YASKAWA

VIPA System MICRO

CPU | M13-CCF0000 | Manual

HB400 | CPU | M13-CCF0000 | en | 16-47

SPEED7 CPU M13C

www.vipa.com/en/service-support/manual

VIPA CONTROLS

VIPA GmbH
Ohmstr. 4
91074 Herzogenaurach
Telephone: 09132-744-0
Fax: 09132-744-1864
Email: info@vipa.com
Internet: www.vipa.com

Table of contents

1 General

1.1 Copyright © VIPA GmbH

All Rights Reserved

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. Nevertheless, the authors retain the right to modify the information.

This customer document describes all the hardware units and functions known at the present time. Descriptions may be included for units which are not present at the customer site. The exact scope of delivery is described in the respective purchase contract.

CE Conformity Declaration

Hereby, VIPA 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 VIPA customer service organization.

Trademarks

VIPA, SLIO, System 100V, System 200V, System 300V, System 300S, System 400V, System 500S and Commander Compact are registered trademarks of VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SPEED7 is a registered trademark of profichip GmbH.

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

Microsoft and Windows are registered trademarks of Microsoft Inc., USA.

Portable Document Format (PDF) and Postscript are registered trademarks of Adobe Systems, Inc.

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

Information product support

Contact your local VIPA Customer Service Organization representative if you wish to report errors or questions regarding the contents of this document. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany
Telefax: +49 9132 744-1204
EMail: documentation@vipa.de

Technical support

Contact your local VIPA Customer Service Organization representative if you encounter problems with the product or have questions regarding the product. If you are unable to locate a customer service centre, contact VIPA as follows:

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany
Tel.: +49 9132 744-1150 (Hotline)
EMail: support@vipa.de

1.2 About this manual

Objective and contents

This manual describes the CPU M13-CCF0000 of the System MICRO from VIPA. It contains a description of the construction, project implementation and usage.

Product	Order number	as of state:	
		CPU-HW	CPU-FW
CPU M13C	M13-CCF0000	01	V2.0.12

Target audience

The manual is targeted at users who have a background in automation technology.

Structure of the manual

The manual consists of chapters. Every chapter provides a self-contained description of a specific topic.

Guide to the document

The following guides are available in the manual:

- An overall table of contents at the beginning of the manual
- References with page numbers

Availability

The manual is available in:

- printed form, on paper
- in electronic form as PDF-file (Adobe Acrobat Reader)

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 information for users

Handling of electrostatic sensitive modules

VIPA modules make use of highly integrated components in MOS-Technology. These components are extremely sensitive to over-voltages that can occur during electrostatic discharges. The following symbol is attached to modules that can be destroyed by electrostatic discharges.

The Symbol is located on the module, the module rack or on packing material and it indicates the presence of electrostatic sensitive equipment. It is possible that electrostatic sensitive equipment is destroyed by energies and voltages that are far less than the human threshold of perception. These voltages can occur where persons do not discharge themselves before handling electrostatic sensitive modules and they can damage components thereby, causing the module to become inoperable or unusable. Modules that have been damaged by electrostatic discharges can fail after a temperature change, mechanical shock or changes in the electrical load. Only the consequent implementation of protection devices and meticulous attention to the applicable rules and regulations for handling the respective equipment can prevent failures of electrostatic sensitive modules.

Shipping of modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatic sensitive modules

When you are conducting measurements on electrostatic sensitive modules you should take the following precautions:

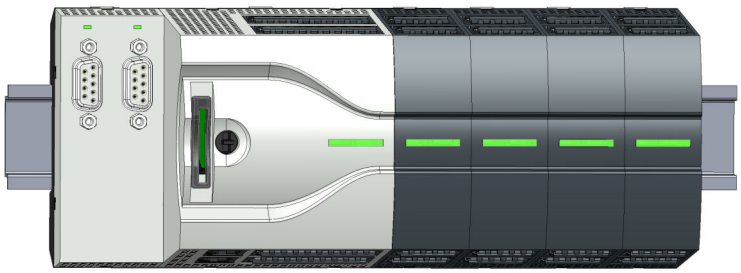
- Floating instruments must be discharged before use.
- Instruments must be grounded.

Modifying electrostatic sensitive modules you should only use soldering irons with grounded tips.

CAUTION!
Personnel and instruments should be grounded when working on electrostatic sensitive modules.

2.2 System conception

Overview

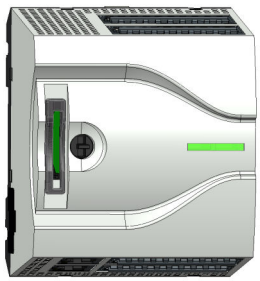


The System MICRO is a modular automation system for assembly on a 35mm mounting rail. By means of periphery modules this system may be adapted matching to your automation tasks. In addition, it is possible to expand your CPU by appropriate interfaces. The wiring complexity is low, because the DC 24V electronic section supply is integrated to the backplane bus and defective modules may be replaced with standing wire.

Components

- CPU
- Extension module
- Periphery module

CPU



With the CPU electronic, input/output components and power supply are integrated to one casing. In addition, up to 8 periphery modules of the System MICRO 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 electronic power supply of the periphery modules, which are connected via backplane bus, the CPU has removable connectors. By installing of up to 8 periphery modules at the backplane bus of the CPU, these are electrically connected, this means these are assigned to the backplane bus and connected to the DC 24V electronic power supply.

Extension module

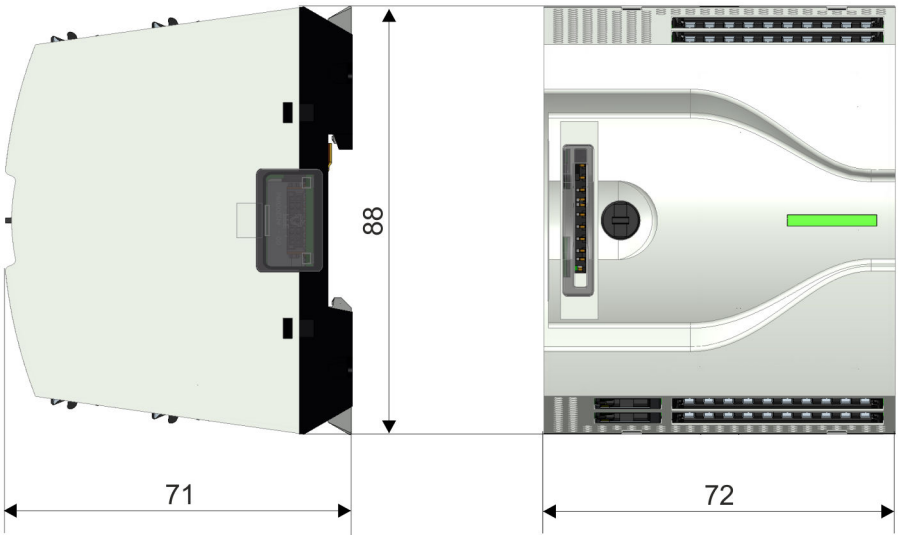
By using extension modules you can extend the interfaces of the CPU. The attachment to the CPU is made by plugging on the left side of the CPU. You can only connect one extension module to the CPU at a time.

Periphery module

By means of up to 8 periphery modules, you can extend the internal I/O areas. The attachment to the CPU is made by plugging them on the right side of the CPU.

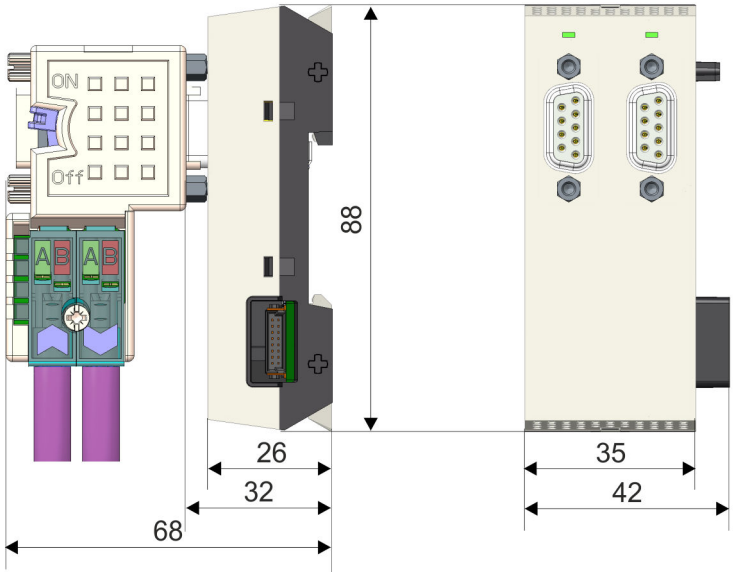
2.3 Dimensions

Dimensions CPU M13C



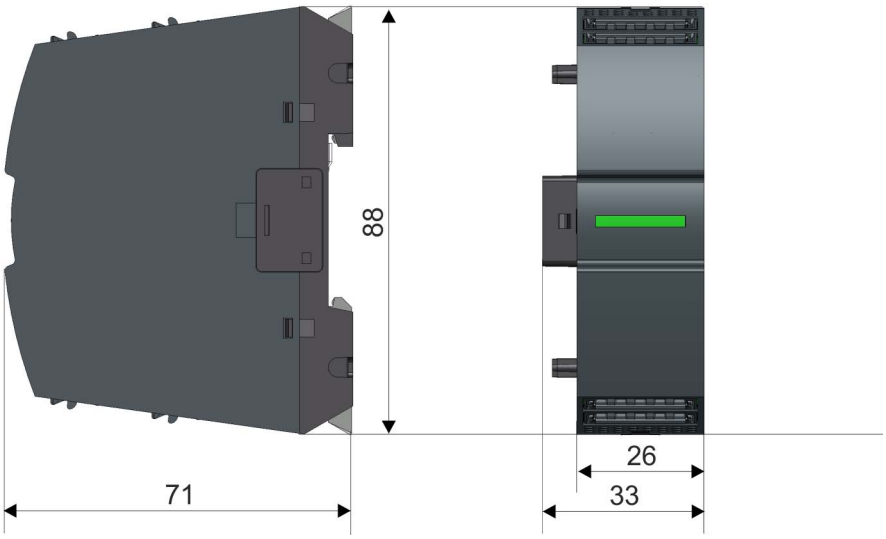
Dimensions in mm

Dimensions extension module EM M09



Dimensions in mm

Dimensions periphery module SM M2x



Dimensions in mm

2.4 Mounting

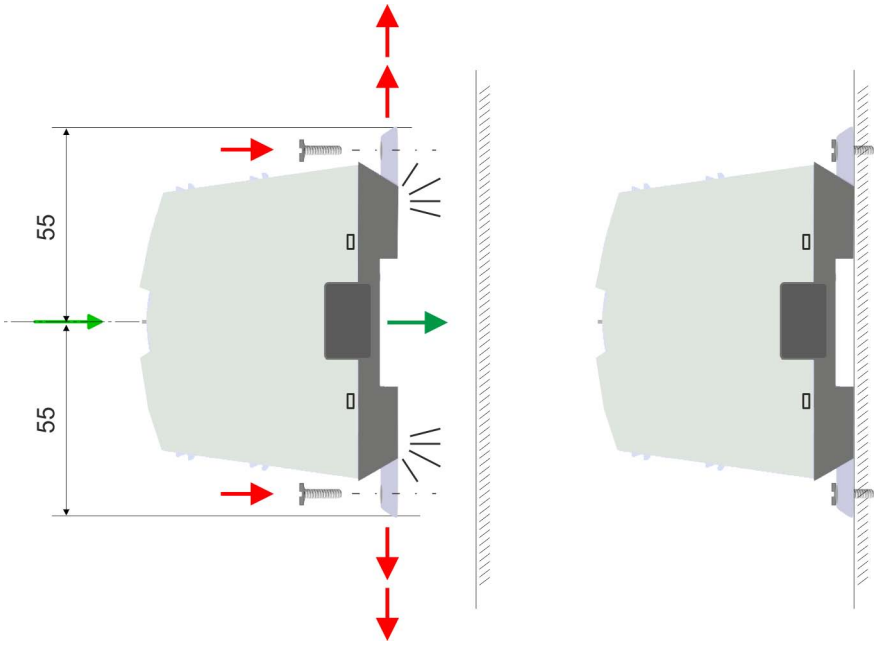
2.4.1 Mounting CPU

2.4.1.1 Mounting CPU without mounting rail

CAUTION!
Mounting without mounting rail is only permitted, if you only want to use the CPU without extension and periphery modules. Otherwise, a mounting rail must always be used for EMC technical reasons.

Proceeding

You can screw the CPU to the back wall by means of screws via the locking levers. The happens with the following proceeding:



Dimensions in mm

1. The CPU has a locking lever on the upper and lower side. Pull these levers outwards as shown in the figure, until these engage 2x audible.

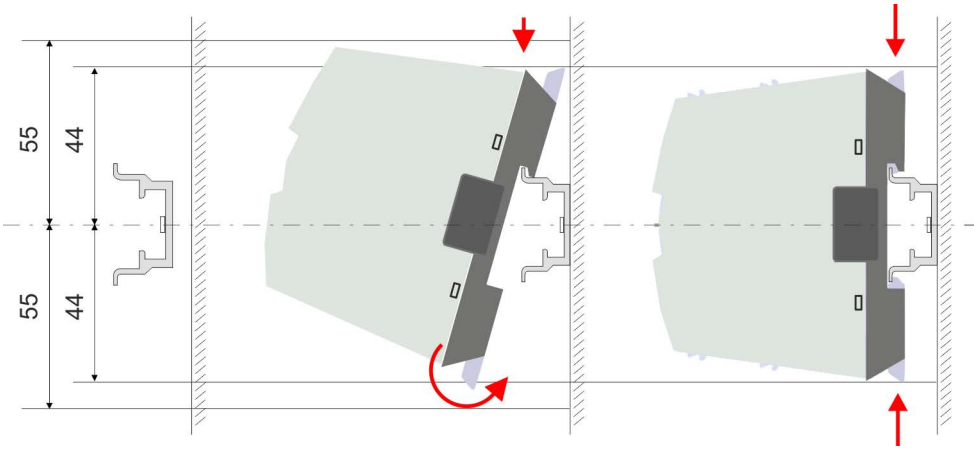
   ⇨ By this openings on the locking levers get visible.

2. Use the appropriate screws to fix your CPU to your back wall. Consider the installation clearances for the CPU.

   ⇨ The CPU is now mounted and can be wired.

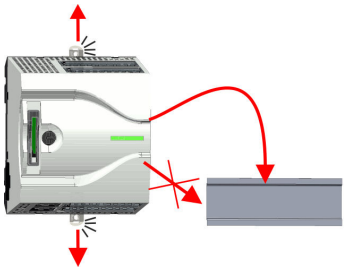
2.4.1.2 Mounting with mounting rail

Proceeding



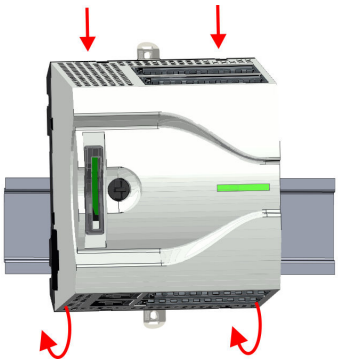
Dimensions in mm

1. Mount the mounting rail. Please consider that a clearance from the middle of the mounting rail of at least 44mm respectively 55mm above and below exists.

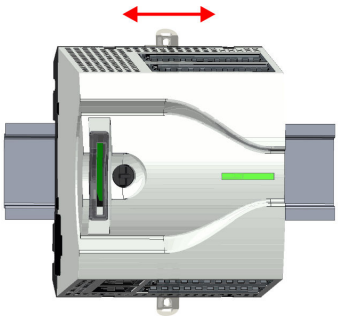


2. The CPU has a locking lever on the upper and lower side. Pull these levers outwards as shown in the figure, until these engage audible.

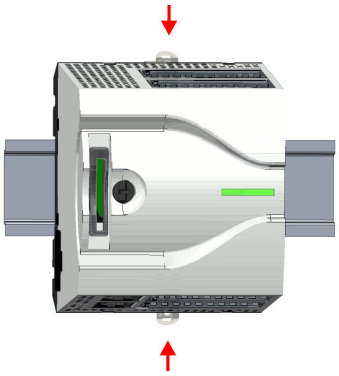
CAUTION!
It is not allowed to mount the module sideways on the mounting rail, as otherwise the module may be damaged.



3. Plug the CPU from the top onto the mounting rail and turn the periphery module downward until it rests on the mounting rail.



4. Move the CPU on the mounting rail at its position.



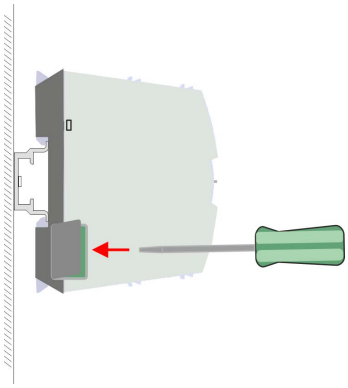
5. To fix the CPU at the mounting rail, move the locking levers back to the initial position.

⇨ The CPU is now mounted and can be wired.

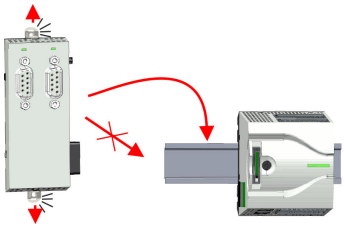
2.4.2 Mounting the extension module

Proceeding

You have the possibility to extend the interfaces of the CPU by plugging an extension module. For this the extension module is plugged at the left side of the CPU. The mountings happens with the following proceeding:

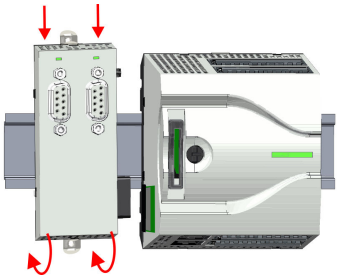


1. Remove the bus cover with a screwdriver on the left side of the CPU.

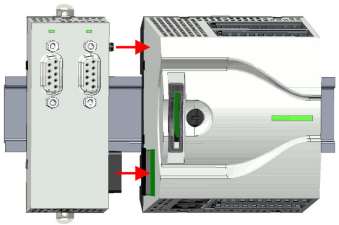


2. The extension module has a locking lever on the upper and lower side. Pull these levers outwards as shown in the figure, until these engage audible.

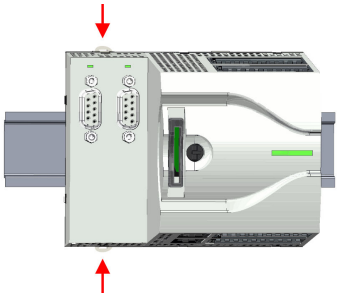
CAUTION!
It is not allowed to mount the module sideways on the mounting rail, as otherwise the module may be damaged.



3. To mount plug the extension module from the top onto the mounting rail and turn the extension module downward until it rests on the mounting rail.



4. Attach the extension module to the CPU by sliding the extension module on the mounting rail to the right until the interface connector slightly locks into the CPU.

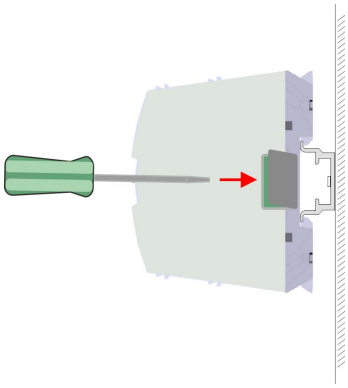


5. To fix the extension module at the mounting rail, move the locking levers back to the initial position.

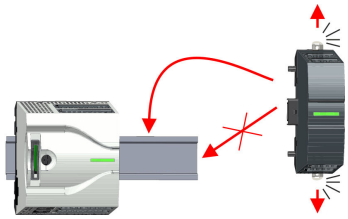
2.4.3 Mounting periphery module

Proceeding

You have the possibility to extend the periphery area of the CPU by plugging up to 8 periphery modules. For this the periphery modules are plugged at the right side of the CPU. The mountings happens with the following proceeding:

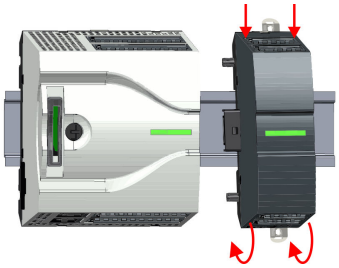


1. Remove the bus cover with a screwdriver on the right side of the CPU.

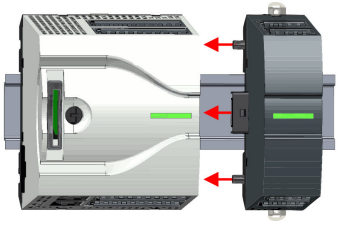


2. Each periphery module has a locking lever on its upper and lower side. Pull these levers outwards as shown in the figure, until these engage audible.

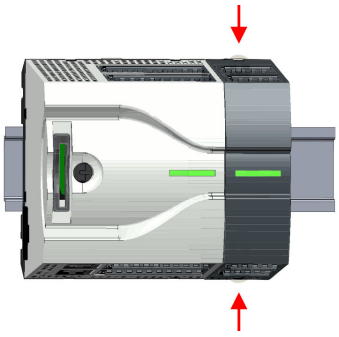
CAUTION!
It is not allowed to mount the module sideways on the mounting rail, as otherwise the module may be damaged.



3. To mount plug the periphery module from the top onto the mounting rail and turn the periphery module downward until it rests on the mounting rail.



4. Attach the periphery module to the CPU by sliding the periphery module on the mounting rail to the left until the interface connector slightly locks into the CPU.



5. To fix the periphery module at the mounting rail, move the locking levers back to the initial position.
6. Proceed in this way with additional periphery modules.

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!

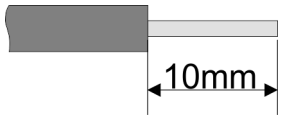
CAUTION!
Separate insulation areas!
The system is specified for SELV/PELV environment. Devices, which are attached to the system must meet theses specifications. Installation and cable routing other than SELV/PELV specification must be separated from the system's equipment!

2.5.1 Wiring CPU

CPU connector

For wiring the CPU has removable connectors. With the wiring of the connectors 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.

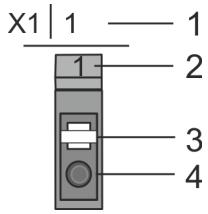
Data



U_{max}	240V AC / 30V DC
I_{max}	10A
Cross section	0.2 ... 1.5mm^2 (AWG 24 ... 16)
Stripping length	10mm

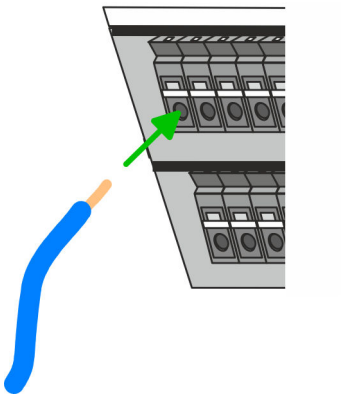
Use for wiring rigid wires respectively use wire sleeves. When using stranded wires you have to press the release button with a screwdriver during the wiring.

Wiring procedure



1 Labeling on the casing
2 Pin no. at the connector
3 Release area
4 Connection hole for wire

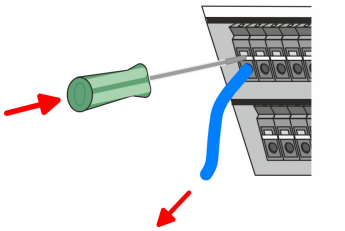
Insert wire



The wiring happens without a tool.

- Determine according to the casing labelling the connection position and 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.

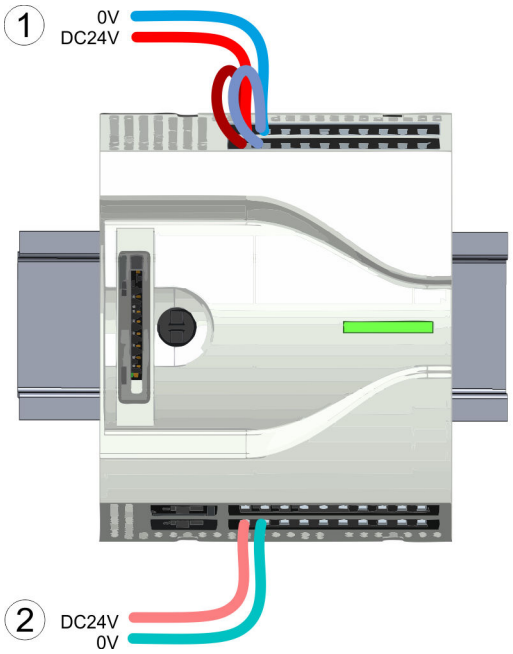
Remove wire



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.

Standard wiring



(1) X2: 4L+: DC 24V power section supply for integrated outputs
X1: 3L+: DC 24V power section supply for integrated inputs
(2) X6: 1L+ DC 24V for electronic power supply

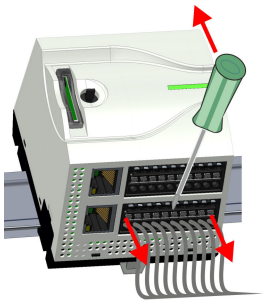
The electronic power section supply is internally protected against higher voltage by fuse. The fuse is located inside the CPU and can not be changed by the user.

Fusing

- It is recommended to externally protect the electronic power supply for CPU and backplane bus with a 3A fuse (fast) respectively by a line circuit breaker 3A characteristics Z.
- The power section supply of the internal I/Os is to be externally protected with a 6A fuse (fast) respectively by a line circuit breaker 6A characteristics Z.

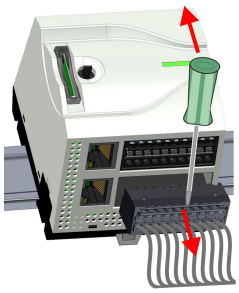
Remove connector

By means of a screwdriver there is the possibility to remove the connectors e.g. for module exchange with a fix wiring. For this each connector has indentations for unlocking at the top. Unlocking takes place by the following proceeding:



1. Remove connector:

Insert your screwdriver from above into one of the indentations.



2. Push the screwdriver backwards:

⇨ The connector is unlocked and can be removed.

CAUTION!
Via wrong operation such as pressing, the screwdriver downward the release lever may be damaged.

3. Plug connector:

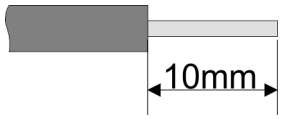
The connector is plugged by plugging it directly into the release lever.

2.5.2 Wiring periphery module

Periphery module connector

For wiring the periphery m module has removable connectors. With the wiring of the connectors 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.

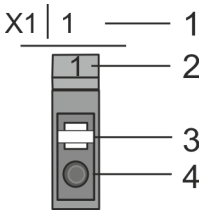
Data



U_{max}	240V AC / 30V DC
I_{max}	10A
Cross section	0.2 ... 1.5mm² (AWG 24 ... 16)
Stripping length	10mm

Use for wiring rigid wires respectively use wire sleeves. When using stranded wires you have to press the release button with a screwdriver during the wiring.

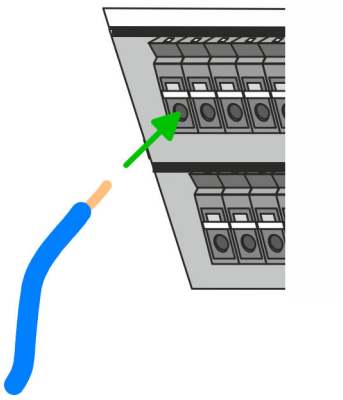
Wiring procedure



1 Labeling on the casing
2 Pin no. at the connector
3 Release area
4 Connection hole for wire

Insert wire

The wiring happens without a tool.

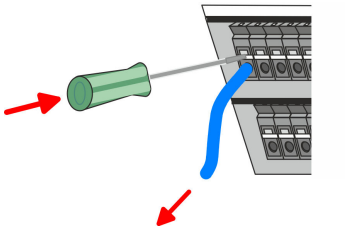


→ Determine according to the casing labelling the connection position and insert through the round connection hole of the according contact your prepared wire until it stops, so that it is fixed.

⇨ 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.



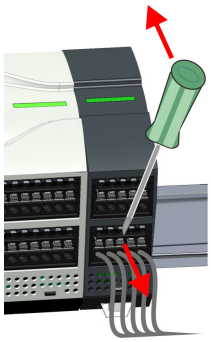
1. → Press with your screwdriver vertically at the release button.

⇨ The contact spring releases the wire.

2. → Pull the wire from the round hole.

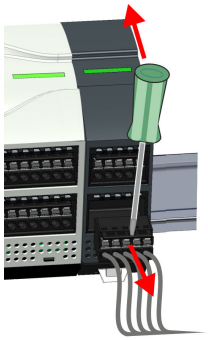
Remove connector

By means of a screwdriver there is the possibility to remove the connectors e.g. for module exchange with a fix wiring. For this each connector has indentations for unlocking at the top. Unlocking takes place by the following proceeding:



1. → Remove connector:

Insert your screwdriver from above into one of the indentations.



2. Push the screwdriver backwards:
   ⇨ The connector is unlocked and can be removed.

> **CAUTION!**
> Via wrong operation such as pressing, the screwdriver downward the release lever may be damaged.

3. Plug connector:
   The connector is plugged by plugging it directly into the release lever.

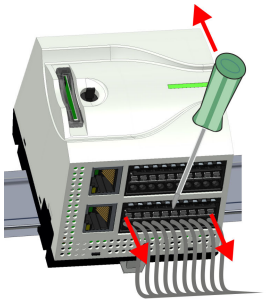
2.6 Demounting

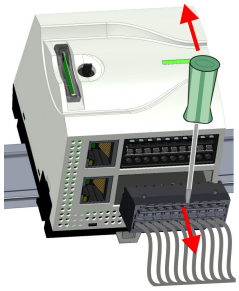
2.6.1 Demounting CPU

Remove connector

By means of a screwdriver there is the possibility to remove the connectors e.g. for module exchange with a fix wiring. For this each connector has indentations for unlocking at the top. Unlocking takes place by the following proceeding:

1. Power-off your system.
2. Remove connector:
   Insert your screwdriver from above into one of the indentations.





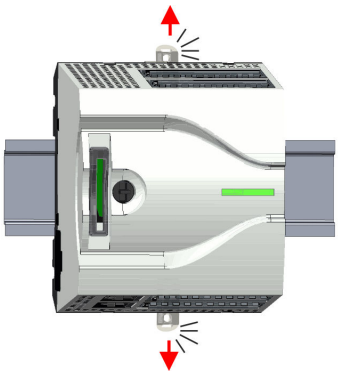
3. Push the screwdriver backwards:
   ⇨ The connector is unlocked and can be removed.

> **CAUTION!**
> Via wrong operation such as pressing, the screwdriver downward the connector may be damaged!

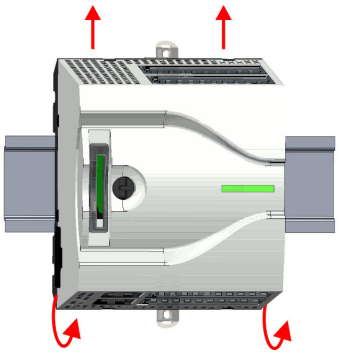
4. In this way, remove all plugged connectors on the CPU.

CPU replacement (stand-alone)

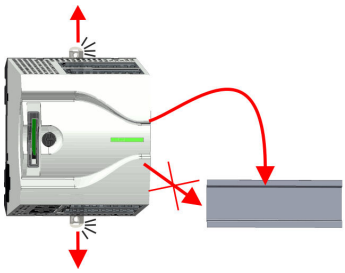
If more modules are connected to the CPU ↳ *'Option: CPU replacement in a system' on page 24*. If no other modules are connected to the CPU, the CPU is replaces according to the following proceeding:



1. Pull the locking levers of the CPU outwards until these engage audible.

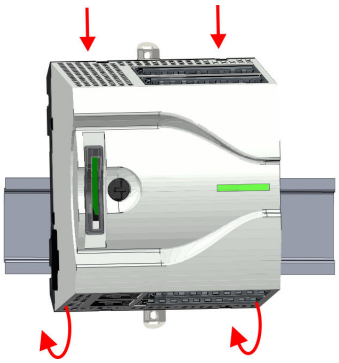


2. Remove the CPU with a rotation upwards from the mounting rail.

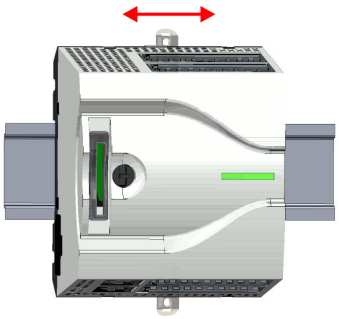


3. Pull the locking levers of the CPU outwards until these engage audible.

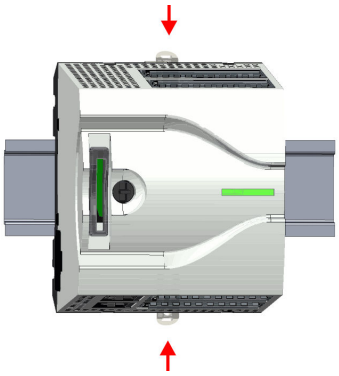
CAUTION!
It is not allowed to mount the module sideways on the mounting rail, as otherwise the module may be damaged!



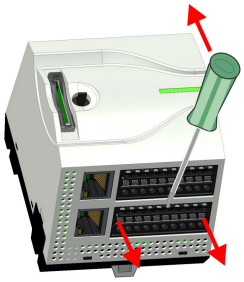
4. Plug the CPU from the top onto the mounting rail and turn the periphery module downward until it rests on the mounting rail.



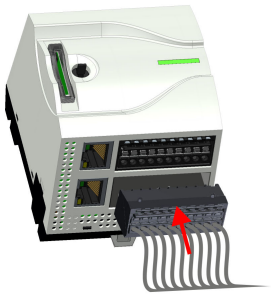
5. Move the CPU on the mounting rail at its position.



6. To fix the CPU at the mounting rail, move the locking levers back to the initial position.



7. Remove the connectors, which are not necessary at the CPU.

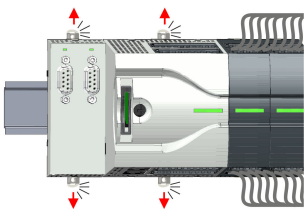


8. Plug again the wired connectors.

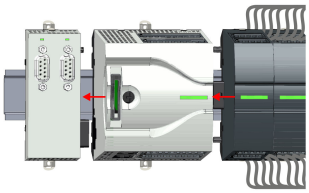
⇨ Now you can bring your system back into operation.

Option: CPU replacement in a system

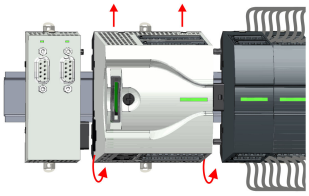
In the following the replacement of a CPU in a system is shown:



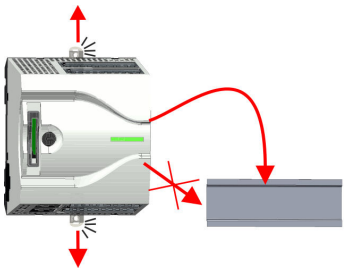
1. If there is an extension module connected to the CPU, you have to remove it from the CPU. For this pull the locking levers of the extension module and CPU outwards until these engage audible.



2. Disconnect all the modules, which are connected to the CPU by moving the CPU along with the extension module on the mounting rail.

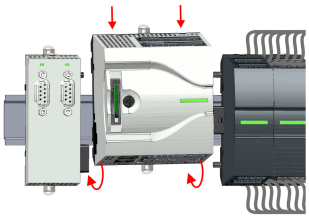


3. Remove the CPU with a rotation upwards from the mounting rail.

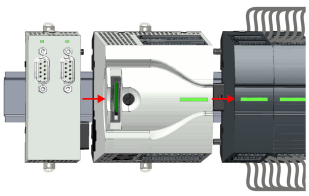


4. Pull the locking levers of the CPU outwards until these engage audible.

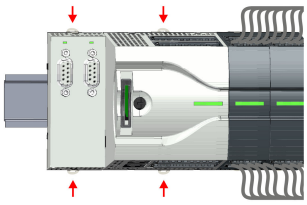
CAUTION!
It is not allowed to mount the module sideways on the mounting rail, as otherwise the module may be damaged!



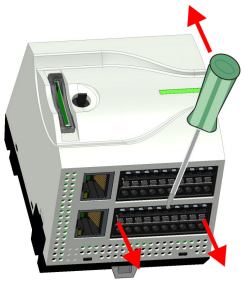
5. For mounting pull the locking levers of the CPU outwards until these engage audible. Plug the CPU from the top onto the mounting rail and turn the periphery module downward until it rests on the mounting rail.



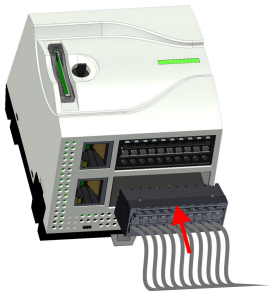
6. Rebind your modules by moving the CPU along with the extension module on the mounting rail.



7. To fix the CPU at the mounting rail, move the locking levers back to the initial position.



8. Remove the connectors, which are not necessary at the CPU.

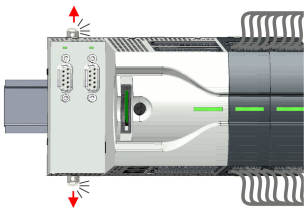


9. Plug again the wired connectors.
   ⇨ Now you can bring your system back into operation.

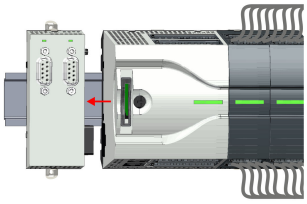
2.6.2 Demounting the extension module

Proceeding

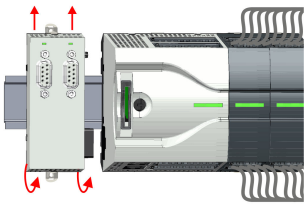
1. Power-off your system.
2. Remove the corresponding bus connectors.



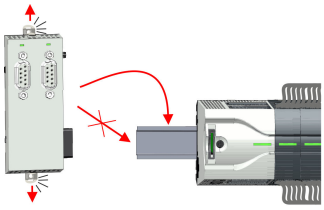
3. Pull the locking levers of the extension module outwards until these engage audible.



4. Remove the extension module from the CPU by sliding it on the mounting rail.

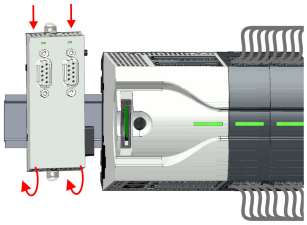


5. Remove the extension module with a rotation upwards from the mounting rail.

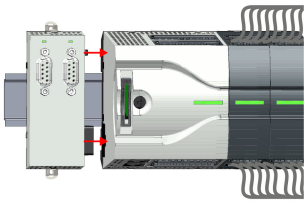


6. Pull the locking levers of the extension module outwards until these engage audible.

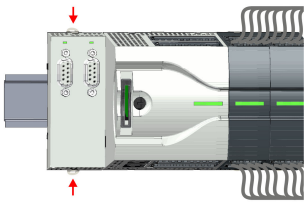
CAUTION!
It is not allowed to mount the module sideways on the mounting rail, as otherwise the module may be damaged!



7. Plug the extension module from the top onto the mounting rail and turn the extension module downward until it rests on the mounting rail.



8. Reattach the extension module to the CPU by sliding the extension module on the mounting rail to the right until the interface connector slightly locks into the CPU.



9. Move the locking levers back to the initial position.
10. Plug the corresponding bus connectors.
    ⇨ Now you can bring your system back into operation.

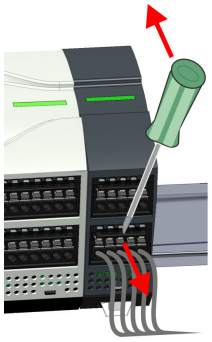
2.6.3 Demounting periphery module

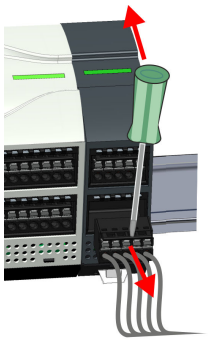
Remove connector

By means of a screwdriver there is the possibility to remove the connectors e.g. for module exchange with a fix wiring. For this each connector has indentations for unlocking at the top. Unlocking takes place by the following proceeding:

1. Power-off your system.
2. Remove connector:

   Insert your screwdriver from above into one of the indentations.



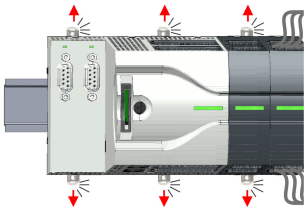


3. Push the screwdriver backwards:
   ⇨ The connector is unlocked and can be removed.

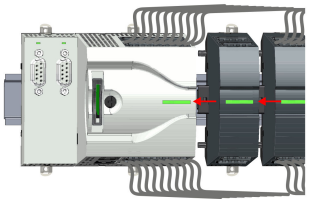
> **CAUTION!**
> Via wrong operation such as pressing, the screwdriver downward the connector may be damaged!

4. In this way, remove all plugged connectors on the periphery module.

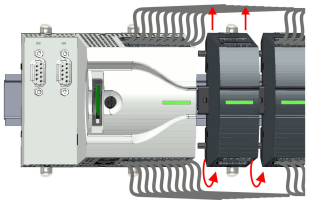
Replace the periphery module



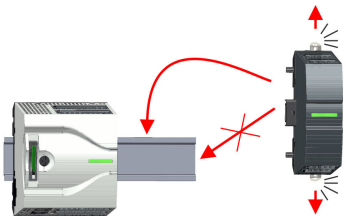
1. Remove the modules that are connected to the module to be replaced by pulling their release levers outwards until these engage audible ...



2. ... and move the modules accordingly.

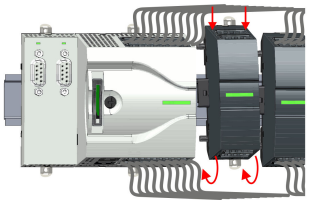


3. Remove the periphery module with a rotation upwards from the mounting rail.

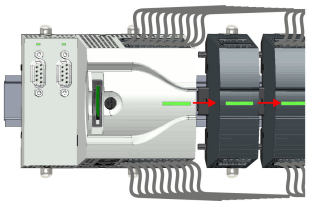


4. Pull the locking levers outwards until these engage audible.

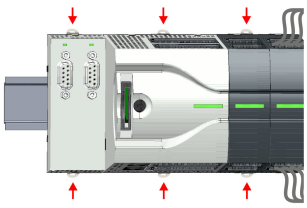
CAUTION!
It is not allowed to mount the module sideways on the mounting rail, as otherwise the module may be damaged!



5. Plug the periphery module from the top onto the mounting rail and turn the periphery module downward until it rests on the mounting rail.



6. Reconnect all modules by pushing them together again on the mounting rail.



7. Move the locking levers back to the initial position.

8. Remove the connectors, which are not necessary.

9. Plug again the wired connectors.
   ⇨ Now you can bring your system back into operation.

2.7 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 of VIPA 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 laid isolated.
  - Analog lines must be laid isolated. When transmitting signals with small amplitudes the one sided laying of the isolation may be favourable.
  - 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

2.8 General data

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

General data

Environmental conditions to EN 61131-2		
Horizontal installation lying	EN 61131-2	0…+60°C
Vertical installation	EN 61131-2	0…+60°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 *

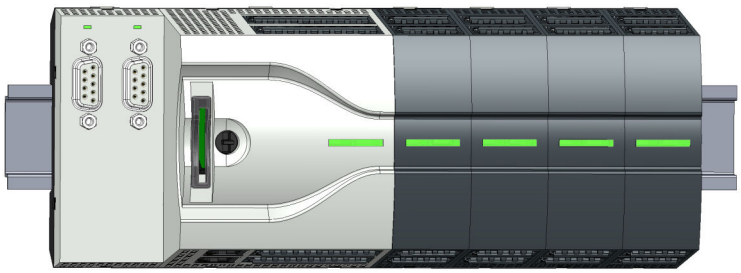
*) Due to the high-energetic single pulses with Surge an appropriate external protective circuit with lightning protection elements like conductors for lightning and overvoltage is necessary.

3 Hardware description

3.1 Properties

M13-CCF0000

- SPEED7 technology integrated
- Programmable via VIPA SPEED7 Studio, Siemens SIMATIC Manager or Siemens TIA Portal
- 64kbyte work memory integrated (32kbyte code, 32kbyte data)
- Work memory expandable up to max. 128kbyte (64kbyte code, 64kbyte data)
- 128kbyte load memory integrated
- Slot for external storage media (lockable)
- Status LEDs for operating state and diagnostics
- X1/X5: DI 16xDC24V with status indication integrated
- X2/X6: DO 12xDC24V 0.5A with status indication integrated
- X3/X4: Ethernet PG/OP channel for active and passive Communication integrated
- X6: AI 2x12Bit (single ended) integrated
- Technological functions: 4 channels for counter, frequency measurement and 2 channels for pulse width modulation
- Option: Extension module 2xRS485
- Option: max. 8 periphery modules
- I/O address area digital/analog 2048byte
- 512 timer/counter, 8192 flag byte



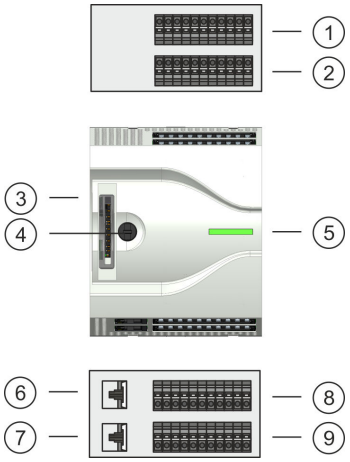
Ordering data

Type	Order number	Description
CPU M13C	M13-CCF0000	System MICRO CPU M13C with options to extend work memory, DI 16xDC24V, DO 12xDC24 0.5A, AI 2x12bit and 4 channels technological functions
EM M09	M09-0CB00	System MICRO extension: Serial interface 2x (RS485/RS422, MPI, option PROFIBUS DP slave)

3.2 Structure

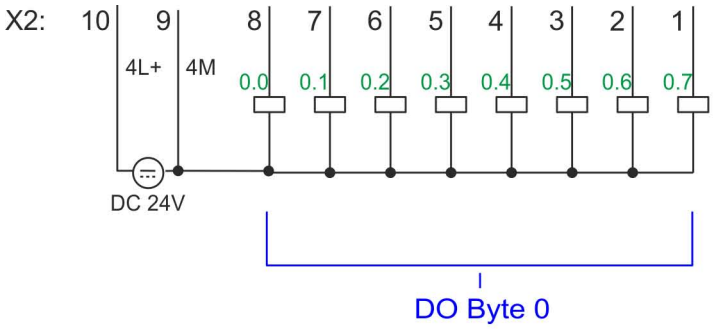
3.2.1 System MICRO CPU M13C

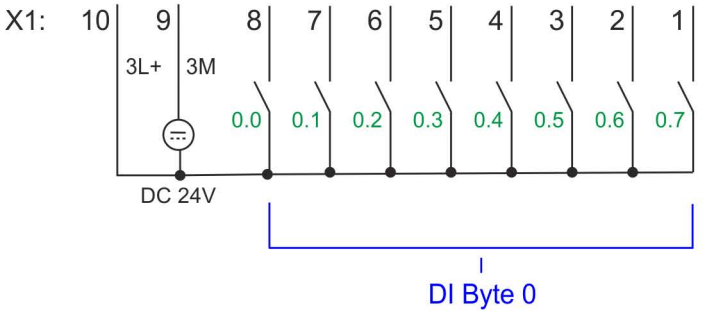
CPU M13-CCF0000



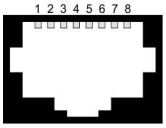
1 X2: Connector DO (byte 0)
2 X1: Connector DI (byte 0)
3 Slot for external storage media (lockable)
4 Operating mode switch CPU
5 Status bar CPU
6 X3: Ethernet PG/OP channel
7 X4: Ethernet PG/OP channel
8 X5: Connector DI (byte 1)
9 X6: Connector electronic section supply, AI, DO (byte 1)

3.2.2 Interfaces

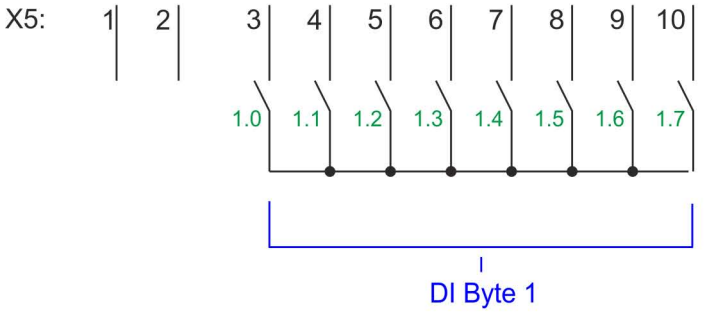




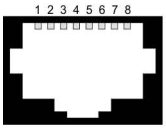
X3 PG/OP 1



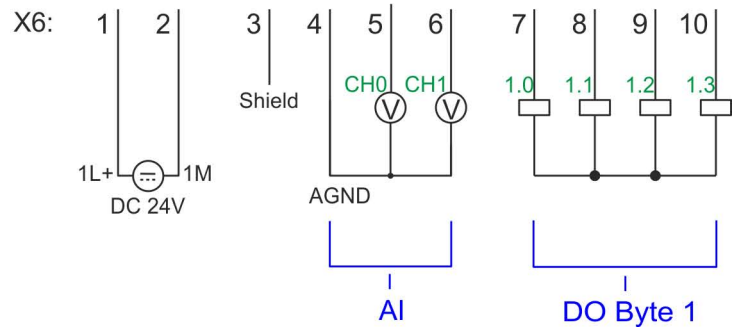
① Transmit +
② Transmit -
③ Receive +
④ -
⑤ -
⑥ Receive -
⑦ -
⑧ -



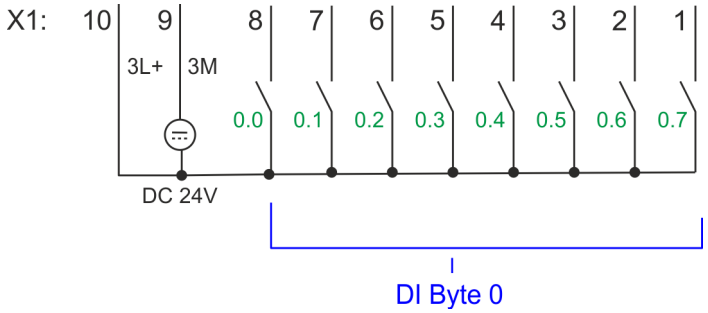
X4 PG/OP 2



① Transmit +
② Transmit -
③ Receive +
④ -
⑤ -
⑥ Receive -
⑦ -
⑧ -



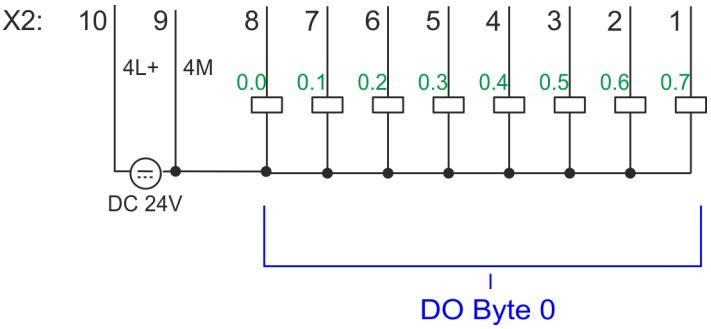
X1: DI byte 0



X1	Function	Type		Description
1	DI 0.7	I	green	Digital input DI 7 / Counter 2 (B) / Frequency 2 *
2	DI 0.6	I	green	Digital input DI 6 / Counter 2 (A) *
3	DI 0.5	I	green	Digital input DI 5
4	DI 0.4	I	green	Digital input DI 4 / Counter 1 (B) / Frequency 1 *
5	DI 0.3	I	green	Digital input DI 3 / Counter 1 (A) *
6	DI 0.2	I	green	Digital input DI 2
7	DI 0.1	I	green	Digital input DI 1 / Counter 0 (B) / Frequency 0 *
8	DI 0.0	I	green	Digital input DI 0 / Counter 0 (A) *
9	0 V	I		3M: GND for onboard DI power section supply
10	DC 24V	I	green	3L+: DC 24V for onboard DI power section supply

*) Max. input frequency 100kHz otherwise 1kHz.

X2: DO byte 0



X2	Function	Type	LED	Description
1	DO 0.7	O	green	Digital output DO 7
2	DO 0.6	O	green	Digital output DO 6
3	DO 0.5	O	green	Digital output DO 5
4	DO 0.4	O	green	Digital output DO 4
5	DO 0.3	O	green	Digital output DO 3 / Output channel counter 3
6	DO 0.2	O	green	Digital output DO 2 / Output channel counter 2
7	DO 0.1	O	green	Digital output DO 1 / PWM 1 / Output channel counter 1
8	DO 0.0	O	green	Digital output DO 0 / PWM 0 / Output channel counter 0
9	0 V	I	red	4M: GND for onboard DO power section supply / GND PWM LED (red) is on at short circuit respectively overload
10	DC 24V	I	green	4L+: DC 24V for onboard DO power section supply

X3/X4: Ethernet PG/OP channel

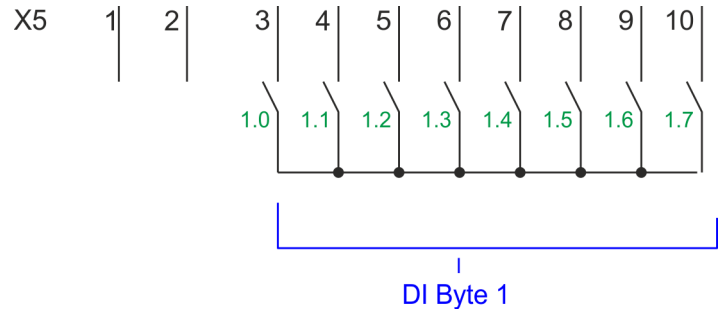
8pin RJ45 jack:

- 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.
- Configurable connections are possible.
- The connection happens via an integrated 2-port switch
- DHCP respectively the assignment of the network configuration with a DHCP server is supported.
- Default diagnostics addresses: 2025 ... 2040
- For online access to the CPU via Ethernet PG/OP channel, you have to assign IP address parameters to this.

↳ *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*

L/A (Link/Activity)	S (Speed)	Description
green	X	The Ethernet PG/OP channel is physically connected to the Ethernet interface.
	X	There is no physical connection.
green blinking	X	Blinking: Shows Ethernet activity.
green	yellow	The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 100Mbit.
green		The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 10Mbit.
not relevant: X		

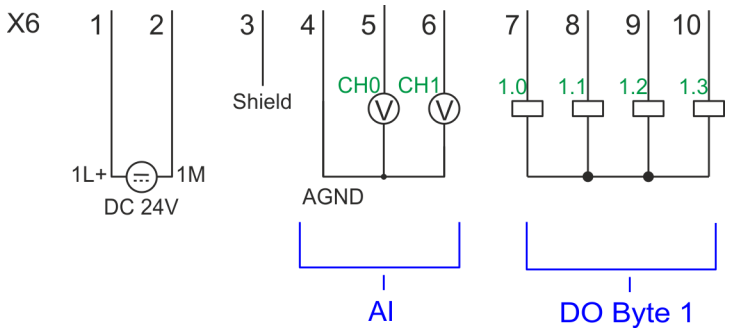
X5: DI byte 1



X5	Function	Type	LED	Description
1	-	-		reserved
2	-	-		reserved
3	DI 1.0	I	green	Digital input DI 8
4	DI 1.1	I	green	Digital input DI 9 / Counter 3 (A) *
5	DI 1.2	I	green	Digital input DI 10 / Counter 3 (B) / Frequency 3 *
6	DI 1.3	I	green	Digital input DI 11 / Gate 3 *
7	DI 1.4	I	green	Digital input DI 12
8	DI 1.5	I	green	Digital input DI 13
9	DI 1.6	I	green	Digital input DI 14
10	DI 1.7	I	green	Digital input DI 15 / Latch 3 *

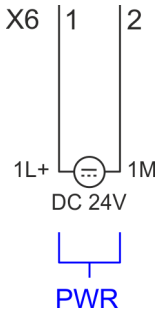
*) Max. input frequency 100kHz otherwise 1kHz.

X6: DC 24V, AI, DO byte 1



X6	Function	Type	LED	Description
1	Sys DC 24V	I	green	1L+: DC 24V for electronic section supply
2	Sys 0V	I		1M: GND for electronic section supply
3	Shield	I		Shield
4	AGND	I		GND for analog inputs
5	AI 0	I		Analog input AI 0
6	AI 1	I		Analog input AI 1
7	DO 1.0	O	green	Digital output DO 8
8	DO 1.1	O	green	Digital output DO 9
9	DO 1.2	O	green	Digital output DO 10
10	DO 1.3	O	green	Digital output DO 11

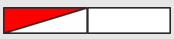
X6: Electronic power supply



The CPU has an integrated power supply. The power supply has to be provided with DC 24V. Via the power supply not only the internal electronic of the CPU is provided with voltage, but also the electronic from the integrated IO components. The power supply is protected against polarity inversion and over current.

3.2.3 LEDs

3.2.3.1 Status bar CPU

LED	Description
Start-up	
	LED yellow blinks with 1Hz: State of the CPU after PowerON
	LEDs green are blinking with 2Hz: During the start-up (OB 100) the status bar blinks for at least 3s.
Operation	
	LED yellow on: CPU is in STOP state.
	LED red on: CPU is in error state.
	LEDs green on: CPU is in RUN state without error.
	LED red blinks with 1Hz and LED green is on: CPU is in RUN state with diagnostics messages.
	LED red on and LED green blinks with 1Hz: CPU is in STOP state, configured holding point reached.
	LED red blinks with 1Hz and LED green blinks with 2Hz: Diagnostic messages detected during start-up.
	LED red on and LED green on: CPU is in error state. There is a system error or an internal error has occurred. If writing of the diagnostic buffer to the memory card has been activated by means of the auto command DIAGBUF, a long write access to the memory card happens on error. During this time the memory card may not be removed. After the write access the CPU restarts.
	LED yellow blinks with 2Hz: Hardware configuration is loaded.
	LEDs green are blinking with 1Hz: Blinking test (started via configuration tool)
	LED green on and LED green flickers: Access to the memory card in the RUN state.
	LED red blinks with 1Hz and LED green flickers: Access to the memory card with diagnostics messages in the RUN state.
	LED yellow flickers: Access to the memory card in STOP state.
Overall reset	
	LED yellow blinks with 1Hz: Overall reset is requested
	LED yellow blinks with 2Hz: Overall reset is executed.
	LED yellow on: Overall reset was successfully finished.
Factory reset	
	LED yellow blinks with 2Hz: Reset to factory setting is executed.
	LED red blinks with 1Hz and LED yellow blinks with 1Hz: Reset to factory settings was finished without errors. Please perform a power cycle!
Firmware update	
	LED red and LED yellow are alternately blinking with 1Hz: A new firmware is available on the memory card.
	LED yellow blinks with 2Hz: A firmware update is in progress.
	LED yellow flickers: Access the memory card during the firmware update.
	LED red and LED yellow are blinking with 1Hz: Firmware update finished without error. Please perform a power cycle!
	LED red blinks with 1Hz: Error during Firmware update.

3.2.3.2 LEDs I/O periphery

Digital input	LED	Description
DI +0.0 DI +0.7	green	Digital I+0.0 ... 0.7 has "1" signal
		Digital I+0.0 ... 0.7 has "0" signal
DI +1.0 ... DI +1.7	green	Digital input I+1.0 ... 1.7 has "1" signal
		Digital input I+1.0 ... 1.7 has "0" signal

Digital output	LED	Description
DO +0.0 ... DO +0.7	green	Digital output Q+0.0 ... 0.7 has "1" signal
		Digital output Q+0.0 ... 0.7 has "0" signal
DO +1.0 ... DO +1.3	green	Digital output Q+1.0 ... 1.3 has "1" signal
		Digital output Q+1.0 ... 1.3 has "0" signal

Power supply	LED	Description
1L+	green	DC 24V electronic section supply
		DC 24V electronic section supply not available
3L+	green	DC 24V power section supply inputs OK
		DC 24V power section supply inputs not available
4L+	green	DC 24V power section supply outputs OK
		DC 24V power section supply outputs not available
4M	red	Error, overload respectively short circuit on the outputs
		no error

3.2.4 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 128kbyte
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 64kbyte
  - There is the possibility to extend the work memory to its maximum capacity 128kbyte by means of a VSC.

3.2.5 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. *Chapter 4.15 'Deployment storage media - VSD, VSC' on page 79*
  - To activate the corresponding card is to be installed and a *Overall reset* is to be established. *Chapter 4.12 'Overall reset' on page 75*

A list of the currently available VSD respectively VSC can be found at www.vipa.com

3.2.6 Buffering mechanisms

The 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. *Chapter 4.19 'Diagnostic entries' on page 85*

3.2.7 Operating mode switch

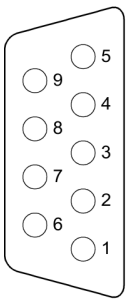
General

- 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.

3.3 Option: Extension module EM M09 2x serial interface

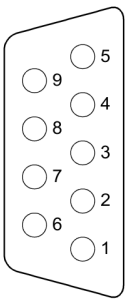
EM M09

X1 PtP



① n. c.
② TxD-P (line B) - RS422
③ RxD-P (line B) - RS422
RxD/TxD-P (line B) - RS485
④ RTS
⑤ M5V
⑥ P5V
⑦ TxD-N (line A) - RS422
⑧ RxD-N (line A) RS422
RxD/TxD-N (line A) - RS485
⑨ n.c.

X2 MPI(PB)



① n. c.
② n. c.
③ RxD/TxD-P (line B)
④ RTS
⑤ M5V
⑥ P5V
⑦ n. c.
⑧ RxD/TxD-N (line A)
⑨ n.c.

X1 PtP (RS422/485)

9pin SubD jack: (isolated)

Using the *PtP* functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source res. target systems.

- Protocols:
  - ASCII
  - STX/ETX
  - 3964R
  - USS
  - Modbus master (ASCII, RTU)
- Serial bus connection
  - Full-duplex Four-wire operation (RS422)
  - Half-duplex Two-wire operation (RS485)
  - Data transfer rate: max. 115 kBaud

↳ *Chapter 7 'Option: PtP communication' on page 155*

X2 MPI(PB)

9pin SubD jack: (isolated)

The interface supports the following functions, which are switch able:

- MPI (default / after overall reset)
  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.
- PROFIBUS DP slave (option)
  The PROFIBUS slave functionality of this interface can be activated by configuring the sub module *'MPI/DP'* of the CPU in the hardware configuration. ↳ *Chapter 8 'Option: Deployment PROFIBUS communication' on page 169*

To switch the interface X2 MPI(PB) to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is activated. ↳ *Chapter 4.15 'Deployment storage media - VSD, VSC' on page 79*

LEDs

X1 PtP TxD	Description
green flickers	Send activity
	No send activity

X2 MPI(PB) DE	Description
green	■ Slave is in DE (data exchange) ■ Slave exchanges data with the master. ■ Slave is in RUN state
green blinking	■ Slave CPU is in state start-up. ■ Slave-CPU is without master.
	■ There is no power supply. ■ Slave has no configuration.

3.4 Technical data

3.4.1 Technical data CPU

Order no.	M13-CCF0000
Type	CPU M13C
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)	120 mA
Current consumption (rated value)	360 mA
Inrush current	3 A
I^2t	0.1 A^2s
Max. current drain at backplane bus	1 A
Max. current drain load supply	-
Power loss	7 W
Technical data digital inputs	
Number of inputs	16
Cable length, shielded	1000 m
Cable length, unshielded	600 m
Rated load voltage	DC 24 V
Reverse polarity protection of rated load voltage	✓

Order no.	M13-CCF0000
Current consumption from load voltage L+ (without load)	25 mA
Rated value	DC 24 V
Input voltage for signal "0"	DC 0...5 V
Input voltage for signal "1"	DC 15...28.8 V
Input voltage hysteresis	-
Frequency range	-
Input resistance	-
Input current for signal "1"	3 mA
Connection of Two-Wire-BEROs possible	✓
Max. permissible BERO quiescent current	0.5 mA
Input delay of "0" to "1"	3 µs – 15 ms / 0.5 ms – 15 ms
Input delay of "1" to "0"	3 µs – 15 ms / 0.5 ms – 15 ms
Number of simultaneously utilizable inputs horizontal configuration	16
Number of simultaneously utilizable inputs vertical configuration	16
Input characteristic curve	IEC 61131-2, type 1
Initial data size	16 Bit
Technical data digital outputs	
Number of outputs	12
Cable length, shielded	1000 m
Cable length, unshielded	600 m
Rated load voltage	DC 24 V
Reverse polarity protection of rated load voltage	✓
Current consumption from load voltage L+ (without load)	20 mA
Total current per group, horizontal configuration, 40°C	6 A
Total current per group, horizontal configuration, 60°C	6 A
Total current per group, vertical configuration	6 A
Output voltage signal "1" at min. current	L+ (-0.8 V)
Output voltage signal "1" at max. current	L+ (-0.8 V)
Output current at signal "1", rated value	0.5 A
Output current, permitted range to 40°C	5 mA to 0.6 A
Output current, permitted range to 60°C	5 mA to 0.6 A
Output current at signal "0" max. (residual current)	0.5 mA
Output delay of "0" to "1"	2 µs / 30 µs
Output delay of "1" to "0"	3 µs / 175 µs
Minimum load current	-

Order no.	M13-CCF0000
Lamp load	10 W
Parallel switching of outputs for redundant control of a load	not possible
Parallel switching of outputs for increased power	not possible
Actuation of digital input	✓
Switching frequency with resistive load	max. 1000 Hz
Switching frequency with inductive load	max. 0.5 Hz
Switching frequency on lamp load	max. 10 Hz
Internal limitation of inductive shut-off voltage	L+ (-45 V)
Short-circuit protection of output	yes, electronic
Trigger level	1 A
Number of operating cycle of relay outputs	-
Switching capacity of contacts	-
Output data size	12 Bit
Technical data analog inputs	
Number of inputs	2
Cable length, shielded	200 m
Rated load voltage	-
Reverse polarity protection of rated load voltage	-
Current consumption from load voltage L+ (without load)	-
Voltage inputs	✓
Min. input resistance (voltage range)	100 kΩ
Input voltage ranges	0 V ... +10 V
Operational limit of voltage ranges	+/-3.5%
Operational limit of voltage ranges with SFU	-
Basic error limit voltage ranges	+/-3.0%
Basic error limit voltage ranges with SFU	-
Destruction limit voltage	max. 30V
Current inputs	-
Max. input resistance (current range)	-
Input current ranges	-
Operational limit of current ranges	-
Operational limit of current ranges with SFU	-
Basic error limit current ranges	-
Radical error limit current ranges with SFU	-
Destruction limit current inputs (electrical current)	-
Destruction limit current inputs (voltage)	-

Order no.	M13-CCF0000
Resistance inputs	-
Resistance ranges	-
Operational limit of resistor ranges	-
Operational limit of resistor ranges with SFU	-
Basic error limit	-
Basic error limit with SFU	-
Destruction limit resistance inputs	-
Resistance thermometer inputs	-
Resistance thermometer ranges	-
Operational limit of resistance thermometer ranges	-
Operational limit of resistance thermometer ranges with SFU	-
Basic error limit thermoresistor ranges	-
Basic error limit thermoresistor ranges with SFU	-
Destruction limit resistance thermometer inputs	-
Thermocouple inputs	-
Thermocouple ranges	-
Operational limit of thermocouple ranges	-
Operational limit of thermocouple ranges with SFU	-
Basic error limit thermoelement ranges	-
Basic error limit thermoelement ranges with SFU	-
Destruction limit thermocouple inputs	-
Programmable temperature compensation	-
External temperature compensation	-
Internal temperature compensation	-
Technical unit of temperature measurement	-
Resolution in bit	12
Measurement principle	successive approximation
Basic conversion time	0.5 ms
Noise suppression for frequency	40 dB
Initial data size	4 Byte
Technical data analog outputs	
Number of outputs	-
Cable length, shielded	-
Rated load voltage	-
Reverse polarity protection of rated load voltage	-

Order no.	M13-CCF0000
Current consumption from load voltage L+ (without load)	-
Voltage output short-circuit protection	-
Voltage outputs	-
Min. load resistance (voltage range)	-
Max. capacitive load (current range)	-
Max. inductive load (current range)	-
Output voltage ranges	-
Operational limit of voltage ranges	-
Basic error limit voltage ranges with SFU	-
Destruction limit against external applied voltage	-
Current outputs	-
Max. in load resistance (current range)	-
Max. inductive load (current range)	-
Typ. open circuit voltage current output	-
Output current ranges	-
Operational limit of current ranges	-
Radical error limit current ranges with SFU	-
Destruction limit against external applied voltage	-
Settling time for ohmic load	-
Settling time for capacitive load	-
Settling time for inductive load	-
Resolution in bit	-
Conversion time	-
Substitute value can be applied	-
Output data size	-
Technical data counters	
Number of counters	4
Counter width	32 Bit
Maximum input frequency	100 kHz
Maximum count frequency	400 kHz
Mode incremental encoder	✓
Mode pulse / direction	✓
Mode pulse	✓
Mode frequency counter	✓
Mode period measurement	✓
Gate input available	✓

Order no.	M13-CCF0000
Latch input available	✓
Reset input available	-
Counter output available	✓
Load and working memory	
Load memory, integrated	128 KB
Load memory, maximum	128 KB
Work memory, integrated	64 KB
Work memory, maximal	128 KB
Memory divided in 50% program / 50% data	✓
Memory card slot	SD/MMC-Card with max. 2 GB
Hardware configuration	
Racks, max.	1
Modules per rack, max.	8
Number of integrated DP master	-
Number of DP master via CP	-
Operable function modules	-
Operable communication modules PtP	-
Operable communication modules LAN	-
Status information, alarms, diagnostics	
Status display	yes
Interrupts	yes
Process alarm	yes
Diagnostic interrupt	yes
Diagnostic functions	yes, parameterizable
Diagnostics information read-out	possible
Supply voltage display	green LED
Group error display	red LED
Channel error display	red LED per group
Isolation	
Between channels	✓
Between channels of groups to	16
Between channels and backplane bus	✓
Between channels and power supply	-
Max. potential difference between circuits	DC 75 V/ AC 50 V
Max. potential difference between inputs (Ucm)	-

Order no.	M13-CCF0000
Max. potential difference between Mana and Mintern (Uiso)	-
Max. potential difference between inputs and Mana (Ucm)	-
Max. potential difference between inputs and Mintern (Uiso)	-
Max. potential difference between Mintern and outputs	-
Insulation tested with	DC 500 V
Command processing times	
Bit instructions, min.	0.02 µs
Word instruction, min.	0.02 µs
Double integer arithmetic, min.	0.02 µs
Floating-point arithmetic, min.	0.12 µs
Timers/Counters and their retentive characteristics	
Number of S7 counters	512
S7 counter remanence	adjustable 0 up to 256
S7 counter remanence adjustable	C0 .. C7
Number of S7 times	512
S7 times remanence	adjustable 0 up to 256
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 256
Bit memories retentive characteristic preset	MB0 .. MB15
Number of data blocks	1024
Max. data blocks size	64 KB
Max. local data size per execution level	4096 Byte
Blocks	
Number of OBs	22
Number of FBs	1024
Number of FCs	1024
Maximum nesting depth per priority class	16
Maximum nesting depth additional within an error OB	4
Time	
Real-time clock buffered	✓
Clock buffered period (min.)	30 d
Accuracy (max. deviation per day)	10 s
Number of operating hours counter	8

Order no.	M13-CCF0000
Clock synchronization	✓
Synchronization via MPI	Master/Slave
Synchronization via Ethernet (NTP)	no
Address areas (I/O)	
Input I/O address area	2048 Byte
Output I/O address area	2048 Byte
Input process image maximal	2048 Byte
Output process image maximal	2048 Byte
Digital inputs	144
Digital outputs	140
Digital inputs central	144
Digital outputs central	140
Integrated digital inputs	16
Integrated digital outputs	12
Analog inputs	2
Analog outputs	0
Analog inputs, central	2
Analog outputs, central	0
Integrated analog inputs	2
Integrated analog outputs	0
Communication functions	
PG/OP channel	✓
Global data communication	✓
Number of GD circuits, max.	8
Size of GD packets, max.	54 Byte
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
PWM data	
PWM channels	2
PWM time basis	1 µs / 0.1 ms / 1 ms
Period length	-

Order no.	M13-CCF0000
Minimum pulse width	0...0.5 * Period duration
Type of output	Highside
Functionality Sub-D interfaces	
Type	X1
Type of interface	RS422/485 isolated
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	-
Type	X2
Type of interface	RS485 isolated
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	✓
MP²I (MPI/RS232)	-
DP master	-
DP slave	optional
Point-to-point interface	-
5V DC Power supply	max. 90mA, isolated
24V DC Power supply	-
Functionality MPI	
Number of connections, max.	32
PG/OP channel	✓
Routing	✓
Global data communication	✓
S7 basic communication	✓
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Transmission speed, min.	19.2 kbit/s

Order no.	M13-CCF0000
Transmission speed, max.	12 Mbit/s
Functionality PROFIBUS slave	
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	✓
Transfer memory inputs, max.	244 Byte
Transfer memory outputs, max.	244 Byte
Address areas, max.	32
User data per address area, max.	32 Byte
Point-to-point communication	
PtP communication	✓
Interface isolated	✓
RS232 interface	-
RS422 interface	✓
RS485 interface	✓
Connector	Sub-D, 9-pin, female
Transmission speed, min.	150 bit/s
Transmission speed, max.	115.2 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	-
Functionality RJ45 interfaces	

Order no.	M13-CCF0000
Type	X3/X4
Type of interface	Ethernet 10/100 MBit Switch
Connector	2 x RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	4
Productive connections	✓
Fieldbus	-
Type	-
Type of interface	-
Connector	-
Electrically isolated	-
PG/OP channel	-
Number of connections, max.	-
Productive connections	-
Fieldbus	-
Ethernet communication via PG/OP	
Number of productive connections via PG/OP, max.	2
Number of productive connections by Siemens NetPro, max.	2
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.	2
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

Order no.	M13-CCF0000
User data per UDP connection, max.	1472 Byte
Housing	
Material	PPE / PPE GF10
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	72 mm x 88 mm x 71 mm
Net weight	221 g
Weight including accessories	221 g
Gross weight	240 g
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	in preparation
KC certification	in preparation

3.4.2 Technical data EM M09

Order no.	M09-0CB00
Type	EM M09
Module ID	-
Housing	
Material	PPE / PPE GF10
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	35 mm x 88 mm x 26 mm
Net weight	54 g
Weight including accessories	54 g
Gross weight	64 g
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL certification	in preparation
KC certification	in preparation

4 Deployment CPU M13-CCF0000

4.1 Assembly

Information about assembly and cabling ↳ Chapter 2 'Basics and mounting' on page 10.

4.2 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 *.pkg file (firmware file) on the memory card. If so, this is shown by the status bar of the CPU and the firmware may be installed by an update request. ↳ *Chapter 4.13 'Firmware update' on page 76*
- The CPU checks if a previously activated VSC is inserted. If not, this is shown by the status bar of the CPU and a diagnostics entry is released. The CPU switches to STOP after 72 hours. With a just installed VSC activated functionalities remain activated. ↳ *Chapter 4.19 'Diagnostic entries' on page 85*

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.

4.3 Addressing

4.3.1 Overview

To provide specific addressing of the installed peripheral 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 peripheral addresses for digital in-/output modules starting with 0 and analog modules are assigned to even addresses starting with 256.

4.3.2 Default address assignment of the I/O part

Sub module	Input address	Access	Assignment
AI5/AO2	800	WORD	Analog input channel 0 (X6)
	802	WORD	Analog input channel 1 (X6)

Sub module	Input address	Access	Assignment
DI24/DO16	136	BYTE	Digital input I+0.0 ... I+0.7 (X1)
	137	BYTE	Digital input I+1.0 ... I+1.7 (X5)

Sub module	Input address	Access	Assignment
Counter	816	DINT	Channel 0: Counter value / Frequency value
	820	DINT	Channel 1: Counter value / Frequency value
	824	DINT	Channel 2: Counter value / Frequency value
	828	DINT	Channel 3: Counter value / Frequency value

Sub module	Output address	Access	Assignment
DI24/DO16	136	BYTE	Digital output Q+0.0 ... Q+0.7 (X2)
	137	BYTE	Digital output Q+1.0 ... Q+1.3 (X6)

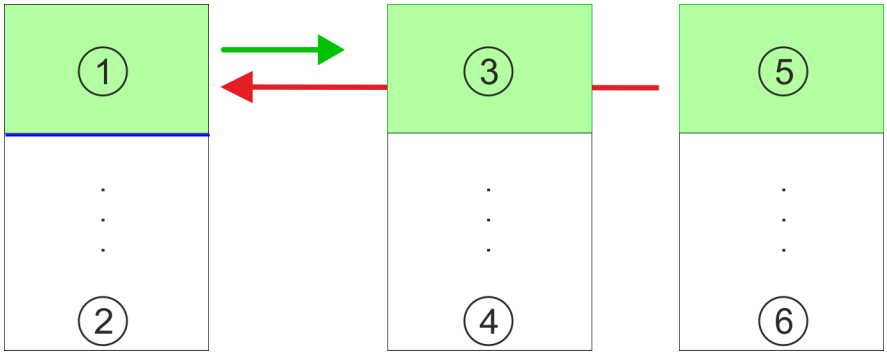
Sub module	Output address	Access	Assignment
Counter	816	DWORD	reserved
	820	DWORD	reserved
	824	DWORD	reserved
	828	DWORD	reserved

4.3.3 Option: Addressing periphery modules

The CPU M13-CCF0000 provides an I/O area (address 0 ... 2047) and a process image of the in- and outputs (each address default 0 ... 127). The process image stores the signal states of the lower address (default 0 ... 127) in an additional memory area. The size of the process image can be preset via the parameterization. ↳ *Chapter 4.7 'Setting standard CPU parameters' on page 63*

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 I/O area: 0 ... 127 (default)
2 I/O area max.: 2047
3 Process image of the inputs (PII): 0 ... 127
4 Process image of the inputs (PII) max.: 2047
5 Process image of the outputs (PIQ): 0 ... 127
6 Process image of the outputs (PIQ) max.: 2047

Max. number of pluggable modules

Up to 8 periphery modules can be connected to the CPU.

Define addresses by hardware configuration

You may access the modules with read res. write accesses to the peripheral 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.
- As soon as the mapping of digital modules exceeds the address 256, by regarding the order, these are mapped starting from address 256.

4.4 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 configuration of the System MICRO CPU happens in the Siemens SIMATIC Manager by means of a virtual PROFINET IO device *'VIPA MICRO PLC'*. The *'VIPA MICRO PLC'* 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!

Installing the IO device VIPA MICRO PLC

The installation of the PROFINET IO devices *'VIPA MICRO PLC'* happens in the hardware catalog with the following approach:

1. Go to the service area of www.vipa.com.
2. Load from the download area at *'PROFINET files'* the file Micro_Vxxx.zip.
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 ➔ VIPA MICRO PLC'*

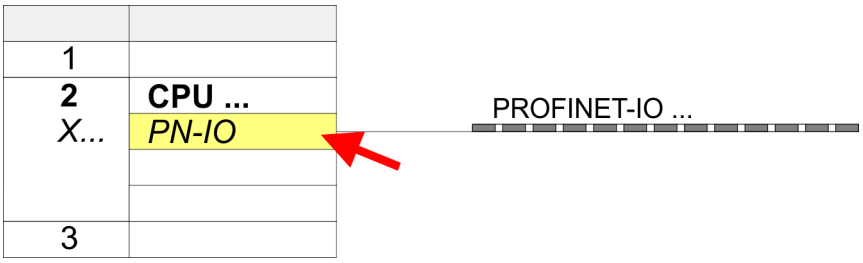
Proceeding

In the Siemens SIMATIC Manager the following steps should be executed:

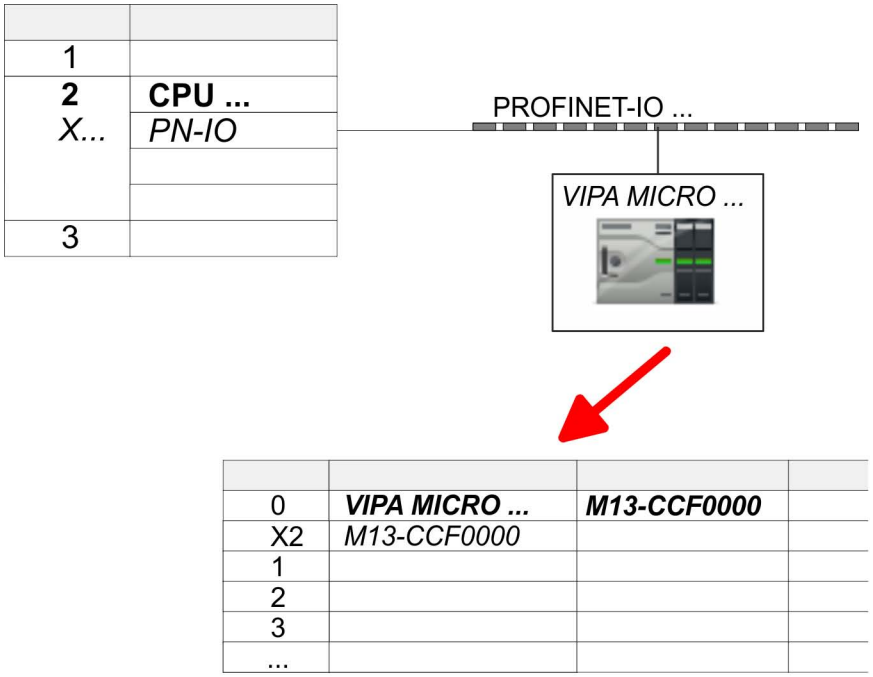
1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at *'Slot'*-Number 2 the CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).

Slot	Module
1	
2	**CPU 314C-2PN/DP**
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.



0	***VIPA MICRO ...***	***M13-CCF0000***	
X2	*M13-CCF0000*		
1			
2			
3			
...			

9. Navigate in the hardware catalog to the directory *'PROFINET IO ➔ Additional field devices ➔ I/O ➔ VIPA MICRO PLC'* and connect the IO device M13-CCF0000 to your PROFINET system.

⇨ In the slot overview of the PROFINET IO device *'VIPA MICRO PLC'* the CPU is already placed at slot 0.

4.5 Hardware configuration - System MICRO modules

System SLIO backplane bus

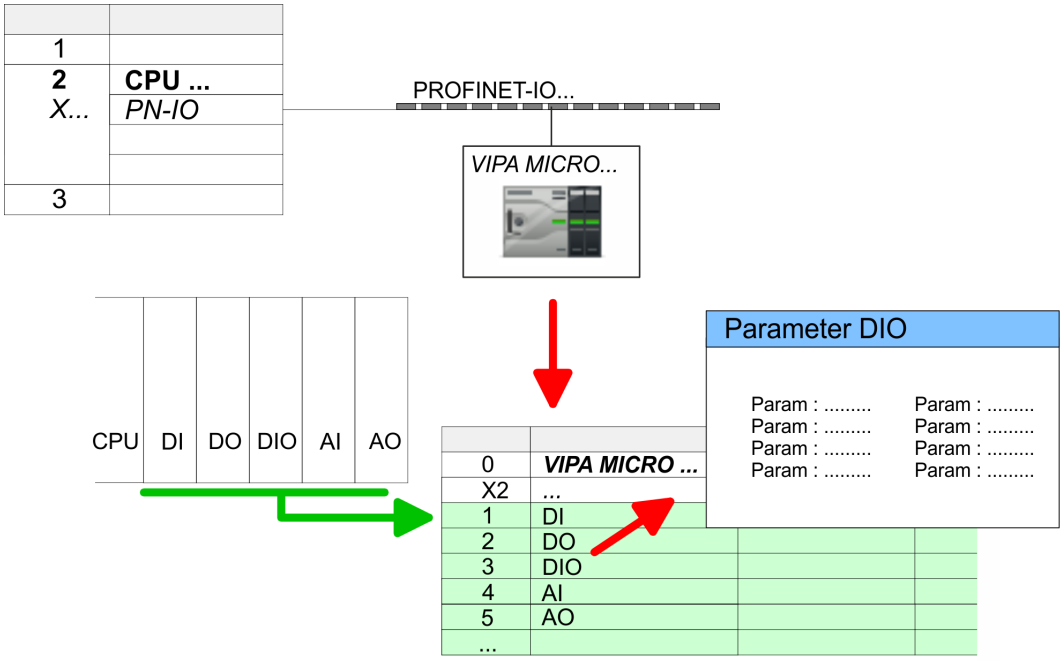
To connect System SLIO modules, the CPU has a backplane bus, which is supplied by the CPU. Here up to 8 System MICRO modules can be connected.

Proceeding

1. Perform, if not already done, a hardware configuration for the CPU. ↬ *Chapter 4.4 'Hardware configuration - CPU' on page 58*

2. Starting with slot 1 place in the slot overview of the PROFINET IO device *'VIPA MICRO PLC'* your System MICRO modules in the plugged sequence.

3. Parametrize if necessary the modules and assign valid addresses, so that they can directly be addressed.



4.6 Hardware configuration - Ethernet PG/OP channel

Overview

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 (X3/X4) is designed as switch. This enables PG/OP communication via the connections X3 and X4.
- The Ethernet PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.
- 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. This is called "initialization".
- This can be done with the Siemens SIMATIC Manager.

Assembly and commissioning

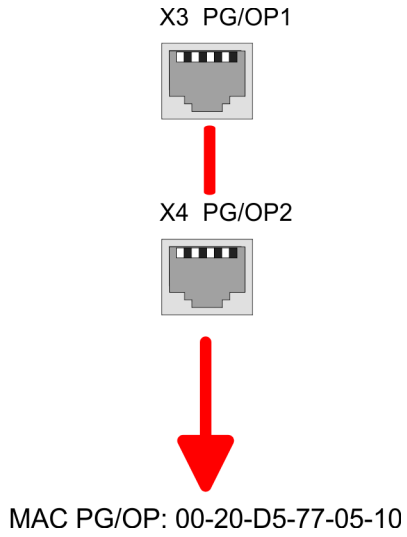
1. Install your System with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the one of the Ethernet jack (X3, X4) of the Ethernet PG/OP channel to Ethernet, to which your programming device (PC) is connected.
4. Switch on the power supply.

   ⇨ After a short boot time the CPU is ready for communication. It 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 5.3 & SP3 with the following proceeding:

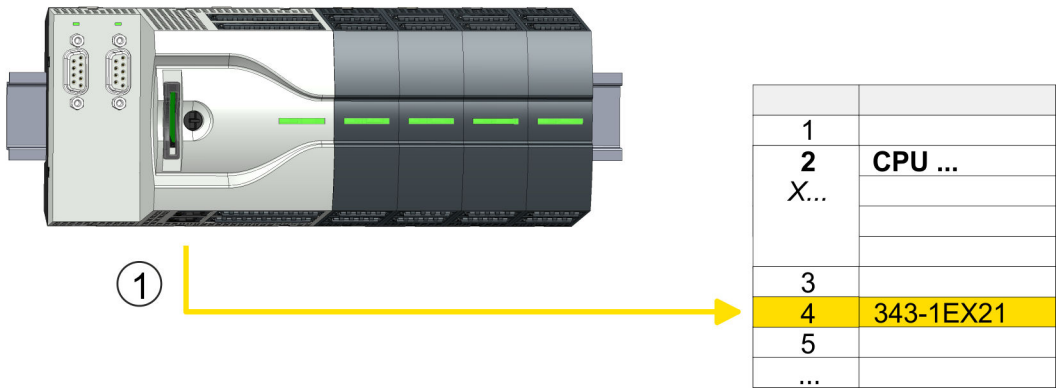
1. Start the Siemens SIMATIC Manager and set via *'Options ➔ Set PG/PC interface'* the access path to *'TCP/IP -> Network card ....'*.
2. Open with *'PLC ➔ Edit Ethernet Node n'* the dialog window with the same name.
3. To get the stations and their MAC address, use the [Browse] button or type in the 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 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
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 343-1EX21 0XE0 V1.2).
3. Open the property window via double-click on the CP 343-1EX21 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.



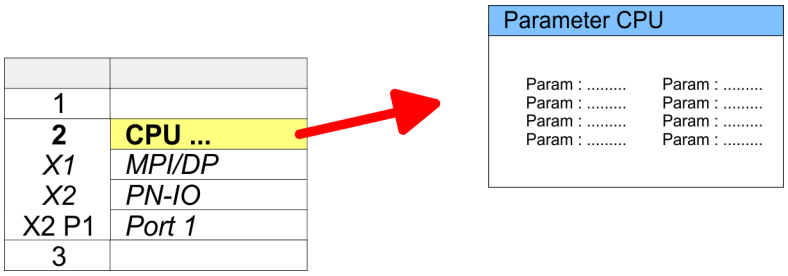
1 Ethernet PG/OP channel

4.7 Setting standard CPU parameters

4.7.1 Parametrization via Siemens CPU

Parametrization via Siemens CPU 314-6EH04

Since the CPU from VIPA is to be configured as Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) in the Siemens hardware configurator, the standard parameters of the VIPA CPU may be set with "Object properties" of the CPU 314C-2 PN/DP during hardware configuration. Via a double-click on the CPU 314C-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.



4.7.2 Parameter CPU

Supported parameters

The CPU does not evaluate each parameter, which may be set at the hardware configuration. The parameters of the following registers are not supported: Synchronous cycle interrupts, communication and web. The following parameters are currently supported:

General

- Short description
  - The short description of the Siemens CPU is CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
- Order No. / Firmware
  - Order number and firmware are identical to the details in the "hardware catalog" window.
- Name
  - The Name field provides the short description of the CPU.
  - If you change the name the new name appears in the Siemens SIMATIC Manager.

- 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 81346-1.
- Location designation
  - The location designation is part of the resource designation.
  - Here the exact location of your module within a plant may be specified.
- Comment
  - In this field information about the module may be entered.

Startup

- Startup when expected/actual configuration differs
  - If the checkbox for *'Startup when expected/actual configuration differ'* is deselected and at least one module is not located at its configured slot or if another type of module is inserted there instead, then the CPU does not switch to RUN mode and remains in STOP mode.
  - If the checkbox for *'Startup when expected/actual configuration differ'* is selected, then the CPU starts even if there are modules not located in their configured slots of if another type of module is inserted there instead, such as during an initial system start-up.
- Monitoring time for ready message by modules [100ms]
  - This operation specifies the maximum time for the ready message of every configured module after PowerON.
  - Here connected PROFIBUS DP slaves are also considered until they are parameterized.
  - If the modules do not send a ready message to the CPU by the time the monitoring time has expired, the actual configuration becomes unequal to the preset configuration.
- Monitoring time for transfer of parameters to modules [100ms]
  - The maximum time for the transfer of parameters to parameterizable modules.
  - Here connected PROFINET IO devices also considered until they are parameterized.
  - If not every module has been assigned parameters by the time this monitoring time has expired; the actual configuration becomes unequal to the preset configuration.

Cycle/Clock memory

- Update OB1 process image cyclically
  - This parameter is not relevant.
- Scan cycle monitoring time
  - Here the scan cycle monitoring time in milliseconds may be set.
  - If the scan cycle time exceeds the scan cycle monitoring time, the CPU enters the STOP mode.
  - Possible reasons for exceeding the time are:
    - Communication processes
    - a series of interrupt events
    - an error in the CPU program
- Minimum scan cycle time
  - This parameter is not relevant.
- Scan cycle load from Communication
  - Using this parameter you can control the duration of communication processes, which always extend the scan cycle time so it does not exceed a specified length.
  - If the cycle load from communication is set to 50%, the scan cycle time of OB 1 can be doubled. At the same time, the scan cycle time of OB 1 is still being influenced by asynchronous events (e.g. hardware interrupts) as well.

- Size of the process image input/output area
  - Here the size of the process image max. 2048 for the input/output periphery may be fixed (default: 256).
- OB85 call up at I/O access error
  - The preset reaction of the CPU may be changed to an I/O access error that occurs during the update of the process image by the system.
  - The VIPA CPU is preset such that OB 85 is not called if an I/O access error occurs and no entry is made in the diagnostic buffer either.
- Clock memory
  - Activate the check box if you want to use clock memory and enter the number of the memory byte.

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

Retentive Memory

- Number of Memory bytes from MB0
  - Enter the number of retentive memory bytes from memory byte 0 onwards.
- Number of S7 Timers from T0
  - Enter the number of retentive S7 timers from T0 onwards. Each S7 timer occupies 2bytes.
- Number of S7 Counters from C0
  - Enter the number of retentive S7 counter from C0 onwards.
- Areas
  - This parameter is not supported.

Interrupts

- Priority
  - Here the priorities are displayed, according to which the hardware interrupt OBs are processed (hardware interrupt, time-delay interrupt, async. error interrupts).

Time-of-day interrupts

- Priority
  - This value is fixed to 2.
- Active
  - By enabling *'Active'* the time-of-day interrupt function is enabled.
- Execution
  - Select how often the interrupts are to be triggered.
  - Intervals ranging from every minute to yearly are available. The intervals apply to the settings made for *start date* and *time*.
- Start date/time
  - Enter date and time of the first execution of the time-of-day interrupt.
- Process image partition
  - This parameter is not supported.

Cyclic interrupts

- Priority
  - Here the priorities may be specified according to which the corresponding cyclic interrupt is processed.
- Execution
  - Enter the time intervals in ms, in which the watchdog interrupt OBs should be processed.
  - The start time for the clock is when the operating mode switch is moved from STOP to RUN.

- Phase offset
  - Enter the delay time in ms for current execution for the watch dog interrupt. This should be performed if several watchdog interrupts are enabled.
  - Phase offset allows to distribute processing time for watchdog interrupts across the cycle.
- Process image partition
  - This parameter is not supported.

Diagnostics/Clock

- Report cause of STOP
  - Activate this parameter, if the CPU should report the cause of STOP to PG respectively OP on transition to STOP.
- Number of messages in the diagnostics buffer
  - This parameter is ignored. The CPU always has a diagnostics buffer (circular buffer) for 100 diagnostics messages.
- Synchronization type
  - Here you specify whether clock should synchronize 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 is to be carried out.
- Correction factor
  - Lose or gain in the clock time may be compensated within a 24 hour period by means of the correction factor in ms.
  - If the clock is 1s slow after 24 hours, you have to specify a correction factor of "+1000" ms.

Protection

- Level of protection
  - Here 1 of 3 protection levels may be set to protect the CPU from unauthorized access.
  - *Protection level 1 (default setting):*
    No password adjustable, no restrictions
  - *Protection level 2 with password:*
    Authorized users: read and write access
    Unauthorized user: read access only
  - *Protection level 3:*
    Authorized users: read and write access
    Unauthorized user: no read and write access

4.8 Setting VIPA specific CPU parameters

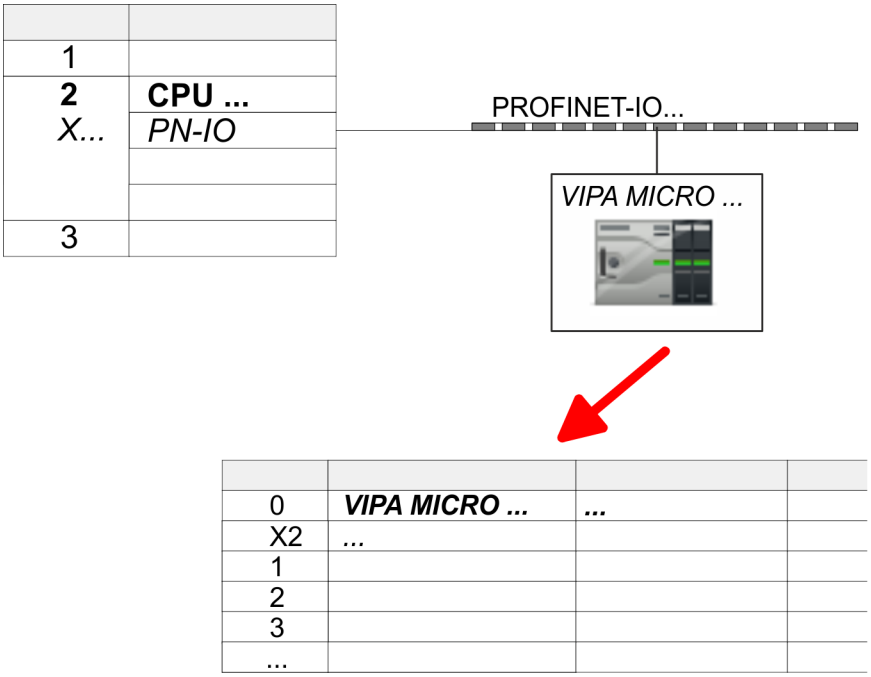
Overview

Except of the VIPA specific CPU parameters the CPU parametrization takes place in the parameter dialog of the CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) from Siemens. After the hardware configuration of the CPU you can set the parameters of the CPU in the virtual IO device *'VIPA MICRO PLC'*. Via double-click at *'VIPA MICRO PLC M13-CCF0000'* the properties dialog is opened.

Here the following parameters may be accessed:

- Additional retentive memory
- Additional retentive timer
- Additional retentive counter
- Diagnostics interrupt DI power section supply

- Diagnostics interrupt DO power section supply
- Diagnostics interrupt DO short circuit/overload



VIPA specific parameters

The following parameters may be accessed by means of the properties dialog of the VIPA CPU.

- Additional retentive memory
  - Here enter the number of retentive memory bytes. With 0 the value *'Retentive memory ➔ Number of memory bytes starting with MB0'* is set, which is pre-set at the parameters of the Siemens CPU.
  - Range of values: 0 (default) ... 8192
- Additional retentive timer
  - Enter the number of S7 timers. With 0 the value *'Retentive memory ➔ Number S7 timers starting with T0'* is set, which is pre-set at the parameters of the Siemens CPU.
  - Range of values: 0 (default) ... 512
- Additional retentive counter
  - Enter the number of S7 counter. With 0 the value *'Retentive memory ➔ Number S7 counters starting with C0'* is set, which is pre-set at the parameters of the Siemens CPU.
  - Range of values: 0 (default) ... 512
- Diagnostics interrupt (default: deactivated)
  - Diagnostics interrupt DI power section supply
    Error: 3L+ (DC 24V DI power section supply) missing respectively <19V
  - Diagnostics interrupt DO power section supply
    Error: 4L+ (DC 24V DO power section supply) missing respectively <19V
  - Diagnostics interrupt DO short circuit/overload
    Error: Short circuit or overload of an digital output respectively current exceeds 0.5A.

4.9 Project transfer

Overview

There is the following possibility for project transfer into the CPU:

- Transfer via Ethernet
- Transfer via memory card
- Option: Transfer via MPI ↳ *Chapter 4.9.3 'Option: Transfer via MPI' on page 69*

4.9.1 Transfer via Ethernet

Initialization

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

- X3/X4: Ethernet PG/OP channel
  - ↳ *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*

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.

4.9.2 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 Siemens SIMATIC Manager with your project
2. Create with *'File ➔ Memory Card File ➔ New'* a new wld file.
3. Copy the blocks from the project blocks folder and the *System data* into the wld file.

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 flickering of the yellow LED of the status bar 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 red LED of the status bar lights up.

4.9.3 Option: Transfer via MPI

General

For the transfer via MPI the use of the optionally available extension module EM M09 is required. The extension module provides the interface X2: MPI(PB) with fixed pin assignment. ↳ *Chapter 2.4 'Mounting' on page 13*

Net structure

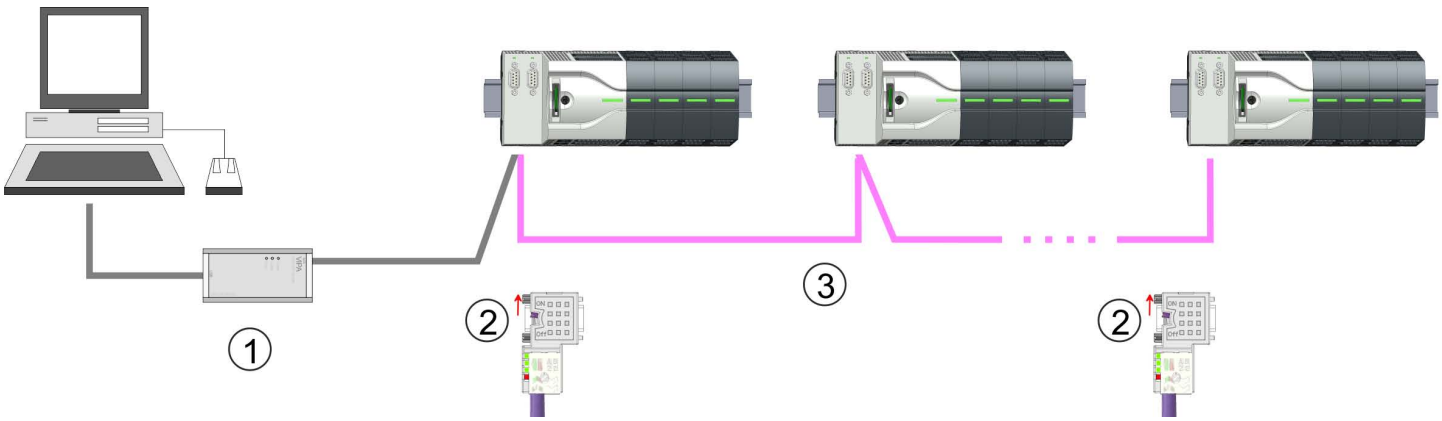
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs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

MPI programming cable

The MPI programming cables are available at VIPA 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

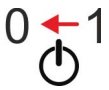
A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment. Please make sure that the participants with the activated terminating resistors are always power supplied. Otherwise it may cause interferences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



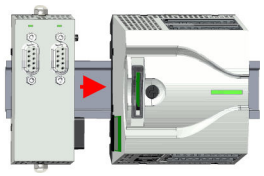
1 MPI programming cable
2 Activate the terminating resistor via switch
3 MPI network

Proceeding enabling the interface

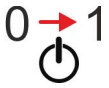
A hardware configuration to enable the MPI interface is not necessary. By installing the extension module EM M09 the MPI interface is enabled.



1. Turn off the power supply.



2. Mount the extension module. ↳ *Chapter 2.4 'Mounting' on page 13*



3. Switch on the power supply.
   ⇨ After a short boot time the interface X2 MPI(PB) is ready for MPI communication with the MPI address 2.

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 from VIPA.
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.

4.10 Accessing the web server

4.10.1 Access via the Ethernet PG/OP channel

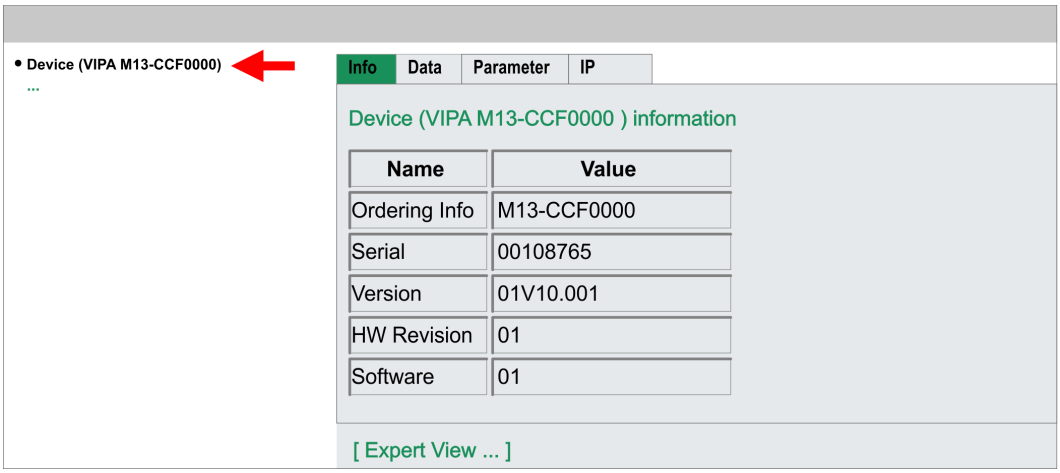
There is a web server, which can be accessed via the IP address of the Ethernet PG/OP channel with an Internet browser. At the web page information about the CPU and its connected modules can be found. ↳ *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*

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

4.10.2 Structure of the web page

The web page is built dynamically. The web page only shows information. The shown values cannot be changed

4.10.3 Web page with selected CPU



Info - Overview

Here order number, serial number and the version of firmware and hardware of the CPU are listed. [Expert View] takes you to the advanced "Expert View".

Info - Expert View

Runtime Information		
Operation Mode	RUN	CPU: Status information
Mode Switch	RUNP	
System Time	03.11.15 14:32:49:561	CPU: Date, time
OB1-Cycle Time	cur = 2000us, min = 2000us, max = 5000us, avg = 2335us	CPU: Cyclic time: min = minimum cur = current max = maximum avg = average
Interface Information		
X1/X5	DI 16	Address 136..137
	AI 2	Address 800..803
	Counter	Address 816..831
X2/X6	DO 12	Address 136..137
	Counter	Address 816..831
X3	PG/OP Ethernet Port 1	Address 256..271
X4	PG/OP Ethernet Port 2	Address 256..271
Serial X1	PTP	
Serial X2	DPS	Address 2047
VIPASetCard Info		

VSD...		Activated VSD respectively VSC with Information for the support
...		
VSC...		
...		
VSC-Trial-Time	71:59	Remaining time in hh:mm for deactivation of the expansion memory respectively bus functionality and the CPU goes to STOP (abnormal operation), if the VSC is removed. This parameter is only visible when the VSC of an enabled function is removed.
Memory Extension	0 bytes	Size of the additional memory, which was activated by means of a VSC.
PROFIBUS	not activated	Type of the PROFIBUS functionality, which was activated by means of a VSC.
Memory Usage		CPU: Information to memory configuration
LoadMem	9208 / 131072 bytes	Load memory, working memory (code/data)
WorkMemCode	304 / 32768 bytes	
WorkMemData	0 / 32768 bytes	
PG/OP Network Information		
Device Name	VIPA M13-CCF0000	Ethernet PG/OP channel:
IP Address	192.168.1.211	Address information
Subnet Mask	255.255.255.0	
Gateway Address	192.168.1.211	
MAC Address	00:20:D5:02:74:48	
Link Mode X3	100 Mbps - Full Duplex	Link mode of the interfaces
Link Mode X4	Not Available	
CPU Firmware Information		
File System	V1.0.2	CPU: Information for the support
PRODUCT	VIPA M13-CCF0000 V2.0.12 Px000292.pkg	CPU: Name, firmware version, package
HARDWARE	V0.1.0.0 5852A-V11 Mx000313.102	CPU: Information for the support
Bx000501	V2.0.12.0	
Ax000136	V1.0.6.0	
fx000018.wld	V1.0.2.0	

syslibex.wld	n/a	
Protect.wld	n/a	
ARM Processor Load		
Measurement Cycle Time	100 ms	
Last Value	79%	
Maximum Load	100%	

Data Currently nothing is displayed here.

Parameter Currently nothing is displayed here.

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

4.11 Operating modes

4.11.1 Overview

The CPU can be in one of 3 operating modes:

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

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.
-  : The yellow LED of the status bar lights up in the STOP state.

Operating mode START-UP

-  : After PowerON the yellow LED of the status bar blinks in the STOP state.
-  : After a short time the flashing changes to a steady light.
- 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.
  -  : The green LEDs 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.
  -  : The green LEDs of the status bar lights up when the START-UP is completed and the CPU is in the RUN state.

**** 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 green LED lights up when the CPU is in the RUN state.
- 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.

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 VIPA 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 lock) 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.

Event	concerns	Effect
	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

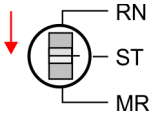
During the overall reset the entire user memory is erased. Data located in the memory card is not affected. You have 2 options to initiate an overall reset:

- Overall reset by means of the operating mode switch
- Overall reset by means of the Siemens SIMATIC Manager

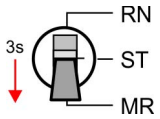
You should always establish an overall reset to your CPU before loading an application program into your CPU to ensure that all blocks have been cleared from the CPU.

4.12.1 Overall reset by means of the operating mode switch

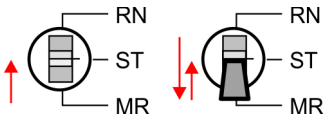
Proceeding



1. Your CPU must be in STOP mode. For this switch the operating mode switch of the CPU to STOP.
   ⇨ Status bar: 



2. Switch the operating mode switch to MR position for about 3 seconds.
   ⇨ The yellow LED blinks with 1Hz  and changes from repeated 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 yellow LED blinks with 2Hz  .
4. The overall reset has been completed when the yellow LED is on permanently  .

4.12.2 Overall reset by means of the Siemens SIMATIC Manager

Proceeding

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. You may place the CPU in STOP by the menu command *'PLC ➔ Operating mode'*.
2. You may request the overall reset by means of the menu command *'PLC ➔ Clean/Reset'*.

   ⇨ 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 the yellow LED of the status bar blinks with 2Hz  . The overall reset has been completed when the yellow LED is on permanently  .

4.12.3 Actions after the overall reset

Activating functionalities by means of a VSC

If there is a VSC memory card from VIPA plugged, after an overall reset the according functionalities are automatically activated. ↳ *Chapter 4.15 'Deployment storage media - VSD, VSC' on page 79*

Automatic reload

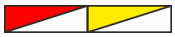
If there is a project S7PROG.WLD on the memory card, after an overall reset the CPU attempts to reload this project from the memory card. Here the yellow LED of the status line flickers  . The operating mode of the CPU will be STOP respectively RUN, depending on the position of the operating mode switch.

Factory reset

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!
↳ *Chapter 4.14 'Reset to factory settings' on page 78*

4.13 Firmware update

Overview

There is the opportunity to execute a firmware update for the CPU and its components via memory card. For this 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 pkg file name is reserved for each update-able component and hardware release, which begins with "px" and differs in a number with 6 digits. At the VIPA System MICRO CPU the pkg 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 *.pkg file at the memory card. If this firmware version is different to the existing firmware version, this is indicated by alternately blinking (1Hz) of the red and yellow LED  and the firmware may be installed by an update request.

Current firmware at www.vipa.com

The latest firmware versions can be found in the service area at www.vipa.com. For example the following files are necessary for the firmware update of the CPU M13-CCF0000 and its components with hardware release 01:

- CPU M13C, Hardware release 01: Px000292.pkg

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.

Display the firmware version via web page

The CPU has an integrated web page that monitors information about the firmware version of the SPEED7 components. The Ethernet PG/OP channel provides the access to this web page. To activate the PG/OP channel you have to enter according IP parameters. This happens in the Siemens SIMATIC Manager either by a hardware configuration, loaded by memory card respectively Ethernet by means of the MAC address with *'PLC ➔ Assign Ethernet Address'*. After that you may access the PG/OP channel with a web browser via the set IP address. *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*

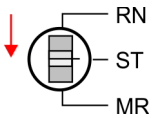
Load firmware and transfer it to memory card

1. Go to www.vipa.com
2. Click at *'Service Support ➔ Downloads ➔ Firmware'*.
3. Via *'System MICRO ➔ CPU'* navigate to your CPU and download the zip file to your PC.
4. Unzip the zip file and copy the pgk file to the root directory of your memory card.

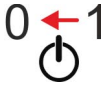
CAUTION!
With a firmware update an overall reset is automatically executed. If your program is only available in the load memory of the CPU it is deleted! Save your program before executing a firmware update! After a firmware update you should execute a "Reset to factory setting". *Chapter 4.14 'Reset to factory settings' on page 78*

Transfer firmware from memory card into CPU

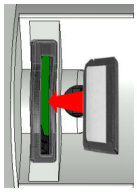
Please note that with some firmware versions an additional firmware update via alternate blinking (1Hz) of the red and yellow LED can be indicated even when the operating mode switch is in RUN position. In this state the CPU can only restart, if you establish a further firmware update process. For this tap the operating mode switch shortly downwards to MR and follow the procedures described below.



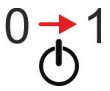
1. Switch the operating mode switch of your CPU in position STOP.



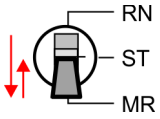
2. Turn off the power supply.



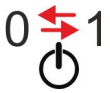
3. Plug the memory card with the firmware files into the CPU. Please take care of the correct plug-in direction of the memory card.



4. Switch on the power supply.
   ⇨ After a short boot-up time, the alternate blinking of the red and yellow LED of the status bar shows that at least 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.
6. During the update process, the yellow LED of the status bar flashes or flickers . This may last several minutes.
7. The update is completed without errors when the red and yellow LEDs of the status bar are flashing (1Hz) . If only the red LED of the status bar is flashing, an error has occurred.

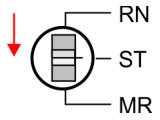


8. Turn power OFF and ON.
9. Now execute a *Reset to factory setting*. After that the CPU is ready for duty.
   ↳ *Chapter 4.14 'Reset to factory settings' on page 78*

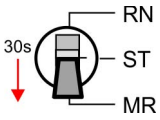
4.14 Reset to factory settings

Proceeding

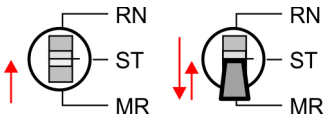
- With the following proceeding the internal RAM of the CPU is completely deleted and the CPU is reset to delivery state.
- Please regard that the MPI address is also reset to default 2 and the IP address of the Ethernet PG/OP channel is reset to 0.0.0.0!
- A *factory reset* may also be executed by the command FACTORY_RESET.
  ↳ *Chapter 4.17 'CMD - auto commands' on page 82*



1. Switch the CPU to STOP.



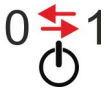
2. Push the operating mode switch down to position MR for 30 seconds. Here the yellow LED of the status bar blinks . After a few seconds the LED changes to static light. Now the LED changes between static light and blinking. Start here to count the static light of the LED.



3. After the 6. static light release the operating mode switch and tip it downwards to MR.
   ⇨ To confirm the reset process the yellow LED of the status bar blinks (2Hz) . This means that the RAM was deleted completely.

> *If the yellow LED of the status bar 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 yellow LED has static light for exact 6 times.*

4. The reset process is completed when the red and yellow LEDs of the status bar are blinking (1Hz) .



5. Turn power OFF and ON.

After a firmware update of the CPU you always should execute a factory reset.

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 - **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.
  - To activate the corresponding card is to be installed and a *Overall reset* is to be established. *Chapter 4.12 'Overall reset' on page 75*

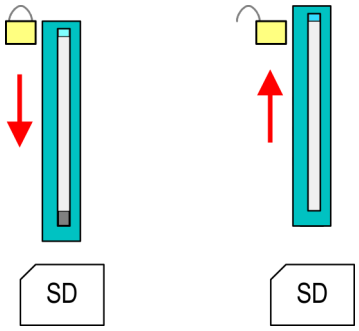
A list of the currently available VSD respectively VSC can be found at www.vipa.com

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. 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.
↳ *Chapter 4.10 'Accessing the web server' on page 70*

CAUTION!
Please regard that the VSC must remain plugged when you've enabled optional functions at your CPU. Otherwise the red LED of the status line turns on and the CPU switches to STOP after 72 hours. As soon as an activated VSC is not plugged, the LED is on and the "Trial-Time" counts downwards from 72 hours to 0. After 72 hours the CPU switches to STOP state. By plugging the VSC, the 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 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 *.pkg file (firmware file). If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. *Chapter 4.13 'Firmware update' on page 76*

Once in STOP state

- If a memory card is plugged, 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 can be found in the manual operation list (HB00_OPL_SP7) of your CPU.

4.16 Extended know-how protection

Overview

Besides the "standard" Know-how protection the CPUs from VIPA 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 developed by VIPA 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 the Siemens SIMATIC Manager 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*. ↳ *Chapter 4.12 'Overall reset' on page 75*
   ⇨ 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. A factory reset does not affect the protected blocks. 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. In addition 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

Command	Description	Diagnostics entry
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

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 via the menu item **Debug** of the Siemens SIMATIC Manager.
- The status of the operands and the RLO can be displayed by means of the test function *'Debug ➔ Monitor'*.
- The status of the operands and the RLO can be displayed by means of the test function *'PLC ➔ Monitor/Modify Variables'*.

'Debug ➔ Monitor'

- This test function displays the current status and the RLO of the different operands while the program is being executed.
- It is also possible to enter corrections to the program.
- The processing of the states may be interrupted by means of jump commands or by timer and process-related interrupts.
- At the breakpoint the CPU stops collecting data for the status display and instead of the required data it only provides the PG with data containing the value 0.
- The interruption of the processing of statuses does not change the execution of the program. It only shows that the data displayed is no longer valid.

When using the test function "Monitor" the PLC must be in RUN mode!

For this reason, jumps or time and process alarms can result in the value displayed during program execution remaining at 0 for the items below:

- the result of the logical operation RLO
- Status / AKKU 1
- AKKU 2
- Condition byte
- absolute memory address SAZ. In this case SAZ is followed by a "?".

'PLC ➔ Monitor/Modify Variables'

This test function returns the condition of a selected operand (inputs, outputs, flags, data word, counters or timers) at the end of program execution. 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.

- Control of outputs
  - Serves to check the wiring and proper operation of output modules.
  - If the CPU is in RUN mode, so only outputs can be controlled, which are not controlled by the user program. Otherwise values would be instantly overwritten.
  - If the CPU is in STOP - even without user program, so you need to disable the command output lock BASP (*'Enable PO'*). Then you can control the outputs arbitrarily
- Controlling variables
  - The following variables may be modified: I, Q, M, T, C and D.
  - The process image of binary and digital operands is modified independently of the operating mode of the CPU.
  - When the operating mode is RUN the program is executed with the modified process variable. When the program continues they may, however, be modified again without notification.
- Forcing variables
  - You can pre-set individual variables of a user program with fixed values so that they can not be changed or overwritten by the user program of the CPU.
  - By pre-setting of variables with fixed values, you can set certain situations for your user program and thus test the programmed functions.

CAUTION!

- Please consider that controlling of output values represents a potentially dangerous condition.
- Even after a power cycle forced variables remain forced with its value, until the force function is disabled.
- These functions should only be used for test purposes respectively for troubleshooting. More information about the usage of these functions may be found in the manual of your configuration tool.

4.19 Diagnostic entries

Accessing diagnostic data

↳ *Appendix 'System specific event IDs' on page 213*

- You may read the diagnostics buffer of the CPU via the Siemens SIMATIC Manager. Besides of the standard entries in the diagnostics buffer, the VIPA CPUs support some additional specific entries as Event-IDs.
- 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 current content of the diagnostic buffer is stored at the memory card by means of the CMD DIAGBUF. ↳ *Chapter 4.17 'CMD - auto commands' on page 82*
- 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 I/O periphery

5.1 Overview

Project engineering and parametrization

- On this CPU the connectors for digital respectively analog signal and *Technological functions* are combined in a one casing.
- The project engineering happens in the Siemens SIMATIC Manager as Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3). Here the CPU M13-CCF0000 is parameterized via the *'Properties'* dialog of the Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
- For parametrization of the digital I/O periphery and the *technological functions* the corresponding sub modules of the CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) is to be used.
- The controlling of the operating modes of the *technological functions* happens by means of handling blocks of the user program.

I/O periphery

- The integrated I/Os of the CPU may be used for *technological functions* or as standard periphery.
- *Technological functions* and standard periphery may be used simultaneously with appropriate hardware.
- Read access to inputs used by *technological functions* is possible.
- Write access to used outputs is not possible.
- – AI 2xUx12Bit (0 ... 10V)
  – The analog channels of the module are not isolated to the electronic power supply.
  – The analog part has no status indication
  ↳ *Chapter 5.3 'Analog input' on page 87*
- – DI 16xDC 24V
  – Interrupt functions parameterizable
  – Status indication via LEDs
  ↳ *Chapter 5.4 'Digital input' on page 90*
- – DO 12xDC 24V, 0.5A
  – Status indication via LEDs
  ↳ *Chapter 5.5 'Digital output' on page 94*

Technological functions

- ↳ *Chapter 5.6 'Counting' on page 96*
  – 4 channels
  – Count once
  – Count continuously
  – Count Periodically
  – Control by the user program (SFB 47)
- ↳ *Chapter 5.7 'Frequency measurement' on page 117*
  – 4 channels
  – Control by the user program (SFB 48)
- ↳ *Chapter 5.8 'Pulse width modulation - PWM' on page 123*
  – 2 channels
  – Control by the user program (SFB 49)

5.2 Address assignment

Sub module	Input address	Access	Assignment
AI5/AO2	800	WORD	Analog input channel 0 (X6)
	802	WORD	Analog input channel 1 (X6)

Sub module	Input address	Access	Assignment
DI24/DO16	136	BYTE	Digital input I+0.0 ... I+0.7 (X1)
	137	BYTE	Digital input I+1.0 ... I+1.7 (X5)

Sub module	Input address	Access	Assignment
Counter	816	DINT	Channel 0: Counter value / Frequency value
	820	DINT	Channel 1: Counter value / Frequency value
	824	DINT	Channel 2: Counter value / Frequency value
	828	DINT	Channel 3: Counter value / Frequency value

Sub module	Output address	Access	Assignment
Counter	816	DWORD	reserved
	820	DWORD	reserved
	824	DWORD	reserved
	828	DWORD	reserved

Sub module	Output address	Access	Assignment
DI24/DO16	136	BYTE	Digital output Q+0.0 ... Q+0.7 (X2)
	137	BYTE	Digital output Q+1.0 ... Q+1.3 (X6)

5.3 Analog input

5.3.1 Properties

- 2xUx12Bit (0 ... 10V) fixed.
- The analog channels of the module are not isolated to the electronic power supply.
- The analog part has no status indication.

Temporarily not used analog inputs must be connected to the concerning ground.

5.3.2 Analog value representation

Number representation in Siemens S7 format

Resolution	Analog value - twos complement															
	High byte (byte 0)								Low byte (byte 1)							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	SG	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
11Bit+sign	SG	Measuring value											X*	X*	X*	X*

*) The lowest value irrelevant bits of the output value (0) are marked with "X".

Sign bit (SG)

Here it is essential:

- Bit 15 = "0": → positive value
- Bit 15 = "1": → negative value

Behavior at error

As soon as a measured value exceeds the overdrive region respectively falls below the underdrive region, the following value is issued:

- Measuring value > end of overdrive region:
  32767 (7FFFh)
- Measuring value < end of underdrive region:
  -32768 (8000h)

At a parameterization error the value 32767 (7FFFh) is issued.

When leaving the defined range during analog output 0V respectively 0A is issued.

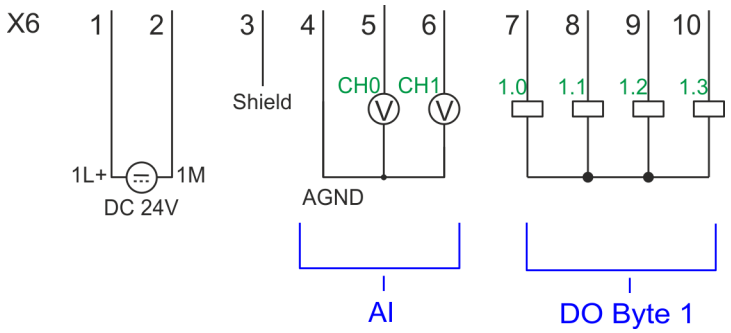
Voltage measurement

0 ... 10V

Measuring range	Voltage (U)	Decimal (D)	Hex	Range	Formulas
0 ... 10V	> 11.759V	32767	7FFFh	overflow	$D = 27648 \cdot \frac{U}{10}$ $U = D \cdot \frac{10}{27648}$ D: decimal value U: voltage value
	11.759V	32511	7EFFh	overdrive range	
	10V	27648	6C00h	nominal range	
	5V	13824	3600h		
	0V	0	0000h		
	-0.8V	-2212	F75Ch	underdrive range	
	< -0.8V	-32768	8000h	underflow	

5.3.3 Wiring

X6: DC 24V, AI, DO byte 1



X6	Function	Type	LED	Description
1	Sys DC 24V	I	green	1L+: DC 24V for electronic section supply
2	Sys 0V	I		1M: GND for electronic section supply
3	Shield	I		Shield
4	AGND	I		GND for analog inputs
5	AI 0	I		Analog input AI 0
6	AI 1	I		Analog input AI 1
7	DO 1.0	O	green	Digital output DO 8
8	DO 1.1	O	green	Digital output DO 9
9	DO 1.2	O	green	Digital output DO 10
10	DO 1.3	O	green	Digital output DO 11

Cables for analog signals

For the analog signals you have to use isolated cables. With this the interferences can be reduced. The shield of the analog cables should be grounded at both ends. If there are potential differences between the cables, a potential compensation current can flow, which could disturb the analog signals. In this case, you should only ground the shield at one end of the cable.

Temporarily not used analog inputs must be connected to the concerning ground.

5.3.4 Parametrization

5.3.4.1 Address assignment

Sub module	Input address	Access	Assignment
AI5/AO2	800	WORD	Analog input channel 0 (X6)
	802	WORD	Analog input channel 1 (X6)

5.3.4.2 Filter

Parameter hardware configuration

The analog input part has a filter integrated. The parametrization of the filter happens in the Siemens SIMATIC Manager via the parameter *'Integration time'*. The default value of the filter is 1000ms. The following values can be entered:

- *'Input 0 ≙ Channel 0'*
  'Input 1 ≙ Channel 1'
  - *'Integration time 2.5ms'* ≙ 2ms (no filter)
  - *'Integration time 16.6ms'* ≙ 100ms (small filter)
  - *'Integration time 20ms'* ≙1000ms (medium filter)

Parametrization during runtime

By using the record set 1 of the SFC 55 "WR_PARM" you may alter the parametrization in the module during runtime.

The time needed until the new parametrization is valid can last up to 2ms. During this time, the measuring value output is 7FFFFh.

Record set 1

Byte	Bit 7 ... Bit 0	Default
0	Bit 7...0: reserved	00h
1	Filter ■ Bit 1, 0: Analog input channel 0 Bit 3, 2: Analog input channel 1 – 00b: *'Integration time 2.5ms'* ≙ 2ms (no filter) – 01b: *'Integration time 16.6ms'* ≙ 100ms (small filter) – 10b: *'Integration time 20ms'* ≙ 1000ms (medium filter) ■ Bit 7...4: reserved	10h
2...12	Bit 7...0: reserved	

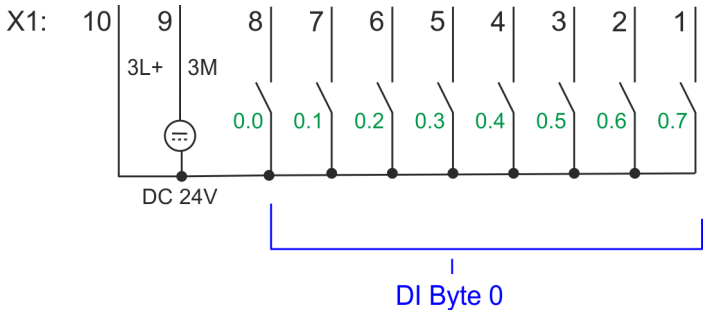
5.4 Digital input

5.4.1 Properties

- 16xDC 24V
- Maximum input frequency
  - 10 inputs: 100kHz
  - 6 inputs: 1kHz
- Interrupt functions parameterizable
- Status indication via LEDs

5.4.2 Wiring

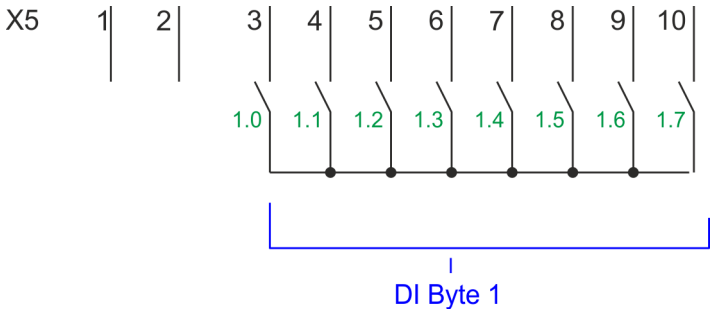
X1: DI byte 0



X1	Function	Type		Description
1	DI 0.7	I	green	Digital input DI 7 / Counter 2 (B) / Frequency 2 *
2	DI 0.6	I	green	Digital input DI 6 / Counter 2 (A) *
3	DI 0.5	I	green	Digital input DI 5
4	DI 0.4	I	green	Digital input DI 4 / Counter 1 (B) / Frequency 1 *
5	DI 0.3	I	green	Digital input DI 3 / Counter 1 (A) *
6	DI 0.2	I	green	Digital input DI 2
7	DI 0.1	I	green	Digital input DI 1 / Counter 0 (B) / Frequency 0 *
8	DI 0.0	I	green	Digital input DI 0 / Counter 0 (A) *
9	0 V	I		3M: GND for onboard DI power section supply
10	DC 24V	I	green	3L+: DC 24V for onboard DI power section supply

*) Max. input frequency 100kHz otherwise 1kHz.

X5: DI byte 1



X5	Function	Type	LED	Description
1	-	-		reserved
2	-	-		reserved
3	DI 1.0	I	green	Digital input DI 8
4	DI 1.1	I	green	Digital input DI 9 / Counter 3 (A) *
5	DI 1.2	I	green	Digital input DI 10 / Counter 3 (B) / Frequency 3 *
6	DI 1.3	I	green	Digital input DI 11 / Gate 3 *
7	DI 1.4	I	green	Digital input DI 12
8	DI 1.5	I	green	Digital input DI 13
9	DI 1.6	I	green	Digital input DI 14
10	DI 1.7	I	green	Digital input DI 15 / Latch 3 *

*) Max. input frequency 100kHz otherwise 1kHz.

5.4.3 Parametrization

5.4.3.1 Adress assignment

Sub module	Input address	Access	Assignment
DI24/DO16	136	BYTE	Digital input I+0.0 ... I+0.7 (X1)
	137	BYTE	Digital input I+1.0 ... I+1.7 (X5)

5.4.3.2 Hardware interrupt

Parameter hardware configuration

With the parameter *'Hardware interrupt at ...'* you can specify a hardware interrupt for each input for the corresponding edge. The hardware interrupt is disabled, if nothing is selected (default setting). A diagnostics interrupt is only supported with *Hardware interrupt lost*. Select with the arrow keys the input and enable the according hardware interrupts.

Here is valid:

- Rising edge: Edge 0-1
- Falling edge: Edge 1-0

5.4.3.3 Input delay

Parameter hardware configuration

- The input delay can be configured per channel in groups of 4.
- An input delay of 0.1ms is only possible with "fast" inputs, which have a max. input frequency of 100kHz ↬ *Chapter 5.4 'Digital input' on page 90*. Within a group, the input delay for slow inputs is limited to 0.5ms.
- Range of values: 0.1ms / 0.5ms / 3ms / 15ms

5.4.4 Status indication

X1	Function	Type		Description
1	DI 0.7	I	green	Digital input DI 7 / Counter 2 (B) / Frequency 2 *
2	DI 0.6	I	green	Digital input DI 6 / Counter 2 (A) *
3	DI 0.5	I	green	Digital input DI 5
4	DI 0.4	I	green	Digital input DI 4 / Counter 1 (B) / Frequency 1 *
5	DI 0.3	I	green	Digital input DI 3 / Counter 1 (A) *
6	DI 0.2	I	green	Digital input DI 2
7	DI 0.1	I	green	Digital input DI 1 / Counter 0 (B) / Frequency 0 *
8	DI 0.0	I	green	Digital input DI 0 / Counter 0 (A) *
9	0 V	I		3M: GND for onboard DI power section supply
10	DC 24V	I	green	3L+: DC 24V for onboard DI power section supply

*) Max. input frequency 100kHz otherwise 1kHz.

X5	Function	Type	LED	Description
1	-	-		reserved
2	-	-		reserved
3	DI 1.0	I	green	Digital input DI 8
4	DI 1.1	I	green	Digital input DI 9 / Counter 3 (A) *
5	DI 1.2	I	green	Digital input DI 10 / Counter 3 (B) / Frequency 3 *
6	DI 1.3	I	green	Digital input DI 11 / Gate 3 *
7	DI 1.4	I	green	Digital input DI 12
8	DI 1.5	I	green	Digital input DI 13
9	DI 1.6	I	green	Digital input DI 14
10	DI 1.7	I	green	Digital input DI 15 / Latch 3 *

*) Max. input frequency 100kHz otherwise 1kHz.

Digital input	LED	Description
DI +0.0 DI +0.7	green	Digital I+0.0 ... 0.7 has "1" signal
	☐	Digital I+0.0 ... 0.7 has "0" signal
DI +1.0 ... DI +1.7	green	Digital input I+1.0 ... 1.7 has "1" signal
	☐	Digital input I+1.0 ... 1.7 has "0" signal

Power supply	LED	Description
1L+	green	DC 24V electronic section supply
		DC 24V electronic section supply not available
3L+	green	DC 24V power section supply inputs OK
		DC 24V power section supply inputs not available
4L+	green	DC 24V power section supply outputs OK
		DC 24V power section supply outputs not available
4M	red	Error, overload respectively short circuit on the outputs
		no error

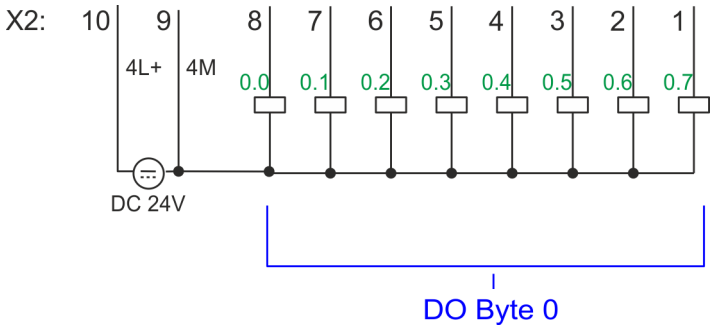
5.5 Digital output

5.5.1 Properties

- 12xDC 24V, 0.5A
- Status indication via LEDs

5.5.2 Wiring

X2: DO byte 0



X2	Function	Type	LED	Description
1	DO 0.7	O	green	Digital output DO 7
2	DO 0.6	O	green	Digital output DO 6
3	DO 0.5	O	green	Digital output DO 5
4	DO 0.4	O	green	Digital output DO 4
5	DO 0.3	O	green	Digital output DO 3 / Output channel counter 3
6	DO 0.2	O	green	Digital output DO 2 / Output channel counter 2
7	DO 0.1	O	green	Digital output DO 1 / PWM 1 / Output channel counter 1
8	DO 0.0	O	green	Digital output DO 0 / PWM 0 / Output channel counter 0
9	0 V	I	red	4M: GND for onboard DO power section supply / GND PWM LED (red) is on at short circuit respectively overload
10	DC 24V	I	green	4L+: DC 24V for onboard DO power section supply

5.5.3 Parametrization

5.5.3.1 Address assignment

Sub module	Output address	Access	Assignment
DI24/DO16	136	BYTE	Digital output Q+0.0 ... Q+0.7 (X2)
	137	BYTE	Digital output Q+1.0 ... Q+1.3 (X6)

5.5.4 Status indication

X2	Function	Type	LED	Description
1	DO 0.7	O	green	Digital output DO 7
2	DO 0.6	O	green	Digital output DO 6
3	DO 0.5	O	green	Digital output DO 5
4	DO 0.4	O	green	Digital output DO 4
5	DO 0.3	O	green	Digital output DO 3 / Output channel counter 3
6	DO 0.2	O	green	Digital output DO 2 / Output channel counter 2
7	DO 0.1	O	green	Digital output DO 1 / PWM 1 / Output channel counter 1
8	DO 0.0	O	green	Digital output DO 0 / PWM 0 / Output channel counter 0
9	0 V	I	red	4M: GND for onboard DO power section supply / GND PWM LED (red) is on at short circuit respectively overload
10	DC 24V	I	green	4L+: DC 24V for onboard DO power section supply

Digital output	LED	Description
DO +0.0 ... DO +0.7	green	Digital output Q+0.0 ... 0.7 has "1" signal
		Digital output Q+0.0 ... 0.7 has "0" signal
DO +1.0 ... DO +1.3	green	Digital output Q+1.0 ... 1.3 has "1" signal
		Digital output Q+1.0 ... 1.3 has "0" signal

Power supply	LED	Description
1L+	green	DC 24V electronic section supply
		DC 24V electronic section supply not available
3L+	green	DC 24V power section supply inputs OK
		DC 24V power section supply inputs not available
4L+	green	DC 24V power section supply outputs OK
		DC 24V power section supply outputs not available
4M	red	Error, overload respectively short circuit on the outputs
		no error

5.6 Counting

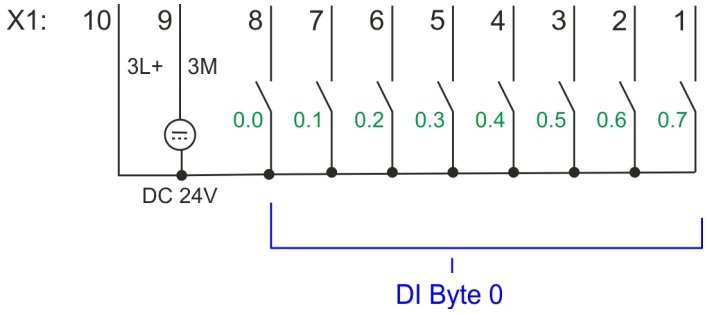
5.6.1 Properties

- 4 channels
- Various counting modes
  - once
  - continuously
  - periodically
- Control by the user program via blocks

5.6.2 Wiring

5.6.2.1 Counter inputs

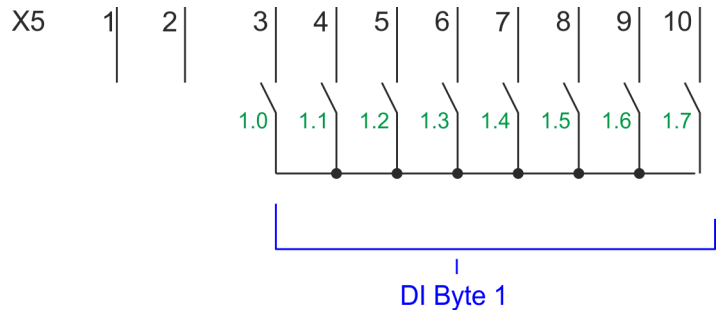
X1: DI byte 0



X1	Function	Type		Description
1	DI 0.7	I	green	Digital input DI 7 / Counter 2 (B) / Frequency 2 *
2	DI 0.6	I	green	Digital input DI 6 / Counter 2 (A) *
3	DI 0.5	I	green	Digital input DI 5
4	DI 0.4	I	green	Digital input DI 4 / Counter 1 (B) / Frequency 1 *
5	DI 0.3	I	green	Digital input DI 3 / Counter 1 (A) *
6	DI 0.2	I	green	Digital input DI 2
7	DI 0.1	I	green	Digital input DI 1 / Counter 0 (B) / Frequency 0 *
8	DI 0.0	I	green	Digital input DI 0 / Counter 0 (A) *
9	0 V	I		3M: GND for onboard DI power section supply
10	DC 24V	I	green	3L+: DC 24V for onboard DI power section supply

*) Max. input frequency 100kHz otherwise 1kHz.

X5: DI byte 1



X5	Function	Type	LED	Description
1	-	-		reserved
2	-	-		reserved
3	DI 1.0	I	green	Digital input DI 8
4	DI 1.1	I	green	Digital input DI 9 / Counter 3 (A) *
5	DI 1.2	I	green	Digital input DI 10 / Counter 3 (B) / Frequency 3 *
6	DI 1.3	I	green	Digital input DI 11 / Gate 3 *
7	DI 1.4	I	green	Digital input DI 12
8	DI 1.5	I	green	Digital input DI 13
9	DI 1.6	I	green	Digital input DI 14
10	DI 1.7	I	green	Digital input DI 15 / Latch 3 *

*) Max. input frequency 100kHz otherwise 1kHz.

Input signals

The following sensors can be connected

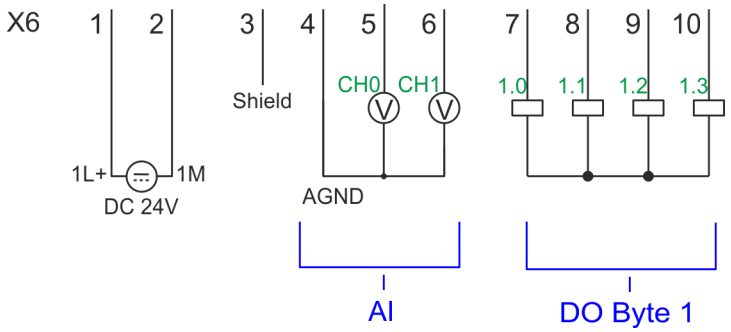
- 24V incremental encoders with two phase-shifted by 90° tracks
- 24V pulse encoder with direction signal
- 24V initiator as BERO or beam sensor

For not all inputs are available at the same time, for every counter you may define the input assignment via the parameterization for the following input signals:

- *Counter$_x$ (A)*
  - Pulse input for counter signal respectively track A of an encoder for 1-, 2- or 4-fold evaluation.
- *Counter$_x$ (B)*
  - Direction signal respectively track B of the encoder. Via the parameterization you may invert the direction signal.
- *Gate 3*
  - Via this input you can if parameterized open the HW gate of *Counter 3* with edge 0-1 and start counting.
- *Latch 3*
  - Via this input via edge 0-1 the current counter value of *Counter 3* is stored in a memory that you may read if needed.

5.6.2.2 Counter outputs

X6: DC 24V, AI, DO byte 1



X6	Function	Type	LED	Description
1	Sys DC 24V	I	green	1L+: DC 24V for electronic section supply
2	Sys 0V	I		1M: GND for electronic section supply
3	Shield	I		Shield
4	AGND	I		GND for analog inputs
5	AI 0	I		Analog input AI 0
6	AI 1	I		Analog input AI 1
7	DO 1.0	O	green	Digital output DO 8
8	DO 1.1	O	green	Digital output DO 9
9	DO 1.2	O	green	Digital output DO 10
10	DO 1.3	O	green	Digital output DO 11

Output channel Counter$_x$

Every counter has an assigned output channel. For each counter you can specify the behavior of the counter output via the parametrization with *'Characteristics of the output'* and *'Pulse duration'*. ↳ *Chapter 5.6.4.4 'Counter' on page 100*

5.6.3 Proceeding

Hardware configuration

In the Siemens SIMATIC Manager the following steps should be executed:

1. Perform a hardware configuration for the CPU. ↳ *Chapter 4.4 'Hardware configuration - CPU' on page 58*
2. Double-click the counter sub module of the CPU CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
   ⇨ The dialog *'Properties'* is opened.
3. As soon as you select the operating mode for the corresponding channel, a dialog box with default values for this counter mode is created and shown. ↳ *Chapter 5.6.5 'Counter operating modes' on page 104*
4. Perform the required parameter settings.
5. Safe your project with *'Station ➔ Safe and compile'*.
6. Transfer your project to your CPU.

User program

- The SFB 47 should cyclically be called (e.g. OB 1) for controlling the counter functions.
- The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.
- Among others the SFB 47 contains a request interface. Hereby you get read and write access to the registers of the appropriate counter.
- So that a new job may be executed, the previous job must have be finished with JOB_DONE = TRUE.
- Per channel you may call the SFB in each case with the same instance DB, since the data necessary for the internal operational are stored here.
- Writing accesses to outputs of the instance DB is not permissible.
- Starting, stopping and interrupting a count function of *Counter 0* to *Counter 2* exclusively happens via the SW gate by setting the SW gate of the SFB 47.
  You can also activate input *'Gate 3'* via the parametrization for *Counter 3*.

More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.

5.6.4 Parametrization

5.6.4.1 Address assignment

Sub module	Input address	Access	Assignment
Counter	816	DINT	Channel 0: Counter value / Frequency value
	820	DINT	Channel 1: Counter value / Frequency value
	824	DINT	Channel 2: Counter value / Frequency value
	828	DINT	Channel 3: Counter value / Frequency value

Sub module	Output address	Access	Assignment
Counter	816	DWORD	reserved
	820	DWORD	reserved
	824	DWORD	reserved
	828	DWORD	reserved

5.6.4.2 Interrupt selection

Via *'Basic parameters'* you can reach *'Select interrupt'*. Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is disabled.
- Process: The following events of the counter can trigger a hardware interrupt (selectable via *'Count'*):
  - Hardware gate opening
  - Hardware gate closing
  - On reaching the comparator
  - on Counting pulse
  - on overflow
  - on underflow
- Diagnostics+process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

5.6.4.3 Operating mode per channel

Parameter hardware configuration

Select via *'Channel'* the channel select via *'Operating'* the operating mode. The following operating modes are supported:

- Not parameterized: Channel is deactivated
- *Chapter 5.6.5.1 'Count continuously' on page 104*
- *Chapter 5.6.5.2 'Count once' on page 105*
- *Chapter 5.6.5.3 'Count periodically' on page 108*
- *Chapter 5.7 'Frequency measurement' on page 117*
- *Chapter 5.8 'Pulse width modulation - PWM' on page 123*

Depending on the selected operating mode default values are loaded and shown in an additional register.

5.6.4.4 Counter

Parameter hardware configuration

Default values and structure of this dialog box depend on the selected *'Operating mode'*.

Parameter overview

Operating parameters	Description	Assignment
Main count direction	■ *None* No restriction of the counting range ■ *Up:* Restricts the up-counting range. The counter starts from 0 or *load value*, counts in positive direction up to the declaration *end value* -1 and then jumps back to *load value* at the next positive transducer pulse. ■ *Down:* Restricts the down-counting range. The counter starts from the declared *start value* or *load value* in negative direction, counts to 1 and then jumps to *start value* at the next negative encoder pulse. Function is disable with *count continuously*.	■ None
Gate function	■ *Cancel count:* The count starts when the gate opens and resumes at the *load value* when the gate opens again. ■ *Stop count:* The count is interrupted when the gate closes and resumed at the last actual counter value when the gate opens again. *Chapter 5.6.6.2 'Gate function' on page 111*	Abort count process
Start value End value	*Start value* with counting direction backward. *End value* with main counting direction forward. Range of values: 2...2147483647 (2^{31}-1)	2147483647 (2^{31}-1)
Comparison value	The count value is compared with the *comparison value*. See also the parameter "Characteristics of the output": ■ No main counting direction – Range of values: -2)^{31} to +2)^{31}-1 ■ Main counting direction forward – Range of values: -2^{31} to end value-1 ■ Main counting direction backward – Range of values: 1 to +2^{31}-1	0
Hysteresis	The *hysteresis* serves the avoidance of many toggle processes of the output, if the counter value is in the range of the *comparison value*. 0, 1: *Hysteresis* disabled Range of values: 0 to 255	0

Input	Description	Assignment
Signal evaluation	Specify the signal of the connected encoder: ■ Pulse/direction At the input count and direction signal are connected ■ At the input there is an encoder connected with the following evaluation: – Rotary encoder single – Rotary encoder double – Rotary encoder quadruple	Pulse/direction
Hardware gate	Gate control exclusively via channel 3: ■ enabled: The gate control for channel 3 happens via SW and HW gate ■ disabled: The gate control for channel 3 exclusively happens via SW gate *Chapter 5.6.6.2 'Gate function' on page 111*	disabled
Count direction inverted	Invert the input signal *'Direction'*: ■ enabled: The input signal is inverted ■ disabled: The input signal is not inverted	disabled

Output	Description	Assignment
Characteristics of the output	The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter. ■ No comparison: The output is used as normal output and STS_CMP remains reset. ■ Comparator – Counter value ≥ Comparison value – Counter value ≤ Comparison value ■ Pulse at *comparison value* – To adapt the used actuators you can specify a *pulse duration*. The output is set for the specified *pulse duration* when the counter value reaches the *comparison value*. When you've set a main counting direction the output is only set at reaching the *comparison value* from the main counting direction.	No comparison
Pulse duration	Here you can specify the *pulse duration* for the output signal. ■ The *pulse duration* starts with the setting of the according digital output. ■ The inaccuracy of the *pulse duration* is less than 1ms. ■ There is no past triggering of the *pulse duration* when the *comparison value* has been left and reached again during pulse output. ■ If the *pulse duration* is changed during operation, it will take effect with the next pulse. ■ If the *pulse duration* = 0, the output is set until the comparison condition is not longer fulfilled. Range of values: 0...510ms in steps of 2ms	0

Hardware interrupt	Description	Assignment
Hardware gate opening	Hardware interrupt by edge 0-1 exclusively at HW gate channel 3 ■ enabled: Process interrupt by edge 0-1 exclusively at HW gate channel 3 with open SW gate ■ disabled: no hardware interrupt	disabled
Hardware gate closing	Hardware interrupt by edge 1-0 exclusively at HW gate channel 3 ■ enabled: Process interrupt by edge 1-0 exclusively at HW gate channel 3 with open SW gate ■ disabled: no hardware interrupt	disabled
On reaching comparator	Hardware interrupt on reaching *comparator* ■ enabled: Hardware interrupt when comparator is triggered, can be configured via *'Characteristics of the output'* ■ disabled: no hardware interrupt	disabled
Overflow	Hardware interrupt overflow ■ enabled: Hardware interrupt on overflow the upper counter limit ■ disabled: no hardware interrupt	disabled
Underflow	Hardware interrupt on underrun ■ enabled: Hardware interrupt on underflow the lower counter limit ■ disabled: no hardware interrupt	disabled

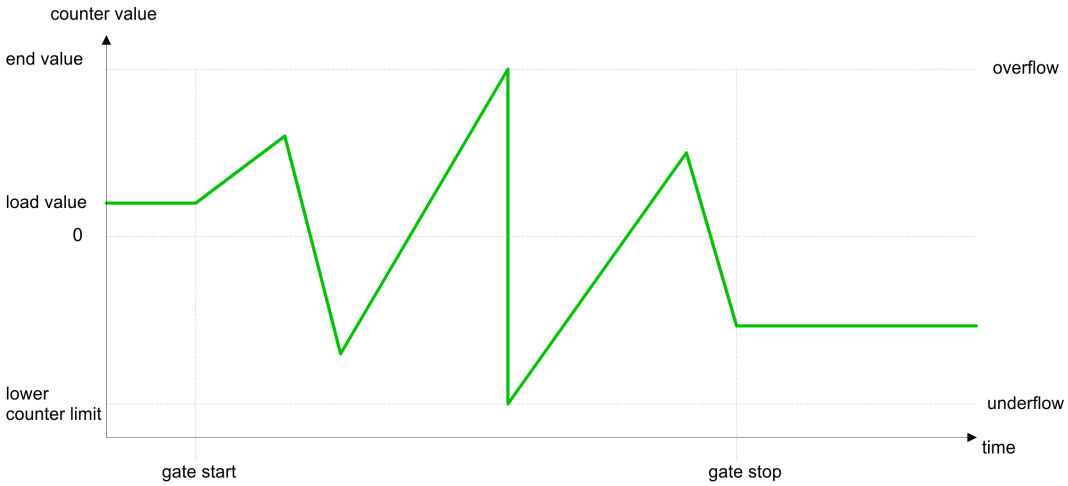
Max. frequency	Description	Assignment
Counting signals/HW gate	Specify the max. frequency for track A/pulse, track B/direction and HW gate Range of values: 1, 2, 5, 10, 30, 60kHz	60kHz
Latch	Specify the max. frequency for the latch signal Range of values: 1, 2, 5, 10, 30, 60kHz	10kHz

5.6.5 Counter operating modes

5.6.5.1 Count continuously

- In this operating mode the counter counts starting with the *load value*.
- When the counter counts forward and reaches the upper count limit and another counting pulse in positive direction arrives, it jumps to the lower count limit and counts from there on.
- When the counter counts backwards and reaches the lower count limit and another counting pulse in negative direction arrives, it jumps to the upper count limit and counts from there on.
- The counter limits are fix set to maximum range.
- With overflow or underflow the status bits STS_OFLW respectively STS_UFLW in the SFB 47 are set. These bits remain set until these are reset with RES_STS. If enabled additionally a hardware interrupt is triggered.

Limits	Valid range of values
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 (2^{31} -1)



5.6.5.2 Count once

5.6.5.2.1 No main counting direction

- The counter counts once starting with *load value*.
- It is counted forward or backward.
- The counter limits are fix set to maximum range.
- At over- or underflow at the count limits, the counter jumps to the according other count limit and the gate is automatically closed.
- To restart the count process, you have to generate an edge 0-1 at the gate *Chapter 5.6.6.2 'Gate function' on page 111*.
- With the configured *'Gate function' 'Interrupt count'* the counting is continued with current *Counter value*.
- With configured *'Gate function' 'Cancel count'* the counter starts with the *Load value*.

Limits	Valid range of values
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 (2^{31} -1)

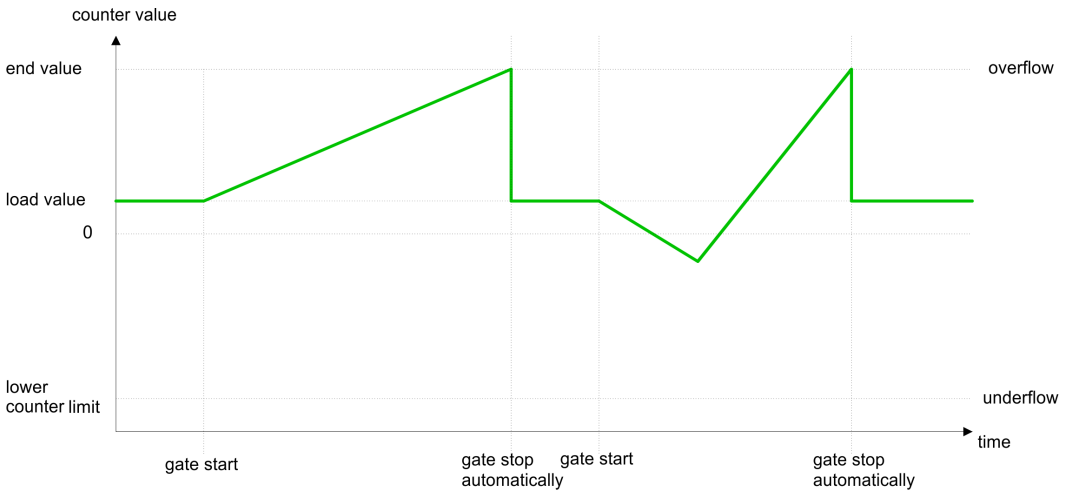
Interrupting gate control

Aborting gate control

5.6.5.2.2 Main counting direction forward

- The counter counts forward starting with the *load value*.
- When the counter reaches the *End value* -1 in positive direction, it jumps to the *load value* at the next count pulse and the gate is automatically closed. If enabled additionally a hardware interrupt is triggered.
- To restart the count process, you have to generate an edge 0-1 at the gate *Chapter 5.6.6.2 'Gate function' on page 111*. The counter counts starting with the *load value*.
- You may exceed the lower count limit.

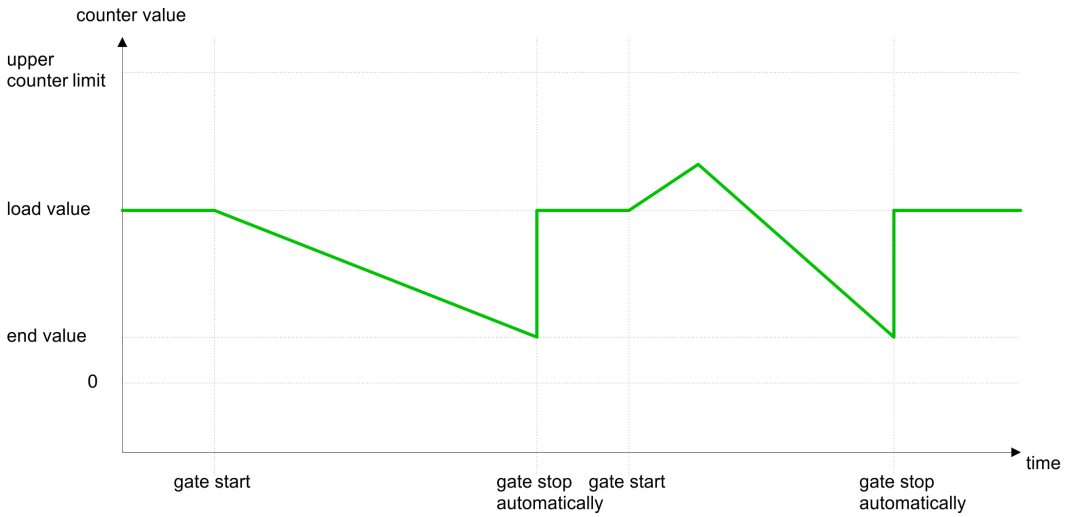
Limits	Valid range of values
End value	-2 147 483 647 (-2^{31} +1) up to +2 147 483 647 (2^{31} -1)
Lower count limit	-2 147 483 648 (-2^{31})



5.6.5.2.3 Main counting direction backward

- The counter counts backward starting with the *load value*.
- When the counter reaches the *End value* +1 in positive direction, it jumps to the *load value* at the next count pulse and the gate is automatically closed. If enabled additionally a hardware interrupt is triggered.
- To restart the count process, you have to generate an edge 0-1 at the gate *Chapter 5.6.6.2 'Gate function' on page 111*. The counter counts starting with the *load value*.
- You may exceed the upper count limit.

Limits	Valid range of values
End value	-2 147 483 648 (-2^{31}) up to +2 147 483 646 (2^{31} -2)
Upper count limit	+2 147 483 647 (2^{31} -1)

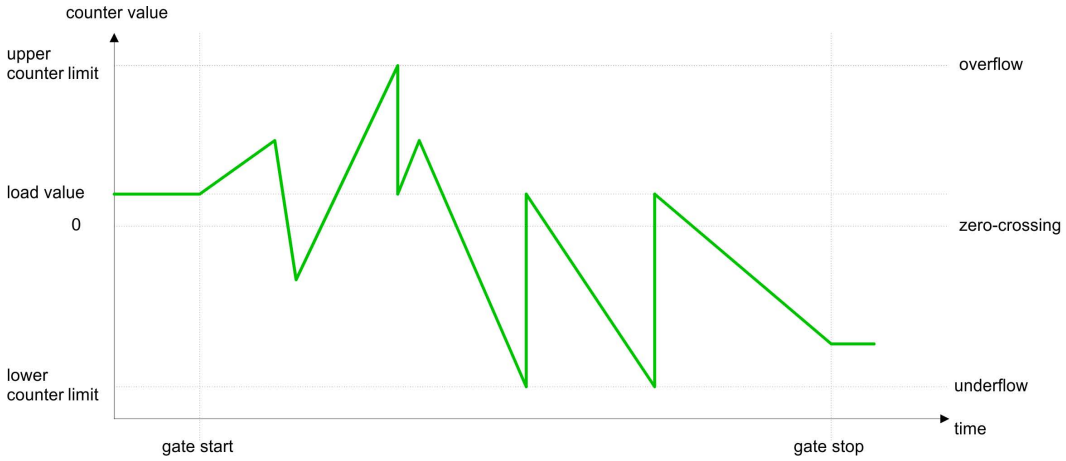


5.6.5.3 Count periodically

5.6.5.3.1 No main counting direction

- The counter counts forward or backwards starting with the *load value*.
- At over- or underrun at the count limits, the counter jumps to the *load value* and continues counting. If enabled additionally a hardware interrupt is triggered.
- The counter limits are fix set to maximum range.

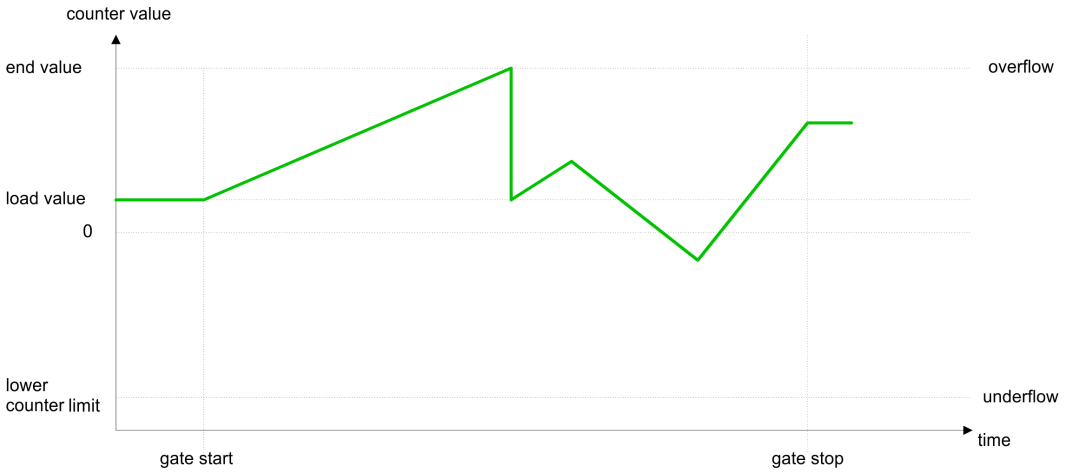
Limits	Valid range of values
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 (2^{31} -1)



5.6.5.3.2 Main counting direction forward

- The counter counts forward starting with the *load value*.
- When the counter reaches the end value -1 in positive direction, it jumps to the *load* value at the next positive count pulse and continues counting. If enabled additionally a hardware interrupt is triggered.
- You may exceed the lower count limit.

Limits	Valid range of values
End value	-2 147 483 647 (-2^{31} +1) up to +2 147 483 647 (2^{31} -1)
Lower count limit	-2 147 483 648 (-2^{31})

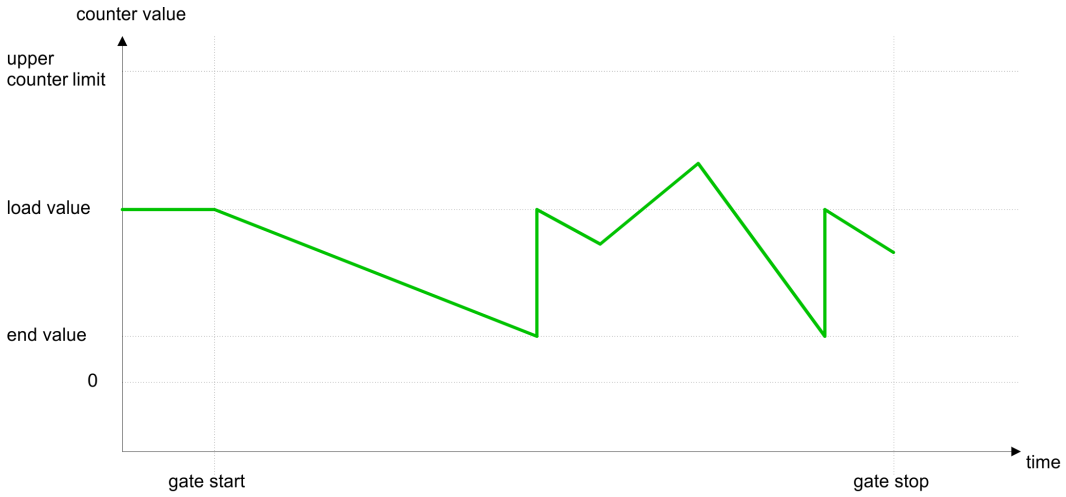


5.6.5.3.3 Main counting direction backward

Main counting direction backward

- The counter counts backward starting with the *load value*.
- When the counter reaches the *end value* +1 in positive direction, it jumps to the *load value* at the next negative count pulse and continues counting. If enabled additionally a hardware interrupt is triggered.
- You may exceed the upper count limit.

Limits	Valid range of values
End value	-2 147 483 648 (-2^{31}) up to +2 147 483 646 (2^{31} -2)
Upper count limit	+2 147 483 647 (2^{31} -1)

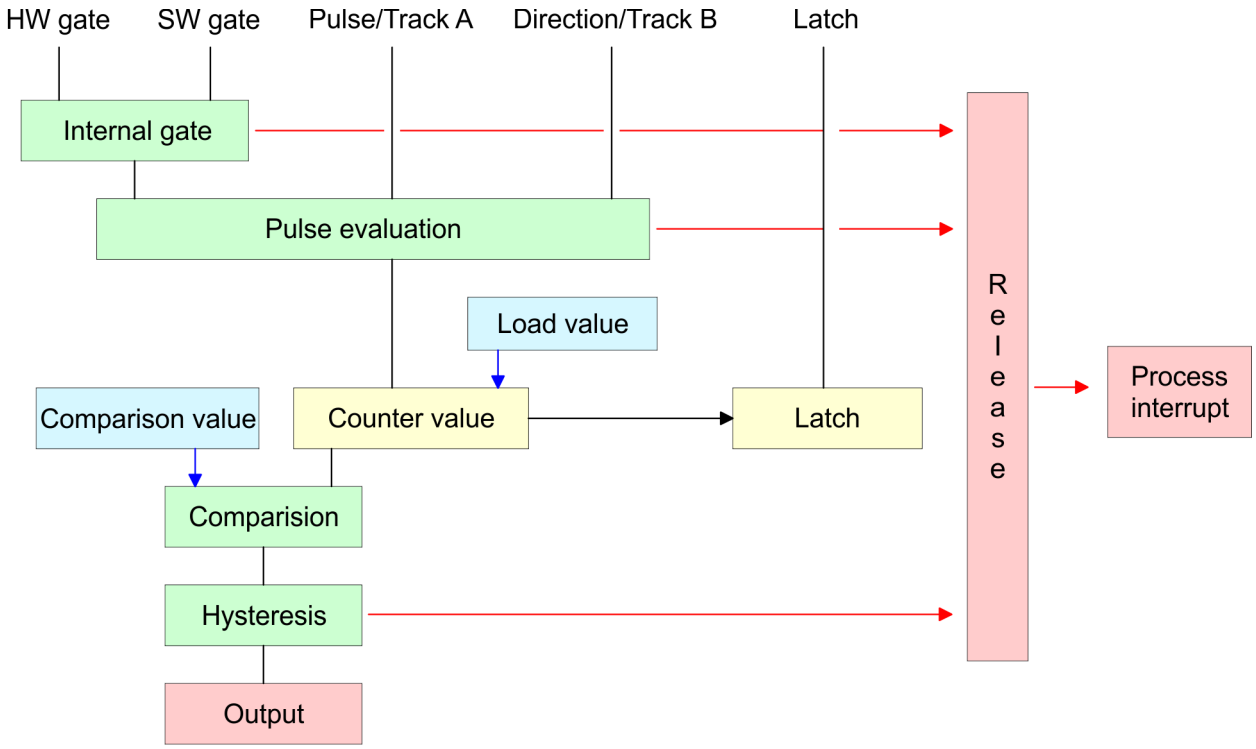


5.6.6 Counter - Additional functions

5.6.6.1 Overview

Schematic structure

The illustration shows how the additional functions influence the counting behavior. The following pages describe these additional functions in detail:



5.6.6.2 Gate function

Function

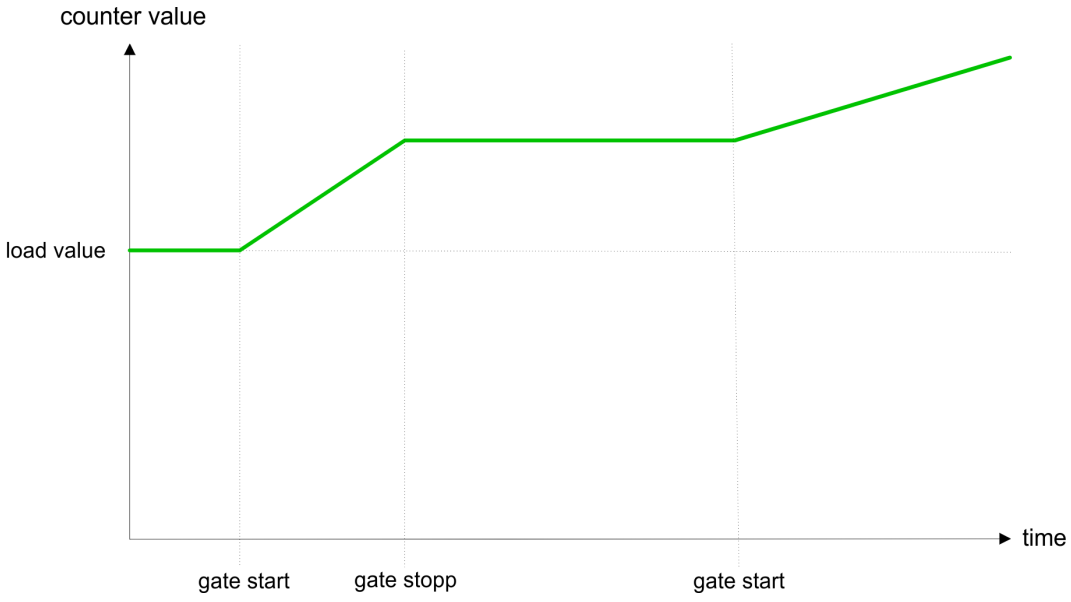
- Starting, stopping and interrupting a count function of *counter 0* to *counter 2* exclusively happens via the SW gate by setting the SW gate of SFB 47.
- Starting, stopping and interrupting a count function of *counter 3* happens via the internal gate (I gate). The i gate is the result of logic operation of HW gate and SW gate. The HW gate evaluation of the connection *'Gate 3'* may be deactivated by the parametrization. With a de-activated HW gate evaluation the triggering exclusively happens by setting the SW gate of SFB 47.

Gate function abort and interrupt

The parametrization defines if the gate interrupts or aborts the counter process.

- At *abort function* the counter starts counting with the *load value* after gate restart.

- At *interrupt function*, the counter starts counting with the last recent counter value after gate restart.



Counter 0 ... 2

SW gate	Gate function	Reaction *counter 0 ... 2*
Edge 0-1	Abort count process	Restart with *load value*
Edge 0-1	Interrupt count process	Continue

5.6.6.3 Comparator

Function

In the CPU a *comparison value* may be stored. During the counting procedure the counter value is compared with the *comparative value*. Depending on the result of the comparison the output channel of the counter and the status bit of STS_CMP of SFB 47 can be set. In addition, you can configure a hardware interrupt. A *comparison value* can be specified via the parametrization respectively the job interface of SFB 47.

5.6.6.4 Additional functions counter 3

Exclusively counter 3 has the following additional functions:

- HW gate via *Gate 3*
- Latch function

5.6.6.4.1 HW gate via *Gate 3*

Starting, stopping and interrupting a count function of counter 3 happens via the internal gate (I gate). The i gate is the result of logic operation of HW gate and SW gate. The HW gate evaluation of the connection *'Gate 3'* may be deactivated by the parametrization. With a de-activated HW gate evaluation the triggering exclusively happens by setting the SW gate of the SFB 47.

Counter 3:

SW gate	HW gate	Gate function	Reaction counter 3:
Edge 0-1	de-activated	Abort count process	Restart with load value
Edge 0-1	de-activated	Interrupt count process	Continue
Edge 0-1	1	Abort count process	Continue
1	Edge 0-1	Abort count process	Restart with load value
Edge 0-1	1	Interrupt count process	Continue
1	Edge 0-1	Interrupt count process	Continue

Counter 3 - count once

If the internal gate has been closed automatically it may only be opened again under the following conditions:		
SW gate	**HW gate**	**I gate**
1	Edge 0-1	1
Edge 0-1 (after edge 0-1 at HW gate)	Edge 0-1	1

5.6.6.4.2 Latch function

Function

- As soon as during a count process an edge 0-1 is recognized at the "Latch" input of counter 3, the current counter value is stored in the according latch register.
- You may access the latch value via the parameter LATCHVAL of the SFB 47.
- A just in LATCHVAL loaded value remains after a STOP-RUN transition.

5.6.6.5 Counter output channel

Characteristics of the output

Each counter has an output channel. You pre-define the behavior of the counter output via the parametrization:

- no comparison:
  - The output is used as normal output.
  - SFB 47:
    The input parameter CTRL_DO is effect less.
    The status bits STS_DO and STS_CMP (status comparator in the instance DB) remain reset.
- Counter value ≥ comparison value respectively counter value ≤ comparison value
  - The output remains set as long as the counter value is higher or equal *comparison value* respectively lower or equal *comparison value*.
  - SFB 47:
    Control bit CTRL_DO must be set.
    The comparison result is shown by the status bit STS_CMP. This status bit may only be reset if the comparison condition is no longer fulfilled.
- Pulse at comparison value
  - When the counter reaches the *comparison value* the output is set for the parametrized *pulse duration*. When you've set a main counting direction the output is only set at reaching the *comparison value* from the main counting direction.
    If the *pulse duration* = 0, the output is set until the comparison condition is not longer fulfilled.
  - SFB 47:
    Control bit CTRL_DO must be set.
    The status of the digital output may be shown by the status bit ST_DO.
    The comparison result is shown by the status bit STS_CMP. The bit may only be reset if the *pulse duration* has expired.
- Pulse duration
  - The *pulse duration* starts with the setting of the according digital output.
  - The inaccuracy of the *pulse duration* is less than 1ms.
  - There is no past triggering of the *pulse duration* when the *comparison value* has been left and reached again during pulse output.
  - If the *pulse duration* is changed during operation, it will take effect with the next pulse.
  - If the *pulse duration* = 0, the output is set until the comparison condition is not longer fulfilled.
  - Range of values: 0...510ms in steps of 2ms

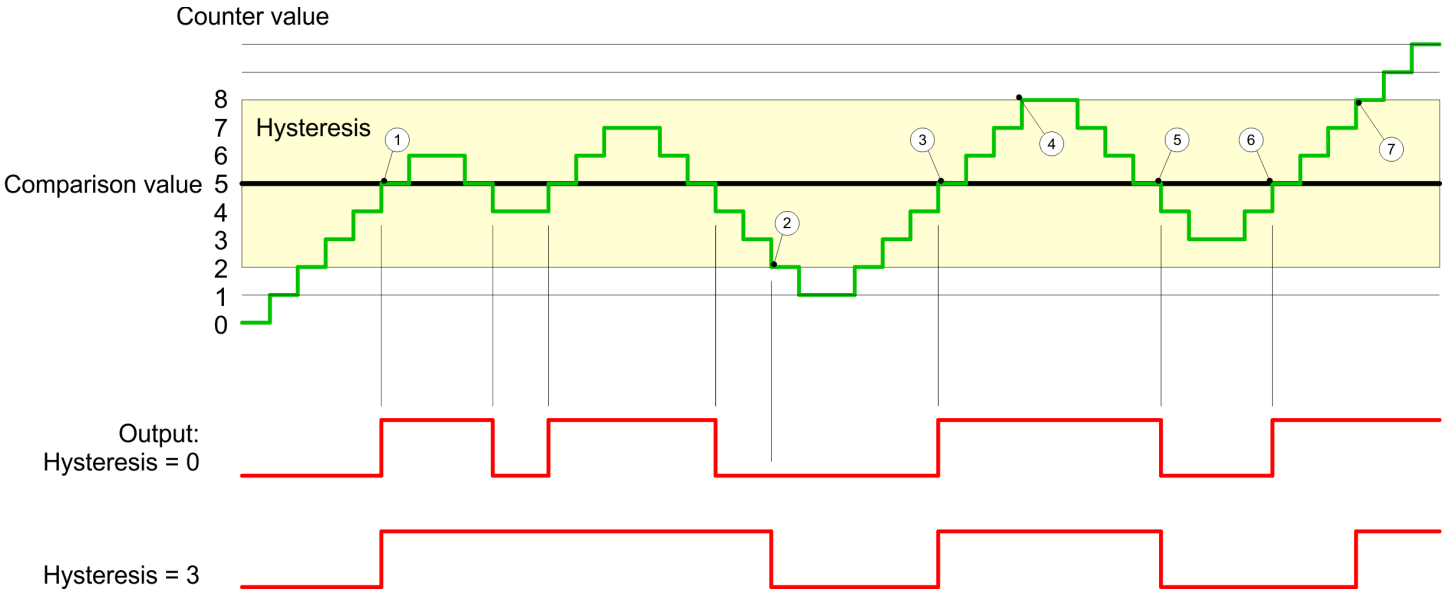
5.6.6.6 Hysteresis function

Hysteresis

- The *hysteresis* serves the avoidance of many toggle processes of the output and the interrupt, if the *counter value* is in the range of the *comparison value*.
- For the *hysteresis* you may set a range of 0 to 255.
- The settings 0 and 1 deactivate the *hysteresis*.
- The *hysteresis* influences zero run, comparison, over- and underflow.
- An activated *hysteresis* remains active after a change. The new *hysteresis* range is activated with the next *hysteresis* event.

The following pictures illustrate the output behavior for *hysteresis* 0 and *hysteresis* 3 for the according conditions:

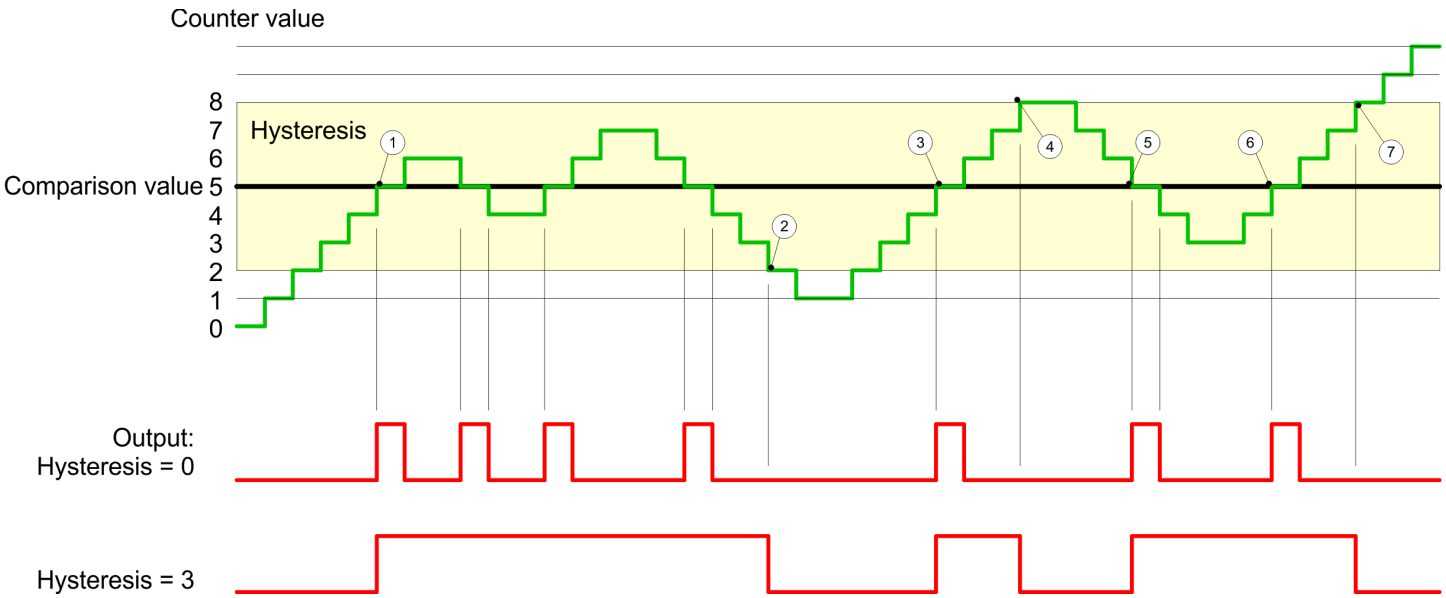
Effect at counter value ≥ comparison value



1 *Counter value* ≥ *comparison value* → output is set and *hysteresis* activated
2 Leave *hysteresis* range → output is reset
3 *Counter value* ≥ *comparison value* → output is set and *hysteresis* activated
4 Leave *hysteresis* range, output remains set for *counter value* ≥ *comparison value*
5 *counter value* < *comparison value* and *hysteresis* active → output is reset
6 *counter value* ≥ *comparison value* → output is not set for *hysteresis* active
7 Leave *hysteresis* range, output remains set for *counter value* ≥ *comparison value*

With reaching the comparison condition the *hysteresis* gets active. At active *hysteresis* the comparison result remains unchanged until the *counter value* leaves the set *hysteresis* range. After leaving the *hysteresis* range a new *hysteresis* is only activated with again reaching the comparison conditions.

Effect at pulse at comparison value with pulse duration Zero

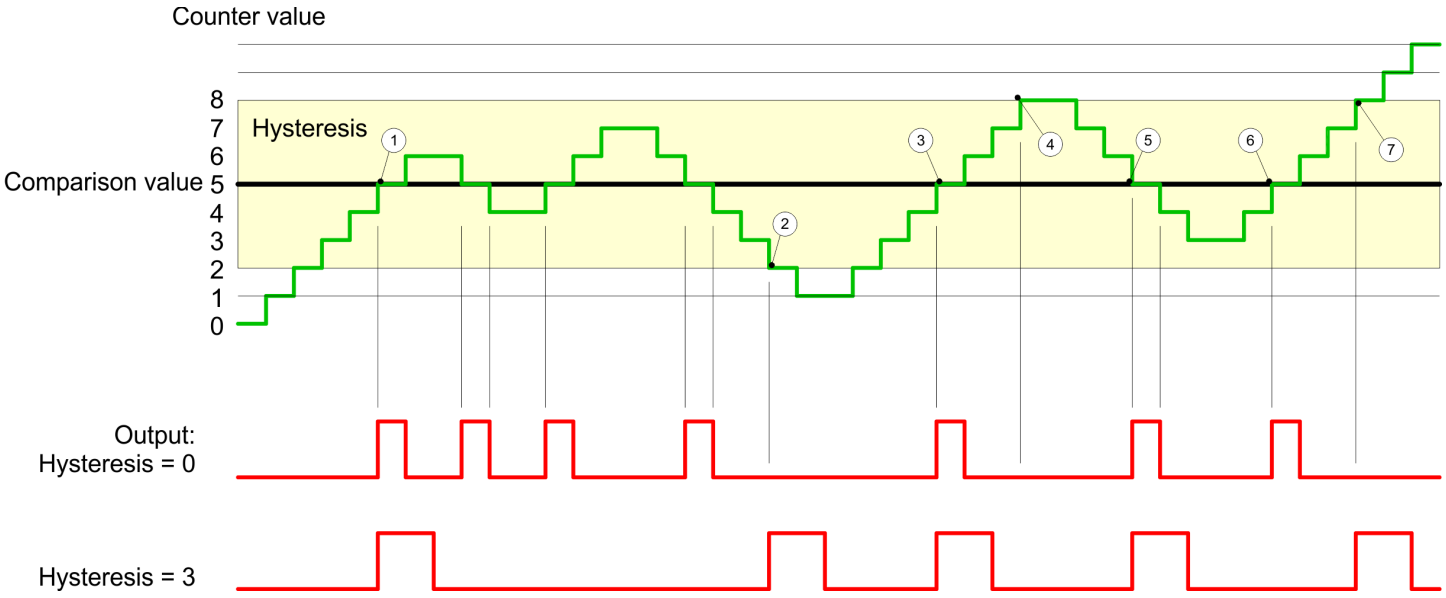


1 *Counter value* = *comparison value* → output is set and *hysteresis* activated
2 Leave *hysteresis* range → output is reset and *counter value* < *comparison value*

3 *Counter value* = *comparison value* → output is set and *hysteresis* activated
4 Output is reset for leaving *hysteresis* range and *counter value* > *comparison value*
5 *Counter value* = *comparison value* → output is set and *hysteresis* activated
6 *Counter value* = *comparison value* and *hysteresis* active → output remains set
7 Leave *hysteresis* range and *counter value* > comparison value → output is reset

With reaching the comparison condition the *hysteresis* gets active. At active *hysteresis* the comparison result remains unchanged until the *counter value* leaves the set *hysteresis range*. After leaving the *hysteresis* range a new *hysteresis* is only activated with again reaching the comparison conditions.

Effect at pulse at comparison value with pulse duration not zero



1 *Counter value* = *comparison value* → pulse of the parameterized *pulse duration* is put out, the *hysteresis* is activated and the counting direction stored
2 Leaving the *hysteresis* range contrary to the stored counting direction → pulse of the parameterized *pulse duration* is put out, the *hysteresis* is de-activated
3 *Counter value* = *comparison value* → pulse of the parameterized *pulse duration* is put out, the *hysteresis* is activated and the counting direction stored
4 Leaving the *hysteresis* range without changing counting direction → *hysteresis* is de-activated
5 *Counter value* = *comparison value* → pulse of the parameterized *pulse duration* is put out, the *hysteresis* is activated and the counting direction stored
6 *Counter value* = *comparison value* and *hysteresis* active → no pulse
7 Leaving the *hysteresis* range contrary to the stored counting direction → pulse of the parameterized *pulse duration* is put out, the *hysteresis* is de-activated

With reaching the comparison condition the *hysteresis* gets active and a pulse of the parameterized duration is put out. As long as the *counter value* is within the *hysteresis* range, no other pulse is put out. With activating the *hysteresis* the counting direction is stored in the module. If the *counter value* leaves the *hysteresis* range contrary to the stored counting direction, a pulse of the parameterized duration is put out. Leaving the *hysteresis* range without direction change, no pulse is put out.

5.6.7 Diagnostics and interrupt

Overview

GSDML

- Edge at an digital interrupt input

Via the hardware configuration you can define the following trigger for a hardware interrupt that can trigger a diagnostics interrupt:

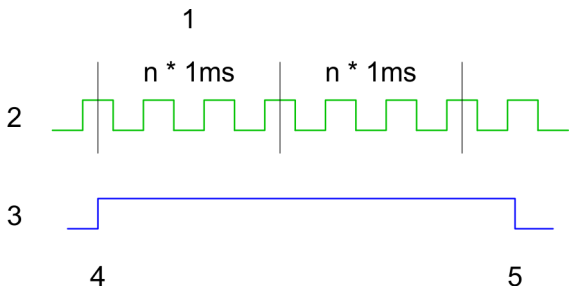
- Reaching the comparison value
- Overflow respectively at overrun upper counter limit
- Underflow respectively at underrun lower counter limit
- Opening the HW gate with open SW gate - except for counter 3
- Closing the HW gate with open SW gate - except for counter 3

5.7 Frequency measurement

5.7.1 Properties

- In this operating mode the CPU counts the incoming pulses during a specified integration time and outputs them as frequency value.
- Integration time 10ms ... 10000ms in steps of 1ms configurable
- Control by the user program via SFB 48

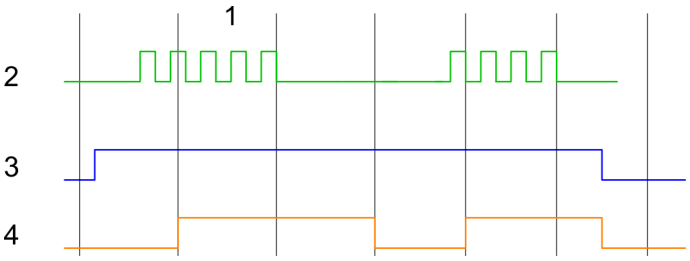
More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.



1 Integration time
2 Counting pulse
3 SW gate
4 Frequency measurement start
5 Frequency measurement stop

Measuring procedure

- The measurement is carried out during the integration time and is updated after the integration time has expired.
- If the period of the measured frequency exceeds the assigned integration time, this means there was no edge 0-1 during the measurement, the measurement value 0 is returned.
- The calculated frequency value is supplied in "mHz" units.
- The measurement value can be read with *MEAS_VAL* from SFB 48.
- The number of activated channels does not influence the max. frequency, which is defined in the technical data.



1 Integration time
2 Counting pulse
3 SW gate
4 Evaluated frequency

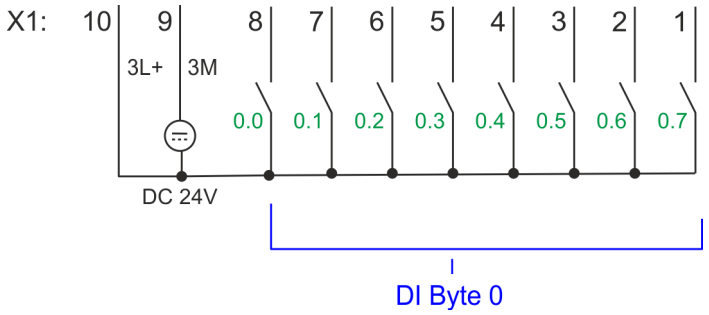
The counting function is disabled during the pulse width modulation on the same channel.

5.7.2 Wiring

5.7.2.1 Frequency measurement inputs

Connect the signal to be measured at input B of the corresponding counter.

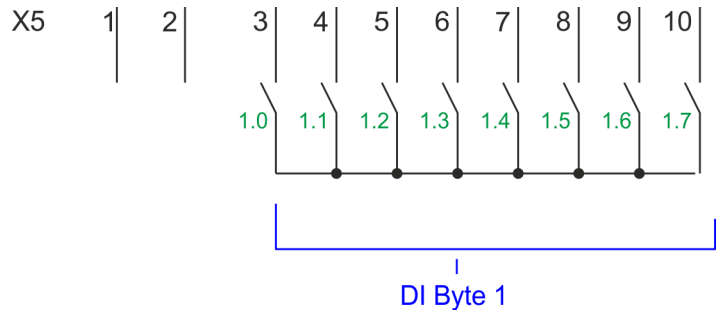
X1: DI byte 0



X1	Function	Type		Description
1	DI 0.7	I	green	Digital input DI 7 / Counter 2 (B) / Frequency 2 *
2	DI 0.6	I	green	Digital input DI 6 / Counter 2 (A) *
3	DI 0.5	I	green	Digital input DI 5
4	DI 0.4	I	green	Digital input DI 4 / Counter 1 (B) / Frequency 1 *
5	DI 0.3	I	green	Digital input DI 3 / Counter 1 (A) *
6	DI 0.2	I	green	Digital input DI 2
7	DI 0.1	I	green	Digital input DI 1 / Counter 0 (B) / Frequency 0 *
8	DI 0.0	I	green	Digital input DI 0 / Counter 0 (A) *
9	0 V	I		3M: GND for onboard DI power section supply
10	DC 24V	I	green	3L+: DC 24V for onboard DI power section supply

*) Max. input frequency 100kHz otherwise 1kHz.

X5: DI byte 1



X5	Function	Type	LED	Description
1	-	-		reserved
2	-	-		reserved
3	DI 1.0	I	green	Digital input DI 8
4	DI 1.1	I	green	Digital input DI 9 / Counter 3 (A) *
5	DI 1.2	I	green	Digital input DI 10 / Counter 3 (B) / Frequency 3 *
6	DI 1.3	I	green	Digital input DI 11 / Gate 3 *
7	DI 1.4	I	green	Digital input DI 12
8	DI 1.5	I	green	Digital input DI 13
9	DI 1.6	I	green	Digital input DI 14
10	DI 1.7	I	green	Digital input DI 15 / Latch 3 *

*) Max. input frequency 100kHz otherwise 1kHz.

5.7.3 Proceeding

Hardware configuration

In the Siemens SIMATIC Manager the following steps should be executed:

1. Perform a hardware configuration for the CPU. *Chapter 4.4 'Hardware configuration - CPU' on page 58*
2. Double-click the counter sub module of the CPU 314C-2 PN/DP.
   ⇨ The dialog *'Properties'* is opened.
3. As soon as you select the operating mode for the corresponding channel, a dialog box with default values for this counter mode is created and shown. Select for the corresponding channel the operating mode *'Frequency counting'*. *Chapter 5.6.5 'Counter operating modes' on page 104*
4. Perform the required parameter settings.
5. Safe your project with *'Station ➔ Safe and compile'*.
6. Transfer your project to your CPU.

User program

- The SFB 48 should cyclically be called (e.g. OB 1) for controlling the frequency measurement.
- The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.

5.7.4 Parametrization

5.7.4.1 Address assignment

Sub module	Input address	Access	Assignment
Counter	816	DINT	Channel 0: Counter value / Frequency value
	820	DINT	Channel 1: Counter value / Frequency value
	824	DINT	Channel 2: Counter value / Frequency value
	828	DINT	Channel 3: Counter value / Frequency value

Sub module	Output address	Access	Assignment
Counter	816	DWORD	reserved
	820	DWORD	reserved
	824	DWORD	reserved
	828	DWORD	reserved

5.7.4.2 Interrupt selection

Via *'Basic parameters'* you can reach *'Select interrupt'*. Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is de-activated.
- Process: The following events of the frequency measurement can trigger a hardware interrupt (selectable via *'Frequency counting'*):
  – End of measurement
- Diagnostics and process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

5.7.4.3 Operating mode per channel

Parameter hardware configuration

Select via *'Channel'* the channel select via *'Operating'* the operating mode. The following operating modes are supported:

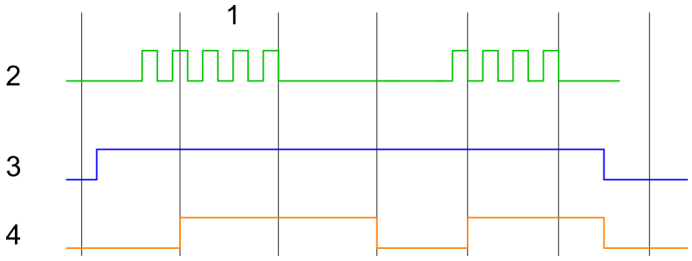
- Not parameterized: Channel is deactivated
- *Chapter 5.6.5.1 'Count continuously' on page 104*
- *Chapter 5.6.5.2 'Count once' on page 105*
- *Chapter 5.6.5.3 'Count periodically' on page 108*
- *Chapter 5.7 'Frequency measurement' on page 117*
- *Chapter 5.8 'Pulse width modulation - PWM' on page 123*

Depending on the selected operating mode default values are loaded and shown in an additional register.

5.7.4.4 Frequency measurement

Parameter hardware configuration

Default values and structure of this dialog box depend on the selected *'Operating mode'*. The following parameters are supported:



1 Integration time
2 Counting pulse
3 SW gate
4 Evaluated frequency

Parameter overview

Operating parameters	Description	Assignment
Integration time	Specify the integration time Range of values: 10ms ... 10000ms in steps of 1ms	100ms
max. counting frequency...	Specify the max. Frequency for the corresponding input Range of values: 1, 2, 5, 10, 30, 60kHz	60kHz

Hardware interrupt	Description	Assignment
End of measurement	Hardware interrupt at end of measurement	de-activated

5.7.5 Status indication

X1	Function	Type		Description
1	DI 0.7	I	green	Digital input DI 7 / Counter 2 (B) / Frequency 2 *
2	DI 0.6	I	green	Digital input DI 6 / Counter 2 (A) *
3	DI 0.5	I	green	Digital input DI 5
4	DI 0.4	I	green	Digital input DI 4 / Counter 1 (B) / Frequency 1 *
5	DI 0.3	I	green	Digital input DI 3 / Counter 1 (A) *
6	DI 0.2	I	green	Digital input DI 2
7	DI 0.1	I	green	Digital input DI 1 / Counter 0 (B) / Frequency 0 *
8	DI 0.0	I	green	Digital input DI 0 / Counter 0 (A) *
9	0 V	I		3M: GND for onboard DI power section supply
10	DC 24V	I	green	3L+: DC 24V for onboard DI power section supply

*) Max. input frequency 100kHz otherwise 1kHz.

X5	Function	Type	LED	Description
1	-	-		reserved
2	-	-		reserved
3	DI 1.0	I	green	Digital input DI 8
4	DI 1.1	I	green	Digital input DI 9 / Counter 3 (A) *
5	DI 1.2	I	green	Digital input DI 10 / Counter 3 (B) / Frequency 3 *
6	DI 1.3	I	green	Digital input DI 11 / Gate 3 *
7	DI 1.4	I	green	Digital input DI 12
8	DI 1.5	I	green	Digital input DI 13
9	DI 1.6	I	green	Digital input DI 14
10	DI 1.7	I	green	Digital input DI 15 / Latch 3 *

*) Max. input frequency 100kHz otherwise 1kHz.

Digital input	LED	Description
DI +0.0 DI +0.7	green	Digital I+0.0 ... 0.7 has "1" signal
	☐	Digital I+0.0 ... 0.7 has "0" signal
DI +1.0 ... DI +1.7	green	Digital input I+1.0 ... 1.7 has "1" signal
	☐	Digital input I+1.0 ... 1.7 has "0" signal

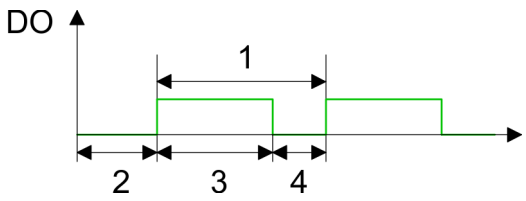
Power supply	LED	Description
1L+	green	DC 24V electronic section supply
		DC 24V electronic section supply not available
3L+	green	DC 24V power section supply inputs OK
		DC 24V power section supply inputs not available
4L+	green	DC 24V power section supply outputs OK
		DC 24V power section supply outputs not available
4M	red	Error, overload respectively short circuit on the outputs
		no error

5.8 Pulse width modulation - PWM

5.8.1 Properties

- By presetting of time parameters, the CPU evaluates a pulse sequence with according pulse/pause ratio and outputs it via the according output channel.
- Channel 0 and 1 are supported
- Control by the user program via SFB 49

More information about the usage of this block may be found in the manual "SPEED7 Operation List" from VIPA.



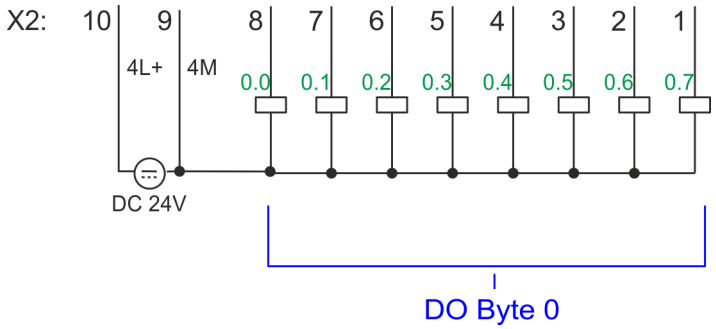
1 Period
2 On-delay
3 Pulse duration
4 Pulse pause

The counting function is disabled during the pulse width modulation on the same channel.

5.8.2 Wiring

5.8.2.1 Pulse width modulation outputs

X2: DO byte 0



X2	Function	Type	LED	Description
1	DO 0.7	O	green	Digital output DO 7
2	DO 0.6	O	green	Digital output DO 6
3	DO 0.5	O	green	Digital output DO 5
4	DO 0.4	O	green	Digital output DO 4
5	DO 0.3	O	green	Digital output DO 3 / Output channel counter 3
6	DO 0.2	O	green	Digital output DO 2 / Output channel counter 2
7	DO 0.1	O	green	Digital output DO 1 / PWM 1 / Output channel counter 1
8	DO 0.0	O	green	Digital output DO 0 / PWM 0 / Output channel counter 0
9	0 V	I	red	4M: GND for onboard DO power section supply / GND PWM LED (red) is on at short circuit respectively overload
10	DC 24V	I	green	4L+: DC 24V for onboard DO power section supply

5.8.3 Proceeding

Hardware configuration

In the Siemens SIMATIC Manager the following steps should be executed:

1. Perform a hardware configuration for the CPU. ↳ *Chapter 4.4 'Hardware configuration - CPU' on page 58*
2. Double-click the counter sub module of the CPU 314C-2 PN/DP.
   ⇨ The dialog *'Properties'* is opened.
3. As soon as you select the operating mode for the corresponding channel, a dialog box with default values for this counter mode is created and shown. Select for the corresponding channel the operating mode *'Pulse width modulation - PWM'*.
   ↳ *Chapter 5.6.5 'Counter operating modes' on page 104*
4. Perform the required parameter settings.
5. Safe your project with *'Station ➔ Safe and compile'*.
6. Transfer your project to your CPU.

User program

- The SFB 49 should cyclically be called (e.g. OB 1) for controlling the pulse width modulation.
- The SFB is to be called with the corresponding instance DB. Here the parameters of the SFB are stored.

5.8.4 Parametrization

5.8.4.1 Address assignment

Sub module	Input address	Access	Assignment
Counter	816	DINT	Channel 0: Counter value / Frequency value
	820	DINT	Channel 1: Counter value / Frequency value
	824	DINT	Channel 2: Counter value / Frequency value
	828	DINT	Channel 3: Counter value / Frequency value

Sub module	Output address	Access	Assignment
Counter	816	DWORD	reserved
	820	DWORD	reserved
	824	DWORD	reserved
	828	DWORD	reserved

5.8.4.2 Operating mode per channel

Parameter hardware configuration

Select via *'Channel'* the channel select via *'Operating'* the operating mode. The following operating modes are supported:

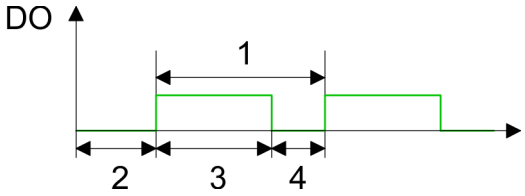
- Not parameterized: Channel is deactivated
- *Chapter 5.6.5.1 'Count continuously' on page 104*
- *Chapter 5.6.5.2 'Count once' on page 105*
- *Chapter 5.6.5.3 'Count periodically' on page 108*
- *Chapter 5.7 'Frequency measurement' on page 117*
- *Chapter 5.8 'Pulse width modulation - PWM' on page 123*

Depending on the selected operating mode default values are loaded and shown in an additional register.

5.8.4.3 Pulse width modulation

Parameter hardware configuration

Default values and structure of this dialog box depend on the selected *'Operating mode'*. The following parameters are supported:



1 Period
2 On-delay
3 Pulse duration
4 Pulse pause

Parameter overview

Operating parameters	Description	Assignment
Output format	Here specify the range of values for the output. The CPU hereby determines the pulse duration: ■ Per mil – Output value is within 0 ... 1000 – Pulse duration = (Output value / 1000) x Period ■ S7 Analog value: – Output value is Siemens S7 analog value 0 ... 27648 – Pulse duration = (Output value / 27648) x Period	Per mil
Time base	Here you can set the time base, which will apply for resolution and range of values of the period duration, minimum pulse duration and on-delay. ■ 1ms: Die Time base is 1ms ■ 0.1ms: Time base is 0.1ms	0.1ms
On-delay	Enter here a value for the time to expire from the start of the output sequence to the output of the pulse. The pulse sequence is output at the output channel, on expiration of the on-delay. Range of values: 0 ... 65535 from this there are the following effective values: ■ Time base 1ms: 0 ... 65535ms ■ Time base 0.1ms: 0 ... 6553.5ms	0

Operating parameters	Description	Assignment
Period	With the period you define the length of the output sequence, which consists of pulse duration and pulse pause. Range of values: Time base 1ms: 1 ... 87ms Time base 0.1ms: 0.4 ... 87.0ms	20000
Minimum pulse duration	With the minimum pulse duration you can suppress short output pulses and short pulse pauses. All pulses or pauses, which are smaller than the minimum pulse duration, are suppressed. This allows you to filter very short pulses (spikes), which can not be recognized by the periphery. Range of values: Time base 1ms: 0 ... Period / 2 * 1ms Time base 0.1ms: 2 ... Period / 2 * 0.1ms	2

5.8.5 Status indication

Digital output	LED	Description
DO +0.0 ... DO +0.7	green	Digital output Q+0.0 ... 0.7 has "1" signal
		Digital output Q+0.0 ... 0.7 has "0" signal
DO +1.0 ... DO +1.3	green	Digital output Q+1.0 ... 1.3 has "1" signal
		Digital output Q+1.0 ... 1.3 has "0" signal

Power supply	LED	Description
1L+	green	DC 24V electronic section supply
		DC 24V electronic section supply not available
3L+	green	DC 24V power section supply inputs OK
		DC 24V power section supply inputs not available
4L+	green	DC 24V power section supply outputs OK
		DC 24V power section supply outputs not available
4M	red	Error, overload respectively short circuit on the outputs
		no error

5.9 Diagnostic and interrupt

5.9.1 Overview

Hardware interrupt

The parametrization allows you to define the following trigger for a hardware interrupt:

- Edge at an digital interrupt input
- Reaching the comparison value
- Overflow respectively at overrun upper counter limit
- Underflow respectively at underrun lower counter limit
- Opening the HW gate with open SW gate - except for counter 3
- Closing the HW gate with open SW gate - except for counter 3

Diagnostics interrupt

The VIPA specific parameters allow you to define the following trigger for a diagnostics interrupt ↬ *Chapter 4.8 'Setting VIPA specific CPU parameters' on page 66*:

- Hardware interrupt lost
- Error: 4L+ DC 24V DO power section supply
- Error: 3L+: DC 24V DI power section supply
- Short circuit overload: DO

5.9.2 Process interrupt

An interrupt for the corresponding channel operating mode can only be triggered if you have additionally parameterized 'Diagnostics+Process' at 'Select interrupt' of the 'Basic parameters'.

A process interrupt causes a call of the OB 40. Within the OB 40 you may find the logical basic address of the module that initialized the process interrupt by using the Local word 6. More detailed information about the initializing event is to find in the *local double word 8*. The assignment of *local double word 8* depends on the parameterized operating mode of each channel.

Local double word 8 of OB 40 at *Alarm Inputs*

Local byte	Bit 7...0
8	■ Bit 0: Edge at I+0.0 ■ Bit 1: Edge at I+0.1 ■ Bit 2: Edge at I+0.2 ■ Bit 3: Edge at I+0.3 ■ Bit 4: Edge at I+0.4 ■ Bit 5: Edge at I+0.5 ■ Bit 6: Edge at I+0.6 ■ Bit 7: Edge at I+0.7
9	■ Bit 0: Edge at I+1.0 ■ Bit 1: Edge at I+1.1 ■ Bit 2: Edge at I+1.2 ■ Bit 3: Edge at I+1.3 ■ Bit 4: Edge at I+1.4 ■ Bit 5: Edge at I+1.5 ■ Bit 6: Edge at I+1.6 ■ Bit 7: Edge at I+1.7
10...11	■ Bit 7 ... 0: reserved

Local double word 8 of OB 40 at *counter function*

Local byte	Bit 7...0
8	■ Bit 0: Edge at I+0.0 ■ Bit 1: Edge at I+0.1 ■ Bit 2: Edge at I+0.2 ■ Bit 3: Edge at I+0.3 ■ Bit 4: Edge at I+0.4 ■ Bit 5: Edge at I+0.5 ■ Bit 6: Edge at I+0.6 ■ Bit 7: Edge at I+0.7
9	■ Bit 0: Edge at I+1.0 ■ Bit 1: Edge at I+1.1 ■ Bit 2: Edge at I+1.2 ■ Bit 3: Edge at I+1.3 ■ Bit 4: Edge at I+1.4 ■ Bit 5: Edge at I+1.5 ■ Bit 6: Edge at I+1.6 ■ Bit 7: Edge at I+1.7

Local byte	Bit 7...0
10	■ Bit 1, 0: reserved ■ Bit 2: Over-/underflow/end value counter 0 ■ Bit 3: Counter 0 reached comparison value ■ Bit 5, 4: reserved ■ Bit 6: Over-/underflow/ end value counter 1 ■ Bit 7: Counter 1 reached comparison value
11	■ Bit 1, 0: reserved ■ Bit 2: Over-/underflow/end value counter 2 ■ Bit 3: Counter 2 reached comparison value ■ Bit 4: Gate counter 3 open (activated) ■ Bit 5: Gate counter 3 closed ■ Bit 6: Over-/underflow/end value counter 3 ■ Bit 7: Counter 3 reached comparison value

Local double word 8 of OB 40 at *frequency measurement*

Local byte	Bit 7...0
8	■ Bit 0: Edge at I+0.0 ■ Bit 1: Edge at I+0.1 ■ Bit 2: Edge at I+0.2 ■ Bit 3: Edge at I+0.3 ■ Bit 4: Edge at I+0.4 ■ Bit 5: Edge at I+0.5 ■ Bit 6: Edge at I+0.6 ■ Bit 7: Edge at I+0.7
9	■ Bit 0: Edge at I+1.0 ■ Bit 1: Edge at I+1.1 ■ Bit 2: Edge at I+1.2 ■ Bit 3: Edge at I+1.3 ■ Bit 4: Edge at I+1.4 ■ Bit 5: Edge at I+1.5 ■ Bit 6: Edge at I+1.6 ■ Bit 7: Edge at I+1.7
10	■ Bit 0: End of measurement channel 0 (end of the integration time) ■ Bit 3 ... 1: reserved ■ Bit 4: End of measurement channel 1 (end of the integration time) ■ Bit 7 ... 5: reserved
11	■ Bit 0: End of measurement channel 2 (end of the integration time) ■ Bit 3 ... 1: reserved ■ Bit 4: End of measurement channel 3 (end of the integration time) ■ Bit 7 ... 5: reserved

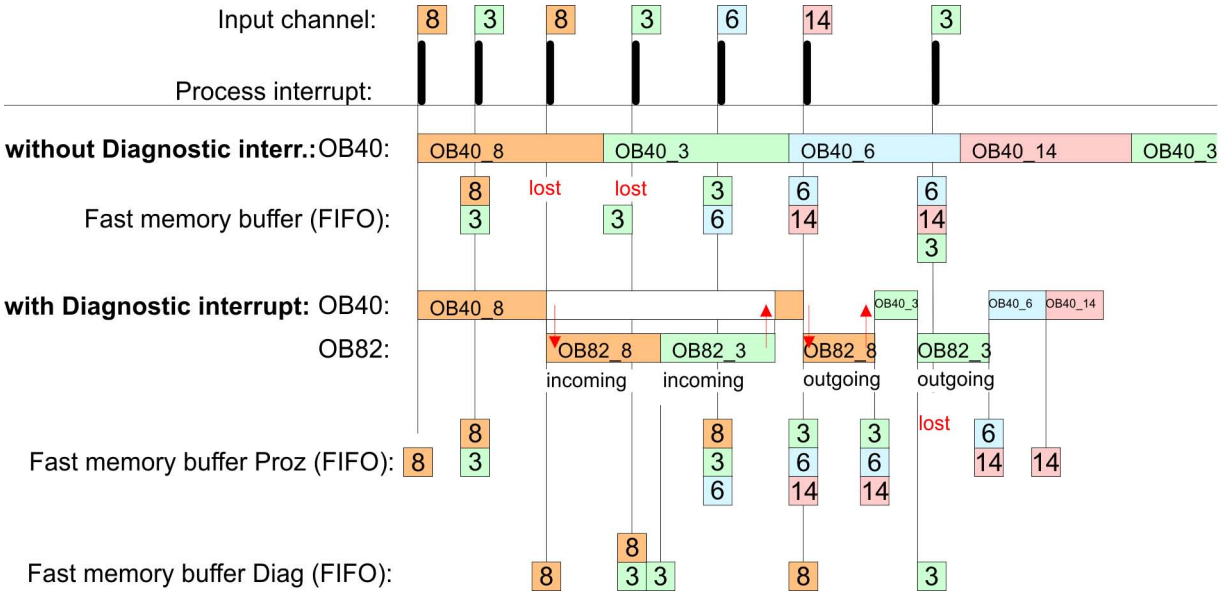
5.9.3 Diagnostic interrupt

Function

An interrupt for the corresponding channel operating mode can only be triggered if you have additionally parameterized 'Diagnostics+Process' at 'Select interrupt' of the 'Basic parameters'.

Via the parameterization (record set 7Fh) you may activate a global diagnostic interrupt for the module. A diagnostic interrupt occurs when during a process interrupt execution in OB 40 another process interrupt is thrown for the same event. The initialization of a diagnostic interrupt interrupts the recent process interrupt execution in OB 40 and branches in OB 82 to diagnostic interrupt processing$_{incoming}$. If during the diagnostic interrupt processing other events are occurring at other channels that may also cause a process res. diagnostic interrupt, these are interim stored. After the end of the diagnostic interrupt processing at first all interim stored diagnostic interrupts are processed in the sequence of their occurrence and then all process interrupts. If a channel where currently a diagnostic interrupt$_{incoming}$ is processed res. interim stored initializes further process interrupts, these get lost. When a process interrupt for which a diagnostic interrupt$_{incoming}$ has been released is ready, the diagnostic interrupt processing is called again as diagnostic interrupt$_{outgoing}$. All events of a channel between diagnostic interrupt$_{incoming}$ and diagnostic interrupt$_{outgoing}$ are not stored and get lost. Within this time window (1. diagnostic interrupt$_{incoming}$ until last diagnostic interrupt$_{outgoing}$) the SF-LED of the CPU is on. Additionally for every diagnostic interrupt$_{incoming/outgoing}$ an entry in the diagnostic buffer of the CPU occurs.

Example:



Diagnostic interrupt processing

Every OB 82 call causes an entry in the diagnostic buffer of the CPU containing error cause and module address. By using the SFC 59 you may read the diagnostic bytes. At de-activated diagnostic interrupt you have access to the last recent diagnostic event. If you've activated the diagnostic function in your hardware configuration, the contents of record set 0 are already in the local double word 8 when calling the OB 82. The SFC 59 allows you to also read the record set 1 that contains additional information. After leaving the OB 82 a clear assignment of the data to the last diagnostic interrupt is not longer possible. The record sets of the diagnostic range have the following structure:

Record set 0 Diagnostic$_{incoming}$

Byte	Bit 7...0
0	■ Bit 0: set at module failure – Counter/Frequency measurement: Process interrupt lost – Digital input: Process interrupt lost – Missing power supply DI or DO – Digital output: short circuit/overload ■ Bit 1: set at internal error – Missing power supply DI or DO – Digital output: short circuit/overload ■ Bit 2: set at external error ■ Bit 3: set at channel error ■ Bit 4: set at missing external power supply ■ Bit 7 ... 5: 0 (fix)
1	■ Bit 3 ... 0: Module class – 1111b: Digital ■ Bit 4: Channel information present – Counter/Frequency measurement: Process interrupt lost – Digital input: Process interrupt lost – Missing power supply DI or DO – Digital output: short circuit/overload ■ Bit 7 ... 5: 0 (fix)
2	■ Bit 3 ... 0: 0 (fix) ■ Bit 4: set at missing internal power supply – Missing power supply DI or DO ■ Bit 7 ... 5: 0 (fix)
3	■ Bit 5 ... 0: 0 (fix) ■ Bit 6: Process interrupt lost ■ Bit 7: 0 (fix)

Record set 0 Diagnostic$_{outgoing}$

After the removing error a diagnostic message$_{outgoing}$ takes place if the diagnostic interrupt release is still active.

Byte	Bit 7...0
0	■ Bit 0: set at module failure – Counter/Frequency measurement: Process interrupt lost – Digital input: Process interrupt lost – Missing power supply DI or DO – Digital output: short circuit/overload ■ Bit 1: set at internal error – Missing power supply DI or DO – Digital output: short circuit/overload ■ Bit 2: set at external error ■ Bit 3: set at channel error ■ Bit 4: set at missing external power supply ■ Bit 7 ... 5: 0 (fix)
1	■ Bit 3 ... 0: Module class – 1111b: Digital ■ Bit 4: Channel information present – Counter/Frequency measurement: Process interrupt lost – Digital input: Process interrupt lost – Missing power supply DI or DO – Digital output: short circuit/overload ■ Bit 7 ... 5: 0 (fix)
2	■ Bit 3 ... 0: 0 (fix) ■ Bit 4: set at missing internal power supply – Missing power supply DI or DO ■ Bit 7 ... 5: 0 (fix)
3	■ Bit 7 ... 0: 0 (fix)

The record set 0 of the alarm interrupts, counter function, frequency measurement and pulse width modulation has the same structure. There are differences in the structure of record set 1.

Diagnostic record set 1 at *Alarm Inputs*

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data. The diagnostic bytes have the following assignment:

Byte	Bit 7...0
0 ... 3	Content record set 0 *'Record set 0 Diagnostic$_{incoming}$' on page 132*
4	■ Bit 6 ... 0: Channel type (here 70h) – 70h: Digital input ■ Bit 7: More channel types present – 0: no – 1: yes
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 08h)

Byte	Bit 7...0
7	■ Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) ■ Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) ■ Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) ■ Bit 3: Error in channel group 2 (I+1.4 ... I+1.7) ■ Bit 7 ... 4: reserved
8	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+0.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.3 ■ Bit 7: 0 (fix)
9	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix)
10	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.3 ■ Bit 7: 0 (fix)
11	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix)
12...15	■ Bit 7 ... 0: reserved

Diagnostic record set 1 at *counter function*

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data. The diagnostic bytes have the following assignment:

Byte	Bit 7...0
0 ... 3	Content record set 0 *'Record set 0 Diagnostic$_{incoming}$' on page 132*
4	■ Bit 6 ... 0: Channel type (here 70h) – 70h: Digital input – 71h: Analog input – 72h: Digital output – 73h: Analog output – 74h: Analog input/output ■ Bit 7: More channel types present – 0: no – 1: yes
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 08h)
7	■ Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) ■ Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) ■ Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) ■ Bit 3: Error in channel group 3 (I+1.4 ... I+1.7) ■ Bit 4: Error in channel group 4 (counter 0) ■ Bit 5: Error in channel group 5 (counter 1) ■ Bit 6: Error in channel group 6 (counter 2) ■ Bit 7: Error in channel group 7 (counter 3)
8	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+0.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.3 ■ Bit 7: 0 (fix)
9	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix)
10	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.3 ■ Bit 7: 0 (fix)

Byte	Bit 7...0
11	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix)
12	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 3 ... 0: reserved ■ Bit 4: ... over-/underflow/end value counter 0 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 0 reached comparison value ■ Bit 7: 0 (fix)
13	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 3 ... 0: reserved ■ Bit 4: ... over-/underflow/end value counter 1 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 1 reached comparison value ■ Bit 7: 0 (fix)
14	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 3 ... 0: reserved ■ Bit 4: ... over-/underflow/end value counter 2 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 2 reached comparison value ■ Bit 7: 0 (fix)
15	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... Gate counter 3 closed ■ Bit 1: 0 (fix) ■ Bit 2: ... Gate counter 3 opened ■ Bit 4: ... over-/underflow/end value counter 3 ■ Bit 5: 0 (fix) ■ Bit 6: ... counter 3 reached comparison value ■ Bit 7: 0 (fix)

Diagnostic Record set 1 at *frequency measurement*

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data. The diagnostic bytes have the following assignment:

Byte	Bit 7...0
0 ... 3	Content record set 0 *'Record set 0 Diagnostic$_{incoming}$' on page 132*
4	■ Bit 6 ... 0: Channel type (here 70h) – 70h: Digital input – 71h: Analog input – 72h: Digital output – 73h: Analog output – 74h: Analog input/output ■ Bit 7: More channel types present – 0: no – 1: yes
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 08h)
7	■ Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) ■ Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) ■ Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) ■ Bit 3: Error in channel group 3 (I+1.4 ... I+1.7) ■ Bit 4: Error in channel group 4 (Frequency meter 0) ■ Bit 5: Error in channel group 5 (Frequency meter 1) ■ Bit 6: Error in channel group 6 (Frequency meter 2) ■ Bit 7: Error in channel group 7 (Frequency meter 3)
8	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+0.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.3 ■ Bit 7: 0 (fix)
9	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix)
10	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.3 ■ Bit 7: 0 (fix)

Byte	Bit 7...0
11	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix)
12	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: End of measurement channel 0 (End of integration time) ■ Bit 7 ... 1: 0 (fix)
13	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: End of measurement channel 1 (End of integration time) ■ Bit 7 ... 1: 0 (fix)
14	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: End of measurement channel 2 (End of integration time) ■ Bit 7 ... 1: 0 (fix)
15	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: End of measurement channel 3 (End of integration time) ■ Bit 7 ... 1: 0 (fix)

Diagnostic record set 1 at *pulse width modulation*

The record set 1 contains the 4byte of the record set 0 and additionally 12byte module specific diagnostic data. The diagnostic bytes have the following assignment:

Byte	Bit 7...0
0 ... 3	Content record set 0 *'Record set 0 Diagnostic$_{incoming}$' on page 132*
4	■ Bit 6 ... 0: Channel type (here 70h) – 70h: Digital input – 71h: Analog input – 72h: Digital output – 73h: Analog output – 74h: Analog input/output ■ Bit 7: More channel types present – 0: no – 1: yes
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 08h)

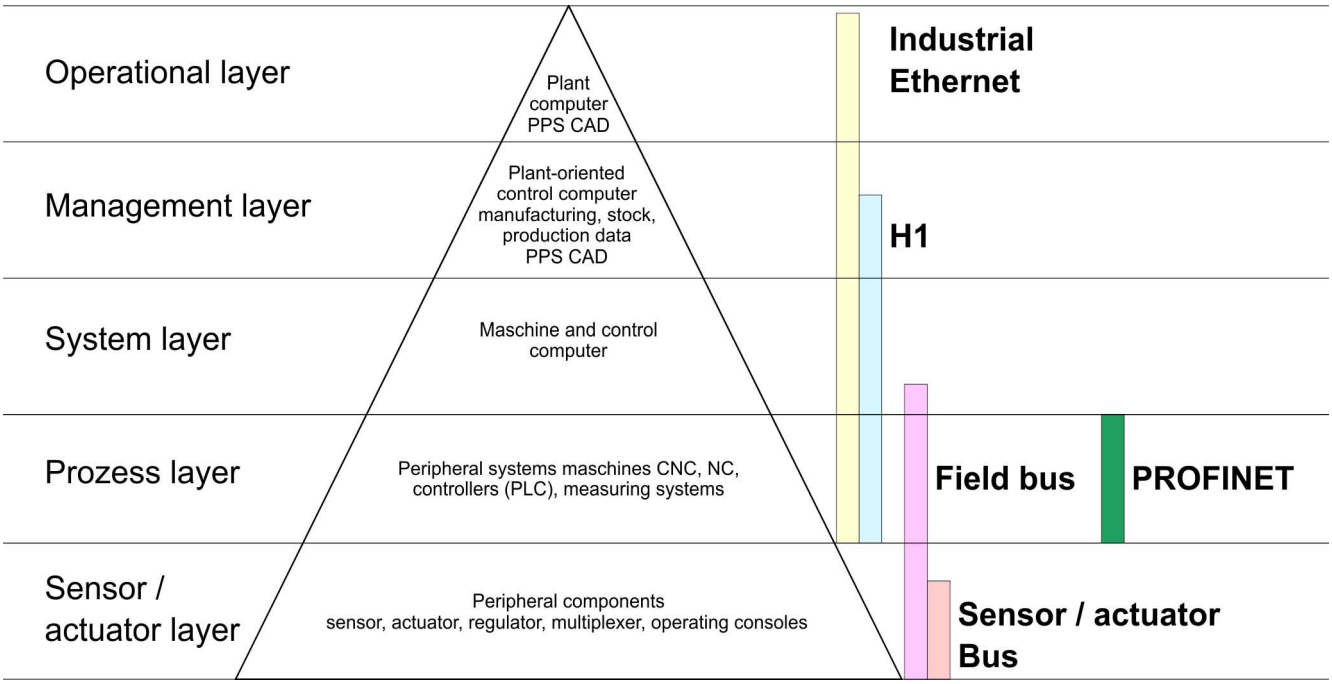
Byte	Bit 7...0
7	■ Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) ■ Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) ■ Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) ■ Bit 3: Error in channel group 3 (I+1.4 ... I+1.7) ■ Bit 7 ... 4: reserved
8	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+0.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.3 ■ Bit 7: 0 (fix)
9	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+0.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+0.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+0.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+0.7 ■ Bit 7: 0 (fix)
10	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+1.0 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.1 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.2 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.3 ■ Bit 7: 0 (fix)
11	Diagnostic interrupt due to "process interrupt lost" at... ■ Bit 0: ... input I+1.4 ■ Bit 1: 0 (fix) ■ Bit 2: ... input I+1.5 ■ Bit 3: 0 (fix) ■ Bit 4: ... input I+1.6 ■ Bit 5: 0 (fix) ■ Bit 6: ... input I+1.7 ■ Bit 7: 0 (fix)
12 ... 15	■ Bit 7... 0: reserved

6 Deployment PG/OP communication - productive

6.1 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.

6.2 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.

6.3 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.

6.4 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 computer communication is the so called ISO/OSI layer model, a model based upon seven layers with rules for the usage of hardware and software *Chapter 6.2 'Basics - ISO/OSI reference model' on page 141*

The following protocols are used:

- Siemens S7 connections
- Open communication
  - TCP native 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. Here NetPro from Siemens may be used.

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 FB/SFB VIPA 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" from VIPA.

Open communication

In the *'open communication'* the communication takes place via the user program by means of handling blocks. These blocks are also part of the Siemens SIMATIC Manager. You will find these in the *'Standard Library'* at *'Communication 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. 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 accord. to RFC 1006:*
    During data transmission, information on the length and the end of the message is also transmitted. 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 accord. 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. 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.

6.5 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 start-up of the CPU, the Ethernet PG/OP channel does not have an IP address.

Information about the assignment of IP address data to the Ethernet PG/OP channel may be found in *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*.

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
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.)

6.6 Fast introduction

Overview

At the first commissioning respectively after an overall reset with PowerON again of the CPU, the Ethernet PG/OP channel has no IP address. This can only be reached by its MAC address. By means of the MAC address, which is printed at the front as *'MAC PG/OP:...'*, you can assign IP address data. The assignment takes place directly via the hardware configuration of the Siemens SIMATIC Manager.

Steps of configuration

For the configuration of the Ethernet PG/OP channel for productive connections please follow the following approach:

- Hardware configuration - CPU
- Hardware configuration - Ethernet PG/OP channel
- Configure connections
  - Siemens S7 connections
    (Configuration via Siemens NetPro, communication via VIPA handling blocks)
  - Open communication
    (Configuration and communication happens by standard handling blocks)
- Transfer of the complete project to CPU

In the Siemens SIMATIC Manager, the CPU M13-CCF0000 from VIPA is to be configured as CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)!

The Ethernet PG/OP channel of the CPU M13-CCF0000 is always to be configured as CP343-1 (343-1EX30 V3.0) from Siemens at slot 4.

6.7 Hardware configuration

Overview

At the first commissioning respectively after an overall reset with PowerON again of the CPU, the Ethernet PG/OP channel has no IP address. This can only be reached by its MAC address. By means of the MAC address, which is printed at the front as *'MAC PG/OP:...'*, you can assign IP address data. The assignment takes place directly via the hardware configuration of the Siemens SIMATIC Manager.

- CPU
  Chapter 4.4 'Hardware configuration - CPU' on page 58
- Ethernet PG/OP channel
  Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61

6.8 Configure Siemens S7 connections

Overview

The project engineering of connections i.e. the "link-up" between stations happens in NetPro from Siemens. NetPro is a graphical user interface for the link-up of stations. A communication connection enables the program controlled communication between two participants at the Industrial Ethernet. The communication partners may here be part of the same project or - at multi projects - separated within related part projects. Communication connections to partners outside of a project are configured via the object "In unknown project" or via deputy objects like "Other stations" or Siemens "SIMATIC S5 Station". The communication is controlled by the user program with VIPA handling blocks. To use this blocks, configured communication connections are always necessary in the active station.

'Link-up stations' on page 148

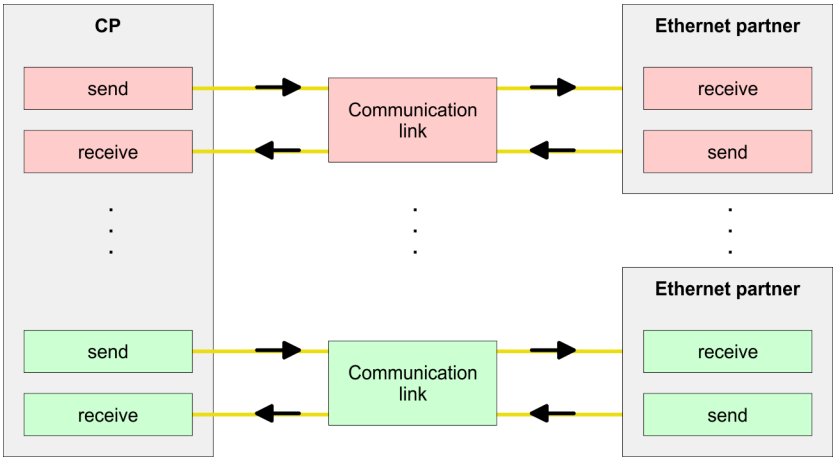
'Projecting connections' on page 149

'Siemens S7 connection - Communication functions' on page 151

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.



Requirements

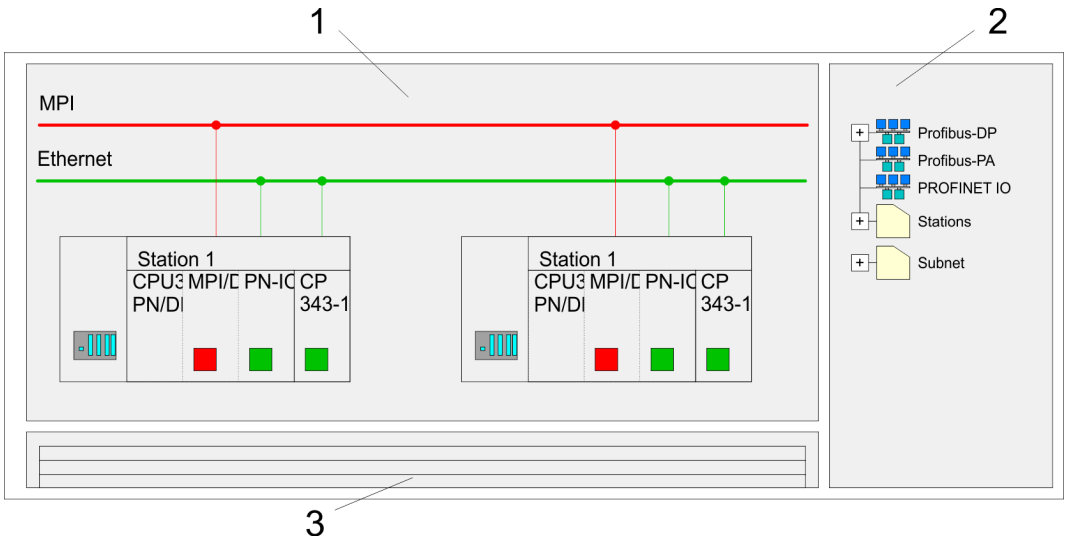
- Siemens SIMATIC Manager V 5.5 SP2 or higher and SIMATIC NET are installed.
- With the hardware configuration the according CP was assigned with IP address data by its properties.

Every station outside of the recent project must be configured as replacement objects like e.g. Siemens "SIMATIC S5" or "other station" or with the object "In unknown project". When creating a connection you may also choose the partner type "unspecified" and set the required remote parameter directly in the connection dialog.

Work environment of NetPro

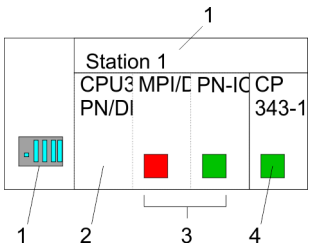
For the project engineering of connections, a thorough knowledge with NetPro from Siemens is required! The following passage only describes the basic usage of NetPro. More detailed information about NetPro is to be found in the according online manual res. documentation. Start NetPro by clicking on a "net" in the Siemens SIMATIC Manager or on "connections" within the CPU.

The environment of NetPro has the following structure:



1 *Graphic 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 *Net objects:* This area displays all available net objects in a directory view. By dragging a wanted object to the net view you may include further net objects and open them in the hardware configurator.
3 *Connection table:* The connection table lists all connections in a table. This list is only shown when you highlighted a connectable module like e.g. a CPU. You may insert new connections into this table with the according command.

PLC stations

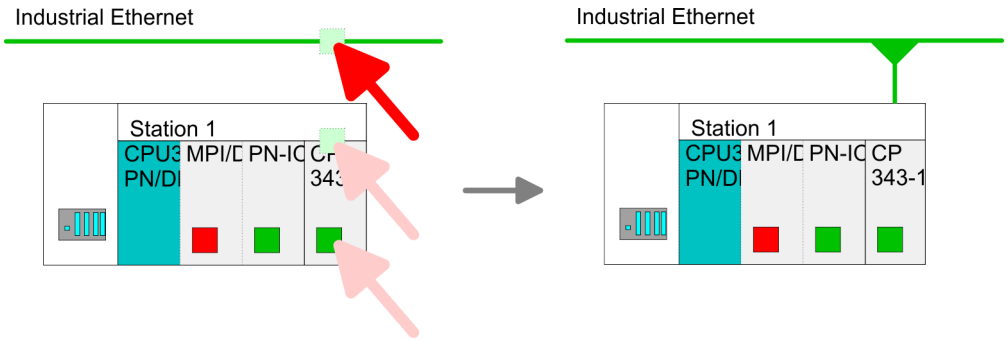


You receive the following graphical display for every PLC station and their component. By selecting the single components, the context menu offers you several functions:

1 *Station:* This includes a PLC station with rack, CPU and communication components. Via the context menu you may configure a station added from the net objects and its concerning components in the hardware configurator. After returning to NetPro, the new configured components are shown.
2 *CPU:* A click onto the CPU shows the connection table. The connection table shows all connections that are configured for the CPU.
3 *Internal communication components:* This displays the communication components that are available in your CPU. The PROFINET IO controller is to be configured by the PN-IO component.
4 *Ethernet PG/OP channel:* The internal Ethernet PG/OP channel must always be configured as external CP in the hardware configuration. This CP only serves the PG/OP communication. Configurable connections are not possible.

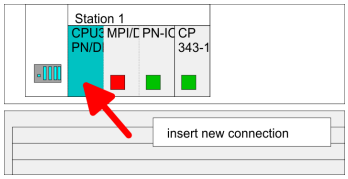
Link-up stations

NetPro offers you the option to link-up the communicating stations. You may link-up the stations via the properties in the hardware configuration or graphically via NetPro. For this you point the mouse on the coloured net mark of the according CP and drag and drop it to the net you want to link. Now the CP is linked up to the wanted net by means of a line.

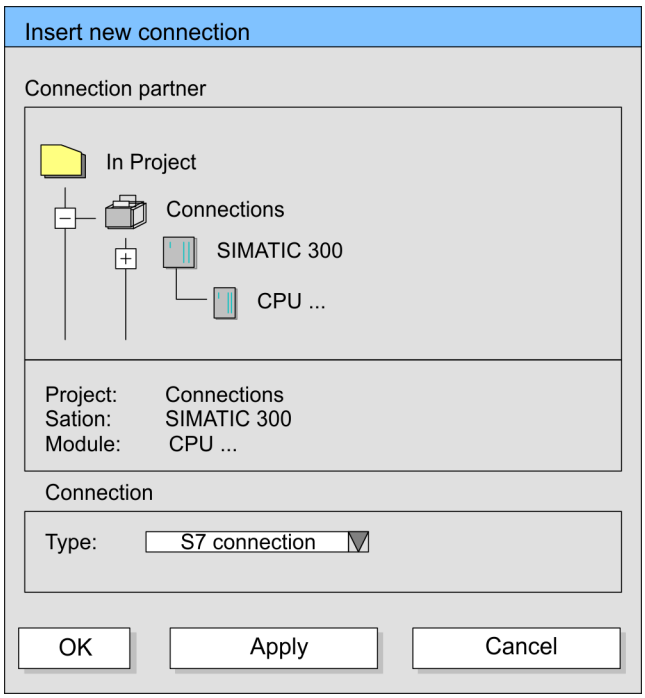


Projecting connections

Projecting connections



1. For the project engineering of connections, open the connection list by selecting the according CPU. Choose *Insert new connection* in the context menu:
   - *Connection partner (partner station)*
     A dialog window opens where you may choose the connection partner and the *connection type*.
   - *Specified connection partner*
     Each station configured in the Siemens SIMATIC Manager is listed in the table of connection partner. These stations are unique specified by an IP address and a subnet mask.
   - *Unspecified connection partner*
     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 connection name, which is to be used in the projects of both stations. Due to this allocation the connection remains *unspecified.*
2. Choose the connection partner and the type of connection and confirm with [OK].
   ⇨ If activated, a properties dialog for the according connection opens as link to your PLC user program.



3. After every connection was configured by this way, you may save and compile your project and exit NetPro.

Connection types

With this CPU exclusively Siemens S7 connection may be configured with Siemens NetPro.

Siemens S7 connection

- For data transfer with Siemens S7 connections the FB/SFB VIPA 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:

The following parameters define a connection end point:

Station A				Station B
remote TSAP	→	Siemens S7 connection	→	local TSAP
local TSAP	←		←	remote TSAP
ID A				ID B

Combination options with deployment of the FB/SFB VIPA handling blocks

Connection partner	Connection establishing	Connection
specified in NetPro (in the current project)	active/passive	specified
unspecified in NetPro (in the current project)	active	specified
	passive	unspecified
unspecified in NetPro (in the unknown project)	active/passive	specified (connection name in an other project)

In the following every relevant parameter of a Siemens S7 connection is described:

- *Local connection end point:*
  Here you may define how the connection is to be established. Since the Siemens SIMATIC Manager can identify the communication options by means of the end points, some options are already preset and may not be changed.
  - *Establish an active connection:*
    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 establish a connection. Here the job is to be made by the partner station.
  - *One-way:*
    If activated only one-way communication blocks like PUT and GET may be used for communication in the user program. Here the partner station acts as server, which neither may send active nor receive active
- Block parameters
  - *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 VIPA handling block.
  - *[Default]:*
    As soon as you click at [Default], the ID is reset to system generated ID.
- *Connection path:*
  In this part of the dialog window the connection path between the local and the partner station may be set. Depending on the linking of the modules the possible interfaces for communication are listed in a selection field.
  - *[Address details]:*
    With this button a dialog window is opened, which shows address information about the local and partner station. The parameters may also be changed.
  - *TSAP:*
    With Siemens S7 connections a TSAP is automatically generated of the connection resource (one-way/two-way) and state of place (rack/slot respectively system internal ID at PC stations).
  - *Connection resource:*
    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.

Siemens S7 connection - Communication functions

With the SPEED7 CPUs of VIPA there are two possibilities for the deployment of the communication functions:

- *Siemens S7-300 communication functions:*
  By integration of the function blocks FB 12 ... FB 15 from VIPA you may access the Siemens S7-300 communication functions.
- *Siemens S7-400 communication functions:*
  For the Siemens S7-400 communication functions the SFB 12 ... SFB 15 are to be used, which were integrated to the operating system of the CPU. Here copy the interface description of the SFBs from the standard library at system function block to the directory container, generate an instance data block for each call and call the SFB with the associated instance data block.

Function blocks

FB/SFB	Label	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. With this type of data transfer, more data can be transported between the communications partners than is possible with all other communication FBs/SFBs for configured S7 connections, namely 65534bytes.
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 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.

6.9 Configure Open Communication

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 Siemens SIMATIC Manager. You will find these in the "Standard Library" at "Communication Blocks". 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. First you have to establish a hardware configuration of the CPU and Ethernet PG/OP channel before you can use the handling blocks.

Hardware configuration:

- CPU
  ↳ *Chapter 4.4 'Hardware configuration - CPU' on page 58*
- Ethernet PG/OP channel
  ↳ *Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 61*

To specify the Ethernet PG/OP channel, the following values are defined in the UDT 65:

- *local_device_id*
  - 00h: Ethernet PG/OP channel of the CPU
- *next_staddr_len*
  - 01h: Ethernet PG/OP channel of the CPU
- *next_staddr*
  - 04h: Ethernet PG/OP channel of the CPU

UDTs

FB	Designation	Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006	Connectionless protocol: UDP according to 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

*) More information about the usage of these blocks may also be found in the manual "SPEED7 Operation List" from VIPA.

FBs

FB	Designation	Connection-oriented protocols: TCP native as per RFC 793, ISO on TCP as per RFC 1006	Connectionless protocol: UDP according to 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 67	TUSEND		Sending data
FB 68	TURCV		Receiving data

*) More information about the usage of these blocks may also be found in the manual "SPEED7 Operation List" from VIPA.

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.

7 *Option:* PtP communication

7.1 Fast introduction

General

For the PtP communication the use of the optionally available extension module EM M09 is required. The extension module provides interface X1: PtP (RS422/485) with fixed pin assignment. *Chapter 2.4 'Mounting' on page 13*

- PtP functionality
  - Using the PtP functionality the interface is allowed to connect via serial point-to-point connection to different source res. 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 instead SFCs

Please regard that the special VIPA SFCs are not shown in the CPU. Please use for programming tools e.g. Siemens SIMATIC Manager and TIA Portal the according FCs of the VIPA library.

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

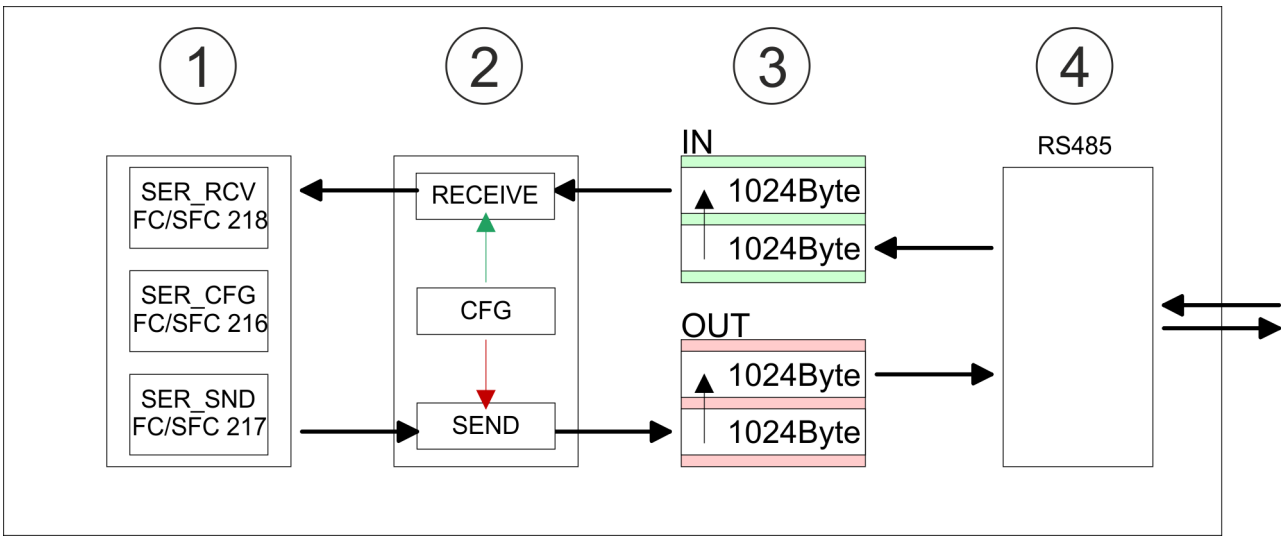
More information about the usage of these blocks may be found in the manual "SPEED7 Operation List" from VIPA.

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 PtP communication via extension module EM M09

X1 PtP (RS422/485)

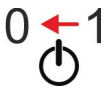
9pin SubD jack: (isolated)

Using the *PtP* functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source res. target systems.

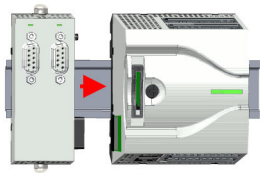
- Protocols:
  - ASCII
  - STX/ETX
  - 3964R
  - USS
  - Modbus master (ASCII, RTU)
- Serial bus connection
  - Full-duplex Four-wire operation (RS422)
  - Half-duplex Two-wire operation (RS485)
  - Data transfer rate: max 115 kBaud

Enable PtP functionality

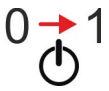
A hardware configuration to enable the PtP functionality is not necessary.



1. Turn off the power supply.



2. Mount the extension module. *Chapter 2.4 'Mounting' on page 13*
3. Establish a cable connection to the communication partner.

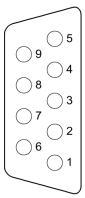


4. Switch on the power supply.

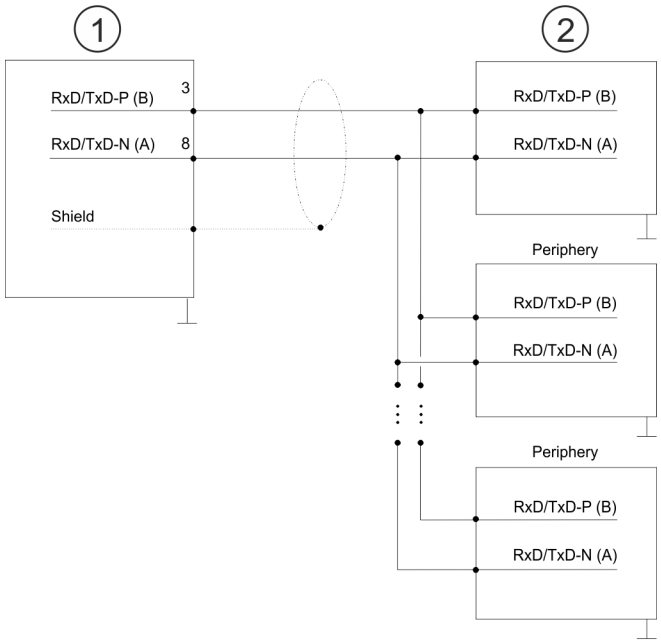
   ⇨ After a short boot time the interface X1 PtP is ready for PtP communication.

RS485 cabling with PROFIBUS cable

X1 PtP



① n. c.
② TxD-P (line B) - RS422
③ RxD-P (line B) - RS422
RxD/TxD-P (line B) - RS485
④ RTS
⑤ M5V
⑥ P5V
⑦ TxD-N (line A) - RS422
⑧ RxD-N (line A) RS422
RxD/TxD-N (line A) - RS485
⑨ n.c.

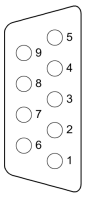


1 X1 PtP interface
2 Periphery

Never connect the cable shield and the M5V (pin 5) together, due to the compensation currents the interfaces could be destroyed!

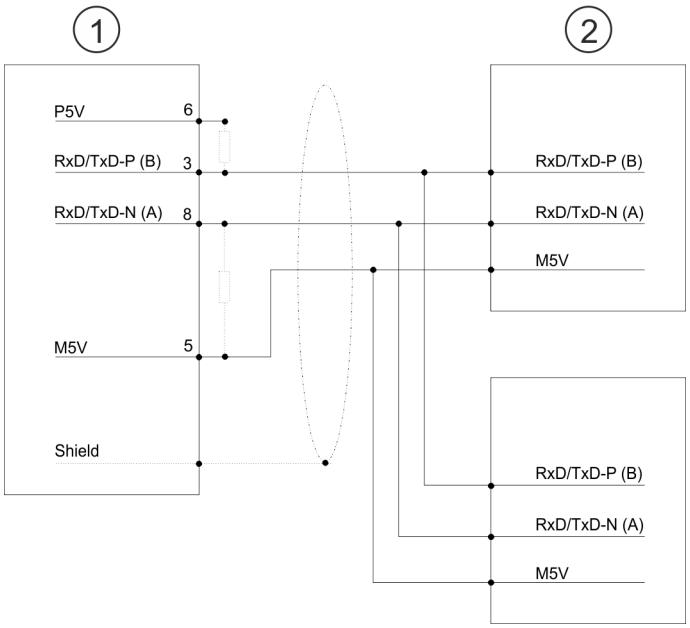
RS485 cabling with defined static voltage levels

X1 PtP



① n. c.
② TxD-P (line B) - RS422
③ RxD-P (line B) - RS422
RxD/TxD-P (line B) - RS485
④ RTS
⑤ M5V
⑥ P5V
⑦ TxD-N (line A) - RS422
⑧ RxD-N (line A) RS422
RxD/TxD-N (line A) - RS485
⑨ n.c.

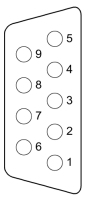
For isolated interfaces you have 5V (P5V) isolated at pin 6 and the corresponding ground (M5V) at pin 5. With this isolated voltage, you can assign defined static voltage levels to the signal lines and so ensure a low reflection level.



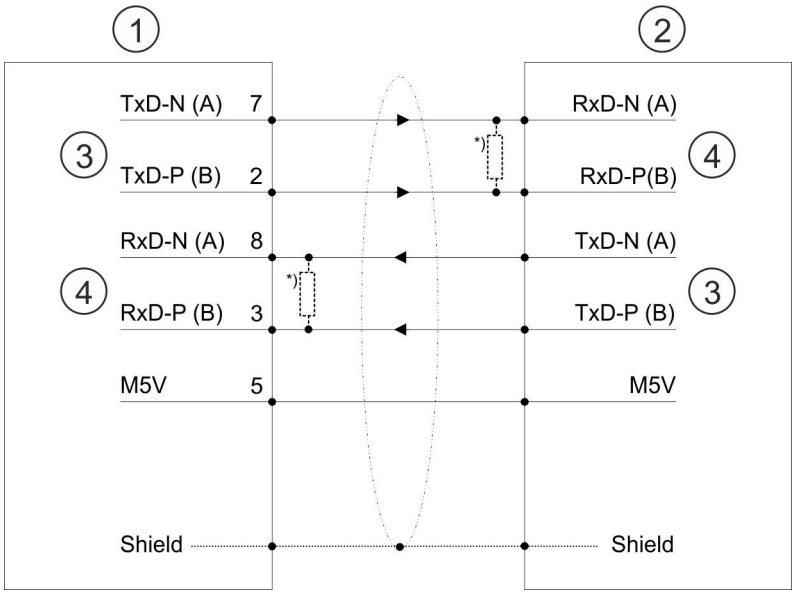
1 X1 PtP interface
2 Periphery

RS422 cabling

X1 PtP



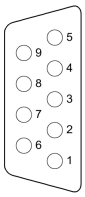
① n. c.
② TxD-P (line B) - RS422
③ RxD-P (line B) - RS422
RxD/TxD-P (line B) - RS485
④ RTS
⑤ M5V
⑥ P5V
⑦ TxD-N (line A) - RS422
⑧ RxD-N (line A) RS422
RxD/TxD-N (line A) - RS485
⑨ n.c.



1 X1 PtP interface
2 Periphery
3 Send
4 Receive
*) For line lengths >50m, you have to solder a terminating resistor of approx. 330Ω on the receiver side for interference-free data traffic.

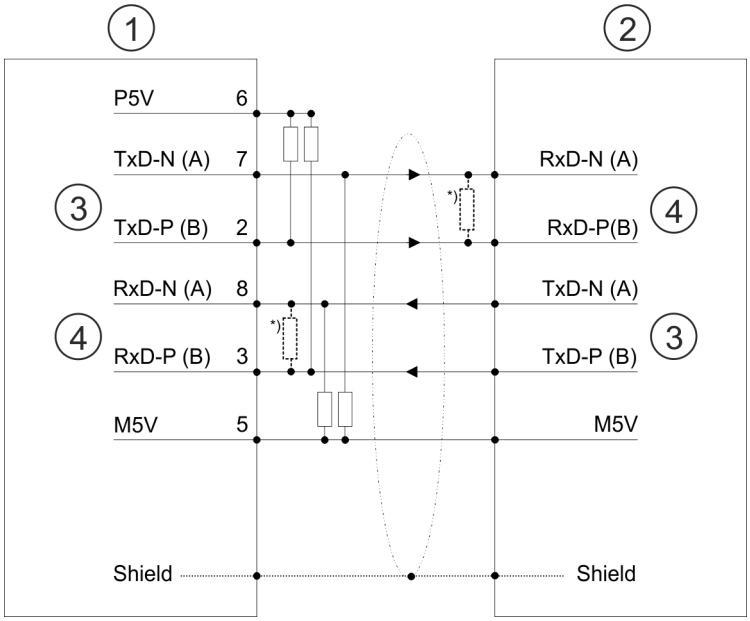
RS422 cabling with defined static voltage levels

X1 PtP



① n. c.
② TxD-P (line B) - RS422
③ RxD-P (line B) - RS422
RxD/TxD-P (line B) - RS485
④ RTS
⑤ M5V
⑥ P5V
⑦ TxD-N (line A) - RS422
⑧ RxD-N (line A) RS422
RxD/TxD-N (line A) - RS485
⑨ n.c.

For isolated interfaces you have 5V (P5V) isolated at pin 6 and the corresponding ground (M5V) at pin 5. With this isolated voltage, you can assign defined static voltage levels to the signal lines and so ensure a low reflection level.



1 X1 PtP interface
2 Periphery
3 Send
4 Receive
*) For line lengths >50m, you have to solder a terminating resistor of approx. 330Ω on the receiver side for interference-free data traffic.

Status indication

X1 PtP TxD	Description
◪ green flickers	Send activity
☐	No send activity

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" from VIPA.

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 parameterized 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. An according Receive_ASCII FB may be found within the VIPA library in the service area of www.vipa.com.

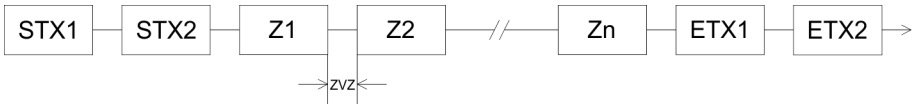
STX/ETX

STX/ETX is a simple protocol with start and end ID, where STX stands for **S**tart of **Tex**t and ETX for **E**nd of **Tex**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

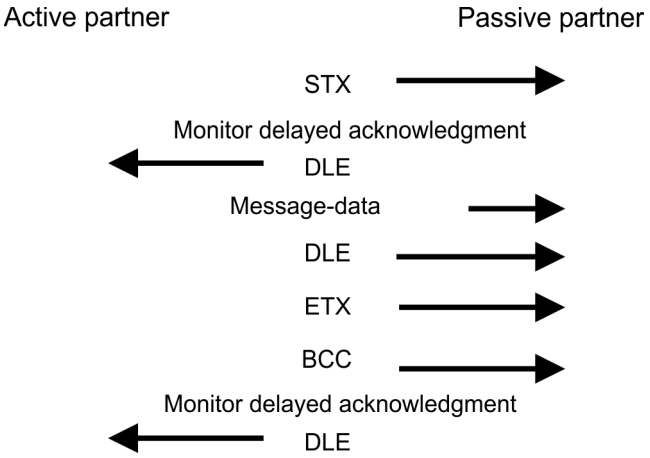
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free receipt.

The procedure employs the following control characters:

- STX: **S**tart of **Te**x**t**
- DLE: **D**ata **L**ink **E**scape
- ETX: **E**nd of **Te**x**t**
- BCC: **B**lock **C**heck **C**haracter
- NAK: **N**egative **Ac**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	

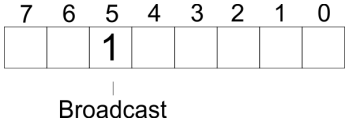
Slave master telegram

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



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

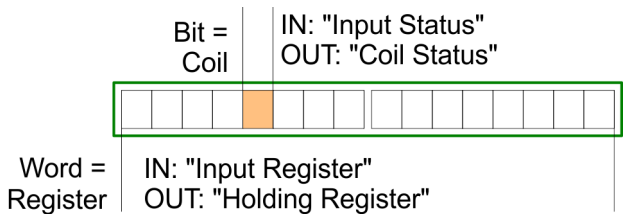
The following Modbus Protocols are supported by the RS485 interface:

- 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 from VIPA 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

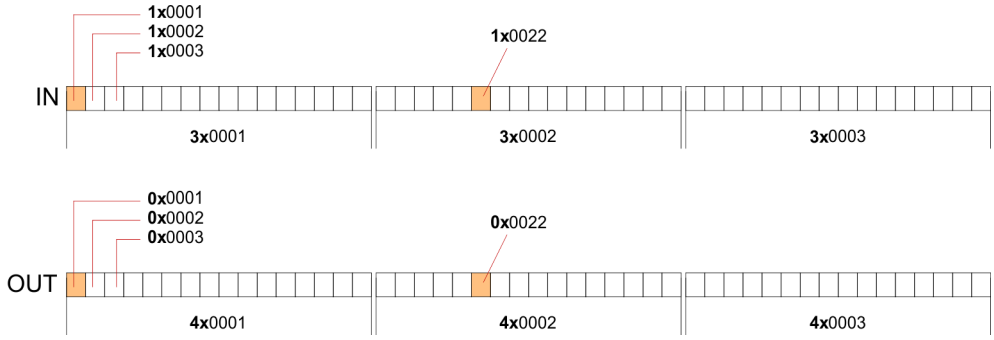
Access via function code 02h

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.

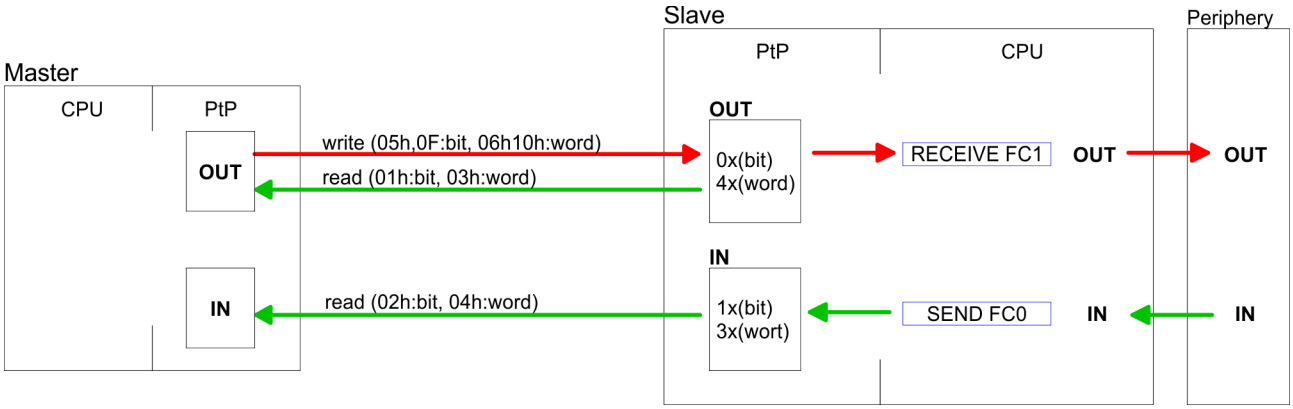
Overview

With the following Modbus function codes a Modbus master can access a Modbus slave:
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
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 *Option:* Deployment PROFIBUS communication

8.1 Fast introduction

Overview

For the PROFIBUS communication the use of the optionally available extension module EM M09 is required. The extension module provides interface X2: MPI(PB) with fixed pin assignment. ↳ *Chapter 2.4 'Mounting' on page 13* The PROFIBUS DP slave 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.

To switch the interface X2 MPI(PB) to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is activated. ↳ *Chapter 4.15 'Deployment storage media - VSD, VSC' on page 79*

Steps of configuration

The configuration of the PROFIBUS DP slave should be done with the following approach:

- **Activating bus functionality by means of a VSC**
- **Hardware configuration - CPU**
- **Use as DP slave**
  - With activating the bus functionality *'PROFIBUS DP slave'* by means of a VSC, the bus functionality *'PROFIBUS DP slave'* is unlocked.
- **Transfer of the entire project to the CPU**

With the Siemens SIMATIC Manager, the CPU M13-CCF0000 from VIPA is to be configured as

CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)

The PROFIBUS DP slave is to be configured and connected via the sub module X1 (MPI/DP).

8.2 PROFIBUS communication

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.

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 a server for communication services with the following characteristics:

- The PROFIBUS interface gets an "passive" PROFIBUS node, this means it is not involved in the token rotation.
- Via this interface you have PG/OP functions (programming, status request, control, test).
- The speed of the PG/OP functions is limited.
- Bus rotation time is not influenced.
- S7 routing is not possible.

8.3 PROFIBUS communication via extension module EM M09

X2 MPI(PB)

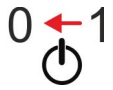
9pin SubD jack: (isolated)

The interface supports the following functionalities, which are switch able by an hardware configuration:

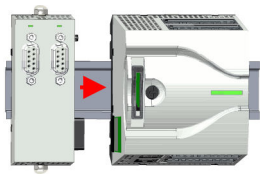
- MPI (default / after overall reset)
  The MPI interface serves for the connection between programming unit and CPU. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU. Standard setting is MPI address 2.
- PROFIBUS DP slave (option)
  The PROFIBUS slave functionality of this interface can be activated by configuring the sub module *'MPI/DP'* of the CPU in the hardware configuration.

Enable PROFIBUS functionality

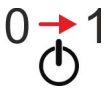
The activation of the PROFIBUS functionality of the extension module EM M09 happens with the following proceeding:



1. Turn off the power supply.



2. Mount the extension module. *Chapter 2.4 'Mounting' on page 13*

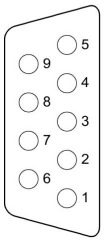


3. Switch on the power supply.

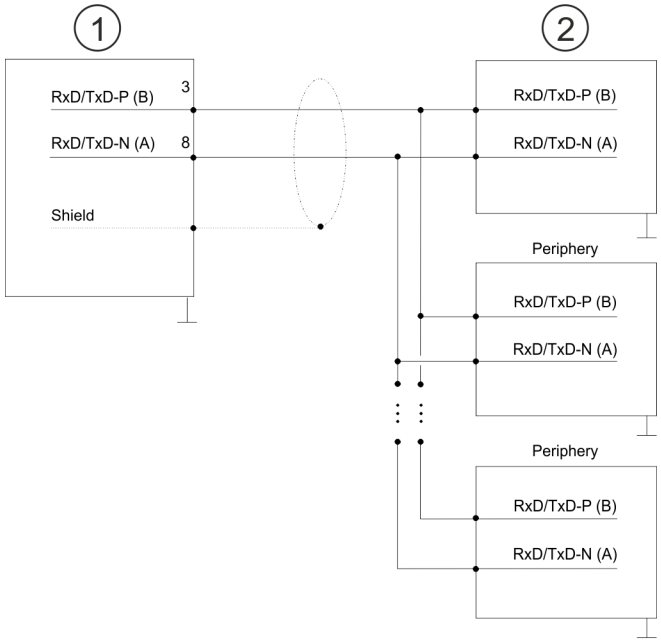
⇨ After a short boot time the interface X2 MPI(PB) is ready for MPI communication with the MPI address 2.

To switch the interface X2 MPI(PB) to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is activated. *Chapter 4.15 'Deployment storage media - VSD, VSC' on page 79*

X2 MPI(PB)



① n. c.
② n. c.
③ RxD/TxD-P (line B)
④ RTS
⑤ M5V
⑥ P5V
⑦ n. c.
⑧ RxD/TxD-N (line A)
⑨ n.c.



1 RS485 interface
2 Periphery

Never connect the cable shield and the M5V (pin 5) together, since the interfaces could be destroyed!

Status indication

X2 MPI(PB) DE	Description
green	■ Slave is in DE (data exchange). ■ Slave exchanges data with the master. ■ Slave is in RUN state
green blinking	■ Slave CPU is in state start-up. ■ Slave-CPU is without master.
	■ There is no power supply. ■ Slave has no configuration.

8.4 Deployment as PROFIBUS DP slave

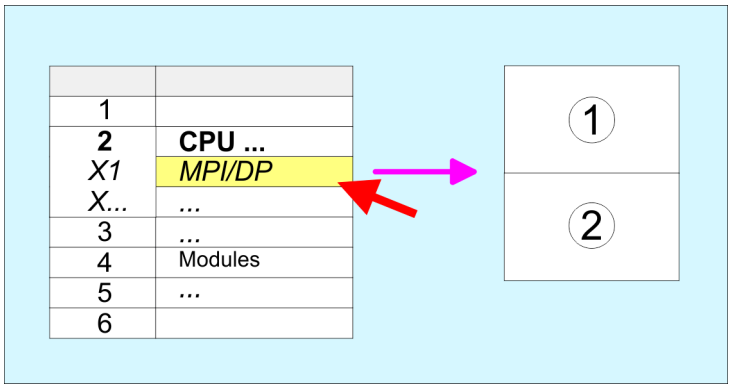
Fast introduction

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.

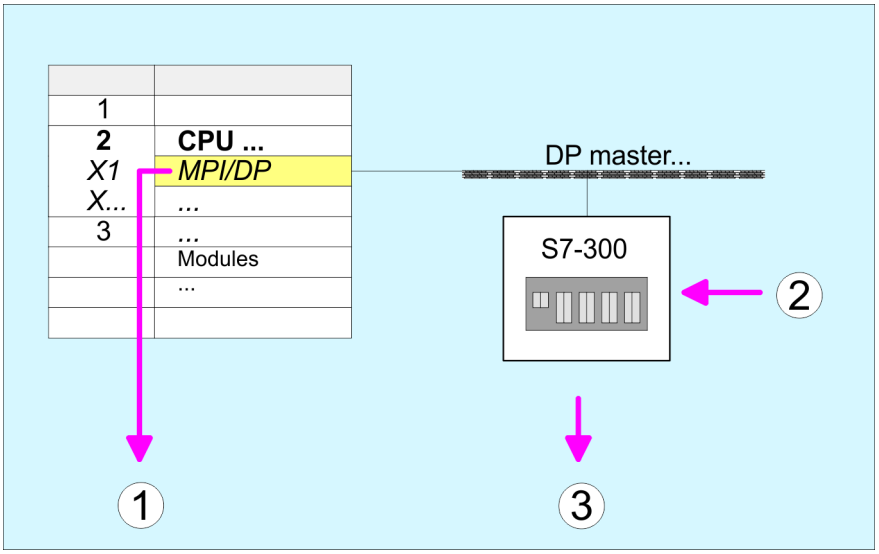


1 Standard bus: Object properties
Operating mode: DP slave
Connect: PROFIBUS
PROFIBUS address > 1
2 Configuration
Input area
Output area

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.



1 Standard bus: Object properties
Operating mode: DP master
PROFIBUS address > 1
2 Hardware catalog: CPU 31x
from *'Configured stations'*
3 DP master system: Object properties
Input area slave CPU = Output area master CPU
Output area slave CPU = Input area master CPU

8.5 PROFIBUS installation guidelines

PROFIBUS in general

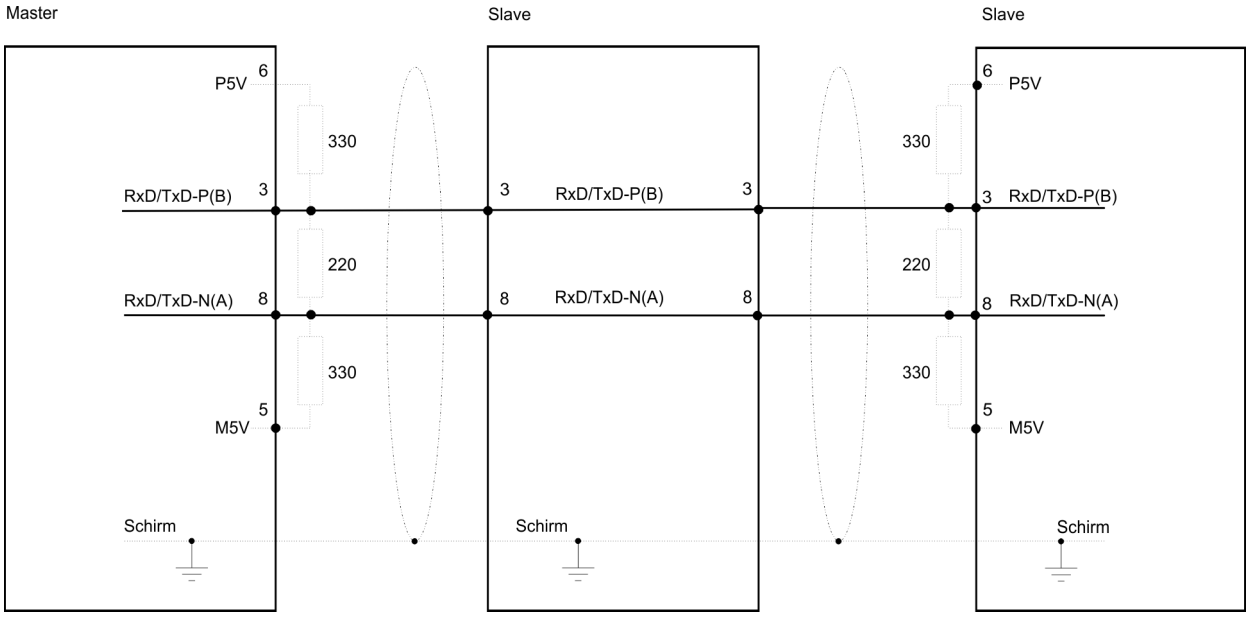
- A PROFIBUS DP network may only be built up in linear structure.
- PROFIBUS DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- PROFIBUS supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- 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

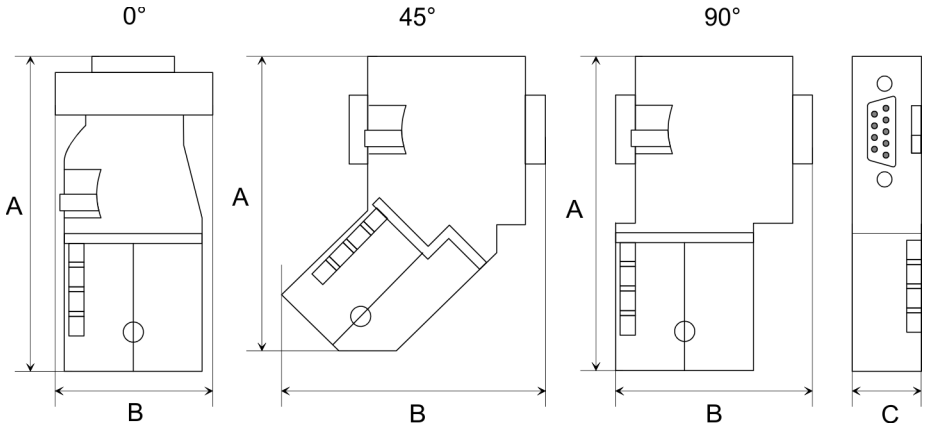
The following picture illustrates the terminating resistors of the respective start and end station.



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 VIPA. 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 Kabel order no: 2170222, 2170822, 2170322.

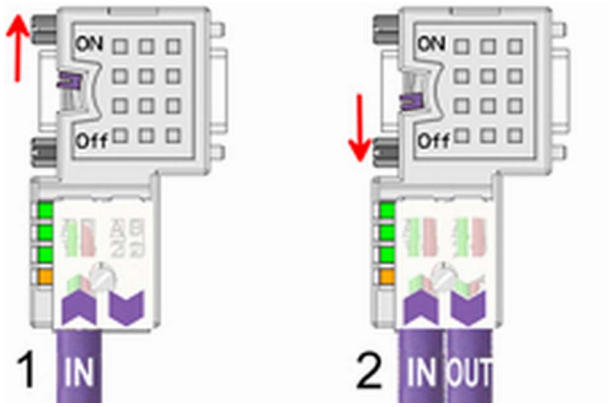
With the order no. 905-6AA00 VIPA offers the "EasyStrip" de-isolating tool that makes the connection of the EasyConn much easier.

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

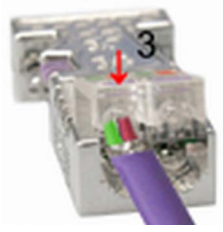
CAUTION!
The terminating resistor is only effective, if the connector is installed at a bus participant and the bus participant is connected to a power supply.

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!

8.6 Commissioning and Start-up behavior

Start-up on delivery

On delivery the CPU is overall reset. The PROFIBUS part is de-activated and the LED (serial port) is off after PowerON.

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 status bar. 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 green DE LED (serial port).

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 to the slaves by the master. Here the DE-LED is blinking.
- 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 to the slaves by the master. Here the DE LED (serial port) is on.
- 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.

9 Configuration with VIPA *SPEED7 Studio*

9.1 *SPEED7 Studio* - Overview

SPEED7 Studio - Working environment

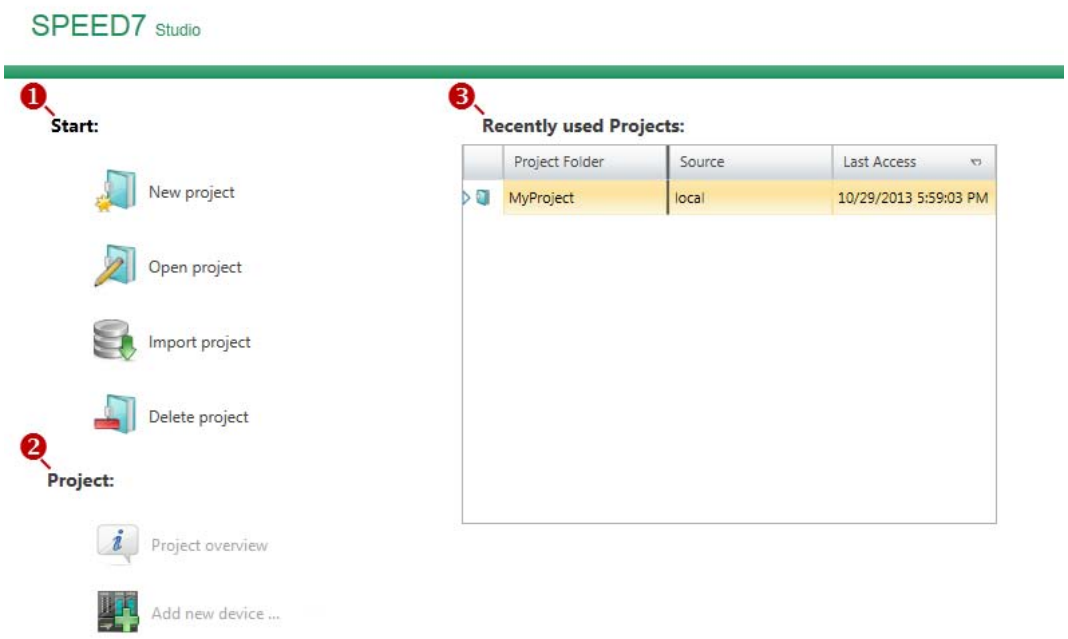
In this part the project engineering of the VIPA CPU in the VIPA *SPEED7 Studio* is shown. Here only the basic usage of the *SPEED7 Studio* together with a VIPA CPU is shown. Please note that software changes can not always be considered and it may thus be deviations to the description. In the *SPEED7 Studio* your VIPA 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 SPEED7 Studio.

Starting the *SPEED7 Studio*

→ Click at the button. You can find *SPEED7 Studio* in Windows Start at *'VIPA'*.

⇨ *SPEED7 Studio* is started. The *start page* is opened.



(1) Start — You can create a new project, open a saved project, or delete projects.
(2) Project — If a project is open, you can open the *'Project overview'* or add a new device.
(3) Last projects — Here recently opened projects are listed.

You can repeatedly run SPEED7 Studio in order to work with different projects. You can not open the same project in the various instances of SPEED7 Studio.

End *SPEED7 Studio*

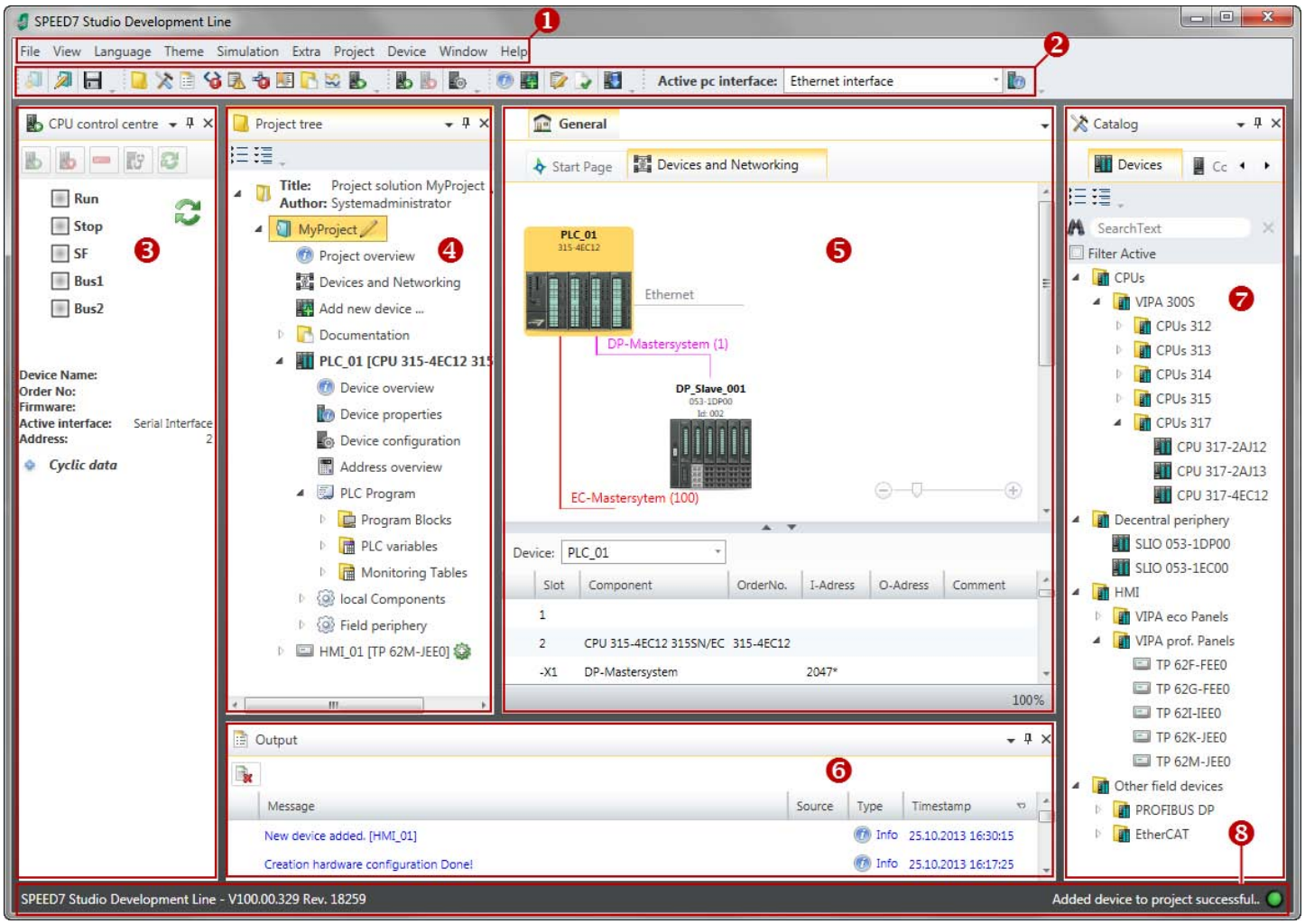
Select one of the following options if you want to end the program:

- **Main window:** Click on the Close button of the *SPEED7 Studio* program window.
- **Menu bar** Select *'File ➔ Exit'*.
- **Keyboard:** Press [Alt] + [F4].

After you have made changes to the project, a dialogue window opens where you can select whether to save or ignore the changes.

⇨ *SPEED7 Studio* is ended.

9.2 *SPEED7 Studio* - Work environment



(1) Menu bar
(2) Toolbar
(3) CPU control centre
(4) Project tree
(5) Area of operations
(6) Output range
(7) Catalog/properties
(8) Status line

You can show and hide additional windows and the arrangement and size of the windows can be adjusted.

(1) Menu bar

Most of the commands you need for working with *SPEED7 Studio* are provided in the menu bar. Further commands can be accessed via the context menus using the right mouse button, e.g. functions of a device in the project tree.

The menu commands *'Project'* and *'Device'* are only shown if a project is open. The menu commands *'Image'* is only shown if a HMI image is open.

You can use the menus with the mouse or the keyboard.

(2) Toolbar

Important commands you need for working with *SPEED7 Studio* are provided in the toolbar. More commands can be accessed via the toolbars and push buttons of different editors.

Some of the commands in the toolbar are only shown if a project is open.

(3) CPU control centre

In the CPU control centre, you can view the current mode and other control data and control the CPU.

(4) Project tree

Any project device and project data can be accessed via the project tree. The project tree includes any object which you have created in the project, e.g. devices, components, program blocks, HMI images. Here you can add or remove devices and components. Furthermore, you can open editors in order to edit settings, configurations, the control program and visualisation.

(5) Area of operations

Devices and project data can be edited in the area of operations. You can open different editors for this purpose. The register in the area of operations is divided into two register levels. You can switch through the editors in the area of operations via the tabs.

(6) Output range

Information on executed activities and background operations are displayed on the output range.

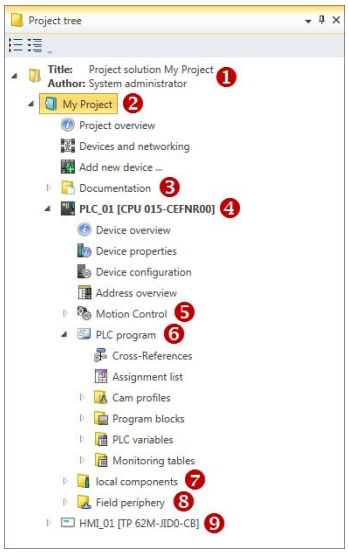
(7) Catalog/properties

Devices and components which you want to add to the project can be selected in the catalog. You can also select objects which you want to add to the PLC program or to HMI images.

(8) Status line

The version of *SPEED7 Studio* is displayed at the left edge of the status line. The progress bar for background operations and status messages is shown at the right edge. As long as there are no background operations, the status message created at last is shown.

9.2.1 Project tree



(1) Title and author
(2) Project
(3) Documentation
(4) PLC
(5) Motion Control
(6) PLC program
(7) Local components
(8) Field periphery
(9) HMI

In the project tree, you can access commands in order to add or delete objects, e.g. add/delete devices or add/delete blocks.

You can open editors via the project tree if you want to edit settings, configurations, the control program and visualisation.

Moreover, you can retrieve information, e.g. project overview, device properties or properties of the bus system.

Show project tree

If the project tree is not displayed, you must select either *'View ➔ Project tree'* or press [Strg]+[Shift]+[P].

Show projects in the project tree

In order to display the project in the project tree, you must create a new project or open a stored project.

It is not possible to edit several projects at the same time. It is possible to run *SPEED7 Studio* simultaneously several times on one PC if you want to use it for various projects.

Show/hide objects

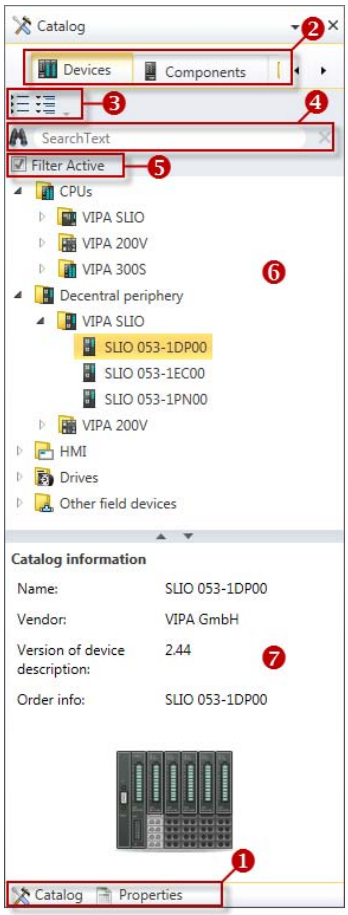
The objects in the project tree are arranged in a tree structure. You can show or hide objects:

- Hide all objects (*'Project ➔ Reduce project tree'*)
- Show all objects (*'Project ➔ Expand project tree'*)
- Hide slave objects / close folder
- Show slave objects / open folder

Recognise object state

Icons behind an object in the project tree provide indications of the object state.

9.2.2 Catalog



(1) Switching to another view
(2) Register
(3) Show/hide objects
(4) Search
(5) Filter
(6) Objects
(7) Catalog information

Devices and components which you want to add to the project can be selected in the catalog. You can also select objects which you want to add to the PLC program or to HMI images.

Show catalog:

If the catalog is not displayed, you must select either *'View ➔ Catalog'* or press [Strg]+[Shift]+[C].

(1) Switch to another view

If the properties are displayed instead of the catalog, you must click on *'Catalog'* at the lower screen edge.

(2) Register

Certain tabs are displayed in the catalog, depending on which editor window is opened in the foreground.

(3) Show/hide objects

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

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

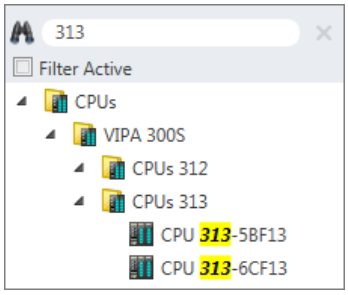
Hide all objects (*'Project ➔ Reduce catalog tree'*)

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

Hide slave objects / close folder

Show slave objects / open folder

(4) Search



You can search for certain objects in the catalog.

1. Enter a search text in the input field.
   ⇨ Only those objects are displayed in the catalog which contain the search text.
2. Click on ☒ to delete the search text.
   ⇨ All objects are displayed in the catalog.

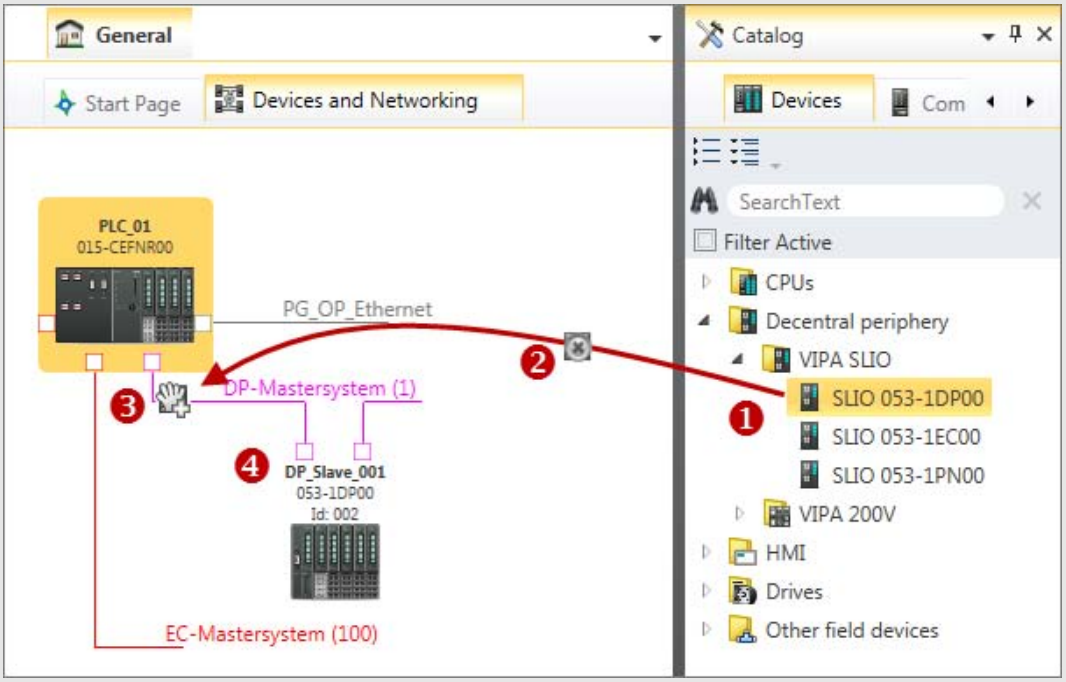
(5) Filter

With *'enabled'* Filter, only these modules are shown in the *Catalog* which are relevant for configuration

(6) Add object

→ Drag the desired object from the catalog to a suitable position.
⇨ The object is added.

Example



(1) Select the desired object (hold left mouse button down)
(2) Drag the object
(3) Drop the object at a suitable place (release the mouse button)
(4) The object is added

(7) Catalog information

The catalog information shows detailed information of the selected object, e.g. name, producer, version and order information.

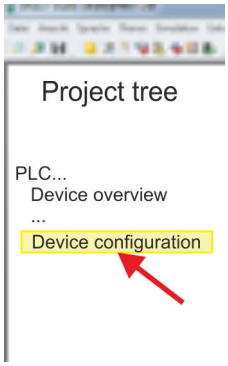
9.3 *SPEED7 Studio* - Hardware configuration - CPU

Precondition

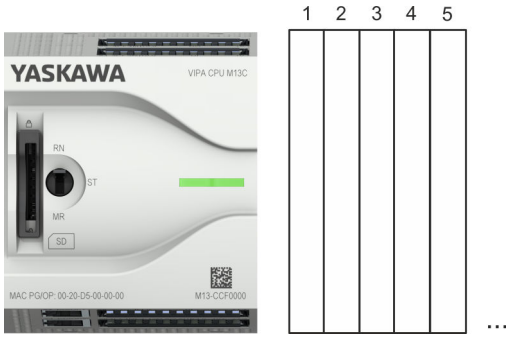
For project engineering a thorough knowledge of the SPEED7 Studio is required!

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 M13-CCF0000				
-X2	MPI interface				
-X3	PG_OP_Ethernet				
...	...			...	

9.4 *SPEED7 Studio* - Hardware configuration - Ethernet PG/OP channel

Overview

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 (X3/X4) is designed as switch. This enables PG/OP communication via the connections X3 and X4.
- The Ethernet PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.
- At the first commissioning respectively after a factory reset the Ethernet PG/OP channel has no IP address.

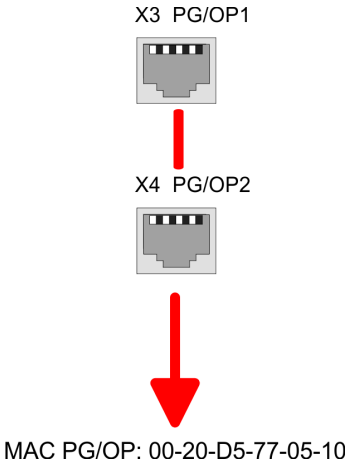
- For online access to the CPU via the Ethernet PG/OP channel, valid IP address parameters have to be assigned to this. This is called "initialization".
- This can be done with the *SPEED7 Studio*.

Assembly and commissioning

1. Install your System MICRO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the one of the Ethernet jacks (X3, X4) 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.

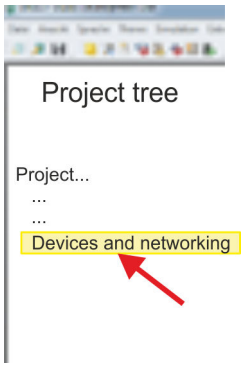
"Initialization"

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:

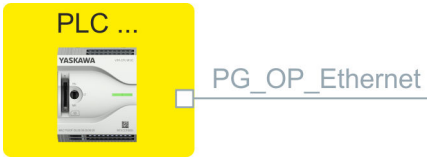


1. Ethernet PG/OP

   Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".
2. Start the *SPEED7 Studio* with your project.

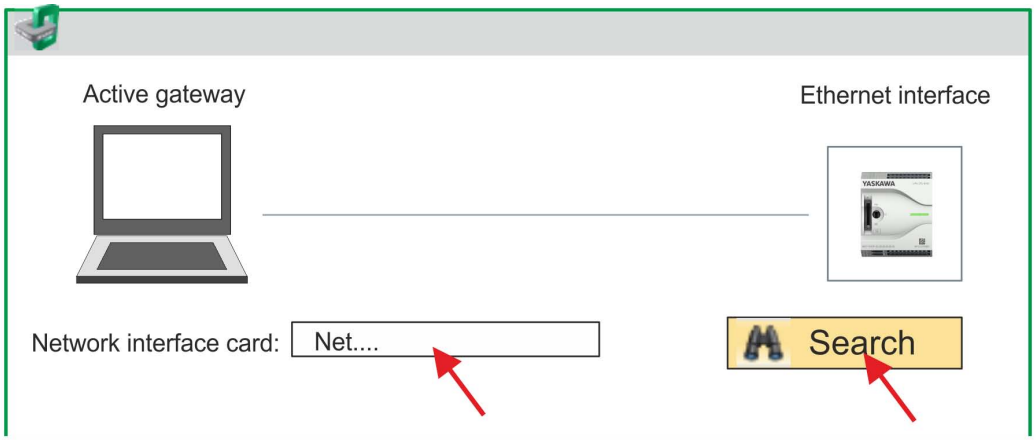


3. Click in the *Project tree* at *'Devices and networking'*.
   ⇨ You will get a graphical object view of your CPU.



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

5. Select *'Context menu ➔ Determine accessible partner'*.
   ⇨ A dialog window opens.



6. 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.

7.

	Devices...	IP...	MAC...	Device...	...	...
1		172.20. ..	00:20: ...	VIPA ...		
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: ...".

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

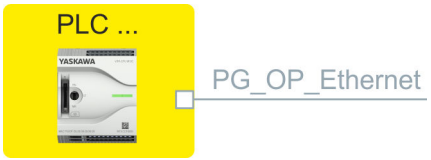
9. Click at *'Set IP address'*.
   ⇨ The IP address is transferred to the module and the list is refreshed. Directly after the assignment the Ethernet PG/OP channel is online reachable using the set IP address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration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

10. 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].

   ⇨ 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.

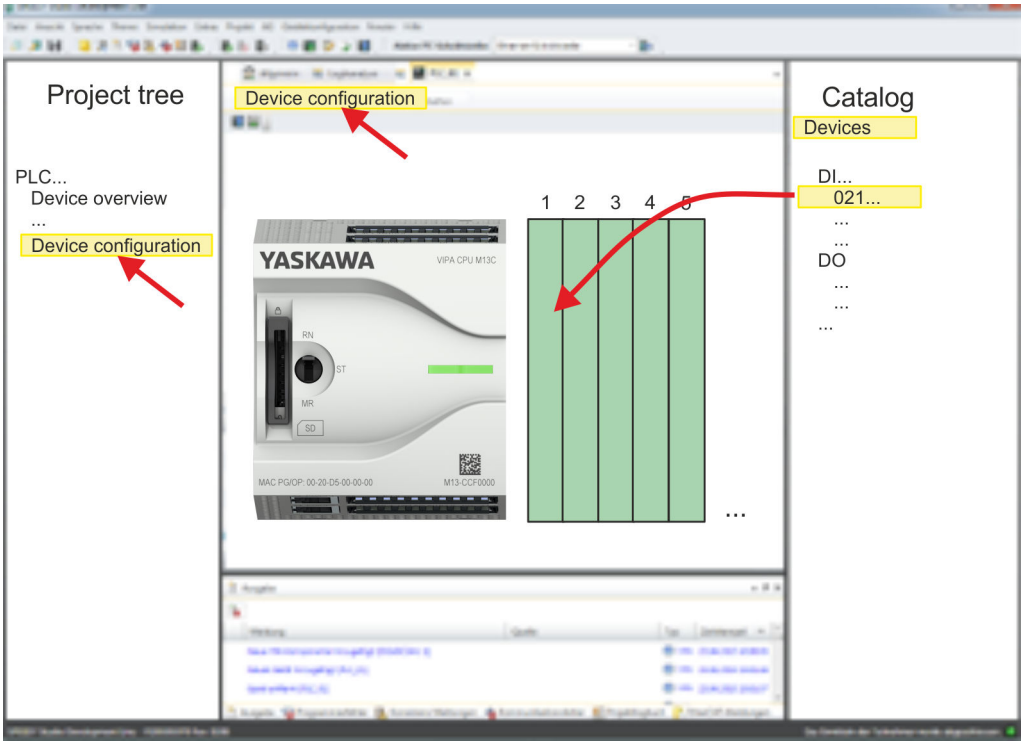
Local components

Slot	Module	...	...	...IP address	...
0	CPU M13-CCF0000			...	
-X2	MPI interface			...	
-X3	PG_OP_Ethernet			172.20.120.40	
...	...			...	

9.5 *SPEED7 Studio* - 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'* yourSystem MICRO 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 parametrize. 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.

9.6 Deployment I/O periphery

9.6.1 Overview

Project engineering and parametrization

- On this CPU the connectors for digital respectively analog signal and *Technological functions* are combined in a one casing.
- Die Project engineering happens in the VIPA *SPEED7 Studio* as CPU M13-CCF0000.
- For parametrization of the digital I/O periphery and the *technological functions* the corresponding sub modules of the CPUM13-CCF0000 are to be used.
- The controlling of the operating modes of the *technological functions* happens by means of handling blocks of the user program.

9.6.2 Analog input

9.6.2.1 Overview

- 2xUx12Bit (0 ... 10V)
- Sub module *'AI2'*
- *Chapter 5.3 'Analog input' on page 87*

9.6.2.2 Parametrization in *SPEED7 Studio*

9.6.2.2.1 *'I/O addresses'*

Sub module	Input address	Access	Assignment
AI2	800	WORD	Analog input channel 0 (X4)
	802	WORD	Analog input channel 1 (X4)

9.6.2.2.2 *'Parameter'*

'Filtering channel 0/1'

The analog input part has a filter integrated. The parametrization of the filter happens via the parameter *'Filter channel 0/1'*. The default value of the filter is 1000ms. The following values can be entered:

- 2ms: no filter
- 100ms: small filter
- 1000ms: medium filter
- 10000ms: maximum filter

9.6.3 Digital input

9.6.3.1 Overview

- 16xDC 24V
- Sub module *'DI16/DO12'*
- *Chapter 5.4 'Digital input' on page 90*

9.6.3.2 Parametrization in *SPEED7 Studio*

9.6.3.2.1 *'I/O addresses'*

Sub module	Input address	Access	Assignment
DI16/DO12	136	BYTE	Digital input I+0.0 ... I+0.7 (X4)
	137	BYTE	Digital input I+1.0 ... I+1.7 (X4)

9.6.3.2.2 *'Inputs'*

'Trigger for process interrupt'

Here you can specify a hardware interrupt for each input in groups of 2 channels for the corresponding edge. The hardware interrupt is disabled, if nothing is selected (default setting). A diagnostics interrupt is only supported with *Hardware interrupt lost*.

Here is valid:

- Rising edge: Edge 0-1
- Falling edge: Edge 1-0

Input delay

- The input delay can be configured per channel in groups of 4.
- An input delay of 0.1ms is only possible with "fast" inputs, which have a max. input frequency of 100kHz *Chapter 5.4 'Digital input' on page 90*. Within a group, the input delay for slow inputs is limited to 0.5ms.
- Range of values: 0.1ms / 0.5ms / 3ms / 15ms

9.6.4 Digital output

9.6.4.1 Overview

- 12xDC 24V, 0.5A
- Sub module *'DI16/DO12'*
- *Chapter 5.5 'Digital output' on page 94*

9.6.4.2 Parametrization in *SPEED7 Studio*

9.6.4.2.1 *'I/O addresses'*

Sub module	Output address	Access	Assignment
DI16/DO12	136	BYTE	Digital output Q+0.0 ... Q+0.7 (X5)
	137	BYTE	Digital output Q+1.0 ... Q+1.3 (X5)

9.6.5 Counter

9.6.5.1 Overview

- 4 channels
- Sub module: *'Count'*
- *Chapter 5.6 'Counting' on page 96*

9.6.5.2 Parametrization in *SPEED7 Studio*

9.6.5.2.1 *'I/O addresses'*

Sub module	Input address	Access	Assignment
Count	816	DINT	Channel 0: Counter value / Frequency value
	820	DINT	Channel 1: Counter value / Frequency value
	824	DINT	Channel 2: Counter value / Frequency value
	828	DINT	Channel 3: Counter value / Frequency value

9.6.5.2.2 Basic parameters

Select interrupt

Via *'Basic parameters'* you can reach *'Select interrupt'*. Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is disabled.
- Process: The following events of the counter can trigger a hardware interrupt (selectable via *'Count'*):
  - Hardware gate opening
  - Hardware gate closing
  - On reaching the comparator
  - on Counting pulse
  - on overflow
  - on underflow
- Diagnostics+process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

9.6.5.2.3 *'Channel x'*

Operating mode

Select via *'Channel'* the channel select via *'Operating'* the counter operating mode. The following counter operating modes are supported:

- Not parametrized: Channel is de-activated
- Count endless
- Count once
- Count periodical

Counter

Operating mode

Default values and structure of this dialog box depend on the selected *'Operating mode'*.

Parameter overview

Operating parameters	Description	Assignment
Main count direction	■ *None* No restriction of the counting range ■ *Up:* Restricts the up-counting range. The counter starts from 0 or *load value*, counts in positive direction up to the declaration *end value* -1 and then jumps back to *load value* at the next positive transducer pulse. ■ *Down:* Restricts the down-counting range. The counter starts from the declared *start value* or *load value* in negative direction, counts to 1 and then jumps to *start value* at the next negative encoder pulse. Function is disable with *count continuously*.	■ None
Gate function	■ *Cancel count:* The count starts when the gate opens and resumes at the *load value* when the gate opens again. ■ *Stop count:* The count is interrupted when the gate closes and resumed at the last actual counter value when the gate opens again. *Chapter 5.6.6.2 'Gate function' on page 111*	Abort count process
Start value End value	*Start value* with counting direction backward. *End value* with main counting direction forward. Range of values: 2...2147483647 (2^{31}-1)	2147483647 (2^{31}-1)
Comparison value	The count value is compared with the *comparison value*. See also the parameter "Characteristics of the output": ■ No main counting direction – Range of values: -2)^{31} to +2)^{31}-1 ■ Main counting direction forward – Range of values: -2^{31} to end value-1 ■ Main counting direction backward – Range of values: 1 to +2^{31}-1	0
Hysteresis	The *hysteresis* serves the avoidance of many toggle processes of the output, if the counter value is in the range of the *comparison value*. 0, 1: *Hysteresis* disabled Range of values: 0 to 255	0

Input	Description	Assignment
Signal evaluation	Specify the signal of the connected encoder: ■ Pulse/direction At the input count and direction signal are connected ■ At the input there is an encoder connected with the following evaluation: – Rotary encoder single – Rotary encoder double – Rotary encoder quadruple	Pulse/direction
Hardware gate	Gate control exclusively via channel 3: ■ enabled: The gate control for channel 3 happens via SW and HW gate ■ disabled: The gate control for channel 3 exclusively happens via SW gate *Chapter 5.6.6.2 'Gate function' on page 111*	disabled
Count direction inverted	Invert the input signal *'Direction'*: ■ enabled: The input signal is inverted ■ disabled: The input signal is not inverted	disabled

Output	Description	Assignment
Characteristics of the output	The output and the "Comparator" (STS_CMP) status bit are set, dependent on this parameter. ■ No comparison: The output is used as normal output and STS_CMP remains reset. ■ Comparator – Counter value ≥ Comparison value – Counter value ≤ Comparison value ■ Pulse at *comparison value* – To adapt the used actuators you can specify a *pulse duration*. The output is set for the specified *pulse duration* when the counter value reaches the *comparison value*. When you've set a main counting direction the output is only set at reaching the *comparison value* from the main counting direction.	No comparison
Pulse duration	Here you can specify the *pulse duration* for the output signal. ■ The *pulse duration* starts with the setting of the according digital output. ■ The inaccuracy of the *pulse duration* is less than 1ms. ■ There is no past triggering of the *pulse duration* when the *comparison value* has been left and reached again during pulse output. ■ If the *pulse duration* is changed during operation, it will take effect with the next pulse. ■ If the *pulse duration* = 0, the output is set until the comparison condition is not longer fulfilled. Range of values: 0...510ms in steps of 2ms	0

Frequency	Description	Assignment
Max. counting frequency	Specify the max. frequency for track A/pulse, track B/direction, Latch and HW gate Range of values: 1, 2, 5, 10, 30, 60, 100kHz	60kHz

Hardware interrupt	Description	Assignment
Hardware gate opening	Hardware interrupt by edge 0-1 exclusively at HW gate channel 3 ■ enabled: Process interrupt by edge 0-1 exclusively at HW gate channel 3 with open SW gate ■ disabled: no hardware interrupt	disabled
Hardware gate closing	Hardware interrupt by edge 1-0 exclusively at HW gate channel 3 ■ enabled: Process interrupt by edge 1-0 exclusively at HW gate channel 3 with open SW gate ■ disabled: no hardware interrupt	disabled
On reaching comparator	Hardware interrupt on reaching *comparator* ■ enabled: Hardware interrupt when comparator is triggered, can be configured via *'Characteristics of the output'* ■ disabled: no hardware interrupt	disabled
Overflow	Hardware interrupt overflow ■ enabled: Hardware interrupt on overflow the upper counter limit ■ disabled: no hardware interrupt	disabled
Underflow	Hardware interrupt on underrun ■ enabled: Hardware interrupt on underflow the lower counter limit ■ disabled: no hardware interrupt	disabled

9.6.6 Frequency measurement

9.6.6.1 Overview

- 4 channels
- Sub module *'Counter'*
- *Chapter 5.7 'Frequency measurement' on page 117*

9.6.6.2 Parametrization in *SPEED7 Studio*

9.6.6.2.1 *'I/O addresses'*

Sub module	Input address	Access	Assignment
Count	816	DINT	Channel 0: Counter value / Frequency value
	820	DINT	Channel 1: Counter value / Frequency value
	824	DINT	Channel 2: Counter value / Frequency value
	828	DINT	Channel 3: Counter value / Frequency value

Sub module	Output address	Access	Assignment
Count	816	DWORD	reserved
	820	DWORD	reserved
	824	DWORD	reserved
	828	DWORD	reserved

9.6.6.2.2 Basic parameters

Select interrupt

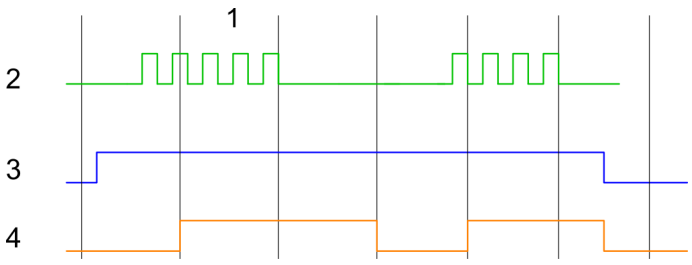
Via *'Basic parameters'* you can reach *'Select interrupt'*. Here you can define the interrupts the CPU will trigger. The following parameters are supported:

- None: The interrupt function is disabled.
- Process: The following events of the frequency measurement can trigger a hardware interrupt (selectable via *'Frequency counting'*):
  - End of measurement
- Diagnostics+process: A diagnostics interrupt is only triggered when a hardware interrupt was lost.

9.6.6.2.3 *'Channel x:'*

Operating mode

Select via *'Channel'* the channel and select for frequency measurement via *'Operating mode'* the operating mode *'Frequency counting'*. Default values and structure of this dialog box depend on the selected *'Operating mode'*. The following parameters are supported:



1 Integration time
2 Counting pulse
3 SW gate
4 Evaluated frequency

Parameter overview

Operating parameters	Description	Assignment
Integration time	Specify the integration time Range of values: 10ms ... 10000ms in steps of 1ms	100ms
max. counting frequency...	Specify the max. Frequency for the corresponding input Range of values: 1, 2, 5, 10, 30, 60, 100kHz	60kHz

Hardware interrupt	Description	Assignment
End of measurement	Hardware interrupt at end of measurement	de-activated

9.6.7 Pulse width modulation - PWM

9.6.7.1 Overview

- 2 channels
- Sub module *'Count'*
- *Chapter 5.8 'Pulse width modulation - PWM' on page 123*

9.6.7.2 Parametrization in *SPEED7 Studio*

9.6.7.2.1 *'I/O addresses'*

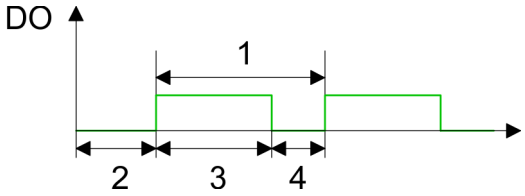
Sub module	Input address	Access	Assignment
Count	816	DINT	reserved
	820	DINT	reserved
	824	DINT	reserved
	828	DINT	reserved

Sub module	Output address	Access	Assignment
Count	816	DWORD	reserved
	820	DWORD	reserved
	824	DWORD	reserved
	828	DWORD	reserved

9.6.7.2.2 *'Channel x:'*

Operating mode

Select via *'Channel'* the channel and select for pulse width modulation via *'Operating mode'* the operating mode *'Pulse width modulation'*. Default values and structure of this dialog box depend on the selected *'Operating mode'*. The following parameters are supported:



1 Period
2 On-delay
3 Pulse duration
4 Pulse pause

Parameter overview

Operating parameters	Description	Assignment
Output format	Here specify the range of values for the output. The CPU hereby determines the pulse duration: ■ Per mil – Output value is within 0 ... 1000 – Pulse duration = (Output value / 1000) x Period ■ S7 Analog value: – Output value is Siemens S7 analog value 0 ... 27648 – Pulse duration = (Output value / 27648) x Period	Per mil
Time base	Here you can set the time base, which will apply for resolution and range of values of the period duration, minimum pulse duration and on-delay. ■ 1ms: The time base is 1ms ■ 0.1ms: The time base is 0.1ms ■ 1µs: The time base is 1µs	0.1ms
On-delay	Enter here a value for the time to expire from the start of the output sequence to the output of the pulse. The pulse sequence is output at the output channel, on expiration of the on-delay. Range of values: 0 ... 65535 from this there are the following effective values: ■ Time base 1ms: 0 ... 65535ms ■ Time base 0.1ms: 0 ... 6553.5ms ■ Time base 1µs: 0 ... 65535µs	0

Operating parameters	Description	Assignment
Period	With the period you define the length of the output sequence, which consists of pulse duration and pulse pause. Range of values: ■ Time base 1ms: 1 ... 87ms ■ Time base 0.1ms: 0.4 ... 87.0ms ■ Time base 1µs: 1 ... 87µs	20000
Minimum pulse duration	With the minimum pulse duration you can suppress short output pulses and short pulse pauses. All pulses or pauses, which are smaller than the minimum pulse duration, are suppressed. This allows you to filter very short pulses (spikes), which can not be recognized by the periphery. Range of values: ■ Time base 1ms: 0 ... Period / 2 * 1ms ■ Time base 0.1ms: 2 ... Period / 2 * 0.1ms ■ Time base 1µs: 0 ... Period / 2 * 1µs	2

9.7 *SPEED7 Studio* - Project transfer

Overview

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

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

9.7.1 Transfer via MPI

General

For the transfer via MPI the use of the optionally available extension module EM M09 is required. The extension module provides the interface X2: MPI(PB) with fixed pin assignment. *Chapter 2.4 'Mounting' on page 13*

Net structure

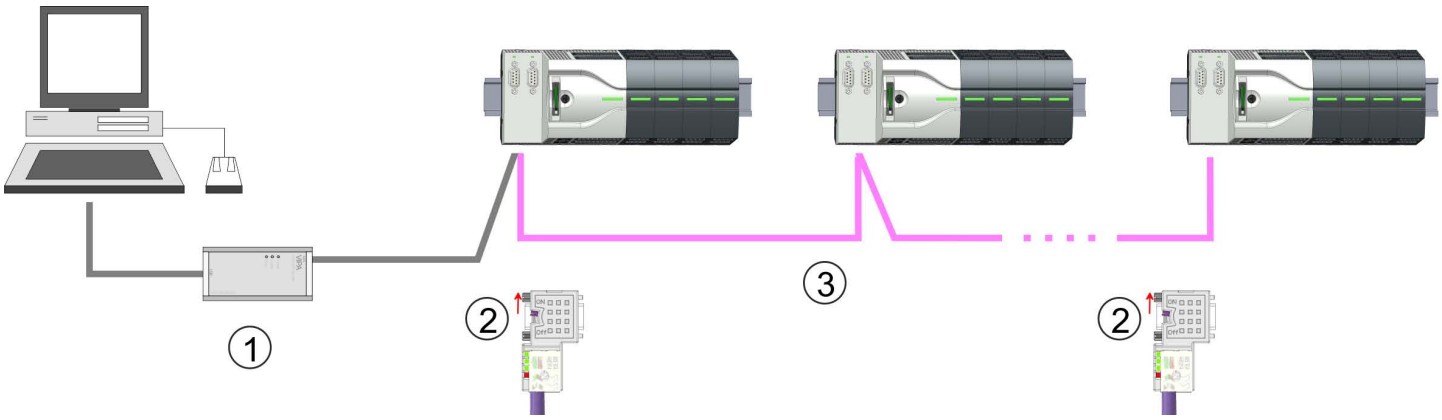
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs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

MPI programming cable

The MPI programming cables are available at VIPA 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

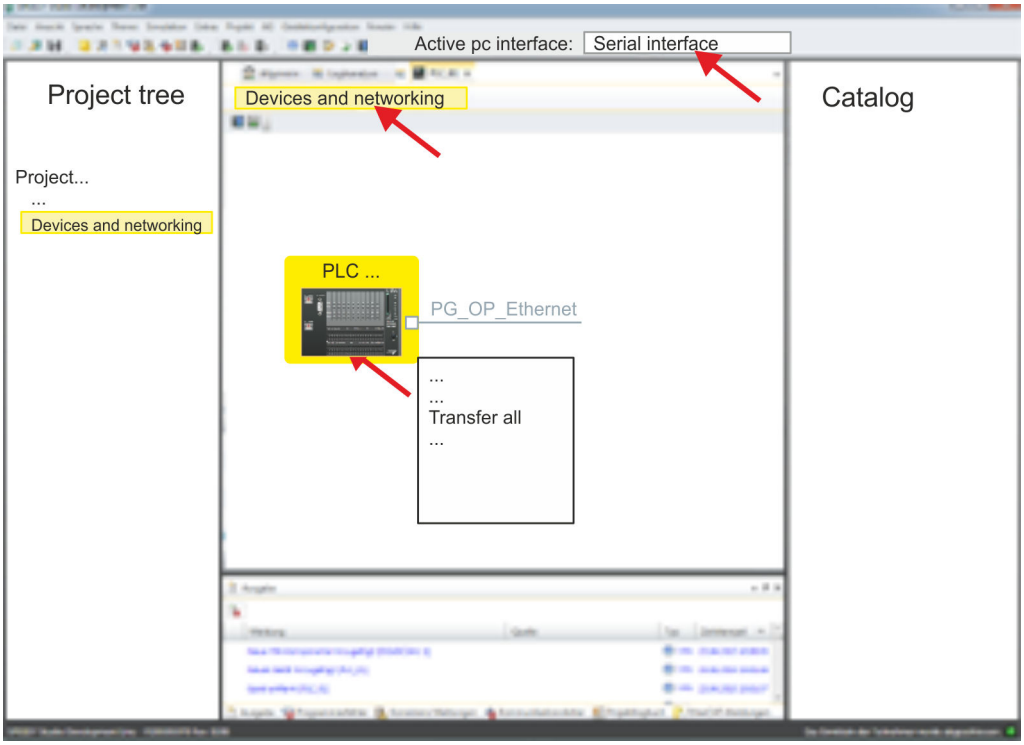
A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment. Please make sure that the participants with the activated terminating resistors are always power supplied. Otherwise it may cause interferences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

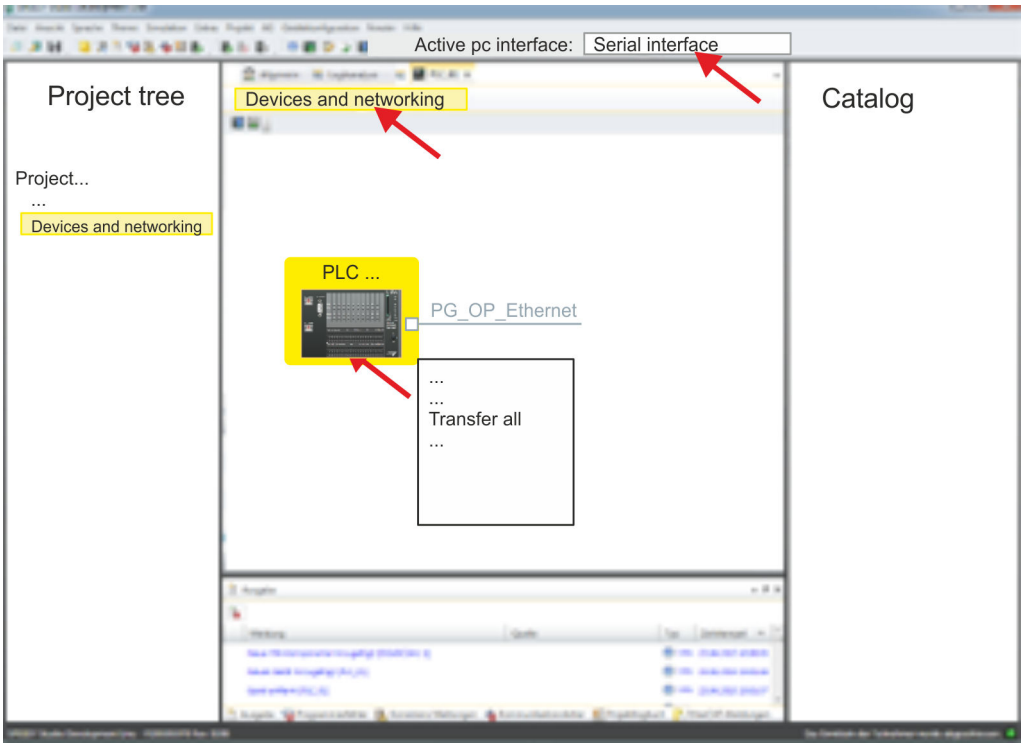
7. Confirm the request that the CPU is to be brought into the state STOP.
   ⇨ The user program and the hardware configuration are transferred via MPI to the CPU.
8. Close after transmission the dialog.
9. With *'Context menu ➔ Copy RAM to ROM'* you can save your project on a memory card, if one is plugged.

9.7.2 Transfer via Ethernet

Proceeding transfer via Ethernet

For transfer via Ethernet the CPU has an Ethernet PG/OP channel. For online access to this, you have to assign IP address parameters to this by means of "initialization" and transfer them into your project. For the transfer, connect, if not already done, the Ethernet PG/OP channel jack to your Ethernet. The connection happens via an integrated 2-port switch (X3, X4).

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



4. 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
5. Select the *'Port type'* "Ethernet interface" and start the transfer with *'Transfer'*.
6. 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.
7. Close after transmission the dialog.

8. With *'Context menu ➔ Copy RAM to ROM'* you can save your project on a memory card, if one is plugged.

9.7.3 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 flickering of the yellow LED  of the status bar 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 red LED  of the status bar lights up.

10 Configuration with TIA Portal

10.1 TIA Portal - Work environment

10.1.1 General

General

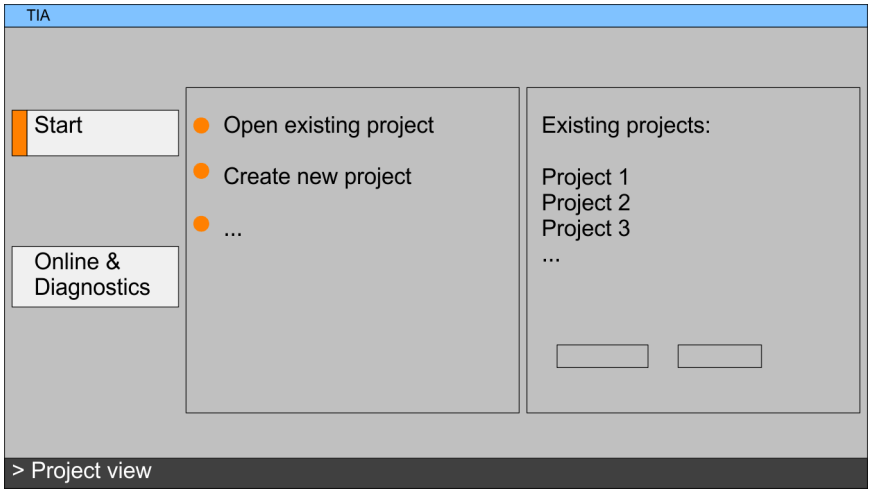
In this chapter the project engineering of the VIPA CPU in the Siemens TIA Portal is shown. Here only the basic usage of the Siemens TIA Portal together with a VIPA CPU is shown. Please note that software changes can not always be considered and it may thus be deviations to the description. TIA means **T**otally **i**ntegrated **A**utomation from Siemens. Here your VIPA 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.

10.1.2 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

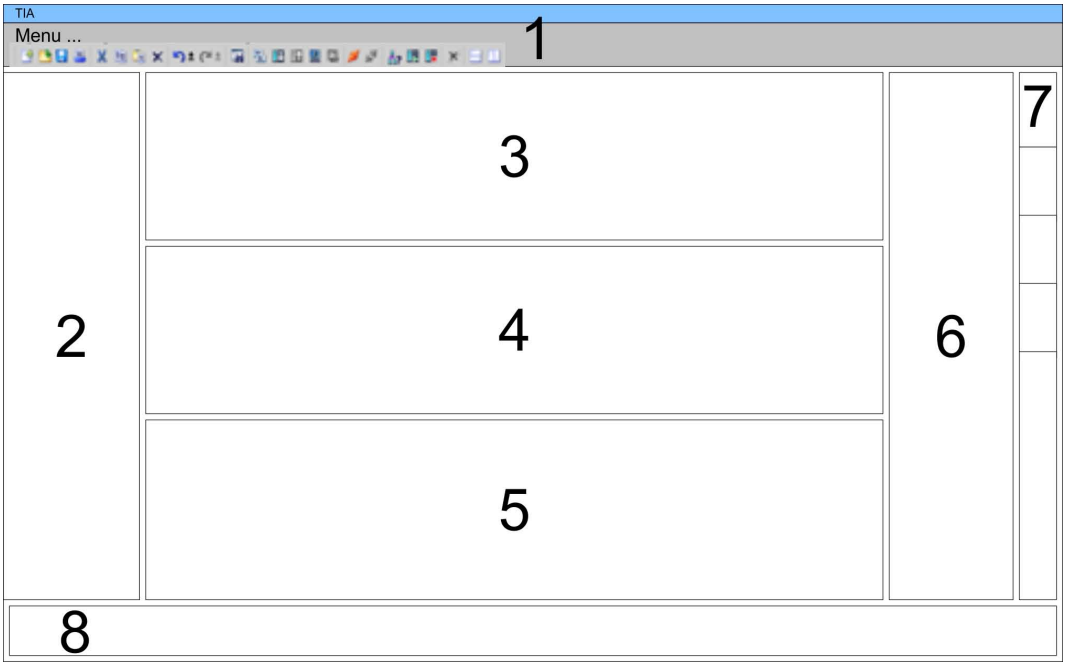
The *'Portal view'* provides a "task oriented" view of the tools for processing your project. Here you have direct access to the tools for a task. If necessary, a change to the Project view takes place automatically for the selected task.

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

10.2 TIA Portal - Hardware configuration - CPU

Overview

The hardware configuration of the CPU 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 GSDML *'VIPA MICRO PLC'* for PROFINET
- Configuration Siemens CPU
- Connection *'VIPA MICRO PLC'* as PROFINET IO device

Installation GSDML CPU for PROFINET

The installation of the PROFINET IO devices *'VIPA MICRO PLC'* happens in the hardware catalog with the following approach:

1. Go to the service area of www.vipa.com.
2. Load from the download area at *'PROFINET files '* the file Micro_PLC_Vxxx.zip.
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 GmbH* > *VIPA MICRO PLC*.

Thus, the VIPA components can be displayed, you have to deactivate the "Filter" of the hardware catalog.

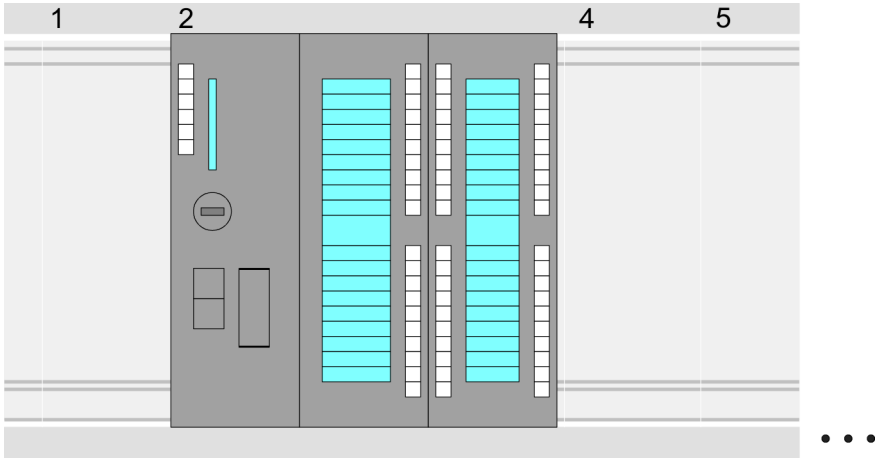
Configuration CPU

With the Siemens TIA Portal, the CPU from VIPA is to be configured as CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) 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 314C-2 PN/DP (314-6EH04-0AB0 V3.3)

   ⇨ The CPU is inserted with a profile rail.



Device overview:

Module	...	Slot	...	Type	...
PLC ...		2		CPU 314C-2PN/DP	
MPI interface...		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
DI24/DO16		2 5		DI24/DO16	
AI5/AO2...		2 6		AI5/AO2	

Counter...		2 7		Counter	
...					

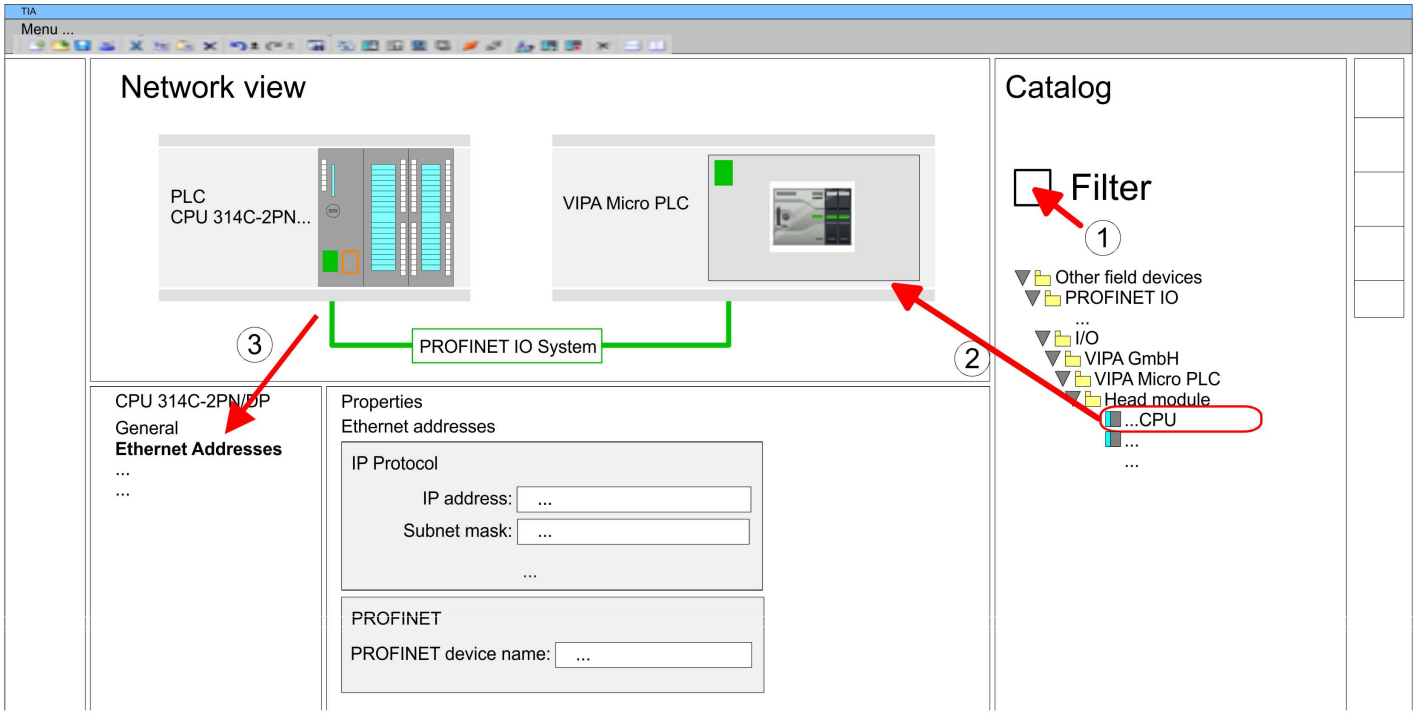
– *For parametrization of the digital I/O periphery and the technological functions the corresponding sub modules of the CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3) is to be used.*
– *The controlling of the operating modes of the technological functions happens by means of handling blocks of the user program.*

Setting standard CPU parameters

Since the CPU from VIPA is configured as Siemens CPU, so the setting of the non- VIPA 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. ↳ *Chapter 4.7 'Setting standard CPU parameters' on page 63*

Connection CPU as PROFINET IO device

1. Switch in the *Project area* to *'Network view'*.
2. After installing the GSDML the IO device for the CPU may be found in the hardware catalog at *Other field devices > PROFINET > IO > VIPA GmbH > VIPA MICRO PLC*. 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 at 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.



5. Select in the *Network view* the IO device *'VIPA MICRO PLC'* and switch to the *Device overview*.

   ⇨ In the *Device overview* of the PROFINET IO device *'VIPA MICRO PLC'* the CPU is already placed at slot 0. From slot 1 you can place your Extension module.

Setting VIPA specific CPU parameters

For parametrization click at the CPU at slot 0 in the *Device overview* of the PROFINET IO device *'VIPA MICRO PLC'*. Then the parameters of the CPU part are shown in the *Properties dialog*. Here you can make your parameter settings. *Chapter 4.8 'Setting VIPA specific CPU parameters' on page 66*

10.3 TIA Portal - Hardware configuration - Ethernet PG/OP channel

Overview

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 (X3/X4) is designed as switch. This enables PG/OP communication via the connections X3 and X4.
- The Ethernet PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.
- 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. This is called "initialization".
- This can be done with the Siemens TIA Portal.

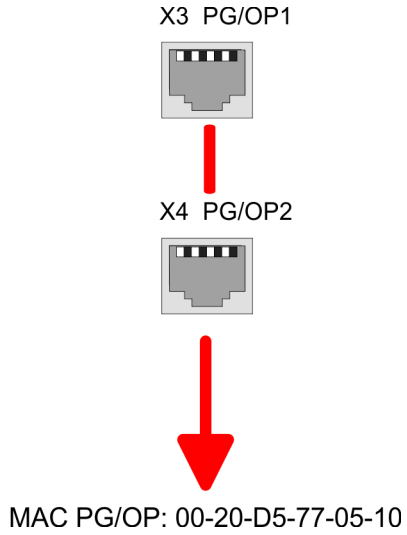
Assembly and commissioning

1. Install your System MICRO with your CPU.
2. Wire the system by connecting cables for voltage supply and signals.
3. Connect the one of the Ethernet jacks (X3, X4) 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 Online functions

The initialization via the Online functions takes place with the following proceeding:

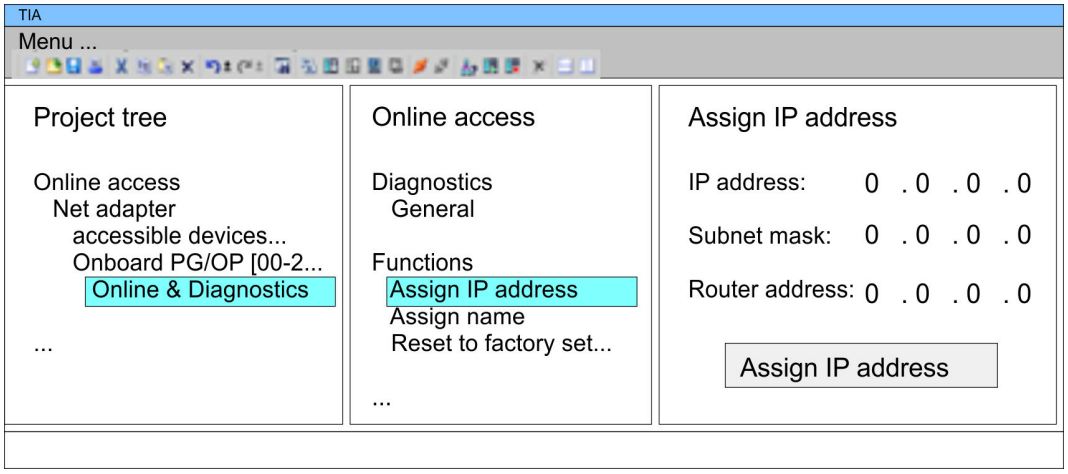
→ Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".



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 TIA Portal with the following proceeding:

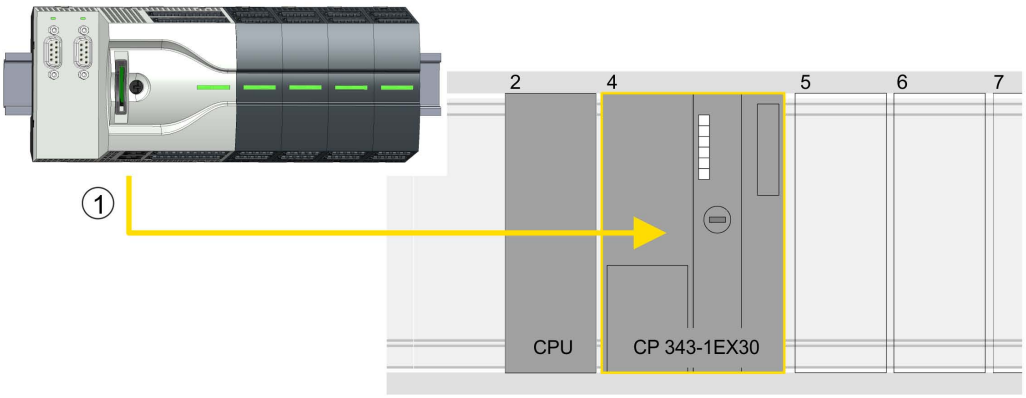
1. Start the Siemens TIA Portal.
2. Switch to the *'Project view'*.
3. Click in the *'Project tree'* at *'Online access'* and choose here by a doubleclick your network card, which is connected to the Ethernet PG/OP channel.
4. To get the stations and their MAC address, use the *'Accessible device'*. 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

1. Open your project.
2. If not already done, configure in the *'Device configuration'* a Siemens CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3).
3. As Ethernet PG/OP channel place at slot 4 the Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
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 314C-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	
CP 343-1		4		CP 343-1	
...		...		...	

10.4 TIA Portal - VIPA-Include library

Overview

- The VIPA specific blocks can be found in the "Service" area of www.vipa.com as library download file at *Downloads > VIPA LIB*.
- The library is available as packed zip file for the corresponding TIA Portal version.
- As soon as you want to use VIPA specific blocks you have to import them into your project.
  Execute the following steps:
  – Load an unzip the file ...TIA_V*xx*.zip (note TIA Portal version)
  – Open library and transfer blocks into the project

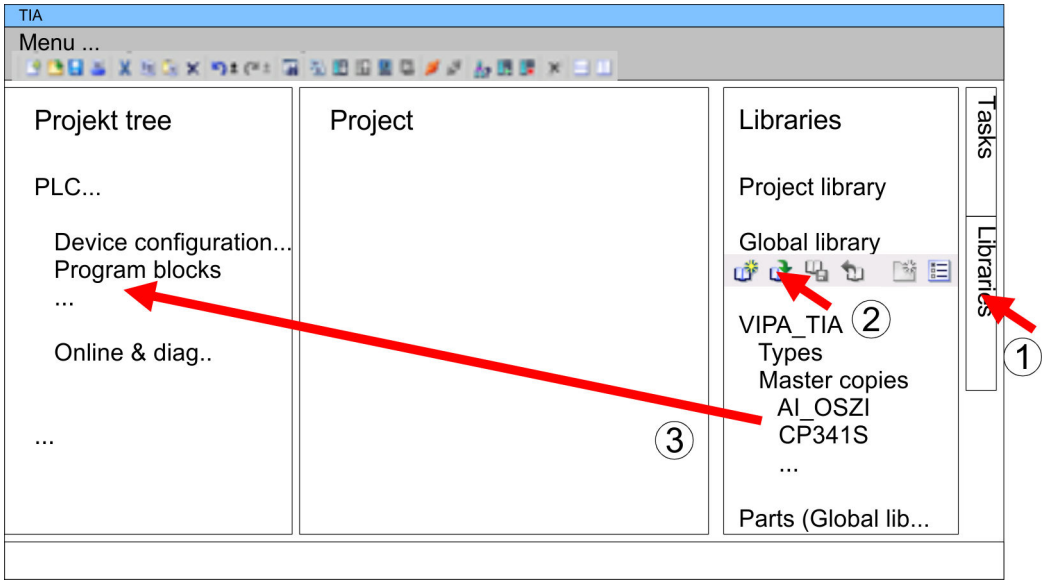
Unzip ...TIA_Vxx.zip

Start your un-zip application with a double click on the file TIA_V*xx*.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.al*xx*.



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 VIPA specific blocks via your user application.

10.5 TIA Portal - Project transfer

Overview

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

- Transfer via Ethernet
- Transfer via memory card
- Option: Transfer via MPI

10.5.1 Transfer via Ethernet

Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X3/X4: Ethernet PG/OP channel via an integrated 2-port switch

Initialization

So that you may the according Ethernet interface, you have to assign IP address parameters by means of the "initialization". *Chapter 10.3 'TIA Portal - Hardware configuration - Ethernet PG/OP channel' on page 205*

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.

10.5.2 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 flickering of the yellow LED  of the status bar 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 red LED  of the status bar lights up.

10.5.3 Option: Transfer via MPI

General

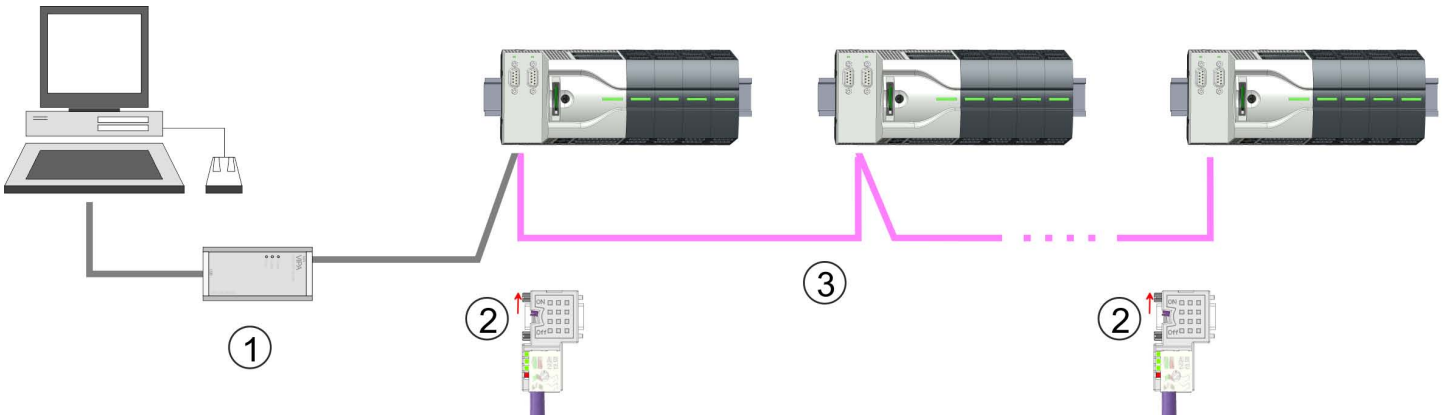
For the transfer via MPI the use of the optionally available extension module EM M09 is required. The extension module provides the interface X2: MPI(PB) with fixed pin assignment. *Chapter 2.4 'Mounting' on page 13*

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. VIPA CPUs are delivered with MPI address 2.

Terminating resistor

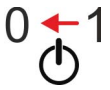
A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment. Please make sure that the participants with the activated terminating resistors are always power supplied. Otherwise it may cause interferences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



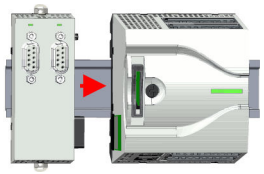
1 MPI programming cable
2 Activate the terminating resistor via switch
3 MPI network

Proceeding enabling the interface

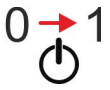
A hardware configuration to enable the MPI interface is not necessary. By installing the extension module EM M09 the MPI interface is enabled.



1. Turn off the power supply.



2. Mount the extension module. *Chapter 2.4 'Mounting' on page 13*



3. Switch on the power supply.
   ⇨ After a short boot time the interface X2 MPI(PB) is ready for MPI communication with the MPI address 2.

Proceeding transfer via MPI interface

Currently the VIPA programming cables for transfer via MPI are not supported. This is only possible with the programming cable from Siemens. 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.

1. Establish a connection to your extension module via MPI with an appropriate programming cable. Information may be found in the corresponding documentation of the programming cable.
2. 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.

Appendix

Content

A System specific event IDs

Event IDs *Chapter 4.19 'Diagnostic entries' on page 85*

Event ID	Description
0x115C	Vendor-specific interrupt (OB 57) at EtherCAT
	OB: OB number
	ZInfo1: Logical address of the slave that triggered the interrupt
	ZInfo2: Interrupt type
	0x00: Reserved
	0x01: Diagnostic interrupt (incoming)
	0x02: Hardware interrupt
	0x03: Pull interrupt
	0x04: Plug interrupt
	0x05: Status interrupt
	0x06: Update interrupt
	0x07: Redundancy interrupt
	0x08: Controlled by the supervisor
	0x09: Enabled
	0x0A: Wrong sub module plugged
	0x0B: Restoration of the sub module
	0x0C: Diagnostic interrupt (outgoing)
	0x0D: Cross traffic connection message
	0x0E: Neighbourhood change message
	0x0F: Synchronisation message (bus)
	0x10: Synchronisation message (device)
	0x11: Network component message
	0x12: Clock synchronisation message (bus)
	0x1F: Pull interrupt module
	ZInfo3: CoE error code
0xE003	Error on accessing the 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!
0xE007	Configured in-/output bytes do not fit into periphery area
0xE008	Internal error - Please contact the hotline!

Event ID	Description
0xE009	Error on accessing the standard backplane bus
0xE010	There is a undefined module at the backplane bus
	ZInfo2 : Slot
	ZInfo3 : Type ID
0xE011	Master project engineering at slave CPU not possible or wrong slave configuration
0xE012	Error at parametrization
0xE013	Error at shift register access to standard bus digital modules
0xE014	Error at Check_Sys
0xE015	Error at access to the master
	ZInfo2 : Slot of the master
	ZInfo2 : Page frame master
0xE016	Maximum block size at master transfer exceeded
	ZInfo1 : Periphery address
	ZInfo2 : Slot
0xE017	Error at access to integrated slave
0xE018	Error at mapping of the master periphery
0xE019	Error at standard back plane bus system recognition
0xE01A	Error at recognition of the operating mode (8 / 9 bit)
0xE01B	Error - maximum number of plug-in modules exceeded
0xE020	Error - Interrupt information undefined
	ZInfo2 : Slot
	ZInfo3 : Not relevant to the user
	DatID : Interrupt type
0xE030	Error of the standard bus
0xE033	Internal error - Please contact the hotline!
0xE0B0	SPEED7 is not stoppable (e.g. undefined BCD value at timer)
	ZInfo1 : Not relevant to the user
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	DatID : Not relevant to the user
0xE0C0	Not enough space in work memory for storing code block (block size exceeded)
0xE0CB	Error at SSL access
	ZInfo1 : Error
	4: SSL wrong
	5: Sub-SSL wrong
	6: Index wrong
	ZInfo2 : SSL ID
	ZInfo3 : Index

Event ID	Description
0xE0CC	Communication errors
	ZInfo1 : Error code
	1: Wrong priority
	2: Buffer overflow
	3: Telegram format error
	4: Wrong SSL request (SSL ID not valid)
	5: Wrong SSL request (SSL sub ID invalid)
	6: Wrong SSL request (SSL-Index not valid)
	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 relevant to the user
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	DatID : Not relevant to the user
0xE0CE	Error: Timeout at sending of the 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 relevant to the user
0xE200	Memory card writing finished (Copy Ram2Rom)
	PK : Not relevant to the user
	OB : Not relevant to the user
0xE210	Memory card reading finished (reload after overall reset)
	ZInfo1 : Not relevant to the user
	PK : Not relevant to the user

Event ID	Description
	OB : Not relevant to the user
0xE21E	Memory card reading: Error at reload (after overall reset), error in block header
	ZInfo1 : Block type
	0x38: OB
	0x41: DB
	0x42: SDB
	0x43: FC
	0x44: SFC
	0x45: FB
	0x46: SFB
	0x6F: VOB
	0x65: VFB
	0x63: VFC
	0x61: VDB
	0x62: VSDB
	0x64: VSFC
	0x66: VSFB
	ZInfo2 : Block number
	ZInfo3 : Block length
0xE21E	Memory card reading: Error at reload (after overall reset), file "Protect.wld" too big
	OB : Not relevant to the user
0xE21F	Memory card reading: Error at reload (after overall reset), checksum error at reading
	PK : Not relevant to the user
	OB : Not relevant to the user
	ZInfo1 : Not relevant to the user
	ZInfo2 : BstTyp
	0x38: OB
	0x41: DB
	0x42: SDB
	0x43: FC
	0x44: SFC
	0x45: FB
	0x46: SFB
	0x6F: VOB
	0x65: VFB
	0x63: VFC
	0x61: VDB
	0x62: VSDB

Event ID	Description
	0x64: VSFC
	0x66: VSFB
	ZInfo3 : BstNr
0xE300	Internal flash writing finished (Copy Ram2Rom)
0xE310	Internal flash writing finished (reload after battery failure)
0xE400	FSC card was plugged
	DatID : FeatureSet Trialtime in minutes
	ZInfo1 : Memory extension in kB
	ZInfo2 : FeatureSet PROFIBUS
	ZInfo2 : FeatureSet field bus
	ZInfo2 : FeatureSet motion
	ZInfo2 : Reserved
0xE401	FSC card was removed
	DatID : FeatureSet Trialtime in minutes
	ZInfo1 : Memory extension in kB
	ZInfo2 : FeatureSet PROFIBUS
	ZInfo2 : FeatureSet field bus
	ZInfo2 : FeatureSet motion
	ZInfo2 : Reserved
	ZInfo3 : Source of the FSC
	0: CPU
	1: Card
0xE402	A configured functionality is not activated
	ZInfo1 : FCS ErrorCode
	1: The PROFIBUS functionality is disabled The interface acts further as MPI interface
	2: The EtherCAT functionality is not enabled
	3: The number of configured axis is not enabled
0xE403	FSC can not be activated in this CPU
	ZInfo1 : Memory extension in kB
	ZInfo2 : FeatureSet PROFIBUS
	ZInfo2 : FeatureSet field bus
	ZInfo2 : FeatureSet motion
	ZInfo2 : Reserved
0xE404	FeatureSet deleted due to CRC error
	DatID : Not relevant to the user
0xE405	The trial time of a feature set or MMC has expired
	DatID : Not relevant to the user
0xE410	A CPU feature set was activated

Event ID	Description
	DatID : Not relevant to the user
0xE500	Memory management: Deleted block without corresponding entry in BstList
	ZInfo2 : Block type
	0x38: OB
	0x41: DB
	0x42: SDB
	0x43: FC
	0x44: SFC
	0x45: FB
	0x46: SFB
	0x6F: VOB
	0x65: VFB
	0x63: VFC
	0x61: VDB
	0x62: VSDB
	0x64: VSFC
	0x66: VSFB
	ZInfo3 : Block no.
0xE501	Parser error
	ZInfo3 : SDB number
	ZInfo1 : ErrorCode
	1: Parser error: SDB structure
	2: Parser error: SDB is not a valid SDB type.
	ZInfo2 : SDB type
0xE502	Invalid block type in protect.wld
	ZInfo2 : Block type
	0x38: OB
	0x41: DB
	0x42: SDB
	0x43: FC
	0x44: SFC
	0x45: FB
	0x46: SFB
	0x6F: VOB
	0x65: VFB
	0x63: VFC
	0x61: VDB
	0x62: VSDB

Event ID	Description
	0x64: VSFC
	0x66: VSFB
	ZInfo3 : Block number
0xE503	Inconsistency of code size and block size in work memory
	ZInfo1 : Code size
	ZInfo2 : Block size (high word)
	ZInfo3 : Block size (low word)
0xE504	Additional information for CRC error in work memory
	ZInfo2 : Block address (high word)
	ZInfo3 : Block address (low word)
0xE505	Internal error - Please contact the hotline!
0xE604	Multiple parametrization 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 : Slot of the interface
	ZInfo2 : Number configured connections
	ZInfo3 : Number of allowed connections
0xE610	Onboard PROFIBUS/MPI: Bus error fixed
	ZInfo1 : Interface
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xE701	Internal error - Please contact the hotline!
0xE703	Internal error - Please contact the hotline!
0xE710	Onboard PROFIBUS/MPI: Bus error occurred
	ZInfo1 : Interface
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xE720	Internal error - Please contact the hotline!
0xE721	Internal error - Please contact the hotline!
0xE722	Internal error - Please contact the hotline!
0xE723	Internal error - Please contact the hotline!
0xE780	Internal error - Please contact the hotline!
0xE801	CMD - Auto command: CMD_START recognized and successfully executed

Event ID	Description
0xE802	CMD - Auto command: CMD_End recognized and successfully executed
0xE803	CMD - Auto command: WAIT1SECOND recognized and successfully executed
0xE804	CMD - Auto command: WEBPAGE recognized and successfully executed
0xE805	CMD - Auto command: LOAD_PROJECT recognized and successfully executed
0xE806	CMD - Auto command: SAVE_PROJECT recognized and successfully executed
	ZInfo3 : Status
	0: Error
	1: OK
	0x8000: Wrong password
0xE807	CMD - Auto command: FACTORY_RESET recognized and successfully executed
0xE808	Internal error - Please contact the hotline!
0xE809	Internal error - Please contact the hotline!
0xE80A	Internal error - Please contact the hotline!
0xE80B	CMD - Auto command: DIAGBUF recognized and successfully executed
	ZInfo3 : Status
	0: OK
	0xFE81: File create error
	0xFEA1: File write error
	0xFEA2: Odd address when reading
0xE80C	Internal error - Please contact the hotline!
0xE80D	Internal error - Please contact the hotline!
0xE80E	CMD - Auto command: SET_NETWORK recognized and successfully executed
0xE80F	Internal error - Please contact the hotline!
0xE810	Internal error - Please contact the hotline!
0xE811	Internal error - Please contact the hotline!
0xE812	Internal error - Please contact the hotline!
0xE813	Internal error - Please contact the hotline!
0xE814	CMD - Auto command: SET_MPI_ADDRESS recognized
0xE816	CMD - Auto command: SAVE_PROJECT recognized but not executed, because the CPU memory is empty
0xE817	Internal error - Please contact the hotline!
0xE820	Internal message
0xE821	Internal message
0xE822	Internal message
0xE823	Internal message
0xE824	Internal message
0xE825	Internal message
0xE826	Internal message
0xE827	Internal message

Event ID	Description
0xE828	Internal message
0xE829	Internal message
0xE82A	CMD - Auto command: CPUTYPE_318 recognized and successfully executed
	ZInfo3 : Error code
	0: No Error
	1: Command not possible
	2: Error on storing the attribute
0xE82B	CMD - Auto command: CPUTYPE_ORIGINAL recognized and successfully executed
	ZInfo3 : Error code
	0: No Error
	1: Command not possible
	2: Error on storing the attribute
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 missing
0xE8FF	CMD - Auto command: Error: Error while reading CMD file (memory card error)
0xE901	Check sum error
	ZInfo1 : Not relevant to the user
	ZInfo2 : Not relevant to the user
	DatID : Not relevant to the user
0xE902	Internal error - Please contact the hotline!
0xEA00	Internal error - Please contact the hotline!
0xEA01	Internal error - Please contact the hotline!
0xEA02	SBUS: Internal error (internal plugged sub module not recognized)
	ZInfo1 : Slot
	ZInfo2 : Type ID set
	ZInfo3 : Type ID
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEA03	SBUS: Communication error between CPU and IO controller
	ZInfo1 : Slot
	ZInfo2 : Status
	0: OK
	1: Error
	2: Empty
	3: Busy
	4: Timeout
	5: Internal blocking

Event ID	Description
	6: Too many frames
	7: Not connected
	8: Unknown
	PK : Not relevant to the user
	DatID : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
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: Parametrized input data width unequal to plugged input data width
	ZInfo1 : Parametrized input data width
	ZInfo2 : Slot
	ZInfo3 : Input data width of the plugged module
0xEA09	SBUS: Parametrized output data width unequal to plugged output data width
	ZInfo1 : Parametrized output data width
	ZInfo2 : Slot

Event ID	Description
	ZInfo3 : Output data width of the plugged module
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 at writing record set
	ZInfo1 : Slot
	ZInfo2 : Record set number
	ZInfo3 : Record set length
0xEA14	SBUS: Multiple parametrization of a periphery address (diagnostics address)
	ZInfo1 : Periphery address
	ZInfo2 : Slot
	ZInfo3 : Data width
0xEA15	Internal error - Please contact the hotline!
0xEA18	SBUS: Error at mapping of the master periphery
	ZInfo2 : Slot of the master
0xEA19	Internal error - Please contact the hotline!
0xEA1A	SBUS: Error at access to the FPGA address table
	ZInfo2 : HW slot
	ZInfo3 : Table
	0: Reading
	1: Writing
	PK : Not relevant to the user
	DatID : Not relevant to the user
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 faulty configured.
0xEA22	Error - RS485 interface X2 - Value exceeds the limits
	ZInfo2 : Project engineering for X2
0xEA23	Error - RS485 interface X3 - Value exceeds the limits
	ZInfo2 : Project engineering for X3
0xEA24	Error - Configuration RS485 interface X2/X3: Interface/protocol missing, default settings are used.
	ZInfo2 : Project engineering for X2
	ZInfo3 : Project engineering for X3

Event ID	Description
0xEA30	Internal error - Please contact the hotline!
0xEA40	Internal error - Please contact the hotline!
0xEA41	Internal error - Please contact the hotline!
0xEA50	PROFINET IO controller: Error in the configuration
	ZInfo1 : Rack/slot of the controller
	ZInfo2 : Device no.
	ZInfo3 : Slot at the device
	OB : Not relevant to the user
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEA51	PROFINET IO CONTROLLER: There is no PROFINET IO controller at the configured slot
	ZInfo1 : Rack/slot of the controller
	ZInfo2 : Recognized ID at the configured slot
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEA53	PROFINET IO CONTROLLER: PROFINET configuration: There are too many PROFINET IO devices configured
	ZInfo1 : Number of configured devices
	ZInfo2 : Slot
	ZInfo3 : Maximum possible number of devices
0xEA54	PROFINET IO controller: IO controller reports multiple parametrization of a periphery address
	ZInfo1 : Periphery address
	ZInfo2 : Slot
	ZInfo3 : Data width
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEA61	Internal error - Please contact the hotline!
0xEA62	Internal error - Please contact the hotline!
0xEA63	Internal error - Please contact the hotline!
0xEA64	PROFINET IO controller/EtherCAT-CP: Error in the configuration
	ZInfo1 : Too many devices
	ZInfo1 : Too many devices per second
	ZInfo1 : Too many input bytes per ms
	ZInfo1 : Too many output bytes per ms
	ZInfo1 : Too many input bytes per ms
	ZInfo1 : Too many output bytes per device
	ZInfo1 : Too many productive connections
	ZInfo1 : Too many input bytes in the process image
	ZInfo1 : Too many output bytes in the process image

Event ID	Description
	ZInfo1 : Configuration not available
	ZInfo1 : Configuration not valid
	ZInfo1 : Refresh time too short
	ZInfo1 : Cycle time too big
	ZInfo1 : Not valid device number
	ZInfo1 : CPU is configured as I device
	ZInfo1 : Use different method to obtain IP address Is not supported for the IP address of the controller
	ZInfo2 : Incompatible configuration (SDB version not supported)
	ZInfo2 : EtherCAT: EoE configured but not supported
	ZInfo2 : DC parameter not valid
0xEA65	Internal error - Please contact the hotline!
0xEA66	PROFINET error in communication stack
	PK : Rack/slot
	OB : StackError.Service
	DatID : StackError.DeviceRef
	ZInfo1 : StackError.Error.Code
	ZInfo2 : StackError.Error.Detail
	ZInfo3 : StackError.Error.AdditionalDetail
	ZInfo3 : StackError.Error.AreaCode
0xEA67	PROFINET IO controller: Error reading record set
	PK : Error type
	0: Record set error local
	1: Record set error stack
	2: Record set error station
	OB : Rack/slot of the controller
	DatID : Device
	ZInfo1 : Record set number
	ZInfo2 : Record set handle (caller)
	ZInfo3 : Internal error code from PN stack
0xEA68	PROFINET IO controller: Error at writing record set
	PK : Error type
	0: Record set error local
	1: Record set error stack
	2: Record set error station
	OB : Rack/slot of the controller
	DatID : Device
	ZInfo1 : Record set number
	ZInfo2 : Record set handle (caller)

Event ID	Description
	ZInfo3 : Internal error code from PN stack
0xEA69	Internal error - Please contact the hotline!
0xEA6A	PROFINET IO controller: Service error in communication stack
	PK : Rack/slot
	OB : Service ID
	ZInfo1 : ServiceError.Code
	ZInfo2 : ServiceError.Detail
	ZInfo3 : StackError.Error.AdditionalDetail
	ZInfo3 : ServiceError.AreaCode
0xEA6B	PROFINET IO controller: Faulty vendor ID
	ZInfo1 : Device ID
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	PK : Rack/slot
	DatID : Not relevant to the user
0xEA6C	PROFINET IO controller: Faulty device ID
	ZInfo1 : Device ID
	PK : Rack/slot

Event ID	Description
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
0xEA6D	PROFINET IO controller: No empty Name
	ZInfo1 : Device ID
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING

Event ID	Description
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	PK : Rack/slot
	DatID : Not relevant to the user
0xEA6E	PROFINET IO controller: Waiting for RPC answer
	ZInfo1 : Device ID
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	PK : Rack/slot
	DatID : Not relevant to the user
0xEA6F	PROFINET IO controller: PROFINET module deviation
	ZInfo1 : Device ID
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user

Event ID	Description
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	PK : Rack/slot
	DatID : Not relevant to the user
0xEA70	PROFINET stack error in configuration
	ZInfo1 : UnsupportedApiError.slot
	ZInfo2 : UnsupportedApiError.subslot
	OB : UnsupportedApiError.api
	PK : Rack Slot No
	DatID : UnsupportedApiError.deviceID
0xEA71	Internal PROFINET error - Please contact the hotline!
0xEA81	Internal error - Please contact the hotline!
0xEA82	Internal error - Please contact the hotline!
0xEA83	Internal error - Please contact the hotline!
0xEA91	Internal error - Please contact the hotline!
0xEA92	Internal error - Please contact the hotline!
0xEA93	Internal error - Please contact the hotline!
0xEA97	Internal error - Please contact the hotline!
0xEA98	Timeout at waiting for reboot of a SBUS module (server)
	PK : Not relevant to the user

Event ID	Description
	DatID : Not relevant to the user
	ZInfo3 : Slot
0xEA99	Error at file reading via SBUS
	ZInfo3 : Slot
	PK : Not relevant to the user
	DatID : Not relevant to the user
	ZInfo2 : File version of the SBUS module (if not equal to 0)
	ZInfo1 : File version at MMC/SD (if not equal 0)
0xEAA0	Internal error - Please contact the hotline!
0xEAB0	Link mode not valid
	ZInfo1 : Diagnostics address of the master
	ZInfo2 : Current connection mode
	0x01: 10Mbit half-duplex
	0x02: 10Mbit full-duplex
	0x03: 100Mbit half-duplex
	0x04: 100Mbit full-duplex
	0x05: Link mode undefined
	0x06: Auto Negotiation
	OB : Current operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set

Event ID	Description
0xEAC0	Internal error - Please contact the hotline!
0xEAD0	Error in configuration SyncUnit
0xEB02	Backplane bus: Present configuration does not match the actual configuration
	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
	ZInfo1 : Type of error
	0x01: SDB parser error
	0x02: Configured address already used
	0x03: Mapping error
	PK : Not relevant to the user
	DatID : Not relevant to the user
	ZInfo2 : Slot (0=not be determined)
0xEB05	System error: Bus structure for Isochron process image not suitable
	PK : Not relevant to the user
	DatID : Not relevant to the user
	ZInfo2 : Slot (0=not be determined)
0xEB10	System error: Bus error
	ZInfo1 : Type of error
	0x60: Bus enumeration error
	0x80: General error
	0x81: Queue execution error
	0x82: Error interrupt
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEB11	System error during bus initialization
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEB20	System error: Interrupt information undefined
0xEB21	System error: Accessing configuration data
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	DatID : Not relevant to the user
0xEC03	EtherCAT: Error in configuration
	ZInfo1 : Error code
	1: Number of slaves is not supported.

Event ID	Description
	2: Master system ID not valid
	3: Slot not valid
	4: Master configuration not valid
	5: Master type not valid
	6: Slave diagnostic address invalid
	7: Slave address not valid
	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)
	PK : Not relevant to the user
	DatID : Not relevant to the user
	ZInfo2 : Error code higher 2 bytes
	ZInfo3 : Error code lower 2 bytes
0xEC04	EtherCAT Multiple configuration of a periphery address
	ZInfo1 : Periphery address
	ZInfo2 : Slot
	PK : Not relevant to the user
	DatID : Not relevant to the user
0xEC05	EtherCAT: Check the set DC mode of the YASKAWA Sigma 5/7 drive
	PK : Not relevant to the user
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE

Event ID	Description
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	DatID : Not relevant to the user
	ZInfo1 : Station address of the EtherCAT device
	ZInfo2 : Errorcode
	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 Beckhoff 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
	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 timeout, 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 relevant to the user
0xEC10	EtherCAT: Restoration bus with its slaves
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the station
	ZInfo3 : Number of stations, which are not in the same state as the master
	DatID : Input address
	DatID : Output address

Event ID	Description
	DatID : Station not available
	DatID : Station available
0xEC11	EtherCAT: Restoration bus with missing slaves
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : Number of stations, which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xEC12	EtherCAT: Restoration slave
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op

Event ID	Description
	ZInfo2 : Diagnostics address of the station
	ZInfo3 : AL Statuscode
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xEC30	EtherCAT: Topology OK
	ZInfo2 : Diagnostics address of the master
0xEC50	EtherCAT: DC out of sync
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : DC State Change
	0: DC master out of sync
	1: DC slaves out of Sync
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
0xED10	EtherCAT: Bus failure
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp

Event ID	Description
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostic address of the master
	ZInfo3 : Number of stations, which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED12	EtherCAT: Slave failure
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the station
	ZInfo3 : AlStatusCode
	0x0000: No Error
	0x0001: Unspecified error
	0x0011: Invalid requested status change
	0x0012: Unknown requested status
	0x0013: Bootstrap not supported

Event ID	Description
	0x0014: No valid firmware
	0x0015: Invalid mailbox configuration
	0x0016: Invalid mailbox configuration
	0x0017: Invalid sync manager configuration
	0x0018: No valid inputs available
	0x0019: No valid outputs available
	0x001A: Synchronisation error
	0x001B: Sync manager watchdog
	0x001C: Invalid sync manager types
	0x001D: Invalid output configuration
	0x001E: Invalid input configuration
	0x001F: Invalid watchdog configuration
	0x0020: Slave needs cold start
	0x0021: Slave needs INIT
	0x0022: Slave needs PreOp
	0x0023: Slave needs SafeOp
	0x002D: Invalid output FMMU configuration
	0x002E: Invalid input FMMU configuration
	0x0030: Invalid DC Sync configuration
	0x0031: Invalid DC Latch configuration
	0x0032: PLL error
	0x0033: Invalid DC IO error
	0x0034: Invalid DC timeout error
	0x0042: Error in acyclic data exchange Ethernet over EtherCAT
	0x0043: Error in acyclic data exchange CAN over EtherCAT
	0x0044: Error in acyclic data exchange file access over EtherCAT
	0x0045: Error in acyclic data exchange servo drive profile over EtherCAT
	0x004F: Error in acyclic data exchange vendor specific over EtherCAT
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED20	EtherCAT: Bus state change without calling OB86
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap

Event ID	Description
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : Number of stations, which are not in the same state as the master
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED21	EtherCAT: Faulty bus status change
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : Error code
	0x0008: Busy
	0x000B: Invalid parameters
	0x000E: Invalid status
	0x0010: Timeout
	0x0004: Abbort (master state change)
	DatID : Input address

Event ID	Description
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED22	EtherCAT: Slave state change without calling OB86
	ZInfo1 : Old status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Diagnostics address of the station
	ZInfo3 : AlStatusCode
	0x0000: No Error
	0x0001: Unspecified error
	0x0011: Invalid requested status change
	0x0012: Unknown requested status
	0x0013: Bootstrap not supported
	0x0014: No valid firmware
	0x0015: Invalid mailbox configuration
	0x0016: Invalid mailbox configuration
	0x0017: Invalid sync manager configuration
	0x0018: No valid inputs available
	0x0019: No valid outputs available
	0x001A: Synchronisation error
	0x001B: Sync manager watchdog
	0x001C: Invalid sync manager types
	0x001D: Invalid output configuration
	0x001E: Invalid input configuration
	0x001F: Invalid watchdog configuration
	0x0020: Slave needs cold start

Event ID	Description
	0x0021: Slave needs INIT
	0x0022: Slave needs PreOp
	0x0023: Slave needs SafeOp
	0x002D: Invalid output FMMU configuration
	0x002E: Invalid input FMMU configuration
	0x0030: Invalid DC Sync configuration
	0x0031: Invalid DC Latch configuration
	0x0032: PLL error
	0x0033: Invalid DC IO error
	0x0034: Invalid DC timeout error
	0x0042: Error in acyclic data exchange Ethernet over EtherCAT
	0x0043: Error in acyclic data exchange CAN over EtherCAT
	0x0044: Error in acyclic data exchange file access over EtherCAT
	0x0045: Error in acyclic data exchange servo drive profile over EtherCAT
	0x004F: Error in acyclic data exchange vendor specific over EtherCAT
	DatID : Input address
	DatID : Output address
	DatID : Station not available
	DatID : Station available
0xED23	EtherCAT: Timeout while changing the master status to OP, after CPU has changed to RUN
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP

Event ID	Description
	0xFE: Watchdog
	0xFF: Not set
	ZInfo1 : Master status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : There is an EtherCAT configuration
	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 : Diagnostics address of the master
0xED31	EtherCAT: Overflow of the interrupt queue
	ZInfo2 : Diagnostics address of the master
0xED50	EtherCAT: DC slaves in sync
	ZInfo2 : Diagnostics address of the master
	ZInfo3 : DC State change
	0: Master
	1: Slave
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING

Event ID	Description
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
0xED60	EtherCAT: Diagnostics buffer CP: Slave state change
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	ZInfo1 : New status
	0x00: Undefined/Unkown
	0x01: INIT
	0x02: PreOp
	0x03: BootStrap
	0x04: SafeOp
	0x08: Op
	ZInfo2 : Slave address
	ZInfo3 : AlStatusCode
	0x0000: No Error
	0x0001: Unspecified error

Event ID	Description
	0x0011: Invalid requested status change
	0x0012: Unknown requested status
	0x0013: Bootstrap not supported
	0x0014: No valid firmware
	0x0015: Invalid mailbox configuration
	0x0016: Invalid mailbox configuration
	0x0017: Invalid sync manager configuration
	0x0018: No valid inputs available
	0x0019: No valid outputs available
	0x001A: Synchronisation error
	0x001B: Sync manager watchdog
	0x001C: Invalid sync manager types
	0x001D: Invalid output configuration
	0x001E: Invalid input configuration
	0x001F: Invalid watchdog configuration
	0x0020: Slave needs cold start
	0x0021: Slave needs INIT
	0x0022: Slave needs PreOp
	0x0023: Slave needs SafeOp
	0x002D: Invalid output FMMU configuration
	0x002E: Invalid input FMMU configuration
	0x0030: Invalid DC Sync configuration
	0x0031: Invalid DC Latch configuration
	0x0032: PLL error
	0x0033: Invalid DC IO error
	0x0034: Invalid DC timeout error
	0x0042: Error in acyclic data exchange Ethernet over EtherCAT
	0x0043: Error in acyclic data exchange CAN over EtherCAT
	0x0044: Error in acyclic data exchange file access over EtherCAT
	0x0045: Error in acyclic data exchange servo drive profile over EtherCAT
	0x004F: Error in acyclic data exchange vendor specific over EtherCAT
	DatID : Cause for slave status change
	0: Regular slave status change
	1: Slave failure
	2: Restoration slave
	3: Slave is in an error state
	4: Slave has unexpectedly changed its status
0xED61	EtherCAT: Diagnostics buffer CP: CoE emergency

Event ID	Description
	PK : EtherCAT station address (low byte)
	OB : EtherCAT station address (high byte)
	DatID : Error code
	ZInfo1 : Error register
	ZInfo1 : MEF-Byte1
	ZInfo2 : MEF-Byte2
	ZInfo2 : MEF-Byte3
	ZInfo3 : MEF-Byte4
	ZInfo3 : MEF-Byte5
0xED62	EtherCAT: Diagnostics buffer CP: Error on SDO access
	PK : EtherCAT station address (low byte)
	OB : EtherCAT station address (high byte)
	DatID : Subindex
	ZInfo1 : Index
	ZInfo2 : SDOErrorCode (high word)
	ZInfo3 : SDOErrorCode (low word)
0xED63	EtherCAT: Diagnostics buffer CP: Error in the response to an INIT command
	PK : EtherCAT station address (low byte)
	OB : EtherCAT station address (high byte)
	ZInfo1 : Error type
	1: No response
	2: Validation error
	3: INIT command failed, requested station could not be reached
	0: Not defined
0xED70	EtherCAT: Diagnostics buffer CP: Twice HotConnect group found
	OB : Operation mode
	0: Configuration in operation mode RUN
	1: STOP (update)
	2: STOP (overall reset)
	3: STOP (own initialization)
	4: STOP (internal)
	5: Start-up (cold start)
	6: Start-up (cold restart/warm start)
	7: Start-up (restart)
	8: RUN
	9: RUN (redundant operation)
	10: HALT
	11: COUPLING

Event ID	Description
	12: UPDATING
	13: DEFECTIVE
	14: Troubleshooting
	15: Without power
	0xFD: Process image enabled in STOP
	0xFE: Watchdog
	0xFF: Not set
	ZInfo1 : Diagnostics address of the master
	ZInfo2 : EtherCAT station address
0xEE00	Additional information at UNDEF_OPCODE
	ZInfo1 : Not relevant to the user
	ZInfo2 : Not relevant to the user
	ZInfo3 : Not relevant to the user
	OB : Not relevant to the user
	DatID : Not relevant to the user
0xEE01	Internal error - Please contact the hotline!
0xEEEE	CPU was completely overall reset, since after PowerON the start-up could not be finished
0xEF00	Internal error - Please contact the hotline!
0xEF01	Internal error - Please contact the hotline!
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!
0xEFFF	Internal error - Please contact the hotline!

B Integrated blocks

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 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 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 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	Name	Description
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

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 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	Name	Description
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 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