YASKAWA

System SLIO

CPU | 015-CEFNR00 | Manual

HB300 | CPU | 015-CEFNR00 | en | 22-30

SPEED7 CPU 015N

YASKAWA Europe GmbH
Hauptstraße 185
65760 Eschborn
Germany
Tel.: +49 6196 569-300
Fax: +49 6196 569-398
Email: info@yaskawa.eu.com
Internet: www.yaskawa.eu.com

015-CEFNR00_000_CPU 015N,10,EN -

Table of contents

1 General

1.1 Copyright © YASKAWA Europe GmbH

Tel.: +49 6196 569 300
Fax.: +49 6196 569 398
Email: info@yaskawa.eu.com
Internet: www.yaskawa.eu.com

EC conformity declaration

Hereby, YASKAWA Europe GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions. Conformity is indicated by the CE marking affixed to the product.

Conformity Information

For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local representative of YASKAWA Europe GmbH.

Trademarks

SLIO and SPEED7 are registered trademarks of YASKAWA Europe GmbH.

SIMATIC, STEP, SINEC, TIA Portal, S7-300 and S7-400 are registered trademarks of Siemens AG.

PROFINET and PROFIBUS are registered trademarks of PROFIBUS and PROFINET International (PI).

EtherCAT is a registered trademark of Beckhoff Automation GmbH.

All Microsoft Windows, Office and Server products mentioned are registered trademarks of Microsoft Inc., USA.

All other trademarks, logos and service or product marks specified herein are owned by their respective companies.

General terms of use

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. We cannot guarantee that the information is free of errors, and we reserve the right to change the information at any time. There is no obligation to inform the customer about any changes. The customer is requested to actively keep his documents up to date. The customer is always responsible for the deployment of the products with the associated documentation, taking into account the applicable directives and standards.
This documentation describes all hardware and software units and functions known today. It is possible that units are described that do not exist at the customer. The exact scope of delivery is described in the respective purchase contract.

Document support

Contact your local representative of YASKAWA Europe GmbH if you have errors or questions regarding the content of this document. You can reach YASKAWA Europe GmbH via the following contact:

Email: Documentation.HER@yaskawa.eu.com

Technical support

Contact your local representative of YASKAWA Europe GmbH if you encounter problems or have questions regarding the product. If such a location is not available, you can reach the Yaskawa customer service via the following contact:

YASKAWA Europe GmbH,
European Headquarters, Hauptstraße 185, 65760 Eschborn, Germany
Tel.: +49 6196 569 500 (hotline)
Email: support@yaskawa.eu.com

1.2 About this manual

Objective and contents

This manual describes the CPU 015-CEFNR00 of the System SLIO.

- It describes the structure, configuration and application.
- The manual is written for users with basic knowledge of automation technology.
- The manual consists of chapters. Each chapter describes a completed topic.
- The following guides are available in the manual:
  – An overall table of contents at the beginning of the manual.
  – References with pages numbers.

Validity of the documentation

Product	Order number	as of version:		
CPU 015N	015-CEFNR00	CPU-HW: 03	CPU-FW: V3.0.5	CP-FW: V3.4.3

Icons Headings

Important passages in the text are highlighted by following icons and headings:

DANGER!
Immediate or likely danger. Personal injury is possible.

CAUTION!
Damages to property is likely if these warnings are not heeded.

Supplementary information and useful tips.

1.3 Safety information

Applications conforming with specifications

The system is constructed and produced for:

- communication and process control
- general control and automation tasks
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle

DANGER!
This device is not certified for applications in

– in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

CAUTION!
The following conditions must be met before using or commissioning the components described in this manual:

– Hardware modifications to the process control system should only be carried out when the system has been disconnected from power!
– Installation and hardware modifications only by properly trained personnel.
– The national rules and regulations of the respective country must be satisfied (installation, safety, EMC ...)

Disposal

National rules and regulations apply to the disposal of the unit!

2 Basics and mounting

2.1 Safety notes for the user

DANGER!

Protection against dangerous voltages

– When using System SLIO modules, the user must be protected from touching hazardous voltage.
– You must therefore create an insulation concept for your system that includes safe separation of the potential areas of ELV and hazardous voltage.
– Here, observe the insulation voltages between the potential areas specified for the System SLIO modules and take suitable measures, such as using PELV/SELV power supplies for System SLIO modules.

Handling of electrostatic sensitive modules

The modules are equipped with highly integrated components in MOS technology. These components are highly sensitive to over-voltages that occur, e.g. with electrostatic discharge. The following symbol is used to identify these hazardous modules:

The symbol is located on modules, module racks or on packaging and thus indicates electrostatic sensitive modules. Electrostatic sensitive modules can be destroyed by energies and voltages that are far below the limits of human perception. If a person who is not electrically discharged handles electrostatic sensitive modules, voltages can occur and damage components and thus impair the functionality of the modules or render the modules unusable. Modules damaged in this way are in most cases not immediately recognized as faulty. The error can only appear after a long period of operation. Components damaged by static discharge can show temporary faults when exposed to temperature changes, vibrations or load changes. Only the consistent use of protective devices and responsible observance of the handling rules can effectively prevent malfunctions and failures on electrostatic sensitive modules.

Shipping of modules

Please always use the original packaging for shipping.

Measurement and modification of electrostatic sensitive modules

For measurements on electrostatic sensitive modules the following must be observed:

- Floating measuring instruments must be discharged before use.
- Measuring instruments used must be grounded.

When modifying electrostatic sensitive modules, ensure that a grounded soldering iron is used.

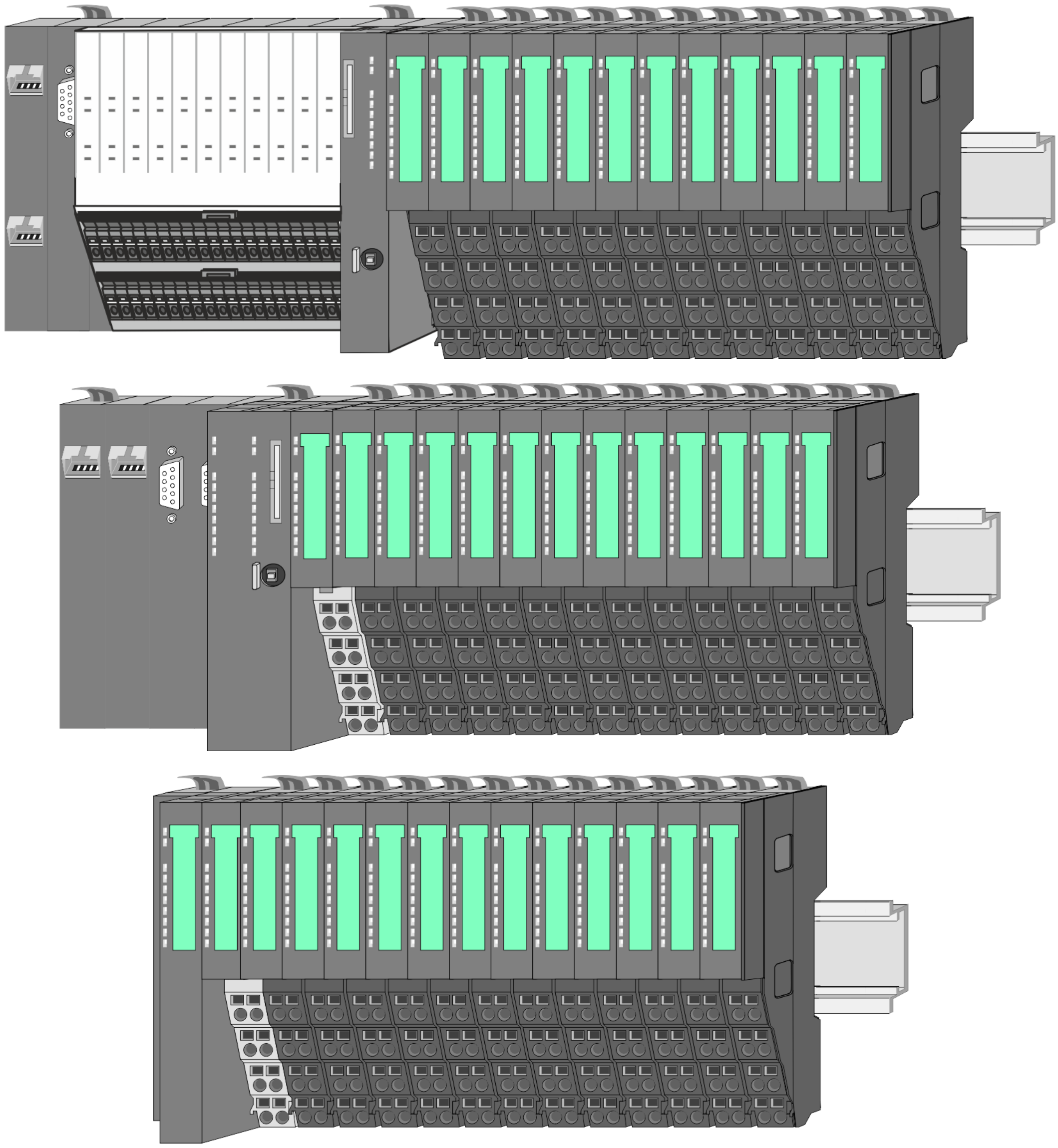
CAUTION!

When working with and on electrostatic sensitive modules, make sure that personnel and equipment are adequately grounded.

2.2 System conception

2.2.1 Overview

The System SLIO is a modular automation system for assembly on a 35mm mounting rail. By means of the periphery modules with 2, 4, 8 and 16 channels this system may properly be adapted matching to your automation tasks. The wiring complexity is low, because the supply of the DC 24V power section supply is integrated to the backplane bus and defective modules may be replaced with standing wiring. By deployment of the power modules in contrasting colors within the system, further isolated areas may be defined for the DC 24V power section supply, respectively the electronic power supply may be extended with 2A.

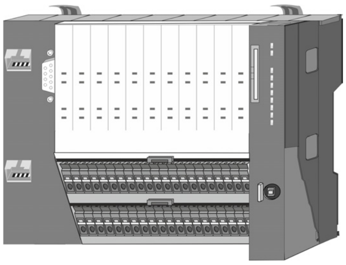


2.2.2 Components

- CPU (head module)
- Bus coupler (head module)
- Line extension
- 8x periphery modules
- 16x periphery modules
- Power modules
- Accessories

CAUTION!
Only Yaskawa modules may be combined. A mixed operation with third-party modules is not allowed!

CPU 01xC



With the CPU 01xC electronic, input/output components and power supply are integrated to one casing. In addition, up to 64 periphery modules of the System SLIO can be connected to the backplane bus. As head module via the integrated power module for power supply CPU electronic and the I/O components are supplied as well as the electronic of the periphery modules, which are connected via backplane bus. To connect the power supply of the I/O components and for DC 24V power section supply of via backplane bus connected periphery modules, the CPU has removable connectors. By installing of up to 64 periphery modules at the backplane bus, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

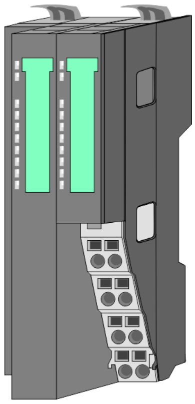
CPU 01x

With this CPU 01x, CPU electronic and power supply are integrated to one casing. As head module, via the integrated power module for power supply, CPU electronic and the electronic of the connected periphery modules are supplied. The DC 24V power section supply for the linked periphery modules is established via a further connection of the power module. By installing of up to 64 periphery modules at the backplane bus, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

CAUTION!
CPU part and power module may not be separated!

Here you may only exchange the electronic module!

Bus coupler

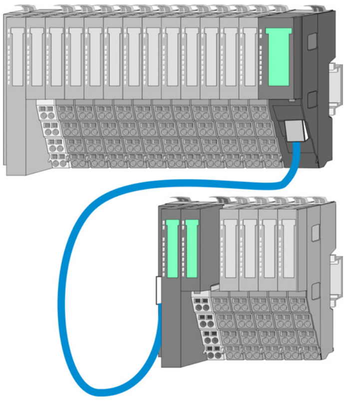


With a bus coupler bus interface and power module is integrated to one casing. With the bus interface you get access to a subordinated bus system. As head module, via the integrated power module for power supply, bus interface and the electronic of the connected periphery modules are supplied. The DC 24V power section supply for the linked periphery modules is established via a further connection of the power module. By installing of up to 64 periphery modules at the bus coupler, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

CAUTION!
Bus interface and power module may not be separated!

Here you may only exchange the electronic module!

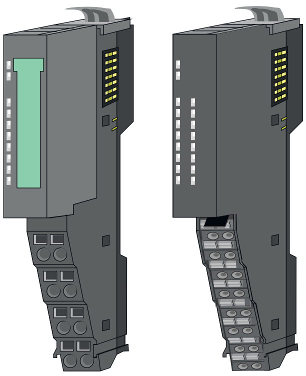
Line extension



In the System SLIO there is the possibility to place up to 64 modules in on line. By means of the line extension you can divide this line into several lines. Here you have to place a line extension master at each end of a line and the subsequent line has to start with a line extension slave. Master and slave are to be connected via a special connecting cable. In this way, you can divide a line on up to 5 lines. For each line extension the maximum number of pluggable modules at the System SLIO bus is decreased by 1. To use the line extension no special configuration is required.

Please note that some modules do not support line extensions due to the system. For more information, see the 'System SLIO - Compatibility List' at www.yaskawa.eu.com

Periphery modules

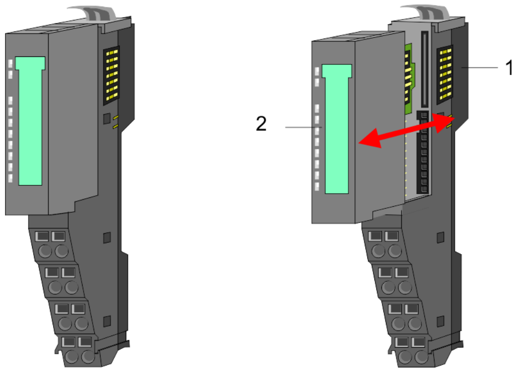


The periphery modules are available in the following 2 versions, whereby of each the electronic part can be replaced with standing wiring:

- 8x periphery module for a maximum of 8 channels.
- 16x periphery module for a maximum of 16 channels.

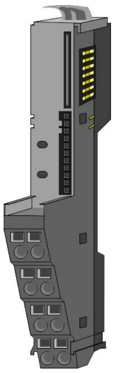
8x periphery modules

Each 8x periphery module consists of a *terminal* and an *electronic module*.



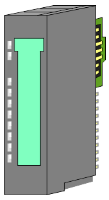
1 Terminal module
2 Electronic module

Terminal module



The *terminal* module serves to carry the electronic module, contains the backplane bus with power supply for the electronic, the DC 24V power section supply and the staircase-shaped terminal for wiring. Additionally the terminal module has a locking system for fixing at a mounting rail. By means of this locking system your system may be assembled outside of your switchgear cabinet to be later mounted there as whole system.

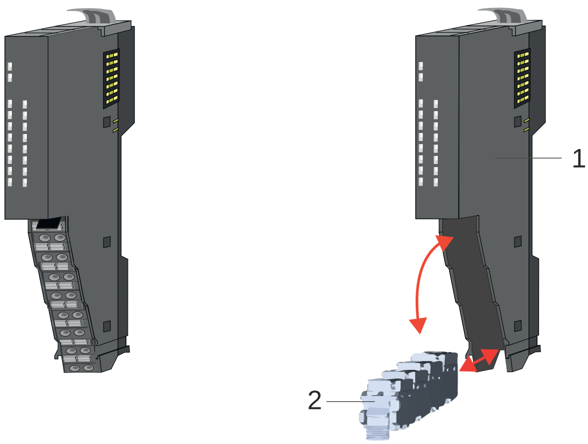
Electronic module



The functionality of a periphery module is defined by the *electronic module*, which is mounted to the terminal module by a sliding mechanism. With an error the defective electronic module may be exchanged for a functional module with standing installation. At the front side there are LEDs for status indication. For simple wiring each module shows corresponding connection information at the front and at the side.

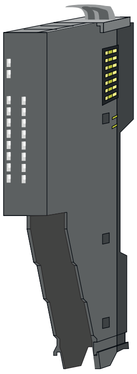
16x periphery modules

Each 16x periphery module consists of an *electronic unit* and a *terminal block*.



1 Electronic unit
2 Terminal block

Electronic unit

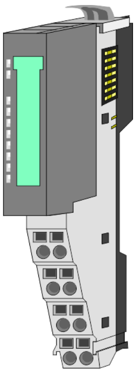


The functionality of a 16x periphery module is defined via the terminal block, which is connected to the *electronic unit* via a secure flap mechanism. In the case of an error you can exchange the defective electronic unit for a functional unit with standing wiring. At the front side there are LEDs for status indication. For easy wiring each electronic unit shows corresponding connection information at the side. The electronic unit provides the slot for the terminal block for the wiring and contains the backplane bus with power supply for the electronic and the connection to the DC 24V power section supply. Additionally the electronic unit has a locking system for fixing it at a mounting rail. By means of this locking system your system may be assembled outside of your switchgear cabinet to be later mounted there as whole system.

Terminal block

The *terminal block* provides the electrical interface for the signalling and supplies lines of the module. When mounting the terminal block, it is attached to the bottom of the electronic unit and turned towards the electronic unit until it clicks into place. With the wiring a "push-in" spring-clip technique is used. This allows a quick and easy connection of your signal and supply lines. The clamping off takes place by means of a screwdriver.

Power module

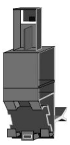


In the System SLIO the power supply is established by power modules. These are either integrated to the head module or may be installed between the periphery modules. Depending on the power module isolated areas of the DC 24V power section supply may be defined respectively the electronic power supply may be extended with 2A. For better recognition the colour of the power modules are contrasting to the periphery modules.

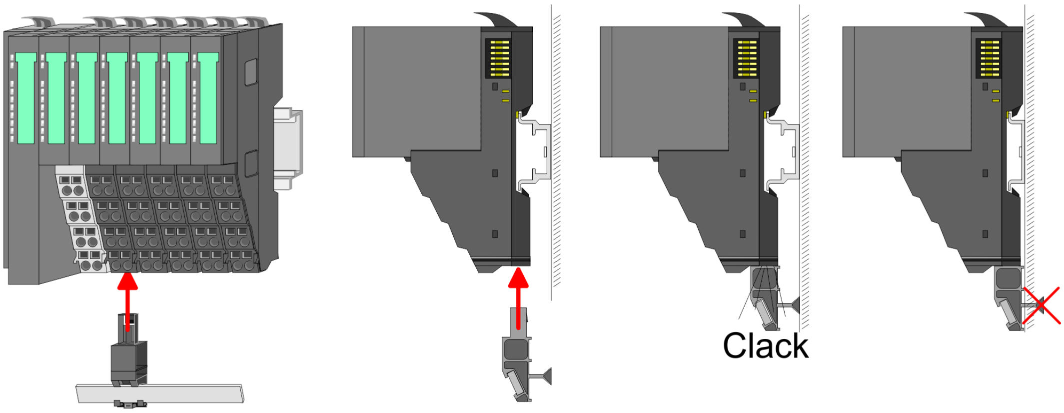
2.2.3 Accessories

Shield bus carrier

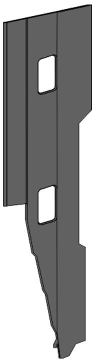
> *Please note that a shield bus carrier cannot be mounted on a 16x periphery module!*



The shield bus carrier (order no.: 000-0AB00) serves to carry the shield bus (10mm x 3mm) to connect cable shields. Shield bus carriers, shield bus and shield fixings are not in the scope of delivery. They are only available as accessories. The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.



Bus cover



With each head module, to protect the backplane bus connectors, there is a mounted bus cover in the scope of delivery. You have to remove the bus cover of the head module before mounting a System SLIO module. For the protection of the backplane bus connector you always have to mount the bus cover at the last module of your system again. The bus cover has the order no. 000-0AA00.

Coding pins

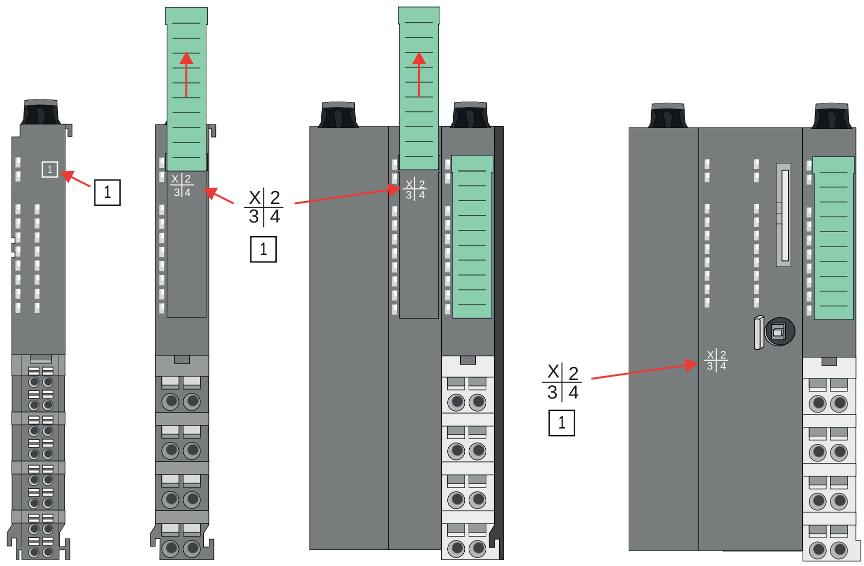
Please note that a coding pin cannot be installed on a 16x periphery module! Here you have to make sure that the associated terminal block is plugged again when the electronics unit is replaced.

There is the possibility to fix the assignment of electronic and terminal module. Here coding pins (order number 000-0AC00) can be used. The coding pin consists of a coding jack and a coding plug. By combining electronic and terminal module with coding pin, the coding jack remains in the electronic module and the coding plug in the terminal module. This ensures that after replacing the electronic module just another electronic module can be plugged with the same encoding.

2.2.4 Hardware revision

Hardware revision on the front

- The hardware revision is printed on every System SLIO module.
- Since a System SLIO 8x periphery module consists of a terminal and electronic module, you will find a hardware revision printed on each of them.
- Authoritative for the hardware revision of a System SLIO module is the hardware revision of the electronic module. This is located under the labeling strip of the corresponding electronic module.
- Depending on the module type, there are the following 2 variants e.g. to indicate hardware revision 1:
  – Current modules have a 1 on the front.
  – With earlier modules, the 1 is marked with *'X'* on a number grid.



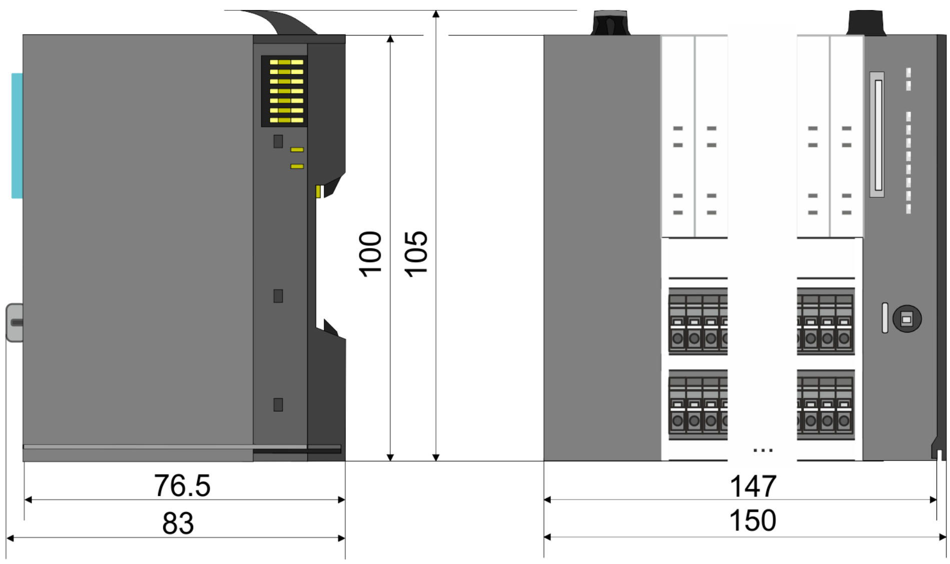
Hardware revision via web server

On the CPUs and some bus couplers, you can check the hardware revision *'HW Revision'* via the integrated web server.

2.3 Dimensions

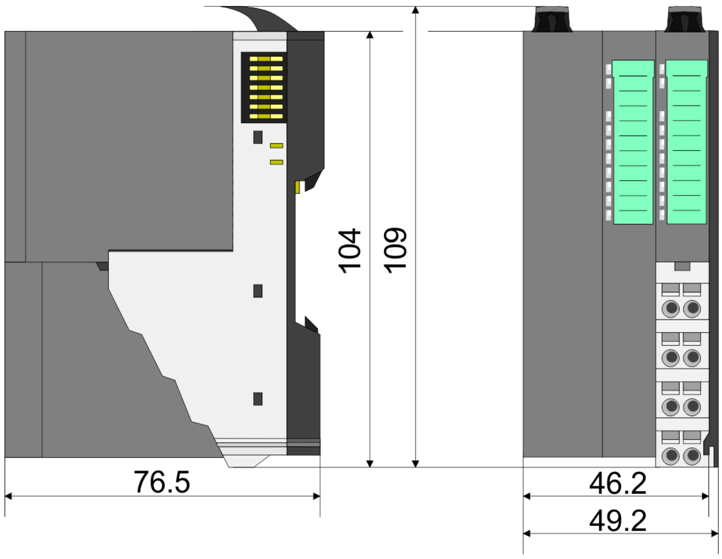
CPU 01xC

All dimensions are in mm.

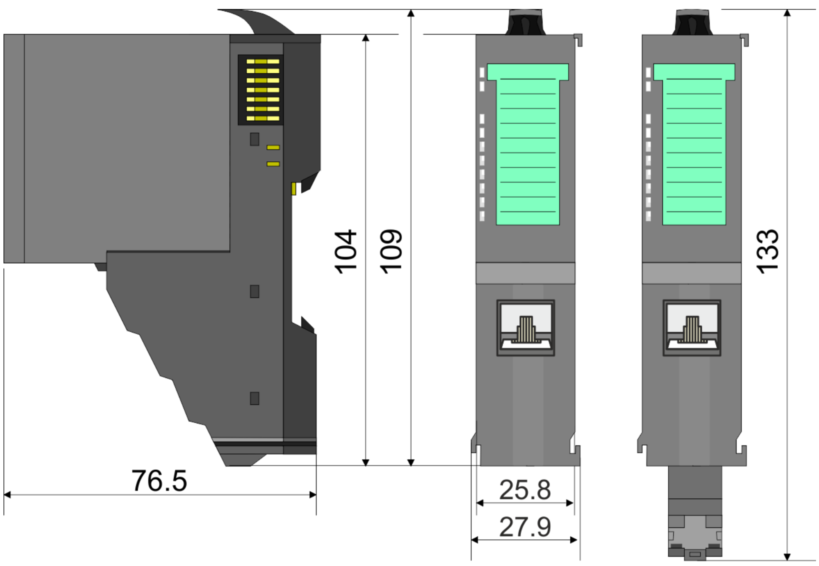


CPU 01x

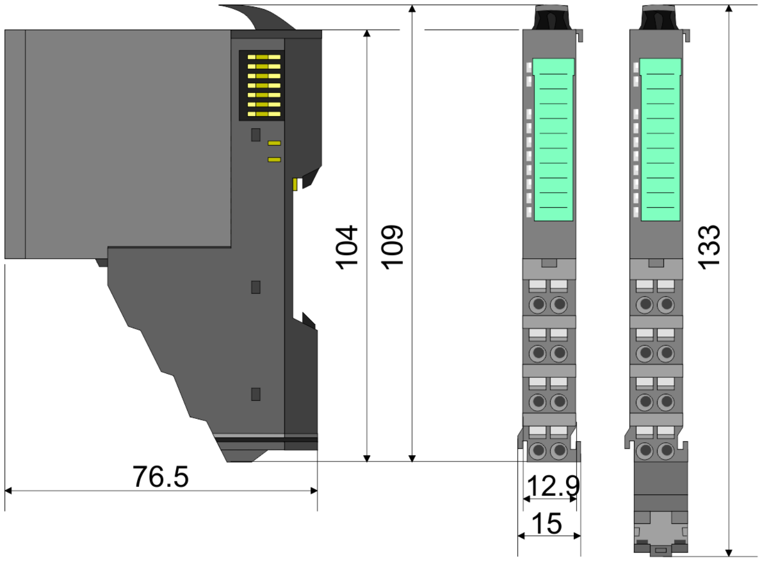
Bus coupler and line extension slave



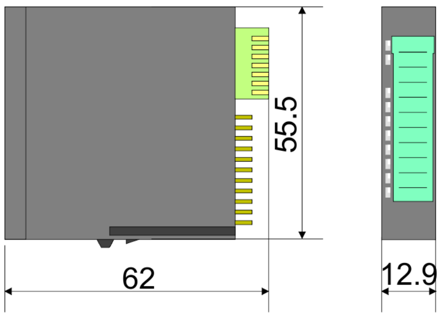
Line extension master



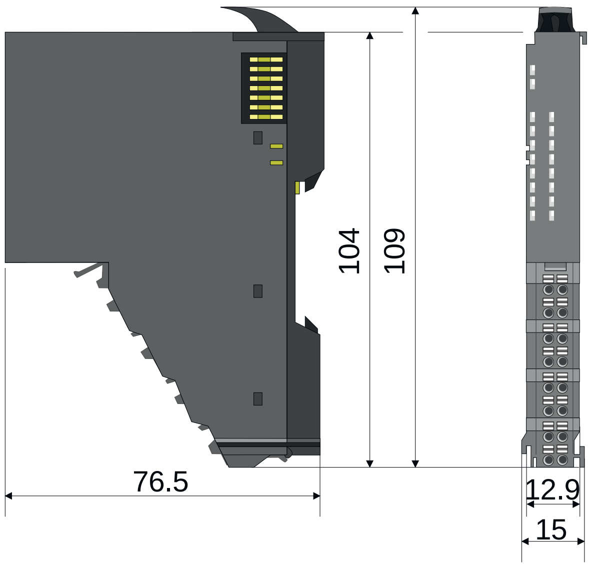
8x periphery module



Electronic module



16x periphery module



2.4 Mounting

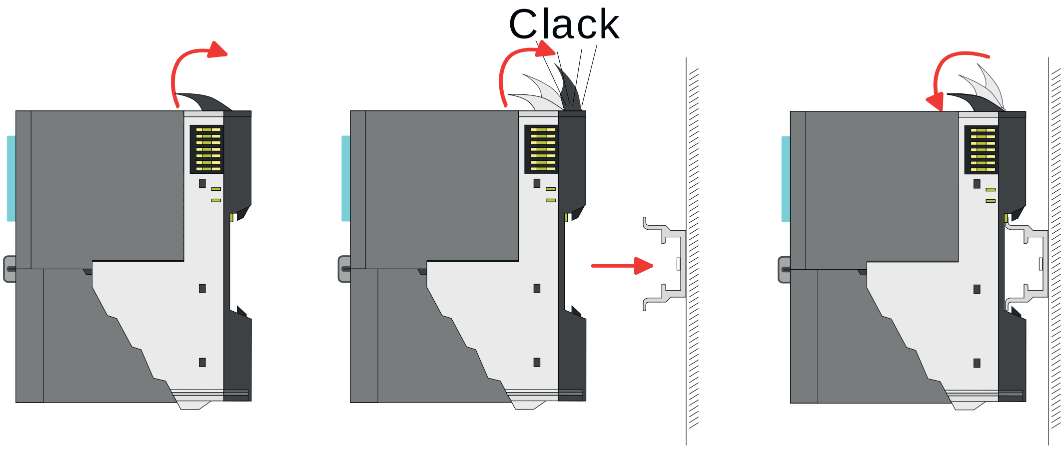
CAUTION!

Requirements for UL compliance use

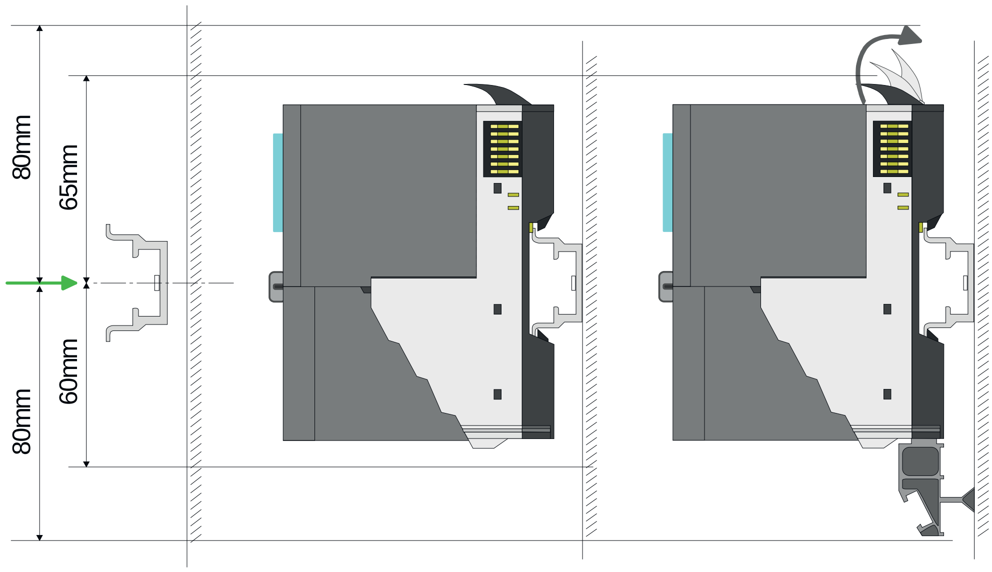
- Use for power supply exclusively SELV/PELV power supplies.
- The System SLIO must be installed and operated in a housing according to IEC 61010-1 9.3.2 c).

2.4.1 Mounting CPU 01x

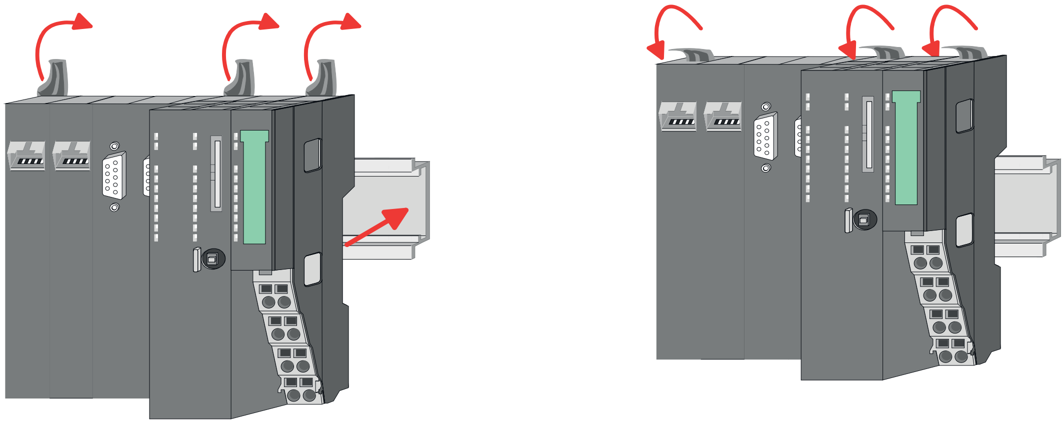
There are locking lever at the top side of the CPU. For mounting and demounting these locking lever are to be turned upwards until these engage. Place the CPU at the mounting rail. The CPU is fixed to the mounting rail by pushing downward the locking levers. The CPU is directly mounted at a mounting rail. Up to 64 modules may be mounted. The electronic and power section supply are connected via the backplane bus. Please consider here that the sum current of the electronic power supply does not exceed the maximum value of 3A. By means of the power module 007-1AB10 the current of the electronic power supply may be expanded accordingly.



Proceeding



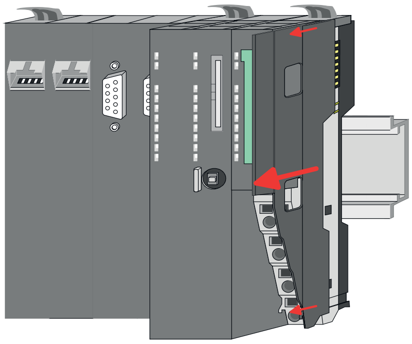
1. Mount the mounting rail! Please consider that a clearance from the middle of the mounting rail of at least 80mm above and 60mm below, respectively 80mm by deployment of shield bus carriers, exist.



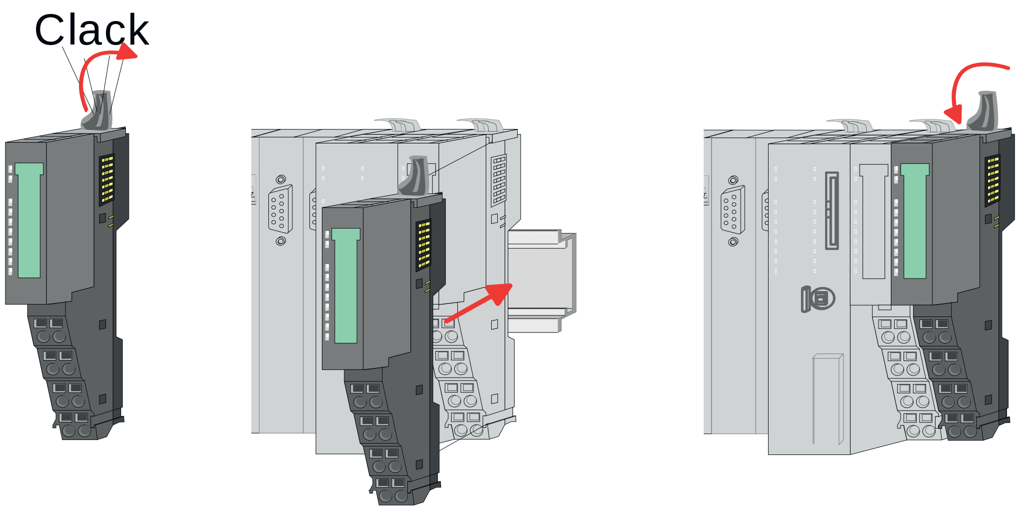
2. Turn the locking lever upwards, place the CPU at the mounting rail and turn the lever downward.

Mounting periphery modules

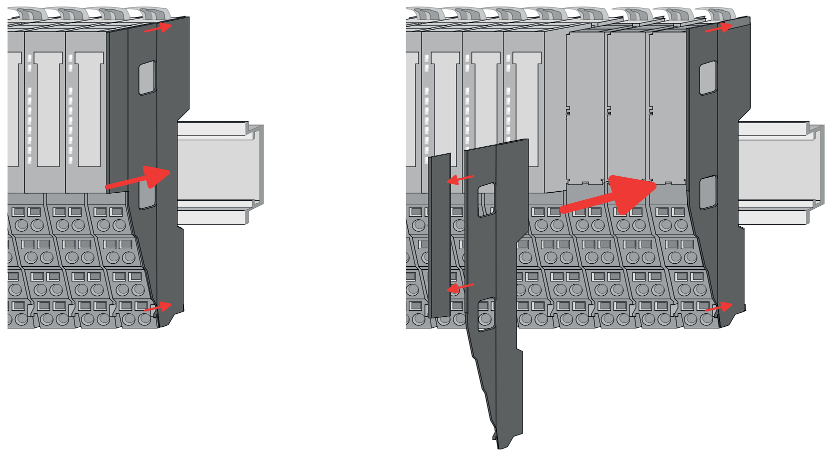
The procedure is identical for 8x and 16x periphery modules.



1. Before mounting the periphery modules you have to remove the bus cover at the right side of the CPU by pulling it forward. Keep the cover for later mounting.



2. Mount the periphery modules you want.



3. After mounting the whole system, to protect the backplane bus connectors at the last module you have to mount the bus cover, now. If the last module is a clamp module, for adaptation the upper part of the bus cover is to be removed.

2.5 Wiring

> **CAUTION!**
> **Consider temperature for external cables!**
> Cables may experience temperature increase due to system heat dissipation. Thus the cabling specification must be chosen 5°C above ambient temperature!

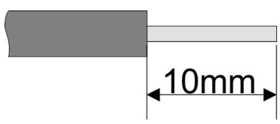
> **CAUTION!**
> **Separate insulation areas!**
> The system is specified for SELV/PELV environment. Devices, which are attached to the system must meet theses specifications. Installation and cable routing other than SELV/PELV specification must be separated from the system’s equipment!

2.5.1 Wiring CPU 01x

Terminal module terminals

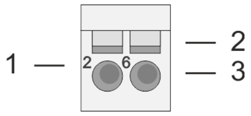
The System SLIO CPUs have a power module integrated. Terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines.

Data

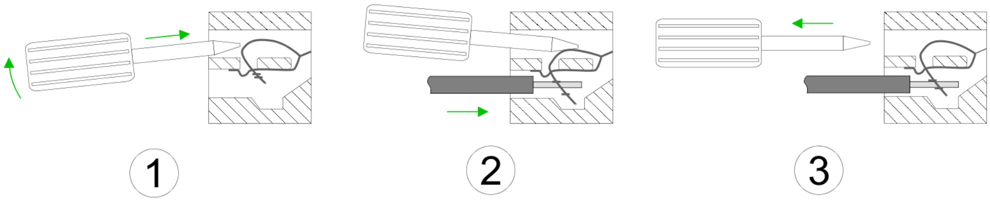


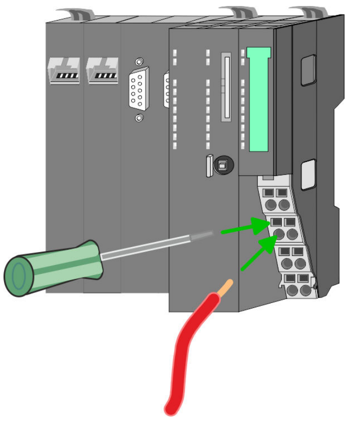
U_{max}	30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm^2 (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



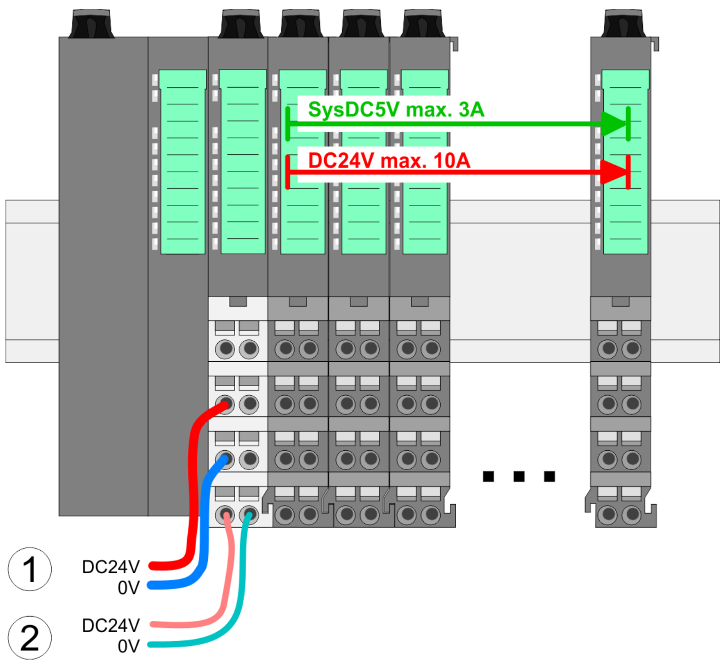
1 Pin number at the terminal module
2 Opening for screwdriver
3 Connection hole for wire





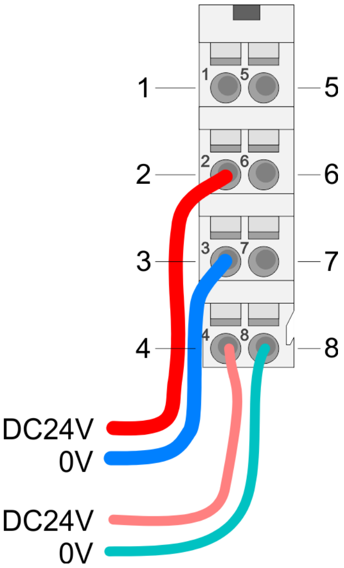
1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can connect wires with a cross section of 0.08mm² up to 1.5mm².
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

Standard wiring



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area

PM - Power module



For wires with a core cross-section of 0.08mm² up to 1.5mm².

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

CAUTION!
Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!

The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!

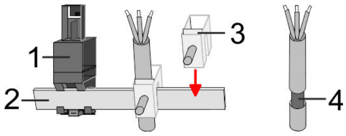
Fusing

- The power section supply is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected with a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
- It is recommended to externally protect the electronic power supply for bus coupler and I/O area with a 2A fuse (fast) respectively by a line circuit breaker 2A characteristics Z.
- The electronic power supply for the I/O area of the power module 007-1AB10 should also be externally protected with a 1A fuse (fast) respectively by a line circuit breaker 1A characteristics Z.

State of the electronic power supply via LEDs

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A. With a sum current greater than 3A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

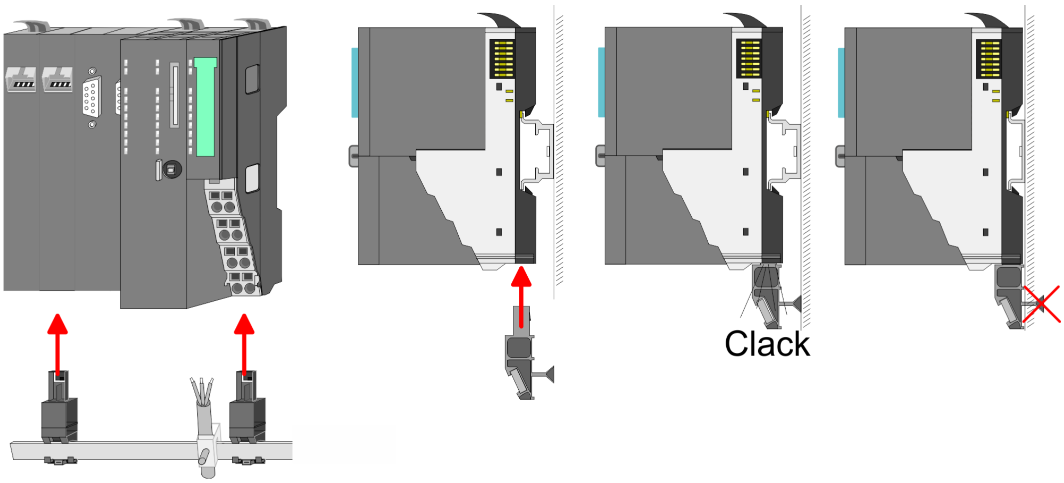
Shield attachment



1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
2. Put your shield bus into the shield bus carrier.



3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

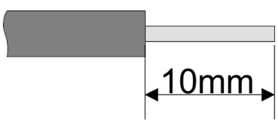
2.5.2 Wiring 8x periphery modules

Terminal module terminals

CAUTION!
Do not connect hazardous voltages!
If this is not explicitly stated in the corresponding module description, hazardous voltages are not allowed to be connected to the corresponding terminal module!

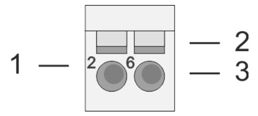
- With wiring the terminal modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

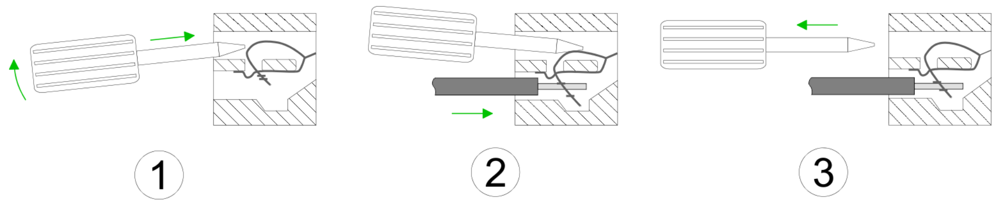


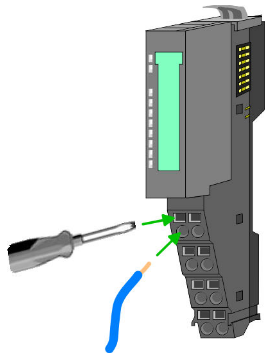
U_{max}	240V AC / 30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm² (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



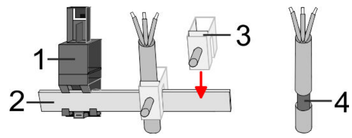
1 Pin number at the connector
2 Opening for screwdriver
3 Connection hole for wire





1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of $0.08mm^2$ up to $1.5mm^2$
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

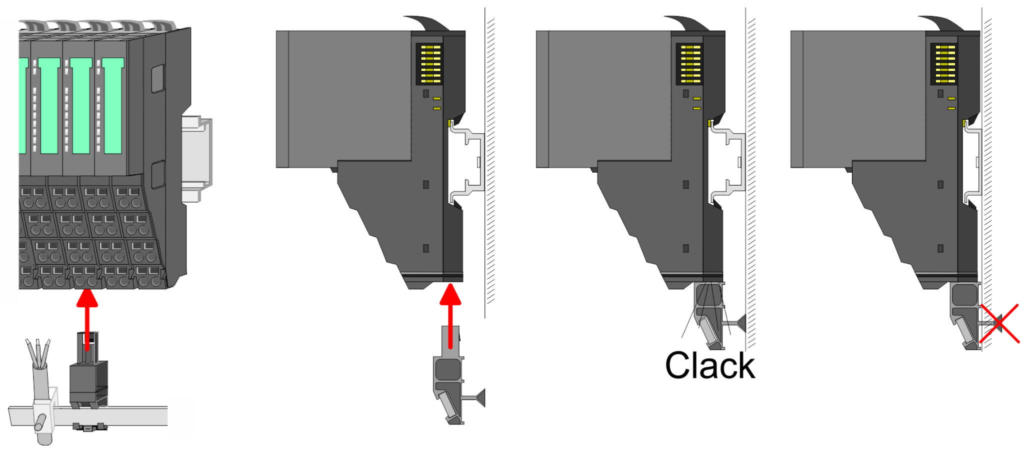
Shield attachment



1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO 8x periphery module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
2. Put your shield bus into the shield bus carrier.



3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

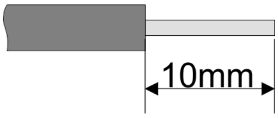
2.5.3 Wiring 16x periphery modules

Terminal block connectors

> **CAUTION!**
> **Do not connect hazardous voltages!**
> If this is not explicitly stated in the corresponding module description, hazardous voltages are not allowed to be connected to the corresponding terminal block!

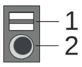
- The 16x periphery module has a removable terminal block for wiring.
- With the wiring of the terminal block a "push-in" spring-clip technique is used. This allows a quick and easy connection of your signal and supply lines.
- The clamping off takes place by means of a screwdriver.
- Please use copper wire only!

Data



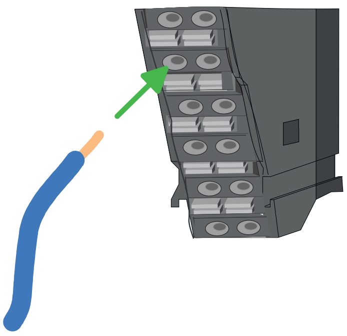
U_{max}	30V DC
I_{max}	10A
Cross section solid wire	0.25 ... 0.75mm^2
Cross section with ferrule	0.14 ... 0.75mm^2
Wire type	CU
AWG	24 ... 16
Stripping length	10mm

Wiring procedure



1 Release area
2 Connection hole for wire

Insert wire

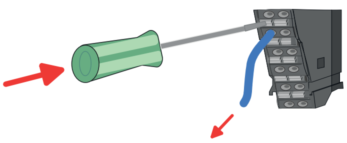


The wiring happens without a tool.

1. Determine according to the casing labelling the connection position.
2. Insert through the round connection hole of the according contact your prepared wire until it stops, so that it is fixed.

   ⇨ By pushing the contact spring opens, thus ensuring the necessary contact pressure.

Remove wire



The wire is to be removed by means of a screwdriver with 2.5mm blade width.

1. Press with your screwdriver vertically at the release button.

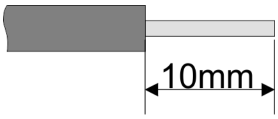
   ⇨ The contact spring releases the wire.
2. Pull the wire from the round hole.

2.5.4 Wiring power modules

Terminal module terminals

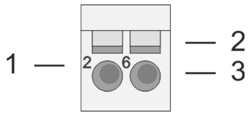
Power modules are either integrated to the head module or may be installed between the periphery modules. With power modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

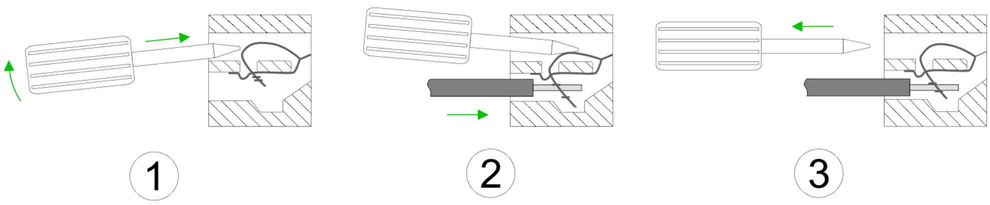


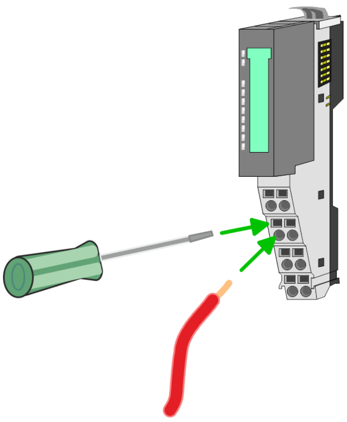
U_{max}	30V DC
I_{max}	10A
Cross section	0.08 ... 1.5mm² (AWG 28 ... 16)
Stripping length	10mm

Wiring procedure



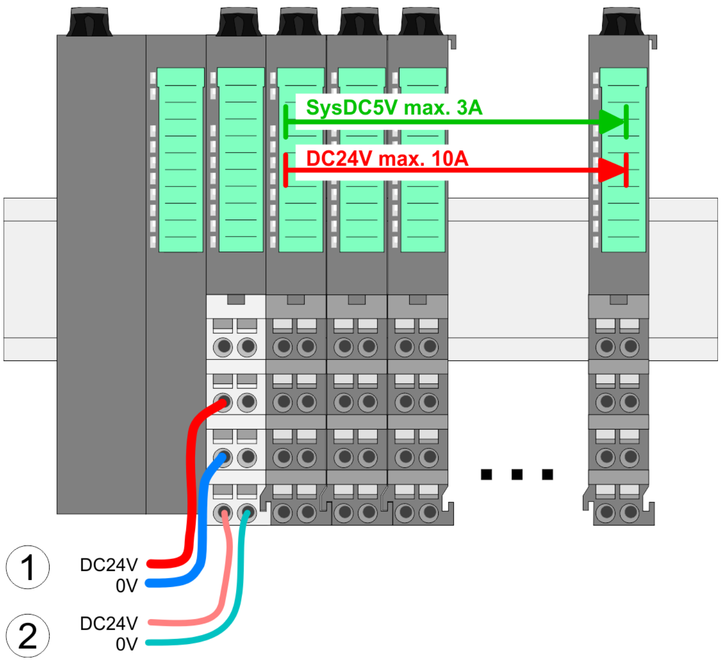
1 Pin number at the connector
2 Opening for screwdriver
3 Connection hole for wire





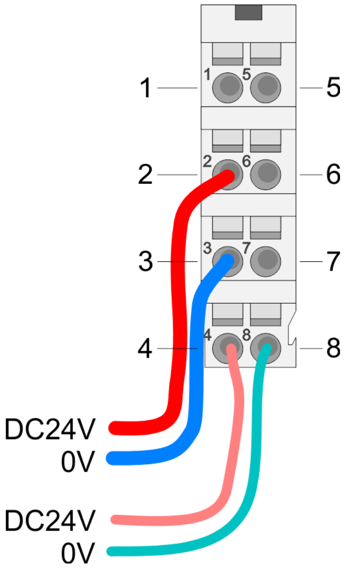
1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm² up to 1.5mm²
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

Standard wiring



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area

PM - Power module



For wires with a core cross-section of 0.08mm² up to 1.5mm².

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

CAUTION!
Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!

The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!

Fusing

- The power section supply is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected with a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
- It is recommended to externally protect the electronic power supply for head modules and I/O area with a 2A fuse (fast) respectively by a line circuit breaker 2A characteristics Z.
- The electronic power supply for the I/O area of the power module 007-1AB10 should also be externally protected with a 1A fuse (fast) respectively by a line circuit breaker 1A characteristics Z.

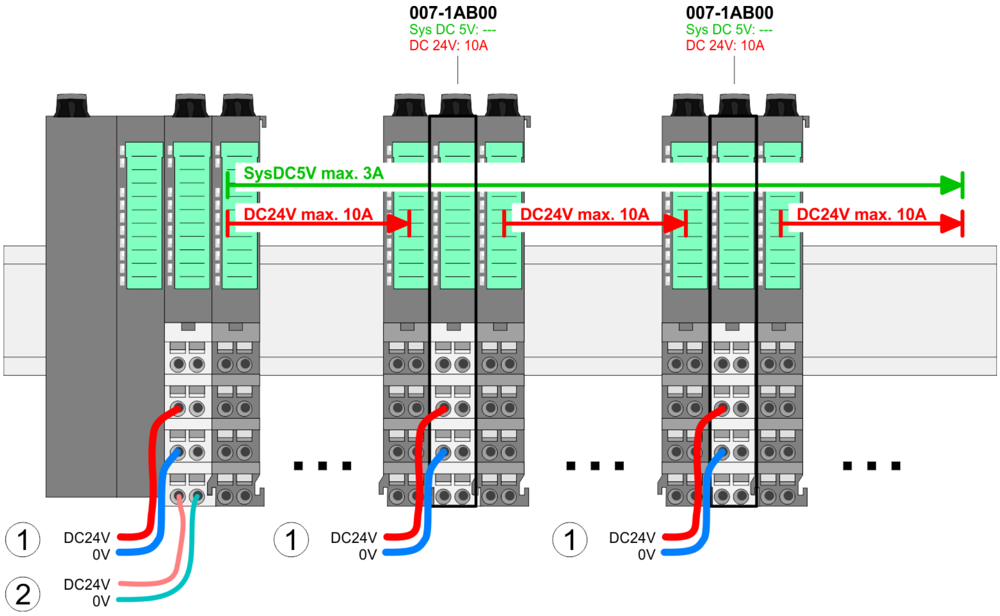
State of the electronic power supply via LEDs

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A. With a sum current greater than 3A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

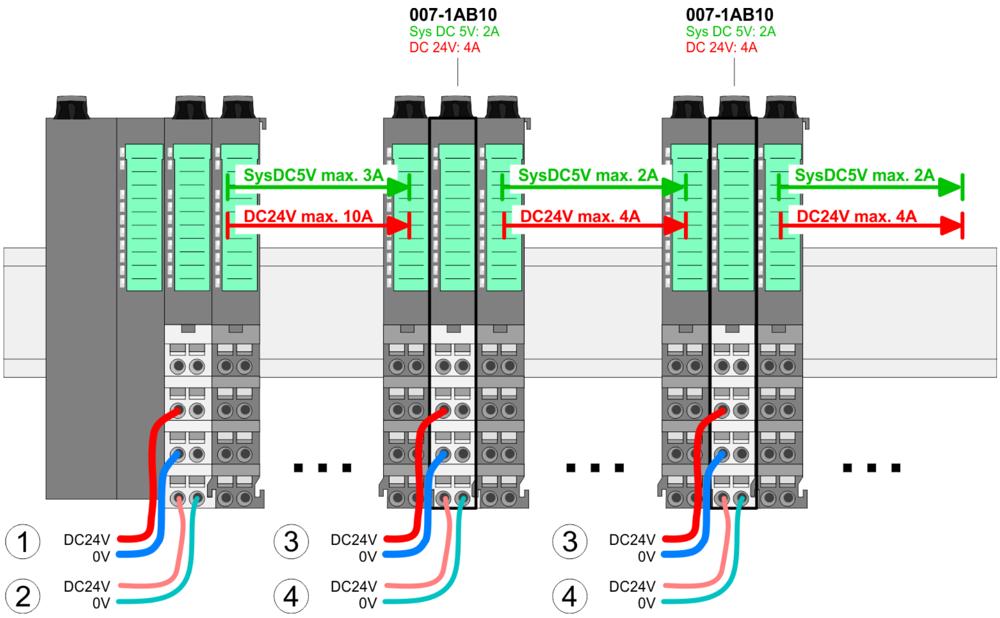
Deployment of the power modules

- If the 10A for the power section supply is no longer sufficient, you may use the power module with the order number 007-1AB00. So you have also the possibility to define isolated groups.
- The power module with the order number 007-1AB10 is to be used if the 3A for the electronic power supply at the backplane bus is no longer sufficient. Additionally you get an isolated group for the DC 24V power section supply with max. 4A.
- By placing the power module 007-1AB10 at the following backplane bus modules may be placed with a sum current of max. 2A. Afterwards a power module is to be placed again. To secure the power supply, the power modules may be mixed used.

Power module 007-1AB00

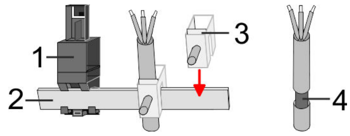


Power module 007-1AB10



(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area
(3) DC 24V for power section supply I/O area (max. 4A)
(4) DC 24V for electronic power supply I/O area

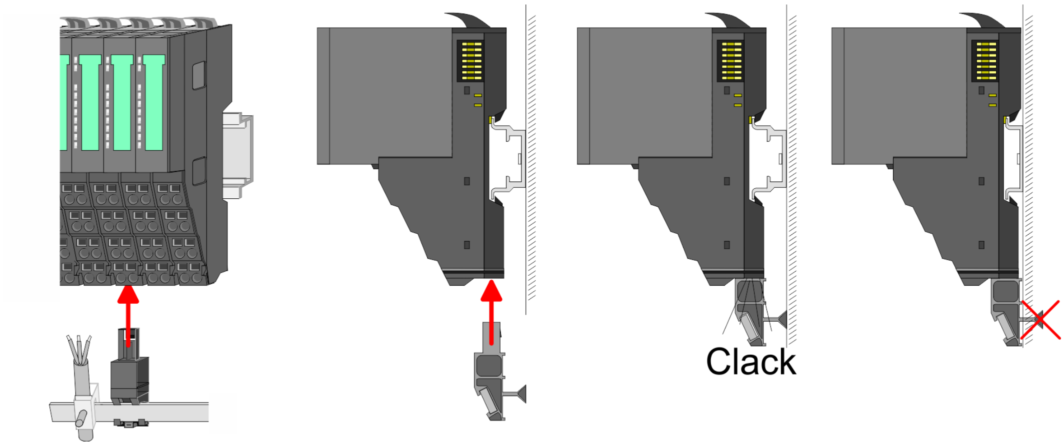
Shield attachment



1 Shield bus carrier
2 Shield bus (10mm x 3mm)
3 Shield clamp
4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO 8x periphery module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
2. Put your shield bus into the shield bus carrier.

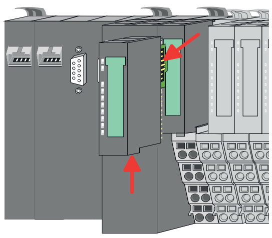


3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

2.6 Demounting

2.6.1 Demounting CPU 01x

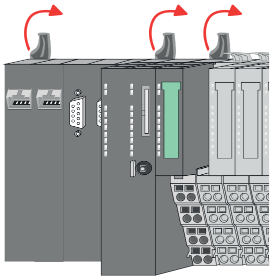
Proceeding



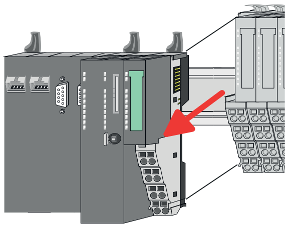
CAUTION!
CPU part and power module may not be separated! Here you may only exchange the electronic module!

1. Power-off your system.
2. Remove if exists the wiring of the CPU.
3. *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

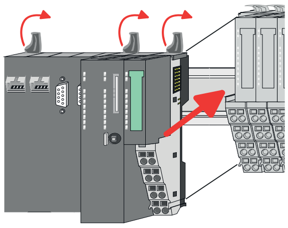
   Press the unlocking lever at the lower side of the just mounted right module near the CPU and pull it forward.



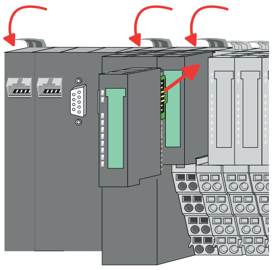
4. Turn all the locking lever of the CPU to be exchanged upwards.



5. Pull the CPU forward.
6. For mounting turn all the locking lever of the CPU to be mounted upwards.



7. To mount the CPU put it to the left periphery module and push it, guided by the stripes, to the mounting rail.
8. Turn all the locking lever downward, again.



9. Plug again the electronic module, which you have removed before. For installation plug the electronic module guided by the strips at the lower side until this engages to the terminal module.
10. Wire your CPU.

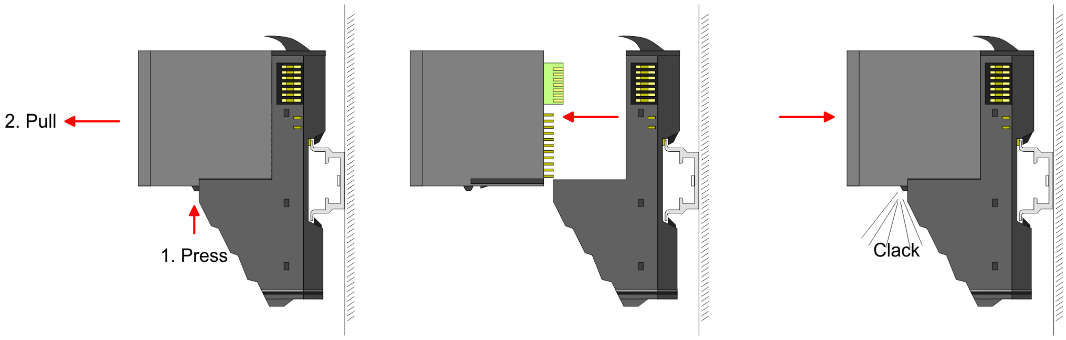
⇨ Now you can bring your system back into operation.

2.6.2 Demounting 8x periphery modules

Proceeding

Exchange of an electronic module

1. Power-off your system.



2. For the exchange of a electronic module, the electronic module may be pulled forward after pressing the unlocking lever at the lower side of the module.
3. For installation plug the new electronic module guided by the strips at the lower side until this engages to the terminal module.

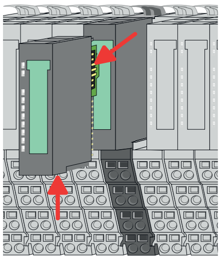
   ⇨ Now you can bring your system back into operation.

> ***Easy Maintenance***
> *'Easy Maintenance' means the support for adding and removing electronic modules during operation without having to restart the system. If this is supported by your head module, you will find more detailed information on this in the "Deployment" chapter.* *➪ Chap. 2.6.4 'Easy Maintenance' page 41*

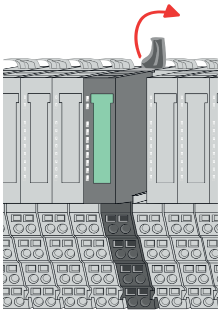
Exchange of a periphery module

1. Power-off your system.
2. Remove if exists the wiring of the module.
3. > *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

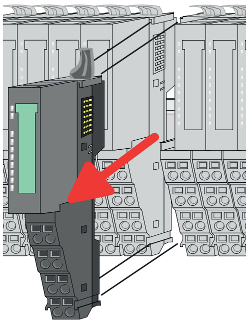
   Press the unlocking lever at the lower side of the just mounted right module and pull it forward.



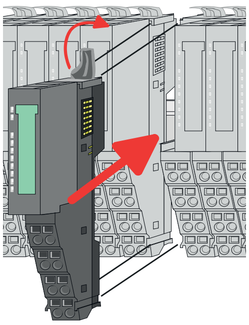
4. Turn the locking lever of the module to be exchanged upwards.



5. Pull the module.
6. For mounting turn the locking lever of the module to be mounted upwards.

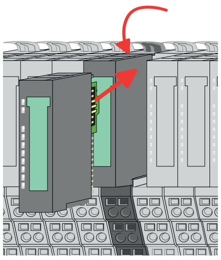


7. To mount the module put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.
8. Turn the locking lever downward, again.

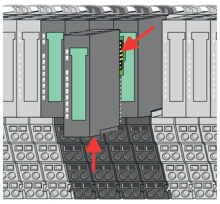


9. Plug again the electronic module, which you have removed before.
10. Wire your module.

   ⇨ Now you can bring your system back into operation.

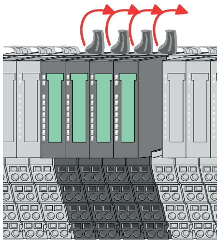


Exchange of a module group

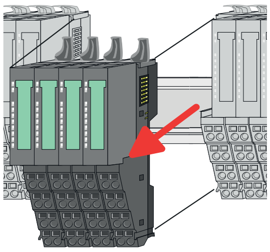


1. Power-off your system.
2. Remove if exists the wiring of the module group.
3. *For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

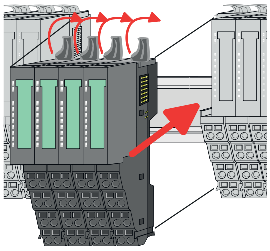
   Press the unlocking lever at the lower side of the just mounted right module near the module group and pull it forward.



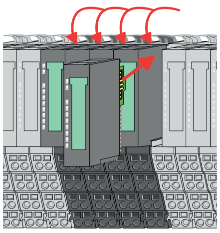
4. Turn all the locking lever of the module group to be exchanged upwards.



5. Pull the module group forward.
6. For mounting turn all the locking lever of the module group to be mounted upwards.



7. To mount the module group put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.
8. Turn all the locking lever downward, again.



9. Plug again the electronic module, which you have removed before.
10. Wire your module group.

   ⇨ Now you can bring your system back into operation.

2.6.3 Demounting 16x periphery modules

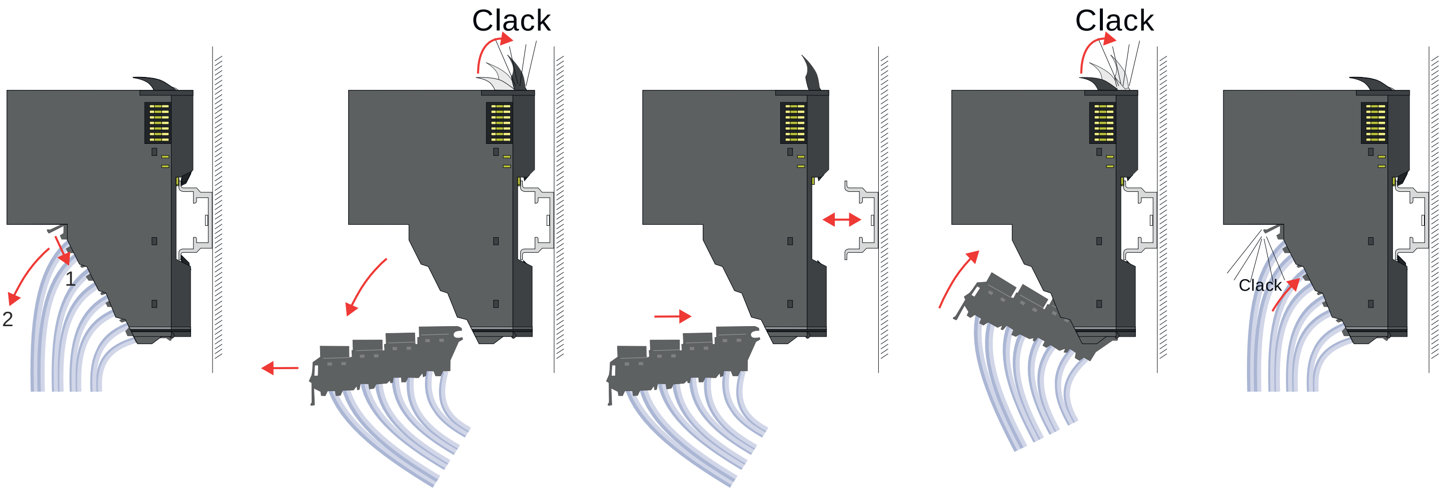
Proceeding

Exchange of an electronic unit

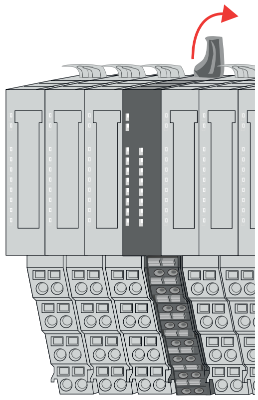
1. Power-off your system.
2. To replace an electronic unit, you can push down and pull off the terminal block after releasing the lock.

   To mount the terminal block, place it horizontally on the lower side of the electronic unit and push it towards the electronic unit until it clicks into place.

   ⇨ Now you can bring your system back into operation.

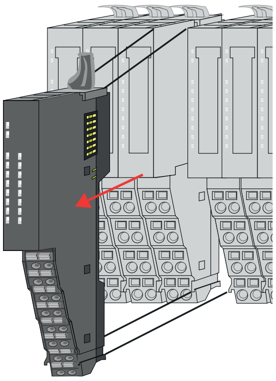


Exchange of a 16x periphery module

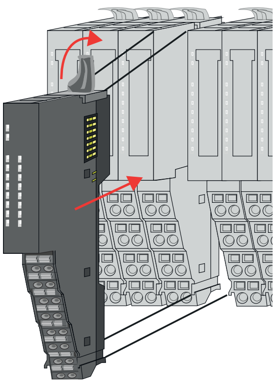


1. Power-off your system.
2. Remove if exists the wiring of the module respectively the wired terminal block.
3. *In contrast to 8x periphery modules, you can directly demount and mount 16x periphery modules.*

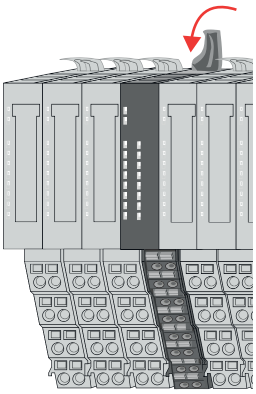
   Turn the locking lever of the module to be exchanged upwards.



4. Pull the module.
5. For mounting turn the locking lever of the module to be mounted upwards.



6. To mount the module put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.



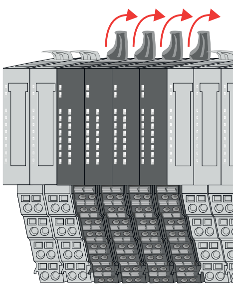
7. Turn the locking lever downward, again.
8. Wire your module respectively plug the wired terminal block again.
   ⇨ Now you can bring your system back into operation.

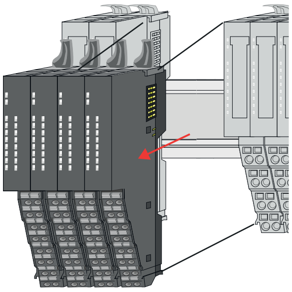
Exchange of a module group

1. Power-off your system.
2. Remove if exists the wiring of the module group respectively the wired terminal blocks.
3.

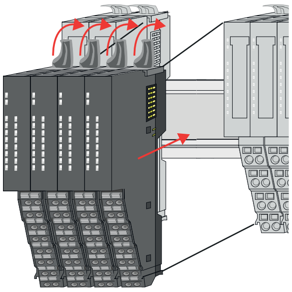
> *In contrast to 8x periphery modules, you can directly demount and mount 16x periphery modules.*

   Turn all the locking lever of the module group to be exchanged upwards.

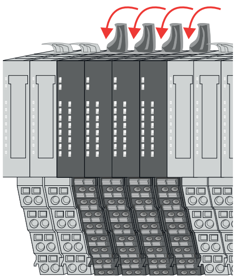




4. Pull the module group forward.
5. For mounting turn all the locking lever of the module group to be mounted upwards.



6. To mount the module group put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.



7. Turn all the locking lever downward, again.
8. Wire your module group respectively plug the wired terminal blocks again.
   ⇨ Now you can bring your system back into operation.

2.6.4 Easy Maintenance

Overview

Easy Maintenance means the support for adding and removing an electronic module during operation without having to restart the system. Here the following behavior is shown by the example of a CPU:

- Electronic module is removed
  - The CPU detects a module failure on the backplane bus.
  - Diagnostic message *'System SLIO bus failure'* (0x39D0) is triggered.
  - OB 86 is called. If this is not available, the CPU switches to STOP otherwise it remains in RUN.
  - The SF LED of the CPU lights up.
  - The I/O data of all modules become invalid.
- Identical electronic module is plugged
  - The CPU detects the module return on the backplane bus.
  - The SF-LED of the CPU gets off.
  - All RUN LEDs on the modules get on and the MF LEDs get off.
  - Diagnostic message *'System SLIO bus recovery'* (0x38D0) is triggered.
  - OB 86 is called. If this is not available, the CPU switches to STOP otherwise it remains in RUN.
  - The I/O data of all modules become valid again.
- Wrong electronic module is plugged
  - The CPU detects the wrong module.
  - Diagnostic message *'System SLIO bus recovery, but expected configuration does not match actual configuration'* (0x38D1) is triggered.
  - The SF LED of the CPU remains on.
  - The MF LED of the wrong module flashes.
  - OB 86 is called. If this is not available, the CPU switches to STOP otherwise it remains in RUN.
  - With the exception of the wrong module, the I/O data of all modules become valid again.

CAUTION!
Please note that only electronic modules may be exchanged during operation! Replacing an 8x or 16x periphery module during operation can damage the module and the system!

Please note that the CPU switches to STOP, if there is no OB 86 configured when adding or removing System SLIO modules!

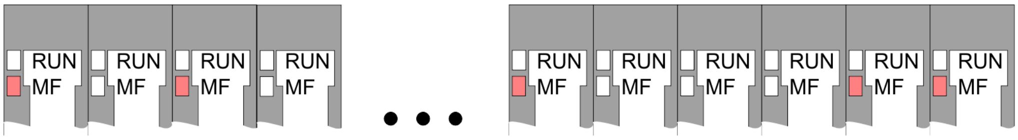
2.7 Trouble shooting - LEDs

General

Each module has the LEDs RUN and MF on its front side. Errors or incorrect modules may be located by means of these LEDs.

In the following illustrations flashing LEDs are marked by ☼.

Sum current of the electronic power supply exceeded

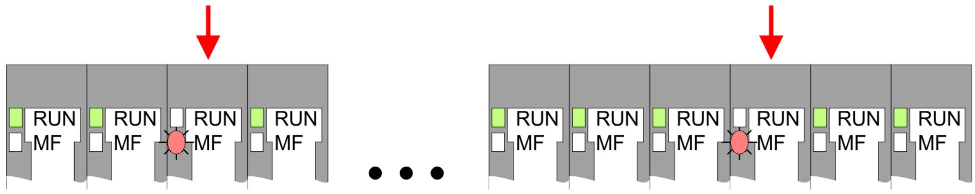


Behaviour: After PowerON the RUN LED of each module is off and the MF LED of each module is sporadically on.

Reason: The maximum current for the electronic power supply is exceeded.

Remedy: As soon as the sum current of the electronic power supply is exceeded, always place the power module 007-1AB10. ↳ *Chap. 2.5.4 'Wiring power modules' page 29*

Error in configuration

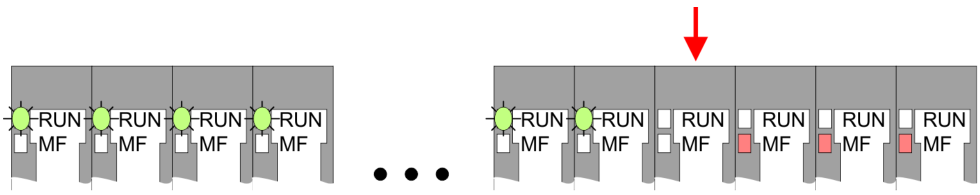


Behaviour: After PowerON the MF LED of one module respectively more modules flashes. The RUN LED remains off.

Reason: At this position a module is placed, which does not correspond to the configured module.

Remedy: Match configuration and hardware structure.

Module failure



Behaviour: After PowerON all of the RUN LEDs up to the defective module are flashing. With all following modules the MF LED is on and the RUN LED is off.

Reason: The module on the right of the flashing modules is defective.

Remedy: Replace the defective module.

2.8 Industrial security and installation guidelines

2.8.1 Industrial security in information technology

Latest version

This chapter can also be found as a guide *'IIndustrial IT Security'* at *www.yaskawa.eu.com*

Hazards

The topic of data security and access protection has become increasingly important in the industrial environment. The increased networking of entire industrial systems to the network levels within the company together with the functions of remote maintenance have all served to increase vulnerability. Hazards can arise from:

- Internal manipulation such as technical errors, operating and program errors and deliberate program or data manipulation.
- External manipulation such as software viruses, worms and Trojans.
- Human carelessness such as password phishing.

Precautions

The most important precautions to prevent manipulation and loss of data security in the industrial environment are:

- Encrypting the data traffic by means of certificates.
- Filtering and inspection of the traffic by means of VPN - "Virtual Private Networks".
- Identification of the user by "Authentication" via save channels.
- Segmenting in protected automation cells, so that only devices in the same group can exchange data.
- Deactivation of unnecessary hardware and software.

Further Information

You can find more information about the measures on the following websites:

- Federal Office for Information Technology *www.bsi.bund.de*
- Cybersecurity & Infrastructure Security Agency *us-cert.cisa.gov*
- VDI / VDE Society for Measurement and Automation Technology *www.vdi.de*

2.8.1.1 Protection of hardware and applications

Precautions

- Do not integrate any components or systems into public networks.
  – Use VPN "Virtual Private Networks" for use in public networks. This allows you to control and filter the data traffic accordingly.
- Always keep your system up-to-date.
  – Always use the latest firmware version for all devices.
  – Update your user software regularly.
- Protect your systems with a firewall.
  – The firewall protects your infrastructure internally and externally.
  – This allows you to segment your network and isolate entire areas.
- Secure access to your plants via user accounts.
  – If possible, use a central user management system.
  – Create a user account for each user for whom authorization is essential.
  – Always keep user accounts up-to-date and deactivate unused user accounts.
- Secure access to your plants via secure passwords.
  – Change the password of a standard login after the first start.
  – Use strong passwords consisting of upper/lower case, numbers and special characters. The use of a password generator or manager is recommended.
  – Change the passwords according to the rules and guidelines that apply to your application.
- Deactivate inactive communication ports respectively protocols.
  – Only the communication ports that are used for communication should be activated.
  – Only the communication protocols that are used for communication should be activated.
- Consider possible defence strategies when planning and securing the system.
  – The isolation of components alone is not sufficient for comprehensive protection. An overall concept is to be drawn up here, which also provides defensive measures in the event of a cyber attack.
  – Periodically carry out threat assessments. Among others, a comparison is made here between the protective measures taken and those required.
- Limit the use of external storage media.
  – Via external storage media such as USB memory sticks or SD memory cards, malware can get directly into a system while bypassing a firewall.
  – External storage media or their slots must be protected against unauthorized physical access, e.g. by using a lockable control cabinet.
  – Make sure that only authorized persons have access.
  – When disposing of storage media, make sure that they are safely destroyed.
- Use secure access paths such as HTTPS or VPN for remote access to your plant.
- Enable security-related event logging in accordance with the applicable security policy and legal requirements for data protection.

2.8.1.2 Protection of PC-based software

Precautions

Since PC-based software is used for programming, configuration and monitoring, it can also be used to manipulate entire systems or individual components. Particular caution is required here!

- Use user accounts on your PC systems.
  - If possible, use a central user management system.
  - Create a user account for each user for whom authorization is essential.
  - Always keep user accounts up-to-date and deactivate unused user accounts.
- Protect your PC systems with secure passwords.
  - Change the password of a standard login after the first start.
  - Use strong passwords consisting of upper/lower case, numbers and special characters. The use of a password generator or manager is recommended.
  - Change the passwords according to the rules and guidelines that apply to your application.
- Enable security-related event logging in accordance with the applicable security policy and legal requirements for data protection.
- Protect your PC systems by security software.
  - Install virus scanners on your PC systems to identify viruses, trojans and other malware.
  - Install software that can detect phishing attacks and actively prevent them.
- Always keep your software up-to-date.
  - Update your operating system regularly.
  - Update your software regularly.
- Make regular backups and store the media at a safe place.
- Regularly restart your PC systems. Only boot from storage media that are protected against manipulation.
- Use encryption systems on your storage media.
- Perform security assessments regularly to reduce the risk of manipulation.
- Use only data and software from approved sources.
- Uninstall software which is not used.
- Disable unused services.
- Activate a password-protected screen lock on your PC systems.
- Always lock your PC systems as soon as you leave your PC workstation.
- Do not click any links that come from unknown sources. If necessary ask, e.g. on e-mails.
- Use secure access paths such as HTTPS or VPN for remote access to your PC system.

2.8.2 Installation guidelines

General

The installation guidelines contain information about the interference free deployment of a PLC system. There is the description of the ways, interference may occur in your PLC, how you can make sure the electromagnetic compatibility (EMC), and how you manage the isolation.

What does EMC mean?

Electromagnetic compatibility (EMC) means the ability of an electrical device, to function error free in an electromagnetic environment without being interfered respectively without interfering the environment.

The components are developed for the deployment in industrial environments and meets high demands on the EMC. Nevertheless you should project an EMC planning before installing the components and take conceivable interference causes into account.

Possible interference causes

Electromagnetic interferences may interfere your control via different ways:

- Electromagnetic fields (RF coupling)
- Magnetic fields with power frequency
- Bus system
- Power supply
- Protected earth conductor

Depending on the spreading medium (lead bound or lead free) and the distance to the interference cause, interferences to your control occur by means of different coupling mechanisms.

There are:

- galvanic coupling
- capacitive coupling
- inductive coupling
- radiant coupling

Basic rules for EMC

In the most times it is enough to take care of some elementary rules to guarantee the EMC. Please regard the following basic rules when installing your PLC.

- Take care of a correct area-wide grounding of the inactive metal parts when installing your components.
  - Install a central connection between the ground and the protected earth conductor system.
  - Connect all inactive metal extensive and impedance-low.
  - Please try not to use aluminium parts. Aluminium is easily oxidizing and is therefore less suitable for grounding.
- When cabling, take care of the correct line routing.
  - Organize your cabling in line groups (high voltage, current supply, signal and data lines).
  - Always lay your high voltage lines and signal respectively data lines in separate channels or bundles.
  - Route the signal and data lines as near as possible beside ground areas (e.g. suspension bars, metal rails, tin cabinet).
- Proof the correct fixing of the lead isolation.
  - Data lines must be shielded.
  - Analog lines must be shielded. When transmitting signals with small amplitudes the one sided laying of the isolation may be favourable.
  - Cables for frequency inverters, servo and stepper motors must be shielded.
  - Lay the line isolation extensively on an isolation/protected earth conductor rail directly after the cabinet entry and fix the isolation with cable clamps.
  - Make sure that the isolation/protected earth conductor rail is connected impedance-low with the cabinet.
  - Use metallic or metallised plug cases for isolated data lines.
- In special use cases you should appoint special EMC actions.
  - Consider to wire all inductivities with erase links.
  - Please consider luminescent lamps can influence signal lines.
- Create a homogeneous reference potential and ground all electrical operating supplies when possible.
  - Please take care for the targeted employment of the grounding actions. The grounding of the PLC serves for protection and functionality activity.
  - Connect installation parts and cabinets with your PLC in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If there are potential differences between installation parts and cabinets, lay sufficiently dimensioned potential compensation lines.

Isolation of conductors

Electrical, magnetically and electromagnetic interference fields are weakened by means of an isolation, one talks of absorption. Via the isolation rail, that is connected conductive with the rack, interference currents are shunt via cable isolation to the ground. Here you have to make sure, that the connection to the protected earth conductor is impedance-low, because otherwise the interference currents may appear as interference cause.

When isolating cables you have to regard the following:

- If possible, use only cables with isolation tangle.
- The hiding power of the isolation should be higher than 80%.
- Normally you should always lay the isolation of cables on both sides. Only by means of the both-sided connection of the isolation you achieve high quality interference suppression in the higher frequency area. Only as exception you may also lay the isolation one-sided. Then you only achieve the absorption of the lower frequencies. A one-sided isolation connection may be convenient, if:
  – the conduction of a potential compensating line is not possible.
  – analog signals (some mV respectively µA) are transferred.
  – foil isolations (static isolations) are used.
- With data lines always use metallic or metallised plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.
- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to your PLC and don't lay it on there again!

CAUTION!
Please regard at installation!
At potential differences between the grounding points, there may be a compensation current via the isolation connected at both sides.

Remedy: Potential compensation line

2.9 General data for the System SLIO

Conformity and approval		
Conformity		
CE	2014/35/EU	Low-voltage directive
	2014/30/EU	EMC directive
Approval		
UL	-	Refer to Technical data
Others		
RoHS	2011/65/EU	Restriction of the use of certain hazardous substances in electrical and electronic equipment

Protection of persons and device protection		
Type of protection	-	IP20
Electrical isolation		
to the field bus	-	electrically isolated
to the process level	-	electrically isolated
Insulation resistance	-	-
Insulation voltage to reference earth		
Inputs / outputs	-	AC / DC 50V, test voltage AC 500V
Protective measures	-	against short circuit

Environmental conditions to EN 61131-2		
Climatic		
Storage / transport	EN 60068-2-14	-25…+70°C
Operation		
Horizontal installation hanging	EN 61131-2	0…+60°C
Horizontal installation lying	EN 61131-2	0…+55°C
Vertical installation	EN 61131-2	0…+50°C
Air humidity	EN 60068-2-30	RH1 (without condensation, rel. humidity 10…95%)
Pollution	EN 61131-2	Degree of pollution 2
Installation altitude max.	-	2000m
Mechanical		
Oscillation	EN 60068-2-6	1g, 9Hz ... 150Hz
Shock	EN 60068-2-27	15g, 11ms

Mounting conditions		
Mounting place	-	In the control cabinet
Mounting position	-	Horizontal and vertical

EMC	Standard		Comment
Emitted interference	EN 61000-6-4		Class A (Industrial area)
Noise immunity zone B	EN 61000-6-2		Industrial area
		EN 61000-4-2	ESD 8kV at air discharge (degree of severity 3), 4kV at contact discharge (degree of severity 2)
		EN 61000-4-3	HF field immunity (casing) 80MHz … 1000MHz, 10V/m, 80% AM (1kHz) 1.4GHz ... 2.0GHz, 3V/m, 80% AM (1kHz) 2GHz ... 2.7GHz, 1V/m, 80% AM (1kHz)
		EN 61000-4-6	HF conducted 150kHz … 80MHz, 10V, 80% AM (1kHz)
		EN 61000-4-4	Burst, degree of severity 3
		EN 61000-4-5	Surge, degree of severity 3[1]

1) Due to the high-energetic single pulses with Surge an appropriate external protective circuit with lightning protection elements like conductors for lightning and overvoltage is necessary.

2.9.1 Use in difficult operating conditions

Without additional protective measures, the products must not be used in locations with difficult operating conditions; e.g. due to:

– *dust generation*
– *chemically active substances (corrosive vapors or gases)*
– *strong electric or magnetic fields*

3 Hardware description

3.1 Properties

CPU 015-CEFNR00

- SPEED7 technology integrated
- Programmable via *SPEED7 Studio*, Siemens SIMATIC Manager or TIA Portal
- 256kbyte work memory integrated (128kbyte code, 128kbyte data)
- Work memory expandable up to 512kbyte (256kbyte code, 256kbyte data)
- 512kbyte load memory integrated
- Slot for external storage media (lockable)
- Status LEDs for operating state and diagnostics
- X1/X5: Ethernet PG/OP channel integrated
- X2: PtP(MPI) interface Serial integrated interface for PtP communication with the protocols: ASCII, STX/ETX , USS, 3964(R), MODBUS RTU, master/slave switch able to MPI communication
- X3: MPI(PB) interface: MPI interface with via VSC unlock able field bus functions
- X4: Ethernet interface Ethernet interface with EtherCAT master functionality
- X6: NET CP: Ethernet-interface for TCP/IP communication
- PROFINET I-Device via Ethernet PG/OP channel
- *OPC UA* project via Ethernet PG/OP channel respectively Ethernet CP
- *WebVisu* project via Ethernet PG/OP channel respectively Ethernet CP
- Up to 64 System SLIO modules placeable
- I/O address area digital/analog 2048byte
- 512 timer/counter, 8192 flag byte

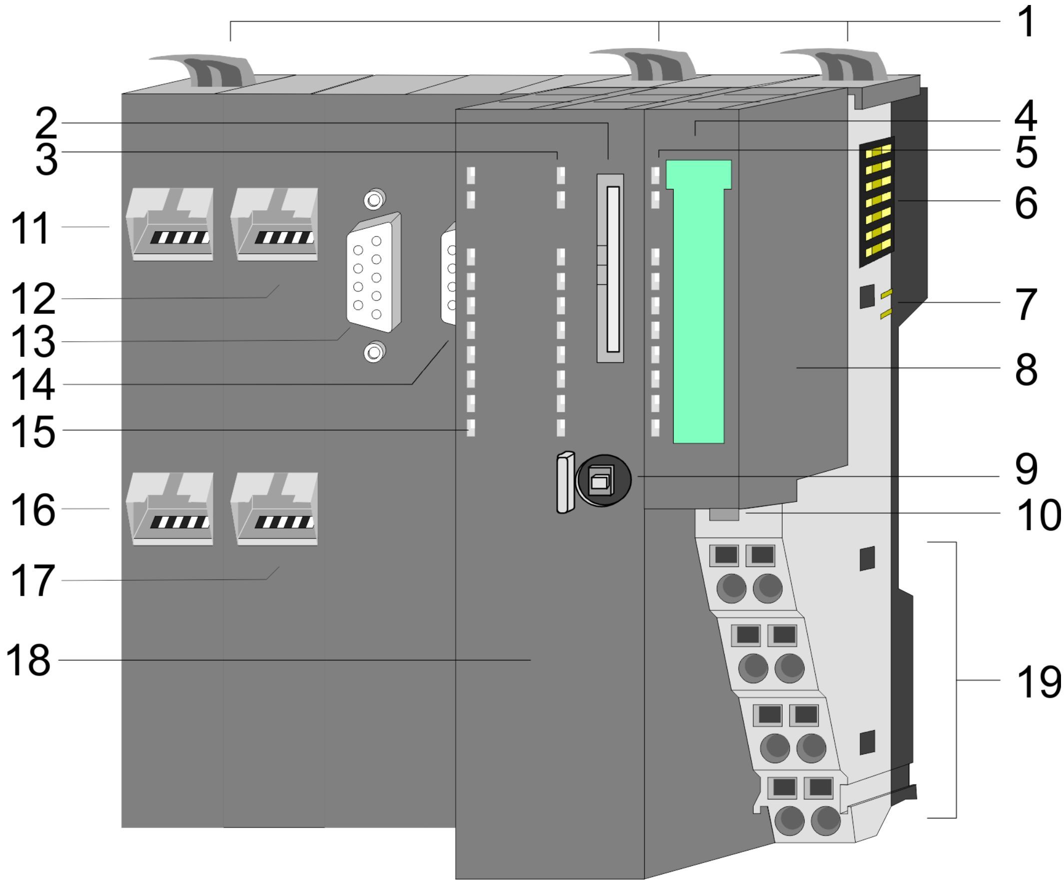
Ordering data

Type	Order number	Description
CPU 015N	015-CEFNR00	Basic CPU 015N with NET CP communication processor and EtherCAT master and options to extend work memory and bus interface.

3.2 Structure

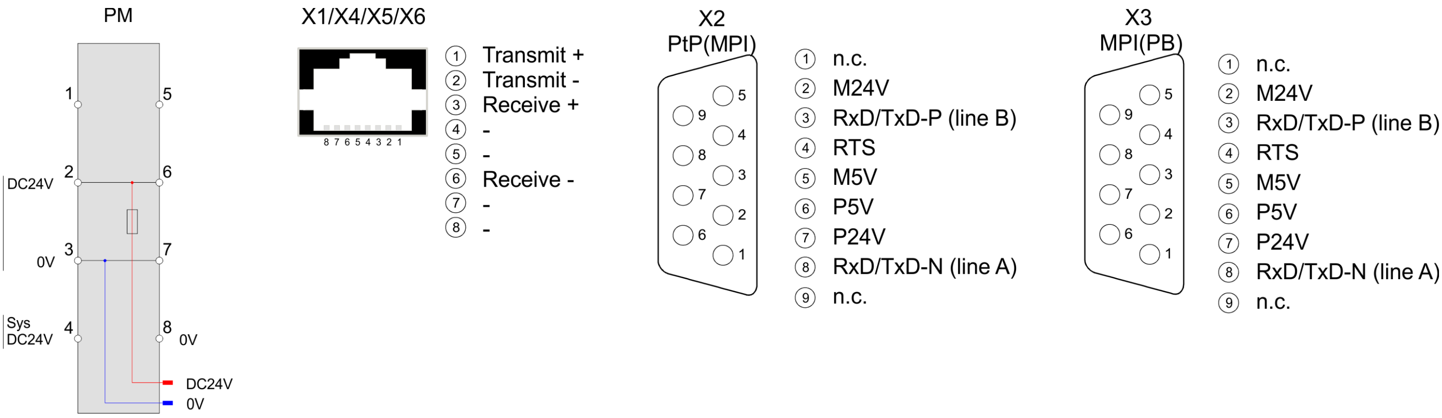
3.2.1 Basic CPU

CPU 015-CEFNR00



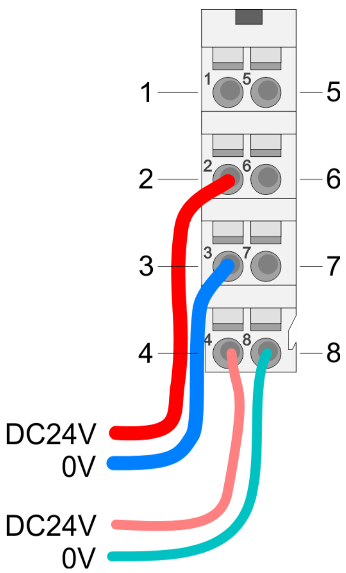
1 Locking lever
2 Slot for external storage media (lockable)
3 PLC: LEDs CPU part
4 Labelling strip power module
5 LEDs power module
6 Backplane bus
7 DC 24V power section supply
8 Power module
9 Operating mode switch CPU
10 Unlocking lever power module
11 X4: EtherCAT master
12 X1: Ethernet PG/OP channel (switch)
13 X2: PtP(MPI) interface
14 X3: MPI(PB) interface
15 CP: LEDs EtherCAT master and Ethernet CP
16 X6: NET CP
17 X5: Ethernet PG/OP channel (switch)
18 CPU part
19 Terminal power module

3.2.2 Interfaces



CAUTION!
CPU part and power module may not be separated! Here you may only exchange the electronic module!

PM - Power module



For wires with a core cross-section of 0.08mm^2 up to 1.5mm^2.

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

X1/X5: Ethernet PG/OP channel

8pin RJ45 jack:

- The RJ45 jack serves as interface to the Ethernet PG/OP channel.
- This interface allows you to program respectively remote control your CPU and to access the internal web server.
- The Ethernet PG/OP channel (X1/X5) is designed as switch. This enables PG/OP communication via the connections X1 and X5.
- Configurable connections are possible.
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040

- At the first commissioning respectively after a factory reset the Ethernet PG/OP channel has no IP address. For online access to the CPU via the Ethernet PG/OP channel, valid IP address parameters have to be assigned to this by means of your configuration tool. This is called "initialization". *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 77*
- Via the Ethernet PG/OP channel, you have access to:
  – Device website, where you can find information on firmware status, connected peripherals, current cycle times, etc.
  – *OPC UA* project, which is to be created in the *OPC UA Configurator*.
  – *WebVisu* project, which is to be created in the *SPEED7 Studio*.
  – PROFINET I-Device.

Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 77

Chap. 8 'Deployment Ethernet communication - productive' page 173

X2: PtP(MPI) interface

9pin SubD jack: (isolated):

The interface supports the following functions, which are switch able via the *product specific CPU parameters* *80*:

- PtP (default / after overall reset)
  Per default, the RS485 interface is set to PtP functionality. Using the *PtP* functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source res. target systems.
  The following protocols are supported:
  – ASCII
  – STX/ETX
  – 3964R
  – USS
  – Modbus master (ASCII, RTU)
- MPI
  The MPI interface serves for the connection between programming unit and CPU. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU. Standard setting is MPI address 2.

X3: MPI(PB) interface

9pin SubD jack: (isolated):

The interface supports the following functions, which are switch able via *'MPI interface'* in the hardware configuration:

- MPI (default / after reset to factory setting *Chap. 4.14 'Reset to factory settings' page 120*)
  Per default, the RS485 interface is set to MPI functionality. The MPI interface serves for the connection between programming unit and CPU. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU. Standard setting is MPI address 2.
- PB
  The PROFIBUS master/slave functionality of this interface can be activated by configuring the *'MPI interface'* of the CPU in the hardware configuration.

Enable bus functionality via VSC

To switch the MPI(PB) interface X3 to PROFIBUS functionality, you have to enable the according bus functionality by means of a VSC storage media from Yaskawa. By plugging the VSC storage card and then an overall reset the according functionality is enabled.

'Overview' page 121

X4: EtherCAT master

8pin RJ45 jack:

- Connect this interface with the RJ45 jack "IN" of your slave station.
- EtherCAT uses Ethernet as transfer medium. Standard CAT5 cables are used. Here distances of about 100m between 2 stations are possible.
- Only EtherCAT components may be used in an EtherCAT network. For topologies, which depart from the line structure, the corresponding EtherCAT components are necessary. Hubs may not be used.
- An EtherCAT network always consists of a master and an various number of EtherCAT salves (coupler).
- Each EtherCAT slave has an "IN" and "OUT" RJ45 jack. The arriving EtherCAT cable from the direction of the master is to be connected to the "IN" jack. The "OUT" jack is to be connected to the next station. With the respective last station the "OUT" jack remains free.

CAUTION!
Using a switch
When using an EoE terminal (Ethernet over EtherCAT) X4 and X6 must not be connected to the same switch! Due to the internal connection, this leads to a ring closure on Ethernet.

X6: NET CP

8pin RJ45 jack:

- NET CP Ethernet interface for TCP/IP communication
- Productive connections via configuration
- Productive connections via user program
- PG/OP connections
- The NET CP (Ethernet CP) provides access to:
  – *Device web page* of the Ethernet CP for information about firmware status, connection status, transfer statistics, and more.
  – Ethernet CP *OPC UA* project, which is to be created in the *OPC UA Configurator*.
  – Ethernet CP WebVisu project, which is to be configured in the SPEED7 Studio.

3.2.3 Memory management

General

The CPU has an integrated memory. Information about the capacity of the memory may be found at the front of the CPU. The memory is divided into the following parts:

- Load memory 512kbyte
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 256kbyte
  – There is the possibility to extend the work memory to its maximum capacity 512kbyte by means of a VSC.

3.2.4 Slot for storage media

Overview

In this slot you can insert the following storage media:

- VSD - **V**IPA **SD**-Card
  – External memory card for programs and firmware.
- VSC - **V**IPA**S**et**C**ard
  – External memory card (VSD) for programs and firmware with the possibility to unlock optional functions like work memory and field bus interfaces.
  – These functions can be purchased separately. → *Chap. 4.15 'Deployment storage media - VSD, VSC' page 121*
  – To activate the corresponding card is to be installed and a *Overall reset* is to be established. → *Chap. 4.12 'Overall reset' page 115*

To avoid malfunctions, you should use memory cards of Yaskawa. These correspond to the industrial standard. A list of the currently available VSD respectively VSC can be found at www.yaskawa.eu.com

3.2.5 Buffering mechanisms

The System SLIO CPU has a capacitor-based mechanism to buffer the internal clock in case of power failure for max. 30 days. With PowerOFF the content of the RAM is automatically stored in the Flash (NVRAM).

CAUTION!
Please connect the CPU for approximately 1 hour to the power supply, so that the internal buffering mechanism is loaded accordingly.

In case of failure of the buffer mechanism Date and Time 01.09.2009 00:00:00 set. Additionally, you receive a diagnostics message. → *Chap. 4.19 'Diagnostic entries' page 128*

3.2.6 Operating mode switch

General

- With the operating mode switch you may switch the CPU between STOP and RUN.
- During the transition from STOP to RUN the operating mode START-UP is driven by the CPU.
- Placing the switch to MR (**M**emory **R**eset), you request an overall reset with following load from memory card, if a project there exists.

3.2.7 LEDs

1 PLC: LEDs CPU
2 CP: LEDs EtherCAT and NET-CP
3 LEDs power module

PLC

LED	Color	Function
PW	green	CPU - Power: The CPU is power supplied.
SF	red	CPU - system fault: System error occurred. ↪ 57
RN	green	CPU - RUN: CPU is in RUN state. ↪ 57
ST	yellow	CPU - STOP: CPU is in STOP state. ↪ 57
FC	yellow	CPU - Forced: Variables are forced. ↪ 57
SD	yellow	CPU - SD memory card: Accessing the memory card. ↪ 57
DE	green	PROFIBUS - data exchange ↪ 59
BF1	red	PROFIBUS - bus error ↪ 59
L/A1	green	Ethernet PG/OP channel X1 - link/activity ↪ 58
L/A2	green	Ethernet PG/OP channel X5 - link/activity ↪ 58

CP

LED	Color	Function
BF2	red	EtherCAT X4 - bus error ↪ 59
BS	green	EtherCAT X4 - bus status ↪ 59
MT	yellow	EtherCAT X4 - maintenance ↪ 59
L/A3	green	EtherCAT X4 - link/activity ↪ 59
L/A4	green	NET-CP X6 - link/activity ↪ 60

Power module

LED	Color	Function
PWR IO	green	Power module - Power IO: Power section supply OK. ↪ 58
PWR	green	Power module - Power: Electronic section supply OK. ↪ 58
PF	red	Power module - Power fault: Fuse electronic section supply defective. ↪ 58

LEDs CPU

SF red	RN green	ST yellow	FC yellow	SD yellow	Description
Boot-up after PowerON - as soon as the CPU is supplied with 5V, the green PW-LED (Power) is on.					
		X			Firmware is loaded, here the SF-LED flickers.
					Initialization: Phase 1
					Initialization: Phase 2
					Initialization: Phase 3
					Initialization: Phase 4
Operation					
X			X	X	CPU is in STOP state.
X	2Hz		X	X	CPU is in start-up state. During the start-up (OB 100) the RUN LED blinks for at least 3s.
X		10Hz	X	X	Activation of a new hardware configuration
			X	X	CPU is in state RUN without error.
	X	X	X	X	There is a system fault. More information can be found in the diagnostics buffer of the CPU. *Chap. 4.19 'Diagnostic entries' page 128*
X	X	X		X	Variables are forced.
X	X	X	X		Accessing the memory card.
X	X	10Hz	X	X	Configuration is loaded.
Overall reset					
X		1Hz	X	X	Overall reset is requested.
X		2Hz	X	X	Overall reset is executed.
X		10Hz	X	X	Overall reset with none hardware configuration respectively hardware configuration from memory card.
Reset to factory setting					
					Reset to factory setting is executed.
					Reset to factory setting was finished without error. Then power OFF/ON is mandatory.
Firmware update					
2Hz			2Hz		The alternate blinking indicates that there is new firmware on the memory card.
2Hz			2Hz		The alternate blinking indicates that a firmware update is executed.
					Firmware update was finished without error.
10Hz		10Hz	10Hz	10Hz	Error during firmware update.
not relevant: X					

LEDs Ethernet PG/OP channel

L/A1 L/A2 green	Description
on	The corresponding Ethernet PG/OP channel is physically connected to the Ethernet.
off	The corresponding Ethernet PG/OP channel is not physically connected to the Ethernet.
flickering	The corresponding Ethernet PG/OP channel indicates Ethernet activity, here the LED flickers.

LEDs power module

PWR IO green	PWR green	PF red	Description
off	off	off	Both power supplies are missing
on	X	off	Power section supply OK
X	on	off	Electronic section OK
X	X	on	Fuse electronic section supply defective
not relevant: X			

LEDs PROFIBUS

Dependent on the mode of operation the LEDs show information about the state of operation of the PROFIBUS part according to the following pattern:

DE ■ green	BF1 ■ red	Description
Master operation		
□	□	Master has no project, this means the interface is deactivated respectively the master configured without slaves with no errors.
◩ 2Hz	□	CPU is in STOP state, the master is in "clear" state. All the slaves are in DE (data exchange) and the outputs of the slaves are disabled.
■	□	CPU is in RUN state, the master is in "operate" state. All the slaves are in DE. The outputs are enabled.
■	◩ 2Hz	CPU is in RUN state, at least 1 slave is missing and at least 1 slave is in DE.
◩ 2Hz	◩ 2Hz	CPU is in STOP state, the master is in "clear" state. At least 1 slave is missing and at least 1 slave is in DE.
□	■	PROFIBUS is interrupted (no communication possible).
□	◩ 2Hz	At least 1 slave is missing and no slave is in DE.
X	◩ 2Hz	At least 1 slave is not in DE.
Slave operation		
□	□	Slave has no configuration.
□	■	There is a bus error.
◩ 2Hz	□	Slave exchanges data with the master. Slave CPU is in state STOP.
■	□	Slave exchanges data with the master. Slave CPU is in state RUN.
not relevant: X		

LEDs EtherCAT

BF2 ■ red	BS ■ green	MT ■ yellow	Description
□	□	□	Master is in INIT state
□	◩ 2Hz	□	Master is in Pre-Op state
□	◩ pulses	□	Master is in Safe-Op state
□	■	□	Master is in OP state
X	X	□	There is no maintenance event pending.
X	X	■	There is a maintenance event pending. More may be found in the diagnostics data
□	X	X	There is no error on the EtherCAT bus pending
■	X	X	■ EtherCAT bus error, no connection to sub net ■ wrong transfer rate ■ Full-duplex-transmission is not activated

BF2 red	BS green	MT yellow	Description
1Hz	X	X	Failure of a connected IO device At least one IO device cannot be reached (topology mismatch) Faulty configuration
4s on, 1s off		4s on, 1s off	Error in configuration: 0xEA64 was added to the diagnostics buffer Additionally the SF LED of the CPU is on
4Hz		4Hz	The alternate blinking indicates that a firmware update of the EtherCAT master is executed.
			Firmware update of the EtherCAT master is finished without error.
not relevant: X			

L/A3 green	Description
	The EtherCAT master is physically connected to the Ethernet.
	The EtherCAT master is not physically connected to the Ethernet.
flickers	The EtherCAT master shows Ethernet activity.

LEDs NET-CP

L/A4 green	Description
	The NET-CP is physically connected to the Ethernet interface.
	The NET-CP is not physically connected to the Ethernet interface.
flickers	The NET-CP shows Ethernet activity.

3.3 Technical data

Order no.	015-CEFNR00
Type	CPU 015N
Module ID	-
Technical data power supply	
Power supply (rated value)	DC 24 V
Power supply (permitted range)	DC 20.4...28.8 V
Reverse polarity protection	✓
Current consumption (no-load operation)	175 mA
Current consumption (rated value)	1.1 A
Inrush current	3 A
I^2t	0.1 A^2s
Max. current drain at backplane bus	3 A
Max. current drain load supply	10 A
Power loss	8 W
Load and working memory	
Load memory, integrated	512 KB
Load memory, maximum	512 KB
Work memory, integrated	256 KB
Work memory, maximal	512 KB
Memory divided in 50% program / 50% data	✓
Memory card slot	SD/MMC-Card with max. 2 GB
Hardware configuration	
Racks, max.	5
Modules per rack, max.	total max. 64 minus number line extensions
Number of integrated DP master	1
Number of DP master via CP	-
Operable function modules	64
Operable communication modules PtP	64
Operable communication modules LAN	-
Status information, alarms, diagnostics	
Status display	yes
Interrupts	no
Process alarm	no
Diagnostic interrupt	no
Diagnostic functions	yes
Diagnostics information read-out	possible

Order no.	015-CEFNR00
Supply voltage display	green LED
Group error display	red SF LED
Channel error display	none
Command processing times	
Bit instructions, min.	0.01 µs
Word instruction, min.	0.01 µs
Double integer arithmetic, min.	0.01 µs
Floating-point arithmetic, min.	0.06 µs
Timers/Counters and their retentive characteristics	
Number of S7 counters	512
S7 counter remanence	adjustable 0 up to 512
S7 counter remanence adjustable	C0 .. C7
Number of S7 times	512
S7 times remanence	adjustable 0 up to 512
S7 times remanence adjustable	not retentive
Data range and retentive characteristic	
Number of flags	8192 Byte
Bit memories retentive characteristic adjustable	adjustable 0 up to 8192
Bit memories retentive characteristic preset	MB0 .. MB15
Number of data blocks	4096
Max. data blocks size	64 KB
Number range DBs	1 ... 8191
Max. local data size per execution level	4096 Byte
Max. local data size per block	4096 Byte
Blocks	
Number of OBs	24
Maximum OB size	64 KB
Total number DBs, FBs, FCs	4096
Number of FBs	4096
Maximum FB size	64 KB
Number range FBs	0 ... 8191
Number of FCs	4096
Maximum FC size	64 KB
Number range FCs	0 ... 8191
Maximum nesting depth per priority class	16
Maximum nesting depth additional within an error OB	4

Order no.	015-CEFNR00
Time	
Real-time clock buffered	✓
Clock buffered period (min.)	30 d
Type of buffering	Goldcap
Load time for 50% buffering period	15 min
Load time for 100% buffering period	1 h
Accuracy (max. deviation per day)	10 s
Number of operating hours counter	8
Clock synchronization	✓
Synchronization via MPI	Master/Slave
Synchronization via Ethernet (NTP)	Slave
Address areas (I/O)	
Input I/O address area	2048 Byte
Output I/O address area	2048 Byte
Process image adjustable	✓
Input process image preset	128 Byte
Output process image preset	128 Byte
Input process image maximal	2048 Byte
Output process image maximal	2048 Byte
Digital inputs	16384
Digital outputs	16384
Digital inputs central	512
Digital outputs central	512
Integrated digital inputs	-
Integrated digital outputs	-
Analog inputs	1024
Analog outputs	1024
Analog inputs, central	512
Analog outputs, central	256
Integrated analog inputs	-
Integrated analog outputs	-
Communication functions	
PG/OP channel	✓
Global data communication	✓
Number of GD circuits, max.	8
Size of GD packets, max.	22 Byte

Order no.	015-CEFNR00
S7 basic communication	✓
S7 basic communication, user data per job	76 Byte
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
S7 communication, user data per job	160 Byte
Number of connections, max.	32
Functionality Sub-D interfaces	
Type	X2
Type of interface	RS485
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	✓
MP²I (MPI/RS232)	-
DP master	-
DP slave	-
Point-to-point interface	✓
5V DC Power supply	max. 90mA, isolated
24V DC Power supply	max. 100mA, non-isolated
Type	X3
Type of interface	RS485
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	✓
MP²I (MPI/RS232)	-
DP master	optional
DP slave	optional
Point-to-point interface	-
5V DC Power supply	max. 90mA, isolated
24V DC Power supply	max. 100mA, non-isolated
Functionality MPI	
Number of connections, max.	32
PG/OP channel	✓
Routing	✓
Global data communication	✓

Order no.	015-CEFNR00
S7 basic communication	✓
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Transmission speed, min.	19.2 kbit/s
Transmission speed, max.	12 Mbit/s
Functionality PROFIBUS master	
Number of connections, max.	32
PG/OP channel	✓
Routing	✓
S7 basic communication	✓
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Activation/deactivation of DP slaves	✓
Direct data exchange (slave-to-slave communication)	-
DPV1	✓
Transmission speed, min.	9.6 kbit/s
Transmission speed, max.	12 Mbit/s
Number of DP slaves, max.	124
Address range inputs, max.	2 KB
Address range outputs, max.	2 KB
User data inputs per slave, max.	244 Byte
User data outputs per slave, max.	244 Byte
Functionality PROFIBUS slave	
Number of connections, max.	32
PG/OP channel	✓
Routing	✓
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Direct data exchange (slave-to-slave communication)	-
DPV1	✓
Transmission speed, min.	9.6 kbit/s
Transmission speed, max.	12 Mbit/s
Automatic detection of transmission speed	-

Order no.	015-CEFNR00
Transfer memory inputs, max.	244 Byte
Transfer memory outputs, max.	244 Byte
Address areas, max.	32
User data per address area, max.	32 Byte
Functionality RJ45 interfaces	
Type	X1
Type of interface	Ethernet 10/100 MBit Switch
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	4
Productive connections	-
Fieldbus	-
Type	X5
Type of interface	Ethernet 10/100 MBit Switch
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	4
Productive connections	-
Fieldbus	-
Type	X4
Type of interface	Ethernet 100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	-
Number of connections, max.	-
Productive connections	-
Type	X6
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓

Order no.	015-CEFNR00
Number of connections, max.	8
Productive connections	✓
Point-to-point communication	
PtP communication	✓
Interface isolated	✓
RS232 interface	-
RS422 interface	-
RS485 interface	✓
Connector	Sub-D, 9-pin, female
Transmission speed, min.	1200 bit/s
Transmission speed, max.	115.5 kbit/s
Cable length, max.	500 m
Point-to-point protocol	
ASCII protocol	✓
STX/ETX protocol	✓
3964(R) protocol	✓
RK512 protocol	-
USS master protocol	✓
Modbus master protocol	✓
Modbus slave protocol	✓
Special protocols	-
Properties PROFINET I/O-Controller via PG/OP	
Realtime Class	-
Conformance Class	-
Number of PN IO devices	-
IRT support	-
Shared Device supported	-
MRP Client supported	-
Prioritized start-up	-
Number of PN IO lines	-
Address range inputs, max.	-
Address range outputs, max.	-
Transmitting clock	-
Update time	-
Isochronous mode	-
Parallel operation as controller and I-Device	-

Order no.	015-CEFNR00
Properties PROFINET I-Device via PG/OP	
I/O Data range, max.	512 Byte
Update time	1 ms .. 512 ms
Mode as Shared I-Device	-
Properties PROFINET I-Device via CP	
I/O Data range, max.	-
Update time	-
Mode as Shared I-Device	-
Management & diagnosis via PG/OP	
Protocols	ICMP DCP LLDP / SNMP NTP
Web based diagnosis	✓
NCM diagnosis	-
Ethernet communication CP	
Number of configurable connections, max.	8
Number of productive connections by Siemens NetPro, max.	8
S7 connections	BSEND, BRCV, GET, PUT, Connection of active and passive data handling
User data per S7 connection, max.	32 KB
TCP-connections	FETCH PASSIV, WRITE PASSIV, Connection of passive data handling
User data per TCP connection, max.	64 KB
ISO-connections	-
User data per ISO connection, max.	-
ISO on TCP connections (RFC 1006)	FETCH PASSIV, WRITE PASSIV, Connection of passive data handling
User data per ISO on TCP connection, max.	32 KB
UDP-connections	-
User data per UDP connection, max.	-
UDP-multicast-connections	-
UDP-broadcast-connections	-
Ethernet open communication	
Number of connections, max.	8
ISO on TCP connections (RFC 1006)	TSEND, TRCV, TCON, TDISCON
User data per ISO on TCP connection, max.	8 KB

Order no.	015-CEFNR00
TCP-Connections native	TSEND, TRCV, TCON, TDISCON
User data per native TCP connection, max.	8 KB
User data per ad hoc TCP connection, max.	1460 Byte
UDP-connections	TUSEND, TURCV
User data per UDP connection, max.	1472 Byte
Ethernet communication via PG/OP	
Number of productive connections via PG/OP, max.	4
Number of productive connections by Siemens NetPro, max.	4
S7 connections	BSEND, BRCV, GET, PUT, Connection of active and passive data handling
User data per S7 connection, max.	64 KB
TCP-connections	FETCH PASSIV, WRITE PASSIV, Connection of passive data handling
User data per TCP connection, max.	8 KB
ISO on TCP connections (RFC 1006)	FETCH PASSIV, WRITE PASSIV, Connection of passive data handling
User data per ISO connection, max.	8 KB
Ethernet open communication via PG/OP	
Number of configurable connections, max.	4
ISO on TCP connections (RFC 1006)	TSEND, TRCV, TCON, TDISCON
User data per ISO on TCP connection, max.	32 KB
TCP-Connections native	TSEND, TRCV, TCON, TDISCON
User data per native TCP connection, max.	32 KB
User data per ad hoc TCP connection, max.	1460 Byte
UDP-connections	TUSEND, TURCV
User data per UDP connection, max.	1472 Byte
EtherCAT Master	
Number of EtherCAT-slaves	128
Update time	1 ms .. 512 ms
Address range inputs, max.	2 KB
Address range outputs, max.	2 KB
EoE support	✓
CoE support	✓
FoE support	✓
Distributed Clock support	✓
Hotconnect Slaves	✓
Isochronous mode	✓

Order no.	015-CEFNR00
Management & diagnosis	
Protocols	ICMP DCP
Web based diagnosis	-
NCM diagnosis	-
WebVisu via PG/OP	
WebVisu is supported	✓
Max. number of connections WebVisu	4
WebVisu supports HTTP	✓
WebVisu supports HTTPS	✓
OPC UA server via PG/OP	
OPC UA server is supported	✓
Max. number of connections per interface	6
Services	Data Access (Read, Write, Subscribe)
Security policies	None, Basic128Rsa15, Basic256, Basic256Sha256
Authentication	Anonymous, username and password
WebVisu via CP	
WebVisu is supported	✓
Max. number of connections WebVisu	4
WebVisu supports HTTP	✓
WebVisu supports HTTPS	✓
OPC UA server via CP	
OPC UA server is supported	✓
Max. number of connections per interface	6
Services	Data Access (Read, Write, Subscribe)
Security policies	None, Basic128Rsa15, Basic256, Basic256Sha256
Authentication	Anonymous, username and password
Housing	
Material	PPE / PPE GF10
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	131.5 mm x 109 mm x 83 mm
Net weight	335 g
Weight including accessories	335 g
Gross weight	365 g
Environmental conditions	

Order no.	**015-CEFNR00**
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	yes
KC certification	yes

4 Deployment CPU 015-CEFNR00

4.1 Please note!

The following descriptions always refer to the usage in the SPEED7 Studio. Information on usage in the Siemens SIMATIC Manager or TIA Portal can be found here:

Chap. 12 'Configuration with Siemens SIMATIC Manager' page 293

Chap. 13 'Configuration with TIA Portal' page 323

4.2 Assembly

Information about assembly and cabling *Chap. 2 'Basics and mounting' page 12*

4.3 Start-up behavior

Turn on power supply

- The CPU checks whether a project AUTOLOAD.WLD exists on the memory card. If so, an overall reset is executed and the project is automatically loaded from the memory card.
- The CPU checks whether a command file with the name VIPA_CMD.MMC exists on the memory card. If so the command file is loaded from the memory card and the commands are executed.
- After PowerON and CPU STOP the CPU checks if there is a *.pkb file (firmware file) on the memory card. If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. *Chap. 4.13 'Firmware update' page 116*
- The CPU checks if a previously activated VSC is inserted. If not, the SF LED gets on and a diagnostics entry is released. The CPU switches to STOP after 72 hours. With a just installed VSC activated functions remain activated. *Chap. 4.19 'Diagnostic entries' page 128*

After this the CPU switches to the operating mode, which is set on the operating mode switch.

Delivery state

In the delivery state the CPU is overall reset. After a STOP→RUN transition the CPU switches to RUN without program.

4.4 Addressing

4.4.1 Overview

To provide specific addressing of the installed periphery modules, certain addresses must be allocated in the CPU. This address mapping is in the CPU as hardware configuration. If there is no hardware configuration, depending on the slot, the CPU assigns automatically periphery addresses for digital in-/output modules starting with 0 and analog modules are assigned to even addresses starting with 256.

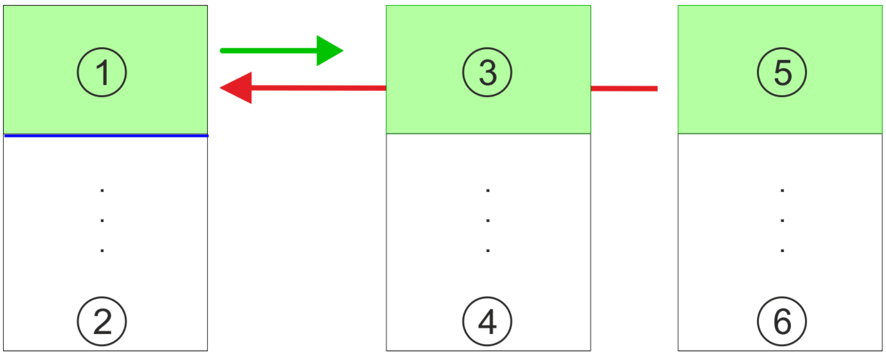
4.4.2 Addressing backplane bus periphery

The CPU 015-CEFNR00 provides an periphery area (address 0 ... max. periphery address) and a process image of the in- and outputs (each address default 0 ... 127). The size of the process image can be preset via the parametrization.

↳ *'Cycle / Clock memory' page 82*

The process image is updated automatically when a cycle has been completed. The process image is divided into two parts:

- Process image to the inputs (PII)
- Process image to the outputs (PIQ)



1 Periphery area: 0 ... 127 (default)
2 max periphery area
3 Process image of the inputs (PII): 0 ... 127
4 max. process image of the inputs (PII)
5 Process image of the outputs (PIQ): 0 ... 127
6 max. process image of the outputs (PIQ)

Max. number of pluggable modules

Up to 64 System SLIO modules can be connected to a System SLIO CPU. This sum includes power and clamp modules.

Define addresses by hardware configuration

You may access the modules with read res. write accesses to the periphery bytes or the process image. To define addresses a hardware configuration may be used. For this, click on the properties of the according module and set the wanted address.

Automatic addressing

If you do not like to use a hardware configuration, an automatic addressing is established. Here the address assignment follows the following specifications:

- Starting with slot 1, the central plugged modules are assigned with ascending logical addresses.
- The length of the memory area corresponds to the size of the process data of the according module. Information about the sizes of the process data can be found in the according manual of the module.

- The memory areas of the modules are assigned without gaps separately for input and output area.
- Digital modules are mapped starting at address 0 and all other modules are mapped starting from address 256. ETS modules are mapped starting from address 256.
- As soon as the mapping of digital modules exceeds the address 256, by regarding the order, these are mapped starting from address 256.

Example for automatic address allocation

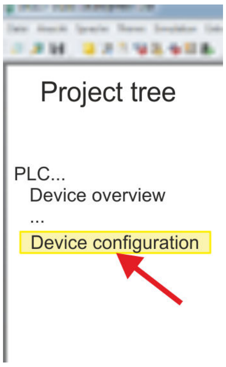
Slot	Type	Description	Length	I address	O address
1	021-1BF00	DI 8x	1 Byte	0	
2	021-1BF00	DI 8x	1 Byte	1	
3	022-1BF00	DO 8x	1 Byte		0
4	031-1BB30	AI 2x	4 Byte	256...259	
5	032-1BB30	AO 2x	4 Byte		256...259
6	031-1BD40	AI 4x	8 Byte	260...267	
7	032-1BD40	AO 4x	8 Byte		260...267
8	022-1BF00	DO 8x	1 Byte		1
9	021-1BF00	DI 8x	1 Byte	2	

4.5 Hardware configuration - CPU

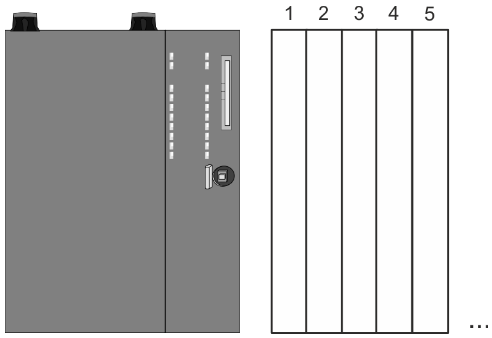
Precondition

> *For project engineering a thorough knowledge of the SPEED7 Studio is required!*

Proceeding



1. Start the *SPEED7 Studio*.
2. Create a new project in the *Work area* with *'New project'*.
   ⇨ A new project is created and the view *'Devices and networking'* is shown.
3. Click in the *Project tree* at *'Add new device ...'*.
   ⇨ A dialog for device selection opens.
4. Select from the *'Device templates'* your CPU and click at [OK].
   ⇨ The CPU is inserted in *'Devices and networking'* and the *'Device configuration'* is opened.



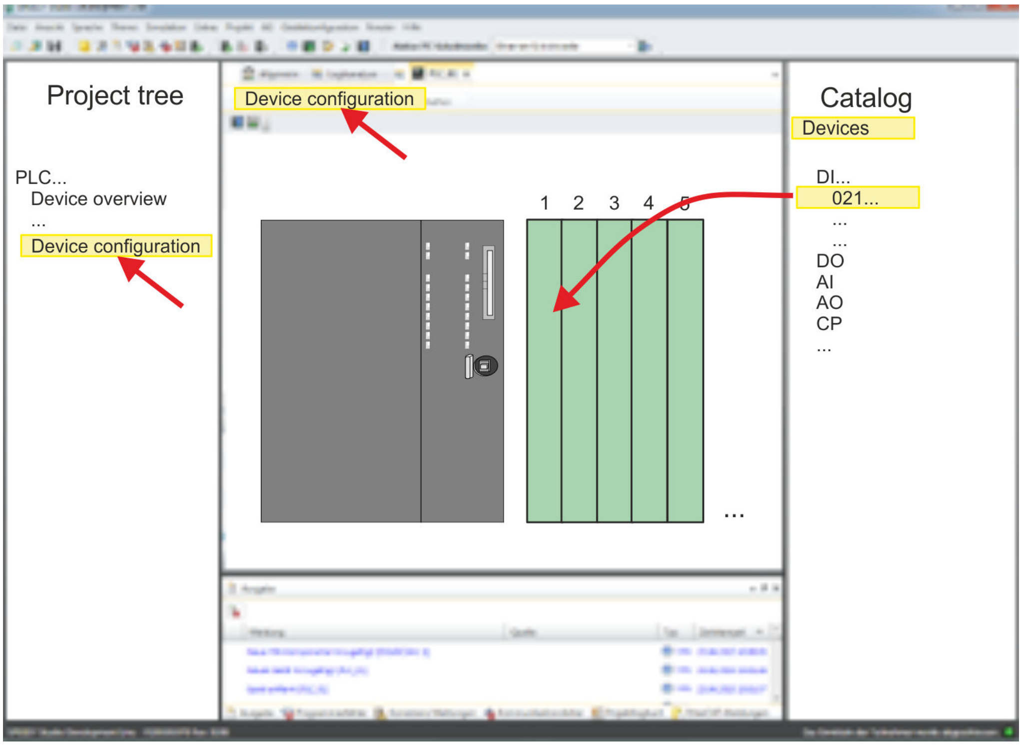
Device configuration

Slot	Module	...	...	...	...
0	CPU 015-CEFNR00				
-X1	PG_OP_Ethernet				
-X3	MPI interface				
...	...			...	

4.6 Hardware Configuration - I/O modules

Hardware configuration of the modules

1. Click in the *'Project tree'* at *'PLC... > Device configuration'*.
2. Starting with slot 1 place in the *'Device configuration'* your System SLIO modules in the plugged sequence. For this drag from the hardware catalog the corresponding module to the corresponding position in the *Device configuration*.



Parametrization

For parametrization double-click in the *'Device configuration'* on the module you want to parameterize. Then the parameters of the module are shown in a dialog. Here you can make your parameter settings.

Parametrization during runtime

By using the SFCs 55, 56 and 57 you may alter and transfer parameters for wanted modules during runtime. For this you have to store the module specific parameters in so called "record sets". More detailed information about the structure of the record sets is to find in the according module description.

4.7 Hardware configuration - Ethernet PG/OP channel

Overview

> ***Please note!***
> – *At the first commissioning respectively after a reset to factory setting the Ethernet interface has no IP address.*
> – *For online access, you have to assign valid IP address data to it by means of "Initialization".*
> – *After initialization, you can transfer the IP address data to your project.*

The CPU has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU.

- The Ethernet PG/OP channel (X1/X5) is designed as switch. This enables PG/OP communication via the connections X1 and X5.
- Configurable connections are possible.
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040
- Via the Ethernet PG/OP channel, you have access to:
  – Device website, where you can find information on firmware status, connected peripherals, current cycle times, etc.
  – *OPC UA* project, which is to be created in the *OPC UA Configurator*.
  – *WebVisu* project, which is to be created in the *SPEED7 Studio*.
  – PROFINET I-Device

Assembly and commissioning

1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the one of the Ethernet jacks (X1, X5) of the Ethernet PG/OP channel to Ethernet.
4. Switch on the power supply.

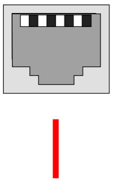
   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization"

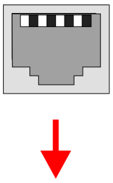
The initialization takes place with the following proceeding:

- Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of your CPU with the name "MAC PG/OP: ...".

X1 PG/OP

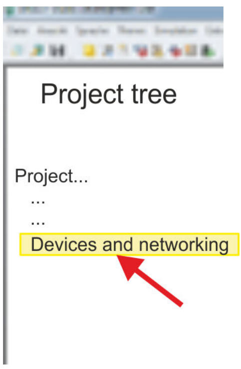


X5 PG/OP



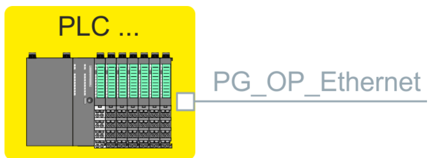
MAC PG/OP: 00-20-D5-77-05-10

Assign IP address parameters

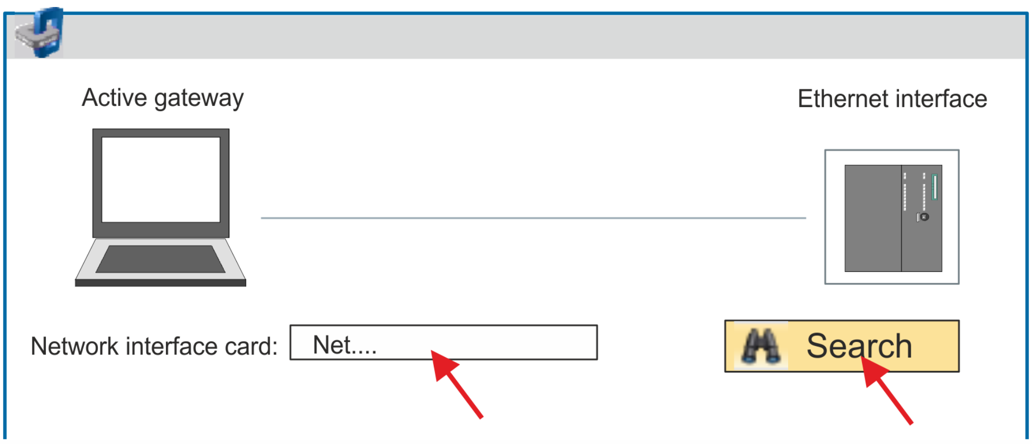


You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the *SPEED7 Studio* with the following proceeding:

1. Start the *SPEED7 Studio* with your project.
2. Click in the *Project tree* at *'Devices and networking'*.
   ⇨ You will get a graphical object view of your CPU.



3. Click at the network *'PG_OP_Ethernet'*.
4. Select *'Context menu ➔ Determine accessible partner'*.
   ⇨ A dialog window opens.



5. Select the according network interface card, which is connected to the Ethernet PG/OP channel and click at *'Search'* to determine the via MAC address reachable device.

   ⇨ The network search is started and the found stations are listed in a table.

6.

	Devices...	IP...	MAC...	Device...	...	...
1		172.20. ..	00:20: ...	Yaskawa ...		
2		...	...	...		

   Click in the list at the module with the known MAC address. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".

7. Click at *'Set IP address'*. Now set the IP configuration by entering *'IP address'*, *'Subnet mask'* and *'Gateway'*.

8. Click at *'Set IP address'*.

   ⇨ The IP address is transferred to the module and the list is refreshed. Directly after the assignment the Ethernet PG/OP channel is online reachable using the set IP address data. The value remains as long as it is reassigned, it is over-written by a hardware configuration or a factory reset is executed.

9. With clicking at *'Apply settings'* the IP address data a stored in the project.

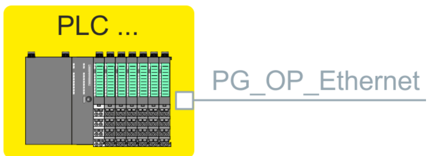
Take IP address parameters in project

If you are not online, you can assign IP address data to your Ethernet PG/OP channel with following proceeding:

1. Start the *SPEED7 Studio* with your project.

2. Click in the *Project tree* at *'Devices and networking'*.

   ⇨ You will get a graphical object view of your CPU.



3. Click at the network *'PG_OP_Ethernet'*.

4. Select *'Context menu ➔ Interface properties'*.

   ⇨ A dialog window opens. Here you can enter the IP address data for your Ethernet PG/OP channel.

5. Confirm with [OK].

   Local components

Slot	Module	...	...	...IP address	...
0	CPU 015-CEFNR00			...	...
-X1	PG_OP_Ethernet			172.20.120.40	
...	...			...	

   ⇨ The IP address data are stored in your project listed in *'Devices and networking'* at *'Local components'*.

   After transferring your project your CPU can be accessed via Ethernet PG/OP channel with the set IP address data.

4.8 Setting CPU parameters

Proceeding

1. Click in the *'Project tree'* at *'PLC... > Device configuration'*.
2. Click at the CPU and select *'Context menu → Components properties'*.
   ⇨ The properties dialog is opened. Here you can adjust all the CPU parameters.

> *Depending on the CPU, the settings are different. Selection or input fields that are grayed out can not be edited at this CPU type.*

4.8.1 Parameter CPU

General

Here you can make general settings for the current CPU.

- Name
  – Name of the PLC. This name is shown in the project tree.
- Plant designation
  – Here is the possibility to specify a plant designation for the CPU.
  – This plant designation identifies parts of the plant according to their function.
  – Its structure is hierarchic according to IEC 1346-1.
- Location designation
  – The location designation is part of the resource designation.
  – Here the exact location of your module within a plant may be specified.
- MPI data
  – Here you can adjust the setting of the MPI subnet (Multi Point Interface) for serial connection between the MPI participants.
  – Address: Here you can enter the MPI address. Standard setting of the CPUs is MPI address 2. The address 0 is reserved for programming devices.
  – Highest address: By specifying the highest address number you can limit the range of addresses.
  – Secondary transmission speed MPI The transmission rate (Bit/s) of the MPI subnet must not be higher than the transmission rate of the slowest accessible MPI partner.

Feature set

Here you can activate the corresponding additional functions in the *SPEED7 Studio*:

- *'Motion Control + ... Axes'*.
  – Isochronous mode with activation of OB 60 and OB 61 for the corresponding number of axes.
- PROFIBUS slave/master functionality
  – Enabling of the PROFIBUS slave or master functionalities

With *'inactive'* the additional functions can be disabled.

> *Please consider the additional functions in the SPEED7 Studio can only be activated, if you have valid license for these functions!*

Start-up

Here you can make general configurations for the start-up behaviour of the current CPU.

- Start-up if present configuration does not match actual configuration
  - The expected configuration is the configuration of the components which is defined in the project and uploaded to the CPU.
  - The actual configuration is the implemented configuration of the components.
  - If this option is deselected, the CPU remains in the STOP mode for the following cases:

    One or more components are not located in the configured slot.

    A component of another type is located in the configured slot.
  - If this option is selected, the CPU switches to the RUN mode even if the components are not located in the configured slots or if components of another type are located there.
- Deleting PAA after warm restart
  - If this option is selected, the process image of the outputs (PAA) is deleted after the warm restart of the CPU.
- Disable hot restart by operator
  - The types of start-up are restricted when triggered by the operation or communication job.
  - If this option is selected, only restart or cold start are possible. Warm restart is not possible.
  - If this option is deselected, all types of start-up are possible.
- Start-up after Power On
  - Here you can select whether a restart, warm restart or cold start should be made after having activated the power supply (powerON).
  - Cold start: All variables and memory blocks are initialised.
  - Restart (warm start) The non-retentive memory areas are initialised, the retentive memory areas are restored.
  - Warm restart: The user program is continued where it has been interrupted.
- Monitoring time for ...
  - The time base of the following parameters is 100 milliseconds. Multiply the entered value with the time base. Example: Entered value 650 * 100 ms = 65.000 ms of monitoring time
  - Finished message from modules (100 ms): Maximum duration of the Ready signal of all configured components after having switched on the power supply (powerON).
  - Transfer of parameters to modules (100 ms) Maximum duration of the parameter transfer to the parameterizable components.
  - Hot restart (100 ms) Maximum duration of the warm restart. If the time between powerOFF and PowerON or between STOP mode and RUN mode is longer than the time entered here, there is no warm restart. The CPU remains in STOP mode.

Synchronous cycle interrupt

Here you can make settings for synchronous cycle and type.

- OB 61
  - Currently you can not make any settings here. These data are information on OB 61 and may not be changed.
- Behavior at runtime violation
  - Here you can set how the CPU should behave when a runtime violation occurs.
- Warn threshold
  - Enter a value in %, which serves as a threshold for the runtime violation, as soon as the application cycle time is exceeded.
- Error behaviour
  - disabled: Runtime violations are ignored.
  - CPU stops: If the runtime is violated, the CPU goes to STOP.
  - OB 80 is requested If the runtime is violated, OB 80 is requested.

- Error count limit
  – Specify here how often the runtime may be violated until it is reported to the system as runtime error.
- Synchronize local System SLIO bus
  – If selected, the address range of the System SLIO modules on the backplane bus is mapped in the process image of OB 61.

> – *In principle, isochronous does not support analog modules on the System SLIO bus. You can add analog modules to the process image of the OB 61. Their input and output data are not processed isochronously. If you use modules on the System SLIO bus system, which do not support isochronous, you will get the diagnostic message 0xEB05 (bus configuration for isochronous process diagram not suitable). The module's error LED blinks.*

Cycle / Clock memory

Here you can make general configurations for the start-up behaviour of the current CPU.

- Refresh process image cyclically
  – If this option is selected, the process image of the organisation block OB 1 is cyclically updated. This expands the cycle time.
- Scan cycle monitoring time (ms)
  – Here you can enter the scan cycle monitoring time in ms.
  – If the run time of the user program exceeds the scan cycle monitoring time, the CPU switches to STOP mode.
  – Reasons for time-out:
    - Communication principle
    - Accumulation of alarm events
    - Error in the CPU program
- Minimum scan cycle time (ms)
  – Guaranteed compliance with a minimum scan cycle time: The start of a new cycle is delayed until the minimum cycle time has been reached.
- Scan cycle load from communication (%)
  – Percentage of communication processes compared to the complete cycle time.
  – With this parameter you can control the duration of communication processes within certain limits, which always extend the cycle time.
  – If e.g. set to 50%, the cycle time might double. In addition, the OB 1 cycle will be extended by asynchronous events (e.g. process interrupts).
- OB 85 calling at periphery access errors
  – Reaction of the CPU after periphery access errors during the update of the process image.
  – The CPU is preset such that on a periphery access error the OB 85 is not called and there is no entry made to the diagnostic buffer.
- Size of the process-image input/output area
  – Here you can define the size of the process image max. 2048 for input and output periphery (default: 128).
- Clock memory
  – Clock memory Enable this option, if the CPU should provide clock memories. Clock memories periodically change their value in pre-set intervals.
  – Memory byte: Number of the memory byte for the clock memory. The memory byte is used only if you select the *'Clock memory'* option.

> *The selected memory byte cannot be used for temporary data storage.*

Retentive memory

In order to maintain data in case of power failure, certain data areas can be marked as retentive. A restart (warm start) will restore the values of the retentive memory areas from the last program cycle.

- Number of memory bytes starting with MB0
  - Here you can enter the number of retentive memory bytes starting from memory byte 0. Example: Example: Input value 16 = memory bytes 0 to 15 are retentive.
- Number of timers starting with T0
  - Here you can enter the number of retentive timers starting from T0.
- Number of counters starting with Z0
  - Here you can enter the number of retentive counter starting from Z0.
- Areas
  - You can define up to 8 retentive memory areas in the data blocks:
  - DB no.: Number of the retentive data block
  - Byte address: Starting address within the retentive data block
  - Number of bytes: Number of retentive bytes from the starting address within the data block.

Interrupts

Here you can define the order for processing the individual interrupt organisation blocks. OBs with the smallest number have lowest priority. OBs with priority 0 are not processed.

- Priority: The following interrupt OBs are listed:
  - OB 40 - OB 47: Hardware interrupts
  - OB 20 - OB 23: Time delay interrupts
  - OB 50, OB 51, OB 55 - OB 57: Communication interrupts
  - OB 81 - OB 87: Async. error interrupts

Time of day interrupts

The time of day interrupt organisation blocks OB10 to OB17 can interrupt the processing of OB1 once or at a certain interval. Depending on the CPU used, you can parameterize up to 8 time of day interrupts.

- Priority
  - Order in which a time of day interrupt organisation block is processed.
  - OBs with the smallest number have lowest priority.
  - OBs with priority 0 are not processed.
- Active
  - By selecting *'Active'* the time of day interrupt is activated.
- Execution
  - Here you can select the execution of interrupts once or at certain intervals.
  - The intervals from every minute to yearly are related to the settings at *Start date* and *Time*.
- Start date/time
  - Here you can define the time of the initial execution of the time of day interrupt.

Cyclic interrupts

The cyclic interrupt organisation blocks OB 30 to OB 38 can interrupt the processing of OB1 once or at a certain interval. Depending on the CPU used, you can parameterize up to 9 cyclic interrupts.

- Priority
  - Order in which a cyclic interrupt organisation block is processed.
  - OBs with the smallest number have lowest priority.
  - OBs with priority 0 are not processed.
- Execution
  - Specify the time intervals in ms, in which the watchdog interrupt OBs should be processed.
  - The starting time is the switching from STOP mode to RUN mode.
- Phase shift
  - Time in milliseconds by which the execution time of the cyclic interrupt is to be delayed.
  - By selecting several cyclic interrupts, you can use the phase offset to make sure that the cyclic interrupts do not start at the same time.

Diagnostics/Clock

Here you can define, which clock is to be synchronised with which clock.

- Extended functions
  - The expanded range of functions for diagnostics is not supported.
- Report cause of STOP
  - Activate this parameter, if on transition to STOP the CPU has to report a cause for the STOP to PG respectively OP.
- Report to process control active
  - Currently this function is not supported.
- Clock
  - Here you can define, which clock is to be synchronised with another clock.
- Synchronization type
  - Here you specify whether the clock synchronizes other clocks or not.
  - As slave: The clock is synchronized by another clock.
  - As master: The clock synchronizes other clocks as master.
  - none: There is no synchronization.
- Time interval
  - Time intervals within which the synchronization should take place.
- Correction factor
  - By specifying a correction factor in ms, you can compensate the deviation of the clock within 24 hours.
  - If the clock runs slow by 1 second within 24 hours, you can adjust this deviation with the correction factor "+1000".

Protection

- Protection Level
  – Here you can configure a protection level to protect the CPU against unauthorised access.
  – *No protection* (default):
  No password adjustable; no restrictions
  – *Write-protection* with password:
  Known password: Read and write access
  Unknown password: Read access only.
  – *Read/write protection* with password:
  Known password: Read and write access
  Unknown password: No read and write access
- Password
  – Here you can specify a password for the write and read protection.
  – Depending on the setting of the protection level for read respectively write access a password is requested.

Advanced configurations

Here you can adjust the functionality of the interface and specify the number of flags, timers and counters:

- Function X2
  – Function PtP(MPI) interface X2
  – PtP (default): With this operating mode the RS485 interface acts as an interface for serial point-to-point communication. Here data may be exchanged between two stations by means of protocols.
  – MPI: With this operating mode the interface serves for the connection between programming unit and CPU via MPI. By means of this e.g. the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU.
- MPI address X2
  – With *MPI* you can specify the MPI address here. With *PTP* this parameter is ignored by the CPU.
  – Range of values: 2 (default) ... 31
- MPI Baud rate X2:
  – With *MPI* you can specify the MPI transmission rate here. With *PTP* this parameter is ignored by the CPU.
  – Range of values: 19.2kB/s ... 12MB/s, default: 187.5kB/s
- Additional retentive memory
  – Here enter the number of retentive memory bytes. With 0 the preset value of *'Number of memory bytes starting with MB0'* of *'Retentive memory'* is taken.
  – Range of values: 0 (default) ... 8192
- Additional retentive timer
  – Here enter the number of timer. With 0 the preset value of *'Number of timer starting with T0'* of *'Retentive memory'* is taken.
  – Range of values: 0 (default) ... 512
- Additional retentive counter
  – Enter here the number of counter. With 0 the preset value of *'Number of counter starting with Z0'* of *'Retentive memory'* is taken.
  – Range of values: 0 (default) ... 512
- Priority OB 57
  – Here you can specify the priority for OB 57.
  – Range of values: 2 (default) ... 24
- OB 80 for cyclic interrupt error
  – Here you can set for which cyclic interrupt OB 80 (time error) should be called.
  – Range of values: Deactivated (default), selection of the corresponding OB

- Direct DX transition
  if this parameter is activated, the integrated PROFIBUS DP master, if activated by VSC, shows the following behavior:
  – As long as a DP slave is in Data Exchange, i.e. in the DP slave standard diagnostic data byte 0, bit 1 and byte 1, bit 0 have the status 0, this DP slave is directly taken from the DP master in Data Exchange. The transition happens without a *SetPrm*- and *CheckConfig* telegram was being sent to the DP Slave.
  – When a DP slave is taken to Data Exchange, the outputs remain active and are not deactivated.
  – If the CPU goes from RUN to STOP, the DP master is deactivated for at least the duration of the *Response monitoring time*, which is to be set in the PROFIBUS parameters. Afterwards, the DP master becomes active again and takes the DP slaves back to Data Exchange. When the DP master transitions to the inactive state, the output data of the DP slaves are not zeroed nor deactivated. Standard-compliant DP slaves automatically will switch off the outputs or zero them if they do not receive DE telegrams from the master during the *Response monitoring time*.
  – If the power supply of the CPU fails, the outputs of the DP slaves are not zeroed and not deactivated. Standard-compliant DP slaves automatically will switch off the outputs or zero them if they do not receive DE telegrams from the master during the *Response monitoring time*.
- PN MultipleWrite
  – In the activated state, parameter record sets are combined at PROFINET to one or more Ethernet frames during the connection setup. This speeds up the connection setup, since a separate Ethernet frame is not used for each parameter record set.
- Free Module Mapping *89*
  – When activated, you can use your CPU in different hardware variants.
  – You specify the mapping at runtime with record set 0x7F.
- Ethernet port ...
  – Here you can disable individual Ethernet interfaces.

> *Please note that by disabling e.g. the Ethernet PG/OP channel after transferring the hardware configuration, the CPU can no longer be configured via this Ethernet PG/OP channel. The access setting can be reset by an overall reset.*

- ... RAS protocol
  – Here you can deactivate the RAS protocol for remote access for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- ... NTP protocol
  – Here you can deactivate the NTP protocol for time synchronisation between the stations for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- ... OPC UA
  – Here you may deactivate the OPC UA protocol for access to an OPC UA project on the CPU respectively the CP part.
- ... Open communication
  – Here you can deactivate the protocol for communication via the user program when using handling blocks for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- ... Device WebSite
  – Here you can deactivate the protocol for access to the integrated web server for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.

- ... Web Visu
  – Here you can deactivate the protocol for access to the Web visualization for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- ... PG/OP protocol
  – Here you can deactivate the protocol for PG/OP communication via Siemens S7 connections for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- ... PG/OP routing (access point)
  – Here you can deactivate routing requests via Siemens S7 connections for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- ... NetPro connections ...
  – Here you can deactivate the protocol for communication between PLC systems based on Siemens STEP®7 by means of configured communication connections for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- ... DCP protocol
  – Here you can deactivate the telegram for determining accessible nodes on PROFINET for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- ... LLDP protocol
  – Here you can deactivate the telegram for determining accessible nodes on PROFINET for the CPU or CP part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- Field bus PN
  – Here you can deactivate the communication via PROFINET telegrams for the CPU part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- MPI(PB) PG/OP protocol
  – Here you can deactivate the protocol for PG/OP communication via MPI(PB) interface X3 for the CPU part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- PtP(MPI) PG/OP protocol
  – Here you can deactivate the protocol for PG/OP communication via PtP(MPI) interface X2 for the CPU part. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- Global data communication
  – Here you can deactivate the protocol for the cyclic data exchange between CPUs via the MPI interface. If a protocol is deactivated, requests via the disabled protocol will be rejected.
- Reduced PDU size
  – When activated, the PDU size is reduced accordingly.
  – For some protocols, some configuration tools require a reduced PDU size. For example, in the Siemens TIA Portal for variable forcing, the PDU size must be reduced. For the processing of single steps at several breakpoints, the PDU size must also be reduced.

4.8.2 Parameter MPI interface

Via double-click at *'MPI interface'* of *'Device configuration'* the properties dialog of the MPI interface is opened.

- MPI data
  – Here you can adjust the setting of the MPI subnet (Multi Point Interface) for serial connection between the MPI participants.
  – Address: Here you can enter the MPI address. Standard setting of the CPUs is MPI address 2. The address 0 is reserved for programming devices.
  – Highest address: By specifying the highest address number you can limit the range of addresses.
  – Secondary transmission speed MPI: The transmission rate (Bit/s) of the MPI subnet must not be higher than the transmission rate of the slowest accessible MPI partner.

4.8.3 Parameter Ethernet

Via double-click at *'Ethernet'* of *'Device configuration'* the properties dialog of the NET CP interface (X6) is opened.

- General
  – General: Here you can enter a name for your station.
  – Subnet ID The subnet ID is used to uniquely identify your network.
  – PDU size: Here you can specify the buffer size for the Ethernet communication.
- IP configuration
  – IP address: Here you can assign an IP address to your NET CP.
  – Subnet: Here you can enter the subnet mask of your network.

4.8.4 Free Module Mapping

4.8.4.1 Overview

Free module mapping - FMM

- With *FMM* you can use your CPU in different hardware variants without adapting your user program. You only have to adapt the FMM configuration in the CPU when configuring the hardware variants. Here you have the following possibilities:
  - Modules from the target configuration can be divided in any order to the slots of the actual configuration.
  - Modules from the target configuration may be missing in the actual configuration.
  - Individual slots of the target configuration can be deactivated, on which modules are located in the actual configuration.
- FMM is a functionality of Yaskawa and is only supported by Yaskawa modules.
- By default, FMM is disabled. To use FMM mapping, you must enable the CPU parameter *'Free Module Mapping'*.
- For the *FMM* the mapping of the slots is to be specified via the record set 0x7F.
- For commissioning, you have to enable the parameter *'Startup when expected/actual configuration differs'* in your CPU.
- If FMM is activated and configured correctly, the system reacts as follows:
  - During start-up, no target/actual difference of the hardware is diagnosed.
  - Output data of missing modules are ignored and not output.
  - Input data of missing modules are set to 0.

4.8.4.2 FMM configuration

Configuration

- The mapping of the modules is defined as configuration by the 64byte record set 0x7F.
- The data record is retentively stored in the CPU.
- The record set must be transferred to the CPU by the user program by means of a write command.
- With the record set read command parts of the active configuration can be read. You have always to write the complete record set.
- Each written and valid configuration is only saved if a difference to the existing configuration exists.

Record set 0x7F

Record set 0x7F						
Byte	0	1	2	3	...	63
Mapping						

- The record 0x7F has a length of 64Byte, where Byte 0 ... 63 corresponds to the slot 1 ... 64 of the target configuration.
- For the FMM configuration, you must specify for each used slot of the target configuration at *'Mapping'* the corresponding value that corresponds to the actual configuration.

The following values can be entered at *Mapping*:

- 0 (0x00) - module is ignored
  – If modules of the target configuration are to be ignored, the value 0x00 must be used. In this way, gaps can be projected.
- 1 ... 64 (0x01 ... 0x40) - position of the module in the actual configuration
  – *'Mapping'* corresponds to the value of $Slot_{actual}$ i.e. the slot of the actual configuration on which the module of the target configuration is located.
- 255 (0xFF) - virtual module
  – If a module from the target configuration is missing, for Mapping the value 255 for "virtual module" is to be used.
  – Behaviour of a *virtual module*:
    - The input area always has the value 0, regardless of its size.
    - The writing to the output area has no effect.

Commissioning

The *target configuration* serves as template for the configuration of hardware variants.

1. Configure your system with a hardware configuration as target configuration and create your user program. The target configuration represents a superset of all available hardware variants.
2. Activate the parameter *'Free Module Mapping'* in your CPU.
3. Activate the parameter *'Startup when expected/actual configuration differs'* in your CPU.
4. Create the configuration by defining the deviation of the actual and target configuration for the current hardware configuration in record set 0x7F.
5. Transfer this record set via write command to your CPU.
   - For this use SFB 53 or SFB 58.
   - The address to be used is the diagnostic address of the CPU in the virtual IO device *'... SLIO CPU'*.

   ⇨ The configuration is permanently stored in the CPU and immediately active.

4.8.4.3 Examples

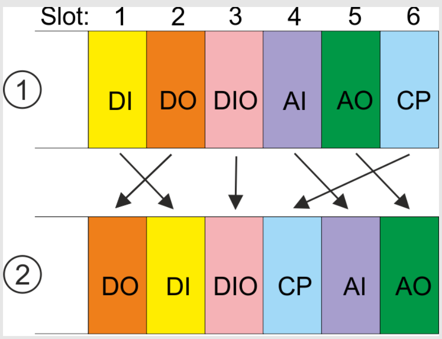
(1): Target configuration	$\text{Slot}_{\text{target}}$
	1
	2
	3
	4
	5
	6

$\text{Slot}_{\text{target}}$ - The mapping always refers to the slot of the target configuration.

Based on the target configuration, the following examples show how to determine the mapping values for the hardware variants.

4.8.4.3.1 Examples of hardware variants

Variant 1: Same type and number of modules but reversed slots

(1): Target configuration (2): Actual configuration	$\text{Slot}_{\text{target}}$	$\text{Slot}_{\text{actual}}$	Record set 0x7F	
			Byte	Mapping
	1	2	0	0x02
	2	1	1	0x01
	3	3	2	0x03
	4	5	3	0x05
	5	6	4	0x06
	6	4	5	0x04

Determination of *Mapping* values of record set 0x7F:

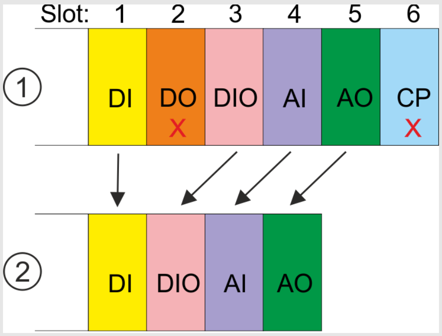
- Byte 0: The module of $Slot_{target}$ = 1 is in the actual configuration at $Slot_{actual}$ = 2 → Mapping = 0x02
- Byte 1: The module of $Slot_{target}$ = 2 is in the actual configuration at $Slot_{actual}$ = 1 → Mapping = 0x01
- Byte 2: The module of $Slot_{target}$ = 3 is in the actual configuration at $Slot_{actual}$ = 3 → Mapping = 0x03
- Byte 3: The module of $Slot_{target}$ = 4 is in the actual configuration at $Slot_{actual}$ = 5 → Mapping = 0x05
- Byte 4: The module of $Slot_{target}$ = 5 is in the actual configuration at $Slot_{actual}$ = 6 → Mapping = 0x06
- Byte 5: The module of $Slot_{target}$ = 6 is in the actual configuration at $Slot_{actual}$ = 4 → Mapping = 0x04

$\text{Slot}_{\text{target}}$ - The mapping always refers to the slot of the target configuration.

$\text{Slot}_{\text{actual}}$ - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 1, *Mapping* corresponds to $Slot_{actual}$, i.e. slot of the actual configuration on which the module of the target configuration is located.

Variant 2: Reversed slots and modules are missing

(1): Target configuration **(2): Actual configuration**	**Slot$_{target}$**	**Slot$_{actual}$**	**Record set 0x7F**	
			Byte	**Mapping**
	1	1	0	0x01
	2	-	1	0xFF
	3	2	2	0x02
	4	3	3	0x03
	5	4	4	0x04
	6	-	5	0xFF

Determination of *Mapping* values of record set 0x7F:

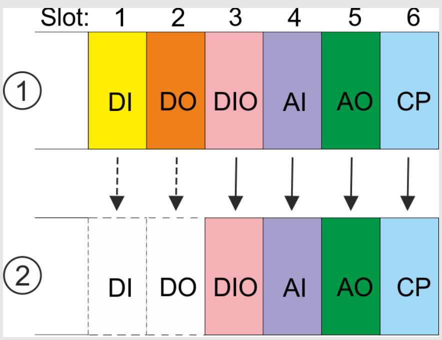
- Byte 0: The module of *Slot$_{target}$* = 1 is in the actual configuration at *Slot$_{actual}$* = 1 → Mapping = 0x01
- Byte 1: The module of *Slot$_{target}$* = 2 is not available in the actual configuration → Mapping = 0xFF
- Byte 2: The module of *Slot$_{target}$* = 3 is in the actual configuration at *Slot$_{actual}$* = 2 → Mapping = 0x02
- Byte 3: The module of *Slot$_{target}$* = 4 is in the actual configuration at *Slot$_{actual}$* = 3 → Mapping = 0x03
- Byte 4: The module of *Slot$_{target}$* = 5 is in the actual configuration at *Slot$_{actual}$* = 4 → Mapping = 0x04
- Byte 5: The module of *Slot$_{target}$* = 6 is not available in the actual configuration → Mapping = 0xFF

Slot$_{target}$ - The mapping always refers to the slot of the target configuration.

Slot$_{actual}$ - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 2, *Mapping* corresponds to the value of *Slot$_{actual}$* , i.e. slot of the actual configuration on which the module of the target configuration is located. If a module from the target configuration is missing, for *Mapping* the value 0xFF for "virtual module" is to be used.

Variant 3: Modules are ignored

(1): Target configuration (2): Actual configuration	$Slot_{target}$	$Slot_{actual}$	Record set 0x7F	
			Byte	**Mapping**
	1	empty	0	0x00
	2	empty	1	0x00
	3	3	2	0x03
	4	4	3	0x04
	5	5	4	0x05
	6	6	5	0x06

Determination of *Mapping* values of record set 0x7F:

- Byte 0: The module of $Slot_{target}$ = 1 is ignored in the actual configuration → Mapping = 0x00
- Byte 1: The module of $Slot_{target}$ = 2 is ignored in the actual configuration → Mapping = 0x00
- Byte 2: The module of $Slot_{target}$ = 3 is in the actual configuration at $Slot_{actual}$ = 3 → Mapping = 0x03
- Byte 3: The module of $Slot_{target}$ = 4 is in the actual configuration at $Slot_{actual}$ = 4 → Mapping = 0x04
- Byte 4: The module of $Slot_{target}$ = 5 is in the actual configuration at $Slot_{actual}$ = 5 → Mapping = 0x05
- Byte 5: The module of $Slot_{target}$ = 6 is in the actual configuration at $Slot_{actual}$ = 6 → Mapping = 0x06

$Slot_{target}$ - The mapping always refers to the slot of the target configuration.

$Slot_{actual}$ - Slot of the actual configuration on which the module of the target configuration is located.

Mapping - For variant 3, *Mapping* corresponds to the value of $Slot_{actual}$, i.e. slot of the actual configuration on which the module of the target configuration is located. If modules of the target configuration are to be ignored, for *Mapping* the value 0x00 is to be used.

The presence of gaps in the System SLIO is not allowed! But you can place modules and define them via the configuration as empty slot for the target hardware configuration.

4.9 Project transfer

Overview

There are the following possibilities for project transfer into the CPU:

- Transfer via MPI
- Transfer via Ethernet
- Transfer via memory card

4.9.1 Transfer via MPI

General

For transfer via MPI the CPU has the following interface:

'X3: MPI(PB) interface' page 54

'X2: PtP(MPI) interface' page 53

> *With an overall reset CPU the configuration via X2 PtP(MPI) is not possible!*

Net structure

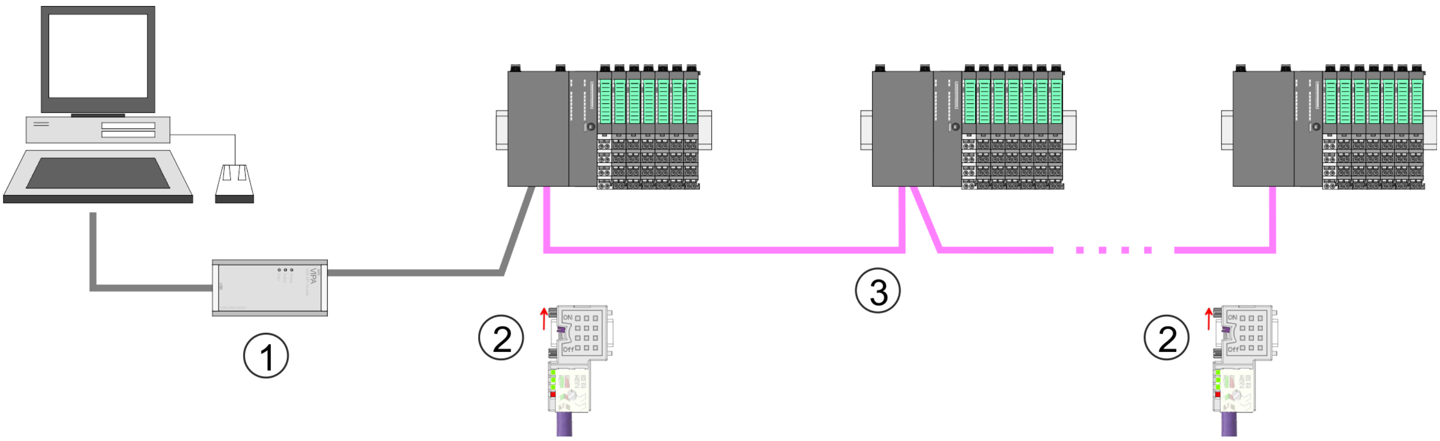
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs with 187.5kbaud. The CPUs are delivered with MPI address 2.

MPI programming cable

The MPI programming cables are available at Yaskawa in different variants. The cables provide a RS232 res. USB plug for the PC and a bus enabled RS485 plug for the CPU. Due to the RS485 connection you may plug the MPI programming cables directly to an already plugged plug on the RS485 jack. Every bus participant identifies itself at the bus with an unique address, in the course of the address 0 is reserved for programming devices.

Terminating resistor

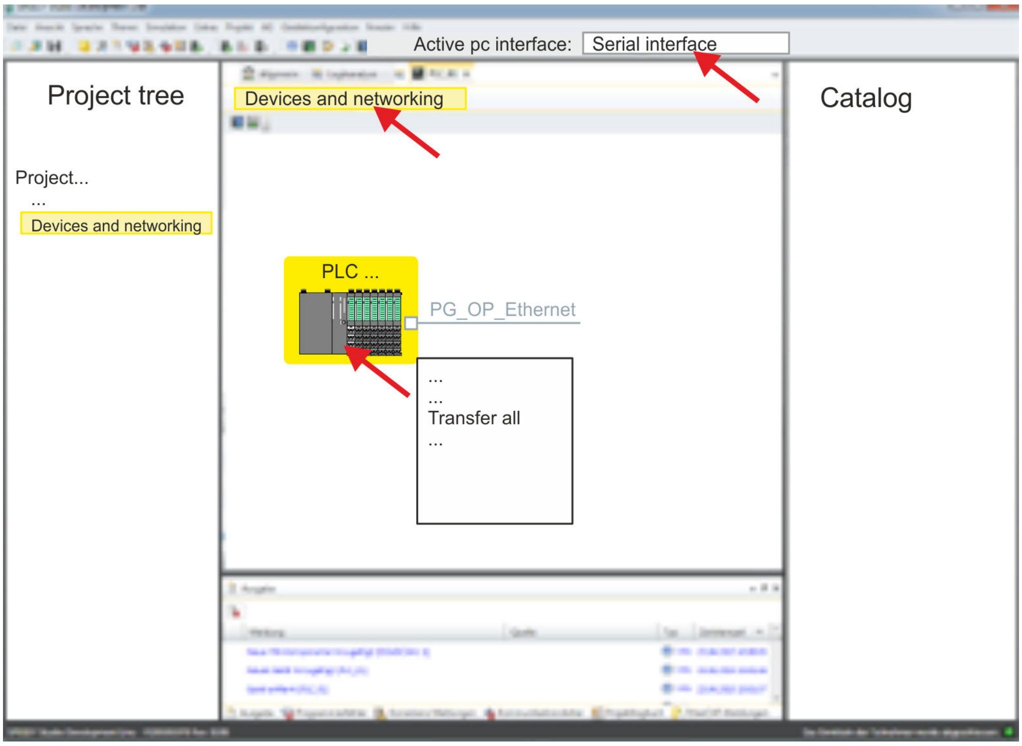
A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment. Please make sure that the participants with the activated terminating resistors are always power supplied. Otherwise it may cause interferences on the bus.



1 MPI programming cable
2 Activate the terminating resistor via switch
3 MPI network

Proceeding transfer via MPI

1. Connect your PC to the MPI jack of your CPU via a MPI programming cable.
2. Switch-ON the power supply of your CPU and start the *SPEED7 Studio* with your project.
3. Set at *'Active PC interface'* the "Serial interface".
4. Click in the *'Project tree'* to your project and select *'Context menu ➔ Recompile'*.
   ⇨ Your project will be translated and prepared for transmission.



5. To transfer the user program and hardware configuration click in the *Project tree* at your CPU and select *'Context menu → Transfer all'*.
   ⇨ A dialog window for project transfer opens
6. Select the *'Port type'* "Serial interface" and start the transfer with *'Transfer'*.
7. Confirm the request that the CPU is to be brought into the state STOP.
   ⇨ The user program and the hardware configuration are transferred via MPI to the CPU.
8. Close after transmission the dialog.
9. With *'Context menu → Copy RAM to ROM'* you can save your project on a memory card, if one is plugged.

4.9.2 Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X1/X5: Ethernet PG/OP channel

Initialization

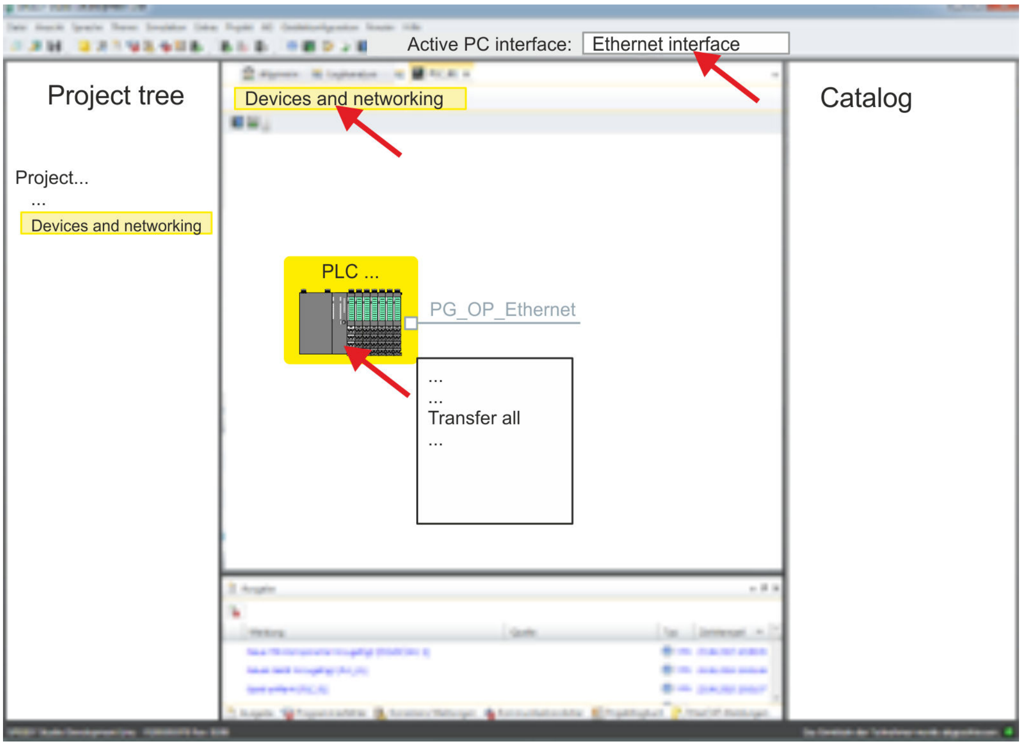
So that you may access the according Ethernet interface you have to assign IP address parameters by means of the "initialization".

- X1/X5: Ethernet PG/OP channel
  – *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 77*

Transfer

1. For the transfer, connect, if not already done, the appropriate Ethernet port to your Ethernet.
2. Switch-ON the power supply of your CPU and start the *SPEED7 Studio* with your project.
3. Set at *'Active PC interface'* the "Ethernet interface".

4. Click in the *'Project tree'* to your project and select *'Context menu ➔ Recompile'*.
   ⇨ Your project will be translated and prepared for transmission.



5. To transfer the user program and hardware configuration click in the *Project tree* at your CPU and select *'Context menu ➔ Transfer all'*.
   ⇨ A dialog window for project transfer opens
6. Select the *'Port type'* "Ethernet interface" and start the transfer with *'Transfer'*.
7. Confirm the request that the CPU is to be brought into the state STOP.
   ⇨ The user program and the hardware configuration are transferred via Ethernet to the CPU.
8. Close after transmission the dialog.
9. With *'Context menu ➔ Copy RAM to ROM'* you can save your project on a memory card, if one is plugged.

4.9.3 Transfer via memory card

Proceeding transfer via memory card

The memory card serves as external storage medium. There may be stored several projects and sub-directories on a memory card. Please regard that your current project is stored in the root directory and has one of the following file names:

- S7PROG.WLD
- AUTOLOAD.WLD

1. Start the *SPEED7 Studio* with your project.
2. Click in the *'Project tree'* at the CPU.
3. Create in the *SPEED7 Studio* with *'Context menu ➔ Export device configuration (WLD)'* a wld file.
   ⇨ The wld file is created. This contains the user program and the hardware configuration

4. Copy the wld file at a suited memory card. Plug this into your CPU and start it again.

⇨ The transfer of the application program from the memory card into the CPU takes place depending on the file name after an overall reset or PowerON.

S7PROG.WLD is read from the memory card after overall reset.

AUTOLOAD.WLD is read from the memory card after PowerON.

The blinking of the SD LED of the CPU marks the active transfer. Please regard that your user memory serves for enough space for your user program, otherwise your user program is not completely loaded and the SF LED gets on.

4.10 Accessing the web server

Overview

The CPU has a web server integrated. This offers the following access:

- via the Ethernet PG/OP channel
  - Device web page of the CPU
  - CPU *OPC UA* project
  - CPU *WebVisu* project
- via Ethernet CP
  - Device web page of the Ethernet CP
  - Ethernet CP *OPC UA* project
  - Ethernet CP *WebVisu* project

↳ *Chap. 5 'Deployment OPC UA' page 129*

↳ *Chap. 6 'Deployment WebVisu - Web visualization' page 156*

4.10.1 Device web page CPU

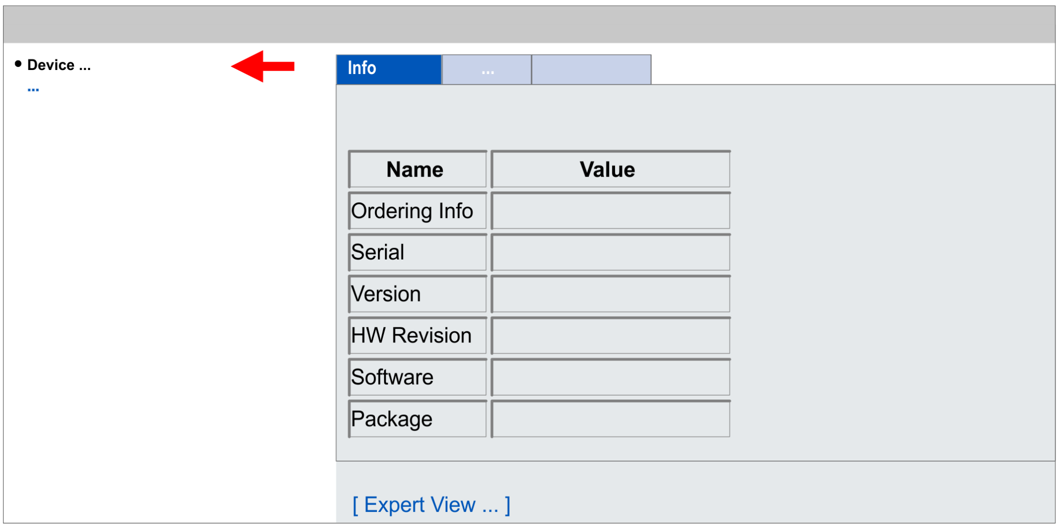
Overview

- Dynamic web page, which exclusively outputs information.
- On the *device web page* you will find information about your CPU, the connected modules and your *WebVisu* project.
- The shown values cannot be changed.
- Access is via the IP address of the Ethernet PG/OP channel.
  ↳ *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 77*
- You can access the IP address with a web browser.

> *It is assumed that there is a connection between PC and CPU with web browser via the Ethernet PG/OP channel. This may be tested by Ping to the IP address of the Ethernet PG/OP channel.*

4.10.1.1 Web page with selected CPU

Tab: *'Info'*



Name	Value	
Ordering Info	015-CEFNR00	Order number of the CPU
Serial	...	Serial number of the CPU
Version	03V...	Version number of the CPU
HW Revision	03	CPU hardware revision
Software	3.0.5	CPU firmware version
Package	Pb000328.pkb	File name for the firmware update

[Expert View] takes you to the advanced "Expert View".

Runtime Information		CPU
Operation Mode	RUN	Operating mode
Mode Switch	RUNP	
System Time	14.03.19 16:09:31:262	Date, time
Up Time	0 days 02 hrs 07 min 08 sec	Time of changing the operating mode
Last Change to RUN	n/a	
Last Change to STOP	29.03.17 16:09:03:494	
OB1-Cycle Time	cur = 0us, min = 0us, max = 0us, avg = 0us	Cyclic time: min = minimum cur = current max = maximum avg = average

Interface Information			Interfaces
X1	PG/OP Ethernet Port 1	Address 2025..2040	Ethernet PG/OP channel port 1
X2	PTP		PtP: Point-to-point operation or DPM: DP master mode
X3	MPI	Address 2047	Operating mode RS485 MPI: MPI operation
X4	EtherCAT Port	Address 2045	EtherCAT interface
X5	PG/OP Ethernet Port 2	Address 2025..2040	Ethernet PG/OP channel port 2
X6	Ethernet Port	Address 2044	CP interface

Card Information		Memory card
Type	SD	Memory card information
Manufacturer ID	9	
Application ID	16720	
Card Name	AF SD	
Card Revision	16	
Card S/N	64C34010	

Card Information		Memory card
Manufacture Month	8	
Manufacture Year	2013	
Size	470.73 MByte	
Free	454.70 MByte	

VSC Information		
VSC Product Number	955-C0ME040	Information about VSC
VSC Product S/N	00007807	
Memory Extension	256 kByte	
Profibus	not activated	
Motion	4 Axes	

Active Feature Set Information		
Status	Media present	Information about enabled functions
VSC Product Number	955-C0ME040	
VSC Product S/N	00007807	
Memory Extension	256 kByte	
Profibus	not activated	
Motion	4 Axes	

Memory Usage				CPU
	free	used	max	Memory configuration information Load memory, working memory (code/data)
LoadMem	512.0 kByte	0 byte	512.0 kByte	
WorkMemCode	256.0 kByte	0 byte	256.0 kByte	
WorkMemData	256.0 kByte	0 byte	256.0 kByte	

PG/OP Network Information		Ethernet PG/OP channel
Device Name	Onboard PG/OP	Name
IP Address	172.20.139.76	Address information
Subnet Mask	255.255.255.0	
Gateway Address	172.20.139.76	
MAC Address	00:20:D5:02:6C:27	
Link Mode X1	100 Mbps - Full Duplex	Link status and speed
Link Mode X5	Not Available	

CP Network Information (According To Project Settings)		NET-CP
Device Name	n/a	Name
IP Address	0.0.0.0	Address information
Subnet Mask	0.0.0.0	
Gateway Address	0.0.0.0	

CP Firmware Information		EtherCAT
Bx000689	V3.0.0.32	Support information
PRODUCT	... EtherCAT-CP V3.4.3 Px000324.pkg	Name, firmware version, package
ExtSvnRev.txt	V128.0.0.0	Support information
MX000337	V0.0.1.0	
Diagnosis Address	2046	Diagnosis Address

CPU Firmware Information		CPU
File System	V1.0.2	Name, firmware version, package
PRODUCT	... 015-CEFNR00 V3.0.5 Pb000328.pkb	
HARDWARE	V0.1.0.0 5841L-V10 MX000311.003	Support information
BOOTLOADER	Bx000645 V125	
Bx000501	V2.2.5.0	
Ax000136	V1.0.6.0	
Ax000150	V1.1.4.0	
fx000018.wld	V1.0.2.0	
syslibex.wld	n/a	
Protect.wld	n/a	

ARM Processor Load		CPU
Measurement Cycle Time	100 ms	Support information
Last Value	5%	
Average Of Last 10 Values	5%	
Minimum Load	5%	
Maximum Load	16%	

Tab: *'IP'*

Here the IP address data of your Ethernet PG/OP channel are shown.

Tab: *'Firmware'*

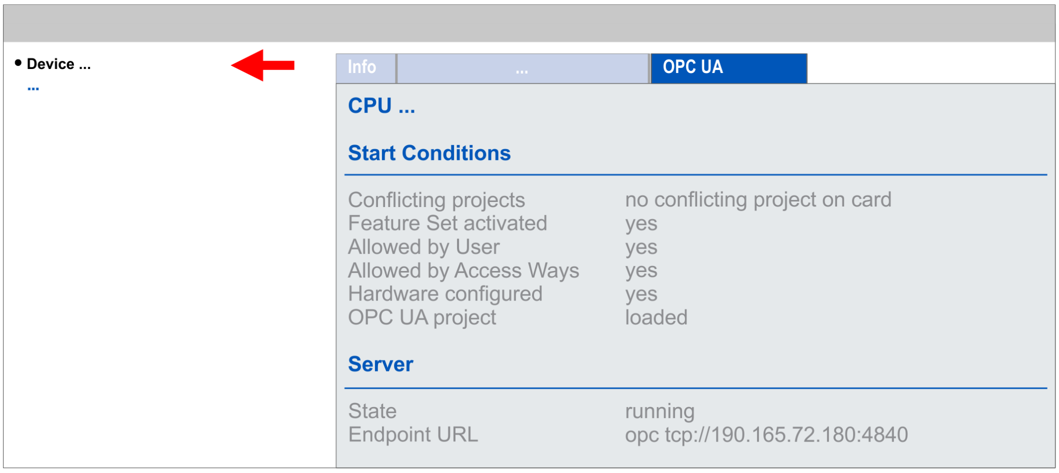
As of CPU firmware version V2.6.0 , you can transfer the firmware file online to the CPU via the *'Firmware'* tab. The firmware update in the CPU is triggered by means of the operating mode switch. *Chap. 4.13 'Firmware update' page 116*

Tab: *'Access Ways'*

As of CPU firmware version V2.6.0 , information about access settings is shown here. In the delivery state, there are no restrictions. You can specify access to interfaces, ports and protocols via the parametrization. *80*

4.10.1.1.1 Tab: *‘OPC UA’*

As of CPU firmware version V3.0.0, information about the *OPC UA* project is shown here. *Chap. 5 ‘Deployment OPC UA’ page 129*



For your CPU can process a OPC UA project, you have to activate the OPC UA functionality. *Chap. 5.3 ‘Activate OPC UA functionality’ page 138*

Start Conditions

Here the start conditions for the *OPC UA* server are listed:

- Conflicting projects
  – Simultaneous use of an *OPC UA* and *WebVisu* project via the same interface is not permitted and results in the message *‘conflicting project on card’*.
  – The status *‘no conflicting project on card’* indicates that there is no conflict with a *WebVisu* project on the same interface.
- Feature Set activated
  – yes: The *OPC UA* functionality is activated.
  – no: The *OPC UA* functionality is not activated.
  – *Chap. 5.3 ‘Activate OPC UA functionality’ page 138*
- Allowed by User
  – yes: The *OPC UA* server is activated and access to it is allowed. As soon as an *OPC UA* project is found on the memory card, it is automatically started and enabled for access.
  – no: You can disable and stop the *OPC UA* server by means of the CMD - auto command *‘OPCUA_PGOP_DISABLE’*. With *‘OPCUA_PGOP_ENABLE’* you can enable to restart the *OPC UA* server.
  – *Chap. 4.17 ‘CMD - auto commands’ page 124*
- Allowed by Access Ways
  – yes: By default, the *OPC UA* protocol is enabled.
  – no: The *OPC UA* protocol is disabled. You can deactivate or activate the *OPC UA* protocol by means of the parameter *‘OPC UA’* in the *‘Access settings’*.

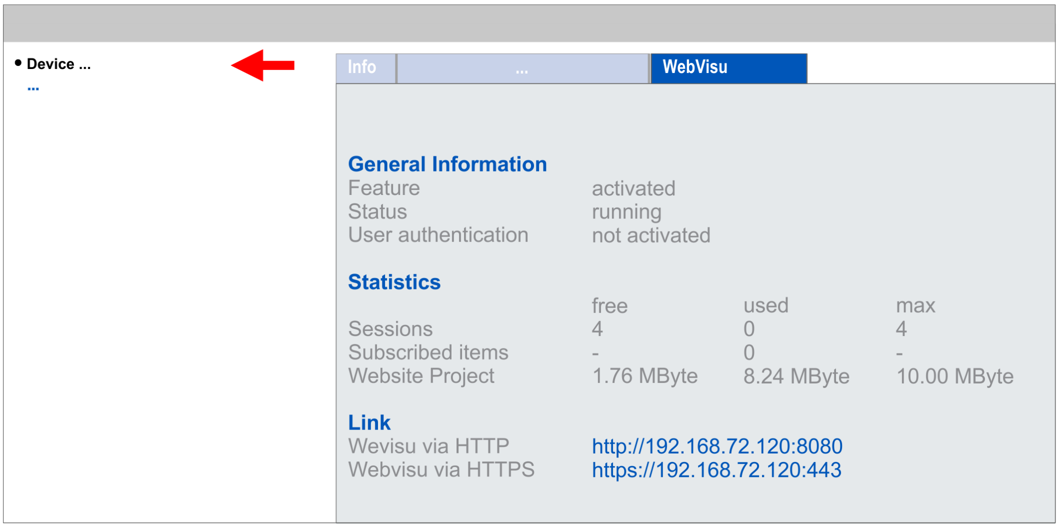
- Hardware configured
  - yes: A hardware configuration as a precondition for the *OPC UA* communication is loaded. The hardware configuration is checked for validity elsewhere.
  - no: A hardware configuration is not loaded e.g. after an overall reset.
  - *Chap. 4.5 'Hardware configuration - CPU' page 75*
- OPC UA project
  - loaded: An *OPC UA*OPC UA project is loaded.
  - not loaded: An *OPC UA*OPC UA project is not loaded.
  - *Chap. 5 'Deployment OPC UA' page 129*

Server

- State
  - Running: The start conditions are fulfilled and the *OPC UA* server is started.
  - Stopped: The *OPC UA* server is stopped.
  - Startup failure: The *OPC UA* server can not be started.
  - Starting: The *OPC UA* server currently starts up.
  - Stopping: The *OPC UA* server currently stopps.
- Endpoint URL
  - As soon as the *OPC UA* server is started, the endpoint URL of the *OPC UA* server is listed here.

4.10.1.1.2 Tab: *'WebVisu'*

Information about the web visualization (*'WebVisu'*) are shown here. The creation of a *'WebVisu'* project is only possible with the *SPEED7 Studio* V1.7.0 and up. *Chap. 6 'Deployment WebVisu - Web visualization' page 156*



For your CPU can process a WebVisu project, you have to activate the WebVisu functionality. *Chap. 6.2 'Activate WebVisu functionality' page 159*

General Information

- Feature
  - activated: The *WebVisu* functionality is activated.
  - not activated: The *WebVisu* functionality is not activated.
- Status
  - The status of your *WebVisu* project is shown here. *Chap. 4.10.1.1.2.1 'Status of the WebVisu' page 104*
- User authentication
  - activated: User authentication is activated. Access to the *WebVisu* happens via a login by user name and password.
  - not activated: User authentication is de-activated. Access to the *WebVisu* is unsecured.

Statistics

Statistical information about your *WebVisu* project are shown here.

- Sessions: Number of sessions, i.e. online connections to this *WebVisu* project. A session corresponds to an open window or tab in a web browser.
  - free: Number of sessions still possible.
  - used: Number of active sessions. For the number of active sessions, it is not relevant whether the sessions were started by the same or different users.
  - max.: Number of sessions still possible. The maximum number of sessions is device specific and specified in the technical data.
- Subscribed items: Number of variables.
  - free: Here nothing is shown.
  - used: Number of variables used.
  - max.: Here nothing is shown.
- WebVisu Project: Information on the memory allocation for the *WebVisu* project.
  - free: Still free space for the *WebVisu* project.
  - used: Size of the current *WebVisu* project.
  - max.: Maximum available space for a *WebVisu* project.

Link

In *Status 'running'* the links to access your *WebVisu* are listed here.

Status of the *WebVisu*

On the device web page at the tab *'WebVisu'* via *'Status'* you get the status of your *WebVisu* project.

Status	Meaning
running	*WebVisu* is active / has started-up and can be opened
loading webvisu project	Loading *WebVisu* project
shutting down	*WebVisu* server shuts down
stop requested	*WebVisu* STOP requested
stopped	*WebVisu* server is down
webvisu feature not activated	*WebVisu* not activated, memory card is not inserted
webvisu is disabled by the user	*WebVisu* was disabled by the user
no webvisu project file found	No *WebVisu* project found
no hardware configuration loaded	No hardware configuration is loaded

Status	Meaning
invalid configuration	Invalid *WebVisu* configuration
internal error: filesystem	Error initializing the file system
webvisu project file too large	Error loading *WebVisu* project, project file too large
loading webvisu project file	Error loading *WebVisu* project, project file may be damaged
deleting webvisu project	Failed to delete the *WebVisu* project
internal error: file system - delete	*WebVisu* project to be deleted was not found in the memory
CRC mismatch	CRC of the *WebVisu* project file is not correct
webvisu stopped	*WebVisu* server has terminated unexpectedly
internal error 1	Internal error - initialization failed step 1
internal error 2	Internal error - initialization failed step 2
internal error 3	Unexpected internal error
unknown error	General error

4.10.1.1.3 Tab: *'Port Mirroring'*

Overview

- *Port Mirroring* offers the possibility to diagnose the communication without additional hardware effort.
- The Ethernet PG/OP interface is designed as switch.
- When *Port Mirroring* of the PG/OP2: X5 (Mirror Port) interface is activated, all telegrams received and sent via the PG/OP1: X1 interface are mirrored to the PG/OP2: X5 interface and vice versa.
- When *Port Mirroring* is activated, for diagnostics with diagnostics software such as Wireshark, you can connect your PC directly to the 2. interface.
- The next power-cycle will automatically disable *Port Mirroring*.

Enable Port Mirroring

When enabled, you can set the parameters for *Port Mirroring*.

PG/OP1: X1

When enabled the frames of PG/OP2: X5 are mirrored to PG/OP1: X1.

PG/OP2: X5

When enabled the frames of PG/OP1: X1 are mirrored to PG/OP2: X5.

Disable communication on the Mirror Port

When enabled additional communication via the mirrored interface (Mirror Port) is prevented.

Save

With *Save*, the setting are taken and activated. The next power-cycle will automatically disable *Port Mirroring*.

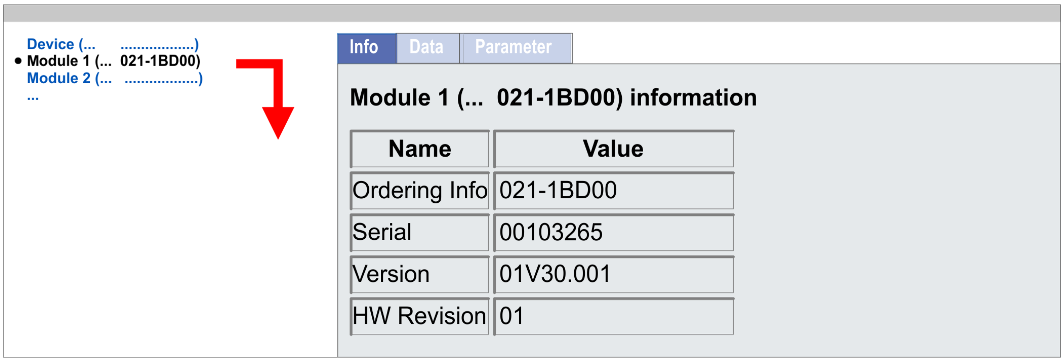
4.10.1.2 Web page with selected module

Structure

The web page is built dynamically and depends on the number of modules, which are connected to the CPU. The web page only shows information. The shown values cannot be changed

Please consider the System SLIO power and clamp modules do not have any module ID. These may not be recognized by the CPU and so are not listed and considered during slot allocation.

Module



Tab: *'Info'*

Here product name, order number, serial number, firmware version and hardware state number of the according module are listed.

Tab: *'Data'*

Here the address and the state of the inputs respectively outputs are listed. Please note with the outputs that here exclusively the states of outputs can be shown, which are within the OB 1 process image.

Tab: *'Parameter'*

With parametrizable modules e.g. analog modules the parameter setting is shown here. These come from the hardware configuration.

4.10.2 Device web page Ethernet CP

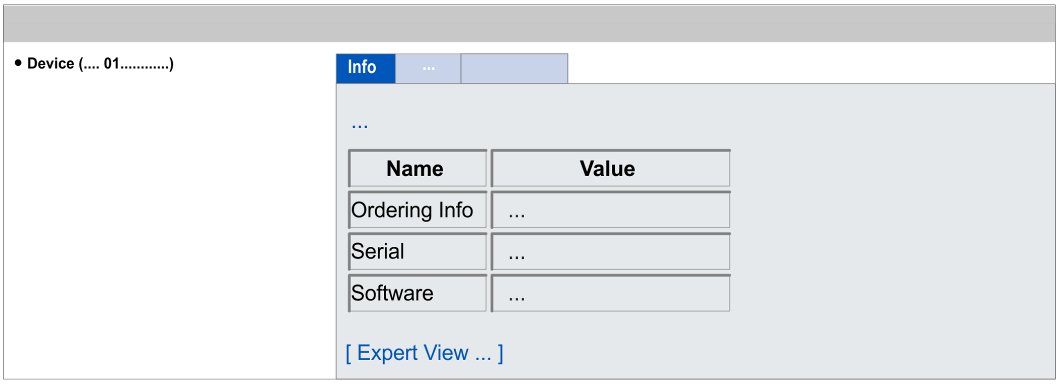
Overview

- Dynamic web page, which exclusively outputs information.
- On the *device web page* you will find information about:
  – Ethernet CP
  – *OPC UA* project
  – *WebVisu* project
- The shown values cannot be changed.
- Access is via the IP address of the Ethernet CP.
  Chap. 8.8 'Commissioning and initialization' page 180
- You can access the IP address with a web browser.

It is assumed that there is a connection between PC and Ethernet CP with web browser via the Ethernet interface. This may be tested by Ping to the IP address of the Ethernet CP.

4.10.2.1 Web page with selected CP

Tab: Info



015-CEFNR00 CP - Information

Name	Value	
Ordering number	015-CEFNR00	Order number of the CPU
Serial	...	Serial number of the CPU
Software	V3.4.3	CP firmware version

[Expert View] takes you to the advanced "Expert View".

Runtime Information		**Ethernet CP**
System Date/Time	Mon Nov 27 07:55:34 2017	Date, time

Network information Port X4		**Interface X4**
Link Mode	No Link	Connection information

Network information Port X6		**Interface X6**
Link Mode	100 Mbps - Full Duplex	Connection information

Hardware Information		**Ethernet CP**
Vendor ID	0x022B	Support information
Device ID	0x0101	
MX-File	MX000335.002	
Semi-Product-Number	5836D-V10	
Slot number	2	

Boot Loader Information		**Ethernet-CP**
Firmware	Bx000644	Support information
Firmware Version	V125	

Firmware Information		Ethernet CP
Package File Name	Px000324.pkg	Package version
Firmware	Bb000714	Firmware name
Firmware Version	3.4.3	Firmware version

ARM Processor Load		Ethernet CP
Measurement Cycle Time	100 ms	Support information
Last Value	4%	
Average Of Last 10 Values	3%	
Minimum Load	3%	
Maximum Load	100	

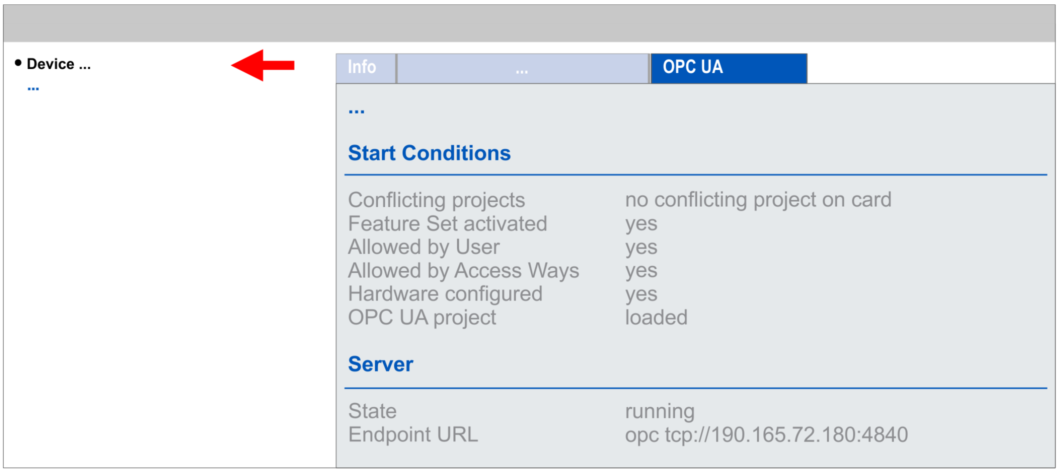
Statistic Port X4		Interface X4
Number Rx bytes received	292526	Statistic data for data to be sent and received
Number Rx frames received	4275	
Number Rx overrun errors	0	
Number Rx CRC errors	0	
Total number of Rx errors	0	
Number Tx bytes sent	292974	
Number Tx frames sent	4266	
Number Tx underrun errors	0	
Number Tx collision errors	0	
Total number of Tx errors	0	

Statistic Port X6		Interface X6
Number Rx bytes received	2580	Statistic data for data to be sent and received
Number Rx frames received	22	
Number Rx overrun errors	0	
Number Rx CRC errors	0	
Total number of Rx errors	0	
Number Tx bytes sent	11983	
Number Tx frames sent	33	
Number Tx underrun errors	0	
Number Tx collision errors	0	
Total number of Tx errors	0	

Tab: *'IP'* Here the IP address data of your Ethernet CP are shown.

4.10.2.1.1 Tab: *'OPC UA'*

As of CPU firmware version V3.0.0, information about the *OPC UA* project is shown here. *Chap. 5 'Deployment OPC UA' page 129*



For your CPU can process a OPC UA project, you have to activate the OPC UA functionality. *Chap. 5.3 'Activate OPC UA functionality' page 138*

Start Conditions

Here the start conditions for the *OPC UA* server are listed:

- Conflicting projects
  - Simultaneous use of an *OPC UA* and *WebVisu* project via the same interface is not permitted and results in the message *'conflicting project on card'*.
  - The status *'no conflicting project on card'* indicates that there is no conflict with a *WebVisu* project on the same interface.
- Feature Set activated
  - yes: The *OPC UA* functionality is activated.
  - no: The *OPC UA* functionality is not activated.
  - *Chap. 5.3 'Activate OPC UA functionality' page 138*
- Allowed by User
  - yes: The *OPC UA* server is activated and access to it is allowed. As soon as an *OPC UA* project is found on the memory card, it is automatically started and enabled for access.
  - no: You can disable and stop the *OPC UA* server by means of the CMD - auto command *'OPCUA_PGOP_DISABLE'*. With *'OPCUA_PGOP_ENABLE'* you can enable to restart the *OPC UA* server.
  - *Chap. 4.17 'CMD - auto commands' page 124*
- Allowed by Access Ways
  - yes: By default, the *OPC UA* protocol is enabled.
  - no: The *OPC UA* protocol is disabled. You can deactivate or activate the *OPC UA* protocol by means of the parameter *'OPC UA'* in the *'Access settings'*.

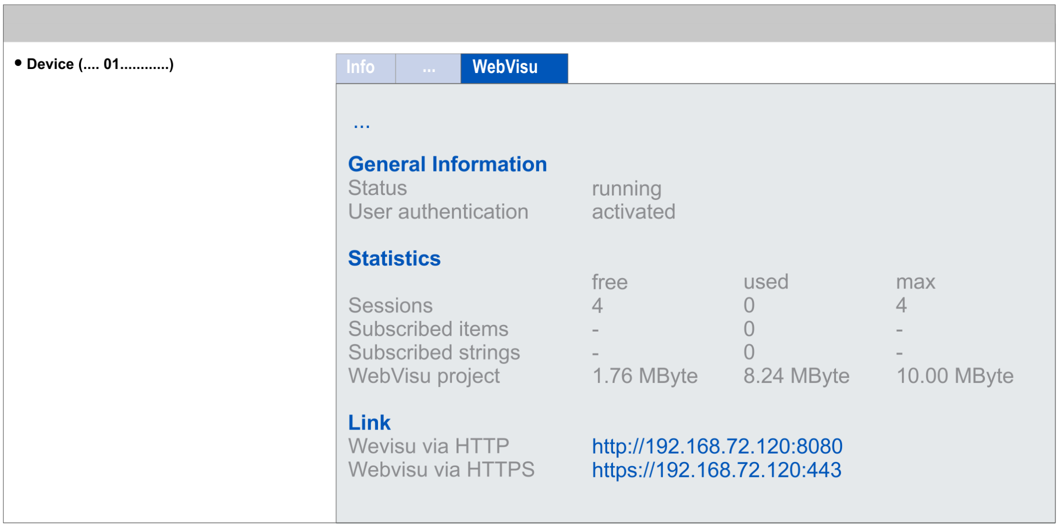
- Hardware configured
  - yes: A hardware configuration as a precondition for the *OPC UA* communication is loaded. The hardware configuration is checked for validity elsewhere.
  - no: A hardware configuration is not loaded e.g. after an overall reset.
  - *Chap. 4.5 ‘Hardware configuration - CPU’ page 75*
- OPC UA project
  - loaded: An *OPC UA*OPC UA project is loaded.
  - not loaded: An *OPC UA*OPC UA project is not loaded.
  - *Chap. 5 ‘Deployment OPC UA’ page 129*

Server

- State
  - Running: The start conditions are fulfilled and the *OPC UA* server is started.
  - Stopped: The *OPC UA* server is stopped.
  - Startup failure: The *OPC UA* server can not be started.
  - Starting: The *OPC UA* server currently starts up.
  - Stopping: The *OPC UA* server currently stopps.
- Endpoint URL
  - As soon as the *OPC UA* server is started, the endpoint URL of the *OPC UA* server is listed here.

4.10.2.1.2 Tab: *‘WebVisu’*

Information about the web visualization (*‘WebVisu’*) of the Ethernet CP are shown here. The creation of a *‘WebVisu’* project for the CP is only possible with the *SPEED7 Studio* V1.7.0 and up.



For your CPU can process a WebVisu project, you have to activate the WebVisu functionality. *Chap. 6.2 ‘Activate WebVisu functionality’ page 159*

General Information

- Status
  – The status of your *WebVisu* project is shown here. ↳ *Chap. 4.10.1.1.2.1 'Status of the WebVisu' page 104*
- User authentication
  – activated: User authentication is activated. Access to the *WebVisu* happens via a login by user name and password.
  – not activated: User authentication is de-activated. Access to the *WebVisu* is unsecured.

Statistics

Statistical information about your *WebVisu* project are shown here.

- Sessions: Number of sessions, i.e. online connections to this *WebVisu* project. A session corresponds to an open window or tab in a web browser.
  – free: Number of sessions still possible.
  – used: Number of active sessions. For the number of active sessions, it is not relevant whether the sessions were started by the same or different users.
  – max.: Number of sessions still possible. The maximum number of sessions is device specific and specified in the technical data.
- Subscribed items: Number of variables including strings.
  – free: Here nothing is shown.
  – used: Number of variables used.
  – max.: Here nothing is shown.
- Subscribed strings: Number of strings or character chains.
  – free: Here nothing is shown.
  – used: Number of strings used.
  – max.: Here nothing is shown.
- WebVisu Project: Information on the memory allocation for the *WebVisu* project.
  – free: Still free space for the *WebVisu* project.
  – used: Size of the current *WebVisu* project.
  – max.: Maximum available space for a *WebVisu* project.

Link

In *Status 'running'* the links to access your *WebVisu* are listed here.

4.11 Operating modes

4.11.1 Overview

The CPU can be in one of 4 operating modes:

- Operating mode STOP
- Operating mode START-UP
  (OB 100 - restart / OB 102 - cold start *)
- Operating mode RUN
- Operating mode HOLD

Certain conditions in the operating modes START-UP and RUN require a specific reaction from the system program. In this case the application interface is often provided by a call to an organization block that was included specifically for this event.

Operating mode STOP

- The application program is not processed.
- If there has been a processing before, the values of counters, timers, flags and the process image are retained during the transition to the STOP mode.
- Command output disable (BASP) is activated this means the all digital outputs are disabled.
- RUN-LED off
- STOP-LED on

Operating mode START-UP

- During the transition from STOP to RUN a call is issued to the start-up organization block OB 100.
  - The processing time for this OB is not monitored.
  - The START-UP OB may issue calls to other blocks.
- All digital outputs are disabled during the START-UP, this means BASP is activated.
- RUN LED
  - The RUN LED blinks as soon as the OB 100 is operated and for at least 3s, even if the start-up time is shorter or the CPU gets to STOP due to an error.
  - This indicates the start-up.
- STOP LED
  - During the processing of the OB 100 the STOP LED is on and then turns off.
- When the CPU has completed the START-UP OB, it assumes the operating mode RUN.

> **** OB 102 (Cold start)***
>
> *If there is a "Watchdog" error the CPU still remains in STOP state. With such an error the CPU must be manually started again. For this the OB 102 (cold start) must exist. The CPU will not go to RUN without the OB 102. Alternatively you can bring your CPU in RUN state again by an overall reset respectively by reloading your project.*
>
> *Please consider that the OB 102 (cold start) may exclusively be used for treatment of a watchdog error.*

Operating mode RUN

- The application program in OB 1 is processed in a cycle. Under the control of alarms other program sections can be included in the cycle.
- All timers and counters being started by the program are active and the process image is updated with every cycle.
- BASP is deactivated, i.e. all outputs are enabled.
- RUN-LED on
- STOP-LED off

Operating mode HOLD

The CPU offers up to 3 breakpoints to be defined for program diagnosis. Setting and deletion of breakpoints happens in your programming environment. As soon as a breakpoint is reached, you may process your program step by step.

Precondition

For the usage of breakpoints, the following preconditions have to be fulfilled:

- Testing in single step mode is possible with STL. If necessary switch the view via *'View ➔ STL'* to STL.
- The block must be opened online and must not be protected.

Approach for working with breakpoints

1. Activate *'View ➔ Breakpoint Bar'*.
2. Set the cursor to the command line where you want to insert a breakpoint.
3. Set the breakpoint with *'Debug ➔ Set Breakpoint'*.
   ⇨ The according command line is marked with a circle.
4. To activate the breakpoint click on *'Debug ➔ Breakpoints Active'*.
   ⇨ The circle is changed to a filled circle.
5. Bring your CPU into RUN.
   ⇨ When the program reaches the breakpoint, your CPU switches to the state HOLD, the breakpoint is marked with an arrow and the register contents are monitored.
6. Now you may execute the program code step by step via *'Debug ➔ Execute Next Statement'* or run the program until the next breakpoint via *'Debug ➔ Resume'*.
7. Delete (all) breakpoints with the option *'Debug ➔ Delete All Breakpoints'*.

Behavior in operating state HOLD

- The RUN-LED blinks and the STOP-LED is on.
- The execution of the code is stopped. No level is further executed.
- All times are frozen.
- The real-time clock runs is just running.
- The outputs were disabled (BASP is activated).
- Configured CP connections remain exist.

The usage of breakpoints is always possible. Switching to the operating mode test operation is not necessary.

With more than 2 breakpoints, a single step execution is not possible.

4.11.2 Function security

The CPUs include security mechanisms like a Watchdog (100ms) and a parameterizable cycle time surveillance (parameterizable min. 1ms) that stop res. execute a RESET at the CPU in case of an error and set it into a defined STOP state. The CPUs are developed function secure and have the following system properties:

Event	concerns	Effect
RUN → STOP	general	BASP (**B**efehls-**A**usgabe-**Sp**erre, i.e. command output disable) is set.
	central digital outputs	The outputs are disabled.
	central analog outputs	The outputs are disabled. ■ Voltage outputs issue 0V ■ Current outputs 0...20mA issue 0mA ■ Current outputs 4...20mA issue 4mA If configured also substitute values may be issued.
	decentral outputs	Same behaviour as the central digital/analog outputs.
	decentral inputs	The inputs are cyclically be read by the decentralized station and the recent values are put at disposal.
STOP → RUN res. PowerON	general	First the PII is deleted, then OB 100 is called. After the execution of the OB, the BASP is reset and the cycle starts with: Delete PIO → Read PII → OB 1.
	decentral inputs	The inputs are be read by the decentralized station and the recent values are put at disposal.
RUN	general	The program is cyclically executed: Read PII → OB 1 → Write PIO.

PII = Process image inputs

PIO = Process image outputs

4.12 Overall reset

Overview

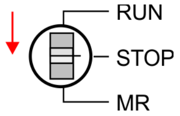
During the overall reset the entire user memory is erased. Data located in the memory card is not affected. You have 2 options to initiate an overall reset:

- Overall reset by means of the operating mode switch
- Overall reset by means of the *SPEED7 Studio*

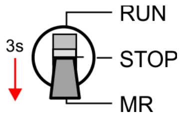
> *You should always establish an overall reset to your CPU before loading an application program into your CPU to ensure that all blocks have been cleared from the CPU.*

Overall reset by means of the operating mode switch

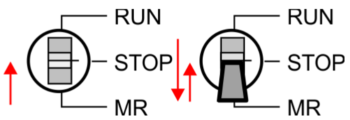
Proceeding



1. Your CPU must be in STOP mode. For this switch the operating mode switch of the CPU to STOP.
   ⇨ The ST LED is on.



2. Switch the operating mode switch to MR position for about 3 seconds.
   ⇨ The ST LED changes from blinking to permanently on.



3. Place the operating mode switch in the position STOP and switch it to MR and quickly back to STOP within a period of less than 3 seconds.
   ⇨ The overall reset is carried out. Here the ST LED flashes.
4. The overall reset has been completed when the ST LED is permanently on.

Overall reset via *SPEED7 Studio*

For the following proceeding you must be online connected to your CPU.

1. For an overall reset the CPU must be switched to STOP state. For this enable with *'View ➔ CPU control centre'* the *CPU control centre* and there switch your CPU to STOP state.
2. Request an overall reset via the *CPU control centre* or with *'Device ➔ Clear memory'*.
   ⇨ A dialog window opens. Here you can bring your CPU in STOP state, if not already done, and start the overall reset. During the overall reset procedure the STOP-LED flashes. When the STOP-LED is on permanently the overall reset procedure has been completed.

Activating functionality by means of a VSC

If there is a VSC of Yaskawa plugged, after an overall reset the according functionality is automatically activated. *'VSD' page 121* *Chap. 4.15 'Deployment storage media - VSD, VSC' page 121*

Automatic reload

If there is a project S7PROG.WLD on the memory card, after an overall reset the CPU attempts to reload this project from the memory card. → The SD LED is on. When the reload has been completed the LED expires. The operating mode of the CPU will be STOP respectively RUN, depending on the position of the operating mode switch.

Reset to factory setting

The *Reset to factory setting* deletes completely the internal RAM of the CPU and resets this to delivery state. Please regard that the MPI address is also set back to default 2! *Chap. 4.14 'Reset to factory settings' page 120*

4.13 Firmware update

Overview

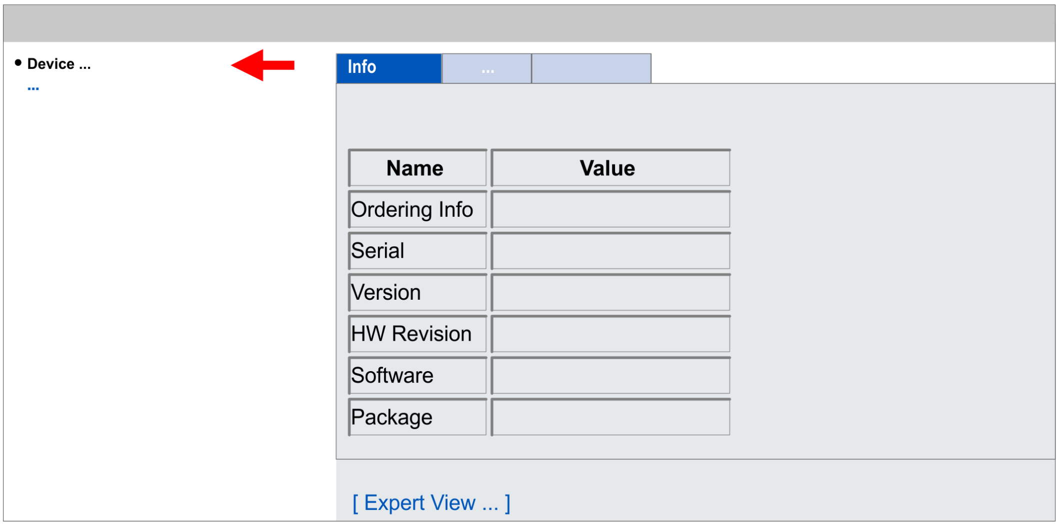
You can find the current firmware versions in the *'Download Center'* of www.yaskawa.eu.com under *'Firmware 015-CEFNR00'*. There are the following possibilities for the firmware update:

- Firmware update online - from FW V2.6.0 ↪ *Chap. 4.13.1 'Firmware update online' page 117*
  – Transfer of the firmware file to the CPU via the CPU web page.
  – Triggering the firmware update by means of the operating mode switch.
- Firmware update via memory card ↪ *Chap. 4.13.2 'Firmware update via memory card' page 118*
  – Transfer of the firmware file to a memory card.
  – The identification of a firmware file on the memory card takes place by means of a defined naming convention.
  – After PowerON and operating mode switch in the STOP position, the firmware update can be triggered by means of operating mode switch.

Show the firmware version via web page

The CPU has an integrated *Device web page* that also shows information about the firmware version via *'Info'*. Here you will also find information about the required firmware *'Package'*. With **[Expert View]** you can access the extended "Expert" overview. ↪ *Chap. 4.10.1 'Device web page CPU' page 98*

Tab: *'Info'*



Name	Value	
Ordering Info	015-CEFNR00	Order number of the CPU
Serial	...	Serial number of the CPU
Version	03V...	Version number of the CPU
HW Revision	03	CPU hardware revision
Software	3.0.5	CPU firmware version
Package	Pb000328.pkb	File name for the firmware update

Current firmware at www.yaskawa.eu.com

You can find the current firmware versions in the *'Download Center'* of www.yaskawa.eu.com under *'Firmware 015-CEFNR00'*. For example the following file is necessary for the firmware update of the CPU 015-CEFNR00 and its components with hardware release 03:

- CPU 015N, Hardware release 03: Pb000328.pkb

CAUTION!
When installing a new firmware you have to be extremely careful. Under certain circumstances you may destroy the CPU, for example if the voltage supply is interrupted during transfer or if the firmware file is defective. In this case, please call our hotline!

Please regard that the version of the update firmware has to be different from the existing firmware otherwise no update is executed.

4.13.1 Firmware update online

Precondition

- Access is via the IP address of the Ethernet PG/OP channel.
  Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 77
- You can access the IP address with a web browser.

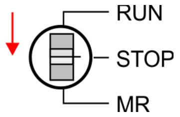
It is assumed that there is a connection via the Ethernet PG/OP channel between the PC with web browser and the CPU. This may be tested by Ping to the IP address of the Ethernet PG/OP channel.

Load firmware and store it in working directory

1. Go to the *'Download Center'* of www.yaskawa.eu.com.
2. Download under *'Firmware 015-CEFNR00'* the corresponding zip file for your CPU to your PC..
3. Extract the zip file into your working directory.

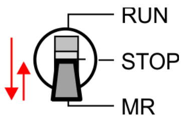
CAUTION!
With a firmware update an overall reset is automatically executed. If your program is only available in the load memory of the CPU it is deleted! Save your program before executing a firmware update!

Perform firmware update



1. Switch the operating mode switch of your CPU in position STOP.
2. Execute an overall reset. *Chap. 4.12 'Overall reset' page 115*
3. Open the CPU web page and select the *'Firmware'* tab.
4. Click at *'Browse ...'* and navigate to the firmware file in your working directory.
5. Click at *'Upload'*.

   ⇨ The firmware file is checked for plausibility and transmitted to the CPU. After the transfer, the firmware versions are matched and listed with the note if a firmware update is possible.



6. You start the firmware update by tipping the operating mode switch downwards to MR and then leaving the switch in STOP position.

   ⇨ During the update process, the LEDs SF and FC are alternately blinking. This may last several minutes.

7. The update is successful finished when the LEDs PW, SF, ST, FC and SD are on. If they are blinking fast, an error occurred.

8. Turn power OFF and ON.

   ⇨ After the start-up, the CPU is ready for operation with the new firmware. The current firmware version can be determined via the web page of the CPU.

4.13.2 Firmware update via memory card

Overview

- For the firmware update via memory card an accordingly prepared memory card must be in the CPU during the start-up.
- So a firmware files can be recognized and assigned with start-up, a pkb file name is reserved for each hardware revision, which begins with "pb" and differs in a number with 6 digits.
- In the System SLIO CPU, the pkb file name can be shown via the web page.
- After PowerON and operating mode switch of the CPU in STOP, the CPU checks if there is a pkb file at the memory card. If this firmware version is different to the existing firmware version, this is indicated by blinking of the LEDs and the firmware may be installed by an update request.

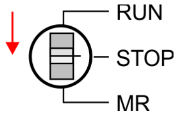
The procedure here describes the update from the CPU firmware version V2.4.0. The update of an older version to the firmware version V2.4.0 has to be done via pkg files. For this refer to the corresponding manual for your CPU version.

Load firmware and transfer it to memory card

1. Go to the *'Download Center'* of www.yaskawa.eu.com.
2. Download under *'Firmware 015-CEFNR00'* the corresponding zip file for your CPU to your PC..
3. Unzip the zip file and copy the pgb file to the root directory of your memory card.

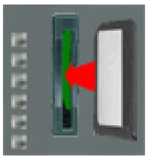
CAUTION!
With a firmware update an overall reset is automatically executed. If your program is only available in the load memory of the CPU it is deleted! Save your program before executing a firmware update!

Transfer firmware from memory card into CPU



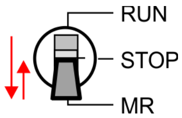
1. Switch the operating mode switch of your CPU in position STOP.

2. Turn off the power supply.



3. Plug the memory card with the firmware file into the CPU. Please take care of the correct plug-in direction of the memory card.

4. Turn on the power supply.
   ⇨ After a short boot-up time, the alternate blinking of the LEDs SF and FC shows that a more current firmware file was found at the memory card.



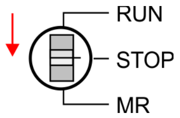
5. You start the transfer of the firmware as soon as you tip the operating mode switch downwards to MR within 10s and then leave the switch in STOP position.
   ⇨ During the update process, the LEDs SF and FC are alternately blinking and SD LED is on. This may last several minutes.
6. The update is successful finished when the LEDs PW, SF, ST, FC and SD are on. If they are blinking fast, an error occurred.

7. Turn the power supply off and on again.
   ⇨ After the start-up, the CPU is ready for operation with the new firmware. The current firmware version can be determined via the web page of the CPU.

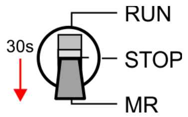
4.14 Reset to factory settings

Proceeding

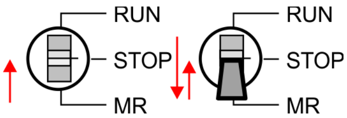
- With the following proceeding the internal RAM of the CPU is completely deleted and the CPU is reset to delivery state.
- Please regard that the MPI address is also reset to default 2 and the IP address of the Ethernet PG/OP channel is reset to 0.0.0.0!
- A factory reset may also be executed by the command *FACTORY_RESET*. ✎ *Chap. 4.17 'CMD - auto commands' page 124*



1. Switch the CPU to STOP.



2. Push the operating mode switch down to position MR for 30 seconds. Here the ST LED blinks. After a few seconds the ST LED changes to static light. Now the ST LED changes between static light and blinking. Start here to count the static light of the ST LED.



3. After the 6. Static light release the operating mode switch and tip it downwards to MR.

   ⇨ For the confirmation of the resetting procedure the green RN LED lights up once. This means that the RAM was deleted completely.

> *If the ST LED is on, only an overall reset has been performed and the reset to factory setting has been failed. In this case you can repeat the procedure. A factory reset can only be executed if the ST LED has static light for exact 6 times.*

4. The update is successful finished when the LEDs PW, ST, SF, FC and MC are on.

5. Turn power OFF and ON.

4.15 Deployment storage media - VSD, VSC

Overview

At the front of the CPU there is a slot for storage media. Here the following storage media can be plugged:

- VSD - **VI**PA **SD**-Card
  - External memory card for programs and firmware.
- VSC - **V**IPA**S**et**C**ard
  - External memory card (VSD) for programs and firmware with the possibility to unlock optional functions like work memory and field bus interfaces.
  - These functions (FSC: **F**eature **S**et **C**ode) can be purchased separately.
  - To activate a FSC the corresponding card is to be installed and a *Overall reset* is to be established. *Chap. 4.12 'Overall reset' page 115*

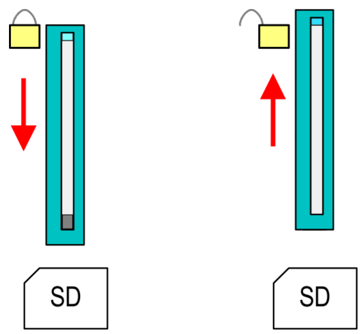
> *To avoid malfunctions, you should use memory cards of Yaskawa. These correspond to the industrial standard. A list of the currently available VSD respectively VSC can be found at www.yaskawa.eu.com*

You can cause the CPU to load a project automatically respectively to execute a command file by means of pre-defined file names.

VSD

VSDs are external storage media based on SD memory cards. VSDs are pre-formatted with the PC format FAT 16 (max. 2GB) and can be accessed via a card reader. After PowerON respectively an overall reset the CPU checks, if there is a VSD with data valid for the CPU.

Push the VSD into the slot until it snaps in leaded by a spring mechanism. This ensures contacting. By sliding down the sliding mechanism, a just installed VSD card can be protected against drop out.



To remove, slide the sliding mechanism up again and push the storage media against the spring pressure until it is unlocked with a click.

> **CAUTION!**
> If the media was already unlocked by the spring mechanism, with shifting the sliding mechanism, a just installed memory card can jump out of the slot!

VSC

The VSC is a VSD with the possibility to enable optional functions (FSCs). Here you have the opportunity to accordingly expand your work memory respectively enable field bus functions. Information about the enabled functions can be shown via the web page. *Chap. 4.10 'Accessing the web server' page 98*

CAUTION!
Please regard that the VSC must remain plugged when you've enabled optional functions at your CPU. Otherwise the SF LED is on and the CPU switches to STOP after 72 hours. As soon as an activated VSC is not plugged, the SF LED is on and the "TrialTime" counts downwards from 72 hours to 0. After 72 hours the CPU switches to STOP state. By plugging the VSC, the SF LED expires and the CPU is running again without any restrictions.

The VSC cannot be replaced by a VSC of the same optional functions. The activation code is fixed to the VSD by means of an unique serial number. Here the function as an external memory card is not affected.

Accessing the storage medium

To the following times an access takes place on a storage medium:

After overall reset

- The CPU checks if a VSC is inserted. If so, the corresponding optional functions (FSCs) are enabled.
- The CPU checks whether a project S7PROG.WLD exists. If so, it is automatically loaded.

After PowerON

- The CPU checks whether a project AUTOLOAD.WLD exists. If so, an overall reset is executed and the project is automatically loaded.
- The CPU checks whether a command file with the name VIPA_CMD.MMC exists. If so the command file is loaded and the commands are executed.
- After PowerON and CPU STOP the CPU checks if there is a *.pkb file (firmware file). If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. *Chap. 4.13 'Firmware update' page 116*

In STOP state when inserting a memory card

- If a memory card is plugged in STOP state, which contains a command file VIPA_CMD.MMC, the command file is loaded and the containing instructions are executed.

The FC/SFC 208 ... FC/SFC 215 and FC/SFC 195 allow you to include the memory card access into your user application. More information about this may be found in the manual "SPEED7 Operation List".

4.16 Extended know-how protection

Overview

Please note that this functionality is not supported by the Siemens TIA Portal!

Besides the "standard" Know-how protection the CPUs provide an "extended" know-how protection that serves a secure block protection for accesses of 3. persons.

- Standard protection
  – The standard protection from Siemens transfers also protected blocks to the PG but their content is not displayed.
  – But with according manipulation the know-how protection is not guaranteed.
- Extended protection
  – The "extended" know-how protection offers the opportunity to store blocks permanently in the CPU.
  – With the "extended" protection you transfer the protected blocks to a memory card into a WLD-file named protect.wld.
  – By plugging the memory card and then an overall reset the blocks in the protect.wld are permanently stored in the CPU.
  – You may protect OBs, FBs and FCs.
  – When back-reading the protected blocks into the PG, exclusively the block header are loaded. The block code that is to be protected remains in the CPU and cannot be read.

Protect blocks with protect.wld

1. Create a new wld file in your project engineering tool with *'File ➔ Memory Card file ➔ New'*.
2. Rename the wld file to "protect.wld".
3. Transfer the according blocks into the file by dragging them with the mouse from the project to the file window of protect.wld.
4. Transfer the file protect.wld to a memory card.
5. Plug the memory card into the CPU and execute an *overall reset*. *Chap. 4.12 'Overall reset' page 115*

   ⇨ The overall reset stores the blocks in protect.wld permanently in the CPU protected from accesses of 3. persons.

Protection behaviour

Protected blocks are overwritten by a new protect.wld. Using a PG 3. persons may access protected blocks but only the block header is transferred to the PG. The block code that is to be protected remains in the CPU and cannot be read.

Change respectively delete protected blocks

Protected blocks in the RAM of the CPU may be substituted at any time by blocks with the same name. This change remains up to next overall reset. Protected blocks may permanently be overwritten only if these are deleted at the protect.wld before. By transferring an empty protect.wld from the memory card with an overall reset, you may delete all protected blocks in the CPU.

Usage of protected blocks

Due to the fact that reading of a "protected" block from the CPU monitors no symbol labels it is convenient to provide the "block covers" for the end user. For this, create a project of all protected blocks. Delete all networks in the blocks so that these only contain the variable definitions in the according symbolism.

4.17 CMD - auto commands

Overview

A *Command* file at a memory card is automatically executed under the following conditions:

- CPU is in STOP and memory card is plugged
- After each PowerON

Command file

- The *Command* file is a text file, which consists of a command sequence to be stored as **VIPA_cmd.mmc** in the root directory of the memory card.
- The file has to be started by CMD_START as 1. command, followed by the desired commands (no other text) and must be finished by CMD_END as last command.
- Text after the last command *CMD_END* e.g. comments is permissible, because this is ignored.
- As soon as the command file is recognized and executed each action is stored at the memory card in the log file logfile.txt.
- For each executed command a diagnostics entry may be found in the diagnostics buffer.

Commands

Please regard the command sequence is to be started with *CMD_START* and ended with *CMD_END*.

Command	Description	Diagnostics entry
CMD_START	In the first line CMD_START is to be located.	0xE801
	There is a diagnostics entry if CMD_START is missing.	0xE8FE
WAIT1SECOND	Waits about 1 second.	0xE803
LOAD_PROJECT	The function "Overall reset and reload from memory card" is executed. The wld file located after the command is loaded else "s7prog.wld" is loaded.	0xE805
SAVE_PROJECT	The recent project (blocks and hardware configuration) is stored as "s7prog.wld" at the memory card. If the file just exists it is renamed to "s7prog.old". If your CPU is password protected so you have to add this as parameter. Otherwise there is no project written. Example: SAVE_PROJECT password	0xE806
FACTORY_RESET	Executes "factory reset".	0xE807
DIAGBUF	The current diagnostics buffer of the CPU is stored as "diagbuff.txt" at the memory card.	0xE80B
SET_NETWORK	IP parameters for Ethernet PG/OP channel may be set by means of this command. The IP parameters are to be given in the order IP address, subnet mask and gateway in the format x.x.x.x each separated by a comma. Enter the IP address if there is no gateway used.	0xE80E
CMD_END	In the last line CMD_END is to be located.	0xE802
WEBPAGE	Saves all information on the device web page (Expert-View) as *webpage.txt* on the memory card *Chap. 4.10 'Accessing the web server' page 98*	0xE804
WEBVISU_PGOP_ENABLE	Enable *WebVisu* project via Ethernet PG/OP channel	0xE82C
WEBVISU_PGOP_DISABLE[1]	Disable *WebVisu* project via Ethernet PG/OP channel	0xE82D

Command	Description	Diagnostics entry
WEBVISU_CP_ENABLE	Enable *WebVisu* project via Ethernet CP	0xE82E
WEBVISU_CP_DISABLE[1]	Disable *WebVisu* project via Ethernet CP	0xE82F
OPCUA_PGOP_ENABLE	Enable *OPC UA* project via Ethernet PG/OP channel	0xE830
OPCUA_PGOP_DISABLE	Disable *OPC UA* project via Ethernet PG/OP channel	0xE831
OPCUA_CP_ENABLE	Enable *OPC UA* project via Ethernet CP	0xE832
OPCUA_CP_DISABLE	Disable *OPC UA* project via Ethernet CP	0xE833
1) After a power cycle or loading a hardware configuration, the settings are retained. With *reset to the factory settings* or *over all reset*, the *WebVisu* project is set to the default value "enabled".		

Examples

The structure of a command file is shown in the following. The corresponding diagnostics entry is put in parenthesizes.

Example 1

`CMD_START`	Marks the start of the command sequence (0xE801)
`LOAD_PROJECT proj.wld`	Execute an overall reset and load "proj.wld" (0xE805)
`WAIT1SECOND`	Wait ca. 1s (0xE803)
`DIAGBUF`	Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)
`CMD_END`	Marks the end of the command sequence (0xE802)
`... arbitrary text ...`	Text after the command CMD_END is not evaluated.

Example 2

`CMD_START`	Marks the start of the command sequence (0xE801)
`LOAD_PROJECT proj2.wld`	Execute an overall reset and load "proj2.wld" (0xE805)
`WAIT1SECOND`	Wait ca. 1s (0xE803)
`WAIT1SECOND`	Wait ca. 1s (0xE803)
	IP parameter (0xE80E)
`SET_NETWORK 172.16.129.210,255.255.224.0,172.16.129.210`	
`WAIT1SECOND`	Wait ca. 1s (0xE803)
`WAIT1SECOND`	Wait ca. 1s (0xE803)
`DIAGBUF`	Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)
`CMD_END`	Marks the end of the command sequence (0xE802)
`... arbitrary text ...`	Text after the command CMD_END is not evaluated.

The parameters IP address, subnet mask and gateway may be received from the system administrator. Enter the IP address if there is no gateway used.

4.18 Control and monitoring of variables with test functions

Overview

For troubleshooting purposes and to display the status of certain variables you can access certain test functions in the *SPEED7 Studio*

- Test of the user program in the PLC simulator
- Monitoring blocks in the editor
- Viewing and changing variables in watch tables
- Recording of signals by logic analysis

4.18.1 Test of the user program in the PLC simulator

With the PLC simulator, you can test your application program in a virtual CPU before loading it into your PLC. The happens with the following proceeding:

1. Load your user program.
2. Compile your user program.
3. Set at *'Active PC interface'* the virtual interface *'Simulation'*.
4. Open the dialog *'PLC simulation settings'* and make your simulation adjustments if necessary.
5. Start the simulation with *'Simulation ➔ Start PLC simulation'*
   ⇨ The simulation is started.
6. Here you can test your application program, e.g. monitor values of variables or signal states or overwrite variables with values.
7. With *'Simulation ➔ Stop PLC simulation'* the simulation is stopped.

4.18.2 Monitoring blocks in the editor

In the *SPEED7 Studio* you can monitor variables of a block in the block editor. For this the block to be monitored must be loaded in the CPU and you must be online connected to the CPU.

1. Open the block of type OB, FB, FC or DB in the block editor.
2. Click at .
   ⇨ The variable values are cyclically read from the PLC and shown. Please note that in this case you can not make any changes at the block.

			VKE	STA	Akku 1	Statuswort
1	UN	M 1.0	1	0	0050	00000000 00000011
2	L	S5T#300MS	1	0	0030	00000000 00000011
3	SE	T 1	1	0	T#000.0	00000000 00000010
4	NOP 0		1	0	0030	00000000 00000010
5	NOP 0		1	0	0030	00000000 00000010
6	NOP 0		1	0	0030	00000000 00000010
7	U	T 1	1	1	T#000.0	00000000 00000111
8	L	S5T#200MS	1	1	0020	00000000 00000111
9	SE	T 2	1	1	T#017.0	00000000 00000110

Depending on the editor the result of logic operation (RLO), status bit (STA) and values of the accu and status word register are shown.

3. To stop the monitoring click again at .

4.18.3 Viewing and changing variables in watch tables

In the watch table you can monitor (read) and control (write) variables. You can specify , which variables of the CPU are to be read and controlled. You can create more watch tables if you want. This information is obtained from the corresponding area of the selected operands. During the controlling of variables respectively in operating mode STOP the input area is directly read. Otherwise only the process image of the selected operands is displayed.

Inputs can be monitored, but not be controlled. Outputs can be controlled, but not be observed.

Add a watch table

1. Click in the project tree within a PLC beneath *'PLC program'* at *'*
   ➔ *Add watch table'*.
   ⇨ The dialog *'Add watch table'* is opened.
2. *'Name'*: Enter a name if needed.
3. *'Comment'*: Enter a comment if needed, e.g. add or description
4. Click at *'OK'*.
   ⇨ The watch table is added and listed in the project tree.
5. Open the watch table
6. Enter via the first line of the table the variables, which you want to monitor respectively control.
7. Marc with ☑ in the column *'Watch'* all the variables, you want to monitor.
8. Click at the icon, to cyclically read data from the PLC.
9. Marc with ☑ in the column *'Control'* all the variables, you want to control.
10. Click at the icon, to write all control values to the PLC with each PLC cycle.

CAUTION!
Please consider that controlling of output values represents a potentially dangerous condition.

These functions should only be used for test purposes respectively for troubleshooting.

4.18.4 Recording of signals by logic analysis

With the logic analysis you can cyclically record PLC signals. Please consider for this an appropriate license for the *SPEED7 Studio* is necessary. To start the logic analysis choose *'View* ➔ *Logic analysis'*. More information about the usage may be found in the online help of the *SPEED7 Studio*.

4.19 Diagnostic entries

Accessing diagnostic data

↳ *Appendix A 'System specific event IDs' page 349*

- You may read the diagnostic buffer of the CPU via the *SPEED7 Studio*. To show the diagnostic entries you choose in the *SPEED7 Studio 'Device ➔ Status of component'*. Here via *'Diagnostic buffer'* you can access the diagnostic buffer.
- The current content of the diagnostic buffer is stored at the memory card by means of the CMD DIAGBUF. ↳ *Chap. 4.17 'CMD - auto commands' page 124*
- The diagnostic is independent from the operating mode of the CPU. You may store a max. of 100 diagnostic entries in the CPU.

5 Deployment *OPC UA*

> *Please note that the simultaneous use of OPC UA and WebVisu on the same interface is not supported! When attempting to activate them, both servers are stopped and the diagnostic message 0xE989 or 0xE9AB is output.*

- With an *OPC UA* project there is the possibility to configure an *OPC UA* server on your CPU respectively Ethernet CP.
- The configuration of an *OPC UA* project is only possible with the *SPEED7 Studio* V1.8.6 and up.
- Since a *OPC UA* project is only executable by memory card, a memory card (VSD, VSC) of Yaskawa must be plugged. Please note that you must always use a VSC card suitable for your CPU. ↳ *Chap. 4.15 'Deployment storage media - VSD, VSC' page 121*
- If the memory card is removed for a short time, the SF LED gets on. This indicates that a feature is missing and the *OPC UA*server will quit after 72 hours.
- When the project is transferred from the *SPEED7 Studio*, the *OPC UA* project is automatically transferred as TAR file to the inserted memory card.
- The *OPC UA* server must be activated in the CPU. ↳ *Chap. 5.3 'Activate OPC UA functionality' page 138*
- The access is made with an OPC UA client via the *'Endpoint URL'* of the corresponding interface. The *'Endpoint URL'* can be found on the *Device web page* of the CPU. ↳ *Chap. 4.10.1.1.1 'Tab: 'OPC UA'' page 102*

5.1 General

Term definitions

- OPC - **O**pen **P**latform **C**ommunications
  - *OPC* is an interoperability standard for secure and reliable data exchange in industrial automation.
  - *OPC* is platform-independent and ensures a seamless flow of information between devices from different manufacturers.
- UA - **U**nified **A**rchitecture
  - *UA* specifies security features and data modelling based on a service-oriented architecture (SOA).

Precondition

- *SPEED7 Studio* from Version V1.8.6
  - The functionality for the *OPC UA* configuration is integrated in the *SPEED7 Studio*.
- Siemens SIMATIC Manager from version V5.5 and *SPEED7 Studio* from version V1.8.6
  - The *OPC UA* configuration is done with the *OPC UA Configurator*. This is part of the *SPEED7 Studio* from version V1.8.6.
  - When calling the *OPC UA Configurator*, the *SPEED7 Studio* opens with functionality limited to *OPC UA* configuration.
  - The *OPC UA Configurator* is to be called from the Siemens SIMATIC Manager as external device tool.
  - To be able to call the *OPC UA Configurator* as an external device tool, you must first register it in the Siemens SIMATIC Manager. This is done with *SPEED7 Tools Integration*, which is automatically installed during the installation of the *SPEED7 Studio*.
  - The *OPC UA Configurator* is to be called from the Siemens SIMATIC Manager after project creation and online configuration.
  - The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens SIMATIC Manager.
  - The *OPC UA* configuration is transferred online from the *OPC UA Configurator*. The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens SIMATIC Manager.
- Siemens TIA Portal from version V15.0 and *SPEED7 Studio* from version V1.8.6
  - The *OPC UA* configuration is done with the *OPC UA Configurator*. This is part of the *SPEED7 Studio* from version V1.8.6.
  - When calling the *OPC UA Configurator*, the *SPEED7 Studio* opens with functionality limited to *OPC UA* configuration.
  - The *OPC UA Configurator* is to be called from the Siemens TIA Portal as external device tool.
  - To be able to call the *OPC UA Configurator* as an external device tool, you must first register it in the Siemens TIA Portal. This is done with *SPEED7 Tools Integration*, which is automatically installed during the installation of the *SPEED7 Studio*.
  - The *OPC UA Configurator* is to be called from the Siemens TIA Portal after project creation and online configuration.
  - The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens TIA Portal.
  - The *OPC UA* configuration is transferred online from the *OPC UA Configurator*. The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens TIA Portal.

5.2 Basics OPC UA

5.2.1 OPC UA

Standard for data and information exchange

OPC UA defines a common standard for data and information exchange in an *'Industry 4.0'* environment. Due to the platform independence, the integrated security concept and the data type information supplied with the data, *OPC UA* provides the basis for machine-readable and cross-level communication.

OPC - Open Platform Communications

- Classic variant not scalable and exclusively for Microsoft Windows®
- For each type of data transfer, such as real-time data, history data, interrupts, events, etc., a separate solution with its own semantics is required, such as OPC DA, OPC HDA, OPC A&E, etc.
- Separate and complex effort required for security settings.
- *OPC* requires a complex *DCOM* configuration.
- *OPC* requires separate complex firewall settings.

OPC UA - Open Platform Communications Unified Architecture

- Scalable and platform-independent communication standard specified in IEC 62541.
- Standardization of classic OPC specifications with integrated security concept.
- The *OPC UA* security concept includes user and application authentication, message signing, and encryption of transmitted data.
- IP-based, optimized, binary protocol for Internet and firewall communication via one port (4840).
- With *OPC UA*, any type of information is available at any time and place for any authorized application and authorized person. For example, raw data and preprocessed information can be transported safely from the sensor and field level to the control system and into the production planning systems.
- SOA (**S**ervice **O**riented **A**rchitecture) replaces the Microsoft *DCOM* technology with open, platform-independent protocols with integrated security mechanisms.
  - Communication takes place via standardized services based on the *Information model* of OPC UA.
  - The services are divided into different task groups.
  - Based on a basic model, arbitrarily complex, object-oriented extensions of the services can be carried out, without affecting interoperability.

OPC UA server

- An *OPC UA server* provides information within a network that can be retrieved from an OPC UA client.
- The data exchange can take place via security certificates, which have to be stored accordingly in the server.
- The *OPC UA server* provides basic services such as data exchange or navigation through the address space.
- The OPC UA configuration is used to define the variables or contents that an OPC UA server should provide.
- The OPC UA configuration is done via an external tool such as for Yaskawa CPUs the *OPC UA Configurator*.

OPC UA client

OPC UA clients are programs with the following functionality:

- Read or write access to information of the OPC UA server.
- Access is controlled by access rights.
- Execute services on the OPC UA server.

Communication types

- Client/Server
  - An *OPC UA client* accesses information from the *OPC UA server* via services, provided by the *OPC UA server*. Here a fix defined connection is used.
  - Example: *OPC UA client* retrieves status of an input in the CPU.
- Publisher/Subscriber
  - A *Publisher* sends to unknown *Subscriber* (clients) without a fixed connection.
  - Example: Sensors send data to the cloud.

5.2.2 Information modeling

Information model

- *Information models* are used to describe devices and their data.
- The basis is the *Core specification*. The *Core specification* describes the structure of the address range and of the services, such as the entry points for the clients in the address space of an *OPC UA* server.
- In an *information model*, the content of the address space of the *OPC UA* server is described.

- The *Information models* are structured in layers. Each higher-order type is based on certain basic rules. Thus, clients who only know the basic rules can still edit complex information models, e.g. navigate through the *address space* and read or write data variables.
- In the *address space*, all information is represented by *Nodes*, which are interconnected via *references*.
- A node is always an instance of a *NodeClass*.
- *OPC UA* offers basic services such as data exchange or navigation through the address space. The services are grouped in *Service Sets*.

Node classes

The following *NodeClasses* are defined in the *OPC UA* specification:

- Variable - class of variables
- Method - class of functions
- Object - class of objects
- View - Class of view of a subset of nodes
- DataType - Class of the data types of the value of a variable
- VariableType - Class of the data types of a variable
- ObjectType - class of object types
- ReferenceType - class of reference types

Node attributes

Each *node* consists of attributes and references. Some attributes may also be optional. The following attributes of each *NodeClass* must be published:

- NodeID - Unique identifier of a *nodes* in the *address space*
- NodeClass - class of *node* instance
- BrowseName - name of the *node* in plain text
- DisplayName - display name of the *Node* for the user
- Description - Description of the *node* (optional)

***OPC UA* services**

- *OPC UA* services are abstract descriptions defined by request and response messages.
- The available services of an *OPC UA* server are defined in the server profile and grouped together in service sets.

Basic service sets

- Discovery Service Set
  – Services for discovering existing servers and endpoints.
- SecureChannel Service Set
  – Services for opening and closing secure communication channels.
- Session Service Set
  – Services for the client to create and manage a session.
- NodeManagement Service Set
  – Services for creating and deleting nodes and references.
- View Service Set
  – Services for the client to navigate in the address space or in the view.
- Query Service Set
  – Services for search queries in the address space.
- Attribute Service Set
  – Services for accessing attributes of nodes.
- Method Service Set
  – Service for calling a method of an object.

- MonitoredItem Service Set
  - Services for the client to create and manage monitored items.
  - Monitored items are used to log in for data and event notifications.
- Subscription Service Set
  - Services for the client to create and manage subscriptions.
  - Subscriptions control the way of the data and event notification.

Access

- To access an *OPC UA* server, the *endpoint* must be known.
- You can navigate via the *endpoint* using the navigation function through the address space of the *OPC UA* server. Here you receive information about the *OPC UA* server and the CPU and have access to the objects created in the *OPC UA* configuration, such as tags, data blocks, etc.
- Lower network load through *'subscriptions'*
  - If variables are to be transmitted only if their value has changed, you have to use *subscriptions*.
  - To activate a *subscription*, enter the transmission interval "Publishing Interval" in the *OPC UA* client.
  - When the *subscription* is created, tell the server which variables to monitor. Among other things, you can specify the amount by which a value must change in order a transfer takes place.
  - Since only a change in value of a transmission takes place, the use of *subscriptions* leads to a reduced network load.
- Fast access through *'registration'*
  - Normally the addressing takes place by means of identifier strings. By using a numeric identifier access can be accelerated. For this reason, you should use the *registration* for regular access to certain variables.
  - During *registration*, the *OPC UA* client assigns the variable to the *OPC UA* server. The *OPC UA* server then generates a numeric identifier and sends it back to the *OPC UA* client.
  - The numerical identifier is valid for the duration of the session.
  - In the properties of the CPU, you can set the maximum number of registered nodes. This must be taken into account by the *OPC UA* clients.
  - Since the registration takes time, you should put them in the start-up phase of the *OPC UA* server.

Due to the system, access to data in complex structures is not consistent.

Setting the sampling intervals (sampling interval, publishing interval) too short may cause too much network load. Always choose intervals that are still sufficient for your application.

5.2.3 *OPC UA* data types and their conversion

Siemens S7 data types are mapped in the namespace via SPEED7 PLC OPC UA data types. Siemens S7 data types do not always match the *OPC UA* built-in data types. The CPU provides variables to the *OPC UA* server as an *OPC UA* built-in data type so that *OPC UA* clients can access these variables with *OPC UA* built-in data types via the server interface. A client can read the "DataType" attribute from such a variable and reconstruct the original data type.

Data type mapping

Siemens S7 data type		SPEED7 PLC *OPC UA* data type		*OPC UA* built-in data type
BOOL	→	BOOL	→	Boolean
BYTE		BYTE		Byte
WORD		WORD		UInt16
DWORD		DWORD		UInt32
INT		INT		Int16
DINT		DINT		Int32
REAL		REAL		Float
S5TIME		S5TIME		UInt16
TIME		TIME		Int32
DATE		DATE		UInt16
TIME_OF_DAY (TOD)		TIME_OF_DAY		UInt32
CHAR		CHAR		Byte
COUNTER		COUNTER		UInt16 (Only valid values)
TIMER		TIMER		UInt16 (Only valid values)
STRING		STRING		String
DT		DT		Byte[8]

Particularities

- String
  – The data type *STRING* in Siemens S7 is a byte array in which the maximum length and the current length are stored in the first 2 bytes. The other bytes store the string.
  – The *OPC UA* data type *String* should be defined in the same way.
- Array
  – A read or write job in *OPC UA* is always an *Array* access, i.e. always provided with index and length.
  – A single variable is a special case of an *Arrays* (index 0 and length 1). On the line, the data type is simply sent several times in succession. For the variable, the *DataType* attribute points to the base data type. The *ValueRank* and *ArrayDimensions* attributes determine if it is an array and how large the array is.
- Structure
  – A structure describes a complex data type.
  – You can describe your own structures as a subtype of the abstract data type *Structures*, which inherits from the data type *BaseDataType*.
  – Since a client may not know user-specific structures, the variables of the data type of this structure are uniformly published in an *ExtensionObject*. The structure *ExtensionObject* can be read by any client and also publishes the *DataTypeId* of the user-specific structure.
  – All structures that are not described by structures of the basic data types are published on the server in a *TypeDictionary*.
  – With the description of the structure by the *TypeDictionary* and the *DataTypeId*, which is published by the *ExtensionObject*, the structure of the *ExtensionObject* can be decoded by a client.
  – If a client knows in advance the description of a user-specific structure, it can be decoded without reading the *TypeDictionary*. In this approach, a client needs to read and decode the entire tree to access individual elements.

5.2.4 Integrated security concept

Generals to data security

The topic of data security and access protection have become increasingly important in the industrial environment. The increased networking of entire industrial systems to the network levels within the company together with the functions of remote maintenance have all served to increase vulnerability. Threats can arise from internal manipulation like technical errors, operator and program errors respectively from external manipulation like software viruses and worms, trojans and password phishing.

The most important precautions to prevent manipulation and loss of data security in the industrial environment are:

- Encrypting the data traffic by means of certificates.
- Filtering and inspection of the traffic by means of VPN - "Virtual Private Networks".
- Identification of the nodes by "Authentication" via save channels.
- Segmenting in protected automation cells, so that only devices in the same group can exchange data.

Guidelines for information security

With the "VDI/VDE 2182 sheet 1", Information Security in the Industrial Automation - General procedural model, VDI guidelines, the VDI/VDE society for measuring and automation engineering has published a guide for implementing a security architecture in the industrial environment. The guideline can be found at www.vdi.de PROFIBUS & PROFINET International (PI) can support you in setting up security standards by means of the "PROFINET Security Guideline". More concerning this can be found at the corresponding web site such as www.profibus.com

Security mechanisms in *OPC UA*

- Verifying the identity of *OPC UA* servers and clients.
- Checking the identity of the users.
- Signed and encrypted data exchange between *OPC UA* server and clients.
- In the connection settings in the *OPC UA Configurator*, you can specify how a user of an *OPC UA* client must legitimize access to the *OPC UA* server.

Safety rules:

- Only activate *'Anonymous-Login'* or *'Unsecured data traffic'* in exceptional cases.
- Only allow access to variables and data blocks via *OPC UA* if it is actually required.

Activate only security guidelines that are compatible with the protection concept for your machine or Application. Deactivate all other security guidelines.

X.509 certificates

OPC UA has integrated security mechanisms in multiple layers. An important component here are X.509 certificates, which are also used in the PC world. When using certificates, the *OPC UA* server delivers data to the client only if the security certificate has been accepted as valid on both sides. An X.509 certificate includes the following information:

- Version and serial number of the certificate.
- Name of the certification authority.
- Information about the algorithm used by the certification authority to sign the certificate.
- Start and end of the validity of the certificate.
- Name of the program, person, or organization for which the certificate was signed by the certification authority.
- The public key of the program, person or organization.

OPC UA uses three types of X.509 certificates when establishing a client-to-server connection:

- *OPC UA* application certificates
- *OPC UA* software certificates
- *OPC UA* user certificates

- Check when establishing a connection
  – When establishing a connection between client and server, the participants check all information from the certificate that is required to establish integrity.
  – Among other things, the period of validity which is stored in the certificate is checked. Please ensure that the date and time are set correctly for the participants, otherwise no communication can take place.
- Sign and encrypt
  – To avoid tampering, certificates are signed.
  – Within the *OPC UA Configurator*, you can use the *'Server settings'* to import certificates or create and sign them yourself.
- Self-signed certificate
  – Each participant generates his own certificate and signs it.
  – Self-signed certificates are to be transferred to the CPU.
  – From a self-signed certificate no new certificates can be derived.
  – Sample applications: Static configuration with limited number of communication participants.
- *CA certificate:*
  – All certificates are created and signed by a certification authority.
  – Only the derived and signed certificate of the certification authority is to be transferred to the CPU.
  – The certification authority can generate new certificates. Adding partner devices is possible at any time.
  – Sample applications: Dynamically growing plants.

Digital signature

The signature can be used to prove the integrity and origin of a message.

1. The sender forms a hash value as a check value from the clear message.
2. The hash value and a private key result in the digital signature.
3. The clear message is sent to the recipient together with the digital signature.
4. The recipient decrypts the received signature with the public key and thus gets back the original hash value.
5. The receiver also forms a hash value from the clear message and checks it with the original hash value. The public key and hash method are included in the X.509 certificate.

   ⇨ - If both hash values are identical, sender and clear message were not manipulated.
     - If both hash values are not identical, the clear message was manipulated or falsified during transmission.

Encrypting

- X.509 certificates are not encrypted; they are public and anyone can see them.
- Encrypting data prevents unauthorized users from knowing the content.
- When encrypting, the sender encrypts the clear message with the recipient's public key from the X.509 certificate.
- The recipient decrypts the message with his private key. Each owner of the private key can decrypt a received message.

Secure Channel

- *OPC UA* uses private and public keys to establish secure channels between client and server
- Once a secure connection is established, the client and server generate a shared private key for signing and encrypting messages.

Security policies

OPC UA uses the following security policies to protect messages:

- *No security*
  All messages are unsecured. To use these security policies, connect to a "None" endpoint of a server.
- *Sign*
  All messages are signed. This allows the integrity of the received messages to be checked. Manipulations are detected. To use these security policies, connect to a "Sign" endpoint of a server.
- *Sign & encrypt*
  All messages are signed and encrypted. This allows the integrity of the received messages to be checked. Manipulations are detected. Due to the encryption, no attacker can read the content of the message. To use these security policies, connect to a "Sign & Encrypt" endpoint of a server.
  The security guidelines are additionally named according to the algorithms used. Example: "Basic256Sha256 - Sign & Encrypt" means: Secure Endpoint, supports a set of algorithms for 256-bit hashing and 256-bit encryption.

Please note that the encryption of the communication can affect CPU performance and therefore the response time of the entire system!

5.3 Activate *OPC UA* functionality

Proceeding

For your CPU can process a *OPC UA* project, you have to activate the *OPC UA* functionality.

1. Insert a Yaskawa memory card (VSD, VSC) into your CPU. Please note that you must always use a VSC card suitable for your CPU. *Chap. 4.15 'Deployment storage media - VSD, VSC' page 121*
2. Turn on the CPU, to activate the *OPC UA* functionality, you have to establish an *Overall reset*.

   ⇨ As long as the memory card is inserted, the *OPC UA* functionality remains activated even after a power cycle. When the project is transferred from the *OPC UA Configurator*, the *OPC UA* project is automatically transferred to the inserted memory card.

Please regard that the memory card must remain plugged when you've executed activated the OPC UA functionality. Otherwise the SF LED is on and the OPC UA functionality is deactivated after 72 hours. As long as an activated memory card is not inserted, the SF LED is on and the "TrialTime" timer counts from 72 hours down to 0. After that the OPC UA functionality is de-activated. By inserting the memory card, the LED goes off and the CPU runs again without restrictions.

Please note that the use of a OPC UA project, depending on the scope of the OPC UA project and the PLC project, can influence the performance and thus the response time of your application.

5.4 Usage in *SPEED7 Studio*

Precondition

- *SPEED7 Studio* from Version V1.8.6
  – The functionality for the *OPC UA* configuration is integrated in the *SPEED7 Studio*. For more information, see the according online help.

5.5 Usage in Siemens SIMATIC Manager

5.5.1 Precondition

Siemens SIMATIC Manager from V5.5 and *SPEED7 Studio* from V1.8.6

- The *OPC UA* configuration happens by the external *OPC UA Configurator*.
- The *OPC UA Configurator* is the *SPEED7 Studio* reduced to *OPC UA* functionality.
- The *OPC UA Configurator* can be registered in the Siemens SIMATIC Manager by means of the *SPEED7 Tools Integration*.
- The *OPC UA Configurator* is to be called from the Siemens SIMATIC Manager after project creation and online configuration.
- The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens SIMATIC Manager.
- The *OPC UA* configuration is transferred online from the *OPC UA Configurator*. The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens SIMATIC Manager.

Please note that only the objects of the LD, FBD and IL languages can be transferred to the OPC UA Configurator.

5.5.2 Installation *OPC UA Configurator*

Proceeding

Installation and activation of *SPEED7 Studio*

The *OPC UA Configurator* is part of the *SPEED7 Studio* with *OPC UA* functionality. With the *SPEED7 Tools Integration*, which is also installed when installing the *SPEED7 Studio* the *OPC UA Configurator* is to be registered in the Siemens SIMATIC Manager as external tool.

1. The latest version of the *SPEED7 Studio* can be found in the *'Download Center'* of www.yaskawa.eu.com. Double-click on the installation program an follow the instructions on the monitor.

   The use of the SPEED7 Studio requires that you agree with the license agreement. During installation, you must confirm this.

   Further components are required in order to operate *SPEED7 Studio*. If the following programs are not already present on your PC, they are automatically installed:

   - Microsoft .NET Framework 4.52
   - Microsoft SQL Server© 2014 SP1
   - WinPcap

2. You can use a 30-day demo version or activate a license.

   In order to use *SPEED7 Studio* without restrictions, you require a licence, which you can obtain from your local Yaskawa customer service organisation.

   If the PC, on which you would like to use the *SPEED7 Studio*, is connected to the Internet, you can activate the licence online. If no license is activated, the dialog box for activating the license opens with each new start of *SPEED7 Studio*.

   Click on *'Yes'*.

   ⇨ The *'Product activation'* dialog window will open.

3. Enter the serial number that you received with your order of *SPEED7 Studio* in the *'Licence key'* input field.

4. Enter your name in the *'Your name'* input field.

5. If you enter your e-mail address in the *'E-mail address'* input field, you receive an e-mail confirmation regarding the product activation.

6. Click at *'Activate'*.

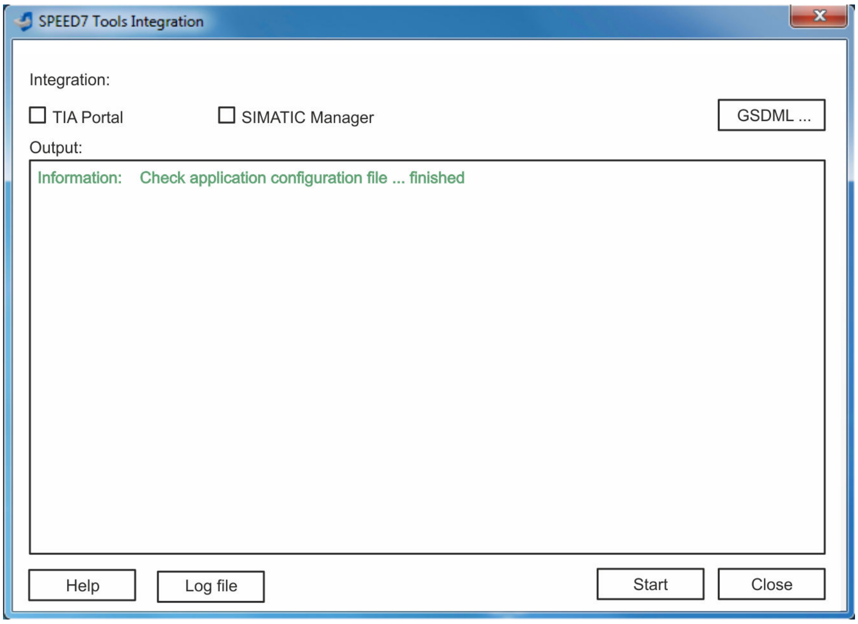
   ⇨ The licence is activated and the *SPEED7 Studio* is started.

Registration of SPEED7 Studio in the Siemens SIMATIC Manager as *OPC UA Configurator*.

SPEED7 Tools Integration is automatically listed in the Windows Start menu during the installation of the *SPEED7 Studio*.

1. To start the *SPEED7 Tools Integration*, click in the Windows Start menu on *'VIPA ... ➔ SPEED7 Tools Integration'*.

   ⇨ For *SPEED7 Tools Integration* can start, you have to acknowledge the security prompt to change the data on your computer with *'Yes'*. Afterwards *SPEED7 Tools Integration* will be started.



2. Click at *'GSDML ...'*.
3. Navigate to your GSDML file of your CPU, which you also use for your configuration in the Siemens *'SIMATIC Manager'*. Select these and click at *'Confirm'*. You can also select and use several GSDML files.

   ⇨ The identified GSDML files are listed and the selection for the configuration tools is enabled.
4. Select the Siemens *'SIMATIC Manager'*, in which the *SPEED7 Studio* is to be registered as *OPC UA Configurator*.
5. Click on *'Start'*.

   ⇨ - *SPEED7 Studio* is registered in the Windows registry as *OPC UA Configurator*.
   - In the Siemens SIMATIC Manager the *OPC UA Configurator* is registered as externally callable program.
   - All changes are recorded in a log file, which you can output via *'Log file'*.
6. *'Close'* closes *SPEED7 Tools Integration*.

   ⇨ With the next start of the Siemens hardware configurator, the *SPEED7 Studio* can be called as *OPC UA Configurator* with to *OPC UA* configuration limited functions. More information about the usage can be found in the in the online help of the *OPC UA Configurator*.

5.5.3 Steps of the *OPC UA* configuration

Steps of configuration

When using the Siemens SIMATIC Manager, the *OPC UA* configuration happens by the following steps:

1. Create your project in the Siemens SIMATIC Manager with the corresponding hardware configuration. *Chap. 4.5 'Hardware configuration - CPU' page 75*
2. Configure the corresponding Ethernet connection for PG/OP communication and establish an online connection. *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 77*
3. Save translate and transfer your project. *Chap. 4.9 'Project transfer' page 93*
4. Call the external *OPC UA Configurator* from the Siemens SIMATIC Manager. For this click in the hardware configurator on the CPU and select *'Start Device Tool ➔ VIPA Framework ➔ OPC UA Configurator'*.
5. Confirm to start an external program with [YES].

> **NOTICE!**
> **Data exchange between platforms of different vendors**
> If you allow access, you permit the exchange of data between *OPC UA Configurator* and your project data of the Siemens SIMATIC Manager.
> – Ensure that the necessary security guidelines are complied with.

⇨ The *OPC UA Configurator* is started. For the *OPC UA* configuration, the data is taken from the Siemens SIMATIC Manager project and listed in the table for the *OPC UA* configuration.

> *Please note that only the objects of the LD, FBD and IL languages can be transferred to the OPC UA Configurator.*

6. Configure the *OPC UA* server and the data for the *OPC UA* communication.
7. In the *OPC UA Configurator* switch to the online dialog and transfer the *OPC UA* configuration. For communication the IP address data are taken from the Siemens SIMATIC Manager project.

⇨ The *OPC UA* configuration is now complete. For check you will find information about your *OPC UA* configuration on the device web page at *'OPC UA'*. *Chap. 4.10.1.1.1 'Tab: 'OPC UA'' page 102*

5.6 Usage in Siemens TIA Portal

5.6.1 Precondition

Siemens TIA Portal from version V15.0 and *SPEED7 Studio* from V1.8.6

- The *OPC UA* configuration happens by the external *OPC UA Configurator*.
- The *OPC UA Configurator* is the *SPEED7 Studio* reduced to *OPC UA* functionality.
- The *OPC UA Configurator* can be registered in the Siemens TIA Portal by means of the *SPEED7 Tools Integration*.
- The *OPC UA Configurator* is to be called from the Siemens TIA Portal after project creation and online configuration.

- The *OPC UA Configurator* automatically imports the data for the *OPC UA* configuration from the project data of the Siemens TIA Portal.
- The *OPC UA* configuration is transferred online from the *OPC UA Configurator*. For the communication the *OPC UA Configurator* automatically uses the IP address data of the Siemens TIA Portal project.

> *Please note that only the objects of the LD, FBD and IL languages can be transferred to the OPC UA Configurator.*

5.6.2 Installation *OPC UA Configurator*

Proceeding

Installation and activation of *SPEED7 Studio*

The *OPC UA Configurator* is part of the *SPEED7 Studio* with *OPC UA* functionality. With the *SPEED7 Tools Integration*, which is also installed when installing the *SPEED7 Studio* the *OPC UA Configurator* is to be registered in the Siemens TIA Portal as external tool.

1. The latest version of the *SPEED7 Studio* can be found in the *'Download Center'* of www.yaskawa.eu.com. Double-click on the installation program an follow the instructions on the monitor.

   > *The use of the SPEED7 Studio requires that you agree with the license agreement. During installation, you must confirm this.*

   Further components are required in order to operate *SPEED7 Studio*. If the following programs are not already present on your PC, they are automatically installed:

   - Microsoft .NET Framework 4.52
   - Microsoft SQL Server© 2014 SP1
   - WinPcap

2. You can use a 30-day demo version or activate a license.

   In order to use *SPEED7 Studio* without restrictions, you require a licence, which you can obtain from your local Yaskawa customer service organisation.

   If the PC, on which you would like to use the *SPEED7 Studio*, is connected to the Internet, you can activate the licence online. If no license is activated, the dialog box for activating the license opens with each new start of *SPEED7 Studio*.

   Click on *'Yes'*.

   ⇨ The *'Product activation'* dialog window will open.

3. Enter the serial number that you received with your order of *SPEED7 Studio* in the *'Licence key'* input field.
4. Enter your name in the *'Your name'* input field.
5. If you enter your e-mail address in the *'E-mail address'* input field, you receive an e-mail confirmation regarding the product activation.
6. Click at *'Activate'*.

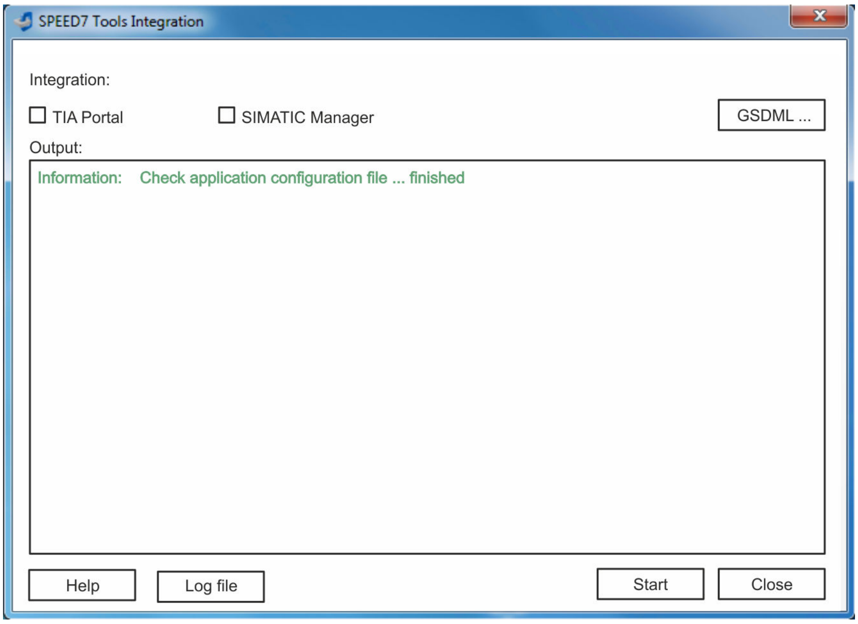
   ⇨ The licence is activated and the *SPEED7 Studio* is started.

Registration of SPEED7 Studio in the Siemens TIA Portal as *OPC UA Configurator*

SPEED7 Tools Integration is automatically listed in the Windows Start menu during the installation of the *SPEED7 Studio*.

1. To start the *SPEED7 Tools Integration*, click in the Windows Start menu on *'VIPA ... ➔ SPEED7 Tools Integration'*.

   ⇨ For *SPEED7 Tools Integration* can start, you have to acknowledge the security prompt to change the data on your computer with *'Yes'*. Afterwards *SPEED7 Tools Integration* will be started.



2. Click at *'GSDML ...'*.
3. Navigate to your GSDML file of your CPU, which you also use for your configuration in the Siemens *'TIA Portal'*. Select these and click at *'Confirm'*. You can also select and use several GSDML files.

   ⇨ The identified GSDML files are listed and the selection for the configuration tools is enabled.

4. Select *'TIA Portal'*, in which the *SPEED7 Studio* is to be registered as *OPC UA Configurator*.
5. Click on *'Start'*.

   ⇨
   - *SPEED7 Studio* is registered in the Windows registry as *OPC UA Configurator*.
   - In the Siemens TIA Portal the *OPC UA Configurator* is registered as externally callable program.
   - The current Windows user is registered in the user group *Siemens TIA Openness* of the Siemens TIA Portal.
   - All changes are recorded in a log file, which you can output via *'Log file'*.

6. *'Close'* closes *SPEED7 Tools Integration*.

   ⇨ With the next start of the Siemens TIA Portal, the *SPEED7 Studio* can be called as *OPC UA Configurator* with to *OPC UA* configuration limited functions. More information about the usage can be found in the in the online help of the *OPC UA Configurator*.

5.6.3 Steps of the *OPC UA* configuration

Steps of configuration

When using the Siemens TIA Portal, the *OPC UA* configuration happens by the following steps:

1. Create your project in the Siemens TIA Portal with the corresponding hardware configuration. *Chap. 13.3 'TIA Portal - Hardware configuration - CPU' page 326*
2. Configure the corresponding Ethernet connection for PG/OP communication and establish an online connection. *Chap. 13.4 'TIA Portal - Hardware configuration - Ethernet PG/OP channel' page 329*
3. Save translate and transfer your project. *Chap. 13.10 'TIA Portal - Project transfer' page 344*
4. Call the external *OPC UA Configurator* from the Siemens TIA Portal. For this click at *'Devices & networks'* on the CPU and select *'Start device tool'*.

   ⇨ A dialog window opens. Select *'OPC UA Configurator'* and click [Start].
5. Ignore the query *'Set interface'* with [OK]

   ⇨ The *OPC UA Configurator* is started.
6. If not yet confirmed, you will now receive an access request in the TIA Portal.

   Please note that due to the software the access request does not appear in the foreground. To show the access request, you must again bring the Siemens TIA Portal to the foreground. Once the access has been selected, you must again bring the 'OPC UA Configurator' to the foreground.

   You have the following options for access:

   - *'No'*: Deny access - the *OPC UA Configurator* is not started.
   - *'Yes'*: Access is permitted once and the *OPC UA Configurator* is started.
   - *'Yes to all'*: Access is permitted and the *OPC UA Configurator* is started. At the next call, the access request is no longer shown.

   Allow access with *'Yes'* respectively *'Yes to all'*.

   ⇨ **NOTICE!**
   Data exchange between platforms of different vendors
   If you allow access, you permit the exchange of data between *OPC UA Configurator* and your project data of the Siemens TIA Portal.
   – Ensure that the necessary security guidelines are complied with.

   For the *OPC UA* configuration, the data is taken from the Siemens TIA Portal project and listed in the table for the *OPC UA* configuration.

   Please note that only the objects of the LD, FBD and IL languages can be transferred to the OPC UA Configurator.
7. Configure the *OPC UA* server and the data for the *OPC UA* communication.

8. In the *OPC UA Configurator* switch to the online dialog and transfer the *OPC UA* configuration. For communication the IP address data are taken from the TIA Portal project.

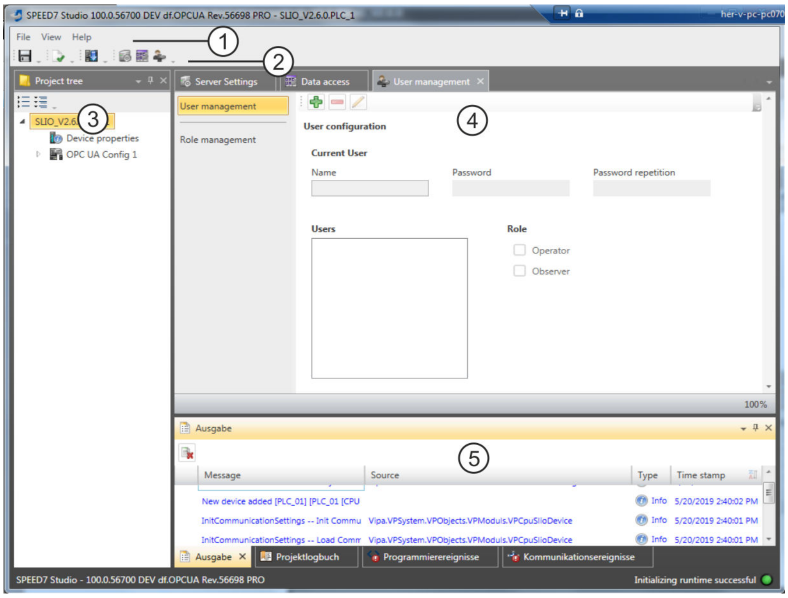
⇨ The *OPC UA* configuration is now complete. For check you will find information about your *OPC UA* configuration on the device web page at *'OPC UA'*. *Chap. 4.10.1.1.1 'Tab: 'OPC UA'' page 102*

> *You can use the CMD auto commands OPCUA_PGOP_ENABLE and OPCUA_PGOP_DISABLE to enable or disable OPC UA. After a power cycle or loading a hardware configuration, the settings are retained. With reset to the factory settings or over all reset, the OPC UA project is set to the default value "enabled".* *Chap. 4.17 'CMD - auto commands' page 124*

5.7 Usage *OPC UA Configurator*

5.7.1 *OPC UA Configurator*

The user interface of the *OPC UA Configurator* is divided into the following areas:



1 Menu bar
2 Toolbar
3 Project tree
4 Workspace
5 Output area

Menu bar

In the menu bar you will find a few general commands on the *OPC UA Configurator*. Further commands can be called up via context menus with the right mouse button, e.g. functions for an object in the project tree.

Toolbar

 Store *OPC UA* configuration

 Compile *OPC UA* configuration

Transfer *OPC UA* configuration into the control

Project tree

The *Project tree* gives you access to the *'Device properties'* and to the following areas of the *'OPC UA configuration'*:

- Server settings
- Data access
- User management

Workspace

In the *Work space*, you can edit the settings in the following areas of the *OPC UA* configuration:

- Device properties - General
  - Information about the CPU such as device name, name and firmware version.
- Device properties - Communication
  - Configuration of the interface for data exchange.
  - The IP address data are automatically imported from the project when the *OPC UA Configurator* is called and can be viewed here.
- Device properties - Server configuration
  - Administration and interface assignment of the *OPC UA* server in the *Project tree*
- Server settings - Connection
  - Legitimation of the user for access to the *OPC UA* server.
  - Port for communication.
  - Security policy for encryption and corresponding exceptions.
- Server settings - Certificate
  - Create, view, import or export X.509 ITU-T standard certificate.
  - Re-creating or importing replaces an existing certificate.
- Data access
  - Selection of the variables that can be accessed via *OPC UA*.
  - Filter option to limit the selection.
- User management
  - Creation of a user list with password and role assignment.

Output area

The output area shows information about activities performed and background operations.

5.7.2 Project tree

You can edit the *OPC UA* configuration via the project tree. The project tree contains the *OPC UA* configurations, which you have created. You can create a maximum of two *OPC UA* configurations: One configuration for the CPU and one configuration for the CP (if exists).

Show project tree

If the project tree is not shown select *'View ➔ Project tree'* or press *[Strg]+[Shift]+[P]*.

Show/hide objects

The objects in the project tree are arranged in a tree structure. You can show or hide objects:

Hide all objects (*'Project ➔ Collapse project tree'*)

Show all objects (*'Project ➔ Expand project tree'*)

Hide slave objects / close folder

▾ Show slave objects / open folder

Edit configurations and *OPC UA* configuration

Device properties	
Device properties	■ Edit device name and comment *Chap. 5.7.3.2 'General device properties' page 147* ■ Perform communication settings *Chap. 5.7.3.3 'Communication settings' page 148* ■ Create *OPC UA* configuration *Chap. 5.7.3.4 'Server configuration' page 150*
OPC UA	
Server settings	*Chap. 5.7.4 'Server settings - Connection ' page 150* *Chap. 5.7.5 'Server settings - Certificate ' page 151*
Data access	*Chap. 5.7.6 'Data access ' page 152*
User management	*Chap. 5.7.7 'User management ' page 153* *Chap. 5.7.8 'Role management ' page 154*

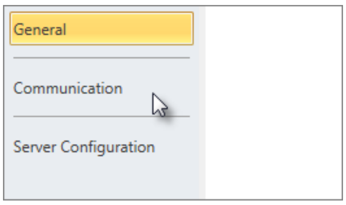
5.7.3 Device properties

5.7.3.1 Overview

Here you can edit the device name and the comment, perform the communication settings as well as create the *OPC UA* configuration.

→ Click in the project tree at *'Device properties'*.

⇨ The *'Device properties'* editor opens.



The *'Device properties'* editor is divided into several sections:

- *Chap. 5.7.3.2 'General device properties' page 147*
- *Chap. 5.7.3.3 'Communication settings' page 148*
- *Chap. 5.7.3.4 'Server configuration' page 150*

5.7.3.2 General device properties

To show or change the device properties, proceed as follows:

1. Click in the project tree at *'Device properties'*.

   ⇨ The editor of the *'Device properties'* opens.

2. Select the area *'General'*.

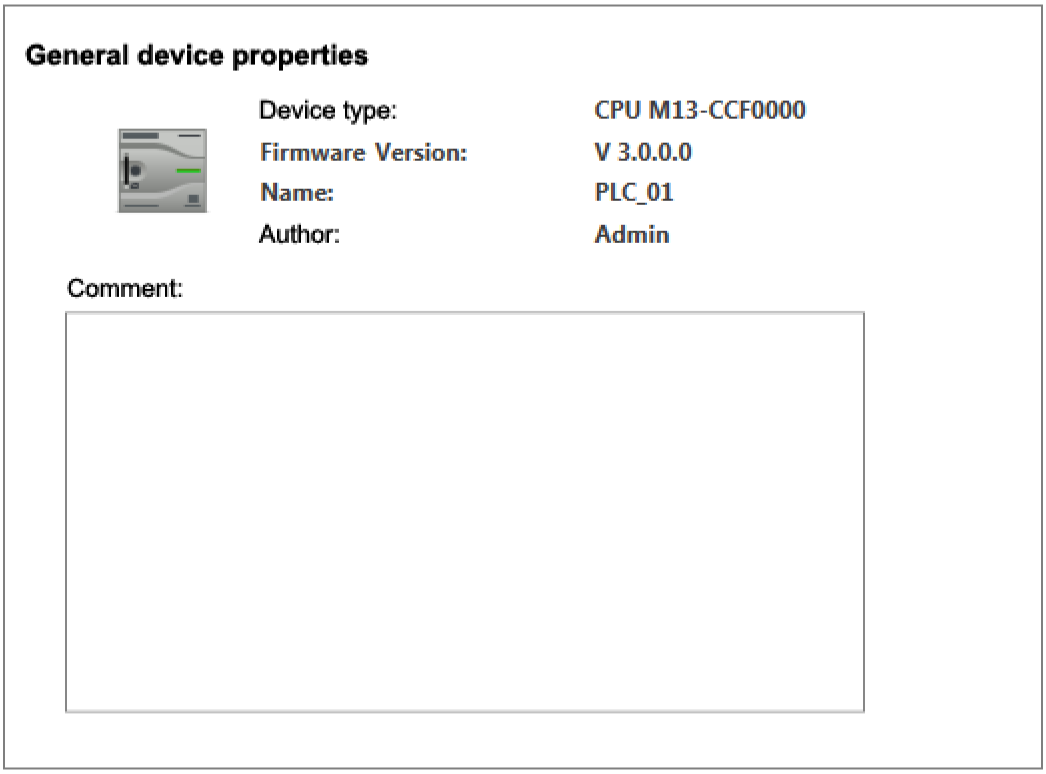


Fig. 1: Device properties of a PLC, as an example

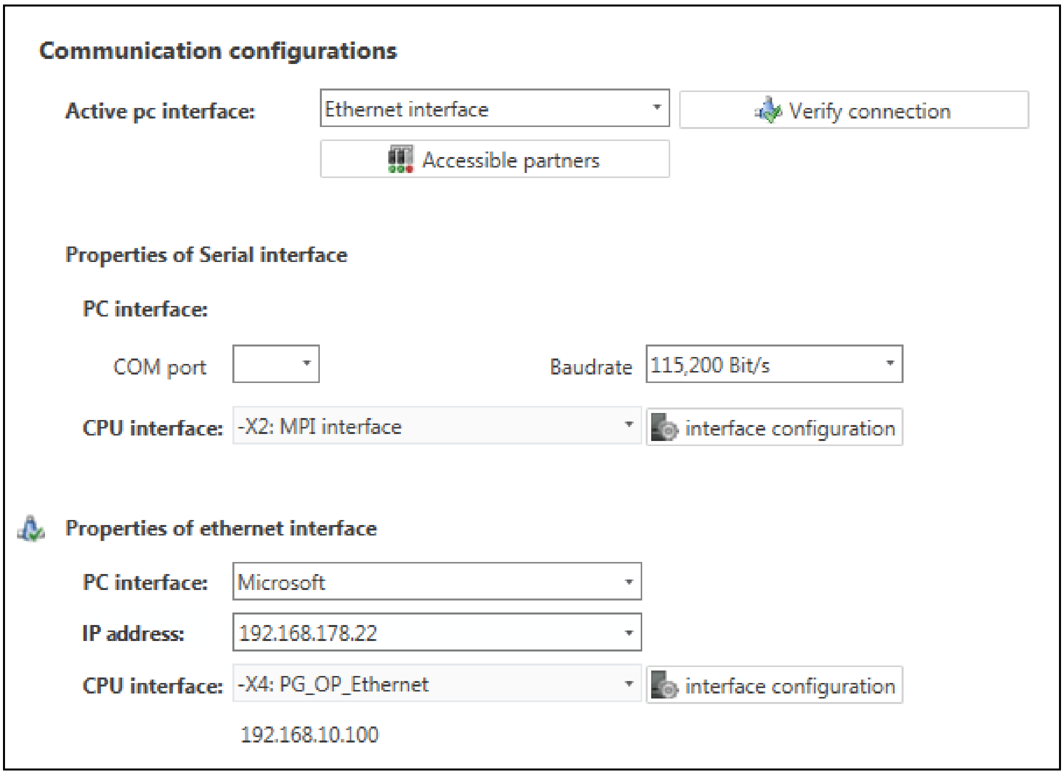
'Device type'	-	Name of the CPU
'Firmware'	-	Firmware version of the CPU
'Name'	-	Device name: This name is shown in the project tree.
'Author'	-	Name of the responsible person who created the device
'Comment'	-	Any comment, e.g. an annotation or explanation

→ Click on the input field and enter any comment, e.g. an annotation or explanation. With the *[Enter]* key, you can add a new line to the input field.

5.7.3.3 Communication settings

The communication settings are used to configure the interface for the data exchange between programming device and destination station. Since the IP address parameters for the *OPC UA* configuration are imported from the project, you simply have to set the interface via which you are connected to the destination station.

1. Click in the project tree at *'Device properties'*.
   ⇨ The editor of the *'Device properties'* opens.
2. Select the area *'Communication settings'*.



Setting the Ethernet interface

1. *'Active PC interface'*: Select *'Ethernet interface'*.
2. *'PC interface'*: Select the network adapter for the communication connection from the list.
   ⇨ If an IP address is already configured in the network adapter, it is shown under the input field *'IP address'*. If necessary, select a different IP address.
3. *'CPU interface'*: Select the interface of the control from the list.
   ⇨ Since the IP address is imported from the project, it is shown below the input field.
4. To configure further settings of the interface, click on *'Interface configuration'*.
   ⇨ The *'Interface properties'* dialog window will open.
5. In order to check whether a connection between the programming device and the control can be established with the selected communication settings, click on *'Verify connection'*.
   ⇨ You can see in the status line, whether the connection could be established successfully.
6. In order to check whether your programming device is connected with the correct control, you can retrieve information from the connected control. For this click on *'Accessible partners'*.
   ⇨ The *'Search for accessible partners'* dialog window will open.

5.7.3.4 Server configuration

Here you can create the *OPC UA* configurations.

1. Click in the project tree at *'Device properties'*.
   ⇨ The editor of the *'Device properties'* opens.
2. Select the area *'Server configuration'*.

You can create a maximum of two *OPC UA* configurations: One configuration for the CPU and one configuration for the CP (if exists).

Create configuration

1. Select in the selection field *'OPC UA Configuration'* and click on *'Add Server'*.
   ⇨ A new *OPC UA* configuration is created and listed in the project tree.
2. Click in the selection field *'Active server CP'* or *'Active server CP'* and choose which configuration is to be assigned. With the selection *'None'* the configuration remains saved in the project. However, it is not transferred to the device.

To swap the two configurations for CP and CPU, click on the button .

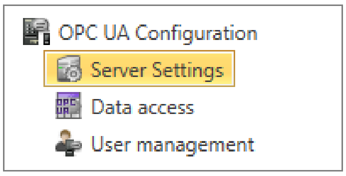
You can create a maximum of two *OPC UA* configurations.

Remove server

→ Right-click on the *OPC UA* configuration (PLC) in the project tree and select *'Remove OPC UA server'*.

5.7.4 Server settings - Connection

Here you can perform the connection settings of the *OPC UA* server.



1. Under *Project tree* at *'OPC UA configuration'* click on *'Server settings'*.
   ⇨ The *'Server settings editor'* editor opens.
2. Select the area *'Connection'*.

General

You can set for the *OPC UA* server how a user of an *OPC UA* client must prove their identity for access to the server. Select at least one of the following login methods. You can also combine the two login methods with each other.

- *'Activate anonymous login'*
  – The *OPC UA* server does not check the authorisation of the *OPC UA* client.
- *'Activate user/password login'*
  – The *OPC UA* server checks using the user name and password whether the access of the *OPC UA* client is authorised. To do this, the server evaluates the role assigned to the user. *Chap. 5.7.8 'Role management' page 154*
- *'Allow obsolete security guideline'*
  – Allows the selection of the two obsolete security guidelines *'Basic128Rsa15'* and *'Basic256'* (not recommended)
- *'Application name'*
  – Clear identification of the application in the OPC name space.

Network

- *'End point port'*
  – TCP port for binary data exchange (standard: 4840).

Security

> *Activate only security guidelines that are compatible with the protection concept for your machine or system. Deactivate all other security guidelines.*

- *'None'*
  – Insecure data traffic between server and client.
- *'Basic128Rsa15'*
  – Secured data traffic, 128-bit encoding with key wrap algorithm RSA-15, (allow option with *'Allow obsolete security guideline'* see above).
- *'Basic256'*
  – Secured data traffic, 256-bit encoding (allow option with *'Allow obsolete security guideline'* see above).
- *'Basic256Sha256'*
  – Secured data traffic, 256-bit encoding with hash algorithm SHA-256 (recommended).

Encoding:

- *'Sign'*
  – Endpoint secures the integrity of the data through signing.
- *'SignAndEncrypt'*
  – Endpoint secures the integrity and confidentiality of the data through signing and encoding.
- *'Both'*
  – The *OPC UA* server offers both encryption methods *'Sign'* and *'SignAndEncrypt'*. The *OPC UA* client can use one of the two encoding methods.

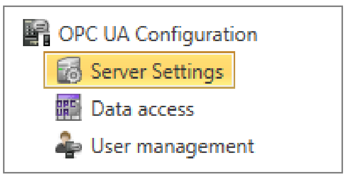
Security Check Overrides

Here you can allow various exceptions in the security check, in order to increase the error tolerance.

5.7.5 Server settings - Certificate

A secure connection between the *OPC UA* client and the server can only be established if the server classifies and accepts the client's digital certificate as trusted. Currently, the server accepts every valid client certificate. The server accepts self-signed certificates. In addition, the client also checks the server's certificate.

Here you can create, show, import or export an ITU-T standardized X.509 certificate for the *OPC UA* server. The certificate shown here is transferred into the *OPC UA* server.



1. Under *Project tree* at *'OPC UA configuration'* click on *'Server settings'*.
   ⇨ The *'Server settings'* editor opens.
2. Select the area *'Certificate'*.

The current X.509 certificate is shown in the work space. If you create or import a new certificate, the previously shown certificate is replaced.

Toolbar

Create new certificate: Opens the dialog window *'Create new certificate'*

Display certificate: Shows information on the current certificate

Export certificate: Opens the dialog window *'Save certificate'*

Import certificate: Opens the dialog window *'Open certificate'*

Create new certificate

1. Click on to create a new certificate.
   ⇨ The dialog window *'Create new certificate'* opens.
2. Enter the data for the certificate and click on *'OK'*.
   ⇨ The previously shown certificate is replaced by the new certificate.

Display certificate

- Click on to show information about the current certificate.
  ⇨ The dialog window *'Certificate'* opens.

Export certificate

You can export the current certificate e.g. to use it on different computers.

1. Click on .
   ⇨ The dialog window *'Save certificate'* opens.
2. Select a directory and enter a file name.
3. Click on *'Save'*.
   ⇨ The current certificate is saved in the export file (pfx file format).

Import certificate

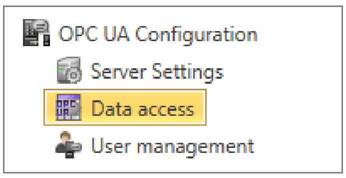
You can import a certificate, e.g. to use it for the current *OPC UA* configuration. For a successful import, the certificate must have the following characteristics:

- The certificate must be available as a PFX file.
- The *'Common name'* and *'Organization'* fields must be completed.
- The maximum key strength must not exceed 2048bit.
- The certificate must contain a valid *Private key*.

1. Click at .
   ⇨ The dialog window *'Open certificate'* opens.
2. Select the desired certificate (pfx file format).
3. Click on *'Open'*.
   ⇨ The previously shown certificate is replaced by the imported certificate.

5.7.6 Data access

Here you can select the variables belonging to the CPU or CP (if exists) that can be accessed via *OPC UA*.



- Under *Project tree* at *'OPC UA configuration'* click on *'Data access'*.
  ⇨ The editor for *'Data access'* opens.

Toolbar

Refresh variables: Apply changed filter settings to the result table.

Filter settings

Here you can select the operands and address ranges that will be shown in the results table.

1. Activate ☑ *'All operand areas'* or individual operand ranges to be shown in the result table.
2. To limit the addresses of an operand range, enter the start and end byte addresses in the two adjacent fields, e.g. `0` to `1000`.
3. Click on or activate ☑ *'Apply filter changes immediately'*.
   ⇨ The result table is updated with the filter settings.

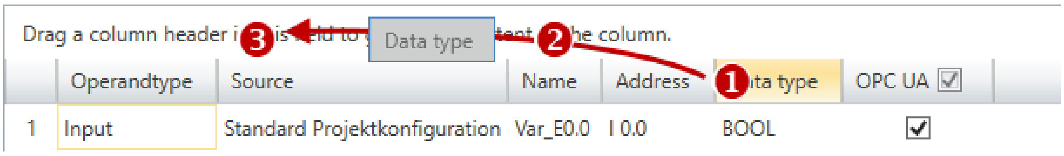
Result

In the results table, select the variables that are to be used in the *OPC UA* configuration. *OPC UA* clients may access these variables.

Activate ☑ *'OPC UA'* of the desired variables.

Group operands

For a better overview, you can sort the table entries by groups.



(1) Select column (hold left mouse button down)
(2) Drag the column
(3) Drop column in the field (release mouse button)

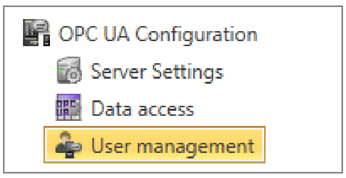
1. Drag the desired column header into the field above the table.
   ⇨ The contents of the column will be grouped. The number of lines is shown for each group.
2. Click on ▸ to open the group. Click on ▾ to close the group.

You can repeat steps 1 to 2 in order to structure the group into further sub-groups.

In order to cancel a grouping, click on the close icon next to the group name.

5.7.7 User management

The user management allows you to create a user list. For each user, you can define a password and a role.



1. Under *Project tree* at *'OPC UA configuration'* click on *'User management'*.
   ⇨ The editor for *'User management'* opens.
2. Select the area *'User management'*.

Toolbar

Add new user: Input mode for new user

Remove user: Deletes the selected user

Edit current user: Input mode for selected user

Save input: Save input Save user settings

Cancel input: Cancel user settings without saving

Adding a user

1. Click on .
2. Enter the desired user name in the input field *'Name'*.
3. Enter the password in the input field *'Password'* and repeat the input under *'Re-enter password'*.
4. Select a role for the user. With this role, the access rights to the *OPC UA* server are established.
5. Click on .
   ⇨ The user will be entered in the user list.

Edit user

1. In the user list, select the user whose data you want to change.
2. Click on .
3. Enter the desired changes and click on .

Removing a user

1. In the user list, select the user you want to delete.
2. Click on .
   ⇨ A dialog box opens where you can choose whether the user should be deleted or not.

5.7.8 Role management

Here you establish the roles and access rights that you can assign to the users. When you activate the authentication via User/password login *Chap. 5.7.4 'Server settings - Connection ' page 150*, the access rights to the *OPC UA* server are issued using the logged-in user and the assigned role.

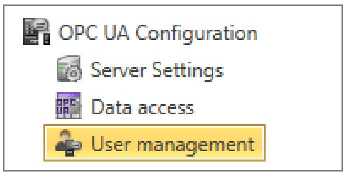
Example:

Role: Operator

Username: "I myself"

Server settings: User/password login activated

The user "Me Self" receives write permission and reading rights to the *OPC UA* server when he has successfully logged in with the password.



1. Under Project tree at *'OPC UA configuration'* click on *'User management'*.
   ⇨ The editor for *'User management'* opens.
2. Select the area *'Role management'*.

Configure roles

The following two roles are currently available for selection; further roles can not be added at the moment.

- Operator: Write permission and reading rights
- Observer: Reading rights only

5.7.9 Output

Information on executed activities and background operations are displayed in the "Output" window.

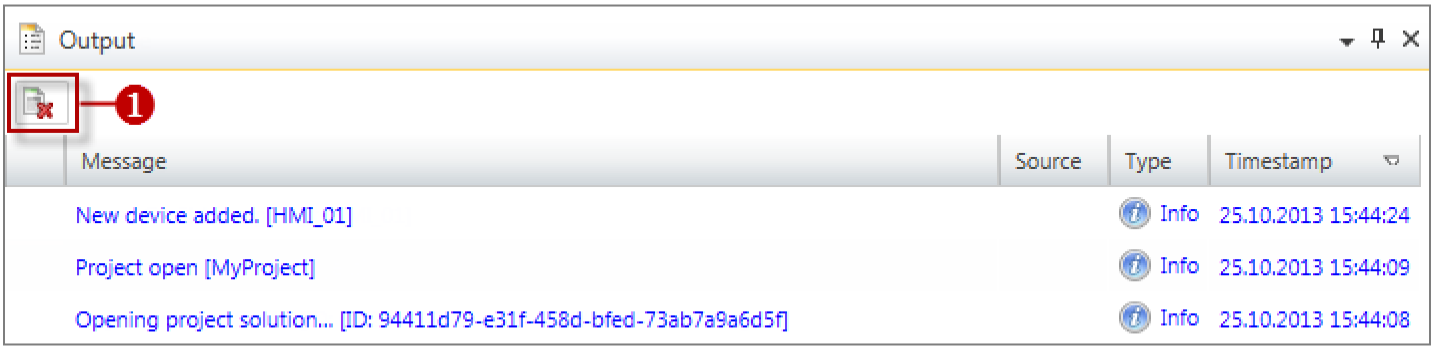


Fig. 2: Output

(1) Delete all messages in the output window

6 Deployment *WebVisu* - Web visualization

> *Please note that the simultaneous use of OPC UA and WebVisu on the same interface is not supported! When attempting to activate them, both servers are stopped and the diagnostic message 0xE989 or 0xE9AB is output.*

- With a *WebVisu* project there is the possibility to configure a web visualization on your CPU respectively Ethernet CP.
- The configuration of a *WebVisu* project is only possible with the *SPEED7 Studio* V1.7.0 and up.
- Since a *WebVisu* project is only executable by memory card, a Yaskawa memory card (VSD, VSC) must be plugged. Please regard that you always use a VSC card that fits to your CPU. *Chap. 4.15 'Deployment storage media - VSD, VSC' page 121*
- If the memory card is removed for a short time, the SF LED lights up. This indicates that a feature is missing and the *WebVisu* will quit after 72 hours.
- The *WebVisu* functionality must be activated in the CPU. *Chap. 6.2 'Activate WebVisu functionality' page 159*
- When the project is transferred from the *SPEED7 Studio*, the *WebVisu* project is automatically transferred as TAR file to the inserted memory card.
- Access to the *WebVisu* project of the CPU takes place via the IP address of the Ethernet PG/OP channel and the correspondingly configured port or via the *device web page* of the CPU.
- Access to the *WebVisu* project of the Ethernet CP takes place via the IP address of the Ethernet CP and the correspondingly configured port or via the *device web page* of the CP.
- You can access your web visualization via a web browser. Web browsers based on Windows CE are currently not supported.

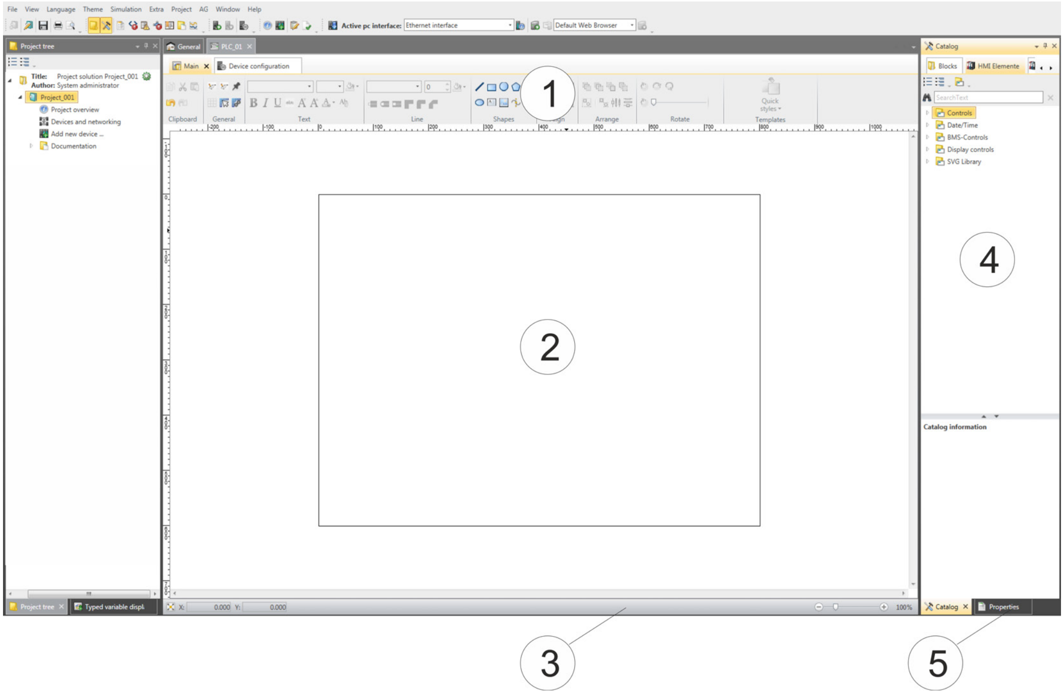
> *Please note that the use of a WebVisu project, depending on the scope of the WebVisu project and the PLC project, can influence the performance and thus the response time of your application. To relieve your CPU, you should run your WebVisu project on the Ethernet CP.*

6.1 *WebVisu* editor

The configuration of a *WebVisu* project is shown below. This is only the basic use of the *WebVisu* editor in the *SPEED7 Studio* together with the CPU. Please note that software changes can not always be considered and it may thus be deviations to the description.

> *For more information on the SPEED7 Studio and how to use the WebVisu editor, refer to its online help.*

6.1.1 Working environment



(1) Toolbar
(2) Editor surface
(3) Status bar
(4) Catalog
(5) Properties window

(1) Toolbar

The toolbar provides important commands for working with the *WebVisu* editor.

(2) Editor area

The editor area is your workspace. Here you can place and edit texts and graphics objects.

(3) Status bar

With a slider you can enlarge or reduce your view.

(4) Catalog

Via *Catalog* you can access all the *WebVisu* elements. You can use Drag & Drop to place them on the *Editor surface* and adjust them using properties.

(5) Properties window

By enabling *'View ➔ Properties'* the *'Properties'* are shown. The properties of the selected element are shown here. You can adjust these if necessary.

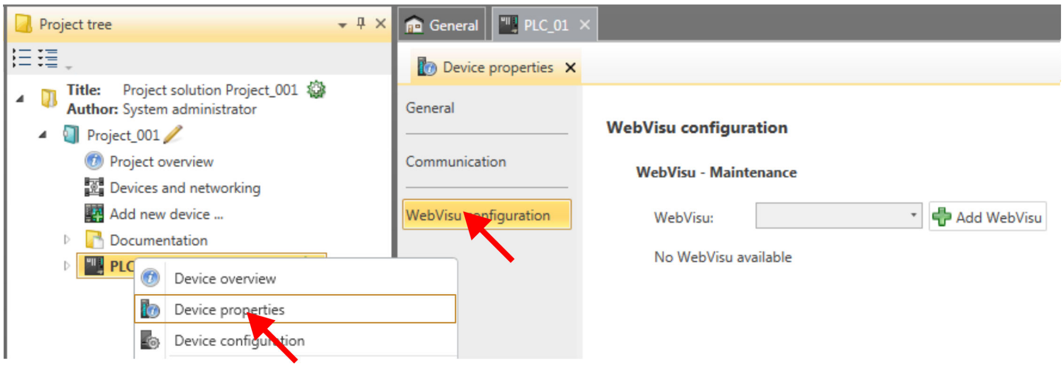
6.1.2 Creating a *WebVisu* project

Add *WebVisu*

1. Start the *SPEED7 Studio* with your project for the CPU for which a *WebVisu* project is to be created.
2. If not already done, add a CPU by clicking *'Add new device'*.
3. Click in the *'Project tree'* at the CPU and select *'Context menu ➔ Device properties'*.
   ⇨ The *'Device properties'* of your CPU opens.

4. Here click at *'WebVisu configuration'*

⇨ In this settings window, you can create a *WebVisu* project for your CPU.



5. To create a *WebVisu* project, click at [Add *WebVisu*].

⇨ A new WebVisu project is created and listed in the *'Project tree'*. At *'WebVisu - general configurations'* and *'WebVisu - SSL configurations'*, you can make further settings.

WebVisu - general configurations

- Port number
  – Enter the port number under which the *WebVisu* should be accessible.
  – *Port number*: 8080 (default): The *WebVisu* can be accessed via the IP address and port 8080. The *Device web page* can be accessed via the IP address and port 80.
  – *Port number*: 80: The *WebVisu* can be accessed via the IP address and port 80. The *Device web page* can be accessed via the IP address and port 8080.
- Polling interval (ms)
  – Enter here the interval for the cyclical refresh of the web content.
- Execution device
  – Select *'CPU'* as device on which this *WebVisu* project is to be executed.
  – *WebVisu* projects for Ethernet CPs are not supported by this CPU.

WebVisu - SSL configurations

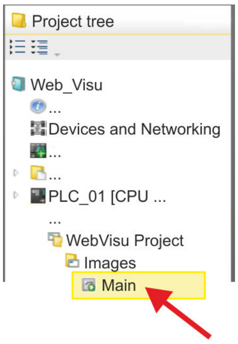
- Enabling encoding
  – When enabled, you have SSL-encrypted access to your *WebVisu*.
- Disable HTTP
  – When activated, the access happens via HTTPS.
- SSL port number
  – SSL port number 443 (default): The secure access to the *WebVisu* takes place via the IP address and port 443. The *Device web page* can be accessed via the IP address of the CPU and port 8080.
- Original path of the certificate used
  – Here you can upload a security certificate.
  – Only security certificates in PEM format are supported.
  – The file must contain the certificate and the private key.

Delete *WebVisu*

Click in the *'Project tree'* at *WebVisu* Project and select *'Context menu → Delete WebVisu'*.

⇨ The *WebVisu* project is removed from the configuration.

Edit *WebVisu*



➡ In the *'Project tree'*, navigate to *'WebVisu Project > Images'* and click at *'Main'*. Select *'Context menu ➔ Open image'*

⇨ The *WebVisu* editor opens. Here, you can configure your web visualization by dragging and dropping elements from the *'Catalog'* onto the editor area and using the *'Properties'* to interconnect them with a variable.

6.2 Activate *WebVisu* functionality

Proceeding

For your CPU can process a *WebVisu* project, you have to activate the *WebVisu* functionality.

1. Insert a Yaskawa memory card (VSD, VSC) into your CPU. Please note that you must always use a VSC card suitable for your CPU. *Chap. 4.15 'Deployment storage media - VSD, VSC' page 121*
2. Turn on the CPU, to activate the *WebVisu* functionality, you have to establish an *Overall reset*.

⇨ As long as the memory card is inserted, the *WebVisu* functionality remains activated even after a power cycle. When the project is transferred from the *SPEED7 Studio*, the *WebVisu* project is automatically transferred to the inserted memory card.

> *Please regard that the memory card must remain plugged when you've executed activated the WebVisu functionality. Otherwise the SF LED is on and the WebVisu functionality is deactivated after 72 hours. As long as an activated memory card is not inserted, the SF LED is on and the "TrialTime" timer counts from 72 hours down to 0. After that the WebVisu functionality is de-activated. By inserting the memory card, the LED goes off and the CPU runs again without restrictions.*

6.3 Start-up of the *WebVisu* project

The following preconditions must be fulfilled for the *WebVisu* project to start-up:

1. Activate *WebVisu* functionality if not already done. *Chap. 6.2 'Activate WebVisu functionality' page 159*
2. Configure your CPU and perform a hardware configuration.
3. Configure your *WebVisu* project.
4. Safe and translate your project.
5. If you are online connected to your CPU, you can transfer your project to the CPU with *'AG ➔ Transfer all'*.

⇨ Here the configuration is transferred in the CPU and the *WebVisu* project is transferred to the memory card. Immediately after the transfer you have access to your *WebVisu*.

> *You can use the CMD auto commands WEBVISU_PGOP_ENABLE and WEBVISU_PGOP_DISABLE to enable or disable the WebVisu. After a power cycle or loading a hardware configuration, the settings are retained. With reset to the factory settings or over all reset, the WebVisu project is set to the default value "enabled".* *Chap. 4.17 'CMD - auto commands' page 124*

6.4 Access to the *WebVisu*

- When connected via Ethernet PG/OP channel, you have access to the *WebVisu* and *Device web page* of the CPU, controlled via ports.
- Access to the *WebVisu* can be password-protected and encrypted by means of SSL certificates. If you want to use SSL certificates, you must integrate them in the *SPEED7 Studio* accordingly.
- You can create users in the *SPEED7 Studio* via *'WebVisu project > User administration'*, who can access the *WebVisu*.
- Via *'Device properties > WebVisu configuration'* you can specify the port, the *WebVisu* can be accessed and upload SSL certificates. This changes the port for accessing the device web page.
  - *Port number*: 8080 (default): The *WebVisu* can be accessed via the IP address and port 8080. The *Device web page* can be accessed via the IP address and port 80.
  - *Port number*: 80: The *WebVisu* can be accessed via the IP address and port 80. The *Device web page* can be accessed via the IP address and port 8080.
  - SSL port number 443 (default): The secure access to the *WebVisu* takes place via the IP address and port 443. The *Device web page* can be accessed via the IP address and port 8080.

> – *Please note that once you have made adjustments to the user management, you will need to restart your web browser. Otherwise you receive system-related error messages about invalid user information!*
> – *Please note that the encryption of the communication can affect CPU performance and therefore the response time of the entire system!*

6.4.1 Status of the *WebVisu*

On the device web page at the tab *'WebVisu'* via *'Status'* you get the status of your *WebVisu* project. *Chap. 4.10.1.1.2 'Tab: 'WebVisu'' page 103*

7 Deployment PtP communication

7.1 Fast introduction

General

The CPU has a RS485 interface, which is per default set to PtP communication (point to point). This allows to connect via serial process connection to different source or target systems.

Protocols

The protocols respectively procedures ASCII, STX/ETX, 3964R, USS and Modbus are supported.

Parametrization

The parametrization of the serial interface happens during runtime using the FC/SFC 216 (SER_CFG). For this you have to store the parameters in a DB for all protocols except ASCII.

Communication

The FCs/SFCs are controlling the communication. Send takes place via FC/SFC 217 (SER_SND) and receive via FC/SFC 218 (SER_RCV). The repeated call of the FC/SFC 217 SER_SND delivers a return value for 3964R, USS and Modbus via RetVal that contains, among other things, recent information about the acknowledgement of the partner station. The protocols USS and Modbus allow to evaluate the receipt telegram by calling the FC/SFC 218 SER_RCV after SER_SND. The FCs/SFCs are included in the consignment of the CPU.

Use FCs in the SPEED7 Studio

To ensure compatibility with other programming tools, these blocks are available as FC and SFC and therefore designated as "FC/SFC". In the SPEED7 Studio you should always use FCs. This increases the compatibility with other programming tools.

Overview FCs/SFCs for serial communication

The following FCs/SFCs are used for the serial communication:

FC/SFC		Description
FC/SFC 216	SER_CFG	RS485 parameterize
FC/SFC 217	SER_SND	RS485 send
FC/SFC 218	SER_RCV	RS485 receive

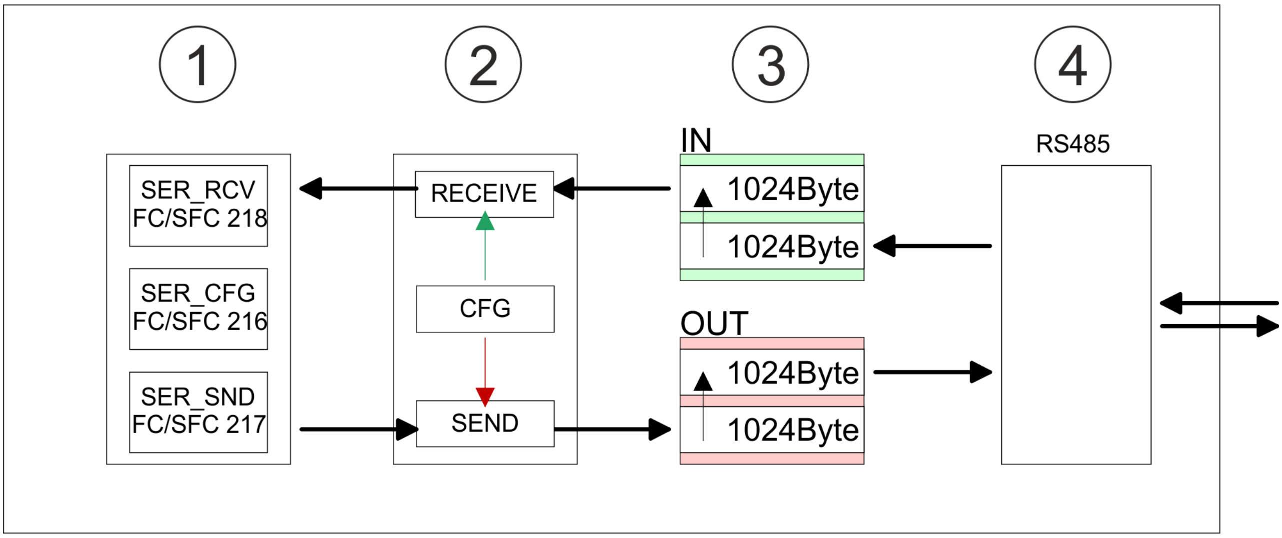
More information about the usage of these blocks may be found in the manual "SPEED7 Operation List".

7.2 Principle of the data transfer

RS485 PtP communication

The data transfer is handled during runtime by using FC/SFCs. The principle of data transfer is the same for all protocols and is shortly illustrated in the following.

- Data, which are written into the according data channel by the CPU, is stored in a FIFO send buffer (first in first out) with a size of 2x1024byte and then put out via the interface.
- When the interface receives data, this is stored in a FIFO receive buffer with a size of 2x1024byte and can there be read by the CPU.
- If the data is transferred via a protocol, the embedding of the data to the according protocol happens automatically.
- In opposite to ASCII and STX/ETX, the protocols 3964R, USS and Modbus require the acknowledgement of the partner.
- An additional call of the FC/SFC 217 SER_SND causes a return value in RetVal that includes among others recent information about the acknowledgement of the partner.
- Further on for USS and Modbus after a SER_SND the acknowledgement telegram must be evaluated by a call of the FC/SFC 218 SER_RCV.



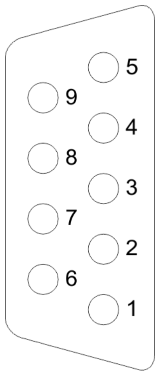
1 Program
2 Protocol
3 FIFO buffer
4 Interface

7.3 Deployment of RS485 interface for PtP

Properties RS485

- Logical states as voltage difference between 2 twisted wires
- Serial bus connection in two-wire technology using half duplex mode
- Data communications up to a max. distance of 500m
- Data communication rate up to 115.2kbaud

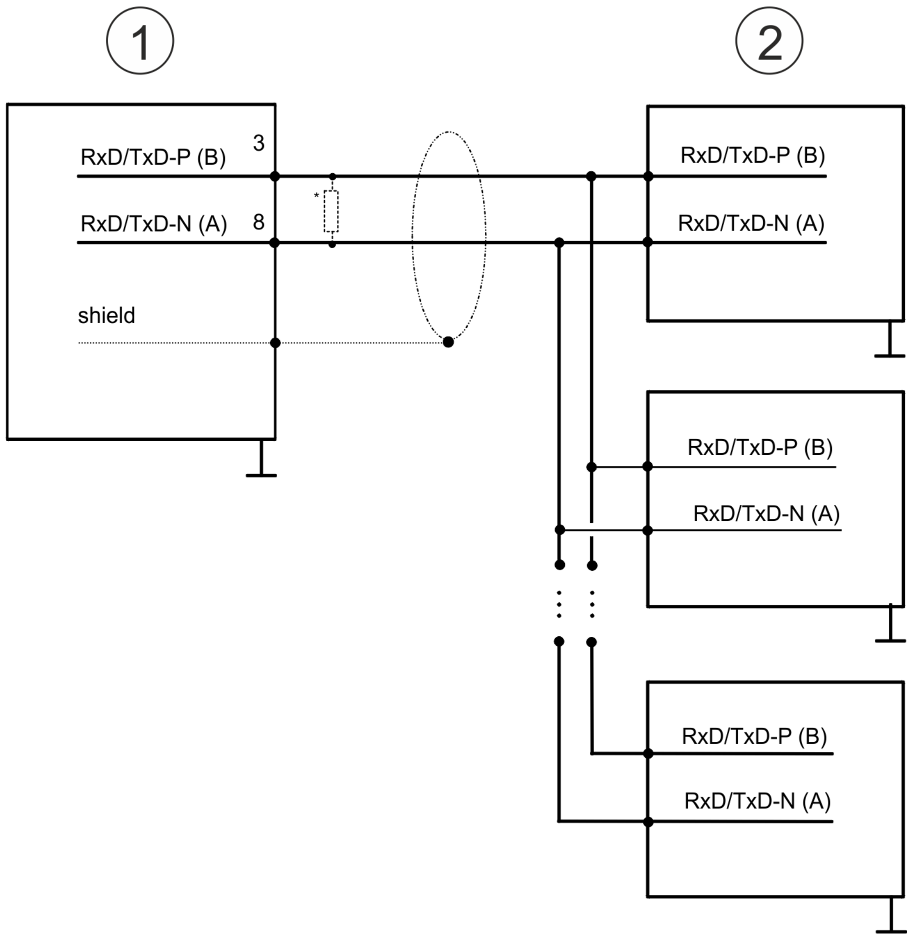
RS485



9pin SubD jack

Pin	RS485
1	n.c.
2	M24V
3	RxD/TxD-P (Line B)
4	RTS
5	M5V
6	P5V
7	P24V
8	RxD/TxD-N (Line A)
9	n.c.

Connection



1 RS485 interface
2 Periphery

> **) For traffic-free data transfer use a terminating resistor of approximately 120Ω.*

7.4 Parametrization

7.4.1 FC/SFC 216 - SER_CFG - Parametrization PtP

The parametrization happens during runtime deploying the FC/SFC 216 (SER_CFG). You have to store the parameters for STX/ETX, 3964R, USS and Modbus in a DB.

7.5 Communication

7.5.1 FC/SFC 217 - SER_SND - Send to PtP

This block sends data via the serial interface. The repeated call of the FC/SFC 217 SER_SND delivers a return value for 3964R, USS and Modbus via RETVAL that contains, among other things, recent information about the acknowledgement of the partner station. The protocols USS and Modbus require to evaluate the receipt telegram by calling the FC/SFC 218 SER_RCV after SER_SND.

7.5.2 FC/SFC 218 - SER_RCV - Receive from PtP

This block receives data via the serial interface. Using the FC/SFC 218 SER_RCV after SER_SND with the protocols USS and Modbus the acknowledgement telegram can be read.

> *More information about the usage of these blocks may be found in the manual "SPEED7 Operation List".*

7.6 Protocols and procedures

Overview

The CPU supports the following protocols and procedures:

- ASCII communication
- STX/ETX
- 3964R
- USS
- Modbus

ASCII

ASCII data communication is one of the simple forms of data exchange. Incoming characters are transferred 1 to 1. At ASCII, with every cycle the read FC/SFC is used to store the data that is in the buffer at request time in a parametrized receive data block. If a telegram is spread over various cycles, the data is overwritten. There is no reception acknowledgement. The communication procedure has to be controlled by the concerning user application. For this you can use the FB 1 - Receive_ASCII.

More information about the usage of this block may be found in the manual "SPEED7 Operation List".

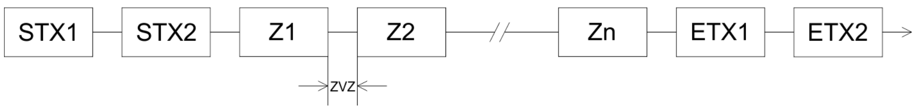
STX/ETX

STX/ETX is a simple protocol with start and end ID, where STX stands for **S**tart of **T**e**x**t and ETX for **E**nd of **T**e**x**t.

- Any data transferred from the periphery must be preceded by a Start followed by the data characters and the end character. Depending of the byte width the following ASCII characters can be transferred: 5bit: not allowed: 6bit: 20...3Fh, 7bit: 20...7Fh, 8bit: 20...FFh.
- The effective data, which includes all the characters between Start and End are transferred to the CPU when the End has been received.
- When data is send from the CPU to a peripheral device, any user data is handed to the FC/SFC 217 (SER_SND) and is transferred with added Start- and End-ID to the communication partner.
- You may work with 1, 2 or no Start- and with 1, 2 or no End-ID.
- If no End-ID is defined, all read characters are transferred to the CPU after a parameterizable character delay time (Timeout).

As Start-res. End-ID all Hex values from 01h to 1Fh are permissible. Characters above 1Fh are ignored. In the user data, characters below 20h are not allowed and may cause errors. The number of Start- and End-IDs may be different (1 Start, 2 End res. 2 Start, 1 End or other combinations). For not used start and end characters you have to enter FFh in the hardware configuration.

Message structure:



3964

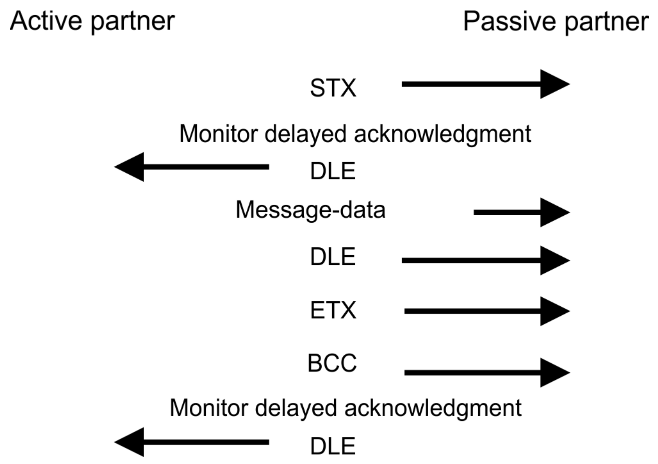
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free receipt.

The procedure employs the following control characters:

- STX: **S**tart of **T**e**x**t
- DLE: **D**ata **L**ink **E**scape
- ETX: **E**nd of **T**e**x**t
- BCC: **B**lock **C**heck **C**haracter
- NAK: **N**egative **A**c**k**nowledge

You may transfer a maximum of 255byte per message.

Procedure



When a DLE is transferred as part of the information it is repeated to distinguish between data characters and DLE control characters that are used to establish and to terminate the connection (DLE duplication). The DLE duplication is reversed in the receiving station.

The 3964R procedure requires that a lower priority is assigned to the communication partner. When communication partners issue simultaneous send commands, the station with the lower priority will delay its send command.

USS

The USS protocol (**U**niverselle **s**erielle **S**chnittstelle = universal serial interface) is a serial transfer protocol defined by Siemens for the drive and system components. This allows to build-up a serial bus connection between a superordinated master and several slave systems. The USS protocol enables a time cyclic telegram traffic by presetting a fix telegram length.

The following features characterize the USS protocol:

- Multi point connection
- Master slave access procedure
- Single master system
- Max. 32 participants
- Simple and secure telegram frame

It is essential:

- You may connect 1 master and max. 31 slaves at the bus
- The single slaves are addressed by the master via an address sign in the telegram.

- The communication happens exclusively in half-duplex operation.
- After a send command, the acknowledgement telegram must be read by a call of the FC/SFC 218 SER_RCV.

The telegrams for send and receive have the following structure:

Master slave telegram

STX	LGE	ADR	PKE		IND		PWE		STW		HSW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

Slave master telegram

STX	LGE	ADR	PKE		IND		PWE		ZSW		HIW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

with

STX - Start sign

STW - Control word

LGE - Telegram length

ZSW - State word

ADR - Address

HSW - Main set value

PKE - Parameter ID

HIW - Main effective value

IND - Index

BCC - Block Check Character

PWE - Parameter value

Broadcast with set bit 5 in ADR byte

7	6	5	4	3	2	1	0
		1					

Broadcast

A request can be directed to a certain slave ore be send to all slaves as broadcast message. For the identification of a broadcast message you have to set bit 5 to 1 in the ADR byte. Here the slave addr. (bit 0 ... 4) is ignored. In opposite to a "normal" send command, the broadcast does not require a telegram evaluation via FC/SFC 218 SER_RCV. Only write commands may be sent as broadcast.

Modbus

- The Modbus protocol is a communication protocol that fixes a hierarchic structure with one master and several slaves.
- Physically, Modbus works with a serial half-duplex connection. There are no bus conflicts occurring, because the master can only communicate with one slave at a time.
- After a request from the master, this waits for a preset delay time for an answer of the slave. During the delay time, communication with other slaves is not possible.
- After a send command, the acknowledgement telegram must be read by a call of the FC/SFC 218 SER_RCV.
- The request telegrams send by the master and the respond telegrams of a slave have the following structure:

Telegram structure

Start sign	Slave address	Function Code	Data	Flow control	End sign

Broadcast with slave address = 0

- A request can be directed to a special slave or at all slaves as broadcast message.
- To mark a broadcast message, the slave address 0 is used.
- In opposite to a "normal" send command, the broadcast does not require a telegram evaluation via FC/SFC 218 SER_RCV.
- Only write commands may be sent as broadcast.

ASCII, RTU mode

Modbus offers 2 different transfer modes. The mode selection happens during runtime by using the FC/SFC 216 SER_CFG.

- ASCII mode: Every byte is transferred in the 2 sign ASCII code. The data are marked with a start and an end sign. This causes a transparent but slow transfer.
- RTU mode: Every byte is transferred as one character. This enables a higher data pass through as the ASCII mode. Instead of start and end sign, a time control is used.

Supported Modbus protocols

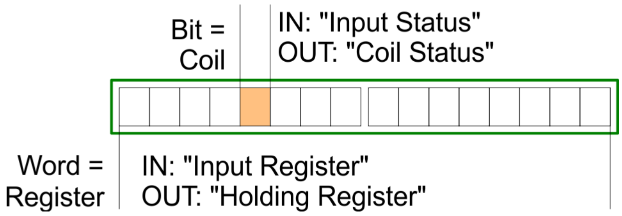
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

7.7 Modbus - Function codes

Naming convention

Modbus has some naming conventions:



- Modbus differentiates between bit and word access; bits = "Coils" and words = "Register".
- Bit inputs are referred to as "Input-Status" and bit outputs as "Coil-Status".
- word inputs are referred to as "Input-Register" and word outputs as "Holding-Register".

Range definitions

Normally the access at Modbus happens by means of the ranges 0x, 1x, 3x and 4x.

0x and 1x gives you access to digital bit areas and 3x and 4x to analog word areas.

For the CPs is not differentiating digital and analog data, the following assignment is valid:

0x - Bit area for master output data

Access via function code 01h, 05h, 0Fh

1x - Bit area for master input data

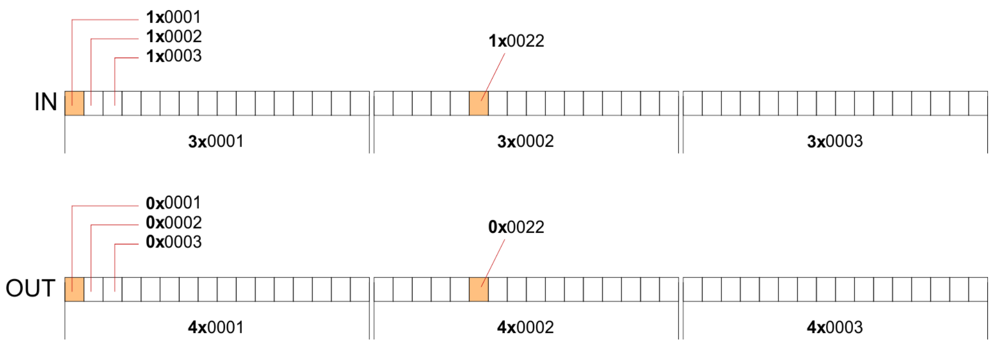
Access via function code 02h

3x - word area for master input data

Access via function code 04h

4x - word area for master output data

Access via function code 03h, 06h, 10h



A description of the function codes follows below.

Overview

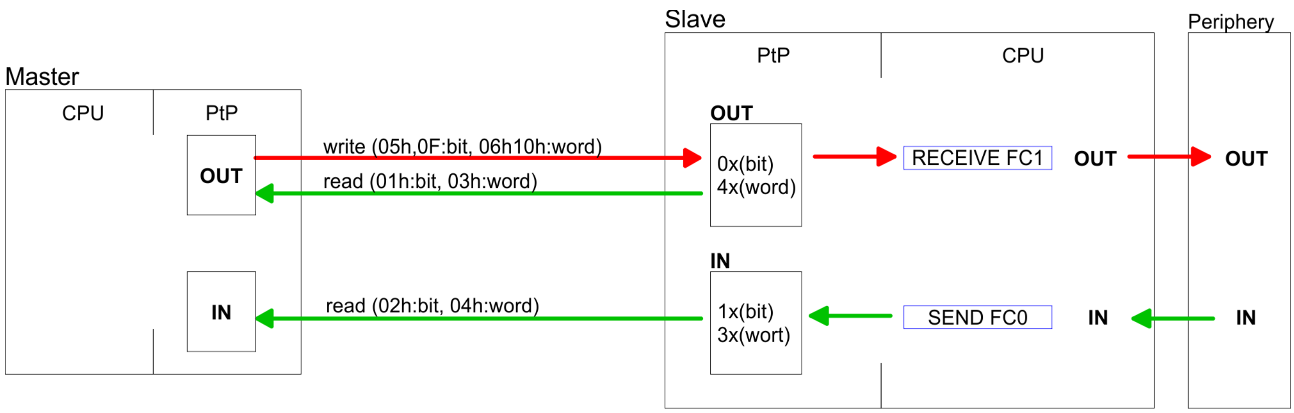
With the following Modbus function codes a Modbus master can access a Modbus slave. The description always takes place from the point of view of the master:

Code	Command	Description
01h	Read n bits	Read n bits of master output area 0x
02h	Read n bits	Read n bits of master input area 1x

Code	Command	Description
03h	Read n words	Read n words of master output area 4x
04h	Read n words	Read n words master input area 3x
05h	Write 1 bit	Write 1 bit to master output area 0x
06h	Write 1 word	Write 1 word to master output area 4x
0Fh	Write n bits	Write n bits to master output area 0x
10h	Write n words	Write n words to master output area 4x

Point of View of "Input" and "Output" data

The description always takes place from the point of view of the master. Here data, which were sent from master to slave, up to their target are designated as "output" data (OUT) and contrary slave data received by the master were designated as "input" data (IN).



Respond of the slave

If the slave announces an error, the function code is send back with an "ORed" 80h.

Without an error, the function code is sent back.

Slave answer:	Function code OR 80h	→ Error
	Function code	→ OK

Byte sequence in a word

1 word	
High-byte	Low-byte

Check sum CRC, RTU, LRC

The shown check sums CRC at RTU and LRC at ASCII mode are automatically added to every telegram. They are not shown in the data block.

Read n bits 01h, 02h

Code 01h: Read n bits of master output area 0x

Code 02h: Read n bits of master input area 1x

Command telegram

Slave address	Function code	Address 1. bit	Number of bits	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Number of read bytes	Data 1. byte	Data 2. byte	...	Check sum CRC/LRC
1byte	1byte	1byte	1byte	1byte		1word
			max. 250byte			

Read n words 03h, 04h

03h: Read n words of master output area 4x

04h: Read n words master input area 3x

Command telegram

Slave address	Function code	Address 1. bit	Number of words	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Number of read bytes	Data 1. word	Data 2. word	...	Check sum CRC/LRC
1byte	1byte	1byte	1word	1word		1word
			max. 125words			

Write 1 bit 05h

Code 05h: Write 1 bit to master output area 0x

A status change is via "Status bit" with following values:

"Status bit" = 0000h → Bit = 0

"Status bit" = FF00h → Bit = 1

Command telegram

Slave address	Function code	Address bit	Status bit	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Address bit	Status bit	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write 1 word 06h

Code 06h: Write 1 word to master output area 4x

Command telegram

Slave address	Function code	Address word	Value word	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Respond telegram

Slave address	Function code	Address word	Value word	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write n bits 0Fh

Code 0Fh: Write n bits to master output area 0x

Please regard that the number of bits has additionally to be set in byte.

Command telegram

Slave address	Function code	Address 1. bit	Number of bits	Number of bytes	Data 1. byte	Data 2. byte	...	Check sum CRC/LRC
1byte	1byte	1word	1word	1byte	1byte	1byte	1byte	1word
					max. 250byte			

Respond telegram

Slave address	Function code	Address 1. bit	Number of bits	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

Write n words 10h

Code 10h: Write n words to master output area 4x

Command telegram

Slave address	Function code	Address 1. word	Number of words	Number of bytes	Data 1. word	Data 2. word	...	Check sum CRC/LRC
1byte	1byte	1word	1word	1byte	1word	1word	1word	1word
					max. 125words			

Respond telegram

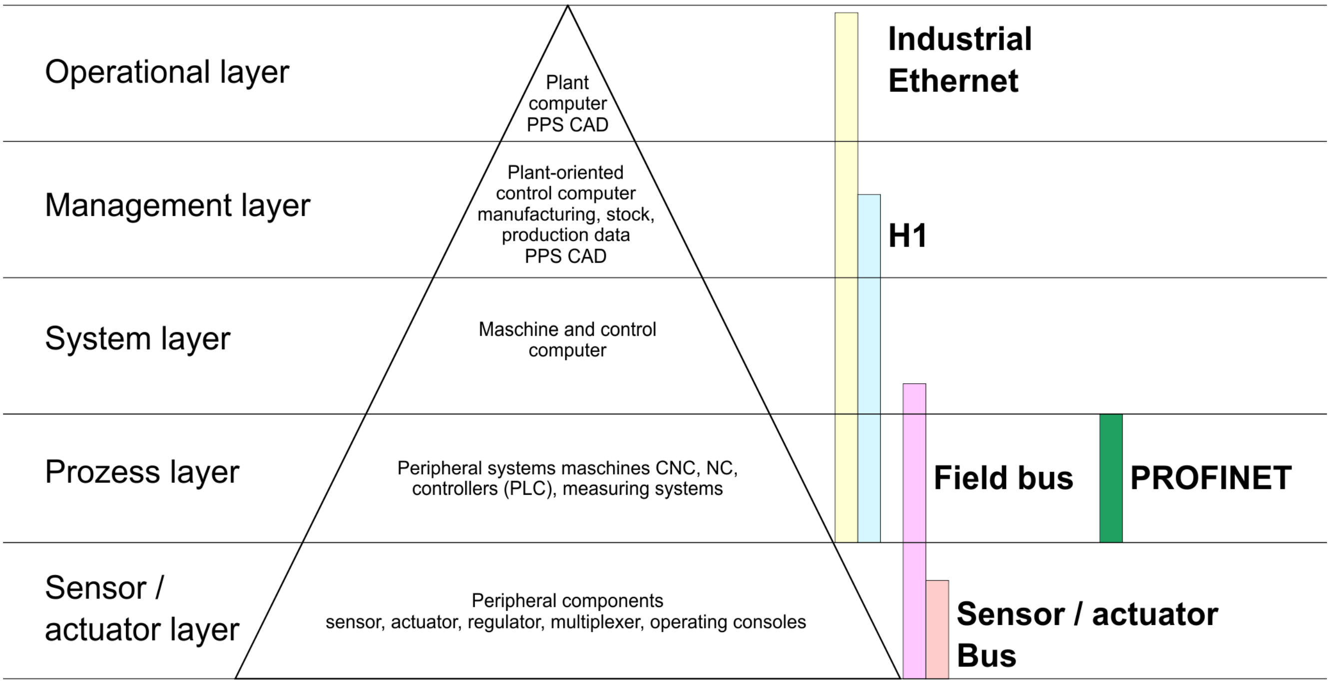
Slave address	Function code	Address 1. word	Number of words	Check sum CRC/LRC
1byte	1byte	1word	1word	1word

8 Deployment Ethernet communication - productive

8.1 Basics - Industrial Ethernet in automation

Overview

The flow of information in a company presents a vast spectrum of requirements that must be met by the communication systems. Depending on the area of business the bus system or LAN must support a different number of users, different volumes of data must be transferred and the intervals between transfers may vary, etc. It is for this reason that different bus systems are employed depending on the respective task. These may be subdivided into different classes. The following model depicts the relationship between the different bus systems and the hierarchical structures of a company:



Industrial Ethernet

Industrial Ethernet is an electrical net based on shielded twisted pair cabling or optical net based on optical fibre. Industrial Ethernet is defined by the international standard IEEE 802.3

The net access of Industrial Ethernet corresponds to IEEE 802.3 - CSMA/CD (**C**arrier **S**ense **M**ultiple **A**ccess/**C**ollision **D**etection) scheme:

- Every station "listens” on the bus cable and receives communication messages that are addressed to it.
- Stations will only initiate a transmission when the line is unoccupied.
- In the event that two participants should start transmitting simultaneously, they will detect this and stop transmitting to restart after a random delay time has expired.
- Using switches there is the possibility for communication without collisions.

8.2 Basics - ISO/OSI reference model

Overview

The ISO/OSI reference model is based on a proposal that was developed by the International Standards Organization (ISO). This represents the first step towards an international standard for the different protocols. It is referred to as the ISO-OSI layer model. OSI is the abbreviation for **O**pen **S**ystem **I**nterconnection, the communication between open systems. The ISO/OSI reference model does not represent a network architecture as it does not define the services and protocols used by the different layers. The model simply specifies the tasks that the different layers must perform. All current communication systems are based on the ISO/OSI reference model, which is defined by the ISO 7498 standard. The reference model structures communication systems into 7 layers that cover different communication tasks. In this manner the complexity of the communication between different systems is divided amongst different layers to simplify the task.

The following layers have been defined:

- Layer 7 - Application Layer
- Layer 6 - Presentation Layer
- Layer 5 - Session Layer
- Layer 4 - Transport Layer
- Layer 3 - Network Layer
- Layer 2 - Data Link Layer
- Layer 1- Physical Layer

Depending on the complexity and the requirements of the communication mechanisms a communication system may use a subset of these layers.

Layer 1 - Bit communication layer (physical layer)

The bit communication layer (physical layer) is concerned with the transfer of data bits via the communication channel. This layer is therefore responsible for the mechanical, electrical and the procedural interfaces and the physical communication medium located below the bit communication layer:

- Which voltage represents a logical 0 or a 1?
- The minimum time the voltage is present to be recognized as a bit.
- The pin assignment of the respective interface.

Layer 2 - Security layer (data link layer)

This layer performs error-checking functions for bit strings transferred between two communicating partners. This includes the recognition and correction or flagging of communication errors and flow control functions. The security layer (data link layer) converts raw communication data into a sequence of frames. This is where frame limits are inserted on the transmitting side and where the receiving side detects them. These limits consist of special bit patterns that are inserted at the beginning and at the end of every frame. The security layer often also incorporates flow control and error detection functions. The data security layer is divided into two sub-levels, the LLC and the MAC level. The MAC (**M**edia **A**ccess **C**ontrol) is the lower level and controls how senders are sharing a single transmit channel. The LLC (**L**ogical **L**ink **C**ontrol) is the upper level that establishes the connection for transferring the data frames from one device into the other.

Layer 3 - Network layer

The network layer is an agency layer. Business of this layer is to control the exchange of binary data between stations that are not directly connected. It is responsible for the logical connections of layer 2 communications. Layer 3 supports the identification of the single network addresses and the establishing and disconnecting of logical communication channels. Additionally, layer 3 manages the prior transfer of data and the error processing of data packets. IP (Internet Protocol) is based on Layer 3.

Layer 4 - Transport layer

Layer 4 connects the network structures with the structures of the higher levels by dividing the messages of higher layers into segments and passes them on to the network layer. Hereby, the transport layer converts the transport addresses into network addresses. Common transport protocols are: TCP, SPX, NWLink and NetBEUI.

Layer 5 - Session layer

The session layer is also called the communication control layer. It relieves the communication between service deliverer and the requestor by establishing and holding the connection if the transport system has a short time fail out. At this layer, logical users may communicate via several connections at the same time. If the transport system fails, a new connection is established if needed. Additionally this layer provides methods for control and synchronization tasks.

Layer 6 - Presentation layer

This layer manages the presentation of the messages, when different network systems are using different representations of data. Layer 6 converts the data into a format that is acceptable for both communication partners. Here compression/decompression and encrypting/decrypting tasks are processed. This layer is also called interpreter. A typical use of this layer is the terminal emulation.

Layer 7 - Application layer

The application layer is the link between the user application and the network. The tasks of the application layer include the network services like file, print, message, data base and application services as well as the according rules. This layer is composed from a series of protocols that are permanently expanded following the increasing needs of the user.

8.3 Basics - Terms

Network (LAN)

A network res. LAN (Local Area Network) provides a link between different stations that enables them to communicate with each other. Network stations consist of PCs, IPCs, TCP/IP adapters, etc. Network stations are separated by a minimum distance and connected by means of a network cable. The combination of network stations and the network cable represent a complete segment. All the segments of a network form the Ethernet (physics of a network).

Twisted Pair

In the early days of networking the Triaxial- (yellow cable) or thin Ethernet cable (Cheapernet) was used as communication medium. This has been superseded by the twisted-pair network cable due to its immunity to interference. The CPU has a twisted-pair connector. The twisted-pair cable consists of 8 cores that are twisted together in pairs. Due to these twists this system is provides an increased level of immunity to electrical interference. For linking please use twisted pair cable which at least corresponds to the category 5. Where the coaxial Ethernet networks are based on a bus topology the twisted-pair network is based on a point-to-point scheme. The network that may be established by means of this cable has a star topology. Every station is connected to the star coupler (hub/switch) by means of a separate cable. The hub/switch provides the interface to the Ethernet.

Hub (repeater)

The hub is the central element that is required to implement a twisted-pair Ethernet network. It is the job of the hub to regenerate and to amplify the signals in both directions. At the same time it must have the facility to detect and process segment wide collisions and to relay this information. The hub is not accessible by means of a separate network address since it is not visible to the stations on the network. A hub has provisions to interface to Ethernet or to another hub res. switch.

Switch

A switch also is a central element for realizing Ethernet on Twisted Pair. Several stations res. hubs are connected via a switch. Afterwards they are able to communicate with each other via the switch without interfering the network. An intelligent hardware analyses the incoming telegrams of every port of the switch and passes them collision free on to the destination stations of the switch. A switch optimizes the bandwidth in every connected segment of a network. Switches enable exclusive connections between the segments of a network changing at request.

8.4 Basics - Protocols

Overview

Protocols define a set of instructions or standards that enable computer to establish communication connections and exchange information as error free as possible. A commonly established protocol for the standardization of the complete communication is the ISO/OSI layer model. *Chap. 8.2 'Basics - ISO/OSI reference model' page 174*

The following protocols are used:

- Communication connections
  – Siemens S7 connections
- Open communication
  – UDP according to RFC 793
  – ISO on TCP according to RFC 1006
  – UDP according to RFC 768

Siemens S7 connections

With the Siemens S7 connection large data sets may be transferred between PLC systems based on Siemens STEP®7. Here the stations are connected via Ethernet. Precondition for the Siemens S7 communication is a configured connection table, which contains the defined connections for communication. This can be configured in *SPEED7 Studio*.

Properties:

- A communication connection is specified by a connection ID for each connection partner.
- The acknowledgement of the data transfer is established from the partner station at level 7 of the ISO/OSI reference model.
- At the PLC side product specific FB/SFB handling blocks are necessary for data transfer for the Siemens S7 connections.

More information about the usage of these blocks may be found in the manual "SPEED7 Operation List".

Open communication

In the *Open communication* the communication takes place via the user program by means of handling blocks. These blocks are part of the *SPEED7 Studio*. These can be found in the *'Catalog'* at *'Blocks'*.

- *Connection-oriented protocols:*
  Connection-oriented protocols establish a (logical) connection to the communication partner before data transmission is started. And if necessary they terminate the connection after the data transfer was finished. Connection-oriented protocols are used for data transmission when reliable, guaranteed delivery is of particular importance. Also the correct order of the received packets is ensured. In general, many logical connections can exist on one physical line. The following connection-oriented protocols are supported with FBs for open communication via industrial Ethernet:
  – *TCP native accord. to RFC 793:*
    During data transmission, no information about the length or about the start and end of a message is transmitted. However, the receiver has no means of detecting where one message ends in the data stream and the next one begins. The transfer is stream-oriented. For this reason, it is recommended that the data length of the FBs is identical for the sending and receiving station. If the number of received data does not fit to the preset length you either will get not the whole data, or you will get data of the following job.
  – *ISO on TCP according to RFC 1006:*
    During data transmission, information on the length and the end of the message is also transmitted. The transfer is block-oriented If you have specified the length of the data to be received greater than the length of the data to be sent, the receive block will copy the received data completely into the receive range.
- *Connection-less protocol:*
  There is thus no establishment and termination of a connection with a remote partner. Connection-less protocols transmit data with no acknowledge and with no reliable guaranteed delivery to the remote partner.
  – *UDP according to RFC 768:*
    In this case, when calling the sending block you have to specify the address parameters of the receiver (IP address and port number). During data transmission, information on the length and the end of the message is also transmitted. Analog after finishing the receive block you get a reference to the address parameter of the sender (IP address and port no.) In order to be able to use the sending and receiving blocks first you have to configure the local communications access point at both sides. With each new call of the sending block, you reference the remote partner by specifying its IP address and its port number.

8.5 Basics - IP address and subnet

IP address structure

Exclusively IPv4 is supported. At IPv4 the IP address is a 32bit address that must be unique within the network and consists of 4 numbers that are separated by a dot. Every IP address is a combination of a *Net-ID* and a *Host-ID* and has the following

Structure: **xxx.xxx.xxx.xxx**

Range: 000.000.000.000 to 255.255.255.255

Net-ID, Host-ID

The **Net**work-ID identifies a network res. a network controller that administrates the network. The Host-ID marks the network connections of a participant (host) to this network.

Subnet mask

The Host-ID can be further divided into a *Subnet-ID* and a new *Host-ID* by using a bit for bit AND assignment with the Subnet mask.

The area of the original Host-ID that is overwritten by 1 of the Subnet mask becomes the Subnet-ID, the rest is the new Host-ID.

Subnet mask	binary all "1"		binary all "0"
IPv4 address	Net-ID	Host-ID	
Subnet mask and IPv4 address	Net-ID	Subnet-ID	new Host-ID

Address at first start-up

At the first commissioning the Ethernet PG/OP channel and the NET-CP have no IP address.

How IP address data are assigned to the Ethernet PG/OP channel *Chap. 4.7 'Hardware configuration - Ethernet PG/OP channel' page 77*.

How IP address data are assigned to the NET-CP *Chap. 8.8 'Commissioning and initialization' page 180*.

Address classes

For IPv4 addresses there are five address formats (class A to class E) that are all of a length of 4byte = 32bit.

Class A	0	Network-ID (1+7bit)	Host-ID (24bit)
Class B	10	Network-ID (2+14bit)	Host-ID (16bit)
Class C	110	Network-ID (3+21bit)	Host-ID (8bit)
Class D	1110	Multicast group	
Class E	11110	Reserved	

The classes A, B and C are used for individual addresses, class D for multicast addresses and class E is reserved for special purposes. The address formats of the 3 classes A, B, C are only differing in the length of Network-ID and Host-ID.

Private IP networks

These addresses can be used as net-ID by several organizations without causing conflicts, for these IP addresses are neither assigned in the Internet nor are routed in the Internet. To build up private IP-Networks within the Internet, RFC1597/1918 reserves the following address areas:

Network class	from IP	to IP	Standard subnet mask
A	10.0.0.0	10.255.255.255	255.0.0.0
B	172.16.0.0	172.31.255.255	255.255.0.0

Network class	from IP	to IP	Standard subnet mask
C	192.168.0.0	192.168.255.255	255.255.255.0
(The Host-ID is underlined.)			

Reserved Host-IDs

Some Host-IDs are reserved for special purposes.

Host-ID = "0"	Identifier of this network, reserved!
Host-ID = maximum (binary complete "1")	Broadcast address of this network

Never choose an IP address with Host-ID=0 or Host-ID=maximum! (e.g. for class B with subnet mask = 255.255.0.0, the "172.16.0.0" is reserved and the "172.16.255.255" is occupied as local broadcast address for this network.)

8.6 Basics - MAC address and TSAP

MAC address

There is a unique MAC address (**M**edia **A**ccess **C**ontrol) necessary for each CP. Usually a module is labelled with its MAC address by the manufacturer. This address should be used for project engineering of the CP. The MAC address has a length of 6bytes. On delivery the first three bytes specify the manufacturer. These bytes are assigned by the IEEE committee. The last 3 bytes may be assigned by the manufacturer. In a network several stations with the same MAC address may not exist. The MAC address may be changed at any time. You will get a valid MAC address from your network administrator.

- Broadcast address
  - The MAC address, with which all bits are set to 1, is:
    FF-FF-FF-FF-FF-FF
    This address is used as Broadcast address and addresses all participants in the net.
- Address at first start-up
  - Each CP of a CPU has an unique MAC address. This may be found on a label beneath the front flap.

TSAP

TSAP means **T**ransport **S**ervice **A**ccess **P**oint. ISO transport connections support TSAP length of 1...16byte. TSAPs may be entered in ASCII format or hexadecimal.

Address parameters

Station A				Station B
remote TSAP	→	ISO transport connection	→	local TSAP
local TSAP	←		←	remote TSAP
MAC address A				MAC address B

An ISO transport connection is specified by a local and a remote connection endpoint. The TSAPs of an ISO transport connection must match as follows:

- Remote TSAP (in CP) = local TSAP (in destination station)
- Local TSAP (in CP) = remote TSAP (in destination station)

8.7 Fast introduction

Overview

At the first commissioning respectively after an overall reset with PowerON again of the CPU, the Ethernet PG/OP channel and the NET-CP have no IP address. These are only reachable by its MAC address. By means of the MAC address, which is printed at the front with the sequence address NET-CP and beneath address Ethernet PG/OP channel, you can assign IP address data to the according component. The assignment takes place directly via the device configuration in *SPEED7 Studio*.

Steps of configuration

For the configuration of the NET-CP for productive connections should be done by the follow approach:

- Assembly and commissioning
- Hardware configuration - CPU
- Configure connections
  - Siemens S7 connections
    (Configuration happens via *'Device and networking'* in the *SPEED7 Studio*, the communication via product specific handling blocks)
  - Open communication
    (Configuration and communication happens by handling blocks)
- Transfer of the entire project to the CPU

8.8 Commissioning and initialization

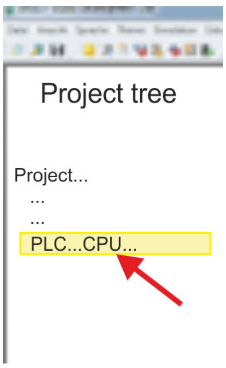
Assembly and commissioning

1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the Ethernet jack of the NET-CP (X6) to Ethernet.
4. Switch on the power supply.
   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization"

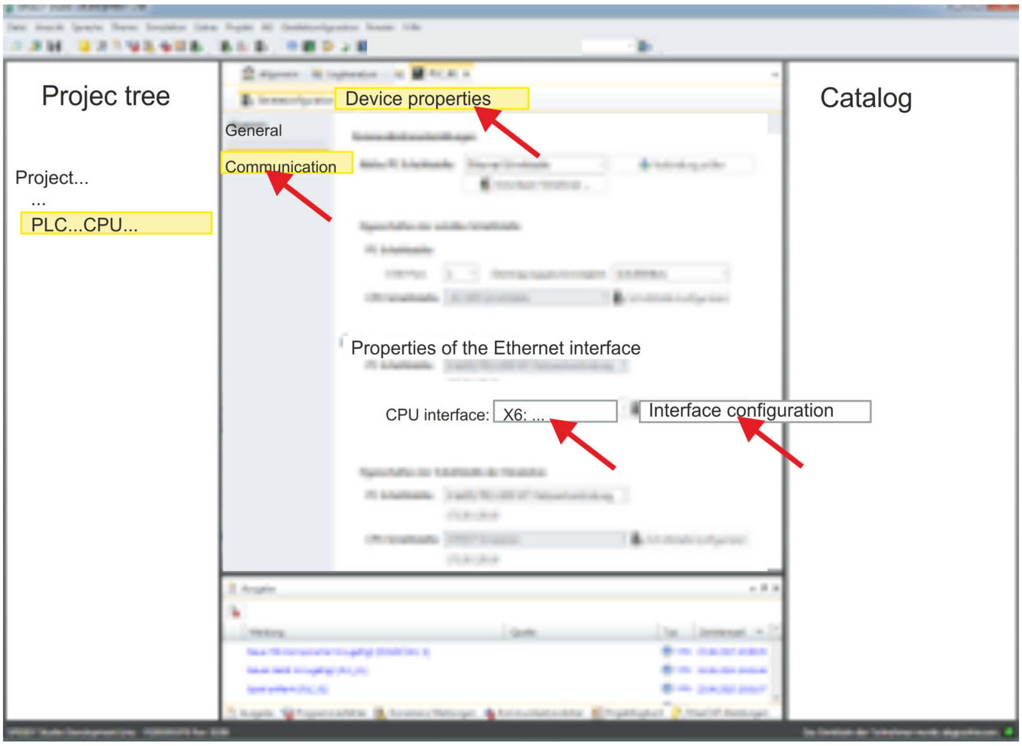
You get valid IP address parameters from your system administrator. The assignment of the IP address data happens in the *SPEED7 Studio*with the following proceeding:

1. Start the *SPEED7 Studio* with your project.
2. Click in the *Project tree* at the CPU *'PLC ... CPU ...'*



3. Select *'Context menu ➔ Device properties'*.

⇨ The dialogue *'Device properties'* opens.



4. Here click at *'Communication'*
5. Select at *'Properties Ethernet interface'* as *'CPU-interface'* the interface *'X6:...'*.
6. Click at the button [Interface configuration].
7. Enter the wanted IP address data and confirm with [OK].

⇨ The IP address data are stored in your current project. After transferring your project, the NET-CP can be accessed via the set IP address data.

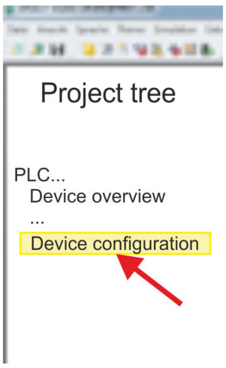
8.9 Hardware configuration - CPU

Precondition

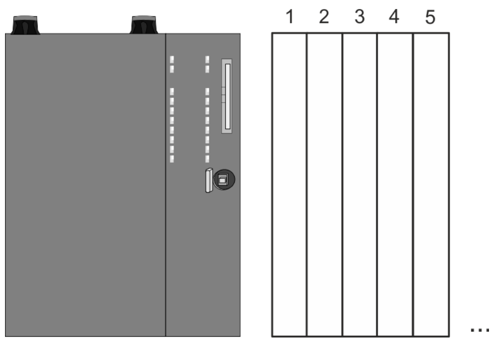
> *For project engineering a thorough knowledge of the SPEED7 Studio is required!*

Proceeding

1. Start the *SPEED7 Studio*.
2. Create a new project in the *Work area* with *'New project'*.
   ⇨ A new project is created and the view *'Devices and networking'* is shown.
3. Click in the *Project tree* at *'Add new device ...'*.
   ⇨ A dialog for device selection opens.



4. Select from the *'Device templates'* your CPU and click at [OK].
   ⇨ The CPU is inserted in *'Devices and networking'* and the *'Device configuration'* is opened.



Device configuration

Slot	Module	...	...	...	...
0	CPU 015-CEFNR00				
-X1	PG_OP_Ethernet				
-X3	MPI interface				
...	...			...	

8.10 Configure Siemens S7 connections

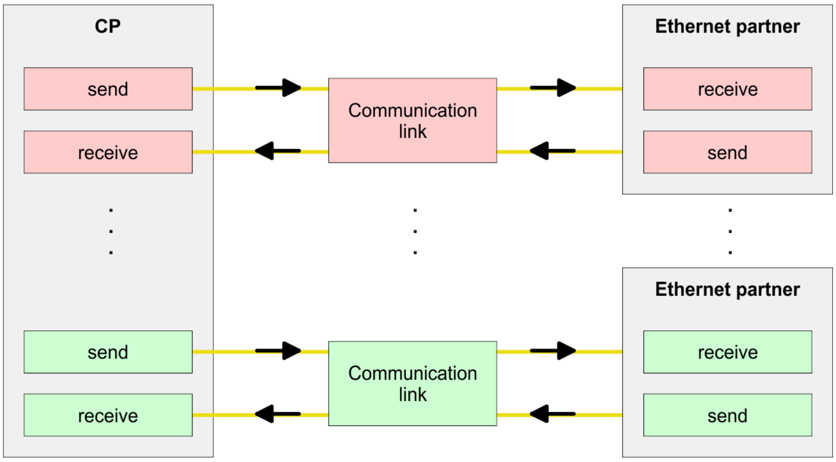
Overview

The configuration of S7 connections, i.e. the "link-up" between the stations happens in the *SPEED7 Studio* at *'Devices and networking'*. Here you can configure in tabular form communication connections In addition, the physical connections between the stations are graphically shown. A communication connection enables the program controlled communication between two participants at the Industrial Ethernet. The communication is controlled by the user program with product specific handling blocks. To use this blocks, configured communication connections are always necessary in the active station.

Properties communication connection

The following properties are characterizing a communication connection:

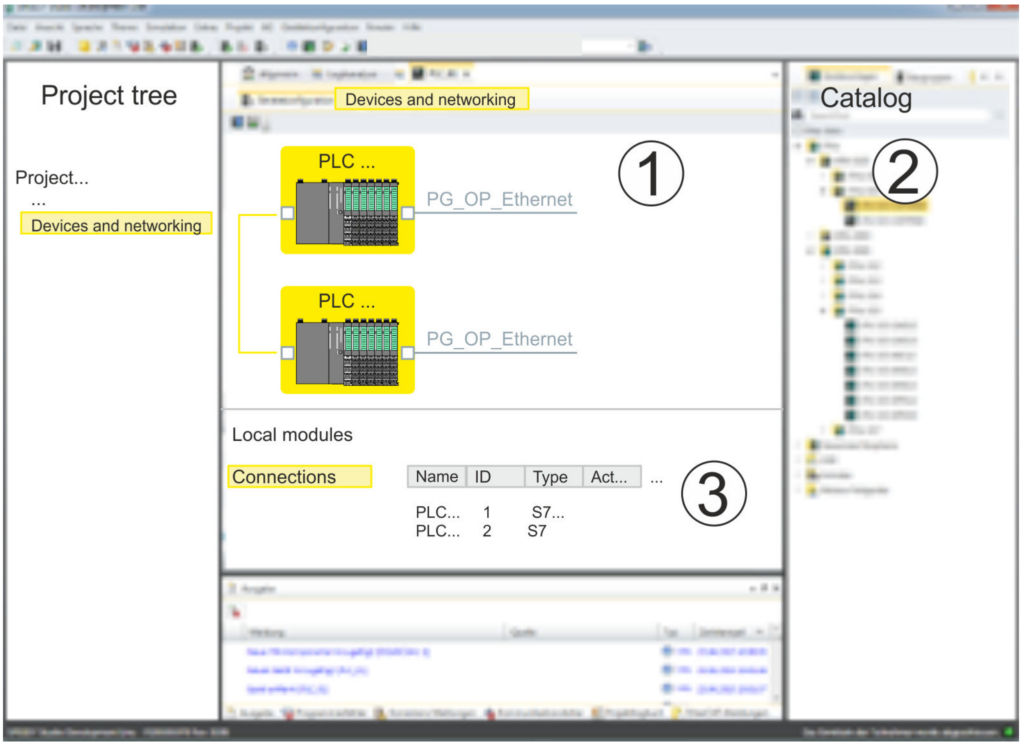
- One station always executes an active connection establishment.
- Bi-directional data transfer (Send and receive on one connection)
- Both participant have equal rights, i.e. every participant may initialize the send res. receive process event controlled.
- Except of the UDP connection, at a communication connection the address of the communication partner is set via the project engineering. Here the connection is active established by one station.



Working environment *'Devices and networking'*

For the configuration of connections a thorough knowledge of the *SPEED7 Studio* is required! The following passage only describes the basic usage *'Devices and networking'*. More information can be found in the online help respectively in documentation. After loading your project, *'Devices and networking'* can directly be called via the *Project tree*.

The working environment of *'Devices and networking'* has the following structure:

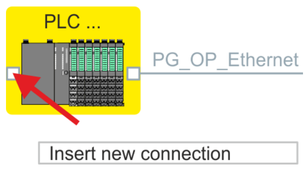


1 *Net view*: All stations and networks are displayed in a graphic view. By clicking on the according component you may access and alter the concerning properties.
2 *Catalog*: This area displays all available modules respectively net objects are shown in a directory view. By dragging a wanted object to the *net view* you may include further net objects.
3 *Connection table*: As soon as you select a module in the *net view*, the configured connections of this module are listed in the connection table. When you select a connection, via the context menu there is the option to edit, delete or to create a new connection.

Link up stations and configure connections

With *SPEED7 Studio* you can link up the communicating stations. The link up takes place in *Devices and networking* via the context menu of the NET-CP net marker with the following approach:

1. Select with the mouse the NET-CP net marker

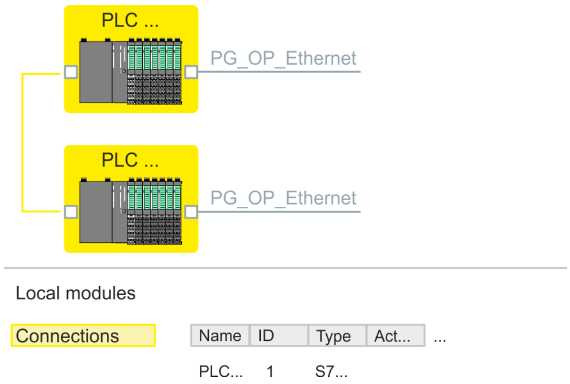


2. Open with *'Context menu ➔ Insert new connection'* the dialog to establish a new connection.
3. Choose partner station and connection type and confirm with [OK].
   - *Connection partner - Station of your project*
     Each station configured in *SPEED7 Studio* is listed in the table of connection partner.
   - *Connection partner - unspecified*
     Here the connection partner may exist in the current project or in an unknown project. Connection jobs to an unknown project must be defined by an unique IP address. Due to this allocation the connection remains *unspecified*.

   ⇨ A dialogue for setting the connection parameters opens.

4. Enter the according parameters and confirm with [OK].

⇨ The connection is established, listed in the *Connection table* and shown as graphical connection line between the stations at *Devices and networking*.



5. Configure in this way further connections. You also can edit you connections via the *Connection table* by selecting a connection and executing an according function via context menu like *'Insert new connection'*.

Connection types

With this CPU exclusively Siemens S7 connections may be configured with *SPEED7 Studio*.

Siemens S7 connection

- For data transfer with Siemens S7 connections the FB/SFB handling blocks are necessary; the deployment is described in the manual "Operation list" of your CPU.
- At Siemens S7 connections the communication connections are specified by a connection ID for each communication partner.
- A connection is specified by the local and partner connection end point.
- At Siemens S7 connections the TSAPs must be congruent crosswise.

The following parameters define a connection end point:

Station A				Station B
remote TSAP	→	Siemens	→	local TSAP
local TSAP	←	S7-Verbindung	←	remote TSAP
ID A				ID B

Combination options with deployment of the FB/SFB handling blocks

Connection partner	Connection establishing	Connection
specified in the current project	active/passive	specified
unspecified	active	specified
	passive	unspecified

In the following every relevant parameter of a Siemens S7 connection is described:

- General
  – *End point:*
    Here you may define how the connection is to be established. Since the *SPEED7 Studio* can identify the communication options by means of the end points, some options are already preset and may not be changed.
  – *Name:*
    Here you can enter a name for your station
  – *Interface:*
    Here you can select the interfaces of your local and partner station.
  – *Address:*
    Here you can enter the MPI address of your local and partner station.
- Local ID
  – The ID is the link to your PLC program. The ID must be identical to the ID of the call interface of the FB/SFB handling block. Here you also can find the parameters *'ID'* and *'LADDR'*, which you have to specify in the handing blocks.
- Special features
  – *Active connection establishment:*
    An established connection is precondition for data transfer. By activating the option *'Establish an active connection'* the local station establishes the connection. Please regard not every station is able to active establish a connection. Here the job is to be made by the partner station.
- Address details
  – *Rack/slot:*
    Here you will find information about rack and slot of the local and the partner station.
  – Via this button a dialog window opens to show or enter address information of the local or partner station.
  – *Connection resources*
    The connection resource is part of the TSAP of the local station respectively of the partner. Not every connection resource may be used for every connection type. Depending on the connection partner and the connection type the range of values is limited respectively the connection resource is fix specified.
  – *TSAP:*
    With Siemens S7 connections a TSAP is automatically generated of the connection resource (one-way/two-way) and state of place (rack/slot).

Function blocks

FB/SFB	Designation	Description
FB/SFB 12	BSEND	Sending data in blocks FB/SFB 12 BSEND sends data to a remote partner FB/SFB of the type BRCV (FB/SFB 13). The data area to be transmitted is segmented. Each segment is sent individually to the partner. The last segment is acknowledged by the partner as it is received, independently of the calling up of the corresponding FB/SFB/FB BRCV. Due to the segmentation up to 65534byte data can be transferred with one send job.
FB/SFB 13	BRCV	Receiving data in blocks: The FB/SFB 13 BRCV can receive data from a remote partner FB/SFB of the type BSEND (FB/SFB 12). The parameter R_ID of both FB/SFBs must be identical. After each received data segment an acknowledgement is sent to the partner FB/SFB and the LEN parameter is updated.

FB/SFB	Designation	Description
FB/SFB 14	GET	Remote CPU read: The FB/SFB 14 GET can be used to read data from a remote CPU. The respective CPU must be in RUN mode or in STOP mode.
FB/SFB 15	PUT	Remote CPU write: The FB/SFB 15 PUT can be used to write data to a remote CPU. The respective CPU may be in RUN mode or in STOP mode.

8.11 Configure Open Communication

Connection-oriented protocols

- Connection-oriented protocols establish a (logical) connection to the communication partner before data transmission is started.
- And if necessary they terminate the connection after the data transfer was finished.
- Connection-oriented protocols are used for data transmission when reliable, guaranteed delivery is of particular importance.
- In general, many logical connections can exist on one physical line.

The following connection-oriented protocols are supported with FBs for open communication via Industrial Ethernet:

- *TCP/IP native according to RFC 793 (connection types 01h and 11h):*
  - During data transmission, no information about the length or about the start and end of a message is transmitted.
  - The receiver has no means of detecting where one message ends in the data stream and the next one begins.
  - The transfer is stream-oriented. For this reason, it is recommended that the data length of the FBs is identical for the sending and receiving station.
  - If the number of received data does not fit to the preset length you either will get not the whole data, or you will get data of the following job. The receive block copies as many bytes into the receive area as you have specified as length. After this, it will set NDR to TRUE and write RCVD_LEN with the value of LEN. With each additional call, you will thus receive another block of sent data.
- *ISO on TCP according to RFC 1006:*
  - During data transmission, information on the length and the end of the message is also transmitted.
  - The transfer is block-oriented
  - If you have specified the length of the data to be received greater than the length of the data to be sent, the receive block will copy the received data completely into the receive range. After this, it will set NDR to TRUE and write RCVD_LEN with the length of the sent data.
  - If you have specified the length of the data to be received less than the length of the sent data, the receive block will not copy any data into the receive range but instead will supply the following error information: ERROR = 1, STATUS = 8088h.

Connection-less protocol

- There is thus no establishment and termination of a connection with a remote partner.
- Connection-less protocols transmit data with no acknowledge and with no reliable guaranteed delivery to the remote partner.

The following connection-oriented protocol is supported with FBs for open communication via Industrial Ethernet:

- *UDP according to RFC 768 (with connection type 13h):*
  - In this case, when calling the sending block you have to specify the address parameters of the receiver (IP address and port number).
  - During data transmission, information on the length and the end of the message is also transmitted.
  - In order to be able to use the sending and receiving blocks first you have to configure the local communications access point at both sides.
  - With each new call of the sending block, you re-reference the remote partner by specifying its IP address and its port number.
  - If you have specified the length of the data to be received greater than the length of the data to be sent, the receive block will copy the received data completely into the receive range. After this, it will set NDR to TRUE and write RCVD_LEN with the length of the sent data.
  - If you have specified the length of the data to be received less than the length of the sent data, the receive block will not copy any data into the receive range but instead will supply the following error information: ERROR = 1, STATUS = 8088h.

Handling blocks

Those in the following listed UTDs and FBs serve for "open communication" with other Ethernet capable communication partners via your user program. These blocks are part of the *SPEED7 Studio*. You will find these in the *'Standard Library'*. Please consider when using the blocks for open communication that the partner station does not have to be configured with these blocks. This can be configured with AG_SEND/AG_RECEIVE or IP_CONFIG.

More information about the usage of these blocks may be found in the manual "SPEED7 Operation List".

UDTs

FB	Designation	Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006	Connectionless protocol: UDP as per RFC 768
UDT 65	TCON_PAR	Data structure for assigning connection parameters	Data structure for assigning parameters for the local communications access point
UDT 66	TCON_ADR		Data structure for assigning addressing parameters for the remote partner

FBs

FB	Designation	Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006	Connectionless protocol: UDP as per RFC 768
FB 63	TSEND	Sending data	
FB 64	TRCV	Receiving data	
FB 65	TCON	Establishing a connection	Configuring the local communications access point
FB 66	TDISCON	Terminating a connection	Closing the local communications access point

FB	Designation	Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006	Connectionless protocol: UDP as per RFC 768
FB 67	TUSEND	Sending data - UDP	Sending data
FB 68	TURCV	Receiving data - UDP	Receiving data

9 Ethernet communication - EtherCAT

9.1 Basics EtherCAT

9.1.1 General

Field buses were established for many years in the automation technology. Since higher speeds are required but the technical limits of this technology have already been reached, new solutions needed to be found.

At least in theory, the Ethernet, which is familiar to all of us from the office world, is fast with its 100Mbit/s speed, which is available everywhere today. However, these networks do not offer real-time capability due to the kind of cabling that they use and the rules governing access rights. This effect was corrected with EtherCAT®.

EtherCAT®

- For EtherCAT® is valid: EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- EtherCAT means Ethernet for Controller and Automation Technology. It was originally developed by Beckhoff Automation GmbH and is now supported and further developed by the EtherCAT Technology Group (ETG). ETG is the word biggest international user and producer connection for industrial Ethernet
- EtherCAT is an open Ethernet based field bus system, which is standardized at the IEC.
- As open field bus system EtherCAT matches the user profile for the part of industrial real-time systems.
- In opposition to the normal Ethernet communication at EtherCAT the data exchange of I/O data takes place during the frame passes the coupler with 100Mbit/s in full-duplex. Since in this way a frame to send and receive direction reaches the data of many stations, EtherCAT has a rate of user data of over 90%.
- The EtherCAT protocol, which is optimized for process data, is directly transported with the Ethernet frame. This again can consist of several sub-frames, which serve for a storage area of the process image.

Transfer medium

EtherCAT uses Ethernet as transfer medium Standard CAT5 cables are used. Here distances of about 100m between 2 stations are possible.

Only EtherCAT components may be used in an EtherCAT network. For topologies, which depart from the line structure, the corresponding EtherCAT components are necessary. Hubs may not be used.

Communication principle

At EtherCAT the master sends a telegram to the first station. The station takes its data from the current data stream, inserts its answer data and sends the frame to the succeeding station. Here the frame is handled with the same way.

When the frame has reached the last station this recognizes that no further is connected and sends the frame back to the master. Here the telegram is sent through every station via the other pair of leads (full-duplex). Due to the plug sequence and the use of the full-duplex technology EtherCAT represents a logical ring.

EtherCAT State Machine

Via the EtherCAT State Machine the state of the EtherCAT members is controlled.

Object dictionary (SDOs)

In the object directory the parameter, diagnostics, interrupt or other data are listed, which may be written or read via EtherCAT. The object directory may be accessed by the SDO information service. Additionally the object directory may be found in the device master file.

Process data (PDOs)

The EtherCAT data link layer is optimized for the fast transfer of process data. Here it is specified how the process data of the device are assigned to the EtherCAT process data and how the application of the device is synchronized to the EtherCAT cycle. The mapping of the process data happens by PDO mapping and by Sync-Manager-PDO-Assign objects. These describe, which objects of the object directory are transferred as object data via EtherCAT. The cycle time to transfer the process data via EtherCAT and how this is synchronized for the transfer is specified with the Sync-Manager-Communication objects.

Emergencies

Via Emergencies diagnostics, process events and errors at state change of the State Machine may be transferred.

Status messages, which show the current state of the device, should directly be transferred within the process data.

Distributed clocks (DC)

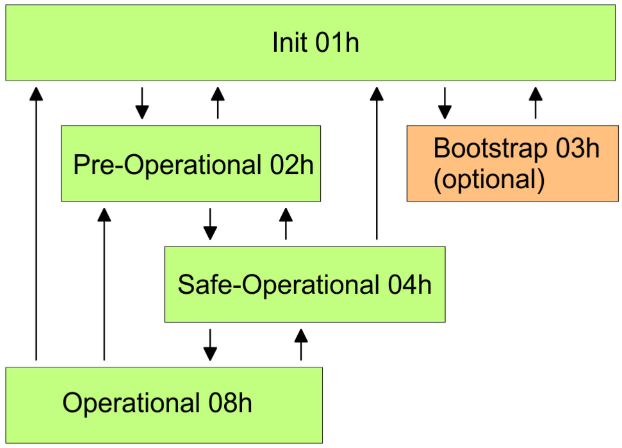
Due to the run time of an EtherCAT frame at the bus, the outputs of the EtherCAT slave stations are activated at different times and the inputs are read at different times. For an isochronous access to the process data EtherCAT provides the "distributed clock" functionality. In EtherCAT "**D**istributed **C**locks" means a logical combination of "clocks", which are located in the EtherCAT devices With this there is the possibility to locally provide a synchronized time in each bus device. With DC current output values are activated in equal time on the slave stations, and the input values read at precisely this time. This moment is called *Sync* signal. If an EtherCAT device supports the DC, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This *reference clock* so represents the system time. Among others there are the following DC setting parameters:

- Master/bus shift
  - Master shift: Based on DC the reference clock takes the "master" function, i.e. all DC slaves are adjusted based on the reference clock.
  - Bus Shift: Based on DC the reference clock takes the "slave" function, i.e. the EtherCAT master specifies how fast or how slow the reference clock has to run.
- Continuous Propagation Compensation
  - A command (datagram) will be inserted in the cyclic frame which allows the EtherCAT master to measure and compensate the propagation delay time by time.
- Sync Window Monitoring
  - A command (datagram) will be inserted in the cyclic frame which allows the EtherCAT master to determine the sync state ((*in-sync* or *out-of-sync*) of the system.

9.1.2 EtherCAT State Machine

States

In each EtherCAT communication device a *state machine* is implemented. For each state there is defined which communication service is active via EtherCAT. The state machine of the slave station is controlled by the state machine of the EtherCAT master.



Init - 01h

After power-on the EtherCAT members are in state *Init*. There is neither mailbox nor process data communication possible. The EtherCAT master initializes the SyncManager channels 0 and 1 for the mailbox communication.

Pre-Operational (Pre-Op) - 02h

The EtherCAT master initializes the SyncManager channels for process data (starting with SyncManager channel 2), the FMMU channels and the PDO mapping respectively the SyncManager PDO assignment. Further in this state the settings for process data transfer and the module-specific parameters, which deviate from the default values are transferred. During the transition from *Init* to *Pre-Op* the EtherCAT slave checks whether the mailbox was correctly initialized. In the state *Pre-Op* mailbox communication and Ethernet over EtherCAT (EoE) are possible but the process data communication is blocked.

Safe-Operational (Safe-Op) - 04h

In *Safe-Op* the input data are cyclically updated but the outputs are de-activated. With the transition from *Pre-Op* to *Safe-Op* the EtherCAT slave checks if the SyncManager channels for process data communication are correct. Before it acknowledges the state change, the EtherCAT slave copies current input data to the corresponding DP RAM areas of the EtherCAT slave controller. In the state *Safe-Op* mailbox and process data communication is possible.

Operational (Op) - 08h

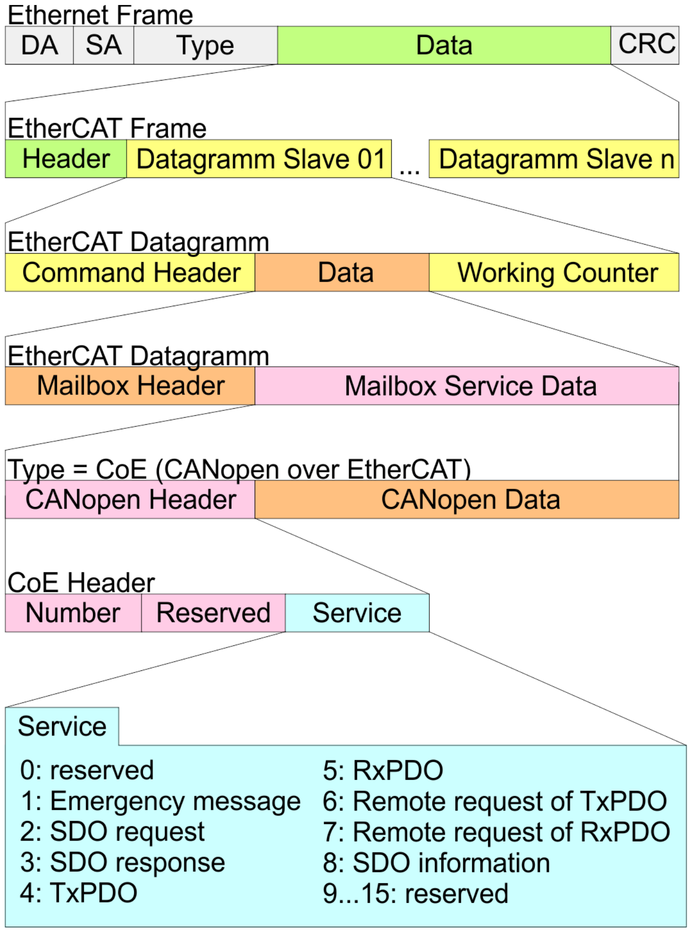
In the state *Op* the input data are cyclically updated and the EtherCAT master sends output data to the EtherCAT slave. The EtherCAT slave copies the output data of the master to its outputs and return input data to the EtherCAT master. In this state process data and mailbox communication is possible.

Bootstrap - option (Boot) - 03h

In state *Boot* the firmware of an EtherCAT slave may be updated via the EtherCAT master. This state may only be reached via Init. In the state *Boot* is mailbox communication via the protocol File-Access over EtherCAT (FoE) possible. Other mailbox and process data communications are de-activated.

9.1.3 CoE - CANopen over Ethernet

CoE means CANopen over EtherCAT. With CANopen you get a standard user interface, which makes a simplified system structure possible with most different devices. With CoE the device parameters may comfortably be accessed and data were may be read or written at the same time. Real-time data may be read by PDOs an the parametrization happens by SDOs. Further there are emergency objects available.



DA Destination address
SA Source address
CRC Checksum

9.2 Commissioning and start-up behavior

9.2.1 Assembly and commissioning

1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect your EtherCAT master to EtherCAT.
4. Switch on the power supply.

9.2.2 Start-up behavior

Preconditions for start-up

After PowerON and start-up (incl. OB100) the CPU is switched to RUN. This brings the EtherCAT master to *Op* state and he requests the *Op* state from its connected EtherCAT slave stations. Before the OB1 is called, the CPU waits for a defined time, that the EtherCAT slave stations have changed to*Op* state. You can specify the *Monitoring time* via the CPU parameter *'Transfer of parameters to modules'* in the property register *'start-up'*.

Using the EtherCAT master the following start-up behavior is distinguished. The terms and conditions can be found in the following table:

- **CPU switches to RUN, if topology is OK**
  The CPU waits for all the slaves, which mandatory have to exist, maximum until the *Monitoring time* expires and then switches to RUN. The topology mus be OK.
- **CPU switches to RUN mode regardless of topology or optional slaves**
  The CPU waits for all the slaves, which mandatory have to exist, maximum until the *Monitoring time* expires and then switches to RUN regardless of topology or optional slaves.

Truth table

Is the CPU parameter: *'Start-up is preset configuration does not match actual configuration'* activated?	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Are all the mandatory slaves configured?	x	x	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	Y	Y
Are there optional slaves configured (hot connect group)?	x	x	N	N	Y	Y	Y	Y	x	x	Y	Y	x	x	N	N	N	N	x	x
Do all the mandatory slaves exist?	x	x	Y	Y	N	N	Y	Y	x	x	Y	Y	x	x	x	x	x	x	N	N
Do optional slaves exist (not all must exist)?	x	x	N	N	Y	Y	Y	Y	x	x	Y	Y	x	x	x	x	x	x	x	x
Is there at least one mandatory slave with a wrong module?	x	x	N	N	N	N	N	N	Y	Y	x	x	x	x	x	x	x	x	x	x
Is there at least one optional slave with a wrong module?	x	x	N	N	N	N	N	N	x	x	Y	Y	x	x	x	x	x	x	x	x
Does at least on not configured slave exist?	x	x	N	N	N	N	N	N	x	x	x	x	Y	Y	Y	Y	N	N	x	x
Is DC (distributed clocks) activated?	N	Y	N	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y
Are all the slaves in-sync?	x	Y	x	Y	x	Y	x	Y	N	x	N	x	N	x	N	x	x	Y	N	x
Is the master in-sync to the first slave?	x	Y	x	Y	x	Y	x	Y	x	N	x	N	x	N	x	N	x	Y	x	N
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
CPU switches to RUN, if topology is OK.	Y	Y																		
CPU switches to RUN mode regardless of topology or optional slaves.			Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	Y	Y	N	N
Yes: Y | No: N | not relevant: X																				

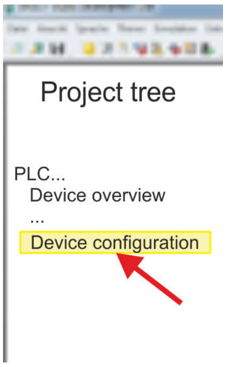
9.3 Hardware configuration - CPU

Please note that this functionality is not supported by the Siemens TIA Portal!

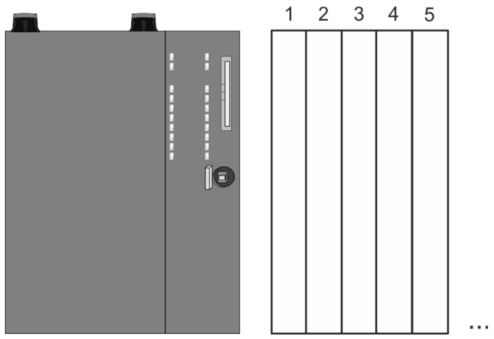
The CPU is to be configured in the *SPEED7 Studio*. With the integrated *SPEED7 EtherCAT Manager* you can configure your EtherCAT network.

Proceeding

1. Start the *SPEED7 Studio*.
2. Create a new project in the *Work area* with *'New project'*.
   ⇨ A new project is created and the view *'Devices and networking'* is shown.
3. Click in the *Project tree* at *'Add new device ...'*.
   ⇨ A dialog for device selection opens.



4. Select from the *'Device templates'* your CPU and click at [OK].
   ⇨ The CPU is inserted in *'Devices and networking'* and the *'Device configuration'* is opened.



Device configuration

Slot	Module	...	...	...	...
0	CPU 015-CEFNR00				
-X1	PG_OP_Ethernet				
-X3	MPI interface				
...	...			...	

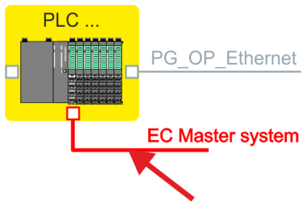
Please consider the additional functions in the SPEED7 Studio can only be activated, if you have valid license for these functions!

Proceeding

1. Click at the CPU in the *'Device configuration'* and select *'Context menu ➔ Components properties'*.
   ⇨ The properties dialog of the CPU is opened
2. Click at *'Feature Sets'* and activate at *'Motion Control'* the parameter *'EtherCAT Master functionality+Motion Control+...'*.
3. Confirm your input with [OK].
   ⇨ The additional functions are now available in your project. More information about the usage may be found in the online help of the *SPEED7 Studio*.

CAUTION!
Please note that with any change in the feature set settings in the *SPEED7 Studio*, due to the system, the EtherCAT fieldbus system together with the motion control configuration are removed from your project!

Configuration EtherCAT master



1. Click in the Project tree at *'Add new device ...'*.
2. Click here at *'EC-Master system'* and select *'Context menu ➔ Bus system properties'*.
   ⇨ The *SPEED7 EtherCAT Manager* opens. Here you can configure the EtherCAT master system.

   More information about the usage of the *SPEED7 EtherCAT Manager* may be found in the online help of the *SPEED7 Studio*.
3. By closing the *SPEED7 EtherCAT Manager* the EtherCAT configuration is taken to the project and the *SPEED7 EtherCAT Manager* is closed. You can always edit your EtherCAT configuration in the *SPEED7 EtherCAT Manager*, since the configuration is stored in your project.

Since slave and module parameters are transmitted by means of SDO respectively SDO Init command, the configuration remains active, until a power cycle is performed or new parameters for the same SDO objects are transferred.

With an overall reset the slave and module parameters are not reset!

9.4 EtherCAT Diagnostics

Overview

There are the following ways to get diagnostics information from your system:

- Diagnostics via *SPEED7 EtherCAT Manager*
- Diagnostics during runtime in the user program (OB 1, SFB 52)
- Diagnostics via system status lists - SSL
- Diagnostics via OB start information
- Diagnostics via diagnostics CPU or CP
- Diagnostics via status LEDs

9.4.1 Diagnostics via *SPEED7 EtherCAT Manager*

Information

The *SPEED7 EtherCAT Manager* offers various opportunities for diagnostics:

- Diagnostics EtherCAT master
- Diagnostics EtherCAT slave station

More information about the usage of the SPEED7 EtherCAT Manager may be found in the online help.

Chap. 9.10.7 'Diagnostics - EC-Mastersystem' page 262

Chap. 9.10.8 'Diagnostics - slave station' page 265

9.4.2 Diagnostics during runtime in the user program (OB 1, SFB 52)

Handling block SFB 52 RDREC

With SFB 52 RDREC (read record) you can access diagnostics data from your user program e.g. in OB1. The SFB 52 RDREC operates asynchronously, that is, processing covers multiple SFB calls.

More information about the usage of the SFB 52 may be found in the online help of your programming tool or in the manual "Operation list".

The following data can be accessed with the SFB 52:

- CoE emergency messages (record set 0x4000 ... 0x4003)
- EtherCAT specific identification data (record set 0x1000)
- EtherCAT interface informations (record set 0x1037)
- EtherCAT register from slave station (record set 0x3000, 0x3001)
- EtherCAT register master (record set 0x3001)
- Analysis DC (record set 0x5000)

9.4.2.1 Accessing the CoE emergency messages

Record set 0x4000 ... 0x4003

With SFB 52 RDREC (read record) you can access CoE emergency messages from your user program e.g. in OB 1 by means of the record sets 0x4000 ... 0x4003. The SFB 52 RDREC operates asynchronously, that is, processing covers multiple SFB calls. An entry for the record sets 0x4000 ... 0x4003, which are described here, consists of the CoE emergency himself (8byte) and the station address of the CoE emergency comes from (2byte).

Record set structure

Index [byte]	Content	Description
0	NumberOfEntries	Number of following CoE emergency entries (0 ...n)
1		
2 + (n*12)	n * *CoE emergency entry*	CoE emergency entry according to the requested record set

CoE emergency entry

Index [byte]	Content	Description
0	Error Code	CoE emergency
1		
2	Error Register	

Index [byte]	Content	Description
3	Error Data	
4		
5		
6		
7		
8	Station Address	Address of the station, which has sent the emergency.
9		
10	Reserved	
11		

Record sets

Record set	Description
0x4000	The record set provides the last CoE emergency of each slave (on CoE emergency entry per slave, which has supplied a CoE emergency). Master data set (logical address for SFB/SFC call: PN-IO system address or address of the EtherCAT master). There are no entries for slaves with no CoE emergency. ■ Parameters: None ■ NumberOfEntries: 0 ... 512
0x4001	The record set provides the last CoE emergency of a specific slave. If a slave ID is passed, which does not exist, an error is returned. Slave data record (logical address for SFB/SFC call: address of the EtherCAT slave from which the information is to be read) → internal parameter: slave ID (1 ... 512). If the slave ID is valid but no CoE emergency for this slave exists, the number of sent entries is equivalent to 0. ■ NumberOfEntries: 0 ... 1

Record set	Description
0x4002	The record set provides the last 20 CoE emergencies of the whole system (this means multiple entries for one slave can be reported). Is there a total of less than 20 entries, the number of messages is correspondingly smaller. Master data set (logical address for SFB/SFC call: PN-IO system address or address of the EtherCAT master). ■ Parameters: None ■ NumberOfEntries: 0 ... 20
0x4003	The record set provides the last 10 CoE emergency of a specific slave. Slave data record (logical address for SFB/SFC call: address of the EtherCAT slave from which the information is to be read) → internal parameter: slave ID (1 ... 512). If a slave ID is passed, which does not exist, an error is returned. If the slave ID is valid but less than 10 CoE emergencies for this slave exist, the number of sent entries is correspondingly smaller. ■ NumberOfEntries: 0 ... 10

Example OB 1

For cyclical access to a record set of the diagnostics data of an EtherCAT slave station, you can use the following example program in OB 1:

```
UN M10.3 'Read process finished (BUSY=0)
UN M10.1 'If there is no job activation
         'then (REQ=0)
S  M10.1 'start record set transfer (REQ:=1)
L  W#16#4001 'record set number(here record set 0x4001)
T  MW12
CALL SFB 52, DB52 'Call SFB 52 with instance DB
  REQ :=M10.1 'Start flag
  ID :=DW#16#0018 'Address of the EtherCAT slave
  INDEX :=MW12
  MLEN :=14 'Length record set 0x4001 with 1. entry
  VALID :=M10.2 'Validity of the record set
  BUSY :=M10.3 'Shows if job just running
  ERROR :=M10.4 'Error bit during read access
  STATUS :=MD14 'Error codes
  LEN :=MW16 'Length of the read record set
  RECORD :=P#M 100.0 Byte 40 'Target (MB100, 40byte)
U M10.1
R M10.1 'Reset of REQ
```

9.4.2.2 Accessing EtherCAT specific identification data

Record set 0x1000

The record set 0x1000 contains EtherCAT specific identification data, which can be read with the SFB 52. The values *Device Type*, *Serial Number*, *Hardware Version* and *Software Version* are directly retrieved via CoE from the slave station. If a slave station does not support CoE or one of these values in the object directory, the values are substituted with 0xFF. The record set has the following structure:

Index	Designation	Data type
1	Address	Unsigned32
2	Device Name	Array of char[32]
3	Vendor ID	Unsigned32
4	Product Code	Unsigned32
5	Device Type	Unsigned32
6	Serial Number	Unsigned32
7	Revision	Unsigned32
8	Hardware Version	Array of char[8]
9	Software Version	Array of char[8]

9.4.2.3 Accessing information of the EtherCAT interface

Record set 0x1037

The record set 0x1037 contains information about the Ethernet interface of the EtherCAT master, which can be read with the SFB 52. The record set has the following structure:

Index	Designation	Data type
1	Logical address	Unsigned16
2	IP address	Unsigned32
3	Subnet mask	Unsigned32
4	Default Router	Unsigned32
5	MAC address	Array of Unsigned8[6]
6	Source	Unsigned8
7	reserved	Unsigned8
8	DCP Mod Timestamp	Array of Unsigned8[8]
9	phys_mode_1	Unsigned8
10	phys_mode_2	Unsigned8
11	phys_mode_3	Unsigned8
12	phys_mode_4	Unsigned8
13	phys_mode_5	Unsigned8
14	phys_mode_6	Unsigned8
15	phys_mode_7	Unsigned8
16	phys_mode_8	Unsigned8
17	phys_mode_9	Unsigned8
18	phys_mode_10	Unsigned8
19	phys_mode_11	Unsigned8
20	phys_mode_12	Unsigned8
21	phys_mode_13	Unsigned8

Index	Designation	Data type
22	phys_mode_14	Unsigned8
23	phys_mode_15	Unsigned8
24	phys_mode_16	Unsigned8
25	reserved	Unsigned8

9.4.2.4 Accessing the EtherCAT register from slave stations

Record set 0x3000

With the record set 0x3000 you can access the registers of an EtherCAT slave station, by calling it with the SFB 52. The record set has the following structure:

Byte	Content	Register
0	AL Status	0x0130, 0x0131
1		
2	AL Control	0x0120, 0x0121
3		
4	Al Status Code	0x0134, 0x0135
5		
6	ESC DL Status	0x0110, 0x0111
7		
8	Processing Unit Error Counter	0x030C
9	PDI Error Counter	0x030D
10	Link Lost Counter Port A	0x0310
11	Link Lost Counter Port B	0x0311
12	Link Lost Counter Port C	0x0312
13	Link Lost Counter Port D	0x0313
14	reserved	-
15	reserved	-

Record set 0x3001

The record set can be used to read the last reported *AL Status Code* of an EtherCAT slave station. The content of the record set remains until an overall reset is made or a new configuration is loaded.

Byte	Content	Register
0	Al Status Code	0x0134, 0x0135
1		

If you use an invalid slave address (slave ID) you will receive an error. If the slave ID is available but the EtherCAT slave station has not just reported an AL Status Code, so you also get an error.

9.4.2.5 Accessing the EtherCAT master register

Record set 0x3001

The record set can be used to read the last reported *AL Status Codes* of all the EtherCAT slave stations. If an EtherCAT slave station did not report a bug by the time of reading, so the returned *Al Status Code* is 0. The content of the record set remains until an overall reset is made or a new configuration is loaded.

Structure record set

Byte	Content
0	Data block for slave ID 1
4	Data block for slave ID 2
...	...
2043	Data block for slave ID 512

Structure data block

Byte	Content	Description
0	Al Status Code	*AL Status Code* of the corresponding EtherCAT slave station
1		
2	Validity	Validity: ■ 0: *AL Status Code* not valid (slave ID is not configured or EtherCAT slave station has not reported an *AL Status Code*, yet). ■ 1: *AL Status Code* is valid
3	reserved	-

Validity is only set to 1 if an Al Status Code is reported from the EtherCAT slave station. With an error-free EtherCAT slave station this byte is 0.

9.4.2.6 Analysis DC

Record set 0x5000

This record set informs about the current status of the DC system.

- These values are only updated with corresponding messages, which also generate a diagnostic buffer entry.
  – The parameters DC_InSync and DC_Deviation are updated with the message "EC_NOTIFY_DC_SLV_SYNC".
  – The parameters DCM_InSync, DCM_CtlErrorCur, DCM_CtlErrorAvg and DCM_CtlErrorMax are updated with the message "EC_NOTIFY_DCM_SYNC".
- Except the counter of "out of sync" the data come from the EtherCAT stack. For this reason, the nomenclature of the EtherCAT stack is taken for this data.

Structure of the data on reading

Index	Name	Type	Description	Default value
1	DC_InSync	DWORD	Indicates whether the DC slaves are synchronized with each other. 0: out of sync 1: in sync	0
2	DC_Deviation	DINT	Deviation in ns	0
3	DC_OutOfSyncCnt	DWORD	Counter, how often DC slaves were "out of sync". The counter is reset when an overall reset is performed or when a new configuration is loaded to the CPU.	0
4	DCM_InSync	DWORD	Indicates whether the DC master and Reference-Clock are synchronized with each other. 0: out of sync 1: in sync	0
5	DCM_CtlErrorCur	DINT	Current DC master deviation in ms	0
6	DCM_CtlErrorAvg	DINT	Average DC master deviation in ns.	0
7	DCM_CtlErrorMax	DINT	Maximum DC master deviation in ns	0
8	DCM_OutOfSyncCnt	DWORD	Counter, how often DC master was "out of sync". The counter is reset when an overall reset is performed or when a new configuration is loaded to the CPU.	0

9.4.3 Diagnostics via system status lists - SSL

SSL partial lists

In the following all the possible SSL partial lists with additional SSL-ID are listed, which are supported by the EtherCAT master system.

More information about the usage of the SSLs can be found in the manual "Operation list".

SSL partial lists	SSL-ID
SSL content	xy00h
Module identification	xy11h
Status of all LEDs	xy19h
Status of the LEDs	xy74h
Status information CPU	xy91h
Stations status information	xy94h
Module status information	xy96h
Diagnostic buffer of the CPU	xyA0h
Information EtherCAT Master/Slave	xyE0h
EtherCAT bus system	xyE1h
Type ID modules central bus system	xyF0h
Status of the VSC features from the System SLIO CPU	xyFCh

9.4.4 Diagnostics via OB start information

On an error the faulty system generates a diagnostics message for the CPU. Then the CPU calls the according diagnostics OB. Here the CPU operating system transfers start information to the local data of the OB. By evaluating the start information of the according OB you can get information about cause and location of the error. During run-time you can access the start information with the system function SFC 6 RD_SINFO. Please consider that you can even read the start information in the OB himself, because the data are temporary data.

Depending on the type of error, the following OBs are called in a diagnostics event:

- OB 82 on an error of an module at the EtherCAT slave station (Diagnostics alarm) *'Interrupt handling in the CPU' page 210*
- OB 86 on failure respectively restart of an EtherCAT slave station *'Enter OB start information and call OB' page 208*
- OB 57 Vendor specific interrupt

More information about OBs may be found in the online help of your programming tool or in the manual "Operation list".

9.4.5 Diagnostics via diagnostics buffer CPU respectively CP

Chap. 4.19 'Diagnostic entries' page 128

9.4.6 Diagnostics via status LEDs

LEDs EtherCAT

BF2 red	BS green	MT yellow	Description
□	□	□	Master is in INIT state
□	◪ 2Hz	□	Master is in Pre-Op state
□	◪ pulses	□	Master is in Safe-Op state
□	■	□	Master is in OP state
X	X	□	There is no maintenance event pending.
X	X	■	There is a maintenance event pending. More may be found in the diagnostics data
□	X	X	There is no error on the EtherCAT bus pending
■	X	X	■ EtherCAT bus error, no connection to sub net ■ wrong transfer rate ■ Full-duplex-transmission is not activated
◪ 1Hz	X	X	■ Failure of a connected IO device ■ At least one IO device cannot be reached (topology mismatch) ■ Faulty configuration
◪ 4s on, 1s off	□	◪ 4s on, 1s off	Error in configuration: ■ 0xEA64 was added to the diagnostics buffer ■ Additionally the SF LED of the CPU is on
◪ 4Hz	□	◪ 4Hz	The alternate blinking indicates that a firmware update of the EtherCAT master is executed.
■	■	■	Firmware update of the EtherCAT master is finished without error.
not relevant: X			

L/A3 green	Description
■	The EtherCAT master is physically connected to the Ethernet.
□	The EtherCAT master is not physically connected to the Ethernet.
◪ flickers	The EtherCAT master shows Ethernet activity.

9.5 Interrupt behaviour

9.5.1 Overview

Once a fault in the EtherCAT system occurs, it is made on OBs and SSLs available. Here, the SSLs are refreshed respectively OBs are called.

9.5.2 Interrupt types

Interrupt types

- MANUFACTURER_SPECIFIC_ALARM_MIN (0x0020 or 0x0021) - OB 57
- PROZESS_ALARM (0x0002) - OB 40 (process interrupt)
- BUS_STATUS_CHANGED (0x8001) - OB 86
- DIAGNOSE_ALARM_GEHEND (0x000C) - OB 82 (diagnostics interrupt going)
- DIAGNOSE_ALARM_KOMMEND (0x0001) - OB 82 (diagnostics interrupt coming)
- SLAVE_STATUS_CHANGED (0x8002) - OB 86
- TOPOLOGY_MISMATCH (0x8004) - OB 86
- TOPOLOGY_OK (0x8003) - OB 86
- DC_STATUS (0x8005)
- BUS_CYCLE_STATUS (0x8006) - OB82
- BUS_STATUS (0x8007) - OB86

9.5.2.1 MANUFACTURER_SPECIFIC_ALARM_MIN (0x0020 or 0x0021) - OB 57

Properties

Triggering event

- EtherCAT CoE emergency

Supplied data

- Slave address
- CoE emergency

Conditions

- The error code of the CoE emergency has to come from a Yaskawa slave station.
  - The error code of the CoE emergency must diver to 0x0000.
  - The error code of the CoE emergency must diver to 0xA000.
  - The error code of the CoE emergency must diver to 0xA001.
  - The error code of the CoE emergency must diver to 0xFF00.
  - If the error code is 0xFF00, then the 2. byte must be equal to 1 or 2.
- The error code of the CoE emergency has to come from another slave station.
  - Each emergency is reported as OB 57.
- A CoE emergency occurred during an topology change.
  - The error code of the CoE emergency must diver to 0x0000.
  - The error code of the CoE emergency must diver to 0xA000 and 0xA001.

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
Eventless	BYTE	0x11
FLT_ID	BYTE	0x5C
PrioLevel	BYTE	0x02
OBNo	BYTE	57
Reserved1	BYTE	0xCC
IoFlag	BYTE	0x54 or 0x55 (depending on the address type of the alarm-triggering module)
Info1	WORD	Diagnostics address of the slave
Info2	WORD	Error code of CoE emergency

Structure element	Data type	Description
Info3	WORD	Slave state of CoE emergency
User1	WORD	InterruptPrio, InterruptRef
User2	WORD	EtherCAT slave address

Update SSL data

Manufacturer specific interrupts do not change SSLs.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	OBNo.	PK:	Dat ID ½	Info1	Info2	Info3
0x115C	57	0x02	0x54CC	Diagnostics address of the slave	Interrupt type	Error code CoE emergency

9.5.2.2 PROZESS_ALARM (0x0002) - hardware interrupt - OB 40

Properties

Triggering event

- EtherCAT CoE emergency from a Yaskawa slave station

Supplied data

- Slave address
- CoE emergency

Conditions

- The error code of the CoE emergency must be equal to 0xFF00 and the CoE emergency has to come from a Yaskawa slave station.
- The 2. byte of *MEF* must be 1.

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0x11
FLT_ID	BYTE	0x41
PrioLevel	BYTE	Priority of the OB 40
OBNo	BYTE	40
Reserved1	BYTE	reserved

Structure element	Data type	Description
IoFlag	BYTE	0x54 or 0x55 (depending on the address type of the alarm-triggering module)
Info1	WORD	Diagnostics address of the slave
Info2	WORD	Error code of CoE emergency
Info3	WORD	Slave state of CoE emergency
User1	WORD	Alarmprio, AlarmRef
User2	WORD	EtherCAT slave address

Update SSL data

Hardware interrupts do not change SSLs.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

There is no diagnostics buffer entry.

9.5.2.3 BUS_STATUS_CHANGED (0x8001) - OB 86

Properties

Triggering event

- EtherCAT bus state was changed.

Supplied data

- Old and new state of the master and the number of slave modules, which are not in master state.

Conditions

- none

Interrupt handling in the CPU

In the event the master switches to "Operational" *Chap. 9.1.2 'EtherCAT State Machine' page 192*, OB86 is released. Via its event class you can see, whether all configured slave stations have carried the state change. Should any or all slave stations are not be able to establish the state to "Operational", so you can check this via a SSL.

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0xEC on restoration or 0xED on failure or other VusStateChanged
FLT_ID	BYTE	0x10 failure or restoration with all slaves, 0x11 restoration with missing salve(s), 0x20 other BusStateChanged

Structure element	Data type	Description
PrioLevel	BYTE	Priority of the OB86
OBNo	BYTE	86
Reserved1	BYTE	1, if slave available, otherwise 0
IoFlag	BYTE	0x54 at input address in ZInfo1, 0x55 at output address
Info1	WORD	0xXXYY: XX=OldState, YY=NewState
Info2	WORD	Diagnostics address of the master
Info3	WORD	Number of missing salves
User1	WORD	0xXXYY: XX=InterruptPrio, YY=InterruptRef
User2	WORD	EtherCAT slave address

Chap. 9.4 ‘EtherCAT Diagnostics’ page 196

Update SSL data

In the SSL xy94 the corresponding bits are updated for each slave. Each to the CPU reported state change as interrupt event generates a diagnostics buffer entry and may be read in the SSL 0xE0.

Update I/O peripheral structure

I/O state of the slaves and its modules are set to EA_STATUS_BG_VORHANDEN (module available) on restoration and EA_STATUS_BG_NICHTVORHANDEN (module do not exist) on failure.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0xEC10, 0xEC11, 0xED10 or 0xED20 (depends on state change)	PrioLevel of OB86	86	see OB-Startinfo Reserved1, IOFlag	old and new state of the slave	Diagnostics address master	Number of slaves, which differ from the status of the master

9.5.2.4 DIAGNOSE_ALARM_GEHEND (0x000C) - diagnostics interrupt going

Properties

Triggering event

- EtherCAT CoE emergency from a Yaskawa slave station

Supplied data

- Slave address
- CoE emergency

Conditions

- The error code of the CoE emergency must be equal to 0x0000 ("no error respectively "error resolved") and the CoE emergency has to come from a Yaskawa slave station.

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0x38
FLT_ID	BYTE	0x42
PrioLevel	BYTE	Priority of the OB 82
OBNo	BYTE	82
Reserved1	BYTE	0xC5
IoFlag	BYTE	0x54
Info1	WORD	Diagnostics address of the slave
Info2	WORD	Error code of CoE emergency
Info3	WORD	Slave state of CoE emergency
User1	WORD	InterruptPrio, InterruptRef
User2	WORD	EtherCAT slave address

Update SSL data

In SSL 0694 and 0692 the corresponding bit is updated for each slave.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0x3842	PrioLevel of OB 82	82	0xC554	Diagnostics address slave	EtherCAT error code	Slave state

9.5.2.5 DIAGNOSE_ALARM_Kommend (0x0001) - diagnostics interrupt coming

Properties

Triggering event

- EtherCAT CoE emergency from a Yaskawa slave station

Supplied data

- Slave address
- CoE emergency

Conditions

- The error code of the CoE emergency must diver to 0x0000
- The error code of the CoE emergency must diver to 0xA000 and 0xA001

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0x39
FLT_ID	BYTE	0x42
PrioLevel	BYTE	Priority of the OB 82
OBNo	BYTE	82
Reserved1	BYTE	0xC5
IoFlag	BYTE	0x54
Info1	WORD	Diagnostics address of the slave
Info2	WORD	Error code of CoE emergency
Info3	WORD	Slave state of CoE emergency
User1	WORD	InterruptPrio, InterruptRef
User2	WORD	EtherCAT slave address

Update SSL data

In SSL 0694 and 0692 the corresponding bit is updated for each slave.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0x3942	PrioLevel of OB 82	82	0xC554	Diagnostics address slave:	EtherCAT error code	Slave state

9.5.2.6 SLAVE_STATUS_CHANGED (0x8002) - OB 86

Properties

Triggering event

- Slave is not in the requested state.
- The application has successfully set a slave in a different state.

Supplied data

- current new state

*Especially when a master status change is performed , this message is **not** sent to the CPU, since the overall result for error slaves of the status change in the event BUS_STATE_CHANGED is transmitted.*

Interrupt handling in the CPU

For each slave the current state is stored inside the CPU.

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0xEC on restoration or 0xED on failure or other VusStateChanged
FLT_ID	BYTE	0x12 failure or restoration, 0x22 other BusState-Changed
PrioLevel	BYTE	Priority of the OB 86
OBNo	BYTE	86
Reserved1	BYTE	1, if slave available, otherwise 0
IoFlag	BYTE	0x54 at input address in ZInfo1, 0x55 at output address
Info1	WORD	0xXXYY: XX=OldState, YY=NewState
Info2	WORD	Diagnostics address of the slave
Info3	WORD	AI Status Code
User1	WORD	0xXXYY: XX=InterruptPrio, YY=InterruptRef
User2	WORD	EtherCAT slave address

Update SSL data

In the SSL xy94 the corresponding bits are updated for each slave. Each to the CPU reported state change as interrupt event generates a diagnostics buffer entry and may be read in the SSL 0xE0.

Update I/O peripheral structure

I/O state of the slaves and its modules are set to EA_STATUS_BG_VORHANDEN (module available) on restoration and EA_STATUS_BG_NICHTVORHANDEN (module do not exist) on failure.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0xEC10, 0xEC11, 0xED10 or 0xED20 (depends on state change)	PrioLevel of OB 86	86	see OB-Star-tinfo Reserved1, IOFlag	old and new state of the slave	Diagnostics address master	Number of slaves, which differ from the status of the master

9.5.2.7 TOPOLOGY_MISMATCH (0x8004) - OB 86

Properties

Triggering event

- Topology differs from the configured topology. The Interrupt is only triggered with an existing configuration.

Supplied data

- none

Conditions

- none

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0xED
FLT_ID	BYTE	0x30
PrioLevel	BYTE	Priority of the OB 86
OBNo	BYTE	86
Reserved1	BYTE	0
IoFlag	BYTE	0
Info1	WORD	0
Info2	WORD	Diagnostics address of the master
Info3	WORD	0
User1	WORD	0
User2	WORD	0

Update SSL data

In the SSL xy94 the difference of set point and effective value is entered.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0xED30	PrioLevel of OB 86	86	0x0000	0	Diagnostics address master	0

9.5.2.8 TOPOLOGY_OK (0x8003) - OB 86

Properties

Triggering event

- Interrupt is triggered, if the topology on the bus corresponds to the configured topoly.

Supplied data

- none

Conditions

- none

Interrupt handling in the CPU

Enter OB start information and call OB

Structure element	Data type	Description
EventClass	BYTE	0xED
FLT_ID	BYTE	0x30
PrioLevel	BYTE	Priority of the OB 86
OBNo	BYTE	86
Reserved1	BYTE	0
IoFlag	BYTE	0
Info1	WORD	0
Info2	WORD	Diagnostics address of the master
Info3	WORD	0
User1	WORD	0
User2	WORD	0

Update SSL data

In the SSL xy94 the difference of set point and effective value is entered.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1, IOFlag	Info1	Info2	Info3
0xED30	PrioLevel of OB 86	86	0x0000	0	Diagnostics address master	0

9.5.2.9 DC_STATUS (0x8005) - OB82

If a change in sync status of a DC slave or the master is recognized by the master, the EtherCAT master sends an appropriate interrupt to the CPU. This then triggers, if available, an OB 82, and writes an entry in the diagnostic buffer.

Interrupt handling in the CPU

Enter OB start information and call OB

Structural element	Data type	Description
EventClass	BYTE	0xEC: Incoming event 0xED Outgoing event
FLT_ID	BYTE	0x50: At least one DC is not synchronized
PrioLevel	BYTE	Priority of the OB82
OBNo	BYTE	82
Reserved1	BYTE	reserved
DatId	WORD	0x0000
Info1	WORD	0x0000
Info2	WORD	Diagnostic address of the master
Info3	WORD	0: DC status change master 1: DC status change slave station

Update SSL data

In the SSL xy94 the corresponding bits are updated for each slave. Each to the CPU reported state change as interrupt event generates a diagnostics buffer entry and may be read in the SSL 0xE0.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	DatId	Info1	Info2	Info3
0xEC50 0xED50	PrioLevel of OB82	82	0x0000	0x0000	Diagnostics address master	0: DC status change master 1: DC status change slave station

9.5.2.10 BUS_CYCLE_STATUS (0x8006) - OB 82

Properties

In the case that the bus cycle time can not be complied with, when reaching a maximum value (penalty) BUS_CYLCE_STATUS_CYCLE_DURATION_TOO_LONG interrupt is triggered. If the bus cycle is OK again, a BUS_CYLCE_STATUS_CYCLE_DURATION_OK interrupt is triggered. The monitoring and recording of bus cycle time violations follows these rules:

- If a bus cycle time violation occurs, *Penalty* is incremented by 3.
- If there is no violation, *Penalty* is decremented by 1.
- If *Penalty* reaches the value 9, the interrupt BUS_CYCLE_STATUS with cause BUS_CYLCE_STATUS_CYCLE_DURATION_TOO_LONG is sent to the CPU. After that no further interrupts of this type are reported on other bus cycle time violations.
- If *Penalty* reaches the value 0, the interrupt BUS_CYCLE_STATUS with cause BUS_CYLCE_STATUS_CYCLE_DURATION_OK is sent to the CPU.

Interrupt handling in the CPU

Enter OB start information and call OB

Structural element	Data type	Description
EventClass	BYTE	0xEC: Incoming event 0xED: Outgoing event
FLT_ID	BYTE	0x40: A Bus cycle time violation has occurred
PrioLevel	BYTE	Priority of the OB82
OBNo	BYTE	82
Reserved1	BYTE	reserved
Info1	WORD	Diagnostics address of the EtherCAT IO system
Info2	WORD	0x0000
Info3	WORD	0x0000

Update SSL data

In the SSL xy94 the corresponding bits are updated for each slave. Each to the CPU reported state change as interrupt event generates a diagnostics buffer entry and may be read in the SSL 0xE0.

Caching the interrupt

Snapshot at the time of interrupt events - can be evaluated via SFB 54.

Write to the diagnostics buffer

EventId:= Eventclass, StartEvent	PrioLevel	OBNo.	Reserved1	Info1	Info2	Info3
0xEC40 0xED40	PrioLevel of OB82	82	reserved	Diagnostics address of the EtherCAT IO system	0x0000	0x0000

9.5.2.11 BUS_STATUS (0x8007) - OB 86

In the case that multiple telegrams can not be received in the master, when reaching a maximum value (penalty) BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE interrupt is triggered. If telegrams can be received again, BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE_OK interrupt is triggered. The monitoring and recording of receive time-out in the master follows these rules:

- If a receive time-out occurs, *Penalty* is incremented by 3.
- If there is no receive time-out, *Penalty* is decremented by 1.
- If *Penalty* reaches the value 9, the interrupt BUS_STATUS with cause BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE_TIMEOUT is sent to the CPU. After that no further interrupts of this type may be reported on other receive time-outs.
- If *Penalty* reaches the value 0, the interrupt BUS_STATUS with cause BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE_OK is sent to the CPU. This interrupt is only sent once an interrupt BUS_STATUS with the cause BUS_STATUS_REASON_CYCLIC_FRAME_RECEIVE_TIMEOUT has been reported to the CPU, before.

OB86_EV_CLASS	OB86_FLT_ID	Description
B#16#EC	B#16#80	Bus error resolved
OB86_Z1:	Diagnostics address of the EtherCAT IO system	
OB86_Z2:	0	
OB86_Z3:	Bit 11 to 14: System ID EtherCAT network - 100	
Bit 15: 1 - flag bit for EtherCAT		

OB86_EV_CLASS	OB86_FLT_ID	Description
B#16#ED	B#16#80	Bus error occurred (receive timeout)
OB86_Z1:	Diagnostics address of the EtherCAT IO system	
OB86_Z2:	0	
OB86_Z3:	Bit 11 to 14: System ID EtherCAT network - 100	
Bit 15: 1 - flag bit for EtherCAT		

9.6 Firmware update

EtherCAT master

Chap. 4.13 ‘Firmware update’ page 116

EtherCAT slave station

Firmware update via *SPEED7 EtherCAT Manager*. More may be found in the according manual respectively online help. *Chap. 9.10.11 ‘Firmware update - System SLIO IM 053-1EC0x’ page 276*

9.7 EtherCAT system limits

Maximum number of EtherCAT slaves and productive connections

The EtherCAT master has the following system limits for the number of EtherCAT slaves and productive connections. By means of the following formula the EtherCAT slaves per ms can be calculated from the EtherCAT *cycle time*. From this, the maximum number of EtherCAT slaves and productive connections can be determined by the table.

$$S = \frac{n}{C}$$

S EtherCAT slaves per ms
n Number of EtherCAT slaves
C EtherCAT *cycle time* in ms

Please note that the value S must always be rounded to the nearest smaller integer!

The EtherCAT master has the following system limits		
EtherCAT slave per ms	**Max. number of EtherCAT slaves**	**Max. number of configurable connections**
64	512	0
56	512	2
48	512	4
40	512	6
32	512	8

Maximum total size of the input and output bytes per ms

The EtherCAT master has the following system limits for the input and output data as a function of the EtherCAT *cycle time*. The sum of the input and output data per ms can be calculated using the following formula. The table can be used to determine whether the calculated value is still within the respective limits.

$$S_{IO} = \frac{I + O}{C}$$

S_{IO} Sum of the input and output bytes per ms
I Number of input bytes
O Number of output bytes
C EtherCAT *cycle time* in ms

The EtherCAT master has the following system limits	
EtherCAT *cycle time*	**Max. sum of the input and output bytes**
1ms	640byte
2ms	1024byte
4ms	2048byte

9.8 Accessing the object dictionary

9.8.1 Overview

Blocks

With the following blocks you have at run-time access to the object dictionary of the EtherCAT slave stations and EtherCAT master:

- FB 52 - Read SDO - Read access to object dictionary
- FB 53 - Write SDO - Write access to object dictionary

These are product specific blocks. More information about the usage of these blocks may be found in the manual "Operation list".

Please consider when accessing the object dictionary, depending on your master system, the byte order can be rotated!

9.9 Object dictionary

9.9.1 Object overview

Index	Object Dictionary Area
0x0000 ... 0x0FFF	Data Type Area Objects
0x1000 ... 0x1FFF	CoE Communication Area Objects
0x2000 ... 0x20FF	Generic Master Area Objects
0x2100 ... 0x21FF	Distributed Clocks Objects
0x3000 ... 0x3FFF	Slave Configuration / Information Objects
0x4000 ... 0x7FFF	Reserved Area
0x8000 ... 0x8FFF	CoE Slave Configuration Objects
0x9000 ... 0x9FFF	CoE Slave Information Objects
0xA000 ... 0xAFFF	CoE Slave Diagnosis Data Objects
0xB000 ... 0xEFFF	Reserved Area
0xF000 ... 0xFFFF	CoE Device Area Objects

9.9.2 CoE Communication Area Objects: 0x1000-0x1FFF

Index	Object Type	Name	Type
0x1000	VAR	Device Type	Unsigned32
0x1001	VAR	Error Register	Unsigned8
0x1008	VAR	Manufacturer Device Name String	VisibleString
0x1009	VAR	Manufacturer Hardware Version String	VisibleString
0x100A	VAR	Manufacturer Software Version String	VisibleString
0x1018	RECORD	Identity Object	Identity (0x23)
0x10F3	RECORD	History Object	History (0x26)

9.9.2.1 Device Type 0x1000

Subindex	Name	Type	Access	Value	Meaning
0x00	Device Type	Unsigned32	ro	0x00001389	0x00001389 means MDP

9.9.2.2 Device Name 0x1008

Subindex	Name	Type	Access	Value	Meaning
0x00	Device name	Visible string	ro	EtherCAT-Master	Name of the EtherCAT device

9.9.2.3 Hardware Version 0x1009

Subindex	Name	Type	Access	Value	Meaning
0x00	Hardware version	Visible string	ro	"V MM.mm.ss.bb"" MM = Major Version mm = Minor Version ss = Service Pack bb = Build e.g. "V 01.05.02.02"	Hardware version of the EtherCAT device

9.9.2.4 Software Version 0x100A

Subindex	Name	Type	Access	Value	Meaning
0x00	Software version	Visible string	ro	"V MM.mm.ss.bb"" MM = Major Version mm = Minor Version ss = Service Pack bb = Build e.g. "V 01.05.02.02"	Software version of the EtherCAT device

9.9.2.5 Identity Object 0x1018

Subindex	Name	Type	Access	Value	Meaning
0x00	Number of Entries	Unsigned8	ro	0x04 (default)	
0x01	Vendor ID	Unsigned32	ro	0x0000022B (default)	Vendor ID of the EtherCAT device
0x02	Product Code	Unsigned32	ro	0x00001636 (default)	Product Code of the EtherCAT device
0x03	Revision Number	Unsigned32	ro	0x00000000 (default)	Revision Number (EtherCAT master software version)
0x04	Serial Number	Unsigned32	ro	0x00000000 (default)	Serial Number of the EtherCAT device

9.9.2.6 History Object 0x10F3

Subindex	Name	Type	Access	Value	Meaning
0	Number of Entries	Unsigned8	ro		
1	Maximum number of Diag messages	Unsigned8	ro		
2	Subindex of newest Diag message	Unsigned8	ro		
3	Subindex of newest acknowledged Diag message	Unsigned8	r/w		
4	New Diag messages available	BOOL32	ro		
5	Flags (UINT16, r/w)	Unsigned16	r/w	0	Bit 0 = 1: Enable Emergency sending (default = 0) Bit 1 = 1: Disable Storing Info Messages (default = 0) Bit 2 = 1: Disable Storing Warning Messages (default = 0) Bit 3 = 1: Disable Storing Error Messages (default = 0) Bit 4...15: reserved for future use
6 ... 255			ro		

9.9.2.6.1 Diagnosis Messages Object 0x10F3: 6-255

Byte-Offset	Name	Type	Access	Value	Meaning
0	Diag-Number	Unsigned32	ro		Bit 0...11: free use Bit 12...15 = 14: to be comp. with Emergency Error Bit 16...31 = 0: reserved Bit 16...31 = 0xFFFE: free use Bit 16...31 = 0xFFFF: reserved
4	Flags	Unsigned16	ro		Bit 0...3: Diag type (0 = Info, 1 = warning, 2 = error) Bit 4...15: reserved
6	Text ID	Unsigned16	ro		0 = no Text ID 1-65535 = Reference to a Text ID with formatted string
8	Time Stamp in ns (from DC)	Unsigned64	ro		
16	Flags parameter 1	Unsigned16	ro		
18	Parameter 1	several	ro		
N	Flags parameter n	Unsigned16	ro		
N+2	Parameter n	several	ro		

9.9.3 Generic Master Objects: 0x2000-0x20FF

Index	Object Type	Name	Type
0x2000	VAR	Master State Change Command Register	Unsigned32
0x2001	VAR	Master State Summary	Unsigned32
0x2002	RECORD	Bus Diagnosis Object	BusDiagnostic (0x40)
0x2005	RECORD	MAC Address	MACAddress (0x41)
0x2010	VAR	Debug Register	Unsigned48
0x2020	RECORD	Master Init. Parameters	MasterInitParm (0x42)

9.9.3.1 Master State Change Command Register 0x2000

Subindex	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	r/w	0 = invalid 1 = init 2 = pre-operational 3 = bootstrap mode 4 = safe operational 8 = operational	

9.9.3.2 Master State Summary 0x2001

Subindex	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro		Bit 0: = 1 Master OK Bit 1...3: reserved Bit 4...7: Master State Bit 8: Slaves in requested State Bit 9: Master in requested State Bit 10: Bus Scan Match Bit 11: reserved Bit 12: DC is enabled Bit 13: DC In-Sync Bit 14: DC Busy Bit 15: Reserved Bit 16: Link Up Bit 17...31: reserved

Master is OK if topology is Ok (Mismatch if slave exists, which is not configured). Master must be in *Op* state, slaves must be in *Op* state and *Distributed Clocks* must be *insync* if activated.

Parameter Flags Bit 12...15	Parameter Flags Bit 0...11	Type of Data	Data
0	CoE DataType e.g. 0x0007 = UINT32	Data Type	Data defined through CoE DataType
1	Length in Byte	Byte Array	Byte stream byData[Size]
2	Length in Byte	ASCII-String	String szString[Length] (not '\0' terminated)
3	Length in Byte	Unicode String	String wszString[Length/2] (not L'\0' terminated)
4	0	Text Id	Text Id (Word)

9.9.3.3 Bus Diagnosis Object 0x2002

Object Type: RECORD, Manufacturer Specific Identity 0x40

Subindex	Description	Type	Access
0x00	Number of Entries	Unsigned8	ro
0x01	Reserved	Unsigned16	ro
0x02	Configuration Checksum CRC32	Unsigned32	ro
0x03	Number of found Slave	Unsigned32	ro
0x04	Number of found DC Slave	Unsigned32	ro
0x05	Number of Slaves in Configuration	Unsigned32	ro
0x06	Number of Mailbox Slaves in Configuration	Unsigned32	ro
0x07	Counter: TX frames	Unsigned32	ro
0x08	Counter: RX frames	Unsigned32	ro
0x09	Counter: Lost frames	Unsigned32	ro
0x10	Counter: Cyclic frames	Unsigned32	ro
0x11	Counter: Cyclic datagrams	Unsigned32	ro
0x12	Counter: Acyclic frames	Unsigned32	ro
0x13	Counter: Acyclic datagrams	Unsigned32	ro
0x14	Clear Counters by writing 1 to bit(s) Bit 0: Clear all Counters Bit 1: Clear Tx Frame Counter (Idx 7) Bit 2: Clear Rx Frame Counter (Idx 8) Bit 3: Clear Lost Frame Counter (Idx 9) Bit 4: Clear Cyclic Frame Counter (Idx 10) Bit 5: Clear Cyclic Datagram Counter (Idx 11) Bit 6: Clear Acyclic Frame Counter (Idx 12) Bit 7: Clear Acyclic DataGram Counter (Idx 13) Bit 8...31: Reserved	Unsigned32	r/w

9.9.3.4 MAC Address 0x2005

Object Type: RECORD, Manufacturer Specific Identity 0x41

Subindex	Description	Type	Access
0x00	Number of Entries	Unsigned8	ro
0x01	Hardware	Unsigned48	ro
0x02	Red Hardware	Unsigned48	ro
0x03	Configuration Source	Unsigned48	ro
0x04	Configuration Destination	Unsigned48	

9.9.3.5 Debug Register 0x2010

Subindex	Name	Type	Access	Value	Meaning
0x00	Debug Register	Unsigned38	r/w	Upper 16Bit: 0: activate LinkError Messages 1...15: reserved Lower 32Bit: Definition of parameter dwStateChangeDebug in structure EC_T_MASTER_CONFIG	

9.9.3.6 Master Init Parameters 0x2020

Object Type: RECORD, Manufacturer Specific Identity 0x42

Subindex	Description	Type	Access
00	Number of Entries	Unsigned8	ro
01	EC_T_INITMASTERPARMS.dwVersion Application	Unsigned32	ro
02	dwVersion Master	Unsigned32	ro
03	EC_T_MASTER_CONFIG.nSlaveMultiplier	Unsigned32	ro
04	EC_T_MASTER_CONFIG.dwEcatCmdTimeout in millisec	Unsigned32	ro
05	EC_T_MASTER_CONFIG.dwEcatCmdMaxRetries	Unsigned32	ro
06	EC_T_MASTER_CONFIG.dwCycTimeout in millisec	Unsigned32	ro
07	EC_T_MASTER_CONFIG.dwEoeTimeout in millisec	Unsigned32	ro
08	EC_T_MASTER_CONFIG.dwFoeBusyTimeout in millisec	Unsigned32	ro
09	EC_T_MASTER_CONFIG.dwMaxQueuedEthFrames	Unsigned32	ro
10	EC_T_MASTER_CONFIG.dwMaxSlaveCmdPerFrame	Unsigned32	ro
11	EC_T_MASTER_CONFIG.dwMaxQueuedCoeSlaves	Unsigned32	ro
12	EC_T_MASTER_CONFIG.dwMaxQueuedCoeCmds	Unsigned32	ro
13	EC_T_MASTER_CONFIG.dwStateChangeDebug	Unsigned32	ro
14	EC_T_LINK_DEV_PARAM.szDriverIdent	VisibleString	ro
15	EC_T_LINK_DEV_PARAM.bPollingModeActive	Bool32	ro
16	EC_T_LINK_DEV_PARAM.bAllocSendFrameActive	Bool32	ro

9.9.4 Distributed Clocks Objects: 0x2100-0x21FF

Index	Object Type	Name	Type
0x2100	VAR	DC Slave Sync Deviation Limit	Unsigned32
0x2101	VAR	DC Current Deviation	Signed32
0x2102	VAR	DC Reserved	Unsigned32
0x2103	VAR	DC Reserved	Unsigned32

9.9.4.1 Distributed Clocks Slave Sync Deviation Limit 0x2100

Subindex	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro	dwDevLimit	

9.9.4.2 Distributed Clocks Current Deviation 0x2101

Subindex	Name	Type	Access	Value	Meaning
0x00	Master State	Unsigned32	ro	dwDeviation	

9.9.4.3 Reserved: 0x2102 / 0x2103

This value is reserved.

9.9.5 Slave specific objects

Slave Configuration / Information Objects: 0x3000-0x3FFF

Index	Object Type	Name	Type
0x3000 ... 0x3FFF	RECORD	Slave Configuration and Information Objects	SlaveCfgInfo (0x43)

CoE Slave Configuration Objects: 0x8000-0x8FFF

Index	Object Type	Name	Type
0x8000 ... 0x8FFF	RECORD	One index entry for each configured slave (from ESI)	SlaveCfg (0x45)

CoE Slave Information Objects: 0x9000-0x9FFF

Index	Object Type	Name	Type
0x9000 ... 0x9FFF	RECORD	One index entry for each connected BUS-slave (updated during BUS scan)	SlaveInfo (0x46)

CoE Slave Diagnosis Data Objects: 0xA000-0xAFFF

Index	Object Type	Name	Type
0xA000 ... 0xAFFF	RECORD	One subindex entry for each connected BUS-slave (cyclic updated)	SlaveDiag (0x47)

9.9.5.1 Slave Configuration and Information Object 0x3000-0x3FFF

Object Type: RECORD, Manufacturer Specific Identity 0x43

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	Entry Valid	Bool32	ro
2	VendorId (Bus)	Unsigned32	ro
3	ProductCode (Bus)	Unsigned32	ro
4	Revision No (Bus)	Unsigned32	ro
5	Serial No (Bus)	Unsigned32	ro
6	Device Name (Config)	Visible_String[80]	ro
7	Auto Increment Address (Bus)	Unsigned16	ro
8	Physical Address (Bus)	Unsigned16	ro
9	Config Physical Address (Config)	Unsigned16	ro
10	Alias Address (Bus)	Unsigned16	ro
11	PortState (Bus)	Unsigned16	ro
12	DC Support (Bus)	Bool32	ro
13	DC Support 64Bit (Bus)	Bool32	ro
14	Mailbox Support (Config)	Bool32	ro
15	Requested State (slave instance)	Unsigned32	r/w
16	Current State (slave instance)	Unsigned32	ro
17	Error Flag Set (slave instance)	Bool32	ro
18	Enable Linkmessages (slave instance)	Bool32	r/w
19	Error code (slave instance)	Unsigned32	ro
20	Sync Pulse active (Config, slave instance)	Bool32	ro
21	DC Sync 0 Period (Config, slave instance)	Unsigned32	ro
22	DC Sync 1 Period (Config, slave instance)	Unsigned32	ro
23	SB Error Code (Bus Topology)	Unsigned32	ro
24	RX Error Counter Port 0 (Bus)	Unsigned16	ro
25	RX Error Counter Port 1 (Bus)	Unsigned16	ro
26	RX Error Counter Port 2 (Bus)	Unsigned16	ro
27	RX Error Counter Port 3 (Bus)	Unsigned16	ro
28	Forwarded RX Error Counter Port 0 (Bus)	Unsigned8	ro

Subindex	Description	Type	Access
29	Forwarded RX Error Counter Port 1 (Bus)	Unsigned8	ro
30	Forwarded RX Error Counter Port 2 (Bus)	Unsigned8	ro
31	Forwarded RX Error Counter Port 3 (Bus)	Unsigned8	ro
32	EtherCAT Processing Unit Error Counter (Bus)	Unsigned8	ro
33	PDI Error Counter (Bus)	Unsigned8	ro
34	Reserved	Unsigned16	ro
35	Lost Link Counter Port 0 (Bus)	Unsigned8	ro
36	Lost Link Counter Port 1 (Bus)	Unsigned8	ro
37	Lost Link Counter Port 2 (Bus)	Unsigned8	ro
38	Lost Link Counter Port 3 (Bus)	Unsigned8	ro
39	FMMU's supported (Bus)	Unsigned8	ro
40	Sync Managers supported (Bus)	Unsigned8	ro
41	RAM Size in kByte (Bus)	Unsigned8	ro
42	Port Descriptor (Bus)	Unsigned8	ro
43	ECS Type (Config)	Unsigned8	ro
44	Slave is optional (Config)	Bool32	ro
45	Slave is present (Bus)	Bool32	ro
46	Hot connect group ID	Unsigned32	ro

9.9.5.2 CoE Slave Configuration Objects: 0x8000-0x8FFF

Object Type: RECORD, Manufacturer Specific Identity 0x45

The configuration data contain information about the EtherCAT slaves.

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	Fixed Station Address	Unsigned16	ro
2	Type	Visible_String[64]	ro
3	Name	Visible_String[64]	ro
4	Device Type	Unsigned32	ro
5	Vendor ID	Unsigned32	ro
6	Product Code	Unsigned32	ro
7	Revision Number	Unsigned32	ro
8	Version Number	Unsigned32	ro
33	Mailbox Out Size (if mailbox slave)	Unsigned16	ro
34	Mailbox In Size (if mailbox slave)	Unsigned16	ro

9.9.5.3 CoE Slave Information Objects: 0x9000-0x9FFF

Object Type: RECORD, Manufacturer Specific Identity 0x46

Information about the connected EtherCAT-Slaves can be found in the information data. They are available when the scan command has been executed.

Subindex	Description	Type	Access
0	Number of Entries	Unsigned8	ro
1	**Fixed Station Address** of the Nth EtherCAT slave found (same value as 0xF040: 01)	Unsigned16	ro
5	**Vendor ID** of the Nth EtherCAT slave found (entry 0x1018: 01 of the EtherCAT slave)	Unsigned32	ro
6	**Product Code** of the Nth EtherCAT slave found (entry 0x1018: 02 of the EtherCAT slave)	Unsigned32	ro
7	**Revision Number** of the first EtherCAT slave found (entry 0x1018: 03 of the EtherCAT slave)	Unsigned32	ro
8	**Version Number** of the first EtherCAT slave found (entry 0x1018: 04 of the EtherCAT slave)	Unsigned32	ro
32	**DL Status** (Register 0x110-0x111) of the Nth EtherCAT slave found.	Unsigned16	ro

9.9.5.4 CoE Slave Diagnosis Data Objects: 0xA000-0xAFFF

Object Type: RECORD, Manufacturer Specific Identity 0x47

The diagnostics data contain status and diagnostics information of the EtherCAT slaves or the connections of the EtherCAT slaves.

Subindex	Description	Type	Access
0	**Number of Entries**	Unsigned8	ro
1	**AL Status** (Register 0x130-0x131) of the Nth EtherCAT slave configured.	Unsigned16	ro
2	**AL Control** (Register 0x120-0x121) of the Nth EtherCAT slave configured.	Unsigned16	r/w

9.9.6 CoE Device Area Objects: 0xF000-0xFFFF

Index	Object Type	Name	Type
0xF000	RECORD	Modular Device Profile	DeviceProfile (0x48)
0xF002	RECORD	Detect Modules Command	DetectCmd (0x49)
0xF020 ... 0xF02F	RECORD	Configured Address List	ConfAddrList (0x50)
0xF040 ... 0xF04F	RECORD	Detected Address List	ConnAddrList (0x51)

9.9.6.1 Modular Device Profile Object 0xF000

Object Type: RECORD, Manufacturer Specific Identity 0x48

Subindex	Description	Type	Access
0	**Number of Entries**	Unsigned8	ro
1	**Index distance** between two modules. This value is always read as 1.	Unsigned16	ro
2	**Maximum number of EtherCAT slaves** connected to the EtherCAT bus. This value is read as 512.	Unsigned16	ro
3	**Available entries in objects 0x8xxx** (number of configured slaves).	Unsigned32	ro
4	**Available entries in objects 0x9xxx** (number of connected slaves).	Unsigned32	ro

9.9.6.2 Configured Address List Object 0xF020-0xF02F

Object Type: RECORD, Manufacturer Specific Identity 0x50

Subindex	Description	Type	Access
0	**Number of Entries**	Unsigned8	ro
1	Fixed Station Address of the first EtherCAT slave configured.	Unsigned16	ro
2	Fixed Station Address of the second EtherCAT slave configured.	Unsigned16	ro
…	…		ro
255	Fixed Station Address of the 255. EtherCAT slave configured.	Unsigned16	ro
0	**Number of Entries**	Unsigned8	ro
1	Fixed Station Address of the 256. EtherCAT slave configured.	Unsigned16	ro
…	...		

9.9.6.3 Detected Address List Object 0xF040-0xF04F

Object Type: RECORD, Manufacturer Specific Identity 0x51

Subindex	Description	Type	Access
0	**Number of Entries**	Unsigned8	ro
1	Fixed Station Address of the first EtherCAT slave detected.	Unsigned16	ro
2	Fixed Station Address of the second EtherCAT slave detected.	Unsigned16	ro
...	...		ro
255	Fixed Station Address of the 255. EtherCAT slave detected.	Unsigned16	ro
0	**Number of Entries**	Unsigned8	ro
1	Fixed Station Address of the 256. EtherCAT slave detected.	Unsigned16	ro
...	...		

9.10 Deployment *SPEED7 EtherCAT Manager*

9.10.1 Overview

Properties

- Serves to configure EtherCAT master.
- Is called within the *SPEED7 Studio*.
- Synchronizes the address areas with the *SPEED7 Studio*.
- Saves the configuration in the *SPEED7 Studio* project.
- Expanded functionality by choose-able *'Expert'* mode.

Functions

- Automatic configuration
- Manual configuration
- Diagnosis

Starting the *SPEED7 EtherCAT Manager*

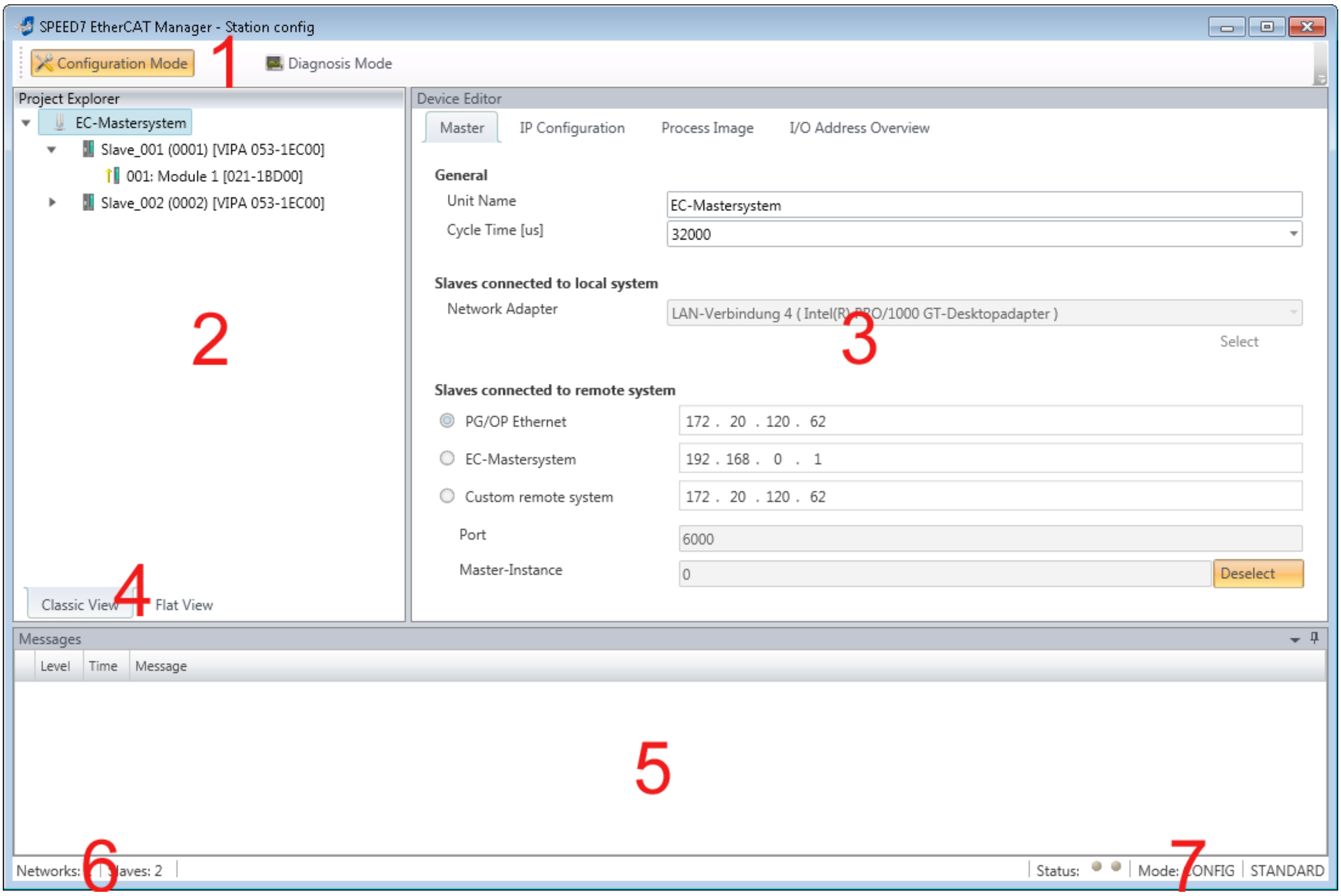
In *SPEED7 Studio* you can call via the *'Project tree'*, *'Field periphery'* of the EtherCAT CPU the *SPEED7 EtherCAT Manager* with *'Bus system properties'*.

Exit the *SPEED7 EtherCAT Manager*

By clicking at [X] in the *SPEED7 EtherCAT Manager*, the dialog is closed and the configuration is taken to the *SPEED7 Studio*.

Work environment of the *SPEED7 EtherCAT Manager*

The work environment of the *SPEED7 EtherCAT Manager* is divided into the following parts:



1 Tool bar: Here you can switch between *Configuration* and *Diagnosis*.
2 Project explorer: Here master and slave stations of your system are listed.
3 Device editor: Properties dialog of a device (parameter) respectively information area.
4 Selection of the view: In *Classic View* all the subordinate stations are shown indented. In *Flat View* all the subordinate stations are shown at the same level.
5 Here all the messages are listed.
6 In this section you can find the number of networks and slave stations.
7 Status area: With an online connection the 2 *Status* LEDs flash alternately. At *Modus* it is shown whether you are in operating mode *Diagnosis* or *Configuration*, followed by the selected dialog sight *Standard* respectively *Expert*.

'Expert mode'

In *SPEED7 Studio* you can call via the *'Project tree'*, *'Field periphery'* of the EtherCAT CPU the *SPEED7 EtherCAT Manager* with *'Bus system properties (Expert)'*. When enabled, the properties dialogs are extended accordingly. In *'Expert mode'* you will have the full scope of the *SPEED7 EtherCAT Manager*. Additionally in the status area *'Expert'* is shown.

Input area - numeric format

Some input fields have [Dec] respectively [Hex] buttons. By selecting the corresponding button you can select the input format *decimal* respectively *hexadecimal* for the input field.

9.10.2 Automatic configuration of a slave system

Precondition

The automatic configuration assumes that your EtherCAT system is mounted and can be reached on-line.

There are the following possibilities for on-line connection:

- Slaves connected to the local system
  – You are directly connected to a slave station via EtherCAT by means of a separate network adapter. Here the on-line connection is established by specifying the *Network Adapter*.
- Slaves connected to remote system
  – You are connected to the Ethernet PG/OP channel of your CPU and can use this to access the EtherCAT master. The on-line connection is established by specifying *IP Address*, *Port* and *Master Instance*. With Yaskawa CPUs *Port* 6000 and *Master Instance* 0 is to be set.

Proceeding

1. Open if not already done the *SPEED7 EtherCAT Manager*
2. Click in the *'Project Explorer'* at *'EC-Mastersystem'*
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly locally connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network adapter* and click at [Select].
   - If you are connected to the Ethernet PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With Yaskawa CPUs set *Port* to 6000 and *Master Instance* to 0.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.

   > *When called from the SPEED7 Studio the IP address is taken from your project. If you change the IP address you need to adjust this in your project and start the SPEED7 EtherCAT Manager again!*

4. Click in the *'Project Explorer'* at *'EC-Mastersystem'* and select from the context menu *'Scan EtherCAT network'*

   ⇨ You might be asked if you want to delete the existing slaves. Confirm with [Yes].

   Then the master is listed with its slaves and the associated PDO configuration in the *'Project Explorer'*, which was found by the network scan. The system can now be configured accordingly.

> *If there is no connection possible with the local master, the an anti virus software could block the connection. Then disabling the packet filter of the protocols of the network card in the anti virus software could help.*

9.10.3 Manual configuration of a slave system

Precondition

With the manual configuration the system need not be built and connected online. The system can freely be configured in the *SPEED7 EtherCAT Manager*.

Proceeding

1. Open if not already done the *SPEED7 EtherCAT Manager*.
2. Click at the *'Project Explorer'* at *'EC-Mastersystem'* and select *'Context menu ➔ Append Slave(s)'*.
   ⇨ A dialog opens to insert slave systems
3. Select the according slave from the list, enter the number of slaves and confirm with [OK].
   ⇨ The corresponding slave systems are inserted and can be configured now.

9.10.4 Configuration - EC-Mastersystem

9.10.4.1 Preparation

Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*. As soon you have configured at least one slave station, the following registers are available:

Chap. 9.10.4.2 'Master' page 238

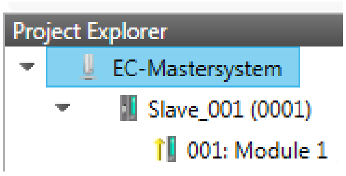
Chap. 9.10.4.3 'Process Data Image' page 239

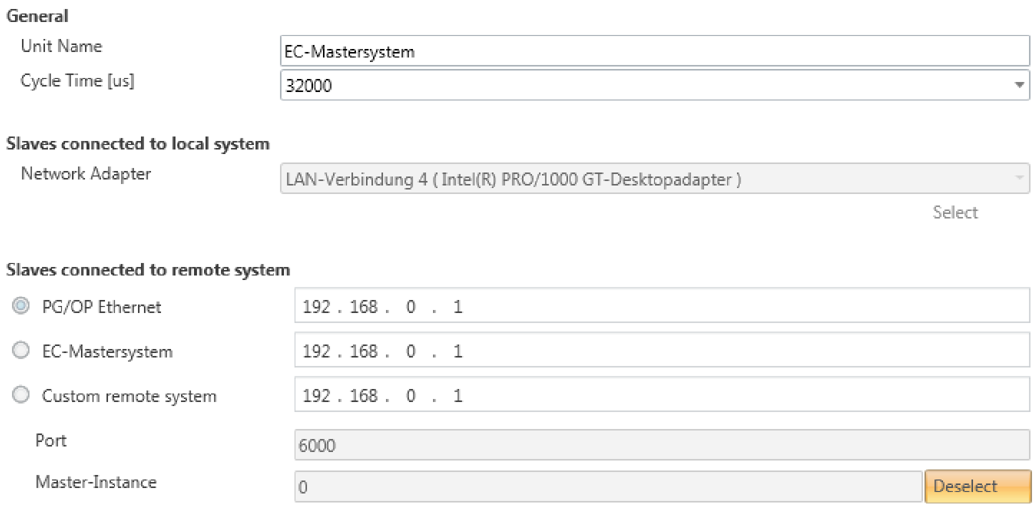
Chap. 9.10.4.4 'Advanced Options (Expert mode)' page 239

Chap. 9.10.4.5 'Distributed Clocks (Expert mode)' page 242

Chap. 9.10.4.6 'I/O Address Overview' page 243

9.10.4.2 Master





Here you can perform master and bus-specific settings.

- General
  - Unit Name: Name of the master
  - Cyclic time: Interval in µs, in which the process data are read and written (PDO cycle time). Here you can choose between different values.
- Slaves connected to the local system
  - You are directly connected to a slave station via EtherCAT by means of a separate network adapter. Here the on-line connection is established by specifying the *Network Adapter*.
- Slaves connected to remote system
  - You are connected to the PG/OP channel of your CPU and can use this to access the EtherCAT master. The on-line connection is established by specifying *IP Address*, *Port* and *Master Instance*.

    IP Address: Enter the IP Address of the PG/OP channel of the remote CPU.

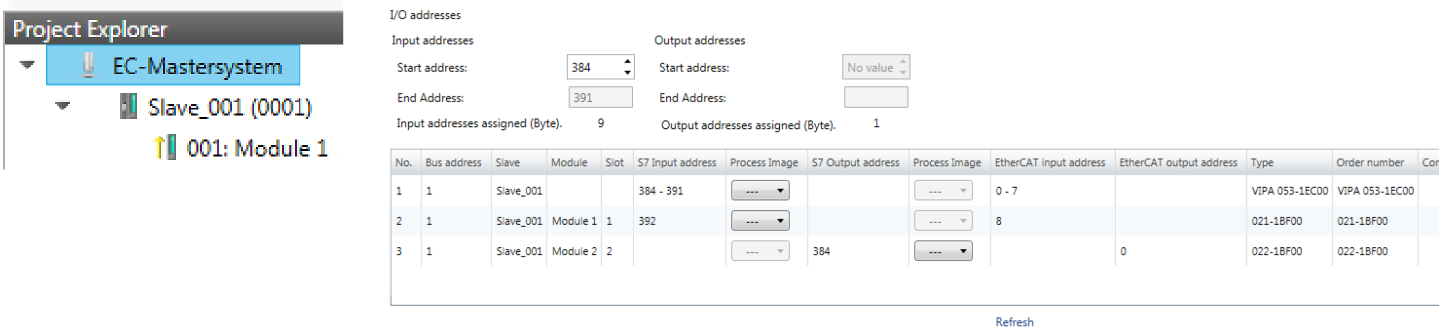
    Port: Port, over which the communication takes place with the remote CPU. With Yaskawa CPUs use Port 6000.

    Master-Instance: Serves for the master instance of the remote system. With Yaskawa systems the master instance is 0.

With [Select] the *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.

When called from the SPEED7 Studio the IP address is taken once from your project. If you change the IP address you need to adjust this in your project and then start the SPEED7 EtherCAT Manager again.

9.10.4.3 Process Data Image



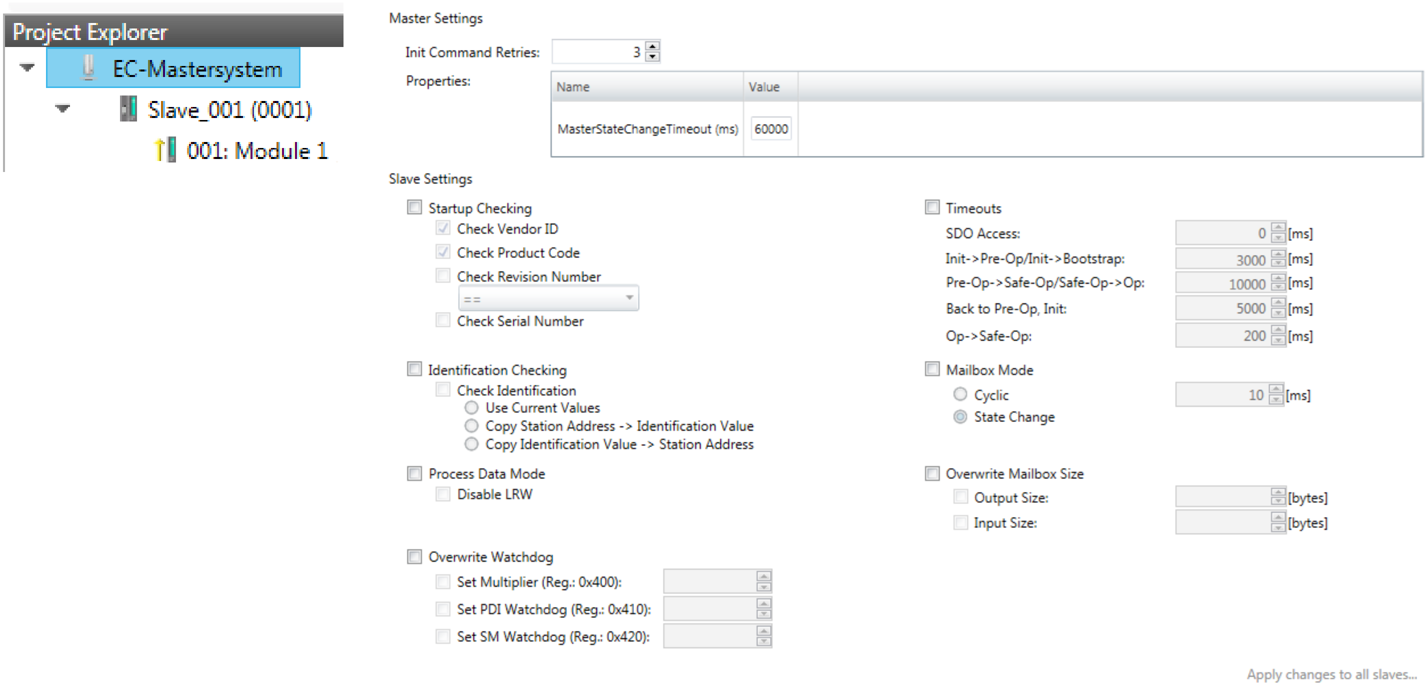
Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of all the slave stations. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start Address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

> *Information about the assignment of the in/output area can be found in the manual of your module.*

The *'I/O Addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O Addresses EtherCAT'* are the offset addresses, which are used within the EtherCAT process image. You cannot change the address. You can use the addresses e.g. for EtherCAT network analysis.

If you have activated "Isochronous mode" via the feature set *'Motion Control + ... axes'*, you can use the *'Process image'* to place the address area of the corresponding module of a slave station in the OB61 process image. Otherwise, the address area is located in the OB1 process image or in the I/O area.

9.10.4.4 Advanced Options (Expert mode)



This dialog is only visible in the *'Expert mode'*! In this dialog the parameters of the master system can be adjusted and the default settings for all the slave stations can be defined.

- Master Settings
  - Init Command Retries: Number of retries, beyond which a transmission error is returned. (default: 3)
  - MasterStateChangeTimeout: Here you can define a timeout for the state change of the master and its slave stations (default: 60000ms). If the *MasterStateChangeTimeout* is too short, the EtherCAT master reports the error message 0xED21.
- Slave Settings
  - In this area default parameters can be applied for all the slave stations. The settings are applied for all slave stations as default setting by clicking on [Apply changes (to all slaves)]. By selecting the slave station in the *'Project Explorer'* you always have the possibility to customize the slave parameters via the register *'Advanced Options'*.

Slave Settings

- Start-up checking:
  Here you can define the items, the EtherCAT master has to check during the transition *'Init→Pre-Op'* (Vendor ID, Product code, Revision number).
  - Revision number can be verified:
    "==" → High word is equal, Low word is equal
    ">=" → High word is equal or greater, Low word is equal or greater
    "LW ==" → Low word is equal
    "LW ==, HW >=" → Low word is equal, High word is equal or greater
    "HW ==" → High word is equal
    "HW ==, LW >=" → High word is equal, Low word is equal or greater
- Identification checking:
  - With these parameters, you determine via which HotConnect address the EtherCAT master should identify the slave station.
  - *'Check identification'*: When activated, the text box below shows the current HotConnect address, which the EtherCAT master has to use to identify the slave station.
  - For identification via the address set on the address switch of the slave station (Explicit Device ID), you have to activate *'Check identification'* and enter the corresponding ESC register address for addressing via the address switch at *'Select local address'*.
  - For identification via SSI (Configured Station Alias) you have to activate *'Check identification'* and enter the corresponding ESC register address for SSI activation at *'Select local address'*. In this case, the *Configured Station Alias* address must be specified via *'EEPROM'* of the slave station in *diagnostics mode*. In addition, you must specify the *Configured Station Alias* address in your configuration in *'Group'* by means *'Identification value'*.
    Chap. 9.10.8.4 'EEPROM (Expert mode)' page 267
    Chap. 9.10.9.3 'Create Hot Connect group' page 273

For more information about the ESC register addresses, refer to the manual for your slave station.

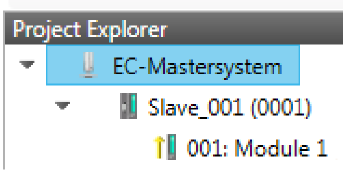
- Process Data Mode:
  Here you specify the command that should be used for process data access.
  – *'LRW activate:'* With one **L**ogical-**R**ea**d**-**L**ogical-**Wr**ite command inputs are read and also outputs are set. This needs 1 frame.
  – *'LRW deactivate:' 'LRD/LWR:'* Read access with **L**ogical-**R**ea**d** command to inputs and write access with **L**ogical-**Wr**ite command to outputs. This needs 2 frames.
- Overwrite Watchdog:
  Writes the configured value in the relevant register of the slave station. Here among others you can set the time of the *'SM Watchdog'* (SyncManager-Watchdog).
  – *'Set Multiplier'*: Writes the configured value to the corresponding slave register: 0x0400
  – *'Set PDI Watchdog'*: Writes the configured value to the corresponding slave register: 0x0410
  – *'Set SM Watchdog'*: Writes the configured value to the corresponding slave register: 0x0420

Please note that even if a watchdog is present, this need not be indicated in the ESI file and this is shown as inactive!

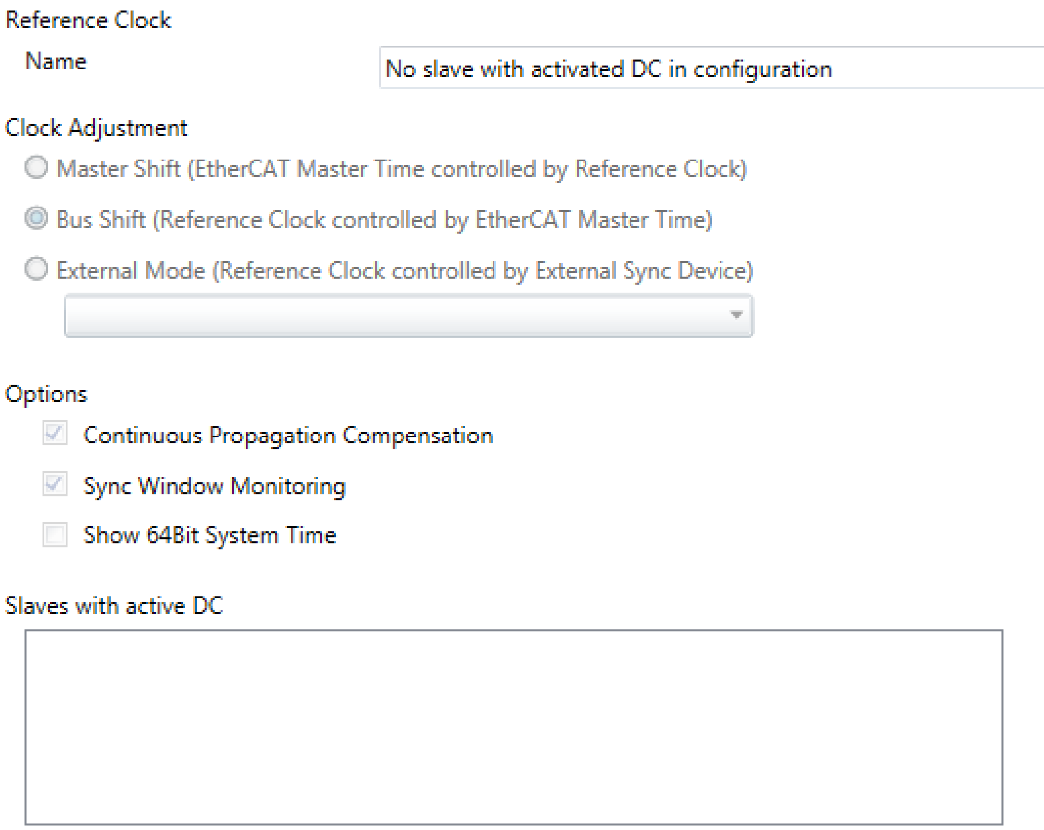
- Timeouts:
  – *'SDO Access'*: Internal master timeout for SDO access
  – *'Init→Pre-Op'*: Internal master timeout for slave state change from *Init* to *Pre-Op*
  – *'Pre-Op→Safe-Op/Safe-Op→Op'*: Internal master timeout for slave state change from *Pre-Op* to *Safe-Op* and then to *Op.*
  – *'Back to Pre-Op, Init'*: Internal master timeout for slave state change to *Pre-Op* and *Init*
  – *'Op→Safe-Op'*: Internal master timeout for slave state change from *Op* to *Safe-Op* *🡆 Chap. 9.1.2 'EtherCAT State Machine' page 192*
- Mailbox Mode:
  The *'Mailbox'* is an a-cyclic communication channel. Here mostly *'Emergencies'* messages and *'SDOs'* are buffered. The way of accessing the just unread mailbox data can be specified here.
  – *'Cyclic'*: Interval in ms within which the mailbox is to be read (polling mode). If you want short interrupt response times, you should select the mode *'Cyclic'* and set a short time e.g. 1ms.
  – *'State change'*: The mailbox is read only on a state bit change.
- Overwrite Mailbox Size
  – *'Output Size'*: Overwrites mailbox output size
  – *'Input Size'*: Overwrites mailbox input size

– When changing the 'Process Data Mode' you have to refresh the addresses in the Register 'Process Image'.
– If the Process Data Mode 'LRW' is used, the input and the output address of the EtherCAT process image must be identical. Here address leaks can occur between slave stations. If an EtherCAT address exceeds the maximum address area of the CPU, the current configuration gets invalid. You need to reduce the configuration or change to process data mode 'LRD/LWR'.
– If you use long cycle times (> 100ms) you should always accordingly raise the 'SM Watchdog'. Otherwise your slave station changes after laps of 'SM Watchdog' time to Safe-Op and releases OB 86. From now on you can only manually set the slave to Op!

9.10.4.5 Distributed Clocks (Expert mode)



Due to the hardware, with local connections the function 'distributed clocks' is not supported.

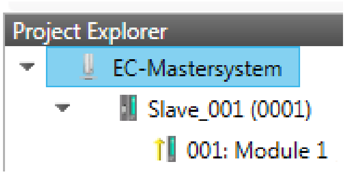


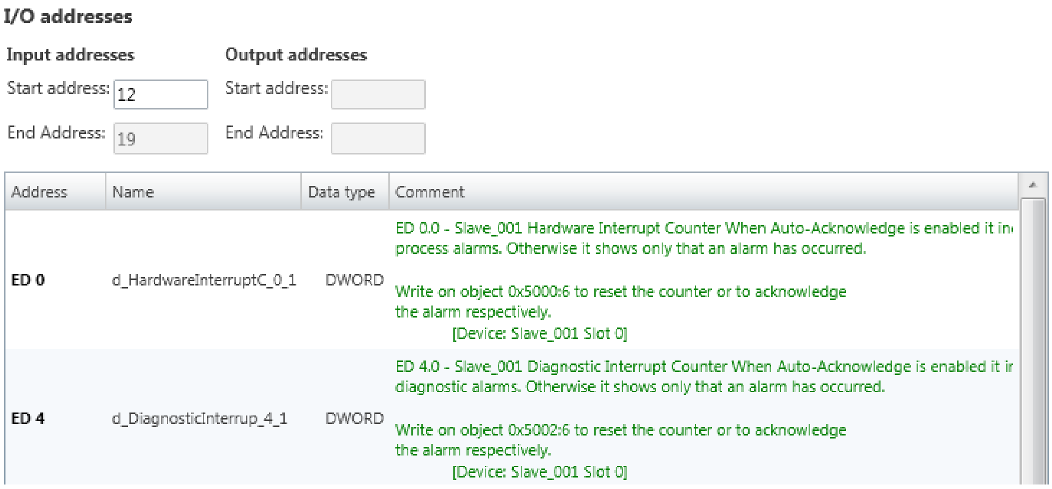
This dialog is only visible in the *'Expert mode'*! Here you can adjust the clock functionality accordingly. In EtherCAT "Distributed Clocks" means a logical combination of "clocks", which are located in the EtherCAT devices. With this there is the possibility to locally provide a synchronized time in each bus device. If an EtherCAT device supports the Distributed Clocks functionality, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This reference clock so represents the system time.

- Reference clock: Here you get information about the clock, which provides the reference time.
  – Name: Name of the reference clock. Per default this is always the 1. slave station, which supports the "Distributed Clock (DC)" functionality.
- Clock adjustment
  – Master Shift: The EtherCAT master time is synchronized by the reference clock.
  – Bus Shift: The reference clock is synchronized by the EtherCAT master time.
  – External Mode: The reference clock is controlled by an external master

- Options
  – Continuous Propagation Compensation: A command (datagram) will be inserted in the cyclic frame which allows the EtherCAT master to measure and compensate the propagation delay time by time.
  – Sync Window Monitoring: A command (datagram) will be inserted in the cyclic frame to read the ESC registers 0x092C. If this is selected the master will throw a notification about the state (*sync* respectively *out-of-sync*) of your system.
  – 64bit system time: Master supports slaves with 32bit and 64bit system time register (0x0910). If this is selected he will interpret it as 64bit system time.
- Slaves with active DC
  – Shows a list of all slave stations with active DC

9.10.4.6 I/O Address Overview





Here you have a list of addresses that are used by the I/O components of all the modules in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

Information about the assignment of the in/output area can be found in the manual of your module.

9.10.5 Configuration - slave station

9.10.5.1 Preparation

Click in the Toolbar at [Configuration] and select the Slave-Station *'Slave_...'* in the *'Project Explorer'*. The following registers are available now:

↳ *Chap. 9.10.5.2 'General' page 245*

↳ *Chap. 9.10.5.3 'Modules' page 246*

↳ *Chap. 9.10.5.4 'PDO Mapping' page 247*

Group - if a group exists for this slave station

↳ *Chap. 9.10.9 'Grouping logic' page 270*

↳ *Chap. 9.10.5.5 'Advanced Options (Expert mode)' page 250*

↳ *Chap. 9.10.5.6 'Ethernet (EoE)' page 253*

↳ *Chap. 9.10.5.7 'Distributed Clocks (Expert mode)' page 254* - if supported

↳ *Chap. 9.10.5.8 'Init Commands (Expert mode)' page 255*

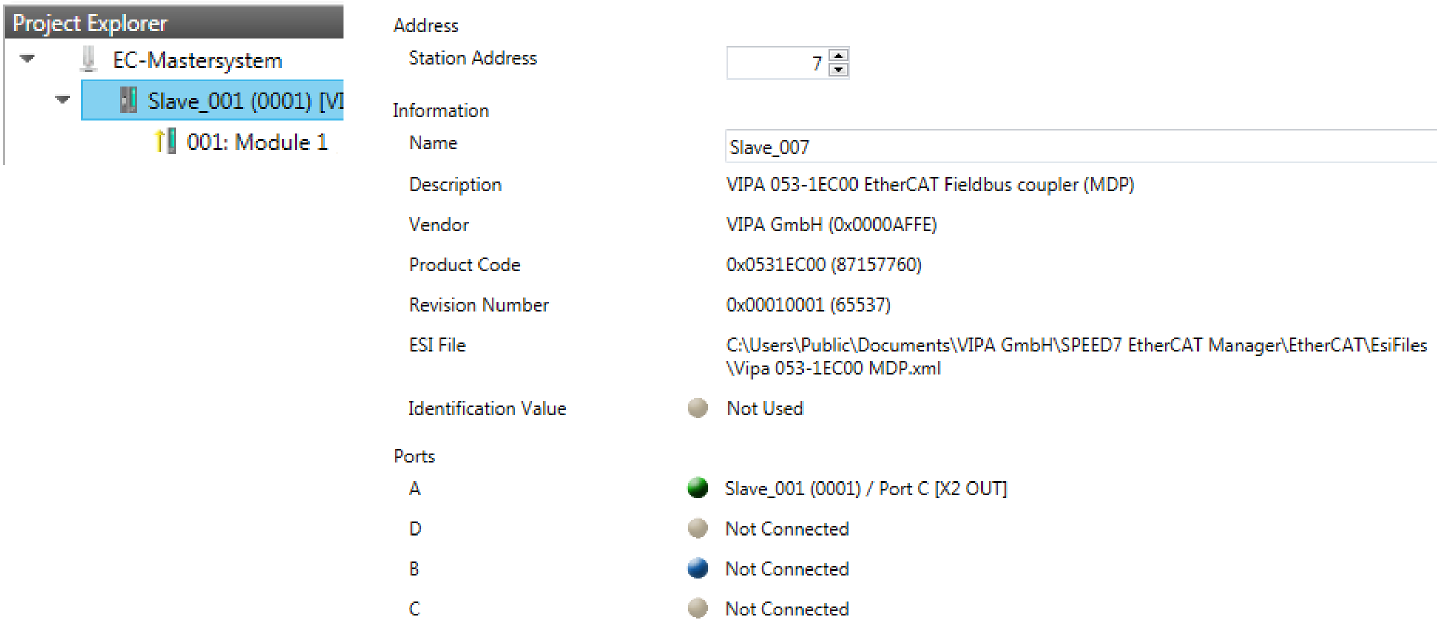
↳ *Chap. 9.10.5.9 'CoE Object Dictionary (Expert mode)' page 257*

↳ *Chap. 9.10.5.10 'Process Image' page 258*

↳ *Chap. 9.10.5.11 'I/O Address Overview' page 258*

↳ *Chap. 9.10.5.12 'Parameter' page 259*

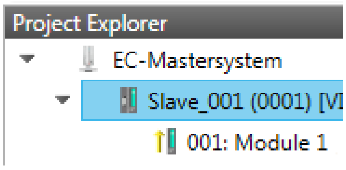
9.10.5.2 General

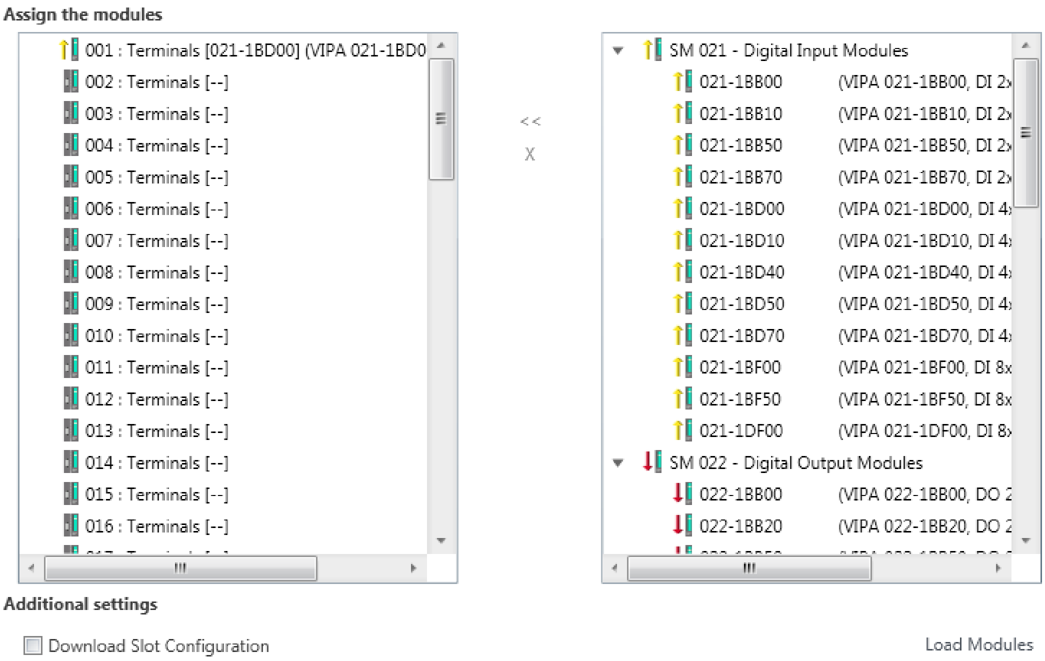


Here you can perform slave-specific settings such as assignment of name and address to a station. It is also possible to change the connection to the station.

- Address
  – Station Address: EtherCAT address of the slave station.
- Information
  – Name: Name of the slave station can be assigned accordingly.
  – Description: Description of the slave station.
  – Vendor: Name of the vendor.
  – Product Code: Internal product code of the slave station.
  – Revision Number: Internal revision number of the slave station.
  – ESI File: Path and name of the device file, in which the data of the slave station is stored.
  – Identification Value: Identification Value of the slave station
- Ports
  – Connected Devices: List of connected slave stations.
  – Predecessor Device: Name of the predecessor device.
  If topology should be changed, please use the *'Edit Topology'* dialog.

9.10.5.3 Modules





With an E-Bus slave this dialog is hidden. *Chap. 9.10.9 'Grouping logic' page 270*

In this dialog you can assign modules to the appropriate slot.

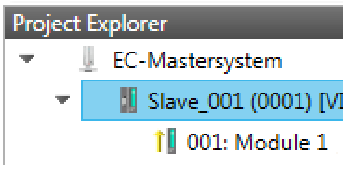
- Connect module to slot ("<<")
  Select your module from the list on the right and add it to a selected slot *'Terminals'* in the left list by clicking [<<]. This takes place according to the following rules:
  – If no modules are configured, the module is connected to the highlighted slot. Each additional module is inserted below.
  – If modules are already exist, the module is added to the highlighted slot and the following modules are moved accordingly.
- Disconnect module from slot ("X")
  – Select from the left list the appropriate slot, which you want to disconnect from the module again and click at ["X"].

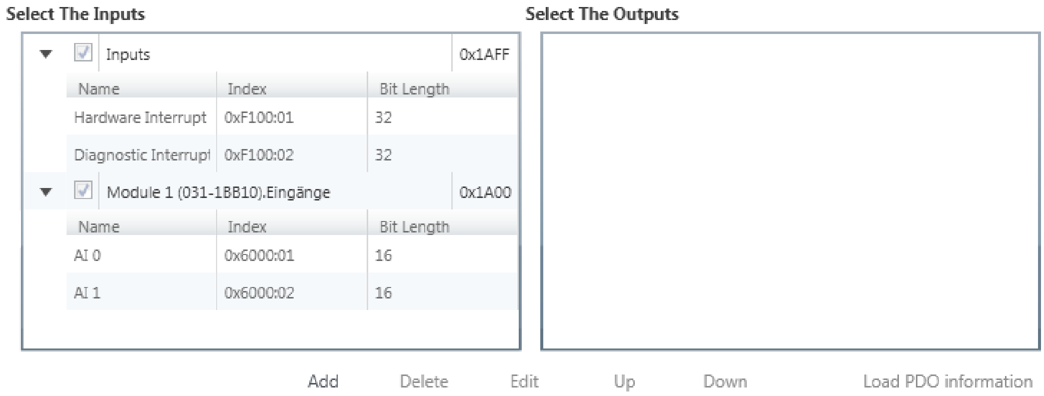
There is also the possibility to add or remove modules via the context menu of the 'Project Explorer'.

- Option field - *'Download slot configuration'*
  When enabled, an Init Command is created, which contains the slot configuration with the unique module identifier. During start-up of the slave station the slot configuration serves for comparison between configured and inserted modules. This can prevent misconfigurations.
- *'Load modules'*
  With this function you can load the configuration from the EtherCAT master for the selected slave station.

9.10.5.4 PDO Mapping

9.10.5.4.1 Description

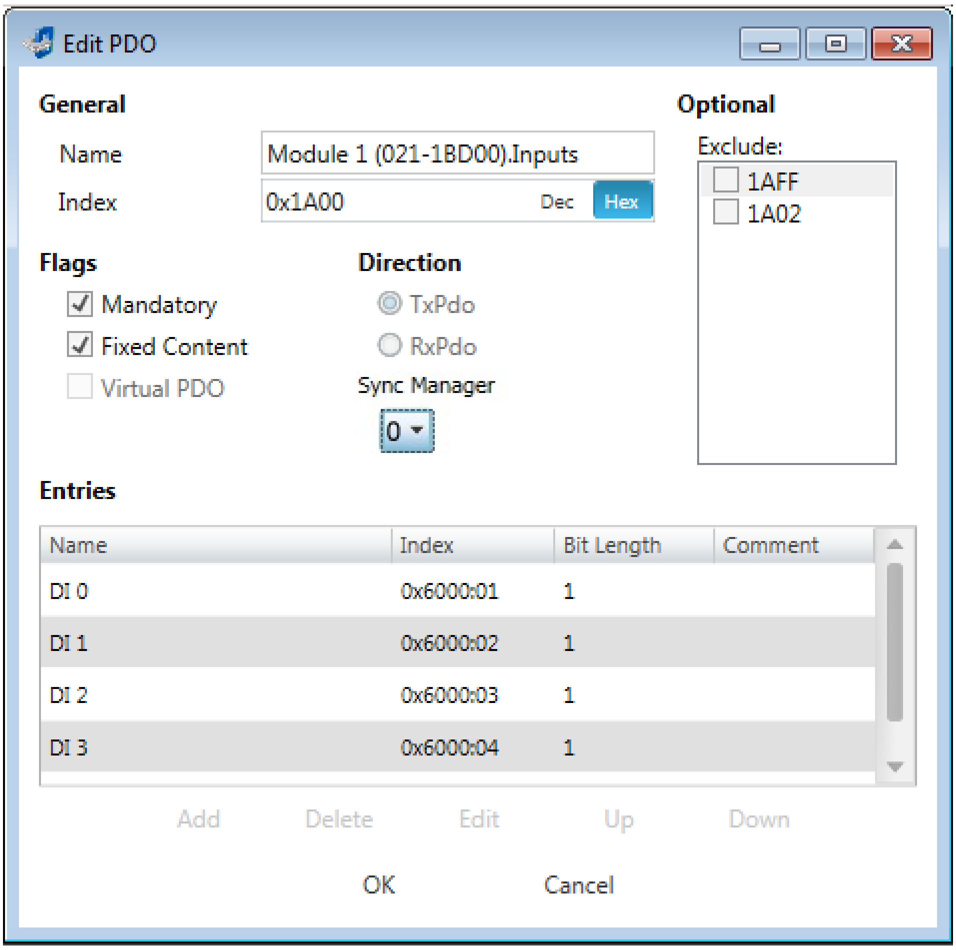




This dialog shows a list of the assigned PDOs. With some slave stations it is possible to activate respectively de-activate certain PDO configurations.

- Select the Inputs
  - If your slave station supports it, you can hide the corresponding input PDO from the configuration by disabling the checkbox.
- Select the Outputs
  - If your slave station supports it, you can hide the corresponding output PDO from the configuration by disabling the checkbox.
- Only *'**Expert mode**'*
  - Add / Delete / Edit:
    Used for changing the lists, if it is allowed by the ESI. First the list, which you want changed, must be selected.
  - Up / Down:
    Moving the selected PDO in the selected list up or down.
  - Load PDO information:
    Here you can load PDO information directly from the slave station.

9.10.5.4.2 Add or edit PDO (Expert mode)



PDOs can only be edited in the *'Expert mode'*! Otherwise, the functions are hidden. With [Edit] the dialog *'Edit PDO'* opens.

- General
  - Name: Name of the PDO
  - Index: Index of the PDO (can be entered in hexadecimal or decimal)
- Flags
  - Mandatory: If activated the PDO cannot be deleted.
  - Fixed Content: If activated the content of the PDO is write protected. to create new or to edit existing PDOs you have to disable *'Fixed Content'*.
  - Virtual PDO: If activated the PDO has no entries.
- Direction
  - TxPDO: Send PDO of the slave station for input data.
  - RxPDO: Receive PDO of the slave station for output data.
- Sync Manager
  - Selected the sync manager, which should be used. The selection is only visible if more than one sync manager can be used.
- Optional
  - Exclude: Select the PDOs which cannot be activated if this PDO is activated.
- Entries
  - Here is the list of configured PDO entries shown.

After editing the PDOs, the addresses need to be re-calculated! For this jump to register 'Process Image' and click at [Recalculate].

9.10.5.4.3 Add PDO (Expert mode)

Via the following dialog the user add a PDO entry.

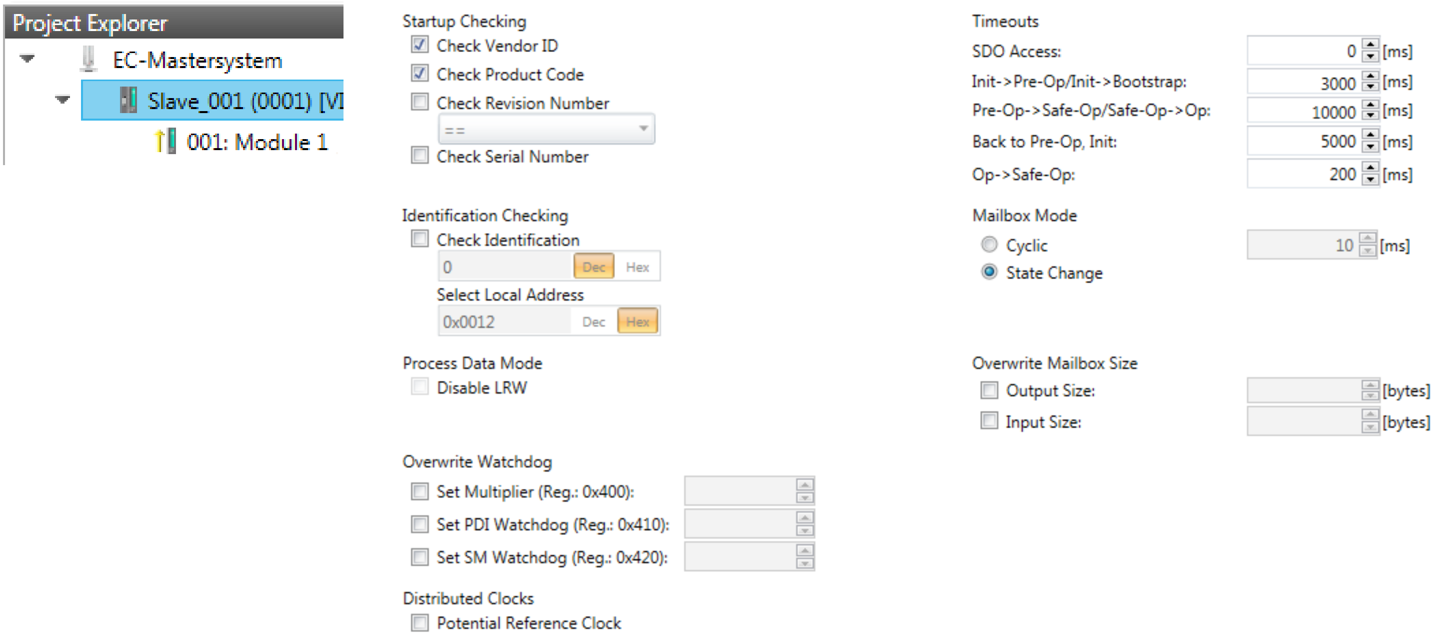
- General
  - Name: Name of the PDO entry
  - Comment: Comment of the PDO entry
  - Swapping: Swapping mode of the PDO entry
- Settings
  - Index: Index of the PDO entry (can be entered in hexadecimal or decimal)
  - Subindex: Subindex of the PDO entry (hexadecimal)
  - Datatype: List of available datatypes
  - Bit Length: Length of the PDO entry in bits
- CoE Object-Dictionary (loaded only if Object-Dictionary is supported by slave)

9.10.5.4.4 Edit PDO (Expert mode)

Via the following dialog the user adit a PDO entry.

- General
  - Name: Name of the PDO entry
  - Comment: Comment of the PDO entry
  - Swapping: Swapping mode of the PDO entry

9.10.5.5 Advanced Options (Expert mode)



This dialog is only visible in the *'Expert mode'*! Here you can make further adjustments to the slave station.

Slave Settings

- Start-up checking:
  Here you can define the items, the EtherCAT master has to check during the transition *'Init→Pre-Op'* (Vendor ID, Product code, Revision number).
  – Revision number can be verified:
    "==" → High word is equal, Low word is equal
    ">=" → High word is equal or greater, Low word is equal or greater
    "LW ==" → Low word is equal
    "LW ==, HW >=" → Low word is equal, High word is equal or greater
    "HW ==" → High word is equal
    "HW ==, LW >=" → High word is equal, Low word is equal or greater
- Identification checking:
  – With these parameters, you determine via which HotConnect address the EtherCAT master should identify the slave station.
  – *'Check identification'*: When activated, the text box below shows the current HotConnect address, which the EtherCAT master has to use to identify the slave station.
  – For identification via the address set on the address switch of the slave station (Explicit Device ID), you have to activate *'Check identification'* and enter the corresponding ESC register address for addressing via the address switch at *'Select local address'*.
  – For identification via SSI (Configured Station Alias) you have to activate *'Check identification'* and enter the corresponding ESC register address for SSI activation at *'Select local address'*. In this case, the *Configured Station Alias* address must be specified via *'EEPROM'* of the slave station in *diagnostics mode*. In addition, you must specify the *Configured Station Alias* address in your configuration in *'Group'* by means *'Identification value'*.
    Chap. 9.10.8.4 'EEPROM (Expert mode)' page 267
    Chap. 9.10.9.3 'Create Hot Connect group' page 273

For more information about the ESC register addresses, refer to the manual for your slave station.

- Process Data Mode:
  Here you specify the command that should be used for process data access.
  – *'LRW activate:'* With one **L**ogical-**R**ea**d**-**L**ogical-**Wr**ite command inputs are read and also outputs are set. This needs 1 frame.
  – *'LRW deactivate:'* *'LRD/LWR:'* Read access with **L**ogical-**R**ea**d** command to inputs and write access with **L**ogical-**Wr**ite command to outputs. This needs 2 frames.
- Overwrite Watchdog:
  Writes the configured value in the relevant register of the slave station. Here among others you can set the time of the *'SM Watchdog'* (SyncManager-Watchdog).
  – *'Set Multiplier'*: Writes the configured value to the corresponding slave register: 0x0400
  – *'Set PDI Watchdog'*: Writes the configured value to the corresponding slave register: 0x0410
  – *'Set SM Watchdog'*: Writes the configured value to the corresponding slave register: 0x0420

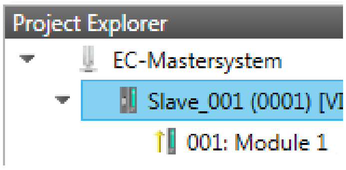
Please note that even if a watchdog is present, this need not be indicated in the ESI file and this is shown as inactive!

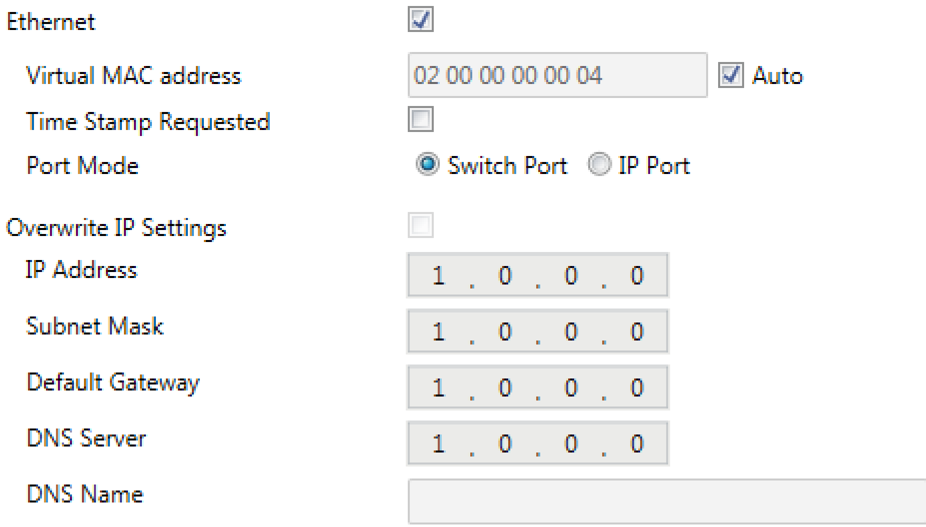
- Timeouts:
  – *'SDO Access'*: Internal master timeout for SDO access
  – *'Init→Pre-Op'*: Internal master timeout for slave state change from *Init* to *Pre-Op*
  – *'Pre-Op→Safe-Op/Safe-Op→Op'*: Internal master timeout for slave state change from *Pre-Op* to *Safe-Op* and then to *Op.*
  – *'Back to Pre-Op, Init'*: Internal master timeout for slave state change to *Pre-Op* and *Init*
  – *'Op→Safe-Op'*: Internal master timeout for slave state change from *Op* to *Safe-Op* *Chap. 9.1.2 'EtherCAT State Machine' page 192*
- Mailbox Mode:
  The *'Mailbox'* is an a-cyclic communication channel. Here mostly *'Emergencies'* messages and *'SDOs'* are buffered. The way of accessing the just unread mailbox data can be specified here.
  – *'Cyclic'*: Interval in ms within which the mailbox is to be read (polling mode). If you want short interrupt response times, you should select the mode *'Cyclic'* and set a short time e.g. 1ms.
  – *'State change'*: The mailbox is read only on a state bit change.
- Overwrite Mailbox Size
  – *'Output Size'*: Overwrites mailbox output size
  – *'Input Size'*: Overwrites mailbox input size

- *When changing the 'Process Data Mode' you have to refresh the addresses in the Register 'Process Image'.*
- *If the Process Data Mode 'LRW' is used, the input and the output address of the EtherCAT process image must be identical. Here address leaks can occur between slave stations. If an EtherCAT address exceeds the maximum address area of the CPU, the current configuration gets invalid. You need to reduce the configuration or change to process data mode 'LRD/LWR'.*
- *If you use long cycle times (> 100ms) you should always accordingly raise the 'SM Watchdog'. Otherwise your slave station changes after laps of 'SM Watchdog' time to Safe-Op and releases OB 86. From now on you can only manually set the slave to Op!*

- Distributed Clocks: *'Potential Reference Clock'*
  - Every slave station can be used as a *'Potential Reference Clock'* if the slave supports the DC registers. The setting is used, when you remove the slave with activated *'Potential Reference Clock'* e.g. via *'Hot Connect'*, then the master searches for a slave station where *'Potential Reference Clock'* is activated. If no slave is available, the first DC slave is used.

9.10.5.6 Ethernet (EoE)

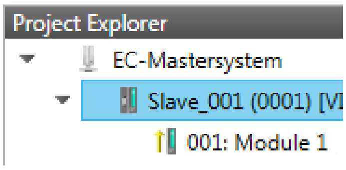


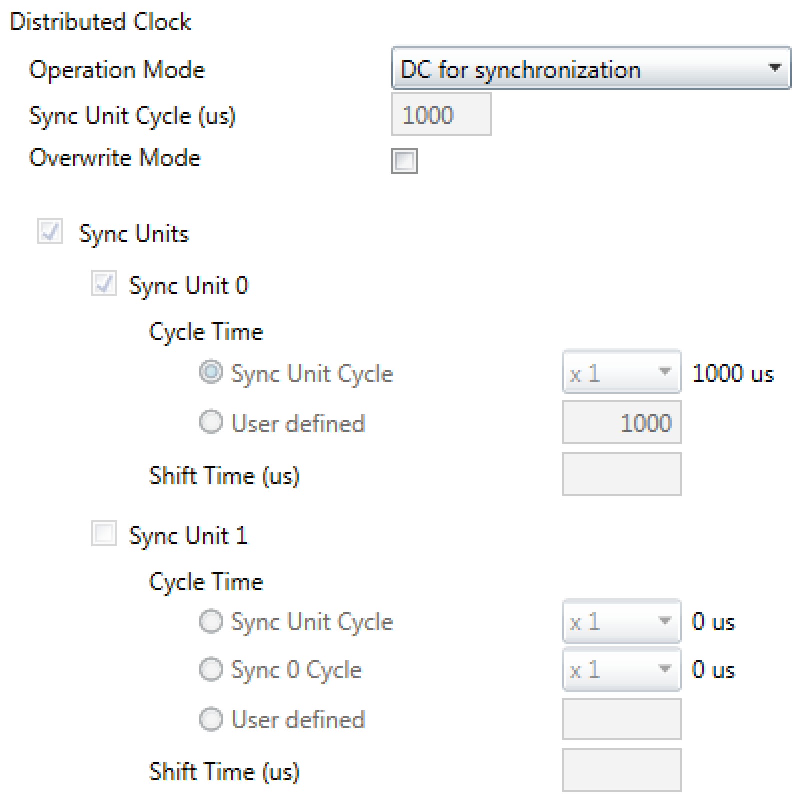


Here you activate or change EoE (**E**thernet **o**ver **E**therCAT) the settings.

- Ethernet (activates EoE support)
  - Virtual MAC address: Virtual MAC address. If *'Auto'* is checked, the Virtual MAC address will be generated from the Station Address, e.g. Station Address is "1010" (= 0x03F2), will generate the Virtual MAC address: "01 00 00 00 03 F2"
  - Time Stamp Requested: Slave station will response with the exact send time and the same Frame number and he should response as soon as possible.
  - Port Mode: Slave station can be run in *'Switch Port'* or *'IP Port'* mode.
- Override IP Settings
  - All IP settings will be overwritten from master like IP Address, Subnet Mask, Default Gateway, DNS Server and DNS Name.

9.10.5.7 Distributed Clocks (Expert mode)





This dialog is only visible in the *'Expert mode'* if this is supported by your slave station! Here you can adjust the settings for *Distributed Clocks* accordingly. In EtherCAT "**D**istributed **C**locks" (DC) means a logical combination of "clocks", which are located in the EtherCAT devices. With this there is the possibility to locally provide a synchronized time in each bus device. If an EtherCAT device supports the *Distributed Clocks* functionality, it has its own clock. After PowerON this first locally works, based on an own pulse generator. By selecting an EtherCAT slave station, which has to provide the reference time, the distributed clocks can be synchronized. This *reference clock* so represents the system time.

- Reference clock
  - Operation Mode: Here you can set the operation mode of the reference clock. More may be found in the manual of your slave station.
  - Sync Unit Cycle: Cycle time of the master. *Chap. 9.10.4 'Configuration - EC-Mastersystem' page 237*
- Sync Units
  - Sync Unit 0

    - Cycle Time: Here you can specify the cycle time in relation to the *'Master Cycle'* or *'User defined'*.

    - Time Shift: Specify here a time shift. This is used for fine adjustment.
  - Sync Unit 1

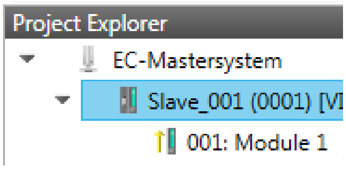
    - Cycle Time: Here you can specify the cycle time in relation to the *'Master Cycle'*, to the cycle of Sync Unit 0 *'Sync 0 Cycle'* or *'User defined'*.

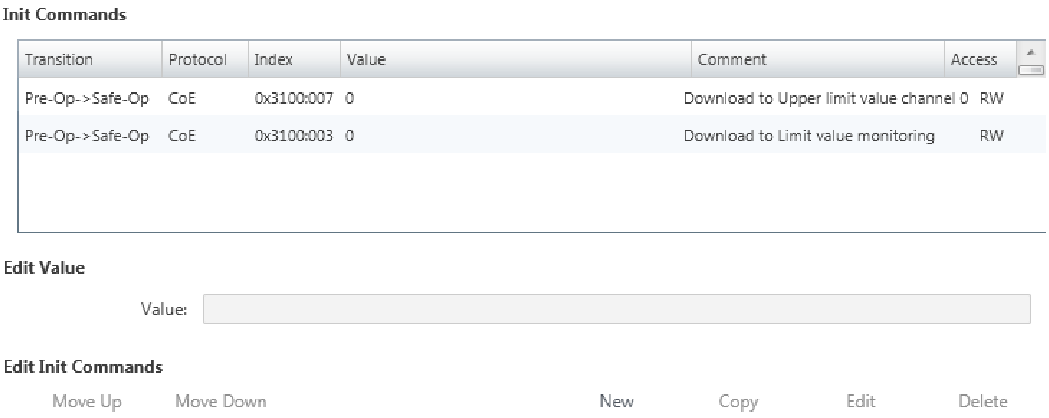
    - Time Shift: Specify here a time shift. This is used for fine adjustment.

Due to the hardware with a local connection Distributed Clocks (connection via network adapter) is not supported!

9.10.5.8 Init Commands (Expert mode)

9.10.5.8.1 Description





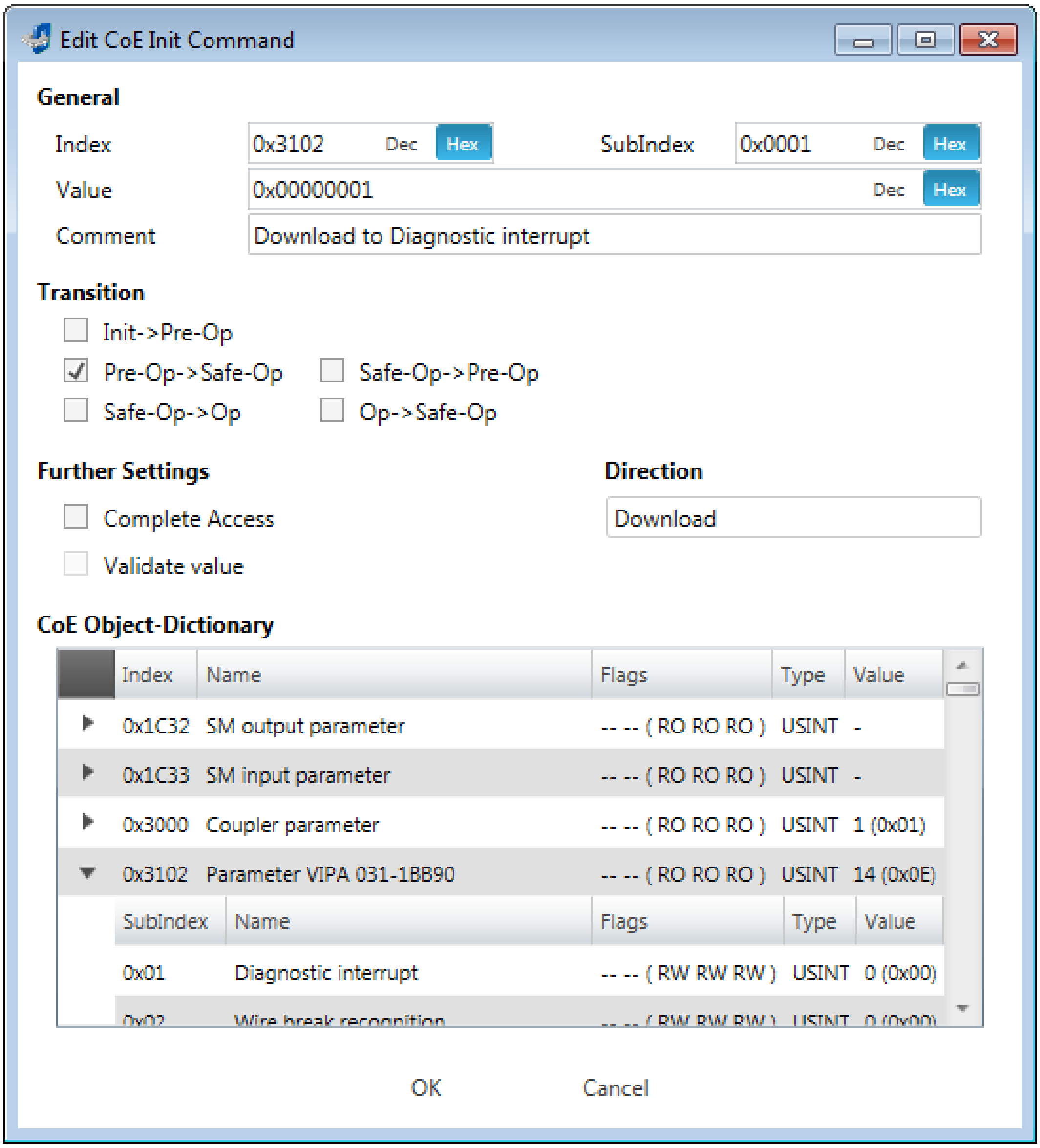
This dialog is only visible in the *'Expert mode'*!

> – *For each parameter of a slave station or module, which differs from the standard setting you have to create an Init command!*
> – *If a write access to an object in the configuration mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init Commands'. ↳ Chap. 9.10.5.9 'CoE Object Dictionary (Expert mode)' page 257*

Here you can see a list of the current configured Init Commands and if it is allowed you can also add/edit/delete the commands.

- Init Commands: Init Commands come from the ESI file or are automatically generated on write access to CoE objects or can be created by the user. You either have full-access (RW = Read/Write) or only read access (RO = Read-only). Init commands from ESI files are automatically listed here. These cannot be changed or deleted.
- Edit Init Commands
  – New, Copy, Edit, Delete: Used for changing Init Commands.
  – Move Up, Move Down: Moving the selected Init Command up or down.

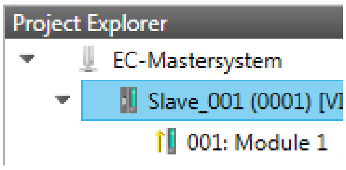
9.10.5.8.2 CoE Init Command (Expert mode)

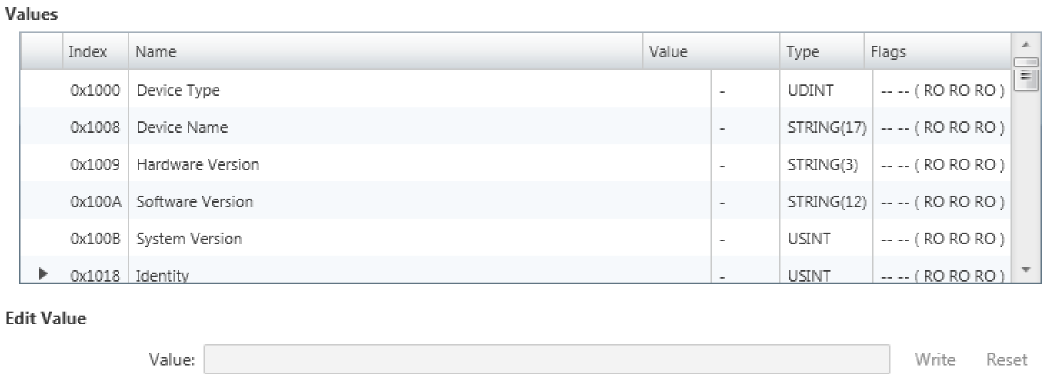


This dialog is only visible in the *'Expert mode'*! With [New] the dialog *'Add CoE Init Command'* opens. This dialog also opens to edit CoE Init Commands, which just exist.

- General
  - Index/Subindex: CoE-Index respectively Subindex of the Init Command
  - Value: Value of the Init Command, which should be written in the chose transition (only available if *'Direction'* is set to *'Download'*). If type of data is unknown, the hex format must be used. (Example: "0011 2233 …").
  - Comment: Here you can comment your Init Command.
- Transition
  - Determines in which transition the Init Command will be executed.
- Further Settings
  - Complete Access: Determines if the complete SDO object should be written/read.
- Direction
  - Download: Writes value to slave station.
  - Upload: Reads value from slave.
- CoE Object Dictionary: Select here the value in the CoE Object Dictionary of the slave station, you want to edit.

9.10.5.9 CoE Object Dictionary (Expert mode)





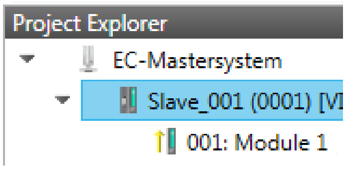
This dialog is only visible in the *'Expert mode'*! Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

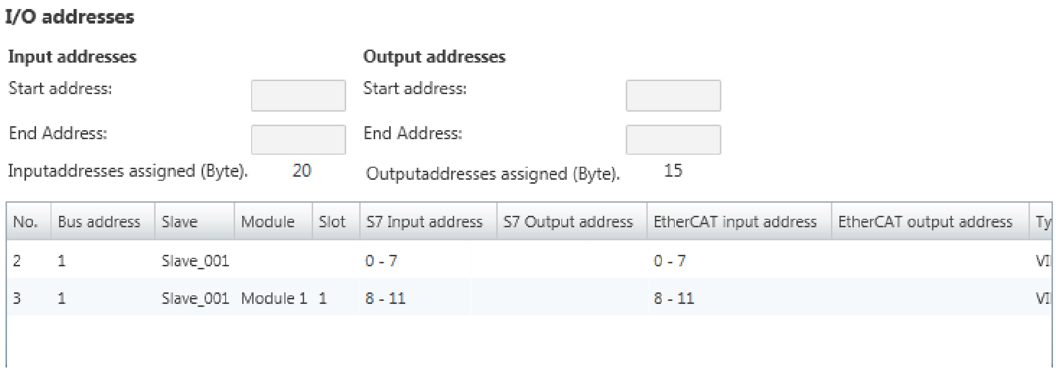
Description of the flags: "AA BB (CC DD EE)"

- AA, BB
  – Rx: Mapping as receive PDO
  – Tx: Mapping as send PDO
  – --: Mapping not allowed
- CC:
  – Access rights for state *PreOp* (RO, WO, RW)
- DD:
  – Access rights for state *SafeOp* (RO, WO, RW)
- EE:
  – Access rights for state *Op* (RO, WO, RW)
  Chap. 9.1.2 'EtherCAT State Machine' page 192
- Edit Value
  – Write: Changes the selected entry
  – Reset: Resets the selected entry to ESI default

If a write access to an object in the configuration mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init Commands'. *Chap. 9.10.5.8 'Init Commands (Expert mode)' page 255*

9.10.5.10 Process Image



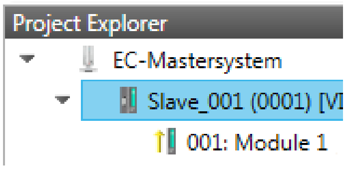


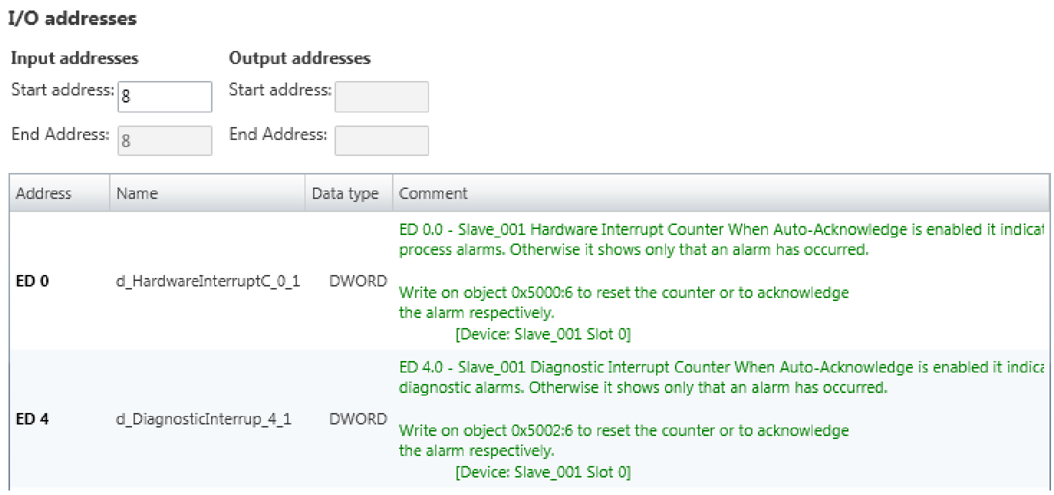
Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of the slave system. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

Information about the assignment of the in/output area can be found in the manual of your module.

The *'I/O addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O addresses EtherCAT'* are the addresses, which are used within the EtherCAT bus. You cannot change the address. You can use the addresses e.g. for EtherCAT network analysis.

9.10.5.11 I/O Address Overview

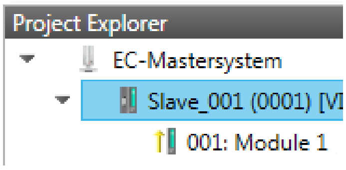




Here you have a list of addresses, which are used by the I/O components of the modules of the selected slave system in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

Information about the assignment of the in/output area can be found in the manual of your module.

9.10.5.12 Parameter



If the parameters of the slave station can be determined such as a slave station, the System SLIO slave parameters can be set here. With [Reset], the parameters of the slave station are reset to their default values.

> *More information about the parameters can be found in the manual of you slave station.*

9.10.6 Configuration - modules

9.10.6.1 Please note

> *With an E-Bus slave the dialog of the module configuration are hidden!*
> *Chap. 9.10.9 ‘Grouping logic’ page 270*

9.10.6.2 Preparation

Select in the configuration mode in the *‘Project Explorer’* the module of the according slave station. The following registers are available now:

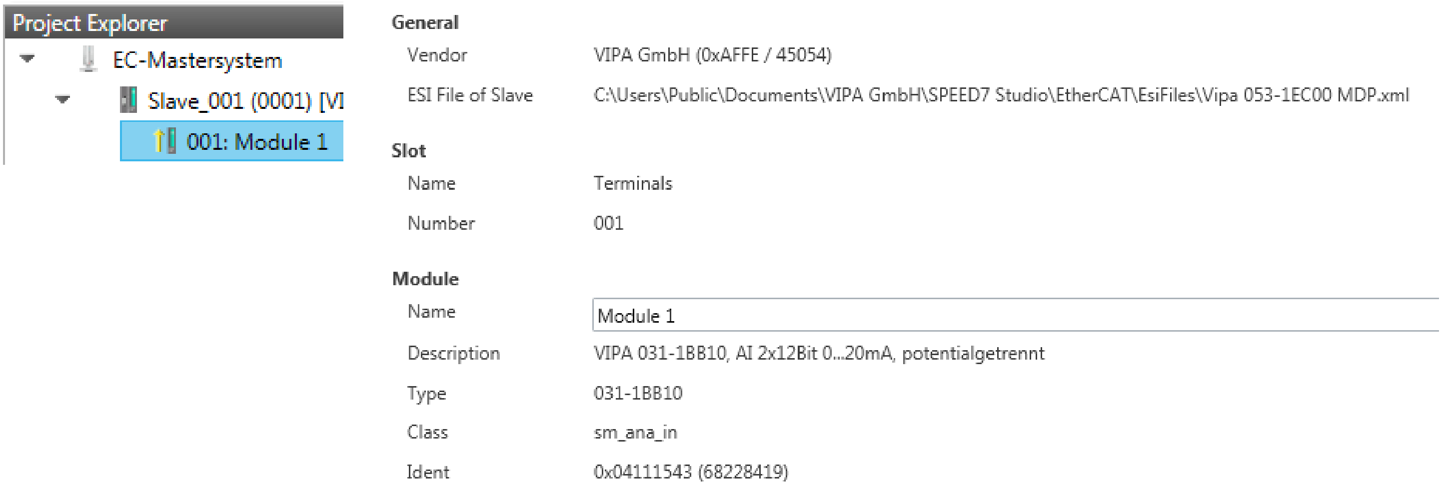
Chap. 9.10.6.3 ‘MDP Slot Properties’ page 260

Chap. 9.10.6.4 ‘Process Image’ page 260

Chap. 9.10.6.5 ‘I/O Address Overview’ page 261

Chap. 9.10.6.6 ‘Parameter’ page 261

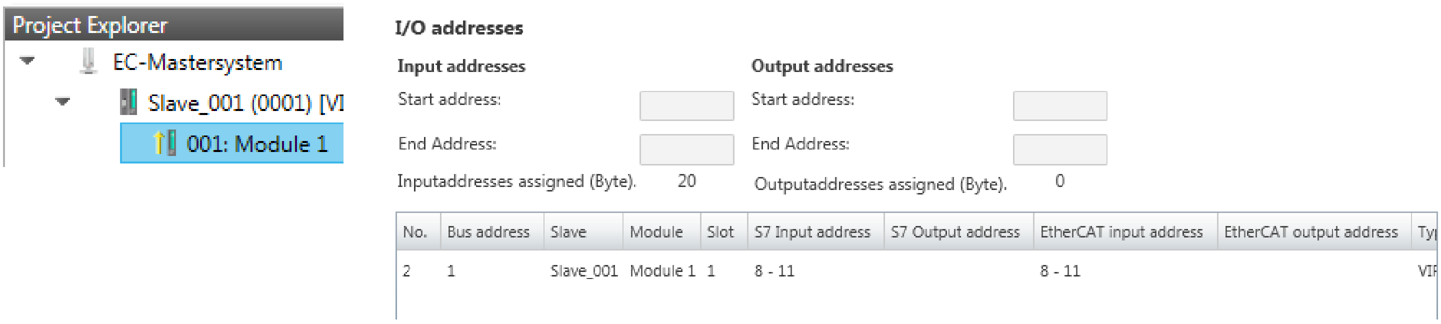
9.10.6.3 MDP Slot Properties



Here you can see the MDP Slot Properties of the corresponding module. This dialog serves for information. You cannot change something.

- General
  – Vendor: Name of the vendor of the module.
  – ESI file: Path and name of the device file, in which the data of the module and the associated slave station is stored.
- Slot
  – Name: Name of the slot
  – Number: Number of the slot
- Module
  – Name: Name of the module
  – Type: Order number of the module
  – Class: Module class
  – Identificator: Identification number of the according module class.

9.10.6.4 Process Image

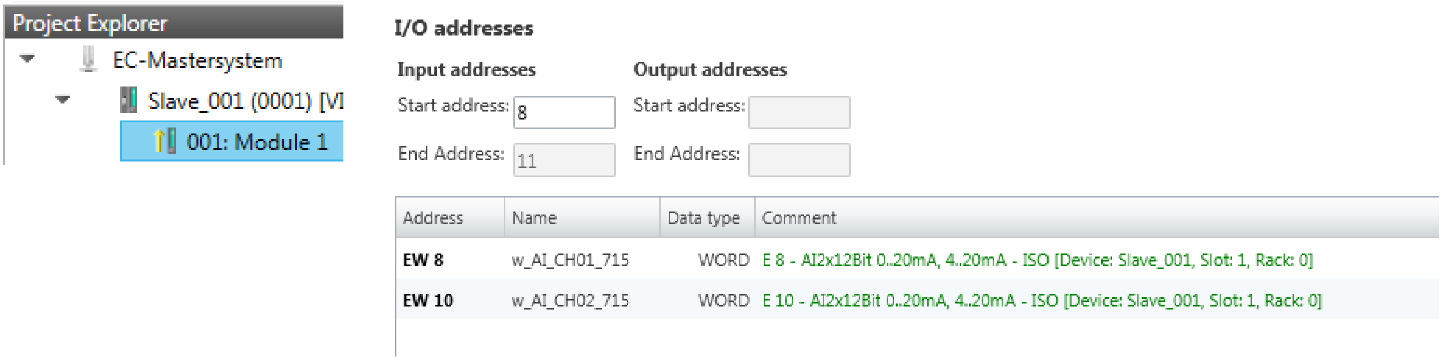


Here you have a list of S7 respectively EtherCAT addresses, which are used by the modules of all the slave stations. The *'S7 address'* corresponds to the address in the address area of the CPU. By entering a new *'Start Address'* you can adjust the S7 addressing of the input and output areas of the modules accordingly.

Information about the assignment of the in/output area can be found in the manual of your module.

The *'I/O Addresses EtherCAT'* are only visible in *'Expert mode'*! *'I/O Addresses EtherCAT'* are the addresses, which are used within the EtherCAT bus. You cannot change the addresses. You can use the addresses e.g. for EtherCAT network analysis.

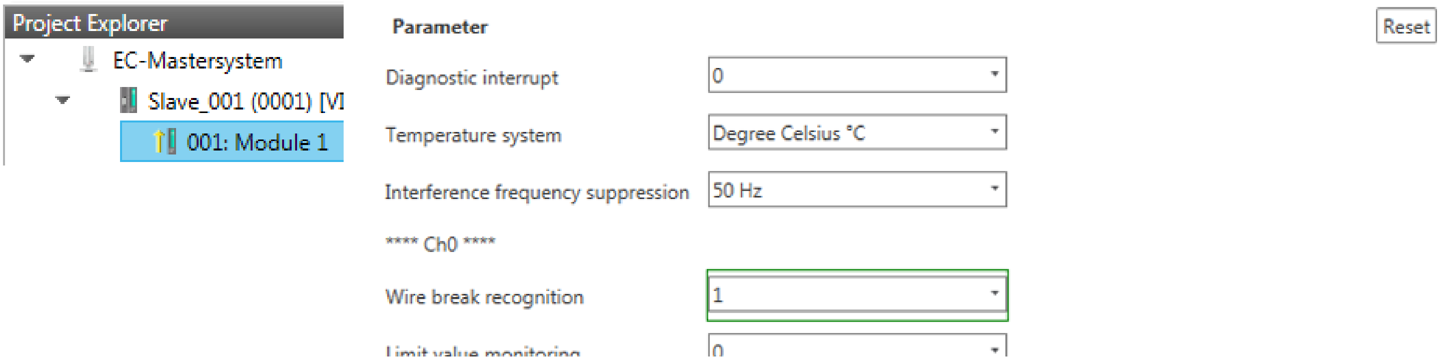
9.10.6.5 I/O Address Overview



Here you have a list of addresses that are used by the module in the address area of the CPU. By entering a new *'Start address'* you can adjust the addressing of the input and output areas accordingly. You can edit *'Name'* and *'Comment'* by clicking at the corresponding entry.

Information about the assignment of the in/output area can be found in the manual of your module.

9.10.6.6 Parameter



If there is a parametrizable module, whose parameters can be determined such as a System SLIO module, the module parameters can be set here. Here also the necessary Init command for the EtherCAT slave station is generated *Chap. 9.10.5.8 'Init Commands (Expert mode)' page 255*. With [Reset], the parameters of the module are reset to their default values.

More information about the parameters can be found in the manual of you module.

9.10.7 Diagnostics - EC-Mastersystem

9.10.7.1 Preparation

To use the *'Diagnostics'* functions, you must be connected online with your EtherCAT system.

1. Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*.
2. Activate in the *'Device editor'* the register *'Master'*.
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network Adapter* and click at [Select].
   - If you are connected to the PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With Yaskawa CPUs *Port* 6000 and *Master Instance* 0 is to be set.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.
4. Click in the Toolbar at [Diagnosis Mode].

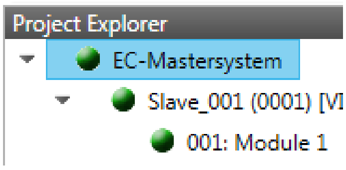
   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project Explorer'*.

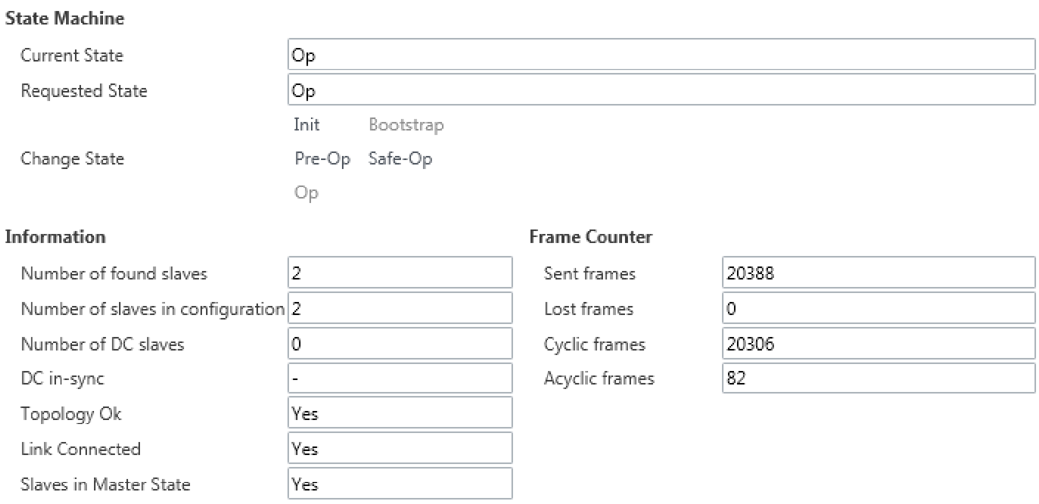
   With an online connection the 2 LEDs flash alternately in the *'Status area'*. In addition *'Modus'* switches to *'Diagnosis'*.
5. Click in the *'Project Explorer'* at Master.

   ⇨ The following registers are available now:

Chap. 9.10.7.2 'General' page 262
Chap. 9.10.7.3 'CoE Object Dictionary' page 264
Chap. 9.10.7.4 'History (Expert mode)' page 264

9.10.7.2 General





Colors and states

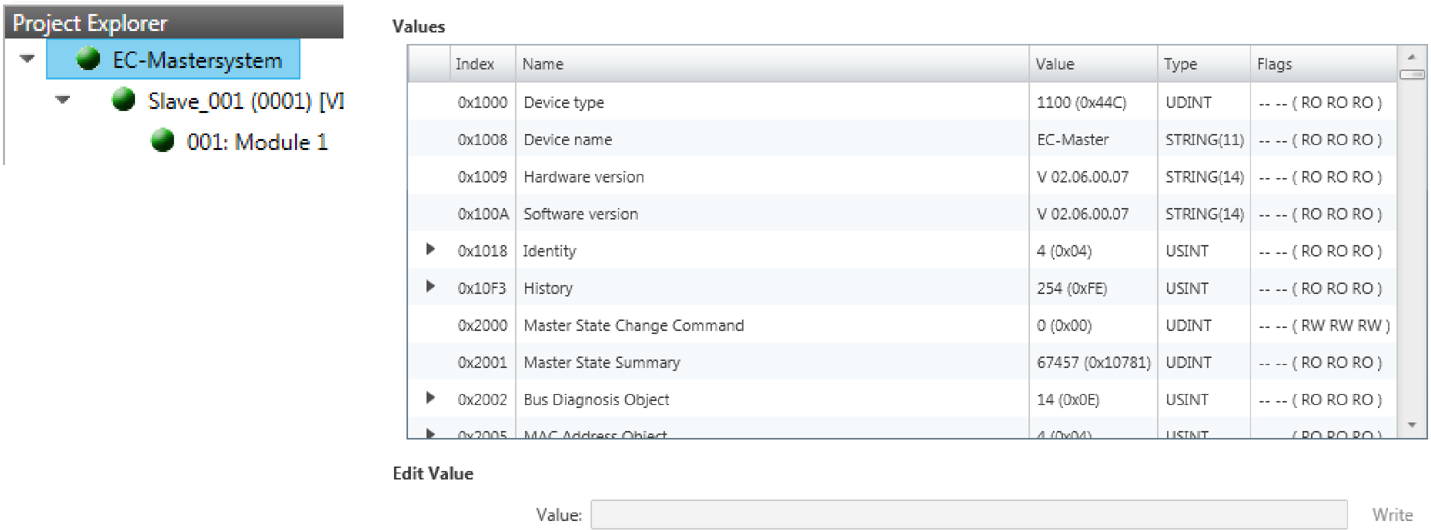
The state of the state machine can be determined via the color according to the following specifications:

Color	State of the state machine
- red	Init / Bootstrap
- blue	Pre-Op
- yellow	Safe-Op
- green	Op

Here you will get master and bus-specific information.

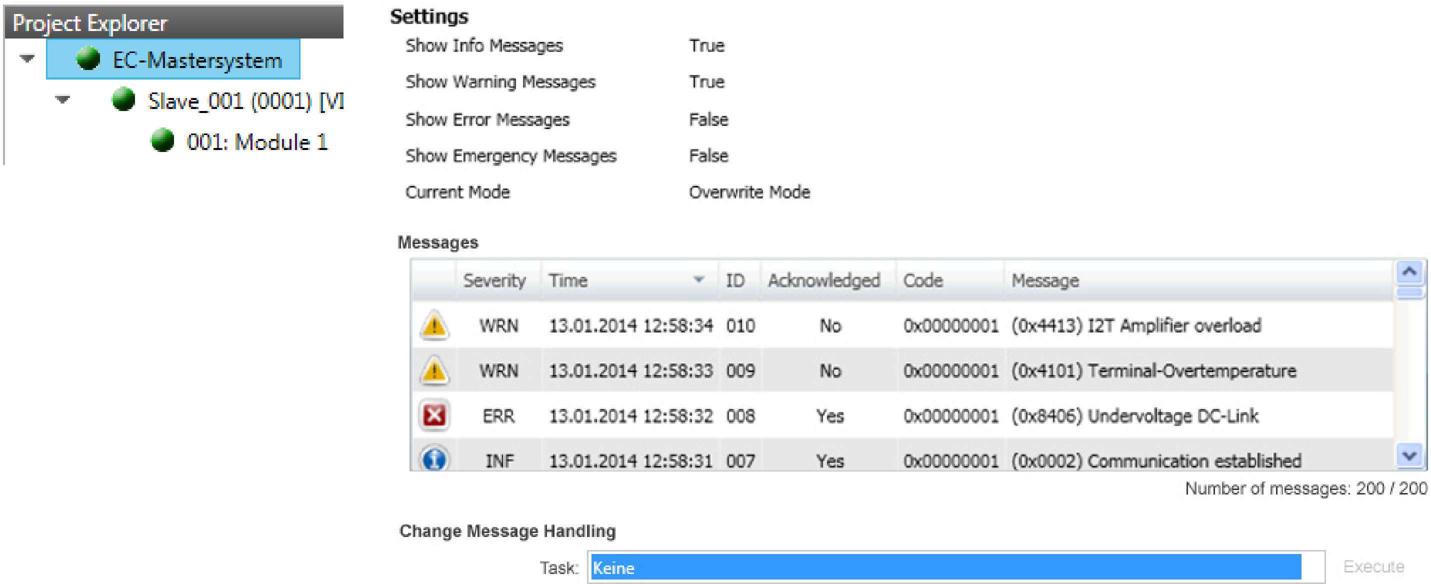
- State Machine
  - Current State: Shows the current state of master. *Chap. 9.1.2 'EtherCAT State Machine' page 192*
  - Requested State: Shows the currently requested state of the master which was requested by *'Change State '*.
  - Change State: Here you can change the state of the master.
- Information
  - Number of found slaves: Shows number of found slave stations at the bus.
  - Number of slaves in configuration: Shows number of configured slave stations at the bus.
  - Number of DC slaves: Shows the number of slave stations, which support distributed clocks functionality (DC).
  - DC in-sync: If distributed clocks is configured you can find here information about the synchronization status of the system.
  - Topology OK: The *'Topology'* is OK (*'Yes'*), if the number of configured matches the number of found slave stations. Here only the mandatory slaves stations are considered.
  - Link Connected: Here you will find *'Yes'*, if there is a physical connection to the configured slave stations.
  - Slaves in Master State: Here you will find *'Yes'*, if every configured slave station is in master state.
- Frame Counter
  - Sent frames: Number of sent frames since the last power cycle.
  - Lost frames: Number of lost frames since the last power cycle.
  - Cyclic frames: Number of cyclic frames since the last power cycle.
  - Acyclic frames: Number of acyclic frames since the last power cycle.

9.10.7.3 CoE Object Dictionary



Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

9.10.7.4 History (Expert mode)



In this dialog box, you can access all the diagnostic messages in the master and edit them if necessary. Via *'Settings'* they may be filtered accordingly.

9.10.8 Diagnostics - slave station

9.10.8.1 Preparation

To use the *'Diagnostics'* functions, you must be connected online with your EtherCAT system.

1. Click in the Toolbar at [Configuration] and select *'EC-Mastersystem'* in the *'Project Explorer'*.
2. Activate in the *'Device editor'* the register *'Master'*.
3. Set depending on the on-line access in the *'Device Editor > Master'* as follows:
   - If you are directly connected to a slave station via EtherCAT by means of a separate network adapter, select your *Network Adapter* and click at [Select].
   - If you are connected to the PG/OP channel of you CPU, please enter *IP Address*, *Port* and *Master Instance* and click at [Select]. With Yaskawa CPUs *Port* 6000 and *Master Instance* 0 is to be set.

   ⇨ The *SPEED7 EtherCAT Manager* uses the set connection for communication. By clicking on [Deselect] you can change the connection parameters.
4. Click in the Toolbar at [Diagnosis Mode].

   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project Explorer'*

   With an online connection the 2 LEDs flash alternately in the *'Status area'*. In addition *'Modus'* switches to *'Diagnosis'*.
5. Click in the *'Project Explorer'* at the according slave station *'Slave_...'*

   The following registers are available now:

Chap. 9.10.8.2 'General' page 265

Chap. 9.10.8.3 'ESC Register (Expert mode)' page 266

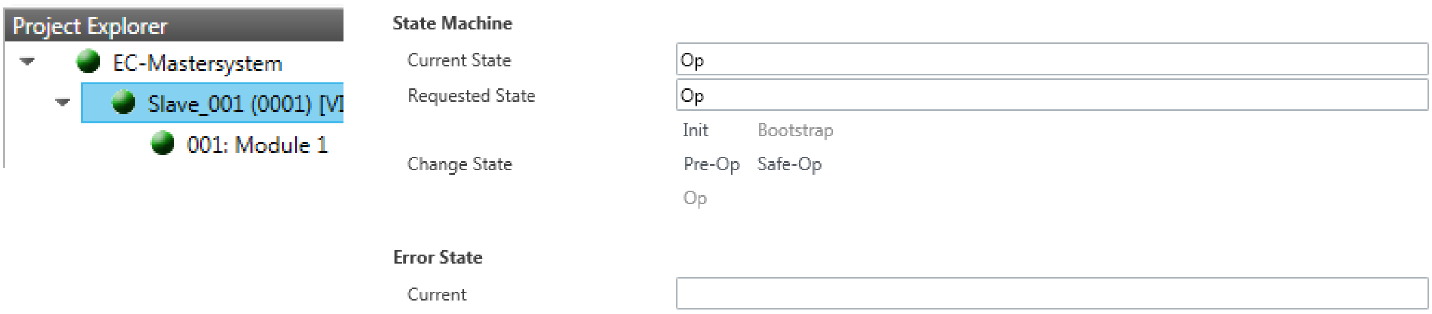
Chap. 9.10.8.4 'EEPROM (Expert mode)' page 267

Chap. 9.10.8.5 'Extended Diagnosis (Expert mode)' page 267

Chap. 9.10.8.6 'DC Diagnosis (Expert mode)' page 268

Chap. 9.10.8.7 'CoE Object Dictionary' page 268

9.10.8.2 General



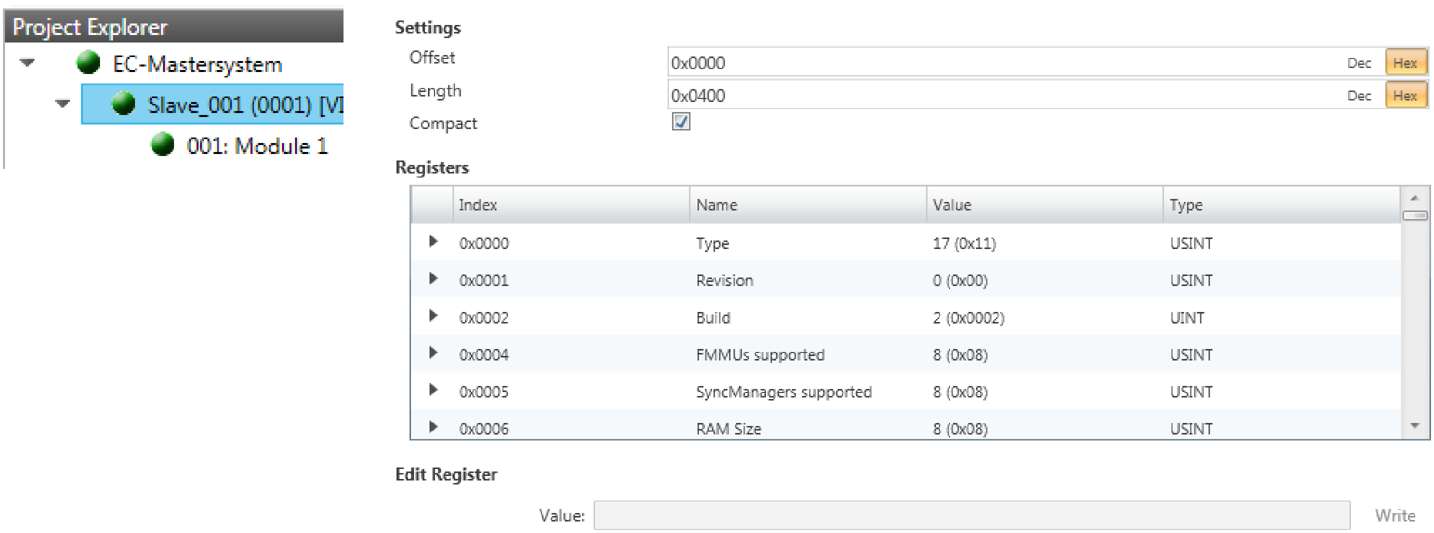
Colors and states

The state of the state machine can be determined via the color according to the following specifications:

Color	State of the state machine
- red	Init / Bootstrap
- blue	Pre-Op
- yellow	Safe-Op
- green	Op

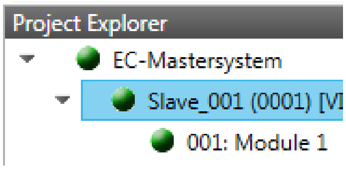
- State Machine
  - Current State: Shows the current state of the state machine of the slave station. *Chap. 9.1.2 'EtherCAT State Machine' page 192*
  - Requested State: Shows the requested state of the slave station.
  - Change State: Here you can change the state of the state machine of the slave station.
- Error State
  - Current: Slave error which occurred during state transition.

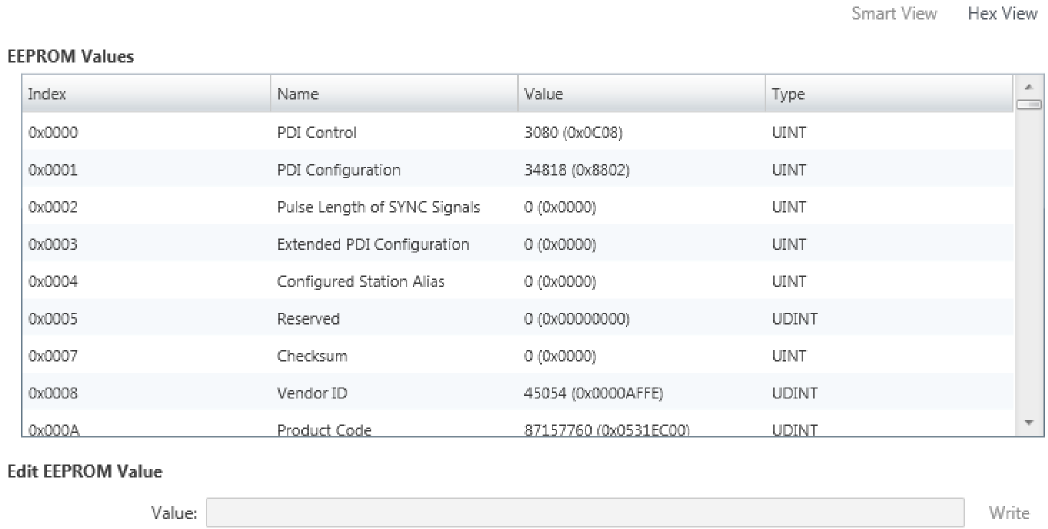
9.10.8.3 ESC Register (Expert mode)



This dialog is only visible in the *'Expert mode'*! Here you can directly access the registers of the EtherCAT ASIC. You should not make any changes here!

9.10.8.4 EEPROM (Expert mode)



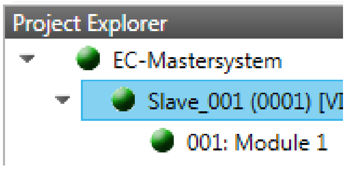


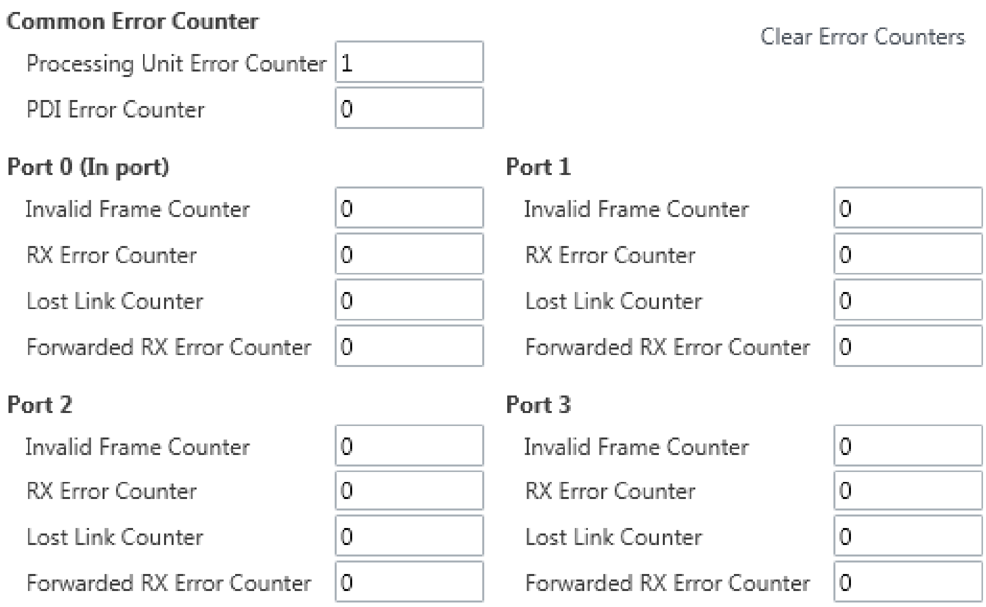
Index	Name	Value	Type
0x0000	PDI Control	3080 (0x0C08)	UINT
0x0001	PDI Configuration	34818 (0x8802)	UINT
0x0002	Pulse Length of SYNC Signals	0 (0x0000)	UINT
0x0003	Extended PDI Configuration	0 (0x0000)	UINT
0x0004	Configured Station Alias	0 (0x0000)	UINT
0x0005	Reserved	0 (0x00000000)	UDINT
0x0007	Checksum	0 (0x0000)	UINT
0x0008	Vendor ID	45054 (0x0000AFFE)	UDINT
0x000A	Product Code	87157760 (0x0531EC00)	UDINT

This dialog is only visible in the *'Expert mode'*! Here you can access the contents of the EEPROM of the slave station. Currently you can only change the parameter *'Configured Station Alias'*. This can be used for forming groups. *Chap. 9.10.9 'Grouping logic' page 270*

CAUTION!
Please regard that your slave station can get unusable by entering incorrect values especially in *'Hex view'*! In this case, any warranty of the vendor is excluded!

9.10.8.5 Extended Diagnosis (Expert mode)

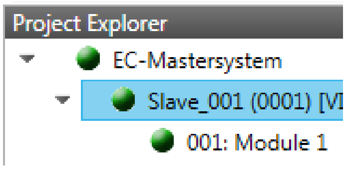


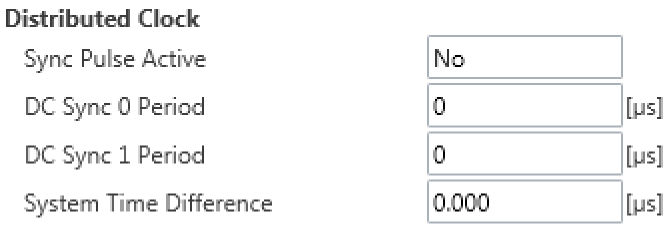


This dialog is only visible in the *'Expert mode'*!

- Common Error Counter
  - Processing Unit Error Counter: Number of received frames by the slave station, which are no EtherCAT frames.
  - PDI Error Counter: Number of PDI access errors (**P**rocess **D**ata **I**nterface). These are physical errors, which were detected by the PDI at the EtherCAT bus.
  - With [Clear Error Counters] the error counters can be reset.
- Port 0...3
  - Invalid Frame Counter: Number of invalid frames from *Port* y (access at register 0x300+y*2)
  - RX Error Counter: Number of RX errors from *Port* y (access at register 0x300+y*2+8bit)
  - Lost Link Counter: Number of lost connections from *Port* y (access at register 0x310+y)
  - Forwarded RX Error Counter: Number of forwarded RX errors from *Port* y (access at register 0x380+y)

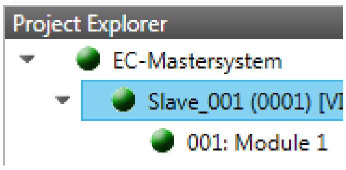
9.10.8.6 DC Diagnosis (Expert mode)

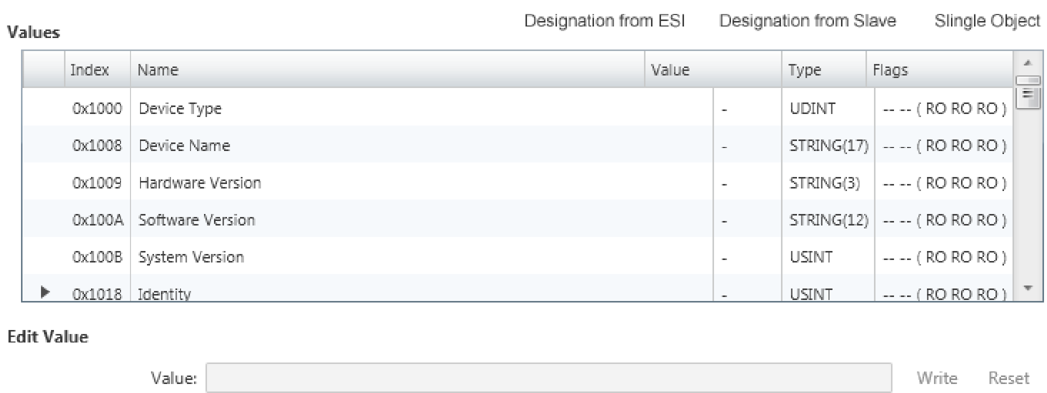




This dialog is only visible in the *'Expert mode'*! Here status information for the distributed clock of your slave station is shown. More may be found in the manual of the slave station.

9.10.8.7 CoE Object Dictionary





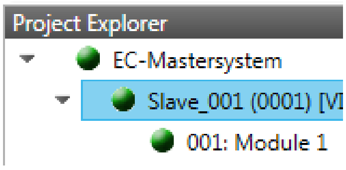
Here you will have read and write access to the CoE Object Dictionary of the slave station. This can be changed if your slave station permits. It is indicated by the *'Flags'* of each object, if write access is permitted. Information about the structure of the Object Dictionary can be found in the manual of your slave station.

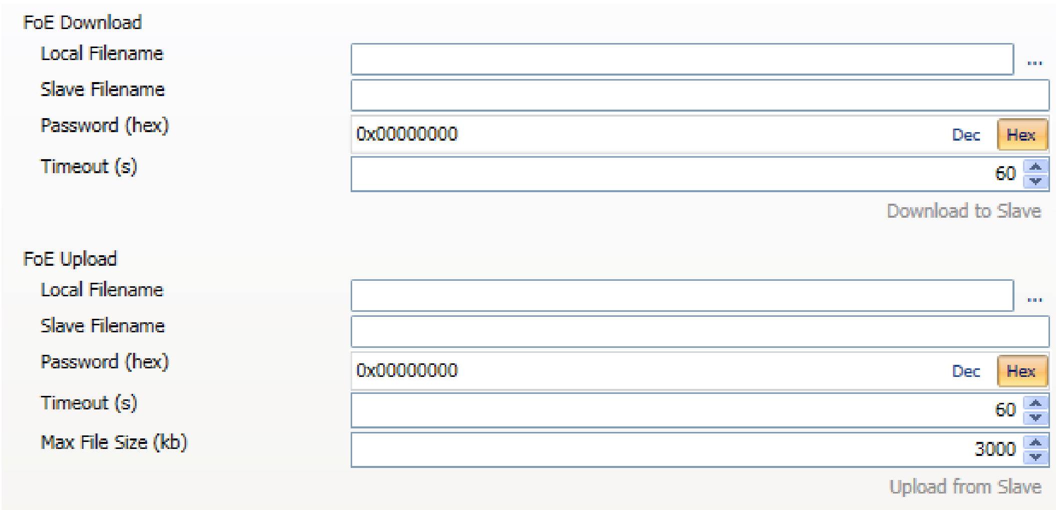
If a write access to an object in the diagnosis mode is performed, and the written value does not reflect to the default value of the object, so this command is automatically added to the 'Init commands'. *Chap. 9.10.5.8 'Init Commands (Expert mode)' page 255*

This dialog is only visible in the *'Expert mode'*:

- Designation from ESI
  – By selecting this function the designations are loaded from the ESI file.
- Designation from slave
  – By selecting this function the designations are loaded from the slave station.
- Single Object
  – With this function you have read and write access to a single object in the object dictionary by specifying index and subindex.

9.10.8.8 FoE Download/Upload





- With this function you have the possibility to transfer files between PC and slave station (if this is supported by the device). If the slave station is in state *Bootstrap*, a firmware update of the slave station can be established via *'FoE Download'*. Here you have to enter the file name without extension. *Chap. 9.10.11 'Firmware update - System SLIO IM 053-1EC0x' page 276*
  – Local filename: Name of the file at the PC.
  – Slave filename: Name of the file at the slave station.
  – Password: Password to access the slave station.
  – Timeout: Maximum time for data transfer.
  – Max. file size: Maximum size of the file, which is to be transferred from the slave station to the PC.

9.10.9 Grouping logic

9.10.9.1 Overview

Slave types

With EtherCAT, the following slave types are distinguished:

- MII slave - MII corresponds to **M**edia **I**ndependant **I**nterface. An MII slave has an EtherCAT interface to connect to EtherCAT for integration into a system bus (backplane bus) for connecting peripheral modules. The MII slave receives data via EtherCAT and passes them through its backplane to the according peripheral module. Conversely, it reads the input data and passes it via EtherCAT. The System SLIO 053-1EC0x e.g. is a MII-Slave.
- E-Bus slave - In an E-Bus slave the EtherCAT protocol is used for communication on the backplane bus For this reason, the attached peripheral modules are also shown as a slave station in the *SPEED7 EtherCAT Manager*

Possibilities

The EtherCAT Manager supports the following ways to group the individual slave stations. Each group may consist of 1 .. n slave stations. Group nesting is not supported:

Chap. 9.10.9.2 'Create group with pinned process data offset' page 272

'Hot Connect group with Dynamic Position in Topology' page 274

'Hot Connect group with Fixed Position in Topology' page 274

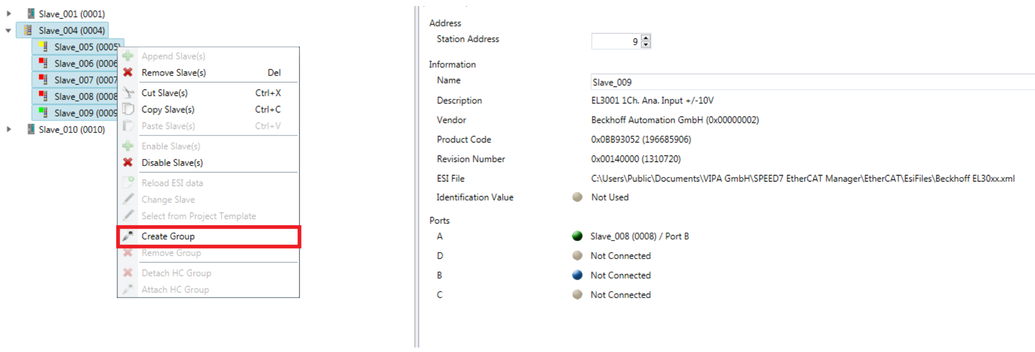
'Hot Connect group with Pinned or Dynamic Process Data Offset' page 274

> *Please consider that Hot Connect groups are not possible with E-Bus slaves!*

Create Group

1. Click in the Toolbar of the *SPEED7 EtherCAT Manager* at [Configuration].
2. Click in the *Project Explorer* at the slave station and select *'Context menu → Create Group'*.

   ⇨ The dialog *'Create Group'* opens. Here always the 1. slave station is selected. You can either select more slave stations or depending on the group type selection, the necessary save stations are automatically selected.

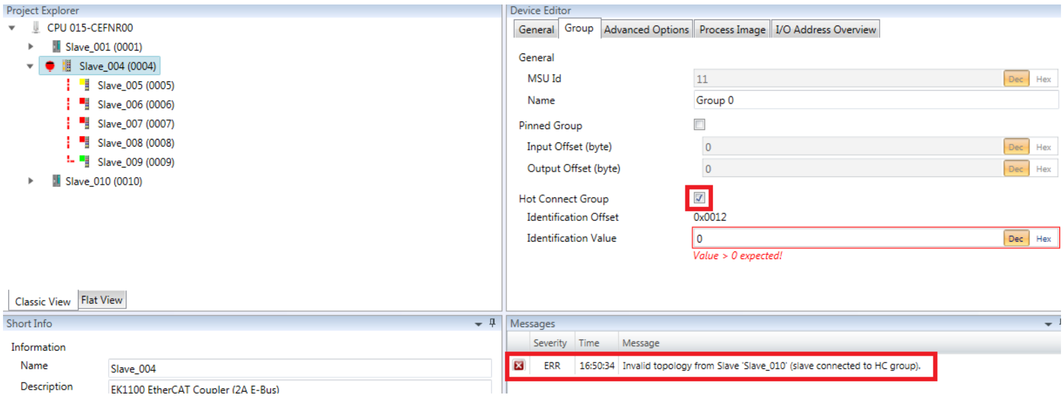


With the *'Create Group'* functionality you have two different functions:

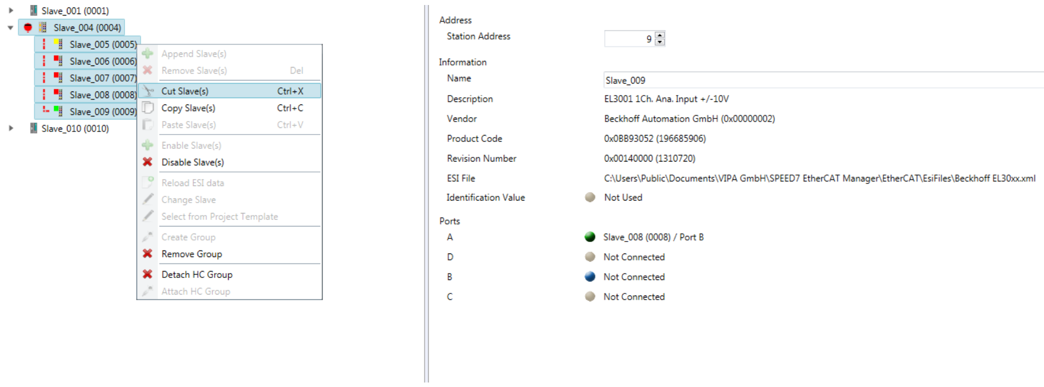
- You can create a new group if the selected slave station is not yet part of a group.
- If the selected slave station is already part of a group, the current group is divided into two sub-groups from the selected slave station.

Edit Group

After creating a group, the *'Device Editor'* of the slave station is extended with the register *'Group'*. Here you can adjust the group properties accordingly.

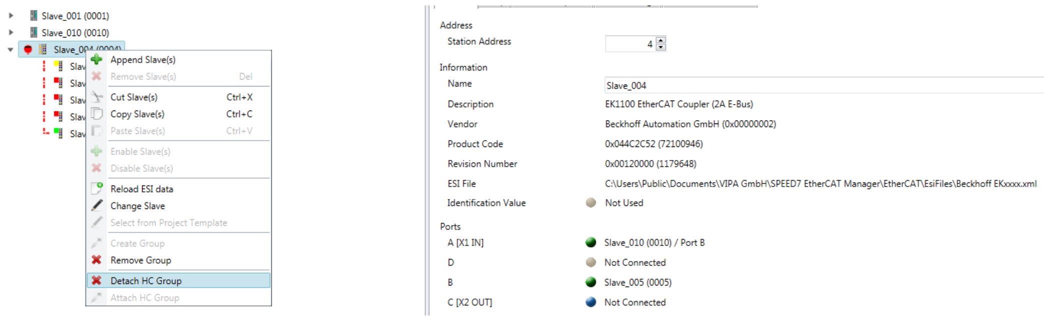


The new group can be selected by selecting this group via *'Cut Slave(s)'* be changed.



Detach HC Group

If you want to connect this group to an other slave station on the network, you can detach the current connection by *'Detach HC Group'*.



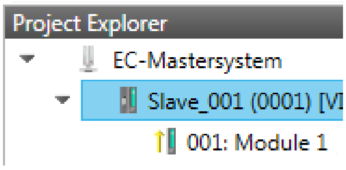
Remove Group

➔ To remove a group click in the *SPEED7 EtherCAT Manager* at a slave station and select *'Context menu ➔ Remove Group'*.

⇨ The group is removed. Depending on the group, the previously grouped slave stations are reintegrated into the topology or remain at the current position.

9.10.9.2 Create group with pinned process data offset

Procedure

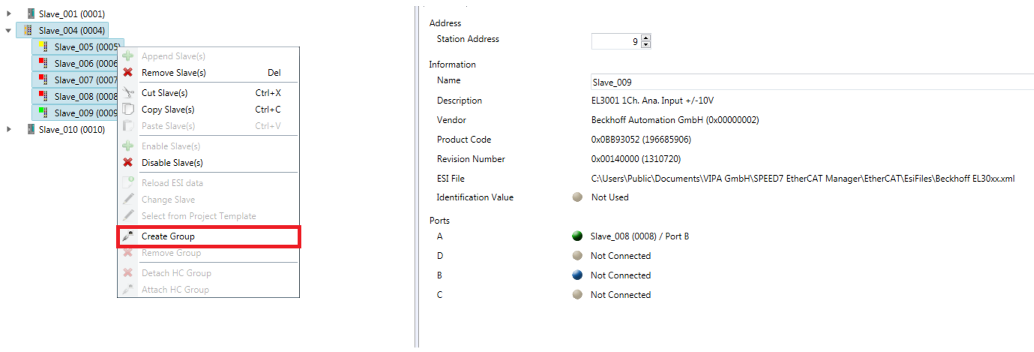


This group may start at any slave station and either end at himself, at a following slave station, at a following group or at the last slave station. The group functionality is possible with each slave type. The slave stations of this group are pinned at a fix position in the topology.

Create Group

1. Click in the Toolbar of the *SPEED7 EtherCAT Manager* at [Configuration].
2. Click in the *Project Explorer* at the slave station and select *'Context menu ➔ Create Group'*.

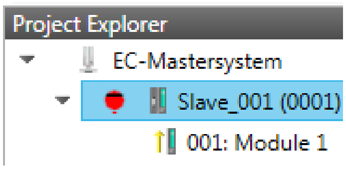
   ⇨ The dialog *'Create Group'* opens. Here always the 1. slave station is selected. You can either select more slave stations or depending on the group type selection, the necessary save stations are automatically selected.

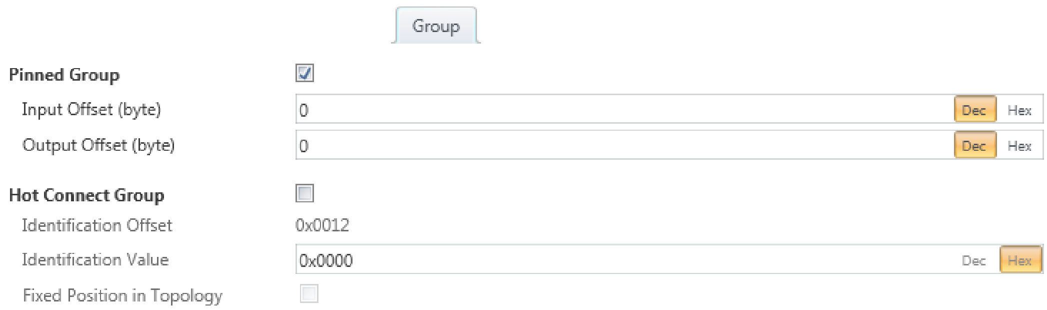


Pinned group

1. Choose from *'Select the slaves'* the slave stations, which you want to include in the *'Pinned group'*.

   ⇨ The dialog is closed, the slave station is marked as group in the *'Project Explorer'* and a tab "Group" is created in the *'Device Editor'*.



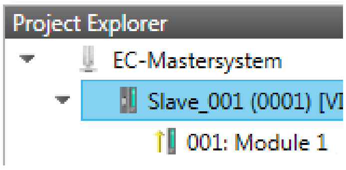


2. Enable the option *'Pinned Group'*.
3. Enable the option *'Input Offset = Output Offset'* if the input and output addresses are identical.

   ⇨ The group is now defined as *Pinned Group*.

9.10.9.3 Create Hot Connect group

9.10.9.3.1 Proceeding



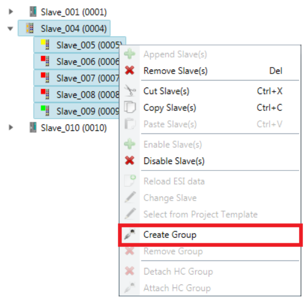
In a *Hot Connect group* several slave stations can be located, which must only optional be available at the EtherCAT bus. So you have the possibility to take or add pre-configured sections from the traffic before starting the system or during the operation. This can be done by disconnecting/connecting the communication path or enabling/disabling the participant.

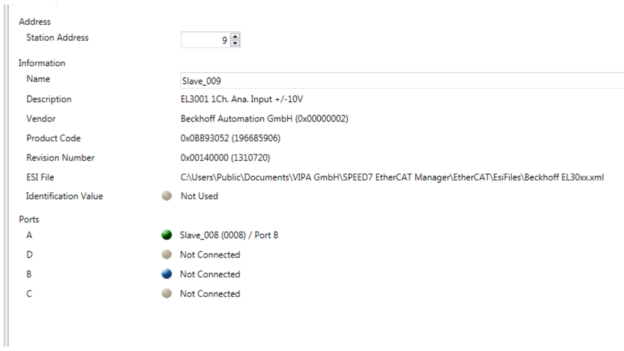
> *Please consider that the first slave station after the EtherCAT master must not be optional!*
>
> *To use the hot connect function with E-Bus slave stations, the E-Bus head station* *and* *the connected slave stations must be in the same group!* *Chap. 9.10.9 'Grouping logic' page 270*

Create Group

1. Click in the Toolbar of the *SPEED7 EtherCAT Manager* at [Configuration].
2. Click in the *Project Explorer* at the slave station and select *'Context menu ➔ Create Group'*.

   ⇨ The dialog *'Create Group'* opens. Here always the 1. slave station is selected. You can either select more slave stations or depending on the group type selection, the necessary save stations are automatically selected.

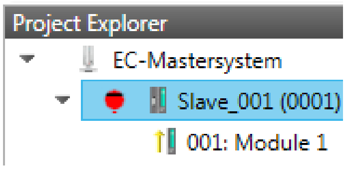


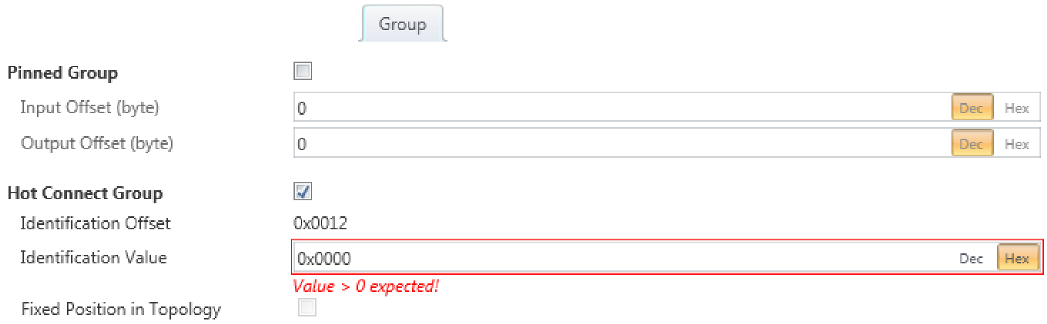


Hot connect group

1. Choose from *'Select the slaves'* the slave stations, which you want to include in the *'Hot connect group'*.

   ⇨ The dialog is closed, the slave station is marked as group in the *'Project Explorer'* and a tab "Group" is created in the *'Device Editor'*.





2. Enable the option *'Hot connect group'*.
3. Enter an *'Identification value'*: This is the *Station-Alias-Address*, which you have to assign before to the slave station in the *'Diagnosis'* Mode. *Chap. 9.10.8.4 'EEPROM (Expert mode)' page 267*

   Please regard that the slave station takes the new address after a power-cycle.
4. For a fix position of the group in the topology the option *'Pinned group'* can be enabled.

9.10.9.3.2 Combination possibilities

Hot Connect group with Dynamic Position in Topology

The group must start with a MII slave. Here, all slave stations below the selected are automatically added to the group. This group ends at himself, at a following slave station, at a following group or at the last slave station.

Hot Connect group with Fixed Position in Topology

The group is fix coupled to a predecessor slave station and its port. You always have the possibility to change the link to the previous slave station via the dialog box. If the group is removed, the slave stations remain in place.

A Hot Connect group with Fixed Position in Topology cannot be removed, if the slave stations before are a part of another Hot Connect group with Fixed Position in Topology!

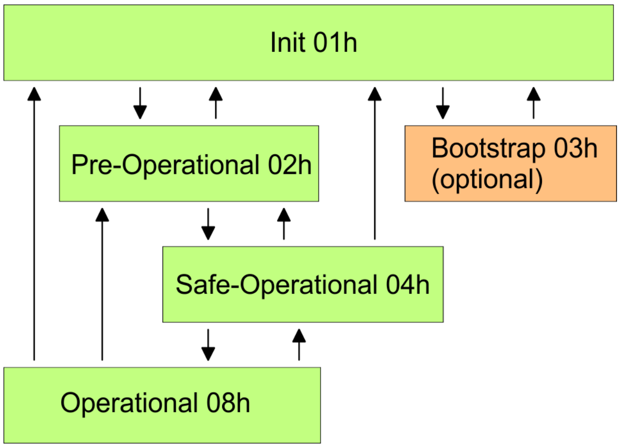
Hot Connect group with Pinned or Dynamic Process Data Offset

This group does not depend on slave station or port. The group has no predecessor slave station and is moved to the end of the tree when created. When the group is removed it is searched for a suited free port starting from the end of the main tree. If there is no suited slave station available, the group will be rejected! Due to the system the group has no predecessor slave station, the connection cannot be changed via the dialog box.

9.10.10 EtherCAT State Machine

States

In each EtherCAT communication device a *state machine* is implemented. For each state there is defined which communication service is active via EtherCAT. The state machine of the slave station is controlled by the state machine of the EtherCAT master.



Init - 01h

After power-on the EtherCAT members are in state *Init*. There is neither mailbox nor process data communication possible. The EtherCAT master initializes the SyncManager channels 0 and 1 for the mailbox communication.

Pre-Operational (Pre-Op) - 02h

The EtherCAT master initializes the SyncManager channels for process data (starting with SyncManager channel 2), the FMMU channels and the PDO mapping respectively the SyncManager PDO assignment. Further in this state the settings for process data transfer and the module-specific parameters, which deviate from the default values are transferred. During the transition from *Init* to *Pre-Op* the EtherCAT slave checks whether the mailbox was correctly initialized. In the state *Pre-Op* mailbox communication and Ethernet over EtherCAT (EoE) are possible but the process data communication is blocked.

Safe-Operational (Safe-Op) - 04h

In *Safe-Op* the input data are cyclically updated but the outputs are de-activated. With the transition from *Pre-Op* to *Safe-Op* the EtherCAT slave checks if the SyncManager channels for process data communication are correct. Before it acknowledges the state change, the EtherCAT slave copies current input data to the corresponding DP RAM areas of the EtherCAT slave controller. In the state *Safe-Op* mailbox and process data communication is possible.

Operational (Op) - 08h

In the state *Op* the input data are cyclically updated and the EtherCAT master sends output data to the EtherCAT slave. The EtherCAT slave copies the output data of the master to its outputs and return input data to the EtherCAT master. In this state process data and mailbox communication is possible.

Bootstrap - option (Boot) - 03h

In state *Boot* the firmware of an EtherCAT slave may be updated via the EtherCAT master. This state may only be reached via Init. In the state *Boot* is mailbox communication via the protocol File-Access over EtherCAT (FoE) possible. Other mailbox and process data communications are de-activated.

9.10.11 Firmware update - System SLIO IM 053-1EC0x

Current firmware at www.yaskawa.eu.com

The latest firmware versions are to be found in the *'Download Center'* at www.yaskawa.eu.com. Load the Px000xxx.pkg file.

CAUTION!

– When installing a new firmware you have to be extremely careful. Under certain circumstances you may destroy the slave station, for example if the voltage supply is interrupted during transfer or if the firmware file is defective. In this case, please call the Yaskawa hotline!

– An update can only be applied if the firmware version to be overwritten differs from the update version.

Precondition

- There is an Ethernet respectively remote connection between the PC and the EtherCAT slave station, where a firmware update is to be established.

Proceeding

Below the proceeding is shown by the example of the Yaskawa System SLIO slave station. For other devices, please follow the procedures described in the according manual.

1. Open if not already done the *SPEED7 EtherCAT Manager*.
2. Click in the *'Project Explorer'* at *'EC-Mastersystem'*.
3. Select in *'Device Editor > Master'* at *'Network Adapter'* your network card and enter at *'IP Address'* the IP address of the PG/OP channel of the CPU and click at [Select].
4. Click in the Toolbar at [Diagnosis Mode].

   ⇨ An online connection to your EtherCAT system is established via the preset communication channel and the current project configuration in the *'Project explorer'*.
5. Click in the *'Project explorer'* at the master.
6. Select in the register *'General'* at *'State Machine'* the state *'Init'*. Wait, until all slave station response the state *'Init'*.
7. Click in the *'Project explorer'* at the slave, where the firmware update is to be established.
8. Select in the register *'General'* at *'State Machine'* the state *'Bootstrap'*.
9. Enter in the register *'FoE '* at *'FoE Download'* as follows:

   - Filename: Px000xxx

   - Password (hex): 0x0000000

   - Timeout (ms): 60000

   - Max File Size (kb): 3000
10. Click at [Download].

    ⇨ A dialog for file selection opens.
11. Select the file. The transfer starts with [OK].

    ⇨ There will be a progress bar displayed, which informs you about the transfer state.
12. After successful download bring your slave in the *'Init'* state.

    ⇨ With this operation the firmware file is taken.

10 Deployment PG/OP communication - PROFINET I-Device

- *With firmware version V2.4.0, there is a PROFINET I-Device available via the Ethernet PG/OP channel.*
- *As soon as you use the PROFINET functionality via the Ethernet PG/OP channel, this affects the performance and response time of your system and due to the system the cycle time of the OB 1 is extended by 2ms.*

10.1 Basics PROFINET

General

- PROFINET is an open Industrial Ethernet Standard from PROFIBUS & PROFINET International (PI) for automation.
- PROFINET is standardized in the IEC 61158.
- PROFINET uses TCP/IP and IT standards and supplements the PROFIBUS technology for applications, where fast data communication with industrial IT functions is demanded.

Properties of PROFINET

PROFINET of IEC 61158 has the following properties:

- Full-duplex transfer with 100MBit/s via copper respectively fibre optics.
- Switched Ethernet
- Auto negotiation (negotiates the transfer parameters)
- Auto crossover (transmission and receipt lines are crossed automatically if necessary)
- Wireless communication via WLAN
- UDP/IP is used as overlaid protocol. UDP means **U**ser **D**atagram **P**rotocol and contains the unprotected connectionless broadcast communication within IP.

GSDML file

- To configure a device I/O connection in your own configuration tool, you've got all the information about your PROFINET components in form of a GSDML file. This file may be found in the *'Download Center'* of www.yaskawa.eu.com.
- Please install the GSDML file in your configuration tool.
- More information about installing the GSDML file may be found at the manual of the according engineering tool.
- Structure and content of the GSDML file are defined by IEC 61158.

The following GSDML files are required for configuring the integrated PROFINET I-Device in the Siemens SIMATIC Manager:

- GSDML for I-Device
- GSDML for I-Device at IO controller

Addressing

In contrast to the PROFIBUS address, in PROFINET each device may be definitely identified with its PROFINET interface:

- Device name
- IP address respectively MAC address

Transfer medium

PROFINET is compatible to Ethernet in accordance with the IEEE standards. The connection of the PROFINET IO field devices is exclusively established via switches as network components. This is made either as star via multi-port switches or as line by means of switches, integrated to the field devices.

10.2 PROFINET installation guidelines

Generals to data security

The topic of data security and access protection have become increasingly important in the industrial environment. The increased networking of entire industrial systems to the network levels within the company together with the functions of remote maintenance have all served to increase vulnerability. Threats can arise from internal manipulation like technical errors, operator and program errors respectively from external manipulation like software viruses and worms, trojans and password phishing.

Precautions

The most important precautions to prevent manipulation and loss of data security in the industrial environment are:

- Encrypting the data traffic by means of certificates.
- Filtering and inspection of the traffic by means of VPN - "Virtual Private Networks".
- Identification of the nodes by "Authentication" via save channels.
- Segmenting in protected automation cells, so that only devices in the same group can exchange data.

Guidelines for information security

With the "VDI/VDE 2182 sheet 1", Information Security in the Industrial Automation - General procedural model, VDI guidelines, the VDI/VDE society for measuring and automation engineering has published a guide for implementing a security architecture in the industrial environment. The guideline can be found at www.vdi.de PROFIBUS & PROFINET International (PI) can support you in setting up security standards by means of the "PROFINET Security Guideline". More concerning this can be found at the corresponding web site e.g. www.profibus.com

Industrial Ethernet

- Due to the open standard of PROFINET standard Ethernet components may be used.
- For industrial environment and due to the high transfer rate of 100MBit/s your PROFINET system should consist of Industrial Ethernet components.
- All the devices interconnected by switches are located in one and the same network. All the devices in a network can communicate directly with each other.
- A network is physically limited by a router.
- If devices need to communicate beyond the limits of a network, you have to configure the router so that it allows this communication to take place.

Topology

- Linear
  - With the linear structure all the communication devices are connected via a linear bus topology.
  - Here the linear bus topology is realized with switches that are already integrated into the PROFINET device.
  - If a communication member fails, communication across the failed member is no longer possible.
- Star
  - If you connect communication devices to a switch with more than 2 PROFINET interfaces, you automatically create a star network topology.
  - If an individual PROFINET device fails, this does not automatically lead to failure of the entire network, in contrast to other structures.
  - It is only if a switch fails that part of the communication network will fail as well.
- Ring
  - In order to increase the availability of a network the both open ends of a linear bus topology may be connected by a switch.
  - By configuring the switch as redundancy manager on a break in the network it ensures that the data is redirected over an intact network connection.
- Tree
  - If you interconnect several star structures, you obtain a tree network topology.

10.3 Deployment as PROFINET I-Device

10.3.1 Steps of configuration

Functionality

> *Please regard that the PROFINET I-Device supports only the PROFINET functions, which are described in this manual, even if the Siemens CPU, which is used for configuration, offers further functionalities!*

The *I-Device* (Intelligent IO device) functionality of a CPU allows data to be exchanged with an IO controller, which are preprocessed by the CPU. In this case, the I-Device is connected as an IO device to a higher-level IO controller. The process values, recorded in central or decentralized periphery, can be preprocessed via a user program and made available to the higher-level PROFINET IO controller by means of PROFINET.

- The configuration of the integrated PROFINET IO controller of the CPU as an I-Device is made via a virtual PROFINET devices, which is to be installed by means of a product specific GSDML in the hardware catalog.
- The communication takes place via input/output areas, which are defined in the I-Device.
- The size of the areas for input and output data is max. 768byte.
- The I-Device is made available to a deterministic PROFINET IO system via a PROFINET IO interface and thus supports the real-time communication *Real-Time* .
- The I-Device functionality meets the requirements of the RT class I (A) and corresponds to the PROFINET specification version V 2.3.
- In order for the higher-level IO controller to communicate with the I-Device, the following must be observed:
  – I-Device and higher-level IO controllers must be configured in different networks. Their IP addresses must be in the same IP circuit.
  – The device name of the PROFINET controller of the I-Device must match the device name of the I-Device at the higher-level IO controller.

> *The PROFINET IO controller supports a maximum IO block size of 512 bytes (consistent).*

Configuration

The configuration of the PROFINET IO controller as I-Device should be done by the following procedure:

1. Installation of the GSDML files
2. Configuration as I-Device
3. Configuration in the higher-level IO controller

Transfer I-Device from *SPEED7 Studio*

If the configuration of the *I-Device* from the *SPEED7 Studio* is to be applied to the IO controller of a third-party system, then you must export from the *SPEED7 Studio* the according GSDLM file and import it into the IO controller of the third-party system.

1. Start the *SPEED7 Studio* with your PROFINET project.
2. Click at the CPU in *'Devices and networking'* and select *'Context menu → Create GSDML file'*. Specify an *'Export path'* and an unique *'Device name'*.
   ⇨ The GSDML file is created and exported. Import this GSDML file into your third-party system.

10.3.2 Installing the GSDML file

The following GSDML files are required for configuring the integrated PROFINET I-Device in the Siemens SIMATIC Manager:

- GSDML for I-Device
- GSDML for I-Device at IO controller

Proceeding

1. You can find the GSDML files in the *‘Download Center’* of www.yaskawa.eu.com. Load the file and unzip it on your PC.
2. Start the Siemens SIMATIC Manager and install via *‘Options ➔ Install new GSD file’* both GSD files.

   ⇨ After the installation you can find the following virtual devices in the *hardware catalog* at *‘PROFINET IO ➔ Additional field devices ➔ ... ➔ ... SLIO System’*:

   - PN I-Device for System SLIO CPU
     – This allows you to configure the Input/output areas in the I-Device of the System SLIO CPU.
   - PN I-Device for higher-level CPU
     – This allows you to connect the I-Device to the higher-level IO controller.

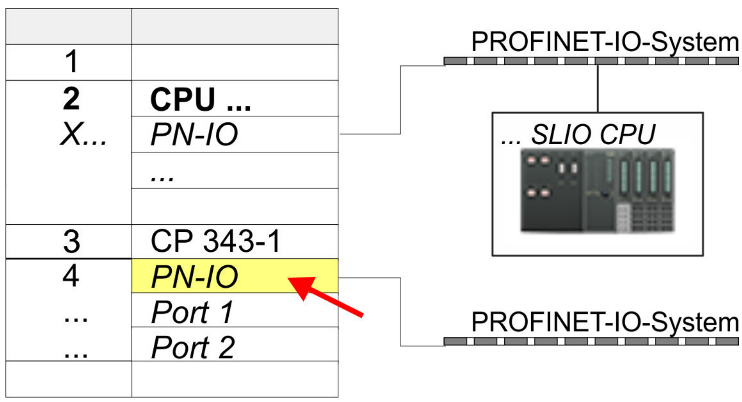
10.3.3 Configuration as I-Device

It is assumed that a hardware configuration of the CPU exists. ↪ *Chap. 4.5 ‘Hardware configuration - CPU’ page 75* ↪ *Chap. 12.2 ‘Siemens SIMATIC Manager - Hardware configuration - CPU’ page 293*

1. Place for the Ethernet PG/OP channel at slot 4 the Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).

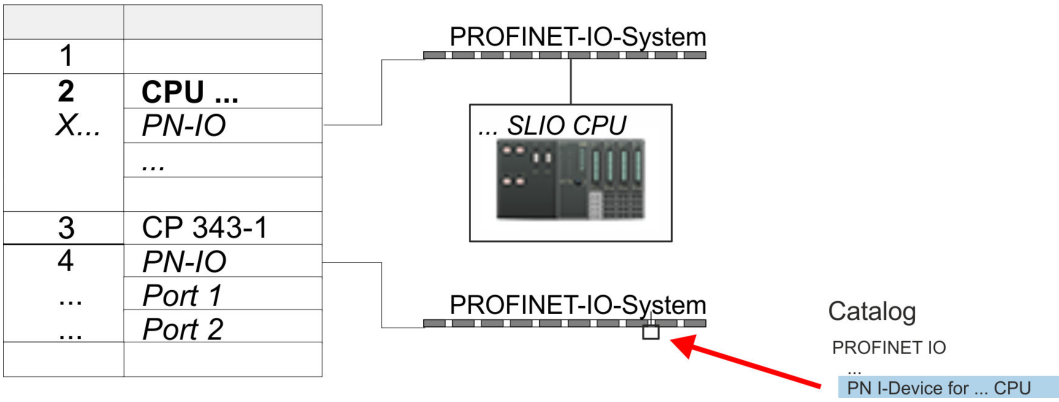
> **CAUTION!**
> Please configure the diagnostic addresses of the CP343-1EX30 for *‘PN-IO’*, *‘Port1’* and *‘Port2’* so that no overlaps occur in the periphery input area. Otherwise your CPU can not start-up and you receive the diagnostic entry 0xE904. These addresses overlaps are not recognized by the Siemens SIMATIC Manager.

2. Click at the sub module *‘PN-IO’* of the CP 343-1.
3. Select *‘Context menu ➔ Insert PROFINET IO System’*.



4. Create with [New] a new subnet and assign valid IP address data and a subnet. The IP address data are not accepted without subnet assignment.

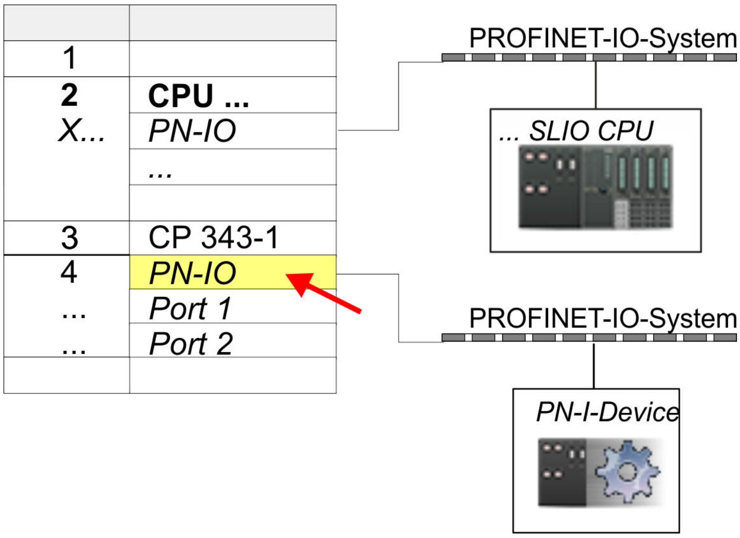
5. For the project engineering of PROFINET I-Device you have to search the virtual device *'PN I-Device for ... CPU'* in the hardware catalog at *PROFINET-IO* and drag&drop it in the PROFINET subnet.



6. Click at the sub module *'PN-IO'* of the CP 343-1.

   Open the properties dialog of the CP 343-1 by a double-click at *'PN-IO'* and assign the name for the I-Device.

> *Write down the Name. This name must also be specified as the 'device name' of the I-Device for the higher-level IO controller.*



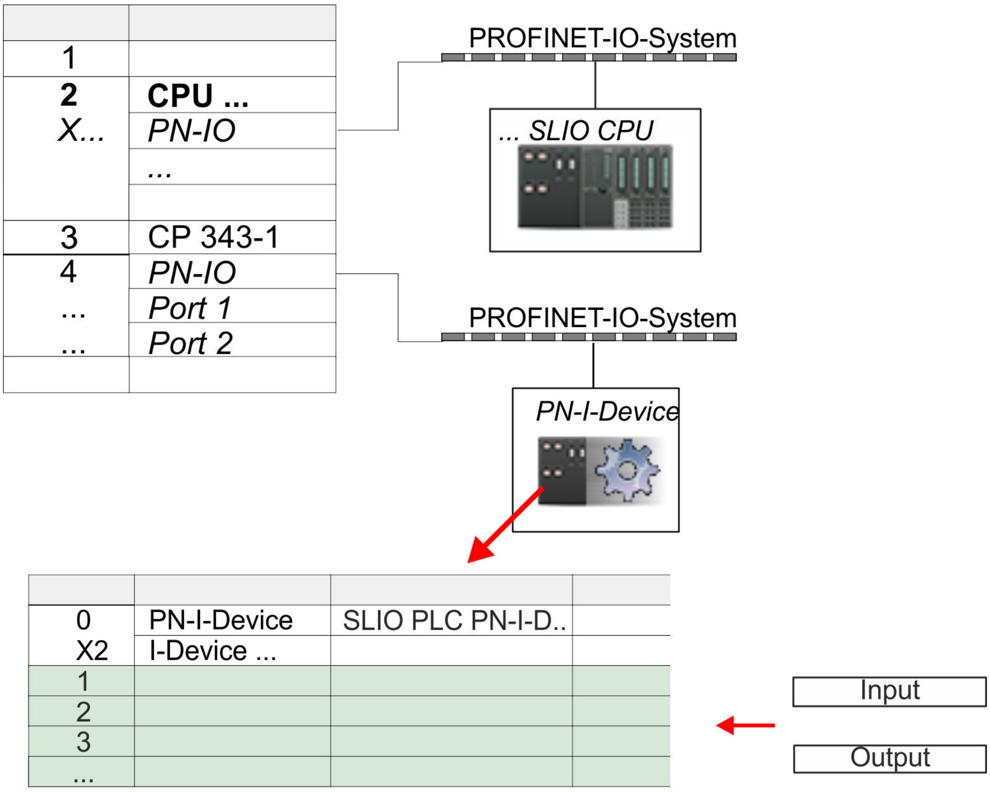
7. For *'PN-IO'* at *'slot'* *'X...'* assign an IP address via the properties dialog.

8. Create the transfer areas by dragging them to the *'slots'* as I/O areas from the hardware catalog. There must be no gaps in the slots. To create the transfer areas, the following input and output areas are available that can be assigned to the virtual I-Device:

- Input: 1, 8, 16, 32, 64, 128, 256, 512 byte
- Output: 1, 8, 16, 32, 64, 128, 256, 512 byte

The data direction for *Input* or *Output* refers to the view of the I-Device.

- *Input* areas define data that are sent from the higher-level IO controller to the I-Device and which are mapped to the input address area of the CPU.
- *Output* areas define data that are sent to the higher-level IO controller and which are to stored in the output address area of the CPU.

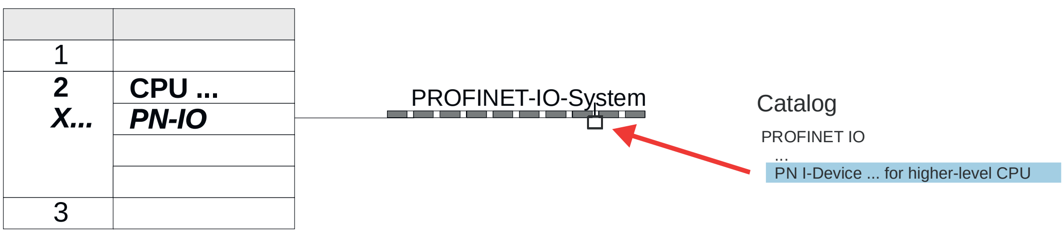


9. Save and transfer your project to the CPU.

10.3.4 Configuration in the higher-level IO controller

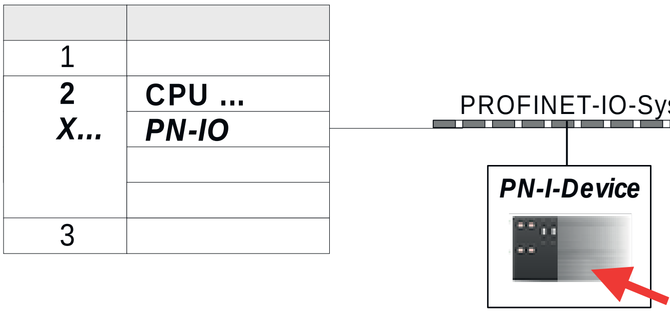
It is assumed that a CPU is configured with IP address with the higher-level IO controller. The IP address must be in the same IP circuit as the IP address of the I-Device.

1. Open the project of the CPU with the higher-level IO controller.
2. For the project engineering of System SLIO I-Device in the higher-level IO controller you have to search the device *'PN I-Device 015-CEFNR00 for high-level CPU'* in the hardware catalog at *PROFINET-IO* and drag&drop it to the PROFINET subnet.



3. Open the properties dialog by double-clicking System SLIO *'PN-I-Device'* and enter at *'Device name'* the previously noted name of the I-Device.

> *The configured name must match the PROFINET IO controller name 'PN-IO' of the I-Device CPU, which you have written down before!* *Chap. 10.3.3 'Configuration as I-Device' page 280*



4. Configure an input area of the same size for each output area of the I-Device in the IO controller and vice versa. Here also no gaps may arise. In particular, make sure that the order of the transfer areas matches that of the I-Device configuration. The following transfer units are available:
   - Input: 1, 8, 16, 32, 64, 128, 256, 512 byte per slot
   - Output: 1, 8, 16, 32, 64, 128, 256, 512 byte per slot
5. Save and transfer your project to the CPU.
   ⇨ Your System SLIO PROFINET CPU is now connected as an PROFINET I-Device to the higher-level PROFINET IO controller.

> ***I-Device with S7 routing***
> *S7 routing is not possible with the procedure above. S7 routing is only possible if the I-Device and the higher-level I/O controller are configured in the same network. The device names must not be identical. By using identical names and extending the name of the I-Device with "-x", this is detected internally and used appropriately for S7 routing.*

10.3.5 Error behavior and interrupts

Error behavior

The system shows the following error behavior ...

- ... at gaps in the *'slot'* configuration:
  – If the configuration of the I-Device contains gaps in the *'slot'* configuration (i.e. there are free *'slots'* before used *'slots'*), the configuration is rejected and 0xEA64 is returned as a configuration error in the diagnostic buffer.
  – If the configuration of the higher-level IO controller contains gaps in the *'slot'* configuration (i.e. there are free *'slots'* before used *'slots'*), the connection is rejected with the PN IO Status *ErrorCode1* = 0x40 and *ErrorCode2* = 0x04 (AR_OUT_OF_RESOURCE).
- ... at modules, which differ from the configured:
  – A *ModuleDiffBlock* is generated and the wrong modules are not served.

- ... if the number of configured modules in the IO controller is greater than the number of configured modules in the I-Device:
  – The IO controller receives a *ModuleDiffBlock* with ModuleStatus "NoModule" for modules that are not configured in the I-Device. The I-Device sets the status of the non-configured modules to "bad".
- ... if the number of configured modules in the I-Device is greater than the number of configured modules in the IO controller:
  – The IO controller does not receive an error because the additional modules are unknown.

Starting position	**IO controller in RUN, I-Device in RUN**
Event	I-Device CPU goes to STOP
Reaction	■ An OB 85 is called in the IO controller for each input and output transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. *82* ■ An OB 122 is triggered in the IO controller for each peripheral direct access to an input or output transfer area.

Starting position	**IO controller in RUN, I-Device in RUN**
Event	IO controller goes to STOP
Reaction	■ An OB 85 is called in the I-Device for each input transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. *82* ■ In the I-Device, an OB 122 is triggered for each peripheral direct access to an input transfer area.
	Note: Output transfer areas can still be accessed!

Starting position	**IO controller in RUN, I-Device in RUN**
Event	Station failure I-Device, e.g. by bus interruption
Condition	I-Device must remain operational without a bus connection, i.e. the power supply must further exist.
Reaction	■ An OB 86 (station failure) is called up in the IO controller. ■ An OB 85 is called in the IO controller for each input and output transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. *82* ■ An OB 122 is triggered in the IO controller for each peripheral direct access to an input or output transfer area. ■ An OB 86 (station failure) is called up in the I-device. ■ An OB 85 is called in the IO controller for each input and output transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. *82* ■ In the I-device, an OB 122 is triggered for each peripheral direct access to an input or output transfer area.

Starting position	**IO controller in RUN, I-Device in RUN**
Event	Station recovery
Reaction	■ An OB 86 (recovery) is called in the IO controller. ■ An OB 85 is called in the IO controller until the OB 86 has been called, for each input and output transfer area, which is in the process diagram, if messages of process image transfer errors are parametrized. ☞ *82* ■ An OB 122 is triggered in the IO controller until the OB 86 is called, for each peripheral direct access to an input or output transfer area. ■ An OB 86 (return) is called in the I-Device. ■ An OB 83 (sub module recovery) is called for each input transfer area in the I-Device. ■ In the I-device, an OB 85 is called for each input transfer area, which is in the process image, if messages of process image transfer errors are parametrized and the corresponding OB 83 has not yet been called. ☞ *82* ■ An OB 122 is triggered in the I-Device for each peripheral direct access to an input transfer area, until the corresponding OB 83 has been called.

Starting position	**Controller in RUN, I-Device in STOP**
Event	I-Device starts
Reaction	■ The OB 100 (start-up) is called in the I-Device. ■ The OB 83 (Return-of-Submodule) for input sub modules of the transfer areas to the higher-level IO controller is called in the I-Device. ■ An OB 85 is called in the I-device for each input transfer area, which is located in the process image, if messages of process image transfer errors are parametrized. ☞ *82* ■ In the I-Device, an OB 122 is triggered for each peripheral direct access to an input transfer area. ■ OB 83 (Return-of-Submodule) for input and output sub modules of the transfer areas to the I-Device is called in the IO controller. ■ An OB 85 is called in the IO controller for each input and output transfer area, which is located in the process diagram, if messages of process image transfer errors are parametrized and the corresponding OB 83 has not yet been called. ☞ *82* ■ In the IO controller, an OB 122 is triggered for each peripheral direct access to an input or output transfer area until the corresponding OB 83 has been called.

Starting position	**IO controller is in STOP, I-Device in RUN**
Event	IO controller starts
Reaction	■ The OB 83 (Return-of-Submodule) for input sub modules of the transfer areas to the higher-level IO controller is called in the I-Device. ■ An OB 85 is called for each transfer area, which is located in the process image, in the I-device if messages of process image transfer errors are parametrized and the corresponding OB 83 has not yet been called. ☞ *82* ■ An OB 122 is triggered in the I-Device for each peripheral direct access to an input transfer area, until the corresponding OB 83 has been called. ■ The OB 100 (startup) is called in the IO controller.

11 *Option:* Deployment - Isochronous

Isochronous only via SPEED7 Studio

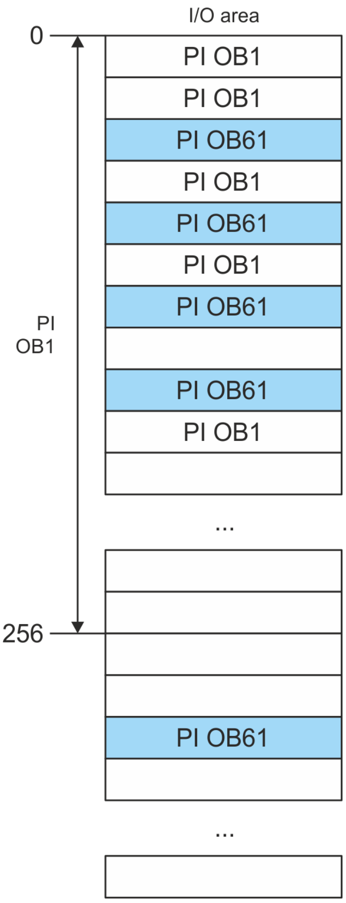
Please note that the use of Isochronous together with motion functions is only possible via the SPEED7 Studio.

Activate additional functions by means of VSC in the CPU

In order to use the additional functions, you must activate them by means of a Yaskawa VSC storage card. By inserting the VSC storage card and then an overall reset, the corresponding additional functions are activated.

'Overview' page 121

11.1 Process Image



The CPU determines by reading the input values the current state of a system and achieves by selectively controlling of output values the required system behavior (functionality). If the operand areas of the process data are accessed by the user program, so an area of the system memory is accessed. This memory area is called *Process image* (PI). Direct access to the process image has the advantage that for the duration of cyclic program processing the CPU has a consistent image of process signals. The update of the process image can be interrupted by an organization block with higher priority. This is only possible on the consistency points, which are defined by module limits. The OB 61 has the highest priority.

Process images

The CPU has an I/O data area to store process images. There are the following PIs:

PI	Remark
PI OB 1	■ Triggering: Internal ■ Assignment: OB 1 ■ Start address: 0 ■ End address configurable ■ Gaps: Usage by PI OB 61 permitted
PI OB 61	■ Triggering: Internal ■ Assignment: OB 61 ■ For System SLIO and EtherCAT bus ■ Any address can be mapped

Here is valid:

- Each address can be assigned to only one PI.
- The data of a PI are consistent for the duration of the OB for which the process image has been configured.
- The input data of the configured process images for the corresponding OB are read before starting the OB and the output data are written after completion of the OB.
- The data of a PI can be accessed from any OB.

> – *By assigning the address ranges of the corresponding modules and EtherCAT slaves (S7 addresses) to the process image of the OB 61, these are synchronized in clock cycles. The other addresses may be assigned to the PI OB 1.*
>
> – *In principle, isochronous does not support analog modules on the System SLIO bus. You can add analog modules to the process image of the OB 61. Their input and output data are not processed isochronously. If you use modules on the System SLIO bus system, which do not support isochronous, you will get the diagnostic message 0xEB05 (bus configuration for isochronous process diagram not suitable). The module's error LED blinks.*

11.2 Isochronous

Isochronous and Sync-signal

The detection or output of input or output signals synchronized with a reference signal in the central system and decentralized via a connected field bus system is called *Isochronous*. In decentralized automation structures many processing cycles are not synchronized to each other. In the process input signals are detected, evaluated in the user program and the according reactions are connected to the output components. Here the cycles correlate to each other. Due to the telegram runtime of the corresponding bus, the process reaction time can vary respectively the process data are not transferred at a consistent time.

For the synchronization of I/O data, a basic clock is required. This is derived as *Sync-signal* from the EtherCAT system. With each *Sync-Signal*, all input data are buffered and the output data are output, i.e. all data of the process image belong together logically and temporally. The *Sync-signal* serves as a clock generator within its cycle the following functions are performed:

- The current centralized and decentralized input data are latched.
- The output data, which were cached *Sync-signal* cycle are centralized and decentralized output. All output data get simultaneously active.

> *The isochronous functionality on EtherCAT is called Distributed Clocks (DC). For synchronization on EtherCAT DC-capable EtherCAT slaves are required, where DC is also activated. If only modules on the System SLIO backplane bus are to be synchronized, so to generate the Sync-signal, you have to configure EtherCAT without slaves.*

Synchronous cycle interrupt OB 61

With OB 1 no isochronous is possible. For this the high-priority OB 61 is to be used. For isochronous use the OB 61 is started in a defined time interval. The OB 61 is executed according to the following steps, whereby the processing of these steps must be within one cycle so that it is ensured that the output data can be output at the next *Sync-Signal*.

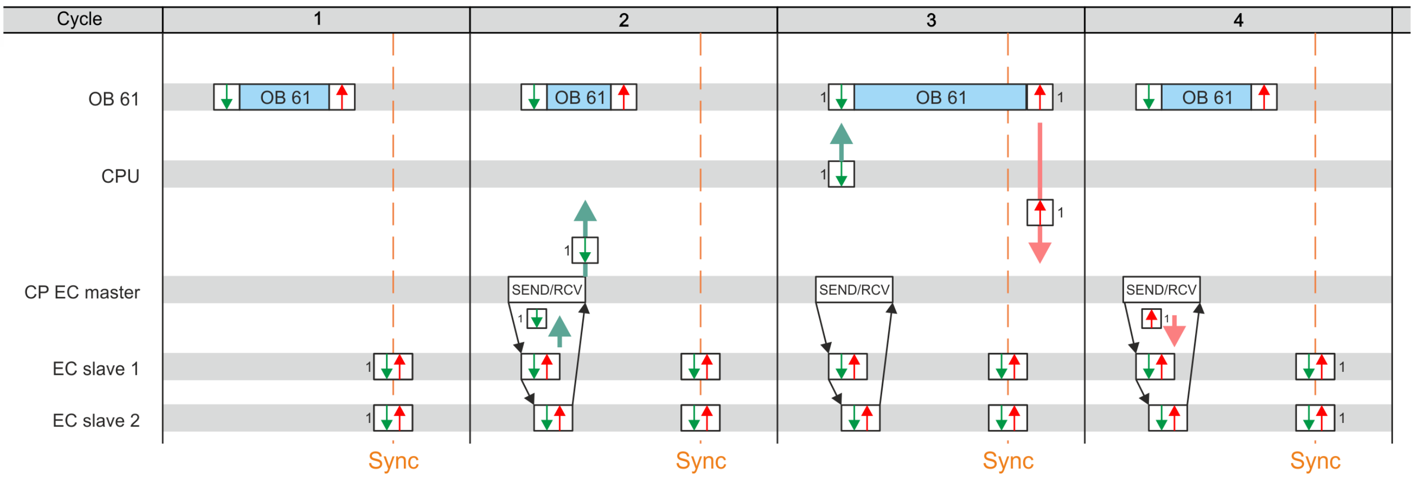
1. The input process image of the OB 61 is refreshed.
2. The user program of the OB 61 is executed.
3. The output process image of the OB 61 is refreshed.

A maximum of 1 EtherCAT cycle can elapse before data changes can be detected to the next following *Sync-Signal*.

> **CAUTION!**
> If a cycle time is exceeded in the OB 61 based on the user program, you can configure an OB 80 (time error). It is called on time error. If this is not available, the CPU switches to STOP.

Sequence OB 61 with EtherCAT master system

The following figure shows a sequence on EtherCAT, how the input data from cycle 1 are buffered and transferred and the resulting output data are output.

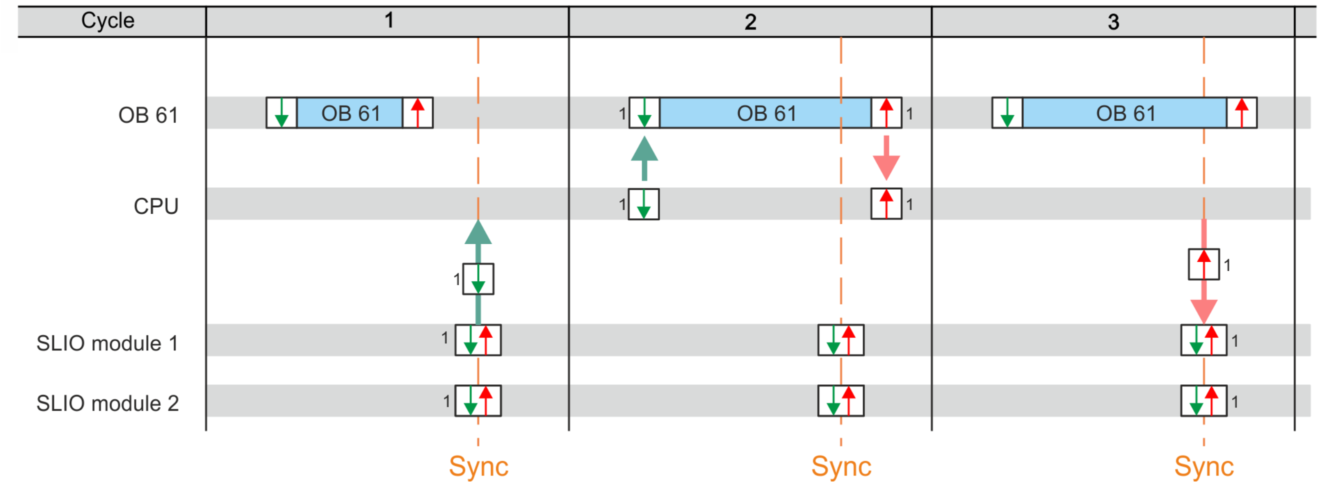


Input process image with the input data from cycle 1
Output process image resulting from the input data from cycle 1
Sync *Sync-Signal* - Sync / Freece I / O

- Cycle 1: The input signals are buffered at the time of the *Sync-Signals* at the EtherCAT input modules and forwarded to the CPU.
- Cycle 2: The input process image is forwarded to the CPU via SEND / RECEIVE via the EtherCAT master.
- Cycle 3: The input process image is transferred to OB 61, OB 61 is processed, and the output process data is transferred from the CPU to the EtherCAT master.
- Cycle 4: The output process image is transferred to the EtherCAT slave via SEND / RECEIVE and is switched to the outputs of the EtherCAT output modules at the time of the *Sync-Signals*.

Sequence OB 61 with System SLIO modules

The following figure shows a sequence in the System SLIO as to how the input data from cycle 1 are buffered and transferred and the resulting output data are output.

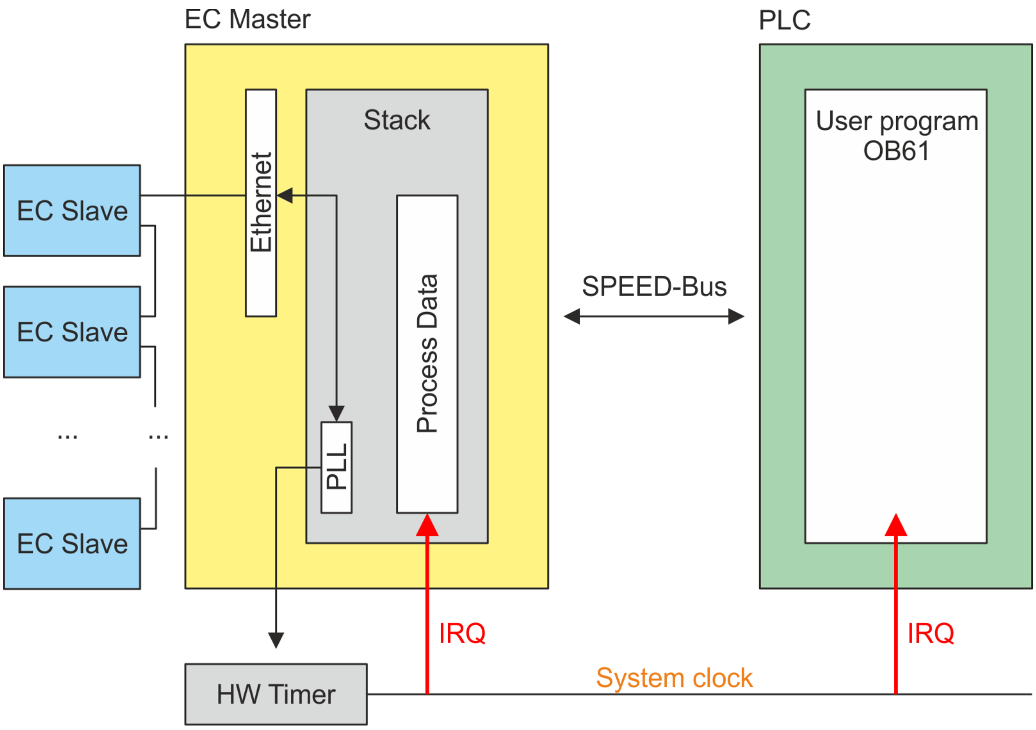


Input process image with the input data from cycle 1
Output process image resulting from the input data from cycle 1
Sync *Sync-Signal* - Sync / Freece I / O

- Cycle 1: The input signals are read from the input modules with the *Sync-signal* and forwarded to the CPU.
- Cycle 2: The input process image is transferred to the OB 61, the OB 61 is processed and then the output process image is forwarded to the System SLIO modules.
- Cycle 3: At the time of the *Sync-Signals* the outputs are enabled on the System SLIO output modules.

Mechanism of synchronization

The CPU components PLC and EtherCAT Master are synchronized by an interrupt. This interrupt is generated from the System SLIO bus timer and the EtherCAT bus cycle time. The synchronization of EtherCAT slaves happens by DC. EtherCAT slaves that do not support DC are not synchronized. With Yaskawa the 1. DC-capable EtherCAT slave in the network always has the DC reference time. The synchronization between the DC-reference time and the EtherCAT master takes place in the EtherCAT master. Here also the System SLIO bus timer is synchronized. The EtherCAT bus cycle time can be configured in *SPEED7 Studio*.

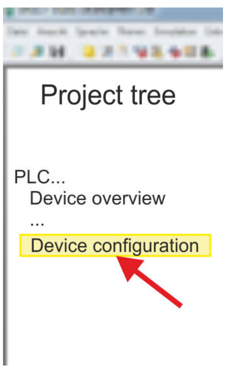


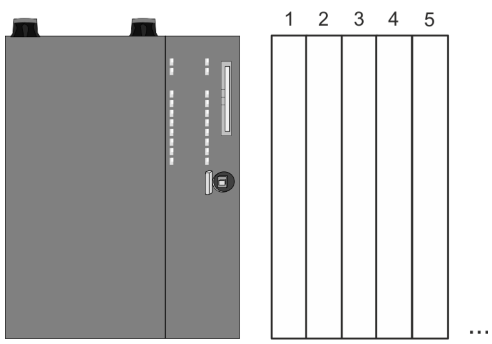
11.3 Configuration

11.3.1 Hardware configuration CPU

Proceeding

1. Start the *SPEED7 Studio*.
2. Create a new project in the *Work area* with *'New project'*.
   ⇨ A new project is created and the view *'Devices and networking'* is shown.
3. Click in the *Project tree* at *'Add new device ...'*.
   ⇨ A dialog for device selection opens.
4. Select from the *'Device templates'* your CPU and click at [OK].
   ⇨ The CPU is inserted in *'Devices and networking'* and the *'Device configuration'* is opened.





Device configuration

Slot	Module	...	...	...	...
0	CPU 015-CEFNR00				

-X1	PG_OP_Ethernet				
-X3	MPI interface				
...	...			...	

11.3.2 Activate isochronous

Please consider the additional functions in the SPEED7 Studio can only be activated, if you have valid license for these functions!

Proceeding

1. Click at the CPU in the *'Device configuration'* and select *'Context menu ➔ Components properties'*.
   ⇨ The properties dialog of the CPU is opened
2. Click at *'Feature Sets'* and activate at *'Motion Control'* the parameter *'EtherCAT Master functionality+Motion Control+...'*.
3. Confirm your input with [OK].
   ⇨ The additional functions are now available in your project. More information about the usage may be found in the online help of the *SPEED7 Studio*.

CAUTION!
Please note that with any change in the feature set settings in the *SPEED7 Studio*, due to the system, the EtherCAT fieldbus system together with the motion control configuration are removed from your project!

Activate additional functions by means of VSC in the CPU
In order to use the additional functions, you must activate them by means of a Yaskawa VSC storage card. By inserting the VSC storage card and then an overall reset, the corresponding additional functions are activated.

'Overview' page 121

OB 60

By activating of the function *'Motion Control'* in the *SPEED7 Studio* the OB 60 is automatically created. The OB is used internally and can not be edited. It used to manage the service data objects (SDO) and diagnostic data. The OB 60 has a higher priority than OB 1. The cycle time for this OB can be configured in the *SPEED7 Studio*. Please note that no peripheral direct access is possible in OB 60.

OB 61

By activating of the function *'Motion Control'* in the *SPEED7 Studio* the OB 61 is automatically created. For the OB a separate process image PI OB 61 is created, which data are consistent during the execution of the OB. OB 61 has a higher priority than OB 60. Please note that no peripheral direct access is possible in OB 61. Within the OB 61 should be the functions which are synchronously should be executed. The following blocks are permitted in the OB 61:

- FBs and FCs without restrictions
- SFCs
  - SFC 20 - BLKMOV
  - SFC 21 - FILL
  - SFC 47 - WAIT
  - SFC 53 - uS_Tick
  - SFC 64 - TIME_TCK
- SFBs
  - SFB 0 - CTU
  - SFB 1 - CTD
  - SFB 2 - CTUD
  - SFB 3 - TP
  - SFB 4 - TON
  - SFB 5 - TOF
  - SFB 7 - TIMEMESS

12 Configuration with Siemens SIMATIC Manager

12.1 Siemens SIMATIC Manager - General

In this part the project engineering of the System SLIO CPU in the Siemens SIMATIC Manager is shown. Here only the basic usage of the Siemens SIMATIC Manager together with a System SLIO CPU is shown. In the Siemens SIMATIC Manager your Yaskawa PLCs may be configured and linked. For diagnostics online tools are available.

> *More information can be found in the online help respectively in documentation of the Siemens SIMATIC Manager.*

12.2 Siemens SIMATIC Manager - Hardware configuration - CPU

Precondition

- The configuration of the CPU takes place at the *'hardware configurator'* of the Siemens SIMATIC Manager V 5.5 SP2 and up.
- The configuration of the System SLIO CPU happens by means of a virtual PROFINET IO device *'VIPA SLIO System'*. The *'VIPA SLIO System'* is to be installed in the hardware catalog by means of the GSDML.

> *For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required!*

Installation IO device *'VIPA SLIO System'*

The installation of the PROFINET IO devices *'VIPA SLIO System'* happens in the hardware catalog with the following approach:

1. Go to the *'Download Center'* of www.yaskawa.eu.com.
2. Load under *'GSDML SLIO'* the according file for your System SLIO.
3. Extract the file into your working directory.
4. Start the Siemens hardware configurator.
5. Close all the projects.
6. Select *'Options → Install new GSD file'*
7. Navigate to your working directory and install the according GSDML file.
   ⇨ After the installation according PROFINET IO device can be found at *'PROFINET IO → Additional field devices → I/O → ... SLIO System'*

Proceeding

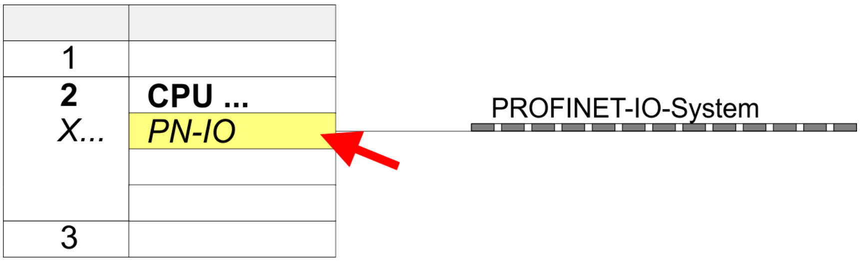
In the Siemens SIMATIC Manager the following steps should be executed:

1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'*-Number 2 the CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).

Slot	Module
1	
2	**CPU 31...**
X1	*MPI/DP*
X2	*PN-IO*
X2...	*Port 1*
X2...	*Port 2*
3	

4. Click at the sub module *'PN-IO'* of the CPU.

5. Select *'Context menu ➔ Insert PROFINET IO System'*.

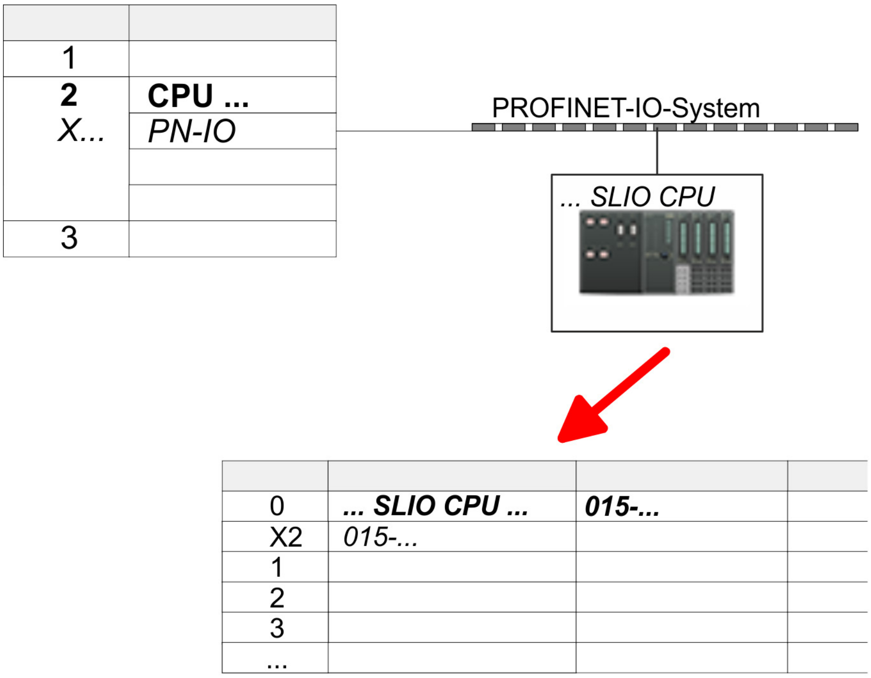


6. Create with [New] a new sub net and assign valid address data

7. Click at the sub module *'PN-IO'* of the CPU and open with *'Context menu ➔ Properties'* the properties dialog.

8. Enter at *'General'* a device name. The device name must be unique at the Ethernet subnet.

> *Please leave 'Send clock' in Tab 'PROFINET' at 1ms, otherwise this leads to a configuration error!*



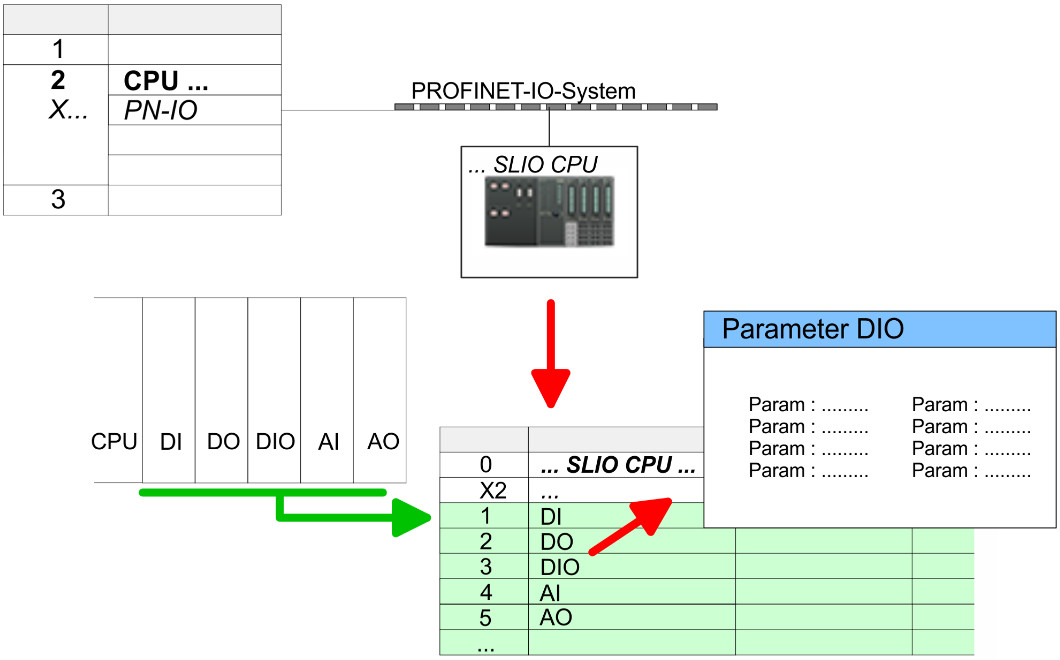
9. Navigate in the hardware catalog to the directory *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ ... SLIO System'* and connect the IO device *'015-CEFNR00 CPU'* to your PROFINET system.

⇨ In the slot overview of the PROFINET IO device *'... SLIO CPU'* the CPU is already placed at slot 0. From slot 1 you can place your System SLIO modules.

12.3 Siemens SIMATIC Manager - Hardware configuration - I/O modules

Hardware configuration of the modules

Starting with slot 1 place in the slot overview of the PROFINET IO device *'... SLIO CPU'* your System SLIO modules in the plugged sequence. To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU.



Parametrization

For parametrization double-click during the project engineering at the slot overview on the module you want to parametrize. In the appearing dialog window you may set the wanted parameters. Here you can make your parameter settings.

Parametrization during runtime

By using the SFCs 55, 56 and 57 you may alter and transfer parameters for wanted modules during runtime. For this you have to store the module specific parameters in so called "record sets". More detailed information about the structure of the record sets is to find in the according module description.

12.4 Siemens SIMATIC Manager - Hardware configuration - Ethernet PG/OP channel

Overview

> ***Please note!***
> - *At the first commissioning respectively after a reset to factory setting the Ethernet interface has no IP address.*
> - *For online access, you have to assign valid IP address data to it by means of "Initialization".*
> - *After initialization, you can transfer the IP address data to your project.*

The CPU has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU.

- The Ethernet PG/OP channel (X1/X5) is designed as switch. This enables PG/OP communication via the connections X1 and X5.
- Configurable connections are possible.
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040
- Via the Ethernet PG/OP channel, you have access to:
  - Device website, where you can find information on firmware status, connected peripherals, current cycle times, etc.
  - *WebVisu* project, which is to be created in the *SPEED7 Studio*.
  - PROFINET I-Device

Assembly and commissioning

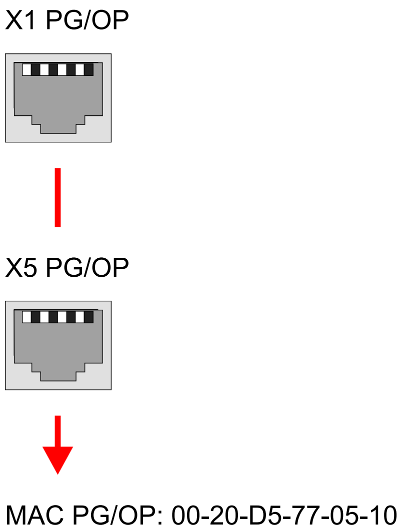
1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the one of the Ethernet jacks (X1, X5) of the Ethernet PG/OP channel to Ethernet.
4. Switch on the power supply.

   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization" via PLC functions

The initialization via PLC functions takes place with the following proceeding:

- Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of your CPU with the name "MAC PG/OP: ...".



Assign IP address parameters

You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens SIMATIC Manager starting with version V 5.5 & SP2 with the following proceeding:

1. Start the Siemens SIMATIC Manager and set via *'Options → Set PG/PC interface'* the access path to *'TCP/IP -> Network card ....'*.
2. Open with *'PLC → Edit Ethernet Node n'* the dialog window with the same name.
3. To get the stations and their MAC address, use the [Browse] button or type in the MAC Address. The Mac address may be found at the 1. label beneath the front flap of the CPU.
4. Choose if necessary the known MAC address of the list of found stations.
5. Either type in the IP configuration like IP address, subnet mask and gateway.
6. Confirm with [Assign IP configuration].

   ⇨ Direct after the assignment the Ethernet PG/OP channel may be reached online by these address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.

Take IP address parameters in project

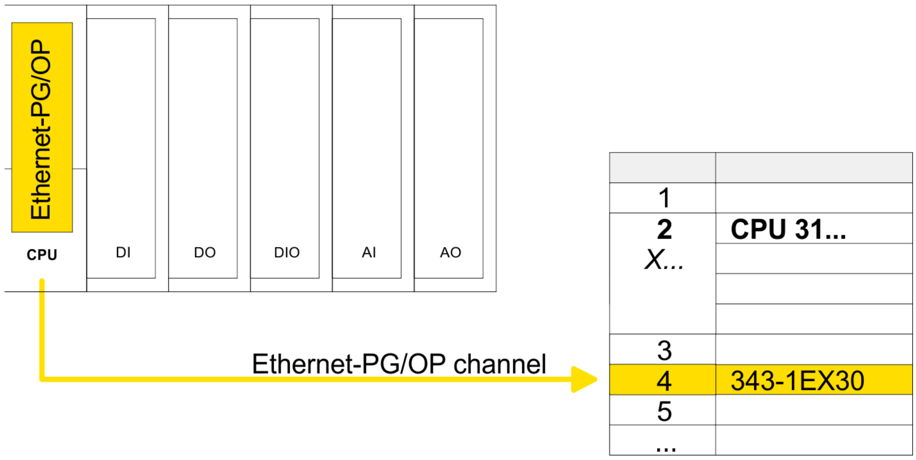
1. Open the Siemens hardware configurator and configure the Siemens CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).
2. For the Ethernet PG/OP channel you have to configure at slot 4 a Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \CP 343-1 \ 6GK7 343-1EX30 0XE0 V3.0).

   > **CAUTION!**
   > Please configure the diagnostic addresses of the CP343-1EX30 for *'PN-IO'*, *'Port1'* and *'Port2'* so that there is no overlap in the peripheral input area. Otherwise, your CPU will not start and you will get the diagnostic entry 0xE904. These address overlaps can not be recognized by the Siemens SIMATIC Manager.

3. Open the property window via double-click on the CP 343-1EX30 and enter for the CP at *'Properties'* the IP address data, which you have assigned before.

4. Assign the CP to a *'Subnet'*. Without assignment the IP address data are not used!

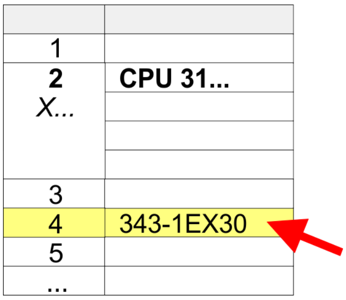
5. Transfer your project.



12.4.1 Time-of-day synchronization

NTP method

In the NTP mode (**N**etwork **T**ime **P**rotocol) the module sends as client time-of-day queries at regular intervals to all configured NTP servers within the sub net. You can define up to 4 NTP server. Based on the response from the servers, the most reliable and most exact time-of-day is determined. Here the time with the lowest *stratum* is used. *Stratum 0* is the time standard (atomic clock). *Stratum 1* are directly linked to this NTP server. Using the NTP method, clocks can be synchronized over subnet boundaries. The configuration of the NTP servers is carried out in the Siemens SIMATIC Manager via the CP, which is already configured.



1. Open the properties dialog via double-click on the CP 343-1EX30.
2. Select the register *'Time-of-day synchronization'*.
3. Activate the NTP method by enabling *'Activate NTP time-of-day synchronization'*.
4. Click at [Add] and add the corresponding NTP server.
5. Select your *'time zone'*. In the NTP method, UTC (**U**niversal **T**ime **C**oordinated) is generally transmitted; this corresponds to GMT (Greenwich Mean Time). By configuring the local time zone, you can set a time offset to UTC.
6. Select your *'Update interval'*. Within this interval the time of the module is synchronized once.
7. Close the dialog with [OK].
8. Save and transfer your project to the CPU.

   ⇨ After transmission, the NTP time is requested by each configured time server and the best response for the time synchronization is used.

Please note that although the time zone is evaluated, an automatic changeover from winter to summer time is not supported. Industrial systems with time-of-day synchronization should always be set in accordance to the winter time.

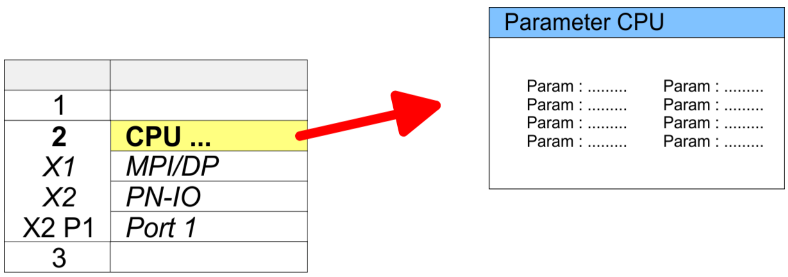
With the FC 61 you can determine the local time in the CPU. More information about the usage of this block may be found in the manual "SPEED7 Operation List".

12.5 Siemens SIMATIC Manager - Hardware configuration - Parametrization

12.5.1 Standard CPU parameters

Parametrization via Siemens CPU 315-2 PN/DP

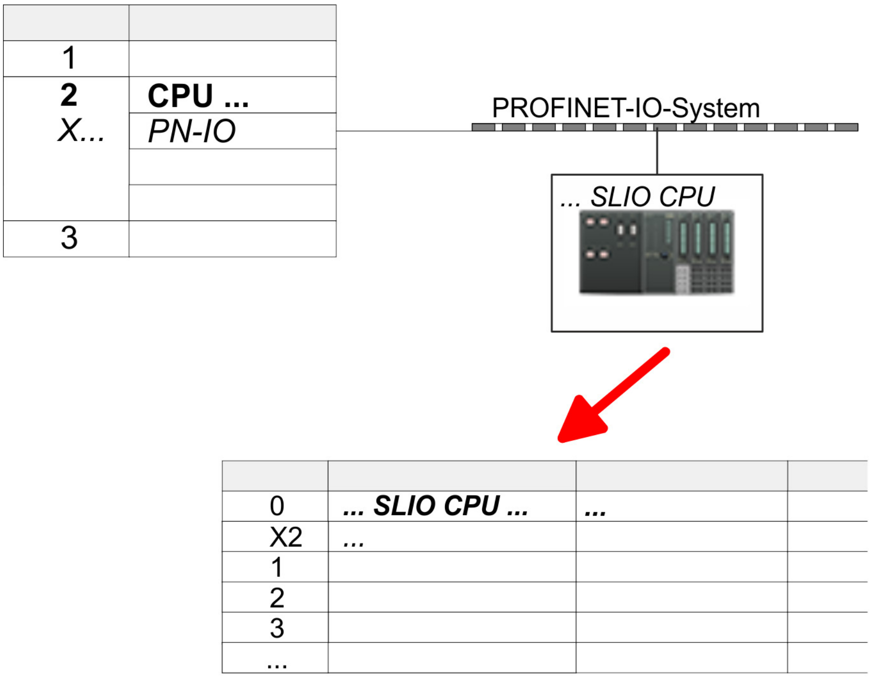
Since the CPU is to be configured as Siemens CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2) in the hardware configurator, the standard parameters of the CPU may be set with "Object properties" of the CPU 315-2 PN/DP during hardware configuration. Via a double-click on the CPU 315-2 PN/DP the parameter window of the CPU may be accessed. Using the registers you get access to every standard parameter of the CPU. ↳ *Chap. 4.8.1 'Parameter CPU' page 80*



12.5.2 Product specific CPU parameters

Except of the product specific CPU parameters the CPU parametrization takes place in the parameter dialog of the CPU 315-2 PN/DP from Siemens. After the hardware configuration of the CPU you can set the parameters of the CPU in the virtual IO device *'VIPA SLIO CPU'*. Via double-click at the *'VIPA SLIO CPU'* the properties dialog is opened Here the following parameters may be accessed:

- Function X2 (PtP/MPI)
- MPI address X2
- MPI Baud rate X2
- Number of retentive memory/timer/counter



12.5.3 Parameters for MPI/DP

The properties dialog of the MPI(PB) interface X3 is opened via a double click to the sub module MPI/DP

To switch the interface to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from Yaskawa. By plugging the VSC storage card and then an overall reset the according functionality is activated. *Chap. 4.15 'Deployment storage media - VSD, VSC' page 121*

12.6 Siemens SIMATIC Manager - Project transfer

Overview

There are the following possibilities for project transfer into the CPU:

- Transfer via MPI (optional via PROFIBUS)
- Transfer via Ethernet
- Transfer via memory card

To switch the interface X3 MPI(PB) to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from Yaskawa. By plugging the VSC storage card and then an overall reset the according functionality is activated. *Chap. 4.15 'Deployment storage media - VSD, VSC' page 121*

12.6.1 Transfer via MPI / optional PROFIBUS

General

For transfer via MPI / optional PROFIBUS there are the following 2 interface:

'X3: MPI(PB) interface' page 54

'X2: PtP(MPI) interface' page 53

With an overall reset CPU the configuration via X2 PtP(MPI) is not possible!

Net structure

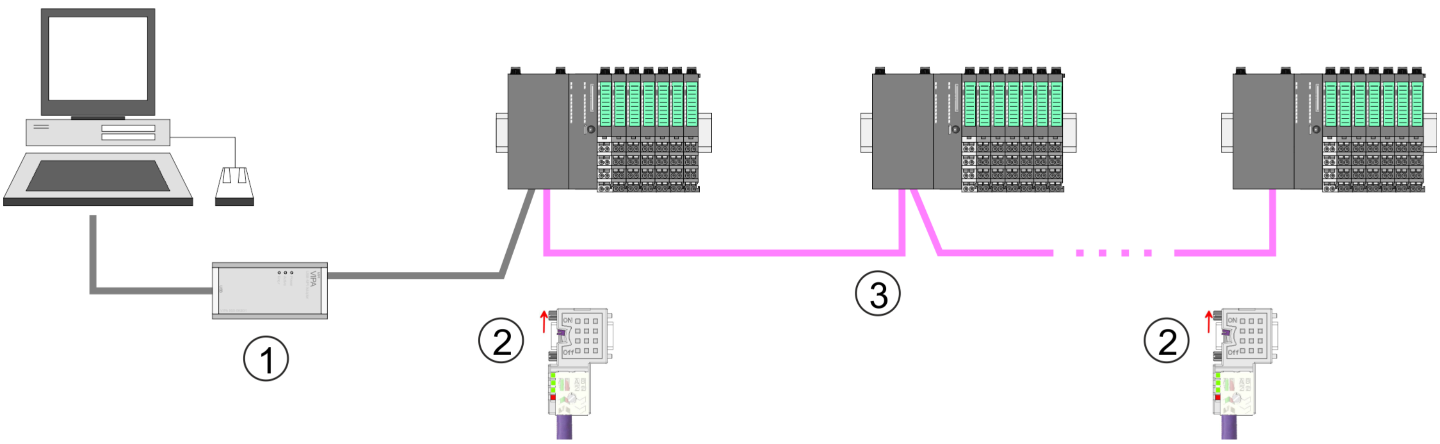
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs with 187.5kbaud. The CPUs are delivered with MPI address 2.

MPI programming cable

The MPI programming cables are available at Yaskawa in different variants. The cables provide a RS232 res. USB plug for the PC and a bus enabled RS485 plug for the CPU. Due to the RS485 connection you may plug the MPI programming cables directly to an already plugged plug on the RS485 jack. Every bus participant identifies itself at the bus with an unique address, in the course of the address 0 is reserved for programming devices.

Terminating resistor

A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment. Please make sure that the participants with the activated terminating resistors are always power supplied. Otherwise it may cause interferences on the bus.



1 MPI programming cable
2 Activate the terminating resistor via switch
3 MPI/PROFIBUS network

Approach transfer via MPI interface

1. Connect your PC to the MPI jack of your CPU via a MPI programming cable.
2. Load your project in the SIMATIC Manager from Siemens.
3. Choose in the menu *'Options ➔ Set PG/PC interface'*.
4. Select in the according list the "PC Adapter (MPI)"; if appropriate you have to add it first, then click on [Properties].
5. Set in the register MPI the transfer parameters of your MPI net and type a valid *address.*
6. Switch to the register *Local connection*.
7. Set the COM port of the PCs and the transfer rate 38400baud for the MPI programming cable.
8. Transfer your project via *'PLC ➔ Load to module'* via MPI to the CPU and save it with *'PLC ➔ Copy RAM to ROM'* on a memory card if one is plugged.

Proceeding Transfer via PROFIBUS interface

To switch the interface to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media of Yaskawa. By plugging the VSC storage card and then an overall reset the according functionality is activated.

1. Connect your PC to the MPI(PB) jack X3 of your CPU via a MPI programming cable.
2. Load your project in the Siemens SIMATIC Manager.
3. Choose in the menu *'Options ➔ Set PG/PC interface'*.
4. Select in the according list the "PC Adapter (PROFIBUS)"; if appropriate you have to add it first, then click at [Properties].
5. Set in the register PROFIBUS the transfer parameters of your PROFIBUS net and enter a valid *PROFIBUS address*. The *PROFIBUS address* must be assigned to the DP master by a project before.
6. Switch to the register *Local connection*.
7. Set the COM port of the PCs and the transfer rate 38400baud for the MPI programming cable.
8. Transfer your project via *'PLC ➔ Load to module'* via PROFIBUS to the CPU and save it with *'PLC ➔ Copy RAM to ROM'* on a memory card if one is plugged.

Transfer via PROFIBUS is available by DP master, if projected as master and assigned with a PROFIBUS address before. In slave operation you have also to enable the option 'Test , Commissioning, Routing ' when selecting the slave mode.

12.6.2 Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X1/X5: Ethernet PG/OP channel

Initialization

So that you may access the according Ethernet interface you have to assign IP address parameters by means of the "initialization".

- X1/X5: Ethernet PG/OP channel
  – *Chap. 12.4 'Siemens SIMATIC Manager - Hardware configuration - Ethernet PG/OP channel' page 297*

Transfer

1. For the transfer, connect, if not already done, the appropriate Ethernet port to your Ethernet.
2. Open your project with the Siemens SIMATIC Manager.
3. Set via *'Options ➔ Set PG/PC Interface'* the access path to "TCP/IP → Network card .... ".
4. Click to *'PLC ➔ Download'* Download → the dialog "Select target module" is opened. Select your target module and enter the IP address parameters of the Ethernet PG/OP channel for connection. Provided that no new hardware configuration is transferred to the CPU, the entered Ethernet connection is permanently stored in the project as transfer channel.
5. With [OK] the transfer is started.

System dependent you get a message that the projected system differs from target system. This message may be accepted by [OK].

→ Your project is transferred and may be executed in the CPU after transfer.

12.6.3 Transfer via memory card

The memory serves as external transfer and storage medium. There may be stored several projects and sub-directories on a memory card. Please regard that your current project is stored in the root directory and has one of the following file names:

- S7PROG.WLD
- AUTOLOAD.WLD

With *'File ➔ Memory Card File ➔ New'* in the Siemens SIMATIC Manager a new wld file may be created. After the creation copy the blocks from the project blocks folder and the *System data* into the wld file.

Transfer memory card → CPU

The transfer of the application program from the memory card into the CPU takes place depending on the file name after an overall reset or PowerON.

- *S7PROG.WLD* is read from the memory card after overall reset.
- *AUTOLOAD.WLD* is read from the memory card after PowerON.

A short lightning up of the SD LED of the CPU marks the active transfer. Please regard that your user memory serves for enough space for your user program, otherwise your user program is not completely loaded and the SF LED gets on.

Transfer CPU → memory card

When a memory card has been installed, the write command stores the content of the RAM as S7PROG.WLD on the memory card.

The write command is controlled by means of the block area of the Siemens SIMATIC Manager *'PLC ➔ Copy RAM to ROM'*. The SD LED lights up during the write access. When the LED expires, the write process is finished.

If this project is to be loaded automatically from the memory card with PowerON, you have to rename this to on the memory card to *AUTOLOAD.WLD*.

Checking the transfer operation

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries you choose in the Siemens SIMATIC manager *'PLC ➔ Module information'*. Via the register "Diagnostic Buffer" you reach the diagnostic window. *Chap. 4.19 'Diagnostic entries' page 128*

12.7 Siemens SIMATIC Manager - Access to diagnostics entries

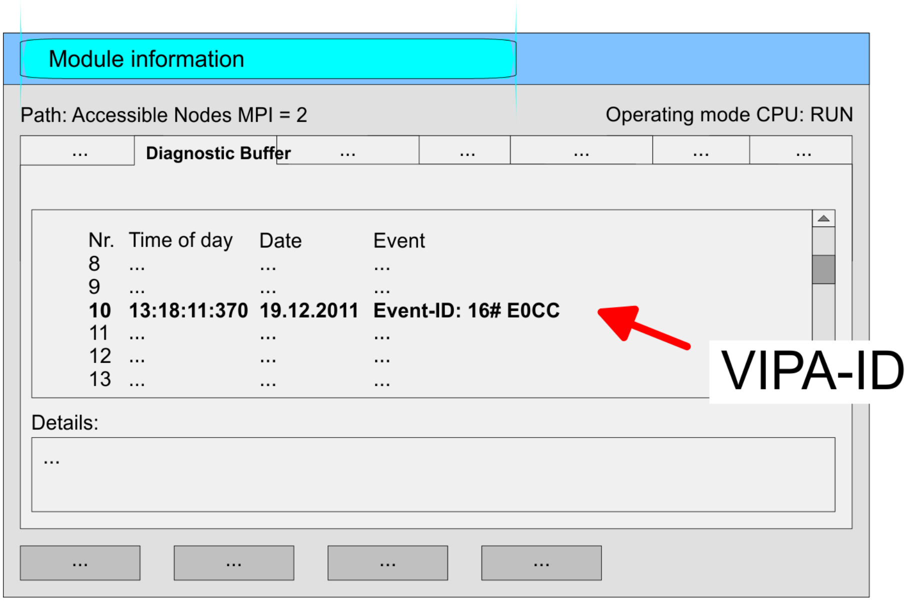
Entries in the diagnostics buffer

- You may read the diagnostics buffer of the CPU via the Siemens SIMATIC Manager.
- Besides of the standard entries in the diagnostics buffer, the Yaskawa CPUs support some additional specific entries as event IDs.
- The current content of the diagnostics buffer is stored at the memory card by means of the CMD DIAGBUF.

Every register of the module information is supported by the Yaskawa CPUs. More information may be found at the online help of the Siemens SIMATIC Manager.

Monitoring the diagnostic entries

To monitor the diagnostics entries you choose in the Siemens SIMATIC manager *'PLC ➔ Module information'*. Via the register "Diagnostics Buffer" you reach the diagnostics window:



The diagnostic is independent from the operating mode of the CPU. Max. 100 diagnostic entries can be stored in the CPU. *Appendix A 'System specific event IDs' page 349*

12.8 Siemens SIMATIC Manager - Option: Deployment PROFIBUS communication

12.8.1 Overview

Enable bus functionality via VSC

To switch the MPI(PB) interface X3 to PROFIBUS functionality, you have to enable the according bus functionality by means of a VSC storage media from Yaskawa. By plugging the VSC storage card and then an overall reset the according functionality is enabled.

'Overview' page 121

PROFIBUS DP

- PROFIBUS is an international standard applicable to an open and serial field bus for building, manufacturing and process automation that can be used to create a low (sensor-/actuator level) or medium (process level) performance network of programmable logic controllers.
- PROFIBUS comprises an assortment of compatible versions. The following details refer to PROFIBUS DP.
- PROFIBUS DP is a special protocol intended mainly for automation tasks in a manufacturing environment. DP is very fast, offers Plug'n'Play facilities and provides a cost-effective alternative to parallel cabling between PLC and remote I/O. PROFIBUS DP was designed for high-speed data communication on the sensor-actuator level.
- The data transfer referred to as "Data Exchange" is cyclical. During one bus cycle, the master reads input values from the slaves and writes output information to the slaves.

CPU with DP master

The PROFIBUS DP master is to be configured in the hardware configurator from Siemens. Here the configuration happens by the sub module X1 (MPI/DP) of the Siemens CPU. After the transmission of the data to the CPU, the configuration data are internally passed on to the PROFIBUS master part. During the start-up the DP master automatically includes his data areas into the address range of the CPU. Project engineering in the CPU is not required.

Deployment of the DP master with CPU

Via the PROFIBUS DP master PROFIBUS DP slaves may be coupled to the CPU. The DP master communicates with the DP slaves and links up its data areas with the address area of the CPU. At every POWER ON respectively overall reset the CPU fetches the I/O mapping data from the master. At DP slave failure, the OB 86 is requested. If this is not available, the CPU switches to STOP and BASP is set. As soon as the BASP signal comes from the CPU, the DP master is setting the outputs of the connected periphery to zero. The DP master remains in the operating mode RUN independent from the CPU.

DP slave operation

For the deployment in a super-ordinated master system you first have to project your slave system as Siemens CPU in slave operation mode with configured in-/output areas. Afterwards you configure your master system. Couple your slave system to your master system by dragging the CPU 31x from the hardware catalog at *Configured stations* onto the master system, choose your slave system and connect it.

Operating mode DP slave: Test, commissioning, routing (active/passive)

There is the possibility to enable the option *'Test, commissioning, routing'* in the hardware configuration by means of the properties dialog of the PROFIBUS via the register *'Operating mode'* at *'DP slave'*. The activation affects as follows:

- The PROFIBUS interface gets an "active" PROFIBUS node, this means it is involved in the token rotation.
- Via this interface you have PG/OP functions (programming, status request, control, test).

- The PROFIBUS interface serves as a gateway (S7 routing).
- The bus rotation time can exceed.

When disabled, the PROFIBUS interface operates as passive DP slave with the following characteristics:

- The PROFIBUS interface gets an "passive" PROFIBUS node, this means it is not involved in the token rotation.
- Bus rotation time is not influenced.
- S7 routing is not possible.

12.8.2 Fast introduction

Overview

The PROFIBUS DP master is to be configured in the hardware configurator. Here the configuration happens by means of the sub module X1 (MPI/DP) of the Siemens CPU.

> ***Enable bus functionality via VSC***
> *To switch the MPI(PB) interface X3 to PROFIBUS functionality, you have to enable the according bus functionality by means of a VSC storage media from Yaskawa. By plugging the VSC storage card and then an overall reset the according functionality is enabled.*
>
> *'Overview' page 121*

Steps of configuration

For the configuration of the PROFIBUS DP master please follow the following approach:

- **Enable bus functionality via VSC**
- **Hardware configuration - CPU**
- **Deployment as DP master or DP slave**
  - With activating the bus function *'PROFIBUS DP master'* by means of the VSC, the bus function *'PROFIBUS DP slave'* is also unlocked.
- **Transfer of the complete project to CPU**

> *With the Siemens SIMATIC Manager, the CPU 015-CEFNR00 is to be configured as*
>
> *CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2)*
>
> *The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X1 (MPI/DP).*

12.8.3 Enable bus functionality via VSC

Enabling

'Overview' page 121

12.8.4 Hardware configuration - CPU

Precondition

The configuration of the CPU takes place at the Siemens *'hardware configurator'*. The hardware configurator is part of the Siemens SIMATIC Manager. It serves for project engineering. Please use for configuration the Siemens SIMATIC Manager V 5.5 SP2 and up. The modules, which may be configured here are listed in the hardware catalog. If necessary you have to update the hardware catalog with *'Options ➔ Update Catalog'*.

> *For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required!*

Proceeding

With the Siemens SIMATIC Manager the following steps should be executed:

1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'*-Number 2 the CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).

Slot	Module
1	
2	**CPU 31...**
X1	*MPI/DP*
X2	*PN-IO*
X2...	*Port 1*
X2...	*Port 2*
3	

The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X1 (MPI/DP).

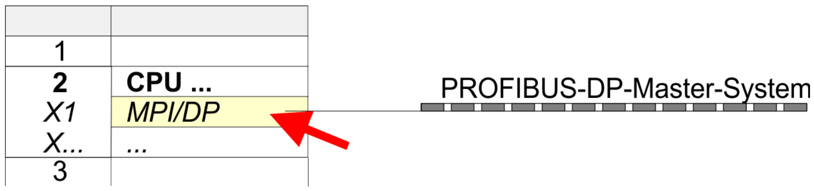
12.8.5 Deployment as PROFIBUS DP master

Precondition

The hardware configuration described before was established.

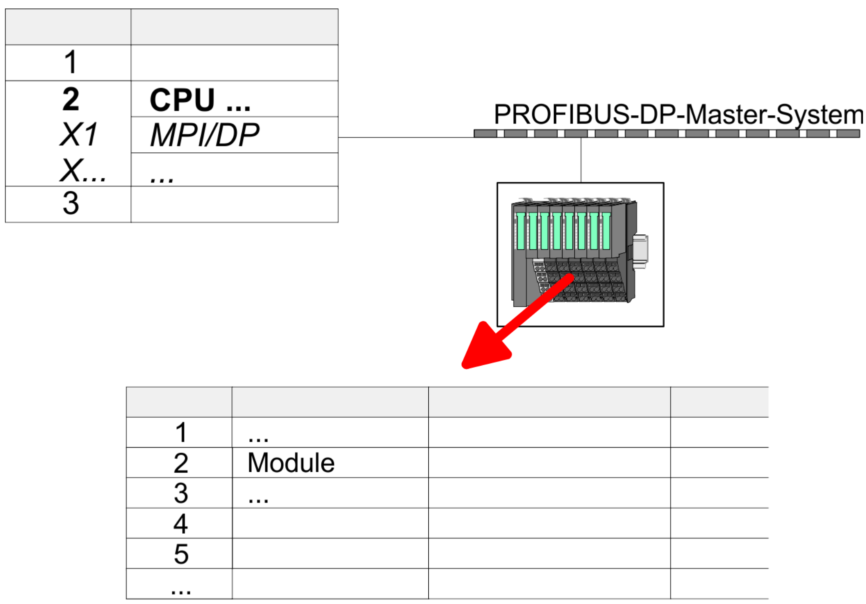
Proceeding

1. Open the properties dialog of the DP interface of the CPU by means of a double-click at *'MPI/DP'*.
2. Set at Interface: Type "PROFIBUS".
3. Connect to PROFIBUS and preset an address (preferably 2). Confirm your input with [OK].
4. Switch at Operating mode to "DP master" and confirm the dialog with [OK].
   ⇨ A PROFIBUS DP master system is inserted:



Now the project engineering of your PROFIBUS DP master is finished. Please link up now your DP slaves with periphery to your DP master.

1. For the project engineering of PROFIBUS DP slaves you search the concerning PROFIBUS DP slave in the hardware catalog and drag&drop it in the subnet of your master.
2. Assign a valid PROFIBUS address to the DP slave.
3. Link up the modules of your DP slave system in the plugged sequence and add the addresses that should be used by the modules.
4. If needed, parametrize the modules.
5. Save, compile and transfer your project.



12.8.6 Deployment as PROFIBUS DP slave

Fast introduction

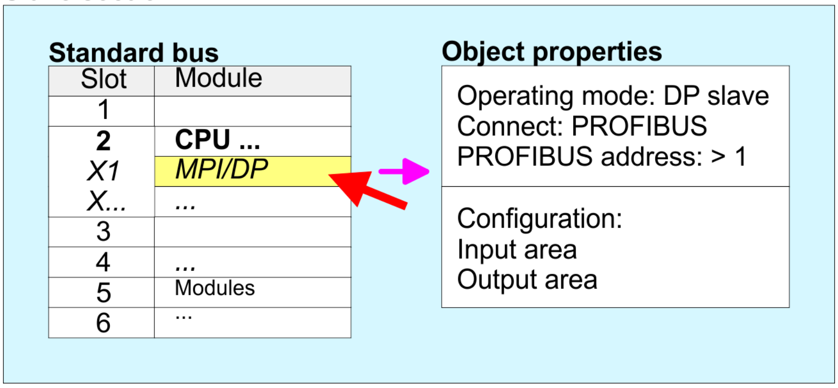
In the following the deployment of the PROFIBUS section as "intelligent" DP slave on master system is described, which exclusively may be configured in the Siemens SIMATIC Manager. The following steps are required:

1. Configure a station with a CPU with operating mode DP slave.
2. Connect to PROFIBUS and configure the in-/output area for the slave section.
3. Save and compile your project.
4. Configure another station with another CPU with operating mode DP master.
5. Connect to PROFIBUS and configure the in-/output ranges for the master section.
6. Save, compile and transfer your project to your CPU.

Project engineering of the slave section

1. Start the Siemens SIMATIC Manager and configure a CPU as described at "Hardware configuration - CPU".
2. Designate the station as "...DP slave".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at *'MPI/DP'*.
5. Set Interface type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 3) and confirm with [OK].
7. Switch at Operating mode to "DP slave" .
8. Via Configuration you define the in-/output address area of the slave CPU, which are to be assigned to the DP slave.
9. Save, compile and transfer your project to your CPU.

Slave section

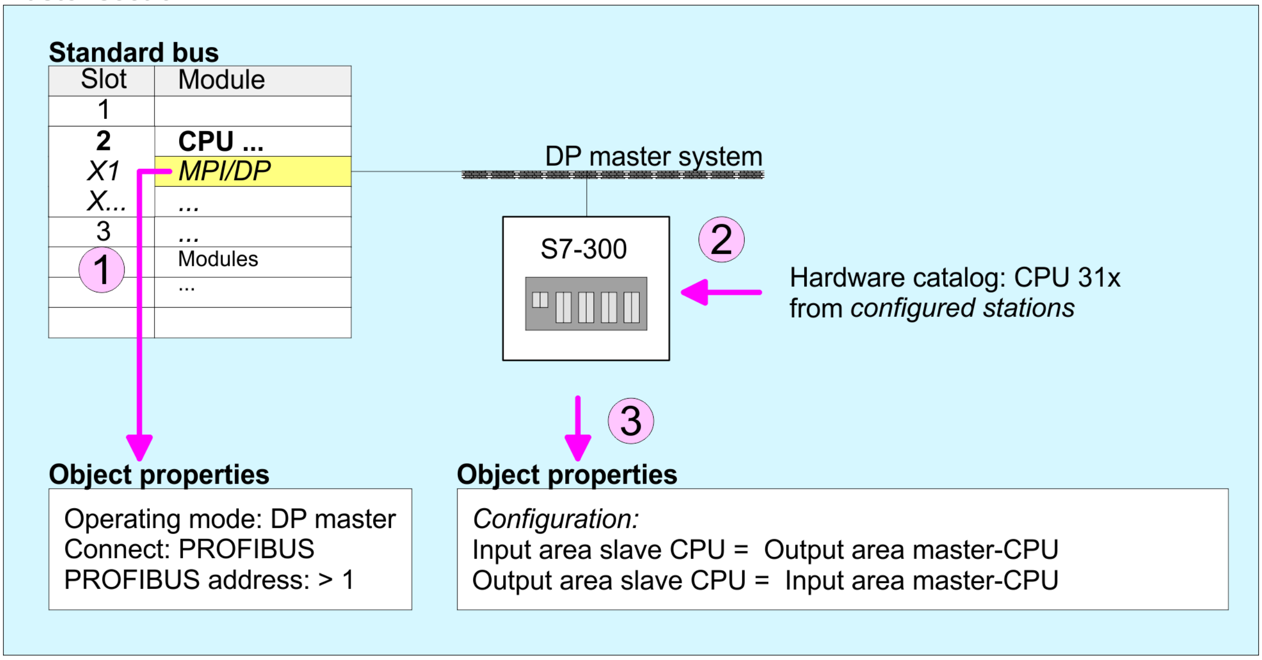


Project engineering of the master section

1. Insert another station and configure a CPU.
2. Designate the station as "...DP master".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at *'MPI/DP'*.
5. Set Interface: type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 2) and confirm with [OK].
7. Switch at Operating mode to "DP master" and confirm the dialog with [OK].

8. Connect your slave system to this master system by dragging the "CPU 31x" from the hardware catalog at *Configured stations* onto the master system and select your slave system to be coupled.
9. Open the *Configuration* at *Object properties* of your slave system.
10. Via double click to the according configuration line you assign the according input address area on the master CPU to the slave output data and the output address area to the slave input data.
11. Save, compile and transfer your project to your CPU.

Master section



12.8.6.1 Diagnostic functions

Overview

PROFIBUS DP provides an extensive set of diagnostic functions for quick error localization. Diagnostic messages are transferred via the bus and collected by the master. In the operating mode DP slave the CPU sends diagnostic data when requested by the master or in case of an error. Since a part of the diagnostic data (Byte 11 ... 15) is located in the peripheral address area of the CPU, you may start the diagnostics and modify the diagnostic data. Diagnostic data consist of:

- Standard diagnostic data (Byte 0 ... 5),
- Device specific diagnostic data (Byte 6 ... 15).

Structure

The diagnostic data have the following structure:

Standard diagnostic data

Byte 0	Station status 1
Byte 1	Station status 2
Byte 2	Station status 3
Byte 3	Master address
Byte 4	Ident number (low)
Byte 5	Ident number High

Device specific diagnostic data

Byte 6	Length and code device specific diagnostic
Byte 7	Device specific diagnostic messages
Byte 8... Byte 10	reserved
Byte 11 ... Byte 15	**User-specific diagnostic data is mapped into the peripheral addressing range of the CPU and may be modified and sent to the master.**

Standard diagnostic data

More detailed information to the structure of the slave standard diagnostic data can be found in the standard papers of the PROFIBUS User Organization. The slave diagnostic data have the following structure:

Byte	Bit 7 ... Bit 0
0	■ Bit 0: 0 (fix) ■ Bit 1: Slave is not yet ready for data exchange ■ Bit 2: Configuration data are not identical ■ Bit 3: Slave has external diagnostic data ■ Bit 4: Slave does not provide this function ■ Bit 5: 0 (fix) ■ Bit 6: Wrong parametrization ■ Bit 7: 0 (fix)
1	■ Bit 0: Slave needs new parametrization ■ Bit 1: Static diagnostic ■ Bit 2: 1 (fix) ■ Bit 3: Response monitoring active ■ Bit 4: Freeze command received ■ Bit 5: Sync command received ■ Bit 6: reserved ■ Bit 7: 0 (fix)
2	■ Bit 0 ... Bit 6: reserved ■ Bit 7: Diagnostic data overflow
3	Master address after parametrization
4	ID number high byte
5	ID number low byte

Device specific diagnostic data

The device related diagnostic data provide detailed information on the slave and the peripheral modules. The length of the device related diagnostic data is fixed at 10byte

Byte	Bit 7 ... Bit 0
6	■ Bit 0 ... 5: Length device specific diagnostic data – 001010: Length 10byte (fix) ■ Bit 6 ... 7: Code for device specific diagnostic – 00: Code 00 (fix)
7	■ Device specific diagnostic message – 12h: Error: Data length parameters – 13h: Error: Data length configuration data – 14h: Error: Configuration entry – 15h: Error: VPC3 buffer calculation – 16h: Missing configuration data – 17h: Error: Comparison DP parametrization and configuration – 40h: User specific diagnostic data is valid
8 ...10	reserved
11 ...15	**User specific diagnostic data that are stored behind the diagnostic status byte in the process picture of the CPU.** **This data may be overwritten and forwarded to the master.**

Release diagnostic

- In case of a diagnostic the contents of Byte 11 ... 15 of the device specific diagnostic data will be transferred to the process image of the CPU and this preceded by a status byte.
- Where this diagnostic block with a length of 6byte is located in the process image can be defined via the CPU parameters.
- You start diagnostics by means of a status change from 0 → 1 in the diagnostic status byte. This transmits the respective diagnostic message to the master.
- **A status of 0000 0011 is ignored!**

The diagnostic block of the CPU has the following structure:

Byte	Bit 7 ... Bit 0
0	Diagnostic status byte ■ Bit 0: User specific diagnostic data – 0: Invalid diagnostic data – 1: Valid diagnostic data (starting a diagnostic) ■ Bit 1: Delete diagnostic – 0: Diagnostic deletion not valid – 1: Diagnostic deletion valid ■ Bit 2 ... Bit 7: reserved
1 ... 5	User specific diagnostic data equal to Byte 11 ... 15 of device specific diagnostic

12.8.7 PROFIBUS installation guidelines

PROFIBUS in general

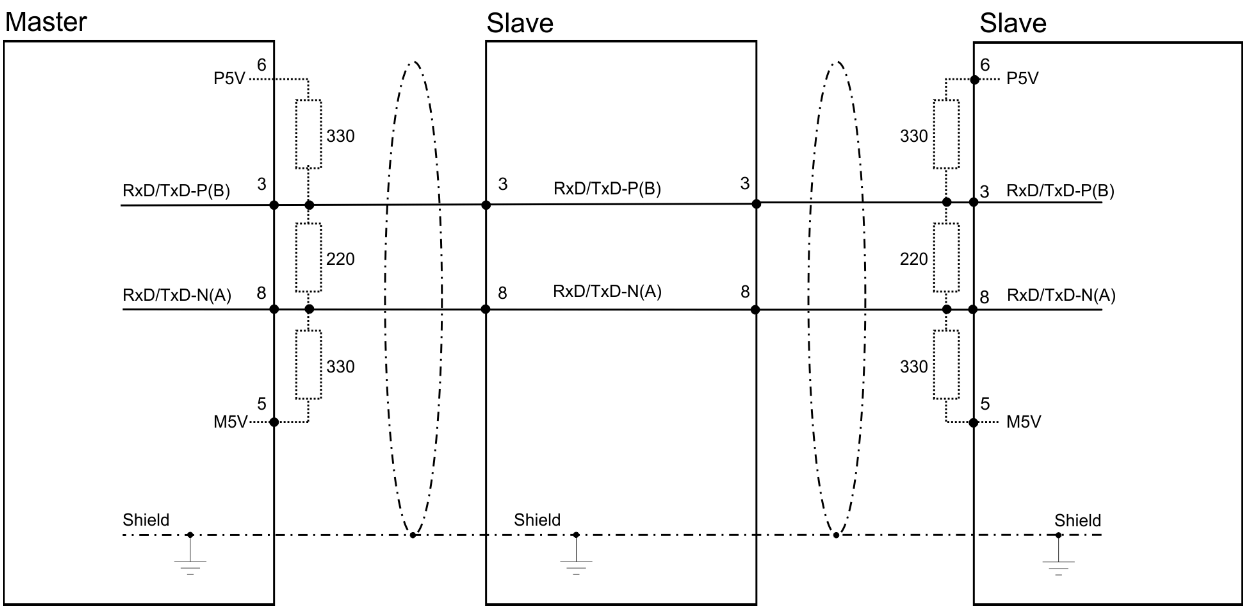
- A PROFIBUS DP network may only be built up in linear structure.
- PROFIBUS DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- PROFIBUS supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- The max. segment length depends on the transfer rate:
  9.6 ... 187.5bit/s → 1000m
  500kbit/s → 400m
  1.5Mbit/s → 200m
  3 ... 12Mbit/s → 100m
- Max. 10 segments may be built up. The segments are connected via repeaters. Every repeater counts for one participant.
- The bus respectively a segment is to be terminated at both ends.
- All participants are communicating with the same transfer rate. The slaves adjust themselves automatically on the transfer rate.

Transfer medium

- As transfer medium PROFIBUS uses an isolated twisted-pair cable based upon the RS485 interface.
- The RS485 interface is working with voltage differences. Though it is less irritable from influences than a voltage or a current interface. You are able to configure the network as well linear as in a tree structure.
- Max. 32 participants per segment are permitted. Within a segment the members are linear connected. The segments are connected via repeaters. The maximum segment length depends on the transfer rate.
- PROFIBUS DP uses a transfer rate between 9.6kbit/s and 12Mbit/s, the slaves are following automatically. All participants are communicating with the same transfer rate.
- The bus structure under RS485 allows an easy connection res. disconnection of stations as well as starting the system step by step. Later expansions don't have any influence on stations that are already integrated. The system realizes automatically if one partner had a fail down or is new in the network.

Bus connection

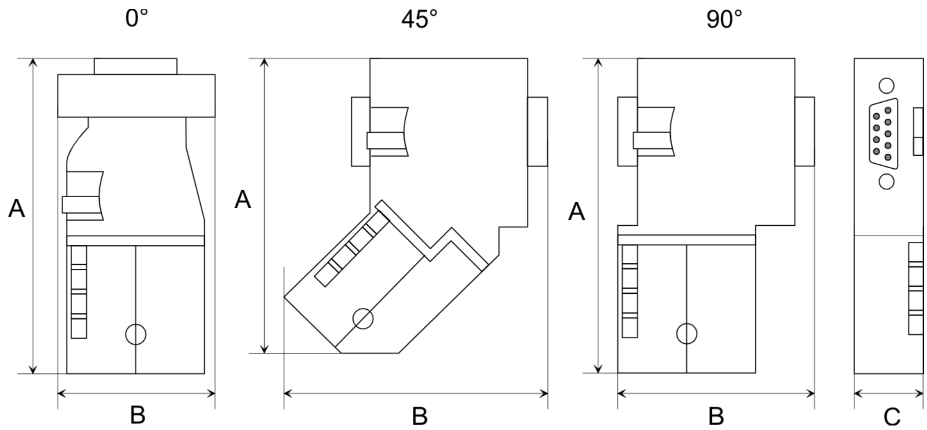
The following picture illustrates the terminating resistors of the respective start and end station.



The PROFIBUS line has to be terminated with its ripple resistor. Please make sure to terminate the last participants on the bus at both ends by activating the terminating resistor.

EasyConn bus connector

In PROFIBUS all participants are wired parallel. For that purpose, the bus cable must be feed-through. Via the order number 972-0DP10 you may order the bus connector "Easy-Conn" from Yaskawa. This is a bus connector with switchable terminating resistor and integrated bus diagnostic.



Dimensions in mm	0°	45°	90°
A	64	61	66
B	34	53	40
C	15.8	15.8	15.8

To connect this EasyConn plug, please use the standard PROFIBUS cable type A (EN50170). Starting with release 5 you also can use highly flexible bus cable:

Lapp cable order no: 2170222, 2170822, 2170322.

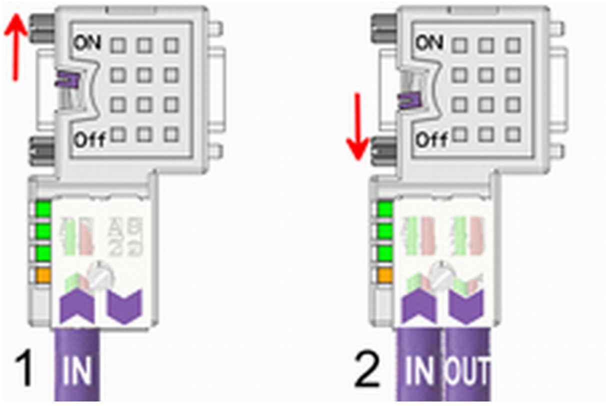
With the order no. 905-6AA00 Yaskawa offers the "EasyStrip" de-isolating tool that makes the connection of the EasyConn much easier.

Dimensions in mm

Termination with "Easy-Conn"

The "EasyConn" bus connector is provided with a switch that is used to activate a terminating resistor.

Wiring



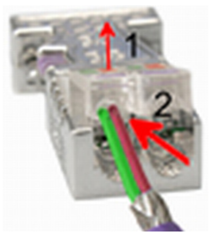
[1] 1./last bus participant
[2] further participants

CAUTION!
The terminating resistor is only effective, if the connector is installed at a bus participant and the bus participant is connected to a power supply.

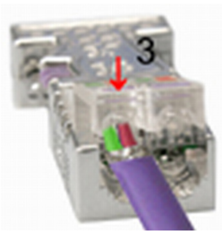
The tightening torque of the screws to fix the connector to a device must not exceed 0.02Nm!

A complete description of installation and deployment of the terminating resistors is delivered with the connector.

Assembly



1. Loosen the screw.
2. Lift contact-cover.
3. Insert both wires into the ducts provided (watch for the correct line colour as below!)
4. Please take care not to cause a short circuit between screen and data lines!



5. Close the contact cover.
6. Tighten screw (max. tightening torque 0.08Nm).

The green line must be connected to A, the red line to B!

12.8.8 Commissioning and Start-up behavior

Start-up on delivery

In delivery the CPU is overall reset. The PROFIBUS part is deactivated and its LEDs are off after Power ON.

Online with bus parameter without slave project

The DP master can be served with bus parameters by means of a hardware configuration. As soon as these are transferred the DP master goes online with his bus parameter. This is shown by the RUN LED. Now the DP master can be contacted via PROFIBUS by means of his PROFIBUS address. In this state the CPU can be accessed via PROFIBUS to get configuration and DP slave project.

Slave configuration

If the master has received valid configuration data, he switches to *Data Exchange* with the DP Slaves. This is indicated by the DE-LED.

CPU state controls DP master

After PowerON respectively a receipt of a new hardware configuration the configuration data and bus parameter were transferred to the DP master. Dependent on the CPU state the following behavior is shown by the DP master:

Master behavior at CPU STOP

- The global control command "Clear" is sent by the master. Then the DP slaves disable the outputs.
- DP slaves with fail safe mode were provided with output telegram length "0".
- DP slaves without fail safe mode were provided with the whole output telegram but with output data = 0.
- The input data of the DP slaves were further cyclically transferred to the input area of the CPU.

Master behavior at CPU RUN

- The global control command "Operate" is sent by the master. Then the DP slaves enable the outputs.
- Every connected DP slave is cyclically attended with an output telegram containing recent output data.
- The input data of the DP slaves were cyclically transferred to the input area of the CPU.

12.9 Siemens SIMATIC Manager - Deployment PROFINET I-Device

Chap. 10 'Deployment PG/OP communication - PROFINET I-Device' page 277

12.10 Siemens SIMATIC Manager - Deployment EtherCAT

Precondition

The configuration of the EtherCAT masters happens in the Siemens SIMATIC Manager by means of a virtual PROFINET IO device *'EtherCAT network'*. The *'EtherCAT network'* is to be installed in the hardware catalog by means of the GSDML and can be configured with the Yaskawa tool *SPEED7 EtherCAT Manager*.

The following preconditions must be fulfilled for the configuration of the EtherCAT master:

- GSDML for *'EtherCAT network'* is installed
- *SPEED7 EtherCAT Manager* for EtherCAT configuration is installed

Installing the IO device EtherCAT network

The installation of the PROFINET IO devices *'EtherCAT Network'* happens in the hardware catalog with the following approach:

1. Go to the *'Download Center'* of www.yaskawa.eu.com
2. Download under *'GSDML EtherCAT'* the GSDML file for your EtherCAT master.
3. Extract the files into your working directory.
4. Start the Siemens hardware configurator.

5. Close all the projects.
6. Select *'Options → Install new GSD file'*.
7. Navigate to your working directory and install the according GSDML file.
   ⇨ After the installation the *'EtherCAT Network'* can be found at *'PROFINET IO → Additional field devices → I/O → VIPA EtherCAT System'*.

Installing the *SPEED7 EtherCAT Manager*

The configuration of the PROFINET IO device *'EtherCAT Network'* happens by means of the Yaskawa *SPEED7 EtherCAT Manager*. This may be found in the *'Download Center'* of www.yaskawa.eu.com at *'EtherCAT Manager'*.

The installation happens with the following proceeding:

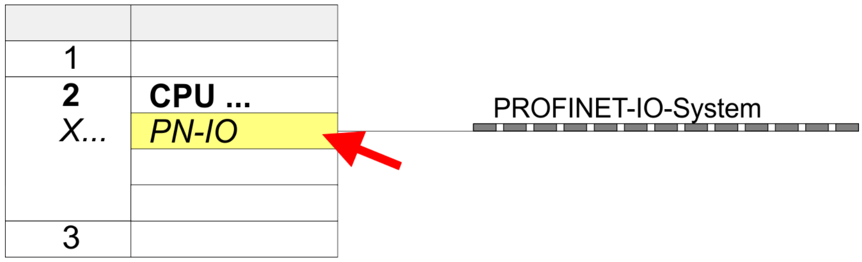
1. Close the Siemens SIMATIC Manager.
2. Go to the *'Download Center'* of www.yaskawa.eu.com
3. Load the *EtherCAT Manager* and unzip it on your PC.
4. For installation start the file EtherCATManager_v... .exe.
5. Select the language for the installation.
6. Accept the licensing agreement.
7. Select the installation directory and start the installation.
8. After installation you have to reboot your PC.
   ⇨ The *SPEED7 EtherCAT Manager* is installed and can now be called via the context menu of the Siemens SIMATIC Manager.

Configuration of the CPU

Slot	Module
1	
2	**CPU 31...**
X1	*MPI/DP*
X2	*PN-IO*
X2...	*Port 1*
X2...	*Port 2*
3	

To be compatible with the Siemens SIMATIC manager the following steps should be executed:

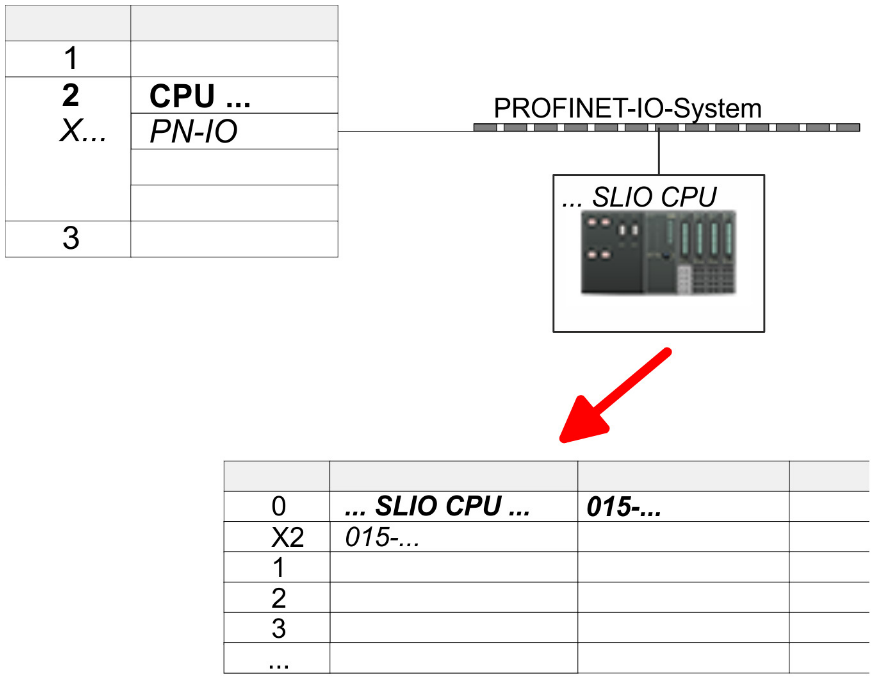
1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'* number 2 the CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).
4. The integrated PROFIBUS DP master (jack X3) is to be configured and connected via the sub module *'X1 MPI/DP'*.
5. The integrated EtherCAT master is to be configured via the sub module *'X2 PN-IO'* as a virtual PROFINET network.
6. Click at the sub module *'PN-IO'* of the CPU.
7. Select *'Context menu → Insert PROFINET IO System'*.



8. Create with [New] a new sub net and assign valid address data

9. Click at the sub module *'PN-IO'* of the CPU and open with *'Context menu ➔ Properties'* the properties dialog.

10. Enter at *'General'* a *'Device name'*. The device name must be unique at the Ethernet subnet.

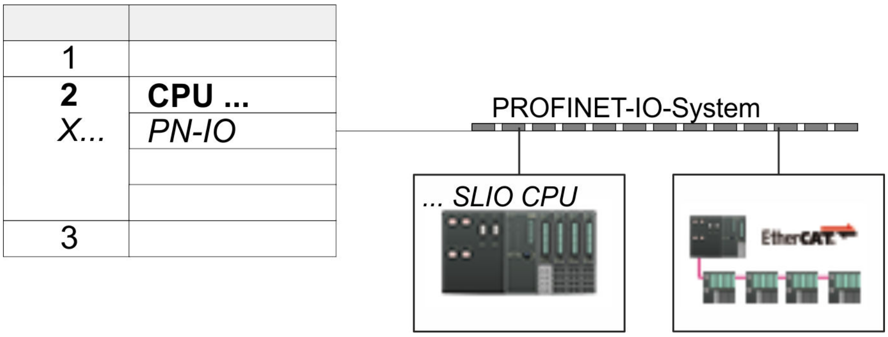


11. Navigate in the hardware catalog to the directory *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ ... SLIO System'* and connect the IO device *'015-CFFNR00 CPU'* to your PROFINET system.

    ⇨ In the Device overview of the PROFINET IO device *'... SLIO CPU'* the CPU is already placed at slot 0. From slot 1 you can place your System SLIO modules.

12. Please note that in the *SPEED7 EtherCAT Manager* and in the hardware configurator the automatic assignment of the device number starts with 1. So to avoid overlapping of the device numbers you have to change these in the hardware configurator to a large value. For this open the object properties of the EtherCAT Systems and set at *'Device number'* e.g. the value 100.

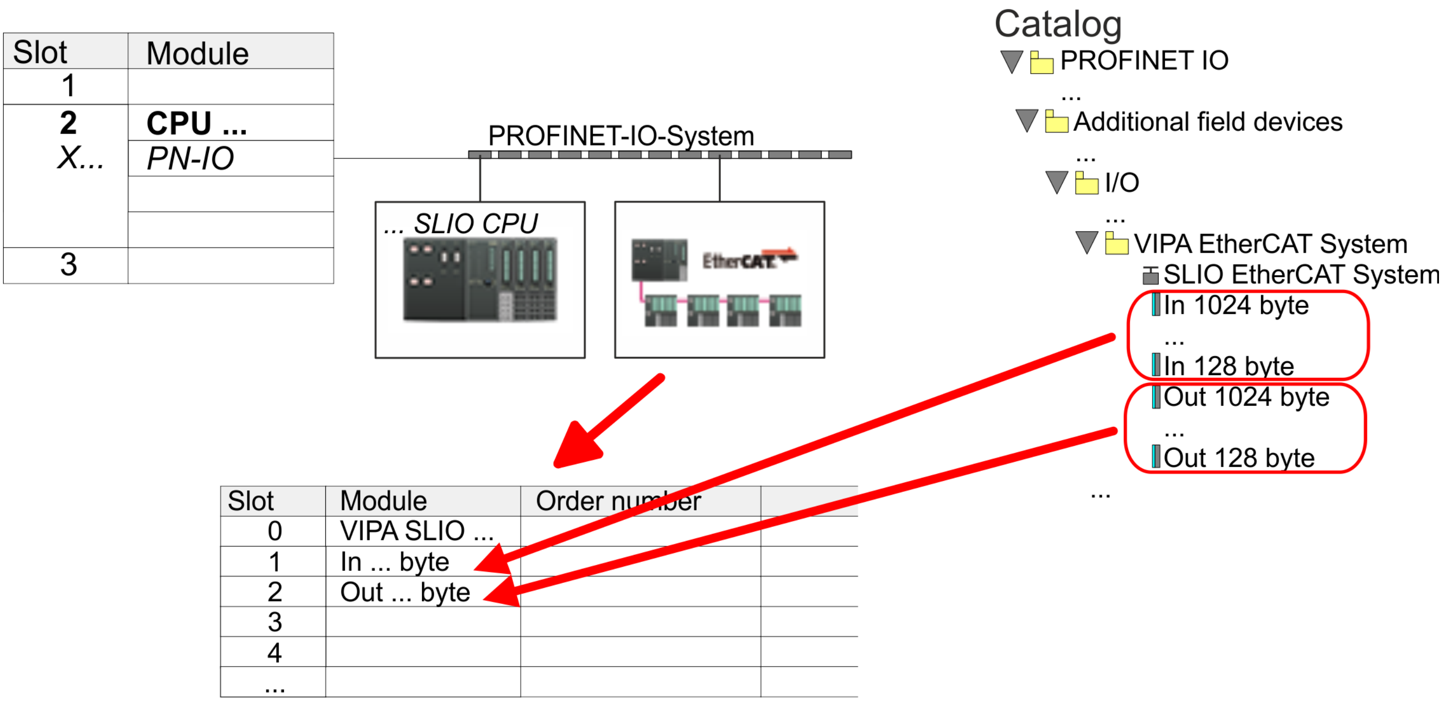
Configuration EtherCAT master



1. Navigate in the hardware catalog to the directory *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ VIPA EtherCAT System'* and connect the IO device *'SLIO EtherCAT System'* to your PROFINET system.
2. Please note that in the *SPEED7 EtherCAT Manager* and in the hardware configurator the automatic assignment of the device number starts with 1. So to avoid overlapping of the device numbers you have to change these in the hardware configurator to a large value. For this open the object properties of the EtherCAT Systems and set at *'Device number'* e.g. the value 101.
3. Click at the inserted IO device *'EtherCAT Network'* and define the areas for in and output by drag and dropping the according *'Out'* or *'In'* area to a slot.

   Here the following rules must be observed:

   - Input and output areas can be mixed.
   - You have a maximum of 4096byte EtherCAT process data for input and output respectively.
   - Data must be consistent in the Siemens hardware configurator, i.e. with PROFINET the maximum number of bytes may not be exceeded. Otherwise you have to connect another *'SLIO EtherCAT System'* to your PROFINET system. In the *SPEED7 EtherCAT Manager* all the areas are automatically detected and combined



4. Select *'Station* ➔ *Save and compile'*

⇨ Now you can configure your EtherCAT system with the *SPEED7 EtherCAT Manager*.

> *Before calling the SPEED7 EtherCAT Manager you have always to save you project with 'Station ➔ Save and compile'.*

5. Click at an inserted IO device *'EtherCAT Network'* and select *'Context menu* ➔ *Start Device-Tool* ➔ *SPEED7 EtherCAT Manager'*.

⇨ The *SPEED7 EtherCAT Manager* opens. Here you can configure the EtherCAT master system.

More information about the usage of the *SPEED7 EtherCAT Manager* may be found in the according manual or online help.

6. By closing the *SPEED7 EtherCAT Manager* the EtherCAT configuration is taken to the project and the *SPEED7 EtherCAT Manager* is closed. You can always edit your EtherCAT configuration in the *SPEED7 EtherCAT Manager*, since the configuration is stored in your project.

7. Choose the Siemens SIMATIC manager and transfer your project into the CPU.

The transfer can only be done by the Siemens SIMATIC Manager - not hardware configurator!

> *Since slave and module parameters are transmitted by means of SDO respectively SDO Init command, the configuration remains active, until a power cycle is performed or new parameters for the same SDO objects are transferred.*
>
> ***With an overall reset the slave and module parameters are not reset!***

13 Configuration with TIA Portal

13.1 TIA Portal - Work environment

13.1.1 General

General

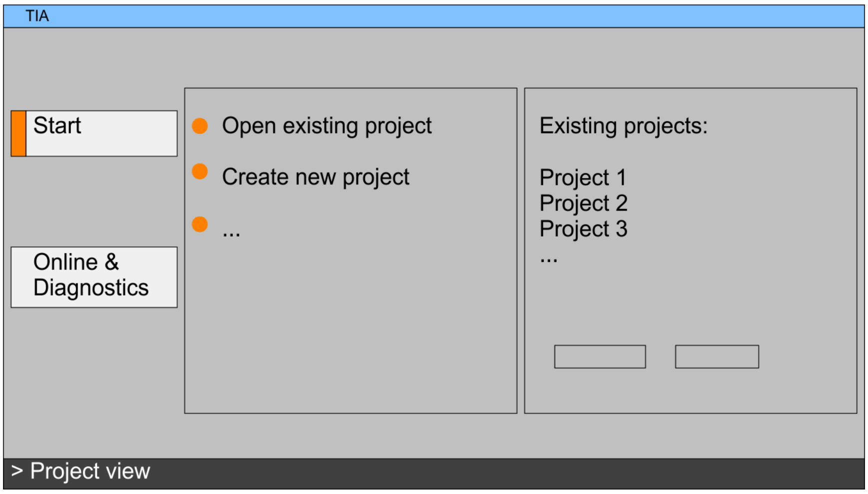
In this chapter the project engineering of the CPU in the Siemens TIA Portal is shown. Here only the basic usage of the Siemens TIA Portal together with a CPU is shown. Please note that software changes can not always be considered and it may thus be deviations to the description. TIA means **T**otally **i**ntegrated **A**utomation from Siemens. Here your PLCs may be configured and linked. For diagnostics online tools are available.

> *Information about the Siemens TIA Portal can be found in the online help respectively in the according online documentation.*

Starting the TIA Portal

To start the Siemens TIA Portal with Windows select *'Start ➔ Programs ➔ Siemens Automation ➔ TIA ...'*

Then the TIA Portal opens with the last settings used.



Exiting the TIA Portal

With the menu *'Project ➔ Exit'* in the *'Project view'* you may exit the TIA Portal. Here there is the possibility to save changes of your project before.

13.1.2 Work environment of the TIA Portal

Basically, the TIA Portal has the following 2 views. With the button on the left below you can switch between these views:

Portal view

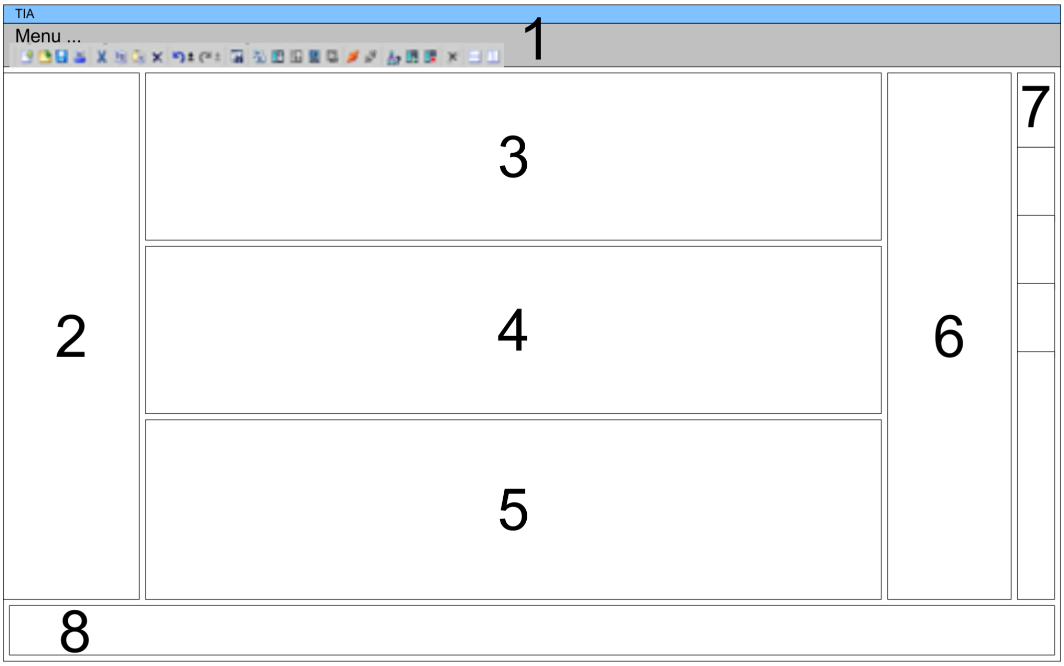
The *'Portal view'* provides a "task oriented" view of the tools for processing your project. Here you have direct access to the tools for a task. If necessary, a change to the Project view takes place automatically for the selected task.

Project view

The *'Project view'* is a "structured" view to all constituent parts of your project.

Areas of the Project view

The Project view is divided into the following areas:



1 Menu bar with toolbars
2 Project tree with Details view
3 Project area
4 Device overview of the project respectively area for block programming
5 Properties dialog of a device (parameter) respectively information area
6 Hardware catalog and tools
7 "Task-Cards" to select hardware catalog, tasks and libraries
8 Jump to Portal or Project view

13.2 TIA Portal - Functional limitations

Limitation of performance data

Please note that the performance data of the CPU is limited to the performance data of the Siemens CPU used for the configuration.

No *'Upload device as new station...'*

Due to the system *'Upload device as new station...'* is currently not supported. Instead, use the *backup* and *restore* functions in the Siemens TIA Portal:

1. To *backup* an online connected CPU, select *'Online ➔ Backup from online device'*.
   ⇨ A *backup* object is created and stored in the *project navigation* at *'Online backups'*. The backup contains all blocks of the project and the current device status.
2. To *restore* into an online connected CPU it must be overall reset first. Then click in the *project navigation* under *'Online backups'* on the created *backup* and select *'Context menu ➔ Download to device'*.
   ⇨ The backup data is transferred online to the CPU.

No online blocks

Due to the system, online blocks of CPUs connected via *'Accessible devices'* are currently not listed.

13.3 TIA Portal - Hardware configuration - CPU

Overview

The hardware configuration of the CPU and its plugged modules happens in the Siemens TIA Portal by means of a virtual PROFINET IO device. For the PROFINET interface is standardized software sided, the functionality is guaranteed by including a GSDML file into the Siemens TIA Portal.

The hardware configuration of the CPU is divided into the following parts:

- Installation PROFINET IO device *'VIPA SLIO System'*
- Configuration Siemens CPU
- Connection System SLIO CPU as PROFINET IO device

Installation PROFINET IO device *'VIPA SLIO System'*

The installation of the PROFINET IO devices *'VIPA SLIO System'* happens in the hardware catalog with the following approach:

1. Go to the *'Download Center'* of www.yaskawa.eu.com.
2. Load under *'GSDML SLIO'* the according file for your System SLIO.
3. Extract the file into your working directory.
4. Start the Siemens TIA Portal.
5. Close all the projects.
6. Switch to the *Project view*.
7. Select *'Options ➔ Install general station description file (GSD)'*.
8. Navigate to your working directory and install the according GSDML file.

   ⇨ After the installation the hardware catalog is refreshed and the Siemens TIA Portal is finished.

   After restarting the Siemens TIA Portal the according PROFINET IO device can be found at *'Other field devices ➔ PROFINET ➔ IO ➔ VIPA ... ➔ ... SLIO System'*.

> *Thus, the components can be displayed, you have to deactivate the "Filter" of the hardware catalog.*

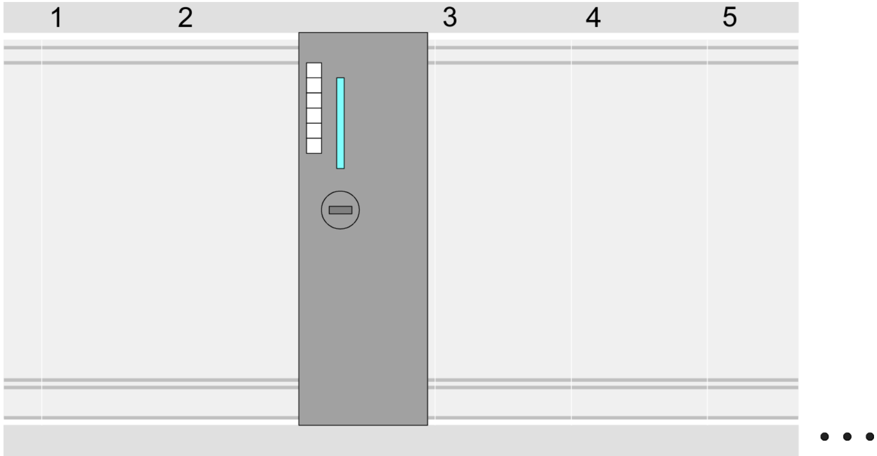
Configuration Siemens CPU

In the Siemens TIA Portal, the CPU from is to be configured as CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2) from Siemens.

1. Start the Siemens TIA Portal.
2. Create a new project in the *Portal view* with *'Create new project'*.
3. Switch to the *Project view*.
4. Click in the *Project tree* at *'Add new device'*.
5. Select the following CPU in the input dialog:

   SIMATIC S7-300 > CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2)

   ⇨ The CPU is inserted with a profile rail.



Device overview

Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	

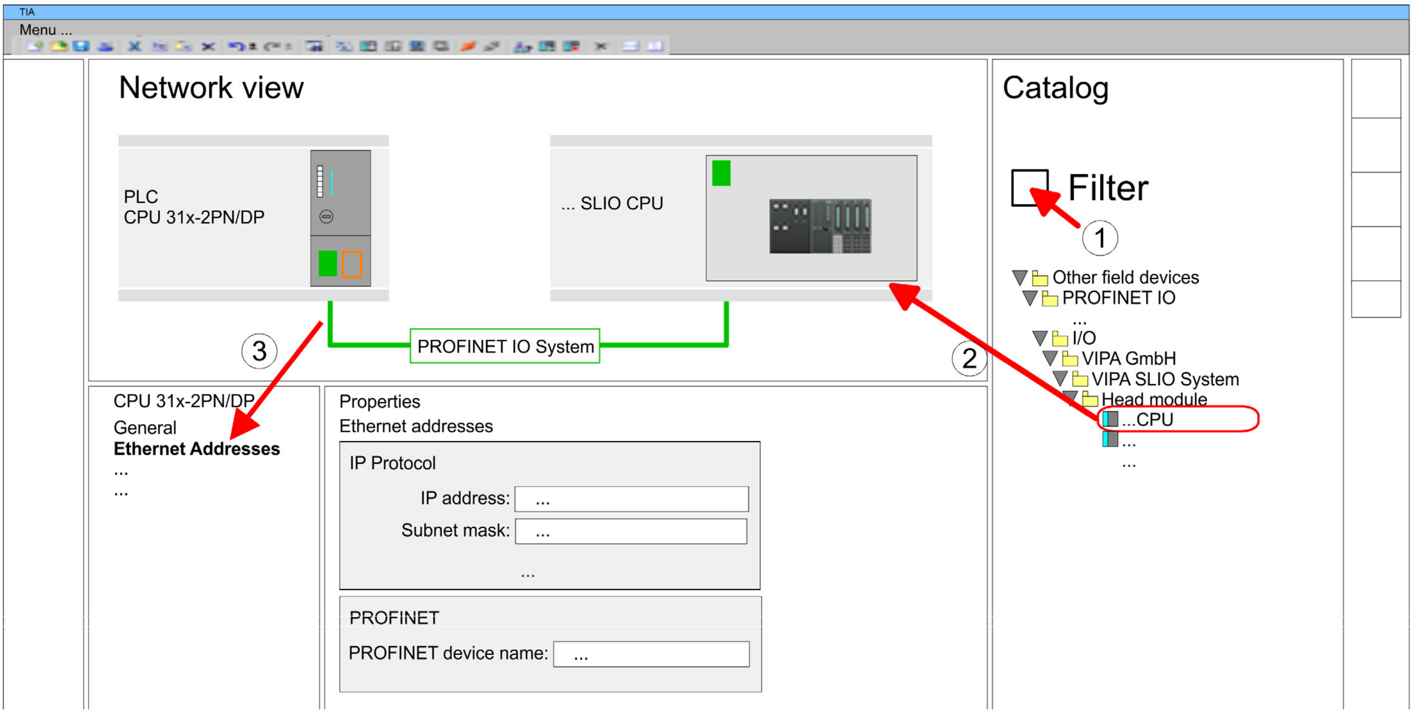
Setting standard CPU parameters

Since the CPU is configured as Siemens CPU, so the setting of the non-product specific parameters takes place via the Siemens CPU. For parametrization click in the *Project area*respectively in the *Device overview* at the CPU part. Then the parameters of the CPU part are shown in the *Properties dialog*. Here you can make your parameter settings. *Chap. 4.8.1 'Parameter CPU' page 80*

Connection System SLIO CPU as PROFINET IO device

1. Switch in the *Project area* to *'Network view'*.
2. After installing the GSDML the IO device for the System SLIO CPU may be found in the hardware catalog at *'Other field devices ➔ PROFINET ➔ IO ➔ VIPA ... ➔ ... SLIO System'*. Connect the slave system to the CPU by dragging&dropping it from the hardware catalog to the *Network view* and connecting it via PROFINET to the CPU.
3. Click in the *Network view* at the PROFINET part of the Siemens CPU and enter a valid IP address data in *'Properties'* at *'Ethernet address'* in the area *'IP protocol'*.
4. Enter at *'PROFINET'* a *'PROFINET device name'*. The device name must be unique at the Ethernet subnet.

> *Please leave 'Send clock' at 'Advanced options ➔ Realtime settings ➔ IO communication' at 1ms, otherwise this leads to a configuration error!*



5. Select in the *Network view* the IO device *'... SLIO CPU...'* and switch to the *Device overview*.

⇨ In the *Device overview* of the PROFINET IO device *'... SLIO CPU'* the CPU is already placed at slot 0. From slot 1 you can place your System SLIO modules.

Setting product specific CPU parameters

For parametrization click at the CPU at slot 0 in the *Device overview* of the PROFINET IO device *'VIPA SLIO System'*. Then the parameters of the CPU part are shown in the *Properties dialog*. Here you can make your parameter settings. *Chap. 4.8.1 'Parameter CPU' page 80*

13.4 TIA Portal - Hardware configuration - Ethernet PG/OP channel

Overview

Please note!

– *At the first commissioning respectively after a reset to factory setting the Ethernet interface has no IP address.*
– *For online access, you have to assign valid IP address data to it by means of "Initialization".*
– *After initialization, you can transfer the IP address data to your project.*

The CPU has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU.

- The Ethernet PG/OP channel (X1/X5) is designed as switch. This enables PG/OP communication via the connections X1 and X5.
- Configurable connections are possible.
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040
- Via the Ethernet PG/OP channel, you have access to:
  – Device website, where you can find information on firmware status, connected peripherals, current cycle times, etc.
  – *OPC UA* project, which is to be created in the *OPC UA Configurator*.
  – *WebVisu* project, which is to be created in the *SPEED7 Studio*.

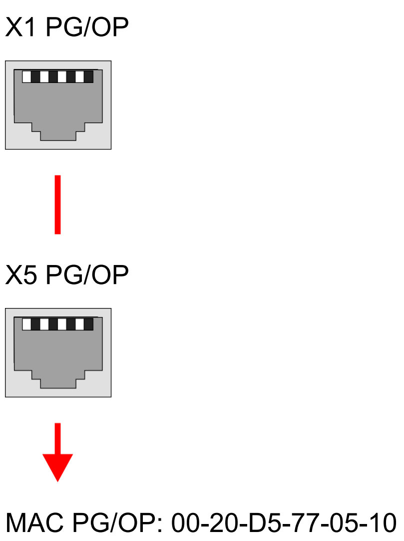
Assembly and commissioning

1. Install your System SLIO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the one of the Ethernet jacks (X1, X5) of the Ethernet PG/OP channel to Ethernet.
4. Switch on the power supply.

   ⇨ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

"Initialization"

The assignment of IP address data takes place via the MAC address. The IP address of your Ethernet PG/OP channel for the interfaces X1 and X5 can be found on the front of your CPU with the name "MAC PG/OP: ...".

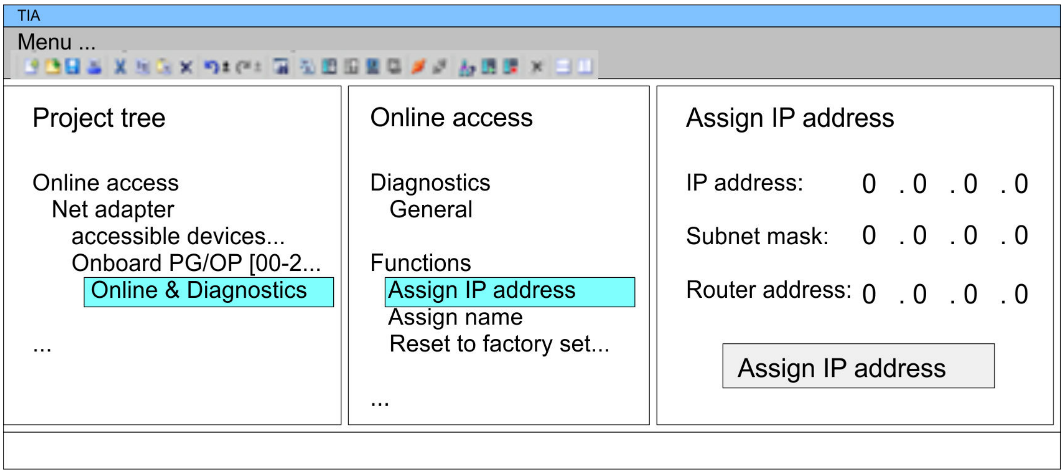


Assign IP address parameters

The assignment of the IP address data happens online in the Siemens TIA Portal with the following proceeding:

1. Start the Siemens TIA Portal.
2. Switch to the *'Project view'*.
3. Click in the *'Project tree'* at *'Online access'* and choose here by a doubleclick your network card, which is connected to the Ethernet PG/OP channel.
4. To get the stations and their MAC address, use the *'Accessible device'*. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".
5. Choose from the list the module with the known MAC address (Onboard PG/OP [MAC address]) and open with "Online & Diagnostics" the diagnostics dialog in the Project area.
6. Navigate to *Functions* > *Assign IP address*. Type in the IP configuration like IP address, subnet mask and gateway.
7. Confirm with [Assign IP configuration].

   ⇨ Directly after the assignment the Ethernet PG/OP channel is online reachable using the set IP address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.



Due to the system you may get a message that the IP address could not be assigned. This message can be ignored.

Take IP address parameters in project

Please note the following restrictions:

- Address overlaps are not recognized in the Siemens TIA Portal.
- For PROFINET devices only the address range 0 ... 1023 is available.
- The addresses of the PROFINET devices are not checked with the address space of the CPU from the Siemens TIA Portal for address overlaps.

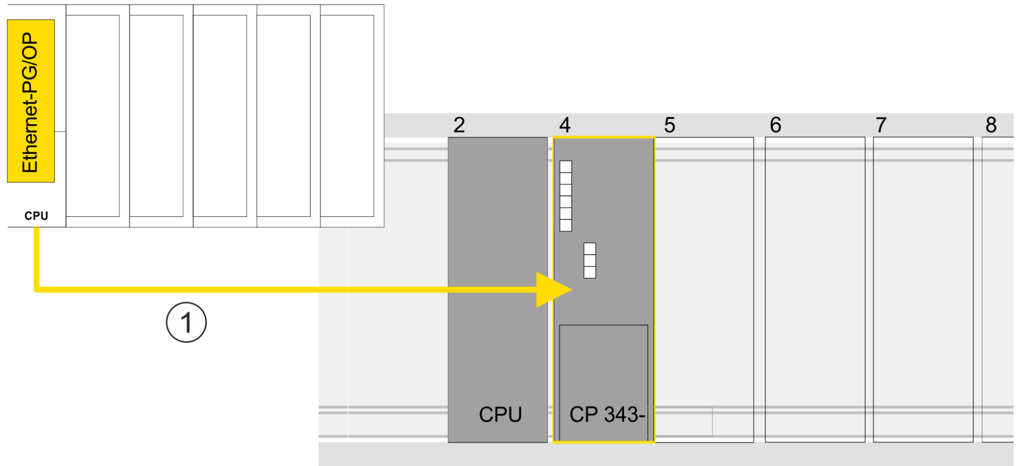
1. Open your project.
2. If not already done, configure in the *'Device configuration'* a Siemens CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2).

3. As Ethernet PG/OP channel place at slot 4 the Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).

> **CAUTION!**
> Please configure the diagnostic addresses of the CP343-1EX30 for *'PN-IO'*, *'Port1'* and *'Port2'* so that no overlaps occur in the periphery input area. Otherwise your CPU can not start-up and you receive the diagnostic entry 0xE904. Address overlaps are not recognized in the Siemens TIA Portal.

4. Open the "Property" dialog by clicking on the CP 343-1EX30 and enter for the CP at "Properties" at "Ethernet address" the IP address data, which you have assigned before.

5. Transfer your project.



1 Ethernet PG/OP channel

Device overview

Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	
CP 343-1		4		CP 343-1	
...		...		...	

13.4.1 Time-of-day synchronization

NTP method

In the NTP mode (**N**etwork **T**ime **P**rotocol) the module sends as client time-of-day queries at regular intervals to all configured NTP servers within the sub net. You can define up to 4 NTP server. Based on the response from the servers, the most reliable and most exact time-of-day is determined. Here the time with the lowest *stratum* is used. *Stratum 0* is the time standard (atomic clock). *Stratum 1* are directly linked to this NTP server. Using the NTP method, clocks can be synchronized over subnet boundaries. The configuration of the NTP servers is carried out in the Siemens TIA Portal via the CP, which is already configured.

1. In the *'Device configuration'*, click the CP 343-1EX30.
2. Click on *'PROFINET interface'* in the *'Device overview'*.
3. In the *'Properties'*, select *'Time-of-day synchronization'*.
4. Enable the NTP method by enabling *'Activate time-of-day synchronization'* and selecting *'NTP'* at *'Method'*.
5. Add the appropriate NTP servers by specifying their IP addresses.
6. Select your *'Time zone'*. In the NTP method, UTC (**U**niversal **T**ime **C**oordinated) is generally transmitted; this corresponds to GMT (Greenwich Mean Time). By configuring the local time zone, you can set a time offset to UTC.
7. Set the *'Update interval'* you want. Within this interval, the time of the module is synchronized once.
8. Save and transfer your project to the CPU.

   ⇨ After transmission, the NTP time is requested by each configured time server and the best response for the time synchronization is used.

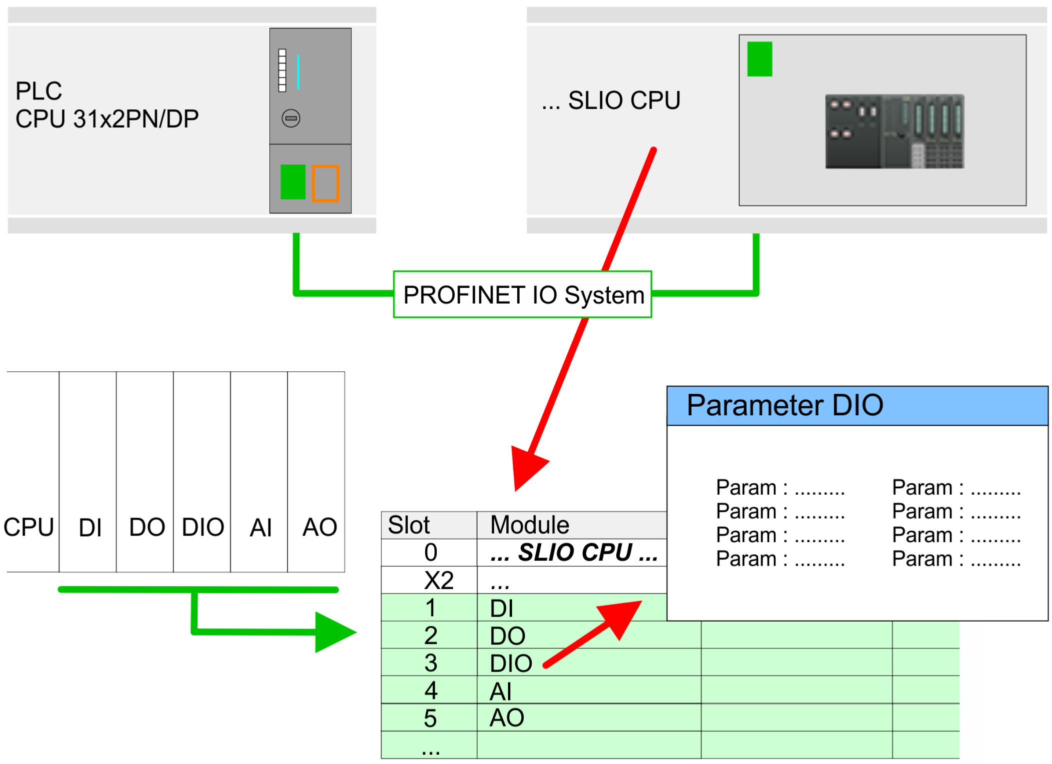
Please note that although the time zone is evaluated, an automatic changeover from winter to summer time is not supported. Industrial systems with time-of-day synchronization should always be set in accordance to the winter time.

With the FC 61 you can determine the local time in the CPU. More information about the usage of this block may be found in the manual "SPEED7 Operation List".

13.5 TIA Portal - Hardware configuration - I/O modules

Hardware configuration of the modules

Starting with slot 1 place in the *Device overview* of the PROFINET IO device *'... SLIO CPU'* your System SLIO modules in the plugged sequence. For this drag from the hardware catalog the corresponding module to the corresponding position in the *Device overview*.



Parametrization

To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU. For parametrization click in the *Device overview* at the module you want to parametrize. Then the parameters of the module are shown in the *Properties* dialog. Here you can make your parameter settings.

13.6 TIA Portal - Deployment PG/OP communication - PROFINET I-Device

- *With firmware version V2.4.0, there is a PROFINET I-Device available via the Ethernet PG/OP channel.*
- *As soon as you use the PROFINET functionality via the Ethernet PG/OP channel, this affects the performance and response time of your system and due to the system the cycle time of the OB 1 is extended by 2ms.*

13.6.1 Deployment as PROFINET I-Device

13.6.1.1 Steps of configuration

Functionality

> ***Range of functions***
>
> *Please regard that the PROFINET IO controller supports only the PROFINET functions, which are described in this manual, even if the Siemens CPU, which is used for configuration, offers further functions! To use some described PROFINET functions, it is necessary to deploy another Siemens CPU for configuration. Here, however, is pointed to explicitly.*

The *I-Device* (Intelligent IO device) functionality of a CPU allows data to be exchanged with an IO controller, which are preprocessed by the CPU. In this case, the I-Device is connected as an IO device to a higher-level IO controller. The process values, recorded in central or decentralized periphery, can be preprocessed via a user program and made available to the higher-level PROFINET IO controller by means of PROFINET.

- The configuration of the integrated PROFINET IO controller of the CPU as an I-Device is made via a virtual PROFINET devices, which is to be installed by means of a product specific GSDML in the hardware catalog.
- The communication takes place via input/output areas, which are defined in the I-Device.
- The size of the areas for input and output data is max. 768byte.
- The I-Device is made available to a deterministic PROFINET IO system via a PROFINET IO interface and thus supports the real-time communication *Real-Time* .
- The I-Device functionality meets the requirements of the RT class I (A) and corresponds to the PROFINET specification version V 2.3.
- The configuration of a PROFINET CPU as an IO controller and at the same time as an I-Device is possible. The influence of the I-Device configuration on the system limits or performance of the PROFINET controller is equated with that of a device. This means that when the IO controller and I-Device are used at the same time on the PROFINET controller, the I-Device is to be regarded as an additional device for determining the system limits.
- In order for the higher-level IO controller to communicate with the I-Device, the following must be observed:
  - The device name of the PROFINET controller of the I-Device must match the device name of the I-Device at the higher-level IO controller.
  - When using the Siemens SIMATIC Manager or the TIA Portal, in order to avoid name conflicts, I-Device and IO controller must be configured in different logical networks.

> *The PROFINET IO controller supports a maximum IO block size of 512 bytes (consistent).*

Configuration

The configuration of the PROFINET IO controller as I-Device should be done by the following procedure:

1. Installation of the GSDML files
2. Configuration as I-Device
3. Configuration in the higher-level IO controller

Transfer I-Device from *SPEED7 Studio*

If the configuration of the *I-Device* from the *SPEED7 Studio* is to be applied to the IO controller of a third-party system, then you must export from the *SPEED7 Studio* the according GSDLM file and import it into the IO controller of the third-party system.

1. Start the *SPEED7 Studio* with your PROFINET project.
2. Click at the CPU in *'Devices and networking'* and select *'Context menu ➔ Create GSDML file'*. Specify an *'Export path'* and an unique *'Device name'*.
   ⇨ The GSDML file is created and exported. Import this GSDML file into your third-party system.

13.6.1.2 Installation of the GSDML files

The following GSDML files are required for configuring the integrated PROFINET IO controller of the CPU as I-Device:

- GSDML for I-Device
- GSDML for I-Device at IO controller

Proceeding

The installation of the PROFINET IO device *'SLIO CPU'* happens in the hardware catalog with the following approach:

1. Go to the *'Download Center'* of www.yaskawa.eu.com.
2. Load under *'GSDML SLIO'* the according file for your System SLIO.
3. Extract the file into your working directory.
4. Start the Siemens TIA Portal.
5. Close all the projects.
6. Switch to the *Project view*.
7. Select *'Options ➔ Install general station description file (GSD)'*.
8. Navigate to your working directory and install the according GSDML file.
   ⇨ After the installation the hardware catalog is refreshed and the Siemens TIA Portal is closed.

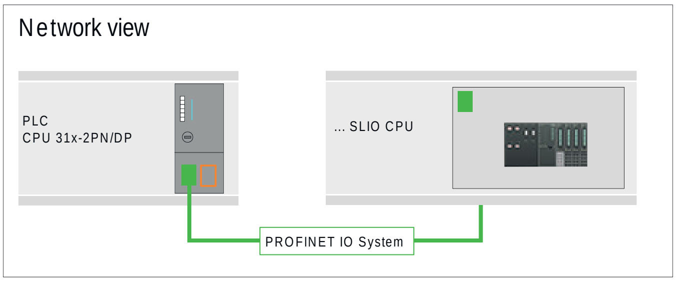
   After restarting the Siemens TIA Portal following virtual devices can be found in the Hardware catalog at *'PROFINET IO ➔ Other field devices ➔ VIPA ... ➔ ... SLIO System ➔ SLIO I-Device'*:

   - PN I-Device for CPU
     – This allows you to configure the Input/output areas in the I-Device of the CPU.
   - PN I-Device config 015-CEFNR00 for higher-level CPU
     – This allows you to connect the I-Device CPU to the higher-level IO controller.

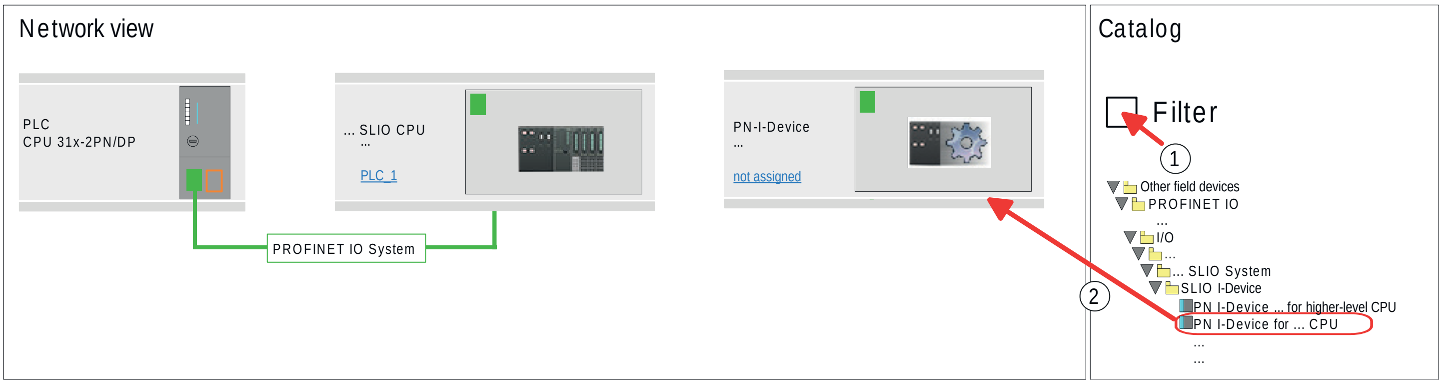
Thus, the components can be shown, you have to deactivate the "Filter" of the hardware catalog.

13.6.1.3 Configuration as I-Device

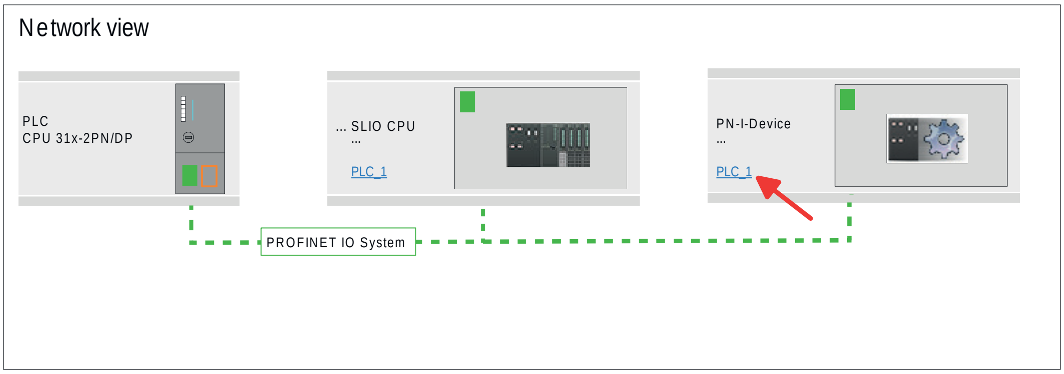
It is assumed that a hardware configuration of the CPU exists. *Chap. 13.3 'TIA Portal - Hardware configuration - CPU' page 326*



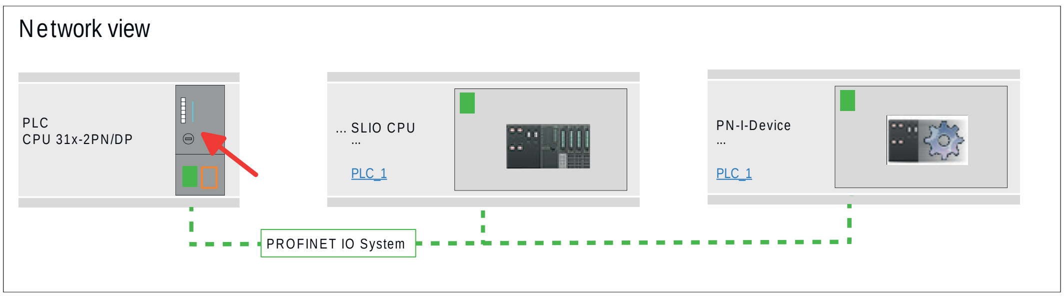
1. After installing the GSDML the *'PN I-Device for ... CPU'* can be found in the hardware catalog at *'Other field devices ➔ PROFINET IO ➔ I/O ➔ VIPA ... ➔ ... SLIO System ➔ SLIO I-Device'*. Drag *'PN I-Device for ... CPU'* from the hardware catalog to *Network view*.



2. To connect to the CPU, click at *'not assigned'* and select the PROFINET interface of the CPU.



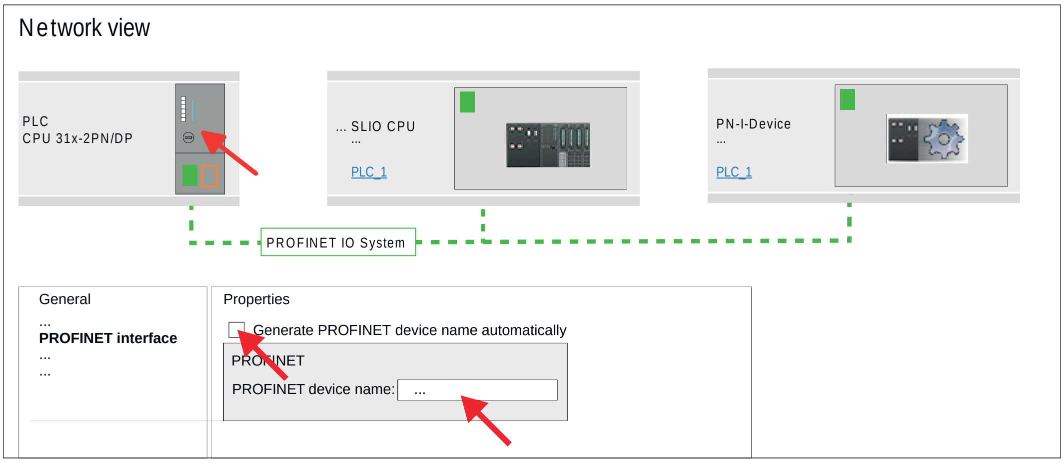
3. Click at the CPU and select *'Context menu ➔ Properties'*.



⇨ The properties dialog of the CPU is opened.

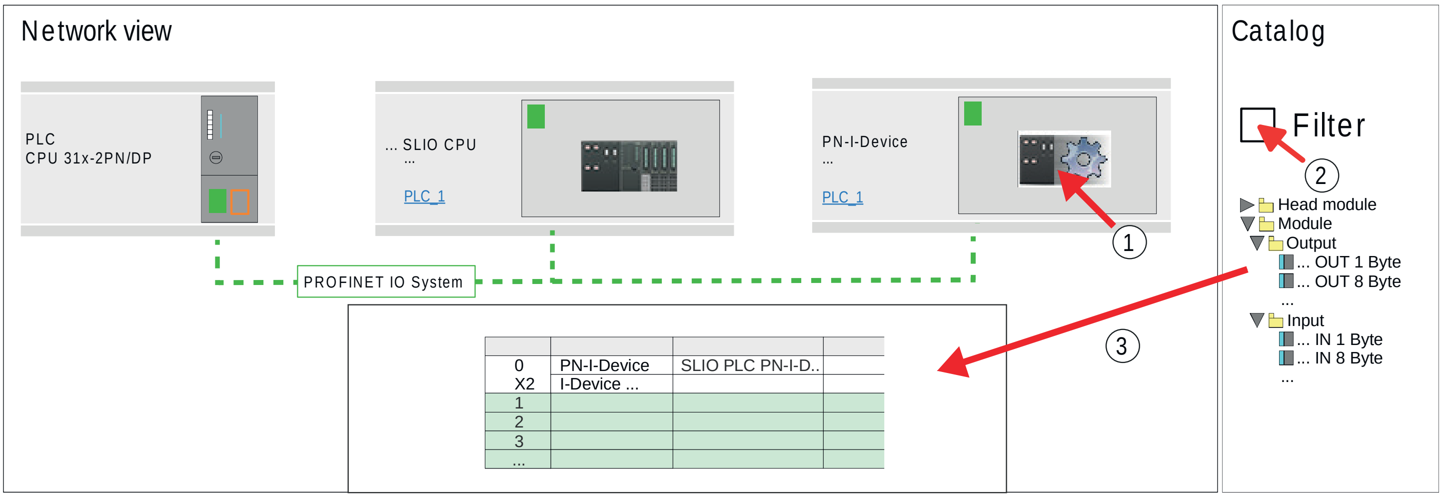
4. Under *'Properties'*, click at *'PROFINET interface ... '* and navigate to *'PROFINET'*. Disable *'Generate PROFINET device name automatically'* and enter a name for the I-Device at *'PROFINET device name'*.

> *Write down the PROFINET device name. This name must also be specified as the 'PROFINET device name' of the I-Device for the higher-level IO controller!*



5. Click in the *Network view* at the PN I-Device and switch to the *Device overview*.
6. Activate the *'Filter'* in the hardware catalog.
7. Create the transfer areas by dragging them to the *'Slots'* as I/O areas from the hardware catalog to the *Device view*. There must be no gaps in the slots. To create the transfer areas, the following input and output areas are available that can be assigned to the virtual I-Device:
   - Input: 1, 8, 16, 32, 64, 128, 256, 512 byte
   - Output: 1, 8, 16, 32, 64, 128, 256, 512 byte

   The data direction for *Input* or *Output* refers to the view of the I-Device.
   - *Input* areas define data that are sent from the higher-level IO controller to the I-Device and which are mapped to the input address area of the CPU.
   - *Output* areas define data that are sent to the higher-level IO controller and which are to stored in the output address area of the CPU.

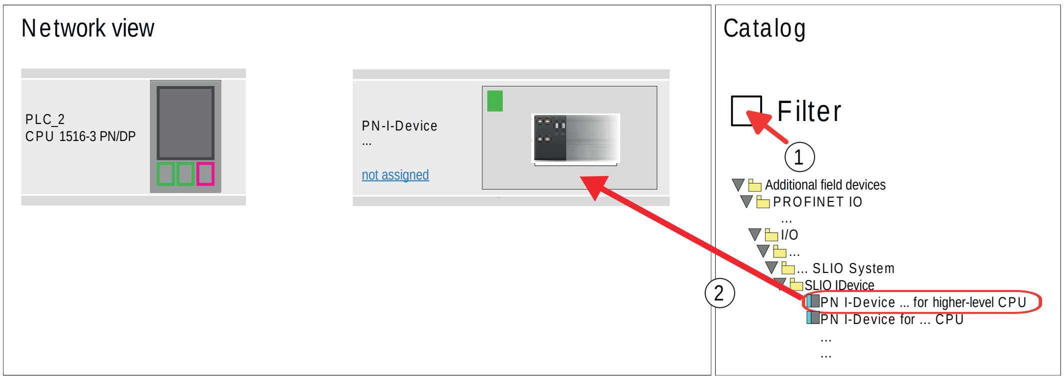


8. Save and transfer your project to the CPU.

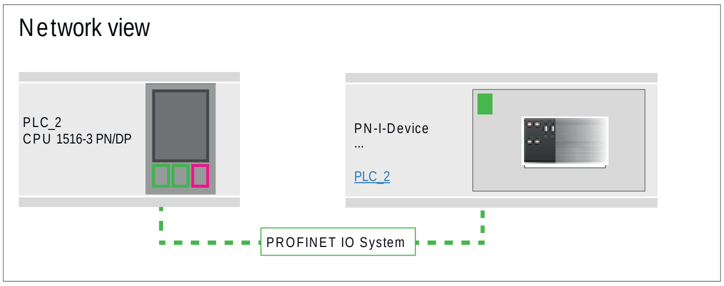
13.6.1.4 Configuration in the higher-level IO controller

It is assumed that a CPU with the higher-level IO controller, such as a Siemens CPU 1516-3 PN/DP with IP address is configured. The IP address must be in the same IP circuit as the IP address of the I-Device.

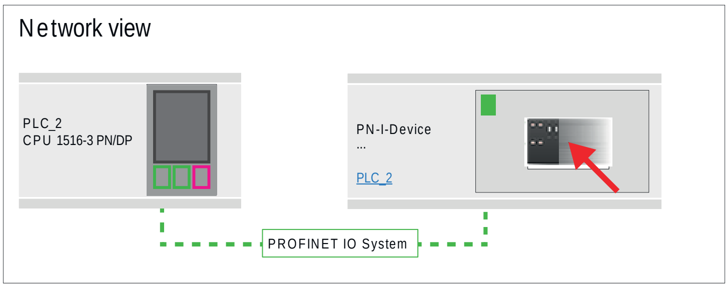
1. Open the project of the CPU with the higher-level IO controller.
2. For the project engineering of I-Device in the higher-level IO controller you have to search the device *'PN I-Device 015-CEFNR00 for higher-level CPU'* in the hardware catalog at *PROFINET-IO* and drag&drop it in the *Network view*.



3. To connect to the CPU, click at *'not assigned'* and select the PROFINET interface of the CPU.



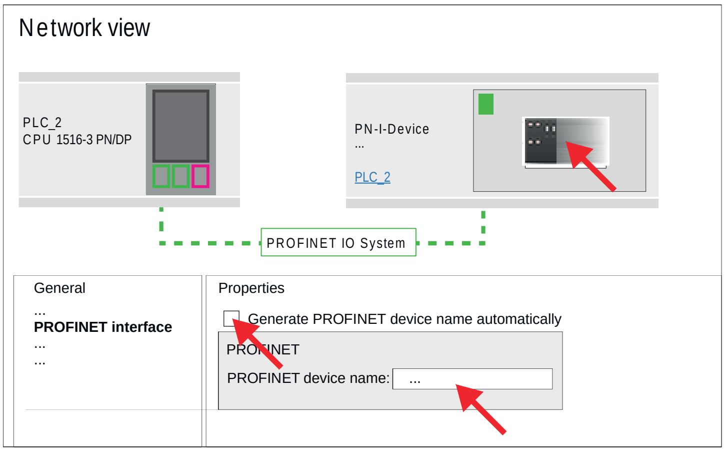
4. Click at *'PN-I-Device'* and select *'Context menu → Properties'*.



⇨ The properties dialog of the CPU is opened.

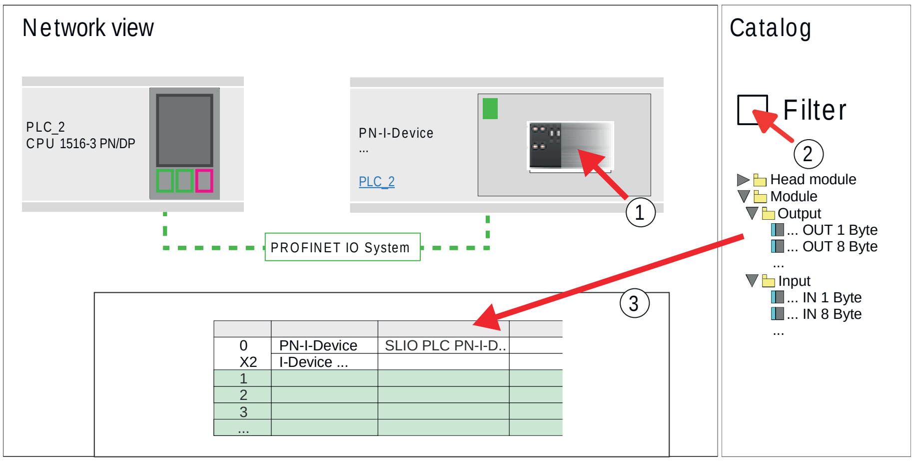
5. Under *'Properties'*, click at *'PROFINET interface ... '* and navigate to *'PROFINET'*. Disable *'Generate PROFINET device name automatically'* and enter at *'PROFINET device name'* the previously noted name of the I-Device.

> *The 'PROFINET device name' must match 'PROFINET device name' of the I-Device CPU, which you have written down before!* *Chap. 13.6.1.3 'Configuration as I-Device' page 335*



7. Configure an input area of the same size for each output area of the I-Device in the IO controller and vice versa. Here also no gaps may arise. In particular, make sure that the order of the transfer areas matches that of the I-Device configuration. The following transfer units are available:

- Input: 1, 8, 16, 32, 64, 128, 256, 512 byte per slot
- Output: 1, 8, 16, 32, 64, 128, 256, 512 byte per slot



8. Save and transfer your project to the CPU.

⇨ Your PROFINET CPU is now connected as an I-Device to the higher-level PROFINET IO controller.

I-Device with S7 routing

S7 routing is not possible with the procedure above. S7 routing is only possible if the I-Device and the higher-level I/O controller are configured in the same network. The device names must not be identical. By using identical names and extending the name of the I-Device with "-x", this is detected internally and used appropriately for S7 routing.

13.7 TIA Portal - Option: Deployment PROFIBUS communication

13.7.1 Fast introduction

Overview

The PROFIBUS DP master must be configured in the *Project view*. Here the configuration happens by the sub module X1 (MPI/DP) of the Siemens CPU.

Enable bus functionality via VSC

To switch the MPI(PB) interface X3 to PROFIBUS functionality, you have to enable the according bus functionality by means of a VSC storage media from Yaskawa. By plugging the VSC storage card and then an overall reset the according functionality is enabled.

‘Overview’ page 121

Steps of configuration

The configuration of the PROFIBUS DP master should be done with the following approach:

- Activating bus functionality by means of a VSC
- Hardware configuration - CPU
- Deployment as DP master *or* deployment as DP slave
- Transfer of the entire project to the CPU

13.7.2 Activating bus functionality by means of a VSC

Proceeding

To use the MPI (PB) interface X3 as PROFIBUS interface, the PROFIBUS functionality must be activated by means of a VSC memory card of Yaskawa.

1. Plug the VSC memory card. *‘Overview’ page 121*
2. Execute an overall reset. *Chap. 4.12 ‘Overall reset’ page 115*

   ⇨
   - The PROFIBUS functionality is enabled.
   - With activating the bus functionality "PROFIBUS DP master", the bus functionality "PROFIBUS DP slave" is also enabled.

13.7.3 Hardware configuration - CPU

Perform a hardware configuration for the CPU. *Chap. 13.3 ‘TIA Portal - Hardware configuration - CPU’ page 326*

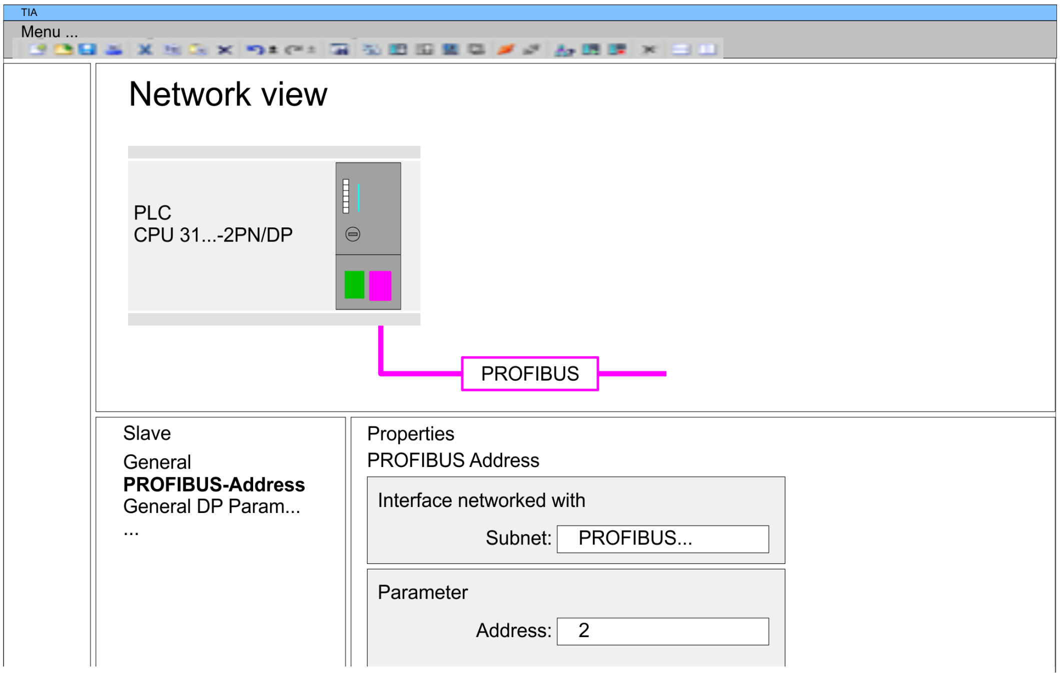
13.7.4 Deployment as PROFIBUS DP master

Precondition

The hardware configuration described before was performed.

Proceeding

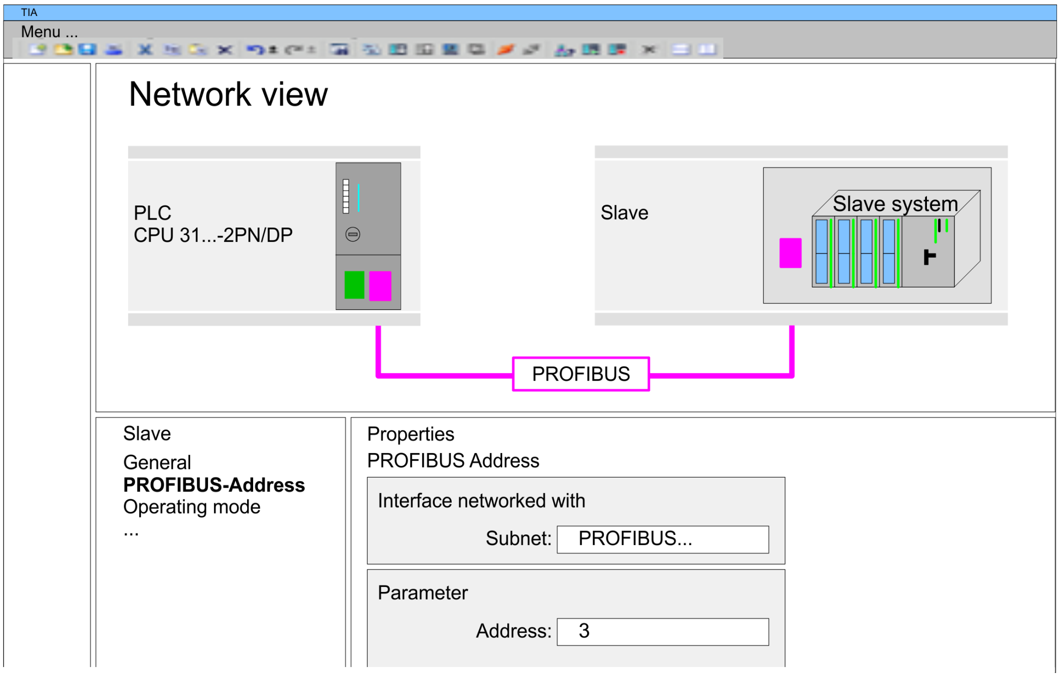
1. Switch to the *'Device view'*.
2. Select the PROFIBUS interface of your CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2) and select *'Context menu ➔ Properties'*.
   ⇨ The *'Properties'* dialog is shown.
3. On the *'General'* tab, click *'MPI Address'*.
   ⇨ The dialog for the MPI address appears.
4. Set the *'Interface type'* to "PROFIBUS".
5. Connect to PROFIBUS and preset an address (preferably 2).
6. Switch at *'Operating mode'* to "DP master".
   ⇨ A master system is inserted.



Now the project engineering of your PROFIBUS DP master is finished. Please link up now your DP slaves with periphery to your DP master.

1. For the project engineering of PROFIBUS DP slaves you search the concerning PROFIBUS DP slave in the hardware catalog and drag&drop it in the subnet of your master.
2. Assign a valid PROFIBUS address to the DP slave.
3. Link up the modules of your DP slave system in the plugged sequence and add the addresses that should be used by the modules.
4. If needed, parametrize the modules.

5. Save, compile and transfer your project.



13.7.5 Deployment as PROFIBUS DP slave

Fast introduction

The following section describes how to use the PROFIBUS part as an "intelligent" DP slave on master systems. Execute the following steps:

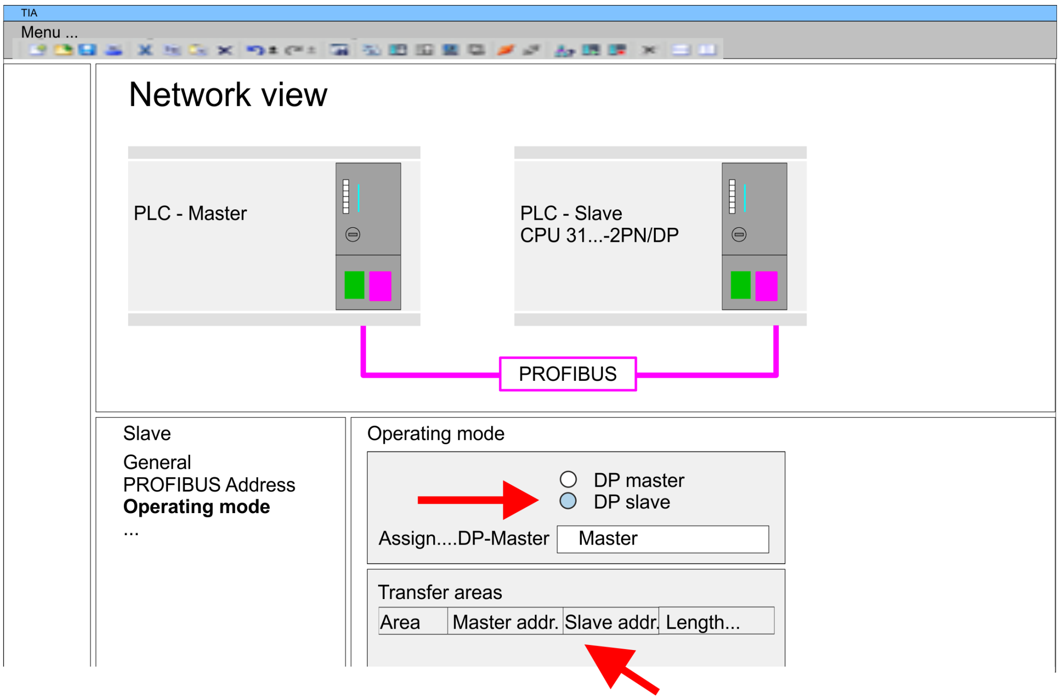
1. Configure a station with a CPU with DP master mode.
2. Network with PROFIBUS
3. Configure another station with a CPU with DP slave operating mode.
4. Network with PROFIBUS and assign the DP slave to the DP master.
5. Configure the I/O areas for the slave part.
   ⇨ The configuration is automatically adapted to the DP master.
6. Save, translate and transfer the respective project into the corresponding master or slave CPU.

Configuration of the master part

1. Start the Siemens TIA Portal and configure a CPU.
2. Designate the station as "... DP master".
3. Integrate the modules according to the hardware configuration of the master system.
4. Select the PROFIBUS interface of the master CPU and select *'Context menu ➔ Properties'*.
   ⇨ The *'Properties'* dialog is shown.
5. Set at *interface*: Type "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 2).
7. Switch at Operating mode to "DP master".

Configuration of the slave part

1. Configure a Siemens CPU 315-2 PN/DP (6ES7 315-2EH14 V3.2) as another CPU ↳ *Chap. 13.3 ‘TIA Portal - Hardware configuration - CPU’ page 326*
2. Designate the station as "... DP slave".
3. Integrate your modules according to your hardware configuration.
4. Select the PROFIBUS interface of your CPU and select *‘Context menu ➔ Properties’*.
   ⇨ The *‘Properties’* dialog is shown.
5. Set at interface: Type "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 3).
7. Switch at *‘Operating mode’* to "DP slave".
8. Under *‘Assigned DP master’*, select your master system.
9. Use *‘Transfer area’* to define the I/O address areas of the slave CPU that are to be assigned to the DP slave.
10. Save, translate and transfer the respective project into the corresponding master or slave CPU.



13.8 Deployment *OPC UA*

↳ *Chap. 5 ‘Deployment OPC UA’ page 129*

13.9 TIA Portal - Include Controls Library

Overview

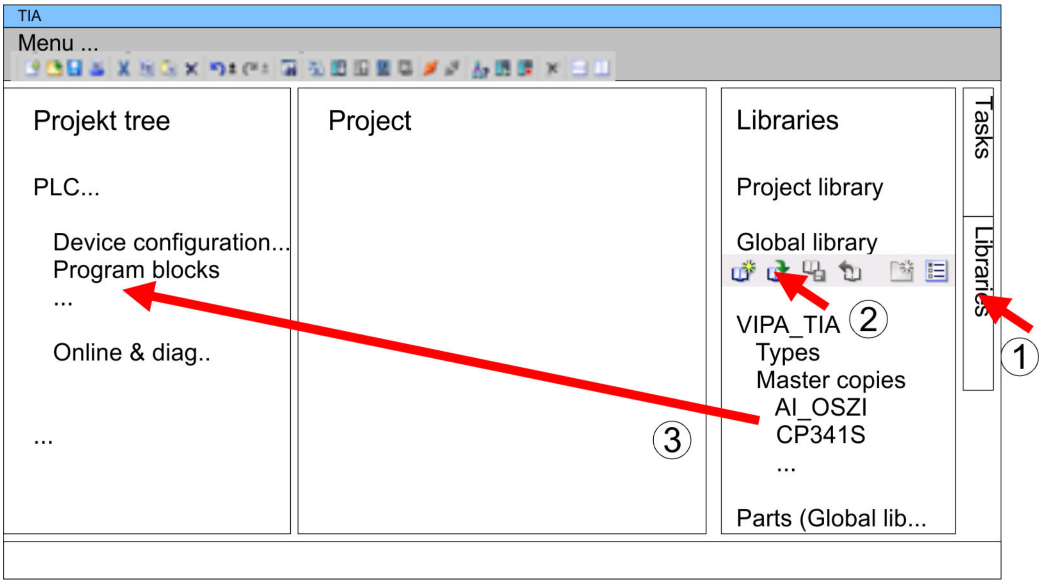
- The product specific blocks can be found at www.yaskawa.eu.com in the *'Download Center'* under *'Controls Library'* as library download file.
- The library is available as packed zip file for the corresponding TIA Portal version.
- As soon as you want to use product specific blocks you have to import them into your project.
  Execute the following steps:
  – Load an unzip the file ...TIA_Vxx.zip (note TIA Portal version)
  – Open library and transfer blocks into the project

Unzip ...TIA_Vxx.zip

Start your un-zip application with a double click on the file TIA_Vxx.zip and copy all the files and folders in a work directory for the Siemens TIA Portal.

Open library and transfer blocks into the project

1. Start the Siemens TIA Portal with your project.
2. Switch to the *Project view*.
3. Choose "Libraries" from the task cards on the right side.
4. Click at "Global libraries".
5. Click at "Open global libraries".
6. Navigate to your directory and load the file ...TIA.alxx.



7. Copy the necessary blocks from the library into the "Program blocks" of the *Project tree* of your project. Now you have access to the product specific blocks via your user application.

13.10 TIA Portal - Project transfer

Overview

There are the following possibilities for project transfer into the CPU:

- Transfer via MPI
- Transfer via Ethernet
- Transfer via memory card

13.10.1 Transfer via MPI

Transfer via MPI

Currently the programming cables from Yaskawa for transfer via MPI are not supported. This is only possible with the programming cable from Siemens.

1. Establish a connection to the CPU via MPI with an appropriate programming cable. Information may be found in the corresponding documentation of the programming cable.
2. Switch-ON the power supply of your CPU and start the Siemens TIA Portal with your project.
3. Select in the *Project tree* your CPU and choose *'Context menu ➔ Download to device ➔ Hardware configuration'* to transfer the hardware configuration.
4. To transfer the PLC program choose *'Context menu ➔ Download to device ➔ Software'*. Due to the system you have to transfer hardware configuration and PLC program separately.

13.10.2 Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X1/X5: Ethernet PG/OP channel

Initialization

So that you may the according Ethernet interface, you have to assign IP address parameters by means of the "initialization". ✎ *Chap. 13.4 'TIA Portal - Hardware configuration - Ethernet PG/OP channel' page 329*

Please consider to use the same IP address data in your project for the CP 343-1.

Transfer

1. For the transfer, connect, if not already done, the appropriate Ethernet jack to your Ethernet.
2. Open your project with the Siemens TIA Portal.
3. Click in the *Project tree* at *Online access* and choose here by a double-click your network card, which is connected to the Ethernet PG/OP interface.
4. Select in the *Project tree* your CPU and click at [Go online].
5. Set the access path by selecting "PN/IE" as type of interface, your network card and the according subnet. Then a net scan is established and the corresponding station is listed.
6. Establish with [Connect] a connection.
7. Click to *'Online ➔ Download to device'*.
   ⇨ The according block is compiled and by a request transferred to the target device. Provided that no new hardware configuration is transferred to the CPU, the entered Ethernet connection is permanently stored in the project as transfer channel.

13.10.3 Transfer via memory card

Proceeding

The memory card serves as external storage medium. There may be stored several projects and sub-directories on a memory card. Please regard that your current project is stored in the root directory and has one of the following file names:

- S7PROG.WLD
- AUTOLOAD.WLD

1. Start the Siemens TIA Portal with your project.
2. Create a wld file with *'Project ➔ Memory card file ➔ New'*.
   ⇨ The wld file is shown in the *Project tree* at "SIMATIC Card Reader" as "Memory card file".
3. Copy the blocks from the *Program blocks* to the wld file. Here the hardware configuration data are automatically copied to the wld file as "System data".
4. Copy the wld file at a suited memory card. Plug this into your CPU and start it again.
   ⇨ The transfer of the application program from the memory card into the CPU takes place depending on the file name after an overall reset or PowerON.

   S7PROG.WLD is read from the memory card after overall reset.

   AUTOLOAD.WLD is read from the memory card after PowerON.

   The blinking of the SD LED of the CPU marks the active transfer. Please regard that your user memory serves for enough space for your user program, otherwise your user program is not completely loaded and the SF LED gets on.

Appendix

Content

A System specific event IDs

Event IDs

↬ *Chap. 4.19 ‘Diagnostic entries’ page 128*

Event ID	Description
0x115C	Manufacture interrupt (OB 57) for EtherCAT
	OB: OB number
	ZINFO1: Input / Output address
	ZINFO2: Interrupt type
	0: Reserved
	1: Diagnostic interrupt (incoming)
	2: Process interrupt
	3: Pull interrupt
	4: Plug interrupt
	5: Status interrupt
	6: Update interrupt
	7: Redundancy interrupt
	8: Controlled by the supervisor
	9: Enabled
	10: Wrong sub module plugged
	11: Recovery of the sub module
	12: Diagnostic interrupt (outgoing)
	13: Cross traffic connection message
	14: Neighbourhood change message
	15: Synchronisation message (bus)
	16: Synchronisation message (device)
	17: Network component message
	18: Clock synchronisation message (bus)
	31: Pull interrupt component
	32: Vendor-specific interrupt min.
	33: Vendor-specific interrupt topology change
	127: Vendor-specific interrupt max.
	ZINFO3: CoE error code
	DatID: Input
	DatID: Output
0x38D0	Bus recovery
	0: OB
	PK: Priority class
	ZINFO1: Logical address of the IO system
	ZINFO2: Logical address of the virtual device

Event ID	Description
	ZINFO3 - Position 0: Station number
	ZINFO3 - Position 11: IO system ID
	ZINFO3 - Bit 15: System ID DP/PN
0x38D1	Bus recovery, but expected configuration does not match actual configuration
	0: OB
	PK: Priority class
	ZINFO1: Logical address of the IO system
	ZINFO2: Logical address of the virtual device
	ZINFO3 - Position 0: Station number
	ZINFO3 - Position 11: IO system ID
	ZINFO3 - Bit 15: System ID DP/PN
0x39D0	Bus failure
	OB: OB number
	PK: Priority class
	ZINFO1: Logical address of the IO system
	ZINFO2: Logical address of the virtual device
	ZINFO3 - Position 0: Station number
	ZINFO3 - Position 11: IO system ID
	ZINFO3 - Bit 15: System ID DP/PN
0x454B	STOP: Maximum number of time-outs of a synchronous OB reached
	OB: CPU mode
	PK:
	ZINFO1: ZInfo1
	ZINFO2: ZInfo2
	ZINFO3: ZInfo3
	DatID: Block type
0x49CA	PROFINET IO system failure (Watchdog)
	0: OB
	1: Cyclic program (OB 1)
	16: Time of day interrupt OB (OB 16)
	17: Time of day interrupt OB (OB 17)
	32: Cyclic interrupt OB (OB 32)
	33: Cyclic interrupt OB (OB 33)
	34: Cyclic interrupt OB (OB 34)
	35: Cyclic interrupt OB (OB 35)
	36: Cyclic interrupt OB (OB 36)
	37: Cyclic interrupt OB (OB 37)
	38: Cyclic interrupt OB (OB 38)

Event ID	Description
	64: Synchronous cycle interrupt (OB 64)
	65: Synchronous technology interrupt (OB 65)
	80: Time error OB (OB 80)
	81: Power supply error OB (OB 81)
	82: Diagnostic interrupt OB (OB 82)
	83: Plug/Pull OB (OB 83)
	85: Program sequence error OB (OB 85)
	86: Component rack failure OB (OB 86)
	PK: Priority class
	ZINFO1: Logical address of the IO system
	ZINFO2: Interrupt reason
	0: Unknown
	1: Alarm overflow
	2: Message box overflow
	3: Cyclic data not in bus cycle
	4: Application bus cycle error
	5: Watchdog
	6: Error handler
	7: Time-out when receiving cyclic data
	8: No not-IO-task available
	ZINFO3 - Position 0: Station number
	ZINFO3 - Position 11: IO system ID
	ZINFO3 - Bit 15: System ID DP/PN
0xE003	Error in access to periphery
	ZINFO1: Transfer type
	ZINFO2: Periphery address
	ZINFO3: Slot
0xE004	Multiple configuration of a periphery address
	ZINFO1: Periphery address
	ZINFO2: Slot
0xE005	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
0xE007	Configured input/output bytes do not fit in the periphery area
0xE008	Internal error - Please contact the hotline!
0xE009	Error on accessing the standard backplane bus
0xE010	Non-defined component recognised at the standard backplane bus

Event ID	Description
	ZINFO2: Slot
	ZINFO3: Type identifier
0xE011	Master project engineering at slave CPU not possible or wrong slave configuration
0xE012	Error at configuration standard backplane bus
0xE013	Error at shift register access to standard backplane bus digital modules
0xE014	Error in Check_Sys
0xE015	Error in access to master
	ZINFO2: Slot of the master
	ZINFO2: Page frame master
0xE016	Maximum block size exceeded in master transfer
	ZINFO1: Periphery address
	ZINFO2: Slot
0xE017	Error in access to integrated slave
0xE018	Error in mapping the master periphery
0xE019	Error on standard backplane bus system detection
0xE01A	Error at detection of the operating mode (8/9 bit)
0xE01B	Error: Maximum number of plug-in components exceeded
0xE020	Error: Interrupt information undefined
	ZINFO1: Rack/Slot
	ZINFO3: Error type
	4: Rack/Slot (in ZINFO1) undefined
	5: Alarm type (in DatID) undefined
	DatID: Interrupt type
0xE030	Error of the standard backplane bus
0xE033	Internal error - Please contact the hotline!
0xE0B0	SPEED7 is not stoppable
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE0C0	Not enough memory space in the working memory for code block (block too large)
0xE0CB	Error on SSL access
	ZINFO1: Error
	4: SSL wrong
	5: Sub-SSL wrong
	6: Index wrong
	ZINFO2: SZL-ID
	ZINFO3: Index

Event ID	Description
0xE0CC	Communication error
	ZINFO1: Error code
	1: Wrong priority
	2: Buffer overrun
	3: Telegram format error
	4: Wrong SSL request (SSL-ID invalid)
	5: Wrong SSL request (SSL-Sub-ID invalid)
	6: Wrong SSL request (SSL-Index invalid)
	7: Wrong value
	8: Wrong return value
	9: Wrong SAP
	10: Wrong connection type
	11: Wrong sequence number
	12: Faulty block number in the telegram
	13: Faulty block type in the telegram
	14: Inactive function
	15: Wrong size in the telegram
	20: Error in writing on MMC
	90: Faulty buffer size
	98: Unknown error
	99: Internal error
0xE0CD	Error at DP-V1 job management
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE0CE	Error: Time out when sending I-Slave diagnostics
0xE100	Memory card access error
0xE101	Memory card error file system
0xE102	Memory card error FAT
0xE104	Memory card error at saving
	ZINFO3: Not user relevant
0xE200	Memory card writing finished (Copy Ram2Rom)
	OB: Not user relevant
	PK: Not user relevant
0xE210	Memory card reading finished (reload after memory reset)
	OB: Not user relevant
	PK: Not user relevant

Event ID	Description
	ZINFO1 - Position 0: Not user relevant
0xE21D	Memory card reading: Error on reload (after memory reset), error in the block header
	ZINFO1: Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO2: Block number
	ZINFO3: Block length
0xE21E	Memory card reading: Error in recharging (after memory reset), "Protect.wld" file too large
	OB: Not user relevant
0xE21F	Memory card reading: Error at reload (after memory reset), checksum error when reading
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Not user relevant
	ZINFO2: Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB

Event ID	Description
	102: VSFB
	111: VOB
	ZINFO3: Block number
0xE300	Internal flash writing completed (copy Ram2Rom)
0xE310	Internal flash reading completed (recharging after battery failure)
0xE400	FSC card was plugged
	OB: FSC activated from this slot (PK)
	OB: The inserted FSC is the activated FSC
	OB: The inserted FSC is compatible with the CPU
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40

Event ID	Description
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
0xE401	FSC card was removed
	OB: Action after the end of the trial time
	0: No action
	1: CPU STOP
	2: CPU STOP and FSC deactivated
	3: Factory reset
	255: FSC was not activated
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060

Event ID	Description
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
	DatID: FeatureSet Trialtime in minutes
0xE402	A configured functionality is not activated. The configuration is accepted, but the PLC can not go to RUN.
	ZINFO1: Required FSC: PROFIBUS
	ZINFO1: Required FSC: MOTION
	ZINFO2: Number of released axes
	ZINFO3: Number of configured axes
0xE403	FSC can not be activated in this CPU
	OB: FCS error code
	PK: FSC source
	0: CPU

Event ID	Description
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020

Event ID	Description
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
0xE404	Feature set deleted due to CRC error
0xE405	The trial time of a feature set/memory card has expired
	OB: Action after the end of the trial time
	0: No action
	1: CPU STOP
	2: CPU STOP and FSC deactivated
	3: Factory reset
	255: FSC was not activated
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10

Event ID	Description
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
	DatID: FeatureSet Trialtime in minutes
0xE406	The inserted feature set is corrupt
	PK: FSC source
	0: CPU
	1: Card
0xE410	A CPU feature set was activated
	PK: FSC source
	0: CPU
	1: Card
	ZINFO1: FSC(CRC)
	1146: 955-C000070
	1736: 955-C0NE040
	2568: FSC-C0ME040
	3450: 955-C000M30
	3903: 955-C000S30
	4361: FSC-C000M30
	4940: FSC-C000S30
	5755: 955-C0ME040
	6843: FSC-C0NE040
	8561: FSC-C000S20

Event ID	Description
	9012: FSC-C000M20
	13895: 955-C000060
	15618: 955-C000S20
	16199: 955-C000M20
	17675: FSC-C000S00
	18254: FSC-C000M00
	20046: FSC-C000040
	21053: 955-C000040
	22904: 955-C000S00
	23357: 955-C000M00
	24576: 955-C000050
	35025: 955-C00MC10
	36351: FSC-C000S40
	36794: FSC-C000M40
	37260: 955-C000S40
	37833: 955-C000M40
	38050: FSC-C00MC10
	41460: 955-C000M50
	41526: 955-C0PE040
	42655: FSC-C00MC00
	47852: 955-C00MC00
	48709: FSC-C0PE040
	50574: 955-C000M70
	52366: 955-C000030
	53501: FSC-C000030
	58048: FSC-C000020
	63411: 955-C000M60
	65203: 955-C000020
	ZINFO2: FSC serial number (high word)
	ZINFO3: FSC serial number (low word)
0xE500	Memory management: Deleted block without corresponding entry in BstList
	ZINFO2: Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB

Event ID	Description
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO3: Block number
0xE501	Parser error
	ZINFO1: Error code
	1: Parser error: SDB structure
	2: Parser error: SDB is not a valid SDB type
	ZINFO2: SDB type
	ZINFO3: SDB number
0xE502	Invalid block type in protect.wld (block was not loaded)
	ZINFO2: Block type
	56: OB
	65: DB
	66: SDB
	67: FC
	68: SFC
	69: FB
	70: SFB
	97: VDB
	98: VSDB
	99: VFC
	100: VSFC
	101: VFB
	102: VSFB
	111: VOB
	ZINFO3: Block number
0xE503	Inconsistency of code sizes and block sizes in the working memory
	ZINFO1: Code size
	ZINFO2: Block size (high word)
	ZINFO3: Block size (low word)
0xE504	Additional information for CRC error in the working memory
	ZINFO2: Block address (high word)

Event ID	Description
	ZINFO3: Block address (low word)
0xE505	Internal error - Please contact the hotline!
	ZINFO1: Cause for MemDump
	0: Unknown
	1: Manual request
	2: Invalid OP value
	3: CRC code error
	4: Processor exception
	5: Processor exception with dump after reboot
	6: Block-CRC error
0xE604	Multiple configuration of a periphery address for Ethernet PG/OP channel
	ZINFO1: Periphery address
	ZINFO3: 0: periphery address is input, 1: periphery address is output
0xE605	Too many productive connections configured
	ZINFO1: Interface slot
	ZINFO2: Number of configured connections
	ZINFO3: Number of admissible connections
0xE610	On-board PROFIBUS/MPI: Bus error removed
	PK: Not user relevant
	ZINFO1: Interface
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE701	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE703	Internal error - Please contact the hotline!
	0: Master system ID
	PK: Not user relevant
	ZINFO1: Not user relevant
	ZINFO2: Slave address
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE705	Too many PROFIBUS slaves configured
	ZINFO1: Diagnostic address of the PROFIBUS master
	ZINFO2: Number of configured slaves

Event ID	Description
	ZINFO3: Number of admissible slaves
0xE70A	PROFIBUS configured, but access way disabled
	ZINFO1: Logical base address of the DP master
	ZINFO2 - Position 8: DP master system ID
0xE710	On-board PROFIBUS/MPI: Bus error occurred
	PK: Not user relevant
	ZINFO1: Interface
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE720	Internal error - Please contact the hotline!
	ZINFO1: Slave no
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Master system ID
0xE721	Internal error - Please contact the hotline!
	PK: Error code
	1: Error when assigning diagnostic address for slave (no. in ZINFO3)
	2: Error when assigning diagnostic address for master
	3: Error when assigning logical address when de/activating for slave (no. in ZINFO3)
	4: Error when assigning slots for slave (no. in ZINFO3)
	5: Error in DPV1 configuration (inputs) for slave (no. in ZINFO3)
	6: Error in DPV1 configuration (outputs) for slave (no. in ZINFO3)
	7: SubnetID for master (in ZINFO2) invalid
	8: Slave (no. in ZINFO3) could not be configured (CFG length in OB)
	ZINFO1: Not user relevant
	ZINFO2: Master system ID
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xE722	Internal error - Please contact the hotline!
	ZINFO1: Channel-Event
	0: Channel offline
	1: Bus error
	2: Internal error
	ZINFO2: Master system ID
	DatID: Not user relevant
0xE723	Internal error - Please contact the hotline!
	ZINFO1: Error code

Event ID	Description
	1: Parameter error
	2: Configuration error
	ZINFO2: Master system ID
	DatID: Not user relevant
0xE780	Error in configuration of a process image
	ZINFO1: Not user relevant
	ZINFO2: Logical address
	ZINFO3: IO Flag
0xE781	Address range exceeds process image limit
	ZINFO1: Address
	ZINFO2: Length of the address range
	ZINFO3: Size of the process image
	DatID: Address range
0xE801	CMD - auto command: CMD_START recognized and executed
0xE802	CMD - auto command: CMD_End recognized and executed
0xE803	CMD - auto command: WAIT1SECOND recognized and executed
0xE804	CMD - auto command: WEBPAGE recognized and executed
0xE805	CMD - auto command: LOAD_PROJECT recognized and executed
0xE806	CMD - auto command: SAVE_PROJECT recognized and executed
	ZINFO3: Status
	0: Error
	1: OK
	32768: Wrong password
0xE807	CMD - auto command: FACTORY_RESET recognized and executed
0xE808	Internal message
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
0xE809	Internal message
	ZINFO3: Not user relevant
0xE80A	Internal message
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80B	CMD - auto command: DIAGBUF recognized and executed
	ZINFO3: Status
	0: OK

Event ID	Description
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80C	Internal message
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE80D	Internal message
0xE80E	CMD - auto command: SET_NETWORK recognized and executed
0xE80F	Internal message
	ZINFO3: Status
	0: OK
	65153: File create error
	65185: File writing error
	65186: Odd address for reading
0xE810	Internal message
0xE811	Internal message
0xE812	Internal message
0xE813	Internal message
0xE814	CMD - auto command: SET_MPI_ADDRESS identified
0xE816	CMD - auto command: SAVE_PROJECT recognized but not executed, because the CPU memory is empty
0xE817	Internal message
	ZINFO3: Not user relevant
0xE820	Internal message
0xE821	Internal message
0xE822	Internal message
0xE823	Internal message
0xE824	Internal message
0xE825	Internal message
0xE826	Internal message
0xE827	Internal message
0xE828	Internal message
0xE829	Internal message
0xE82A	CMD - auto command: CPUTYPE_318 recognized and executed
	ZINFO3: Error code
0xE82B	CMD - auto command: CPUTYPE_ORIGINAL recognized and executed

Event ID	Description
	ZINFO3: Error code
0xE82C	CMD - auto command: WEBVISU_PGOP_ENABLE recognized and executed
0xE82D	CMD - auto command: WEBVISU_PGOP_DISABLE recognized and executed
0xE82E	CMD - auto command: WEBVISU_CP_ENABLE recognized and executed
0xE82F	CMD - auto command: WEBVISU_CP_DISABLE recognized and executed
0xE830	CMD - auto command: OPCUA_PGOP_ENABLE recognized and executed
0xE831	CMD - auto command: OPCUA_PGOP_DISABLE recognized and executed
0xE832	CMD - auto command: OPCUA_CP_ENABLE recognized and executed
0xE833	CMD - auto command: OPCUA_CP_DISABLE recognized and executed
0xE8FB	CMD - auto command: Error: Initialization of the Ethernet PG/OP channel by means of SET_NETWORK is faulty
0xE8FC	CMD - auto command: Error: Some IP parameters missing in SET_NETWORK
0xE8FE	CMD - auto command: Error: CMD_START not found
0xE8FF	CMD - auto command: Error while reading CMD file (memory card error)
0xE901	Checksum error
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	DatID: Not user relevant
0xE902	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	DatID: Not user relevant
0xE904	PG/OP: Multiple parametrization of a peripheral address
	ZINFO1: Peripheral address
	ZINFO2: Slot
	ZINFO3: Data width
	DatID: 0x54 Peripheral address is input address
	DatID: 0x55 Peripheral address is output address
0xE90A	PROFINET configured, but access way disabled
	ZINFO1: Logical address of the IO system
	ZINFO3: Station number
	ZINFO3: IO system ID
	ZINFO3: System ID DP/PN
0xE910	PG/OP: Input peripheral address out of peripheral area
	ZINFO1: Peripheral address
	ZINFO2: Slot
	ZINFO3: Data width
0xE911	PG/OP: Output peripheral address out of peripheral area
	ZINFO1: Peripheral address

Event ID	Description
	ZINFO2: Slot
	ZINFO3: Data width
0xE920	Configuration error PROFINET
	ZINFO1 - Position 0: Error code
	1: Double IP/PROFINET configuration in slot 2 and 4
	2: PROFINET IO system configured in slot 4
	3: Too many PROFINET IO controller configured
	4: Virtual device multiple configured
	5: EtherCAT devices for PROFINET CP configured
	6: PROFINET devices for EtherCAT CP configured
	7: PROFINET CP configured at slot 2, although this is not supported
	8: A PROFINET IO system (I-Device) must not be configured on the CP on slot 4
	9: A PROFINET IO system (controller) must not be configured on the CP on slot 4
0xE980	Error when loading the WebVisu project file
	ZINFO1: Platform
0xE981	Error in the configuration of the WebVisu project
	ZINFO1: Platform
0xE982	Internal error of the WebVisu server
	ZINFO1: Platform
0xE983	Hardware configuration of the control is not loaded, WebVisu is not started
	ZINFO1: Platform
0xE984	WebVisu is blocked by the user, start of the WebVisu was prevented
	ZINFO1: Platform
0xE985	WebVisu was started
	ZINFO1: Platform
0xE986	WebVisu was stopped
	ZINFO1: Platform
0xE987	WebVisu was enabled by the user
	ZINFO1: Platform
0xE988	WebVisu was disabled by the user
	ZINFO1: Platform
0xE989	WebVisu and OPC UA project not allowed at the same time
	ZINFO1: Platform
0xE9A0	Error when loading the OPC UA project file
	ZINFO1: Platform
	ZINFO3 - Bit 0: Error code
0xE9A1	OPC UA: No FSC enabled
	ZINFO1: Platform

Event ID	Description
0xE9A2	OPC UA: TAR file invalid
	ZINFO1: Platform
	ZINFO3: Error code
0xE9A3	OPC UA: Internal error of the OPC UA server
	ZINFO1: Platform
	ZINFO3: Error code
0xE9A4	OPC UA: Hardware configuration of the control is not loaded, Server is not started
	ZINFO1: Platform
0xE9A5	OPC UA blocked by user, start of the server was prevented
	ZINFO1: Platform
0xE9A6	OPC UA server was started
	ZINFO1: Platform
0xE9A7	OPC UA server was stopped
	ZINFO1: Platform
0xE9A8	OPC UA was disabled by the user
	ZINFO1: Platform
0xE9A9	OPC UA was enabled by the user
	ZINFO1: Platform
0xE9AA	OPC UA: Lock by S7 configuration (access settings)
	ZINFO1: Platform
0xE9AB	OPC UA and WebVisu project not allowed at the same time
	ZINFO1: Platform
0xEA00	Internal error - Please contact the hotline!
	PK: Not relevant to user
	DatID: Not user relevant
0xEA01	Internal error - Please contact the hotline!
	PK: Not user relevant
	ZINFO1: Slot
	DatID: Not user relevant
0xEA02	SBUS: Internal error (internal plugged sub module not recognized)
	PK: Not user relevant
	ZINFO1: Slot
	ZINFO2: Type identifier target
	ZINFO3: Type identifier
	DatID: Not user relevant
0xEA03	SBUS: Communication error between CPU and IO controller
	OB: Operating mode
	0: Configuration in operating condition RUN

Event ID	Description
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Not user relevant
	ZINFO1: Slot
	ZINFO2: Status
	0: OK
	1: Error
	2: Empty
	3: Busy
	4: Time out
	5: Internal blocking
	6: Too many telegrams
	7: Not Connected
	8: Unknown
	DatID: Not user relevant
0xEA04	SBUS: Multiple configuration of a periphery address
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA05	Internal error - Please contact the hotline!
0xEA07	Internal error - Please contact the hotline!
0xEA08	SBUS: Configured input data width not the same as the connected input data width
	ZINFO1: Configured input data width

Event ID	Description
	ZINFO2: Slot
	ZINFO3: Input data width of the connected component
0xEA09	SBUS: Configured output data width not the same as the connected output data width
	ZINFO1: Configured output data width
	ZINFO2: Slot
	ZINFO3: Output data width of the plugged component
0xEA0A	SBUS: Internal error (internal plugged sub module wrong)
	PK: Not user relevant
	ZINFO1: Slot
	ZINFO2: Type identifier target
	3: PROFINET-CPU
	4: EtherCAT-CPU
	ZINFO3: Type identifier
	3: PROFINET-CPU
	4: EtherCAT-CPU
	DatID: Not user relevant
0xEA10	SBUS: Input periphery address outside the periphery area
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA11	SBUS: Output periphery address outside the periphery area
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA12	SBUS: Error in writing dataset
	ZINFO1: Slot
	ZINFO2: Dataset number
	ZINFO3: Dataset length
0xEA14	SBUS: Multiple configuration of a periphery address (diagnostic address)
	ZINFO1: Periphery address
	ZINFO2: Slot
	ZINFO3: Data width
0xEA15	Internal error - Please contact the hotline!
	ZINFO2: Slot of the master
0xEA18	SBUS: Error in mapping the master periphery
	ZINFO2: Slot of the master
0xEA19	Internal error - Please contact the hotline!
	PK: Not user relevant

Event ID	Description
	ZINFO2: HW slot
	ZINFO3: Interface type
	DatID: Not user relevant
0xEA1A	SBUS: Error in access to SBUS FPGA address table
	PK: Not user relevant
	ZINFO2: HW slot
	ZINFO3: Table
	0: Read
	1: Writing
	DatID: Not user relevant
0xEA20	Error: RS485 interface is not pre-set to PROFIBUS DP master bus a PROFIBUS DP master is configured
0xEA21	Error: Configuration RS485 interface X2/X3: PROFIBUS DP master is configured but missing
	ZINFO2: Interface X is configured incorrectly
0xEA22	Error: Configuration RS485 interface X2: Value is outside the limits
	ZINFO2: Configuration for X2
0xEA23	Error: Configuration RS485 interface X3: Value is outside the limits
	ZINFO2: Configuration for X3
0xEA24	Error: Configuration RS485 interface X2/X3: Interface/protocol missing, default settings are used
	ZINFO2: Configuration for X2
	ZINFO3: Configuration for X3
0xEA30	Internal error - Please contact the hotline!
	ZINFO1: Status
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
0xEA40	Internal error - Please contact the hotline!
	OB: Slot of the CP
	PK: File number
	ZINFO1: Version of the CP
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Line
0xEA41	Internal error - Please contact the hotline!
	OB: Slot of the CP
	PK: File number
	ZINFO1: Version of the CP
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Line

Event ID	Description
0xEA50	PROFINET IO controller: Error in the configuration
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Rack/slot of the controller
	ZINFO2: Device number
	ZINFO3: Slot at the device
	DatID: Not user relevant
0xEA51	PROFINET IO controller: There is no PROFINET IO controller at the configured slot
	PK: Not user relevant
	ZINFO1: Rack/slot of the controller
	ZINFO2: Recognized type identifier at the configured slot
	DatID: Not user relevant
0xEA52	PROFINET IO controller: Too many configured PROFINET IO controllers
	PK: Not user relevant
	ZINFO1: Number of configured controllers
	ZINFO2: Slot of the excessively configured controller
	DatID: Not user relevant
0xEA53	PROFINET IO controller: Too many configured PROFINET IO devices
	ZINFO1: Number of configured devices
	ZINFO2: Slot
	ZINFO3: Maximum possible number of devices
0xEA54	PROFINET IO controller: Multiple configuration of a periphery address or range too long
	0: Error type
	0: No error
	1: Range too long
	2: Input address already used
	3: Output address already used
	PK: Not user relevant
	ZINFO1: Logical address of the IO system
	ZINFO2: Rack/slot of the controller
	ZINFO3: Base address of the block which is too large
	DatID: Not user relevant
0xEA55	PROFINET IO controller: Too many slots or too high slot number configured
	ZINFO1: Rack/slot of the controller
	ZINFO2: Device number
	ZINFO3: Number of configured slots or too high slot number
0xEA56	PROFINET IO controller: Too many subslots or too high sub slot number configured
	ZINFO1: Rack/slot of the controller

Event ID	Description
	ZINFO2: Device number
	ZINFO3: Number of configured subslots or too high sub slot number
0xEA57	PROFINET IO controller: The port configuration in the virtual device has no effect.
0xEA61	Internal error - Please contact the hotline!
	OB: File number
	PK: Slot of the controller
	ZINFO1: Firmware major version
	ZINFO2: Firmware minor version
	DatID: Line
0xEA62	Internal error - Please contact the hotline!
	OB: File number.
	PK: Slot of the controller
	ZINFO1: Firmware major version
	ZINFO2: Firmware minor version
	DatID: Line
0xEA63	Internal error - Please contact the hotline!
	OB: File number
	PK: Slot of the controller
	ZINFO1: Firmware major version
	ZINFO2: Firmware minor version
	DatID: Line
0xEA64	PROFINET IO controller/EtherCAT-CP: Error in configuration
	PK: Interface
	ZINFO1 - Bit 0: Too many devices
	ZINFO1 - Bit 1: Too many devices per second
	ZINFO1 - Bit 2: Too many input bytes per millisecond
	ZINFO1 - Bit 3: Too many output bytes per millisecond
	ZINFO1 - Bit 4: Too many input bytes per device
	ZINFO1 - Bit 5: Too many output bytes per device
	ZINFO1 - Bit 6: Too many productive connections
	ZINFO1 - Bit 7: Too many input bytes in the process image
	ZINFO1 - Bit 8: Too many output bytes in the process image
	ZINFO1 - Bit 9: Configuration not available
	ZINFO1 - Bit 10: Configuration invalid
	ZINFO1 - Bit 11: Refresh interval too small
	ZINFO1 - Bit 12: Refresh interval too large
	ZINFO1 - Bit 13: Invalid device number
	ZINFO1 - Bit 14: CPU is configured as an I device

Event ID	Description
	ZINFO1 - Bit 15: Assume IP address in another way. Is not supported for the IP address of the controller.
	ZINFO2 - Bit 0: Incompatible configuration (SDB version not supported)
	ZINFO2 - Bit 1: EtherCAT: EoE configured but not supported (Possible cause is a too short cycle time of the EtherCAT master system. When using EoE terminals, at least a cycle time of 4ms must be configured.)
	ZINFO2 - Bit 2: DC parameter invalid
	ZINFO2 - Bit 3: I device configuration invalid (slot gap)
	ZINFO2 - Bit 4: MRP configuration invalid (client)
	ZINFO2 - Bit 5: Transfer rate 10 MBit (HD/FD) configured but not supported
0xEA65	Internal error - Please contact the hotline!
	PK: Platform
	0: none
	8: CP
	9: Ethernet CP
	10: PROFINET CP
	12: EtherCAT CP
	16: CPU
	ZINFO1: ServiceID in which the error occurred
	ZINFO2: Command in which the error occurred
	1: Request
	2: Connect
	3: Error
0xEA66	PROFINET IO controller: Error in the communication stack
	OB: StackError.Service
	PK: Rack/slot
	ZINFO1: StackError.Error.Code
	ZINFO2: StackError.Error.Detail
	ZINFO3 - Position 0: StackError.Error.AdditionalDetail
	ZINFO3 - Position 8: StackError.Error.AreaCode
	DatID: StackError.DeviceRef
0xEA67	PROFINET IO controller: Error reading dataset
	OB: Rack/slot of the controller
	PK: Error type
	0: Dataset error local
	1: Dataset error stack
	2: Dataset error station
	ZINFO1: Dataset number
	ZINFO2: Dataset handle (caller)
	ZINFO3: Internal error code from PN stack

Event ID	Description
	DatID: Device
0xEA68	PROFINET IO controller: Error writing dataset
	OB: Rack/slot of the controller
	PK: Error type
	0: Dataset error local
	1: Dataset error stack
	2: Dataset error station
	ZINFO1: Dataset number
	ZINFO2: Dataset handle (caller)
	ZINFO3: Internal error code from PN stack
	DatID: Device
0xEA69	Internal error - Please contact the hotline!
	ZINFO1: Minimum version for the FPGA
	ZINFO2: Loaded FPGA version
0xEA6A	PROFINET IO controller: Service error in the communication stack
	OB: Service ID
	PK: Rack/slot
	ZINFO1: ServiceError.Code
	ZINFO2: ServiceError.Detail
	ZINFO3 - Position 0: ServiceError.AdditionalDetail
	ZINFO3 - Position 8: ServiceError.AreaCode
0xEA6B	PROFINET IO controller: Incorrect Vendor-ID
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised

Event ID	Description
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA6C	PROFINET IO controller: Incorrect Device-ID
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID
0xEA6D	PROFINET IO controller: No empty name
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)

Event ID	Description
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA6E	PROFINET IO controller: Wait for RPC response
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot

Event ID	Description
	ZINFO1: Device ID
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA6F	PROFINET IO controller: PROFINET module deviation
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Rack/slot
	ZINFO1: Device ID
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEA70	PROFINET IO controller: PROFINET stack configuration error
	OB: UnsupportedApiError.api
	PK: Rack/slot
	ZINFO1: UnsupportedApiError.slot
	ZINFO2: UnsupportedApiError.subslot
	DatID: UnsupportedApiError.deviceID
0xEA71	Internal error - Please contact the hotline!
	PK: Rack/slot
	ZINFO1: functionIndex

Event ID	Description
	ZINFO2: Not user relevant
0xEA72	Internal error - Please contact the hotline!
	OB: Connection number
	PK: Slot of the controller
	ZINFO1: Error cause
	129: PNIO
	207: RTA error
	218: AlarmAck
	219: IODConnectRes
	220: IODReleaseRes
	221: IOD/IOXControlRes
	222: IODReadRes
	223: IODWriteRes
	ZINFO2: ErrorDecode
	128: PNIORW: Service Read Write
	129: PNIO: Other Service or internal e.g. RPC errors
	130: Vendor specific
	ZINFO3: Error code (PN spec. V2.722 chapter 5.2.6)
	DatID: Device ID
0xEA81	Internal error - Please contact the hotline!
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: SvnRevision
0xEA82	Internal error - Please contact the hotline!
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: SvnRevision
0xEA83	Internal error - Please contact the hotline!
	OB: Not user relevant
	PK: Not user relevant
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]

Event ID	Description
	ZINFO3: Line
	DatID: SvnRevision
0xEA91	Internal error - Please contact the hotline!
	OB: Current OB number
	PK: Core status
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: Current job number
0xEA92	Internal error - Please contact the hotline!
	OB: Current OB number
	PK: Core status
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: Current job number
0xEA93	Internal error - Please contact the hotline!
	OB: Current OB number
	PK: Core status
	0: INIT
	1: STOP
	2: READY
	3: PAUSE
	4: RUN
	ZINFO1: Filenamehash[0-3]
	ZINFO2: Filenamehash[4-7]
	ZINFO3: Line
	DatID: Current job number

Event ID	Description
0xEA97	Internal error - Please contact the hotline!
	ZINFO3: Slot
0xEA98	Error in file reading via SBUS
	PK: Not user relevant
	ZINFO3: Slot
	DatID: Not user relevant
0xEA99	Parameter assignment job could not be executed
	PK: Not user relevant
	ZINFO1: File version on MMC/SD (if not 0)
	ZINFO2: File version of the SBUS module (if not 0)
	ZINFO3: Slot
	DatID: Not user relevant
0xEAA0	Internal error - Please contact the hotline!
	OB: Current operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1: Diagnostic address of the master
	ZINFO2: Not user relevant
	ZINFO3: Number of errors which occurred
0xEAB0	Invalid link mode
	OB: Current operating mode
	0: Configuration in operating condition RUN

Event ID	Description
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1: Diagnostic address of the master
	ZINFO2: Current connection mode
	1: 10Mbit half-duplex
	2: 10Mbit full-duplex
	3: 100Mbit half-duplex
	4: 100Mbit full-duplex
	5: Connection mode undefined
	6: Auto Negotiation
0xEAC0	Internal error - Please contact the hotline!
	ZINFO1: Error code
	2: Internal error
	3: Internal error
	4: Internal error
	5: Internal error
	6: Internal error
	7: Internal error
	8: Internal error
	8: Internal error
0xEAD0	SyncUnit configuration error
	ZINFO1: Status
0xEB02	System error: Preset configuration does not match actual configuration

Event ID	Description
	ZINFO1: Bit mask slots 1-16
	ZINFO2: Bit mask slots 17-32
	ZINFO3: Bit mask slots 33-48
	DatID: Bit mask slots 49-64
0xEB03	System error: IO mapping
	PK: Not user relevant
	ZINFO1: Error type
	1: SDB parser error
	2: Configured address already used
	3: Mapping error
	ZINFO2: Slot (0=cannot be determined)
	DatID: Not user relevant
0xEB04	Bus: Multiple configuration of a periphery address
	ZINFO1: Periphery address
	ZINFO2: Slot
	DatID: Input
	DatID: Output
0xEB05	System error: Bus structure for isochronous process image not suitable
	PK: Not user relevant
	ZINFO2: Slot (0=cannot be determined)
	DatID: Not user relevant
0xEB06	System error: Timeout with the isochronous process image
0xEB10	System error: Bus error
	PK: Not user relevant
	ZINFO1: Error type
	96: Bus enumeration error
	128: General error
	129: Queue execution error
	130: Error interrupt
	ZINFO2: Error on bus enumeration error (ZINFO1)
	DatID: Not user relevant
0xEB11	System error: Error during bus initialization
	PK: Not user relevant
	DatID: Not user relevant
0xEB15	Bus FMM message
	ZINFO1: FMM message
0xEB20	System error: Interrupt information undefined
0xEB21	System error: Accessing configuration data

Event ID	Description
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEC02	EtherCAT: Configuration warning
	ZINFO1: Error code
	1: Number of slave stations is not supported
	2: Master system ID invalid
	3: Slot invalid
	4: Master configuration invalid
	5: Master type invalid
	6: Slave diagnostic address invalid
	7: Slave address invalid
	8: Slave module IO configuration invalid
	9: Logical address already in use
	10: Internal error
	11: IO mapping error
	12: Error
	13: Error in initialising the EtherCAT stack (is entered by the CP)
	14: Slave station number already occupied by virtual device
	ZINFO2: Station number
0xEC03	EtherCAT: Configuration error
	PK: Not user relevant
	ZINFO1: Error code
	1: Number of slave stations is not supported
	2: Master system ID invalid
	3: Slot invalid
	4: Master configuration invalid
	5: Master type invalid
	6: Slave diagnostic address invalid
	7: Slave address invalid
	8: Slave module IO configuration invalid
	9: Logical address already in use
	10: Internal error
	11: IO mapping error
	12: Error
	13: Error in initialising the EtherCAT stack (is entered by the CP)
	14: Slave station number already occupied by virtual device
	ZINFO2: Station number

Event ID	Description
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEC04	EtherCAT: Multiple configuration of a periphery address
	PK: Not user relevant
	ZINFO1: Periphery address
	ZINFO2: Slot
	DatID: Input
	DatID: Output
0xEC05	EtherCAT: Check the set DC mode of the YASKAWA Sigma 5/7 drive
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	PK: Not user relevant
	ZINFO1: Station address of the EtherCAT device
	ZINFO2: Error code
	1: WARNING: For the drive the DC Beckhoff mode is recommended (DC reference clock is not in Beckhoff Mode)!
	2: NOTE: For the drive the DC Hilscher mode is recommended (DC reference clock is not in Beckhoff Mode)!
	3: The station address could not be determined for checking (station address in ZINFO1 is accordingly 0)
	4: The slave information could not be determined for checking (station address in ZINFO1 is accordingly 0)
	5: The EtherCAT status of the drive could not be determined

Event ID	Description
	6: Error when sending the SDO request (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	7: Drive returns error in the SDO response (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	8: SDO time out, DC mode could not be determined (for further information, the (subsequent) event with the ID 0xED60 is to be analysed on the CP)
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEC10	EtherCAT: Recovery bus with all slaves
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the station
	ZINFO3: Number of stations, which are not in the same state as the master
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xEC11	EtherCAT: Recovery bus with missing slaves
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init

Event ID	Description
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the master
	ZINFO3: Number of stations which are not in the same state as the master
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xEC12	EtherCAT: Recovery slave
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the station
	ZINFO3: AL status code
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xEC30	EtherCAT: Topology OK
	ZINFO2: Diagnostic address of the master
0xEC40	Bus cycle time infringement resolved
	ZINFO2: Logical address of the IO system
0xEC50	EtherCAT: Distributed clocks (DC) out of sync
	OB: Operating mode
	0: Configuration in operating condition RUN

Event ID	Description
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO2: Diagnostic address of the master
	ZINFO3: DC state change
	0: DC master out of sync
	1: DC slave stations out of sync
0xEC80	EtherCAT: Bus error resolved
	ZINFO1: Logical address of the IO system
	ZINFO3 - Position 0: Station number
	ZINFO3 - Position 11: IO system ID
	ZINFO3 - Bit 15: System ID DP/PN
0xED10	EtherCAT: Breakdown bus
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp

Event ID	Description
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the master
	ZINFO3: Number of stations which are not in the same state as the master
	DatID: Station available
	DatID: Station not available
	DatID: Input address
	DatID: Output address
0xED12	EtherCAT: Breakdown slave
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the station
	ZINFO3: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog

Event ID	Description
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xED20	EtherCAT: Bus state change without calling OB86
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp

Event ID	Description
	8: Op
	ZINFO2: Diagnostic address of the master
	ZINFO3: Number of stations which are not in the same state as the master
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xED21	EtherCAT: Incorrect bus status change
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the master
	ZINFO3: Error code
	4: Cancel (master state change)
	8: Busy
	11: Invalid parameters
	14: Invalid status
	16: Time out
	DatID: Station available
	DatID: Station not available
	DatID: Output address
	DatID: Input address
0xED22	EtherCAT: Slave status change that does not generate an OB86
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp

Event ID	Description
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO1 - Position 8: Previous status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Diagnostic address of the station
	ZINFO3: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error

Event ID	Description
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID: Station not available
	DatID: Station available
	DatID: Input address
	DatID: Output address
0xED23	EtherCAT: Time out while changing the master state to OP, after CPU has changed to RUN
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1: Master status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: EtherCAT configuration present

Event ID	Description
	0: There is no EC configuration
	1: There is an EC configuration
	ZINFO3: DC in sync
	0: Not in sync
	1: In sync
0xED30	EtherCAT: Topology deviation
	ZINFO2: Diagnostic address of the master
0xED31	EtherCAT: Overflow of the interrupt queue
	ZINFO2: Diagnostic address of the master
0xED40	Bus cycle time infringement occurred
	ZINFO1: Logical address of the IO system
0xED50	EtherCAT: Distributed clocks (DC) in sync
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO2: Diagnostic address of the master
	ZINFO3: DC state change
	0: Master
	1: Slave
0xED60	EtherCAT: Diagnostic buffer CP: Slave status change
	OB: Operating mode
	0: Configuration in operating condition RUN

Event ID	Description
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1 - Position 0: New status
	0: Undefined/Unkown
	1: Init
	2: PreOp
	3: Bootstrap
	4: SafeOp
	8: Op
	ZINFO2: Slave address
	ZINFO3: AlStatusCode
	0: No error
	1: Unspecified error
	17: Invalid requested status change
	18: Unknown requested status
	19: Bootstrap not supported
	20: No valid firmware
	22: Invalid mailbox configuration
	23: Invalid sync manager configuration
	24: No valid inputs available
	25: No valid outputs available
	26: Synchronisation error
	27: Sync manager watchdog

Event ID	Description
	28: Invalid sync manager types
	29: Invalid output configuration
	30: Invalid input configuration
	31: Invalid watchdog configuration
	32: Slave station needs cold start
	33: Slave station needs to be in INIT state
	34: Slave station needs to be in PreOp state
	35: Slave station needs to be in SafeOp state
	45: Invalid output FMMU configuration
	46: Invalid input FMMU configuration
	48: Invalid DC Sync configuration
	49: Invalid DC Latch configuration
	50: PLL error
	51: Invalid DC IO error
	52: Invalid DC time out error
	66: Error in acyclic data exchange Ethernet Over EtherCAT
	67: Error in acyclic data exchange CAN Over EtherCAT
	68: Error in acyclic data exchange Fileaccess Over EtherCAT
	69: Error in acyclic data exchange Servo Drive Profile Over EtherCAT
	79: Error in acyclic data exchange Vendorspecific Over EtherCAT
	DatID: Cause for slave status change
	0: Regular slave status change
	1: Slave failure
	2: Recovery slave
	3: Slave is in an error state
	4: Slave has unexpectedly changed its status
0xED61	EtherCAT: Diagnostic buffer CP: CoE emergency
	OB: EtherCAT station address (high byte)
	PK: EtherCAT station address (low byte)
	ZINFO1 - Position 0: Error register
	ZINFO1 - Position 8: MEF-Byte1
	ZINFO2 - Position 0: MEF-Byte2
	ZINFO2 - Position 8: MEF-Byte3
	ZINFO3 - Position 0: MEF-Byte4
	ZINFO3 - Position 8: MEF-Byte5
	DatID: Error code
0xED62	EtherCAT: Diagnostic buffer CP: Error on SDO access
	OB: EtherCAT station address (high byte)

Event ID	Description
	PK: EtherCAT station address (low byte)
	ZINFO1: Index
	ZINFO2: SDO error code (high word)
	ZINFO3: SDO error code (low word)
	DatID: Sub index
0xED63	EtherCAT: Diagnostic buffer CP: Error in the response to an INIT command
	OB: EtherCAT station address (high byte)
	PK: EtherCAT station address (low byte)
	ZINFO1: Error type
	0: Not defined
	1: No response
	2: Validation error
	3: INIT command failed, requested station could not be reached
0xED70	EtherCAT: Diagnostic buffer CP: Twofold hot connect group recognised
	OB: Operating mode
	0: Configuration in operating condition RUN
	1: STOP (update)
	2: STOP (memory reset)
	3: STOP (auto initialization)
	4: STOP (internal)
	5: STARTUP (cold start)
	6: STARTUP (restart/warm start)
	7: STARTUP (hot restart)
	9: RUN
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Error search mode
	15: De-energised
	253: Process image release in STOP
	254: Watchdog
	255: Not set
	ZINFO1: Diagnostic address of the master
	ZINFO2: EtherCAT station address
0xED80	Bus error occurred (receive time-out)
	ZINFO1: Logical address of the IO system
	ZINFO3 - Position 0: Station number

Event ID	Description
	ZINFO3 - Position 11: IO system ID
	ZINFO3 - Bit 15: System ID DP/PN
0xEE00	Additional information at UNDEF_OPCODE
	OB: Not user relevant
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEE01	Internal error - Please contact the hotline!
	ZINFO3: SFB number
0xEEEE	CPU was completely deleted, since after PowerON the start-up could not be finished
0xEF00	Internal error - Please contact the hotline!
	DatID: Not user relevant
0xEF01	Internal error - Please contact the hotline!
	ZINFO1: Not user relevant
	ZINFO2: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEF11	Internal error - Please contact the hotline!
0xEF12	Internal error - Please contact the hotline!
0xEF13	Internal error - Please contact the hotline!
0xEFFE	Internal error - Please contact the hotline!
	PK: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xEFFF	Internal error - Please contact the hotline!
	PK: Not user relevant
	ZINFO3: Not user relevant
	DatID: Not user relevant
0xF9C1	Restart of the component
	OB: NCM_EVENT
	1: OVS: Component start-up request was denied
	3: Component data basis invalid
	6: IP_CONFIG: New IP address assigned by STEP7 configuration
	10: IP_CONFIG: A non-configured new IP address was assigned
	13: HW reset at P bus (for CPU memory reset)
	19: Switch actuation from STOP to RUN causes the restart of the component
	20: MGT: PG command causes the restart of the component

Event ID	Description
	21: MGT: Take-over of component data basis causes the hot restart of the component
	23: Stopping the sub-system after having loaded the already existing consistency-secured SDBs xxxx by the rack component
	25: The SIMATIC procedure has been selected for the time synchronisation of the component.
	26: Component actively established a connection
	28: The SDB xxxx loaded by the rack component is the consistency securing object (SDB type 0x3118)
	29: The component actively disconnected the system connection to the CPU
	31: Inconsistency of the component data base by loading SDB xxxx by the rack component (SDB type 0x3100)
	32: Periphery enabled by S7-CPU
	33: Periphery disabled by S7-CPU
	34: Component STOP due to switch actuation
	35: Component STOP due to invalid configuration
	36: Component STOP due to PG command
	38: SDB xxxx is not registered in the still valid consistency securing object, or it has an incorrect time stamp (SDB type 0x3107), the error is being corrected
	40: Memory reset executed
	44: Consistency of the data base achieved after loading the SDBs xxxx by the rack component (SDB type xxxx)
	45: Remanent part of the component data base is deleted by the rack component after being loaded
	70: Restore factory defaults (same as memory reset of CPU!)
	83: Network interface: automatic configuration, TP/ITP with 10 Mbit/s semi-duplex
	96: The MAC address was retrieved from the system SDB. This is the configured address.
	97: The MAC address was retrieved from the boot EPROM. This is the factory-provided address.
	100: Restart of the component
	101: Component STOP due to deletion of system SDBs
	104: PG command start was denied due to missing or inconsistent configuration
	105: Component STOP due to double IP address
	107: Start-up request by switch actuation was denied due to missing or inconsistent configuration
	PK: NCM_SERVICE
	2: Management
	3: Object management system
	6: Time synchronisation
	10: IP_CONFIG
	38: SEND/RECEIVE

B Integrated blocks

More information about this may be found in the manual "SPEED7 Operation List".

OB	Name	Description
OB 1	CYCL_EXC	Program Cycle
OB 10	TOD_INT0	Time-of-day Interrupt
OB 20	DEL_INT0	Time delay interrupt
OB 21	DEL_INT1	Time delay interrupt
OB 28	CYC_INT_250us	Cyclic interrupt
OB 29	CYC_INT_500us	Cyclic interrupt
OB 32	CYC_INT2	Cyclic interrupt
OB 33	CYC_INT3	Cyclic interrupt
OB 34	CYC_INT4	Cyclic interrupt
OB 35	CYC_INT5	Cyclic interrupt
OB 40	HW_INT0	Hardware interrupt
OB 55	DP: STATUS ALARM	Status interrupt
OB 56	DP: UPDATE ALARM	Update interrupt
OB 57	DP: MANUFACTURE ALARM	Vendor specific interrupt
OB 60	CYCL_EXC_FOR_SYNC_1	Multi-computing interrupt
OB 61	SYNC_1	Synchronous cycle interrupt
OB 80	CYCL_FLT	Time error
OB 81	PS_FLT	Power supply error
OB 82	I/O_FLT1	Diagnostics interrupt
OB 83	I/O_FLT2	Insert / remove module
OB 85	OBNL_FLT	Priority class error
OB 86	RACK_FLT	Slave failure / restart
OB 100	COMPLETE RESTART	Start-up
OB 102	COLD RESTART	Start-up
OB 121	PROG_ERR	Programming error
OB 122	MOD_ERR	Periphery access error

SFB	Name	Description
SFB 0	CTU	Up-counter
SFB 1	CTD	Down-counter

SFB	Name	Description
SFB 2	CTUD	Up-down counter
SFB 3	TP	Create pulse
SFB 4	TON	On-delay
SFB 5	TOF	Create turn-off delay
SFB 7	TIMEMESS	Time measurement
SFB 12	BSEND	Sending data in blocks
SFB 13	BRCV	Receiving data in blocks:
SFB 14	GET	Remote CPU read
SFB 15	PUT	Remote CPU write
SFB 32	DRUM	Realize a step-by-step switch
SFB 47	COUNT	Control counter
SFB 48	FREQUENC	Frequency measurement
SFB 49	PULSE	Pulse width modulation
SFB 52	RDREC	Read record set
SFB 53	WRREC	Write record set
SFB 54	RALRM	Receiving an interrupt from a periphery module
SFB 238	EC_RWOD	Function is used internally

SFC	Name	Description
SFC 0	SET_CLK	Set system clock
SFC 1	READ_CLK	Read system clock
SFC 2	SET_RTM	Set run-time meter
SFC 3	CTRL_RTM	Control run-time meter
SFC 4	READ_RTM	Read run-time meter
SFC 5	GADR_LGC	Logical address of a channel
SFC 6	RD_SINFO	Read start information
SFC 7	DP_PRAL	Triggering a hardware interrupt on the DP master
SFC 12	D_ACT_DP	Activating and deactivating of DP slaves
SFC 13	DPNRM_DG	Read diagnostic data of a DP salve
SFC 14	DPRD_DAT	Read consistent data
SFC 15	DPWR_DAT	Write consistent data
SFC 17	ALARM_SQ	ALARM_SQ
SFC 18	ALARM_SQ	ALARM_S
SFC 19	ALARM_SC	Acknowledgement state last alarm
SFC 20	BLKMOV	Block move
SFC 21	FILL	Fill a field
SFC 22	CREAT_DB	Create a data block

SFC	Name	Description
SFC 23	DEL_DB	Deleting a data block
SFC 24	TEST_DB	Test data block
SFC 28	SET_TINT	Set time-of-day interrupt
SFC 29	CAN_TINT	Cancel time-of-day interrupt
SFC 30	ACT_TINT	Activate time-of-day interrupt
SFC 31	QRY_TINT	Query time-of-day interrupt
SFC 32	SRT_DINT	Start time-delay interrupt
SFC 33	CAN_DINT	Cancel time-delay interrupt
SFC 34	QRY_DINT	Query time-delay interrupt
SFC 36	MSK_FLT	Mask synchronous errors
SFC 37	MSK_FLT	Unmask synchronous errors
SFC 38	READ_ERR	Read error register
SFC 39	DIS_IRT	Disabling interrupts
SFC 40	EN_IRT	Enabling interrupts
SFC 41	DIS_AIRT	Delaying interrupts
SFC 42	EN_AIRT	Enabling delayed interrupts
SFC 43	RE_TRIGR	Re-trigger the watchdog
SFC 44	REPL_VAL	Replace value to ACCU1
SFC 46	STP	STOP the CPU
SFC 47	WAIT	Delay the application program
SFC 49	LGC_GADR	Read the slot address
SFC 51	RDSYSST	Read system status list SSL
SFC 52	WR_USMSG	Write user entry into diagnostic buffer
SFC 53	µS_TICK	Time measurement
SFC 54	RD_DPARM	Reading predefined parameters
SFC 55	WR_PARM	Write dynamic parameter
SFC 56	WR_DPARM	Write default parameter
SFC 57	PARM_MOD	Parametrize module
SFC 58	WR_REC	Write record set
SFC 59	RD_REC	Read record set
SFC 64	TIME_TCK	Read system time tick
SFC 65	X_SEND	Sending data
SFC 66	X_RCV	Receiving data
SFC 67	X_GET	Read data
SFC 68	X_PUT	Write data
SFC 69	X_ABORT	Disconnect

SFC	Name	Description
SFC 70	GEO_LOG	Determining the start address of a module
SFC 71	LOG_GEO	Determining the slot belonging to a logical address
SFC 81	UBLKMOV	Copy data area without gaps
SFC 101	HTL_RTM	Handling runtime meters
SFC 102	RD_DPARA	Reading predefined parameters
SFC 105	READ_SI	Reading dynamic system resources
SFC 106	DEL_SI	Releasing dynamic system resources
SFC 107	ALARM_DQ	ALARM_DQ
SFC 108	ALARM_DQ	ALARM_DQ

C SSL partial list

More information about this may be found in the manual "SPEED7 Operation List".

SSL-ID	SSL partial list
xy11h	Module identification
xy12h	CPU characteristics
xy13h	User memory areas
xy14h	System areas
xy15h	Block Types
xy19h	Status of all LEDs
xy1Ch	Identification of the component
xy22h	Interrupt status
xy32h	Communication status data
xy37h	Ethernet details of the module
xy3Ah	Status of the TCON Connections
xy3Eh	Web server diagnostic information
xy3Fh	Configuration of Access settings
xy74h	Status of the LEDs
xy91h	Status information CPU
xy92h	Stations status information (DPM)
xy94h	Stations status information (DPM, PROFINET IO and EtherCAT)
xy95h	Status information (DPM-, PROFINET IO system)
xy96h	Module status information (PROFIBUS DP, PROFINET IO, EtherCAT)
xyA0h	Diagnostic buffer of the CPU
xyB3h	Module diagnostic information (record set 1) via logical address
xyB4h	Diagnostic data of a DP slave
xyE0h	Information EtherCAT master/slave
xyE1h	EtherCAT bus system
xyFAh	Statistics information to OBs
xyFCh	Status of the VSC features from the System SLIO CPU