<table>
<thead>
<tr>
<th>CPU</th>
<th>014-CEF0R00</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>HB300</td>
<td>CPU</td>
<td>014-CEF0R00</td>
</tr>
<tr>
<td>SPEED7 CPU 014</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Table of contents

1 General ............................................................................................................. 6  
1.1 Copyright © VIPA GmbH ........................................................................... 6  
1.2 About this manual .................................................................................... 7  
1.3 Safety information ............................................................................... 8  

2 Basics and mounting ...................................................................................... 10  
2.1 Safety information for users .................................................................... 10  
2.2 System conception .................................................................................. 11  
2.2.1 Overview .......................................................................................... 11  
2.2.2 Components ..................................................................................... 11  
2.2.3 Accessories ....................................................................................... 13  
2.3 Dimensions ............................................................................................. 14  
2.4 Mounting ................................................................................................... 16  
2.4.1 Mounting CPU 01x .......................................................................... 16  
2.5 Wiring ......................................................................................................... 18  
2.5.1 Wiring CPU 01x ................................................................................ 18  
2.5.2 Wiring periphery modules .................................................................. 21  
2.5.3 Wiring power modules ....................................................................... 23  
2.6 Demounting ............................................................................................... 27  
2.6.1 Demounting CPU 01x ....................................................................... 27  
2.6.2 Demounting periphery modules ....................................................... 28  
2.7 Trouble shooting - LEDs ......................................................................... 30  
2.8 Installation guidelines ............................................................................. 31  
2.9 General data .............................................................................................. 34  

3 Hardware description ..................................................................................... 36  
3.1 Properties .................................................................................................. 36  
3.2 Structure ..................................................................................................... 37  
3.2.1 Basic CPU ........................................................................................ 37  
3.2.2 Interfaces ........................................................................................... 37  
3.2.3 Memory management ......................................................................... 40  
3.2.4 Slot for storage media ....................................................................... 40  
3.2.5 Buffering mechanisms ...................................................................... 40  
3.2.6 Operating mode switch ...................................................................... 41  
3.2.7 LEDs .................................................................................................. 41  
3.3 Technical data ........................................................................................... 44  

4 Deployment CPU 014 .................................................................................. 51  
4.1 Assembly .................................................................................................. 51  
4.2 Start-up behavior ..................................................................................... 51  
4.3 Addressing ................................................................................................ 51  
4.3.1 Overview .......................................................................................... 51  
4.3.2 Addressing backplane bus periphery .............................................. 51  
4.4 Hardware configuration - CPU ................................................................. 53  
4.5 Hardware configuration - I/O modules ................................................... 55  
4.6 Hardware configuration - Ethernet PG/OP channel ................................ 56  
4.7 Hardware configuration - Communication ........................................... 58  
4.8 Setting standard CPU parameters .......................................................... 58  
4.8.1 Parametrization via Siemens CPU ..................................................... 58  
4.8.2 Parameter CPU ................................................................................ 59
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
EMail: 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.
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.

About this manual

This manual describes the CPU 014 of the System SLIO from VIPA. It contains a description of the construction, project implementation and usage.

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.

<table>
<thead>
<tr>
<th>Product</th>
<th>Order no.</th>
<th>as of state:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HW</td>
</tr>
<tr>
<td>Basic CPU 014</td>
<td>014-CEF0R00</td>
<td>01</td>
</tr>
</tbody>
</table>
Guide to the document

The following guides are available in the manual:
- An overall table of contents at the beginning of the manual
- References with page numbers

Availability

The manual is available in:
- printed form, on paper
- in electronic form as PDF-file (Adobe Acrobat Reader)

Icons Headings

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

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

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

Supplementary information and useful tips.

1.3 Safety information

Applications conforming with specifications

The system is constructed and produced for:
- communication and process control
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle

DANGER!
This device is not certified for applications in
- in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the
- project design department
- installation department
- commissioning
- operation
CAUTION!
The following conditions must be met before using or commissioning the components described in this manual:

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

Disposal

National rules and regulations apply to the disposal of the unit!
2 Basics and mounting

2.1 Safety information for users

Handling of electrostatic sensitive modules

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

![Symbol]

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

2.2.1 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 colors within the system, further isolated areas may be defined for the DC 24V power section supply, respectively the electronic power supply may be extended with 2A.

2.2.2 Components

- CPU (head module)
- Bus coupler (head module)
- Line extension
- Periphery modules
- Accessories

CAUTION!
Only modules of VIPA may be combined. A mixed operation with third-party modules is not allowed!
**CPU 01x**

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

**CAUTION!**

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

**Bus coupler**

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

**CAUTION!**

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

**Line extension**

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

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

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 01x
Dimensions

Dimensions bus coupler and line extension slave

Dimensions line extension master

Dimension periphery module
2.4 Mounting

2.4.1 Mounting CPU 01x

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

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

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

2.5 Wiring

2.5.1 Wiring CPU 01x

Terminal module terminals

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

Data

\[
\begin{align*}
U_{\text{max}} & \quad 240\text{V AC} / 30\text{V DC} \\
I_{\text{max}} & \quad 10\text{A} \\
\text{Cross section} & \quad 0.08 \ldots 1.5\text{mm}^2 \ (\text{AWG} \ 28 \ldots 16) \\
\text{Stripping length} & \quad 10\text{mm}
\end{align*}
\]
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 connect wires with a cross section of 0.08mm² up to 1.5mm².

3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

Standard wiring

(1) DC 24V for power section supply I/O area (max. 10A)
(2) DC 24V for electronic power supply bus coupler and I/O area
For wires with a core cross-section of 0.08mm² up to 1.5mm².

<table>
<thead>
<tr>
<th>Pos.</th>
<th>Function</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>---</td>
<td>---</td>
<td>not connected</td>
</tr>
<tr>
<td>2</td>
<td>DC 24V</td>
<td>I</td>
<td>DC 24V for power section supply</td>
</tr>
<tr>
<td>3</td>
<td>0V</td>
<td>I</td>
<td>GND for power section supply</td>
</tr>
<tr>
<td>4</td>
<td>Sys DC 24V</td>
<td>I</td>
<td>DC 24V for electronic section supply</td>
</tr>
<tr>
<td>5</td>
<td>---</td>
<td>---</td>
<td>not connected</td>
</tr>
<tr>
<td>6</td>
<td>DC 24V</td>
<td>I</td>
<td>DC 24V for power section supply</td>
</tr>
<tr>
<td>7</td>
<td>0V</td>
<td>I</td>
<td>GND for power section supply</td>
</tr>
<tr>
<td>8</td>
<td>Sys 0V</td>
<td>I</td>
<td>GND for electronic section supply</td>
</tr>
</tbody>
</table>

I: Input

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

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

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

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

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

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

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

2. Put your shield bus into the shield bus carrier.

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

2.5.2 Wiring periphery modules

Terminal module terminals

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

Data

- $U_{\text{max}}$: 240V AC / 30V DC
- $I_{\text{max}}$: 10A
- Cross section: 0.08 ... 1.5mm$^2$ (AWG 28 ... 16)
- Stripping length: 10mm

Wiring procedure

1. Pin number at the connector
2. Opening for screwdriver
3. Connection hole for wire
1. Insert a suited screwdriver at an angel into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.

2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm$^2$ up to 1.5mm$^2$

3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

Shield attachment

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

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

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

2. Put your shield bus into the shield bus carrier.

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

Terminal module terminals

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

Data

- \( U_{\text{max}} \) 240V AC / 30V DC
- \( I_{\text{max}} \) 10A
- Cross section 0.08 ... 1.5mm\(^2\) (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\(^2\) up to 1.5mm\(^2\)
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.
(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

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

<table>
<thead>
<tr>
<th>Pos.</th>
<th>Function</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>---</td>
<td>---</td>
<td>not connected</td>
</tr>
<tr>
<td>2</td>
<td>DC 24V</td>
<td>I</td>
<td>DC 24V for power section supply</td>
</tr>
<tr>
<td>3</td>
<td>0V</td>
<td>I</td>
<td>GND for power section supply</td>
</tr>
<tr>
<td>4</td>
<td>Sys DC 24V</td>
<td>I</td>
<td>DC 24V for electronic section supply</td>
</tr>
<tr>
<td>5</td>
<td>---</td>
<td>---</td>
<td>not connected</td>
</tr>
<tr>
<td>6</td>
<td>DC 24V</td>
<td>I</td>
<td>DC 24V for power section supply</td>
</tr>
<tr>
<td>7</td>
<td>0V</td>
<td>I</td>
<td>GND for power section supply</td>
</tr>
<tr>
<td>8</td>
<td>Sys 0V</td>
<td>I</td>
<td>GND for electronic section supply</td>
</tr>
</tbody>
</table>

I: Input

---

**CAUTION!**

Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!

Fusing

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

State of the electronic power supply via LEDs

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

Deployment of the power modules

- If the 10A for the power section supply is no longer sufficient, you may use the power module 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 max. 4A.
- By placing the power module 007-1AB10 at the following backplane bus modules may be placed with a sum current of max. 2A. Afterwards a power module is to be placed again. To secure the power supply, the power modules may be mixed used.

Power module 007-1AB00

![Diagram of Power Module 007-1AB00](image-url)
Power module
007-1AB10

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

Shield attachment

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

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

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

2. Put your shield bus into the shield bus carrier.

3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.
2.6 Demounting
2.6.1 Demounting CPU 01x

Proceeding

2.6 Demounting
2.6.1 Demounting CPU 01x

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

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

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

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

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

7. To mount the CPU put it to the left periphery module and push it, guided by the stripes, to the mounting rail.
8. Turn all the locking lever downward, again.
9. Plug again the electronic module, which you have removed before. For installation plug the electronic module guided by the strips at the lower side until this engages to the terminal module.

10. Wire your CPU.

⇒ Now you can bring your system back into operation.

2.6.2 Demounting periphery modules

**Proceeding**

**Exchange of an electronic module**

1. Power-off your system.

2. For the exchange of a electronic module, the electronic module may be pulled forward after pressing the unlocking lever at the lower side of the module.

3. For installation plug the new electronic module guided by the strips at the lower side until this engages to the terminal module.

⇒ Now you can bring your system back into operation.

**Exchange of a periphery module**

1. Power-off your system.

2. Remove if exists the wiring of the module.

3. For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.

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

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

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

9. Plug again the electronic module, which you have removed before.
10. Wire your module.
    ⇒ Now you can bring your system back into operation.

Exchange of a module group

1. Power-off your system.
2. Remove if exists the wiring of the module group.
3. For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.
   - Press the unlocking lever at the lower side of the just mounted right module near the module group and pull it forward.
4. Turn all the locking lever of the module group to be exchanged upwards.
5. Pull the module group forward.
6. For mounting turn all the locking lever of the module group to be mounted upwards.

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

9. Plug again the electronic module, which you have removed before.
10. Wire your module group.
    ⇒ Now you can bring your system back into operation.

2.7 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 2.5.3 ‘Wiring power modules’ on page 22

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.

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

Remedy: Match configuration and hardware structure.

Reason: 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.

Remedy: 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 compatibility (EMC), and how you manage the isolation.

What does EMC mean?

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

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

Possible interference causes

Electromagnetic interferences may interfere your control via different ways:

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

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

There are:

- galvanic coupling
- capacitive coupling
- inductive coupling
- radiant coupling
Basic rules for EMC

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

- Take care of a correct area-wide grounding of the inactive metal parts when installing your components.
  - Install a central connection between the ground and the protected earth conductor system.
  - Connect all inactive metal extensive and impedance-low.
  - Please try not to use aluminium parts. Aluminium is easily oxidizing and is therefore less suitable for grounding.

- When cabling, take care of the correct line routing.
  - Organize your cabling in line groups (high voltage, current supply, signal and data lines).
  - Always lay your high voltage lines and signal respectively data lines in separate channels or bundles.
  - Route the signal and data lines as near as possible beside ground areas (e.g. suspension bars, metal rails, tin cabinet).

- Proof the correct fixing of the lead isolation.
  - Data lines must be laid isolated.
  - Analog lines must be laid isolated. When transmitting signals with small amplitudes the one sided laying of the isolation may be favourable.
  - Lay the line isolation extensively on an isolation/protected earth conductor rail directly after the cabinet entry and fix the isolation with cable clamps.
  - Make sure that the isolation/protected earth conductor rail is connected impedance-low with the cabinet.
  - Use metallic or metallised plug cases for isolated data lines.

- In special use cases you should appoint special EMC actions.
  - Consider to wire all inductivities with erase links.
  - Please consider luminescent lamps can influence signal lines.

- Create a homogeneous reference potential and ground all electrical operating supplies when possible.
  - Please take care for the targeted employment of the grounding actions. The grounding of the PLC serves for protection and functionality activity.
  - Connect installation parts and cabinets with your PLC in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If there are potential differences between installation parts and cabinets, lay sufficiently dimensioned potential compensation lines.

Isolation of conductors

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

When isolating cables you have to regard the following:

- If possible, use only cables with isolation tangle.
- The hiding power of the isolation should be higher than 80%.
Normally you should always lay the isolation of cables on both sides. Only by means of the both-sided connection of the isolation you achieve high quality interference suppression in the higher frequency area. Only as exception you may also lay the isolation one-sided. Then you only achieve the absorption of the lower frequencies. A one-sided isolation connection may be convenient, if:
- the conduction of a potential compensating line is not possible.
- analog signals (some mV respectively µA) are transferred.
- foil isolations (static isolations) are used.

With data lines always use metallic or metallised plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!

At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.

To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.

Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to your PLC and don't lay it on there again!

**CAUTION!**
**Please regard at installation!**
At potential differences between the grounding points, there may be a compensation current via the isolation connected at both sides.
Remedy: Potential compensation line
### 2.9 General data

#### Conformity and approval

<table>
<thead>
<tr>
<th>Conformity</th>
<th>Approval</th>
<th>RoHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE 2006/95/EG</td>
<td>Low-voltage directive</td>
<td>2011/65/EU</td>
</tr>
<tr>
<td>2004/108/EG</td>
<td>EMC directive</td>
<td></td>
</tr>
<tr>
<td>UL</td>
<td>Refer to Technical Data</td>
<td></td>
</tr>
<tr>
<td>others</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Protection of persons and device protection

| Type of protection | - | IP20 |
| Electrical isolation | - | electrically isolated |
| to the field bus | - | electrically isolated |
| to the process level | - | |
| 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 | Operation | Mechanical |
| Storage / transport | EN 60068-2-14 | -25…+70°C |
| Horizontal installation hanging | EN 61131-2 | 0…+60°C |
| Horizontal installation lying | EN 61131-2 | 0…+55°C |
| Vertical installation | EN 61131-2 | 0…+50°C |
| Air humidity | EN 60068-2-30 | RH1 (without condensation, rel. humidity 10…95%) |
| Pollution | EN 61131-2 | Degree of pollution 2 |
| Installation altitude max. | - | 2000m |
| Oscillation | EN 60068-2-6 | 1g, 9Hz ... 150Hz |
| Shock | EN 60068-2-27 | 15g, 11ms |
## Mounting conditions

<table>
<thead>
<tr>
<th>Mounting place</th>
<th>-</th>
<th>In the control cabinet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mounting position</td>
<td>-</td>
<td>Horizontal and vertical</td>
</tr>
</tbody>
</table>

## EMC

<table>
<thead>
<tr>
<th>Emitted interference</th>
<th>Standard</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A (Industrial area)</td>
<td>EN 61000-6-4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Noise immunity</th>
<th>Standard</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN 61000-6-2</td>
<td>EN 61000-4-2</td>
<td>Industrial area</td>
</tr>
<tr>
<td>EN 61000-4-3</td>
<td>EN 61000-4-6</td>
<td>ESD</td>
</tr>
<tr>
<td>EN 61000-4-4</td>
<td>EN 61000-4-5</td>
<td>HF field immunity (casing)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HF conducted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Burst, degree of severity 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Surge, installation class 3</td>
</tr>
</tbody>
</table>

*) 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 014

- SPEED7 technology integrated
- Programmable via SPEED7 Studio, Siemens SIMATIC Manager or TIA Portal
- Work memory 64kbyte integrated (32kbyte code, 32kbyte data)
- Work memory expandable up to 192kbyte (96kbyte code, 96kbyte data)
- 192kbyte load memory integrated
- Slot for external storage media (lockable)
- Status LEDs for operating state and diagnostics
- X1: Ethernet PG/OP channel integrated
- X2: PtP(MPI) interface: Serial integrated interface for PtP communication with the protocols: ASCII, STX/ETX, USS, 3964(R), MODBUS RTU, master/slave switch able to MPI communication
- X3: MPI(PB) interface: MPI interface with via VSC unlock able field bus functions
- up to 64 SLIO modules placeable
- I/O address area digital/analog 2048byte
- 512 timer/counter, 8192 flag byte

Ordering data

<table>
<thead>
<tr>
<th>Type</th>
<th>Order number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU 014</td>
<td>014-CEF0R00</td>
<td>Basic CPU 014 with options to extend work memory and bus interface.</td>
</tr>
</tbody>
</table>
3.2  Structure

3.2.1  Basic CPU

CPU 014

1. Locking lever
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. X1: Ethernet PG/OP channel
11. X2: PtP(MPI) interface
12. X3: MPI(PB) interface
13. Operating mode switch CPU
14. CPU part
15. 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 0.08\(\text{mm}^2\) up to 1.5\(\text{mm}^2\).

<table>
<thead>
<tr>
<th>Pos.</th>
<th>Function</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>---</td>
<td>---</td>
<td>not connected</td>
</tr>
<tr>
<td>2</td>
<td>DC 24V</td>
<td>I</td>
<td>DC 24V for power section supply</td>
</tr>
<tr>
<td>3</td>
<td>0V</td>
<td>I</td>
<td>GND for power section supply</td>
</tr>
<tr>
<td>4</td>
<td>Sys DC 24V</td>
<td>I</td>
<td>DC 24V for electronic section supply</td>
</tr>
<tr>
<td>5</td>
<td>---</td>
<td>---</td>
<td>not connected</td>
</tr>
<tr>
<td>6</td>
<td>DC 24V</td>
<td>I</td>
<td>DC 24V for power section supply</td>
</tr>
<tr>
<td>7</td>
<td>0V</td>
<td>I</td>
<td>GND for power section supply</td>
</tr>
<tr>
<td>8</td>
<td>Sys 0V</td>
<td>I</td>
<td>GND for electronic section supply</td>
</tr>
</tbody>
</table>

I: Input

---

**X1: Ethernet PG/OP channel**

### 8pin RJ45 jack:

- The RJ45 jack serves as interface to the Ethernet PG/OP channel.
- This interface allows you to program respectively remote control your CPU and to access the internal web server.
- Configurable connections are 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 56*
**X2: PtP(MPI) interface**

9pin SubD jack: (isolated):

The interface supports the following functions, which are switchable via the **VIPA specific CPU parameters** in Chapter 4.9 ‘Setting VIPA specific CPU parameters’ on page 63:

- PtP (default / after overall reset)
  
  Per default, the RS485 interface is set to PtP functionality. Using the PtP functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source res. target systems.
  
  The following protocols are supported:
  - ASCII
  - STX/ETX
  - 3964R
  - USS
  - Modbus master (ASCII, RTU)

- MPI
  
  The MPI interface serves for the connection between programming unit and CPU. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU. Standard setting is MPI address 2.

**X3: MPI(PB) interface**

9pin SubD jack: (isolated):

The interface supports the following functions, which are switchable via the sub module X1 ‘MPI/DP’ in the hardware configuration:

- MPI (default / after reset to factory setting in Chapter 4.15 ‘Factory reset’ on page 77)
  
  Per default, the RS485 interface is set to MPI functionality. The MPI interface serves for the connection between programming unit and CPU. By means of this the project engineering and programming happens. In addition MPI serves for communication between several CPUs or between HMIs and CPU. Standard setting is MPI address 2.

- PB
  
  The PROFIBUS master/slave functionality of this interface can be activated by configuring the 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 78
3.2.3 Memory management

General

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

- Load memory 192kbyte
- Code memory (50% of the work memory)
- Data memory (50% of the work memory)
- Work memory 64kbyte
  - There is the possibility to extend the work memory to its maximum capacity 192kbyte 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 4.16 ‘Deployment storage media - VSD, VSC’ on page 78
  - To activate the corresponding card is to be installed and a Overall reset is to be established. ☞ Chapter 4.13 ‘Overall reset’ on page 74

A list of the currently available VSD respectively VSC can be found at 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. ☞ ‘VIPA specific diagnostic entries’ on page 83
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 (Memory Reset), you request an overall reset with following load from memory card, if a project there exists.

3.2.7 LEDs

**CPU part**

<table>
<thead>
<tr>
<th>PW</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>● As soon as the CPU is supplied with 5V, the green PW-LED (Power) is on.</td>
</tr>
<tr>
<td></td>
<td>○ The CPU is not power-supplied.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RN</th>
<th>ST</th>
<th>SF</th>
<th>FC</th>
<th>SD</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>yellow</td>
<td>red</td>
<td>yellow</td>
<td>yellow</td>
<td>Boot-up after PowerON</td>
</tr>
<tr>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Flickers: Firmware is loaded.</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Initialization: Phase 1</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>Initialization: Phase 2</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>Initialization: Phase 3</td>
</tr>
<tr>
<td>○</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>Initialization: Phase 4</td>
</tr>
</tbody>
</table>

**Operation**

- ○ ● X X X CPU is in STOP state.
- 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 ● Accessing the memory card
- X BB X X X Blinking with 10Hz: Configuration is loaded
<table>
<thead>
<tr>
<th>Meanings</th>
<th>RN</th>
<th>ST</th>
<th>SF</th>
<th>FC</th>
<th>SD</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall reset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>○ BB X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Blinking with 1Hz: Overall reset is requested</td>
</tr>
<tr>
<td>○ BB X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Blinking with 2Hz: Overall reset is executed</td>
</tr>
<tr>
<td>○ BB X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Blinking with 10Hz: Overall reset with none hardware configuration respectively with hardware configuration from memory card.</td>
</tr>
<tr>
<td>Reset to factory setting</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>Reset to factory setting is executed</td>
</tr>
<tr>
<td>○ ● ● ● ●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reset to factory setting finished without error. Then a power cycle is necessary</td>
</tr>
<tr>
<td>Firmware update</td>
<td>○</td>
<td>●</td>
<td>BB</td>
<td>BB</td>
<td>●</td>
<td></td>
</tr>
<tr>
<td>○ ○ BB BB ●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The alternate blinking indicates that there is new firmware on the memory card.</td>
</tr>
<tr>
<td>○ ○ BB BB ●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The alternate blinking indicates that a firmware update is executed.</td>
</tr>
<tr>
<td>○ ● ● ● ●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Firmware update finished without error.</td>
</tr>
<tr>
<td>○ BB BB BB ●</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Blinking with 10Hz: Error during Firmware update.</td>
</tr>
</tbody>
</table>

Ethernet PG/OP channel:

<table>
<thead>
<tr>
<th>Meaning</th>
<th>L/A (Link/Activity)</th>
<th>S (Speed)</th>
<th>on: ●</th>
<th>off: ○</th>
<th>blinking: BB</th>
<th>not relevant: X</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>green</td>
<td>green</td>
<td>●</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>X</td>
<td>The Ethernet PG/OP channel is physically connected to the Ethernet interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>○</td>
<td>X</td>
<td>There is no physical connection.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB</td>
<td>X</td>
<td>Shows Ethernet activity.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 100Mbit.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>●</td>
<td>○</td>
<td>The Ethernet interface of the Ethernet PG/OP channel has a transfer rate of 10Mbit.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### LEDs power module

<table>
<thead>
<tr>
<th>PWR</th>
<th>PWR</th>
<th>PF</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>green</td>
<td>red</td>
<td>Both power supplies are missing</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>Power section supply OK</td>
</tr>
<tr>
<td>●</td>
<td>X</td>
<td>○</td>
<td>Electronic section supply OK</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>●</td>
<td>Fuse electronic section supply defective</td>
</tr>
</tbody>
</table>

On: ● | Off: ○ | Not relevant: X

---

### 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

<table>
<thead>
<tr>
<th>DE (Data Exchange)</th>
<th>BF (Bus error)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>red</td>
<td></td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>Master has no project, this means the interface is deactivated respectively the master configured without slaves with no errors.</td>
</tr>
<tr>
<td>BB</td>
<td>○</td>
<td>CPU is in STOP state, the master is in &quot;clear&quot; state. All the slaves are in DE and the outputs are of the slaves are disabled.</td>
</tr>
<tr>
<td>●</td>
<td>○</td>
<td>CPU is in STOP state, the master is in &quot;operate&quot; state. All the slaves are in DE. The outputs are enabled.</td>
</tr>
<tr>
<td>●</td>
<td>BB</td>
<td>CPU is in RUN state, at least 1 slave is missing and at least 1 slave is in DE.</td>
</tr>
<tr>
<td>BB</td>
<td>BB</td>
<td>CPU is in STOP state, the master is in &quot;clear&quot; state. At least 1 slave is missing and at least 1 slave is in DE.</td>
</tr>
<tr>
<td>○</td>
<td>●</td>
<td>PROFIBUS is interrupted (no communication possible)</td>
</tr>
<tr>
<td>○</td>
<td>BB</td>
<td>At least 1 slave is missing and no slave is in DE.</td>
</tr>
</tbody>
</table>
### Data Exchange (DE) and Bus Error (BF)

<table>
<thead>
<tr>
<th>DE (Data Exchange)</th>
<th>BF (Bus Error)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>BB</td>
<td>At least 1 slave is not in DE.</td>
</tr>
</tbody>
</table>

- on: ● | off: ○ | blinking (2Hz): BB

### Slave Operation

<table>
<thead>
<tr>
<th>DE (Data Exchange)</th>
<th>BF (Bus Error)</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>green</td>
<td>red</td>
<td>Slave has no project.</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>There is a bus error.</td>
</tr>
<tr>
<td>○</td>
<td>●</td>
<td>Slave is in state data exchange with master.</td>
</tr>
<tr>
<td>BB</td>
<td>○</td>
<td>Slave is in state data exchange with master. Slave CPU is in STOP state.</td>
</tr>
<tr>
<td>●</td>
<td>○</td>
<td>Slave is in state data exchange with master. Slave CPU is in RUN state.</td>
</tr>
</tbody>
</table>

- on: ● | off: ○ | blinking (2Hz): BB

### Technical Data

#### Order no.

- 014-CEF0R00

#### Type

- CPU 014

#### Module ID

- 

#### Technical Data: Power Supply

- Power supply (rated value): 24 V
- Power supply (permitted range): 20.4...28.8 V
- Reverse polarity protection: ✔
- Current consumption (no-load operation): 120 mA
- Current consumption (rated value): 1 A
- Inrush current: 3 A
- \( I^2t \): 0.1 A²s
- Max. current drain at backplane bus: 3 A
- Max. current drain load supply: 10 A
- Power loss: 6 W

#### Load and Working Memory

- VIPA System SLIO
- Hardware description

---

HB300 | CPU | 014-CEF0R00 | GB | 16-03

43
<table>
<thead>
<tr>
<th><strong>Order no.</strong></th>
<th><strong>014-CEF0R00</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Load memory, integrated</td>
<td>192 KB</td>
</tr>
<tr>
<td>Load memory, maximum</td>
<td>192 KB</td>
</tr>
<tr>
<td>Work memory, integrated</td>
<td>64 KB</td>
</tr>
<tr>
<td>Work memory, maximal</td>
<td>192 KB</td>
</tr>
<tr>
<td>Memory divided in 50% program / 50% data</td>
<td>✓</td>
</tr>
<tr>
<td>Memory card slot</td>
<td>SD/MMC-Card with max. 2 GB</td>
</tr>
</tbody>
</table>

**Hardware configuration**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Racks, max.</td>
<td>1</td>
</tr>
<tr>
<td>Modules per rack, max.</td>
<td>64</td>
</tr>
<tr>
<td>Number of integrated DP master</td>
<td>1</td>
</tr>
<tr>
<td>Number of DP master via CP</td>
<td>-</td>
</tr>
<tr>
<td>Operable function modules</td>
<td>64</td>
</tr>
<tr>
<td>Operable communication modules PtP</td>
<td>64</td>
</tr>
<tr>
<td>Operable communication modules LAN</td>
<td>-</td>
</tr>
</tbody>
</table>

**Command processing times**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit instructions, min.</td>
<td>0.02 µs</td>
</tr>
<tr>
<td>Word instruction, min.</td>
<td>0.02 µs</td>
</tr>
<tr>
<td>Double integer arithmetic, min.</td>
<td>0.02 µs</td>
</tr>
<tr>
<td>Floating-point arithmetic, min.</td>
<td>0.12 µs</td>
</tr>
</tbody>
</table>

**Timers/Counters and their retentive characteristics**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of S7 counters</td>
<td>512</td>
</tr>
<tr>
<td>S7 counter remanence</td>
<td>adjustable 0 up to 512</td>
</tr>
<tr>
<td>S7 counter remanence adjustable</td>
<td>C0 .. C7</td>
</tr>
<tr>
<td>Number of S7 times</td>
<td>512</td>
</tr>
<tr>
<td>S7 times remanence</td>
<td>adjustable 0 up to 512</td>
</tr>
<tr>
<td>S7 times remanence adjustable</td>
<td>not retentive</td>
</tr>
</tbody>
</table>

**Data range and retentive characteristic**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of flags</td>
<td>8192 Byte</td>
</tr>
<tr>
<td>Bit memories retentive characteristic adjustable</td>
<td>adjustable 0 up to 8192</td>
</tr>
<tr>
<td>Bit memories retentive characteristic preset</td>
<td>MB0 .. MB15</td>
</tr>
<tr>
<td>Number of data blocks</td>
<td>1024</td>
</tr>
<tr>
<td>Max. data blocks size</td>
<td>64 KB</td>
</tr>
<tr>
<td>Number range DBs</td>
<td>1 ... 8191</td>
</tr>
<tr>
<td>Max. local data size per execution level</td>
<td>4096 Byte</td>
</tr>
<tr>
<td>Max. local data size per block</td>
<td>4096 Byte</td>
</tr>
<tr>
<td>Order no.</td>
<td>014-CEF0R00</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Blocks</strong></td>
<td></td>
</tr>
<tr>
<td>Number of OBs</td>
<td>22</td>
</tr>
<tr>
<td>Maximum OB size</td>
<td>64 KB</td>
</tr>
<tr>
<td>Total number DBs, FBs, FCs</td>
<td>1024</td>
</tr>
<tr>
<td>Number of FBs</td>
<td>1024</td>
</tr>
<tr>
<td>Maximum FB size</td>
<td>64 KB</td>
</tr>
<tr>
<td>Number range FBs</td>
<td>0 ... 8191</td>
</tr>
<tr>
<td>Number of FCs</td>
<td>1024</td>
</tr>
<tr>
<td>Maximum FC size</td>
<td>64 KB</td>
</tr>
<tr>
<td>Number range FCs</td>
<td>0 ... 8191</td>
</tr>
<tr>
<td>Maximum nesting depth per priority class</td>
<td>16</td>
</tr>
<tr>
<td>Maximum nesting depth additional within an error OB</td>
<td>4</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td></td>
</tr>
<tr>
<td>Real-time clock buffered</td>
<td>✓</td>
</tr>
<tr>
<td>Clock buffered period (min.)</td>
<td>30 d</td>
</tr>
<tr>
<td>Type of buffering</td>
<td>Goldcap</td>
</tr>
<tr>
<td>Load time for 50% buffering period</td>
<td>15 min</td>
</tr>
<tr>
<td>Load time for 100% buffering period</td>
<td>1 h</td>
</tr>
<tr>
<td>Accuracy (max. deviation per day)</td>
<td>10 s</td>
</tr>
<tr>
<td>Number of operating hours counter</td>
<td>8</td>
</tr>
<tr>
<td>Clock synchronization</td>
<td>✓</td>
</tr>
<tr>
<td>Synchronization via MPI</td>
<td>Master/Slave</td>
</tr>
<tr>
<td>Synchronization via Ethernet (NTP)</td>
<td>no</td>
</tr>
<tr>
<td><strong>Address areas (I/O)</strong></td>
<td></td>
</tr>
<tr>
<td>Input I/O address area</td>
<td>2048 Byte</td>
</tr>
<tr>
<td>Output I/O address area</td>
<td>2048 Byte</td>
</tr>
<tr>
<td>Process image adjustable</td>
<td>✓</td>
</tr>
<tr>
<td>Input process image preset</td>
<td>128 Byte</td>
</tr>
<tr>
<td>Output process image preset</td>
<td>128 Byte</td>
</tr>
<tr>
<td>Input process image maximal</td>
<td>2048 Byte</td>
</tr>
<tr>
<td>Output process image maximal</td>
<td>2048 Byte</td>
</tr>
<tr>
<td>Digital inputs</td>
<td>16384</td>
</tr>
<tr>
<td>Digital outputs</td>
<td>16384</td>
</tr>
<tr>
<td>Digital inputs central</td>
<td>512</td>
</tr>
<tr>
<td>Digital outputs central</td>
<td>512</td>
</tr>
</tbody>
</table>
### Hardware description

VIPA System SLIO

#### Technical data

<table>
<thead>
<tr>
<th>Order no.</th>
<th>014-CEF0R00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated digital inputs</td>
<td>-</td>
</tr>
<tr>
<td>Integrated digital outputs</td>
<td>-</td>
</tr>
<tr>
<td>Analog inputs</td>
<td>1024</td>
</tr>
<tr>
<td>Analog outputs</td>
<td>1024</td>
</tr>
<tr>
<td>Analog inputs, central</td>
<td>256</td>
</tr>
<tr>
<td>Analog outputs, central</td>
<td>256</td>
</tr>
<tr>
<td>Integrated analog inputs</td>
<td>-</td>
</tr>
<tr>
<td>Integrated analog outputs</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Communication functions

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PG/OP channel</td>
<td>✓</td>
</tr>
<tr>
<td>Global data communication</td>
<td>✓</td>
</tr>
<tr>
<td>Number of GD circuits, max.</td>
<td>8</td>
</tr>
<tr>
<td>Size of GD packets, max.</td>
<td>22 Byte</td>
</tr>
<tr>
<td>S7 basic communication</td>
<td>✓</td>
</tr>
<tr>
<td>S7 basic communication, user data per job</td>
<td>76 Byte</td>
</tr>
<tr>
<td>S7 communication</td>
<td>✓</td>
</tr>
<tr>
<td>S7 communication as server</td>
<td>✓</td>
</tr>
<tr>
<td>S7 communication as client</td>
<td>-</td>
</tr>
<tr>
<td>S7 communication, user data per job</td>
<td>160 Byte</td>
</tr>
<tr>
<td>Number of connections, max.</td>
<td>32</td>
</tr>
</tbody>
</table>

#### Functionality Sub-D interfaces

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>X2</td>
</tr>
<tr>
<td>Type of interface</td>
<td>RS485</td>
</tr>
<tr>
<td>Connector</td>
<td>Sub-D, 9-pin, female</td>
</tr>
<tr>
<td>Electrically isolated</td>
<td>✓</td>
</tr>
<tr>
<td>MPI</td>
<td>✓</td>
</tr>
<tr>
<td>MP²I (MPI/RS232)</td>
<td>-</td>
</tr>
<tr>
<td>DP master</td>
<td>-</td>
</tr>
<tr>
<td>DP slave</td>
<td>-</td>
</tr>
<tr>
<td>Point-to-point interface</td>
<td>✓</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>X3</td>
</tr>
<tr>
<td>Type of interface</td>
<td>RS485</td>
</tr>
<tr>
<td>Connector</td>
<td>Sub-D, 9-pin, female</td>
</tr>
<tr>
<td>Electrically isolated</td>
<td>✓</td>
</tr>
<tr>
<td>MPI</td>
<td>✓</td>
</tr>
</tbody>
</table>
### Technical data

<table>
<thead>
<tr>
<th><strong>Order no.</strong></th>
<th><strong>014-CEF0R00</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MP²I (MPI/RS232)</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>DP master</strong></td>
<td>optional</td>
</tr>
<tr>
<td><strong>DP slave</strong></td>
<td>optional</td>
</tr>
<tr>
<td><strong>Point-to-point interface</strong></td>
<td>-</td>
</tr>
</tbody>
</table>

**Functionality MPI**

- Number of connections, max.: 32
- PG/OP channel: ✓
- Routing: ✓
- Global data communication: ✓
- S7 basic communication: ✓
- S7 communication: ✓
- S7 communication as server: ✓
- S7 communication as client: -
- Transmission speed, min.: 19.2 kbit/s
- Transmission speed, max.: 12 Mbit/s

**Functionality PROFIBUS master**

- PG/OP channel: ✓
- Routing: ✓
- S7 basic communication: ✓
- S7 communication: ✓
- S7 communication as server: ✓
- S7 communication as client: -
- Activation/deactivation of DP slaves: -
- Direct data exchange (slave-to-slave communication): -
- DPV1: ✓
- Transmission speed, min.: 9.6 kbit/s
- Transmission speed, max.: 12 Mbit/s
- Number of DP slaves, max.: 124
- Address range inputs, max.: 2 KB
- Address range outputs, max.: 2 KB
- User data inputs per slave, max.: 244 Byte
- User data outputs per slave, max.: 244 Byte

**Functionality PROFIBUS slave**

- PG/OP channel: ✓
- Routing: ✓
<table>
<thead>
<tr>
<th><strong>Order no.</strong></th>
<th>014-CEF0R00</th>
</tr>
</thead>
<tbody>
<tr>
<td>S7 communication</td>
<td>✓</td>
</tr>
<tr>
<td>S7 communication as server</td>
<td>✓</td>
</tr>
<tr>
<td>S7 communication as client</td>
<td>-</td>
</tr>
<tr>
<td>Direct data exchange (slave-to-slave communication)</td>
<td>-</td>
</tr>
<tr>
<td>DPV1</td>
<td>✓</td>
</tr>
<tr>
<td>Transmission speed, min.</td>
<td>9.6 kbit/s</td>
</tr>
<tr>
<td>Transmission speed, max.</td>
<td>12 Mbit/s</td>
</tr>
<tr>
<td>Automatic detection of transmission speed</td>
<td>-</td>
</tr>
<tr>
<td>Transfer memory inputs, max.</td>
<td>244 Byte</td>
</tr>
<tr>
<td>Transfer memory outputs, max.</td>
<td>244 Byte</td>
</tr>
<tr>
<td>Address areas, max.</td>
<td>32</td>
</tr>
<tr>
<td>User data per address area, max.</td>
<td>32 Byte</td>
</tr>
</tbody>
</table>

**Point-to-point communication**

| PtP communication | ✓ |
| Interface isolated | ✓ |
| RS232 interface | - |
| RS422 interface | - |
| RS485 interface | ✓ |
| Connector | Sub-D, 9-pin, female |
| Transmission speed, min. | 150 bit/s |
| Transmission speed, max. | 115.5 kbit/s |
| Cable length, max. | 500 m |

**Point-to-point protocol**

| ASCII protocol | ✓ |
| STX/ETX protocol | ✓ |
| 3964(R) protocol | ✓ |
| RK512 protocol | - |
| USS master protocol | ✓ |
| Modbus master protocol | ✓ |
| Modbus slave protocol | ✓ |
| Special protocols | - |

**Functionality RJ45 interfaces**

| Type | X1 |
| Type of interface | Ethernet 10/100 MBit |
| Connector | RJ45 |
### Technical data

<table>
<thead>
<tr>
<th><strong>Order no.</strong></th>
<th>014-CEF0R00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrically isolated</td>
<td>✓</td>
</tr>
<tr>
<td>PG/OP channel</td>
<td>✓</td>
</tr>
<tr>
<td>Number of connections, max.</td>
<td>4</td>
</tr>
<tr>
<td>Productive connections</td>
<td>-</td>
</tr>
</tbody>
</table>

**Housing**
- Material: PPE / PPE GF10
- Mounting: Profile rail 35 mm

**Mechanical data**
- Dimensions (WxHxD): 131.5 mm x 109 mm x 83 mm
- Weight: 280 g

**Environmental conditions**
- Operating temperature: 0 °C to 60 °C
- Storage temperature: -25 °C to 70 °C

**Certifications**
- UL certification: in preparation
- KC certification: yes
4 Deployment CPU 014

4.1 Assembly

Information about assembly and cabling "Basics and mounting" on page 9

4.2 Start-up behavior

Turn on power supply

- The CPU checks whether a project AUTOLOAD.WLD exists on the memory card. If so, an overall reset is executed and the project is automatically loaded from the memory card.
- The CPU checks whether a command file with the name VIPA_CMD.MMC exists on the memory card. If so the command file is loaded from the memory card and the commands are executed.
- After PowerON and CPU STOP the CPU checks if there is a *.pkg file (firmware file) on the memory card. If so, this is shown by the CPU by blinking LEDs and the firmware may be installed by an update request. Further information on page 76
- 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. "VIPA specific diagnostic entries" on page 83

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

Delivery state

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

4.3 Addressing

4.3.1 Overview

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

4.3.2 Addressing backplane bus periphery

The CPU 014 provides an I/O area (address 0 ... 2047) and a process image of the in- and outputs (each address default 0 ... 127). The process image stores the signal states of the lower address (default 0 ... 127) in an additional memory area. The size of the process image can be preset via the parameterization. Cycle/Clock memory on page 60
The process image is 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 pluggable modules
Up to 64 SLIO modules can be connected to a 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**

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

**4.4 Hardware configuration - CPU**

**Precondition**
- The configuration of the CPU takes place at the Siemens ‘hardware configurator’. The hardware configurator is part of the Siemens SIMATIC Manager. It serves for project engineering.
- Please use for configuration the Siemens SIMATIC Manager V 5.5 SP2 and up.
- The configuration of the System 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.

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

**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:

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

1. Start the Siemens hardware configurator with a new project.
2. Insert a profile rail from the hardware catalog.
3. Place at ‘Slot’-Number 2 the CPU 315-2 PN/DP (315-2EH14 V3.2).
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.
9. Navigate in the hardware catalog to the directory ‘PROFINET IO → Additional field devices → I/O → VIPA SLIO System’ and connect the IO device ‘014-CEF0R00 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.
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 014 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 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 5.3 & SP3 with the following proceeding:

1. Start the Siemens SIMATIC Manager and set via ‘Options ➔ Set PG/PC interface the access path to ‘TCP/IP → Network card ....’.

2. Open with ‘PLC ➔ Edit Ethernet Node n’ the dialog window with the same name.

3. To get the stations and their MAC address, use the [Browse] button or type in the MAC Address. The Mac address may be found at the 1. label beneath the front flap of the CPU.

4. Choose if necessary the known MAC address of the list of found stations.

5. Either type in the IP configuration like IP address, subnet mask and gateway.

6. Confirm with [Assign IP configuration].

⇒ Direct after the assignment the Ethernet PG/OP channel may be reached online by these address data. The value remains as long as it is reassigned, it is overwritten by a hardware configuration or an factory reset is executed.

Take IP address parameters in project

1. Open the Siemens hardware configurator and configure the Siemens CPU 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 V1.2).

3. Open the property window via double-click on the CP 343-1EX21 and enter for the CP at ‘Properties’ the IP address data, which you have assigned before.

4. Assign the CP to a ‘Subnet’. Without assignment the IP address data are not used!
5. Transfer your project.

4.7 Hardware configuration - Communication

The hardware configuration of PROFIBUS and PtP is described at the following pages:

- PROFIBUS DP
  - Master operation: § Chapter 6.5 ‘Deployment as PROFIBUS DP master’ on page 125
  - Slave operation: § Chapter 6.6 ‘Deployment as PROFIBUS DP slave’ on page 127

- PtP
  - PtP: § Chapter 5.3 ‘Deployment of RS485 interface for PtP’ on page 101

4.8 Setting standard CPU parameters

4.8.1 Parametrization via Siemens CPU

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.

### Parametrization via Siemens CPU 315-2EH14

#### Parameter CPU

<table>
<thead>
<tr>
<th>Slot</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CPU...</td>
</tr>
<tr>
<td>X1</td>
<td>MPI/DP</td>
</tr>
<tr>
<td>X2</td>
<td>PN-IO</td>
</tr>
<tr>
<td>X2 P1</td>
<td>Port 1</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
### 4.8.2 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 315-2PN/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 of if another type of module is inserted there instead, such as during an initial system start-up.
- **Monitoring time for ready message by modules [100ms]**
  - This operation specifies the maximum time for the ready message of every configured module after PowerON.
  - Here connected PROFIBUS DP slaves are also considered until they are parameterized.
  - If the modules do not send a ready message to the CPU by the time the monitoring time has expired, the actual configuration becomes unequal to the preset configuration.
- **Monitoring time for transfer of parameters to modules [100ms]**
  - The maximum time for the transfer of parameters to parameterizable modules.
  - Here connected PROFINET IO devices also considered until they are parameterized.
  - If not every module has been assigned parameters by the time this monitoring time has expired; the actual configuration becomes unequal to the preset configuration.
Cycle/Clock memory

- Update OB1 process image cyclically
  - This parameter is not relevant.
- Scan cycle monitoring time
  - Here the scan cycle monitoring time in milliseconds may be set.
  - If the scan cycle time exceeds the scan cycle monitoring time, the CPU enters the STOP mode.
  - Possible reasons for exceeding the time are:
    - Communication processes
    - a series of interrupt events
    - an error in the CPU program
- Minimum scan cycle time
  - This parameter is not relevant.
- Scan cycle load from Communication
  - Using this parameter you can control the duration of communication processes, which always extend the scan cycle time so it does not exceed a specified length.
  - If the cycle load from communication is set to 50%, the scan cycle time of OB 1 can be doubled. At the same time, the scan cycle time of OB 1 is still being influenced by asynchronous events (e.g. hardware interrupts) as well.
- Size of the process image input/output area
  - Here the size of the process image max. 2048 for the input/output periphery may be fixed (default: 128).
- 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 2 bytes.
- Number of S7 Counters from C0
  - Enter the number of retentive S7 counter from C0 onwards.
- Areas
  - This parameter is not supported.

Interrupts

- Priority
  - Here the priorities are displayed, according to which the hardware interrupt OBs are processed (hardware interrupt, time-delay interrupt, async. error interrupts).
Time-of-day interrupts
- Priority
  - This value is fixed to 2.
- Active
  - By enabling ‘Active’ the time-of-day interrupt function is enabled.
- Execution
  - Select how often the interrupts are to be triggered.
  - Intervals ranging from every minute to yearly are available. The intervals apply to the settings made for start date and time.
- Start date/time
  - Enter date and time of the first execution of the time-of-day interrupt.
- Process image partition
  - This parameter is not supported.

Cyclic interrupts
- Priority
  - Here the priorities may be specified according to which the corresponding cyclic interrupt is processed.
  - 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 watchdog interrupt. This should be performed if several watchdog interrupts are enabled.
  - Phase offset allows to distribute processing time for watchdog interrupts across the cycle.
- Process image partition
  - This parameter is not supported.

Diagnostics/Clock
- Report cause of STOP
  - Activate this parameter, if the CPU should report the cause of STOP to PG respectively OP on transition to STOP.
- Number of messages in the diagnostics buffer
  - This parameter is ignored. The CPU always has a diagnostics buffer (circular buffer) for 100 diagnostics messages.
- Synchronization type
  - Here you specify whether clock should synchronize other clocks or not.
  - as slave: The clock is synchronized by another clock.
  - as master: The clock synchronizes other clocks as master.
  - none: There is no synchronization
- Time interval
  - Time intervals within which the synchronization is to be carried out.
- Correction factor
  - Lose or gain in the clock time may be compensated within a 24 hour period by means of the correction factor in ms.
  - If the clock is 1s slow after 24 hours, you have to specify a correction factor of "+1000" ms.
Protection

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

4.8.3 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.  

General

- Short description
  - Here the short description "MPI/DP" for the interface is specified.
- 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 function 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.
4.9 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 1 Module

<table>
<thead>
<tr>
<th>Slot</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CPU ...</td>
</tr>
<tr>
<td>2</td>
<td>PN-IO</td>
</tr>
</tbody>
</table>

### Slot 2 Module

<table>
<thead>
<tr>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU ...</td>
</tr>
</tbody>
</table>

### Slot 3 Module

<table>
<thead>
<tr>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>X2</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

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
4.10 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. ☞ ‘VSC’ on page 78

4.10.1 Transfer via MPI / optional PROFIBUS

General

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

- X3: MPI(PB) ☞ ‘X3: MPI(PB) interface’ on page 38
- X2: PtP(MPI) ☞ ‘X2: PtP(MPI) interface’ on page 38

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

The structure of a MPI net is electrically identical with the structure of a PROFIBUS net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and PROFIBUS cables. Per default the MPI net runs with 187.5kbaud. 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.
Proceding 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 38400 baud 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. In slave operation you have also to enable the option ‘Test, Commissioning, Routing’ when selecting the slave mode.

4.10.2 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.

4.10.3 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
- AUTOLEAD.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.

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.
- AUTOLEAD.WLD is read from the memory card after PowerON.

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

Transfer memory card ➔ CPU

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

The write command is controlled by means of the block area of the Siemens SIMATIC Manager ‘PLC ➔ Copy RAM to ROM’. The SD LED lights up during the write access. When the LED expires, the write process is finished.

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

After accessing the memory card you can find a diagnostics entry in the CPU. To monitor the diagnostics entries you choose in the Siemens SIMATIC manager ‘PLC → Module information’. Via the register “Diagnostic Buffer” you reach the diagnostic window. Chapter 4.19 ‘VIPA specific diagnostic entries’ on page 83

4.11 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

Web page with selected CPU

Info - Overview

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

Info - Expert View

<table>
<thead>
<tr>
<th></th>
<th>Data</th>
<th>Parameter</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device (VIPA 014-CEF0R00) information</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td>Ordering Info</td>
<td>014-CEF0R00</td>
</tr>
<tr>
<td>Value</td>
<td></td>
<td>Serial</td>
<td>00108765</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Version</td>
<td>01V08.001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HW Revision</td>
<td>01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software</td>
<td>01</td>
</tr>
</tbody>
</table>

[ Expert View ... ]

Runtime Info

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation Mode</td>
<td>RUN</td>
<td>CPU: Status information</td>
</tr>
<tr>
<td>Mode Switch</td>
<td>RUNP</td>
<td>CPU: Date, time</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>System Time</td>
<td>31.10.13 18:58:01</td>
<td>CPU: Cyclic time:</td>
</tr>
<tr>
<td></td>
<td>cur = 1000µs, min = 0µs, max = 2000µs, avg = 281µs</td>
<td>min = minimum</td>
</tr>
<tr>
<td></td>
<td>cur = current</td>
<td>cur = current</td>
</tr>
<tr>
<td>ArmLoad</td>
<td>cur = 44%, max = 50%</td>
<td>max = maximum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>avg = average</td>
</tr>
<tr>
<td>Cycle Time</td>
<td>cur = 1000µs, min = 0µs, max = 2000µs, avg = 281µs</td>
<td>CPU: Cyclic time:</td>
</tr>
<tr>
<td></td>
<td>cur = current</td>
<td>cur = current</td>
</tr>
<tr>
<td></td>
<td>max = maximum</td>
<td>max = maximum</td>
</tr>
<tr>
<td></td>
<td>avg = average</td>
<td>avg = average</td>
</tr>
<tr>
<td>RS485 X2</td>
<td>PTP</td>
<td>Operating mode RS485</td>
</tr>
<tr>
<td>RS485 X3</td>
<td>MPI</td>
<td></td>
</tr>
<tr>
<td><strong>Onboard Ethernet</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device Name</td>
<td>Onboard PG/OP</td>
<td>Ethernet PG/OP channel:</td>
</tr>
<tr>
<td>MAC</td>
<td>00:20:D5:01:7A:D1</td>
<td>Address</td>
</tr>
<tr>
<td>IP</td>
<td>172.20.120.40</td>
<td></td>
</tr>
<tr>
<td>Mask</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>Gateway</td>
<td>172.20.120.40</td>
<td></td>
</tr>
<tr>
<td><strong>Memory Usage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoadMem</td>
<td>0/196608 bytes</td>
<td>CPU: Information to memory configuration</td>
</tr>
<tr>
<td>WorkMemCode</td>
<td>0/32768 bytes</td>
<td>Load memory, work memory (code/data)</td>
</tr>
<tr>
<td>WorkMemData</td>
<td>0/32768 bytes</td>
<td></td>
</tr>
<tr>
<td><strong>VIPASetCard Info</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSD...</td>
<td>Information for the support</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSC...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSC-Trial-Time</td>
<td>71:59</td>
<td>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 (abnormal operating mode). This parameter is only visible when the VSC of an enabled function is removed.</td>
</tr>
<tr>
<td>Memory Extension</td>
<td>0 bytes</td>
<td>Size of the additional memory, which was activated by means of a VSC.</td>
</tr>
<tr>
<td>Profibus</td>
<td>PB NO</td>
<td>Type of the PROFIBUS functionality, which was activated by means of a VSC.</td>
</tr>
<tr>
<td>Flash System</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VIPA System SLIO

Deployment CPU 014

Accessing the web server

| File System | V1.0.2 | CPU: Information for the support |
| PRODUCT | VIPA 014-CEF0R00 V1.0.1.4 Px000197.pkg | CPU: Name, firmware version, package |
| HARDWARE | V0.1.0.0 5816A-V10 MX000225.004 | CPU: Information for the support |
| Bx000501 | V1.0.1.4 |
| 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

*VIPA*

| Module 1 (VIPA 021-1BD00) information |
|---|---|
| Name | Value |
| Ordering Info | 021-1BD00 |
| Serial | 00103265 |
| Version | 01V30.001 |
| HW Revision | 01 |

Info

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

Data

Here the address and the state of the inputs respectively outputs are listed.

Parameter

With parameterizable modules e.g. analog modules the parameter setting is shown here. These come from the hardware configuration.
4.12 Operating modes

4.12.1 Overview

The CPU can be in one of 4 operating modes:
- Operating mode STOP
- Operating mode START-UP (OB 100 - restart / OB 102 - cold start *)
- Operating mode RUN
- Operating mode 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.
- Command output disable (BASP) is activated this means the all digital outputs are disabled.
- RUN-LED off
- STOP-LED on

Operating mode START-UP
- During the transition from STOP to RUN a call is issued to the start-up organization block OB 100. The processing time for this OB is not monitored. The START-UP OB may issue calls to other blocks.
- All digital outputs are disabled during the START-UP, this means BASP is activated.
- RUN-LED 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.

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

Please consider that the OB 102 (cold start) may exclusively be used for treatment of a watchdog error.

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

**CAUTION!**

**Breakpoint operation**

In the breakpoint operation, the outputs are enabled and activated during program execution. If the CPU program reaches a breakpoint, all outputs are disabled respectively substitute values are issued.

The breakpoint mode is not suitable for productive systems this may even be "dangerous"!

---

**Precondition**

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

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

**Approach for working with breakpoints**

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

**Behavior in operating state HOLD**

- The RUN-LED blinks and the STOP-LED is on.
- The execution of the code is stopped. No level is further executed.
- The real-time clock runs just running.
- The outputs were disabled (BASP is activated).
- Configured CP connections remain exist.
4.12.2 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:

<table>
<thead>
<tr>
<th>Event</th>
<th>concerns</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUN → STOP</td>
<td>general</td>
<td>BASP (Befehls-Ausgabe-Sperre, i.e. command output lock) is set.</td>
</tr>
<tr>
<td></td>
<td>central digital outputs</td>
<td>The outputs are disabled.</td>
</tr>
<tr>
<td></td>
<td>central analog outputs</td>
<td>The outputs are disabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Voltage outputs issue 0V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Current outputs 0...20mA issue 0mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Current outputs 4...20mA issue 4mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If configured also substitute values may be issued.</td>
</tr>
<tr>
<td></td>
<td>decentral outputs</td>
<td>Same behaviour as the central digital/analog outputs.</td>
</tr>
<tr>
<td></td>
<td>decentral inputs</td>
<td>The inputs are cyclically be read by the decentralized station and the recent values are put at disposal.</td>
</tr>
<tr>
<td>STOP → RUN res. PowerON</td>
<td>general</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>decentral inputs</td>
<td>The inputs are be read by the decentralized station and the recent values are put at disposal.</td>
</tr>
<tr>
<td>RUN</td>
<td>general</td>
<td>The program is cyclically executed: Read PII → OB 1 → Write PIO.</td>
</tr>
</tbody>
</table>

PII = Process image inputs
PIO = Process image outputs
4.13 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.

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 78

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!  

issing Chapter 4.15 ‘Factory reset’ on page 77

### 4.14 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 operating mode switch in STOP position, 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.

The latest firmware versions can be found in the "service" area at www.vipa.com. For example the following files are necessary for the firmware update of the CPU 014 and its components with hardware release 1:

- CPU 014, Hardware release 1: Px000197.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.  

issing Chapter 4.6 ‘Hardware configuration - Ethernet PG/OP channel’ on page 55

Load firmware and transfer it to memory card

1. ➔ Go to www.vipa.com
2. ➔ Click ‘Service Support ➔ Downloads ➔ Firmware’.
3. ➔ Via ‘System SLIO ➔ CPU’ navigate to your CPU and download the zip file to your PC.
4. 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 4.15 'Factory reset' on page 77

---

**Transfer firmware from memory card into CPU**

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

1. Switch the operating mode switch of your CPU in position STOP. 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 4.15 'Factory reset' on page 77
4.15 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 FACTORY_RESET. *CMD - Auto commands* on page 81

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.
4.16 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 - VIPA SetCard**
  - 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 an **Overall reset** is to be established. See Chapter 4.13 ‘Overall reset’ on page 73

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

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

VSD

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

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

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

**CAUTION!**

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

VSC

The VSC is a VSD with the possibility to enable optional functions. Here you have the opportunity to accordingly expand your work memory respectively enable field bus functions. Information about the enabled functions can be shown via the web page. See Chapter 4.11 ‘Accessing the web server’ on page 67
CAUTION!
Please regard that the VSC must remain plugged when you've enabled optional functions at your CPU. Otherwise the SF LED is on and the CPU switches to STOP after 72 hours. As soon as an activated VSC is not plugged, the SF LED is on and the "TrialTimer" counts downwards from 72 hours to 0. After 72 hours the CPU switches to STOP state. By plugging the VSC, the SF LED expires and the CPU is running again without any restrictions.

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

Accessing the storage medium

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

After overall reset
- The CPU checks if a VSC is inserted. If so, the corresponding optional functions 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.  
  & further information on page 75

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

The FC/SFC 208 ... FC/SFC 215 and FC/SFC 195 allow you to include the memory card access into your user application. More can be found in the manual operation list (HB00_OPL_SP7) of your CPU.

4.17 Extended know-how protection

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

Standard protection
The standard protection from Siemens transfers also protected blocks to the PG but their content is not displayed But with according manipulation the Know-how protection is not guaranteed.
**Extended protection**

The "extended" know-how protection developed by VIPA offers the opportunity to store blocks permanently in the CPU. With the "extended" protection you transfer the protected blocks to a memory card into a WLD-file named protect.wld. By plugging the memory card and then an overall 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.

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 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.

4.18 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.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Diagnostics entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD_START</td>
<td>In the first line CMD_START is to be located.</td>
<td>0xE801</td>
</tr>
<tr>
<td></td>
<td>There is a diagnostics entry if CMD_START is missing.</td>
<td>0xE8FE</td>
</tr>
<tr>
<td>WAIT1SECOND</td>
<td>Waits about 1 second.</td>
<td>0xE803</td>
</tr>
<tr>
<td>LOAD_PROJECT</td>
<td>The function &quot;Overall reset and reload from memory card&quot; is executed.</td>
<td>0xE805</td>
</tr>
<tr>
<td>SAVE_PROJECT</td>
<td>The recent project (blocks and hardware configuration) is stored as &quot;s7prog.wld&quot; at the memory card. If the file just exists it is renamed to &quot;s7prog.old&quot;. If your CPU is password protected so you have to add this as parameter. Otherwise there is no project written. Example: SAVE_PROJECT password</td>
<td>0xE806</td>
</tr>
<tr>
<td>FACTORY_RESET</td>
<td>Executes &quot;factory reset&quot;.</td>
<td>0xE807</td>
</tr>
<tr>
<td>DIAGBUF</td>
<td>The current diagnostics buffer of the CPU is stored as &quot;diagbuff.txt&quot; at the memory card.</td>
<td>0xE80B</td>
</tr>
</tbody>
</table>
### Command Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Diagnostics entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET_NETWORK</td>
<td>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.</td>
<td>0xE80E</td>
</tr>
<tr>
<td>CMD_END</td>
<td>In the last line CMD_END is to be located.</td>
<td>0xE802</td>
</tr>
</tbody>
</table>

### Examples

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

#### Example 1

```
CMD_START
LOAD_PROJECT proj.wld
WAIT1SECOND
DIAGBUF
CMD_END
... arbitrary text ...
```

- **Marks the start of the command sequence (0xE801)**
- **Execute an overall reset and load "proj.wld" (0xE805)**
- **Wait ca. 1s (0xE803)**
- **Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)**
- **Marks the end of the command sequence (0xE802)**

#### Example 2

```
CMD_START
LOAD_PROJECT proj2.wld
WAIT1SECOND
WAIT1SECOND
SET_NETWORK 172.16.129.210,255.255.224.0,172.16.129.210
WAIT1SECOND
WAIT1SECOND
DIAGBUF
CMD_END
... arbitrary text ...
```

- **Marks the start of the command sequence (0xE801)**
- **Execute an overall reset and load "proj2.wld" (0xE805)**
- **Wait ca. 1s (0xE803)**
- **Wait ca. 1s (0xE803)**
- **IP parameter (0xE80E)**
- **Wait ca. 1s (0xE803)**
- **Wait ca. 1s (0xE803)**
- **Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)**
- **Marks the end of the command sequence (0xE802)**

---

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.
4.19 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

<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 |
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 |
| 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 backplane 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 |
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xE0CC</td>
<td>Communication error MPI / Serial</td>
</tr>
<tr>
<td></td>
<td>ZInfo1: Code</td>
</tr>
<tr>
<td></td>
<td>1: Wrong priority</td>
</tr>
<tr>
<td></td>
<td>2: Buffer overflow</td>
</tr>
<tr>
<td></td>
<td>3: Frame format error</td>
</tr>
<tr>
<td></td>
<td>4: Wrong SSL request (SSL-ID not valid)</td>
</tr>
<tr>
<td></td>
<td>5: Wrong SSL request (SSL-SubID not valid)</td>
</tr>
<tr>
<td></td>
<td>6: Wrong SSL request (SSL-Index not valid)</td>
</tr>
<tr>
<td></td>
<td>7: Wrong value</td>
</tr>
<tr>
<td></td>
<td>8: Wrong RetVal</td>
</tr>
<tr>
<td></td>
<td>9: Wrong SAP</td>
</tr>
<tr>
<td></td>
<td>10: Wrong connection type</td>
</tr>
<tr>
<td></td>
<td>11: Wrong sequence number</td>
</tr>
<tr>
<td></td>
<td>12: Faulty block number in the telegram</td>
</tr>
<tr>
<td></td>
<td>13: Faulty block type in the telegram</td>
</tr>
<tr>
<td></td>
<td>14: Inactive function</td>
</tr>
<tr>
<td></td>
<td>15: Wrong size in the telegram</td>
</tr>
<tr>
<td></td>
<td>20: Error writing to memory card</td>
</tr>
<tr>
<td></td>
<td>90: Faulty buffer size</td>
</tr>
<tr>
<td></td>
<td>98: Unknown error</td>
</tr>
<tr>
<td></td>
<td>99: Internal error</td>
</tr>
<tr>
<td>0xE0CD</td>
<td>Error at DP-V1 job management</td>
</tr>
<tr>
<td>0xE0CE</td>
<td>Error: Timeout at sending of the i-slave diagnostics</td>
</tr>
<tr>
<td>0xE0CF</td>
<td>Timeout at loading of a new HW configuration (timeout: 39 seconds)</td>
</tr>
<tr>
<td>0xE100</td>
<td>Memory card access error</td>
</tr>
<tr>
<td>0xE101</td>
<td>Memory card error file system</td>
</tr>
<tr>
<td>0xE102</td>
<td>Memory card error FAT</td>
</tr>
<tr>
<td>0xE104</td>
<td>Memory card error at saving</td>
</tr>
<tr>
<td>0xE200</td>
<td>Memory card writing finished (Copy Ram2Rom)</td>
</tr>
<tr>
<td>0xE210</td>
<td>Memory card reading finished (reload after overall reset)</td>
</tr>
<tr>
<td>0xE21E</td>
<td>Memory card reading: Error at reload (after overall reset), file &quot;Protect.wld&quot;</td>
</tr>
<tr>
<td></td>
<td>too big</td>
</tr>
<tr>
<td>0xE21F</td>
<td>Memory card reading: Error at reload (after overall reset), file read error,</td>
</tr>
<tr>
<td></td>
<td>out of memory</td>
</tr>
<tr>
<td>0xE300</td>
<td>Internal flash writing finished (Copy Ram2Rom)</td>
</tr>
<tr>
<td>0xE310</td>
<td>Internal flash writing finished (reload after battery failure)</td>
</tr>
<tr>
<td>0xE311</td>
<td>Internal flash fx0000yy.wld file too big, load failure</td>
</tr>
<tr>
<td>Event-ID</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>0xE400</td>
<td>Memory card with the option memory expansion was plugged</td>
</tr>
<tr>
<td>0xE401</td>
<td>Memory card with the option memory expansion was removed</td>
</tr>
<tr>
<td>0xE402</td>
<td>The PROFIBUS DP master functionality is disabled. The interface acts further as MPI interface</td>
</tr>
<tr>
<td>0xE403</td>
<td>The PROFIBUS DP slave functionality is disabled. The interface acts further as MPI interface</td>
</tr>
<tr>
<td>0xE500</td>
<td>Memory management: Deleted block without corresponding entry in Block List ZInfo2: BlockType ZInfo3: BlockNo</td>
</tr>
<tr>
<td>0xE604</td>
<td>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</td>
</tr>
<tr>
<td>0xE701</td>
<td>Internal error - Please contact the VIPA Hotline!</td>
</tr>
<tr>
<td>0xE703</td>
<td>Internal error - Please contact the VIPA Hotline!</td>
</tr>
<tr>
<td>0xE720</td>
<td>Internal error - Please contact the VIPA Hotline!</td>
</tr>
<tr>
<td>0xE721</td>
<td>Internal error - Please contact the VIPA Hotline!</td>
</tr>
<tr>
<td>0xE801</td>
<td>CMD - Auto command: CMD_START recognized and successfully executed</td>
</tr>
<tr>
<td>0xE802</td>
<td>CMD - Auto command: CMD_End recognized and successfully executed</td>
</tr>
<tr>
<td>0xE803</td>
<td>CMD - Auto command: WAIT1SECOND recognized and successfully executed</td>
</tr>
<tr>
<td>0xE804</td>
<td>CMD - Auto command: WEBPAGE recognized and successfully executed</td>
</tr>
<tr>
<td>0xE805</td>
<td>CMD - Auto command: LOAD_PROJECT recognized and successfully executed</td>
</tr>
<tr>
<td>0xE806</td>
<td>CMD - Auto command: SAVE_PROJECT ZInfo3: 0x0000: SAVE_PROJECT recognized and successfully executed ZInfo3: 0x8000: Error during SAVE_PROJECT e.g. wrong password</td>
</tr>
<tr>
<td>0xE807</td>
<td>CMD - Auto command: FACTORY_RESET recognized and successfully executed</td>
</tr>
<tr>
<td>0xE80B</td>
<td>CMD - Auto command: DIAGBUF recognized and successfully executed</td>
</tr>
<tr>
<td>0xE80E</td>
<td>CMD - Auto command: SET_NETWORK recognized and successfully executed</td>
</tr>
<tr>
<td>0xE816</td>
<td>CMD - Auto command: SAVE_PROJECT: Error - CPU has been reset - no wld file was created.</td>
</tr>
<tr>
<td>Event-ID</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>0xE8FB</td>
<td>CMD - Auto command: Error: Initialization of the Ethernet PG/OP channel by means of SET_NETWORK is faulty</td>
</tr>
<tr>
<td>0xE8FC</td>
<td>CMD - Auto command: Error: Some IP parameters missing in SET_NETWORK</td>
</tr>
<tr>
<td>0xE8FE</td>
<td>CMD - Auto command: Error: CMD_START missing</td>
</tr>
<tr>
<td>0xE8FF</td>
<td>CMD - Auto command: Error: Error while reading CMD file (memory card error)</td>
</tr>
<tr>
<td>0xE901</td>
<td>Check sum error</td>
</tr>
<tr>
<td>0xEA00</td>
<td>Internal error - Please contact the VIPA Hotline!</td>
</tr>
<tr>
<td>0xEA01</td>
<td>Internal error - Please contact the VIPA Hotline!</td>
</tr>
<tr>
<td>0xEA02</td>
<td>SBUS: Internal error (internal plugged sub module not recognized) ZInfo1: Internal slot</td>
</tr>
<tr>
<td>0xEA03</td>
<td>SBUS: Communication error CPU - PROFINET I/O controller: ZInfo1: Slot ZInfo2: Status (0: OK, 1: ERROR, 2: BUSY, 3: TIMEOUT, 4: LOCKED, 5: UNKNOWN)</td>
</tr>
<tr>
<td>0xEA04</td>
<td>SBUS: Multiple parametrization of a periphery address ZInfo1: Periphery address ZInfo2: Slot ZInfo3: Data width</td>
</tr>
<tr>
<td>0xEA05</td>
<td>Internal error - Please contact the VIPA Hotline!</td>
</tr>
<tr>
<td>0xEA07</td>
<td>Internal error - Please contact the VIPA Hotline!</td>
</tr>
<tr>
<td>0xEA08</td>
<td>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</td>
</tr>
<tr>
<td>0xEA09</td>
<td>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</td>
</tr>
<tr>
<td>0xEA10</td>
<td>SBUS: Input periphery address outside the periphery area ZInfo1: Periphery address ZInfo2: Slot ZInfo3: Data width</td>
</tr>
<tr>
<td>Event-ID</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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!                                                                                                                                                                                                                                                                                                                     |
| 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                                                                                                                                                                                                                                                                                 |
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 |
| 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 |
<p>| 0xEA65   | Internal error - Please contact the VIPA Hotline! |</p>
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 |
| 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: ServicelIdentifier  
|           | DatId: 0  
|           | ZInfo1: ServiceError.Code  
|           | ZInfo2: ServiceError.Detail  
<p>|           | ZInfo3: StackError.Error.AdditionalDetail |</p>
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0xEA6B   | PROFINET IO controller  
Vendor ID mismatch  
PK: Rackslot  
OBNo: PLC Mode  
DatId: 0  
ZInfo1: Device ID  
ZInfo2: -  
ZInfo3: - |
| 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: - |
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xEA6F</td>
<td>PROFINET IO controller</td>
</tr>
<tr>
<td></td>
<td>PN module mismatch</td>
</tr>
<tr>
<td></td>
<td>PK: Rackslot</td>
</tr>
<tr>
<td></td>
<td>OBNo: PLC-Mode</td>
</tr>
<tr>
<td></td>
<td>DatId: 0</td>
</tr>
<tr>
<td></td>
<td>ZInfo1: Device ID</td>
</tr>
<tr>
<td></td>
<td>ZInfo2: -</td>
</tr>
<tr>
<td></td>
<td>ZInfo3: -</td>
</tr>
<tr>
<td>0xEA97</td>
<td>Storage error SBUS service channel</td>
</tr>
<tr>
<td></td>
<td>ZInfo3 = Slot</td>
</tr>
<tr>
<td>0xEA98</td>
<td>Timeout at waiting for reboot of a SBUS module (server)</td>
</tr>
<tr>
<td>0xEA99</td>
<td>Error at file reading via SBUS</td>
</tr>
<tr>
<td>0xEAA0</td>
<td>Emac Error occurred</td>
</tr>
<tr>
<td></td>
<td>OBNo: Current PLC mode</td>
</tr>
<tr>
<td></td>
<td>ZInfo1: Diagnostics address of the master / controller</td>
</tr>
<tr>
<td></td>
<td>ZInfo2:</td>
</tr>
<tr>
<td></td>
<td>0: None Rx queue is full</td>
</tr>
<tr>
<td></td>
<td>1: No send buffer available</td>
</tr>
<tr>
<td></td>
<td>2: Send stream was cut off; sending failed</td>
</tr>
<tr>
<td></td>
<td>3: Exhausted retries</td>
</tr>
<tr>
<td></td>
<td>4: No receive buffer available in Emac DMA</td>
</tr>
<tr>
<td></td>
<td>5: Emac DMA transfer aborted</td>
</tr>
<tr>
<td></td>
<td>6: Queue overflow</td>
</tr>
<tr>
<td></td>
<td>7: Unexpected frame received</td>
</tr>
<tr>
<td></td>
<td>ZInfo3: Number of errors, which occurred</td>
</tr>
<tr>
<td>0xEB03</td>
<td>SLIO error on IO mapping</td>
</tr>
<tr>
<td>0xEB10</td>
<td>SLIO error: Bus error</td>
</tr>
<tr>
<td></td>
<td>ZInfo1: Type of error</td>
</tr>
<tr>
<td></td>
<td>0x82: ErrorAlarm</td>
</tr>
</tbody>
</table>
### VIPA specific diagnostic entries

<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xEB20</td>
<td>SLIO error: Interrupt information undefined</td>
</tr>
<tr>
<td>0xEB21</td>
<td>SLIO error on accessing the configuration data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xEC03</td>
<td>EtherCAT: Configuration error</td>
</tr>
<tr>
<td></td>
<td>ZInfo1: Errorcode</td>
</tr>
<tr>
<td></td>
<td>1: NUMBER_OF_SLAVES_NOT_SUPPORTED</td>
</tr>
<tr>
<td></td>
<td>2: SYSTEM_IO_NR_INVALID</td>
</tr>
<tr>
<td></td>
<td>3: INDEX_FROM_SLOT_ERROR</td>
</tr>
<tr>
<td></td>
<td>4: MASTER_CONFIG_INVALID</td>
</tr>
<tr>
<td></td>
<td>5: MASTER_TYPE_ERROR</td>
</tr>
<tr>
<td></td>
<td>6: SLAVE_DIAG_ADDR_INVALID</td>
</tr>
<tr>
<td></td>
<td>7: SLAVE_ADDR_INVALID</td>
</tr>
<tr>
<td></td>
<td>8: SLAVE_MODULE_IO_CONFIG_INVALID</td>
</tr>
<tr>
<td></td>
<td>9: LOG_ADDR_ALREADY_IN_USE</td>
</tr>
<tr>
<td></td>
<td>10: NULL_PTR_CHECK_ERROR</td>
</tr>
<tr>
<td></td>
<td>11: IO_MAPPING_ERROR</td>
</tr>
<tr>
<td></td>
<td>12: ERROR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xEC04</td>
<td>EtherCAT: Multiple configuration of a periphery address</td>
</tr>
<tr>
<td></td>
<td>ZInfo1: Periphery address</td>
</tr>
<tr>
<td></td>
<td>ZInfo2: Slot</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xEC10</td>
<td>EtherCAT: Restoration bus with its slaves</td>
</tr>
<tr>
<td></td>
<td>OB start Info (Local data) StartEvent and Eventclass: 0xEC10</td>
</tr>
<tr>
<td></td>
<td>DatID: 0xXXYY:</td>
</tr>
<tr>
<td></td>
<td>XX=0x54 with input address in ZInfo1,</td>
</tr>
<tr>
<td></td>
<td>XX=0x55 with output address.</td>
</tr>
<tr>
<td></td>
<td>YY=0x00 Station not available,</td>
</tr>
<tr>
<td></td>
<td>YY=0x01 Station available (process data)</td>
</tr>
<tr>
<td></td>
<td>ZInfo1: 0xXXYY (XX=OldState, YY=NewState)</td>
</tr>
<tr>
<td></td>
<td>ZInfo2: Diagnostics address of the master</td>
</tr>
<tr>
<td></td>
<td>ZInfo3: Number of stations, which are not in the same state as the master (&gt; 0)</td>
</tr>
<tr>
<td>Event-ID</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 0xEC11   | EtherCAT: Restoration bus with missing slaves  
OB start Info (Local data) StartEvent and Eventclass: 0xEC11  
DatID:  
0xXXYY:  
XX=0x54 with input address in ZInfo1,  
XX=0x55 with output address.  
YY=0x00 Station not available,  
YY=0x01 Station available (process data)  
ZInfo1: 0xXXYY (XX=OldState, YY=NewState)  
ZInfo2: Diagnostics address of the master  
ZInfo3: Number of stations, which are not in the same state as the master (> 0) |
| 0xEC12   | EtherCAT: restoration slave  
OB start Info (Local data) StartEvent and Eventclass: 0xEC12  
DatID:  
0xXXYY:  
XX=0x54 with input address in ZInfo1,  
XX=0x55 with output address.  
YY=0x00 Station not available,  
YY=0x01 Station available (process data)  
ZInfo1: 0xXXYY (XX=OldState, YY=NewState)  
ZInfo2: Diagnostics of the Station  
ZInfo3: AlStatusCode |
| 0xEC30   | EtherCAT: Topology OK  
OB start Info (Local data) StartEvent and Eventclass: 0xEC30  
ZInfo2: Diagnostics address of the master |
| 0xEC50   | EtherCAT: DC not in Sync  
ZInfo1: Diagnostics address of the master |
| 0xED10   | EtherCAT: Bus failure  
OB start Info (Local data) StartEvent and Eventclass: 0xED10  
DatID:  
0xXXYY:  
XX=0x54 with input address in ZInfo1,  
XX=0x55 with output address.  
YY=0x00 Station not available,  
YY=0x01 Station available (process data)  
ZInfo1: 0xXXYY (XX=OldState, YY=NewState)  
ZInfo2: Diagnostics address of the master  
ZInfo3: Number of stations, which are not in the same state as the master |
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0xED12   | EtherCAT: Failure slave  
OB start Info (Local data) StartEvent and Eventclass: 0xED12  
DatID:  
0xXXYY:  
XX=0x54 with input address in ZInfo1,  
XX=0x55 with output address.  
YY=0x00 Station not available,  
YY=0x01 Station available (process data)  
ZInfo1: 0xXXYY (XX=OldState, YY=NewState)  
ZInfo2: Diagnostics of the Station  
ZInfo3: AlStatusCode |
| 0xED20   | EtherCAT: Bus state change without calling OB86  
OB start Info (Local data) StartEvent and Eventclass: 0xED20  
DatID:  
0xXXYY:  
XX=0x54 with input address in ZInfo1,  
XX=0x55 with output address.  
YY=0x00 Station not available,  
YY=0x01 Station available (process data)  
ZInfo1: 0xXXYY (XX=OldState, YY=NewState)  
ZInfo2: Diagnostics address of the master  
ZInfo3: Number of stations, which are not in the same state as the master |
| 0xED21   | EtherCAT: error in bus state change  
OB: 0x00  
PK: 0x00  
DatID:  
0xXXYY:  
XX=0x54 with input address in ZInfo1,  
XX=0x55 with output address.  
YY=0x00 Station not available,  
YY=0x01 Station available (process data)  
ZInfo1: 0xXXYY (XX = current state, YY = expected state)  
ZInfo2: Diagnostics address of the master  
ZInfo3: ErrorCode:  
0x0008: Busy  
0x000B: Invalid Parameter  
0x000E: Invalid State  
0x0010: Timeout |
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| 0xED22   | EtherCAT: Bus state change without calling OB86  
OB start Info (Local data) StartEvent and Eventclass: 0xED22  
DatID:  
0xXXYY:  
XX=0x54 with input address in ZInfo1,  
XX=0x55 with output address.  
YY=0x00 Station not available,  
YY=0x01 Station available (process data)  
ZInfo1: 0xXXYY (XX=OldState, YY=NewState)  
ZInfo2: Diagnostics of the Station  
ZInfo3: AIStatusCode |
| 0xED30   | EtherCAT: Topology Mismatch  
OB start Info (Local data) StartEvent and Eventclass: 0xED30  
ZInfo2: Diagnostics address of the master |
| 0xED31   | EtherCAT: Interrupt Queue Overflow  
OB start Info (Local data) StartEvent and Eventclass: 0xED31  
ZInfo2: Diagnostics address of the master |
| 0xED40 ... 0xED4F | Internal error - Please contact the VIPA Hotline! |
| 0xED50   | EtherCAT: DC not in Sync  
ZInfo1: Diagnostics address of the master |
| 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  
ZInfo2: EtherCAT station address  
ZInfo3: AIStatusCode (EtherCAT specific error code) |
<table>
<thead>
<tr>
<th>Event-ID</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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  
ZInfo2: 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  
ZInfo2: EtherCAT station address  
ZInfo3: 0 |
| 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! |
### Control and monitoring of variables with test functions

#### Overview
- For troubleshooting purposes and to display the status of certain variables you can access certain test functions via the menu item **Debug** of the Siemens SIMATIC Manager.
- The status of the operands and the RLO can be displayed by means of the test function **'Debug ➔ Monitor'**.
- The status of the operands and the RLO can be displayed by means of the test function **'PLC ➔ Monitor/Modify Variables'**.

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

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

For this reason, jumps or time and process alarms can result in the value displayed during program execution remaining at 0 for the items below:
- the result of the logical operation RLO
- Status / AKKU 1
- AKKU 2
- Condition byte
- absolute memory address SAZ. In this case SAZ is followed by a "?".
This test function returns the condition of a selected operand (inputs, outputs, flags, data word, counters or timers) at the end of program execution. This information is obtained from the corresponding area of the selected operands. During the controlling of variables respectively in operating mode STOP the input area is directly read. Otherwise only the process image of the selected operands is displayed.

Control of outputs
- Serves to check the wiring and proper operation of output modules.
- If the CPU is in RUN mode, so only outputs can be controlled, which are not controlled by the user program. Otherwise values would be instantly overwritten.
- If the CPU is in STOP - even without user program, so you need to disable the command output lock BASP (‘Enable PO’). Then you can control the outputs arbitrarily

Controlling variables
- The following variables may be modified: I, Q, M, T, C and D.
- The process image of binary and digital operands is modified independently of the operating mode of the CPU.
- When the operating mode is RUN the program is executed with the modified process variable. When the program continues they may, however, be modified again without notification.

Forcing variables
- You can pre-set individual variables of a user program with fixed values so that they can not be changed or overwritten by the user program of the CPU.
- By pre-setting of variables with fixed values, you can set certain situations for your user program and thus test the programmed functions.

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

Even after a power cycle forced variables remain forced with its value, until the force function is disabled.

These functions should only be used for test purposes respectively for troubleshooting. More information about the usage of these functions may be found in the manual of your configuration tool.
5 Deployment PtP communication

5.1 Fast introduction

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

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

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

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

**Use FCs 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:

<table>
<thead>
<tr>
<th>FC/SFC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC/SFC 216</td>
<td>SER_CFG</td>
</tr>
<tr>
<td>FC/SFC 217</td>
<td>SER_SND</td>
</tr>
<tr>
<td>FC/SFC 218</td>
<td>SER_RCV</td>
</tr>
</tbody>
</table>
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
### 9pin SubD jack

<table>
<thead>
<tr>
<th>Pin</th>
<th>RS485</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>n.c.</td>
</tr>
<tr>
<td>2</td>
<td>M24V</td>
</tr>
<tr>
<td>3</td>
<td>RxD/TxD-P (Line B)</td>
</tr>
<tr>
<td>4</td>
<td>RTS</td>
</tr>
<tr>
<td>5</td>
<td>M5V</td>
</tr>
<tr>
<td>6</td>
<td>P5V</td>
</tr>
<tr>
<td>7</td>
<td>P24V</td>
</tr>
<tr>
<td>8</td>
<td>RxD/TxD-N (Line A)</td>
</tr>
<tr>
<td>9</td>
<td>n.c.</td>
</tr>
</tbody>
</table>

### 5.4 Parametrization

#### 5.4.1 FC/SFC 216 - SER_CFG

**Description**

The parametrization happens during runtime deploying the FC/SFC 216 (SER_CFG). You have to store the parameters for STX/ETX, 3964R, USS and Modbus in a DB.
### Parameters

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

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: 19200bit/s x 0.008s ≈ 154bit → (9Ah)
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.
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:

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Declaration</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAPTR</td>
<td>IN</td>
<td>ANY</td>
<td>Pointer to Data Buffer for sending data</td>
</tr>
<tr>
<td>DATALEN</td>
<td>OUT</td>
<td>WORD</td>
<td>Length of data sent</td>
</tr>
<tr>
<td>RETVAL</td>
<td>OUT</td>
<td>WORD</td>
<td>Return value (0 = OK)</td>
</tr>
</tbody>
</table>

**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)

**Error code**

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h</td>
<td>Send data - ready</td>
</tr>
<tr>
<td>1000h</td>
<td>Nothing sent (data length 0)</td>
</tr>
<tr>
<td>20xxh</td>
<td>Protocol executed error free with xx bit pattern for diagnosis</td>
</tr>
<tr>
<td>7001h</td>
<td>Data is stored in internal buffer - active (busy)</td>
</tr>
<tr>
<td>7002h</td>
<td>Transfer - active</td>
</tr>
<tr>
<td>80xxh</td>
<td>Protocol executed with errors with xx bit pattern for diagnosis (no acknowledgement by partner)</td>
</tr>
<tr>
<td>90xxh</td>
<td>Protocol not executed with xx bit pattern for diagnosis (no acknowledgement by partner)</td>
</tr>
<tr>
<td>Error code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 8x24h      | Error in FC/SFC parameter x, where x:  
1: Error in DATAPTR  
2: Error in DATALEN |
| 8122h      | Error in parameter DATAPTR (e.g. DB too short) |
| 807Fh      | Internal error |
| 809Ah      | interface not found e.g. interface is used by PROFIBUS |
| 809Bh      | interface not configured |

### ASCII

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9000h</td>
<td>Buffer overflow (no data send)</td>
</tr>
<tr>
<td>9002h</td>
<td>Data too short (0byte)</td>
</tr>
</tbody>
</table>

### STX/ETX

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9000h</td>
<td>Buffer overflow (no data send)</td>
</tr>
<tr>
<td>9001h</td>
<td>Data too long (&gt;1024byte)</td>
</tr>
<tr>
<td>9002h</td>
<td>Data too short (0byte)</td>
</tr>
<tr>
<td>9004h</td>
<td>Character not allowed</td>
</tr>
</tbody>
</table>

### 3964R

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000h</td>
<td>Send ready without error</td>
</tr>
<tr>
<td>80FFh</td>
<td>NAK received - error in communication</td>
</tr>
<tr>
<td>80FEh</td>
<td>Data transfer without acknowledgement of partner or error at acknowledgement</td>
</tr>
<tr>
<td>9000h</td>
<td>Buffer overflow (no data send)</td>
</tr>
<tr>
<td>9001h</td>
<td>Data too long (&gt;1024byte)</td>
</tr>
<tr>
<td>9002h</td>
<td>Data too short (0byte)</td>
</tr>
</tbody>
</table>

### USS

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000h</td>
<td>Send ready without error</td>
</tr>
<tr>
<td>8080h</td>
<td>Receive buffer overflow (no space for receipt)</td>
</tr>
<tr>
<td>8090h</td>
<td>Acknowledgement delay time exceeded</td>
</tr>
</tbody>
</table>
### Error code

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>80F0h</td>
<td>Wrong checksum in respond</td>
</tr>
<tr>
<td>80FEh</td>
<td>Wrong start sign in respond</td>
</tr>
<tr>
<td>80FFh</td>
<td>Wrong slave address in respond</td>
</tr>
<tr>
<td>9000h</td>
<td>Buffer overflow (no data send)</td>
</tr>
<tr>
<td>9001h</td>
<td>Data too long (&gt;1024byte)</td>
</tr>
<tr>
<td>9002h</td>
<td>Data too short (&lt;2byte)</td>
</tr>
</tbody>
</table>

### Modbus RTU/ASCII Master

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000h</td>
<td>Send ready (positive slave respond)</td>
</tr>
<tr>
<td>2001h</td>
<td>Send ready (negative slave respond)</td>
</tr>
<tr>
<td>8080h</td>
<td>Receive buffer overflow (no space for receipt)</td>
</tr>
<tr>
<td>8090h</td>
<td>Acknowledgement delay time exceeded</td>
</tr>
<tr>
<td>80F0h</td>
<td>Wrong checksum in respond</td>
</tr>
<tr>
<td>80FDh</td>
<td>Length of respond too long</td>
</tr>
<tr>
<td>80FEh</td>
<td>Wrong function code in respond</td>
</tr>
<tr>
<td>80FFh</td>
<td>Wrong slave address in respond</td>
</tr>
<tr>
<td>9000h</td>
<td>Buffer overflow (no data send)</td>
</tr>
<tr>
<td>9001h</td>
<td>Data too long (&gt;1024byte)</td>
</tr>
<tr>
<td>9002h</td>
<td>Data too short (&lt;2byte)</td>
</tr>
</tbody>
</table>
The following text shortly illustrates the structure of programming a send command for the different protocols.
USS / Modbus

```
SFC 217 SER_SND

Busy ?

Y

SFC 218 SER_RCV

Error evaluation

N

RetVal 80xxh / 90xxh ?

Y

RetVal 2001h ?

N

RetVal 2000h ?

Y

RetVal 2000h ?

N

SFC 218 SER_RCV

Data evaluation

End
```

ASCII / STX/ETX

```
SFC 217 SER_SND

RetVal 900xh

Y

Error evaluation

N

End
```
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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Declaration</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAPTR</td>
<td>IN</td>
<td>ANY</td>
<td>Pointer to Data Buffer for received data</td>
</tr>
<tr>
<td>DATALEN</td>
<td>OUT</td>
<td>WORD</td>
<td>Length of received data</td>
</tr>
<tr>
<td>ERROR</td>
<td>OUT</td>
<td>WORD</td>
<td>Error Number</td>
</tr>
<tr>
<td>RETVAL</td>
<td>OUT</td>
<td>WORD</td>
<td>Return value (0 = OK)</td>
</tr>
</tbody>
</table>

**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 124 byte.

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**

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

<table>
<thead>
<tr>
<th>Bit</th>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>overflow</td>
<td>The received telegram exceeds the size of the receive buffer.</td>
</tr>
<tr>
<td>1</td>
<td>char</td>
<td>A sign outside the range 20h ... 7Fh has been received.</td>
</tr>
<tr>
<td>3</td>
<td>overflow</td>
<td>Buffer is full.</td>
</tr>
</tbody>
</table>

### 3964R / Modbus RTU/ASCII Master

<table>
<thead>
<tr>
<th>Bit</th>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>overflow</td>
<td>The received telegram exceeds the size of the receive buffer.</td>
</tr>
</tbody>
</table>

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

Return values of the block:

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

STX/ETX

STX/ETX is a simple protocol with start and end ID, where STX stands for Start of Text and ETX for End of Text.
Any data transferred from the periphery must be preceded by a Start followed by the data characters and the end character. Depending on 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 sent 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:**

```
| STX1 | STX2 | Z1 | Z2 | Zn | ETX1 | ETX2 |
```

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**: Start of Text
- **DLE**: Data Link Escape
- **ETX**: End of Text
- **BCC**: Block Check Character
- **NAK**: Negative Acknowledge

You may transfer a maximum of 255 byte per message.

**Procedure**

<table>
<thead>
<tr>
<th>Active partner</th>
<th>Passive partner</th>
</tr>
</thead>
<tbody>
<tr>
<td>STX</td>
<td></td>
</tr>
<tr>
<td>Monitor delay acknowledgment</td>
<td></td>
</tr>
<tr>
<td>DLE</td>
<td></td>
</tr>
<tr>
<td>Message-data</td>
<td></td>
</tr>
<tr>
<td>DLE</td>
<td></td>
</tr>
<tr>
<td>ETX</td>
<td></td>
</tr>
<tr>
<td>BCC</td>
<td></td>
</tr>
<tr>
<td>Monitor delay acknowledgment</td>
<td></td>
</tr>
<tr>
<td>DLE</td>
<td></td>
</tr>
</tbody>
</table>

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 (Universelle serielle Schnittstelle = 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

<table>
<thead>
<tr>
<th>STX</th>
<th>LGE</th>
<th>ADR</th>
<th>PKE</th>
<th>IND</th>
<th>PWE</th>
<th>STW</th>
<th>HSW</th>
<th>BCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>02h</td>
<td></td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H L</td>
</tr>
</tbody>
</table>

### Slave master telegram

<table>
<thead>
<tr>
<th>STX</th>
<th>LGE</th>
<th>ADR</th>
<th>PKE</th>
<th>IND</th>
<th>PWE</th>
<th>ZSW</th>
<th>HIW</th>
<th>BCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>02h</td>
<td></td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H L</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Telegram structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start sign</td>
</tr>
</tbody>
</table>

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

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:

<table>
<thead>
<tr>
<th>Code</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01h</td>
<td>Read n bits</td>
<td>Read n bits of master output area 0x</td>
</tr>
<tr>
<td>02h</td>
<td>Read n bits</td>
<td>Read n bits of master input area 1x</td>
</tr>
<tr>
<td>03h</td>
<td>Read n words</td>
<td>Read n words of master output area 1x</td>
</tr>
<tr>
<td>04h</td>
<td>Read n words</td>
<td>Read n words master input area 3x</td>
</tr>
<tr>
<td>05h</td>
<td>Write 1 bit</td>
<td>Write 1 bit to master output area 0x</td>
</tr>
<tr>
<td>06h</td>
<td>Write 1 word</td>
<td>Write 1 word to master output area 4x</td>
</tr>
<tr>
<td>0Fh</td>
<td>Write n bits</td>
<td>Write n bits to master output area 0x</td>
</tr>
<tr>
<td>10h</td>
<td>Write n words</td>
<td>Write n words to master output area 4x</td>
</tr>
</tbody>
</table>

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).

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

<table>
<thead>
<tr>
<th>1 word</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-byte</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address 1. bit</th>
<th>Number of bits</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td>1word</td>
</tr>
</tbody>
</table>

Respond telegram

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Number of read bytes</th>
<th>Data 1. byte</th>
<th>Data 2. byte</th>
<th>...</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1byte</td>
<td>1byte</td>
<td>1byte</td>
<td></td>
<td>1word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>max. 250byte</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Read n words 03h, 04h

03h: Read n words of master output area 4x
04h: Read n words master input area 3x

**Command telegram**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address 1. bit</th>
<th>Number of words</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td>1word</td>
</tr>
</tbody>
</table>

**Respond telegram**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Number of read bytes</th>
<th>Data 1. word</th>
<th>Data 2. word</th>
<th>...</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td></td>
<td>1word</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address bit</th>
<th>Status bit</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td>1word</td>
</tr>
</tbody>
</table>

**Respond telegram**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address bit</th>
<th>Status bit</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td>1word</td>
</tr>
</tbody>
</table>
### Deployment PtP communication

**Modbus - Function codes**

#### Write 1 word 06h

Code 06h: Write 1 word to master output area 4x

**Command telegram**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address word</th>
<th>Value word</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td>1word</td>
</tr>
</tbody>
</table>

**Respond telegram**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address word</th>
<th>Value word</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td>1word</td>
</tr>
</tbody>
</table>

#### 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**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address 1. bit</th>
<th>Number of bits</th>
<th>Number of bytes</th>
<th>Data 1. byte</th>
<th>Data 2. byte</th>
<th>...</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td>1byte</td>
<td>1byte</td>
<td>1byte</td>
<td>1byte</td>
<td>1byte</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>max. 250byte</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Respond telegram**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address 1. bit</th>
<th>Number of bits</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1byte</td>
<td>1byte</td>
<td>1word</td>
<td>1word</td>
<td>1word</td>
</tr>
</tbody>
</table>
**Write n words 10h**  
Code 10h: Write n words to master output area 4x

**Command telegram**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address 1. word</th>
<th>Number of words</th>
<th>Number of bytes</th>
<th>Data 1. word</th>
<th>Data 2. word</th>
<th>...</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>1 word</td>
<td>1 word</td>
<td>1 byte</td>
<td>1 word</td>
<td>1 word</td>
<td>1 word</td>
<td>1 word</td>
</tr>
</tbody>
</table>

max. 125 words

**Respond telegram**

<table>
<thead>
<tr>
<th>Slave address</th>
<th>Function code</th>
<th>Address 1. word</th>
<th>Number of words</th>
<th>Check sum CRC/LRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>1 word</td>
<td>1 word</td>
<td>1 word</td>
</tr>
</tbody>
</table>
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.

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 submodule X1 (MPI/DP) of the Siemens CPU. After the transmission of the data to the CPU, the configuration data are internally passed on to the PROFIBUS master part. During the start-up the DP master automatically includes his data areas into the address range of the CPU. Project engineering in the CPU is not required.

Deployment of the DP master with CPU

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

DP slave operation

For the deployment in a super-ordinated master system you first have to project your slave system as Siemens CPU in slave operation mode with configured in-/output areas. Afterwards you configure your master system. Couple your slave system to your master system by dragging the CPU 31x from the hardware catalog at Configured stations onto the master system, choose your slave system and connect it.
Operating mode DP slave: Test, commissioning, routing (active/passive)

There is the possibility to enable the option ‘Test, commissioning, routing’ in the hardware configuration by means of the properties dialog of the PROFIBUS via the register ‘Operating mode’ at ‘DP slave’. The activation affects as follows:

- The PROFIBUS interface gets an "active" PROFIBUS node, this means it is involved in the token rotation.
- Via this interface you have PG/OP functions (programming, status request, control, test).
- The PROFIBUS interface serves as a gateway (S7 routing).
- The bus rotation time can exceed.

When disabled, the PROFIBUS interface operates as a server for communication services with the following characteristics:

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

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 submodule 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 77

Steps of configuration

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

- Enable bus functionality via VSC
- Hardware configuration - CPU
- Deployment as DP master or DP slave
  - With activating the bus function ‘PROFIBUS DP master’ by means of the VSC, the bus function ‘PROFIBUS DP slave’ is also unlocked.
- Transfer of the complete project to CPU
With the Siemens SIMATIC Manager, the CPU 014 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 77

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

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).

<table>
<thead>
<tr>
<th>Slot</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CPU 315-2PN/DP</td>
</tr>
<tr>
<td>X1</td>
<td>MPI/DP</td>
</tr>
<tr>
<td>X2</td>
<td>PN-IO</td>
</tr>
<tr>
<td>X2...</td>
<td>Port 1</td>
</tr>
<tr>
<td>X2...</td>
<td>Port 2</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Slot</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
| 2    | CPU ...
| X1   | PROFIBUS DP master system |
| X... |        |
| 3    | ...    |

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.

<table>
<thead>
<tr>
<th>Slot</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
| 2    | CPU ...
| X1   | PROFIBUS DP master system |
| X... |        |
| 3    | ...    |
Option: PROFIBUS communication

Deployment as PROFIBUS DP slave

Fast introduction

In the following the deployment of the PROFIBUS section as "intelligent" DP slave on master system is described, which exclusively may be configured in the Siemens SIMATIC Manager. The following steps are required:

1. Configure a station with a CPU with operating mode DP slave.
2. Connect to PROFIBUS and configure the in-/output area for the slave section.
3. Save and compile your project.
4. Configure another station with another CPU with operating mode DP master.
5. Connect to PROFIBUS and configure the in-/output ranges for the master section.
6. Save, compile and transfer your project to your CPU.

Project engineering of the slave section

1. Start the Siemens SIMATIC Manager and configure a CPU as described at "Hardware configuration - CPU".
2. Designate the station as "...DP slave".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at 'MPI/DP'.
5. Set Interface type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 3) and confirm with [OK].
7. Switch at Operating mode to "DP slave".
8. Via Configuration you define the in-/output address area of the slave CPU, which are to be assigned to the DP slave.
9. Save, compile and transfer your project to your CPU.

Slave section

<table>
<thead>
<tr>
<th>Slot</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CPU ...</td>
</tr>
<tr>
<td>X1</td>
<td>MPI/DP</td>
</tr>
<tr>
<td>X</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Modules</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

Object properties

- Operating mode: DP slave
- Connect: PROFIBUS
- PROFIBUS address: > 1
- Configuration:
- Input area
- Output area

Project engineering of the master section

1. Insert another station and configure a CPU.
2. Designate the station as "...DP master".
3. Add your modules according to the real hardware assembly.
4. Open the properties dialog of the DP interface of the CPU by means of a double-click at 'MPI/DP'.
5. Set Interface: type to "PROFIBUS".
6. Connect to PROFIBUS and preset an address (e.g. 2) and confirm with [OK].
7. Switch at Operating mode to "DP master" and confirm the dialog with [OK].
8. Connect your slave system to this master system by dragging the "CPU 31x" from the hardware catalog at Configured stations onto the master system and select your slave system to be coupled.
9. Open the Configuration at Object properties of your slave system.
10. Via double click to the according configuration line you assign the according input address area on the master CPU to the slave output data and the output address area to the slave input data.
11. Save, compile and transfer your project to your CPU.

Master section

<table>
<thead>
<tr>
<th>Slot</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CPU ...</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
</tr>
</tbody>
</table>

Object properties

Operating mode: DP master
Connect: PROFIBUS
PROFIBUS address: > 1

Object properties

Configuration:
Input area slave CPU = Output area master-CPU
Output area slave CPU = Input area master-CPU
6.7 PROFIBUS installation guidelines

**PROFIBUS in general**
- A PROFIBUS DP network may only be built up in linear structure.
- PROFIBUS DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- PROFIBUS supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- The max. segment length depends on the transfer rate:
  - 9.6 \( \ldots \) 187.5bit/s \( \rightarrow \) 1000m
  - 500kbit/s \( \rightarrow \) 400m
  - 1.5Mbit/s \( \rightarrow \) 200m
  - 3 \( \ldots \) 12Mbit/s \( \rightarrow \) 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 resp. 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.

<table>
<thead>
<tr>
<th>Dimensions in mm</th>
<th>0°</th>
<th>45°</th>
<th>90°</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>64</td>
<td>61</td>
<td>66</td>
</tr>
<tr>
<td>B</td>
<td>34</td>
<td>53</td>
<td>40</td>
</tr>
<tr>
<td>C</td>
<td>15.8</td>
<td>15.8</td>
<td>15.8</td>
</tr>
</tbody>
</table>
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!*

---

**6.8 Commissioning and Start-up behavior**

**Start-up on delivery**

In delivery the CPU is overall reset. The PROFIBUS part is deacti-vated and its LEDs are off after Power ON.

**Online with bus parameter without slave project**

The DP master can be served with bus parameters by means of a hardware configuration. As soon as these are transferred the DP master goes online with his bus parameter. This is shown by the RUN LED. Now the DP master can be contacted via PROFIBUS by means of his PROFIBUS address. In this state the CPU can be accessed via PROFIBUS to get configuration and DP slave project.

**Slave configuration**

If the master has received valid configuration data, he switches to *Data Exchange* with the DP slaves. This is indicated by the DE-LED.

**CPU state controls DP master**

After PowerON respectively a receipt of a new hardware configuration the configuration data and bus parameter were transferred to the DP master. Dependent on the CPU state the following behavior is shown by the DP master:

**Master behavior at CPU STOP**

- The global control command "Clear" is sent to the slaves by the master. Here the DE-LED is blinking.
- DP slaves with fail safe mode were provided with output telegram length "0".
- DP slaves without fail safe mode were provided with the whole output telegram but with output data = 0.
- The input data of the DP slaves were further cyclically transferred to the input area of the CPU.
Master behavior at CPU RUN

- The global control command "Operate" is sent to the slaves by the master. Here the DE-LED is on.
- Every connected DP slave is cyclically attended with an output telegram containing recent output data.
- The input data of the DP slaves were cyclically transferred to the input area of the CPU.
7 Configuration with TIA Portal

7.1 TIA Portal - Work environment

7.1.1 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 Totally Integrated Automation 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.

7.1.2 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

7.2 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:

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 PROFINET IO device can be found at Other field devices > PROFINET > 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

In 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

<table>
<thead>
<tr>
<th>Module</th>
<th>Slot</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLC</td>
<td>2</td>
<td>CPU 315-2 PN/DP</td>
</tr>
<tr>
<td>MPI/DP interface</td>
<td>2 X1</td>
<td>MPI/DP interface</td>
</tr>
</tbody>
</table>
**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 58

**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 a valid IP address data in ‘Properties’ at ‘Ethernet address’ in the area ‘IP protocol’.

4. Enter at ‘PROFINET’ a ‘PROFINET device name’. The device name must be unique at the Ethernet subnet.

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 4.9 ‘Setting VIPA specific CPU parameters’ on page 62

7.3 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: ...".

![Ethernet PG/OP](image)

MAC PG/OP: 00-20-D5-77-05-10
Assign IP address
parameters

You get valid IP address parameters from your system administrator. The assignment of the IP address data happens online in the 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.

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 V1.2).
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.

Due to the system you may get a message that the IP address could not be assigned. This message can be ignored.
Device overview

<table>
<thead>
<tr>
<th>Module</th>
<th>Slot</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLC</td>
<td>2</td>
<td>CPU 315-2 PN/DP</td>
</tr>
<tr>
<td>MPI/DP interface</td>
<td>2 X1</td>
<td>MPI/DP interface</td>
</tr>
<tr>
<td>PROFINET interface</td>
<td>2 X2</td>
<td>PROFINET interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP 343-1</td>
<td>4</td>
<td>CP 343-1</td>
</tr>
</tbody>
</table>

7.4 TIA Portal - Hardware configuration - I/O modules

Hardware configuration of the modules

Starting with slot 1 place in the Device overview of the PROFINET IO device ‘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.

### 7.5 TIA Portal - Include VIPA library

**Overview**

- The VIPA specific blocks may be found at 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.

7.6 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 use the according Ethernet interface, you have to assign IP address parameters by means of the "initialization". See Chapter 7.3 'TIA Portal - Hardware configuration - Ethernet PG/OP channel' on page 137.
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.
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.
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.

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". ‘VIPA specific diagnostic entries’ on page 82