# VIPA System 100V

## CPU | Manual

HB100E_CPU | Rev. 12/12

March 2012

VIPA art of automation

VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH
Ohmstraße 4, D-91074 Herzogenaurach, Germany
Tel.: +49 (91 32) 744 -0
Fax.: +49 9132 744 1864
EMail: info@vipa.de
http://www.vipa.de

**Note**

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**

Hereby, VIPA GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions of the following directives:

- 2004/108/EC Electromagnetic Compatibility Directive
- 2006/95/EC Low Voltage Directive

Conformity is indicated by the CE marking affixed to the product.

**Conformity Information**

For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local VIPA customer service organization.

**Trademarks**

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

SPEED7 is a registered trademark of profichip GmbH.

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

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

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

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

**Information product support**

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax:+49 9132 744 1204
EMail: documentation@vipa.de

**Technical support**

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telephone: +49 9132 744 1150 (Hotline)
EMail: support@vipa.de

# Contents

# About this Manual

This manual describes the available System 100V Micro-PLC CPUs from VIPA. Besides of a product overview you will find the detailed description of the CPUs.

You'll get information about installing and operating a Micro-PLC CPU.

**Overview**

**Chapter 1: Basics**

This introduction includes recommendations on the handling of the modules of the VIPA System 100V as central resp. decentral automation system.

Besides a system overview you will find general information to the System 100V like dimensions, installation and operating conditions.

**Chapter 2: Hardware description Micro-PLC CPU 11x**

The Micro-PLC CPU 11x is available in different variants that will be described in this chapter.

Here you will find information about the structure, connection diagrams, working method and technical data.

**Chapter 3: Deployment Micro-PLC CPU 11x**

This chapter includes all information required for the deployment of the Micro-PLC CPU 11x, from the project engineering to the commissioning.

**Chapter 4: Deployment Micro-PLC CPU 11xDP**

Content of this chapter is the deployment of the Micro-PLC CPU 11xDP under PROFIBUS. You will get all information required for the deployment of an intelligent PROFIBUS-DP slave.

**Chapter 5: Deployment Micro-PLC CPU 11xSER**

Content of this chapter is the deployment of the Micro-PLC CPU with serial interface. After an introduction to protocols and procedures connection and project engineering are descript.

**Objective and contents**

This manual describes the installation, project engineering and usage of the Micro-PLC CPU 11x of the System 100V.

This manual is relevant for:

| Product | Order number | as of state: HW | CPU-FW |
|---|---|---|---|
| CPU 11x | VIPA 11x | 01 | V412 |

**Target audience**

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

**Structure of the manual**

This manual consists of chapters. Every chapter provides the description of one specific topic.

**Guide to the document**

This manual provides the following guides:

- An overall table of contents at the beginning of the manual
- An overview of the topics for every chapter
- An index at the end of the manual.

**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.

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

**Note!**
Supplementary information and useful tips.

# Safety information

**Application specifications**

The System 100V is constructed and manufactured for

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

**Danger!**

The System 100V is not certified for applications in

- explosive environments (EX-zone)

**Documentation**

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

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

- Hardware modification 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!**

# Chapter 1 Basics

**Overview**

Main theme of this chapter is to give you information and hints about deployment areas and usage of the System 100V.

**Content**

# Safety information for Users

**Handling of electrostatic sensitive modules**

VIPA modules make use of highly integrated components in MOS-technology. These components are extremely sensitive to over-voltages that can occur during electrostatic discharges.

The following symbol is attached to modules that can be destroyed by electrostatic discharges:

The symbol is located on the module, the module rack or on packing material and it indicates the presence of electrostatic sensitive equipment.

It is possible that electrostatic sensitive equipment is destroyed by energies and voltages that are far less than the human threshold of perception. These voltages can occur where persons do not discharge themselves before handling electrostatic sensitive modules and they can damage components thereby, causing the module to become inoperable or unusable. Modules that have been damaged by electrostatic discharges may 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 electrostatic sensitive modules**

Modules have to 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.

**Attention!**

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

# Overview System 100V

**General**

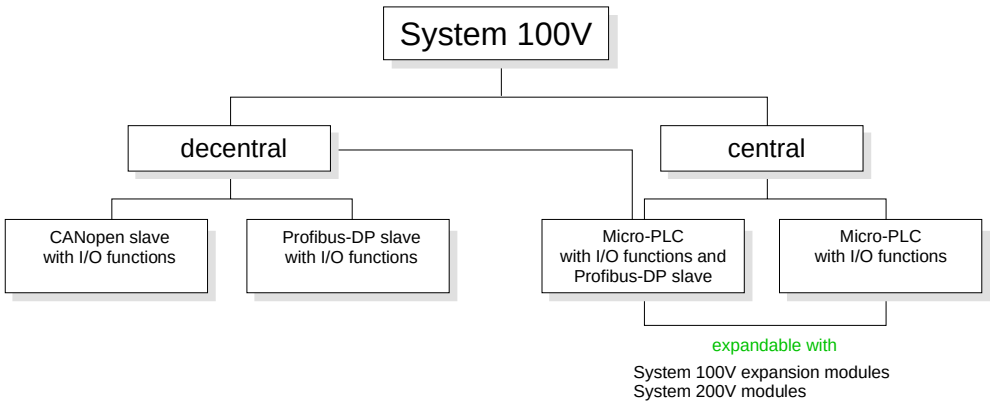
The System 100V from VIPA is a compact central and decentral usable automation system from VIPA. The system is recommended for lower and middle performance needs.

At a System 100V module, CPU res. bus coupler are integrated together with in-/output functions in one case.

System 100V modules are installed directly to a 35mm norm profile rail.

You may expand the number of I/Os of the Micro-PLC by means of expansion modules res. connect System 200V modules via bus couplers.

The following picture shows the performance range of the System 100V:




**Central system**

The central system is built of one CPU and integrated I/O-functions. The CPU is instruction compatible to the S7-300 from Siemens and may be programmed and projected by means of S7 programming tools from Siemens and VIPA via MPI.

By means of bus couplers you may connect modules of the System 200V family res. enlarge the number of I/Os by installing System 100V expansion modules.

The CPUs are available in different variants.

**Central system with DP slave**

At the central system besides the CPU and I/O functions, a PROFIBUS-DP slave is included that acknowledges itself within the address range of the CPU.

**Decentral system**

This system contains a PROFIBUS-DP res. CANopen slave with I/O functions instead of the CPU. The system is not expandable.

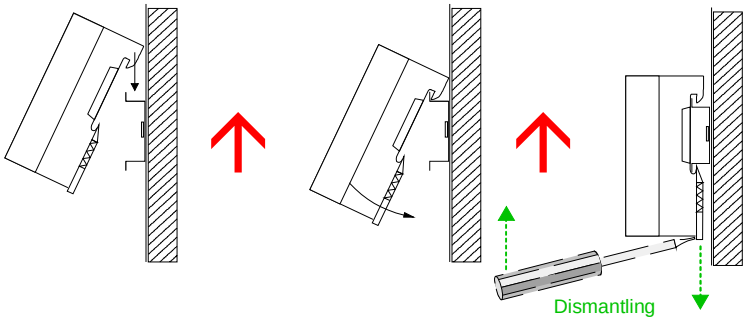
# General Description of the System 100V

**Structure and dimensions**

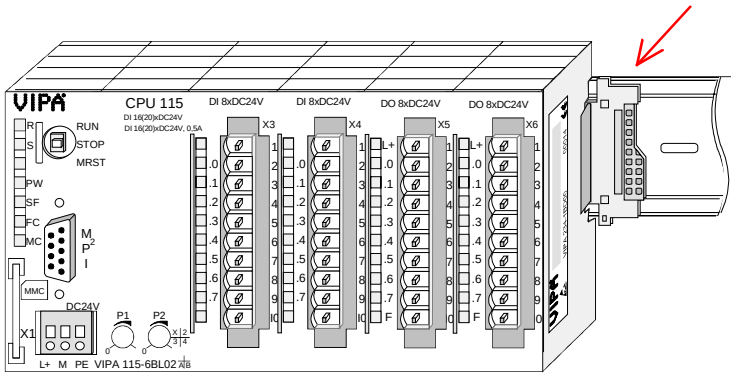
- Norm profile head rail 35mm
- Dimensions basic module:
  4tier width: (WxHxD) in mm: 101.6x76x48 / in inches: 4x3x1.9
  6tier width: (WxHxD) in mm: 152.4x76x48 / in Inches: 6x3x1.9

**Installation**

The installation of a System 100V module works via snapping on a norm profile head rail.




When using expansion modules, you have to clip the included 1tier bus connector at the right side to the module from behind before the installation.




**Operation security**

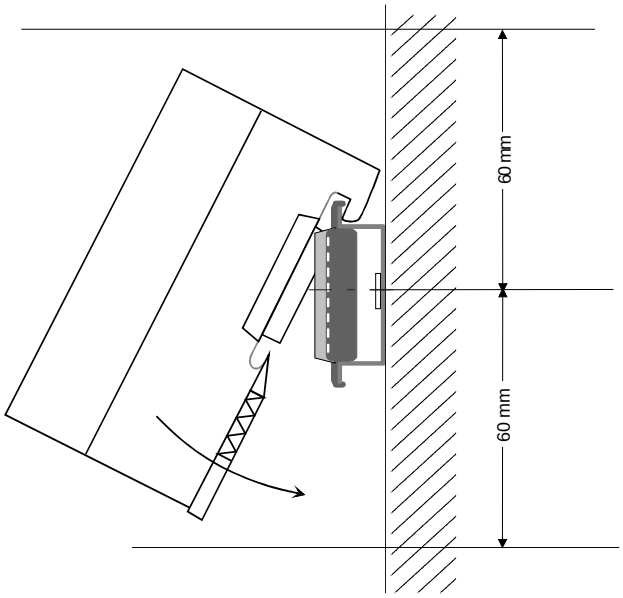
- Plug in via CageClamps, core cross-section 0.08...2.5mm$^2$
- Total isolation of the cables during module changes
- EMV resistance ESD/Burst acc. IEC 61000-4-2 / IEC 61000-4-4 (to level 3)
- Shock resistance acc. IEC 60068-2-6 / IEC 60068-2-27 (1G/12G)

**Environmental conditions**

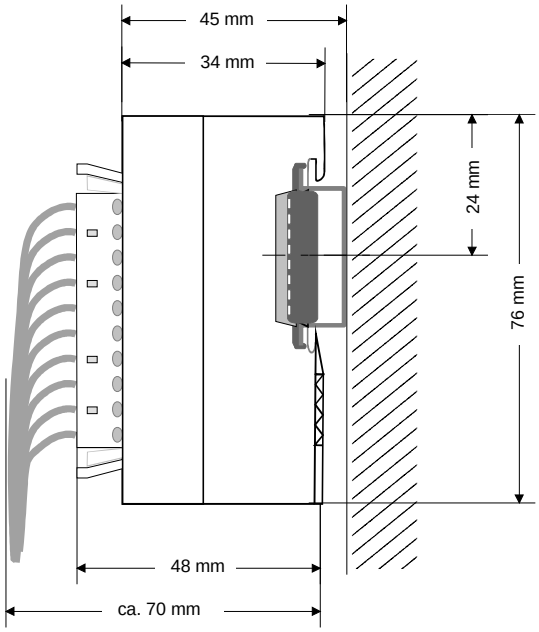
- Operating temperature: 0... + 60°C
- Storage temperature: -25... + 70°C
- Relative humidity: 5 ... 95% without condensation
- fan-less operation

# Assembly dimensions

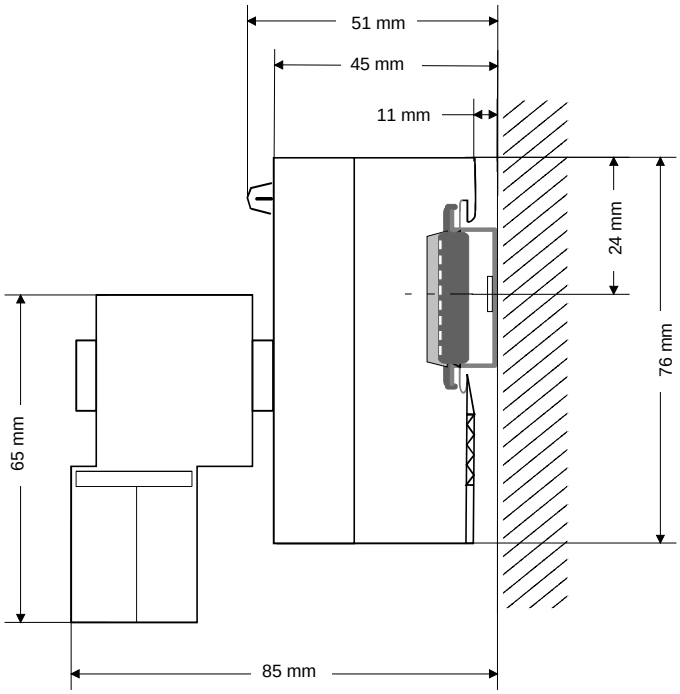
**Installation dimensions**




**Installed and wired dimensions**



**CPU 11x with EasyConn from VIPA**



# Installation Guidelines

**General**

The installation guidelines contain information about the interference free deployment of System 100V systems. There is the description of the ways, interference may occur in your control, how you can make sure the electromagnetic digestibility (EMC), and how you manage the isolation.

**What means EMC?**

Electromagnetic digestibility (EMC) means the ability of an electrical device, to function error free in an electromagnetic environment without being interferenced res. without interferencing the environment.

All System 100V components are developed for the deployment in hard industrial environments and fulfill 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:

- Fields
- I/O signal conductors
- Bus system
- Current 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.

One differs:

- 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 aluminum parts. Aluminum 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 res. 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 favorable.
  - Lay the line isolation extensively on a 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 metalized plug cases for isolated data lines.
- In special use cases you should appoint special EMC actions.
  - Wire all inductivities with erase links that are not addressed by the System 100V modules.
  - -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 is a protection and functionality activity.
  - Connect installation parts and cabinets with the System 100V in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If potential differences between installation parts and cabinets occur, lay sufficiently dimensioned potential compensation lines.

**Isolation of conductors**

Electrical, magnetic 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. Hereby 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 a 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 res. µA) are transferred
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metalized 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 de-isolate the isolated 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 the System 100V module and **don't** lay it on there again!

**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

# Chapter 2 Hardware description Micro-PLC CPU 11x

**Overview**

The Micro-PLC CPU 11x is available in different variants that will be described in this chapter.

Further on you'll get some suggestions to the programming and the according code data of the CPUs.

**Content**

# System overview

**General**

With a Micro-PLC CPU 11x you always have a closed system with CPU and input/output modules.

The CPUs have a $MP^2I$ interface and support the standard MPI protocol and a serial point-to-point communication.

Thus enables, together with the "Green Cable" from VIPA, a direct and economic programming.

The modules are clipped directly at a 35mm norm profile rail.

The CPU 11x has an integrated power supply that has to be provided with DC 24V via the front-side. The power supply is protected against polarity inversion and overcurrent.

The CPU 11x has Counter-, Alarm- and Pulse- output functions, interfaces for expansions modules and 2 analog potentiometers depending of type from CPU 11x.


# Security hints for deployment of DIO channels

**Attention!**

Please regard that the voltage applied to an output channel must be ≤ the voltage supply applied to L+.

Due to the parallel connection of in- and output channel per group, a set output channel may be supplied via an applied input signal.

Thus, a set output remains active even at power-off of the voltage supply with the applied input signal.

Non-observance may cause module demolition.

**Micro-PLC**

The Micro-PLC of the System 100V is especially suitable for the deployment at controls with a low amount of in-/outputs, where you abstained deploying a CPU in the past.
The following System 100V Micro-PLCs are available:

|  | Module width | Number of inputs DC 24V | Number of outputs DC 24V, 0.5A | Number of Relay outputs DC 30V/AC 230V, 5A | Input data | Output data | Alarminputs/ Counter max. | Pulse output | RS232/485 interface | PROFIBUS slave integrated | Work-/Load memory | Current consumption |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Micro-PLC Digital I/O | | | | | | | | | | | | |
| 112-4BH02 | 4tier | 8(12) | 8(4) | - | 3Byte | 3Byte | 4/- | - | - | - | 8/16kB | 50mA |
| 114-6BJ02 | 6tier | 16(20) | 8(4) | - | 3Byte | 3Byte | 4/4 | 2 | - | - | 16/24kB | 80mA |
| 114-6BJ03 | 6tier | 16(20) | 8(4) | - | 3Byte | 3Byte | 4/4 | 2 | - | - | 24/32kB | 80mA |
| 114-6BJ04 | 6tier | 16(20) | 8(4) | - | 3Byte | 3Byte | 4/4 | 2 | - | - | 32/40kB | 80mA |
| 114-6BJ52 | 6tier | 16 | | 8 | 3Byte | 3Byte | 4/4 | - | - | - | 16/24kB | 150mA |
| 114-6BJ53 | 6tier | 16 | | 8 | 3Byte | 3Byte | 4/4 | - | - | - | 24/32kB | 150mA |
| 114-6BJ54 | 6tier | 16 | | 8 | 3Byte | 3Byte | 4/4 | - | - | - | 32/40kB | 150mA |
| 115-6BL02 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | - | - | 16/24kB | 90mA |
| 115-6BL03 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | - | - | 24/32kB | 90mA |
| 115-6BL04 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | - | - | 32/40kB | 90mA |
| 115-6BL12 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | 232 | - | 16/24kB | 100mA |
| 115-6BL13 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | 232 | - | 24/32kB | 100mA |
| 115-6BL14 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | 232 | - | 32/40kB | 100mA |
| 115-6BL22 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | - | ja | 16/24kB | 160mA |
| 115-6BL23 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | - | ja | 24/32kB | 160mA |
| 115-6BL24 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | - | ja | 32/40kB | 160mA |
| 115-6BL32 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | 485 | - | 16/24kB | 110mA |
| 115-6BL33 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | 485 | - | 24/32kB | 110mA |
| 115-6BL34 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | 485 | - | 32/40kB | 110mA |
| 115-6BL72 | 6tier | 16(20) | 16(12) | - | 3Byte | 3Byte | 4/4 | 2 | - | - | 16/24kB | 90mA |

CPU 112

- CPU with in-/output components
- Isolation each I/O group respectively I/O periphery
- Instruction set compatible to S7-300 from Siemens
- MP$^2$I adapter for the data transfer between PC and CPU res. between different MPI participants
- MMC storage module external
- Real-time clock

CPU 114
CPU 115

like CPU 112 additionally

- Interface for expansion modules
- Max. 4 inputs parameterizable as high speed counter (max. 30kHz) or alarm inputs
- Max. 2 outputs parameterizable as pulse outputs with standard PWM or high-frequency PWM to max. 50kHz (not CPU 114-6BJ5x)
- Analog potentiometer (2)

CPU 115DP

Like CPU 115 additionally with integrated PROFIBUS-DP slave

CPU 115SER

Like CPU 115 additionally

CPU 115-6BL1x with RS232 interface
CPU 115-6BL3x with RS485 interface

**Expansion modules**

For expanding your Micro-PLC you may connect up to 4 expansion modules. You may also connect up to 4 modules of the System 200V family. A combination of expansion and System 200V modules, which results to the sum 4 is likewise possible.

At the Micro-SPS CPU with order-no. 115-6BL72 maximum 7 modules may be connected.

Please consider the maximum current of the expansion slot may amount to maximally 0.9A!

More information about the expansion modules may be found in the manual HB100_EM.

**General**

A CPU is an intelligent module. Here your control applications are processed. Depending on your performance needs, you may choose between three CPU variants.

These CPUs 11x are recommended for small and middle range applications with integrated 24V power supply. The CPUs contain a standard processor with internal program memory to store the application program. Additionally every CPU 11x has a slot for a storage module at the front-side.

Every CPU has a MPI interface and is instruction compatible to S7-300 from Siemens.

By connecting up to 4 expansion modules (max 7 modules at VIPA 115-6BL72) you may increase the number of your in- and outputs. Due to the fact that the System 100V and 200V are using identical backplane bus connectors, you may also connect up to 4 (7) modules of the System 200V family.

With the CPU series you have access to the peripheral modules of the System 200V. You may request sensors and control actors via standardized commands and programs.

Via the integrated MPI interface you are able to project your CPU.

**The further description in this chapter refers to the CPU family CPU 11x.**

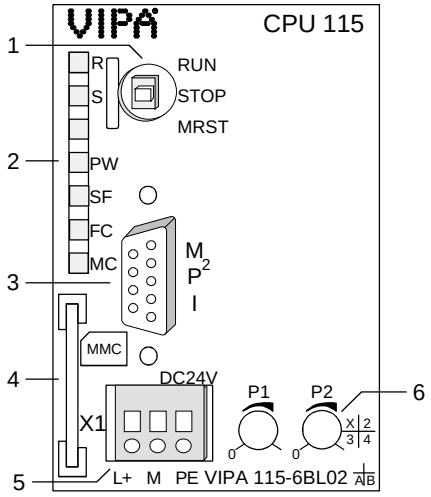
**Properties**

- Instruction compatible to S7-300 from Siemens
- Project engineering via the Siemens SIMATIC manager
- Integrated 24V power supply
- Isolation per I/O-column respectively -periphery
- work / load memory: 16/24 kByte (8/16 kByte only CPU 112)
  11x-xxxx3 work / load memory: 24/32 kByte
  11x-xxxx4 work / load memory: 32/40 kByte
- Max. 4 inputs parameterizable as high speed counter[1)] (max. 30kHz) or alarm inputs
- Max. 2 outputs parameterizable as pulse outputs[1) 2)] with standard PWM or high-frequency PWM to max. 50kHz
- 2 analog potentiometer[1)] for presetting analog values
- PROFIBUS-DP slave at CPU11xDP integrated
- Internal Flash-ROM
- battery buffered real-time clock
- Slot for memory card
- MPI interface
- Integrated VBUS-Controller for controlling the System 100V and 200V peripheral modules
- 256 timers
- 256 counters
- 8192 Bits marker

[1)] not CPU 112 (112-4BH02)
[2)] not CPU 114 (114-6BJ5x)

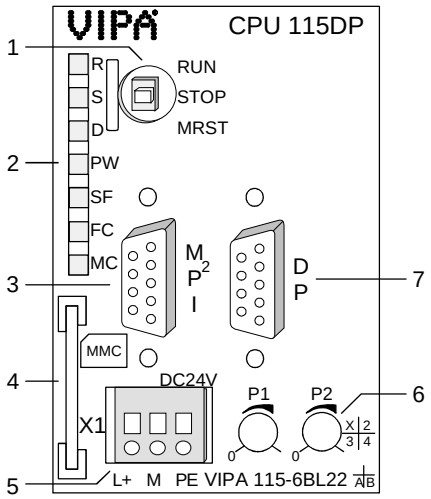
# Structure CPU 11x

**Front view CPU 11x**



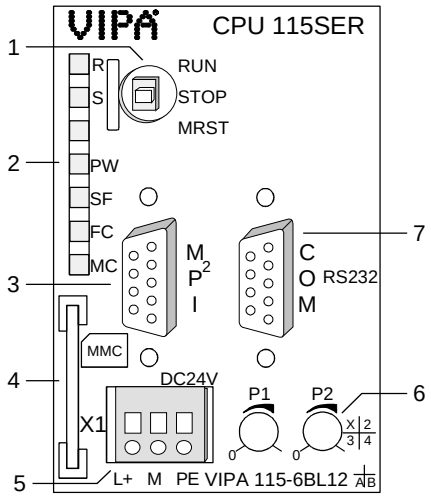

[1] Operating mode switch RUN/STOP/RESET
[2] Diagnostic LEDs
[3] MP²I interface
[4] Slot for MMC storage module
[5] Connection for DC 24V power supply
[6] 2 Analog potentiometer (not CPU 112)

**Front view CPU 11xDP**




[1] Operating mode switch RUN/STOP/RESET
[2] Diagnostic LEDs
[3] MP²I interface
[4] Slot for MMC storage module
[5] Connection for DC 24V power supply
[6] 2 Analog potentiometer
[7] PROFIBUS-DP slave interface

**Front view CPU 11xSER**




[1] Operating mode switch RUN/STOP/RESET
[2] Diagnostic LEDs
[3] MP²I interface
[4] Slot for MMC storage module
[5] Connection for DC 24V power supply
[6] 2 Analog potentiometer
[7] VIPA 115-6BL1x: RS232
VIPA 115-6BL3x: RS485

# Components

**CPU 11x**

The components for the CPU 11x that are described here are also part of all CPUs portrayed in this manual except the CPU 112. The CPU 112 has no counter input nor pulse outputs. The CPU 112 is not expandable with modules. All CPUs have alarm inputs.

**LEDs**

The CPUs 11x have different LEDs for bus diagnosis and program state monitoring. The usage and the colors of the diagnostic LEDs are to find in the following table. These LEDs are part of every CPU in this manual.

| Label | Color | Description |
|---|---|---|
| R | green | CPU is in the operating mode RUN. |
| S | yellow | CPU is in the operating mode STOP. |
| D | green | **only CPU 11xDP**<br>D (Data exchange) indicates PROFIBUS communication activity. |
| PW | green | Signalizes the started CPU. |
| SF | red | Blinks at system errors (hardware defect) |
| FC | yellow | Blinks, if variables are forced (fixed). |
| MC | yellow | Blinking shows accesses at the MMC. |

**Power supply**

The CPU contains an integrated power supply. The connection is via 3 connection clamps at the front-side.

The power supply has to be provided with DC 24V. By means of the supply voltage the electronic parts of the CPU as well as the connected modules are provided via the backplane bus.

The CPU electronics are not isolated from the supply voltage. The power supply is protected against polarity inversion and overcurrent.

**Note!**

Please take care of the correct polarity at the power supply.

**Operating mode switch RN/STOP/MRST**

With the operating mode switch you may choose between the operating modes STOP and RUN. The operating mode START-UP is processed automatically by the CPU between STOP and RUN.

By means of the switch location Memory Reset (MRST) you request an overall reset.

**MMC slot storage module**

As external storage medium you may plug in a MMC storage module from VIPA (Order-No.: VIPA 953-0KX10). Access to the MMC always takes place after an overall reset.

Also available at VIPA is an external MMC reading device (Order-No: VIPA 950-0AD00). This allows you to write onto res. read your MMC at the PC.

The MMCs are delivered preformatted with the FAT16 file system.

This allows you to create programs at the PC, copy them to the MMC and transfer them into the VIPA CPU by plugging-in the MMC.

By means of the MMC you may easily execute a firmware update of your System 100V.

More detailed information is in the chapter "Deployment of the CPU 11x".

**Battery buffer for clock and RAM**

Every CPU 11x has an internal accu for protecting the RAM at black-out. Additionally the internal real-time clock is buffered via the accu.

The accu is loaded directly via the integrated power supply by means of special loading electronics and guarantees a buffer for max. 30 days.

**Attention!**

That the CPU is able to switch to RUN, the accu has to be in good condition.

If there is a defect at the accu, the CPU switches to STOP and announces a sum error. In this case you should check the CPU. Please contact VIPA for that purpose!

**Internal Flash-ROM**

Additional to the battery buffered RAM, the CPU 11x has an internal Flash-ROM in the size of the load memory.

Via the writing command **PLC** > *Copy RAM to ROM* from the destination system functions of the hardware configurator from Siemens, the contents of the load memory are transferred into the Flash-ROM and simultaneously to the MMC, if plugged in.

The CPU only accesses the contents of the Flash-ROM if the battery buffered RAM is empty.

The Flash-ROM is not deleted by an OVERALL RESET. The Flash-ROM may be cleared by means of requesting an OVERALL RESET and then transferring the now empty load memory into the Flash-ROM via the PLC function *Copy RAM to ROM*.

**Note!**

Please regard, that an error message occurs, when you initiate a write command and there is no MMC plugged in.

Nevertheless the data is saved in the internal Flash-ROM.

**MP²I interface**

The MP²I interface provides the data transfer between CPUs and PCs. The MP²I jack combines 2 interfaces in 1:

- *MP interface*
  During a bus communication you may transmit applications and data between the CPUs that are connected with each other via MPI.
- *RS232 interface*
  Serial data transfer by means of Green Cable from VIPA.

**Important notes for the deployment of MPI cables!**

Deploying MPI cables at the CPUs from VIPA, you have to make sure that Pin 1 is not connected. This may cause transfer problems and in some cases damage the CPU!

Especially PROFIBUS cables from Siemens, like e.g. the 6XV1 830-1CH30, must not be deployed at MP²I jack.

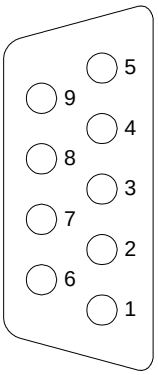
For damages caused by nonobservance of these notes and at improper deployment, VIPA does not take liability!

For a serial transmission from your PC, you normally need a MPI transducer. Instead of this you may also use the VIPA "Green Cable" (order-no. VIPA 950-0KB00).

The Green Cable is a green connection cable, manufactured exclusively for the deployment at VIPA System components.

It is a programming and download cable for VIPA CPUs with MP²I jack and VIPA field bus masters.

The MP²I jack has the following pin assignment:




*9pin jack*

| Pin | Assignment |
|---|---|
| 1 | reserved (must not be connected) |
| 2 | M24V |
| 3 | RxD/TxD-P (Line B) |
| 4 | RTS |
| 5 | M5V |
| 6 | P5V |
| 7 | P24V |
| 8 | RxD/TxD-N (Line A) |
| 9 | n.c. |

**MPI connection PC - CPU via Green Cable**
MPI addr.=2 (default)

For deployment of the Green Cable together with the MP$^2$I jack, you have to assign a COM port to the interface. Execute the following steps:

- Start the SIMATIC manager from Siemens.
- Open the dialog for the MPI adapter via **Options** > *PG/PC interface* and choose "PC adapter (MPI)" from the list.
- Click on [Properties...] to open another window with different register cards.
- The default settings of the "MPI" options are recommended. Please regard that [Standard] has also an influence on the settings at "Local connection".
- At "Local connection" you choose the COM port and set, for the communication via MP$^2$I, the transfer rate at 38400bps. Close both windows with [OK].

To test the connection, plug the VIPA Green Cable to the COM interface of your PC and to the MP$^2$I jack of your CPU.

Via **PLC** > *Display Accessible Nodes* you reach the CPU with the preset MPI address 2.

**Important notes for the deployment of the Green Cable**

Nonobservance of the following notes may cause damages on system components.

For damages caused by nonobservance of the following notes and at improper deployment, VIPA does not take liability!

**Note to the application area**

The Green Cable may exclusively deployed directly at the concerning jacks of the VIPA components (in between plugs are not permitted). E.g. a MPI cable has to be disconnected if you want to connect a Green Cable.

At this time, the following components support the Green Cable:

VIPA CPUs with MP$^2$I jack and field bus master from VIPA.

**Note to the lengthening**

The lengthening of the Green Cable with another Green Cable res. The combination with further MPI cables is not permitted and causes damages of the connected components!

The Green Cable may only be lengthened with a 1:1 cable (all 9 Pins are connected 1:1).

**Counter / alarm inputs, pulse outputs**

The first 4 inputs of X3 may be used as counter or as alarm input, the last 2 outputs of the output area X5 may be used as pulse outputs *).

The properties and the behavior of the in- res. outputs are defined via the hardware configurator at the CPU parameters.

These functions are deactivated in delivery state.

- Alarm input

  The function "alarm input" means that an alarm is initialized after a selectable delay time and edge evaluation.

- Counter input

  The setting "Counter" allows you to control up to 4 counters with a frequency of up to 30kHz via the 4 inputs. An alarm output at limit value overrun is parameterizable.

  The following counter modes are available:

  *Pulses*

  Occupies 1 input and counts in the parameterized direction with every pulse (max. 4 counter).

  *Pulse with direction*

  Occupies 2 inputs and counts with every pulse in the direction given by a second input (max. 2 counter)

  *Pulse with hardware gate*

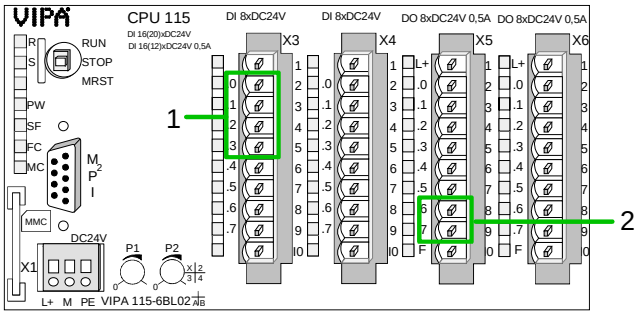
  Occupies 2 inputs with input 1 as counter and input 2 as release.

  *Rotary encoder single, double, quadruple*

  Occupies 2 inputs for every rotary encoder, i.e. a max. of 2 encoders.

- Pulse output *)

  The last two outputs of X5 may be parameterized as pulse width modulated (PWM) output with a max. frequency of up to 50kHz. Via parameterization of time presets, the CPU calculates a pulse sequence with according pulse/pause ratio.




[1] Counter or alarm inputs

[2] Pulse outputs

**Note!**

A more detailed description and the parameterization of these functions is to find in the chapter "Deployment of the Micro-PLC CPU 11x".

*) not CPU 112 (112-4BH02) and CPU 114 (114-6BJ5x)

**Default address allocation CPU 11x**

If no hardware configuration had taken place yet, the following addresses in the CPU 11x are occupied:

| Address allocation input area | |
|---|---|
| 0...2 | DI |
| 3...127 | **free for more inputs** |
| 128, 129 | Potentiometer P1 |
| 130, 131 | Potentiometer P2 |
| 132...135 | reserved |
| 136...139 | Counter 0 |
| 140...143 | Counter 1 |
| 144...147 | Counter 2 |
| 148...151 | Counter 3 |
| 152...1021 | **free for more inputs** |
| 1022 | reserved |
| Address allocation output area | |
| 0...2 | DO |
| 3...1021 | **free for more outputs** |
| 1022 | reserved |

**Potentiometer**

At the front-side, there are 2 potentiometer (not at CPU 112) for the direct input of analog values.

The potentiometers occupy 1 input word each. Per default, the potentiometer are at the following addresses: P1: 128, P2: 130.

The address allocation for the potentiometer takes place via your hardware configuration in the CPU parameters.

You may parameterize values between 0h and 03FFh.

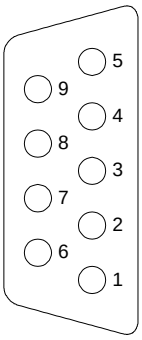
## CPU 11xDP

Additional to the components described before, the CPU 11xDP has a PROFIBUS interface.

### PROFIBUS-DP interface

Via a 9pin RS485 interface you include your Micro-PLC CPU 11xDP into your PROFIBUS.

The pin assignment is as follows:




*9pin jack*

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

### LED

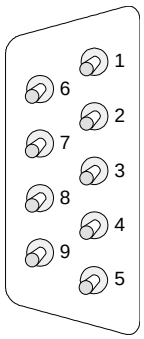
The CPU 11xDP has additionally a "D"-LED (Data exchange) that indicates data exchange via the PROFIBUS-DP interface.

## CPU 11xSER

Additional to the components described before, the CPU 115-6BL1x has an RS232 interface the CPU 115-6BL3x an RS485 interface.

### RS232 interface

Via 9pin jack, you may establish a serial point-to-point connection.

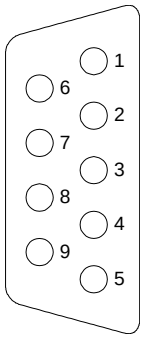



*9pin Plug* (*CPU 115-6BL1x)*

| Pin | RS232 |
|---|---|
| 1 | CD- |
| 2 | RxD |
| 3 | TxD |
| 4 | DTR- |
| 5 | GND |
| 6 | DSR- |
| 7 | RTS- |
| 8 | CTS- |
| 9 | RI- |

### RS485 interface

Via 9pin slot, you may establish a serial point-to-point connection.




*9pin Slot* (*CPU 115-6BL3x)*

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

# Structure of the in-/outputs

## Input section

The digital input section of a System 100V module collects the binary control signals of the process level and stores them in a definable address area of the CPU.

Each input channel occupies 1 Bit and shows its state via a green LED.

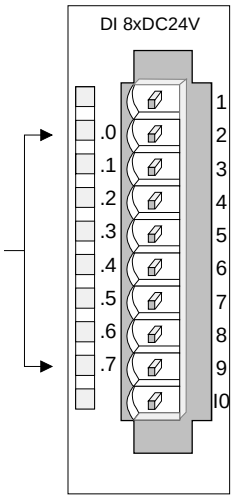
The nominal input voltage is DC 24V. Hereby 0 ... 5V mean the signal state "0" and 15 ... 28.8V the signal state "1".

The input and output areas are always occupying 3Byte input and 3Byte output data in the CPU.

Like mentioned above, you may assign counter res. alarm properties to the first 4 input channels of the first input row. The assignment takes place via the hardware configurator at the CPU parameters. More to that topic is to find in the chapter "Deployment of the Micro-PLC CPU 11x".

### Status monitor pin assignment

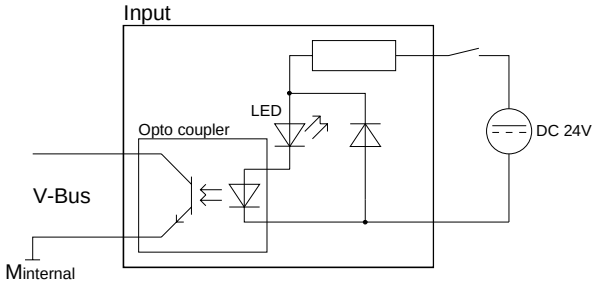
| LED | Description |
|---|---|
| .0... .7 | LEDs (green)<br>I+0.0 to I+0.7<br>from ca. 15V on the signal "1" is recognized and the according LED is addressed |




| Pin | Assignment |
|---|---|
| 1 | not used |
| 2 | Input I+0.0* |
| 3 | Input I+0.1* |
| 4 | Input I+0.2* |
| 5 | Input I+0.3* |
| 6 | Input I+0.4 |
| 7 | Input I+0.5 |
| 8 | Input I+0.6 |
| 9 | Input I+0.7 |
| 10 | Ground |

*) At X3 parameterizable as counter res. alarm input.

### Schematic diagram input section



## Output section

The output section has to be additionally provided with DC 24V via the front-facing connector (see also schematic diagrams). The available supply voltage is shown via the yellow LED (L+).

Every digital output channel shows its state via a green LED. At activated output, the according LED is on.

If an overload, overheat or short circuit occurs, the error-LED, marked with "F", is blinking red. Each channel is loadable with max 0.5A.

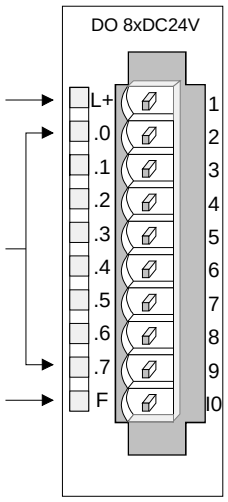
The input and output areas are always occupying 3Byte input and 3Byte output data in the CPU.

Like mentioned above, you may assign pulse functions to the last two output channels at X5.

The assignment takes place via the hardware configurator at the CPU parameters. More to that topic is to find in the chapter "Deployment of the Micro-PLC CPU 11x".

### Status monitor pin assignment

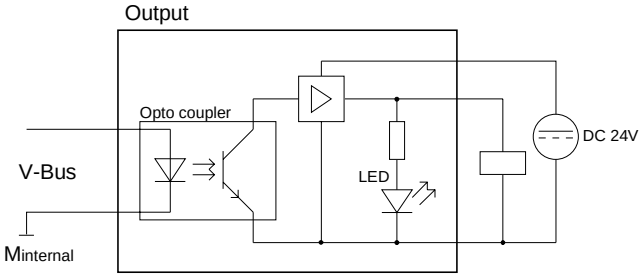
| LED | Description |
|---|---|
| L+ | LED (green)<br>Supply voltage is available |
| .0... .7 | LEDs (green)<br>Q+0.0 to Q+0.7<br>as soon as an output is active, the according LED is addressed |
| F | LED (red)<br>Error at overload, overheat or short circuits. |




| Pin | Assignment |
|---|---|
| 1 | Supply voltage DC 24V |
| 2 | Output Q+0.0 |
| 3 | Output Q+0.1 |
| 4 | Output Q+0.2 |
| 5 | Output Q+0.3 |
| 6 | Output Q+0.4 |
| 7 | Output Q+0.5 |
| 8 | Output Q+0.6* |
| 9 | Output Q+0.7* |
| 10 | Supply voltage ground |

*) At X5 parameterizable as pulse output with max. output current of 0.5A per channel.

### Schematic diagram output section



## In-/Output section

The In-/Output section has 4 I/O channels that may be used as input or as output channels and 4 normal outputs. Every I/O channel is provided with a diagnostic function, i.e. when an output is active the respective input is set to "1".

The In-/output section has to be additionally provided with DC 24V via the front-facing connector (see also schematic diagrams). The available supply voltage is shown via the green LED (L+).

The input and output areas are always occupying 3Byte input and 3Byte output data in the CPU.

When a short circuit occurs at the load, the input is held at "0" and the error is detectable by analyzing the input.

If an overload, overheat or short circuit occurs, the error-LED, marked with "F", is blinking red. Each channel is loadable with max 0.5A.

Like mentioned above, you may assign pulse functions to the last two output channels at X5.
The assignment takes place via the hardware configurator at the CPU parameters. More to that topic is to find in the chapter "Deployment of the Micro-PLC CPU 11x".

**Attention!**

Please regard that the voltage applied to an output channel must be ≤ the voltage supply applied to L+.

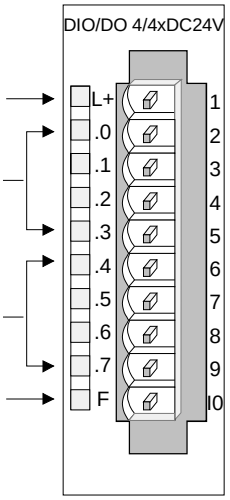
Due to the parallel connection of in- and output channel per group, a set output channel may be supplied via an applied input signal.

Thus, a set output remains active even at power-off of the voltage supply with the applied input signal.

Non-observance may cause module demolition.

### Status monitor pin assignment

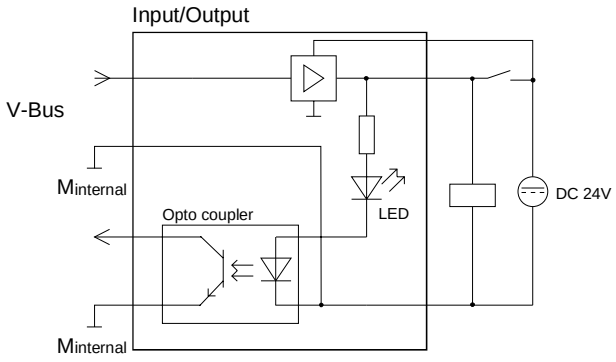
| LED | Description |
|---|---|
| L+ | LED (green)<br>Supply voltage is available |
| .0... .3 | LEDs (green)<br>I/Q+0.0 to I/Q+0.3<br>as soon as an I/O=1 the according LED is addressed |
| .4... .7 | LEDs (green)<br>Q+0.4 to Q+0.7<br>as soon as an output is active, the according LED is addressed |
| F | LED (red)<br>Error at overload, overheat or short circuits. |




| Pin | Assignment |
|---|---|
| 1 | Supply voltage DC 24V |
| 2 | In-/Output I/Q+0.0 |
| 3 | In-/Output I/Q+0.1 |
| 4 | In-/Output I/Q+0.2 |
| 5 | In-/Output I/Q+0.3 |
| 6 | Output Q+0.4 |
| 7 | Output Q+0.5 |
| 8 | Output Q+0.6* |
| 9 | Output Q+0.7* |
| 10 | Supply voltage ground |

*) At X5 parameterizable as pulse output with max. output current of 0.5A per channel.

### Schematic diagram output section

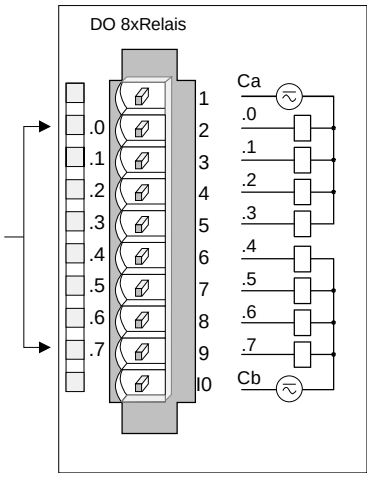


## Relay output

The relay output is segmented in 2 groups with 4 relays. A LED for errors and applied load voltage is not available. The relay output unit is not processing diagnosis.

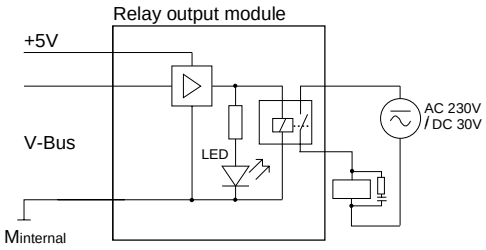
### Status monitor pin assignment

| LED | Description |
|---|---|
| .0... .7 | LED (green)<br>Q+0.0 to Q+0.7<br>as soon as an output is active, the according LED is addressed |



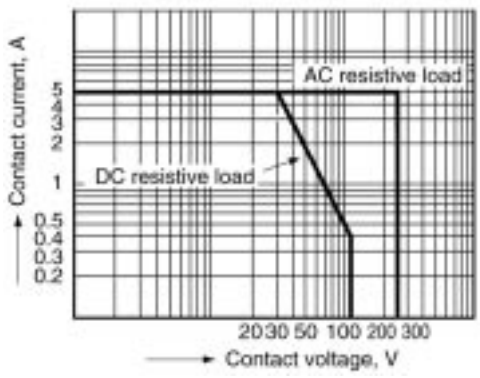

| Pin | Assignment |
|---|---|
| 1 | Supply voltage Ca |
| 2 | Relay output Q+0.0 |
| 3 | Relay output Q+0.1 |
| 4 | Relay output Q+0.2 |
| 5 | Relay output Q+0.3 |
| 6 | Relay output Q+0.4 |
| 7 | Relay output Q+0.5 |
| 8 | Relay output Q+0.6 |
| 9 | Relay output Q+0.7 |
| 10 | Supply voltage Cb |

### Schematic diagram relay output

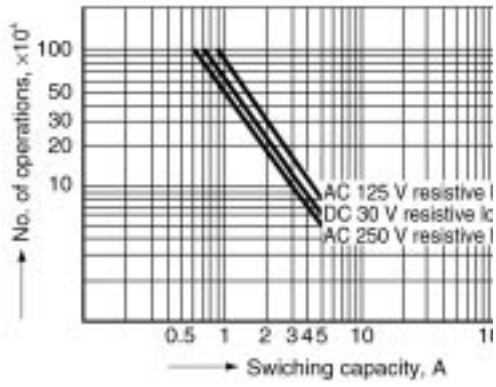



**Note:** When using inductive load please take a suitable protector (i.e. RC-combination).

### Max. toggle performance




### Life time



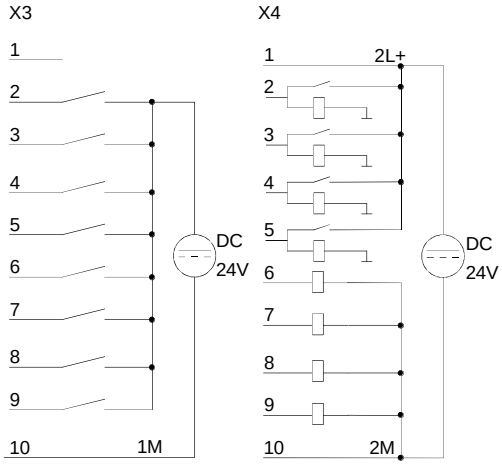
# Circuit diagrams

**Micro-PLC CPU 112**

VIPA 112-4BH02:
DI 8(12)xDC 24V / DO 8(4)xDC 24V 0.5A

DI
IB0

DIO
IB1
OB0



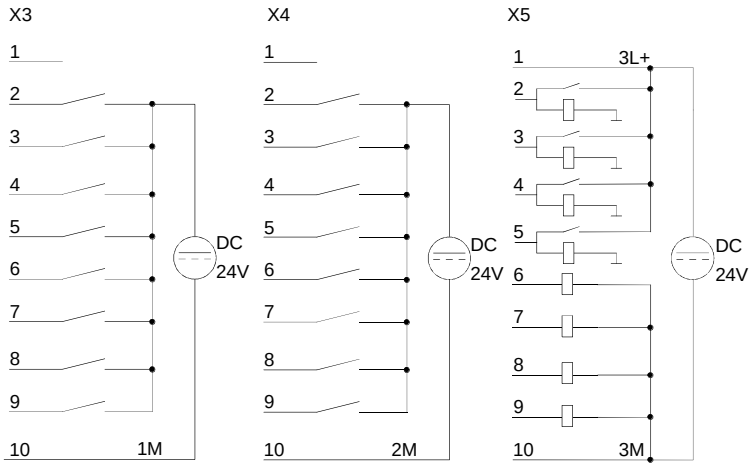

**Micro-PLC CPU 114**

VIPA 114-6BJ02/3/4:
DI 16(20)xDC 24V / DO 8(4)xDC 24V 0.5A

DI
IB0

DI
IB1

DIO
IB2
OB0



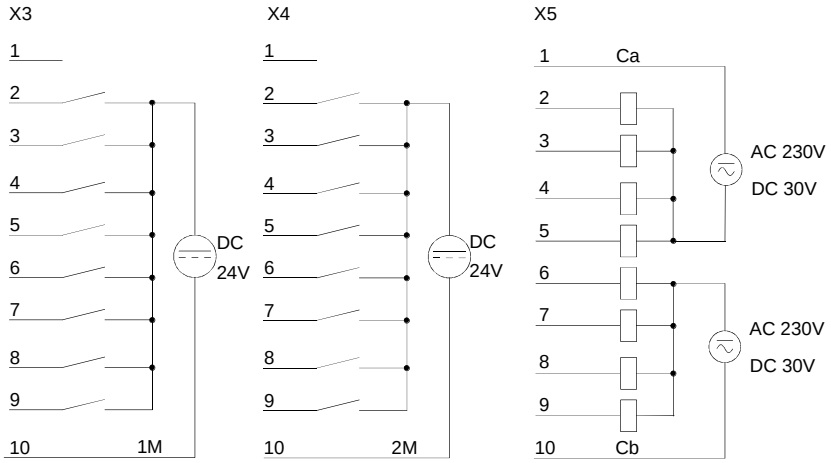
**Micro-PLC CPU 114R**

VIPA 114-6BJ52/3/4:
DI 16xDC 24V / DO 8xRelay

DI
IB0

DI
IB1

DO
OB0




**Micro-PLC CPU 115**

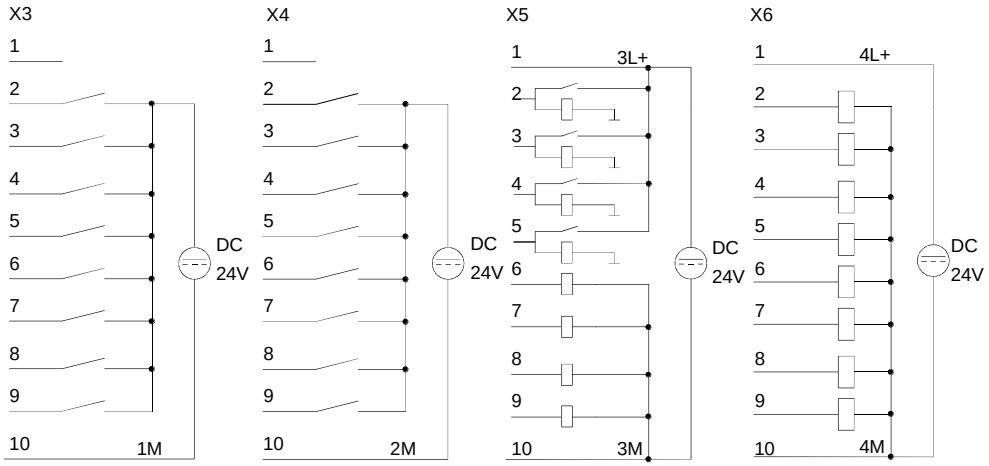
VIPA 115-6BLx2/3/4:
DI 16(20)xDC 24V / DO 16(12)xDC 24V 0.5A

DI
IB0

DI
IB1

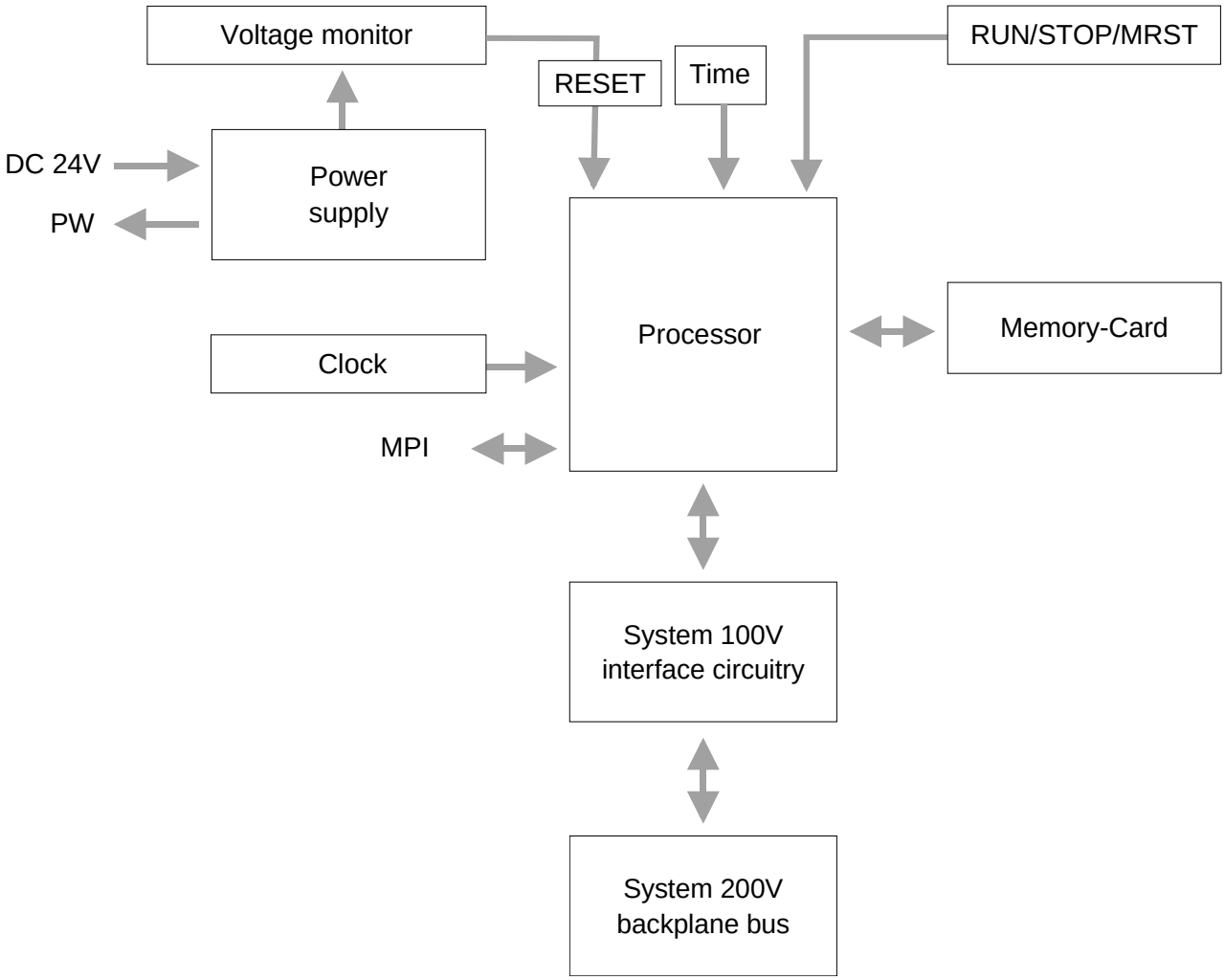
DIO
IB2
OB0

DO
OB1



# Block diagram

The following block diagram shows the hardware construction of the CPU section in principal:



# Function security of the VIPA CPUs

**Security mechanisms**

The CPUs include security mechanisms like a watchdog (100ms) and a parameterizable cycle time surveillance (parameterizable min. 1ms) that stop res. execute a RESET at the CPU in case of an error and set it into a defined STOP state.

The VIPA CPUs are developed function secure and have the following system properties:

| Event | concerns | Effect |
|---|---|---|
| RUN → STOP | general | BASP (**B**efehls-**A**usgabe-**Sp**erre, i.e. command output lock) is set. |
| | central digital outputs | The outputs are set to 0V. |
| | central analog outputs | The voltage supply for the output channels is switched off. |
| | decentral outputs | The outputs are set to 0V. |
| | decentral inputs | The inputs are read constantly from the slave and the recent values are put at disposal. |
| STOP → RUN res. Power on | general | First the PII is deleted, the call of the OB100 follows. After the execution of the OB, the BASP is set back and the cycle starts with: Delete PIQ → Read PII → OB1. |
| | central analog outputs | The behavior of the outputs at restart can be preset. |
| | decentral inputs | The inputs are read constantly from the slave and the recent values are put at disposal. |
| RUN | general | The program execution happens cyclically and can therefore be foreseen: Read PII → OB1 → Write PIQ. |

PII: = Process image inputs

PIQ: = Process image outputs

# Operation modes of the CPU section

**General**

These CPUs are intended for small and medium sized applications and are supplied with an integrated 24V power supply. The CPU contains a standard processor with internal program memory. The unit provides a powerful solution for process automation applications within the System 100V family.

A CPU supports the following modes of operation:

- **cyclic processing**

  Cyclic processing represents the major portion of all the processes that are executed in the CPU. Identical sequences of operations are repeated in a never ending cycle.
- **timer processing**

  Where a process requires control signals at constant intervals you can initiate certain operations based upon a timer, e.g. not critical monitoring functions at one-second intervals.
- **alarm controlled processing**

  If a process signal requires a quick response you would allocate this signal to an alarm controlled procedure. An alarm may activate a procedure in your program.
- **priority based processing**

  The above processes are handled by the CPU in accordance with their priority. Since a timer or an alarm event requires a quick reaction the CPU will interrupt the cyclic processing when these high-priority events occur to react to the event. Cyclic processing will resume once the reaction has been processed. This means that cyclic processing has the lowest priority.

**Software**

The software available in every CPU is parted as follows:

- **System application**

  The system application organizes all functions and processes of the CPU that are not related to a specific control task.
- **User application**

  Here you may find all functions that you need for processing specific control tasks. The operation blocks (OBs) provide the interfaces to the system application.

## Operands

The CPU 11x supports the following operand areas for the project engineering:

- Process image and periphery
- Marker
- Timers and counters
- Data blocks

### Process image and periphery

The user program is able to access the process image of the inputs and outputs PAA/PAE very quickly.

You have access to the following types of data:

Individual bits, bytes, words, double words

You may also gain direct access from your user program to peripheral modules via the bus.

The following types of data are available: bytes, words, blocks

### Marker

Bit memory is an area of memory that is accessible to the user program by means of certain operations. The marker area is intended to store frequently used working data.

You may access the following types of data: individual bits, bytes, words, double words

### Timer and counter

With your program you may load a time cell with a value between 10ms and 9990s. As soon as the user program executes a start operation, the value of this timer is decremented by the interval that you have specified until it reaches zero.

You may load counter cells with an initial value (max. 999) and increment or decrement this as required. Additionally your Micro-PLC includes parameterizable HSC inputs (high-speed counter).

### Data blocks

A data block contains constants or variables in form of bytes, words or double words. You may always access the current data block by means of operands.

You may access the following types of data: individual bits, bytes, words, double words.

# Technical data

## CPU 11x standard

| Order number | 112-4BH02 | 114-6BJ02 | 114-6BJ03 |
|---|---|---|---|
| Type | CPU 112 | CPU 114 | CPU 114 |
| **Technical data power supply** | | | |
| Power supply (rated value) | DC 24 V | DC 24 V | DC 24 V |
| Power supply (permitted range) | DC 20.4...28.8 V | DC 20.4...28.8 V | DC 20.4...28.8 V |
| Reverse polarity protection | ✓ | ✓ | ✓ |
| Current consumption (rated value) | 50 mA | 80 mA | 80 mA |
| **Technical data digital inputs** | | | |
| Number of inputs | 8 (12) | 16 (20) | 16 (20) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ | ✓ | ✓ |
| Current consumption from load voltage L+ (without load) | - | - | - |
| Rated value | DC 24 V | DC 24 V | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V | DC 0...5 V | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V | DC 15...28.8 V | DC 15...28.8 V |
| Input current for signal "1" | 7 mA | 7 mA | 7 mA |
| Connection of Two-Wire-BEROs possible | - | - | - |
| Max. permissible BERO quiescent current | 1.5 mA | 1.5 mA | 1.5 mA |
| Input delay of "0" to "1" | 3 ms | 3 ms | 3 ms |
| Input delay of "1" to "0" | 3 ms | 3 ms | 3 ms |
| Input characteristic curve | IEC 61131, type 1 | IEC 61131, type 1 | IEC 61131, type 1 |
| Initial data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data digital outputs** | | | |
| Number of outputs | 8 (4 | 8 (4 | 8 (4 |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | - | - | - |
| Current consumption from load voltage L+ (without load) | 50 mA | 50 mA | 50 mA |
| Output voltage signal "1" at min. current | L+ (-0.8 V) | L+ (-0.8 V) | L+ (-0.8 V) |
| Output voltage signal "1" at max. current | - | - | - |
| Output current at signal "1", rated value | 0.5 A | 0.5 A | 0.5 A |
| Output delay of "0" to "1" | max. 100 µs | max. 100 µs | max. 100 µs |
| Output delay of "1" to "0" | max. 350 µs | max. 350 µs | max. 350 µs |
| Minimum load current | - | - | - |
| Lamp load | 5 W | 5 W | 5 W |
| Switching frequency with resistive load | max. 1000 Hz | max. 1000 Hz | max. 1000 Hz |
| Switching frequency with inductive load | max. 0.5 Hz | max. 0.5 Hz | max. 0.5 Hz |
| Switching frequency on lamp load | max. 10 Hz | max. 10 Hz | max. 10 Hz |
| Internal limitation of inductive shut-off voltage | L+ (-52 V) | L+ (-52 V) | L+ (-52 V) |
| Short-circuit protection of output | yes, electronic | yes, electronic | yes, electronic |
| Trigger level | 1 A | 1 A | 1 A |
| Output data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data counters** | | | |
| Number of counters | - | 4 | 4 |
| Counter width | - | 32 Bit | 32 Bit |
| Maximum input frequency | - | 30 kHz | 30 kHz |
| Maximum count frequency | - | 30 kHz | 30 kHz |
| Mode incremental encoder | - | ✓ | ✓ |
| Mode pulse / direction | - | ✓ | ✓ |
| Mode pulse | - | ✓ | ✓ |
| Mode frequency counter | - | - | - |
| Mode period measurement | - | - | - |
| Gate input available | - | ✓ | ✓ |
| Latch input available | - | - | - |
| Reset input available | - | - | - |
| Counter output available | - | - | - |
| **Load and working memory** | | | |
| Load memory, integrated | 16 KB | 24 KB | 32 KB |
| Work memory, integrated | 8 KB | 16 KB | 24 KB |
| Memory card slot | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB |

| **Order number** | **112-4BH02** | **114-6BJ02** | **114-6BJ03** |
|---|---|---|---|
| **Hardware configuration** | | | |
| Racks, max. | - | 1 | 1 |
| Modules per rack, max. | - | 4 | 4 |
| Number of integrated DP master | - | - | - |
| Number of DP master via CP | - | 4 | 4 |
| Operable function modules | - | 4 | 4 |
| Operable communication modules PtP | - | 4 | 4 |
| **Status information, alarms, diagnostics** | | | |
| Status display | yes | yes | yes |
| Interrupts | yes | yes | yes |
| Process alarm | yes | yes | yes |
| Diagnostic interrupt | yes | yes | yes |
| Diagnostic functions | no | no | no |
| Diagnostics information read-out | possible | possible | possible |
| Supply voltage display | green LED | green LED | green LED |
| Group error display | red SF LED | red SF LED | red SF LED |
| Channel error display | none | none | none |
| **Isolation** | | | |
| Between channels of groups to | 8 | 8 | 8 |
| Between channels and backplane bus | ✓ | ✓ | ✓ |
| Insulation tested with | DC 500 V | DC 500 V | DC 500 V |
| **Command processing times** | | | |
| Bit instructions, min. | 0.25 µs | 0.25 µs | 0.25 µs |
| Word instruction, min. | 1.2 µs | 1.2 µs | 1.2 µs |
| Double integer arithmetic, min. | - | - | - |
| Floating-point arithmetic, min. | - | - | - |
| **Timers/Counters and their retentive characteristics** | | | |
| Number of S7 counters | 256 | 256 | 256 |
| S7 counter remanence | adjustable 0 up to 64 | adjustable 0 up to 64 | adjustable 0 up to 64 |
| S7 counter remanence adjustable | C0 .. C7 | C0 .. C7 | C0 .. C7 |
| Number of S7 times | 256 | 256 | 256 |
| S7 times remanence | adjustable 0 up to 128 | adjustable 0 up to 128 | adjustable 0 up to 128 |
| S7 times remanence adjustable | not retentive | not retentive | not retentive |
| **Data range and retentive characteristic** | | | |
| Number of flags | 8192 Bit | 8192 Bit | 8192 Bit |
| Bit memories retentive characteristic adjustable | adjustable 0 up to 256 | adjustable 0 up to 256 | adjustable 0 up to 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 | MB0 .. MB15 | MB0 .. MB15 |
| Number of data blocks | 2047 | 2047 | 2047 |
| Max. data blocks size | 16 KB | 16 KB | 16 KB |
| Max. local data size per execution level | 1024 Byte | 1024 Byte | 1024 Byte |
| **Blocks** | | | |
| Number of OBs | 14 | 14 | 14 |
| Maximum OB size | 16 KB | 16 KB | 16 KB |
| Number of FBs | 1024 | 1024 | 1024 |
| Maximum FB size | 16 KB | 16 KB | 16 KB |
| Number of FCs | 1024 | 1024 | 1024 |
| Maximum FC size | 16 KB | 16 KB | 16 KB |
| Maximum nesting depth per priority class | 8 | 8 | 8 |
| Maximum nesting depth additional within an error OB | 1 | 1 | 1 |
| **Time** | | | |
| Real-time clock buffered | ✓ | ✓ | ✓ |
| Clock buffered period (min.) | 30 d | 30 d | 30 d |
| Accuracy (max. deviation per day) | - | - | - |
| Number of operating hours counter | 8 | 8 | 8 |
| Value range operating hours counter | 32767 | 32767 | 32767 |
| Clock synchronization | - | - | - |
| **Address areas (I/O)** | | | |
| Input I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Input I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Input process image preset | 128 Byte | 128 Byte | 128 Byte |
| Output process image preset | 128 Byte | 128 Byte | 128 Byte |
| Digital inputs | 8 (12) | 16 (20) | 16 (20) |
| Digital outputs | 8 (4) | 8 (4) | 8 (4) |
| Integrated digital inputs | 8 (12) | 16 (20) | 16 (20) |
| Integrated digital outputs | 8 (4) | 8 (4) | 8 (4) |
| **Communication functions** | | | |
| PG/OP channel | ✓ | ✓ | ✓ |

| Order number | 112-4BH02 | 114-6BJ02 | 114-6BJ03 |
|---|---|---|---|
| Global data communication | ✓ | ✓ | ✓ |
| Number of GD circuits, max. | 4 | 4 | 4 |
| Size of GD packets, max. | 22 Byte | 22 Byte | 22 Byte |
| S7 basic communication | ✓ | ✓ | ✓ |
| S7 basic communication, user data per job | 76 Byte | 76 Byte | 76 Byte |
| S7 communication | ✓ | ✓ | ✓ |
| S7 communication as server | ✓ | ✓ | ✓ |
| S7 communication as client | - | - | - |
| S7 communication, user data per job | 160 Byte | 160 Byte | 160 Byte |
| Number of connections, max. | 16 | 16 | 16 |
| **Functionality Sub-D interfaces** | | | |
| Type | - | - | - |
| Type of interface | RS485 | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | - | - | - |
| MPI | ✓ | ✓ | ✓ |
| MP²I (MPI/RS232) | ✓ | ✓ | ✓ |
| DP master | - | - | - |
| DP slave | - | - | - |
| Point-to-point interface | - | - | - |
| **Mechanical data** | | | |
| Dimensions (WxHxD) | 101.6 x 76 x 48 mm | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm |
| Weight | 219 g | 266 g | 266 g |
| **Environmental conditions** | | | |
| Operating temperature | 0 °C to 60 °C | 0 °C to 60 °C | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C | -25 °C to 70 °C | -25 °C to 70 °C |
| **Certifications** | | | |
| UL508 certification | yes | yes | yes |

| **Order number** | **114-6BJ04** | **114-6BJ52** | **114-6BJ53** |
|---|---|---|---|
| Type | CPU 114 | CPU 114R | CPU 114R |
| **Technical data power supply** | | | |
| Power supply (rated value) | DC 24 V | DC 24 V | DC 24 V |
| Power supply (permitted range) | DC 20.4...28.8 V | DC 20.4...28.8 V | DC 20.4...28.8 V |
| Reverse polarity protection | ✓ | ✓ | ✓ |
| Current consumption (rated value) | 80 mA | 150 mA | 150 mA |
| **Technical data digital inputs** | | | |
| Number of inputs | 16 (20) | 16 | 16 |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ | ✓ | ✓ |
| Current consumption from load voltage L+ (without load) | - | - | - |
| Rated value | DC 24 V | DC 24 V | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V | DC 0...5 V | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V | DC 15...28.8 V | DC 15...28.8 V |
| Input current for signal "1" | 7 mA | 7 mA | 7 mA |
| Connection of Two-Wire-BEROs possible | - | ✓ | ✓ |
| Max. permissible BERO quiescent current | 1.5 mA | 1.5 mA | 1.5 mA |
| Input delay of "0" to "1" | 3 ms | 3 ms | 3 ms |
| Input delay of "1" to "0" | 3 ms | 3 ms | 3 ms |
| Input characteristic curve | IEC 61131, type 1 | IEC 61131, type 1 | IEC 61131, type 1 |
| Initial data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data digital outputs** | | | |
| Number of outputs | 8 (4) | 8 | 8 |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 30 V/ AC 230 V | DC 30 V/ AC 230 V |
| Reverse polarity protection of rated load voltage | - | - | - |
| Current consumption from load voltage L+ (without load) | 50 mA | - | - |
| Output voltage signal "1" at min. current | L+ (-0.8 V) | - | - |
| Output voltage signal "1" at max. current | - | - | - |
| Output current at signal "1", rated value | 0.5 A | 5 A | 5 A |
| Output delay of "0" to "1" | max. 100 µs | 10 ms | 10 ms |
| Output delay of "1" to "0" | max. 350 µs | 5 ms | 5 ms |
| Minimum load current | - | - | - |
| Lamp load | 5 W | - | - |
| Switching frequency with resistive load | max. 1000 Hz | max. 10 Hz | max. 10 Hz |
| Switching frequency with inductive load | max. 0.5 Hz | - | - |
| Switching frequency on lamp load | max. 10 Hz | - | - |
| Internal limitation of inductive shut-off voltage | L+ (-52 V) | - | - |
| Short-circuit protection of output | yes, electronic | - | - |
| Trigger level | 1 A | - | - |
| Output data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data counters** | | | |
| Number of counters | 4 | 4 | 4 |
| Counter width | 32 Bit | 32 Bit | 32 Bit |
| Maximum input frequency | 30 kHz | 30 kHz | 30 kHz |
| Maximum count frequency | 30 kHz | 30 kHz | 30 kHz |
| Mode incremental encoder | ✓ | ✓ | ✓ |
| Mode pulse / direction | ✓ | ✓ | ✓ |
| Mode pulse | ✓ | ✓ | ✓ |
| Mode frequency counter | - | - | - |
| Mode period measurement | - | - | - |
| Gate input available | ✓ | ✓ | ✓ |
| Latch input available | - | - | - |
| Reset input available | - | - | - |
| Counter output available | - | - | - |
| **Load and working memory** | | | |
| Load memory, integrated | 40 KB | 24 KB | 32 KB |
| Work memory, integrated | 32 KB | 16 KB | 24 KB |
| Memory card slot | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB |
| **Hardware configuration** | | | |
| Racks, max. | 1 | 1 | 1 |
| Modules per rack, max. | 4 | 4 | 4 |
| Number of integrated DP master | - | - | - |
| Number of DP master via CP | 4 | 4 | 4 |

| Order number | 114-6BJ04 | 114-6BJ52 | 114-6BJ53 |
|---|---|---|---|
| Operable function modules | 4 | 4 | 4 |
| Operable communication modules PtP | 4 | 4 | 4 |
| **Status information, alarms, diagnostics** | | | |
| Status display | yes | yes | yes |
| Interrupts | yes | yes | yes |
| Process alarm | yes | yes | yes |
| Diagnostic interrupt | yes | yes | yes |
| Diagnostic functions | no | no | no |
| Diagnostics information read-out | possible | possible | possible |
| Supply voltage display | green LED | green LED | green LED |
| Group error display | red SF LED | red SF LED | red SF LED |
| Channel error display | none | none | none |
| **Isolation** | | | |
| Between channels of groups to | 8 | 8 | 8 |
| Between channels and backplane bus | ✓ | ✓ | ✓ |
| Insulation tested with | DC 500 V | DC 500 V | DC 500 V |
| **Command processing times** | | | |
| Bit instructions, min. | 0.25 µs | 0.25 µs | 0.25 µs |
| Word instruction, min. | 1.2 µs | 1.2 µs | 1.2 µs |
| Double integer arithmetic, min. | - | - | - |
| Floating-point arithmetic, min. | - | - | - |
| **Timers/Counters and their retentive characteristics** | | | |
| Number of S7 counters | 256 | 256 | 256 |
| S7 counter remanence | adjustable 0 up to 64 | adjustable 0 up to 64 | adjustable 0 up to 64 |
| S7 counter remanence adjustable | C0 .. C7 | C0 .. C7 | C0 .. C7 |
| Number of S7 times | 256 | 256 | 256 |
| S7 times remanence | adjustable 0 up to 128 | adjustable 0 up to 128 | adjustable 0 up to 128 |
| S7 times remanence adjustable | not retentive | not retentive | not retentive |
| **Data range and retentive characteristic** | | | |
| Number of flags | 8192 Bit | 8192 Bit | 8192 Bit |
| Bit memories retentive characteristic adjustable | adjustable 0 up to 256 | adjustable 0 up to 256 | adjustable 0 up to 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 | MB0 .. MB15 | MB0 .. MB15 |
| Number of data blocks | 2047 | 2047 | 2047 |
| Max. data blocks size | 16 KB | 16 KB | 16 KB |
| Max. local data size per execution level | 1024 Byte | 1024 Byte | 1024 Byte |
| **Blocks** | | | |
| Number of OBs | 14 | 14 | 14 |
| Maximum OB size | 16 KB | 16 KB | 16 KB |
| Number of FBs | 1024 | 1024 | 1024 |
| Maximum FB size | 16 KB | 16 KB | 16 KB |
| Number of FCs | 1024 | 1024 | 1024 |
| Maximum FC size | 16 KB | 16 KB | 16 KB |
| Maximum nesting depth per priority class | 8 | 8 | 8 |
| Maximum nesting depth additional within an error OB | 1 | 1 | 1 |
| **Time** | | | |
| Real-time clock buffered | ✓ | ✓ | ✓ |
| Clock buffered period (min.) | 30 d | 30 d | 30 d |
| Accuracy (max. deviation per day) | - | - | - |
| Number of operating hours counter | 8 | 8 | 8 |
| Value range operating hours counter | 32767 | 32767 | 32767 |
| Clock synchronization | - | - | - |
| **Address areas (I/O)** | | | |
| Input I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Input I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Input process image preset | 128 Byte | 128 Byte | 128 Byte |
| Output process image preset | 128 Byte | 128 Byte | 128 Byte |
| Digital inputs | 16 (20) | 16 | 16 |
| Digital outputs | 8 (4) | 8 | 8 |
| Integrated digital inputs | 16 (20) | 16 | 16 |
| Integrated digital outputs | 8 (4) | 8 | 8 |
| **Communication functions** | | | |
| PG/OP channel | ✓ | ✓ | ✓ |
| Global data communication | ✓ | ✓ | ✓ |
| Number of GD circuits, max. | 4 | 4 | 4 |
| Size of GD packets, max. | 22 Byte | 22 Byte | 22 Byte |
| S7 basic communication | ✓ | ✓ | ✓ |
| S7 basic communication, user data per job | 76 Byte | 76 Byte | 76 Byte |
| S7 communication | ✓ | ✓ | ✓ |

| **Order number** | **114-6BJ04** | **114-6BJ52** | **114-6BJ53** |
|---|---|---|---|
| S7 communication as server | ✓ | ✓ | ✓ |
| S7 communication as client | - | - | - |
| S7 communication, user data per job | 160 Byte | 160 Byte | 160 Byte |
| Number of connections, max. | 16 | 16 | 16 |
| **Functionality Sub-D interfaces** | | | |
| Type | - | - | - |
| Type of interface | RS485 | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | - | - | - |
| MPI | ✓ | ✓ | ✓ |
| MP²I (MPI/RS232) | ✓ | ✓ | ✓ |
| DP master | - | - | - |
| DP slave | - | - | - |
| Point-to-point interface | - | - | - |
| **Mechanical data** | | | |
| Dimensions (WxHxD) | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm |
| Weight | 266 g | 280 g | 280 g |
| **Environmental conditions** | | | |
| Operating temperature | 0 °C to 60 °C | 0 °C to 60 °C | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C | -25 °C to 70 °C | -25 °C to 70 °C |
| **Certifications** | | | |
| UL508 certification | yes | yes | yes |

| Order number | 114-6BJ54 | 115-6BL02 | 115-6BL03 |
|---|---|---|---|
| Type | CPU 114R | CPU 115 | CPU 115 |
| **Technical data power supply** | | | |
| Power supply (rated value) | DC 24 V | DC 24 V | DC 24 V |
| Power supply (permitted range) | DC 20.4...28.8 V | DC 20.4...28.8 V | DC 20.4...28.8 V |
| Reverse polarity protection | ✓ | ✓ | ✓ |
| Current consumption (rated value) | 150 mA | 90 mA | 90 mA |
| **Technical data digital inputs** | | | |
| Number of inputs | 16 | 16 (20) | 16 (20) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ | ✓ | ✓ |
| Current consumption from load voltage L+ (without load) | - | - | - |
| Rated value | DC 24 V | DC 24 V | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V | DC 0...5 V | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V | DC 15...28.8 V | DC 15...28.8 V |
| Input current for signal "1" | 7 mA | 7 mA | 7 mA |
| Connection of Two-Wire-BEROs possible | ✓ | ✓ | ✓ |
| Max. permissible BERO quiescent current | 1.5 mA | 1.5 mA | 1.5 mA |
| Input delay of "0" to "1" | 3 ms | 3 ms | 3 ms |
| Input delay of "1" to "0" | 3 ms | 3 ms | 3 ms |
| Input characteristic curve | IEC 61131, type 1 | IEC 61131, type 1 | IEC 61131, type 1 |
| Initial data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data digital outputs** | | | |
| Number of outputs | 8 | 16 (12) | 16 (12) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 30 V/ AC 230 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | - | - | - |
| Current consumption from load voltage L+ (without load) | - | 50 mA | 50 mA |
| Output voltage signal "1" at min. current | - | L+ (-0.8 V) | L+ (-0.8 V) |
| Output voltage signal "1" at max. current | - | - | - |
| Output current at signal "1", rated value | 5 A | 0.5 A | 0.5 A |
| Output delay of "0" to "1" | 10 ms | max. 100 µs | max. 100 µs |
| Output delay of "1" to "0" | 5 ms | max. 350 µs | max. 350 µs |
| Minimum load current | - | - | - |
| Lamp load | - | 5 W | 5 W |
| Switching frequency with resistive load | max. 10 Hz | max. 1000 Hz | max. 1000 Hz |
| Switching frequency with inductive load | - | max. 0.5 Hz | max. 0.5 Hz |
| Switching frequency on lamp load | - | max. 10 Hz | max. 10 Hz |
| Internal limitation of inductive shut-off voltage | - | L+ (-52 V) | L+ (-52 V) |
| Short-circuit protection of output | - | yes, electronic | yes, electronic |
| Trigger level | - | 1 A | 1 A |
| Output data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data counters** | | | |
| Number of counters | 4 | 4 | 4 |
| Counter width | 32 Bit | 32 Bit | 32 Bit |
| Maximum input frequency | 30 kHz | 30 kHz | 30 kHz |
| Maximum count frequency | 30 kHz | 30 kHz | 30 kHz |
| Mode incremental encoder | ✓ | ✓ | ✓ |
| Mode pulse / direction | ✓ | ✓ | ✓ |
| Mode pulse | ✓ | ✓ | ✓ |
| Mode frequency counter | - | - | - |
| Mode period measurement | - | - | - |
| Gate input available | ✓ | ✓ | ✓ |
| Latch input available | - | - | - |
| Reset input available | - | - | - |
| Counter output available | - | - | - |
| **Load and working memory** | | | |
| Load memory, integrated | 40 KB | 24 KB | 32 KB |
| Work memory, integrated | 32 KB | 16 KB | 24 KB |
| Memory card slot | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB |
| **Hardware configuration** | | | |
| Racks, max. | 1 | 1 | 1 |
| Modules per rack, max. | 4 | 4 | 4 |
| Number of integrated DP master | - | - | - |
| Number of DP master via CP | 4 | 4 | 4 |

| **Order number** | **114-6BJ54** | **115-6BL02** | **115-6BL03** |
|---|---|---|---|
| Operable function modules | 4 | 4 | 4 |
| Operable communication modules PtP | 4 | 4 | 4 |
| **Status information, alarms, diagnostics** | | | |
| Status display | yes | yes | yes |
| Interrupts | yes | yes | yes |
| Process alarm | yes | yes | yes |
| Diagnostic interrupt | yes | yes | yes |
| Diagnostic functions | no | no | no |
| Diagnostics information read-out | possible | possible | possible |
| Supply voltage display | green LED | green LED | green LED |
| Group error display | red SF LED | red SF LED | red SF LED |
| Channel error display | none | none | none |
| **Isolation** | | | |
| Between channels of groups to | 8 | 8 | 8 |
| Between channels and backplane bus | ✓ | ✓ | ✓ |
| Insulation tested with | DC 500 V | DC 500 V | DC 500 V |
| **Command processing times** | | | |
| Bit instructions, min. | 0.25 µs | 0.25 µs | 0.25 µs |
| Word instruction, min. | 1.2 µs | 1.2 µs | 1.2 µs |
| Double integer arithmetic, min. | - | - | - |
| Floating-point arithmetic, min. | - | - | - |
| **Timers/Counters and their retentive characteristics** | | | |
| Number of S7 counters | 256 | 256 | 256 |
| S7 counter remanence | adjustable 0 up to 64 | adjustable 0 up to 64 | adjustable 0 up to 64 |
| S7 counter remanence adjustable | C0 .. C7 | C0 .. C7 | C0 .. C7 |
| Number of S7 times | 256 | 256 | 256 |
| S7 times remanence | adjustable 0 up to 128 | adjustable 0 up to 128 | adjustable 0 up to 128 |
| S7 times remanence adjustable | not retentive | not retentive | not retentive |
| **Data range and retentive characteristic** | | | |
| Number of flags | 8192 Bit | 8192 Bit | 8192 Bit |
| Bit memories retentive characteristic adjustable | adjustable 0 up to 256 | adjustable 0 up to 256 | adjustable 0 up to 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 | MB0 .. MB15 | MB0 .. MB15 |
| Number of data blocks | 2047 | 2047 | 2047 |
| Max. data blocks size | 16 KB | 16 KB | 16 KB |
| Max. local data size per execution level | 1024 Byte | 1024 Byte | 1024 Byte |
| **Blocks** | | | |
| Number of OBs | 14 | 14 | 14 |
| Maximum OB size | 16 KB | 16 KB | 16 KB |
| Number of FBs | 1024 | 1024 | 1024 |
| Maximum FB size | 16 KB | 16 KB | 16 KB |
| Number of FCs | 1024 | 1024 | 1024 |
| Maximum FC size | 16 KB | 16 KB | 16 KB |
| Maximum nesting depth per priority class | 8 | 8 | 8 |
| Maximum nesting depth additional within an error OB | 1 | 1 | 1 |
| **Time** | | | |
| Real-time clock buffered | ✓ | ✓ | ✓ |
| Clock buffered period (min.) | 30 d | 30 d | 30 d |
| Accuracy (max. deviation per day) | - | - | - |
| Number of operating hours counter | 8 | 8 | 8 |
| Value range operating hours counter | 32767 | 32767 | 32767 |
| Clock synchronization | - | - | - |
| **Address areas (I/O)** | | | |
| Input I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Input I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Input process image preset | 128 Byte | 128 Byte | 128 Byte |
| Output process image preset | 128 Byte | 128 Byte | 128 Byte |
| Digital inputs | 16 | 16 (20 | 16 (20 |
| Digital outputs | 8 | 16 (12 | 16 (12 |
| Integrated digital inputs | 16 | 16 (20 | 16 (20 |
| Integrated digital outputs | 8 | 16 (12 | 16 (12 |
| **Communication functions** | | | |
| PG/OP channel | ✓ | ✓ | ✓ |
| Global data communication | ✓ | ✓ | ✓ |
| Number of GD circuits, max. | 4 | 4 | 4 |
| Size of GD packets, max. | 22 Byte | 22 Byte | 22 Byte |
| S7 basic communication | ✓ | ✓ | ✓ |
| S7 basic communication, user data per job | 76 Byte | 76 Byte | 76 Byte |
| S7 communication | ✓ | ✓ | ✓ |

| Order number | 114-6BJ54 | 115-6BL02 | 115-6BL03 |
|---|---|---|---|
| S7 communication as server | ✓ | ✓ | ✓ |
| S7 communication as client | - | - | - |
| S7 communication, user data per job | 160 Byte | 160 Byte | 160 Byte |
| Number of connections, max. | 16 | 16 | 16 |
| **Functionality Sub-D interfaces** | | | |
| Type | - | - | - |
| Type of interface | RS485 | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | - | - | - |
| MPI | ✓ | ✓ | ✓ |
| MP²I (MPI/RS232) | ✓ | ✓ | ✓ |
| DP master | - | - | - |
| DP slave | - | - | - |
| Point-to-point interface | - | - | - |
| **Mechanical data** | | | |
| Dimensions (WxHxD) | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm |
| Weight | 280 g | 292 g | 292 g |
| **Environmental conditions** | | | |
| Operating temperature | 0 °C to 60 °C | 0 °C to 60 °C | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C | -25 °C to 70 °C | -25 °C to 70 °C |
| **Certifications** | | | |
| UL508 certification | yes | yes | yes |

| **Order number** | **115-6BL04** | **115-6BL72** |
|---|---|---|
| Type | CPU 115 | CPU 115 |
| **Technical data power supply** | | |
| Power supply (rated value) | DC 24 V | DC 24 V |
| Power supply (permitted range) | DC 20.4...28.8 V | DC 20.4...28.8 V |
| Reverse polarity protection | ✓ | ✓ |
| Current consumption (rated value) | 90 mA | 90 mA |
| **Technical data digital inputs** | | |
| Number of inputs | 16 (20) | 16 (20) |
| Cable length, shielded | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ | ✓ |
| Current consumption from load voltage L+ (without load) | - | - |
| Rated value | DC 24 V | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V | DC 15...28.8 V |
| Input current for signal "1" | 7 mA | 7 mA |
| Connection of Two-Wire-BEROs possible | ✓ | ✓ |
| Max. permissible BERO quiescent current | 1.5 mA | 1.5 mA |
| Input delay of "0" to "1" | 3 ms | 3 ms |
| Input delay of "1" to "0" | 3 ms | 3 ms |
| Input characteristic curve | IEC 61131, type 1 | IEC 61131, type 1 |
| Initial data size | 3 Byte | 3 Byte |
| **Technical data digital outputs** | | |
| Number of outputs | 16 (12) | 16 (12) |
| Cable length, shielded | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | - | - |
| Current consumption from load voltage L+ (without load) | 50 mA | 50 mA |
| Output voltage signal "1" at min. current | L+ (-0.8 V) | L+ (-0.8 V) |
| Output voltage signal "1" at max. current | - | - |
| Output current at signal "1", rated value | 0.5 A | 0.5 A |
| Output delay of "0" to "1" | max. 100 µs | max. 100 µs |
| Output delay of "1" to "0" | max. 350 µs | max. 350 µs |
| Minimum load current | - | - |
| Lamp load | 5 W | 5 W |
| Switching frequency with resistive load | max. 1000 Hz | max. 1000 Hz |
| Switching frequency with inductive load | max. 0.5 Hz | max. 0.5 Hz |
| Switching frequency on lamp load | max. 10 Hz | max. 10 Hz |
| Internal limitation of inductive shut-off voltage | L+ (-52 V) | L+ (-52 V) |
| Short-circuit protection of output | yes, electronic | yes, electronic |
| Trigger level | 1 A | 1 A |
| Output data size | 3 Byte | 3 Byte |
| **Technical data counters** | | |
| Number of counters | 4 | 4 |
| Counter width | 32 Bit | 32 Bit |
| Maximum input frequency | 30 kHz | 30 kHz |
| Maximum count frequency | 30 kHz | 30 kHz |
| Mode incremental encoder | ✓ | ✓ |
| Mode pulse / direction | ✓ | ✓ |
| Mode pulse | ✓ | ✓ |
| Mode frequency counter | - | - |
| Mode period measurement | - | - |
| Gate input available | ✓ | ✓ |
| Latch input available | - | - |
| Reset input available | - | - |
| Counter output available | - | - |
| **Load and working memory** | | |
| Load memory, integrated | 40 KB | 24 KB |
| Work memory, integrated | 32 KB | 16 KB |
| Memory card slot | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB |
| **Hardware configuration** | | |
| Racks, max. | 1 | 1 |
| Modules per rack, max. | 4 | 7 |
| Number of integrated DP master | - | - |
| Number of DP master via CP | 4 | 7 |

| Order number | 115-6BL04 | 115-6BL72 |
|---|---|---|
| Operable function modules | 4 | 7 |
| Operable communication modules PtP | 4 | 7 |
| **Status information, alarms, diagnostics** | | |
| Status display | yes | yes |
| Interrupts | yes | yes |
| Process alarm | yes | yes |
| Diagnostic interrupt | yes | yes |
| Diagnostic functions | no | no |
| Diagnostics information read-out | possible | none |
| Supply voltage display | green LED | green LED |
| Group error display | red SF LED | red SF LED |
| Channel error display | none | none |
| **Isolation** | | |
| Between channels of groups to | 8 | 8 |
| Between channels and backplane bus | ✓ | ✓ |
| Insulation tested with | DC 500 V | DC 500 V |
| **Command processing times** | | |
| Bit instructions, min. | 0.25 µs | 0.25 µs |
| Word instruction, min. | 1.2 µs | 1.2 µs |
| Double integer arithmetic, min. | - | - |
| Floating-point arithmetic, min. | - | - |
| **Timers/Counters and their retentive characteristics** | | |
| Number of S7 counters | 256 | 256 |
| S7 counter remanence | adjustable 0 up to 64 | adjustable 0 up to 64 |
| S7 counter remanence adjustable | C0 .. C7 | C0 .. C7 |
| Number of S7 times | 256 | 256 |
| S7 times remanence | adjustable 0 up to 128 | adjustable 0 up to 128 |
| S7 times remanence adjustable | not retentive | not retentive |
| **Data range and retentive characteristic** | | |
| Number of flags | 8192 Bit | 8192 Bit |
| Bit memories retentive characteristic adjustable | adjustable 0 up to 256 | adjustable 0 up to 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 | MB0 .. MB15 |
| Number of data blocks | 2047 | - |
| Max. data blocks size | 16 KB | - |
| Max. local data size per execution level | 1024 Byte | - |
| **Blocks** | | |
| Number of OBs | 14 | - |
| Maximum OB size | 16 KB | - |
| Number of FBs | 1024 | 1024 |
| Maximum FB size | 16 KB | - |
| Number of FCs | 1024 | 1024 |
| Maximum FC size | 16 KB | - |
| Maximum nesting depth per priority class | 8 | - |
| Maximum nesting depth additional within an error OB | 1 | - |
| **Time** | | |
| Real-time clock buffered | ✓ | ✓ |
| Clock buffered period (min.) | 30 d | 30 d |
| Accuracy (max. deviation per day) | - | - |
| Number of operating hours counter | 8 | - |
| Value range operating hours counter | 32767 | - |
| Clock synchronization | - | - |
| **Address areas (I/O)** | | |
| Input I/O address area | 1024 Byte | 1024 Bit |
| Output I/O address area | 1024 Byte | 1024 Bit |
| Input I/O address area, decentral | 1024 Byte | - |
| Output I/O address area, decentral | 1024 Byte | - |
| Input process image preset | 128 Byte | 128 Byte |
| Output process image preset | 128 Byte | 128 Byte |
| Digital inputs | 16 (20) | 16 (20) |
| Digital outputs | 16 (12) | 16 (12) |
| Integrated digital inputs | 16 (20) | - |
| Integrated digital outputs | 16 (12) | - |
| **Communication functions** | | |
| PG/OP channel | ✓ | ✓ |
| Global data communication | ✓ | ✓ |
| Number of GD circuits, max. | 4 | 4 |
| Size of GD packets, max. | 22 Byte | 22 Byte |
| S7 basic communication | ✓ | ✓ |
| S7 basic communication, user data per job | 76 Byte | 76 Byte |
| S7 communication | ✓ | ✓ |

| **Order number** | **115-6BL04** | **115-6BL72** |
|---|---|---|
| S7 communication as server | ✓ | ✓ |
| S7 communication as client | - | - |
| S7 communication, user data per job | 160 Byte | 160 Byte |
| Number of connections, max. | 16 | 16 |
| **Functionality Sub-D interfaces** | | |
| Type | - | - |
| Type of interface | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | - | - |
| MPI | ✓ | ✓ |
| MP²I (MPI/RS232) | ✓ | ✓ |
| DP master | - | - |
| DP slave | - | - |
| Point-to-point interface | - | - |
| **Mechanical data** | | |
| Dimensions (WxHxD) | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm |
| Weight | 292 g | 292 g |
| **Environmental conditions** | | |
| Operating temperature | 0 °C to 60 °C | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C | -25 °C to 70 °C |
| **Certifications** | | |
| UL508 certification | yes | yes |

## CPU 11x PtP

| Order number | 115-6BL12 | 115-6BL13 | 115-6BL14 |
|---|---|---|---|
| Type | CPU 115SER | CPU 115SER | CPU 115SER |
| **Technical data power supply** | | | |
| Power supply (rated value) | DC 24 V | DC 24 V | DC 24 V |
| Power supply (permitted range) | DC 20.4...28.8 V | DC 20.4...28.8 V | DC 20.4...28.8 V |
| Reverse polarity protection | ✓ | ✓ | ✓ |
| Current consumption (rated value) | 100 mA | 100 mA | 100 mA |
| **Technical data digital inputs** | | | |
| Number of inputs | 16 (20) | 16 (20) | 16 (20) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ | ✓ | ✓ |
| Current consumption from load voltage L+ (without load) | - | - | - |
| Rated value | DC 24 V | DC 24 V | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V | DC 0...5 V | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V | DC 15...28.8 V | DC 15...28.8 V |
| Input current for signal "1" | 7 mA | 7 mA | 7 mA |
| Connection of Two-Wire-BEROs possible | ✓ | ✓ | ✓ |
| Max. permissible BERO quiescent current | 1.5 mA | 1.5 mA | 1.5 mA |
| Input delay of "0" to "1" | 3 ms | 3 ms | 3 ms |
| Input delay of "1" to "0" | 3 ms | 3 ms | 3 ms |
| Input characteristic curve | IEC 61131, type 1 | IEC 61131, type 1 | IEC 61131, type 1 |
| Initial data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data digital outputs** | | | |
| Number of outputs | 16 (12) | 16 (12) | 16 (12) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | - | - | - |
| Current consumption from load voltage L+ (without load) | 50 mA | 50 mA | 50 mA |
| Output voltage signal "1" at min. current | L+ (-0.8 V) | L+ (-0.8 V) | L+ (-0.8 V) |
| Output voltage signal "1" at max. current | - | - | - |
| Output current at signal "1", rated value | 0.5 A | 0.5 A | 0.5 A |
| Output delay of "0" to "1" | max. 100 µs | max. 100 µs | max. 100 µs |
| Output delay of "1" to "0" | max. 350 µs | max. 350 µs | max. 350 µs |
| Minimum load current | - | - | - |
| Lamp load | 5 W | 5 W | 5 W |
| Switching frequency with resistive load | max. 1000 Hz | max. 1000 Hz | max. 1000 Hz |
| Switching frequency with inductive load | max. 0.5 Hz | max. 0.5 Hz | max. 0.5 Hz |
| Switching frequency on lamp load | max. 10 Hz | max. 10 Hz | max. 10 Hz |
| Internal limitation of inductive shut-off voltage | L+ (-52 V) | L+ (-52 V) | L+ (-52 V) |
| Short-circuit protection of output | yes, electronic | yes, electronic | yes, electronic |
| Trigger level | 1 A | 1 A | 1 A |
| Output data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data counters** | | | |
| Number of counters | 4 | 4 | 4 |
| Counter width | 32 Bit | 32 Bit | 32 Bit |
| Maximum input frequency | 30 kHz | 30 kHz | 30 kHz |
| Maximum count frequency | 30 kHz | 30 kHz | 30 kHz |
| Mode incremental encoder | ✓ | ✓ | ✓ |
| Mode pulse / direction | ✓ | ✓ | ✓ |
| Mode pulse | ✓ | ✓ | ✓ |
| Mode frequency counter | - | - | - |
| Mode period measurement | - | - | - |
| Gate input available | ✓ | ✓ | ✓ |
| Latch input available | - | - | - |
| Reset input available | - | - | - |
| Counter output available | - | - | - |
| **Load and working memory** | | | |
| Load memory, integrated | 24 KB | 32 KB | 40 KB |
| Work memory, integrated | 16 KB | 24 KB | 32 KB |
| Memory card slot | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB |
| **Hardware configuration** | | | |
| Racks, max. | 1 | 1 | 1 |
| Modules per rack, max. | 4 | 4 | 4 |
| Number of integrated DP master | - | - | - |

| **Order number** | **115-6BL12** | **115-6BL13** | **115-6BL14** |
|---|---|---|---|
| Number of DP master via CP | 4 | 4 | 4 |
| Operable function modules | 4 | 4 | 4 |
| Operable communication modules PtP | 4 | 4 | 4 |
| **Status information, alarms, diagnostics** | | | |
| Status display | yes | yes | yes |
| Interrupts | yes | yes | yes |
| Process alarm | yes | yes | yes |
| Diagnostic interrupt | yes | yes | yes |
| Diagnostic functions | no | no | no |
| Diagnostics information read-out | possible | possible | possible |
| Supply voltage display | green LED | green LED | green LED |
| Group error display | red SF LED | red SF LED | red SF LED |
| Channel error display | none | none | none |
| **Isolation** | | | |
| Between channels of groups to | 8 | 8 | 8 |
| Between channels and backplane bus | ✓ | ✓ | ✓ |
| Insulation tested with | DC 500 V | DC 500 V | DC 500 V |
| **Command processing times** | | | |
| Bit instructions, min. | 0.25 µs | 0.25 µs | 0.25 µs |
| Word instruction, min. | 1.2 µs | 1.2 µs | 1.2 µs |
| Double integer arithmetic, min. | - | - | - |
| Floating-point arithmetic, min. | - | - | - |
| **Timers/Counters and their retentive characteristics** | | | |
| Number of S7 counters | 256 | 256 | 256 |
| S7 counter remanence | adjustable 0 up to 64 | adjustable 0 up to 64 | adjustable 0 up to 64 |
| S7 counter remanence adjustable | C0 .. C7 | C0 .. C7 | C0 .. C7 |
| Number of S7 times | 256 | 256 | 256 |
| S7 times remanence | adjustable 0 up to 128 | adjustable 0 up to 128 | adjustable 0 up to 128 |
| S7 times remanence adjustable | not retentive | not retentive | not retentive |
| **Data range and retentive characteristic** | | | |
| Number of flags | 8192 Bit | 8192 Bit | 8192 Bit |
| Bit memories retentive characteristic adjustable | adjustable 0 up to 256 | adjustable 0 up to 256 | adjustable 0 up to 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 | MB0 .. MB15 | MB0 .. MB15 |
| Number of data blocks | 2047 | 2047 | 2047 |
| Max. data blocks size | 16 KB | 16 KB | 16 KB |
| Max. local data size per execution level | 1024 Byte | 1024 Byte | 1024 Byte |
| **Blocks** | | | |
| Number of OBs | 14 | 14 | 14 |
| Maximum OB size | 16 KB | 16 KB | 16 KB |
| Number of FBs | 1024 | 1024 | 1024 |
| Maximum FB size | 16 KB | 16 KB | 16 KB |
| Number of FCs | 1024 | 1024 | 1024 |
| Maximum FC size | 16 KB | 16 KB | 16 KB |
| Maximum nesting depth per priority class | 8 | 8 | 8 |
| Maximum nesting depth additional within an error OB | 1 | 1 | 1 |
| **Time** | | | |
| Real-time clock buffered | ✓ | ✓ | ✓ |
| Clock buffered period (min.) | 30 d | 30 d | 30 d |
| Accuracy (max. deviation per day) | - | - | - |
| Number of operating hours counter | 8 | 8 | 8 |
| Value range operating hours counter | 32767 | 32767 | 32767 |
| Clock synchronization | - | - | - |
| **Address areas (I/O)** | | | |
| Input I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Input I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Input process image preset | 128 Byte | 128 Byte | 128 Byte |
| Output process image preset | 128 Byte | 128 Byte | 128 Byte |
| Digital inputs | 16 (20) | 16 (20) | 16 (20) |
| Digital outputs | 16 (12) | 16 (12) | 16 (12) |
| Integrated digital inputs | 16 (20) | 16 (20) | 16 (20) |
| Integrated digital outputs | 16 (12) | 16 (12) | 16 (12) |
| **Communication functions** | | | |
| PG/OP channel | ✓ | ✓ | ✓ |
| Global data communication | ✓ | ✓ | ✓ |
| Number of GD circuits, max. | 4 | 4 | 4 |
| Size of GD packets, max. | 22 Byte | 22 Byte | 22 Byte |
| S7 basic communication | ✓ | ✓ | ✓ |
| S7 basic communication, user data per job | 76 Byte | 76 Byte | 76 Byte |

| Order number | 115-6BL12 | 115-6BL13 | 115-6BL14 |
|---|---|---|---|
| S7 communication | ✓ | ✓ | ✓ |
| S7 communication as server | ✓ | ✓ | ✓ |
| S7 communication as client | - | - | - |
| S7 communication, user data per job | 160 Byte | 160 Byte | 160 Byte |
| Number of connections, max. | 16 | 16 | 16 |
| **Functionality Sub-D interfaces** | | | |
| Type | - | - | - |
| Type of interface | RS485 | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | - | - | - |
| MPI | ✓ | ✓ | ✓ |
| MP²I (MPI/RS232) | ✓ | ✓ | ✓ |
| DP master | - | - | - |
| DP slave | - | - | - |
| Point-to-point interface | - | - | - |
| | | | |
| Type | - | - | - |
| Type of interface | RS232 | RS232 | RS232 |
| Connector | Sub-D, 9-pin, male | Sub-D, 9-pin, male | Sub-D, 9-pin, male |
| Electrically isolated | - | - | - |
| MPI | - | - | - |
| MP²I (MPI/RS232) | - | - | - |
| DP master | - | - | - |
| DP slave | - | - | - |
| Point-to-point interface | ✓ | ✓ | ✓ |
| CAN | - | - | - |
| **Point-to-point communication** | | | |
| PtP communication | ✓ | ✓ | ✓ |
| Interface isolated | - | - | - |
| RS232 interface | ✓ | ✓ | ✓ |
| RS422 interface | - | - | - |
| RS485 interface | - | - | - |
| Connector | Sub-D, 9-pin, male | Sub-D, 9-pin, male | Sub-D, 9-pin, male |
| Transmission speed, min. | 150 bit/s | 150 bit/s | 150 bit/s |
| Transmission speed, max. | 115.2 kbit/s | 115.2 kbit/s | 115.2 kbit/s |
| Cable length, max. | 15 m | 15 m | 15 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 | - | - | - |
| **Mechanical data** | | | |
| Dimensions (WxHxD) | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm |
| Weight | 302 g | 302 g | 302 g |
| **Environmental conditions** | | | |
| Operating temperature | 0 °C to 60 °C | 0 °C to 60 °C | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C | -25 °C to 70 °C | -25 °C to 70 °C |
| **Certifications** | | | |
| UL508 certification | yes | yes | yes |

| **Order number** | **115-6BL32** | **115-6BL33** | **115-6BL34** |
|---|---|---|---|
| Type | CPU 115SER | CPU 115SER | CPU 115SER |
| **Technical data power supply** | | | |
| Power supply (rated value) | DC 24 V | DC 24 V | DC 24 V |
| Power supply (permitted range) | DC 20.4...28.8 V | DC 20.4...28.8 V | DC 20.4...28.8 V |
| Reverse polarity protection | ✓ | ✓ | ✓ |
| Current consumption (rated value) | 110 mA | 110 mA | 110 mA |
| **Technical data digital inputs** | | | |
| Number of inputs | 16 (20) | 16 (20) | 16 (20) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ | ✓ | ✓ |
| Current consumption from load voltage L+ (without load) | - | - | - |
| Rated value | DC 24 V | DC 24 V | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V | DC 0...5 V | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V | DC 15...28.8 V | DC 15...28.8 V |
| Input current for signal "1" | 7 mA | 7 mA | 7 mA |
| Connection of Two-Wire-BEROs possible | ✓ | - | - |
| Max. permissible BERO quiescent current | 1.5 mA | 1.5 mA | 1.5 mA |
| Input delay of "0" to "1" | 3 ms | 3 ms | 3 ms |
| Input delay of "1" to "0" | 3 ms | 3 ms | 3 ms |
| Input characteristic curve | IEC 61131, type 1 | IEC 61131, type 1 | IEC 61131, type 1 |
| Initial data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data digital outputs** | | | |
| Number of outputs | 16 (12) | 16 (12) | 16 (12) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | - | - | - |
| Current consumption from load voltage L+ (without load) | 50 mA | 50 mA | 50 mA |
| Output voltage signal "1" at min. current | L+ (-0.8 V) | L+ (-0.8 V) | L+ (-0.8 V) |
| Output voltage signal "1" at max. current | - | - | - |
| Output current at signal "1", rated value | 0.5 A | 0.5 A | 0.5 A |
| Output delay of "0" to "1" | max. 100 µs | max. 100 µs | max. 100 µs |
| Output delay of "1" to "0" | max. 350 µs | max. 350 µs | max. 350 µs |
| Minimum load current | - | - | - |
| Lamp load | 5 W | 5 W | 5 W |
| Switching frequency with resistive load | max. 1000 Hz | max. 1000 Hz | max. 1000 Hz |
| Switching frequency with inductive load | max. 0.5 Hz | max. 0.5 Hz | max. 0.5 Hz |
| Switching frequency on lamp load | max. 10 Hz | max. 10 Hz | max. 10 Hz |
| Internal limitation of inductive shut-off voltage | L+ (-52 V) | L+ (-52 V) | L+ (-52 V) |
| Short-circuit protection of output | yes, electronic | yes, electronic | yes, electronic |
| Trigger level | 1 A | 1 A | 1 A |
| Output data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data counters** | | | |
| Number of counters | 4 | 4 | 4 |
| Counter width | 32 Bit | 32 Bit | 32 Bit |
| Maximum input frequency | 30 kHz | 30 kHz | 30 kHz |
| Maximum count frequency | 30 kHz | 30 kHz | 30 kHz |
| Mode incremental encoder | ✓ | ✓ | ✓ |
| Mode pulse / direction | ✓ | ✓ | ✓ |
| Mode pulse | ✓ | ✓ | ✓ |
| Mode frequency counter | - | - | - |
| Mode period measurement | - | - | - |
| Gate input available | ✓ | ✓ | ✓ |
| Latch input available | - | - | - |
| Reset input available | - | - | - |
| Counter output available | - | - | - |
| **Load and working memory** | | | |
| Load memory, integrated | 24 KB | 32 KB | 40 KB |
| Work memory, integrated | 16 KB | 24 KB | 32 KB |
| Memory card slot | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB |
| **Hardware configuration** | | | |
| Racks, max. | 1 | 1 | 1 |
| Modules per rack, max. | 4 | 4 | 4 |
| Number of integrated DP master | - | - | - |
| Number of DP master via CP | 4 | 4 | 4 |

| Order number | 115-6BL32 | 115-6BL33 | 115-6BL34 |
|---|---|---|---|
| Operable function modules | 4 | 4 | 4 |
| Operable communication modules PtP | 4 | 4 | 4 |
| **Status information, alarms, diagnostics** | | | |
| Status display | yes | yes | yes |
| Interrupts | yes | yes | yes |
| Process alarm | yes | yes | yes |
| Diagnostic interrupt | yes | yes | yes |
| Diagnostic functions | no | no | no |
| Diagnostics information read-out | possible | possible | possible |
| Supply voltage display | green LED | green LED | green LED |
| Group error display | red SF LED | red SF LED | red SF LED |
| Channel error display | none | none | none |
| **Isolation** | | | |
| Between channels of groups to | 8 | 8 | 8 |
| Between channels and backplane bus | ✓ | ✓ | ✓ |
| Insulation tested with | DC 500 V | DC 500 V | DC 500 V |
| **Command processing times** | | | |
| Bit instructions, min. | 0.25 µs | 0.25 µs | 0.25 µs |
| Word instruction, min. | 1.2 µs | 1.2 µs | 1.2 µs |
| Double integer arithmetic, min. | - | - | - |
| Floating-point arithmetic, min. | - | - | - |
| **Timers/Counters and their retentive characteristics** | | | |
| Number of S7 counters | 256 | 256 | 256 |
| S7 counter remanence | adjustable 0 up to 64 | adjustable 0 up to 64 | adjustable 0 up to 64 |
| S7 counter remanence adjustable | C0 .. C7 | C0 .. C7 | C0 .. C7 |
| Number of S7 times | 256 | 256 | 256 |
| S7 times remanence | adjustable 0 up to 128 | adjustable 0 up to 128 | adjustable 0 up to 128 |
| S7 times remanence adjustable | not retentive | not retentive | not retentive |
| **Data range and retentive characteristic** | | | |
| Number of flags | 8192 Bit | 8192 Bit | 8192 Bit |
| Bit memories retentive characteristic adjustable | adjustable 0 up to 256 | adjustable 0 up to 256 | adjustable 0 up to 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 | MB0 .. MB15 | MB0 .. MB15 |
| Number of data blocks | 2047 | 2047 | 2047 |
| Max. data blocks size | 16 KB | 16 KB | 16 KB |
| Max. local data size per execution level | 1024 Byte | 1024 Byte | 1024 Byte |
| **Blocks** | | | |
| Number of OBs | 14 | 14 | 14 |
| Maximum OB size | 16 KB | 16 KB | 16 KB |
| Number of FBs | 1024 | 1024 | 1024 |
| Maximum FB size | 16 KB | 16 KB | 16 KB |
| Number of FCs | 1024 | 1024 | 1024 |
| Maximum FC size | 16 KB | 16 KB | 16 KB |
| Maximum nesting depth per priority class | 8 | 8 | 8 |
| Maximum nesting depth additional within an error OB | 1 | 1 | 1 |
| **Time** | | | |
| Real-time clock buffered | ✓ | ✓ | ✓ |
| Clock buffered period (min.) | 30 d | 30 d | 30 d |
| Accuracy (max. deviation per day) | - | - | - |
| Number of operating hours counter | 8 | 8 | 8 |
| Value range operating hours counter | 32767 | 32767 | 32767 |
| Clock synchronization | - | - | - |
| **Address areas (I/O)** | | | |
| Input I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Input I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Input process image preset | 128 Byte | 128 Byte | 128 Byte |
| Output process image preset | 128 Byte | 128 Byte | 128 Byte |
| Digital inputs | 16 (20) | 16 (20) | 16 (20) |
| Digital outputs | 16 (12) | 16 (12) | 16 (12) |
| Integrated digital inputs | 16 (20) | 16 (20) | 16 (20) |
| Integrated digital outputs | 16 (12) | 16 (12) | 16 (12) |
| **Communication functions** | | | |
| PG/OP channel | ✓ | ✓ | ✓ |
| Global data communication | ✓ | ✓ | ✓ |
| Number of GD circuits, max. | 4 | 4 | 4 |
| Size of GD packets, max. | 22 Byte | 22 Byte | 22 Byte |
| S7 basic communication | ✓ | ✓ | ✓ |
| S7 basic communication, user data per job | 76 Byte | 76 Byte | 76 Byte |
| S7 communication | ✓ | ✓ | ✓ |

| **Order number** | **115-6BL32** | **115-6BL33** | **115-6BL34** |
|---|---|---|---|
| S7 communication as server | ✓ | ✓ | ✓ |
| S7 communication as client | - | - | - |
| S7 communication, user data per job | 160 Byte | 160 Byte | 160 Byte |
| Number of connections, max. | 16 | 16 | 16 |
| **Functionality Sub-D interfaces** | | | |
| Type | - | - | - |
| Type of interface | RS485 | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | - | - | - |
| MPI | ✓ | ✓ | ✓ |
| MP²I (MPI/RS232) | ✓ | ✓ | ✓ |
| DP master | - | - | - |
| DP slave | - | - | - |
| Point-to-point interface | - | - | - |
| | | | |
| Type | - | - | - |
| Type of interface | RS485 | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | ✓ | ✓ | ✓ |
| MPI | - | - | - |
| MP²I (MPI/RS232) | - | - | - |
| DP master | - | - | - |
| DP slave | - | - | - |
| Point-to-point interface | ✓ | ✓ | ✓ |
| CAN | - | - | - |
| **Point-to-point communication** | | | |
| PtP communication | ✓ | ✓ | ✓ |
| Interface isolated | ✓ | ✓ | ✓ |
| RS232 interface | - | - | - |
| RS422 interface | - | - | - |
| RS485 interface | ✓ | ✓ | ✓ |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Transmission speed, min. | 150 bit/s | 150 bit/s | 150 bit/s |
| Transmission speed, max. | 115.2 kbit/s | 115.2 kbit/s | 115.2 kbit/s |
| Cable length, max. | 500 m | 500 m | 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 | - | - | - |
| **Mechanical data** | | | |
| Dimensions (WxHxD) | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm |
| Weight | 302 g | 302 g | 302 g |
| **Environmental conditions** | | | |
| Operating temperature | 0 °C to 60 °C | 0 °C to 60 °C | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C | -25 °C to 70 °C | -25 °C to 70 °C |
| **Certifications** | | | |
| UL508 certification | yes | yes | yes |

## CPU 11x PROFIBUS-DP slave

| Order number | 115-6BL22 | 115-6BL23 | 115-6BL24 |
|---|---|---|---|
| Type | CPU 115DP | CPU 115DP | CPU 115DP |
| **Technical data power supply** | | | |
| Power supply (rated value) | DC 24 V | DC 24 V | DC 24 V |
| Power supply (permitted range) | DC 20.4...28.8 V | DC 20.4...28.8 V | DC 20.4...28.8 V |
| Reverse polarity protection | ✓ | ✓ | ✓ |
| Current consumption (rated value) | 160 mA | 160 mA | 160 mA |
| **Technical data digital inputs** | | | |
| Number of inputs | 16 (20) | 16 (20) | 16 (20) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | ✓ | ✓ | ✓ |
| Current consumption from load voltage L+ (without load) | - | - | - |
| Rated value | DC 24 V | DC 24 V | DC 24 V |
| Input voltage for signal "0" | DC 0...5 V | DC 0...5 V | DC 0...5 V |
| Input voltage for signal "1" | DC 15...28.8 V | DC 15...28.8 V | DC 15...28.8 V |
| Input current for signal "1" | 7 mA | 7 mA | 7 mA |
| Connection of Two-Wire-BEROs possible | - | - | - |
| Max. permissible BERO quiescent current | 1.5 mA | 1.5 mA | 1.5 mA |
| Input delay of "0" to "1" | 3 ms | 3 ms | 3 ms |
| Input delay of "1" to "0" | 3 ms | 3 ms | 3 ms |
| Input characteristic curve | IEC 61131, type 1 | IEC 61131, type 1 | IEC 61131, type 1 |
| Initial data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data digital outputs** | | | |
| Number of outputs | 16 (12) | 16 (12) | 16 (12) |
| Cable length, shielded | 1000 m | 1000 m | 1000 m |
| Cable length, unshielded | 600 m | 600 m | 600 m |
| Rated load voltage | DC 24 V | DC 24 V | DC 24 V |
| Reverse polarity protection of rated load voltage | - | - | - |
| Current consumption from load voltage L+ (without load) | 50 mA | 50 mA | 50 mA |
| Output voltage signal "1" at min. current | L+ (-0.8 V) | L+ (-0.8 V) | L+ (-0.8 V) |
| Output voltage signal "1" at max. current | - | - | - |
| Output current at signal "1", rated value | 0.5 A | 0.5 A | 0.5 A |
| Output delay of "0" to "1" | max. 100 µs | max. 100 µs | max. 100 µs |
| Output delay of "1" to "0" | max. 350 µs | max. 350 µs | max. 350 µs |
| Minimum load current | - | - | - |
| Lamp load | 5 W | 5 W | 5 W |
| Switching frequency with resistive load | max. 1000 Hz | max. 1000 Hz | max. 1000 Hz |
| Switching frequency with inductive load | max. 0.5 Hz | max. 0.5 Hz | max. 0.5 Hz |
| Switching frequency on lamp load | max. 10 Hz | max. 10 Hz | max. 10 Hz |
| Internal limitation of inductive shut-off voltage | L+ (-52 V) | L+ (-52 V) | L+ (-52 V) |
| Short-circuit protection of output | yes, electronic | yes, electronic | yes, electronic |
| Trigger level | 1 A | 1 A | 1 A |
| Output data size | 3 Byte | 3 Byte | 3 Byte |
| **Technical data counters** | | | |
| Number of counters | 4 | 4 | 4 |
| Counter width | 32 Bit | 32 Bit | 32 Bit |
| Maximum input frequency | 30 kHz | 30 kHz | 30 kHz |
| Maximum count frequency | 30 kHz | 30 kHz | 30 kHz |
| Mode incremental encoder | ✓ | ✓ | ✓ |
| Mode pulse / direction | ✓ | ✓ | ✓ |
| Mode pulse | ✓ | ✓ | ✓ |
| Mode frequency counter | - | - | - |
| Mode period measurement | - | - | - |
| Gate input available | ✓ | ✓ | ✓ |
| Latch input available | - | - | - |
| Reset input available | - | - | - |
| Counter output available | - | - | - |
| **Load and working memory** | | | |
| Load memory, integrated | 24 KB | 32 KB | 40 KB |
| Work memory, integrated | 16 KB | 24 KB | 32 KB |
| Memory card slot | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB | MMC-Card with max. 512 MB |
| **Hardware configuration** | | | |
| Racks, max. | 1 | 1 | 1 |

| **Order number** | **115-6BL22** | **115-6BL23** | **115-6BL24** |
|---|---|---|---|
| Modules per rack, max. | 4 | 4 | 4 |
| Number of integrated DP master | - | - | - |
| Number of DP master via CP | 4 | 4 | 4 |
| Operable function modules | 4 | 4 | 4 |
| Operable communication modules PtP | 4 | 4 | 4 |
| **Status information, alarms, diagnostics** | | | |
| Status display | yes | yes | yes |
| Interrupts | yes | yes | yes |
| Process alarm | yes | yes | yes |
| Diagnostic interrupt | yes | yes | yes |
| Diagnostic functions | no | no | no |
| Diagnostics information read-out | possible | possible | possible |
| Supply voltage display | green LED | green LED | green LED |
| Group error display | red SF LED | red SF LED | red SF LED |
| Channel error display | none | none | none |
| **Isolation** | | | |
| Between channels of groups to | 8 | 8 | 8 |
| Between channels and backplane bus | ✓ | ✓ | ✓ |
| Insulation tested with | DC 500 V | DC 500 V | DC 500 V |
| **Command processing times** | | | |
| Bit instructions, min. | 0.25 µs | 0.25 µs | 0.25 µs |
| Word instruction, min. | 1.2 µs | 1.2 µs | 1.2 µs |
| Double integer arithmetic, min. | - | - | - |
| Floating-point arithmetic, min. | - | - | - |
| **Timers/Counters and their retentive characteristics** | | | |
| Number of S7 counters | 256 | 256 | 256 |
| S7 counter remanence | adjustable 0 up to 64 | adjustable 0 up to 64 | adjustable 0 up to 64 |
| S7 counter remanence adjustable | C0 .. C7 | C0 .. C7 | C0 .. C7 |
| Number of S7 times | 256 | 256 | 256 |
| S7 times remanence | adjustable 0 up to 128 | adjustable 0 up to 128 | adjustable 0 up to 128 |
| S7 times remanence adjustable | not retentive | not retentive | not retentive |
| **Data range and retentive characteristic** | | | |
| Number of flags | 8192 Bit | 8192 Bit | 8192 Bit |
| Bit memories retentive characteristic adjustable | adjustable 0 up to 256 | adjustable 0 up to 256 | adjustable 0 up to 256 |
| Bit memories retentive characteristic preset | MB0 .. MB15 | MB0 .. MB15 | MB0 .. MB15 |
| Number of data blocks | 2047 | 2047 | 2047 |
| Max. data blocks size | 16 KB | 16 KB | 16 KB |
| Max. local data size per execution level | 1024 Byte | 1024 Byte | 1024 Byte |
| **Blocks** | | | |
| Number of OBs | 14 | 14 | 14 |
| Maximum OB size | 16 KB | 16 KB | 16 KB |
| Number of FBs | 1024 | 1024 | 1024 |
| Maximum FB size | 16 KB | 16 KB | 16 KB |
| Number of FCs | 1024 | 1024 | 1024 |
| Maximum FC size | 16 KB | 16 KB | 16 KB |
| Maximum nesting depth per priority class | 8 | 8 | 8 |
| Maximum nesting depth additional within an error OB | 1 | 1 | 1 |
| **Time** | | | |
| Real-time clock buffered | ✓ | ✓ | ✓ |
| Clock buffered period (min.) | 30 d | 30 d | 30 d |
| Accuracy (max. deviation per day) | - | - | - |
| Number of operating hours counter | 8 | 8 | 8 |
| Value range operating hours counter | 32767 | 32767 | 32767 |
| Clock synchronization | - | - | - |
| **Address areas (I/O)** | | | |
| Input I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area | 1024 Byte | 1024 Byte | 1024 Byte |
| Input I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Output I/O address area, decentral | 1024 Byte | 1024 Byte | 1024 Byte |
| Input process image preset | 128 Byte | 128 Byte | 128 Byte |
| Output process image preset | 128 Byte | 128 Byte | 128 Byte |
| Digital inputs | 16 (20) | 16 (20) | 16 (20) |
| Digital outputs | 16 (12) | 16 (12) | 16 (12) |
| Integrated digital inputs | 16 (20) | 16 (20) | 16 (20) |
| Integrated digital outputs | 16 (12) | 16 (12) | 16 (12) |
| **Communication functions** | | | |
| PG/OP channel | ✓ | ✓ | ✓ |
| Global data communication | ✓ | ✓ | ✓ |
| Number of GD circuits, max. | 4 | 4 | 4 |
| Size of GD packets, max. | 22 Byte | 22 Byte | 22 Byte |

| Order number | 115-6BL22 | 115-6BL23 | 115-6BL24 |
|---|---|---|---|
| S7 basic communication | ✓ | ✓ | ✓ |
| S7 basic communication, user data per job | 76 Byte | 76 Byte | 76 Byte |
| S7 communication | ✓ | ✓ | ✓ |
| S7 communication as server | ✓ | ✓ | ✓ |
| S7 communication as client | - | - | - |
| S7 communication, user data per job | 160 Byte | 160 Byte | 160 Byte |
| Number of connections, max. | 16 | 16 | 16 |
| **Functionality Sub-D interfaces** | | | |
| Type | - | - | - |
| Type of interface | RS485 | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | - | - | - |
| MPI | ✓ | ✓ | ✓ |
| MP²I (MPI/RS232) | ✓ | ✓ | ✓ |
| DP master | - | - | - |
| DP slave | - | - | - |
| Point-to-point interface | - | - | - |
| | | | |
| Type | - | - | - |
| Type of interface | RS485 | RS485 | RS485 |
| Connector | Sub-D, 9-pin, female | Sub-D, 9-pin, female | Sub-D, 9-pin, female |
| Electrically isolated | ✓ | ✓ | ✓ |
| MPI | - | - | - |
| MP²I (MPI/RS232) | - | - | - |
| DP master | - | - | - |
| DP slave | ✓ | ✓ | ✓ |
| Point-to-point interface | - | - | - |
| CAN | - | - | - |
| **Functionality PROFIBUS slave** | | | |
| PG/OP channel | - | - | - |
| Routing | - | - | - |
| S7 communication | - | - | - |
| S7 communication as server | - | - | - |
| S7 communication as client | - | - | - |
| Direct data exchange (slave-to-slave communication) | - | - | - |
| DPV1 | - | - | - |
| Transmission speed, min. | 9.6 kbit/s | 9.6 kbit/s | 9.6 kbit/s |
| Transmission speed, max. | 12 Mbit/s | 12 Mbit/s | 12 Mbit/s |
| Automatic detection of transmission speed | - | - | - |
| Transfer memory inputs, max. | 64 Byte | 64 Byte | 64 Byte |
| Transfer memory outputs, max. | 64 Byte | 64 Byte | 64 Byte |
| Address areas, max. | 1 | 1 | 1 |
| User data per address area, max. | 64 Byte | 64 Byte | 64 Byte |
| **Mechanical data** | | | |
| Dimensions (WxHxD) | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm | 152.4 x 76 x 48 mm |
| Weight | 330 g | 330 g | 330 g |
| **Environmental conditions** | | | |
| Operating temperature | 0 °C to 60 °C | 0 °C to 60 °C | 0 °C to 60 °C |
| Storage temperature | -25 °C to 70 °C | -25 °C to 70 °C | -25 °C to 70 °C |
| **Certifications** | | | |
| UL508 certification | yes | yes | yes |

# Chapter 3 Deployment Micro-PLC CPU 11x

**Overview**

At the beginning of the chapter you get information about Installation and Commissioning of the System 100V. The chapter is continued by the addressing and the address areas, that are occupied by the System 100V per default, followed by the approach at the project engineering and parameterization of the CPU.

Another part is the description of the operating modes, the overall reset, the firmware update, the employment of the MMC and the MPI slot.

The chapter closes with VIPA specific diagnostics and the test functions "Control and monitor variable".

**Content**

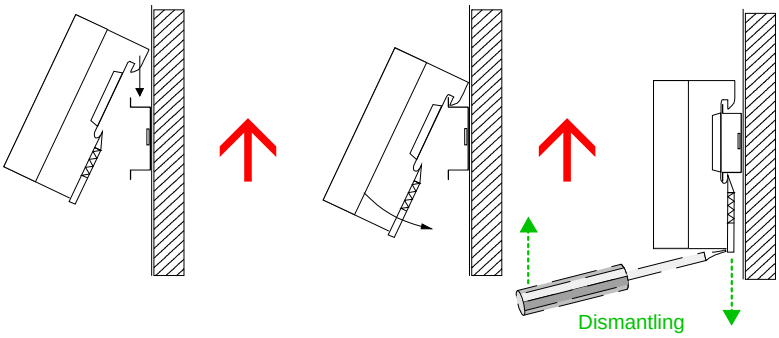
# Installation and Commissioning

**Checklist for commissioning**

- Turn off your power supply.
- Build up your system.
- Cable your system.
- Turn on your power supply.
- Request an Overall reset.

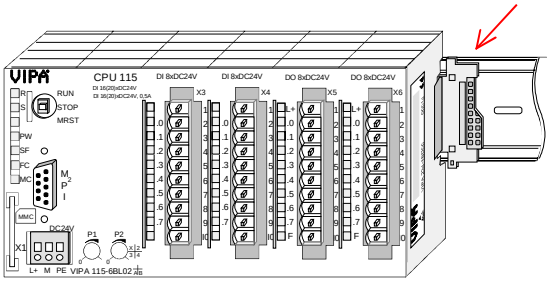
**Installation/ Dismantling**

System 100V modules are clipped to 35mm standard norm profile rails.




For *mounting*, you set the module onto the head rail from above, using an angle of 45°. Rotate the module down until it clips to the rail with a hearable click.

For the *dismantling* you have to pull down the locker with a screwdriver and lift the module from the head rail.

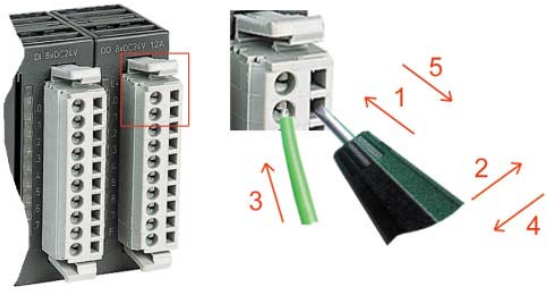



Every expansion module includes a 1tier bus connector. When using expansion modules you have to plug this to the right backside before assembling the module to the system.

**Cabling**

Take a fitting screwdriver and push the cage clamp in the rectangular opening to the back, then insert the cable into the round opening.

The cage clamp locks securely by removing the screwdriver.



# Start-up behavior

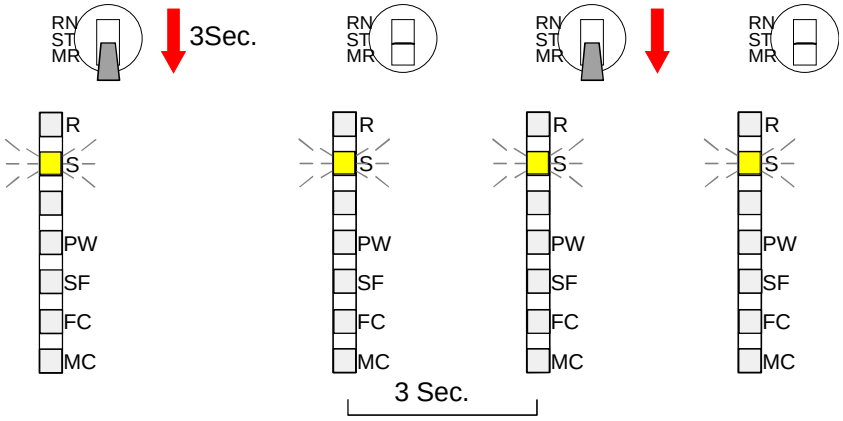
**Turn on power supply**

After you turned on the power supply, the CPU switches to the operating mode that has been selected at the operating mode switch.

Now you may transfer your project from your projecting tool into the CPU via MPI res. Plug-in a MMC containing your project and request an Overall reset.

**Overall reset**

The following picture shows the approach:




**Note!**

The transfer of the user application from the MMC into the CPU takes always place after an Overall reset!

**Default boot procedure, as delivered**

When the CPU is delivered it has been reset.

After a STOP→RUN transition the CPU switches to RUN without program.

**Boot procedure with valid CPU data**

The CPU switches to RUN with the program stored in the battery buffered RAM.

**Start-up with empty accu**

The accu is loaded directly via the integrated power supply by means of a load electronic and guarantees a buffer of ca. 30 days. If this time is exceeded, the accu may be totally discharged and the battery buffered RAM is erased.

Now the CPU executes an Overall reset. If a MMC is plugged-in, the program on the MMC is transferred into the RAM. Otherwise a stored program of the internal flash memory is transferred to RAM.

This procedure is fixed in the diagnostic buffer with this entry: "Automatic start overall reset (unbuffered Power ON)".

The CPU stops after a start-up with empty accu.

# Principles of the address allocation

**Overview**

At the start-up of the CPU, the input and output sections are automatically linked up to the address area of the CPU starting at address 0.

Input and output section is each occupying 3byte. The address from where on the input res. output data is stored may be altered in your projecting tool.

The address allocation of the input/output periphery takes place in the Siemens SIMATIC manager as a virtual PROFIBUS system. For the PROFIBUS interface is standardized also software sided, the functionality is guaranteed by including a GSD-file into the Siemens SIMATIC manager.

Transfer your project into the CPU via a serial connection to the MPI interface.

**Note!**

The configuration of the CPU requires a thorough knowledge of the SIMATIC manager and the hardware configurator from Siemens!

**Automatic addressing**

To provide specific addressing of the input and output areas, certain addresses have to be assigned in the CPU.

The CPU contains a peripheral area (addresses 0 ... 1023) and a process image of the inputs and the outputs (for every address 0 ... 127).

When the CPU is initialized it automatically assigns 3 addresses to the input area and 3 addresses to the output area, starting from 0.

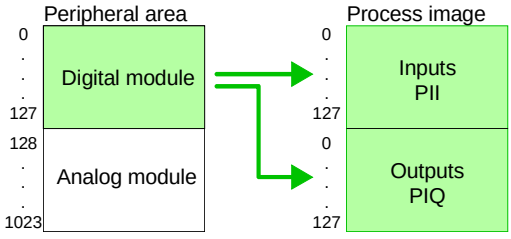
**Signal states in the process image**

The signal states of the lower addresses (0 ... 127) are saved in a special memory area called the *process image*.

After every cycle the process image is updated.

The process image is divided into two parts:

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

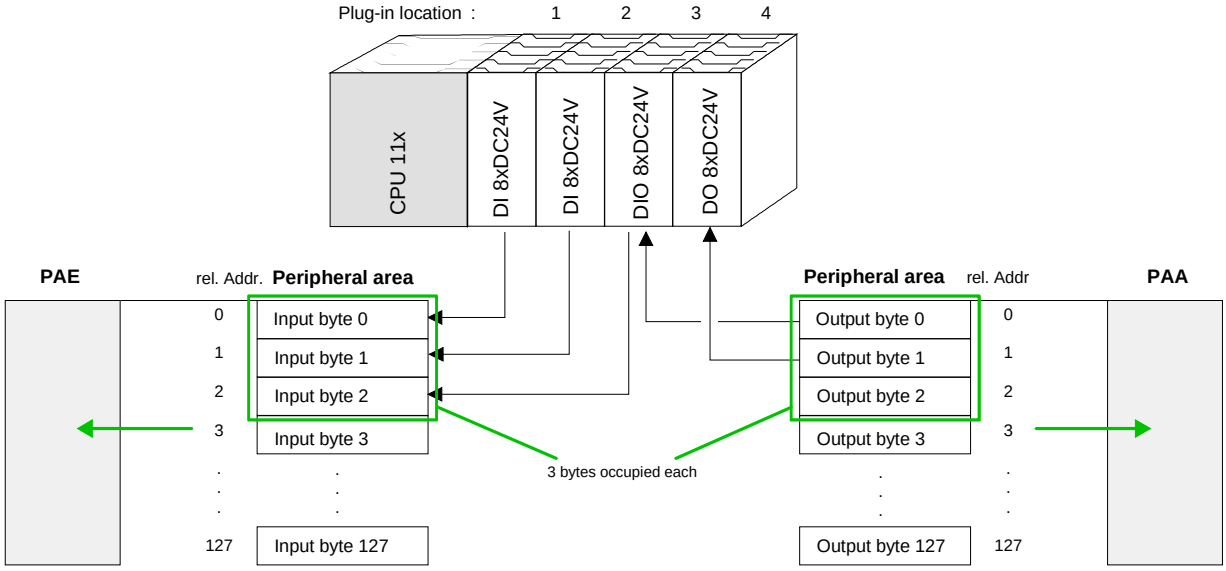



**Read/write access**

You access the modules by means of read or write operations on the peripheral bytes or on the process image.

**Example for the automatic address allocation**

The following figure illustrates the automatic allocation of addresses:




**Note!**

Please regard that you may access different modules by means of read and write operations on the same address.

**Default address allocation of the CPU 11x**

If there hasn't taken place a hardware configuration yet, the following addresses in the CPU 11x are occupied:

| Address allocation Input area | Function |
|---|---|
| 0...2 | DI |
| 3...127 | **free for more DI** |
| 128, 129 | Potentiometer P1 |
| 130, 131 | Potentiometer P2 |
| 132...135 | reserved |
| 136...139 | Counter 0 |
| 140...143 | Counter 1 |
| 144...147 | Counter 2 |
| 148...151 | Counter 3 |
| 152...1021 | **free for more AI** |

| Address allocation Output area | Function |
|---|---|
| 0...2 | DO |
| 3...127 | **free for more DO** |
| 128...1021 | **free for more AO** |

**Change address allocation via configuration**

Using the Siemens SIMATIC manager you may change the address allocation at any time and put in-/output areas into the process image area (0...127).

# Fast introduction project engineering

**Overview**

The address allocation, parameterization and PROFIBUS-DP project engineering takes place in the Siemens SIMATIC manager as a virtual PROFIBUS system. For the PROFIBUS interface is also standardized in software, we are able to guarantee the full functionality with the Siemens SIMATIC manager by including a GSD file.

Your project is transferred into your CPU via the $MP^2$interface.

**Requirements**

- Siemens SIMATIC manager installed on PC res. PG
- GSD files have been included in the hardware configurator
- serial connection to the CPU (e.g. via the "Green Cable" from VIPA)

**Note!**

For the project engineering a thorough knowledge of the Siemens SIMATIC manager and the hardware configurator from Siemens are required!

**Compatibility to Siemens SIMATIC manager via GSD-file**

The project engineering of a CPU 11x takes place in the Siemens SIMATIC manager in form of a virtual PROFIBUS system based on the CPU 315-2DP.

Due to the software standardized PROFIBUS interface, we are able to guarantee the full functionality of the System 100V family with Siemens SIMATIC manager by including a GSD file.

**To be compatible with the Siemens SIMATIC manager, you have to execute the following steps:**

- **Project the PROFIBUS-DP master system with CPU 315-2DP (6ES7 315-2AF03). Please use for the project engineering of the CPUs starting from Firmware V. 3.5.0 the CPU 6ES7-315-2AF03 V1.2 from Siemens.**
- **Insert a PROFIBUS slave with address 1.**
- **Place your CPU 11x at slot 0 of the slave system.**

**Project engineering as virtual PROFIBUS master system**

- Create a new project System 300 in the Siemens SIMATIC manager and add a profile rail from the hardware catalog.
- You'll find the CPU with PROFIBUS master in the hardware catalog at: `Simatic300/CPU-300/CPU315-2DP/`**`6ES7 315-2AF03-0AB0.`**
- Insert the CPU 315-2DP (**6ES7 315-2AF03-0AB0 V1.2**).
- Assign a PROFIBUS address (except 1) to your master.
- Click on "DP", select the operation mode "DP master" at *Object properties* and confirm your entry with OK.
- With a right-click on "DP" a context menu opens. Choose "Add master system". Create a new PROFIBUS subnet via NEW.

**Project CPU 11x**

You have to include the CPU section explicitly.

- Add the system "VIPA_CPU11x" to the subnet. This can be found in the hardware catalog at *PROFIBUS DP > Additional field devices > IO >* VIPA_System_100V. Assign the PROFIBUS address 1 to this slave (VIPA_11x.GSD required).
- Place your System 100V CPU at slot 0 in the configurator like e.g. 115-6BL02.
  **Slot 0 is mandatory!**
  The address areas of the in-/output periphery are created and may be altered at any time.
- Save your project.

**Project CPU 11xDP**

For connecting to a DP master system the following steps for the System 100V are necessary:

- Engineer the CPU 315-2DP with DP master system project (address 2).
- Add the PROFIBUS slave "VIPA_CPU11x" with address 1 (VIPA_11x.GSD required)
- Include the CPU-Type **11xDP** at slot 0 of the slave system.
- Choose the PROFIBUS parameters of the CPU 11xDP.
- Select the parameters of the in-/output periphery.
- Transfer the project engineering into the CPU 11xDP via MPI.

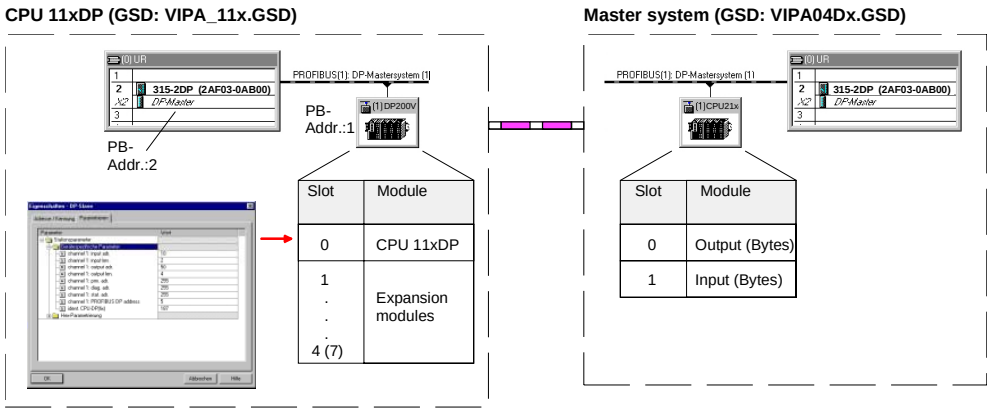
**Steps of the master project engineering**

The master side requires the following steps:

- Engineer the CPU with DP master system (address 2).
- Add the PROFIBUS slave "VIPA_CPU11xDP" (VIPA04Dx.GSD required).
- Type the PROFIBUS in- and output areas starting with slot 0 in Byte res. Words.

**Relation between master and slave**

The following illustration summarizes the project engineering at the slave and the master:




**Attention!**

The length entries for the input and output area have to be congruent with the Byte entry at the master project engineering. Otherwise no PROFIBUS communication is possible (slave failure).

**Note!**

If your DP master system is a System 200V from VIPA, you may parameterize the directly connected modules by including a "DP100V" slave system.

To enable the VIPA-CPU to recognize the project as central system, you have to assign the PROFIBUS address 1 to the "DP100V" slave system!

**At the deployment of the IM 208 PROFIBUS-DP master, please make sure that this has a firmware version V3.0 or higher; otherwise it is not deployable at a CPU 11x with firmware version >V3.0. The firmware versions are to find on the label at the backside of the module.**

On the following pages a closer description of project engineering and configuration of your System 100V can be found.

# Conditions for the project engineering Micro-PLC CPU 11x

**General**

To make the in-/output periphery addressable, you have to assign certain addresses in the CPU.

The project engineering and the address allocation takes place in the Siemens SIMATIC manager as a virtual PROFIBUS system. For the PROFIBUS interface is standardized also software sided, the functionality is guaranteed by including a GSD-file into the Siemens SIMATIC manager.

Transfer your project into the CPU via a serial connection to the MPI interface.

**Conditions**

For the project engineering of your Micro-PLC the following requirements have to be fulfilled:

- Siemens SIMATIC manager is installed on PC res. PG.
- The GSD-file is included to the hardware configurator from Siemens.
- Serial connection to the CPU (e.g. "Green Cable" from VIPA).

**Note!**

The configuration of the CPU requires a thorough knowledge of the Siemens SIMATIC manager and the hardware configurator from Siemens!

**Installation of the Siemens hardware configurator**

The hardware configurator is a component of the Siemens SIMATIC manager. A list of modules that can be configured by this tool can be obtained from the hardware catalog.

Before the PROFIBUS-DP slaves of the System 100V are ready for usage, the modules have to be included in the hardware catalog by means of the VIPA GSD-file.

**Including the GSD file**

- Copy the VIPA GSD-file into **`VIPA_11x.GSD`** your GSD directory ... \siemens\step7\s7data\gsd.
- Start the Siemens hardware configurator.
- Close all projects.
- Go to **Options** > *Install New GSD.*
- Enter **VIPA_11x.GSD**.
- Refresh the hardware catalog via **Options** > *update catalog.*

Now the modules of the VIPA System 100V have been integrated into the hardware catalog and are available for configuration.

# Project engineering Micro-PLC CPU 11x

**Configuration as virtual PROFIBUS master system**

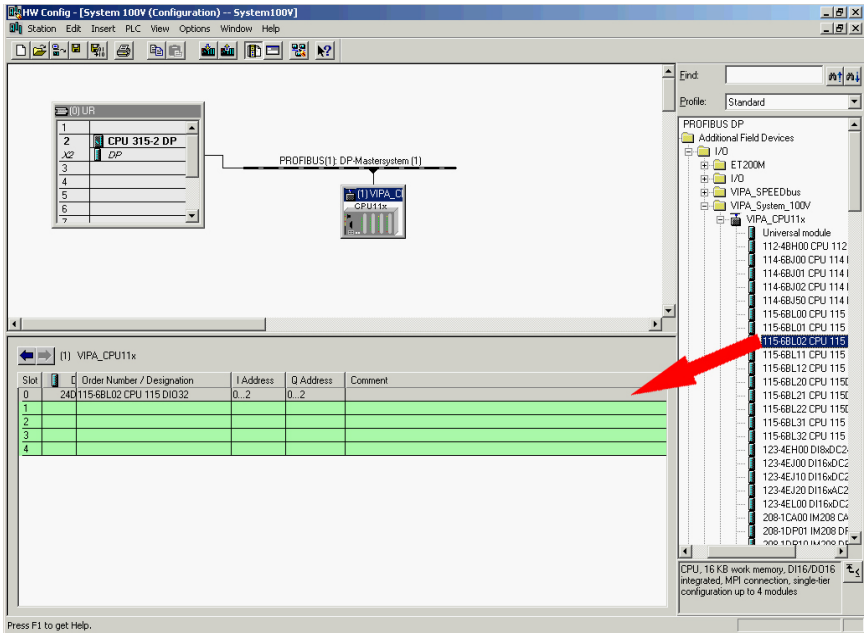
To be compatible with the Siemens SIMATIC manager, you have to configure the Micro-PLC CPU 11x as a virtual PROFIBUS system following these steps:

- Create a new project System 300.
- Include a profile rail from the hardware catalog.
- You find the CPU with PROFIBUS master in the hardware catalog under:
  `Simatic300/CPU-300/CPU315-2DP/`**`6ES7 315-2AF03-0AB0`**
- Add the CPU 315-2DP (**6ES7 315-2AF03-0AB0**).
- Assign a PROFIBUS address for your master (except 1).
- Click on DP and choose the operating mode "DP master" in the *object properties* and confirm with OK.
- Via a click on "DP" with the right mouse button, the context menu opens. Choose "Insert master system". Create a new PROFIBUS subnet via NEW.

**Configuration Micro-PLC**

As said before, you have to explicitly include the CPU section, to be compatible with the Siemens SIMATIC manager.

- Attach the system "VIPA_CPU11x" to the subnet. The respective entries are located in the hardware catalog under *PROFIBUS DP > Additional Field Devices > IO >* VIPA_System_100V. Assign the PROFIBUS address 1 to this slave (VIPA_11x.GSD required).
- Place your System 100V CPU, e.g. 115-6BL02, in the configurator at slot 0.




**Slot 0 is mandatory!**

The address areas of the in-/output periphery are created and may be changed at any time.

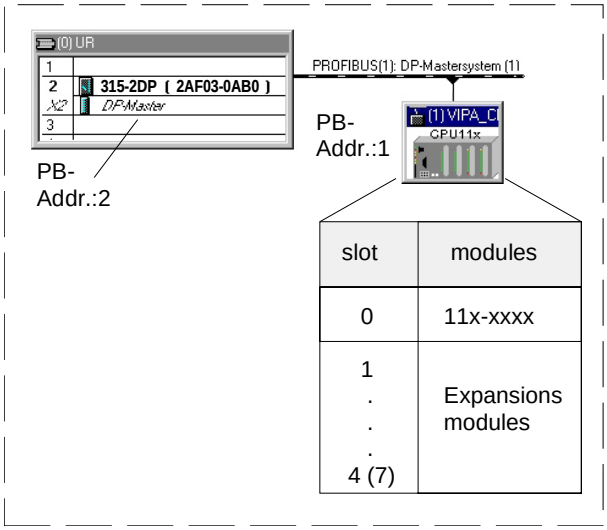
- Save your project.

**Configuration expansion and System 200V modules**

Via slot 1...4 you may include some further modules into your System 100V. With the Micro-PLC CPU with order number VIPA 115-6BL72 maximum 7 modules may be connected.

Choose the wanted module in the hardware catalog from Siemens and place it on the according slot.

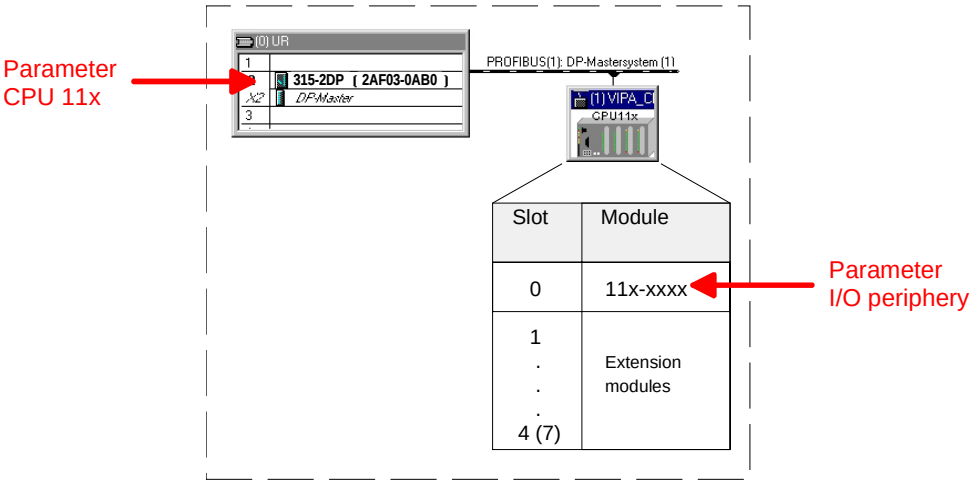
**Micro-PLC 11x**




**Configuration**

The CPU part is configured by the *Properties* of the Siemens CPU 315-2DP. The I/O periphery is configured in the virtual PROFIBUS system by means of the *Properties* of the CPU 11x.

**Mikro-SPS 11x**

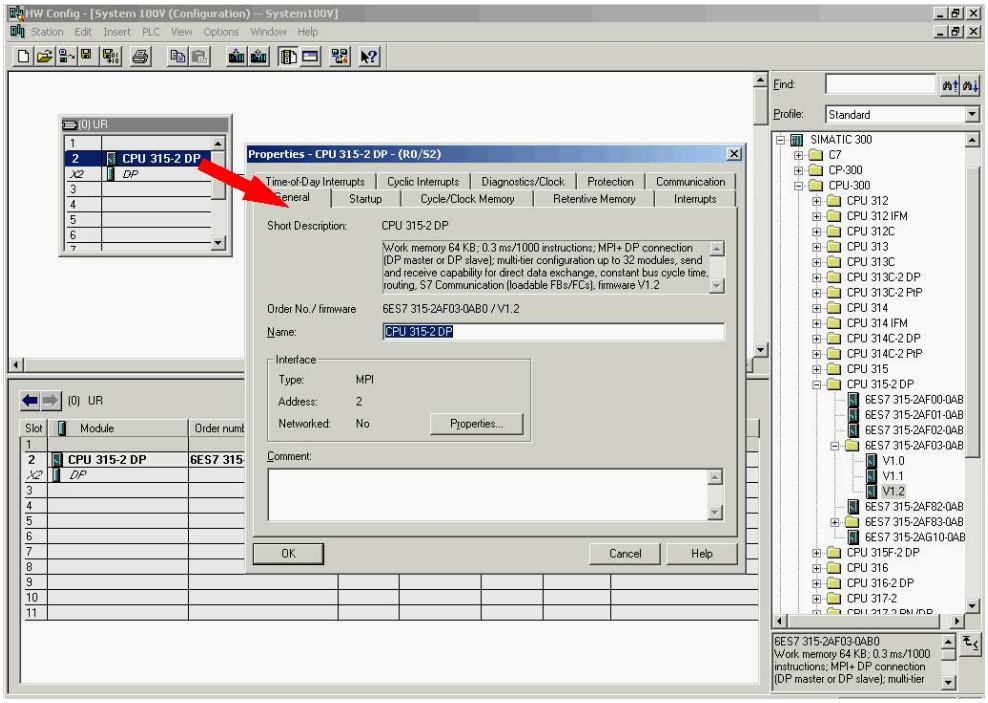



The possibility for configuration is described at the following pages.

# Parameter adjustment System 100V CPU

**Overview**

The general parameters concerning the CPU section of your System 100V have to be configured in the hardware configurator from Siemens under the properties of the CPU 315-2DP.




**Approach**

With a double click at the CPU 315-2DP you reach the parameterization window for your CPU. Via the register tabs you may access all parameters of your System 100V CPU.

Please regard, that at this time not all parameters are supported.

**Supported parameters**

The CPU doesn't evaluate all parameters that you may parameterize in your projecting tool. The following parameters are evaluated by the CPU at this time:

*General:*

MPI address of the CPU
baudrate (19.2kBaud, 187kBaud)
maximum MPI address

*Start-up:*

Start-up at scheduled configuration not equal...
Ready message from module
Transfer of parameters to...

*Remanence:*

No. of bit memory bytes from MB0
Number of S7-Timer from T0
Number of S7-Counter from Z0

*Protection:*

Protection level via password ...

*Time alarm:*

OB10: Execution
Active
Start date
Time-of-day

*Prompter alarm:*

OB35: Execution

*Cycle / pulse marker:*

Cycle watching time
Cycle load due to communication
OB85 call at periphery access error
Timing flags with marker byte no.

# Parameter adjustment System 100V periphery

**Overview**

The Micro-PLC CPU 11x has different parameters, that you may parameterize in the hardware configurator from Siemens via the concerning CPU-"properties".

- The adjustments, concerning the CPU may be found at the properties of the CPU 315-2 DP.
- Adjustments concerning the I/O periphery are to find under the "Properties" of the System 100V CPU like e.g. the 115-6BL02.

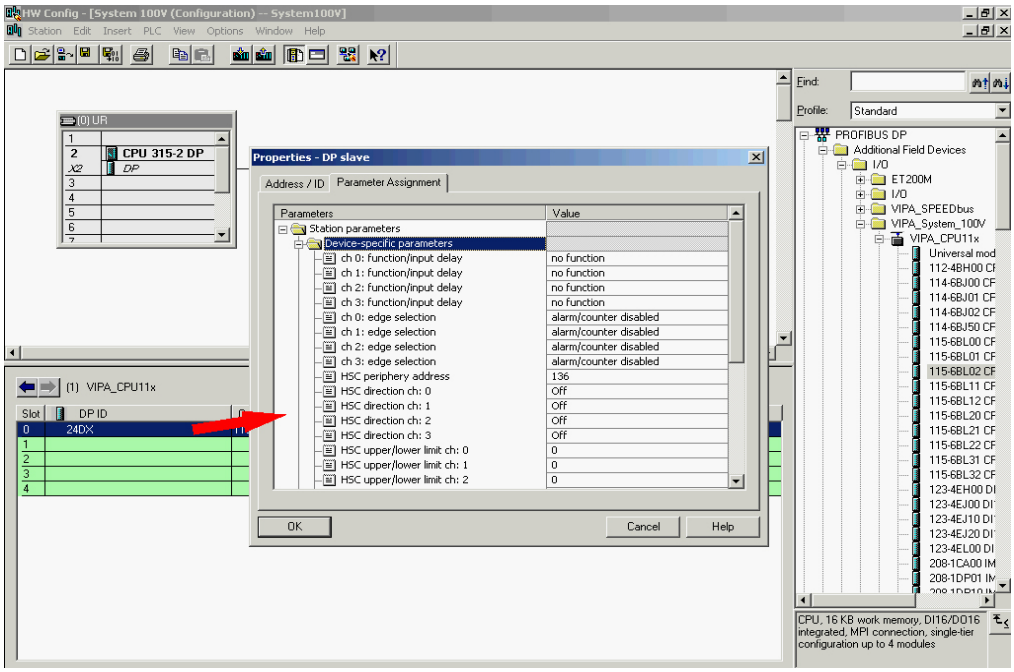
The parameterization of your System 100V I/O periphery shall be described here.

**Approach**

For parameterization you click on the "VIPA_CPU11x" PROFIBUS slave inserted before. At the according slot your System 100V CPU is shown.

Via double click on the System 100V CPU, you reach the dialog window "Properties DP-Slave".

Via the registers you have access to all parameters of the Micro-PLC CPU 11x, which are described in the following:




All parameter are described in the following:

**Address/Code**

**Output/Input**

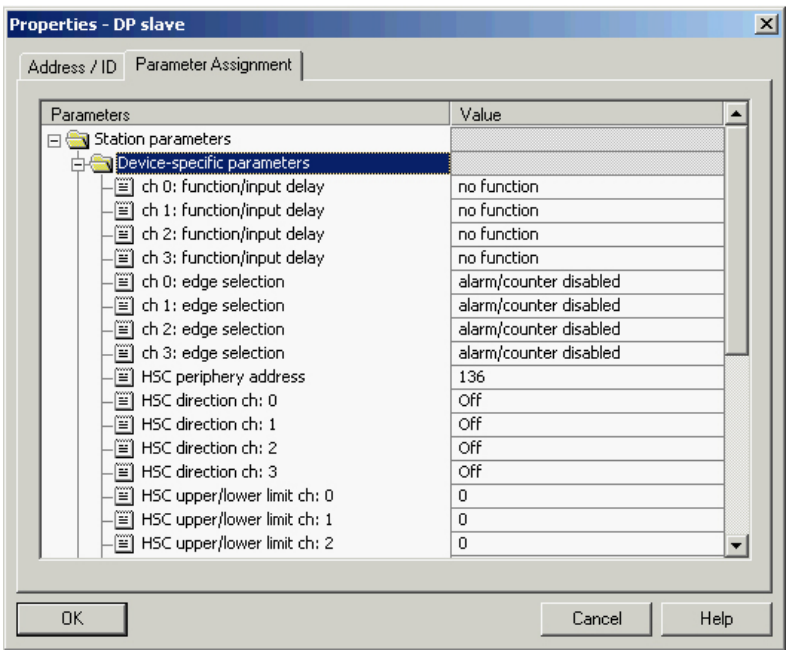
Input and output area is each occupying 3byte in the address area of the CPU. Please type the according start address, from where on the 3byte shall be stored.

Value range: 0 ... 125

**Parameterization**

The following parameterization is possible:

- Counter/Alarm behavior of the first 4 inputs
- Address assignment of the potentiometers P1 and P2
- PWM output behavior of the last 2 outputs at X5




In the following the parameters are listed:

**Ch x: Function/ input delay**

Here you may activate and deactivate the counter res. alarm functions for each channel x. Possible functions:

- disabled
- alarm: 0.1ms input delay
- alarm: 0.5ms input delay
- alarm: 3ms input delay
- alarm: 15ms input delay
- counter: pulse
- counter: pulse with direction
- counter: rotary encoder single
- counter: rotary encoder double
- counter: rotary encoder quadruple
- counter: pulse with HW gate

**Ch x: Edge selection**

Via this parameter is fixed if there should be a reaction after ascending or descending edges.

**Counter: Periphery address**

Please type the start address from where on the content of the 4 counters shall be stored. The length is 16byte.

Value range: 0 ... 1008

Default: 136 (Counter 0) ... 151 (Counter 3)

**Counter x: Direction**

This parameter describes the counting direction if the counter is activated.

**Counter x: Upper/ lower limit**

By fixing an upper res. lower limit, you may realize a counter that will initiate an alarm when reaching a predefined limit value (if wanted), sets itself back and starts counting again.

**Alarm type**

Activate the process alarm that is started as soon as a limit value is reached. You can parameterize the following alarm types:

- Process alarm
- Process+diagnostics alarm

**Note!**

For software-technical reasons attention should be paid that the delay time for all 4 alarms is configured equal.

**AI periphery address (P1, P2)**

At the front side of the System 100V you may see the potentiometers P1 and P2. You are able to predefine values between 0 and 1023 that are stored in the periphery area of the CPU.

Per default the values from P1 are stored at 128, 129 and the values from P2 at 130, 131 in row.

If needed, you may also assign another start address for this range by typing the wanted address in "AI periphery address".

Value range: 0 ... 1020

**PWM parameter**

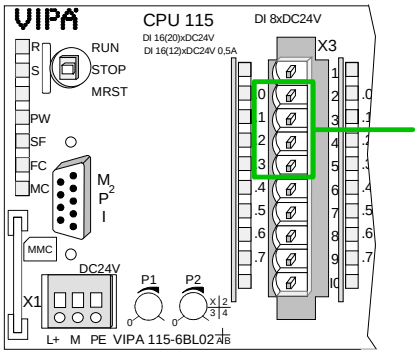
Depending on the chosen PWM mode, you may parameterize the time parameters for the pulse width modulation. A more detailed description of the PWM parameters is to be found at "Deployment PWM".

# Deployment counter and alarm input

**Overview**

Depending on the CPU 11x the first 4 inputs of X3 may be configured as counter respectively as alarm input. The properties and the behavior of the inputs are defined at the hardware configurator of the Siemens SIMATIC manager by means of the CPU parameter of the CPU 11x. These functions are deactivated in delivery state.

There is also the possibility to change the counter parameter at run-time by means of the VIPA SFC 224. More details about this may be found in the Manual "VIPA Operation List Standard" (HB00_OPL_STD).




Counter respectively alarm inputs

**Counter inputs**

Via the parameter *Ch x: Function/ input delay* the setting "Counter ..." allows you to control up to 4 counter with a frequency of up to 30kHz via the 4 inputs. In addition an alarm output at limit value overrun can be configured.

The following counter modes are at the disposal:

Counter: pulse

Counter: pulse with direction

Counter: encoder single

Counter: encoder double

Counter: encoder quadruple

Counter: pulse with hw-gate

**Counter: pulse**

1 input is occupied and it is counted in the configured direction with every pulse. 4 counter are available with this function.

**Counter: pulse with direction**

2 inputs are occupied and it is counted in the direction given by the 2nd input with each pulse at the 1st input. In this functionality maximally 2 counters are available. Here the polarity of the direction can be affected by means of the parameter *Direction*.

Direction "up"

Counts up by low and down by high level at the *Direction* input.

Direction "down"

Counts up by high and down by low level at the *Direction* input.

**Counter: pulse with hw-gate**

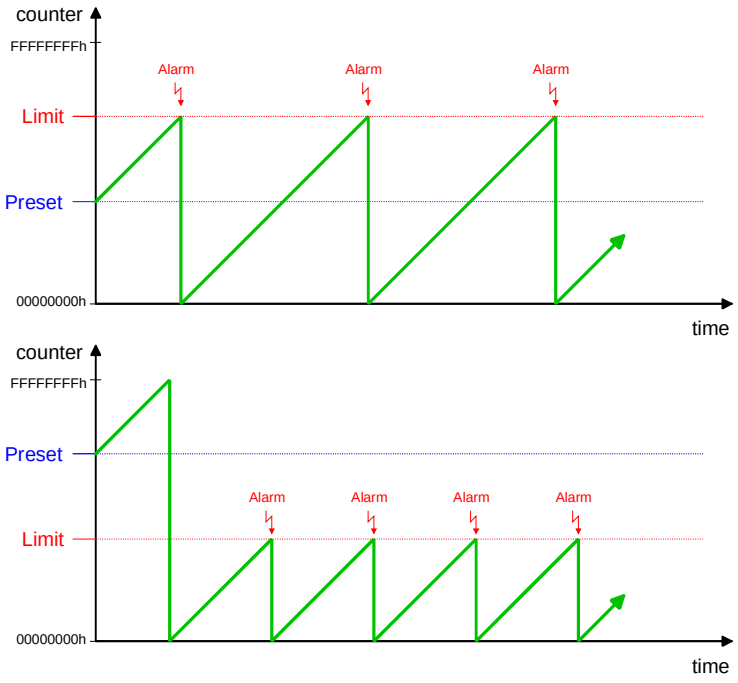
2 inputs are occupied, the 1st input is for counting and the hw-gate is released by the 2nd input.

**Counter behavior**

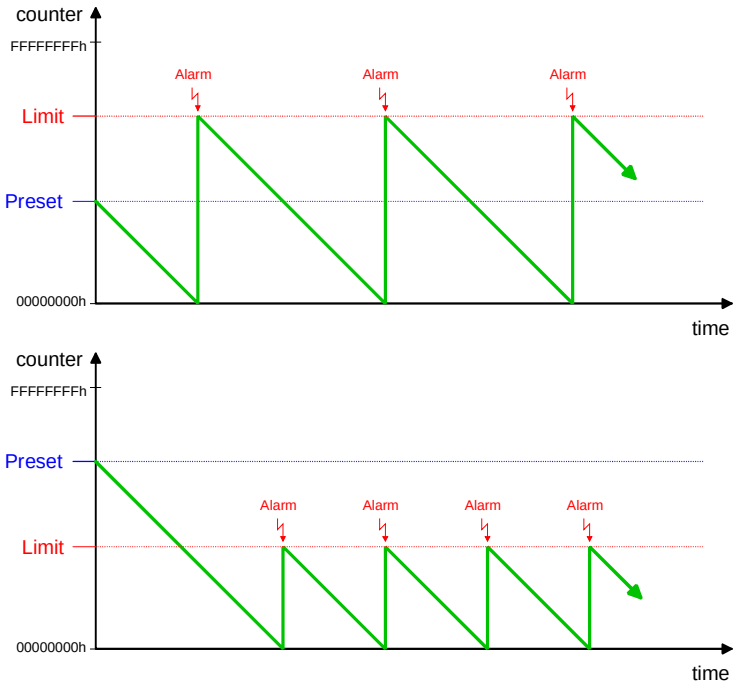
The counter values are, if no limit is fixed, in the range between 00000000h and FFFFFFFFh. As reaching FFFFFFFFh while counting up, the counter starts at 00000000h again. As reaching 00000000h while counting down, the counter starts at FFFFFFFFh again.

By fixing an upper res. lower boundary (limit) you may restrict the counter area. As soon as the counter reaches the limit, an alarm occurs if you activated it at the parameterization. By using the SFC 224 you may influence the counter during runtime, e.g. load it with an initial value (Preset). In the following illustrations the counter behavior is summarized:

Counting up




Counting down



Maximum counter frequency

The maximum counter frequency is influenced by the following facts:

- *Number of activated counters*

  The higher the number of activated counters; the lower is the maximum counter frequency.

- *PWM enabled res. disabled*

  The activating of the **P**ulse **W**idth **M**odulation (PWM) as normal or high frequency function lowers the maximum counter frequency.

- *Counter type is periodic res. continuous*

  The maximum counter frequency is lower when activating the periodic counting.

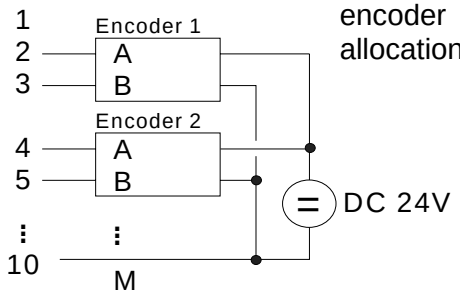
  At the periodic counting the counter value is permanently compared with a predefined limit.

  At continuous counting, the counter counts from a start value until overflow. This is less influence to the maximum counter frequency.

The following table shows the maximum counter frequencies:

| PWM disable / HF PWM enable | | |
|---|---|---|
| Number of counters | Continuous count | Periodic count |
| 1 Counter | 30kHz | 27kHz |
| 2 Counters | 23kHz | 19kHz |
| 3 Counters | 19kHz | 16kHz |
| 4 Counters | 15kHz | 13kHz |
| | | |
| PWM enable | | |
| Number of counters | Continuous count | Periodic count |
| 1 Counter | 16kHz | 16kHz |
| 2 Counters | 14kHz | 14kHz |
| 3 Counters | 13kHz | 13kHz |
| 4 Counters | 11kHz | 11kHz |

## Encoder



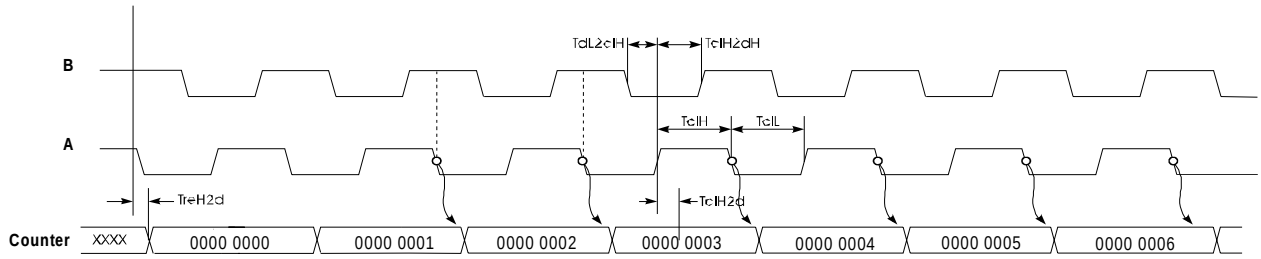

With this functionality 2 encoders can be connected. Configuring "Counter: encoder ..." 2 inputs (A, B) are switched to connect 1 encoder. Here the allocations for the respective 2nd input (B) are ignored.

**Encoder single**

With *encoder single* the counter is decremented respectively incremented by 1 with each falling edge of the respective 1st input (A) corresponding to the direction of rotation. This applies to:

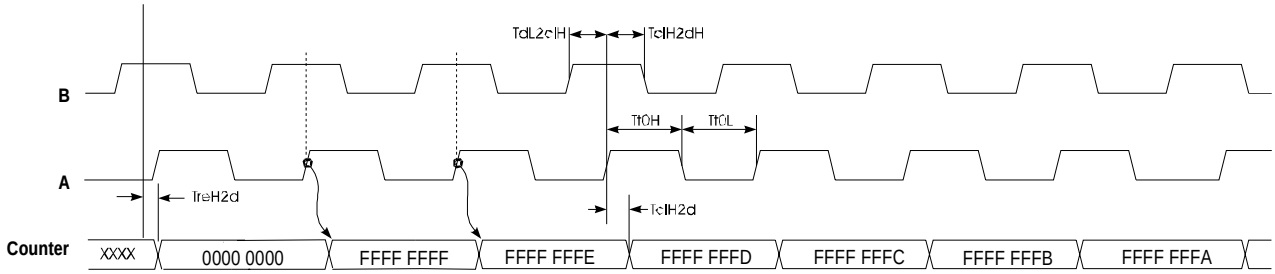
Up counter

Every falling edge of the signal at input A increments the counter if input B is at HIGH level at this moment.




Down counter

Every rising edge of the signal at input A decrements the internal counter if input B is at HIGH level at this moment.

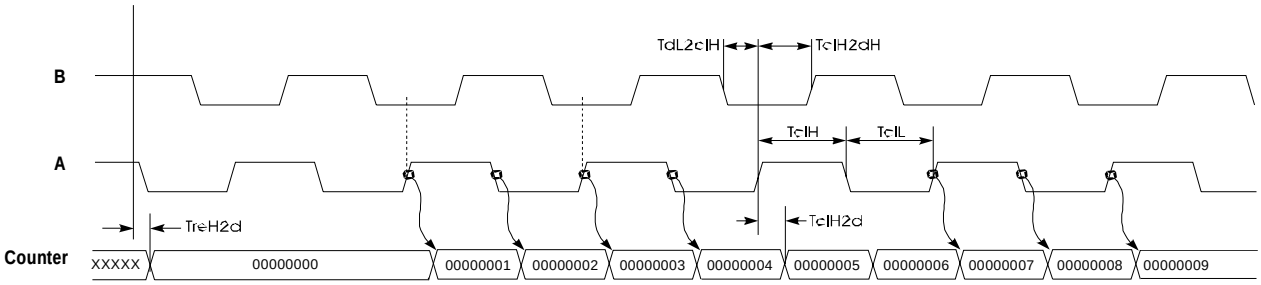



**Encoder double**

The counter is changed by 1 with each rising respectively falling edge of the signal at the 1st input (A). Here the counting direction is influenced by the level of the 2nd input (B).

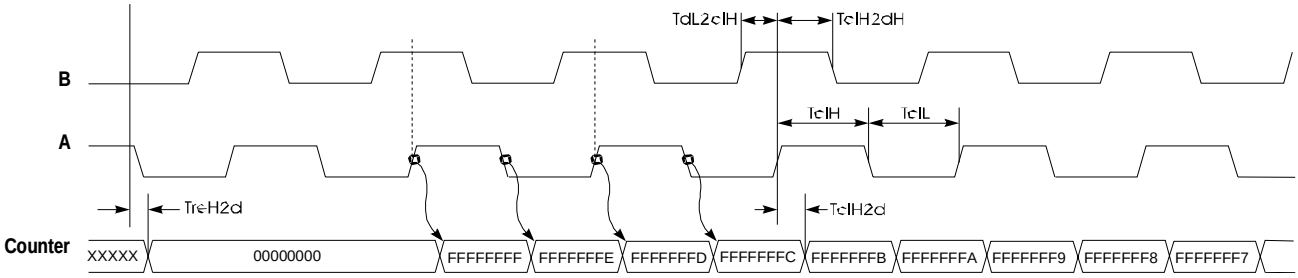
Up counter

The counter is incremented by the rising edge of signal A if input B is at a LOW level or by the falling edge of input A when input B is at a HIGH level.



Down counter

The counter is decremented by the rising edge of signal A if input B is at a HIGH level or by the falling edge of input A when input B is at a LOW level.

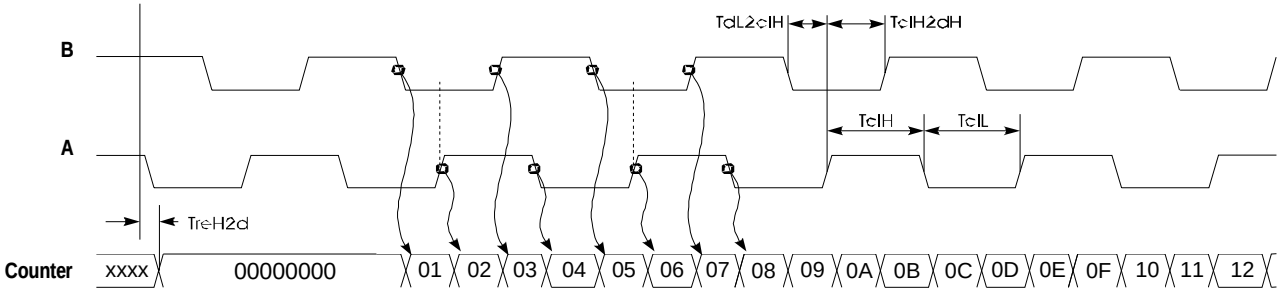



**Encoder quadruple**

The counter is changed by 1 with each rising respectively falling edge of the signal at one of the input A respectively B. Here the counting direction is influenced by the level of the other input (B or A).

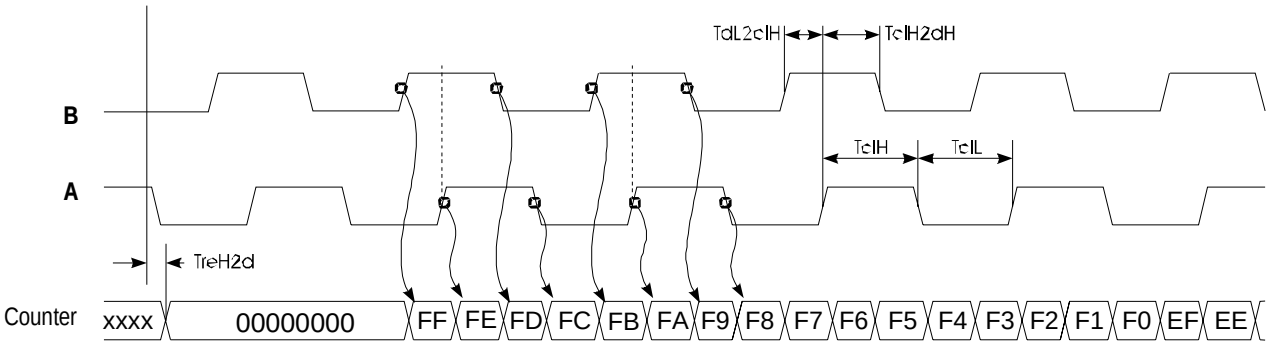
Up counter

The counter is incremented when a rising edge is applied to B while input A is at a HIGH level or if a falling edge is applied to B when input A is at a LOW level. Alternatively it is also incremented when a rising edge is applied to A when input B is at a LOW level or by a falling edge at A when input B is at a HIGH level.




Down counter

The counter is decremented when a rising edge is applied to B while input A is at a LOW level or if a falling edge is applied to B when input A is at a HIGH level. Alternatively it is also decremented when a rising edge is applied to A when input B is at a HIGH level or by a falling edge at input A when input B is at a LOW level.



## Alarm input

The first 4 inputs of X3 may be parameterized as alarm input.

The function "alarm input" means that an alarm is initialized after a selectable delay time and edge evaluation.

The delay time is the time a signal is to be applied, so an alarm is to be released. Here applies to:

- Rising edge with high level evaluation
- Falling edge with low level evaluation

**Note!**

For software-technical reasons attention should be paid that the delay time for all 4 alarms is configured equal.

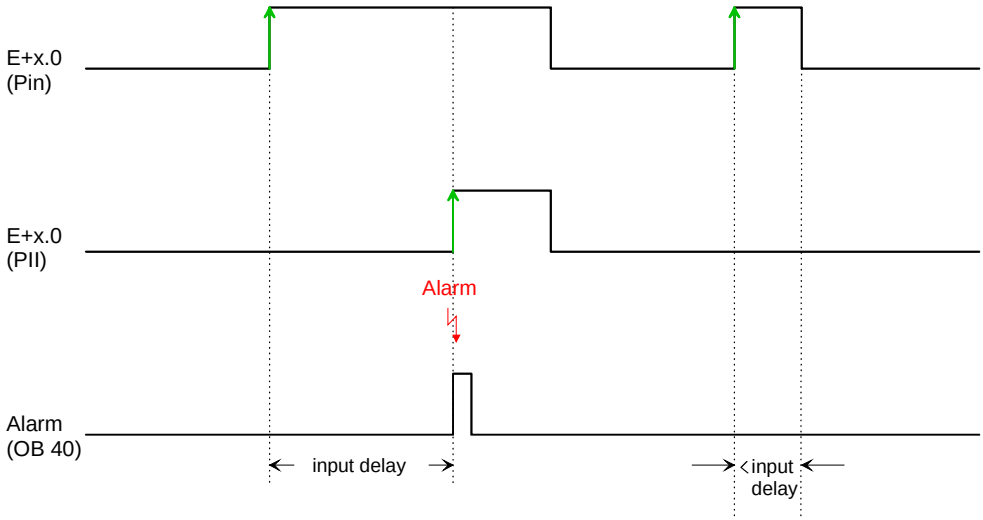
Please regard to use an identical delay time for each 4 alarm inputs. Here the following delay times may be selected:

- disabled (no delay)
- alarm: 0.1ms input delay
- alarm: 0.5ms input delay
- alarm: 3ms input delay
- alarm: 15ms input delay

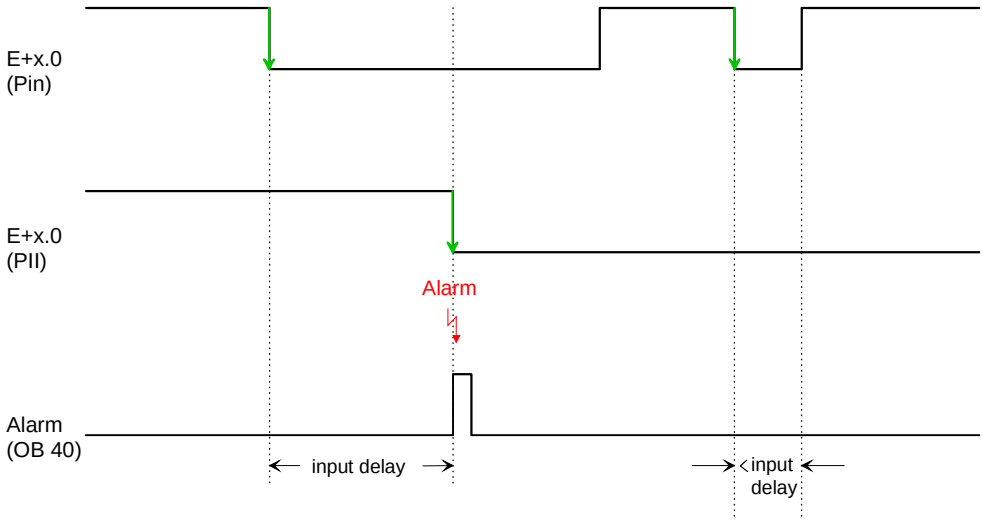
## Edge selection

Depending on the edge type which can be selected by the edge selection there is the following alarm behavior:

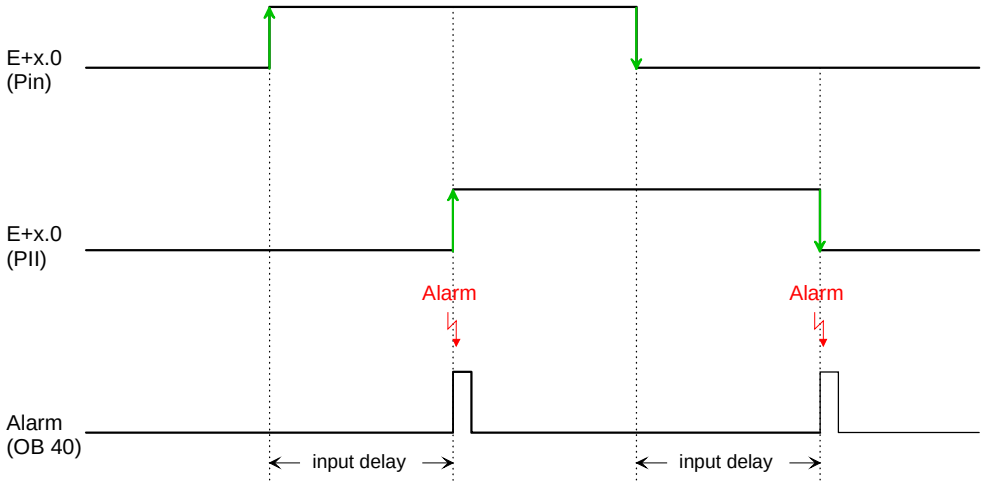
*Alarm at rising edge:*



*Alarm at falling edge:*




*Alarm at both edges:*




**Mixed configuration counter and alarm input**

A simultaneous use of the inputs as counter and alarm should be avoided, since by setting a delay time during the alarm setting the max. counter frequency is influenced:

| Delay time | max. counter frequency |
|---|---|
| 0.1ms | 5kHz |
| 0.5ms | 2kHz |
| 3ms | 333.33Hz |
| 15ms | 66.67Hz |

In the following cases a mixed configuration can be nevertheless meaningful:

*Pulse with direction*

The *direction* input can also be configured as counter respectively alarm input to count changes of the direction respectively release an alarm at changes of direction.

*Pulse with hw-gate*

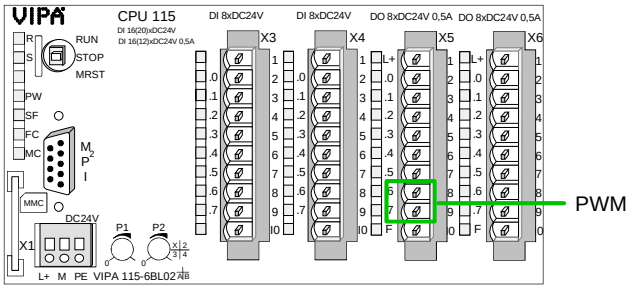
A gate input can be configured as counter respectively as alarm input as well to count the changes of the hw-gate (open/close) respectively to release an alarm.

# Deployment PWM

**Overview**

Depending on the CPU 11x the last 2 outputs of the output part X5 may be configured as pulse output. The properties and the behavior of the inputs are defined at the hardware configurator of the Siemens SIMATIC manager by means of the CPU parameter of the CPU 11x. These functions are deactivated in delivery state.

There is also the possibility to change the PWM parameter at run-time by means of the VIPA SFC 223 (PWM) and SFC 225 (HF_PWM). More details about this may be found in the Manual "VIPA Operation List Standard" (HB00_OPL_STD).

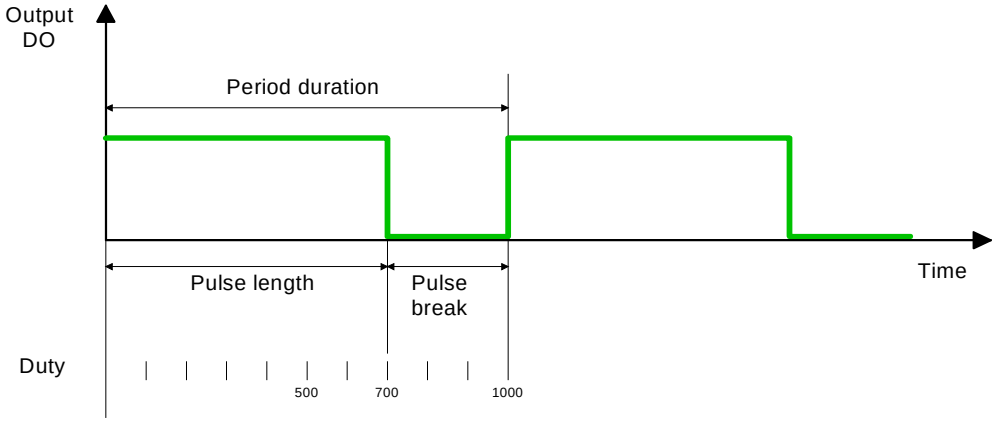



**What is PWM?**

PWM stands for **P**ulse **w**idth **m**odulation. By presetting of time parameter the CPU evaluates a pulse sequence with according pulse/break ratio and issues it via the depending output channel. You have 2 modes for the pulse width modulation:

- *Standard PWM (short: PWM)*
  Settings: time base, period, duty and min. pulse
- High frequency PWM (short: HF-PWM)
  Settings: frequency, duty and min. pulse

The PWM parameters have the following ratio:




*Period duration* = **PWM time base** x **PWM Period**

(at HF-PWM: *Period duration* = 1 / **HF PWM Freq**)

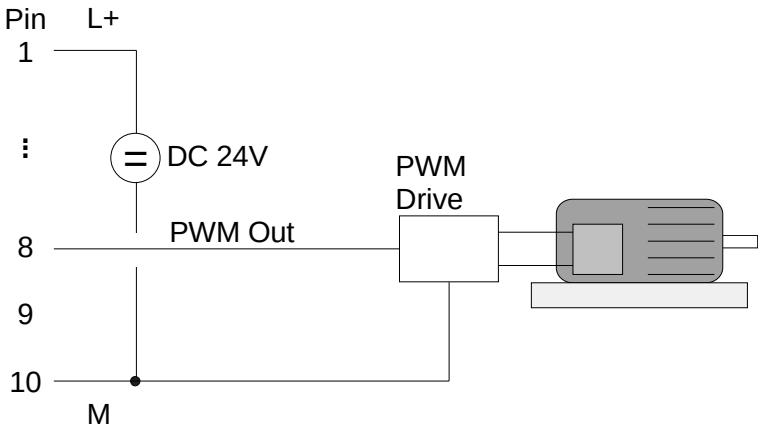
*Pulse duration* = (*Period duration* / 1000) x **PWM duty**

*Pulse break* = *Period duration* - *Pulse duration*

*Pulse duration* and *pulse break* must always be longer that the **min. pulse** (minimal pulse duration)!

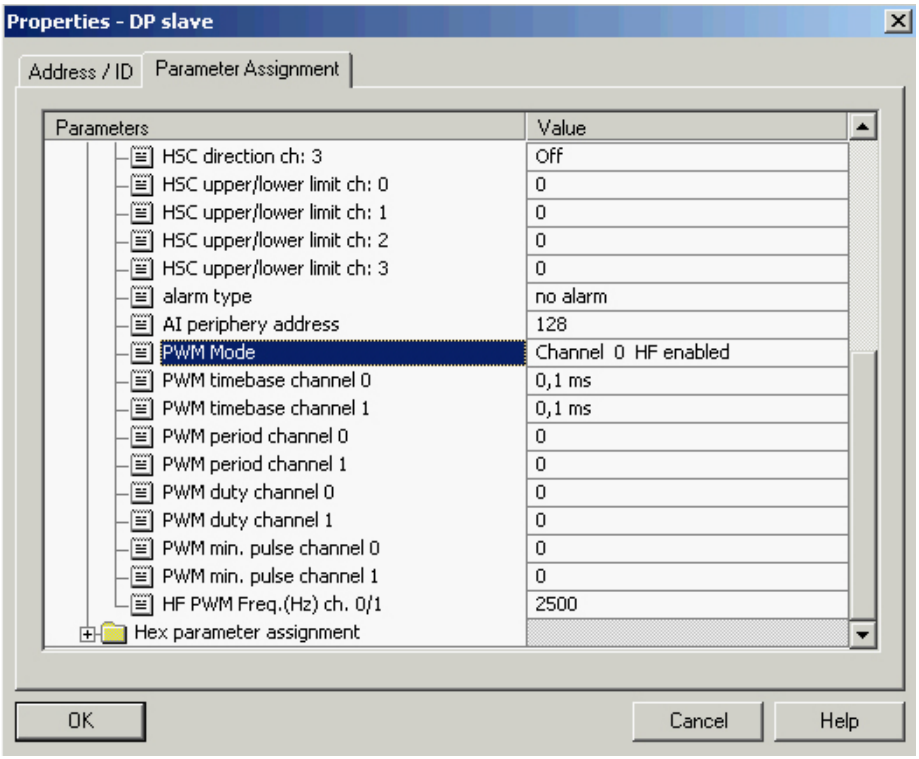
**Connecting a drive**

The connection of a drive with PWM power controller to your System 100V is shown in the following drawing:




**Parameterize PWM**

Activation and parameterization takes place in the register "Parameter assignment" of the CPU 11x:




In the following you'll find the parameters for PWM:

**PWM Mode**

Here you may activate res. deactivate the (HF)-PWM function for the according channel. At deactivated PWM function, the channel may be used as "normal" output channel.

For each of this 2 channels you may either parameterize PWM or HF-PWM. Only one PWM type is possible at a time. Mixing PWM and HF-PWM is not allowed.

**PWM time base**
(only at PWM)

At PWM mode: *PWM* the resolution and the value range of the pulse, period and minimum pulse duration per channel may be chosen.

As time base you may choose 0.1ms or 1ms.

**PWM period**
(only at PWM)

At PWM mode: *PWM* the duration of the period which is the result of the multiplication with the time base may be adjusted.

Value range: 2 ... 60000

**PWM duty**

By fixing the pulse duty ratio in "per mil" you define the ratio between pulse duration and pulse break in one period for each channel.

1 per mil = 1 time base

If the calculated pulse duration is no multiplication of the time base, it is rounded down to the next smaller time base limit.

Value range: 1 ... 1000

**PWM min. pulse duration**

If you predefine a minimal pulse duration, any pulse consequences only occur if the pulse exceeds the minimal pulse duration.

Thus you may filter very small pulses (spikes), which are not noted from the periphery anymore.

Please regard that the time base for the minimal pulse duration depends on the chosen PWM mode:

- PWM mode *PWM*
  The time base is fixed via "PWM time base" in 0.1ms or 1ms.
  Value range: 1 ... 60000
- PWM mode *HF-PWM*
  The time base for the minimal pulse duration is µs. The lowest value is 2µs.
  Value range: 2 ... 60000

**HF-PWM Freq.**
(only at HF-PWM)

At PWM mode: *HF-PWM* it fixes the frequency for both channels. Together with the pulse duty ratio and the minimal pulse duration, this enables the CPU to calculate a pulse sequence including the according pulse/break ratio.

The frequency is fixed in Hz.

Value range: 2500 ... 50000

# Diagnostic and alarm

**Overview**

An alarm can be released by the following events if parameterized:

The parameterization allows you to define the following trigger for a process alarm that may initialize a diagnostic alarm:

- 0 is reached by counting down
- *Limit* is reached counting up respectively down
- After delay time the rising edge at the alarm input with high level evaluation.
- After delay time a falling edge at the alarm input with low level evaluation.

**Alarm type**

The following alarm types can be configured by means of a hardware configuration:

- Process alarm

  A process alarm causes a call of the OB 40. Within the OB 40 you may find information about the event that initialized the process alarm.
- Process+Diagnostics alarm

  A diagnostic alarm occurs when during a process alarm execution in OB 40 another process alarm is thrown for the same event. The initialization of a diagnostic alarm interrupts the recent process alarm execution in OB 40 and branches in OB 82 to diagnostic alarm processing

**Process alarm**

At a process alarm the OB 40 is called. Here by using the local word 6 the logical basis address of the module that initialized the process alarm can be found. More detailed information about the initializing event may be found in the *local double word 8*. The bytes have the following allocation:

| Local byte | Bit 7 ... Bit 0 |
|---|---|
| 8 | Bit 3 ... 0: input which released alarm<br>Bit 0: I+0.0<br>Bit 1: I+0.1<br>Bit 2: I+0.2<br>Bit 3: I+0.3<br>Bit 7...4: reserved |
| 9 | reserved |
| 10 | Bit 3 ... 0: state of input<br>Bit 0: I+0.0<br>Bit 1: I+0.1<br>Bit 2: I+0.2<br>Bit 3: I+0.3<br>Bit 7...4: reserved |
| 11 | reserved |

**Release diagnostics alarm**

During a process alarm is processed by the CPU a diagnostic alarm can be released (if activated with Process+Diagnostic alarm) by the same event at the same channel.

This interrupts the current process alarm execution in OB40 and branches to OB82 for processing the diagnostic alarm (incoming). This OB allows you with an according programming to monitor detailed diagnostic information via the SFCs 51 and 59 and to react to it. If during the diagnostic alarm execution further events at other channels occur that may also initialize a process res. diagnostic alarm, these are temporarily stored. After finishing the current diagnostic alarm execution, the sum diagnostic message "process alarm lost" informs the CPU that in the meantime other process alarms has occurred. After the execution of the OB 82 the user application processing is continued. The diagnostic data is consistent until leaving the OB 82.

After error correction automatically a diagnostic (going) occurs if the diagnostic alarm release is still active.

In the following the record sets for diagnostic (incoming) and diagnostic (going) are specified:

Record set 0
Diagnostic (incoming)

*Record set 0*

| Local byte | Bit 7 ... Bit 0 |
|---|---|
| 8 | Bit 3 ... 0: Module class<br>1000: Function module<br>Bit 7 ... 4: reserved |
| 9 | Bit 0: Module malfunction<br>Bit 1: internal error<br>Bit 7 ... 2: reserved |
| 10 | Bit 5 ... 0: reserved<br>Bit 6: Process alarm lost<br>Bit 7: reserved |
| 11 | Bit 7 ... 0: 00h (fix) |

Record set 0
Diagnostic (going)

After error correction automatically a diagnostic (going) occurs if the diagnostic alarm release is still active.

*Record set 0*

| Local byte | Bit 7 ... Bit 0 |
|---|---|
| 8 | Bit 3 ... 0: Module class<br>1000: Function module<br>Bit 7 ... 4: reserved |
| 9 | Bit 0: Module malfunction<br>Bit 1: internal error<br>Bit 7 ... 2: reserved |
| 10 | 00h (fix) |
| 11 | 00h (fix) |

# Project transfer

**Overview**

There are 2 possibilities for the transfer of your project into the CPU:

- Transfer via MPI
- Transfer via MMC at deployment of a MMC reading device

**Transfer via MPI**

The structure of a MPI network is in principal the same as the structure of a 1.5MBaud PROFIBUS network. That means, the same rules are valid and you use for both networks the same components.

Per default, the MPI network is working with 187kBaud.

Every participant at the bus identifies itself with an unique MPI address.

You connect the single participants via bus interface plugs and the PROFIBUS bus cable.

**Terminating resistor**

A cable has to be terminated with its ripple resistor. 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 provided with voltage during start-up and operation.

**Approach**

- Connect your PG res. your PC via MPI with your CPU.
  If your PU has no MPI functionality you may use the VIPA "Green Cable" for a point-to-point connection.
  The VIPA "Green Cable" has the order no. VIPA 950-0KB00 and may only be used with VIPA CPUs of the System 100V, 200V, 300V and 500V!
- Configure the MPI slot of your PC.
- Transfer the project into the CPU by means of **PLC** > *Upload Station* in your project configuration tool.
- For more security, install a MMC and transfer the application program to the MMC by clicking on **PLC** > *Copy RAM to ROM*.
  During the write operation the MC-LED of the CPU blinks. For internal reasons the message signalizing completion of the write operation arrives too soon. The write operation is only complete when the LED has been extinguished.

## Configure MPI

Hints for the configuration of a MPI interface are to find in the documentation of your programming software.

Here we only want to show the usage of the "Green Cable" from VIPA together with the programming tool from Siemens.

The "Green Cable" establishes via MPI a serial connection between the COM-interface of the PC and the MP$^2$I jack of the CPU.

**Attention!**

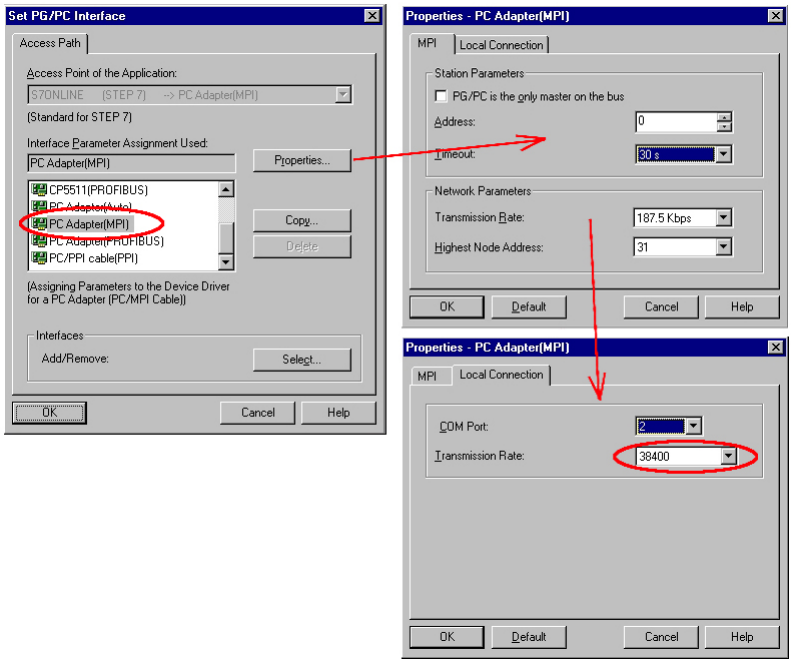
Please regard, that you may use the "Green Cable" exclusively at the MP$^2$I jacks of the Systems 100V, 200V, 300V and 500V from VIPA!

## Approach

- Start the Siemens SIMATIC manager.
- Choose **Options** > *Set PG/PC Interface*

  → The following dialog window appears, where you may configure the according MPI slot:
- Choose the "PC Adapter (MPI)" in the list, probably you may have to add it first and Click on [Properties].

  → In the following 2 sub dialogs you may configure your PC adapter like shown in the picture.




- In the Register "MPI", the **default settings** are recommended. Please regard that [Standard] has influence on the settings under "Local connection".
- At "Local connection" you choose the COM port and set, for the communication via MP$^2$I, the **transfer rate at 38400bps**.
- Close both windows with [OK].

## Test

To test the connection, plug the VIPA Green Cable to the COM interface of your PC and to the MP$^2$I jack of your CPU.

Via **PLC** > *Display Accessible Nodes* you reach the CPU with the preset MPI address 2.

## Transfer via MMC

As external storage medium a MMC is deployed. The MMC (**M**ulti **M**edia **C**ard) serves as external transfer medium for programs and firmware for, among others, it provides the PC compatible FAT16 file system. With an overall reset or PowerON the MMC is automatically read. There may be stored several projects and sub-directories on a MMC storage module. Please consider that the current project is stored in the root directory and has one of the in the following described file names.

### Transfer MMC→RAM→ROM

Always after overall reset and PowerON the CPU tries to load a user program from the MMC into the battery-buffered RAM or in the Flash memory. Here the following file names may be assigned to the project depending upon the desired functionality:

- **S7PROG.WLD**
  After overall reset the user program S7PROG.WLD is transferred into the battery-buffered RAM.
- **S7PROGF.WLD (starting with Firmware-Version V. 3.8.6)**
  After overall reset the user program S7PROG.WLD is transferred into the battery-buffered RAM and additionally into the Flash memory. An access to the Flash memory takes only place at empty battery of the buffer and when no MMC with user program is plugged-in.
- **AUTOLOAD.WLD**
  After PowerON the user program AUTOLOAD.WLD is transferred into the battery-buffered RAM.

### Transfer RAM→MMC→ROM

When the MMC has been plugged-in, the write command stores the content of the battery-buffered RAM as **S7PROG.WLD** at the MMC. The write command is controlled by means of the Siemens hardware configurator via **PLC** > *Copy RAM to ROM*. During the write process the "MC"-LED of the CPU is blinking. When the LED expires the write process is finished. Simultaneously a write process into the internal Flash memory of the CPU takes place. If there is no MMC plugged, system dependent the Siemens SIMATIC manager reacts with an error message, which may be ignored, here.

### Transfer control

After a write process onto the MMC, an according ID event is written into the diagnostic buffer of the CPU. To monitor the diagnosis entries, you select **PLC** > *Module Information* in the Siemens SIMATIC Manager. Via the register "Diagnostic Buffer" you reach the diagnosis window.

The following events may occur:

| Event-ID | Meaning |
|---|---|
| 0xE100 | MMC access error |
| 0xE101 | MMC error file system |
| 0xE102 | MMC error FAT |
| 0xE200 | MMC writing finished |
| 0xE300 | Internal Flash writing finished |

More information to the event IDs may be found at the end of this chapter.

**Note!**

If the size of the user application exceeds the user memory of the CPU, the content of the MMC is not transferred to the CPU. Execute a compression before the transfer, for this does not happen automatically.

# Operating modes

**Overview**

The CPU can be in one of 3 operating modes:

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

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

**Operating mode STOP**

- The application program is not processed.
- If there has been a processing before, the values of counters, timers, marker and the process image are retained during the transition to the STOP mode.
- Outputs are inhibited, i.e. all digital outputs are disabled.
- RUN-LED off
- STOP-LED on

**Operating mode START-UP**

- During the transition from STOP to RUN a call is issued to the start-up organization block OB 100. The length of this OB is not limited. The processing time for this OB is not monitored. The start-up OB may issue calls to other blocks.
- All digital outputs are disabled during the start-up, i.e. outputs are inhibited.
- RUN-LED blinks
- STOP-LED off

When the CPU has completed the start-up OB, it assumes the operating mode RUN.

**Operating mode RUN**

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

# Overall Reset

**Overview**

During the Overall reset the entire user memory (RAM) is erased. Data located in the memory card is not affected.

You have 2 options to initiate an Overall reset:

- initiate the overall reset by means of the function selector switch
- initiate the overall reset by means of the Siemens SIMATIC manager

**Note!**

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

**Overall reset by means of the function selector**

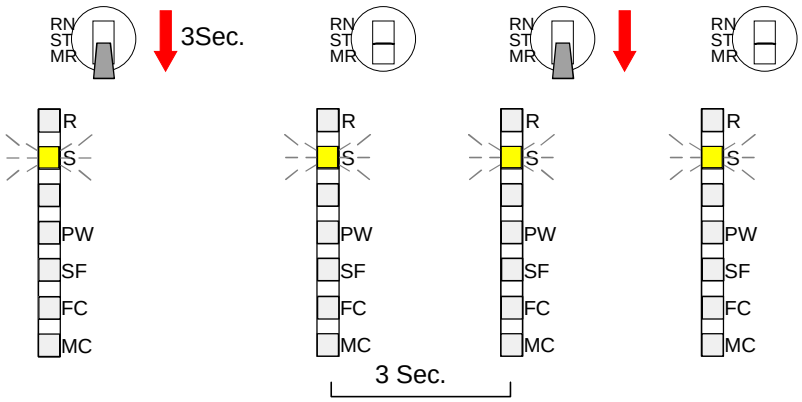
*Condition*

The operating mode of the CPU is STOP. Place the function selector on the CPU in position "ST" → The S-LED is on.

*Overall reset*

- Place the function selector in the position MR and hold it in this position for app. 3 seconds. → The S-LED changes from blinking to permanently on.
- Place the function selector in the position ST and switch it to MR and quickly back to ST within a period of less than 3 seconds.
  → The S-LED blinks (overall reset procedure).
- The overall reset has been completed when the S-LED is on permanently. → The S-LED is on.

The following figure illustrates the above procedure:



**Automatic reload**

At this point the CPU attempts to reload the parameters and the program from the memory card. → The lower LED (MC) blinks.

When the reload has been completed the LED is extinguished. The operating mode of the CPU will be STOP or RUN, depending on the position of the function selector.

**Overall reset via the Siemens SIMATIC Manager**

*Condition*

The operating mode of the CPU has to be STOP.
You may place the CPU in STOP mode by the menu command **PLC** > *Operating mode*.

*Overall reset*

You may request the Overall reset by means of the menu command **PLC** > *Clear/Reset*.

In the dialog window you may place your CPU in STOP mode and start the overall reset if this has not been done as yet.

The S-LED blinks during the overall reset procedure.

When the S-LED is on permanently, the overall reset procedure has been completed.

**Automatic reload**

At this point the CPU attempts to reload the parameters and the program from the memory card. → The "MC"-LED blinks.

When the reload has been completed, the LED is extinguished. The operating mode of the CPU will be STOP or RUN, depending on the position of the function selector.

# Firmware update

**Overview**

All CPUs of the System 100V starting with firmware version 3.3.0 allow you to update the firmware with a MMC via the reserved file name *firmware.bin* or via the update software and the Green Cable from VIPA.

The 2 last recent firmware versions can be downloaded in the service area of www.vipa.de and from the ftp server ftp.vipa.de.

**Attention!**

Please be very careful with loading a new firmware. Under certain circumstances you may destroy your 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 also regard that the update version has to be different from the existing version, otherwise no update will happen.

**Read firmware version**

If you didn't execute a firmware update before, you may find the recent firmware version on the label on the backside of your System 100V module. You may also request the recent firmware version via **PLC** > *Module status,* register tab "General".

**Load firmware via ftp.vipa.de**

To display ftp-sites in your web browser you may have to adjust the following settings:

*Internet Explorer (ftp access ability since V. 5.5)*

**Options** > *Internet options*, register "extended" in the area "Browsing":

- activate: "Activate directory view for ftp-sites"
- activate: "Use passive ftp..."

*Netscape (ftp access ability without further adjustments since V. 6.0)*

If you have problems with the ftp access, please ask your local system operator.

To download the firmware file, order no. and version no. (HW) are required. These ID numbers mark the storage directory of the concerning firmware. For example the firmware file of a System 100V CPU with the order no. 115-6BL02 and HW no. 1 may be found with the file name 115-6BL02B.xxx (xxx is the according firmware version).

- Type the address www.vipa.de.
- Click *Service* > *Download* > *Firmware Updates* in the navigation bar and download the according firmware.
- Extract the zip-file into the wanted directory on your PC.
- If you want to execute the update with the Green Cable, an update software is required that you may download under "Software Tools" in the download area.

**Transfer firmware from MMC into CPU**

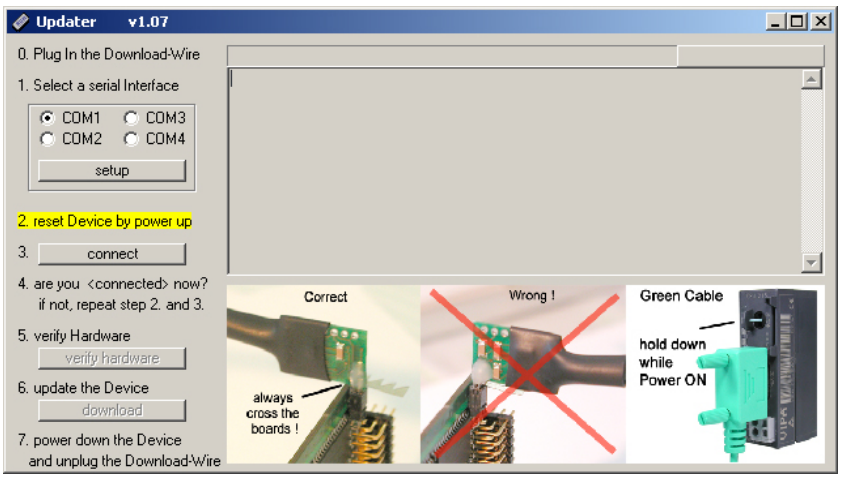
There may be several projects and directories on one MMC. Please regard that the recent firmware file for the CPU has to be stored in the root directory, i.e. on the most upper level. To enable the identification of this file as firmware, rename the file into **firmware.bin**.

- Install your MMC reading device and plug in a MMC. Transfer the file firmware.bin to your MMC.
- Set the RUN-STOP lever of the CPU in position STOP.
- Turn off the power supply.
- Plug the MMC with the firmware file into the CPU. Please take care of the plug-in direction of the MMC.
- Turn on the power supply.
- After a short boot time, the alternate blinking of the LEDs SF and FC shows that a file has been found on the MMC.
- Start the transfer of the firmware by tipping the RUN/STOP lever into position MRST within 10s. The CPU shows the transfer via a LED running light.
- During the update process, the LEDs SF, FC and MC are blinking alternately. This process may last several minutes.
- The update is ready and error free when all CPU-LEDs are on. At fast blinking, an error has occurred.

**Firmware update via Green Cable and "Updater"**

To update the firmware via Green Cable, the Green Cable from VIPA and the software tool "Updater" are required. The software can be downloaded from www.vipa.de. Load the Updater and extract the zip-file into a directory of your PC.

Start the Updater with cpu_up.exe. The following dialog window appears:




A more detailed description of the approach is on the following page.

**Continued firmware update via Green Cable and "Updater"**


to 0. Connect the COM interface of the PC and the MP$^2$I jack of your CPU via the Green Cable.

to 1. Type the COM interface (you should not alter the setup)

to 2. Turn off the power supply of your CPU, hold the RUN/STOP lever in position MRST and turn on the power supply.

Now the CPU is ready for the firmware update and monitors this by turning all LEDs on.

to 3. Click on connect in the Updater.

to 4. A connection to the CPU is established and shown via the message [connected].

If an error message appears instead, repeat the steps above with another COM interface.

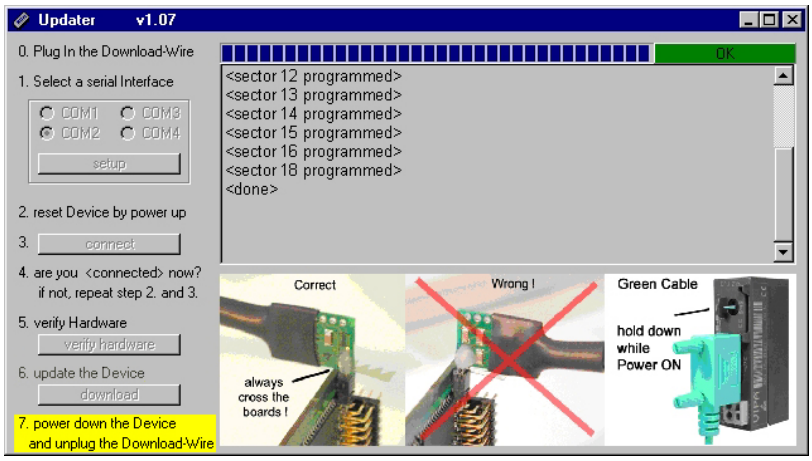
to 5. At error free connection click on verify hardware.

You CPU is now prepared for data transfer.

to 6. A click on download opens a file selection window. Choose the according firmware and start the download with Open.

If the error message "The selected file doesn't fit to your hardware" appears you may have been tried to download a firmware that is not compatible to your CPU. With a valid firmware version, the update process starts. This process may last several minutes and is shown in a process bar.

After the download, the following window should appear:




to 7. Turn off the power supply of your CPU, disconnect the Green Cable and turn on the power supply again. Now the CPU is ready with the new firmware.

If your CPU does not start anymore, an error occurred during the firmware update. Please call the VIPA hotline.

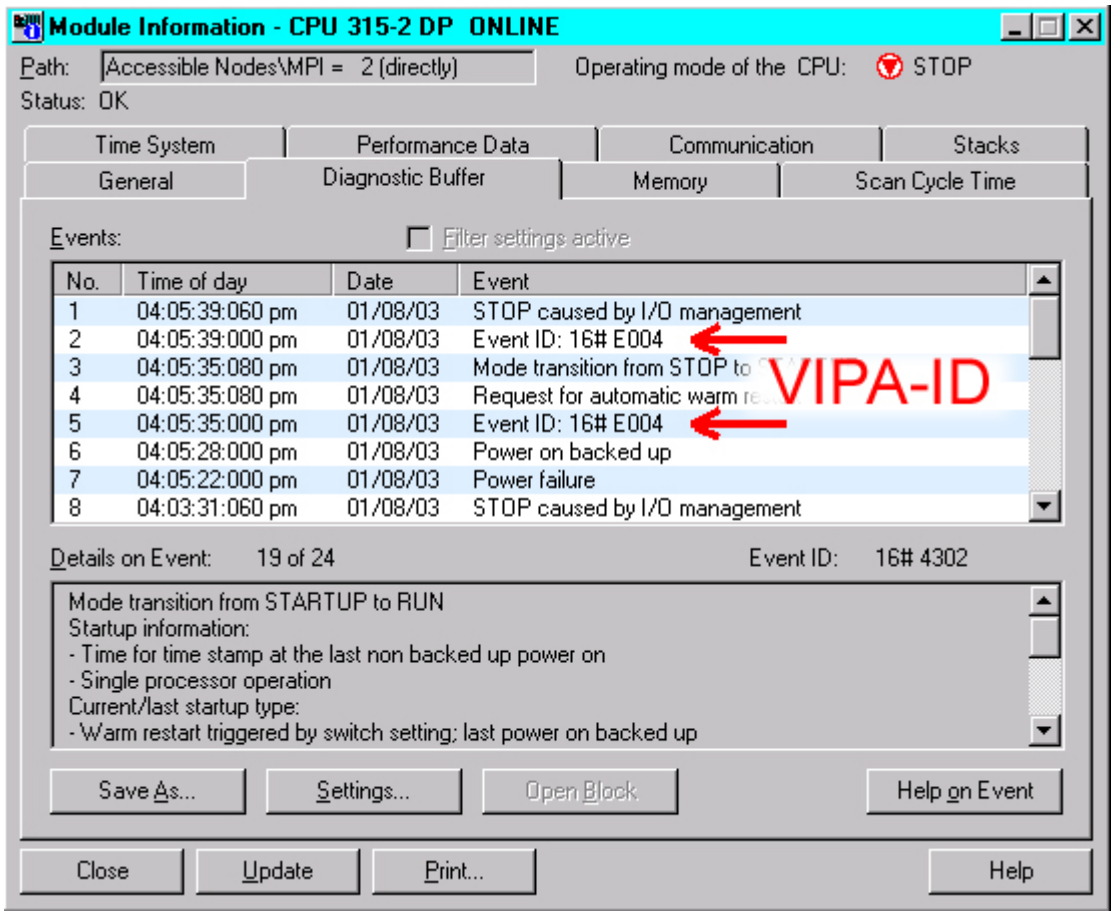
# VIPA specific diagnostic entries

**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.

**Monitoring the diagnostic entries**

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




The diagnosis is independent from the operating mode of the CPU. You may store a max. of 100 diagnostic entries in the CPU.

The following page shows an overview of the VIPA specific Event-IDs.

**Overview of the Event-IDs**

| Event-ID | Description |
|---|---|
| 0xE003 | Error at access to I/O devices<br>Zinfo1: I/O address<br>Zinfo2: Slot |
| 0xE004 | Multiple parameterization of a I/O address<br>Zinfo1: I/O address<br>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 I/O area |
| 0xE008 | Internal error – Please contact the VIPA-Hotline! |
| 0xE009 | Error at access to standard back plane bus |
| 0xE010 | Not defined module group at backplane bus recognized<br>Zinfo2: Slot<br>Zinfo3: Type ID |
| 0xE011 | Master project engineering at Slave-CPU not possible or wrong slave configuration |
| 0xE012 | Error at parameterization |
| 0xE013 | Error at shift register access to VBUS digital modules |
| 0xE014 | Error at Check_Sys |
| 0xE015 | Error at access to the master<br>Zinfo2: Slot of the master (32=page frame master) |
| 0xE016 | Maximum block size at master transfer exceeded<br>Zinfo1: I/O address<br>Zinfo2: Slot |
| 0xE017 | Error at access to integrated slave |
| 0xE018 | Error at mapping of the master I/O devices |
| 0xE019 | Error at standard back plane bus system recognition |
| 0xE01A | Error at recognition of the operating mode (8 / 9 Bit) |
| 0xE0CC | Communication error MPI / Serial |
| 0xE100 | MMC access error |
| 0xE101 | MMC error file system |
| 0xE102 | MMC error FAT |
| 0xE104 | MMC error at saving |
| 0xE200 | MMC writing finished (Copy RAM to ROM) |
| 0xE210 | MMC reading finished (reload after overall reset) |
| 0xE300 | Internal Flash writing ready (Copy RAM to ROM) |
| 0xE310 | Internal Flash reading ready (reload after battery failure) |

# Using test functions for control and monitoring variables

**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 VKE can be displayed by means of the test function **Debug** > *Monitor*.

You can modify and/or display the status of variables by means of the test function **PLC** > *Monitor/Modify variables*.

**Debug** > *Monitor*

This test function displays the current status and the VKE of the different operands while the program is being executed.

It is also possible to enter corrections to the program.

**Note!**

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

The processing of statuses can be interrupted by means of jump commands or by timer and process alarms. At the breakpoint the CPU stops collecting data for the status display and instead of the required data it only provides the PU with data containing the value 0.

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

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

The interruption of the processing of statuses does not change the execution of the program. It only shows that the data displayed is no longer valid from that point on where the interrupt occurred.

**PLC** >
*Monitor/Modify variables*

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

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

*Control of outputs*

It is possible to check the wiring and proper operation of output modules.

You can set outputs to any desired status with or without a control program. The process image is not modified but outputs are no longer inhibited.

*Control of variables*

The following variables may be modified:

I, Q, M, T, C and D.

The process image of binary and digital operands is modified independently of the operating mode of the CPU 11x.

When the operating mode is RUN the program is executed with the modified process variable. When the program continues they may, however, be modified again without notification.

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

# Chapter 4 Deployment Micro-PLC CPU 11xDP

**Overview**

Content of this chapter is the deployment of the Micro-PLC CPU 11xDP under PROFIBUS. It includes all information required for deploying an intelligent PROFIBUS-DP slave.

The chapter closes with a detailed example for the Micro-PLC CPU 11xDP.

**Content**

# Principles

**General**

PROFIBUS is an open field bus standard for building, manufacturing and process automation. PROFIBUS defines the technical and functional properties of a serial field bus system that can be used to create a network of distributed digital field-automation equipment on the lower (sensor-/drive level) to middle performance level (process level).

PROFIBUS comprises various compatible versions. The specifications contained in this description refer to PROFIBUS-DP.

**PROFIBUS-DP**

PROFIBUS-DP is particularly suitable for applications in production automation. DP is very fast, offers Plug & Play and is a cost-effective alternative to parallel cabling between CPU and the distributed periphery. PROFIBUS-DP is conceived for high-speed data exchange on the sensor-drive level. This is where central controllers like CPUs communicate via fast, serial connections with distributed in- and output devices.

During a single bus cycle the master reads the input values from the various slaves and writes new output information into the slaves.

**Master and Slaves**

PROFIBUS distinguishes between active stations (masters) and passive stations (slaves).

*Master equipment*

Master equipment controls the data traffic on the bus. There may be also several masters at one PROFIBUS. This is referred to as multi-master operation. The bus protocol establishes a logical token ring between the intelligent devices connected to the bus.

A master can send unsolicited messages if it has the bus access permission (Token). In the PROFIBUS protocol these masters are also referred to as active stations.

*Slave equipment*

Typical slave equipment holds data of peripheral equipment, sensors, drives and transducers. The VIPA PROFIBUS couplers are modular slave equipment that transfer data between the system 100V periphery and the leading master.

These devices do not have bus access permission in accordance with the PROFIBUS standard. They can only acknowledge messages or transfer messages to a master if requested by the respective master. Slaves occupy a very limited part of the bus protocol. Slaves are also referred to as passive stations.

## Communication

The bus communication protocol provides two procedures for accessing the bus:

**Master to master**

Communications with the master is also referred to as token passing procedure. Token passing guarantees that the station receives access permission to the bus. This access right to the bus is passed between the stations in form of a "token". A token is a specific message that is transferred via the bus.

When a master is in the possession of the token it also has the access right to the bus and can communicate with all other active and passive stations. The token retention time is defined when the system is being configured. When the token retention time has expired the token is passed along to the next master that acquires the bus access rights with the token so that it can communicate with all other stations.

**Master slave procedure**

Data is exchanged in a fixed repetitive sequence between the master and the slaves assigned to the respective master. When you configure the system you define which slaves are assigned to a certain master. You can also specify which DP-slave is included in the cyclic exchange of application data and which ones are excluded.

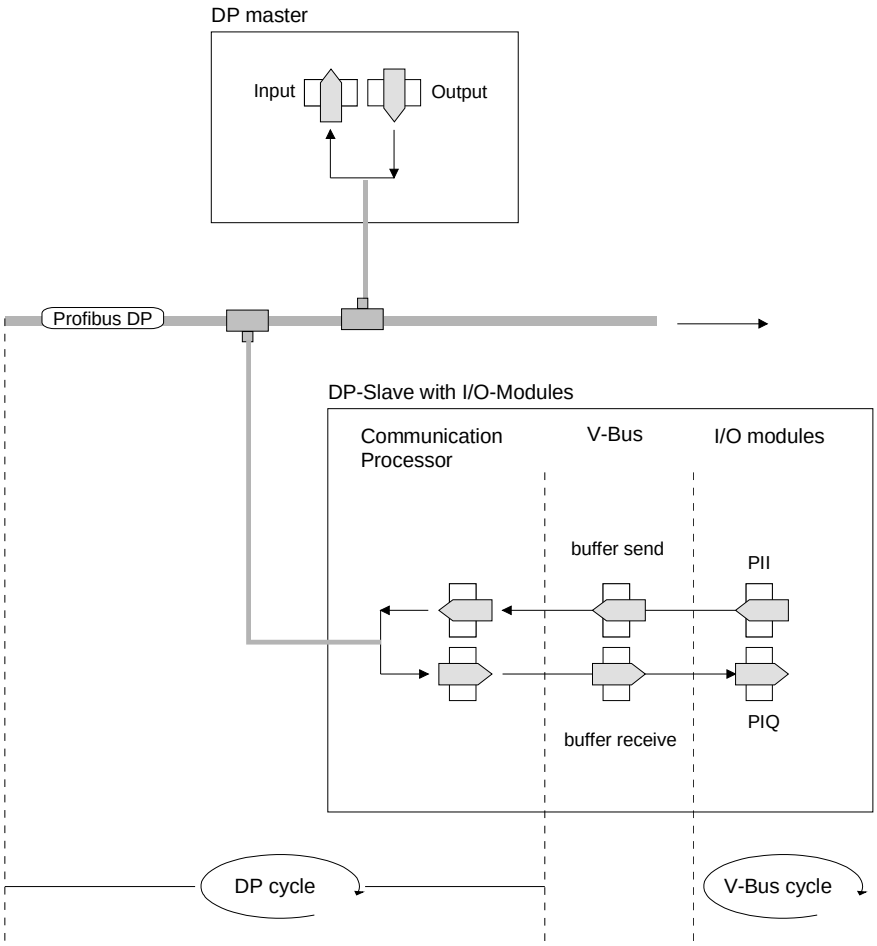
The master-slave data transfer is divided into parameterization, configuration and data transfer phases. Before a DP slave is included in the data transfer phase the master verifies during the parameterization and configuration phase, whether the specified configuration agrees with the effective configuration. This verification process checks the device type, format and length as well as the number of inputs and outputs. This provides you with effective protection against configuration errors.

The master handles application data transfers independently. In addition you can also send new configuration data to a bus coupler.

If in the status DE „Data Exchange“ the master is sending new basic data to the slave and the responding telegram of the slave transfers the recent input data to the master.

## The principle of data transfer operations

The data exchange between the DP master and the DP slave is performed in a cycle using send and receive buffers.




PII: Process image of the Inputs

PIQ: Process image of the Outputs

### V-Bus cycle

In one V-Bus cycle (i.e. VIPA backplane bus) all input data of the single modules are collected in the PII and all output data from the PIQ are transferred to the output modules. After the data exchange is completed, the PII is transferred to the sending buffer (buffer send) and the content of the input buffer (buffer receive) is transferred to PIQ.

### DP cycle

In one PROFIBUS cycle the master contacts all its slaves with a data exchange. There the memory areas assigned to the PROFIBUS are written res. read.

Afterwards the DP-master transmits data of the input area to the receive buffer of the communication processor and the data of the send buffer is transferred into the PROFIBUS output area.

The DP master to DP slave data exchange on the bus is repeated cyclically and does not depend on the V-Bus cycle.

**V-Bus cycle vs. DP cycle**

To guarantee a simultaneous data transfer the V-Bus cycle time should always be same or lower than the DP cycle time.

In the delivered EDS you'll find the parameter **min_slave_interval = 3ms**.

Thus guarantees that the PROFIBUS data on the V-Bus is updated latest every 3ms. Though you are allowed to execute one Data Exchange with the slave every 3ms.

**Data consistency**

Data is referred to as being consistent, if it has the same logical contents. Data that belongs together is: the high- and low-byte of an analog value (word consistency) and the control and the status byte with the respective parameter word required to access the registers.

The data consistency during the interaction between the peripherals and the controller is only guaranteed for 1 byte. That is, the bits of one byte are acquired together and they are transmitted together. Byte-wise consistency is sufficient for the processing of digital signals.

Where the length of the data exceeds a single byte, e.g. analog-values the data consistency must be expanded. PROFIBUS guarantees consistency for the required length of data. Please ensure that you use the correct method to read consistent data from the PROFIBUS master into your CPU.

For additional information please refer to the manual on your PROFIBUS master as well as the one for the interface module.

**Restrictions**

When a high-level master fails this is not recognized automatically by the CPU. You should always pass along a control byte to indicate the presence of the master thereby identifying valid master data.

The example at the end of this chapter also explains the use of the control byte.

**Diagnostic**

There is a wide range of diagnostic functions under PROFIBUS-DP to allow a fast error localization. The diagnostic data are broadcasted by the bus system and summarized at the master.

### Transfer medium

As transfer medium PROFIBUS uses an isolated twisted-pair cable based upon the RS485 interface or a duplex photo cable. The transfer rate is for both methods max. 12Mbaud.

More information about this theme is available at „installation guideline“.

### PROFIBUS DP via RS485

The RS485 interface is working with voltage differences. Though it is less irritable from failures than a voltage or a current interface. You are able to configure the network as well linear as in a tree structure. Your Micro-PLC CPU 11xDP includes a 9pin slot where you connect the Micro-PLC CPU 11xDP into the PROFIBUS network as a slave.

The bus structure under RS485 allows an easy connection res. disconnection of stations as well as starting the system step by step. Later expansions don’t have any influence on stations that are already integrated. The system realizes automatically if one partner had a fail down or is new in the network.

### Addressing

Every partner of the PROFIBUS network has to identify itself with a certain address. This address may exist only one time in the bus system and has a value between 0 and 125.

At the CPU 11xDP you choose the address via the Siemens SIMATIC Manager.

### GSD files

To configure the slave connections in the Siemens SIMATIC Manager, you’ve got all the information about your VIPA-modules in form of an electronic data sheet file.

Structure and content of this file are dictated by the PROFIBUS User Organization (PNO) and can be seen there.

Install this file in the Siemens SIMATIC Manager. Look for more information below under "Project engineering CPU 11xDP".

The following GSD-files are required:

| GSD | required for |
|---|---|
| VIPA_11x.GSD | Configuration CPU 11x and CPU 11xDP at slave |
| VIPA04Dx.GSD | Configuration CPU 11xDP at master |

# Project engineering CPU 11xDP

**Overview**

In contrast to a stand-alone slave, the Micro-PLC CPU 11xDP is an "intelligent coupler".

The "intelligent coupler" processes data that is available from an input or an output area of the CPU. Separate memory areas are used for input and for output data. The areas may be accessed via your CPU application.

Please ensure that none of the addresses overlap since the addressing areas that are occupied by the DP slave may not be displayed directly.

**Note!**

For configuring the CPU and the PROFIBUS-DP master a thorough knowledge of the Siemens SIMATIC manager and the hardware configurator from Siemens is required!

**Configuration in the Siemens SIMATIC manager**

The address allocation and the parameterization takes place in the Siemens SIMATIC manager as a virtual PROFIBUS system. For the PROFIBUS interface is also standardized in software, we are able to guarantee the full functionality under the Siemens SIMATIC manager by including a GSD file.

**Steps of the CPU 11xDP configuration**

To be compatible with the Siemens SIMATIC manager, you have to follow this steps:

- Create a complete CPU 315-2DP with DP master system (address 2)
- Add a PROFIBUS slave "VIPA_CPU11x" with address 1 (VIPA_11x.GSD required)
- Include the CPU type **11xDP** at plug-in location 0 of the slave system
- Select PROFIBUS parameters for the CPU 11xDP
- Enter I/O periphery parameters
- Transfer project via MPI into the CPU 11xDP

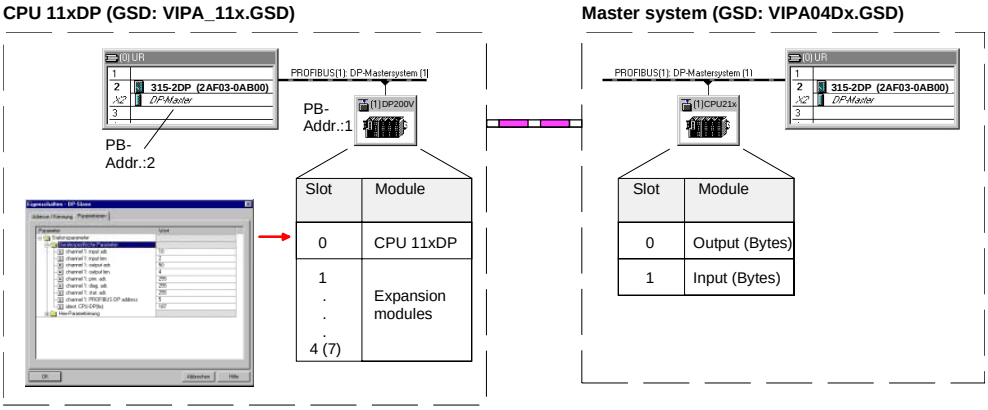
**Steps of the master configuration**

At the master you have to execute the following steps:

- Create CPU with DP master system (address 2)
- Add PROFIBUS slave VIPA_CPU11xDP (VIPA04Dx.GSD required)
- Enter the PROFIBUS in- and output areas starting with plug-in location 0 in Byte res. words

**Relation between master and slave**

The following illustration summarizes the project engineering at the slave and the master:




## Configuration CPU 11xDP

The following section describes the single steps for the slave project engineering.

**Conditions**

For the project engineering of the CPU 11xDP in a system 200V res. system 300V master system the following conditions must be met:

- Siemens SIMATIC manager is installed.
- GSD-file of the CPU11xDP is included in the hardware configurator.
- Transfer possibilities between hardware configurator and CPUs are available.

**Install hardware configurator from Siemens**

The hardware configurator is part of the Siemens SIMATIC manager. The modules that may be parameterized are listed in the hardware catalog.

For the deployment of the PROFIBUS-DP slaves of the systems 100V, 200V and 300V from VIPA, you have to include the modules in the hardware catalog via the GSD-file from VIPA.

**GSD: Include VIPA_11x.GSD**

Start the hardware configurator from Siemens. To include a new GSD, no project may be open.

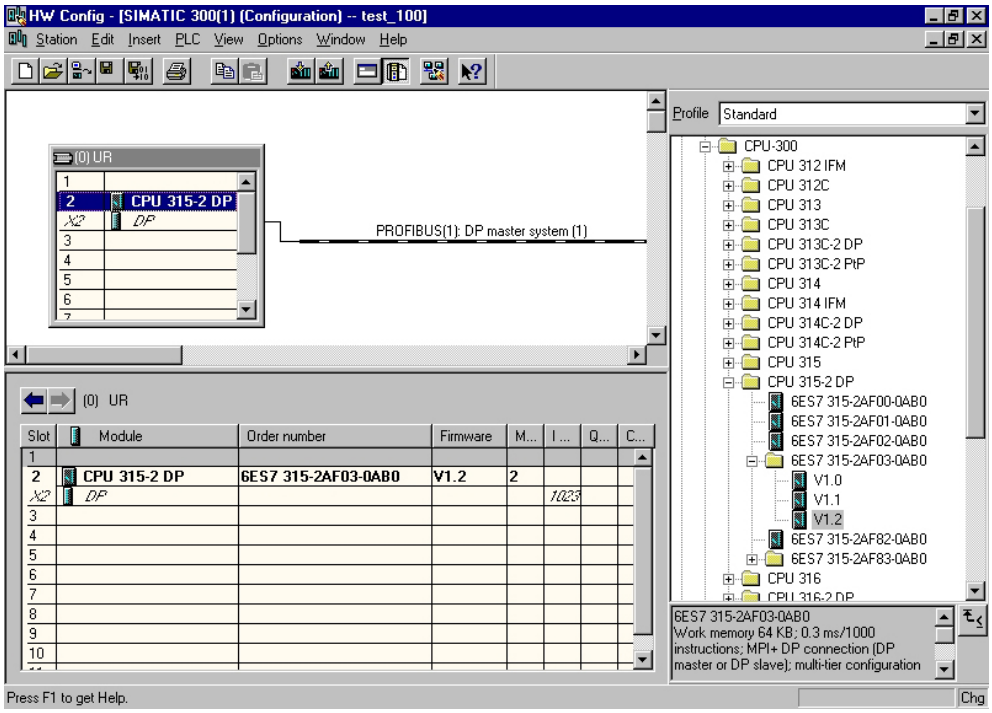
Open the file window for installing GSDs via **Options** > *Install GSD...*. Insert the delivered data medium and select the according GSD. The installation starts with [Open].

Normally you'll find the modules from VIPA installed via the GSD in the hardware catalog under *PROFIBUS-DP* > *Additional field devices* > *I/O* > *VIPA*.

**Create a virtual PROFIBUS system**

- Create a new project system 300 and add a profile rail from the hardware catalog.
- Insert the CPU 315-2DP. This CPU with PROFIBUS master is to find in the hardware catalog under: *Simatic300* > *CPU-300* > *CPU315-2DP* > **6ES7 315-2AF03-0AB0**
- Assign the PROFIBUS address 2 to your master
- Click on "DP" and choose the operating mode "DP master" under *Object properties.* Confirm with OK.
- Via right-click on "DP", the context menu opens. Choose "Add master system". Create a new PROFIBUS subnet via NEW.

  The following picture shows the created master system:



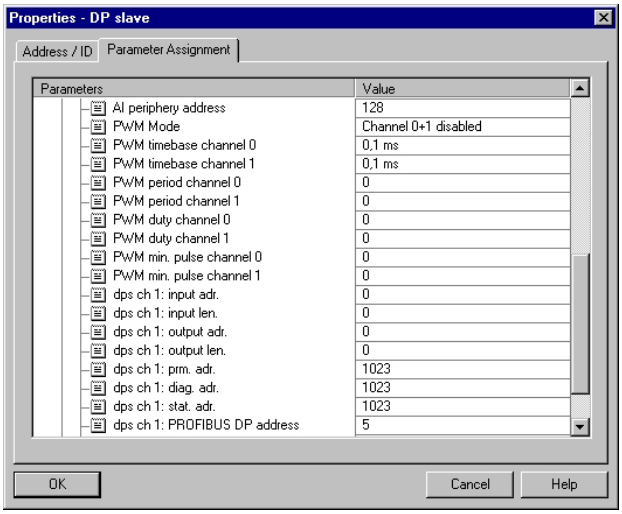

**Configure CPU 11xDP and modules**

To be compatible with the Siemens SIMATIC Manager, you have to include the CPU 11xDP explicitly.

- Add the system "VIPA_CPU11x" to your subnet. This is in the hardware catalog under *PROFIBUS DP* > *Additional field devices* > *I/O* > *VIPA_System_100V*. Assign the PROFIBUS address 1 to the DP slave.
- Place your CPU 11xDP from VIPA on plug-in location 0 in the hardware configurator.
  **The plug-in location 0 is mandatory!**
- Parameterize the in-/output periphery.
- In the CPU parameter window you may adjust the data areas of the PROFIBUS section. You can find more detailed information at the following pages.
- Save your project.
- Transfer your project via MPI to the CPU 11xDP.

**Parameterize PROFIBUS section**

The PROFIBUS section shows its data areas in the memory area of the CPU 11xDP. The allocation of these areas is fixed at the properties of the CPU 11xDP. Via a double-click on the CPU 11xDP you reach the dialog window for parameterizing the data areas for the PROFIBUS slave. More detailed information is contained in "DP slave parameters".



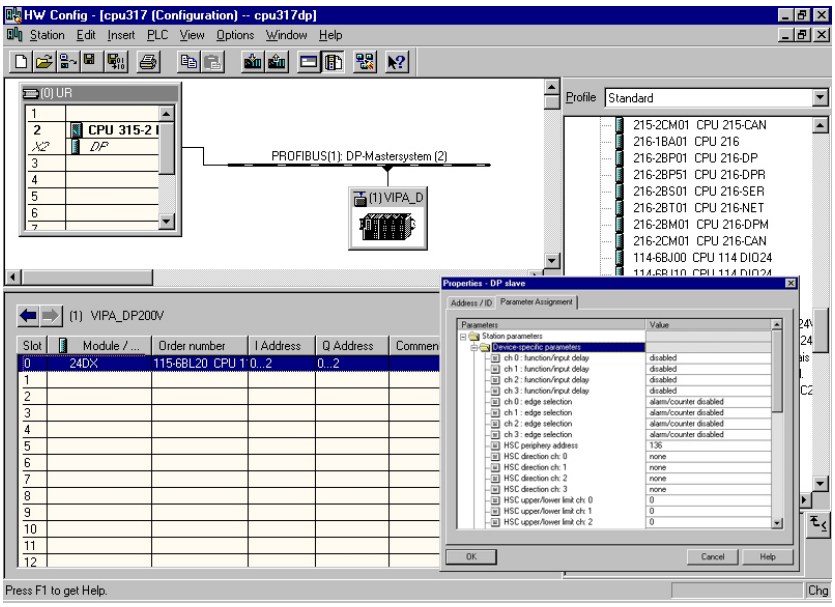

**Attention!**

Please take care of identical data areas length values at master and slave configuration.

The data areas that are occupied in the CPU by the PROFIBUS section may only be monitored in the CPU parameter window.

**View in the hardware configurator from Siemens**

In the following all relevant dialog windows of the slave parameterization are listed. You will also see how to include your System 100V:

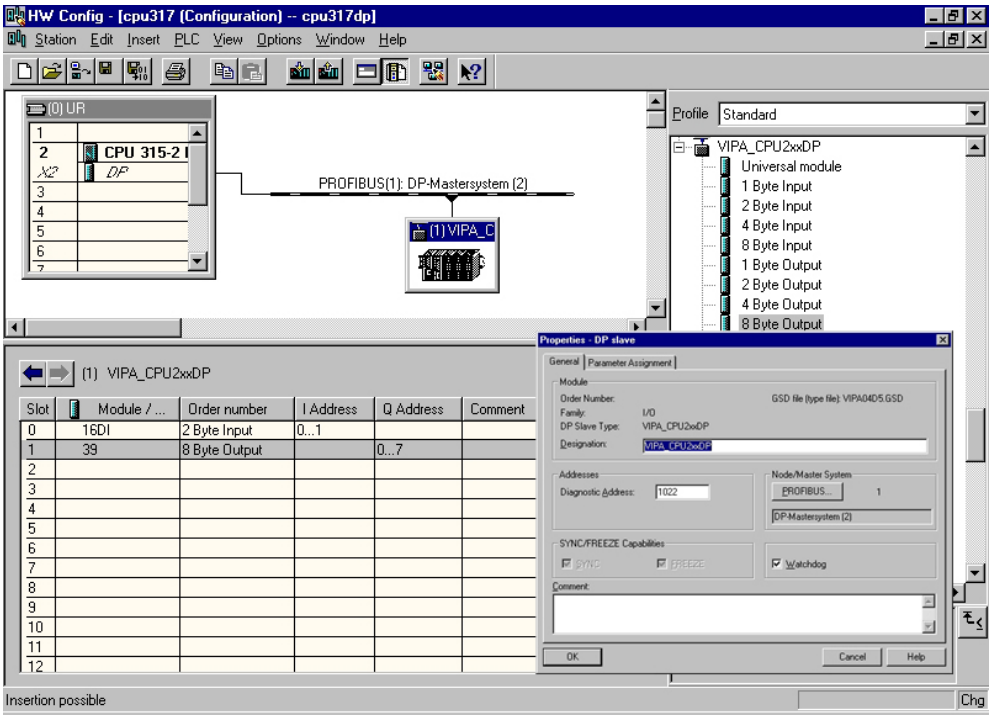


## Project engineering of a master system

To engineer a master system on a higher level, you have to include the GSD: VIPA04Dx.GSD.

- Start your configuration tool and project a PROFIBUS-DP master that is leading your CPU 11xDP.
- Add a DP slave system "VIPA_CPU11xDP" to the master. This is to find in the hardware catalog under:
  *PROFIBUS-DP > Additional field devices > I/O > VIPA > VIPA_System_100V*.
- Select a valid PROFIBUS address for your DP slave.
- Assign memory areas of the CPU address range to the PROFIBUS section for the inputs and outputs in form of "modules". Input and output section always need a not interrupted block of addresses!
- Save your project and transfer it into the CPU of your master system

In the following all relevant dialog windows of the master parameterization are listed:




**Note!**

When your DP master system is a System 200V from VIPA, you may parameterize the directly plugged-in modules by adding a "DP100V" slave system.

To enable the VIPA-CPU to recognize the project as central system, you have to assign the PROFIBUS address 1 to the "DP100V" slave system!

**When deploying a IM 208 PROFIBUS-DP master, please ensure that this has a firmware version > V3.0; otherwise this is not compatible with the CPU 11x with a firmware version >V3.0. The firmware version is to find on the label on the backside of every module.**

# DP slave parameters

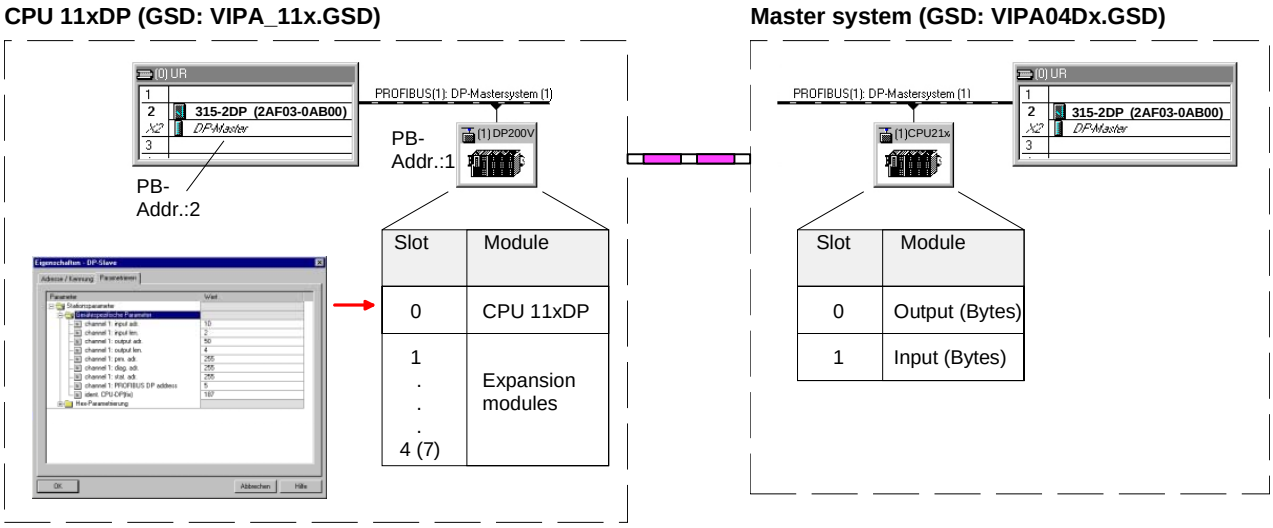
**Overview**

"Intelligent" slave means that the PROFIBUS section includes its data areas into the memory range of the CPU. The allocation of the ranges takes place in the "Properties" of the CPU 11xDP.

The in- res. output areas have to be supported with an according CPU program.

**Attention!**

The length entries for the input and output area have to be congruent with the Byte entry at the master project engineering. Otherwise no PROFIBUS communication is possible (slave failure)!




**Release memory in the CPU**

When you enter a length of 0, the according data do not occupy memory space in the CPU.

Entering 255 (memory limit) at the parameters PRN, DIAG and STAT you may also release memory areas of the CPU.

**Note!**

Using the CPU firmware version V2.2.0 or lower, the CPU 11x and the PROFIBUS-DP system support an address range from 0 to 255.

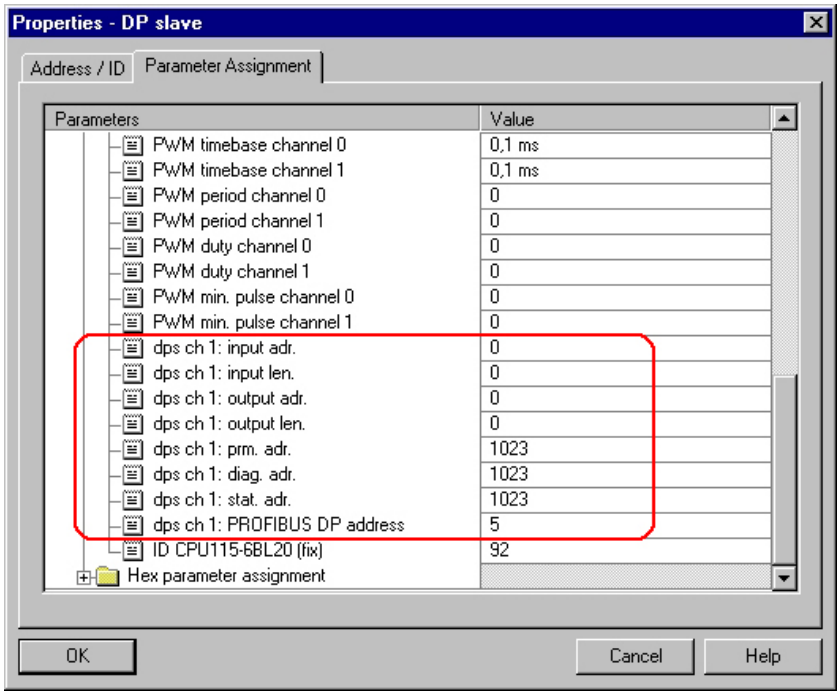
Starting with firmware version V3.0, the CPU 11x and PROFIBUS-DP system from VIPA support an address range from 0 to 1023.

The firmware level is to find on the label at the backside of the modules.

Here, the value 1023 deactivates *PRN, DIAG* and *STAT*.

**Description parameter data**

Via a double-click on the CPU 11xDP in the hardware configurator, the dialog window for the parameterization of the data areas for the PROFIBUS slave:




**Input add., length**

Address, from where on the data coming via PROFIBUS have to be stored in the CPU with the according "length".

When you enter a length of 0, the input areas do not occupy memory space in the CPU. The length is entered at the master in form of Byte groups for the PROFIBUS output section.

**Output add., length**

Address, where the data that has to be send via PROFIBUS is starting. Here too, you define the data width with *len*.

When you enter a length of 0, the input areas do not occupy memory space in the CPU. The length is entered at the master in form of Byte groups for the PROFIBUS input section.

**Prm. add.**
(24Byte fix)

The parameter data is an excerpt of the parameter telegram. The parameter telegram is created at master engineering and sent to the slave when:

- the CPU 11xDP is in start-up
- the connection between CPU 11xDP and master was interrupted, like e.g. disconnection of the bus connector.

A parameter telegram consists of PROFIBUS specific data (bus parameters) and user specific data, where the in-and output bytes at the CPU 11xDP are defined.

The user specific data (Byte 7 ... 31) are shown in the memory area of the CPU with a fixed length of 24Byte starting with the address selected in *prm*. This allows to proof the parameters that your slave gets form the master.

**Diag. add.**
(5Byte fix)

The wide range of diagnostic facilities of PROFIBUS-DP allow a fast error localization. The diagnostic messages are transferred via the bus and collected at the master.

The CPU 11xDP is sending diagnostic data either on master request or in error case. The diagnostic data contain:

- Norm diagnostic data (Byte 0 ... 5),
- Device related diagnostic data (Byte 6 ... 10)
- **User specific diagnostic data (Byte 11 ... 15)**

Via *diag* you define the start address of the 5Byte user specific diagnostic data in the CPU.

With targeted access to this area you may initialize and influence diagnostic.

**Note!**

More detailed information about structure and possibilities with diagnostic messages is under "Diagnostic functions".

**Stat. add.**
(2Byte fix)

The current status of the PROFIBUS communication can be seen in a 2Byte status area, stored in the periphery address range of the CPU starting at the status address.

**Note!**

More detailed information about the structure of a status message is under "Status message internal to CPU".

**PROFIBUS DP address**

Via this parameter you assign a PROFIBUS address to your PROFIBUS slave. Please regard that every PROFIBUS address may be assigned only once!

**Release areas in the CPU**

When entering the length 0, the according data do not occupy space in the CPU.

You may also release memory areas in the CPU by entering the address range limit (255 res. 1023 with CPU versions > 2.2.0) at the parameters *PRN, DIAG* and *STAT*.

# Diagnostic functions

**Overview**

The wide range of diagnostic functions of PROFIBUS DP allow a fast error localization. The diagnostic data is broadcasted via the bus and summarized at the DP master.

The CPU 11xDP is sending diagnostic data either on master request or in error case. For a part of the diagnostic data is stored in the periphery address area (Byte 11 ... 15) of the CPU, you may initialize and influence diagnostic. The diagnostic data contain:

- Norm diagnostic data (Byte 0 ... 5),
- Device related diagnostic data (Byte 6 ... 15).

**Structure**

The diagnostic data have the following structure:

*Norm diagnostic data*

| | |
|---|---|
| Byte 0 | Station state 1 |
| Byte 1 | Station state 2 |
| Byte 2 | Station state 3 |
| Byte 3 | Master address |
| Byte 4 | Ident no. (low) |
| Byte 5 | Ident no. (high) |

*Device related diagnostic data*

| | |
|---|---|
| Byte 6 | length and code device related diagnostic |
| Byte 7 | device related diagnostic messages |
| Byte 8 ... Byte 10 | reserved |
| Byte 11 ... Byte 15 | User specific diagnostic data are shown in the CPU periphery address range and may be altered and send to the master. |

**Norm diagnostic data**

More detailed information about the structure of the norm diagnostic data is available in the PROFIBUS Norm Papers. These papers are delivered by the PROFIBUS User Organization.

The slave norm diagnostic data have the following structure:

| Byte | Bit 7 ... Bit 0 |
|---|---|
| 0 | Bit 0: fixed at 0<br>Bit 1: Slave not ready for data transfer<br>Bit 2: Configuration data is not congruent<br>Bit 3: Slave has external diagnostic data<br>Bit 4: Slave does not support requested function<br>Bit 5: fixed at 0<br>Bit 6: Wrong parameterization<br>Bit 7: fixed at 0 |
| 1 | Bit 0: Slave needs new parameterization<br>Bit 1: Statistic diagnostic<br>Bit 2: fixed at 1<br>Bit 3: Response control active<br>Bit 4: Hold freeze command<br>Bit 5: Hold Sync command<br>Bit 6: reserved<br>Bit 7: fixed at 0 |
| 2 | Bit 6 ... Bit 0: reserved<br>Bit 7: Diagnostic data overflow |
| 3 | Master address after parameterization<br>FFh: Slave without parameterization |
| 4 | Ident no. High-Byte |
| 5 | Ident no. Low-Byte |

**Device related diagnostic data**

The device related diagnostic data give detailed information about the slave and the in-/output periphery. The length of the device related diagnostic data is fixed at 10Byte.

| Byte | Bit 7 ... Bit 0 |
|---|---|
| 6 | Bit 5 ... 0: Length device related diagnostic data<br>001010: Length 10Byte (fix)<br>Bit 7 ... 6: Code for device related diagnostic<br>00: Code 00 (fix) |
| 7 | Bit 7 ... 0: Device related diagnostic messages<br>12h: Error: Parameter data length<br>13h: Error: Configuration data length<br>14h: Error: Configuration entry<br>15h: Error: VPC3 buffer calculation<br>16h: Error: missing configuration data<br>17h: Error: Compare DP parameterization with project<br>40h: User defined diagnostic is valid |
| 8 ... 10 | reserved |
| **11 ... 15** | **User specific diagnostic data that are stored after the diagnostic status byte in the process image of the CPU. They may be overwritten and forwarded to the master.** |

**Initialize diagnostic**

In case of diagnostic the contents of Byte 11...15 of the device related diagnostic data are transferred into the process image of the CPU with the status byte as prefix. The position of this 6Byte diagnostic block in the process image is defined at the CPU parameter adjustment.

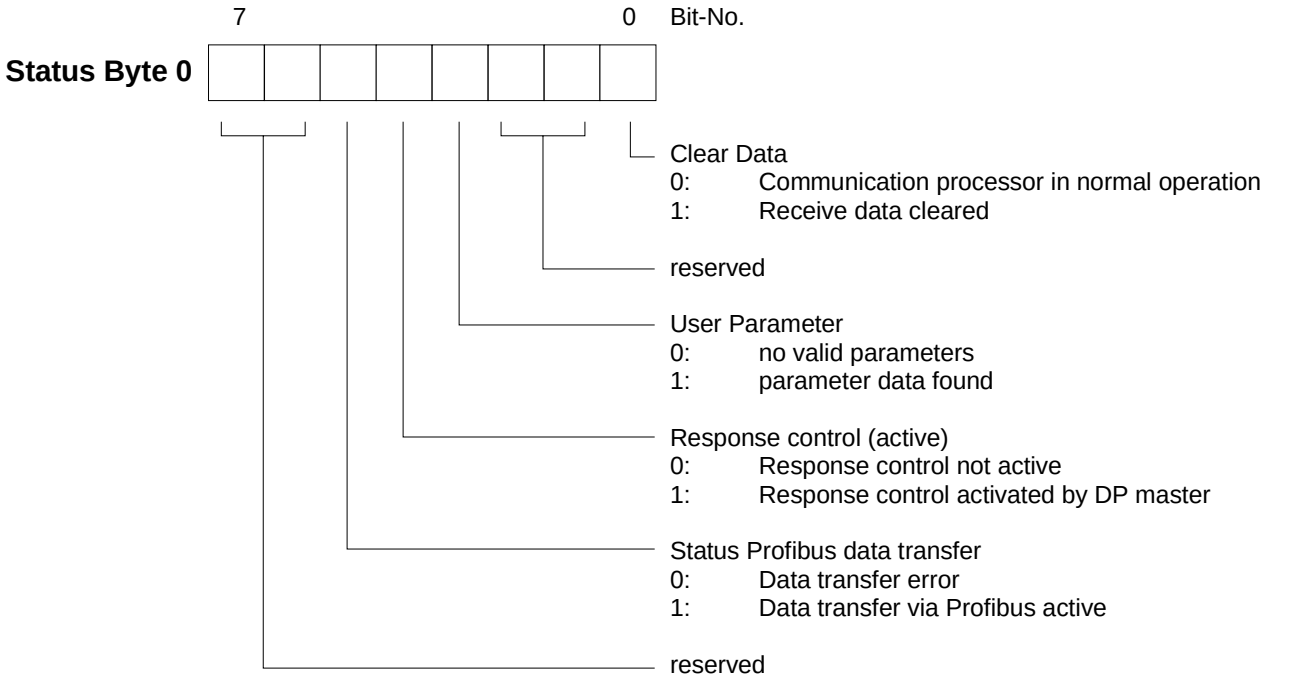
A status change 0 $\rightarrow$ 1 in the diagnostic status byte initializes a diagnostic and the according diagnostic telegram is transferred to the master.
**The status 0000 0011 is ignored!**

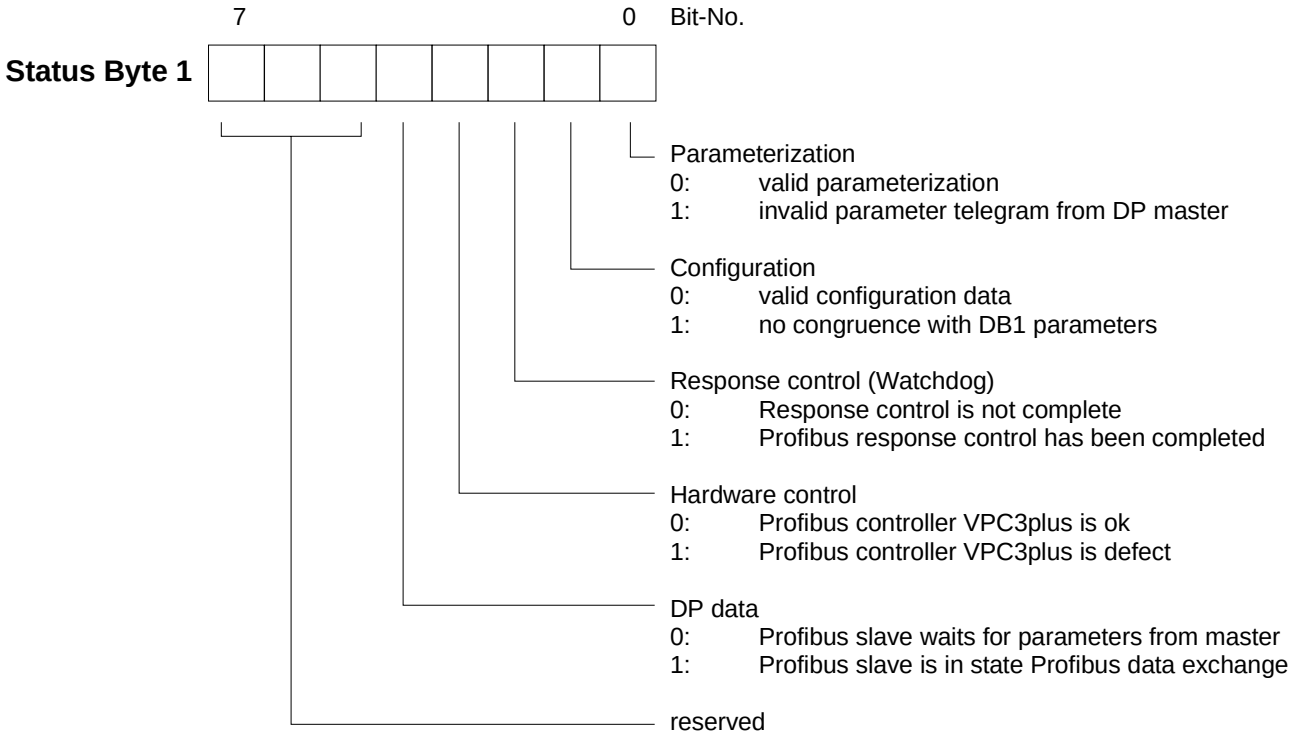
The diagnostic block in the CPU has the following structure:

| Byte | Bit 7 ... Bit 0 |
|---|---|
| 0 | Diagnostic status byte:<br>Bit 0: user specific diagnostic data<br>0: invalid diagnostic data<br>1: valid diagnostic data (initialize diagnostic)<br>Bit 1: Delete diagnostic<br>0: Delete diagnostic invalid<br>1: Delete diagnostic valid<br>Bit 7 ... 2: reserved |
| 1 ... 5 | Bit 7 ... 0: User specific diagnostic data equal to<br>Byte 11 ... 15 of the device related diagnostic |

# Status message internal to CPU

The current status of the PROFIBUS communication is shown in the status messages that are included in the periphery address range of the CPU. The status messages consist of 2Byte and have the following structure:



## Parameters

| | |
|---|---|
| **Clear Data** | In error case, the send and receive buffers are deleted. |
| **reserved** | These two Bits are reserved for future expansions. |
| **User parameters** | Shows the validity of the parameter data. The parameter data are entered at the master parameterization tool. |
| **Response control (active)** | Shows the activation status of the response control in the next higher PROFIBUS master. When the response control time is exceeded, the slave terminated the communication. |
| **Status PROFIBUS data transfer** | Status monitor of the communication with the higher master. With invalid configuration or invalid parameters, the communication is terminated and the error is shown via this Bit. |
| **Parameterization** | Shows the status of the parameter data. The length of the parameter data and the number of parameter bytes is compared. Only if these are identical and not more than 31Byte parameter data are transferred, the parameterization is correct. |
| **Configuration** | Status monitor of the configuration data that are send by the PROFIBUS master. The configuration is created in the master project engineering tool. |
| **Response control (Watchdog)** | The status of the response control in the PROFIBUS master is monitored. When the response control is active and the response time in the slave is exceeded, an error is shown here. |
| **Hardware control** | If a Bit is set here, this shows a failure in the PROFIBUS controller of the CPU 11xDP. Please contact the VIPA hotline. |
| **DP data** | This Bit is set at a transfer error. |

# 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. 125 participants.
- Per segment a max. of 32 participants is permitted.
- The max. segment length depends on the transfer rate:

| | | |
|---|---|---|
| 9.6 ... 187.5kBaud | → | 1000m |
| 500kBaud | → | 400m |
| 1.5MBaud | → | 200m |
| 3 ... 12MBaud | → | 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 baud rate. The slaves adjust themselves automatically on the baud 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.

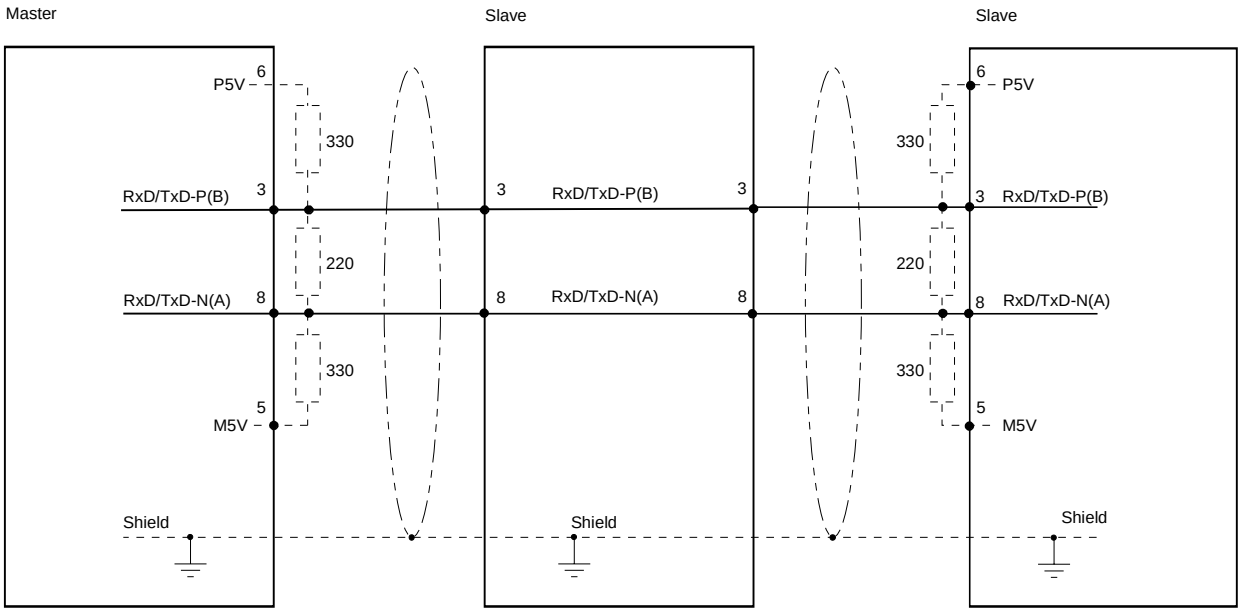
Max. 32 participants per segment are permitted. Within a segment the members are linear connected. The segments are connected via repeaters. The maximum segment length depends on the transfer rate.

PROFIBUS DP uses a transfer rate between 9.6kbaud and 12Mbaud, the slaves are following automatically. All participants are communicating with the same transfer rate.

The bus structure under RS485 allows an easy connection res. disconnection of stations as well as starting the system step by step. Later expansions don't have any influence on stations that are already integrated. The system realizes automatically if one partner had a fail down or is new in the network.

**Bus connection**

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



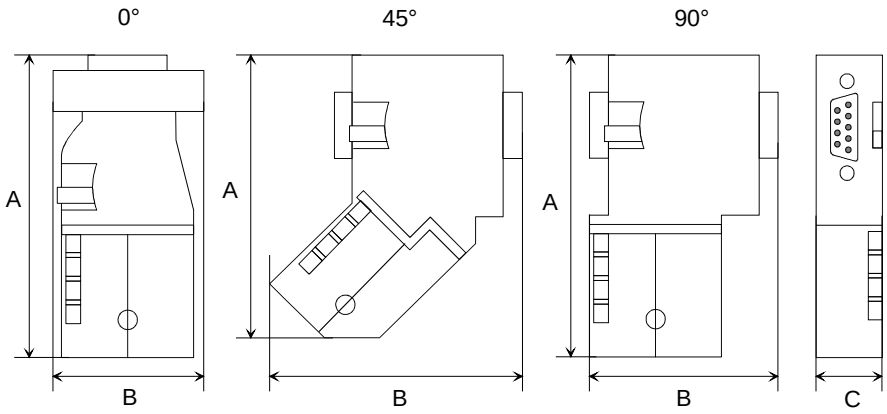

**Note!**

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 VIPA 972-0DP10 you may order the bus connector "EasyConn". This is a bus connector with switchable terminating resistor and integrated bus diagnostic.

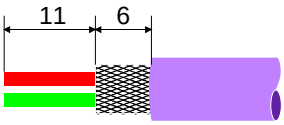


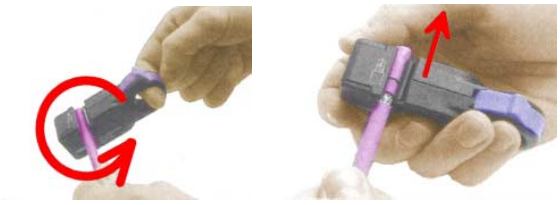

| | 0° | 45° | 90° |
|---|---|---|---|
| A | 64 | 61 | 66 |
| B | 34 | 53 | 40 |
| C | 15.8 | 15.8 | 15.8 |

all in mm

**Note!**

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.



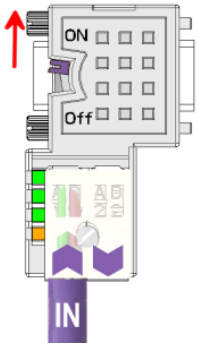
Dimensions in mm

Termination with "EasyConn"

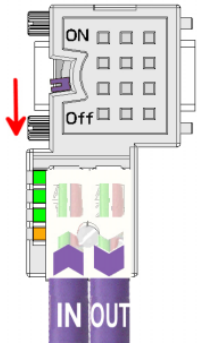
The "EasyConn" bus connector is provided with a switch that is used to activate a terminating resistor.

Wiring

1./last bus participant




further participants



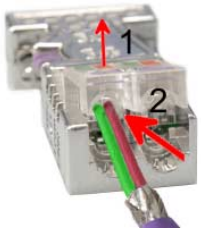

**Attention!**

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.

**Note!**

A complete description of installation and deployment of the terminating resistors is delivered with the connector.

Assembly





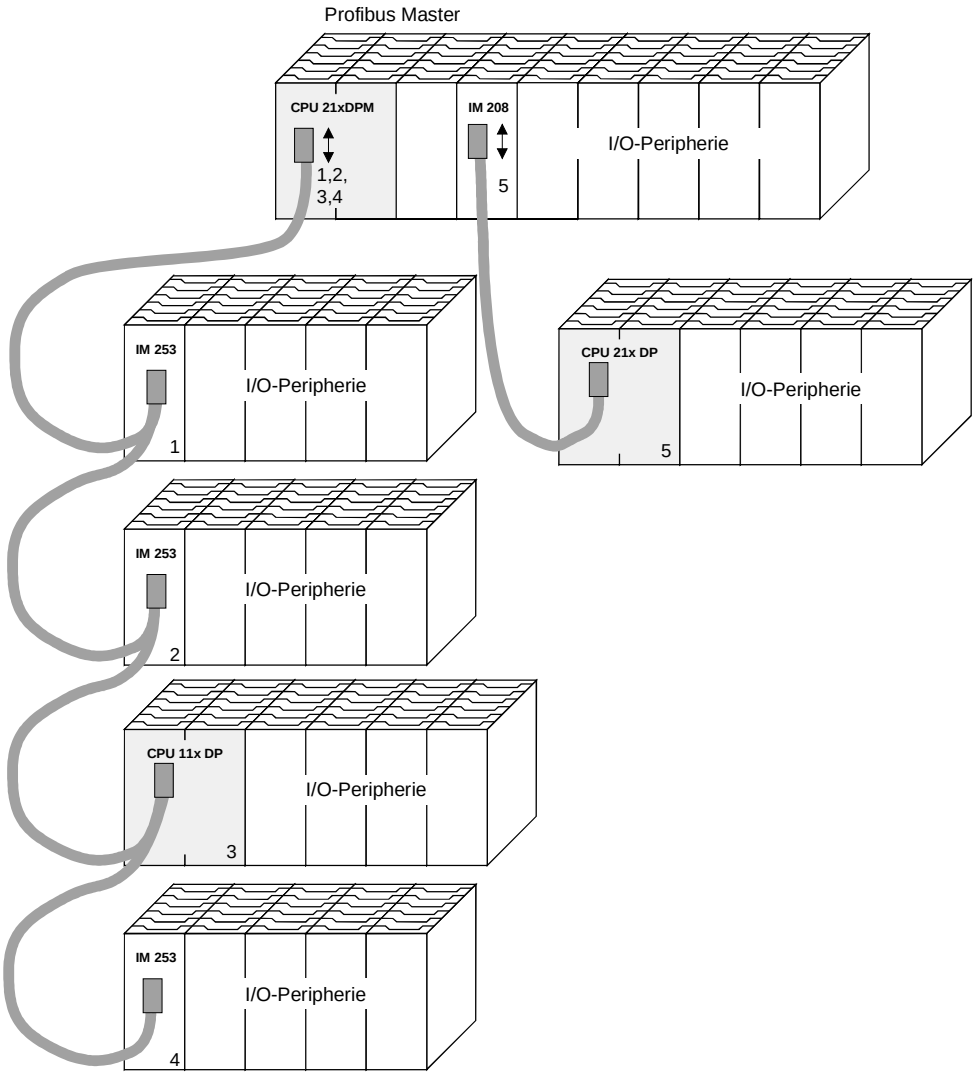
- Loosen the screw.
- Lift contact-cover.
- Insert both wires into the ducts provided (watch for the correct line color as below!)
- Please take care not to cause a short circuit between screen and data lines!
- Close the contact cover.
- Tighten screw (max. tightening torque 4Nm).

**Please note:** The green line must be connected to A, the red line to B!

**Examples for PROFIBUS networks**

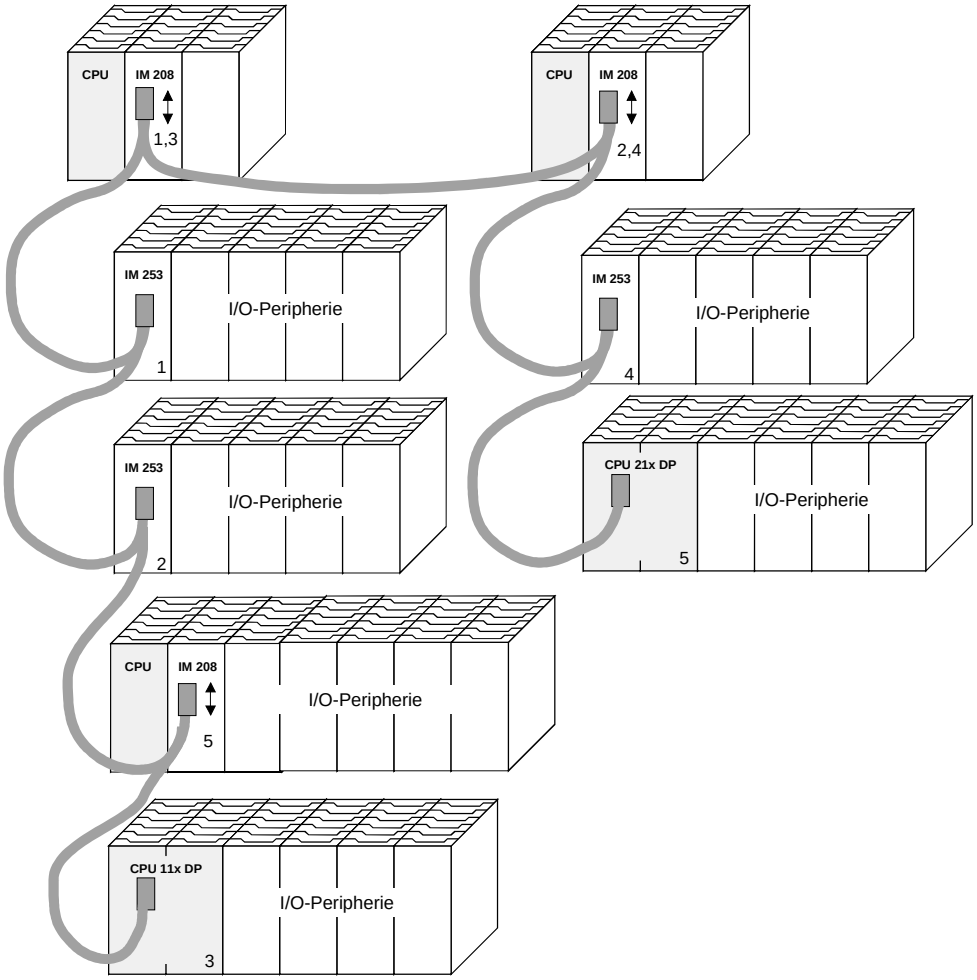
### One CPU and several master lines

The CPU should have a short cycle time to guarantee the actuality of the data in slave no. 5 (right side). This structure is only convenient when there are only slaves coupled to the slow line (left), which data actuality is not important. No alarm throwing modules should be placed here.



### Multi master system

Several master connections at one bus together with several slaves:



# Commissioning

**Overview**

- Build up your CPU 11xDP.
- Project the CPU 11xDP at the master.
- Project the CPU 11xDP at the slave together with in-/output periphery.
- Connect the CPU 11xDP with the PROFIBUS.
- Turn on the power supply.
- Transfer your project into the CPUs.

**Assembly**

Build up your CPU 11xDP.

**Note!**
To avoid transfer irritations from reflections, the bus cable has always to be terminated with its ripple resistor at the cable ends!

**Configuration at the master**

Project your CPU 11xDP in your master system. To engineer the System 100V PROFIBUS slaves from VIPA, you have to include the GSD VIPA04Dx.GSD.

Transfer your project in the master CPU.

**Configuration CPU 11xDP and I/O periphery**

Project your CPU 11xDP at the slave. You need the GSD VIPA_11x.GSD.

The in-/output periphery is automatically overlaid in the CPU address range. The address allocation may be altered in the hardware configurator from Siemens at any time.

Transfer your project via MPI in the CPU 11xDP.

**Power supply**

The CPU 11xDP has an integrated mains power supply. It has to be provided with DC 24V.

Via the supply voltage not only the CPU and the bus coupler is provided but also the connected modules via the backplane bus. Please regard that the internal power supply may provide the backplane bus with max. 3A.

PROFIBUS and backplane bus are isolated.

**Transfer project**

The transfer of the hardware configuration into the CPU takes place via MPI.

- Connect your PG res. the PC via MPI with the CPU.
  If your programming device has no MPI slot, you may use the VIPA Green Cable to establish a serial point-to-point connection.

  The Green Cable has the order no. VIPA 950-0KB00 and only be used with the VIPA CPUs with MP$^2$I-Slot.
- Configure the MP interface of your PC.
- With **PLC** > *Load to module* in hardware configurator you transfer your project into the CPU.
- For the additional security copy of your project on MMC, you plug-in a MMC and transfer the user application to the MMC via **PLC** > *Copy RAM to ROM*.
  During write operation the MC-LED on the CPU blinks. Due to the system, the successful writing is signalized too soon. The write command has only been completed, when the LED extinguishes.

**Attention!**

Please regard the hints for deploying the Green Cable and the MP$^2$I jack at "Hardware description".

**Initialization phase**

After the start-up, the CPU 11xDP executes a self-test. It proofs its internal functions, the communication via backplane bus and to PROFIBUS.

At successful test the parameters are read from the CPU and the PROFIBUS slave parameters are proofed.

After successful boot procedure the CPU 11xDP switches to "READY".

Communication problems at the backplane bus cause the CPU 11xDP to go in STOP and start again after app. 2 seconds. When the test has been completed positive, the RD-LED blinks.

At starting communication, the DE-LED is on.

# Example

**Task description**

This example shows a communication between the master system CPU 214DPM and a slave system CPU 11xDP.

Counter values have to be transferred via PROFIBUS and monitored at the output section of the partner.

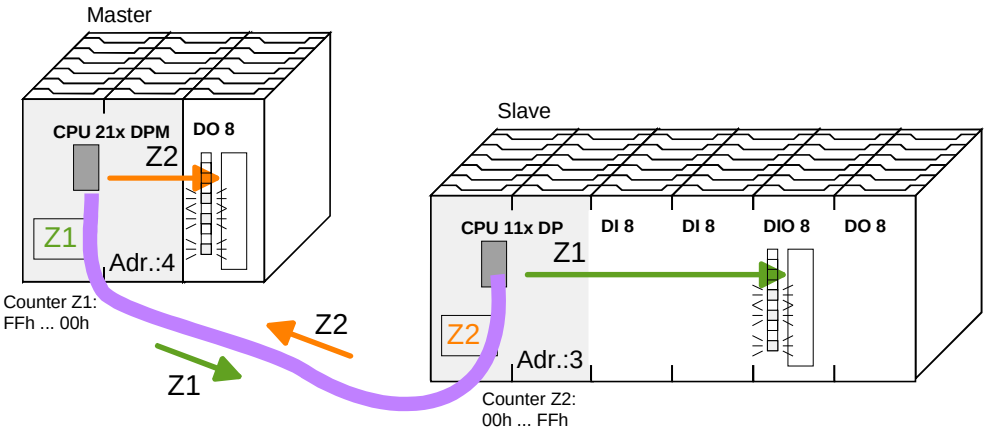
**Task description in detail**

The CPU 214DPM shall count from FFh to 00h and transfer the counter value to the output section of the PROFIBUS master cyclically. The master should send this value to the slave of the Micro-PLC CPU 11xDP.

The received value has to be stored in the input periphery area in the CPU and be monitored in the output section at address 0.

Vice versa, the Micro-PLC CPU 11xDP shall count from 00h to FFh and transfer the counter value to the master.

This value should be monitored at the output module (address 0) of the CPU 214DPM.




**Configuration data**

**CPU 21xDPM**

| | | |
|---|---|---|
| Counter value: | MB 0 (FFh ... 00h) | |
| PROFIBUS address: | 4 | |
| Input area: | address 10 | length: 2Byte |
| Output area: | address 20 | length: 2Byte |

**CPU 11xDP**

| | | |
|---|---|---|
| Counter value: | MB 0 (00h...FFh) | |
| PROFIBUS address: | 3 | |
| Input area: | address 30 | length: 2Byte |
| Output area: | address 40 | length: 2Byte |
| Parameter data: | address 50 | length: 24Byte(fix) |
| Diagnostic data: | address 60 | length: 6Byte (fix) |
| Status data: | address 100 | length: 2Byte (fix) |

## Configuration CPU 21xDPM

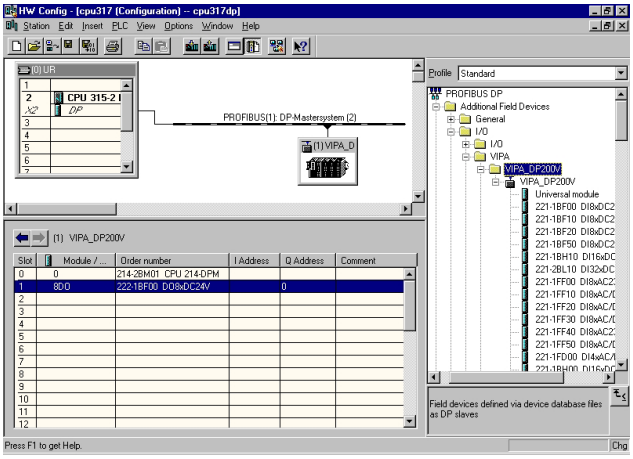
To be compatible with the Siemens SIMATIC Manager, you have to execute the following steps for the System 200V:

- Start the hardware configurator from Siemens
- Project a CPU 315-2DP with DP master system (address 2).
  Use for the project engineering the CPU 6ES7-315-2AF03 V 1.2 from Siemens of the Hardware catalogue.
- Add a PROFIBUS slave **"VIPA_CPU21x"** with address 1. The VIPA_21x.GSD from VIPA is required.
- Include the CPU at slot 0 of the slave system **214-2BM01**.
- Include the output module 222-1BF00 at plug-in location 1.

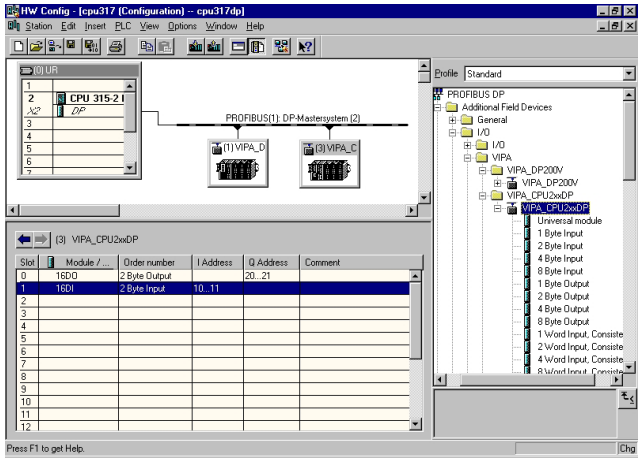
To connect your CPU 11xDP with the PROFIBUS master you have to follow these steps:

- Add the PROFIBUS slave **"VIPA_CPU11xDP"** (address 3). The DP slave is in the hardware catalog under:
  *PROFIBUS-DP > Additional field devices > I/O > VIPA > VIPA_System_100V*.
- Assign memory areas of the CPU to the in- and output of the PROFIBUS-DP master section in form of Byte blocks. For this, you have to include the "2Byte output" element on plug-in location 0 and select the output address 20. Include the "2Byte input" element on plug-in location 1 and select the input address 10.
- Save your project.

Including directly connected modules



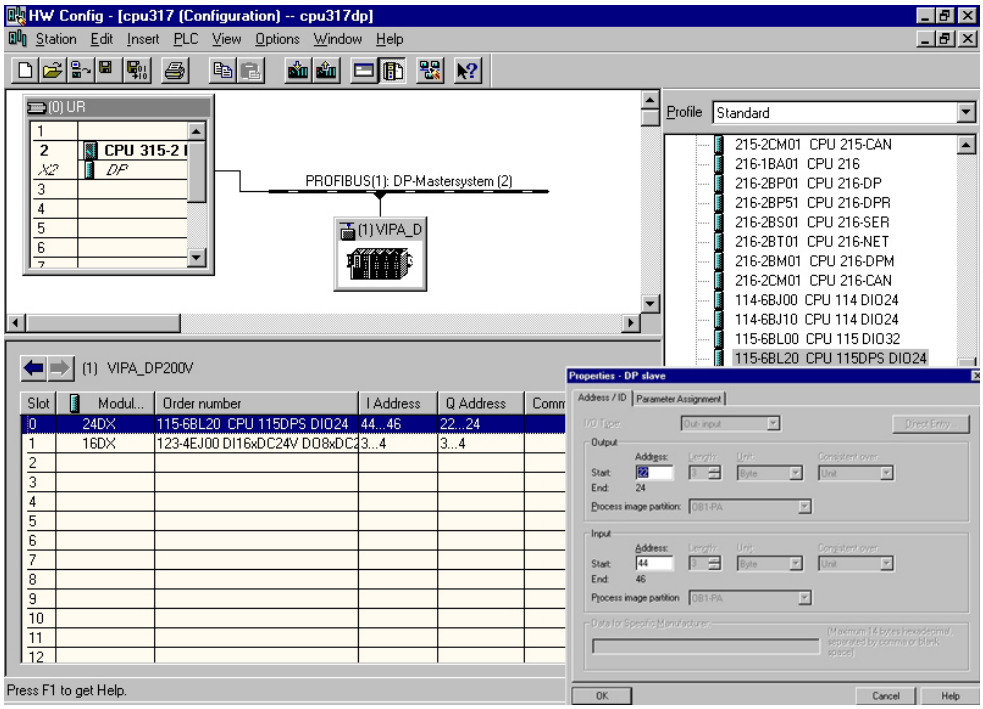

Including CPU 11xDP



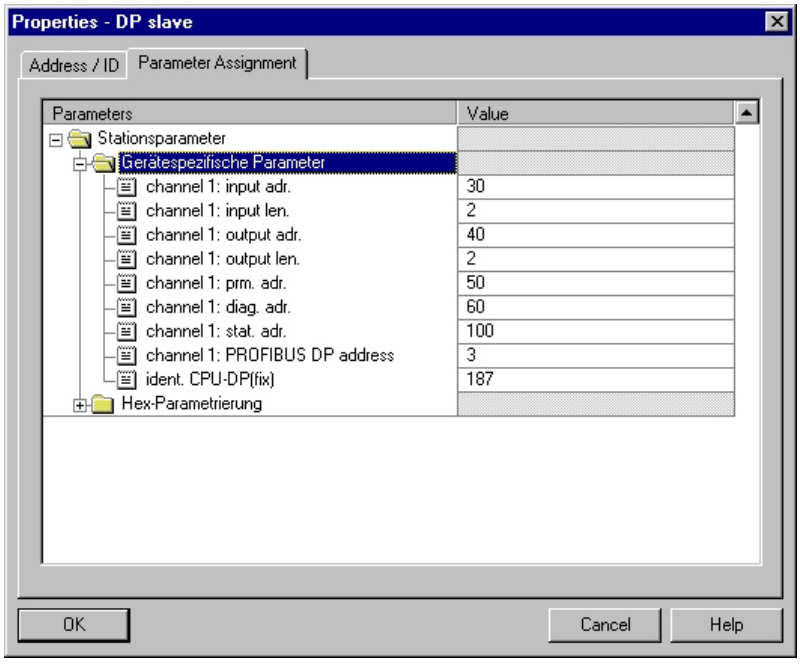
**Configuration CPU 11xDP**

To be compatible with the Siemens SIMATIC Manager, you have to execute the following steps for the System 100V:

- Start the hardware configurator from Siemens
- Project a CPU 315-2DP with DP master system (address 2).
  Use for the project engineering the CPU 6ES7-315-2AF03 V 1.2 from Siemens of the Hardware catalogue
- Add a PROFIBUS slave "VIPA_CPU11x" with address 1.
- Include the CPU 11xDP at plug-in location 0 of the slave system




- Choose the following parameters in the parameter window of the CPU 11xDP:




- Save your project.

**User application CPU 214DPM**

The user application of the CPU 214DPM is for two purposes, shared on two OBs:

- Test communication via the control byte.
  Load the input byte from PROFIBUS and monitor the value at the output module.

  OB 1 (cyclic call)

```
L    B#16#FF
T    AB  20             Control byte for slave CPU

L    B#16#FE            Load control value 0xFE
L    EB  10             Has the control byte been
<>I                     transmitted correctly?
BEB                     No -> End

                        ------------------------------
                        Data transfer via PROFIBUS

L    EB  11             Load input byte 11 (output data
                        CPU11xDP) and
T    AB   0             transfer into output byte 0

BE
```

- Read counter value from the MB 0, decrement, save in MB 0 and send via PROFIBUS to CPU 11xDP.

  OB 35 (time OB)

```
L    MB   0             Counter from 0xFF to 0x00
L    1
-I
T    MB   0

T    AB  21             Transfer to output byte 21
                        (input data CPU11xDP)
BE
```

Now the programming of the CPU 214DPM is complete as well as the PROFIBUS communication at both sides.

Transfer your project into the CPU 214DPM using the PLC functions via MPI.

**User application CPU 11xDP**

Like shown above, the user application is for two purposes, shared on two OBs:

- Load input byte from the PROFIBUS slave and monitor the value at the output module.

OB 1 (cyclic call)

```
L     EW 100              Load status data and store in
T     MW 100              marker word

UN    M  100.5            Commissioning by the DP master
BEB                       complete? No -> End

U     M  101.4            Valid receive data?
BEB                       No -> End
L     B#16#FF             Load control value and
L     EB  30              compare control byte (1st
<>I                       input byte)
BEB                       Received data without
                          valid values
L     B#16#FE             Control byte for master CPU
T     AB  40
                          ------------------------------
                          Data transfer via PROFIBUS

L     EB  31              Load input byte 31 (input data
                          PROFIBUS slaves) and
T     AB   0              transfer to output byte 0

BE
```

- Read counter value from MB 0, increment, save in MB 0 and transfer to the DP master via PROFIBUS.

OB 35 (time-OB)

```
L     MB   0              Counter from 0x00 to 0xFF
L     1
+I
T     MB   0

T     AB  41              Transfer counter value to
                          output byte 41 (output data
                          PROFIBUS slaves)
BE
```

# Chapter 5 Deployment Micro-PLC CPU 11xSER

**Overview**

Content of this chapter is the deployment of the Micro-PLC CPU 11xSER with RS232/RS485 interface.

Here you'll find all information about the deployment of the serial interfaces of the CPU 11xSER.

**Content**

# Principles

**General**

The CPU 11xSER provides serial interfacing facilities between the processes of different source and destination systems. For the serial communication the
CPU 115-6BL1x has a RS232 interface and the
CPU 115-6BL3x has a RS485 interface.

**Protocols**

The CPU 11xSER supports the ASCII, STX/ETX, 3964R, USS and Modbus protocols and procedures.

**Parameterization**

The parameterization happens during runtime by means of the SFC 216 (SER_CFG). The parameters for STX/ETX, 3964R, USS and Modbus have to be stored in a DB.

**Communication**

With the help of SFCs you control the communication. The sending is executed with the SFC 217 (SER_SND) and the reception via SFC 218 (SER_RCV).

Another call of the SFC 217 SER_SND, 3964R, USS and Modbus provides you via RetVal with a return value which contains among others recent information about the acknowledgement of the partner.

The protocols USS and Modbus allows you to read the acknowledgement telegram by calling the SFC 218 SER_RCV after a SER_SND.

The SFCs are included in the consignment of the CPU 11xSER.

**Overview over the SFCs for the serial communication**

The following SFCs are deployed for the serial communication:

| SFC | | Description |
|---|---|---|
| SFC 207 | SER_CTRL | Modem functionality |
| SFC 216 | SER_CFG | RS232/RS485 Parameterization |
| SFC 217 | SER_SND | RS232/RS485 Send |
| SFC 218 | SER_RCV | RS232/RS485 Receive |

# Protocols and procedures

**Overview**

The CPU 11xSER 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-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 is to find at ftp.vipa.de.

## STX/ETX

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

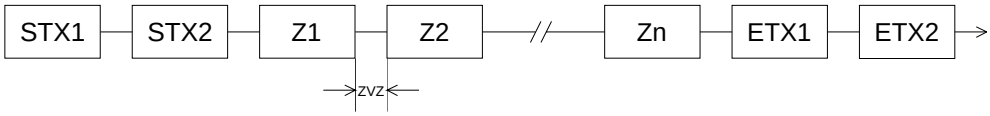
The STX/ETX procedure is suitable for the transfer of ASCII characters. It does not use block checks (BCC). Any data transferred from the periphery must be preceded by a Start followed by the data characters and the end character.

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 send from the CPU to a peripheral device, any user data is handed to the SFC 217 (SER_SND) and is transferred with added Start- and End-ID to the communication partner.

*Message structure:*




You may define up to 2 start and end characters.

You may work with 1, 2 or no Start- and with 1, 2 or no End-ID. 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). If no End-ID is defined, all read characters are transferred to the CPU after a parameterizable character delay time (Timeout).

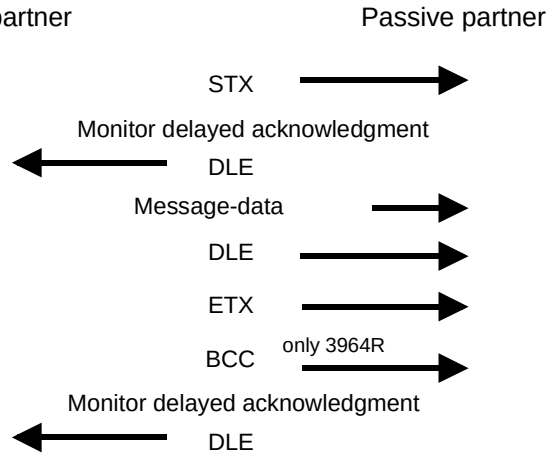
## 3964R

The 3964R procedure controls the data transfer of a point-to-point link between the CPU 11xSER and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free receipt.

The procedure employs the following control characters:

- STX **S**tart of **T**e**x**t
- DLE **D**ata **L**ink **E**scape
- ETX **E**nd of **T**e**x**t
- BCC **B**lock **C**heck **C**haracter
- NAK **N**egative **A**c**k**nowledge

### Procedure




You may transfer a maximum of 255byte per message.

**Note!**

When a DLE is transferred as part of the information it is repeated to distinguish between data characters and DLE control characters that are used to establish and to terminate the connection (DLE duplication). The DLE duplication is reversed in the receiving station.

The 3964R procedure requires that a lower priority is assigned to the communication partner. When communication partners issue simultaneous send commands, the station with the lower priority will delay its send command.

## USS

The USS protocol (**U**niverselle **s**erielle **S**chnittstelle = universal serial interface) is a serial transfer protocol defined by Siemens for the drive and system components. This allows to build-up a serial bus connection between a superordinated master and several slave systems.

The USS protocol enables a time cyclic telegram traffic by presetting a fix telegram length.

The following features characterize the USS protocol:

- Multi point connection
- Master-Slave access procedure
- Single-Master-System
- Max. 32 participants
- Simple and secure telegram frame

You may connect 1 master and max. 31 slaves at the bus where 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 SFC 218 SER_RCV.

The telegrams for send and receive have the following structure:

*Master-Slave telegram*

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

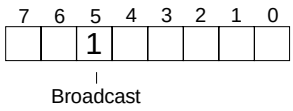
*Slave-Master telegram*

| STX | LGE | ADR | PKE | | IND | | PWE | | ZSW | | HIW | | BCC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 02h | | | H | L | H | L | H | L | H | L | H | L | |

where

| | | | | |
|---|---|---|---|---|
| STX: | Start sign | | STW: | Control word |
| LGE: | Telegram length | | ZSW: | State word |
| ADR: | Address | | HSW: | Main set value |
| PKE: | Parameter ID | | HIW: | Main effective value |
| IND: | Index | | BCC: | Block Check Character |
| PWE: | Parameter value | | | |

Broadcast with set Bit 5 in ADR-Byte




A request can be directed to a certain slave ore be send to all slaves as broadcast message. For the identification of a broadcast message you have to set Bit 5 to 1 in the ADR-Byte. Here the slave address (Bit 0 ... 4) is ignored. In opposite to a "normal" send command, the broadcast does not require a telegram evaluation via SFC 218 SER_RCV. Only write commands may be send 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 SFC 218 SER_RCV.

The request telegrams send by the master and the respond telegrams of a slave have the following structure:

| Start sign | Slave address | Function Code | Data | Flow control | End sign |
|---|---|---|---|---|---|

Broadcast with slave address = 0

A request can be directed to a special slave or at all slaves as broadcast message. To mark a broadcast message, the slave address 0 is used.

In opposite to a "normal" send command, the broadcast does not require a telegram evaluation via SFC 218 SER_RCV.

Only write commands may be send as broadcast.

ASCII, RTU mode

Modbus offers 2 different transfer modes:

- 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.

The mode selection happens during runtime by using the SFC 216 SER_CFG.

# Deployment of the serial interface

**Overview**

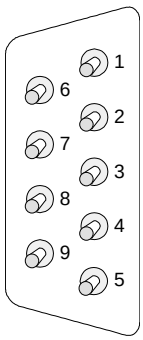
The CPU has got a RS232- (Best.-Nr.: 115-6BL1x) or RS485-interface (Best.-Nr.: 115-6BL3x). The booth interfaces are following described.

## RS232 interface

- Logical signals as voltage levels (compatible to COM of PC)
- Point-to-point links with serial full-duplex transfer in 2-wire technology over distances of up to 15m
- Data transfer rate up to 115.2kBaud
- Receive buffer and send buffer each with 2x256byte
- The maximum telegram length is 255byte

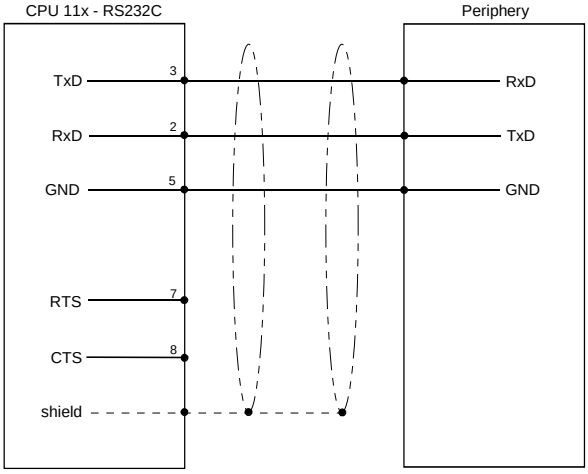
**RS232**

*9pin plug*




| Pin | RS232 |
|---|---|
| 1 | CD- |
| 2 | RxD |
| 3 | TxD |
| 4 | DTR |
| 5 | GND |
| 6 | DSR- |
| 7 | RTS- |
| 8 | CTS- |
| 9 | RI- |

Connection RS232

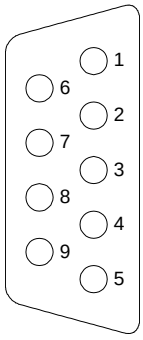



## RS485 interface

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

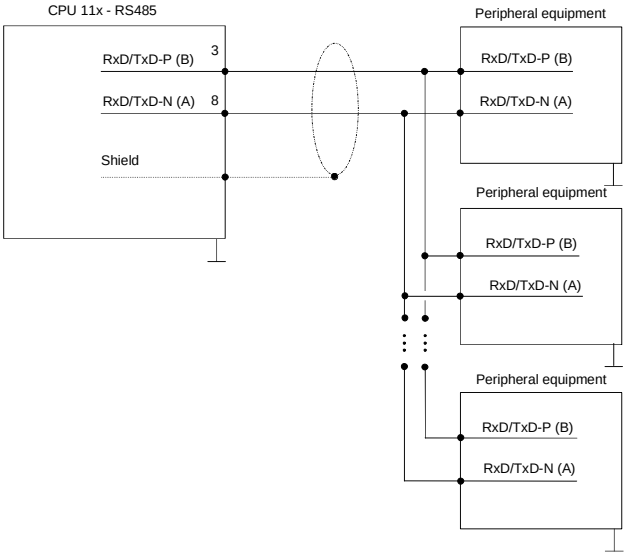
**RS485**

*9pin jack*




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

Connection RS485



# Principals of the data transfer

**Overview**

The data transfer is handled during runtime by using SFCs. The principles of data transfer are the same for all protocols and is shortly illustrated in the following.

**Principle ASCII, STX/ETX, 3964R, Modbus-Master and USS**

Data that is into the according data channel by the CPU, is stored in a send buffer with a size of 2x256byte and then put out via the interface.

When the interface receives data, this is stored in a receive buffer with a size of 2x256byte and can there be read by the CPU.

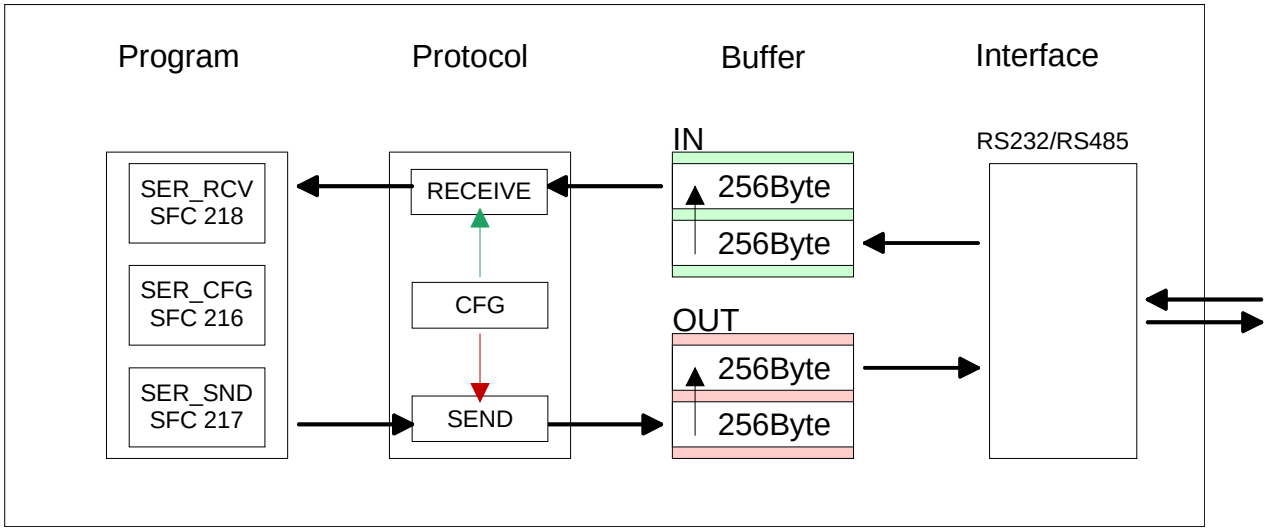
If the data is transferred via a protocol, the adaptation of the data to the according protocol happens automatically.

In opposite to ASCII and STX/ETX, the protocols 3964R, USS and Modbus require the acknowledgement of the partner.

An additional call of the 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 call of the SFC 218 SER_RCV.

CPU 11xSER



**Principles for Modbus Slave**

Data that the CPU has to provide for the Modbus master are stored in a send buffer with a size of 2x256byte. The data remain in the send buffer until they are overwritten by the CPU. Here the data can be requested by the master (function code 02h, 04h).

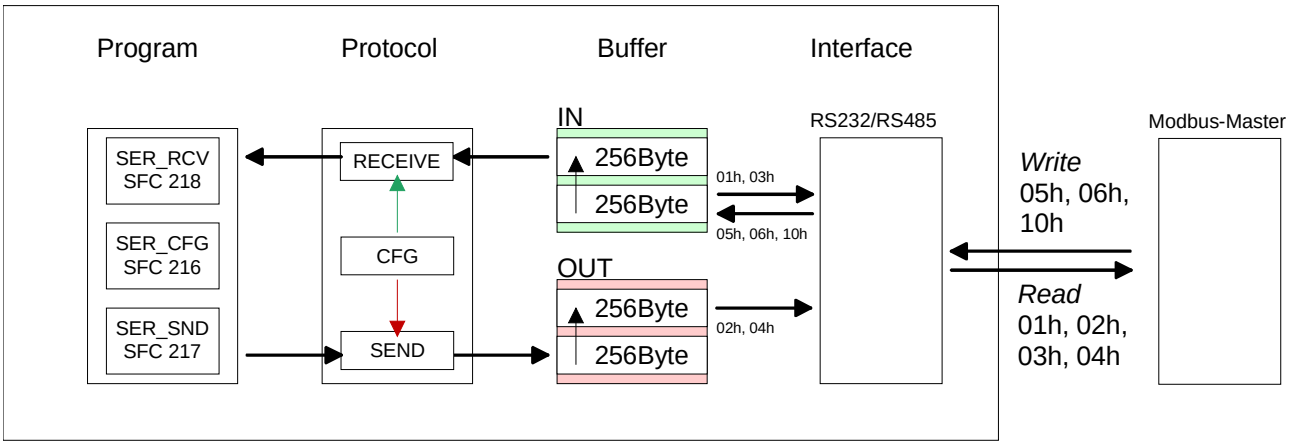
If the interface receives data from the master (function code 05h, 06h, 10h) these are stored in a receive buffer with a size of 2x256byte and may there be read by the CPU.

The embedding of the data into the Modbus protocol happens automatically.

Please regard that the Modbus master may access the IN res. OUT buffer by according presetting of the read function code. By means of a read access to the IN buffer (function code 01h, 03h) the master may read data that it has sent to the Modbus slave before. The data remain in the buffer until they are overwritten by the Modbus master.

The following picture shows the communication principle. More information is also to be found in the chapter "Modbus slave function codes" further below.

CPU 11xSER



# Parameterization

**SFC 216 (SER_CFG)**

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

Please regard that not all protocols support the complete value range of the parameters. More detailed information is to be found in the description of the according parameter.

**Note!**

Please regard that the SFC 216 is not called again during a communication because as a result of this all buffers are cleared.

If you don't want to alter the communication parameter any more, you should place the call of the SFC 216 in the start-up OB OB 100.

| Name | Declaration | Type | Comment |
|---|---|---|---|
| Protocol | IN | BYTE | No. of protocol |
| Parameter | IN | ANY | Pointer to protocol-parameters |
| Baudrate | IN | BYTE | No of Baud rate |
| CharLen | IN | BYTE | 0=5bit, 1=6bit, 2=7bit, 3=8bit |
| Parity | IN | BYTE | 0=None, 1=Odd, 2=Even |
| StopBits | IN | BYTE | 1=1bit, 2=1,5bit, 3=2bit |
| FlowControl | IN | BYTE | 1 (fix) |
| RetVal | OUT | WORD | Error Code ( 0 = OK ) |

**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
7: Modbus RTU Slave
8: Modbus ASCII Slave

**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 in a time window of 10ms) |

**Note!**

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

*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) |
| DBW4: | CharTimeout | WORD | (Char. delay time in 10ms time window) |
| DBW6: | ConfTimeout | WORD | (Ackn. delay time in 10ms time window) |

*Data block at USS*

| | | | |
|---|---|---|---|
| DBW0: | Timeout | WORD | (Delay time in 10ms time grid) |

*Data block at Modbus-Master*

| | | | |
|---|---|---|---|
| DBW0: | Timeout | WORD | (Respond delay time in 10ms time grid) |

*Data block at Modbus-Slave*

| | | | |
|---|---|---|---|
| DBB0: | Address | BYTE | (Address in the Modbus network) |
| DBW1: | Timeout | WORD | (Respond delay time in 10ms time grid) |

**Baud rate**

Velocity of data transfer in bit/s (Baud).

| | | | |
|---|---|---|---|
| 01h: 150 Baud | 05h: 1800 Baud | 09h: 9600 Baud | 0Dh: 57600 Baud |
| 02h: 300 Baud | 06h: 2400 Baud | 0Ah: 14400 Baud | 0Eh: 115200 Baud |
| 03h: 600 Baud | 07h: 4800 Baud | 0Bh: 19200 Baud | |
| 04h: 1200 Baud | 08h: 7200 Baud | 0Ch: 38400 Baud | |

**CharLen**

Number of data bits where a character is mapped to.

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

Supported values:

| Bit | ASCII | STX/ETX | 3964R | USS | Modbus RTU | Modbus ASCII |
|---|---|---|---|---|---|---|
| 5 | x | | x | | | |
| 6 | x | x | x | | | |
| 7 | x | x | x | | | x |
| 8 | x | x | x | x | x | x |

**Parity**

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

0: NONE 1: ODD 2: EVEN

**StopBits**

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

1: 1bit 2: 1.5bit 3: 2bit

1.5bit can only be used with CharLen 5 at this number of data 2bit is not allowed.

**FlowControl**

With this bit you affect the behavior from signal **R**equest **t**o **s**end

"0" = RTS off

"1" = RTS is "0" at Receive (AutoRTS)
RTS is "1" at Send (AutoRTS)

"2" = HW flow (only at ASCII protocols)

Note: For RS485 FlowControl is not evaluated.
It is set automatically to "1" (AutoRTS)!

**RetVal (Error message)**

*Error ID*:

| Error code | Description |
|---|---|
| 0000h | no error |
| 809Ah | interface not found |
| 8x24h | Error at SFC-Parameter x, with x:<br>1: Error at "Protocol"<br>2: Error at "Parameter"<br>3: Error at "Baudrate"<br>4: Error at "CharLength"<br>5: Error at "Parity"<br>6: Error at "StopBits"<br>7: Error at "FlowControl" |
| 809xh | Error in SFC parameter value x, where x:<br>1: Error at "Protocol"<br>3: Error at "Baudrate"<br>4: Error at "CharLength"<br>5: Error at "Parity"<br>6: Error at "StopBits"<br>7: Error at "FlowControl" |
| 8092h | Access error in parameter DB (DB too short) |
| 828xh | Error in parameter x of DB parameter, where x:<br>1: Error 1st parameter<br>2: Error 2nd parameter<br>... |

# Communication

**Overview**

The communication happens via the send and receive blocks SFC 217 (SER_SND) and SFC 218 (SER_RCV).

If data is transferred by means of a protocol, the embedding of the data into the according protocol happens automatically. Depending on the protocol you have to regard the following aspects

ASCII
STX/ETX

With ASCII res. STX/ETX the sending of the data happens without acknowledgement of the partner.

3964R

Another call of the SFC 217 SER_SND provides you via RetVal with a return value, which contains among others recent information about the acknowledgement of the partner.

Modbus master
USS

Sending happens with acknowledgement of the partner. Another call of the SFC 217 SER_SND provides you via RetVal with a return value which contains among others recent information about the acknowledgement of the partner. After the transfer with SER_Send you receive the acknowledgement telegram of the partner by calling the SFC 218 SER_RCV.

**Note!**

Please regard that the SFC 216 is not called again during a communication because as a result of this all buffers are cleared.

**SFC 217 (SER_SND)**

This block allows to send data via the serial interface.

| Name | Declaration | Type | Comment |
|---|---|---|---|
| DataPtr | IN | ANY | Pointer to Data Buffer for sending data |
| DataLen | OUT | WORD | Length of data sent |
| RetVal | OUT | WORD | Error Code ( 0 = OK ) |

**DataPtr**

Here you define a range of the type Pointer for the send buffer where the data that has to be send is stored. You have to set type, start and length.

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

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

**DataLen**

Word where the number of sent bytes is stored.

At **STX/ETX** and **3964R**, the length set in DataPtr or 0 is entered.

At **ASCII** if data were sent by means of 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.

**RetVal (Return value)**

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

Protocol specific RetVal values

*ASCII*

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

*STX/ETX*

| Value | Description |
|---|---|
| 9000h | Buffer overflow (no data send) |
| 9001h | Data too long (>256byte) |
| 9002h | Data too short (0byte) |
| 9004h | Character not allowed |

*3964R*

| Value | Description |
|---|---|
| 2000h | Send ready without error |
| 80FFh | NAK received - error in communication |
| 80FEh | Data transfer without acknowledgement of partner or error at acknowledgement |
| 9000h | Buffer overflow (no data send) |
| 9001h | Data too long (>256byte) |
| 9002h | Data too short (0byte) |

*USS*

| Value | Description |
|---|---|
| 2000h | Send ready without error |
| 8080h | Receive buffer overflow (no space for receipt) |
| 8090h | Acknowledgement delay time exceeded |
| 80F0h | Wrong checksum in respond |
| 80FEh | Wrong start sign in respond |
| 80FFh | Wrong slave address in respond |
| 9000h | Buffer overflow (no data send) |
| 9001h | Data too long (>256byte) |
| 9002h | Data too short (<2byte) |

*Modbus RTU/ASCII Master*

| Value | Description |
|---|---|
| 2000h | Send ready without error |
| 2001h | Send ready with error |
| 8080h | Receive buffer overflow (no space for receipt) |
| 8090h | Acknowledgement delay time exceeded |
| 80F0h | Wrong checksum in respond |
| 80FDh | Length of respond too long |
| 80FEh | Wrong function code in respond |
| 80FFh | Wrong slave address in respond |
| 9000h | Buffer overflow (no data send) |
| 9001h | Data too long (>256byte) |
| 9002h | Data too short (<2byte) |

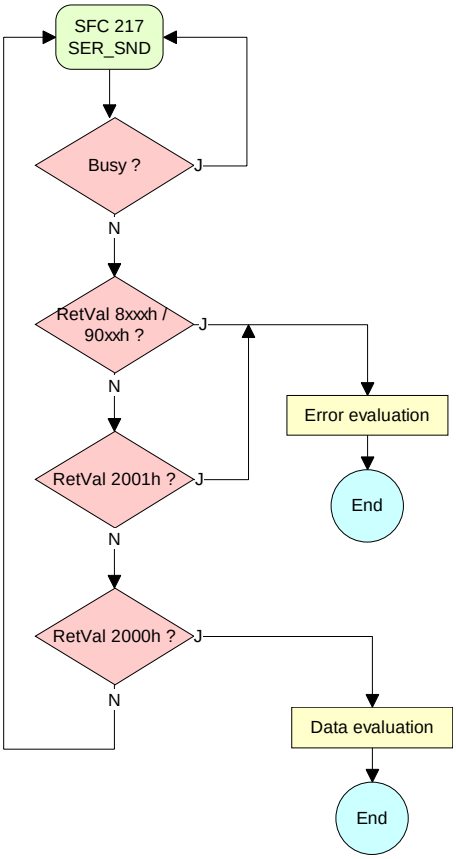
*Modbus RTU/ASCII Slave*

| Value | Description |
|---|---|
| 0000h | Send data - ready |
| 9001h | Data too long (>256byte) |

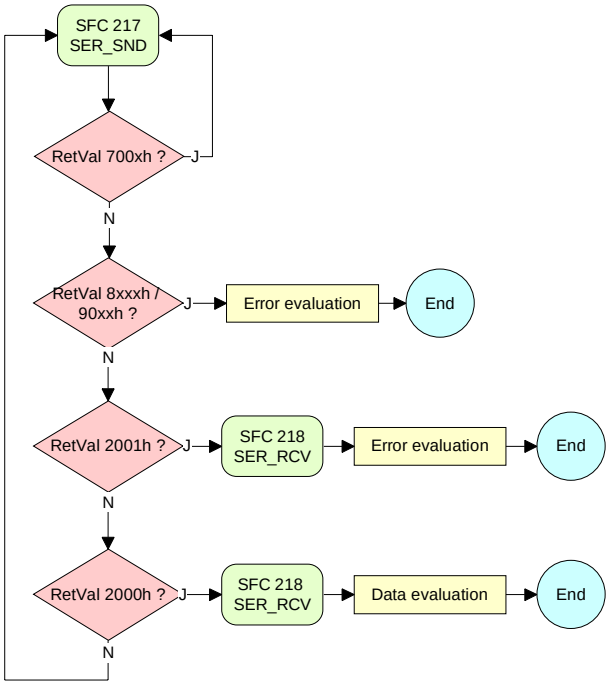
**Principles of programming**

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

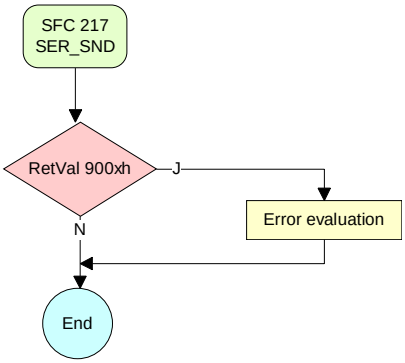
### 3964R



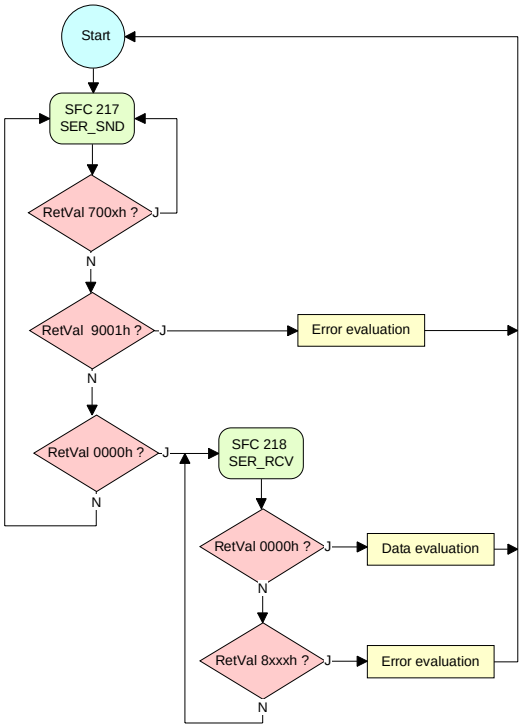

### USS / Modbus master




### ASCII / STX/ETX




### Modbus slave



## SFC 218 (SER_RCV)

This block receives data via the serial interface.

**Parameter**

| Name | Declaration | Type | Comment |
|---|---|---|---|
| DataPtr | IN | ANY | Pointer to Data Buffer for received data |
| DataLen | OUT | WORD | Length of received data |
| Error | OUT | WORD | Error Number |
| RetVal | OUT | WORD | Error Code ( 0 = OK ) |

**DataPtr**

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

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

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

**DataLen**

Word where the number of received bytes is stored.

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

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

**Error**

At ASCII, this word gets an entry in case of an error. The following error messages are possible:

| Bit | Error | Description |
|---|---|---|
| 1 | overrun | Overrun when a character could not be read from the interface fast enough. |
| 2 | parity | Parity error |
| 3 | framing error | Error that shows that a defined bit frame is not met, exceeds the allowed length or contains an additional bit sequence (stop bit error). |

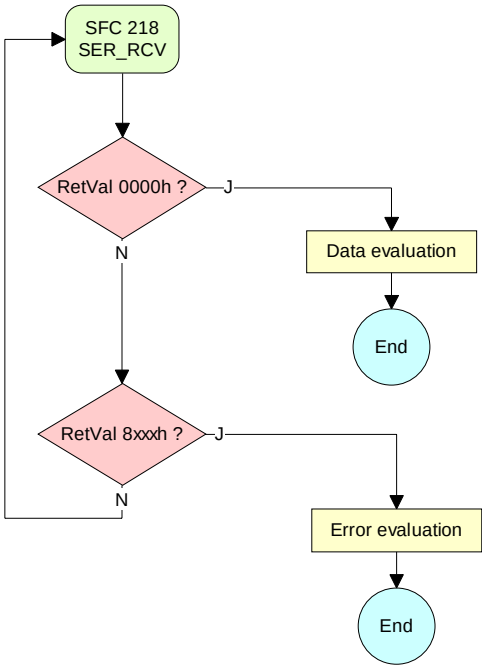
**RetVal (Error message)**

Error that is thrown in case of an error:

| Error code | Description |
|---|---|
| 0000h | no error |
| 1000h | Receive buffer too small (data loss) |
| 8x24h | Error at SFC-Parameter x, with x:<br>1: Error at "DataPtr"<br>2: Error at "DataLen"<br>3: Error at "Error" |
| 8122h | Error in parameter "DataPtr" (e.g. DB too short) |
| 809Ah | serial interface not found |
| 809Bh | serial interface not configured |

**Principles of programming**

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



# Modem functionality

**SFC 207**
**SER_CTRL**

Using the RS232 interface by means of ASCII protocol the serial modem lines can be accessed with this SFC during operation.

Depending on the parameter *FlowControl*, which is set by *SFC 216 (SER_CFG)*, this SFC has the following functionality:

| | | |
|---|---|---|
| *FlowControl=0*: | Read: | DTR, RTS, DSR, RI, CTS, CD |
| | Write: | DTR, RTS |
| *FlowControl>0:* | Read: | DTR, RTS, DSR, RI, CTS, CD |
| | Write: | not possible |

**Parameter**

| Name | Declaration | Type | Comment |
|---|---|---|---|
| Write | IN | BYTE | Bit 0: New state DTR<br>Bit 1: New state RTS |
| MaskWrite | IN | BYTE | Bit 0: Set state DTR<br>Bit 1: Set state RTS |
| Read | OUT | BYTE | Status flags (CTS, DSR, RI, CD, DTR, RTS) |
| ReadDelta | OUT | BYTE | Status flags of change between 2 accesses |
| RetVal | OUT | WORD | Return Code ( 0 = OK ) |

**Write**

With this parameter the status of DTR and RTS is set and activated by *MaskWrite*. The byte has the following allocation:

Bit 0 = DTR
Bit 1 = RTS
Bit 7 ... Bit 2: reserved

**MaskWrite**

Here with "1" the status of the appropriate parameter is activated. The byte has the following allocation:

Bit 0 = DTR
Bit 1 = RTS
Bit 7 ... Bit 2: reserved

**Read**

You get the current status by *Read*. The current status changed since the last access is returned by *ReadDelta*. The bytes have the following structure:

| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Read | x | x | RTS | DTR | CD | RI | DSR | CTS |
| ReadDelta | x | x | x | x | CD | RI | DSR | CTS |

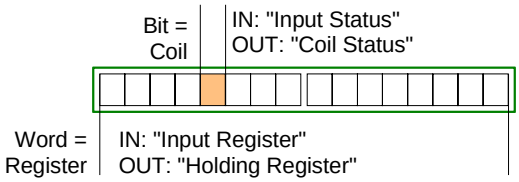
**RetVal**
**(Return value)**

| Return Code | Description |
|---|---|
| 0000h | no error |
| 8x24h | Error SFC parameter x, with x:<br>1: Error at *Write*<br>2: Error at *MaskWrite*<br>3: Error at *Read*<br>4: Error at *ReadDelta* |
| 809Ah | Interface missing |
| 809Bh | Interface not configured (SFC 216) |

# Modbus slave 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 under 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.

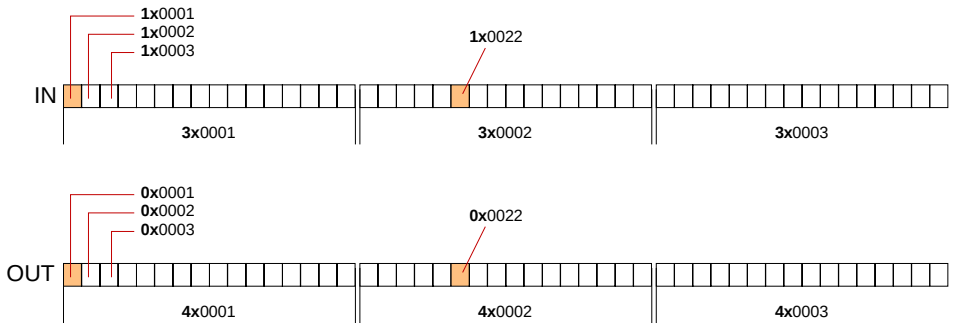
For the CPU 11xSER-1 from VIPA is not differentiating digital and analog data, the following assignment is valid:

0x: Bit area for output
Access via function code 01h, 05h

1x: Bit area for input
Access via function code 02h

3x: Word area for input
Access via function code 04h

4x: Word area for output
Access via function code 03h, 06h, 10h




A description of the function codes follows below.

**Overview**

With the following function codes a slave can be accessed by the master:

| Code | Command | Description |
|---|---|---|
| 01h | Read n bits | Read n bits of slave input area 0x |
| 02h | Read n bits | Read n bits of slave output area 1x |
| 03h | Read n Words | Read n Words of slave input area 4x |
| 04h | Read n Words | Read n Words of slave output area 3x |
| 05h | Write 1 bit | Write 1 bit to slave input area 0x |
| 06h | Write 1 Word | Write 1 Word to slave input area 4x |
| 10h | Write n Words | Write n Words to slave input area 4x |

**Note!**

The telegrams are automatically integrated into the according checksum circle of ASCII res. RTU.

Always valid for the byte sequence in a word is:

| 1 Word | |
|---|---|
| High Byte | Low Byte |

**Response of the slave**

If the slave announces an error, the function code is sent back with an "OR" and 80h. Without an error, the function code is sent back.

Slave answer: Function code OR 80h → Error
Function code → OK

**Read n bits 01h, 02h**

This function enables the reading from a slave bit by bit.

Command telegram

| RTU/ASCII frame | Slave address | Function code | Address 1. bit | Number of bits | RTU/ASCII frame |
|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Word | 1Word | 1Word |

Respond telegram

| RTU/ASCII frame | Slave address | Function code | Number of read Bytes | Data 1. Byte | Data 2. Byte | ... | RTU/ASCII frame |
|---|---|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Byte | 1Byte | 1Byte max. 252Byte | | 1Word |

**Read n Words 03h, 04h**

This function allows to read the slave word by word.

Command telegram

| RTU/ASCII-frame | Slave-address | Functions code | Address 1. Word | Number of words | RTU/ASCII frame |
|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Word | 1Word | 1Word |

Respond telegram

| RTU/ASCII frame | Slave address | Functions code | No. of read Bytes | Data 1. Word | Data 2. Word | ... | RTU/ASCII frame |
|---|---|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Byte | 1Word | 1Word | | 1Word |
| | | | | max. 125 Words | | | |

**Write 1 bit 05h**

This function allows to alter a bit in your slave. A status change happens via "Status Bit" with the following values:

"Status Bit" = 0000h → Bit = 0, " Status Bit" = FF00h → Bit = 1

Command telegram

| RTU/ASCII frame | Slave address | Function code | Address Bit | Status Bit | RTU/ASCII frame |
|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Word | 1Word | 1Word |

Respond telegram

| RTU/ASCII frame | Slave address | Function code | Address Bit | Status Bit | RTU/ASCII frame |
|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Word | 1Word | 1Word |

**Write 1 word 06h**

This function sends a word to the slave. This allows to overwrite a register in the coupler.

Command telegram

| RTU/ASCII frame | Slave address | Function code | Address Word | Value Word | RTU/ASCII frame |
|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Word | 1Word | |

Respond telegram

| RTU/ASCII frame | Slave address | Function code | Address Word | Value Word | RTU/ASCII frame |
|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Word | 1Word | |

**Write n Words 10h**

This function allows you to send n words to the slave.

Command telegram

| RTU/ASCII frame | Slave address | Functions code | Address 1. Word | Number of words | Number of Bytes | Data 1. Word | Data 2. Word | ... | RTU/ASCII frame |
|---|---|---|---|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Word | 1Word | 1Byte | 1Word | 1Word | 1Word | 1Word |
| | | | | | | max. 124Words | | | |

Respond telegram

| RTU/ASCII frame | Slave address | Functions code | Address 1. Word | Number of words | RTU/ASCII frame |
|---|---|---|---|---|---|
| | 1Byte | 1Byte | 1Word | 1Word | 1Word |

# Appendix

## A Index