

# VIPA System SLIO

**CPU | 015-CEFPR00 | Manual**

HB300 | CPU | 015-CEFPR00 | GB | 14-04

VIPA GmbH  
Ohmstr. 4  
91074 Herzogenaurach  
Telephone: 09132-744-0  
Fax: 09132-744-1864  
email: [info@vipa.com](mailto:info@vipa.com)  
Internet: [www.vipa.com](http://www.vipa.com)

## Table of contents

<b>1</b>	<b>General</b> .....	<b>6</b>
1.1	Copyright © VIPA GmbH .....	6
1.2	About this manual.....	7
1.2.1	Safety information.....	8
<b>2</b>	<b>Basics and Assembly</b> .....	<b>10</b>
2.1	Safety Information for Users.....	10
2.2	System conception.....	11
2.3	Dimensions.....	14
2.4	Installation.....	15
2.5	Demounting and module exchange.....	20
2.6	Wiring.....	24
2.7	Trouble shooting - LEDs.....	28
2.8	Installation guidelines.....	28
2.9	General data.....	31
<b>3</b>	<b>Hardware description</b> .....	<b>33</b>
3.1	Properties.....	33
3.2	Structure.....	34
3.2.1	Basic CPU.....	34
3.2.2	Interfaces.....	35
3.2.3	Memory management.....	37
3.2.4	Slot for storage media.....	37
3.2.5	Buffering mechanisms.....	37
3.2.6	Operating mode switch.....	38
3.2.7	LEDs.....	38
3.3	Technical Data.....	42
<b>4</b>	<b>Deployment CPU 015</b> .....	<b>50</b>
4.1	Assembly.....	50
4.2	Start-up behaviour.....	50
4.3	Addressing.....	50
4.3.1	Overview.....	50
4.3.2	Addressing backplane bus periphery.....	51
4.4	Hardware configuration - CPU.....	52
4.5	Hardware configuration - I/O modules.....	54
4.6	Hardware configuration - Ethernet PG/OP channel.....	55
4.7	Setting standard CPU parameters.....	57
4.7.1	Parametrization via Siemens CPU.....	57
4.7.2	Parameter CPU.....	57
4.7.3	Parameter for MPI/DP .....	60
4.8	Setting VIPA specific CPU parameters.....	60
4.9	Project transfer.....	62
4.9.1	Transfer via MPI / optional PROFIBUS.....	62
4.9.2	Transfer via Ethernet.....	64
4.9.3	Transfer via memory card.....	65
4.10	Accessing the web server.....	66
4.11	Operating modes.....	69
4.11.1	Overview.....	69
4.11.2	Function security.....	71

4.12	Overall reset.....	71
4.13	Firmware update.....	73
4.14	Factory reset.....	74
4.15	Deployment storage media - VSD, VSC.....	75
4.16	Extended know-how protection.....	77
4.17	Cmd - Auto commands.....	78
4.18	VIPA specific diagnostic entries.....	80
4.19	Control and monitoring of variables with test functions..	95
<b>5</b>	<b>Deployment PtP communication.....</b>	<b>97</b>
5.1	Fast introduction.....	97
5.2	Principle of the data transfer.....	98
5.3	Deployment of RS485 interface for PtP.....	98
5.4	Parameterization.....	99
5.4.1	FC/SFC 216 - SER_CFG.....	99
5.5	Communication.....	102
5.5.1	Overview.....	102
5.5.2	FC/SFC 217 - SER_SND.....	103
5.5.3	FC/SFC 218 - SER_RCV.....	107
5.6	Protocols and procedures.....	110
5.7	Modbus - Function codes .....	114
<b>6</b>	<b>Option: PROFIBUS communication.....</b>	<b>120</b>
6.1	Overview.....	120
6.2	Fast introduction.....	120
6.3	Enable bus functionality via VSC.....	121
6.4	Hardware configuration - CPU.....	121
6.5	Deployment as PROFIBUS DP master.....	122
6.6	PROFIBUS installation guidelines.....	123
6.7	Commissioning and Start-up behaviour.....	127
<b>7</b>	<b>Deployment Ethernet communication - productive.....</b>	<b>128</b>
7.1	Basics - Industrial Ethernet in automation.....	128
7.2	Basics - ISO/OSI reference model.....	129
7.3	Basics - Terms.....	130
7.4	Basics - Protocols.....	131
7.5	Basics - IP address and subnet.....	133
7.6	Fast introduction.....	134
7.7	Commissioning and Initialization.....	135
7.8	Hardware configuration - CPU.....	52
7.9	Configure Siemens S7 connections.....	138
7.10	Configure Open Communication.....	144
7.11	NCM diagnostic - Help for error diagnostic.....	146
<b>8</b>	<b>Deployment Ethernet communication - PROFINET.....</b>	<b>149</b>
8.1	Basics PROFINET.....	149
8.2	PROFINET installation guidelines.....	151
8.3	Fast introduction.....	152
8.4	Commissioning and Initialization.....	135
8.5	Hardware configuration - CPU.....	52
8.6	Parameters - PROFINET IO controller.....	155
8.6.1	Precondition.....	155
8.6.2	PN-IO.....	156



---

8.6.3	Port 1.....	157
8.7	Configuration PROFINET IO device.....	157
8.8	Commissioning and start-up behaviour.....	158
8.9	PROFINET diagnostics.....	159
8.9.1	Overview.....	159
8.9.2	Diagnostics with the configuration and engineering tool.....	159
8.9.3	Diagnostics during runtime in the user program.....	160
8.9.4	Diagnostics via OB start information.....	162
8.9.5	Diagnostics via status LEDs.....	162
<b>9</b>	<b>Configuration with TIA Portal.....</b>	<b>164</b>
9.1	TIA Portal - Work environment .....	164
9.1.1	General.....	164
9.1.2	Work environment of the TIA Portal.....	164
9.2	TIA Portal - Hardware configuration - CPU .....	165
9.3	TIA Portal - Hardware configuration - Ethernet PG/OP channel.....	168
9.4	Hardware Configuration - I/O modules.....	170
9.5	TIA Portal - Include VIPA library.....	171
9.6	TIA Portal - Project transfer.....	172

# 1 General

## 1.1 Copyright © VIPA GmbH

### All Rights Reserved

This document contains proprietary information of VIPA and is not to be disclosed or used except in accordance with applicable agreements.

This material is protected by the copyright laws. It may not be reproduced, distributed, or altered in any fashion by any entity (either internal or external to VIPA), except in accordance with applicable agreements, contracts or licensing, without the express written consent of VIPA and the business management owner of the material.

For permission to reproduce or distribute, please contact: VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH Ohmstraße 4, D-91074 Herzogenaurach, Germany

Tel.: +49 9132 744 -0

Fax.: +49 9132 744-1864

E-Mail: [info@vipa.de](mailto:info@vipa.de)

<http://www.vipa.com>



*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

E-Mail: [documentation@vipa.de](mailto: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)

E-Mail: [support@vipa.de](mailto:support@vipa.de)

**1.2 About this manual****Objective and contents**

This manual describes the CPU 015 of the System SLIO from VIPA. It contains a description of the construction, project implementation and usage.

Product	Order no.	as of state:		
		CPU HW	CPU FW	CP FW
Basic CPU 015	015-CEFPR00	01	V1.0.0	V1.0.0

**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.*

### 1.2.1 Safety information

#### Applications conforming with specifications

The system is constructed and produced for:

- communication and process control
- 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 Assembly

### 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.



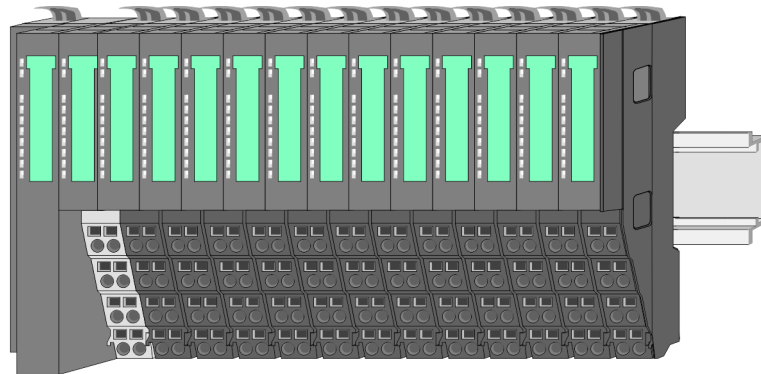
#### CAUTION!

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

## 2.2 System conception

### Overview

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



### Components

- CPU (head module)
- Bus coupler (head module)
- Periphery modules
- Power modules
- Accessories



#### CAUTION!

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

### CPU



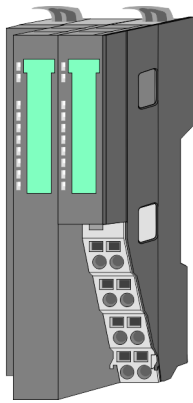
With a CPU, CPU electronic and power module are integrated to one casing. As head module via the integrated power module for power supply the CPU electronic is supplied as well as the electronic of the connected periphery modules. The DC 24 power section supply for the linked periphery modules is established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



#### CAUTION!

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

**Bus coupler**



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

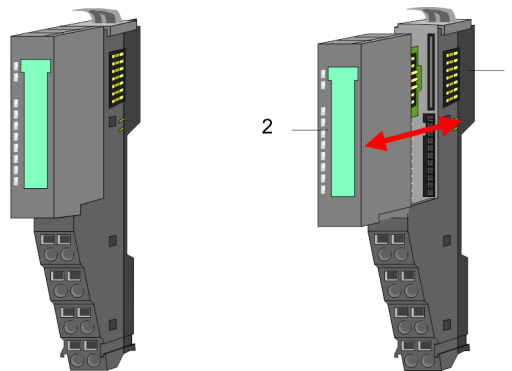


**CAUTION!**

Bus interface and power module of the bus coupler may not be separated! Here you may only exchange the electronic module!

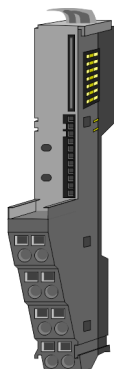
**Periphery modules**

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



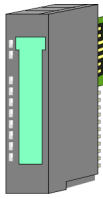
- 1 Terminal module
- 2 Electronic module

**Terminal module**



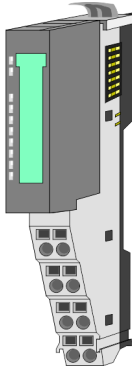
The *terminal module* serves to carry the electronic module, contains the backplane bus with power supply for the electronic, the DC 24V power section supply and the staircase-shaped terminal for wiring. Additionally the terminal module has a locking system for fixing at a mounting rail. By means of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



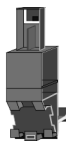
**Electronic module**

The functionality of a SLIO peripheral module is defined by the *electronic module*, which is mounted to the terminal module by a safe sliding mechanism. With an error the defective module may be exchanged for a functional module with standing installation.

At the front side there are LEDs for status indication. For simple wiring each module shows a corresponding connection diagram at the front and at the side.

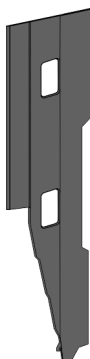
**Power module**

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

**Accessories****Shield bus carrier**

The shield bus carrier (order no. 000-0AB00) serves to carry the shield bus (10mm x 3mm) to connect cable shields. Shield bus carriers, shield bus and shield fixings are not in the scope of delivery. They are only available as accessories.

The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaption to a flat mounting rail you may remove the spacer of the shield bus carrier.

**Bus cover**

With each bus coupler, to protect the backplane bus connectors, there is a mounted bus cover in the scope of delivery. You have to remove the bus cover of the bus coupler before mounting a SLIO module. For the protection of the backplane bus connector you always have to mount the bus cover at the last module of your system again.

The bus cover has the order no. 000-0AA00.

**Coding pins**



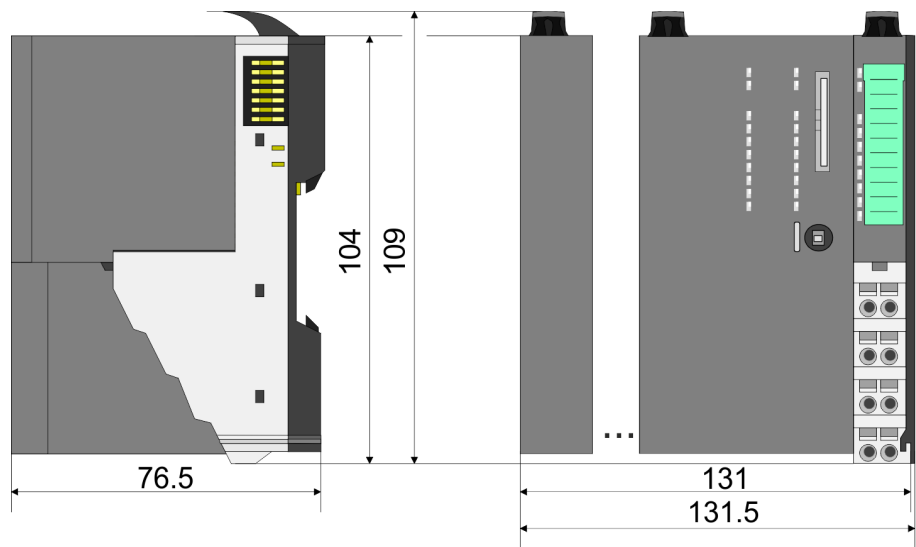
There is the possibility to fix the assignment of electronic and terminal module. Here coding pins (order number 000-0AC00) from VIPA can be used.

The coding pin consists of a coding jack and a coding plug. By combining electronic and terminal module with coding pin, the coding jack remains in the electronic module and the coding plug in the terminal module.

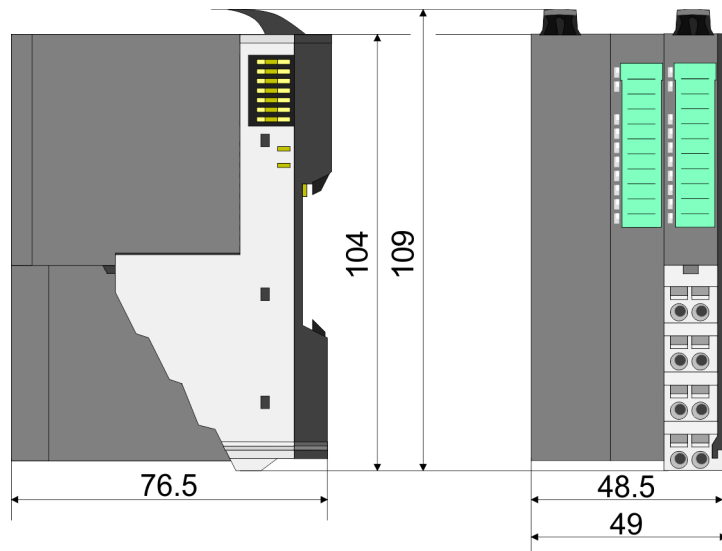
This ensures that after replacing the electronics module just another electronic module can be plugged with the same encoding.

**2.3 Dimensions**

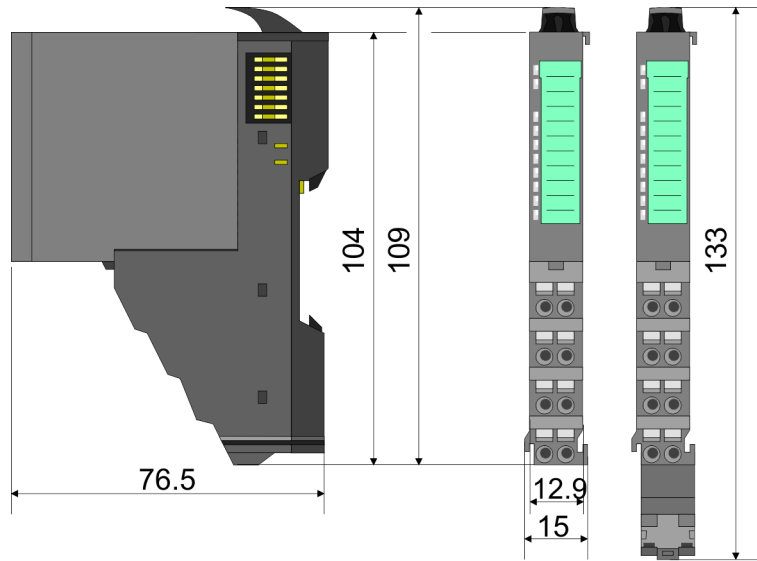
**Dimensions CPU**



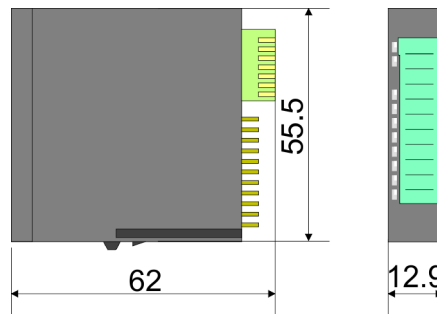
**Dimensions bus coupler**



**Dimensions periphery module**



**Dimensions electronic module**

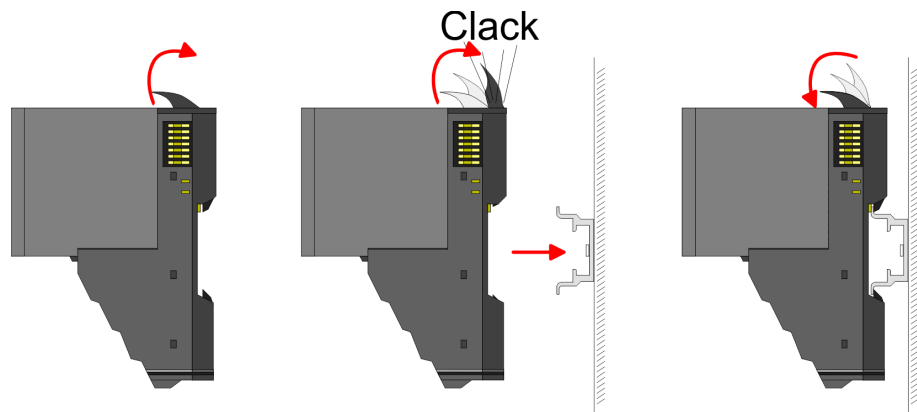


Dimensions in mm

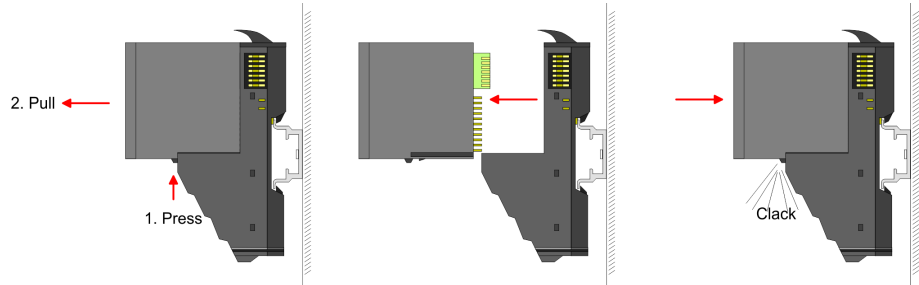
**2.4 Installation**

**Functional principle**

There is a locking lever at the top side of the terminal module. For mounting and demounting this locking lever is to be turned upwards until this engages audible. Now the module may be pulled forward. For mounting plug the module to the module installed before and push the module to the mounting rail guided by the strips at the upper and lower side of the module. The module is fixed to the mounting rail by pushing downward the locking lever. The modules may either separately be mounted to the mounting rail or as block. Here is to be considered that each locking lever is opened.



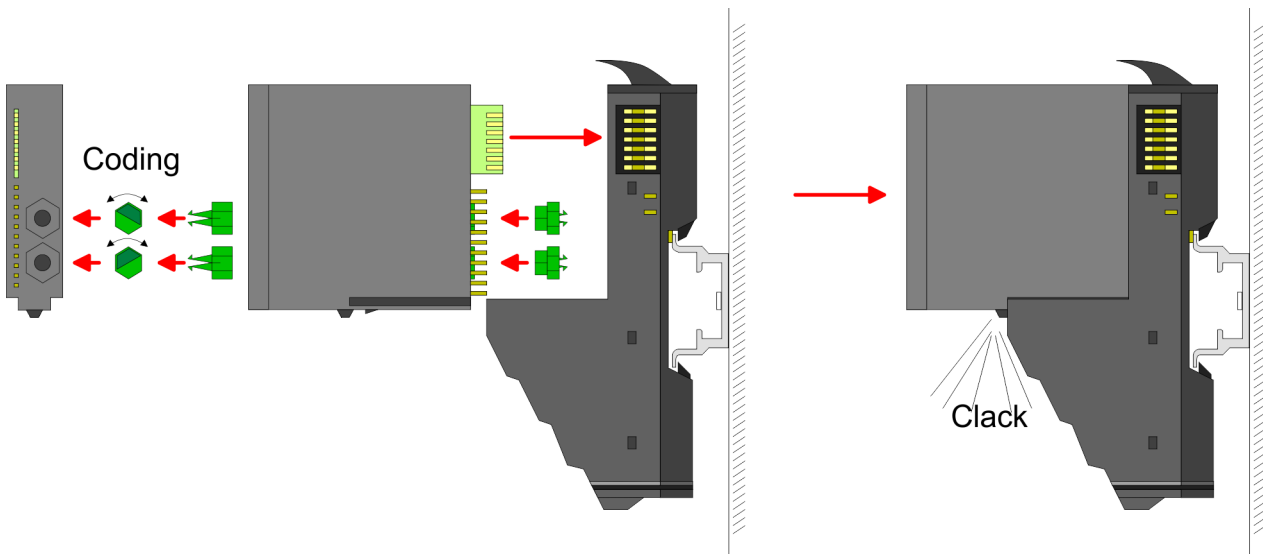
For the exchange of a electronic module, the electronic module may be pulled forward after pressing the unlocking lever at the lower side of the module. For installation plug the electronic module guided by the strips at the lower side until this engages audible to the terminal module.



**Coding**



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



Each electronic module has on its back 2 coding sockets for coding jacks. Due to the characteristics, with the coding jack 6 different positions can be plugged, each. Thus there are 36 possible combinations for coding with the use of both coding sockets.

1. ► Plug, according to your coding, 2 coding jacks in the coding sockets of your electronic module until they lock.
2. ► Now plug the according coding plugs into the coding jacks.
3. ► To fix the coding put both the electronic and terminal module together until they lock.

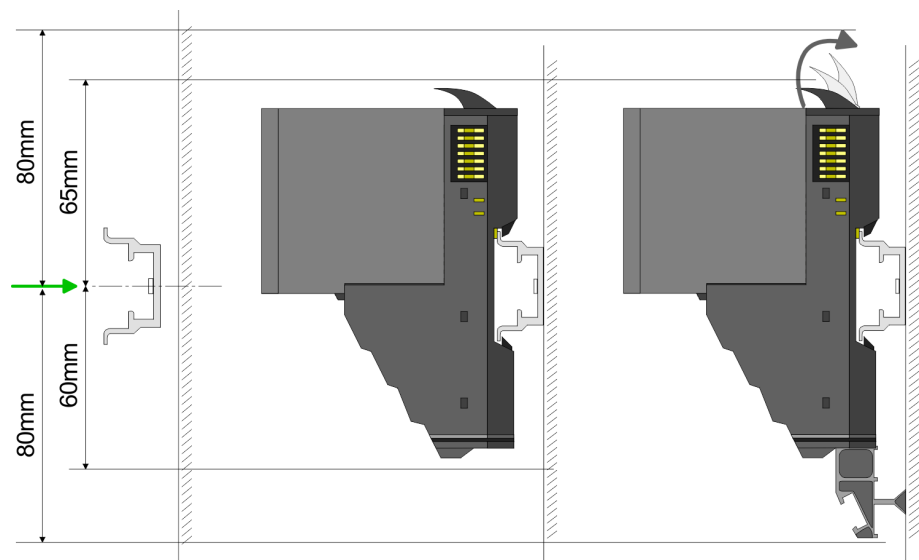
**CAUTION!**

Please consider that when replacing an already coded electronic module, this is always be replaced by an electronic module with the same coding.

Even with an existing coding on the terminal module, you can plug an electronic module without coding. The user is responsible for the correct usage of the coding pins. VIPA assumes no liability for incorrectly attached electronic modules or for damages which arise due to incorrect coding!

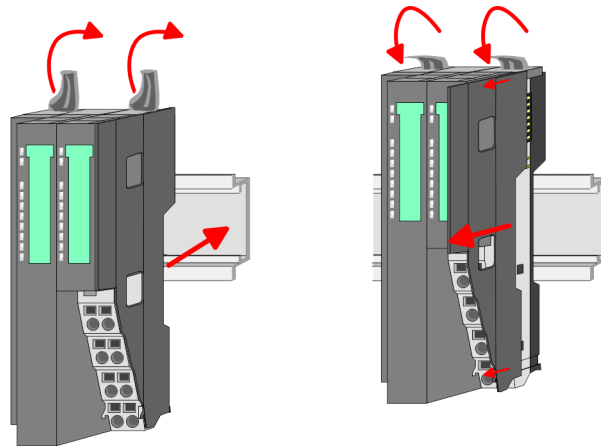
**Mounting Proceeding**

The modules were directly be mounted to the mounting rail and so connected to the backplane bus and the power supply for the electronic and power section. Up to 64 modules may be mounted. Please consider here that the sum current of the electronic power supply does not exceed the maximum value of 3A. By means of the power module 007-1AB10 the current of the electronic power supply may be expanded with 2A. ↪ *Chapter 2.6 'Wiring' on page 24*

**Mounting rail**

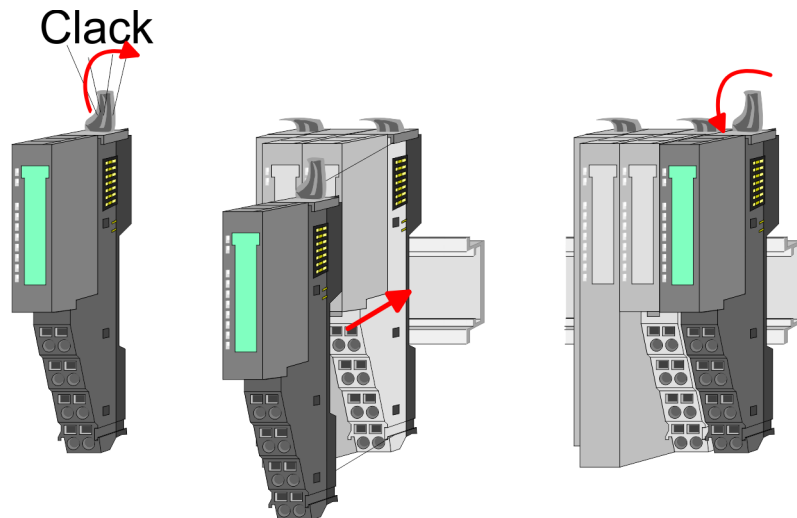
- ➔ Mount the mounting rail! Please consider that a clearance from the middle of the mounting rail of at least 80mm above and 60mm below, respectively 80mm by deployment of shield bus carriers, exist.

Mounting Head module (e.g. bus coupler)



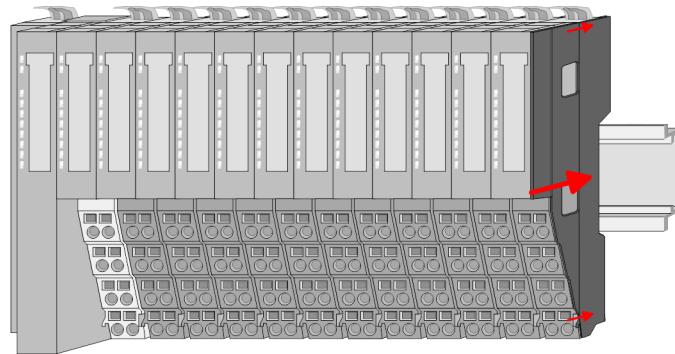
1. ▶ Start at the left side with the head module (e.g. bus coupler). For this turn both locking lever upwards, put the head module to the mounting rail and turn both locking lever downward.
2. ▶ Before mounting the periphery modules you have to remove the bus cover at the right side of the Head module by pulling it forward. Keep the cover for later mounting.

Mounting periphery modules



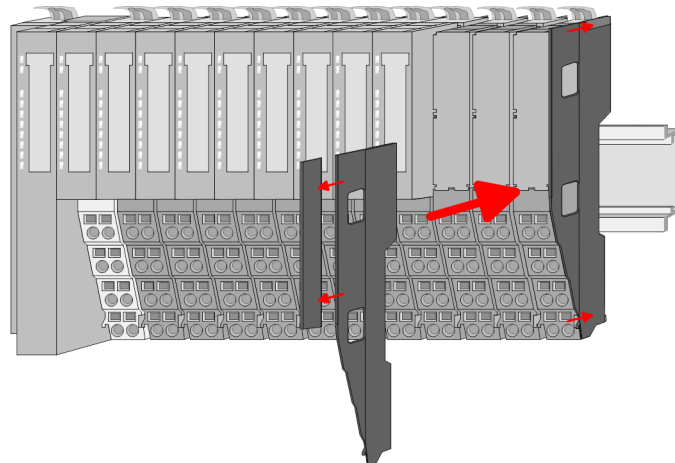
- ▶ Mount the periphery modules you want.

Mounting the bus cover



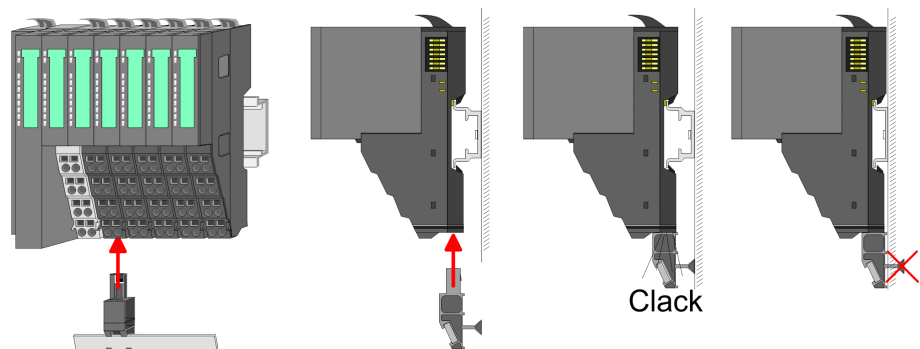
- ➔ After mounting the whole system, to protect the backplane bus connectors at the last module you have to mount the bus cover, now.

Mounting the bus cover at a clamp module



- ➔ If the last module is a clamp module, for adaptation the upper part of the bus cover is to be removed

Mounting shield bus carrier



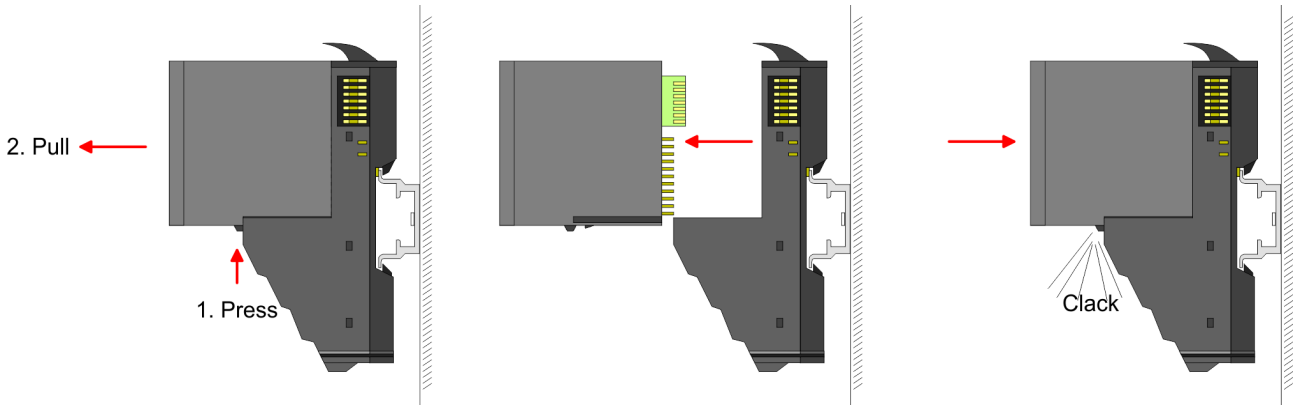
- ➔ The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields. The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaption to a flat mounting rail you may remove the spacer of the shield bus carrier.

## 2.5 Demounting and module exchange

### Proceeding

With demounting and exchange of a module, head module (e.g. bus coupler) or a group of modules for mounting reasons you have always to remove the electronic module of the just mounted right module. After the mounting it may be plugged again.

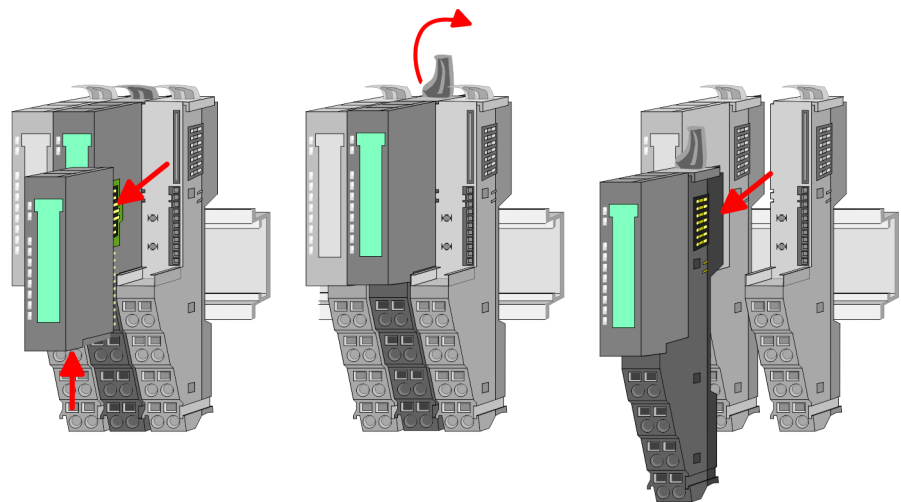
#### Exchange of an electronic module



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

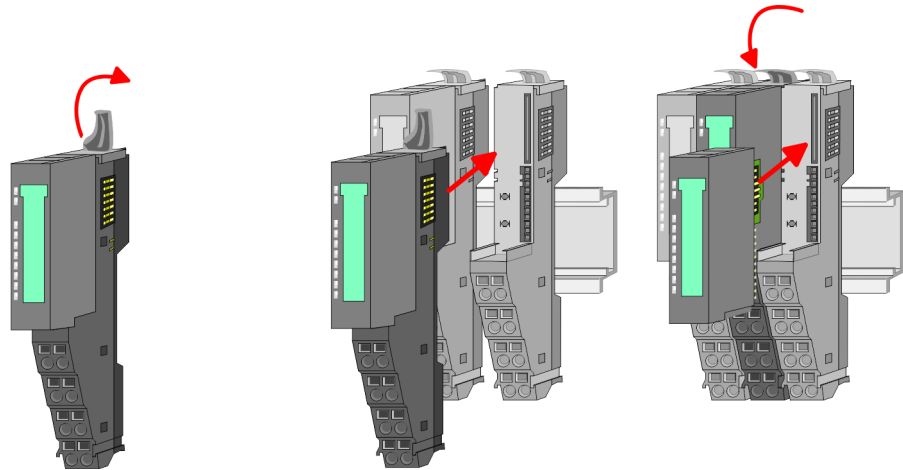
#### Exchange of a module

1. ▶ Remove if exists the wiring. ↪ *Chapter 2.6 'Wiring' on page 24.*



2. ▶ Press the unlocking lever at the lower side of the just mounted right module and pull it forward.
3. ▶ Turn the locking lever of the module to be exchanged upwards.
4. ▶ Pull the module forward.





5. ▶ For mounting turn the locking lever of the module to be mounted upwards.
6. ▶ To mount the module put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.
7. ▶ Turn the locking lever downward again.
8. ▶ Plug again the electronic module, which you have removed before.

Exchange of a head module (e.g. bus coupler)

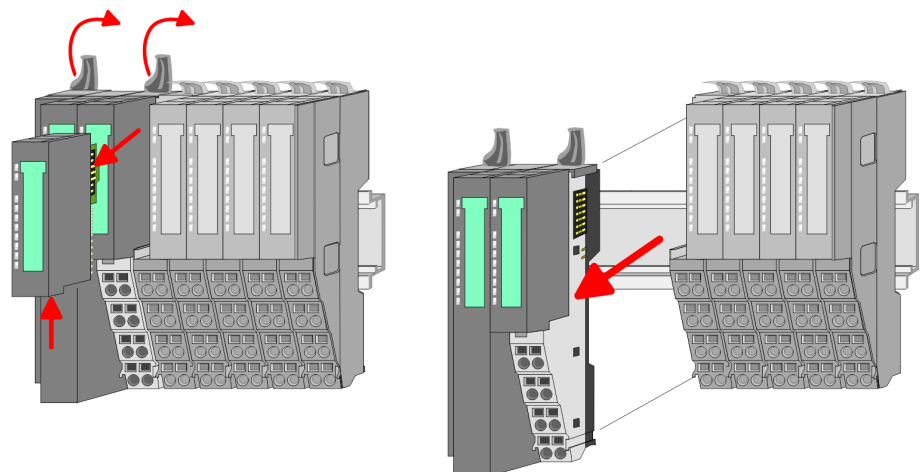


**CAUTION!**

Bus interface and power module of a head module may not be separated!

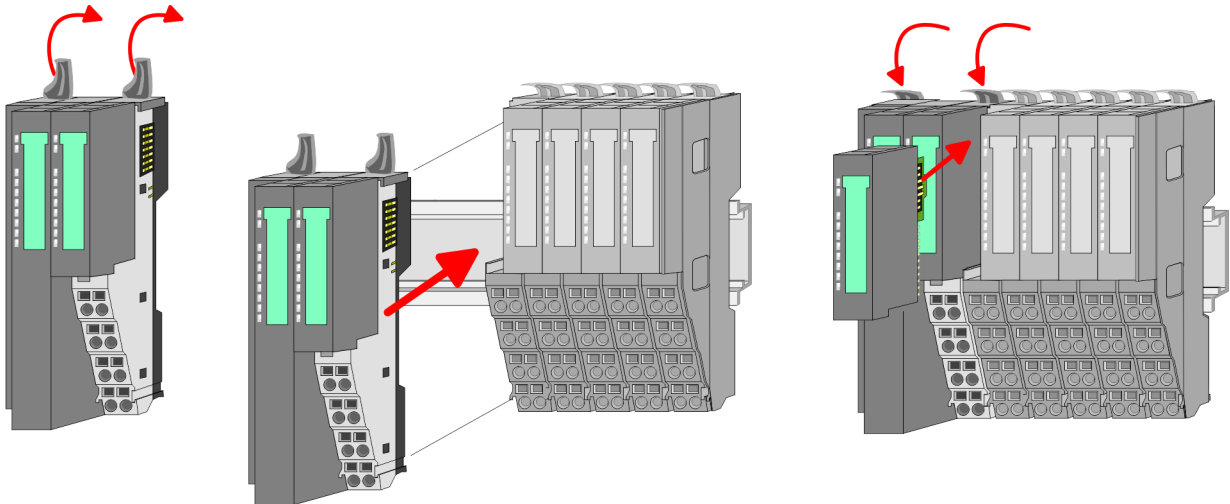
Here you may only exchange the electronic module!

1. ▶ Remove if exists the wiring of the head module. ↪ *Chapter 2.6 'Wiring' on page 24.*



2. ▶ Press the unlocking lever at the lower side of the just mounted right module and pull it forward.
3. ▶ Turn all the locking lever of the head module to be exchanged upwards.
4. ▶ Pull the head module forward.

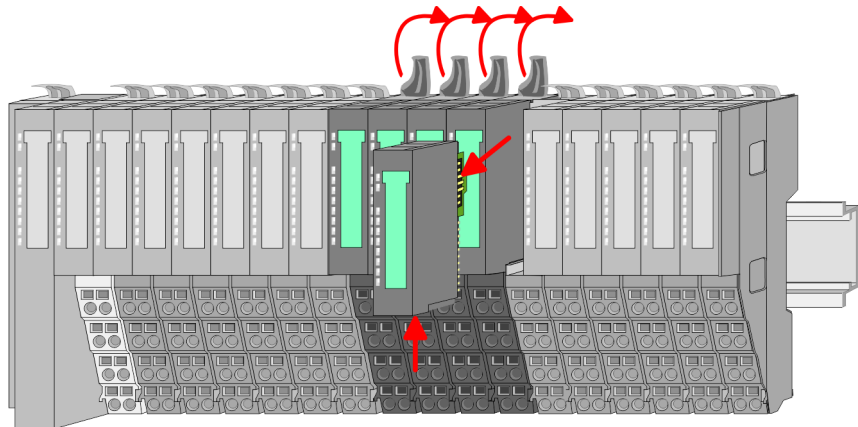
Demounting and module exchange



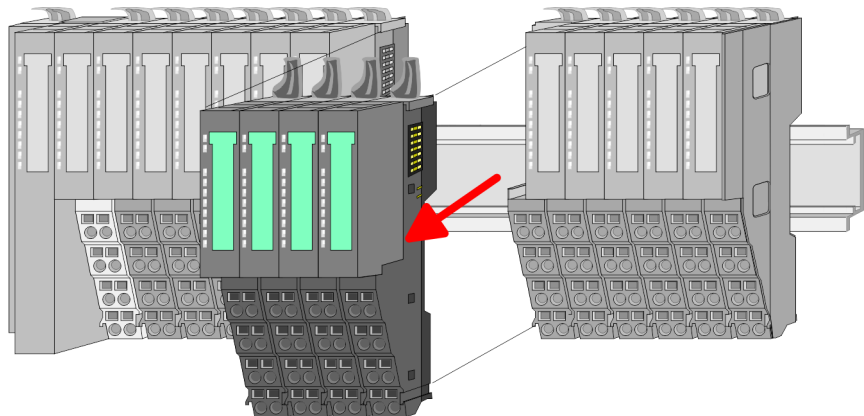
5. ▶ For mounting turn all the locking lever of the head module to be mounted upwards.
6. ▶ To mount the head module put it to the left module and push it, guided by the stripes, to the mounting rail.
7. ▶ Turn all the locking lever downward again.
8. ▶ Plug again the electronic module, which you have removed before.

Exchange of a module group

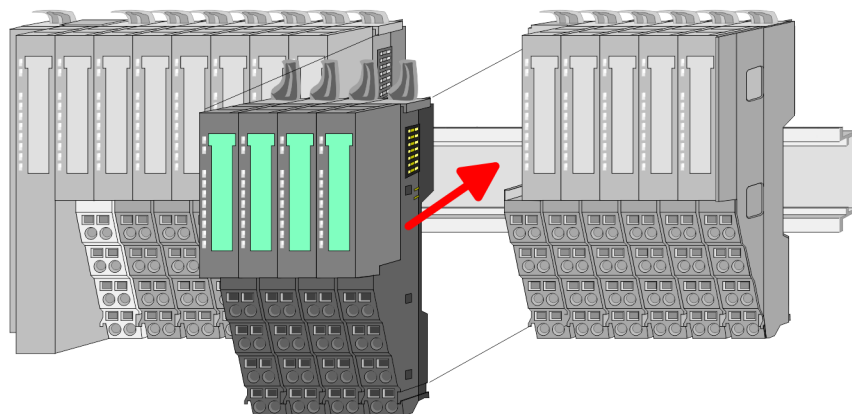
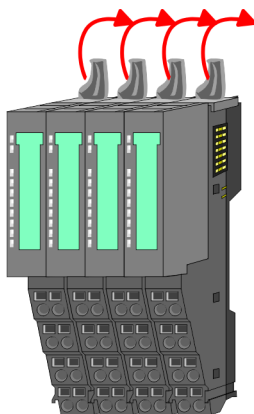
1. ▶ Remove if exists the wiring of the module group. ↪ *Chapter 2.6 'Wiring' on page 24.*



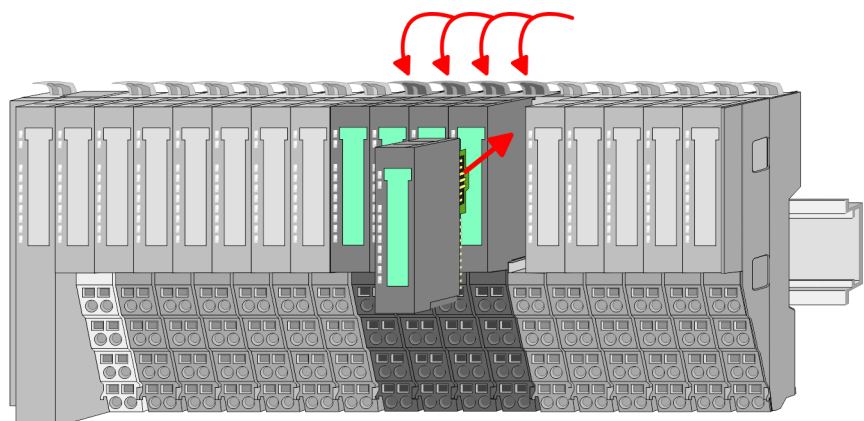
2. ▶ Press the unlocking lever at the lower side of the just mounted right module of the module group and pull it forward.



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



5. For mounting turn all the locking lever of the module group to be mounted upwards.
6. To mount the module group put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.



7. Turn all the locking lever downward again.
8. Plug again the electronic module, which you have removed before.

## 2.6 Wiring

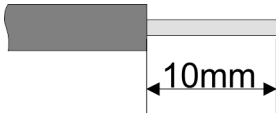
### Connectors

Terminals with spring clamp technology are used for wiring.

The spring clamp technology allows quick and easy connection of your signal and supply lines.

In contrast to screw terminal connections this type of connection is vibration proof.

### Data



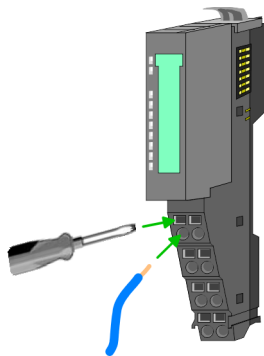
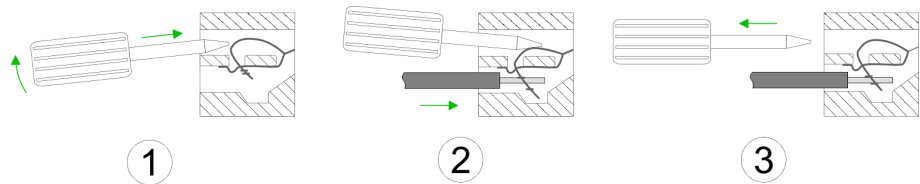
$U_{max}$ : 240V AC / 30V DC

$I_{max}$ : 10A

Cross section: 0.08 ... 1.5mm<sup>2</sup> (AWG 28 ... 16)

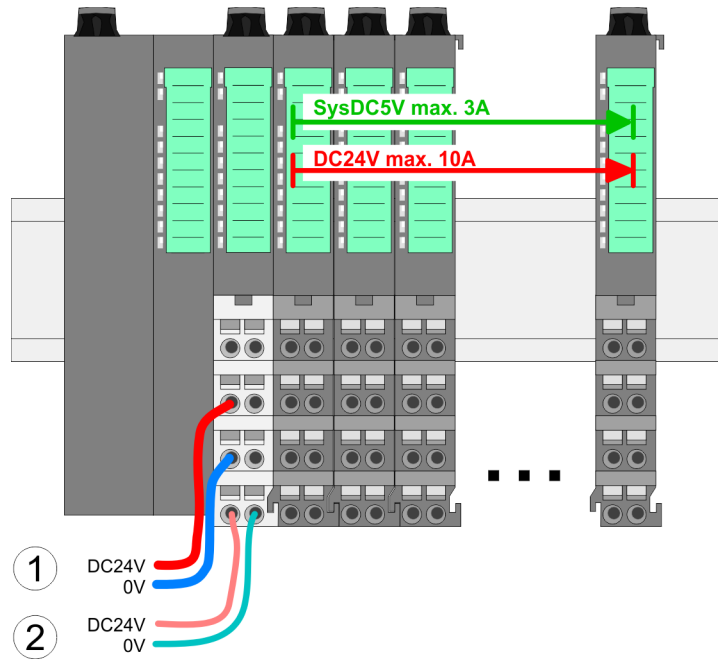
Stripping length: 10mm

### Wiring procedure



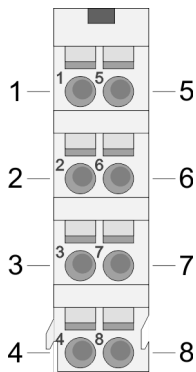
1. ➤ Insert a suited screwdriver at an angle into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. ➤ Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm<sup>2</sup> to 1.5mm<sup>2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

Standard wiring



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

PM - Power module



For wires with a core cross-section of 0.08mm<sup>2</sup> up to 1.5mm<sup>2</sup>.

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

I Input



**CAUTION!**

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



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

**Fusing**

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

**State of the electronic power supply via LEDs**

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A.

With a sum current greater than 3A the LEDs may not be activated.

Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

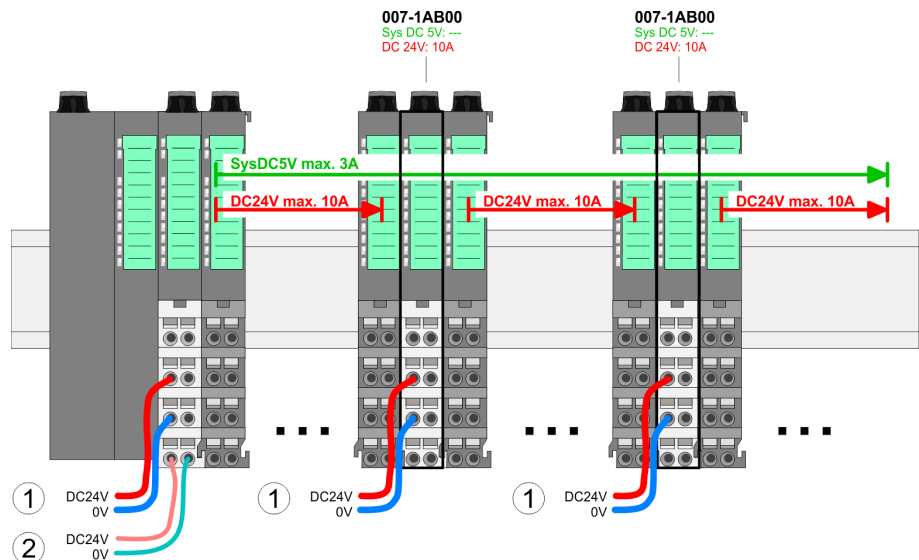
**Deployment of the power modules**

If the 10A for the power section supply is no longer sufficient, you may use the power module from VIPA with the order number 007-1AB00. So you have also the possibility to define isolated groups.

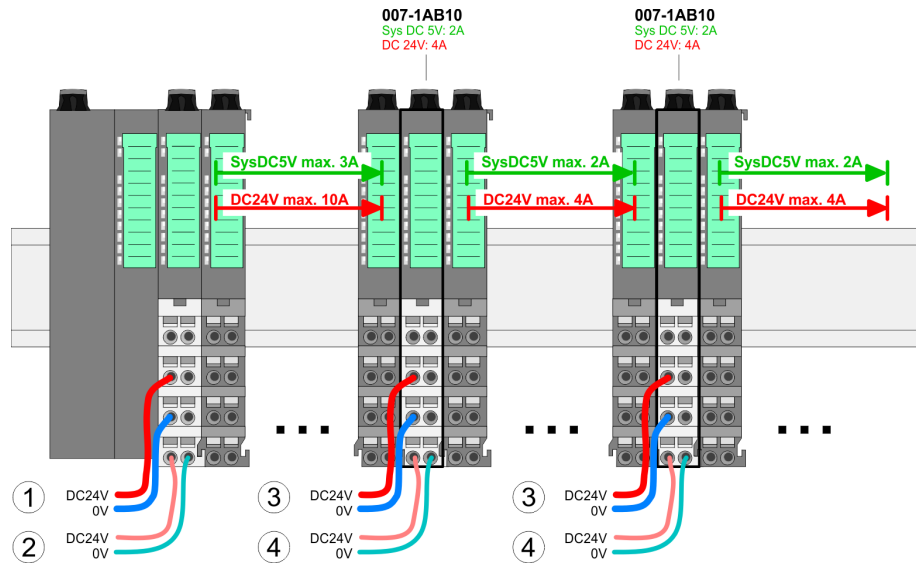
The power module with the order number 007-1AB10 is to be used if the 3A for the electronic power supply at the backplane bus is no longer sufficient. Additionally you get an isolated group for the DC 24V power section supply with 4A.

By placing the power module 007-1AB10 at the following backplane bus modules may be placed with a sum current of max. 2A. Afterwards the power module 007-1AB10 is to be placed again. To secure the power supply, the power modules may be mixed used.

**Power module 007-1AB00**



**Power module  
007-1AB10**



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

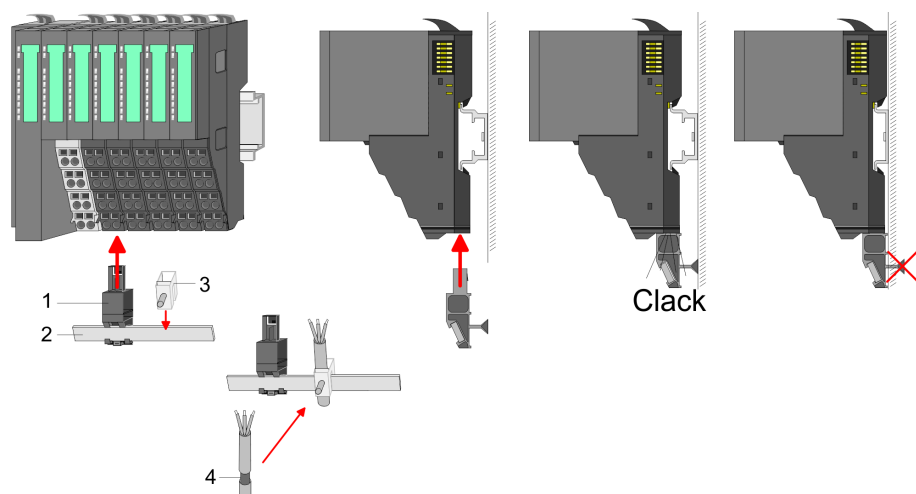
**Shield attachment**

To attach the shield the mounting of shield bus carriers are necessary.

The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaption to a flat mounting rail you may remove the spacer of the shield bus carrier.

After mounting the shield bus carrier with the shield bus, the cables with the accordingly stripped cable screen may be attached and fixed by the shield clamp.



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

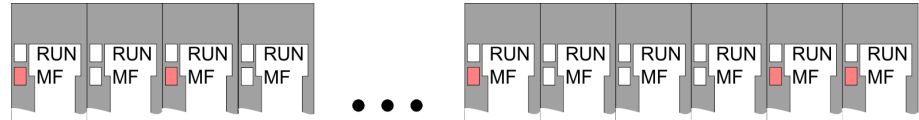
## 2.7 Trouble shooting - LEDs

### General

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

In the following illustrations flashing LEDs are marked by ☼.

### Sum current of the electronic power supply exceeded



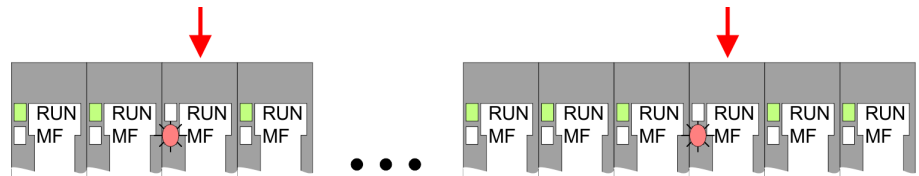
**Behaviour:** After PowerON the RUN LED of each module is off and the MF LED of each module is sporadically on.

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

**Remedy:** As soon as the sum current of the electronic power supply is exceeded, always place the power module 007-1AB10.

☞ Chapter 2.6 'Wiring' on page 24.

### Error in configuration

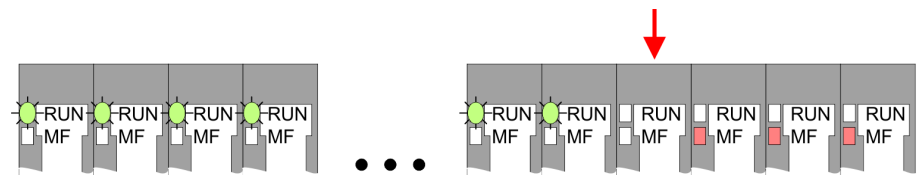


**Behaviour:** After PowerON the MF LED of one module respectively more modules flashes. The RUN LED remains off.

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

**Remedy:** Match configuration and hardware structure.

### Module failure



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

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

**Remedy:** Replace the defective module.

## 2.8 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 digestibility (EMC), and how you manage the isolation.



- What does EMC mean?** Electromagnetic digestibility (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  $\mu\text{A}$ ) are transferred.
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metallised plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.
- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to your PLC and don't lay it on there again!



#### **CAUTION!**

#### **Please regard at installation!**

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

Remedy: Potential compensation line

## 2.9 General data

### Conformity and approval

Conformity		
CE	2006/95/EG	Low-voltage directive
	2004/108/EG	EMC directive
Approval		
UL	UL 508	Approval for USA and Canada
others		
RoHS	2011/65/EU	Product is lead-free; 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	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
Mechanical		
Oscillation	EN 60068-2-6	1g, 9Hz ... 150Hz
Shock	EN 60068-2-27	15g, 11ms

General data

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 irradiation (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, installation class 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

##### CPU 015

- SPEED7 technology integrated
- Programmable via SPEED7 Studio, Siemens SIMATIC Manager or TIA Portal
- 512kbyte load memory integrated (256kbyte code, 256kbyte data)
- 256kbyte work memory integrated (128kbyte code, 128kbyte data), expandable up to 512kbyte
- Slot for external storage media (lockable)
- Status LEDs for operating state and diagnostics
- RJ45 interface: Ethernet PG/OP interface integrated
- RJ45 interface: PROFINET IO controller with integrated Ethernet CP, PROFINET according conformance class A
- RS485 interface: switchable PtP(MPI): Serial integrated interface for PtP communication with the protocols: ASCII, STX/ETX , USS, 3964(R), MODBUS RTU, master/slave
- RS485 interface: MPI(DP) field bus functionality unlock able via VSC
- up to 64 SLIO modules placeable
- I/O address area digital/analog 8191byte
- 512 timer/counter, 8192 flag byte



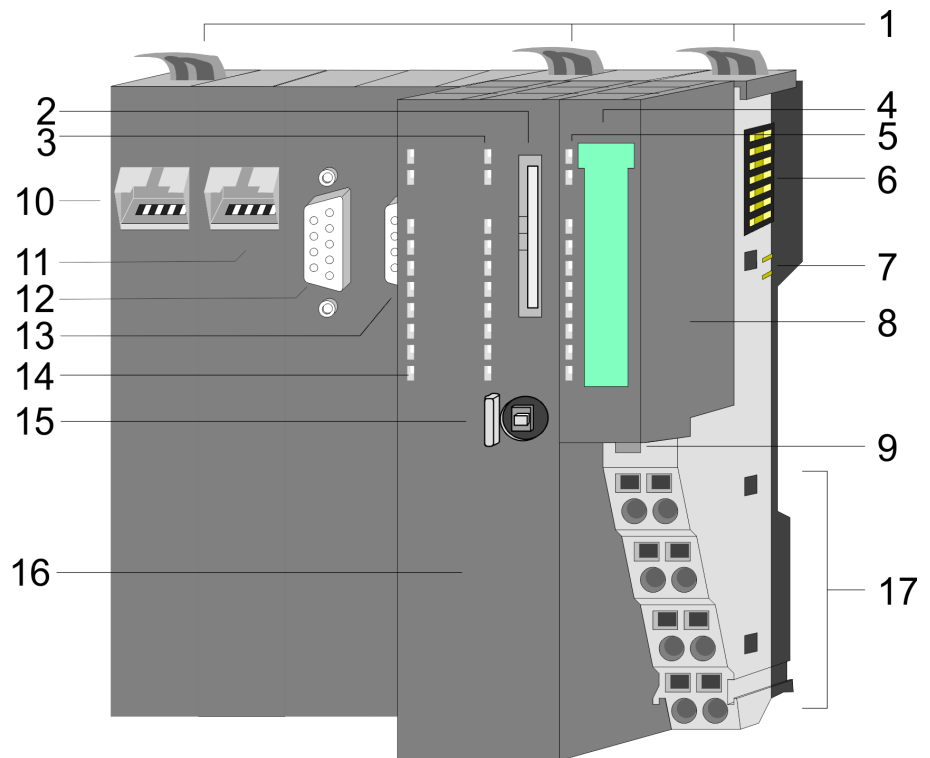
#### Ordering data

Type	Order number	Description
CPU 015	015-CEFPR00	Basic CPU 015 with PROFINET IO controller and options to extend work memory and bus interface.

## 3.2 Structure

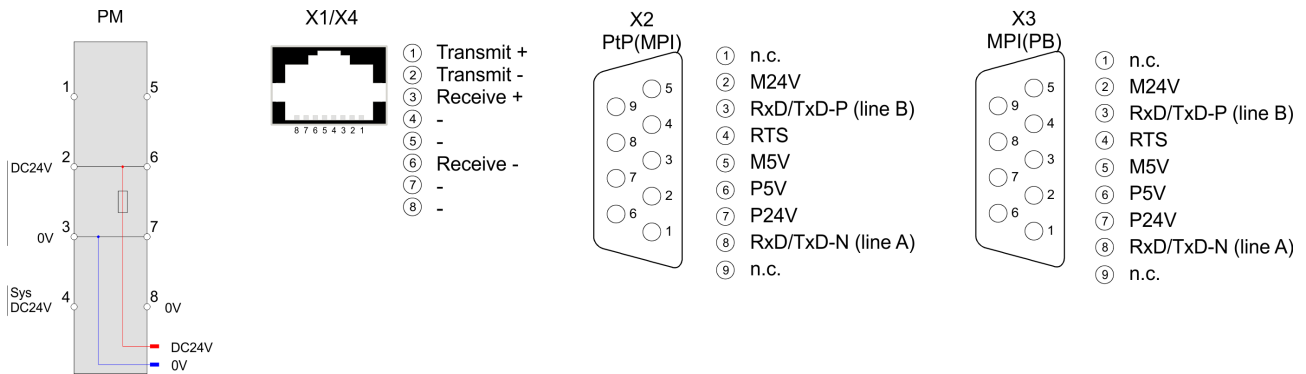
### 3.2.1 Basic CPU


#### CPU 015



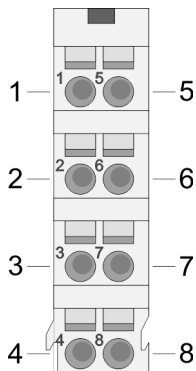
- 1 Locking lever terminal module
- 2 Slot for external storage media (lockable)
- 3 LED status indication CPU part
- 4 Labelling strip power module
- 5 LED status indication power module
- 6 Backplane bus
- 7 DC 24V power section supply
- 8 Power module
- 9 Unlocking lever power module
- 10 Twisted pair interface for PROFINET
- 11 Twisted pair interface for Ethernet PG/OP channel
- 12 PtP(MPI) RS485 interface
- 13 MPI(PB) RS485 interface
- 14 LED status indication PROFINET IO controller
- 15 Operating mode switch CPU
- 16 CPU part
- 17 Terminal power module

### 3.2.2 Interfaces



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

#### PM - Power module



For wires with a core cross-section of 0.08mm<sup>2</sup> up to 1.5mm<sup>2</sup>.

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

I Input

#### Ethernet PG/OP channel X1

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 not possible.
- For online access to the CPU via Ethernet PG/OP channel valid IP address parameters have to be assigned to this.

🔗 Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 55

**PROFINET IO controller X4***8pin RJ45 jack:*

- The CPU has an PROFINET IO controller integrated, which is to be configured by means of the PROFINET sub module in the hardware configurator of Siemens.
- The PROFINET IO controller serves the following possibilities for connections:
  - PROFINET IO controller to connect PROFINET IO devices
  - Ethernet PG/OP channel
  - Ethernet Siemens S7 connections
  - Ethernet open communication

**PtP(MPI) interface X2**

*9pin SubD jack: (isolated):* The interface supports the following functions, which are switch able via the *VIPA specific CPU parameters* ↪ *Chapter 4.8 'Setting VIPA specific CPU parameters' on page 60:*

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



*Please consider the MPI interface X2 can only be configured, if X3 is configured as PROFIBUS.*

**MPI(PB) interface X3**

*9pin SubD jack: (isolated):* The interface supports the following functions, which are switch able via the sub module X1 '*MPI/DP*' in the hardware configuration:

- MPI (default / after reset to factory setting ↪ *Chapter 4.14 'Factory reset' on page 74*)  
Per default, the RS485 interface is set to MPI functionality. The MPI interface serves for the connection between programming unit and CPU. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU. Standard setting is MPI address 2.
- PB  
The PROFIBUS master functionality of this interface can be activated by configuring the sub module X1 '*MPI/DP*' of the CPU in the hardware configuration.



**Enable bus functionality via VSC**

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

🔗 'Overview' on page 75

### 3.2.3 Memory management

**General**

The CPU has an integrated memory. Information about the capacity (min. capacity ... max. capacity) of the memory may be found at the front of the CPU. The memory is divided into the following 3 parts:

- Load memory (integrated/maximum) 256/512kbyte
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)

The work memory has 256kbyte. There is the possibility to extend the work memory to its maximum capacity 512kbyte by means of a VSC.

### 3.2.4 Slot for storage media

**Overview**

In this slot you can insert the following storage media:

- VSD - **VIPA SD-Card**: External memory card for programs and firmware
- VSC - **VIPASetCard**: 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 75*

To activate the corresponding card is to be installed and a *Overall reset* is to be established. 🔗 *Chapter 4.12 'Overall reset' on page 71*




A list of the currently available VSD respectively VSC can be found at [www.vipa.com](http://www.vipa.com).

### 3.2.5 Buffering mechanisms

The SLIO CPU has a capacitor-based mechanism to buffer the internal clock in case of power failure for max. 30 days.

With PowerOFF the content of the RAM is automatically stored in the Flash (NVRAM).



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

In case of failure of the buffer mechanism Date and Time 01.09.2009 00:00:00 set. Additionally, you receive a diagnostics message. ↪ *Chapter 4.18 'VIPA specific diagnostic entries' on page 80*

### 3.2.6 Operating mode switch

#### General




With the operating mode switch you may switch the CPU between STOP and RUN.

During the transition from STOP to RUN the operating mode START-UP is driven by the CPU.






Placing the switch to MR (**M**emory **R**eset), you request an overall reset with following load from memory card, if a project there exists.

### 3.2.7 LEDs

#### CPU part

PW		Meaning
green 	●	As soon as the CPU is supplied with 5V, the green PW-LED (Power) is on.
	○	The CPU is not power-supplied.

on: ● | off: ○

RN	ST	SF	FC	SD	Meaning
green 	yellow 	red 	yellow 	yellow 	

#### Boot-up after PowerON



●	BB	●	●	●	Blinking with 10Hz: Firmware is loaded.
●	●	●	●	●	Initialization: Phase 1
●	●	●	●	○	Initialization: Phase 2
●	●	●	○	○	Initialization: Phase 3
○	●	●	○	○	Initialization: Phase 4

#### Operation

○	●	X	X	X	CPU is in STOP state.
---	---	---	---	---	-----------------------

RN	ST	SF	FC	SD	Meaning
BB	○	X	X	X	CPU is in start-up state. Blinking with 2Hz: The RUN LED blinks during start-up (OB100) at least for 3s.
○	BB	X	X	X	Blinking with 10Hz: Activation of a new hardware configuration
●	○	○	X	X	CPU is in state RUN without error.
X	X	●	X	X	There is a system fault. More information can be found in the diagnostics buffer of the CPU.
X	X	X	●	X	Variables are forced.
X	X	X	X	●	Accessing the memory card
Overall reset					
○	BB	X	X	X	Blinking with 1Hz: Overall reset is requested
○	BB	X	X	X	Blinking with 2Hz: Overall reset is executed
○	BB	X	X	X	Blinking with 10Hz: Overall reset with activation of a default hardware configuration is executed.
Reset to factory setting					
●	●	○	○	○	Reset to factory setting is executed
○	●	●	●	●	Reset to factory setting finished without error
Firmware update					
○	●	BB	BB	●	The alternate blinking indicates that there is new firmware on the memory card.
○	○	BB	BB	●	The alternate blinking indicates that a firmware update is executed.
○	●	●	●	●	Firmware update finished without error.
○	BB	BB	BB	BB	Blinking with 10Hz: Error during Firmware update.
on: ●   off: ○   blinking: BB   not relevant: X					

**Ethernet PG/OP channel**

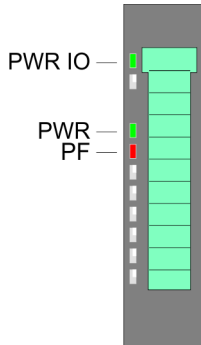
L/A (Link/ Activity)	S (Speed)	Meaning
green 	green 	
●	X	The Ethernet PG/OP channel is physically connected to the Ethernet interface.
○	X	There is no physical connection.
BB	X	Shows Ethernet activity.
X	●	The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 100Mbit.

Structure > LEDs

L/A (Link/ Activity)	S (Speed)	Meaning
X	○	The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 10Mbit.


on: ● | off: ○ | blinking: BB | not relevant: X

LEDs power module



PWR IO	PWR	PF	Description
green	green	red	
●	X	○	Power section supply OK
●	●	○	Electronic section supply OK
X	X	●	Fuse electronic section supply defective

on: ● | off: ○ | not relevant: X

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

LEDs PROFIBUS interface X3

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



Master operation

DE (Data Exchange)	BF (Bus error)	Meaning
green ●	red ■	
○	○	Master has no project, this means the interface is deactivated respectively the master configured without slaves with no errors.
BB	○	CPU is in STOP state, the master is in "clear" state. All the slaves are in DE and the outputs are of the slaves are disabled.
●	○	CPU is in STOP state, the master is in "operate" state. All the slaves are in DE. The outputs are enabled.
●	BB	CPU is in RUN state, at least 1 slave is missing and at least 1 slave is in DE.

DE (Data Exchange)	BF (Bus error)	Meaning
BB	BB	CPU is in STOP state, at least 1 slave is missing and at least 1 slave is in DE.
○	●	PROFIBUS is interrupted (no communication possible)
○	BB	At least 1 slave is missing and no slave is in DE.
X	BB	At least 1 slave is not in DE.



on: ● | off: ○ | blinking (2Hz): BB

**LEDs PROFINET IO controller X4**

BF1 (Bus error)	MT (Data Exchange)	Meaning
red 	yellow 	
●	X	<ul style="list-style-type: none"> <li>■ Bus error, no connection to sub net/switch</li> <li>■ wrong transfer rate</li> <li>■ Full-duplex-transmission</li> </ul>
BB	X	<ul style="list-style-type: none"> <li>■ Failure of a connected IO device</li> <li>■ At least one IO device is not access-able</li> <li>■ Faulty configuration</li> </ul>
X	●	Maintenance event is pending.
BB*	BB*	Simultaneous blinking (4s on, 1s off) indicates a not valid configuration.
BB*	BB*	* The alternate blinking with 4Hz indicates that a firmware update of the PROFINET IO controller is executed.
●	●	Firmware update of the PROFINET IO controller is finished without error.

Technical Data

BF1 (Bus error)	MT (Data Exchange)	Meaning
X	BB	As long as the PROFINET IO controller is not yet configured, for identification you may cause the LED to blink by means of the "Network search of your configuration tool.
on: ●   off: ○   blinking (2Hz): BB   not relevant: X		

L/A1 (Link/Activity)	S1 (Speed)	Meaning
green 	green 	
●	X	The PROFINET IO controller is physically connected to the Ethernet interface.
○	X	There is no physical connection.
B	X	blinks: shows Ethernet activity.
X	●	The Ethernet interface of the PROFINET IO controller has a transfer rate of 100Mbit.
X	○	The Ethernet interface of the PROFINET IO controller has a transfer rate of 10Mbit.
on: ●   off: ○   not relevant: X		

### 3.3 Technical Data

<b>Order no.</b>	<b>015-CEFPR00</b>
Type	SLIO CPU 015
SPEED-Bus	-
<b>Technical data power supply</b>	
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)	150 mA
Current consumption (rated value)	1.1 A
Inrush current	3 A
I²t	0.1 A²s
Max. current drain at backplane bus	3 A
Power loss	7.5 W
<b>Load and working memory</b>	

Order no.	015-CEFPR00
Load memory, integrated	512 KB
Load memory, maximum	512 KB
Work memory, integrated	256 KB
Work memory, maximal	512 KB
Memory divided in 50% program / 50% data	✓
Memory card slot	SD/MMC-Card with max. 2 GB
<b>Hardware configuration</b>	
Racks, max.	1
Modules per rack, max.	64
Number of integrated DP master	1
Number of DP master via CP	-
Operable function modules	64
Operable communication modules PtP	64
Operable communication modules LAN	-
<b>Status information, alarms, diagnostics</b>	
Status display	yes
Interrupts	no
Process alarm	no
Diagnostic interrupt	no
<b>Command processing times</b>	
Bit instructions, min.	0.01 µs
Word instruction, min.	0.01 µs
Double integer arithmetic, min.	0.01 µs
Floating-point arithmetic, min.	0.06 µs
<b>Timers/Counters and their retentive characteristics</b>	
Number of S7 counters	512
Number of S7 times	512
<b>Data range and retentive characteristic</b>	
Number of flags	8192 Byte
Number of data blocks	4096
Max. data blocks size	64 KB
Max. local data size per execution level	4096 Byte
<b>Blocks</b>	
Number of OBs	24
Number of FBs	4096

Order no.	015-CEFPR00
Number of FCs	4096
Maximum nesting depth per priority class	16
Maximum nesting depth additional within an error OB	4
<b>Time</b>	
Real-time clock buffered	✓
Clock buffered period (min.)	30 d
Accuracy (max. deviation per day)	10 s
Number of operating hours counter	8
Clock synchronization	✓
Synchronization via MPI	Master/Slave
Synchronization via Ethernet (NTP)	Slave
<b>Address areas (I/O)</b>	
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	16384
Digital outputs	16384
Digital inputs central	512
Digital outputs central	512
Integrated digital inputs	-
Integrated digital outputs	-
Analog inputs	1024
Analog outputs	1024
Analog inputs, central	256
Analog outputs, central	256
Integrated analog inputs	-
Integrated analog outputs	-
<b>Communication functions</b>	
PG/OP channel	✓
Global data communication	✓
Number of GD circuits, max.	8
Size of GD packets, max.	22 Byte
S7 basic communication	✓
S7 basic communication, user data per job	76 Byte



Order no.	015-CEFPR00
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
S7 communication, user data per job	160 Byte
Number of connections, max.	32
<b>Functionality Sub-D interfaces</b>	
Type	X2
Type of interface	RS485
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	✓
MP <sup>2</sup> I (MPI/RS232)	-
Point-to-point interface	✓
<b>Functionality X3 interfaces</b>	
Type	X3
Type of interface	RS485
Connector	Sub-D, 9-pin, female
Electrically isolated	✓
MPI	-
MP <sup>2</sup> I (MPI/RS232)	-
Point-to-point interface	-
<b>Functionality MPI</b>	
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
Transmission speed, max.	12 Mbit/s
<b>Functionality PROFIBUS master</b>	
PG/OP channel	✓
Routing	✓
S7 basic communication	✓

Technical Data

Order no.	015-CEFPR00
S7 communication	✓
S7 communication as server	✓
S7 communication as client	-
Equidistance support	-
Isochronous mode	-
SYNC/FREEZE	-
Activation/deactivation of DP slaves	-
Direct data exchange (slave-to-slave communication)	-
DPV1	✓
Transmission speed, min.	9.6 kbit/s
Transmission speed, max.	12 Mbit/s
Number of DP slaves, max.	124
Address range inputs, max.	2 KB
Address range outputs, max.	2 KB
User data inputs per slave, max.	244 Byte
User data outputs per slave, max.	244 Byte
<b>Functionality PROFIBUS slave</b>	
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
<b>Functionality PROFINET I/O controller</b>	
Realtime Class	-
Conformance Class	PROFINET IO
Number of PN IO devices	128

Order no.	015-CEFPR00
IRT support	-
Prioritized start-up	-
Number of PN IO lines	1
Address range inputs, max.	2 KB
Address range outputs, max.	2 KB
Transmitting clock	1 ms
Update time	1 ms .. 512 ms
<b>Point-to-point communication</b>	
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.	115.5 kbit/s
Cable length, max.	500 m
<b>Point-to-point protocol</b>	
ASCII protocol	✓
STX/ETX protocol	✓
3964(R) protocol	✓
RK512 protocol	-
USS master protocol	✓
Modbus master protocol	✓
Modbus slave protocol	✓
Special protocols	-
<b>Functionality RJ45 interfaces</b>	
Type	X1
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	4
Productive connections	-
Type	X4

## Technical Data

<b>Order no.</b>	<b>015-CEFPR00</b>
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Electrically isolated	✓
PG/OP channel	✓
Number of connections, max.	8
Productive connections	✓
<b>Ethernet communication CP</b>	
Number of productive connections, max.	8
Number of productive connections by Siemens NetPro, max.	8
S7 connections	BSEND, BRCV, GET, PUT, Connection of active and passive data handling
User data per S7 connection, max.	32 KB
TCP-connections	FETCH PASSIV, WRITE PASSIV, Connection of passive data handling
User data per TCP connection, max.	64 KB
ISO-connections	-
User data per ISO connection, max.	-
ISO on TCP connections (RFC 1006)	FETCH PASSIV, WRITE PASSIV, Connection of passive data handling
User data per ISO on TCP connection, max.	32 KB
UDP-connections	-
User data per UDP connection, max.	-
UDP-multicast-connections	-
UDP-broadcast-connections	-
<b>Ethernet open communication</b>	
Number of connections, max.	8
User data per ISO on TCP connection, max.	8 KB
User data per native TCP connection, max.	8 KB
User data per ad hoc TCP connection, max.	1460 Byte
User data per UDP connection, max.	1472 Byte
<b>Housing</b>	
Material	PPE
Mounting	Profile rail 35 mm
<b>Mechanical data</b>	
Dimensions (WxHxD)	131.5 mm x 109 mm x 83 mm
Weight	310 g
<b>Environmental conditions</b>	

<b>Order no.</b>	<b>015-CEFPR00</b>
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
<b>Certifications</b>	
UL508 certification	in preparation

## 4 Deployment CPU 015

### 4.1 Assembly



Information about assembly and cabling ↪ Chapter 2 'Basics and Assembly' on page 10

### 4.2 Start-up behaviour

#### Turn on power supply

- 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. ↪ on page 74
- The CPU checks if a previously activated VSC is inserted. If not, the SD LED gets on and a diagnostics entry is released. The CPU switches to STOP after 72 hours. With a just installed VSC activated functions remain activated. ↪ Chapter 4.18 'VIPA specific diagnostic entries' on page 80

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. At the start-up of the CPU, this assigns automatically peripheral addresses for digital in-/output modules starting with 0 and ascending depending on the slot location.

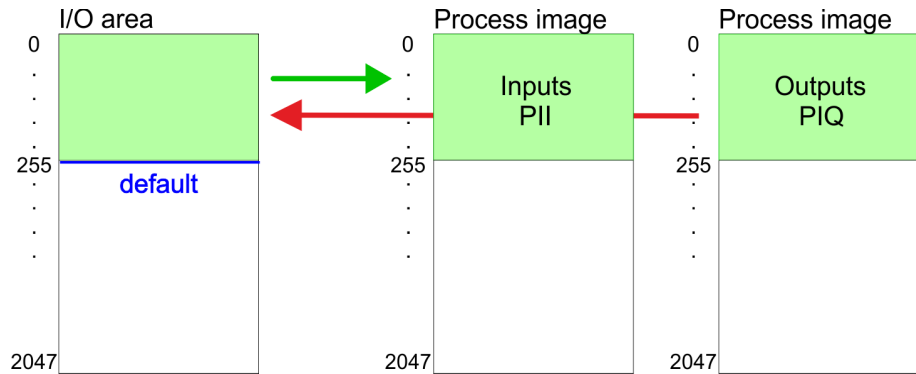
If no hardware project engineering is available, the CPU stores at the addressing analog modules to even addresses starting with 256.

### 4.3.2 Addressing backplane bus periphery

The CPU 015 provides an I/O area (address 0 ... 2047) and a process image of the in- and outputs (each address default 0 ... 255). The process image stores the signal states of the lower address (default 0 ... 255) additionally in a separate memory area. The size of the process image can be preset via the parameterization. ↪ 'Cycle/Clock memory' on page 58

The process image this divided into two parts:

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



The process image is updated automatically when a cycle has been completed.

**Max. number of plug-gable modules**

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

**Define addresses by hardware configuration**

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

**Example for automatic address allocation**

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

**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.



*For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required!*

The configuration of the System SLIO CPU happens in the Siemens SIMATIC Manager by means of a virtual PROFINET IO device 'VIPA SLIO CPU'. The 'VIPA SLIO System' is to be installed in the hardware catalog by means of the GSDML.

**Installing the IO device VIPA SLIO System**

The installation of the PROFINET IO devices 'VIPA SLIO CPU' happens in the hardware catalog with the following approach:

1. Go to the service area of [www.vipa.com](http://www.vipa.com).
2. Load from the download area at 'PROFINET files' the file System SLIO\_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 SLIO System'



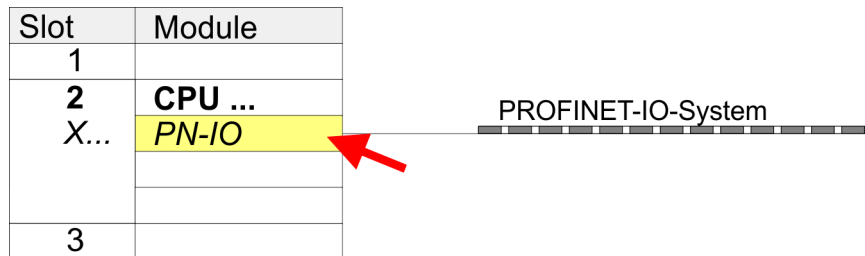
**Proceeding**

To be compatible with the Siemens SIMATIC manager the following steps should be executed:

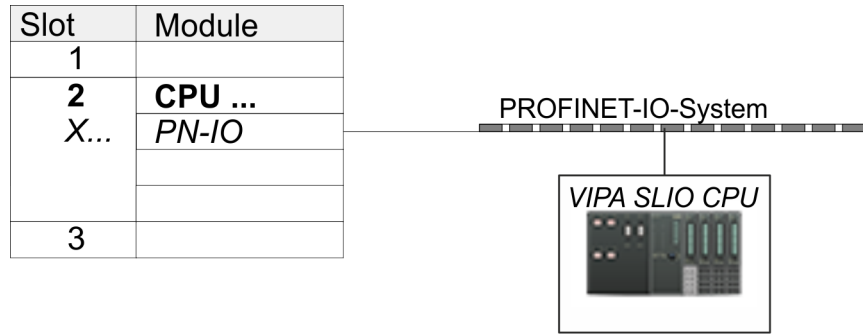
1. ▶ Start the Siemens hardware configurator with a new project.
2. ▶ Insert a profile rail from the hardware catalog.
3. ▶ Place at 'Slot'-Number 2 the CPU 315-2 PN/DP (315-2EH14 V3.2).

Slot	Module
1	
<b>2</b>	<b>CPU 315-2PN/DP</b>
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. ▶ Insert at 'General' a 'Device name' The device name must be unique at the Ethernet subnet.



Slot	Module	Order number
0	<b>VIPA SLIO CPU ...</b>	<b>015-CEFRPR00</b>
X2	<i>015-CEFRPR00</i>	
1		
2		
3		
...		

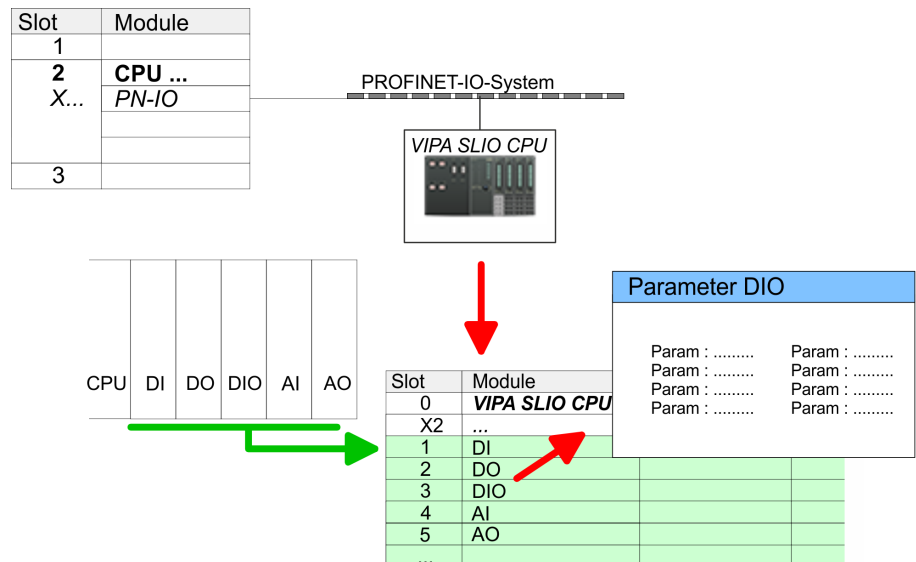
9. Navigate in the hardware catalog to the directory 'PROFINET IO → Additional field devices → I/O → VIPA SLIO System' and connect the IO device '015-CEFRPR00 CPU' to your PROFINET system.
  - ⇒ In the slot overview of the PROFINET IO device 'VIPA SLIO CPU' the CPU is already placed at slot 0. From slot 1 you can place your system SLIO modules.

### 4.5 Hardware configuration - I/O modules

#### Hardware configuration of the modules

Starting with slot 1 place in the slot overview of the PROFINET IO device 'VIPA SLIO CPU' your System SLIO modules in the plugged sequence.

To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU.



**Parametrization** For parametrization double-click during the project engineering at the slot overview on the module you want to parametrize. In the appearing dialog window you may set the wanted parameters.

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

## 4.6 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 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.

With the first start-up respectively after an overall reset the Ethernet PG/OP channel does not have any IP address.

For online access to the CPU via Ethernet PG/OP channel valid IP address parameters have to be assigned to this by means of the Siemens SIMATIC Manager. This is called "initialization".

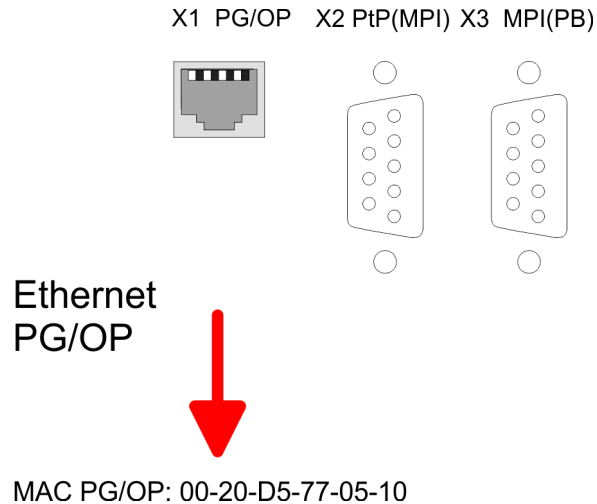
### Assembly and commissioning

1. ► Install your System SLIO with your CPU.
2. ► Wire the system by connecting cables for voltage supply and signals.
3. ► Connect the Ethernet jack (X1) of the Ethernet PG/OP channel to Ethernet
4. ► Switch on the power supply.
  - ⇒ After a short boot time the CP is ready for communication. He possibly has no IP address data and requires an initialization.

### "Initialization" via PLC functions

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

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



### Assign IP address parameters

You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens SIMATIC manager starting with version V 5.5 & SP2 with the following proceeding:

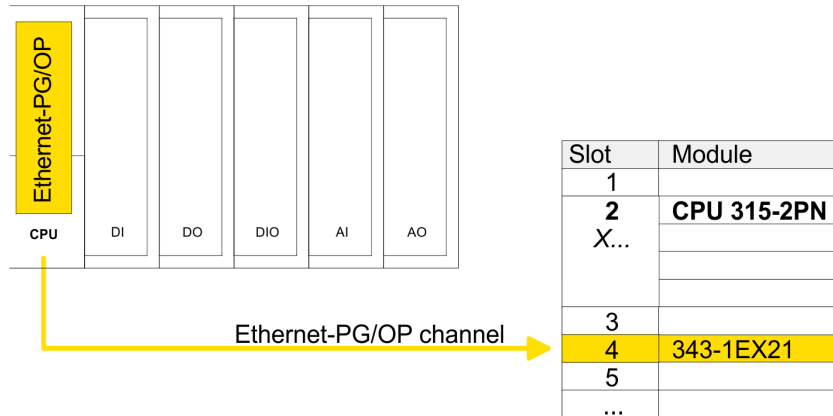
1. ▶ Start the SIMATIC Manager and set via 'Options → Set PG/PC interface' the access path to 'TCP/IP -> Network card ....'.
2. ▶ Open with 'PLC → Edit Ethernet Node' the dialog window with the same name.
3. ▶ To get the stations and their MAC address, use the [Browse] button or type in the MAC address of your 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 a factory reset is executed.*

### Take IP address parameters in project

1. ▶ Open the Siemens hardware configurator and configure the Siemens CPU 315-2 PN/DP (6ES7 315-2EH14-0AB0 V3.2).
2. ▶ For the Ethernet PG/OP channel you have to configure at slot 4 a Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX21 0XE0 V.1.2).
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. ▶ Transfer your project.

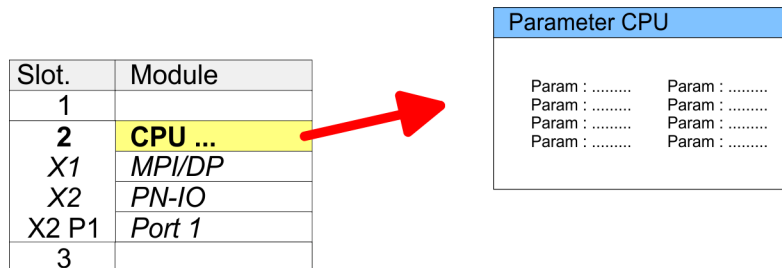


## 4.7 Setting standard CPU parameters

### 4.7.1 Parametrization via Siemens CPU

#### Parametrization via Siemens CPU 315-2EH14

Since the CPU from VIPA is to be configured as Siemens CPU 315-2 PN/DP (6ES7 315-2EH14-0AB0 V3.2) in the Siemens hardware configurator, the standard parameters of the VIPA CPU may be set with "Object properties" of the CPU 315-2 PN/DP during hardware configuration. Via a double-click on the CPU 315-2 PN/DP the parameter window of the CPU may be accessed. Using the registers you get access to every standard parameter of the CPU.



### 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 315-2EH14 is CPU 315-2 PN/DP.
- 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 1346-1.

- Location designation: The location designation is part of the resource designation. Here the exact location of your module within a plant may be specified.
- 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 or 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 PRO-FIBUS 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 ready message by modules [100ms]
- 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: This parameter is not relevant.
- 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.
- 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: Here the priorities may be specified according to which the time-of-day interrupt is processed. With priority "0" the corresponding OB is deactivated.
  - Active: Activate the check box of the time-of-day interrupt OBs if these are to be automatically started on complete restart.
  - 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. With priority "0" the corresponding interrupt is deactivated.
  - 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: Here the number of diagnostics are displayed, which may be stored in the diagnostics buffer (circular buffer).
  - 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

**4.7.3 Parameter for MPI/DP**

The properties dialog of the MPI(PB) interface X3 is opened via a double click to the sub module MPI/DP



To switch the interface to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from 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 75

**General**

- Short description: Here the short description "MPI/DP" for the interface is specified.
- Order no.: Here nothing is shown.
- Name: At *Name* "MPI/DP" is shown. If you change the name, the new name appears in the Siemens SIMATIC Manager.
- Type: Here you can choose between the functionality MPI and PROFIBUS.
- Interface: Here the MPI respectively PROFIBUS address is shown.
- Properties With this button the properties of the interface may be pre-set.
- Comment You can enter the purpose of the interface.

**Address**

- Diagnostics: A diagnostics address for the interface is to be pre-set here. In the case of an error the CPU is informed via this address.
- Operating mode: With the interface type '*PROFIBUS*' here you can pre-set the '*Operating mode*' DP master.
- Configuration, Clock: These parameters are not supported.

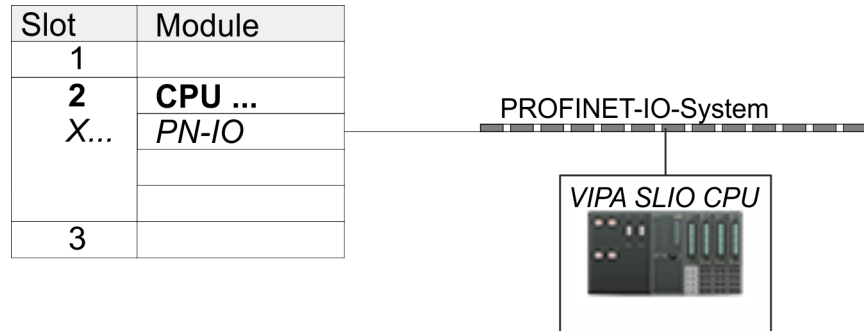
**4.8 Setting VIPA specific CPU parameters****Overview**

Except of the VIPA specific CPU parameters the CPU parametrization takes place in the parameter dialog of the CPU 315-2 PN/DP from Siemens. After the hardware configuration of the CPU you can set the parameters of the CPU in the virtual IO device '*VIPA SLIO CPU*'. Via double-click at the VIPA SLIO CPU the properties dialog is opened.



Here the following parameters may be accessed:

- Function X2 (PtP/MPI)
- MPI address X2
- MPI Baud rate X2
- Additional retentive memory/timer/counter



Slot	Module	Order number	
0	VIPA SLIO CPU ...	...	
X2	...		
1			
2			
3			
...			

**VIPA specific parameter**

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

- Function X2: Function PtP(MPI) interface X2
  - PtP (default): With this operating mode the RS485 interface acts as an interface for serial point-to-point communication. Here data may be exchanged between two stations by means of protocols.
  - MPI: With this operating mode the interface serves for the connection between programming unit and CPU via MPI. By means of this e.g. the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU.
- MPI address X2: With *MPI* you can specify the MPI address here. With *PTP* this parameter is ignored by the CPU.  
Range of values: 2 (default) ... 31
- MPI Baud rate X2: With *MPI* you can specify the MPI transfer rate here. With *PTP* this parameter is ignored by the CPU.  
Range of values: 19.2kb/s ... 12Mb/s, default: 187.5kb/s
- Additional retentive memory: 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

## 4.9 Project transfer

### Overview

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

- Transfer via MPI (optional via PROFIBUS)
- Transfer via Ethernet
- Transfer via memory card



*To switch the interface X3 MPI(PB) to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from 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 75*

### 4.9.1 Transfer via MPI / optional PROFIBUS

#### General

For transfer via MPI / optional PROFIBUS there are the following 2 interface:

- X2: PtP(MPI) ↪ *'PtP(MPI) interface X2' on page 36*
- X3: MPI(PB) ↪ *'MPI(PB) interface X3' on page 36*

#### Net structure

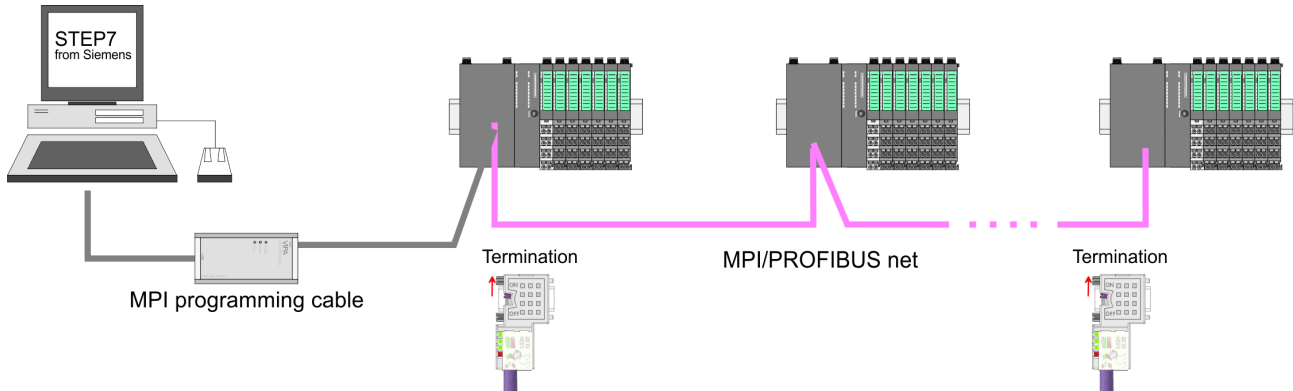
The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs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

**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.

**Proceeding Transfer via PROFIBUS interface**

To switch the interface to PROFIBUS functionality you have to activate the according bus functionality by means of a VSC storage media from VIPA. By plugging the VSC storage card and then an overall reset the according functionality is activated.

1. ▶ Connect your PC to the MPI(PB) jack X3 of your CPU via a MPI programming cable.
2. ▶ Load your project in the Siemens SIMATIC Manager.
3. ▶ Choose in the menu 'Options → Set PG/PC interface'.
4. ▶ Select in the according list the "PC Adapter (PROFIBUS)"; if appropriate you have to add it first, then click at [Properties].
5. ▶ Set in the register PROFIBUS the transfer parameters of your PROFIBUS net and enter a valid PROFIBUS address. The PROFIBUS address must be assigned to the DP master by a project before.
6. ▶ Switch to the register Local connection.
7. ▶ Set the COM port of the PCs and the transfer rate 38400baud for the MPI programming cable from VIPA.
8. ▶ Transfer your project via 'PLC → Load to module' via PROFIBUS to the CPU and save it with 'PLC → Copy RAM to ROM' on a memory card if one is plugged.



Transfer via PROFIBUS is available by DP master, if projected as master and assigned with a PROFIBUS address before.

**4.9.2 Transfer via Ethernet**

For transfer via Ethernet the CPU has the following interface:

- X1: Ethernet PG/OP channel

**Initialization**

So that you may access the Ethernet PG/OP channel you have to assign IP address parameters by means of the "initialization".

↳ Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 55

**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.*

### 4.9.3 Transfer via memory card

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

- S7PROG.WLD
- AUTOLOAD.WLD

With '*File → Memory Card File → New*' in the Siemens SIMATIC Manager a new wld file may be created. After the creation copy the blocks from the project blocks folder and the *System data* into the wld file.

#### Transfer memory card → CPU

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

- *S7PROG.WLD* is read from the memory card after overall reset.
- *AUTOLOAD.WLD* is read from the memory card after PowerON.

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

#### Transfer CPU → memory card

When a memory card has been installed, the write command stores the content of the RAM as *S7PROG.WLD* on the memory card.

The write command is controlled by means of the block area of the Siemens SIMATIC Manager '*PLC → Copy RAM to ROM*'. The SD LED blinks during the write access. When the LED expires, the write process is finished.

If this project is to be loaded automatically from the memory card with PowerON, you have to rename this to on the memory card to *AUTOLOAD.WLD*.

Accessing the web server

**Checking the transfer operation**

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries you choose in the Siemens SIMATIC manager 'PLC → Module information'. Via the register "Diagnostic Buffer" you reach the diagnostic window.  
 ↪ Chapter 4.18 'VIPA specific diagnostic entries' on page 80

**4.10 Accessing the web server**

**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 55

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.

**Structure of the Web page**

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



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

**Web page with selected CPU**

Name	Value
Ordering Info	015-CEFPR00
Serial	00108765
Version	01V08.001
HW Revision	01
Software	01

[ Expert View ... ]

**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 Info		
Operation Mode	RUN	CPU: Status information

Mode Switch	RUNP	
System Time	31.10.13 11:52:48	CPU: Date, time
Cycle Time	cur = 1000 us, min = 0 us, max = 2000 us, avg = 281 us	CPU: Cyclic time: min= minimum cur= current max= maximum avg= average
ArmLoad	cur = 44%, max = 50%	Information for the support
RS485 X2	PTP	Operating mode RS485
RS485 X3	MPI	
<b>Onboard Ethernet</b>		
Device Name	Onboard PG/OP	Ethernet PG/OP channel: Address
MAC	00:20:D5:01:7A:D1	
IP	172.20.120.42	
Mask	255.255.255.0	
Gateway	172.20.120.42	
<b>Onboard Profinet</b>		
Device Name	n/a	PROFINET I/O controller: Address
IP	n/a	
Mask	n/a	
Gateway	n/a	
Bb000429	V1.1.1.1	
PRODUCT	VIPA PROFINET CP V1.1.1.1 Px000206.pkg	PROFINET I/O controller: Name, firmware version, package
Hx000081	V1.1.0.0	Information for the support
Diagnosis Address	2046	
<b>Memory Usage</b>		
LoadMem	0/524288 bytes	CPU: Information to memory configuration
WorkMemCode	0/131072 bytes	Load memory, work memory (code/data)
WorkMemData	0/131072 bytes	
<b>VIPASetCard Info</b>		
VSD...		Information for the support
...		
VSC...		
...		

Accessing the web server

VSC-Trial-Time	71:59	Remaining time in hh:mm for deactivation of the expansion memory respectively bus functionality if memory card is removed. Then the CPU switches to STOP state.
Memory Extension	0 bytes	Size of the additional memory, which was activated by means of a VSC.
Profibus	PB NO	Type of the the PROFIBUS functionality, which was activated by means of a VSC.
<b>Flash System</b>		
File System	V1.0.2	CPU: Information for the support
PRODUCT	015-CEFPR00 V1.1.6.0 Px000198.pkg	CPU: Name, firmware version, package
HARDWARE	V0.1.0.0 5816B-V10 MX000226.004	CPU: Information for the support
Bx000501	V1.1.6.0	
Ax000132	V1.0.2.0	
fx000018.wld	V1.0.1.0	
syslibex.wld	n/a	
Protect.wld	n/a	

**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

**Web page with selected module**

The screenshot shows the VIPA web interface. On the left, there is a sidebar with the following items:
 

- Device (VIP A 015-CEFPR00)
- Module 1 (VIP A 021-1BD00) - This item is selected and has a red arrow pointing to it.
- Module 2 (VIP A 022-1BD00)

 The main content area has tabs for 'Info', 'Data', and 'Parameter'. The 'Info' tab is active, displaying 'Module 1 (VIP A 021-1BD00) information'. Below this, there is a table with the following data:
 

Name	Value
Ordering Info	021-1BD00
Serial	00103265
Version	01V30.001
HW Revision	01



<b>Info</b>	Here product name, order number, serial number, firmware version and hardware state number of the according module are listed.
<b>Data</b>	At Data the states of the inputs respectively outputs are listed.
<b>Parameter</b>	If available the parameter data of the corresponding module may be shown.

## 4.11 Operating modes

### 4.11.1 Overview

The CPU can be in one of 4 operating modes:

- Operating mode STOP
- Operating mode START-UP
- Operating mode RUN
- Operating mode HALT

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.
- Outputs are inhibited, i.e. all digital outputs are disabled.
- RUN-LED off
- STOP-LED on

#### Operating mode START-UP

- During the transition from STOP to RUN a call is issued to the start-up organization block OB 100. The processing time for this OB is not monitored. The START-UP OB may issue calls to other blocks.
- All digital outputs are disabled during the START-UP, i.e. outputs are inhibited.
- RUN-LED  
blinks as soon as the OB 100 is operated and for at least 3s, even if the start-up time is shorter or the CPU gets to STOP due to an error. This indicates the start-up.
- STOP-LED off

When the CPU has completed the START-UP OB, it assumes the operating mode RUN.

#### Operating mode RUN

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

- RUN-LED on
- STOP-LED off

### Operating mode HOLD

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

### Precondition

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

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

### Approach for working with breakpoints

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

### Behaviour in operating state HOLD

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



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

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

### 4.11.2 Function security

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

Event	concerns	Effect
RUN → STOP	general	BASP ( <b>B</b> efehls- <b>A</b> usgabe- <b>S</b> perre, i.e. command output lock) is set.
	central digital outputs	The outputs are disabled.
	central analog outputs	The outputs are disabled. <ul style="list-style-type: none"> <li>■ Voltage outputs issue 0V</li> <li>■ Current outputs 0...20mA issue 0mA</li> <li>■ Current outputs 4...20mA issue 4mA</li> </ul> If configured also substitute values may be issued.
	decentral outputs	Same behaviour as the central digital/analog outputs.
	decentral inputs	The inputs are cyclically be read by the decentralized station and the recent values are put at disposal.
STOP → RUN res. PowerON	general	First the PII is deleted, then OB 100 is called. After the execution of the OB, the BASP is reset and the cycle starts with: Delete PIO → Read PII → OB 1.
	decentral inputs	The inputs are once be read by the decentralized station and the recent values are put at disposal.
RUN	general	The program execution happens cyclically and can therefore be foreseen: 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.*

**Overall reset by means of the operating mode switch**

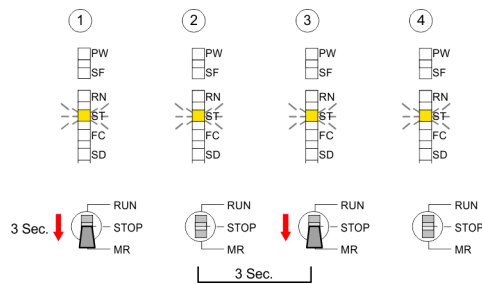
Precondition

- ➔ Your CPU must be in STOP state. For this switch the operating mode switch to "STOP".
- ⇒ The STOP-LED is on.

Overall reset

1. ➔ Hold the operating mode switch for ca. 3 seconds in MR position.
  - ⇒ The STOP-LED changes from blinking to permanently on.
2. ➔ Switch the operating mode switch in STOP position and switch it to MR and quickly back to STOP within a period of 3 seconds.
  - ⇒ The STOP-LED blinks (overall reset procedure).
3. ➔ The overall reset has been completed when the STOP-LED is on permanently.
  - ⇒ The STOP-LED is on.

The following figure illustrates the above procedure:



**Activating functionality by means of a VSC**

If there is a VSC from VIPA plugged, after an overall reset the according functionality is automatically activated. ↪ 'VSD' on page 76

**Automatic reload**

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

**Reset to factory setting**

The *Reset to factory setting* deletes completely the internal RAM of the CPU and resets this to delivery state. Please regard that the MPI address is also set back to default 2! ↪ Chapter 4.14 'Factory reset' on page 74

## 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. The pkg file name of every update-able component can be found at a label on the module. The SLIO CPU has no label. Here the pkg file name can be shown via the web page. After PowerON and CPU 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 blinking of the LEDs and the firmware may be installed by an update request.

### Current firmware at [www.vipa.com](http://www.vipa.com)

The latest firmware versions can be found in the "service" area at [www.vipa.com](http://www.vipa.com). For example the following files are necessary for the firmware update of the CPU and its components with hardware release 1:

- CPU 015, Hardware release 1: Px000198.pkg
- PROFINET I/O controller: Px000206.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 the VIPA 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 MPI or via 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 55*

### Load firmware and transfer it to memory card

- Go to [www.vipa.com](http://www.vipa.com)
- Click at '*Service Support → Downloads → Firmware*'.
- Via '*System SLIO → CPU*' navigate to your CPU and download the zip file to your PC.
- Unzip the zip file and copy the file to the root directory of your memory card.

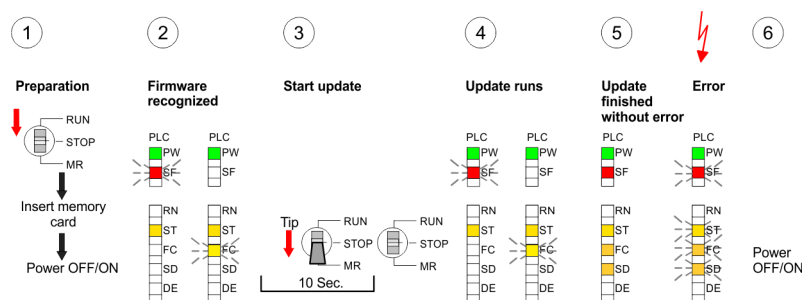


**CAUTION!**

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 'Factory reset' on page 74

**Transfer firmware from memory card into CPU**

1. ➤ Switch the operating mode switch of your CPU in position STOP. Turn off the power supply. Plug the memory card with the firmware files into the CPU. Please take care of the correct plug-in direction of the memory card. Turn on the power supply.
2. ➤ After a short boot-up time, the alternate blinking of the LEDs SF and FC shows that at least a more current firmware file was found at the memory card.
3. ➤ 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.
4. ➤ During the update process, the LEDs SF and FC are alternately blinking and SD LED is on. This may last several minutes.
5. ➤ The update is successful finished when the LEDs PW, ST, SF, FC and SD are on. If they are blinking fast, an error occurred.
6. ➤ Turn power OFF and ON. Now it is checked by the CPU, whether further firmware updates are to be executed. If so, again the LEDs SF and FC flash after a short start-up period. Continue with 3. If the LEDs do not flash, the firmware update is finished.
7. ➤ Now a *Reset to factory setting* as described next should be executed. After that the CPU is ready for duty. ↪ Chapter 4.14 'Factory reset' on page 74



**4.14 Factory reset**

**Proceeding**

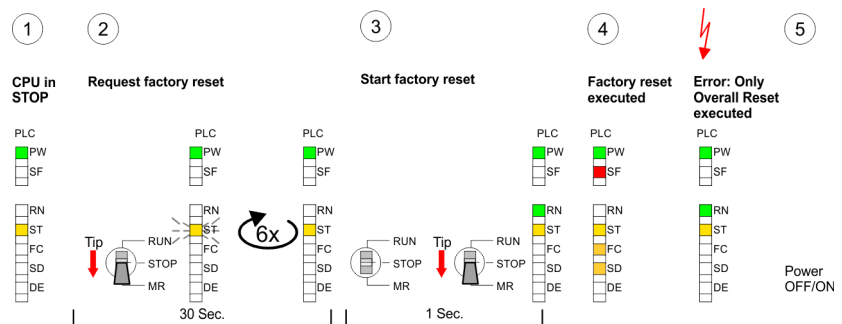
With the following proceeding the internal RAM of the CPU is completely deleted and the CPU is reset to delivery state.

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 `FAC-TORY_RESET`. *☞ 'Commands' on page 79*

1. **▶** Switch the CPU to STOP.
2. **▶** Push the operating mode switch down to position MR for 30 seconds. Here the STOP-LED flashes. After a few seconds the STOP LED changes to static light. Now the STOP LED changes between static light and flashing. Start here to count the static light of the STOP LED.
3. **▶** After the 6. static light release the operating mode switch and tip it downwards to MR. Now the RUN LED lights up once. This means that the RAM was deleted completely.
4. **▶** For the confirmation of the resetting procedure the LEDs PW, ST, SF, FC and MC get on. If not, the factory reset has failed and only an overall reset was executed. In this case you can repeat the procedure. A factory reset can only be executed if the STOP LED has static light for exact 6 times.
5. **▶** The end of factory reset is shown by static light of the LEDs PW, ST, SF, FC and SD. Switch the power supply off and on.

The following figure illustrates the procedure above:



*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 - VIPA SD-Card:** External memory card for programs and firmware.
- **VSC - VIPASetCard:** 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 71*





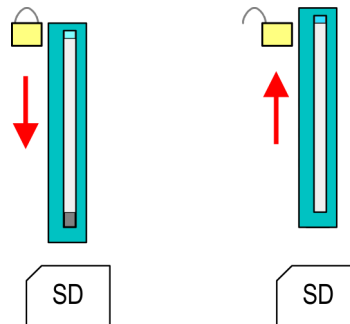
A list of the currently available VSD respectively VSC can be found at [www.vipa.com](http://www.vipa.com).

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 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 led 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 66*



### CAUTION!

Please regard that the VSC must remain plugged when you've enabled optional functions at your CPU. Otherwise the CPU switches to STOP after 72 hours. The VSC cannot be exchanged with a VSC of the same optional functions. The activation code is fixed to the VSD by means of an unique serial number. Here the functionality 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.  
 ↪ on page 74

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.

**4.16 Extended know-how protection**

**Overview**

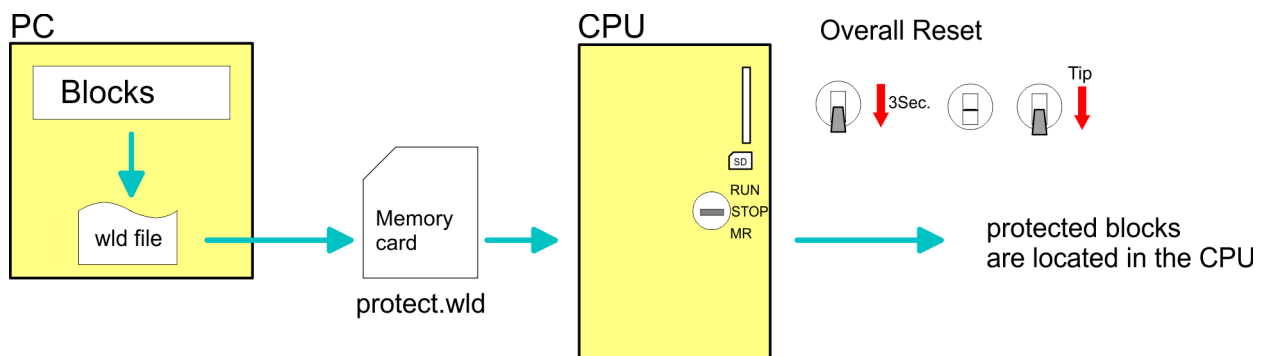
Besides the "standard" Know-how protection the SPEED7-CPU's from VIPA provide an "extended" know-how protection that serves a secure block protection for accesses of 3. persons.

**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 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.

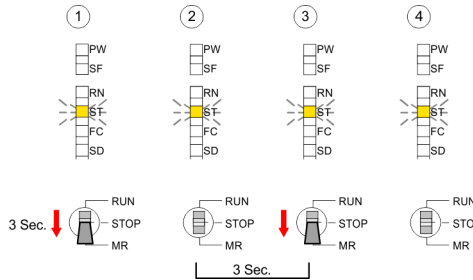


**Protect blocks with protect.wld**

Create a new wld-file in your project engineering tool with 'File → Memory Card file → New' and rename it to "protect.wld". Transfer the according blocks into the file by dragging them with the mouse from the project to the file window of protect.wld.

**Transfer protect.wld to CPU with overall reset**

Transfer the file protect.wld to a memory card, plug the memory card into the CPU and execute an overall reset with the following approach:



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
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 parentheses.

**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)
SET_NETWORK 172.16.129.210,255.255.224.0, 172.16.129.210	IP parameter(0xE80E)
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 VIPA specific diagnostic entries****Entries in the diagnostic buffer**

You may read the diagnostic buffer of the CPU via the Siemens SIMATIC Manager. Besides of the standard entries in the diagnostic buffer, the VIPA CPUs support some additional specific entries in form of event-IDs.

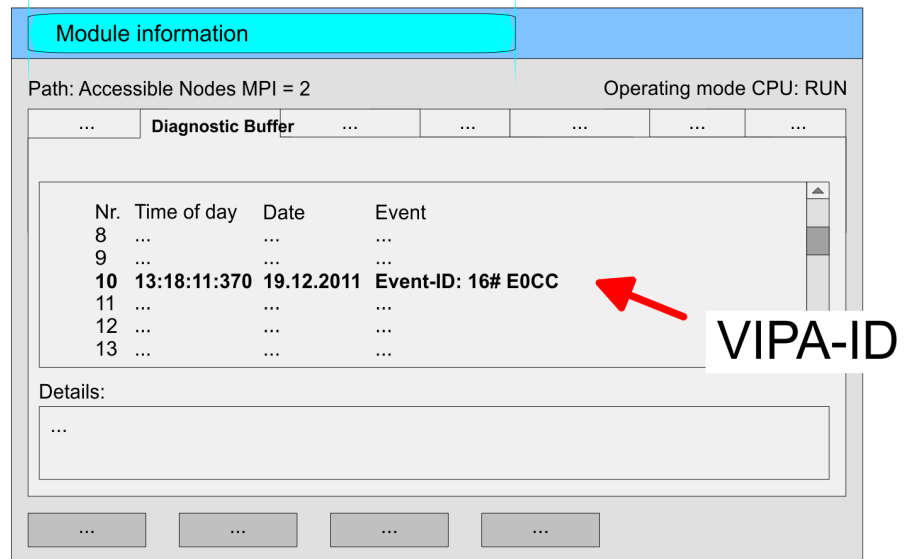
The current content of the diagnostics buffer is stored at the memory card by means of the CMD DIAGBUF.



*Every register of the module information is supported by the VIPA CPUs. More information may be found at the online help of the Siemens SIMATIC Manager.*

**Monitoring the diagnostic entries**

To monitor the diagnostic entries you choose the option 'PLC → Module Information' in the Siemens SIMATIC Manager. Via the register "Diagnostic Buffer" you reach the diagnostic window:



The diagnosis is independent from the operating mode of the CPU. You may store a max. of 100 diagnostic entries in the CPU. The following page shows an overview of the VIPA specific Event-IDs.

**Overview of the Event-IDs**

Event-ID	Description
0x115C	Vendor-specific interrupt (OB 57) at EtherCAT OB: OB number (57) ZInfo1: Logical address of the slave, which has released the interrupt ZInfo2: Interrupt type ZInfo3: reserved
0xE003	Error on accessing the periphery ZInfo1: Periphery address ZInfo2: Slot
0xE004	Multiple parametrization of a periphery address ZInfo1: Periphery address ZInfo2: Slot
0xE005	Internal error - Please contact the VIPA Hotline!
0xE006	Internal error - Please contact the VIPA Hotline!
0xE007	Configured in-/output bytes do not fit into periphery area
0xE008	Internal error - Please contact the VIPA Hotline!
0xE009	Error on accessing the standard backplane bus

Event-ID	Description
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 (32=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 is not defined
0xE030	Error of the standard bus
0xE033	Internal error - Please contact the VIPA Hotline!
0xE0B0	SPEED7 is not stoppable (probably undefined BCD value at timer)
0xE0C0	Not enough space in work memory for storing code block (block size exceeded)
0xE0CB	Error at SSL access ZInfo1: 4=SSL wrong, 5=SubSSL wrong, 6=Index wrong ZInfo2: SSL-ID ZInfo3: Index

Event-ID	Description
0xE0CC	Communication error MPI / Serial ZInfo1: Code 1: Wrong priority 2: Buffer overflow 3: Frame format error 4: Wrong SSL request (SSL-ID not valid) 5: Wrong SSL request (SSL-SubID not valid) 6: Wrong SSL request (SSL-Index not valid) 7: Wrong value 8: Wrong RetVal 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 writing to memory card 90: Faulty buffer size 98: unknown error 99: internal error
0xE0CD	Error at DP-V1 job management
0xE0CE	Error: Timeout at sending of the i-slave diagnostics
0xE0CF	Timeout at loading of a new HW configuration (timeout: 39 seconds)
0xE100	Memory card access error
0xE101	Memory card error file system
0xE102	Memory card error FAT
0xE104	Memory card error at saving
0xE200	Memory card writing finished (Copy Ram2Rom)
0xE210	Memory card reading finished (reload after overall reset)
0xE21E	Memory card reading: Error at reload (after overall reset), file "Protect.wld" too big
0xE21F	Memory card reading: Error at reload (after overall reset), file read error, out of memory
0xE300	Internal flash writing finished (Copy Ram2Rom)
0xE310	Internal flash writing finished (reload after battery failure)
0xE311	Internal flash fx0000yy.wld file too big, load failure

## VIPA specific diagnostic entries

Event-ID	Description
0xE400	Memory card with the option memory expansion was plugged
0xE401	Memory card with the option memory expansion was removed
0xE402	The PROFIBUS DP master functionality is disabled The interface acts further as MPI interface
0xE403	The PROFIBUS DP slave functionality is disabled The interface acts further as MPI interface
0xE500	Memory management: Deleted block without corresponding entry in BstList ZInfo2: BlockType ZInfo3: BlockNo
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
0xE701	Internal error - Please contact the VIPA Hotline!
0xE703	Internal error - Please contact the VIPA Hotline!
0xE720	Internal error - Please contact the VIPA Hotline!
0xE721	Internal error - Please contact the VIPA Hotline!
0xE801	CMD - Auto command: CMD_START recognized and successfully executed
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
0xE807	CMD - Auto command: FACTORY_RESET recognized and successfully executed
0xE80B	CMD - Auto command: DIAGBUF recognized and successfully executed
0xE80E	CMD - Auto command: SET_NETWORK recognized and successfully executed
0xE8FB	CMD - Auto command: Error: Initialization of the Ethernet PG/OP channel by means of SET_NETWORK is faulty
0xE8FC	CMD - Auto command: Error: Some IP parameters missing in SET_NETWORK
0xE8FE	CMD - Auto command: Error: CMD_START missing
0xE8FF	CMD - Auto command: Error: Error while reading CMD file (memory card error)
0xE901	Check sum error



Event-ID	Description
0xEA00	Internal error - Please contact the VIPA Hotline!
0xEA01	Internal error - Please contact the VIPA Hotline!
0xEA02	SBUS: Internal error (internal plugged sub module not recognized) ZInfo1: internal slot
0xEA03	SBUS: Communication error CPU - PROFINET I/O controller: ZInfo1: Slot ZInfo2: Status (0: OK, 1: ERROR, 2: BUSSY, 3: TIMEOUT, 4: LOCKED, 5: UNKNOWN)
0xEA04	SBUS: Multiple parametrization of a periphery address ZInfo1: Periphery address ZInfo2: Slot ZInfo3: Data width
0xEA05	Internal error - Please contact the VIPA Hotline!
0xEA07	Internal error - Please contact the VIPA 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 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 VIPA Hotline!

## VIPA specific diagnostic entries

Event-ID	Description
0xEA18	SBUS: Error at mapping of the master periphery ZInfo2: Slot of the master
0xEA19	Internal error - Please contact the VIPA Hotline!
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
0xEA22	Error - RS485 interface X2 - Value exceeds the limits ZInfo: Configured value of X2
0xEA23	Error - RS485 interface X3 - Value exceeds the limits ZInfo: Configured value of X3
0xEA24	Error - Configuration RS485 interface X2/X3: Interface/protocol missing, default settings are used ZInfo2: Configured value for X2 ZInfo3: Configured value for X3
0xEA30	Internal error - Please contact the VIPA Hotline!
0xEA40	Internal error - Please contact the VIPA Hotline!
0xEA41	Internal error - Please contact the VIPA Hotline!
0xEA50	Error - PROFINET configuration ZInfo1: User slot of the PROFINET I/O controller ZInfo2: IO-Device-No. ZInfo3: IO-Device slot
0xEA51	Error - there is no PROFINET IO controller at the configured slot ZInfo1: User slot of the PROFINET I/O controller ZInfo2: Recognized ID at the configured slot
0xEA53	Error - PROFINET configuration - There are too many PROFINET IO devices configured ZInfo1: Number of configured devices ZInfo2: Slot ZInfo3: Maximum possible number of devices
0xEA54	Error - PROFINET IO controller reports multiple parametrization of a periphery address ZInfo1: Periphery address ZInfo2: User slot of the PROFINET I/O controller ZInfo3: Data width

Event-ID	Description
0xEA61 ... 0xEA63	Internal error - Please contact the VIPA Hotline!
0xEA64	PROFINET/EtherCAT CP Configuration error: Zinfo1: Bit 0: too many devices Bit 1: too many devices per ms Bit 2: too many input bytes per ms Bit 3: too many output bytes per ms Bit 4: too many input bytes per device Bit 5: too many output bytes per device Bit 6: too many productive connections Bit 7: too many input bytes in the process image Bit 8: too many output bytes in the process image Bit 9: Configuration not available Bit 10: Configuration not valid Bit 11: Cycle time too small Bit 12: Cycle time too big Bit 13: Not valid device number Bit 14: CPU is configured as I device Bit 15: Obtain an IP address in a different way is not supported for the IP address of the controller
0xEA65	Internal error - Please contact the VIPA Hotline!
0xEA66	PROFINET IO controller Error in communication stack PK: Rackslot OBNr: StackError.Service DatId: StackError.DeviceRef ZInfo1: StackError.Error.Code ZInfo2: StackError.Error.Detail ZInfo3: StackError.Error.AdditionalDetail << 8 + StackError.Error.AreaCode

VIPA specific diagnostic entries

Event-ID	Description
0xEA67	Error - PROFINET IO controller - reading record set PK: Error type 0: DATA_RECORD_ERROR_LOCAL 1: DATA_RECORD_ERROR_STACK 2: DATA_RECORD_ERROR_REMOTE OBNr: PROFINET IO controller slot DatId: Device-No. ZInfo1: Record set number ZInfo2: Record set handle ZInfo3: Internal error code for service purposes
0xEA68	Error - PROFINET IO controller - writing record set PK: Error type 0: DATA_RECORD_ERROR_LOCAL 1: DATA_RECORD_ERROR_STACK 2: DATA_RECORD_ERROR_REMOTE OBNr: PROFINET IO controller slot DatId: Device-No. ZInfo1: Record set number ZInfo2: Record set handle ZInfo3: Internal error code for service purposes
0xEA69	Internal error - Please contact the VIPA Hotline!
0xEA6A	PROFINET IO controller Service error in communication stack PK: Rackslot OBNr: ServiceIdentifier DatId: 0 ZInfo1: ServiceError.Code ZInfo2: ServiceError.Detail ZInfo3: StackError.Error.AdditionalDetail
0xEA6B	PROFINET IO controller Vendor ID mismatch PK: Rackslot OBNr: PLC Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -

Event-ID	Description
0xEA6C	PROFINET IO controller Device ID mismatch PK: Rackslot OBNo: PLC Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -
0xEA6D	PROFINET IO controller No empty name PK: Rackslot OBNo: PLC Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -
0xEA6E	PROFINET IO controller RPC response missing PK: Rackslot OBNo: PLC Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -
0xEA6F	PROFINET IO controller PN module mismatch PK: Rackslot OBNo: PLC-Mode DatId: 0 ZInfo1: Device ID ZInfo2: - ZInfo3: -
0xEA97	Storage error SBUS service channel ZInfo3 = Slot
0xEA98	Timeout at waiting for reboot of a SBUS module (server)
0xEA99	Error at file reading via SBUS

Event-ID	Description
0xEAA0	Emac Error occurred OBN0: current PLC mode ZInfo1: Diagnostics address of the master / controller ZInfo2: 0: None Rx queue is full 1: No send buffer available 2: Send stream was cut off; sending failed 3: Exhausted retries 4: No receive buffer available in Emac DMA 5: Emac DMA transfer aborted 6: Queue overflow 7: Unexpected frame received ZInfo3: Number of errors, which occurred
0xEAB0	Link mode not valid OBN0: current PLC mode ZInfo1: Diagnostics address master / controller Zinfo2: Current LinkMode 0x01: 10Mbit full-duplex 0x02: 100Mbit half-duplex 0x03: 100Mbit full-duplex 0x05: 10Mbit half-duplex 0xFF: Link mode not defined
0xEB03	SLIO error on IO mapping
0xEB10	SLIO error: Bus error ZInfo1: Type of error 0x82: ErrorAlarm
0xEB20	SLIO error: Interrupt information undefined
0xEB21	SLIO error on accessing the configuration data

Event-ID	Description
0xEC03	<p>EtherCAT configuration error</p> <p>ZInfo1: Errorcode</p> <p>1: NUMBER_OF_SLAVES_NOT_SUPPORTED</p> <p>2: SYSTEM_IO_NR_INVALID</p> <p>3: INDEX_FROM_SLOT_ERROR</p> <p>4: MASTER_CONFIG_INVALID</p> <p>5: MASTER_TYPE_ERROR</p> <p>6: SLAVE_DIAG_ADDR_INVALID</p> <p>7: SLAVE_ADDR_INVALID</p> <p>8: SLAVE_MODULE_IO_CONFIG_INVALID</p> <p>9: LOG_ADDR_ALREADY_IN_USE</p> <p>10: NULL_PTR_CHECK_ERROR</p> <p>11: IO_MAPPING_ERROR</p> <p>12: ERROR</p>
0xEC04	<p>EtherCAT Multiple configuration of a periphery address</p> <p>ZInfo1: Periphery address</p> <p>ZInfo2: Slot</p>
0xEC10	<p>EtherCAT restoration bus with its slaves</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xEC10</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 with input address in ZInfo1,</p> <p>XX=0x55 with output address.</p> <p>YY=0x00 Station not available,</p> <p>YY=0x01 Station available (process data)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnostics address of the master</p> <p>ZInfo3: Number of stations, which are not in the same state as the master (&gt; 0)</p>
0xEC11	<p>EtherCAT restoration bus with missing slaves</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xEC11</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 with input address in ZInfo1,</p> <p>XX=0x55 with output address.</p> <p>YY=0x00 Station not available,</p> <p>YY=0x01 Station available (process data)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnostics address of the master</p> <p>ZInfo3: Number of stations, which are not in the same state as the master (&gt; 0)</p>

Event-ID	Description
0xEC12	<p>EtherCAT restoration slave</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xEC12</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 with input address in ZInfo1, XX=0x55 with output address.</p> <p>YY=0x00 Station not available, YY=0x01 Station available (process data)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnostics of the Station</p> <p>ZInfo3: AIStatusCode</p>
0xEC30	<p>EtherCAT topology OK</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xEC30</p> <p>ZInfo2: Diagnostics address of the master</p>
0xEC50	<p>EtherCAT DC not in Sync</p> <p>ZInfo1: Diagnostics address of the master</p>
0xED10	<p>EtherCAT bus failure</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xED10</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 with input address in ZInfo1, XX=0x55 with output address.</p> <p>YY=0x00 Station not available, YY=0x01 Station available (process data)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnostics address of the master</p> <p>ZInfo3: Number of stations, which are not in the same state as the master</p>
0xED12	<p>EtherCAT failure slave</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xED12</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 with input address in ZInfo1, XX=0x55 with output address.</p> <p>YY=0x00 Station not available, YY=0x01 Station available (process data)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnostics of the Station</p> <p>ZInfo3: AIStatusCode</p>



Event-ID	Description
0xED20	<p>EtherCAT bus state change without calling OB86</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xED20</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 with input address in ZInfo1, XX=0x55 with output address.</p> <p>YY=0x00 Station not available, YY=0x01 Station available (process data)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnostics address of the master</p> <p>ZInfo3: Number of stations, which are not in the same state as the master</p>
0xED22	<p>EtherCAT bus state change without calling OB86</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xED22</p> <p>DatID:</p> <p>0xXXYY:</p> <p>XX=0x54 with input address in ZInfo1, XX=0x55 with output address.</p> <p>YY=0x00 Station not available, YY=0x01 Station available (process data)</p> <p>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</p> <p>ZInfo2: Diagnostics of the Station</p> <p>ZInfo3: AIStatusCode</p>
0xED30	<p>EtherCAT Topology Mismatch</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xED30</p> <p>ZInfo2: Diagnostics address of the master</p>
0xED31	<p>EtherCAT Interrupt Queue Overflow</p> <p>OB start Info (Local data) StartEvent and Eventclass: 0xED31</p> <p>ZInfo2: Diagnostics address of the master</p>
0xED40 ... 0xED4F	Internal error - Please contact the VIPA Hotline!
0xED50	<p>EtherCAT DC not in Sync</p> <p>ZInfo1: Diagnostics address of the master</p>

Event-ID	Description
0xED60	EtherCAT: Diagnostics buffer CP: Slave state change PK: 0 OB: PLC-Mode DatID 1/2: 0 ZInfo1: 0x00YY: YY: new EtherCAT state of the slave ZInfo 2: EtherCAT station address ZInfo3: AIStatusCode (EtherCAT specific error code)
0xED61	EtherCAT: Diagnostics buffer CP: CoE emergency PK: EtherCAT station address (low byte) OB: EtherCAT station address (high byte) DatID 1/2: Error code ZInfo1: 0xYYZZ: YY: Error register ZZ: MEF byte 1 ZInfo 2: 0xYYZZ: YY: MEF byte 2 ZZ: MEF byte 3 ZInfo3: 0xYYZZ: YY: MEF byte 4 ZZ: MEF byte 5
0xED62	EtherCAT: Diagnostics buffer CP: Error on SDO access during state change PK: EtherCAT station address (low byte) OB: EtherCAT station address (high byte) DatID 1/2: Subindex ZInfo1: Index ZInfo 2: SDO error code (high word) ZInfo3: SDO error code (low word)
0xED70	EtherCAT: Diagnostics buffer CP: Twice HotConnect group found PK: 0 OB: PLC-Mode DatID 1/2: 0 ZInfo1: Diagnostics address of the master ZInfo 2: EtherCAT station address ZInfo3: 0

Event-ID	Description
0xEE00	Additional information at UNDEF_OPCODE
0xEE01	Internal error - Please contact the VIPA Hotline!
0xEEEE	CPU was completely overall reset, since after PowerON the start-up could not be finished.
0xEF11 ... 0xEF13	Internal error - Please contact the VIPA Hotline!
0xEFFF	Internal error - Please contact the VIPA Hotline!
PK: C-Source module number   DatID: Line number	

### 4.19 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'*.

You can modify and/or display the status of variables 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.



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

The processing of the states may be interrupted by means of jump commands or by timer and process-related alarms. 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.

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 "?".

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.

**'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 process image of the selected operands. During the "processing check" or in operating mode STOP the periphery is read directly from the inputs. Otherwise only the process image of the selected operands is displayed.

*Control of outputs*

It is possible to check the wiring and proper operation of output modules. You can set outputs to any desired status with or without a control program. The process image is not modified but outputs are no longer inhibited.

*Control of 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. Process variables are controlled asynchronously to the execution sequence of the program.

## 5 Deployment PtP communication

### 5.1 Fast introduction

<b>General</b>	The CPU has a RS485 interface, which is per default set to PtP communication (point to point). This allows to connect via serial process connection to different source or target systems.
<b>Protocols</b>	The protocols respectively procedures ASCII, STX/ETX, 3964R, USS and Modbus are supported.
<b>Parameterization</b>	The parameterization 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.
<b>Communication</b>	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 SLIO 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

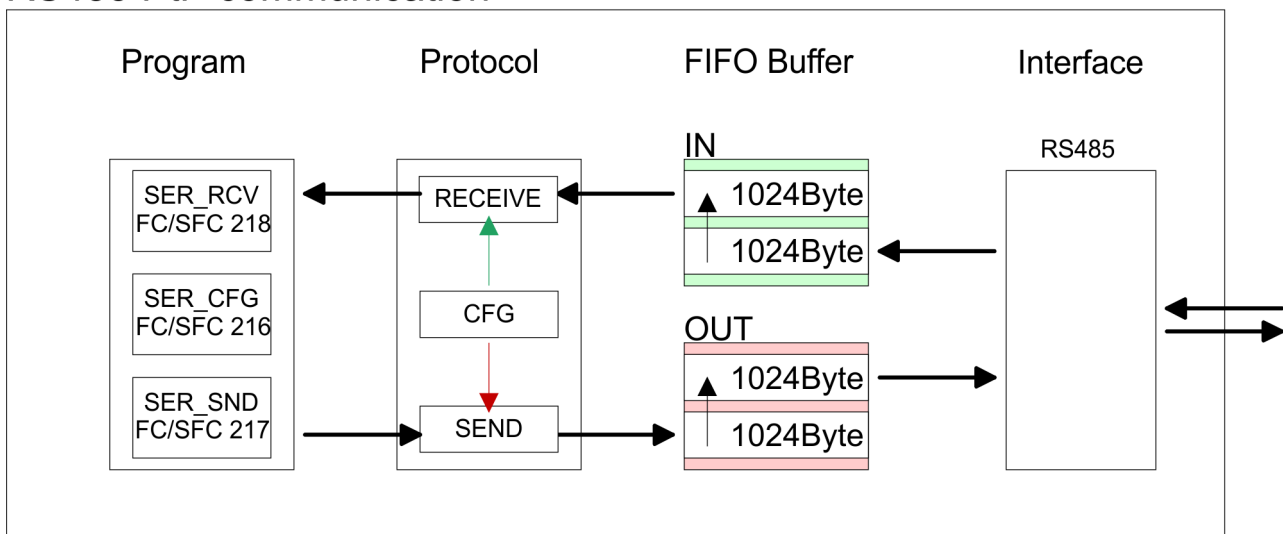
## 5.2 Principle of the data transfer

### Overview

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.

### RS485 PtP communication



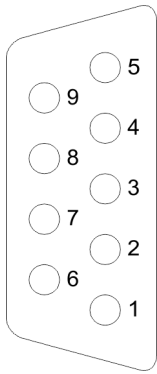
## 5.3 Deployment of RS485 interface for PtP

### Properties RS485

- Logical states represented by voltage differences between the two cores of a twisted pair cable
- Serial bus connection in two-wire technology using half duplex mode
- Data communications up to a max. distance of 500m
- Data communication rate up to 115.2kbaud

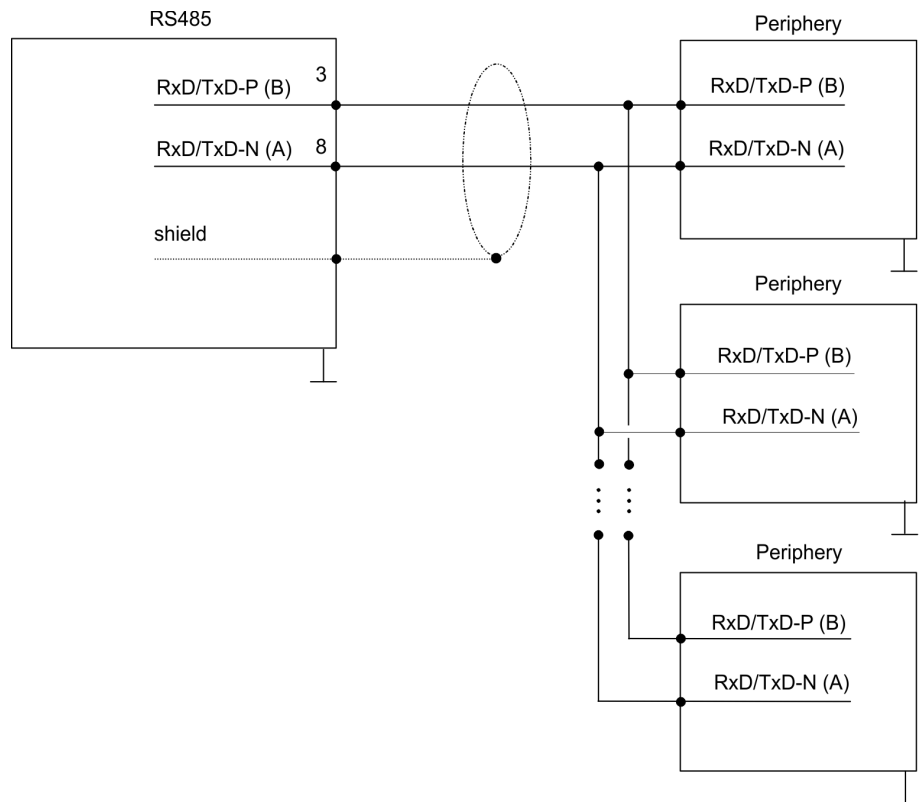
### RS485

9pin SubD jack



Pin	RS485
1	n.c.
2	M24V
3	RxD/TxD-P (Line B)
4	RTS
5	M5V
6	P5V
7	P24V
8	RxD/TxD-N (Line A)
9	n.c.

**Connection**



**5.4 Parameterization**

**5.4.1 FC/SFC 216 - SER\_CFG**

**Description** The parameterization 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.

**Parameters**

Parameter	Declaration	Data type	Description
PROTOCOL	IN	BYTE	1=ASCII, 2=STX/ETX, 3=3964R
PARAMETER	IN	ANY	Pointer to protocol-parameters
BAUDRATE	IN	BYTE	Number of baudrate
CHARLEN	IN	BYTE	0=5bit, 1=6bit, 2=7bit, 3=8bit
PARITY	IN	BYTE	0=Non, 1=Odd, 2=Even
STOPBITS	IN	BYTE	1=1bit, 2=1.5bit, 3=2bit
FLOWCONTROL	IN	BYTE	1 (fix)
RETVAL	OUT	WORD	Return value (0 = OK)

All time settings for timeouts must be set as hexadecimal value. Find the Hex value by multiply the wanted time in seconds with the baudrate.

Example:

Wanted time 8ms at a baudrate of 19200baud

Calculation:  $19200\text{bit/s} \times 0.008\text{s} \approx 154\text{bit} \rightarrow (9\text{Ah})$

Hex value is 9Ah.

**PROTOCOL**

Here you fix the protocol to be used.

You may choose between:

- 1: ASCII
- 2: STX/ETX
- 3: 3964R
- 4: USS Master
- 5: Modbus RTU Master
- 6: Modbus ASCII Master

**PARAMETER (as DB)**

At ASCII protocol, this parameter is ignored.

At STX/ETX, 3964R, USS and Modbus you fix here a DB that contains the communication parameters and has the following structure for the according protocols:

Data block at STX/ETX			
DBB0:	STX1	BYTE	(1. Start-ID in hexadecimal)
DBB1:	STX2	BYTE	(2. Start-ID in hexadecimal)
DBB2:	ETX1	BYTE	(1. End-ID in hexadecimal)
DBB3:	ETX2	BYTE	(2. End-ID in hexadecimal)
DBW4:	TIMEOUT	WORD	(max. delay time between 2 telegrams)





The start res. end sign should always be a value <20, otherwise the sign is ignored!

With not used IDs please always enter FFh!

Data block at 3964R

DBB0:	Prio	BYTE	(The priority of both partners must be different)
DBB1:	ConnAttmptNr	BYTE	(Number of connection trials)
DBB2:	SendAttmptNr	BYTE	(Number of telegram retries)
DBB4:	CharTimeout	WORD	(Char. delay time)
DBW6:	ConfTimeout	WORD	(Acknowledgement delay time )

Data block at USS

DBW0:	Timeout	WORD	(Delay time)
-------	---------	------	--------------

Data block at Modbus master

DBW0:	Timeout	WORD	(Respond delay time)
-------	---------	------	----------------------

**BAUDRATE**

Velocity of data transfer in bit/s (baud)

04h:	1200baud	05h:	1800baud	06h:	2400baud	07h:	4800baud
08h:	7200baud	09h:	9600baud	0Ah:	14400baud	0Bh:	19200baud
0Ch:	38400baud	0Dh:	57600baud	0Eh:	115200baud		

**CHARLEN**

Number of data bits where a character is mapped to.

0: 5bit	1: 6bit	2: 7bit	3: 8bit
---------	---------	---------	---------

**PARITY**

The parity is -depending on the value- even or odd. For parity control, the information bits are extended with the parity bit, that amends via its value ("0" or "1") the value of all bits to a defined status. If no parity is set, the parity bit is set to "1", but not evaluated.

0: NONE	1: ODD	2: EVEN
---------	--------	---------

**STOPBITS**

The stop bits are set at the end of each transferred character and mark the end of a character.

1: 1bit

2: 1.5bit

3: 2bit

**FLOWCONTROL**

The parameter *FLOWCONTROL* is ignored. When sending RTS=1, when receiving RTS=0.

**RETVAL FC/SFC 216 (Return values)**

Return values send by the block:

Error code	Description
0000h	no error
809Ah	Interface not found e. g. interface is used by PROFIBUS In the VIPA SLIO CPU and FeatureSet PTP_NO only the ASCII protocol is configurable. If another protocol is selected the FC/SFC216 also leave with this error code.
8x24h	Error at FC/SFC-Parameter x, with x: 1: Error at <i>PROTOCOL</i> 2: Error at <i>PARAMETER</i> 3: Error at <i>BAUDRATE</i> 4: Error at <i>CHARLENGTH</i> 5: Error at <i>PARITY</i> 6: Error at <i>STOPBITS</i> 7: Error at <i>FLOWCONTROL</i>
809xh	Error in FC/SFC parameter value x, where x: 1: Error at <i>PROTOCOL</i> 3: Error at <i>BAUDRATE</i> 4: Error at <i>CHARLENGTH</i> 5: Error at <i>PARITY</i> 6: Error at <i>STOPBITS</i> 7: Error at <i>FLOWCONTROL</i>
8092h	Access error in parameter DB (DB too short)
828xh	Error in parameter x of DB parameter, where x: 1: Error 1. parameter 2: Error 2. parameter...

**5.5 Communication****5.5.1 Overview**

The communication happens via the send and receive blocks FC/SFC 217 (SER\_SND) and FC/SFC 218 (SER\_RCV). The FCs/SFCs are included in the consignment of the CPU.

### 5.5.2 FC/SFC 217 - SER\_SND

**Description** 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.

**Parameters**

Parameter	Declaration	Data type	Description
DATAPTR	IN	ANY	Pointer to Data Buffer for sending data
DATALEN	OUT	WORD	Length of data sent
RETVAL	OUT	WORD	Return value (0 = OK)

**DATAPTR** Here you define a range of the type Pointer for the send buffer where the data to be sent are stored. You have to set type, start and length.  
 Example:  
 Data is stored in DB5 starting at 0.0 with a length of 124byte.  
 DataPtr:=P#DB5.DBX0.0 BYTE 124

**DATALEN** Word where the number of the sent Bytes is stored.  
 At **ASCII** if data were sent by means of FC/SFC 217 faster to the serial interface than the interface sends, the length of data to send could differ from the DATALEN due to a buffer overflow. This should be considered by the user program.  
 With **STX/ETX, 3964R, Modbus** and **USS** always the length set in *DATAPTR* is stored or 0.

**RETVAL FC/SFC 217 (Return values)** Return values of the block:

Error code	Description
0000h	Send data - ready
1000h	Nothing sent (data length 0)
20xxh	Protocol executed error free with xx bit pattern for diagnosis
7001h	Data is stored in internal buffer - active (busy)
7002h	Transfer - active
80xxh	Protocol executed with errors with xx bit pattern for diagnosis (no acknowledgement by partner)
90xxh	Protocol not executed with xx bit pattern for diagnosis (no acknowledgement by partner)

Error code	Description
8x24h	Error in FC/SFC parameter x, where x: 1: Error in <i>DATAPTR</i> 2: Error in <i>DATALEN</i>
8122h	Error in parameter <i>DATAPTR</i> (e.g. DB too short)
807Fh	Internal error
809Ah	interface not found e.g. interface is used by PRO-FIBUS
809Bh	interface not configured

### Protocol specific RETVAl values

#### ASCII

Value	Description
9000h	Buffer overflow (no data send)
9002h	Data too short (0byte)

#### STX/ETX

Value	Description
9000h	Buffer overflow (no data send)
9001h	Data too long (>1024byte)
9002h	Data too short (0byte)
9004h	Character not allowed

#### 3964R

Value	Description
2000h	Send ready without error
80FFh	NAK received - error in communication
80FEh	Data transfer without acknowledgement of partner or error at acknowledgement
9000h	Buffer overflow (no data send)
9001h	Data too long (>1024byte)
9002h	Data too short (0byte)

#### USS

Error code	Description
2000h	Send ready without error
8080h	Receive buffer overflow (no space for receipt)
8090h	Acknowledgement delay time exceeded

Error code	Description
80F0h	Wrong checksum in respond
80FEh	Wrong start sign in respond
80FFh	Wrong slave address in respond
9000h	Buffer overflow (no data send)
9001h	Data too long (>1024byte)
9002h	Data too short (<2byte)

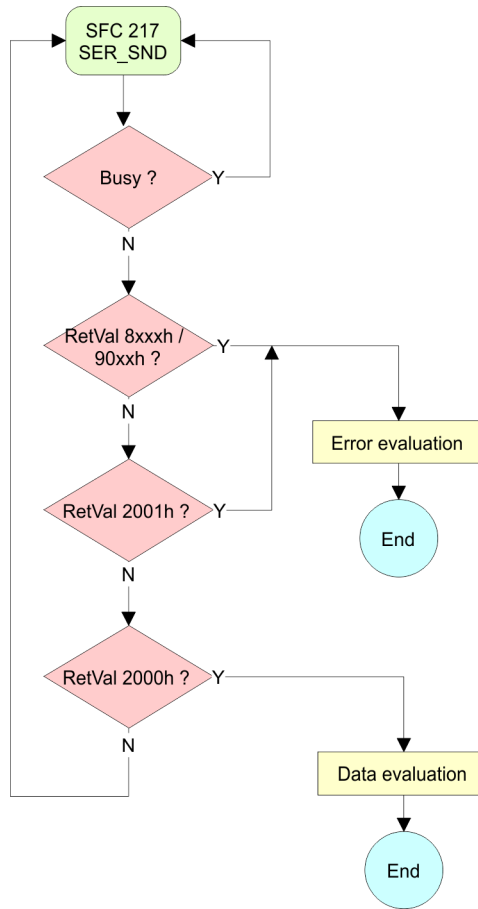
### Modbus RTU/ASCII Master

Error code	Description
2000h	Send ready (positive slave respond)
2001h	Send ready (negative slave respond)
8080h	Receive buffer overflow (no space for receipt)
8090h	Acknowledgement delay time exceeded
80F0h	Wrong checksum in respond
80FDh	Length of respond too long
80FEh	Wrong function code in respond
80FFh	Wrong slave address in respond
9000h	Buffer overflow (no data send)
9001h	Data too long (>1024byte)
9002h	Data too short (<2byte)

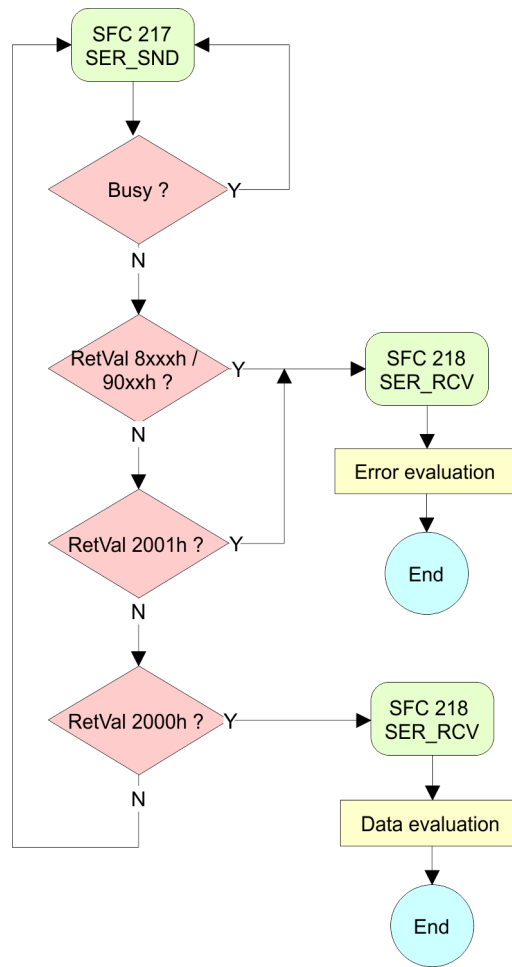
**Principles of programming**

The following text shortly illustrates the structure of programming a send command for the different protocols.

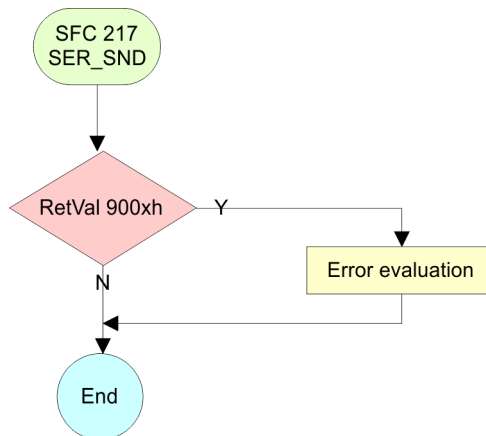
3964R



USS / Modbus



ASCII / STX/ETX



5.5.3 FC/SFC 218 - SER\_RCV

**Description**

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.

**Parameters**

Parameter	Declaration	Data type	Description
DATAPTR	IN	ANY	Pointer to Data Buffer for received data
DATALEN	OUT	WORD	Length of received data
ERROR	OUT	WORD	Error Number
RETVAL	OUT	WORD	Return value (0 = OK)

**DATAPTR**

Here you set a range of the type Pointer for the receive buffer where the reception data is stored. You have to set type, start and length.

Example:

Data is stored in DB5 starting at 0.0 with a length of 124byte.

DataPtr:=P#DB5.DBX0.0 BYTE 124

**DATALEN**

Word where the number of received Bytes is stored.

At **STX/ETX** and **3964R**, the length of the received user data or 0 is entered.

At **ASCII**, the number of read characters is entered. This value may be different from the read telegram length.

**ERROR**

This word gets an entry in case of an error.

The following error messages may be created depending on the protocol:

**ASCII**

Bit	Error	Description
0	overrun	Overflow, a sign couldn't be read fast enough from the interface
1	framing error	Error that shows that a defined bit frame is not coincident, exceeds the allowed length or contains an additional bit sequence (Stop bit error)
2	parity	Parity error
3	overflow	Buffer is full

**STX/ETX**

Bit	Error	Description
0	overflow	The received telegram exceeds the size of the receive buffer.
1	char	A sign outside the range 20h ... 7Fh has been received.
3	overflow	Buffer is full.



**3964R / Modbus RTU/ASCII Master**

Bit	Error	Description
0	overflow	The received telegram exceeds the size of the receive buffer.

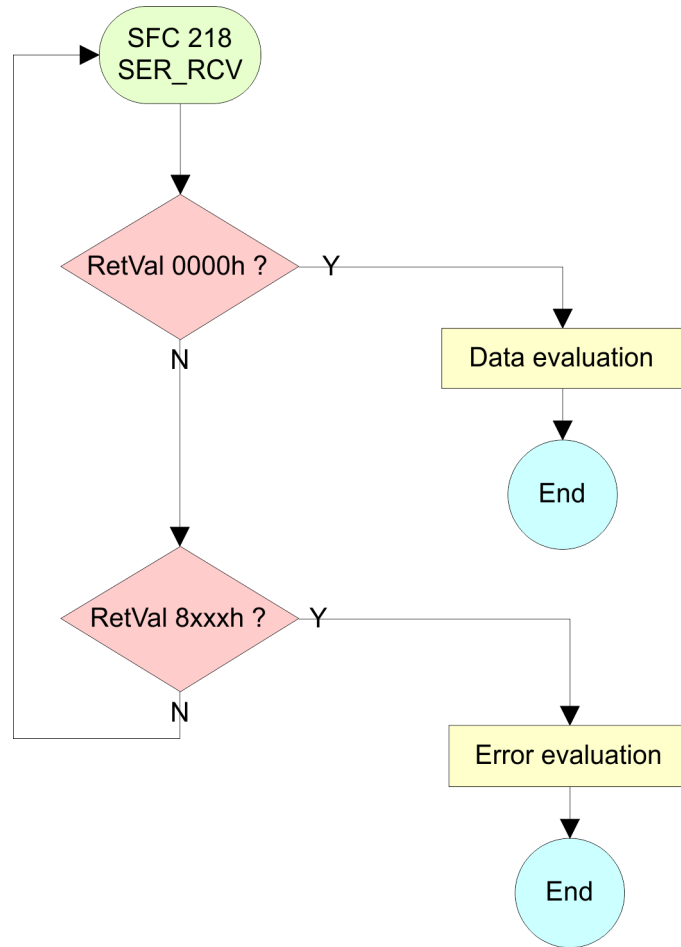
**RETVAL FC/SFC 218  
(Return value)**

Return values of the block:

Error code	Description
0000h	no error
1000h	Receive buffer too small (data loss)
8x24h	Error at FC/SFC-Parameter x, with x: 1: Error at <i>DATAPTR</i> 2: Error at <i>DATALEN</i> 3: Error at <i>ERROR</i>
8122h	Error in parameter <i>DATAPTR</i> (e.g. DB too short)
809Ah	Serial interface not found res. interface is used by PROFIBUS
809Bh	Serial interface not configured

**Principles of programming**

The following picture shows the basic structure for programming a receive command. This structure can be used for all protocols.



**5.6 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](http://www.vipa.com).

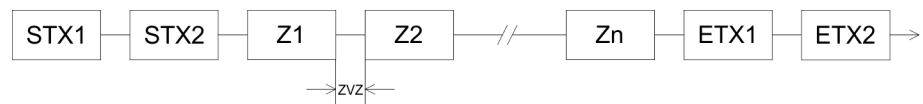
**STX/ETX**

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

- Any data transferred from the periphery must be preceded by a Start followed by the data characters and the end character. Depending of the byte width the following ASCII characters can be transferred: 5bit: not allowed: 6bit: 20...3Fh, 7bit: 20...7Fh, 8bit: 20...FFh.
- The effective data, which includes all the characters between Start and End are transferred to the CPU when the End has been received.
- When data is send from the CPU to a peripheral device, any user data is handed to the FC/SFC 217 (SER\_SND) and is transferred with added Start- and End-ID to the communication partner.
- You may work with 1, 2 or no Start- and with 1, 2 or no End-ID.
- If no End-ID is defined, all read characters are transferred to the CPU after a parameterizable character delay time (Timeout).

As Start-res. End-ID all Hex values from 01h to 1Fh are permissible. Characters above 1Fh are ignored. In the user data, characters below 20h are not allowed and may cause errors. The number of Start- and End-IDs may be different (1 Start, 2 End res. 2 Start, 1 End or other combinations). For not used start and end characters you have to enter FFh in the hardware configuration.

Message structure:



3964

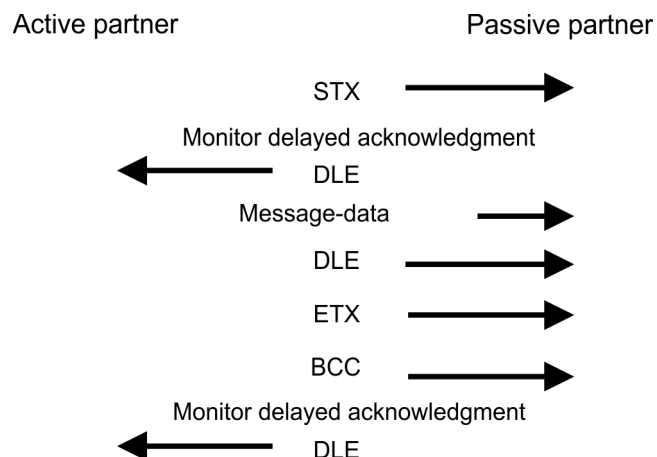
The 3964R procedure controls the data transfer of a point-to-point link between the CPU and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free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

The procedure employs the following control characters:

- STX: **S**tart of **T**ext
- DLE: **D**ata **L**ink **E**scape
- ETX: **E**nd of **T**ext
- BCC: **B**lock **C**heck **C**haracter
- NAK: **N**egative **A**cknowledge

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**

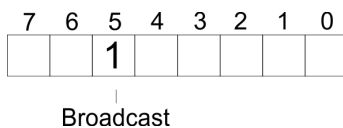
STX	LGE	ADR	PKE		IND		PWE		STW		HSW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

**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 or 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**

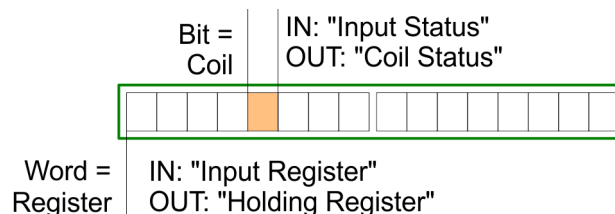
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

**5.7 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".

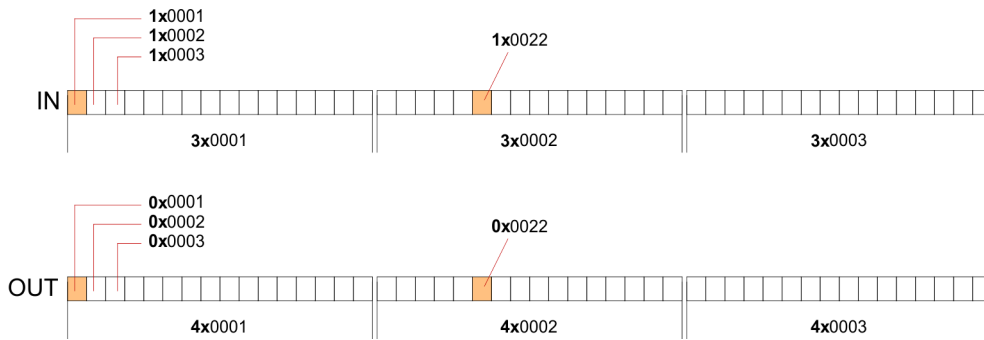
**Range definitions**

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.

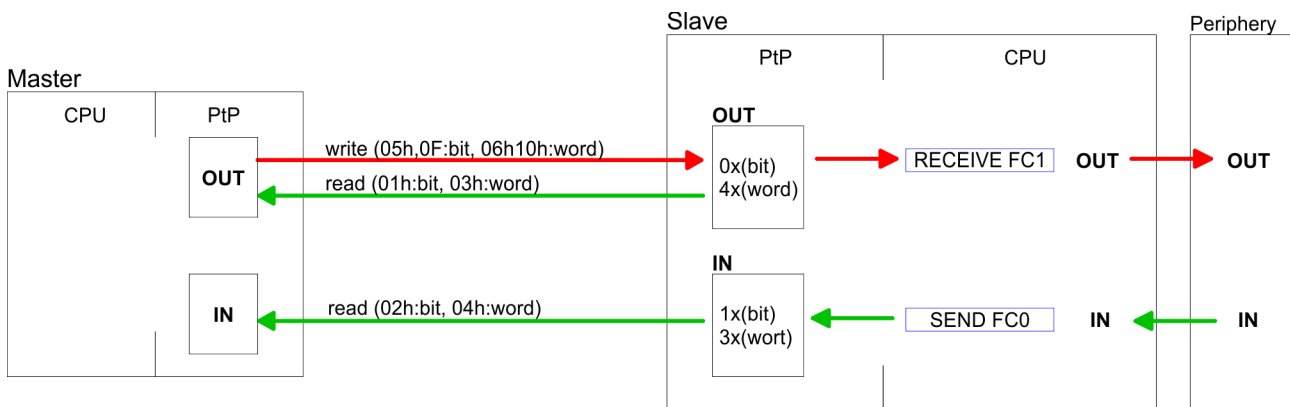
**Overview**

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

## 6 Option: PROFIBUS communication

### 6.1 Overview



#### **Enable bus functionality via VSC**

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

🔗 'Overview' on page 75

#### **PROFIBUS DP**

- PROFIBUS is an international standard applicable to an open and serial field bus for building, manufacturing and process automation that can be used to create a low (sensor-/actuator level) or medium (process level) performance network of programmable logic controllers.
- PROFIBUS comprises an assortment of compatible versions. The following details refer to PROFIBUS DP.
- PROFIBUS DP is a special protocol intended mainly for automation tasks in a manufacturing environment. DP is very fast, offers Plug'n'Play facilities and provides a cost-effective alternative to parallel cabling between PLC and remote I/O. PROFIBUS DP was designed for high-speed data communication on the sensor-actuator level.
- The data transfer referred to as "Data Exchange" is cyclical. During one bus cycle, the master reads input values from the slaves and writes output information to the slaves.

#### **CPU with DP master**

The PROFIBUS DP master is to be configured in the hardware configurator from Siemens. Here the configuration happens by the sub module X1 (MPI/DP) of the Siemens CPU. After the transmission of the data to the CPU, the configuration data are internally passed on to the PROFIBUS master part. During the start-up the DP master automatically includes his data areas into the address range of the CPU. Project engineering in the CPU is not required.

#### **Deployment of the DP master with CPU**

Via the PROFIBUS DP master PROFIBUS DP slaves may be coupled to the CPU. The DP master communicates with the DP slaves and links up its data areas with the address area of the CPU. At every POWER ON respectively overall reset the CPU fetches the I/O mapping data from the master. At DP slave failure, the OB 86 is requested. If this is not available, the CPU switches to STOP and BASP is set. As soon as the BASP signal comes from the CPU, the DP master is setting the outputs of the connected periphery to zero. The DP master remains in the operating mode RUN independent from the CPU.

## 6.2 Fast introduction

### **Overview**

The PROFIBUS DP master is to be configured in the hardware configurator. Here the configuration happens by means of the sub module X1 (MPI/DP) of the Siemens CPU.

**Enable bus functionality via VSC**

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

↳ 'Overview' on page 75

**Steps of configuration**

For the configuration of the PROFIBUS DP master please follow the following approach:

- **Enable bus functionality via VSC**
- **Hardware configuration - CPU**
- **Deployment as DP master**
- **Transfer of the complete project to CPU**



To be compatible to the Siemens SIMATIC Manager, the CPU 015 from VIPA is to be configured as

CPU 315-2 PN/DP (315-2EH14-0AB00 V3.2)

The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X1 (MPI/DP).

**6.3 Enable bus functionality via VSC****Enabling**

↳ 'Overview' on page 75

**6.4 Hardware configuration - CPU****Precondition**

The configuration of the CPU takes place at the Siemens 'hardware configurator'. The hardware configurator is part of the Siemens SIMATIC Manager. It serves for project engineering. Please use for configuration the Siemens SIMATIC Manager V 5.5 SP2 and up. The modules, which may be configured here are listed in the hardware catalog. If necessary you have to update the hardware catalog with 'Options → Update Catalog'.



For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required!

**Proceeding**

To be compatible with the Siemens SIMATIC manager the following steps should be executed:

1. ▶ Start the Siemens hardware configurator with a new project.
2. ▶ Insert a profile rail from the hardware catalog.

3. Place at 'Slot'-Number 2 the CPU 315-2 PN/DP (315-2EH14 V3.2).

Slot	Module
1	
2	<b>CPU 315-2PN/DP</b>
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
3	

The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X1 (MPI/DP).

### 6.5 Deployment as PROFIBUS DP master

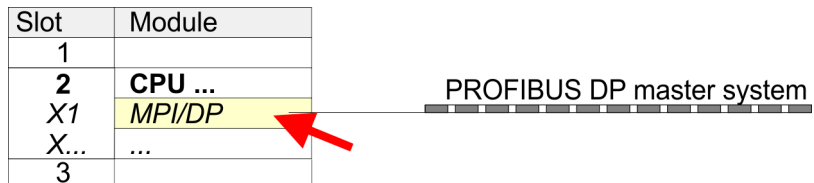
**Precondition**

The hardware configuration described before was established.

**Proceeding**

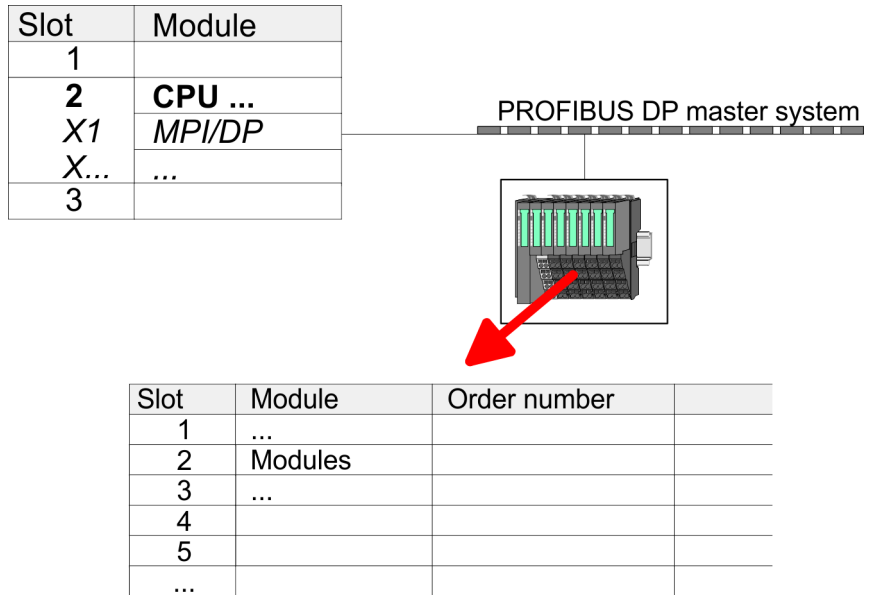
1. Open the properties dialog of the DP interface of the CPU by means of a double-click at 'MPI/DP'.
2. Set at Interface: Type "PROFIBUS".
3. Connect to PROFIBUS and preset an address (preferably 2). Confirm your input with [OK].
4. Switch at Operating mode to "DP master" and confirm the dialog with [OK].

⇒ A PROFIBUS DP master system is inserted:



Now the project engineering of your PROFIBUS DP master is finished. Please link up now your DP slaves with periphery to your DP master.

1. For the project engineering of PROFIBUS DP slaves you search the concerning PROFIBUS DP slave in the hardware catalog and drag&drop it in the subnet of your master.
2. Assign a valid PROFIBUS address to the DP slave.
3. Link up the modules of your DP slave system in the plugged sequence and add the addresses that should be used by the modules.
4. If needed, parametrize the modules.
5. Save, compile and transfer your project.



### 6.6 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.

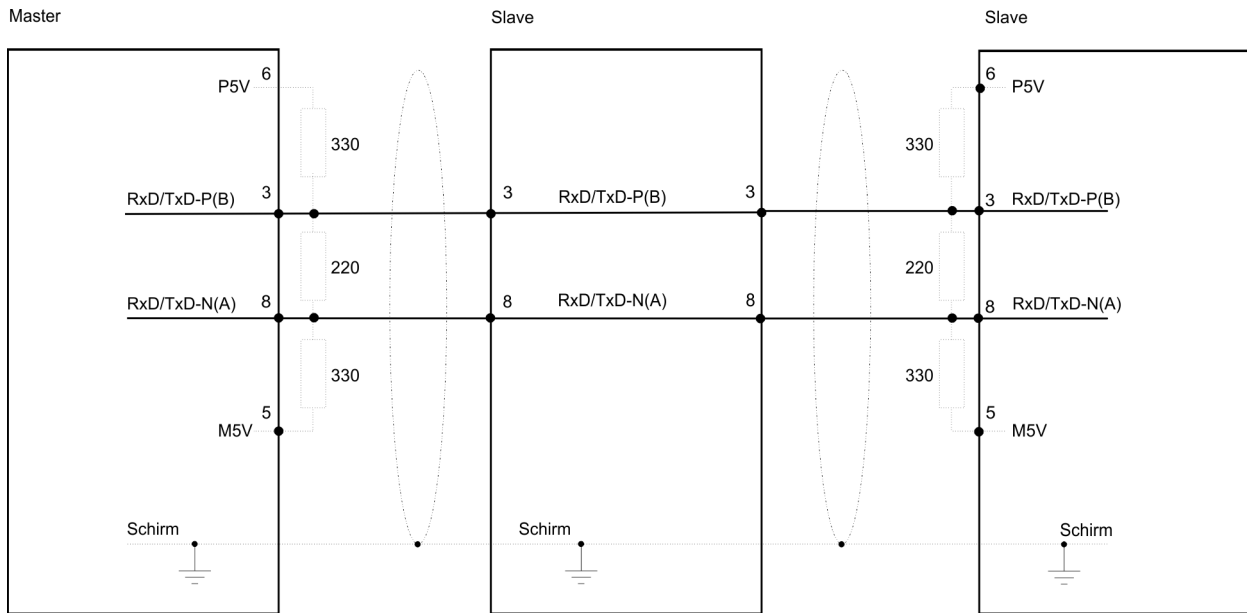
#### Transfer medium

- 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.



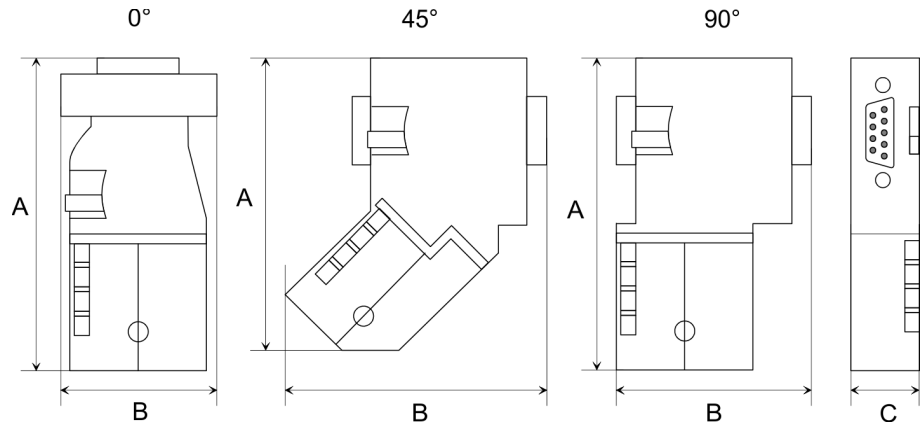
*The PROFIBUS line has to be terminated with its ripple resistor. Please make sure to terminate the last participants on the bus at both ends by activating the terminating resistor.*

**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 "EasyConn". This is a bus connector with switchable terminating resistor and integrated bus diagnostic.





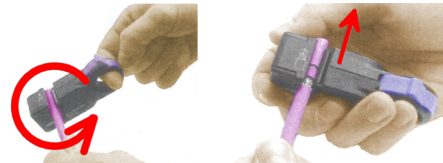
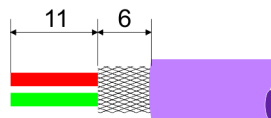
Dimensions in mm	0°	45°	90°
A	64	61	66
B	34	53	40
C	15.8	15.8	15.8



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.

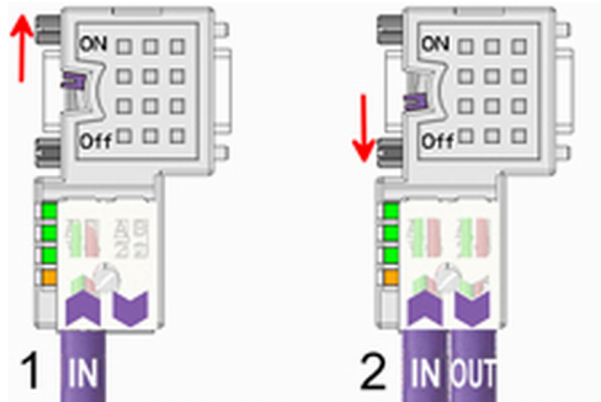


Dimensions in mm

**Termination with "EasyConn"**

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!**

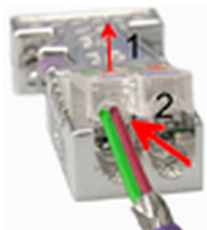
The terminating resistor is only effective, if the connector is installed at a bus participant and the bus participant is connected to a power supply.

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!*

## 6.7 Commissioning and Start-up behaviour

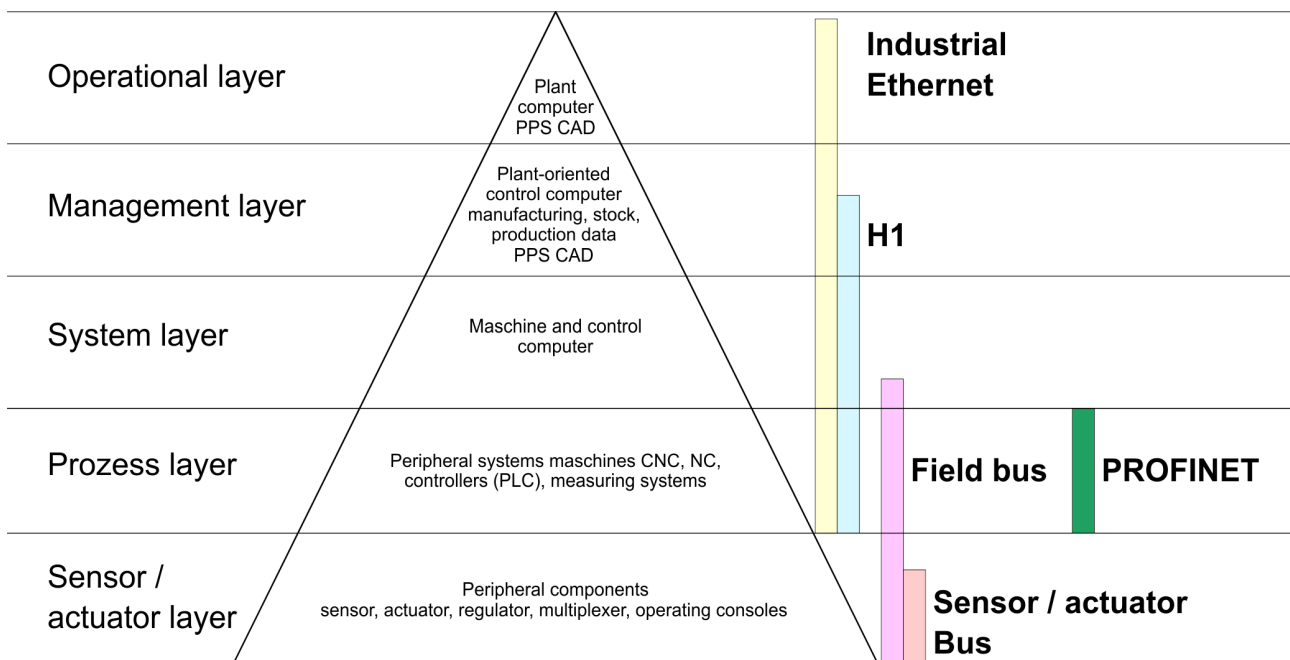
<b>Start-up on delivery</b>	In delivery the CPU is overall reset. The PROFIBUS part is deactivated and its LEDs are off after Power ON.
<b>Online with bus parameter without slave project</b>	The DP master can be served with bus parameters by means of a hardware configuration. As soon as these are transferred the DP master goes online with his bus parameter. This is shown by the RUN LED. Now the DP master can be contacted via PROFIBUS by means of his PROFIBUS address. In this state the CPU can be accessed via PROFIBUS to get configuration and DP slave project.
<b>Slave configuration</b>	If the master has received valid configuration data, he switches to <i>Data Exchange</i> with the DP Slaves. This is indicated by the DE-LED.
<b>CPU state controls DP master</b>	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 behaviour is shown by the DP master:
<b>Master behaviour at CPU STOP</b>	<ul style="list-style-type: none"> <li>■ The global control command "Clear" is sent to the slaves by the master. Here the DE-LED is blinking.</li> <li>■ DP slaves with fail safe mode were provided with output telegram length "0".</li> <li>■ DP slaves without fail safe mode were provided with the whole output telegram but with output data = 0.</li> <li>■ The input data of the DP slaves were further cyclically transferred to the input area of the CPU.</li> </ul>
<b>Master behaviour at CPU RUN</b>	<ul style="list-style-type: none"> <li>■ The global control command "Operate" is sent to the slaves by the master. Here the DE-LED is on.</li> <li>■ Every connected DP slave is cyclically attended with an output telegram containing recent output data.</li> <li>■ The input data of the DP slaves were cyclically transferred to the input area of the CPU.</li> </ul>

## 7 Deployment Ethernet communication - productive

### 7.1 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.

## 7.2 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 - 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.

<b>Layer 3 - Network layer</b>	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.
<b>Layer 4 - Transport layer</b>	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.
<b>Layer 5 - Session layer</b>	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.
<b>Layer 6 - Presentation layer</b>	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.
<b>Layer 7 - Application layer</b>	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.

### 7.3 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.

## 7.4 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 7.2 'Basics - ISO/OSI reference model' on page 129*

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 about the usage of the handling blocks may be found in the manual Operation list HB00\_OPL\_SP7 in chapter "VIPA specific blocks".

## Open communication

In the 'open communication' the communication takes place via the user program by means of handling blocks. These blocks are part of the 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.



### 7.5 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 **Network-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 and the PROFINET connection do 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 55.

Information about the assignment of IP address data to the PROFINET connection may be found in [Chapter 7.7 'Commissioning and Initialization'](#) on page 135

**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. <u>0.0.0</u>	10. <u>255.255.255</u>	255. <u>0.0.0</u>
B	172.16. <u>0.0</u>	172.31. <u>255.255</u>	255.255. <u>0.0</u>
C	192.168. <u>0.0</u>	192.168.255. <u>255</u>	255.255.255. <u>0</u>

(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.)*

**7.6 Fast introduction**

**Overview**

At the first start-up respectively at an over all reset with an PowerON again, the Ethernet PG/OP channel and PROFINET IO controller do not have any IP address. These may only be reached via its MAC address. IP address parameters may be assigned to the corresponding component by means of the MAC addresses, which may be found printed at the front with the sequence 1. address PROFINET IO controller and beneath address Ethernet PG/OP channel. The assignment takes place directly via the hardware configuration of the Siemens SIMATIC Manager.

**Steps of configuration**

For the configuration of the PROFINET IO controller for productive connections please follow the following approach:

- Assembly and commissioning
- Hardware configuration - CPU

- 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



*To be compatible to the Siemens SIMATIC Manager, the CPU 015 from VIPA is to be configured as CPU 315-2 PN/DP (6ES7 315-2EH14-0AB0 V3.2)!*

*The PROFINET controller is to be configured via the CPU sub module X2 (PN-IO).*

*The Ethernet PG/OP channel of the CPU 015 is always to be configured as CP343-1 (343-1EX21) from Siemens at slot 4.*

## 7.7 Commissioning and Initialization

### Assembly and commissioning

1. ▶ Install your System SLIO with your CPU.
2. ▶ Wire the system by connecting cables for voltage supply and signals
3. ▶ Connect your PROFINET IO controller with Ethernet.
4. ▶ Switch on the power supply.

⇒ After a short boot time, the CP is in idle.

At the first commissioning respectively after an overall reset of the CPU, the PROFINET IO controller and the Ethernet PG/OP channel have no IP address.

### Assign IP address parameters

This function is supported only if the PROFINET IO controller is not yet configured. You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens SIMATIC Manager starting with version V 5.5 & SP2 with the following proceeding:

1. ▶ Start the Siemens SIMATIC Manager.
2. ▶ Switch to "TCP/IP -> Network card ...." using 'Options → Set PG/PC interface →'.
3. ▶ Open the dialog for initialization of a station with 'PLC → Edit Ethernet node'.
4. ▶ 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 front of the CPU.
5. ▶ Choose if necessary the known MAC address of the list of found stations. To check this with [Blink] you may cause the MT LED to blink.

6. ➤ Either type in the IP configuration like IP address, subnet mask and gateway. Or your station is automatically provided with IP parameters by means of a DHCP server. Depending of the chosen option the DHCP server is to be supplied with MAC address, equipment name or client ID. The client ID is a numerical order of max. 63 characters. The following characters are allowed: Hyphen "-", 0-9, a-z, A-Z
7. ➤ Confirm with [Assign IP configuration].

Directly after the assignment the PROFINET IO controller is online reachable using the set IP address data.



*Since the IP address data, which were assigned here, are deleted at PowerOFF, you have to take them to a project by means of the hardware configuration, which is described next.*

*Initialization of the Ethernet PG/OP channel ↗ Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 55*

## 7.8 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.



*For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required!*

The configuration of the System SLIO CPU happens in the Siemens SIMATIC Manager by means of a virtual PROFINET IO device 'VIPA SLIO CPU'. The 'VIPA SLIO System' is to be installed in the hardware catalog by means of the GSDML.

### Installing the IO device VIPA SLIO System

The installation of the PROFINET IO devices 'VIPA SLIO CPU' happens in the hardware catalog with the following approach:

1. ➤ Go to the service area of [www.vipa.com](http://www.vipa.com).
2. ➤ Load from the download area at 'PROFINET files' the file System SLIO\_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 SLIO System'

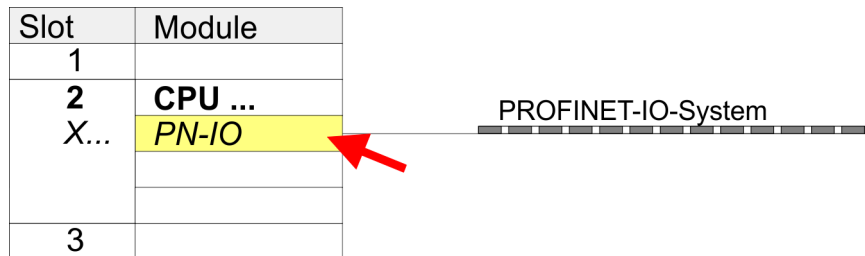
**Proceeding**

To be compatible with the Siemens SIMATIC manager the following steps should be executed:

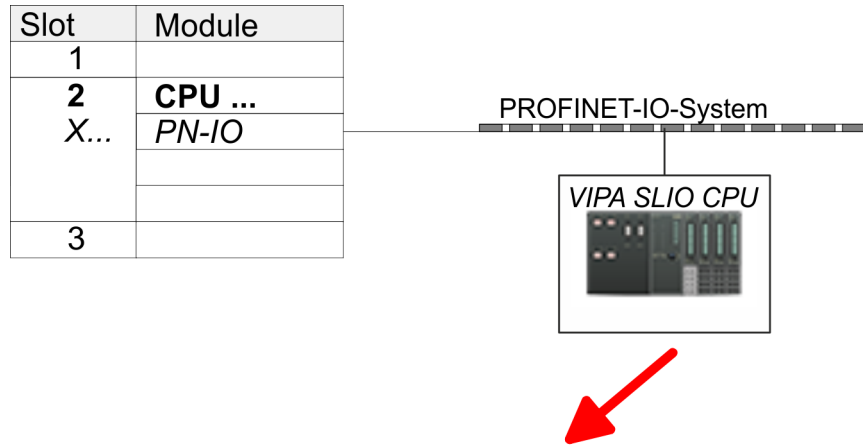
1. ▶ Start the Siemens hardware configurator with a new project.
2. ▶ Insert a profile rail from the hardware catalog.
3. ▶ Place at 'Slot'-Number 2 the CPU 315-2 PN/DP (315-2EH14 V3.2).

Slot	Module
1	
<b>2</b>	<b>CPU 315-2PN/DP</b>
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. ▶ Insert at 'General' a 'Device name' The device name must be unique at the Ethernet subnet.



Slot	Module	Order number	
0	<b>VIPA SLIO CPU ...</b>	<b>015-CEFRPR00</b>	
X2	<i>015-CEFRPR00</i>		
1			
2			
3			
...			

9. Navigate in the hardware catalog to the directory 'PROFINET IO → Additional field devices → I/O → VIPA SLIO System' and connect the IO device '015-CEFRPR00 CPU' to your PROFINET system.
  - ⇒ In the slot overview of the PROFINET IO device 'VIPA SLIO CPU' the CPU is already placed at slot 0. From slot 1 you can place your system SLIO modules.

## 7.9 Configure Siemens S7 connections

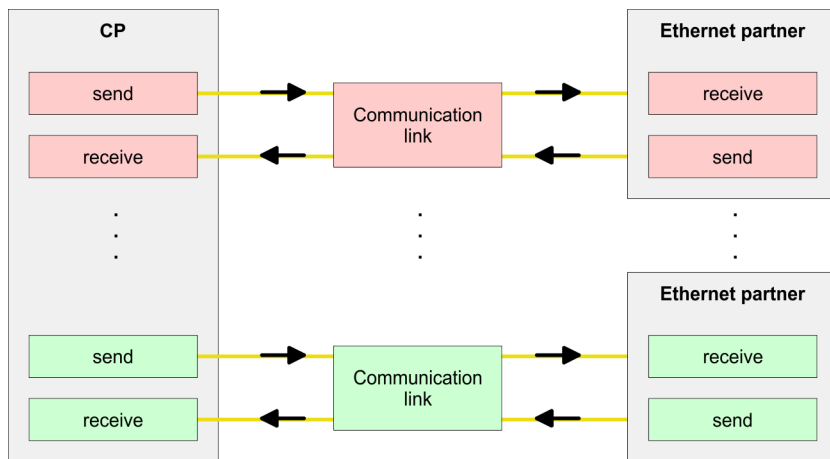
### Overview

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.

### 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 CP was assigned with IP address data by the properties of PN-IO.

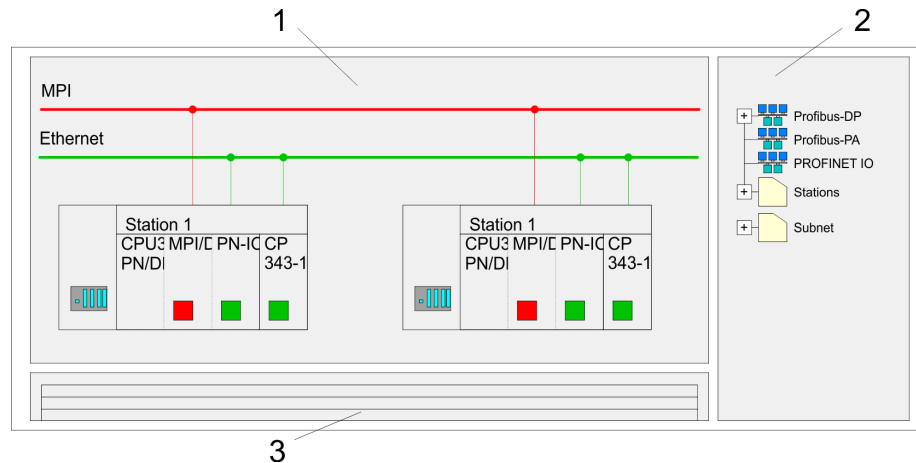


*Every station outside of the recent project must be configured as replacement objects like e.g. Siemens "SIMATIC S5" or "other station" or with the object "In unknown project". When creating a connection you may also choose the partner type "unspecified" and set the required remote parameter directly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

**Work environment of NetPro**

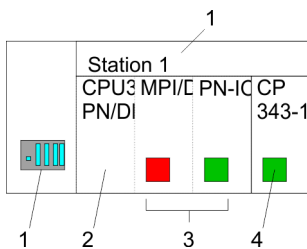
For the project engineering of connections, a thorough knowledge with NetPro from Siemens is required! The following passage only describes the basic usage of NetPro. More detailed information about NetPro is to be found in the according online manual res. documentation. Start NetPro by clicking on a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

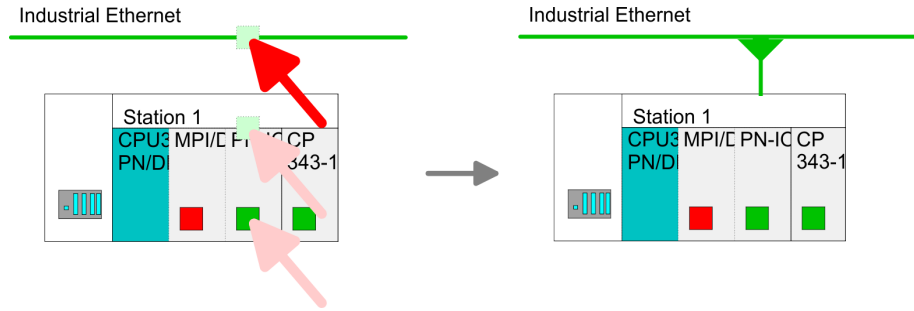
- 1 **Station:** This includes a PLC station with rack, CPU and communication components. Via the context menu you may configure a station added from the net objects and its concerning components in the hardware configurator. After returning to NetPro, the new configured components are shown.
- 2 **CPU:** 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**

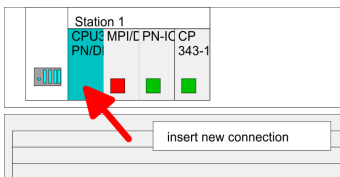
NetPro offers you the option to link-up the communicating stations. You may link-up the stations via the properties in the hardware configuration or graphically via NetPro. For this you point the mouse on the coloured net mark of the according CP and drag and drop it to the net you want to link. Now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



Configure Siemens S7 connections



**Projecting connections**

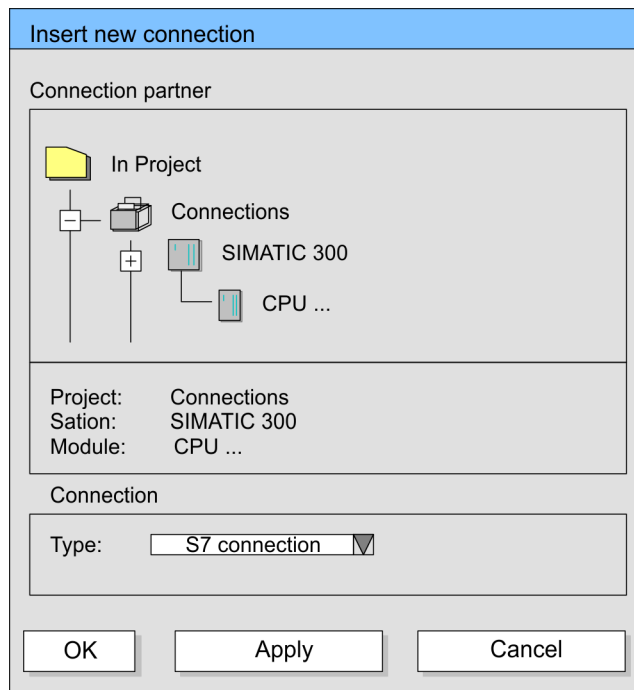


1. For the project engineering of connections, open the connection list by selecting the according CPU. Choose *Insert new connection* in the context menu:

- *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 a 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**

Exclusively Siemens S7 connection may be configured with NetPro.



*All broadcast stations and All multicast stations are not supported by this CPU.*

**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 cross-wise. The following parameters define a connection end point:

**The following parameters define a connection end point:**

Station A				Station B
remote TSAP	→	Siemens	→	local TSAP
local TSAP	←	S7 connection	←	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 8 ... 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 8 ... 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/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.

**7.10 Configure Open Communication**

**Connection-oriented protocols**

- Connection-oriented protocols establish a (logical) connection to the communication partner before data transmission is started.
- And if necessary they terminate the connection after the data transfer was finished.
- Connection-oriented protocols are used for data transmission when reliable, guaranteed delivery is of particular importance.
- In general, many logical connections can exist on one physical line.

The following connection-oriented protocols are supported with FBs for open communication via Industrial Ethernet:

- *TCP/IP native according to RFC 793 (connection types 01h and 11h)*: During data transmission, no information about the length or about the start and end of a message is transmitted. 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. 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. If you have specified the length of the data to be received greater than the length of the data to be sent, the receive block will copy the received data completely into the receive range. After this, it will set NDR to TRUE and write RCVD\_LEN with the length of the sent data. If you have specified the length of the data to be received less than the length of the sent data, the receive block will not copy any data into the receive range but instead will supply the following error information: ERROR = 1, STATUS = 8088h.

### Connection-less protocol

- There is thus no establishment and termination of a connection with a remote partner.
- Connection-less protocols transmit data with no acknowledge and with no reliable guaranteed delivery to the remote partner.

The following connection-oriented protocol is supported with FBs for open communication via Industrial Ethernet:

- *UDP according to RFC 768 (with connection type 13h)*: In this case, when calling the sending block you have to specify the address parameters of the receiver (IP address and port number). During data transmission, information on the length and the end of the message is also transmitted. In order to be able to use the sending and receiving blocks first you have to configure the local communications access point at both sides. With each new call of the sending block, you re-reference the remote partner by specifying its IP address and its port number. If you have specified the length of the data to be received greater than the length of the data to be sent, the receive block will copy the received data completely into the receive range. After this, it will set NDR to TRUE and write RCVD\_LEN with the length of the sent data. If you have specified the length of the data to be received less than the length of the sent data, the receive block will not copy any data into the receive range but instead will supply the following error information: ERROR = 1, STATUS = 8088h.

**Handling blocks**

Those in the following listed UDTs 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.

**UDTs**

FB	Label	Connection-oriented protocols: TCP native as per RFC 793,ISO on TCP as per RFC 1006	Connectionless protocol: UDP as per RFC 768
UDT 65	TCON_PAR	Data structure for assigning connection parameters	Data structure for assigning parameters for the local communications access point
UDT 66	TCON_ADR		Data structure for assigning addressing parameters for the remote partner

**FBs**

FB	Label	Connection-oriented protocols: TCP native as per RFC 793,ISO on TCP as per RFC 1006	Connectionless protocol: UDP as per RFC 768
FB 63	TSEND	Sending data	
FB 64	TRCV	Receiving data	
FB 65	TCON	Establishing a connection	Configuring the local communications access point
FB 66	TDISCON	Terminating a connection	Closing the local communications access point
FB 67	TUSEND		Sending data
FB 68	TURCV		Receiving data

**7.11 NCM diagnostic - Help for error diagnostic**

**Siemens NCM S7 diagnostic**

The VIPA PROFINET IO controller supports the Siemens NCM diagnostic tool. The NCM diagnostic tool is part of the Siemens SIMATIC Manager. This tool delivers information about the operating state of the communication functions of the online CPs dynamically.

The following diagnostic functions are available:

- Check operating state at Ethernet
- Read the diagnostic buffer of the PROFINET IO controller
- Diagnostic of connections



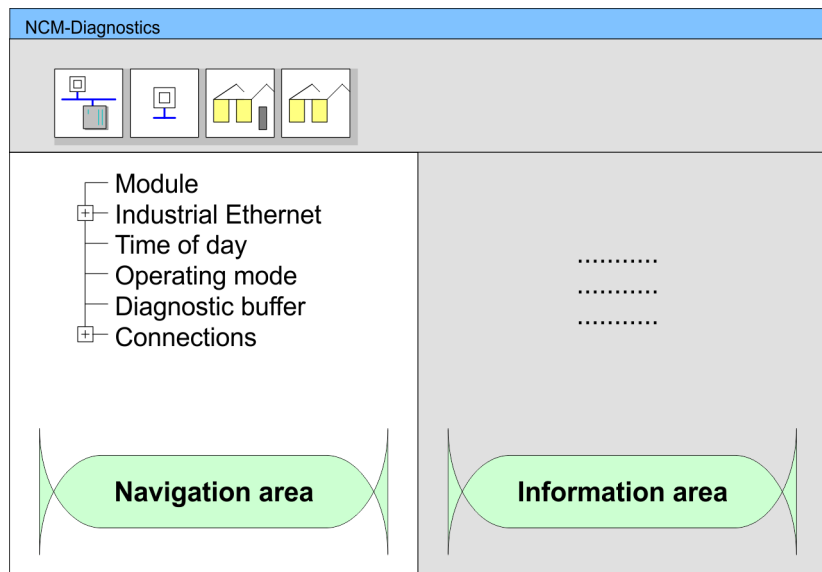
Please always enter for the PROFINET IO controller as destination parameter 0 as module rack and 125 as slot.

The following pages contain a short description of the NCM diagnostic. More details about the function range and for the deployment of the Siemens NCM diagnostic tool is to be found in the according online help res. the manual from Siemens.

**Start NCM diagnostic**

The diagnostic tool is started by 'Windows-START menu → SIMATIC → ... NCM S7 → Diagnostic'.


**Structure**



The working surface of the diagnostic tool has the following structure:

- The 'navigation area' at the left side contains the hierarchical listed diagnostic objects. Depending on CP type and configured connections there is an adjusted object structure in the navigation area.
- The 'information area' at the right side always shows the result of the navigation function you chose in the navigation area.

**No diagnostic without connection**

A diagnostic always requires an online connection to the CP you want to control. For this click at  the symbol bar.

The following dialog window appears:

Set the following parameters at destination station:

- *Attachment...*: Ind. Ethernet TCP/IP
- *Node addr.*: Enter the IP address of the CP
- *Rack/slot*: For the VIPA PROFINET IO controller please enter 0 for module rack and 125 as slot. Set your PG/PC interface to "TCP/IP -> Network card .... ". Via [OK] you start the online diagnostic.


### Read diagnostic buffer

The PROFINET IO controller has a diagnostic buffer. This has the architecture of a ring memory and may store up to 100 diagnostic messages. The NCM diagnostic allows you to monitor and evaluate the diagnostic messages via the diagnostic object Diagnostic buffer. Via a double click on a diagnostic message the NCM diagnostic shows further information.

### Approach for diagnostic

You execute a diagnostic by clicking on a diagnostic object in the navigation area. More functions are available via the menu and the symbol bar.

For the aimed diagnostic deployment the following approach is convenient:

1. ▶ Start diagnostic.
2. ▶ Open the dialog for the online connection with  enter connection parameters and establish the online connection with [OK].
3. ▶ Identify the PROFINET IO controller and check the recent state of the PROFINET IO controller via module status.
4. ▶ Check the connections for particularities like:
  - Connection status
  - Receive status
  - Send status
5. ▶ Control and evaluate the diagnostic buffer of the PROFINET IO controller via 'diagnostic buffer'.
6. ▶ As needed, alter project engineering res. programming and restart diagnostic.



## 8 Deployment Ethernet communication - PROFINET

### 8.1 Basics PROFINET

#### General

PROFINET is an open Industrial Ethernet Standard from PROFIBUS & PROFINET International (PI) for automation. PROFINET is standardized in the IEC 61158.

PROFINET uses TCP/IP and IT standards and supplements the PROFIBUS technology for applications, where fast data communication with industrial IT functions is demanded.

There are 2 PROFINET function classes:

- PROFINET IO
- PROFINET CBA

These may be realized in 3 performance steps:

- TCP/IP communication
- RT communication
- IRT communication

#### PROFINET IO

With PROFINET IO an I/O data sight to the distributed periphery is described. PROFINET IO describes the whole data transfer between IO controller and IO device. PROFINET is configured like PROFIBUS.

PROFINET IO always contains the real time concept.

Contrary to the master-slave procedure of PROFIBUS, PROFINET uses the provider-consumer model. This supports the communication relations (AR = Application Relation) between equal participants in the Ethernet. Here the provider sends its data without a request of the communication partner. Apart from the user data exchange also functions for parameterization and diagnostics are supported.

#### PROFINET CBA

PROFINET CBA means Component Based Automation. This component model describes the communication between autonomously working stations. It makes a simple modularization of complex plants possible, by distributed intelligence by means of graphic configuration for communication of intelligent modules.

#### TCP/IP Communication

This is the open communication via Ethernet TCP/IP without any demand on real-time.

#### RT Communication

RT means Real-Time. The RT communication represents the basics for data transfer at PROFINET IO. Here RT data are handled with higher priority.

#### IRT Communication

IRT means Isochronous Real-Time. With the IRT communication the bus cycle begins clock-exactly i.e. with a maximum permissible tolerance and is again synchronized. Thereby the time-controlled and synchronous transfer of data is guaranteed. Here sync telegrams of a sync master in the network serve for.

<b>Properties of PROFINET</b>	<p>PROFINET of IEC 61158 has the following properties:</p> <ul style="list-style-type: none"><li>■ Full-duplex transfer with 100MBit/s via copper respectively fibre optics.</li><li>■ Switched Ethernet</li><li>■ Auto negotiation (negotiates the transfer parameters)</li><li>■ Auto crossover (transmission and receipt lines are crossed automatically if necessary)</li><li>■ Wireless communication via Bluetooth respectively WLAN</li><li>■ UDP/IP is used as overlaid protocol. UDP means User Datagram Protocol und contains the unprotected connectionless broadcast communication within IP.</li></ul>
<b>PROFINET devices</b>	<p>Like PROFIBUS-DP also with PROFINET IO the following devices are classified according to their tasks:</p> <ul style="list-style-type: none"><li>■ IO controller</li><li>■ IO device</li><li>■ IO supervisor</li></ul>
<b>IO controller</b>	<p>The <i>IO controller</i> is equivalent to the master of PROFIBUS. This is the PLC with PROFINET connection, in which the PLC program runs.</p>
<b>IO device</b>	<p>The <i>IO device</i> is a distributed I/O field device, which is connected to PROFINET. The IO device is equal to the slave of PROFIBUS.</p>
<b>IO supervisor</b>	<p>The <i>IO supervisor</i> is an engineering station as e.g. programming unit, PC or HMI interface for commissioning and diagnostics.</p>
<b>AR</b>	<p>AR (<b>A</b>pplication <b>R</b>elation) corresponds to a connection to an IO controller or IO supervisor.</p>
<b>GSDML file</b>	<p>From VIPA there is a GSDML files for your IO device available. This file may either be found on the supplied storage media or at the download area of <a href="http://www.vipa.de">www.vipa.de</a>. Please install the GSDML file in your configuration tool. Details on the installation of the GSDML file are available from the manual supplied with your configuration tool.</p> <p>For configuration in your configuration tool every SLIO module may the found in the GSDML file as XML data.</p>
<b>Addressing</b>	<p>In contrast to the PROFIBUS address in PROFINET each device may be identified with its PROFINET interface:</p> <ul style="list-style-type: none"><li>■ IP address or MAC address</li><li>■ Device name</li></ul>
<b>Transfer medium</b>	<p>PROFINET is compatible to Ethernet in accordance with the IEEE standards. The connection of the PROFINET IO field devices is exclusively established via switches as network components. This is made either as star via multi-port switches or as line by means of switches, integrated to the field devices.</p>

## 8.2 PROFINET installation guidelines

### Generals to data security

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

### Precautions

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

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

### Guidelines for information security

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

### Industrial Ethernet

- Due to the open standard of PROFINET standard Ethernet components may be used.
- For industrial environment and due to the high transfer rate of 100MBit/s your PROFINET system should consist of Industrial Ethernet components.
- All the devices interconnected by switches are located in one and the same network. All the devices in a network can communicate directly with each other.
- A network is physically limited by a router.
- If devices need to communicate beyond the limits of a network, you have to configure the router so that it allows this communication to take place.

## Topology

- Linear
  - With the linear structure all the communication devices are connected via a linear bus topology.
  - Here the linear bus topology is realized with switches that are already integrated into the PROFINET device.
  - If a communication member fails, communication across the failed member is no longer possible.
- Star
  - If you connect communication devices to a switch with more than 2 PROFINET interfaces, you automatically create a star network topology.
  - If an individual PROFINET device fails, this does not automatically lead to failure of the entire network, in contrast to other structures.
  - It is only if a switch fails that part of the communication network will fail as well.
- Ring
  - In order to increase the availability of a network the both open ends of a linear bus topology may be connected by a switch.
  - By configuring the switch as redundancy manager on a break in the network it ensures that the data is redirected over an intact network connection.
- Tree
  - If you interconnect several star structures, you obtain a tree network topology.

## 8.3 Fast introduction

### Overview

At the first commissioning respectively after an overall reset with PowerON again of the CPU, the Ethernet PG/OP channel and the PROFINET IO controller have no IP address. These are only reachable by its MAC address. By means of the MAC address, which is printed at the front with the sequence 1. address PROFINET IO controller and beneath address Ethernet PG/OP channel, you can assign IP address data to the according component. The assignment takes place directly via the hardware configuration of the Siemens SIMATIC Manager.

### Steps of configuration

The configuration of the PROFINET IO controller for PROFINET communication should be done by the following procedure:

- Commissioning and Initialization (assignment IP address data)
- Hardware configuration - CPU
- Configuration PROFINET IO controller
- Configuration PROFINET IO device
- Transfer of the entire project to the CPU



*To be compatible with the Siemens SIMATIC Manager the CPU 015 from VIPA is to be configured as*

*CPU 315-2 PN/DP (315-2EH14-0AB00 V3.2)!*

*The integrated PROFIBUS DP master (X3) is to be configured and connected via the sub module X1 (DP). The Ethernet PG/OP channel of the CPU 015 is to be configured as 1. module as CP343-1 (343-1EX21) from Siemens after the really plugged modules at the standard bus.*

## 8.4 Commissioning and Initialization

### Assembly and commissioning

1. ▶ Install your System SLIO with your CPU.
2. ▶ Wire the system by connecting cables for voltage supply and signals
3. ▶ Connect your PROFINET IO controller with Ethernet.
4. ▶ Switch on the power supply.

⇒ After a short boot time, the CP is in idle.

At the first commissioning respectively after an overall reset of the CPU, the PROFINET IO controller and the Ethernet PG/OP channel have no IP address.

### Assign IP address parameters

This function is supported only if the PROFINET IO controller is not yet configured. You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the Siemens SIMATIC Manager starting with version V 5.5 & SP2 with the following proceeding:

1. ▶ Start the Siemens SIMATIC Manager.
2. ▶ Switch to "TCP/IP -> Network card .... " using 'Options → Set PG/PC interface → '.
3. ▶ Open the dialog for initialization of a station with 'PLC → Edit Ethernet node'.
4. ▶ 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 front of the CPU.
5. ▶ Choose if necessary the known MAC address of the list of found stations. To check this with [Blink] you may cause the MT LED to blink.
6. ▶ Either type in the IP configuration like IP address, subnet mask and gateway. Or your station is automatically provided with IP parameters by means of a DHCP server. Depending of the chosen option the DHCP server is to be supplied with MAC address, equipment name or client ID. The client ID is a numerical order of max. 63 characters. The following characters are allowed: Hyphen "-", 0-9, a-z, A-Z
7. ▶ Confirm with [Assign IP configuration].

Directly after the assignment the PROFINET IO controller is online reachable using the set IP address data.



*Since the IP address data, which were assigned here, are deleted at PowerOFF, you have to take them to a project by means of the hardware configuration, which is described next.*

*Initialization of the Ethernet PG/OP channel ↪ Chapter 4.6 'Hardware configuration - Ethernet PG/OP channel' on page 55*

## 8.5 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.



*For project engineering a thorough knowledge of the Siemens SIMATIC Manager and the Siemens hardware configurator is required!*

The configuration of the System SLIO CPU happens in the Siemens SIMATIC Manager by means of a virtual PROFINET IO device 'VIPA SLIO CPU'. The 'VIPA SLIO System' is to be installed in the hardware catalog by means of the GSDML.

### Installing the IO device VIPA SLIO System

The installation of the PROFINET IO devices 'VIPA SLIO CPU' happens in the hardware catalog with the following approach:

1. ▶ Go to the service area of [www.vipa.com](http://www.vipa.com).
2. ▶ Load from the download area at 'PROFINET files' the file System SLIO\_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 SLIO System'

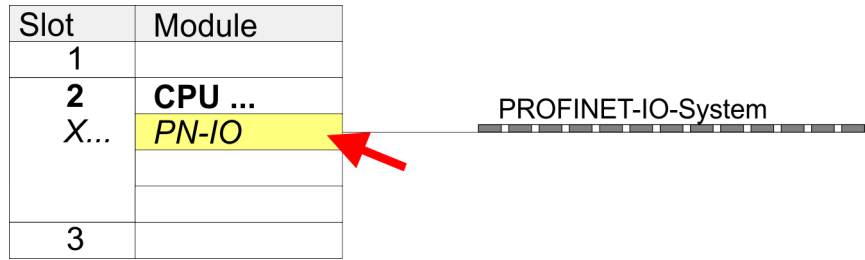
### Proceeding

To be compatible with the Siemens SIMATIC manager the following steps should be executed:

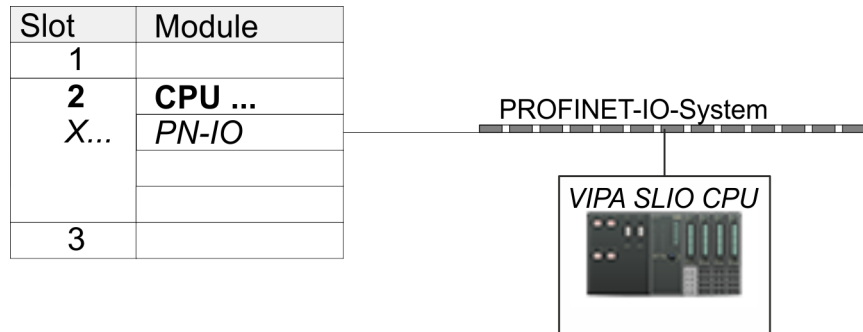
1. ▶ Start the Siemens hardware configurator with a new project.
2. ▶ Insert a profile rail from the hardware catalog.
3. ▶ Place at 'Slot'-Number 2 the CPU 315-2 PN/DP (315-2EH14 V3.2).

Slot	Module
1	
<b>2</b>	<b>CPU 315-2PN/DP</b>
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. Insert at 'General' a 'Device name' The device name must be unique at the Ethernet subnet.



Slot	Module	Order number
0	<b>VIPA SLIO CPU ...</b>	<b>015-CEFRPR00</b>
X2	015-CEFRPR00	
1		
2		
3		
...		

9. Navigate in the hardware catalog to the directory 'PROFINET IO → Additional field devices → I/O → VIPA SLIO System' and connect the IO device '015-CEFRPR00 CPU' to your PROFINET system.
  - ⇒ In the slot overview of the PROFINET IO device 'VIPA SLIO CPU' the CPU is already placed at slot 0. From slot 1 you can place your system SLIO modules.

## 8.6 Parameters - PROFINET IO controller

### 8.6.1 Precondition

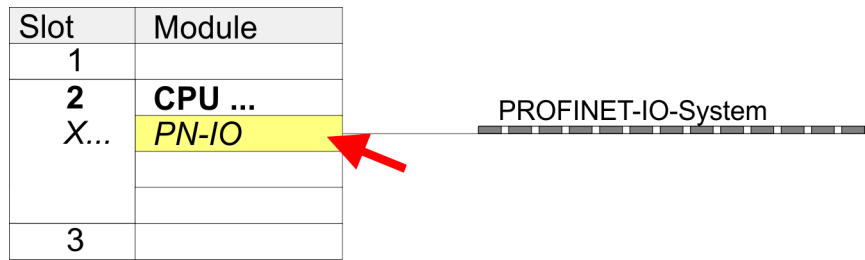
To parametrize the PROFINET IO controller of the CPU, the following conditions must be fulfilled:

- The PROFINET IO controller is online reachable, this means an initialization was established.
- The hardware configuration described before was established and the PROFINET IO controller is networked.



**Proceeding**

➔ Open the properties dialog of the PROFINET IO controller by a double-click at PN-IO.



The PROFINET interface of the PROFINET IO controller is parametrized with PN-IO, the port with Port 1. In the following these parameters for PN-IO and Port 1 are described.

**8.6.2 PN-IO**

**General**

**Short description**

Designation of the IO controller. The IO controller from VIPA always has the *short description* "PN-IO".

**Device name**

The device name on the Ethernet subnet must be unique. For an integrated PROFINET interface the device name is derived from the short description.

**Comment**

Here the purpose may be entered for which the IO controller is being used.

**Properties**

With properties you can enter the IP address, subnet mask and gateway for the PROFINET interface and select the subnet to be connected.

**Addresses**

The CPU reports errors of the IO controller via the *interface address*, as soon as e.g. an error during synchronization of the IO controller occurs. With the *PROFINET IO system address* the CPU reports e.g. failure/return of the PROFINET IO system. This address is also used to identify the IO system to which the device belongs, if an IO device fails.

**PROFINET**

With the operation field "OB82 / I/O fault task..." you can cause the CPU to call the OB 82 at an error event of the PROFINET interface. An entry to the diagnostics buffer is always done.

The other parameters in this tab are not relevant for the use of the VIPA PROFINET CPU.

**Synchronization**

This tab shows the synchronization properties of the IO controller. Here nothing can be changed.



**Time-of-day synchronization** Here you can configure time-of-day master for time-of-day synchronization in the network. NTP (**N**etwork **T**ime **P**rotocol) is used to implement a TCP/IP protocol for time-of-day synchronization in networks. In the NTP mode the module sends out time-of-day queries at regular intervals to all configured NTP servers. Based on the response from the servers, the most reliable and most exact time-of-day is determined and used to synchronize the time-of-day of the module. Configure with [Add] a NTP server and enter the update interval. The time-of-day of the module is synchronized once within this interval.

### 8.6.3 Port 1

#### General

Shown is the short name "Port...". In the field Name another designation may be selected, which is also shown in the configuration table. At *comment* you may describe your entry near more. The comment also appears in the configuration table.

#### Addresses

Via the *port* address the diagnostics information of the IO controller may be accessed.

#### Topology

These parameters serve for the handling of the ports and should not be changed.

#### Options

These parameters serve for the handling of the ports and should not be changed.

## 8.7 Configuration PROFINET IO device

### Install GSDML

The modules, which may be configured here are listed in the hardware catalog.

For the deployment of the PROFINET IO devices from VIPA you have to include the modules into the hardware catalog by means of the GSDML file from VIPA.

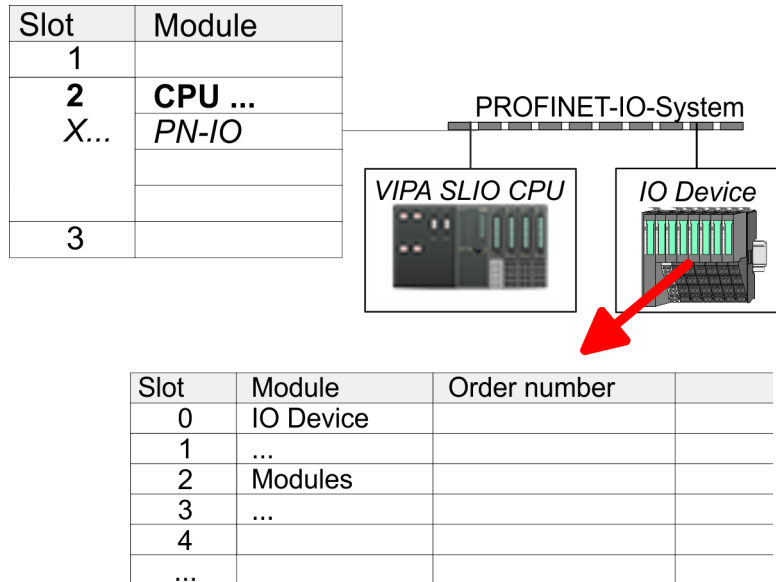
After the installation of the GSDML file the PROFINET IO devices from VIPA may be found in the hardware catalog at '*PROFINET IO* → *Additional field devices* → *I/O* → *VIPA ...*'

### Configure IO devices

Now the project engineering of the PROFINET IO controller is finished. Please link up now your IO devices with periphery to your IO controller.

1. ▶ For the project engineering of PROFINET IO device you search the concerning PROFINET IO device in the hardware catalog at *PROFINET-IO* and drag&drop it in the subnet of your IO controller.
2. ▶ Assign a name to the IO device. The configured name must match the name of the device. Information about setting the device name can be found in the manual of the IO device.
3. ▶ Enter a valid IP address. The IP address is normally assigned automatically by the hardware configurator. If this is not desired, you can assign the IP address manually.

- 4. ▶ Link up the modules of your IO device in the plugged sequence and add the addresses that should be used by the modules.
- 5. ▶ If needed, parametrize the modules.
- 6. ▶ Save, compile and transfer your project. ↪ Chapter 4.9 'Project transfer' on page 62



### 8.8 Commissioning and start-up behaviour

**Start-up on delivery**

In the delivery state the CPU is overall reset. The PROFINET part is deactivated and its LEDs are off after PowerON.

**Online with bus parameters without project**

For the communication between IO controller and IO device the ways for the communication are to be defined before. For the clear specification of the communication ways, these are established during the start-up by the IO controller, based on the project data.

Here the configuration takes place by a hardware configuration. As soon as the project data were transmitted, the IO controller switches to system start-up.

In this state the IO controller may be accessed and its CPU may be configured via Ethernet by the IO controller by means of the IP address.

**IO device configuration**

The PROFINET IO controller is configured by a hardware configuration. After the transmission of the project into the IO controller with connected IO devices, the IO controller has the whole information for the addressing of and the data exchange with the IO devices.

During the system start-up of the IO controller the IO devices are supplied with their configured IP address by means of the DCP protocol. After PowerON due to the project data the system start-up of the IO controller is initialized and it runs off independently. During the system start-up the IO controller establishes a clear communication relation (CR) and an application relation (AR) to an IO-Device. Here the cyclic IO data, the acyclic R/W services and the expected modules/sub modules are specified.

If the IO controller has received valid project engineering data, a system start-up with the IO devices is initialized and this is indicated by flashing BF LED. The BF LED is on with configured PROFINET IO device and bus cable is missing.

If at least one IO device is not in cyclic data exchange during start-up, the BF LED blinks.

If all IO devices are in cyclic data exchange, the BF LED gets off. This state does not depend on the state of the operating mode switch of the CPU. After a successful system start-up the system is ready for communication.

### CPU state influences the IO process data

After PowerON respectively a receipt of a new hardware configuration the configuration data are automatically transferred to the IO controller. Dependent on the CPU state the following behaviour is shown by the IO controller:

#### Behaviour at CPU STOP

- In the STOP state of the CPU an output telegram is further cyclically sent but this is designated as "not valid" and the output data are set to 0.
- The IO controller further receives the input data of the IO devices and transfers them cyclically to the input area of the CPU.

#### Behaviour at CPU RUN

- The IO controller cyclically reads the output data from the CPU and transfers these as telegram to the connected IO devices.
- The IO controller receives the input data of the IO devices and transfers them cyclically to the input area of the CPU.

## 8.9 PROFINET diagnostics

### 8.9.1 Overview

There are the following possibilities to get diagnostics information from your system:

- Diagnostics with the configuration and engineering tool
- Diagnostics during runtime in the user program (OB 1, SFB 52)
- Diagnostics via OB start information
- Diagnostics via status LEDs

### 8.9.2 Diagnostics with the configuration and engineering tool

If you are connected from your configuration respectively engineering tool via Ethernet with the PROFINET IO controller, online diagnostics information may be accessed.

E.g. with '*Station → Open online*' you get information about the state of your system. Here missing respectively faulty components are shown by symbols.

In the following figure e.g. there is shown that the configured device 3 is missing and device 4 reports an error.

**Diagnostics Online**

Slot	Module
1	
2	✘ CPU ...
X...	PN-IO
3	

Slot	Module	I Addr.	Q Addr.	Diag.
0	IM 053PN			8187
1	021-1BB00	20		
2	022-1BB00		30	
3	✘ 050-1BA00	24	32	
4	...			
5				

### 8.9.3 Diagnostics during runtime in the user program

With SFB 52 RDREC (read record) you can access diagnostics data from your user program e.g. in OB1. The SFB 52 RDREC operates asynchronously, that is, processing covers multiple SFB calls.



*More information about the usage of the SFB 52 may be found in the online help of your programming tool or in the manual "SPEED7 Operation list" from VIPA.*

**Example OB1**

For the cyclic access to the diagnostics data of the system SLIO module 050-1BA00 the following example may be used in the OB 1:

```

AN M10.3 'If the reading terminated (BUSY=0) and
AN M10.1 'there is no job triggered (REQ=0) then
S M10.1 'start transfer of record (REQ:=1)
L W#16#4000 'Number of record set (0x4000)
T MW12
CALL SFB 52, DB52 'Call SFB 52 with Instance DB
    REQ :=M10.1      'Trigger flag
    ID :=DW#16#0018 'Smaller addr. of mixed module
    INDEX :=MW12
    MLEN :=14        'Length record set 0x4000
                    'with 1 entry
    VALID :=M10.2   'Validity of the record set
    BUSY :=M10.3    'Flag job just running
    ERROR :=M10.4   'Error bit during read access
    STATUS :=MD14   'Error codes
    LEN :=MW16      'Length of the read record set
    RECORD :=P#M 100.0 Byte 40
                    'Target (MB100, 40byte)
U M10.1
R M10.1            'Reset REQ
    
```

**Diagnostics data**

The system SLIO module 050-1BA00 serves for 20 byte diagnostics data. The diagnostics data of the system SLIO module 050-1BA00 have the following structure:

Name:	Bytes	Function	Default
ERR_A	1	Diagnostics	00h
MODTYP	1	Module information	18h
ERR_C	1	reserved	00h
ERR_D	1	Diagnostics	00h
CHTYP	1	Channel type	76h
NUMBIT	1	Number diagnostics bits per channel	08h
NUMCH	1	Number channels of the module	01h
CHERR	1	Channel error	00h
CH0ERR	1	Channel-specific error	00h
CH1ERR...CH7ERR	7	reserved	00h
DIAG_US	4	µs ticker	00h



More information about the diagnostics data may be found in the system SLIO manual HB300\_FM\_050-1BA00.

### 8.9.4 Diagnostics via OB start information

On an error the faulty system generates a diagnostics message for the CPU. Then the CPU calls the according diagnostics OB. Here the CPU operating system transfers start information to the local data of the OB.

By evaluating the start information of the according OB you can get information about cause and location of the error.

During runtime you can access the start information with the system function SFC 6 RD\_SINFO.

Please consider that you can even read the start information in the OB himself, because the data are temporary data.

Depending on the type of error, the following OBs are called in a diagnostics event:



- OB 82 on an error of an module at the IO device (Diagnostics interrupt)
- OB 83 on inserting respectively removing a module on a IO device
- OB 86 on failure respectively return of a IO device





*More information about the OBs and their start information may be found in the online help of your programming tool and in the manual "SPEED7 operation list" from VIPA.*

### 8.9.5 Diagnostics via status LEDs

#### LEDs PROFINET IO controller X4

BF1 (Bus error)	MT (Data Exchange)	Meaning
red 	yellow 	
•	X	<ul style="list-style-type: none"> <li>■ Bus error, no connection to sub net/switch</li> <li>■ wrong transfer rate</li> <li>■ Full-duplex-transmission</li> </ul>
BB	X	<ul style="list-style-type: none"> <li>■ Failure of a connected IO device</li> <li>■ At least one IO device is not access-able</li> <li>■ Faulty configuration</li> </ul>
X	•	Maintenance event is pending.
BB*	BB*	Simultaneous blinking (4s on, 1s off) indicates a not valid configuration.
BB*	BB*	* The alternate blinking with 4Hz indicates that a firmware update of the PROFINET IO controller is executed.
•	•	Firmware update of the PROFINET IO controller is finished without error.

BF1 (Bus error)	MT (Data Exchange)	Meaning
X	BB	As long as the PROFINET IO controller is not yet configured, for identification you may cause the LED to blink by means of the "Network search of your configuration tool.
on: ●   off: ○   blinking (2Hz): BB   not relevant: X		

L/A1 (Link/ Activity)	S1 (Speed)	Meaning
green 	green 	
●	X	The PROFINET IO controller is physically connected to the Ethernet interface.
○	X	There is no physical connection.
B	X	blinks: shows Ethernet activity.
X	●	The Ethernet interface of the PROFINET IO controller has a transfer rate of 100Mbit.
X	○	The Ethernet interface of the PROFINET IO controller has a transfer rate of 10Mbit.
on: ●   off: ○   not relevant: X		

## 9 Configuration with TIA Portal

### 9.1 TIA Portal - Work environment

#### 9.1.1 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. 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.

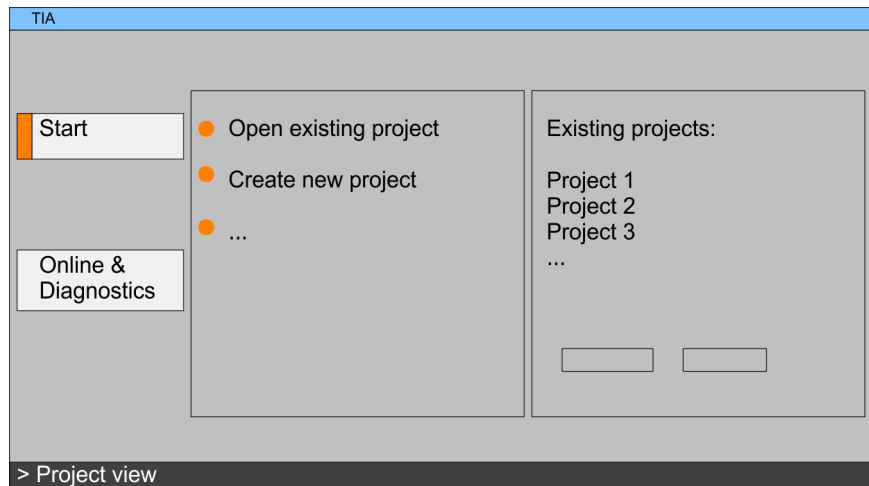


*Information about the Siemens TIA Portal can be found in the online help respectively in the according online documentation.*

##### 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.

#### 9.1.2 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.

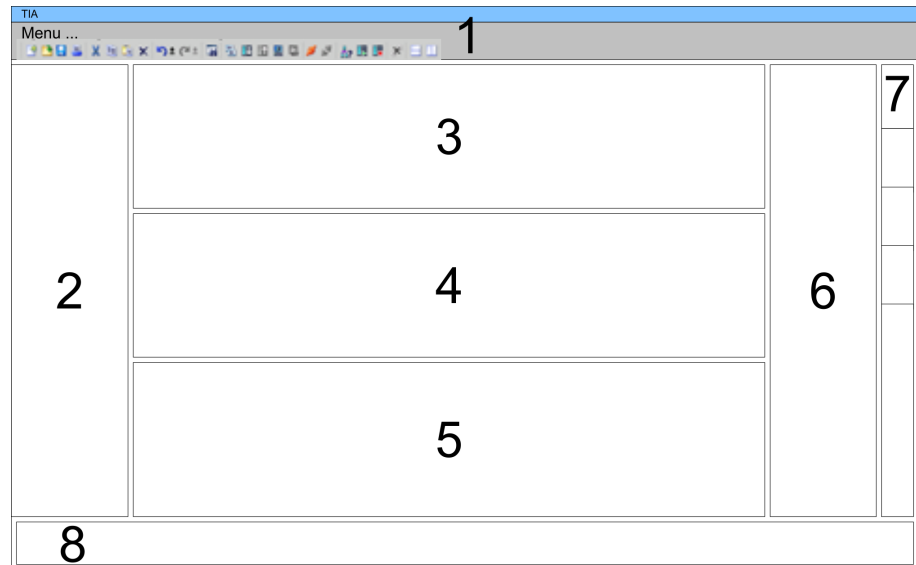
##### Project view

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

## 9.2 TIA Portal - Hardware configuration - CPU

### Overview

The hardware configuration of the CPU and its plugged modules happens in the Siemens TIA Portal by means of a virtual PROFINET IO device. For the PROFINET interface is standardized software sided, the functionality is guaranteed by including a GSDML file into the Siemens TIA Portal.

The hardware configuration of the CPU is divided into the following parts:

- Installation GSDML SLIO CPU PROFINET
- Configuration Siemens CPU
- Connection SLIO CPU as PROFINET IO device

### Installation GSDML SLIO CPU for PROFINET

The installation of the PROFINET IO devices 'VIPA SLIO CPU' happens in the hardware catalog with the following approach:

1. ▶ Go to the service area of [www.vipa.com](http://www.vipa.com).
2. ▶ Load from the download area at 'PROFINET files' the file System SLIO\_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 PRO-FINET IO device can be found at *Other field devices > PRO-FINET > IO > VIPA GmbH > VIPA SLIO System*.



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

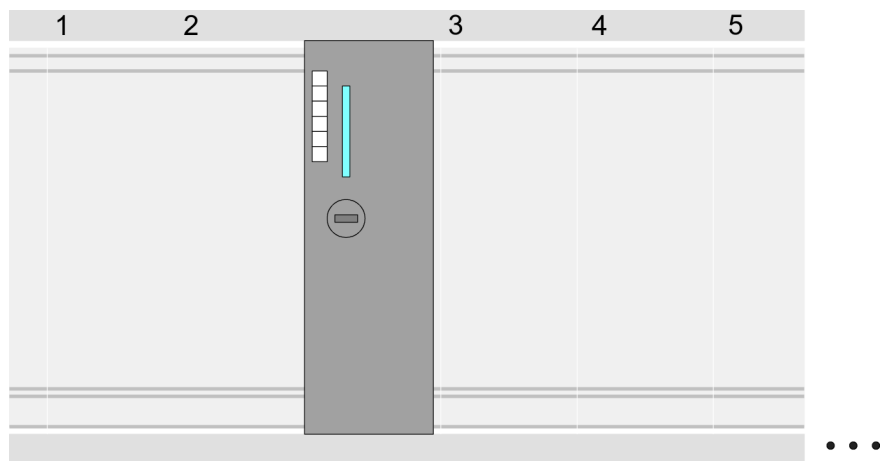
**Configuration Siemens CPU**

To be compatible with the Siemens TIA Portal, the CPU from VIPA is to be configured as CPU 315-2 PN/DP (6ES7 315-2EH14-0AB0 V3.2) from Siemens.

- 1. ▶ Start the Siemens TIA Portal.
- 2. ▶ Create a new project in the *Portal view* with 'Create new project'.
- 3. ▶ Switch to the *Project view*.
- 4. ▶ Click in the *Project tree* at 'Add new device'.
- 5. ▶ Select the following CPU in the input dialog:

SIMATIC S7-300 > CPU 315-2 PN/DP > 6ES7 315-2EH14-0AB0 V3.2

⇒ The CPU is inserted with a profile rail.



Device overview:

Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	

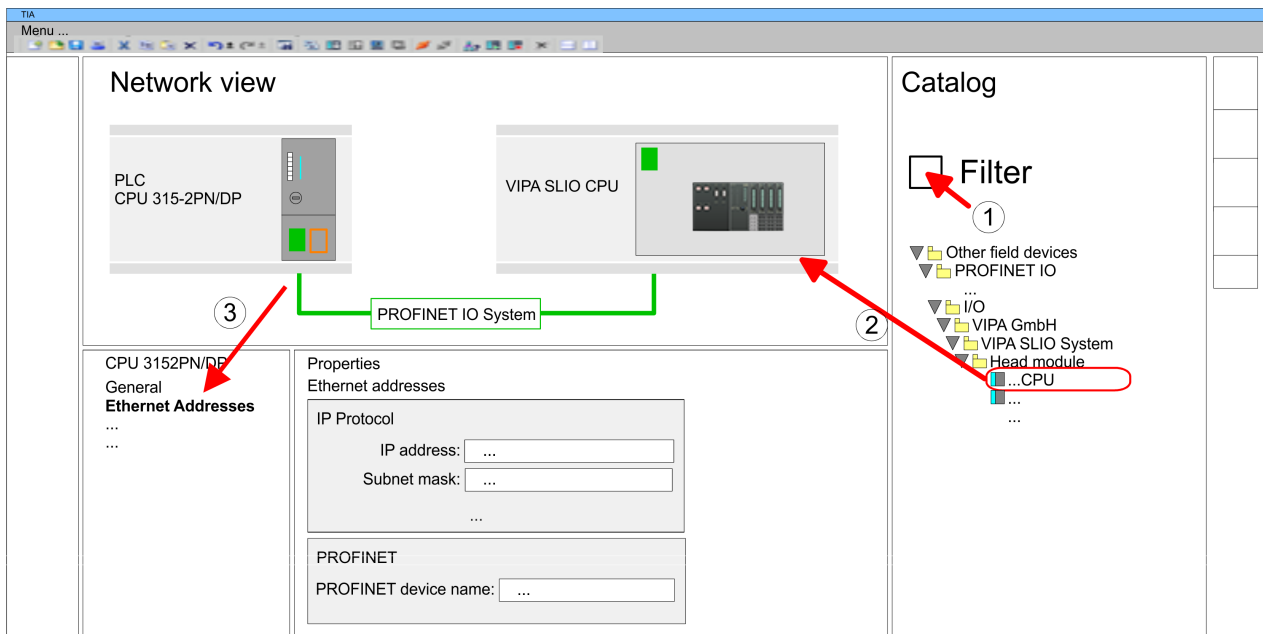
PROFINET interface	2 X2	PROFINET interface
...	...	...

**Setting standard CPU parameters**

Since the CPU 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. ↪ *'General'* on page 57

**Connection SLIO CPU as PROFINET IO device**

1. ➤ Switch in the *Project area* to *'Network view'*.
2. ➤ After installing the GSDML the IO device for the SLIO CPU may be found in the hardware catalog at *Other field devices > PROFINET > IO > VIPA GmbH > VIPA SLIO System*. Connect the slave system to the CPU by dragging&dropping it from the hardware catalog to the *Network view* and connecting it via PROFINET to the CPU.
3. ➤ Click in the *Network view* at the PROFINET part of the Siemens CPU and enter 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 SLIO CPU...'* and switch to the *Device overview*.  
 ⇒ In the *Device overview* of the PROFINET IO device *'VIPA SLIO CPU'* the CPU is already placed at slot 0. From slot 1 you can place your system SLIO modules.

**Setting VIPA specific CPU parameters**

For parametrization click at the CPU at slot 0 in the *Device overview* of the PROFINET IO device 'VIPA SLIO CPU'. 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 60*

**9.3 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 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.

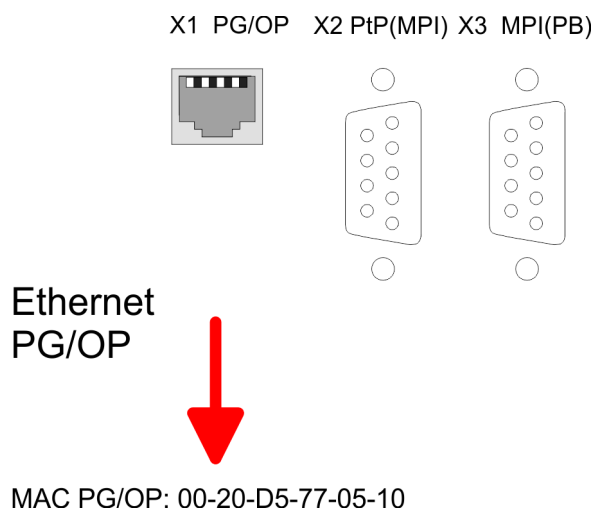
**Assembly and commissioning**

1. ➤ Install your System SLIO with your CPU.
2. ➤ Wire the system by connecting cables for voltage supply and signals.
3. ➤ Connect the Ethernet jack (X1) 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:

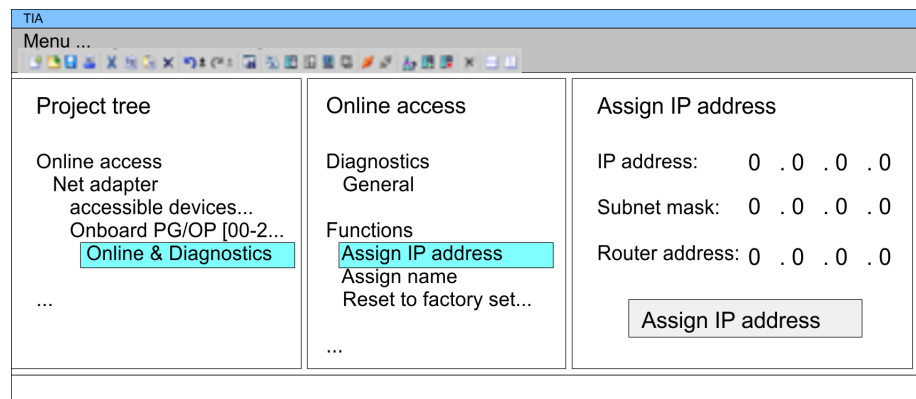
- Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This can be found at the front of the CPU labelled as "MAC PG/OP: ...".



**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:

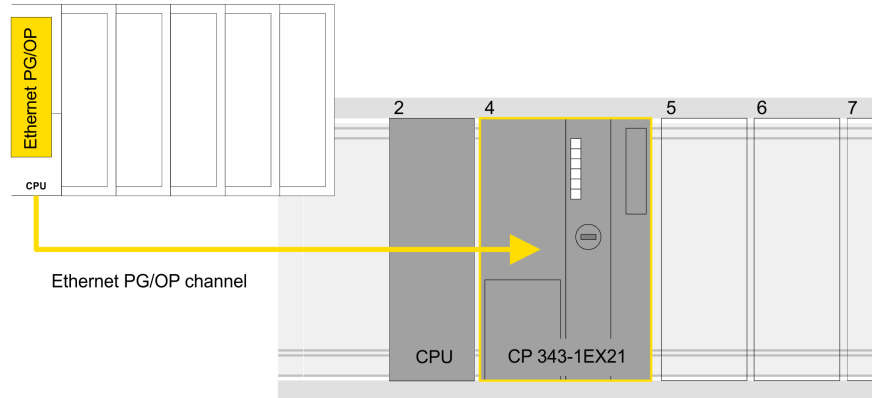
1. ▶ Start the Siemens TIA Portal.
2. ▶ Switch to the 'Project view'.
3. ▶ Click in the 'Project tree' at 'Online access' and choose here by a doubleclick your network card, which is connected to the Ethernet PG/OP channel.
4. ▶ To get the stations and their MAC address, use the 'Accessible device'. 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 315-2 PN/DP (6ES7 315-2EH14-0AB0 V3.2).
3. ▶ As Ethernet PG/OP channel place at slot 4 the Siemens CP 343-1 (6GK7 343-1EX21 0XE0 V.1.2).
4. ▶ Open the "Property" dialog by clicking on the CP 343-1EX21 and enter for the CP at "Properties" at "Ethernet address" the IP address data, which you have assigned before.
5. ▶ Transfer your project.



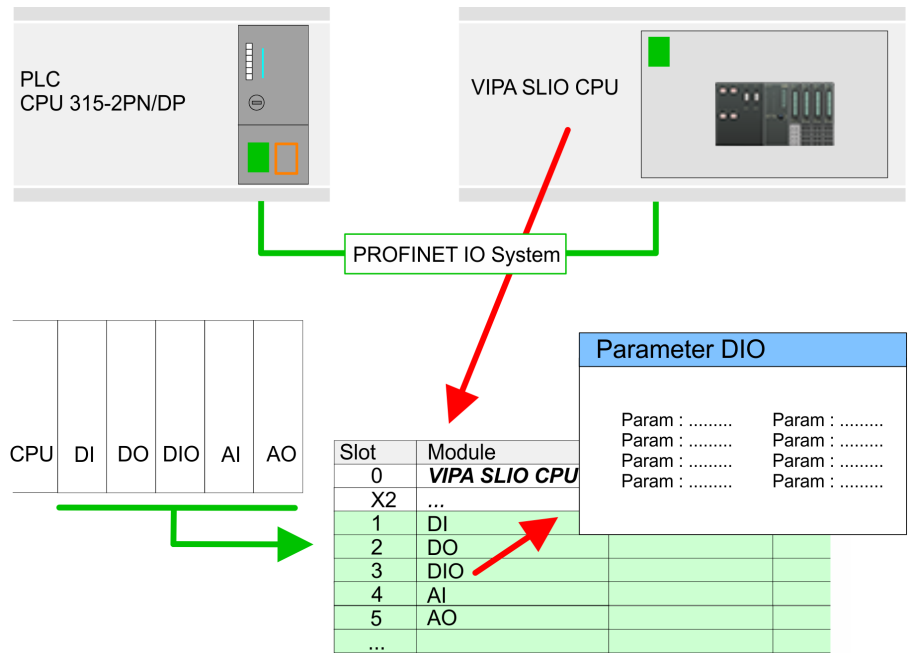
Device overview:

Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP inter- face		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	
CP 343-1		4		CP 343-1	
...		...		...	

### 9.4 Hardware Configuration - I/O modules

#### Hardware configuration of the modules

Starting with slot 1 place in the *Device overview* of the PROFINET IO device 'VIPA SLIO CPU' your System SLIO modules in the plugged sequence. For this drag from the hardware catalog the corresponding module to the corresponding position in the *Device overview*.



**Parametrization**

To provide specific addressing of the installed peripheral modules, certain addresses must be allocated in the CPU. For parametrization click in the *Device overview* at the module you want to parametrize. Then the parameters of the module are shown in the *Properties* dialog. Here you can make your parameter settings.

**9.5 TIA Portal - Include VIPA library**

**Overview**

- The VIPA specific blocks may be found at [www.vipa.com](http://www.vipa.com) as downloadable library at the "service" area with *Downloads > VIPA LIB*.
- The library is available as packed zip-file `Fx000020_V...`.
- If you want to use VIPA specific blocks, you have to import the library into your project.  
Execute the following steps:
  - Extract `Fx000020_V... .zip`
  - Open library and transfer blocks into the project

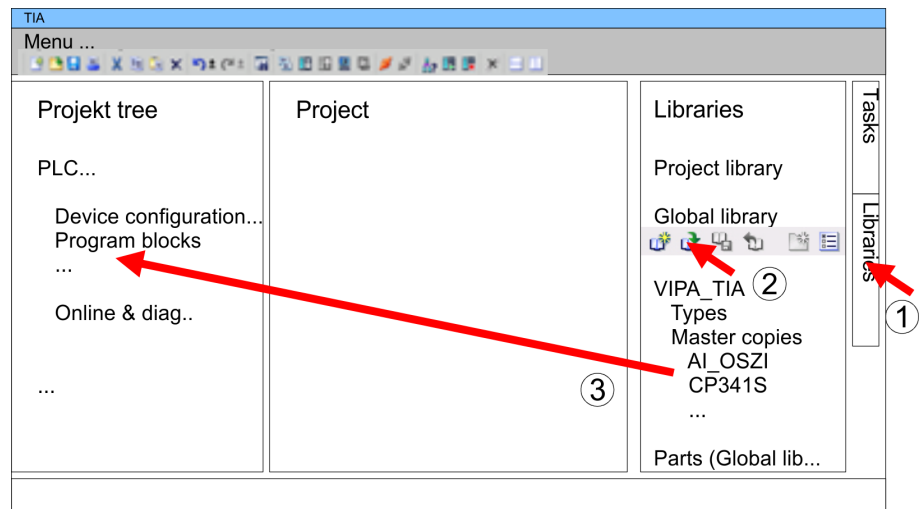
**Unzip  
Fx000020\_V... .zip**

Start your un-zip application with a double click on the file `Fx000020_V... .zip` and copy all the files and folders in a work directory for the Siemens TIA Portal.

**Open library and transfer blocks to project**

1. ▶ Start the Siemens TIA Portal with your project.
2. ▶ Select the *Project view*.
3. ▶ Choose "Libraries" from the Task cards on the right side.
4. ▶ Click at "Global libraries".
5. ▶ Click at "Open global library".

6. ▶ Navigate to your directory and load the file VIPA\_TIA.al11.



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.

## 9.6 TIA Portal - Project transfer

### Overview

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

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

### Transfer via MPI

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

1. ▶ Establish a connection to the CPU via MPI with an appropriate programming cable. Information may be found in the corresponding documentation of the programming cable.
2. ▶ Switch-ON the power supply of your CPU and start the Siemens TIA Portal with your project.
3. ▶ Select in the *Project tree* your CPU and choose *Context menu* → *Download to device* → *Hardware configuration* to transfer the hardware configuration.
4. ▶ To transfer the PLC program choose *Context menu* → *Download to device* → *Software*. Due to the system you have to transfer hardware configuration and PLC program separately.

### Transfer via Ethernet

For transfer via Ethernet the CPU has the following interface:

- X1 Ethernet PG/OP channel

### Initialization

So that you may the according Ethernet interface, you have to assign IP address parameters by means of the "initialization". ↪ *Chapter 9.3 'TIA Portal - Hardware configuration - Ethernet PG/OP channel' on page 168*



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.

### 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. ▶ Create in the Siemens TIA Portal 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".
2. ▶ 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".

### Transfer memory card → CPU

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

- *S7PROG.WLD* is read from the memory card after overall reset.
- *AUTOLOAD.WLD* is read from the memory card after PowerON.

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

**Transfer CPU →  
Memory card**

When a memory card has been installed, the write command stores the content of the RAM as S7PROG.WLD on the memory card. The write command can be found in the Siemens TIA Portal in the Task card "Online tools" in the command area at "Memory" as button [Copy RAM to ROM]. The SD LED blinks during the write access. When the LED expires, the write process is finished. If this project is to be loaded automatically from the memory card with PowerON, you have to rename this to on the memory card to *AUTOLOAD.WLD*.

**Checking the transfer  
operation**

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries, you select *Online & Diagnostics* in the Siemens TIA Portal. Here you can access the "Diagnostics buffer". ↪ *Chapter 4.18 'VIPA specific diagnostic entries' on page 80*