Handbücher/Manuals

art of automation

VIPA
Gesellschaft für Visualisierung
und Prozessautomatisierung mbH

Ohmstraße 4
D-91074 Herzogenaurach
Tel.: +49-9132-744-0
Fax: +49-9132-744-144
Internet: www.vipa.de
E-Mail: Info@vipa.de

Manual

VIPA System 300S
SPEED7 - CPU

Order No.: VIPA HB140E_CPU
Rev. 07/43

The information contained in this manual is supplied without warranties. The information is subject to change without notice.

Ohmstraße 4, D-91074 Herzogenaurach,
Tel.: +49 (91 32) 744 -0
Fax.: +49 (91 32) 744-144
EMail: info@vipa.de
http://www.vipa.de

Hotline: +49 (91 32) 744-114

Disclaimer of liability

The contents of this manual were verified with respect to the hard- and software.

However, we assume no responsibility for any discrepancies or errors. The information in this manual is verified on a regular basis and any required corrections will be included in subsequent editions.

Suggestions for improvement are always welcome.

Trademarks

VIPA, SPEED7 and System 300V

are registered trademarks of VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

SIMATIC, STEP and S7-300

are registered trademarks of Siemens AG.

Any other trademarks referred to in the text are the trademarks of the respective owner and we acknowledge their registration.

About this manual

This manual describes the System 300S SPEED7 CPUs from VIPA with firmware version 3.0.0 and up. Here you may find -besides of a product overview- a detailed description of the single modules. You'll receive information about the connection and the deployment of the System 300S CPUs.

Outline

Chapter 1: Principles

This chapter introduces the System 300 from VIPA as central res. decentral automation system.

Chapter 2: Assembly and installation guidelines

All information required for assembly and cabling of a control consisting of System 300 components are collected in this chapter.

Chapter 3: Hardware description CPU 31xS

The SPEED7 CPU is available in different variants. This chapter describes the hardware of the different versions.

Chapter 4: Deployment CPU 31xS

General information about the deployment of the CPU like address assignment, operating modes, deployment of the MCC and communication via PG/OP and MPI is collected in this chapter.

Chapter 5: Deployment I/O periphery CPU 314ST

Here you'll see a description of the I/O periphery of the CPU 314ST. Illustrated are functionality, project engineering and diagnosis of the integrated analog and digital portion.

Chapter 6: Deployment CPU 31xS with Profibus

This chapter contains a description of the deployment and the project engineering of the SPEED7 CPUs from VIPA under Profibus.

Chapter 7: Deployment RS485 for PtP communication

Via the integrated RS485 interface you may establish a PtP communication. This chapter describes the according approach.

Chapter 8: Deployment CPU 31xS with TCP/IP

This chapter shows you the deployment of the CPU 31xSN/NET and the communication with TCP/IP.

Contents

User considerations

Objective and contents

This manual describes the System 300S SPEED7 CPUs from VIPA. It contains a description of the construction, project implementation and usage.

Target audience

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

Structure of the manual

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

Guide to the document

The following guides are available in the manual:

- 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

Applications conforming with specifications

The SPEED7 is constructed and produced for:

- all VIPA System 300 components
- communication and process control
- general control and automation applications
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle

Danger!

This device is not certified for applications in

- in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

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

- Modification to the process control system should only be carried out when the system has been disconnected from power!
- Installation and 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

This Basics contain hints for the usage and information about the project engineering of a SPEED7 system from VIPA.

General information about the System 300S like dimensions and environment conditions will also be found.

The hints for the MPI interface and the Green Cable should be regarded in this chapter!

The following text describes:

- Safety information for the usage of CPU, MP^{2}I interface and Green Cable
- Components of the System 300S
- General description like dimensions, operating security and environment conditions
- Summary of the project engineering
- Structure, working methods and basics of the programming

Content

Safety Information for Users

Handling of electrostatically 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 electrostatically sensitive equipment.

It is possible that electrostatically 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 electrostatically sensitive modules and they can damage components thereby, causing the module to become inoperable or unusable.

Modules that have been damaged by electrostatic discharges can fail after a temperature change, mechanical shock or changes in the electrical load.

Only the consequent implementation of protection devices and meticulous attention to the applicable rules and regulations for handling the respective equipment can prevent failures of electrostatically sensitive modules.

Shipping of electrostatically sensitive modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatically sensitive modules

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

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

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

Attention!

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

Hints for the deployment of the MPI interface

What is MP²I?

The MP²I jack combines 2 interfaces in 1:

- MPI interface
- RS232 interface

Please regard that the RS232 functionality is only available by using the Green Cable from VIPA.

Deployment as MPI interface

The MPI interface provides the data transfer between CPUs and PCs. In a bus communication you may transfer programs and data between the CPUs interconnected via MPI.

Connecting a common MPI cable, the MPI jack supports the full MPI functionality.

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!

Deployment as RS232 interface only via "Green Cable"

For the serial data transfer from your PC, you normally need a MPI transducer. Fortunately you may also use the "Green Cable" from VIPA. You can order this under the order no. VIPA 950-0KB00.

The "Green Cable" supports a serial point-to-point connection for data transfer via the MP²I jack exclusively for VIPA CPU's.

Please regard the hints for the deployment of the "Green Cable" on the following page.

Green Cable from VIPA

What is the Green Cable?

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

The Green Cable is a programming and download cable for VIPA CPUs with MP^{2}I jack and VIPA fieldbus masters. The Green Cable from VIPA is available under the order no. VIPA 950-0KB00.

The Green Cable allows you to:

- *transfer projects serial*
  Avoiding high hardware needs (MPI transducer, etc.) you may realize a serial point-to-point connection via the Green Cable and the MP^{2}I jack. This allows you to connect components to your VIPA-CPU that are able to communicate serial via an MPI adapter like e.g. a visualization system.
- *execute firmware updates of the CPUs and fieldbus masters*
  Via the Green Cable and an upload application you may update the firmware of all recent VIPA CPU's with MP^{2}I jack and certain fieldbus masters (see Note).

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).

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

VIPA CPU's with MP^{2}I jack and the fieldbus masters 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).

General description of the System 300

The System 300

The System 300 is a modular automation system for middle and high performance needs, that you can use either central or decentral. The single modules are directly clipped to the profile rail and are connected together with the help of bus clips at the backside.

The CPUs of the System 300 are instruction set compatible to S7-300 from Siemens.

System 300V
System 300S

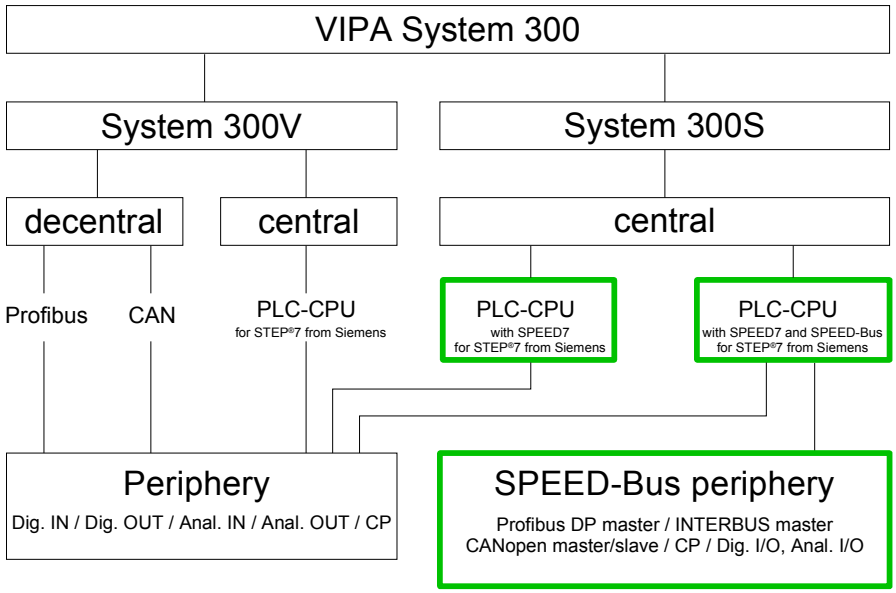
VIPA differentiates between System 300V and System 300S.

- System 300V

  The System 300V allows you to resolve automation tasks central and decentral. The single modules of the System 300V from VIPA are similar in construction to Siemens. Due to the compatible backplane bus, the modules from VIPA and Siemens can be mixed.

- System 300S

  The System 300S extends the central area with high-speed CPUs that have the integrated SPEED7 chip. Additionally some CPU's have got a parallel SPEED-Bus that allows the modular connection of fast peripheral modules like IOs or bus master.



Manual overview

This manual describes the System 300S. This includes the SPEED7-CPUs 31xS and the peripheral modules for SPEED-Bus (framed thick green).

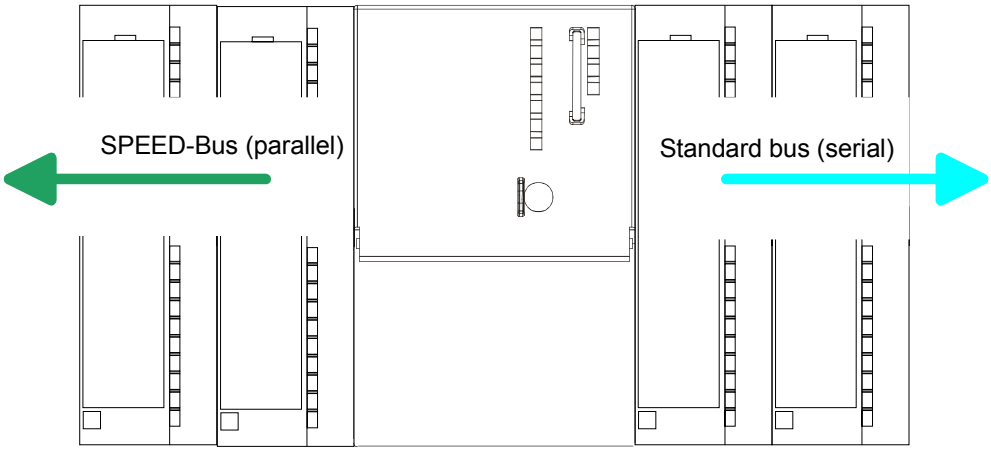
The description of the System 300V CPU 31x without SPEED7 and the concerning peripheral modules like digital and analog in-/output modules, power supplies and bus coupler is to find in the HB 130.

System 300S

Overview

The CPUs 31xS are based upon the SPEED7 technology. This supports the CPU at programming and communication by means of co-processors that causes a power improvement for highest needs.

Except of the basic variant, all SPEED7-CPUs are provided with a parallel SPEED-Bus that enables the additional connection of up to 16 modules from the SPEED-Bus periphery. While the standard peripheral modules are plugged-in at the right side of the CPU, the SPEED bus peripheral modules are connected via a SPEED-Bus bus connector at the left side of the CPU.



CPU 31xS

The System 300S series consists of a number of CPUs. These are programmed in STEP®7 from Siemens. For this you may use WinPLC7 from VIPA or the Siemens SIMATIC Manager.

CPUs with integrated Ethernet interfaces or additional serial interfaces simplify the integration of the CPU into an existing network or the connection of additional peripheral equipment.

The user application is stored in the battery buffered RAM or on an additionally pluggable MMC storage module.

Due to the automatic address allocation, the deployment of the CPUs 31xS allows to address 32 peripheral modules.

Additionally all CPU 31xS except of the basic version have got a parallel SPEED-Bus that allows the modular connection of fast peripheral modules like IOs or bus master.

SPEED-Bus

The SPEED-Bus is a 32Bit parallel bus developed from VIPA with a maximum data rate of 40MByte/s. Via the SPEED-Bus you may connect up to 16 SPEED-Bus modules to your CPU 31xS.

In opposite to the "standard" backplane bus where the modules are plugged-in at the right side of the CPU by means of single bus connectors, the modules at the SPEED-Bus are plugged-in at the left side of the CPU via a special SPEED-Bus rail.

VIPA delivers profile rails with integrated SPEED-Bus for 2, 6, 10 or 16 SPEED-Bus peripheral modules with different lengths.

SPEED-Bus peripheral modules

The SPEED-Bus peripheral modules may exclusively plugged at the SPEED-Bus slots at the left side of the CPU. The following SPEED-Bus modules are in preparation:

- Fast fieldbus modules like Profibus DP, Interbus, CANopen master and CANopen slave
- Fast CP 343 (CP 343 Communication processor for Ethernet)
- Fast digital input-/output modules (Fast Digital IN/OUT)

Memory management

Every CPU 31xS has an integrated work memory. During program run the total memory is divided into 50% for program code and 50% for data.

Starting with CPU firmware 3.0.0 there is the possibility to extend the total memory to its maximum by means of a MCC memory extension card.

Integrated Profibus DP master

The CPUs of the System 300S series have an integrated Profibus DP master. Via the DP master with a data range of 1kByte for in- and output you may address up to 125 DP slaves.

The project engineering takes place in WinPLC7 from VIPA or in the hardware configurator from Siemens.

Integrated Ethernet PG/OP channel

Every CPU 31xS has an Ethernet interface for PG/OP communication. Via the "PLC" functions you may directly access the Ethernet PG/OP channel and program res. remote control your CPU. A max. of 2 PG/OP connections is available.

You may also access the CPU with a visualization software via these connections.

Operation Security

- Wiring by means of spring pressure connections (CageClamps) at the front connector
- Core cross-section 0.08...2.5mm^2
- Total isolation of the wiring at module change
- Potential separation of all modules to the backplane bus
- ESD/Burst acc. IEC 61000-4-2/IEC 61000-4-4 (up 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
- Ventilation by means of a fan is not required

Dimensions/ Weight

- Available lengths of the profile rail in mm: 160, 482, 530, 830 and 2000
- Dimensions of the basic enclosure:
  1tier width: (HxWxD) in mm: 40x125x120
  2tier width: (HxWxD) in mm: 80x125x120

Compatibility

Modules and CPUs of the System 300 from VIPA and Siemens may be used at the "Standard" bus as a mixed configuration.

The project engineering takes place in WinPLC7 from VIPA or in the hardware configurator from Siemens.

The SPEED7 CPUs from VIPA are instruction compatible to the programming language STEP®7 from Siemens and may be programmed via WinPLC7 from VIPA or via the Siemens SIMATIC Manager.

Here the instruction set of the S7-400 from Siemens is used.

Note!

Please do always use the **CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0)** from Siemens of the hardware catalog to project a SPEED7-CPU from VIPA.

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

Green Cable

For project engineering of your DP slave you may transfer your projects from your PC to the CPU serial via MPI by using the "Green Cable". Please also regard the hints to the Green Cable in this chapter.

Integrated power supply

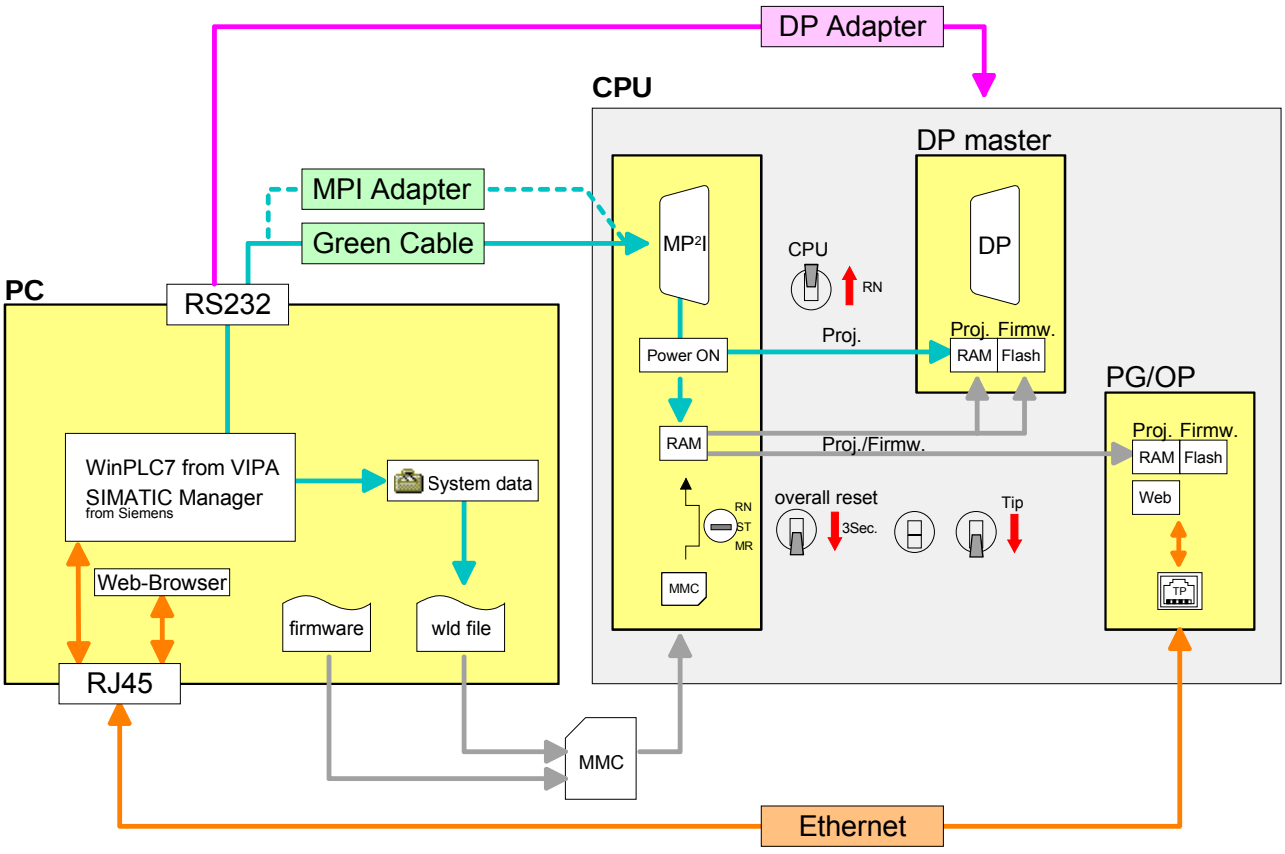
Every CPU res. bus coupler comes with an integrated power supply. The power supply has to be supplied with DC 24V. By means of the supply voltage, the bus coupler electronic is supplied as well as the connected modules via backplane bus. Please regard that the integrated power supply may supply the backplane bus the backplane bus (SPEED-Bus and Standard-Bus) depending on the CPU with a sum with max. 5A.

The power supply is protected against inverse polarity and overcurrent.

Every SPEED-Bus rail has a plug-in option for an external power supply. This allows you to raise the maximum current at the backplane bus for 6A.

Access options for project engineering and firmware update

The following overview shows all access options for project engineering and firmware update.



Hints for the Project Engineering

Overview

For the project engineering of a SPEED7 system please follow this approach:

- Project engineering of the SPEED7-CPU and the internal DP master (if existing) as CPU 318-2DP (318-2AJ00-0AB00)
- Project engineering of the real plugged modules at the standard bus
- Project engineering of the internal Ethernet PG/OP channel after the real plugged modules as virtual CP 343-1 (Setting of IP address, subnet mask and gateway for online project engineering)
- Project engineering of an internal CP343 (if existing) as 2nd CP 343-1
- Project engineering and connection of the SPEED-Bus-CPs res. -DP master as CP 343-1 (343-1EX11) res. CP 342-5 (342-5DA02 V5.0)
- Project engineering of all SPEED-Bus modules as single DP slaves in a virtual DP master module (speedbus.gsd required)

Note!

Please do always use the **CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0)** from Siemens in the hardware catalog to configure a CPU 31xS from VIPA.

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

Requirements

The hardware configurator is part of the Siemens SIMATIC Manager. it serves the project engineering. Please look at the hardware catalog for the modules that may be configured.

For the deployment of the System 300S modules at the SPEED-Bus the inclusion of the System 300S modules into the hardware catalog via the GSD-file speedbus.gsd from VIPA is necessary.

Note about the Green Cable

Please regard the hints for the deployment of the Green Cable in this chapter. For damages caused by nonobservance of these hints and/or improper deployment, VIPA does not take liability!

Approach

The project engineering of the SPEED7-CPU has the following components:

To be compatible with the Siemens SIMATIC Manager, the following steps are required:

- *Preparation*

  Start the hardware configurator from Siemens and include the speedbus.gsd for the SPEED-Bus from VIPA.

- *Project engineering of the CPU*

  Project a CPU 318-2DP (318-2AJ00-0AB00 V3.0). If your SPEED7-CPU contains a DP master, you may now connect it with Profibus and configure your DP slaves.

- *Project engineering of the real plugged modules at the standard bus*

  Set the modules that are at the right side of the CPU at the standard bus starting with slot 4.

- *Project engineering of the integrated CPs*

  For the internal Ethernet PG/OP channel you have to set a CP 343-1 (343-1EX11) as 1st module at the real plugged modules. If your SPEED7-CPU has additionally an integrated CP 343, this is also configured as CP 343-1 but always below the former placed CP 343-1.

- *Project engineering of the SPEED-Bus-CPs and -DP master*

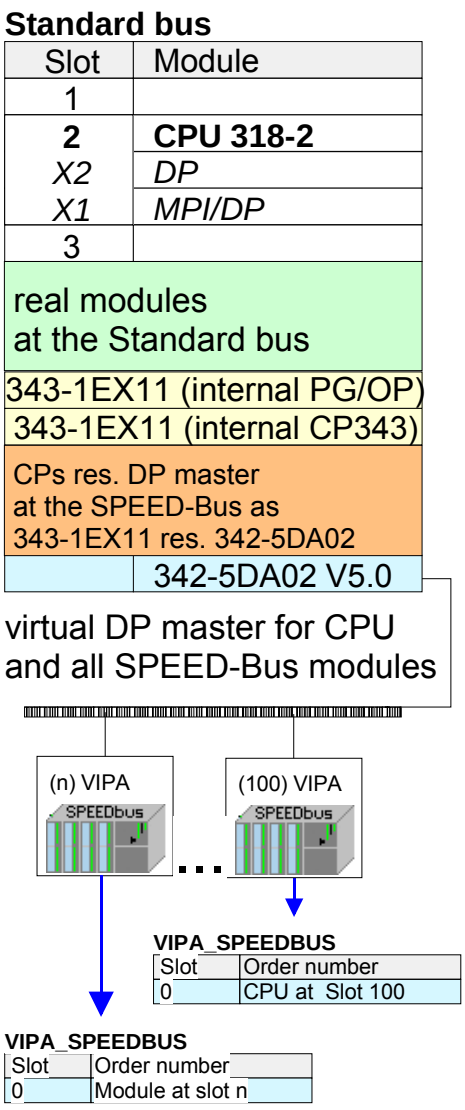
  Plug and connect all CPs as 343-1EX11 and DP master as 342-5DA02 V5.0 at the SPEED-Bus below the former configured internal CPU components.

  Please regard that the sequence within a function group (CP res. DP master) corresponds the sequence at the SPEED-Bus from right to left.

- *Project engineering of the CPU and all SPEED-Bus modules in a virtual master system*

  The slot assignment of the SPEED-Bus modules and the parameterization of the in-/output periphery happens via a virtual Profibus DP master system. For this, place a DP master (342-5DA02 V5.0) with master system as last module. The Profibus address must be <100!

  Now include the slave "vipa_speedbus" for the CPU and every module at the SPEED-Bus. After the installation of the speedbus.gsd you may find this under *Profibus-DP / Additional field devices / I/O / VIPA_SPEEDbus*. Set the slot number of the module (100...116) as Profibus address and plug the according module at slot 0 of the slave system.

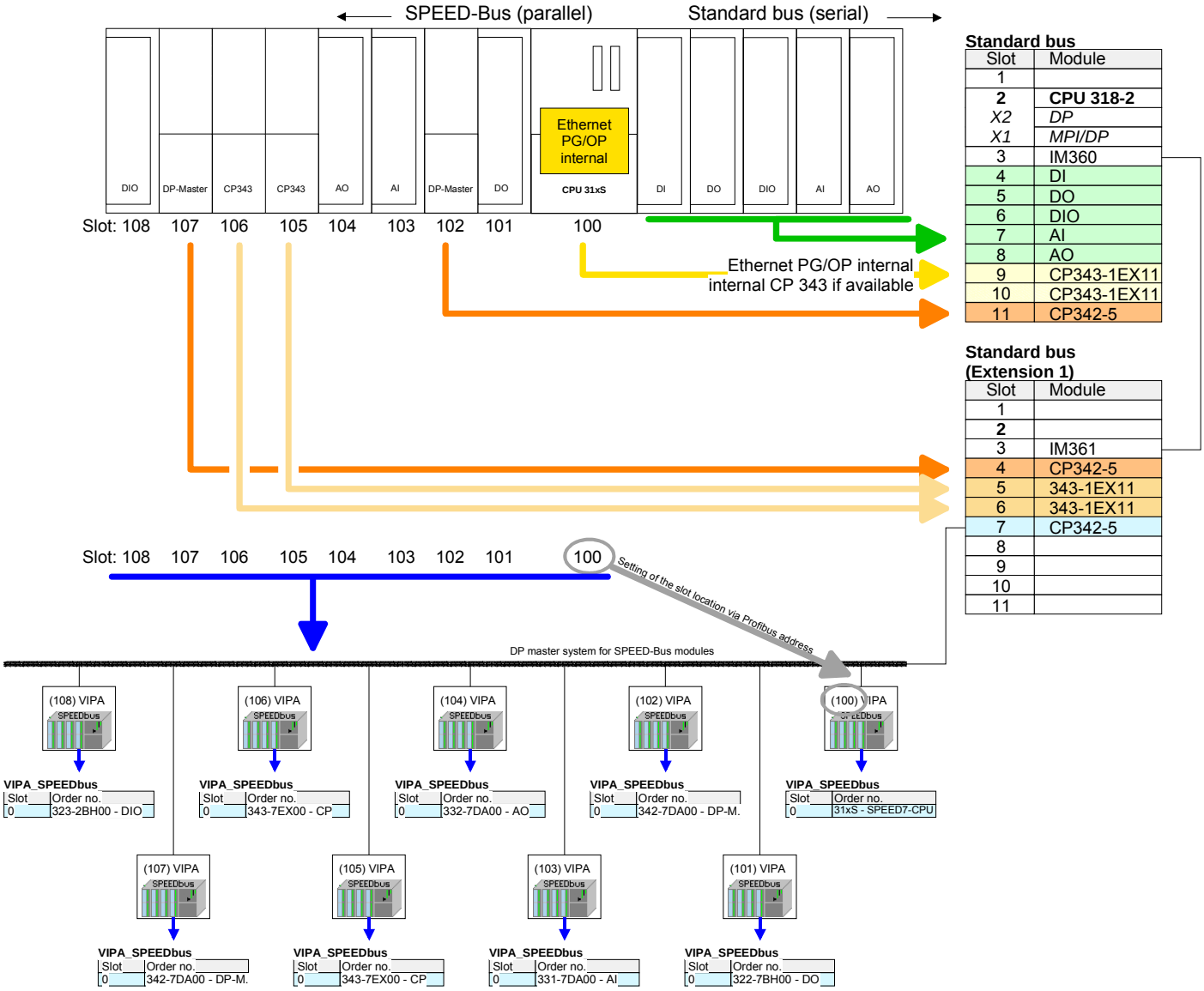


Bus extension with IM 360 and IM 361

To extend the bus you may use the IM 360 from Siemens, where 3 further extensions racks can be connected via the IM 361. Bus extensions must be placed at slot 3.

More detailed information is to be found in the chapter "Deployment CPU 31xS" at "Addressing".

Summary The following illustration summarizes all project engineering steps:



The according module is to be taken over from the HW catalog of vipa_speedbus on slot 0.

Note!

The sequence of the DPM- and CP function groups is insignificant. You only have to take care to regard the sequence within a function group (DP1, DP2... res. CP1, CP2 ...).

Hint, valid for all SPEED-Bus modules!

The SPEED-Bus always requires the Siemens DP master CP 342-5 (342-5DA02 V5.0) as last module to be included, connected and parameterized to the *operation mode* DP master. Every SPEED-Bus module has to be connected as VIPA_SPEEDbus slave into this master system.

By setting the SPEED-Bus slot number via the Profibus address and by including the according SPEED-Bus module at slot 0, the SIMATIC Manager receives information about the modules at the SPEED-Bus.

Additionally the following configurations are required depending on the module.

Project engineering of the DP master at the SPEED-Bus

The hardware configuration and Profibus project engineering happens in the SIMATIC Manager from Siemens. You have to parameterize a virtual CP 342-5 (342-5DA02 V5.0) for every SPEED-Bus-DP master at the standard bus following the real modules and connect it with the depending DP slaves.

Project engineering CP 343 at the SPEED-Bus

SPEED-Bus-CPs have to be configured in the Siemens SIMATIC Manager at the standard bus behind the real modules as virtual CP 343 (343-1EX11) and are then connected with the according Ethernet components. For the connection, the Siemens project engineering tool NetPro is required.

Project engineering of the CAN master at the SPEED-Bus

The project engineering of the CANopen master at the SPEED-Bus happens in WinCoCT (**Win**dows **C**AN**o**pen **C**onfiguration **T**ool) from VIPA.

you export your project from WinCoCT as wld-file. This wld-file can be imported into the hardware configurator from Siemens.

An additional inclusion at the standard bus is not necessary.

Project engineering of the Interbus master at the SPEED-Bus

The project engineering of the IBS master system takes place in your CPU user application using the VIPA FCs.

An additional inclusion at the standard bus is not necessary.

Operating structure of a CPU

General

The CPU contains a standard processor with internal program memory. In combination with System 300S peripherals the unit provides a powerful solution for process automation applications within the System 300S family.

A CPU supports the following modes of operation:

- cyclic operation
- timer processing
- alarm controlled operation
- priority based processing

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 can 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.

CPU 31xS Applications

Overview

The program that is present in every CPU is divided as follows:

- System routine
- User application

System routine

The system routine organizes all those functions and procedures of the CPU that are not related to a specific control application.

User application

This consists of all the functions that are required for the processing of a specific control application. The operating modules provide the interfaces to the system routines.

Operands of the CPU 31xS

Overview

The following series of operands is available for programming the CPU 31xS:

- Process image and periphery
- Bit memory
- Timers and counters
- Data blocks

Process image and periphery

The user application can quickly access the process image of the inputs and outputs PAA/PAE. You may manipulate the following types of data:

- individual Bits
- Bytes
- Words
- Double words

You may also gain direct access to peripheral modules via the bus from user application. The following types of data are available:

- Bytes
- Words
- Blocks

Bit Memory

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

You may access the following types of data:

- individual Bits
- Bytes
- Words
- Double words

Timers and counters

In your program you may load cells of the timer with a value between 10ms and 9990s. As soon as the user application 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 these when required.

Data Blocks

A data block contains constants or variables in the 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

Chapter 2 Assembly and installation guidelines

Overview

In this chapter you will find all information, required for the installation and the cabling of a process control with the components of the System 300.

The following text describes:

- General overview
- Steps of installation and cabling
- EMC-guidelines for installing a System 300

Content

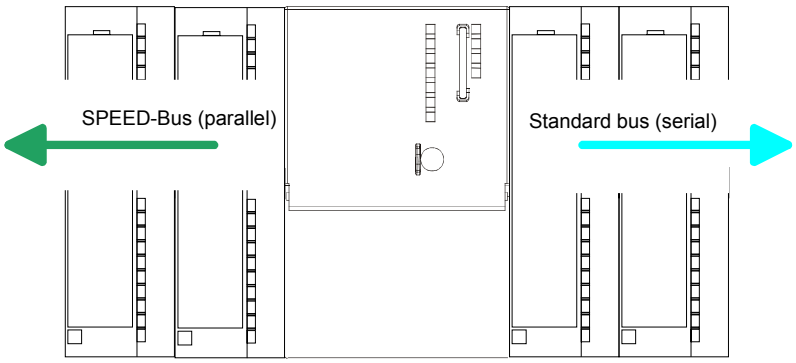
Overview

General

Except of the basic variant, all SPEED7-CPUs are provided with a parallel SPEED-Bus that enables the additional connection of up to 16 modules from the SPEED-Bus periphery.

While the standard peripheral modules are plugged-in at the right side of the CPU, the SPEED-Bus peripheral modules are connected via a SPEED-Bus bus connector at the left side of the CPU.

VIPA delivers profile rails with integrated SPEED-Bus for 2, 6, 10 or 16 SPEED-Bus peripheral modules with different lengths.



Serial Standard bus

The single modules are directly installed on a profile rail and connected via the backplane bus coupler. Before installing the modules you have to clip the backplane bus coupler to the module from the backside.

The backplane bus coupler are included in the delivery of the peripheral modules.

Parallel SPEED-Bus

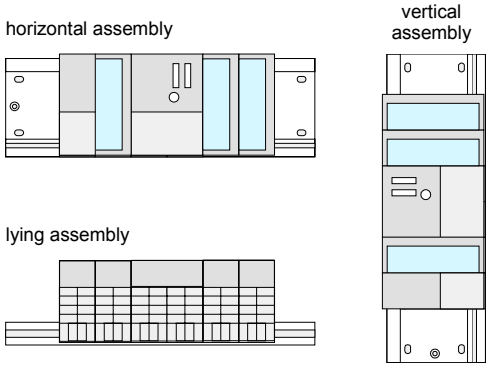
With SPEED-Bus the bus connection happens via a SPEED-Bus rail integrated in the profile rail at the left side of the CPU. Due to the parallel SPEED-Bus not all slots must be occupied in sequence.

SLOT 1 for additional power supply

At SLOT 1 DCDC) you may plug either a SPEED-Bus module or an additional power supply.

Assembly possibilities

You may assemble the System 300 horizontally, vertically or lying.



Please regard the allowed environment temperatures:

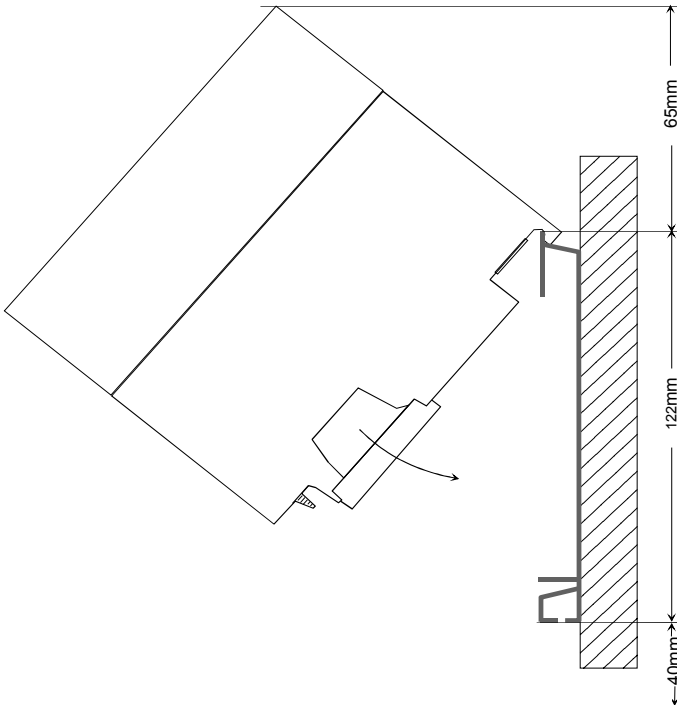
- horizontal assembly: from 0 to 60°C
- vertical assembly: from 0 to 40°C
- lying assembly: from 0 to 40°C

Installation dimensions

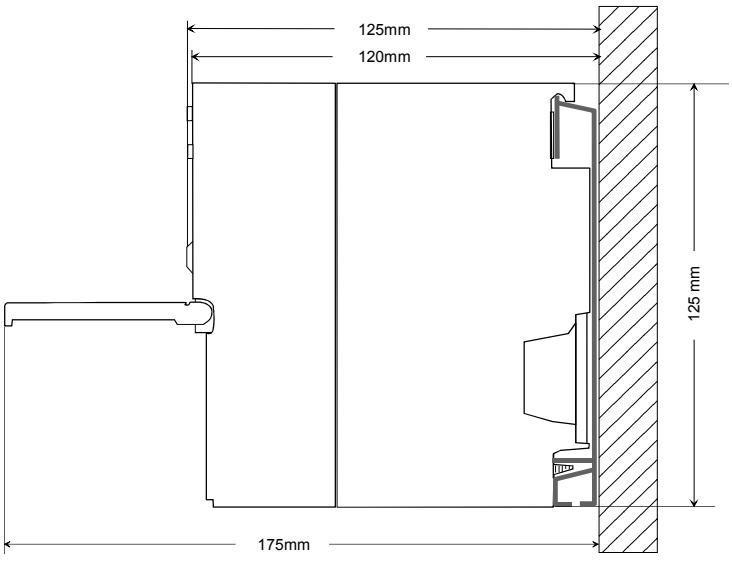
Overview Here follow all the important dimensions of the System 300.

Dimensions
Basic enclosure
1tier width (WxHxD) in mm: 40 x 125 x 120
2tier width (WxHxD) in mm: 80 x 125 x 120

Dimensions



Installation dimensions

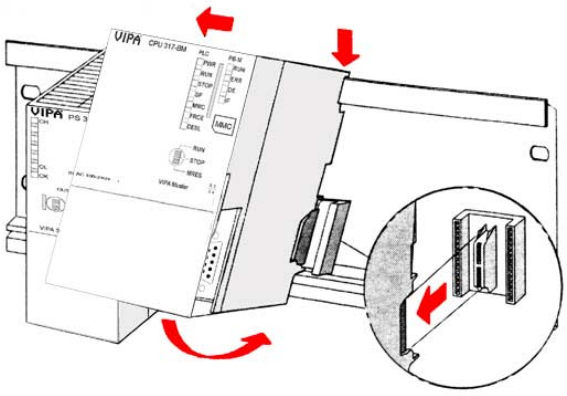


Installation Standard-Bus

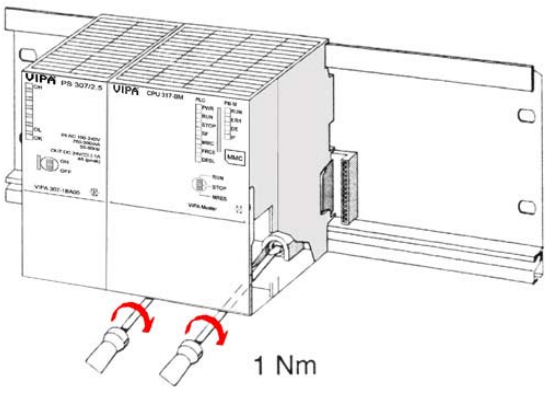
Approach

If you do not deploy SPEED-Bus modules, the assembly at the standard bus happens at the right side of the CPU with the following approach:

- Bolt the profile rail with the background (screw size: M6), so that you still have minimum 65mm space above and 40mm below the profile rail.
- If the background is a grounded metal or device plate, please look for a low-impedance connection between profile rail and background.
- Connect the profile rail with the protected earth conductor. For this purpose there is a bolt with M6-thread.
- The minimum cross-section of the cable to the protected earth conductor has to be 10mm^2.



- Stick the power supply to the profile rail and pull it to the left side up to 5mm to the grounding bolt of the profile rail.
- Take a bus coupler and click it at the CPU from behind like shown in the picture.
- Stick the CPU to the profile rail right from the power supply and pull it to the power supply.



- Click the CPU downwards and bolt it like shown.
- Repeat this procedure with the peripheral modules, by clicking a backplane bus coupler, stick the module right from the modules you've already fixed, click it downwards and connect it with the backplane bus coupler of the last module and bolt it.

Danger!

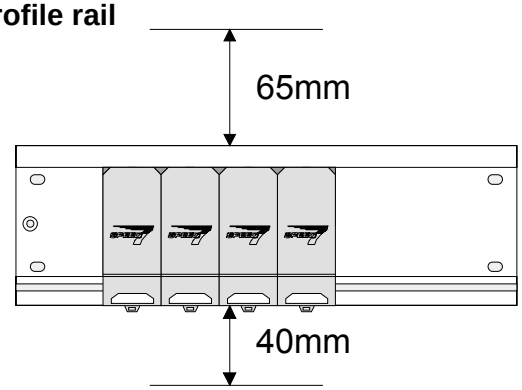
- Before installing or overhauling the System 300, the power supplies must be disconnected from voltage (pull the plug or remove the fuse)!
- Installation and modifications only by properly trained personnel!

Assembly SPEED-Bus

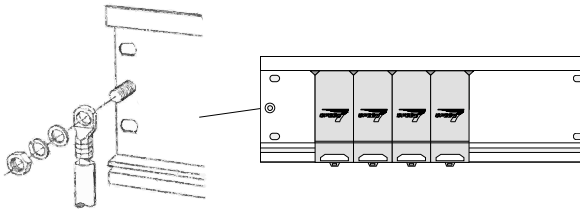
Pre-manufactured SPEED-Bus profile rail

For the deployment of SPEED-Bus modules, a pre-manufactured SPEED-Bus rail is required. This is available mounted on a profile rail with 2, 6, 10 or 16 extension plug-in locations.

Installation of the profile rail



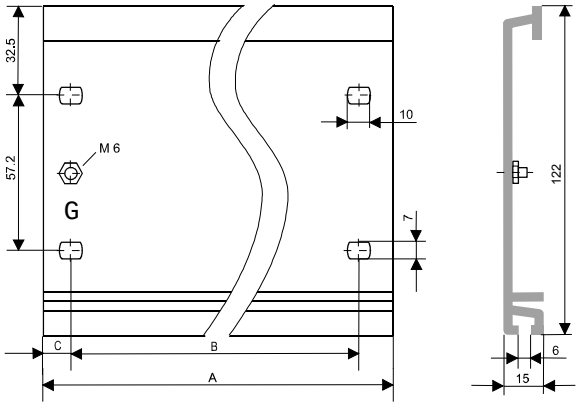
- Bolt the profile rail with the background (screw size: M6), so that you still have minimum 65mm space above and 40mm below the profile rail.
- Please look for a low-impedance connection between profile rail and background



- Connect the profile rail with the protected earth conductor.

  The minimum cross-section of the cable to the protected earth conductor has to be 10mm^2.

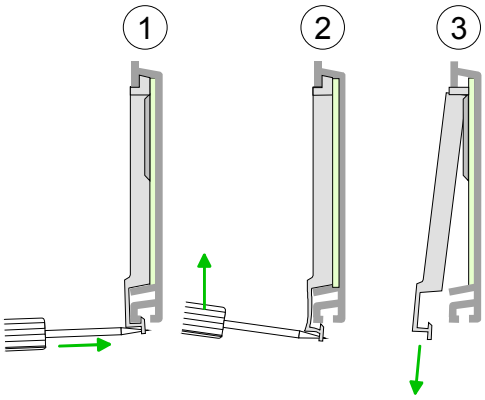
Profile rail



Order number	SPEED-Bus slots	A	B	C
VIPA 390-1AB60	-	160mm	140mm	10mm
VIPA 390-1AE80	-	482mm	466mm	8.3mm
VIPA 390-1AF30	-	530mm	500mm	15mm
VIPA 390-1AJ30	-	830mm	800mm	15mm
VIPA 390-9BC00*	-	2000mm	-	15mm
VIPA 391-1AF10	2	530mm	500mm	15mm
VIPA 391-1AF30	6	530mm	500mm	15mm
VIPA 391-1AF50	10	530mm	500mm	15mm
VIPA 391-1AF80	16	830mm	800mm	15mm

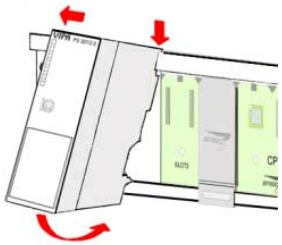
* Unit pack 10 pieces

Installation SPEED-Bus-Module

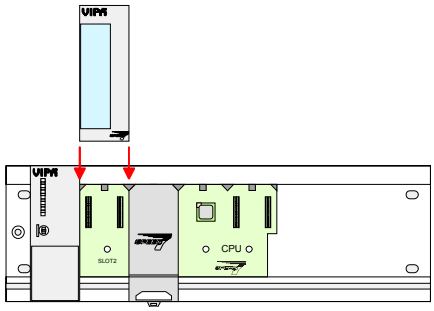


- Dismantle the according protection flaps of the SPEED-Bus plug-in locations with a screw driver (open and pull down).

  For the SPEED-Bus is a parallel bus, not all SPEED-Bus plug-in locations must be used in series. Leave the protection flap installed at an unused SPEED-Bus plug-in location.

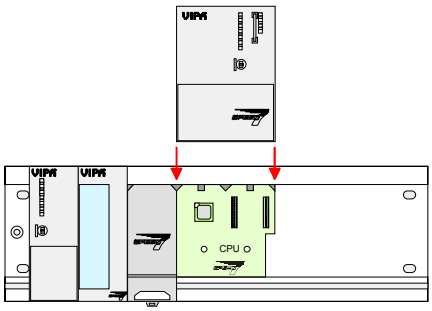


- At deployment of a DC 24V power supply, install it at the shown position at the profile rail at the left side of the SPEED-Bus and push it to the left to the isolation bolt of the profile rail.
- Fix the power supply by screwing.



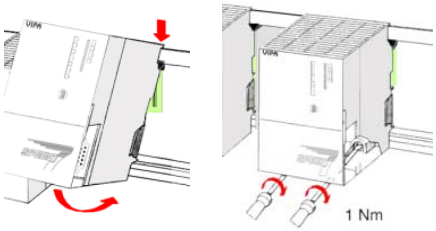
- To connect the SPEED-Bus modules, plug it between the triangular positioning helps to a plug-in location marked with "SLOT ..." and pull it down.
- Only the "SLOT1 DCDC" allows you to plug-in either a SPEED-Bus module or an additional power supply.
- Fix the modules by screwing.

Installation CPU without Standard-Bus-Modules

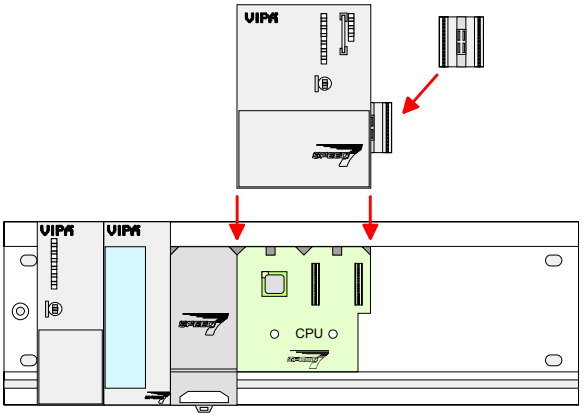


- To deploy the SPEED7-CPU exclusively at the SPEED-Bus, plug it between the triangular positioning helps to the plug-in location marked with "CPU SPEED7" and pull it down.
- Fix the CPU by screwing.

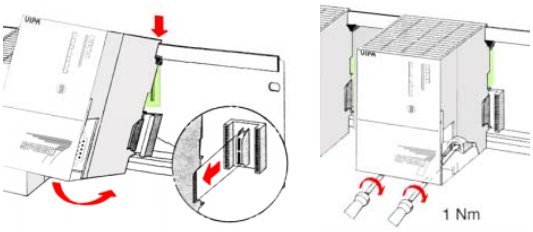
Please regard that only the CPU 317S may be deployed at the SPEED-Bus!



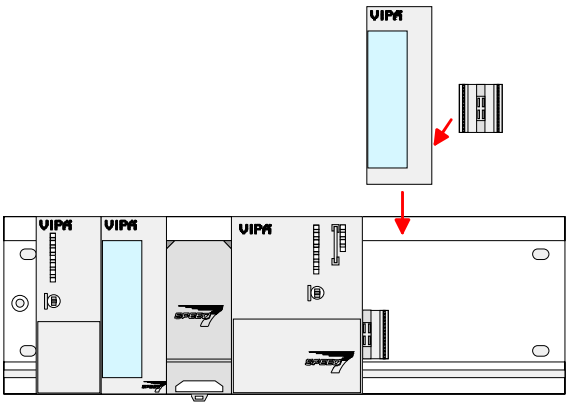
Installation CPU with Standard-Bus-Modules



- If also standard modules shall be plugged, take a bus coupler and click it at the CPU from behind like shown in the picture.
- Plug the CPU between the triangular positioning helps to the plug-in location marked with "CPU SPEED7" and pull it down.
- Fix the CPU by screwing.



Installation Standard-Bus-Modules



- Repeat this procedure with the peripheral modules, by clicking a backplane bus coupler, stick the module right from the modules you've already fixed, click it downwards and connect it with the backplane bus coupler of the last module and bolt it.

Danger!

- Before installing or overhauling the System 300V, the power supplies must be disconnected from voltage (pull the plug or remove the fuse)!
- Installation and modifications only by properly trained personnel!

Cabling

Overview

The power supplies and CPUs are exclusively delivered with CageClamp contacts. For the signal modules the front connectors are available from VIPA with screw contacts. In the following all connecting types of the power supplies, CPUs and input/output modules are described.

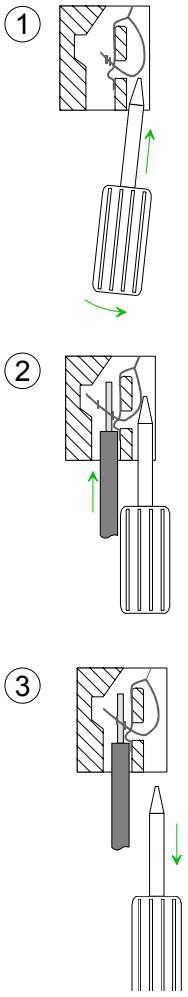
Danger!

- Before installation or overhauling, the power supplies must be disconnected from voltage (pull the plug or remove the fuse)!
- Installation and modifications only by properly trained personnel!

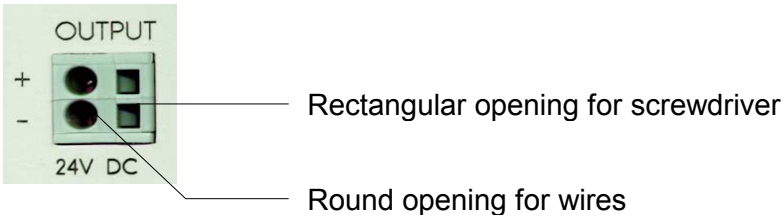
CageClamp technology (gray)

For the cabling of power supplies, bus couplers and parts of the CPU, gray connectors with CageClamp technology are used.

You may connect wires with a cross-section of 0.08mm^2 to 2.5mm^2. You can use flexible wires without end case as well as stiff wires.



You fix the conductors to the CageClamps like this:



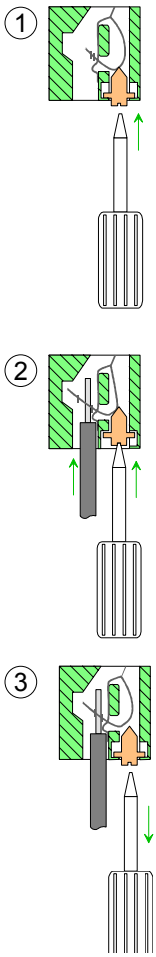
The picture on the left side shows the cabling step by step from top view.

- To conduct a wire you plug a fitting screwdriver obliquely into the rectangular opening like shown in the picture.
- To open the contact spring you have to push the screwdriver in the opposite direction and hold it.
- Insert the insulation striped wire into the round opening. You may use wires with a cross-section from 0.08mm^2 to 2.5mm^2.
- By removing the screwdriver the wire is connected safely with the plug connector via a spring.

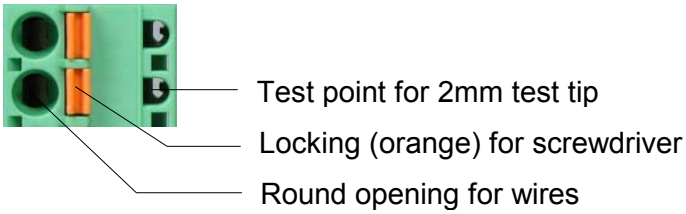
CageClamp technology (green)

For the cabling of e.g. the power supply of a CPU, green plugs with CageClamp technology are deployed.

Here also you may connect wires with a cross-section of 0.08mm^2 to 2.5mm^2. You can use flexible wires without end case as well as stiff wires.



You fix the conductors to the CageClamps like this:



The picture on the left side shows the cabling step by step from top view.

- For cabling you push the locking vertical to the inside with a suiting screwdriver and hold the screwdriver in this position.
- Insert the insulation striped wire into the round opening. You may use wires with a cross-section from 0.08mm^2 to 2.5mm^2.
- By removing the screwdriver the wire is connected safely with the plug connector via a spring.

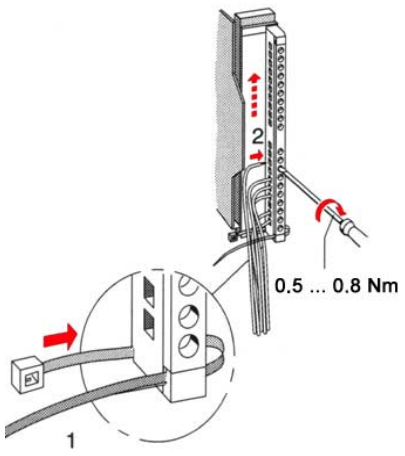
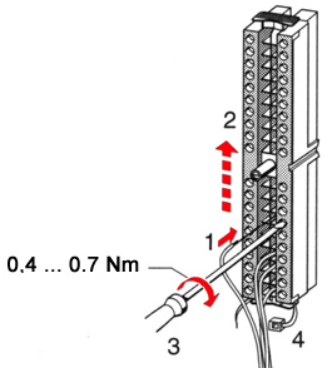
Note!

In opposite to the gray connection clamp from above, the green connection clamp is realized as plug that can be clipped off carefully even if it is still cabled.

Front connectors of the in-/output modules

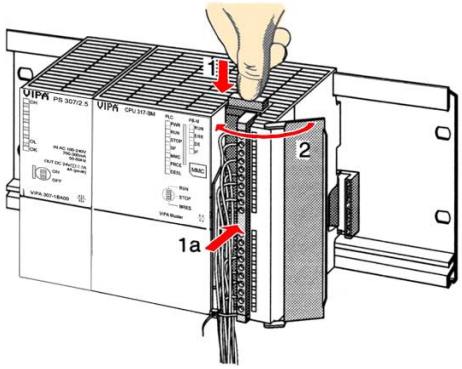
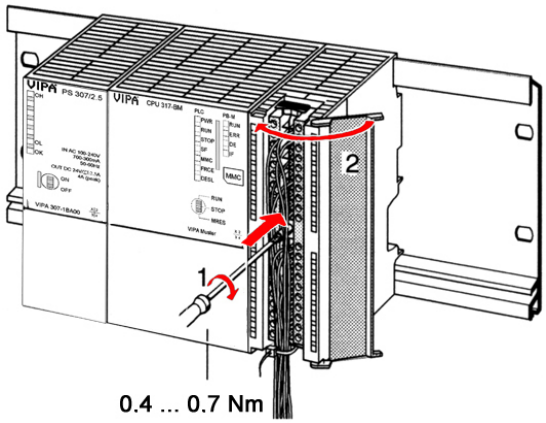
In the following the cabling of the three variants of the front-facing connector is shown:

For the I/O modules the following plugs are available at VIPA:

<table>
<tr><th>20pole screw connection
VIPA 392-1AJ00</th><th>40pole screw connection
VIPA 392-1AM00</th></tr>
<tr><td>  </td><td>  </td></tr>
<tr><td colspan="2">Open the front flap of your I/O module.</td></tr>
<tr><td colspan="2">Bring the front connector in cabling position.
Herefore you plug the front connector on the module until it locks. In this position the front connector juts out of the module and has no contact yet.</td></tr>
<tr><td colspan="2">Strip the insulation of your wires. If needed, use core end cases.</td></tr>
<tr><td>Thread the included cable binder into the front connector.</td><td></td></tr>
<tr><td colspan="2">If you want to lead out your cables from the bottom of the module, start with the cabling from bottom to top, res. from top to bottom, if the cables should be led out at the top.</td></tr>
<tr><td colspan="2">Bolt also the connection screws of not cabled screw clamps.</td></tr>
<tr><td> 
</td><td>Put the included cable binder around the cable bundle and the front connector.
 
</td></tr>
<tr><td colspan="2">Fix the cable binder for the cable bundle.</td></tr>
</table>

continued ...

... continue

<table>
<tr><th>20pole screw connection
VIPA 392-1AJ00</th><th>40pole screw connection
VIPA 392-1AM00</th></tr>
<tr><td>Push the release key at the front connector on the upper side of the module and at the same time push the front connector into the module until it locks.
 
</td><td>Bolt the fixing screw of the front connector.
 
</td></tr>
<tr><td colspan="2">Now the front connector is electrically connected with your module.</td></tr>
<tr><td colspan="2">Close the front flap.</td></tr>
<tr><td colspan="2">Fill out the labeling strip to mark the single channels and push the strip into the front flap.</td></tr>
</table>

Installation Guidelines

General

The installation guidelines contain information about the interference free deployment of System 300 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 300 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 metallized 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 300V modules.
  - For lightening cabinets you should prefer incandescent lamps and avoid luminescent lamps.
- Create an 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 300V 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, magnetical and electromagnetical 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 metallized plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.
- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to the System 300V 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 3 Hardware description CPU 31xS

Outline

The CPUs 31xS are available in different versions that are described in the following chapter.
The chapter closes with the technical data.

The following text describes:

- operating and display elements of the SPEED7 CPUs
- In-/Output-Range of the CPU 314ST
- Technical data

Content

System Overview

SPEED7-CPUs

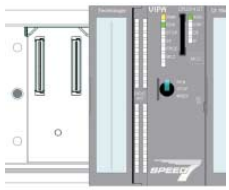
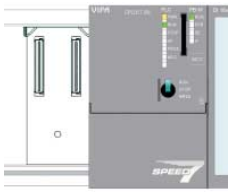
These CPUs are instruction set compatible to STEP®7 from Siemens and are designed for medium and large applications with integrated 24V power supply unit. Every CPU has a slot for memory cards at the front side, an integrated Ethernet interface for PG/OP, a RS485 interface for Profibus DP master communication and a MPI interface and is developed for future memory extensions by means of MCC.

You may poll sensors and control actuators via standardized commands and programs.

Depending on the CPU type you have additionally an integrated CP 343 or a RS485 interface for communication tasks. This CPU series gains you access to the peripheral modules of the System 300V for the standard bus.

Additionally all CPUs 31xS except of the basic version provide a parallel SPEED-Bus that allows you to connect fast peripheral modules like IOs or bus master modular.

The following descriptions in this manual refers to the complete SPEED7-CPU family CPU 31xS from VIPA with firmware version 3.0.0 and up if nothing else is mentioned.

	Basis		Technologie	Extension	
	CPU 315SB DPM	CPU 315SN NET	CPU 314ST DPM	CPU 317SE DPM	CPU 317SN NET
					
Order no.	315-2AG10 315-2AG12	315-4NE11 315-4NE12	314-6CF01 314-6CF02	317-2AJ11 317-2AJ12	317-4NE11 317-4NE12
Memory (50% code / 50% data) via MCC expandable up to	1MB 2MB		512kB 2MB	2MB 8MB	
MP^{2}I	yes *)				
Real time clock	yes				
Ethernet PG/OP	yes				
SPEED-Bus	-		yes	yes	
16 DIO/AIO: DI 8...16xDC24V DO 8...0 DC24V 0.5A 4 Counter AI 4x12Bit/AO 2x12Bit AI 1xPt100	-		yes	-	
Profibus Master/PtP	yes				
CP 343 integrated	-	yes	-	-	yes
Width	1tier	2tier	2tier	2tier	

*) The CPUs, whose order no. ends with 2, only have a MPI interface. Here the deployment of the VIPA Green Cable is not possible.

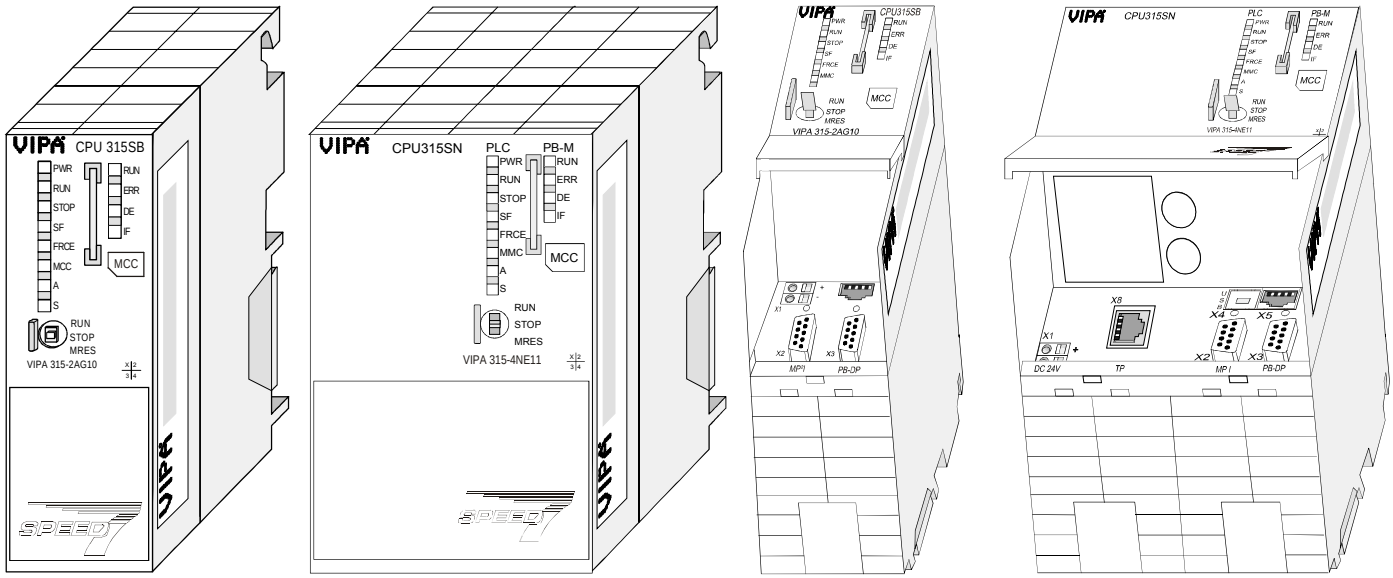
Basis Version

CPU 315SB/DPM
315-2AG10

CPU 315SN/NET
315-4NE11

- SPEED7 technology integrated
- 1MByte total memory (512kByte code, 512kByte data)
- Memory expandable to max. 2MByte (1MByte code, 1MByte data)
- Profibus DP master integrated (DP-V0, DP-V1)
- MP^{2}I-interface
- MCC slot for external memory cards and memory extension
- Status LEDs for operating state and diagnosis
- Real-time clock battery buffered
- Ethernet PG/OP interface integrated
- RS485 interface configurable for Profibus DP master respectively PtP communication
- CP 343 integrated for max. 8 configurable connections (only VIPA 315-4NE11)
- I/O address range digital/analog 8191byte
- 512 timer
- 512 counter
- 8192 flag byte

DPM - front NET - front DPM - bottom NET - bottom



Ordering data

Type	Order number	Description
315SB/DPM	VIPA 315-2AG10	MP^{2}I interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master
315SN/NET	VIPA 315-4NE11	MP^{2}I interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master, CP 343

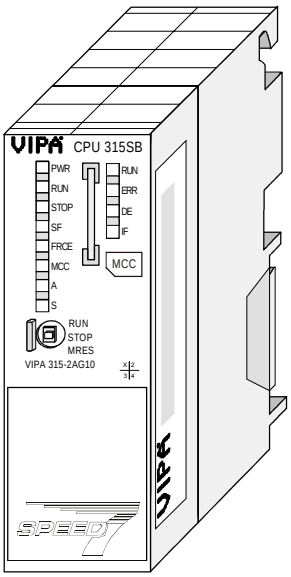
Continued Basis Version

CPU 315SB/DPM
315-2AG12

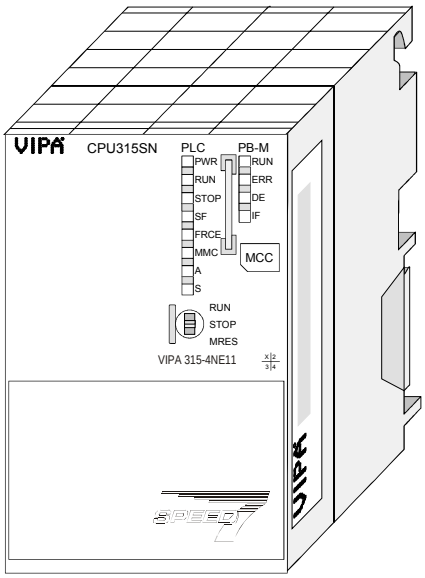
CPU 315SN/NET
315-4NE12

- SPEED7 technology integrated
- 1MByte total memory (512kByte code, 512kByte data)
- Memory expandable to max. 2MByte (1MByte code, 1MByte data)
- Profibus DP master integrated (DP-V0, DP-V1)
- MPI interface (connecting the Green Cable not possible)
- MCC slot for external memory cards and memory extension
- Status LEDs for operating state and diagnosis
- Real-time clock battery buffered
- Ethernet PG/OP interface integrated
- RS485 interface configurable for Profibus DP master respectively PtP communication
- CP 343 integrated for max. 8 configurable connections (only VIPA 315-4NE11)
- I/O address range digital/analog 8191byte
- 512 timer
- 512 counter
- 8192 flag byte

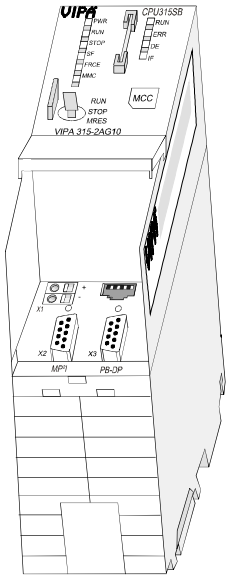
DPM - front



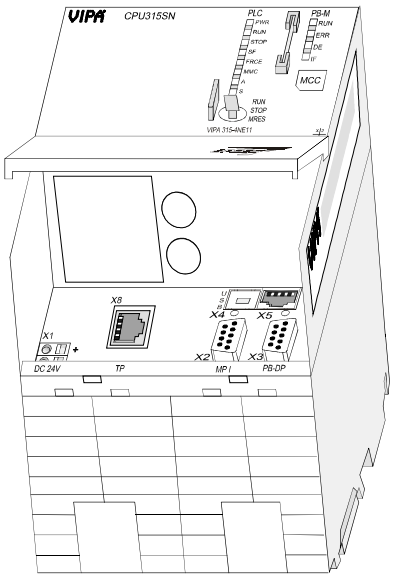
NET - front



DPM - bottom



NET - bottom



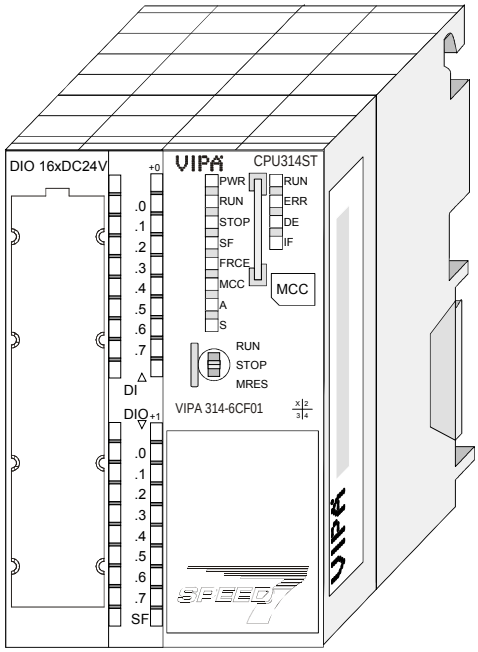
Ordering data

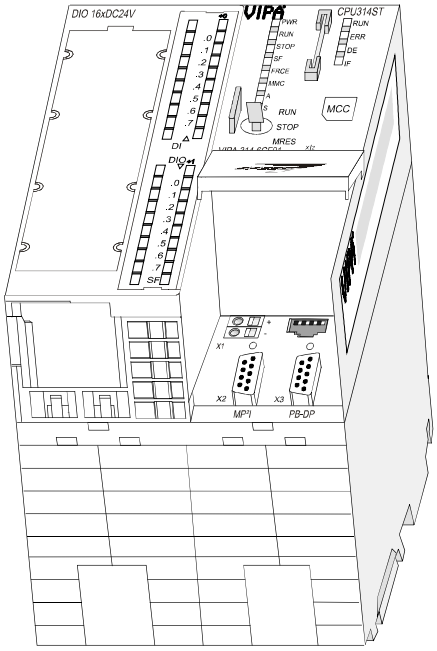
Type	Order number	Description
315SB/DPM	VIPA 315-2AG12	MPI interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master
315SN/NET	VIPA 315-4NE12	MPI interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master, CP 343

Technology Version

CPU 314ST/DPM
314-6CF01

- SPEED7 technology and SPEED-Bus integrated
- 512kByte total memory (256kByte code, 256kByte data)
- Memory expandable to max. 2MByte (1MByte code, 1MByte data)
- Profibus DP master integrated supports DP-V0 and DP-V1
- MP^2I interface
- MCC slot for external memory cards and memory extension
- Status-LEDs for operating state and diagnosis
- Real-time clock battery buffered
- Ethernet PG/OP interface integrated
- RS485 interface configurable for Profibus DP master respectively PtP communication
- Fast digital I/Os: DI 8...16xDC24V / DO 8...0xDC 24V, 0.5A
- Analog I/Os: AI 4x12Bit / AO 2x12Bit / AI 1xPt100
- 4 counter (100kHz)
- I/O address range digital/analog 8191byte
- 512 timer
- 512 counter
- 8192 flag byte





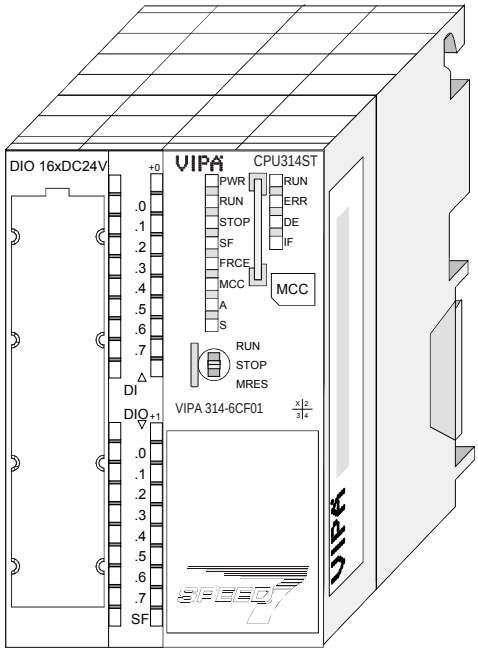
Ordering data

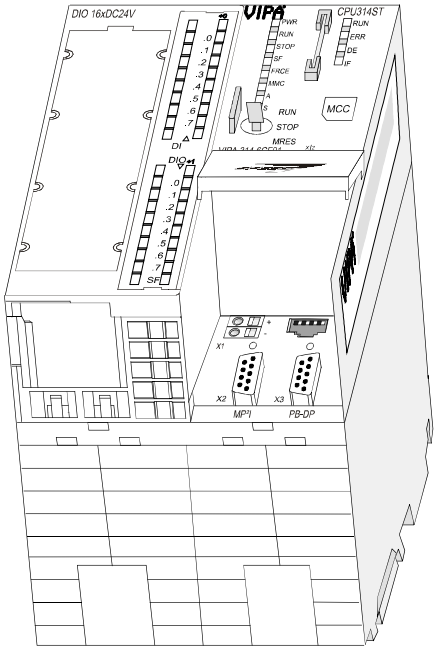
Type	Order number	Description
314ST/DPM	VIPA 314-6CF01	MP^2I interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master, SPEED-Bus, DI 8...16xDC24V / DO 8...0xDC24V, 0.5A, AI 4x12Bit / AO 2x12Bit / AI 1xPt100, 4 Counter

Continued Technology Version

CPU 314ST/DPM
314-6CF02

- SPEED7 technology and SPEED-Bus integrated
- 512kByte total memory (256kByte code, 256kByte data)
- Memory expandable to max. 2MByte (1MByte code, 1MByte data)
- Profibus DP master integrated supports DP-V0 and DP-V1
- MPI interface (connecting the Green Cable not possible)
- MCC slot for external memory cards and memory extension
- Status-LEDs for operating state and diagnosis
- Real-time clock battery buffered
- Ethernet PG/OP interface integrated
- RS485 interface configurable for Profibus DP master respectively PtP communication
- Fast digital I/Os: DI 8...16xDC24V / DO 8...0xDC 24V, 0.5A
- Analog I/Os: AI 4x12Bit / AO 2x12Bit / AI 1xPt100
- 4 counter (100kHz)
- I/O address range digital/analog 8191byte
- 512 timer
- 512 counter
- 8192 flag byte





Ordering data

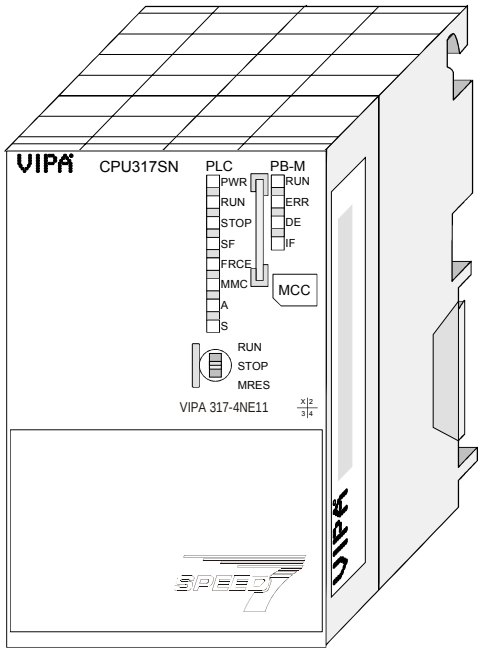
Type	Order number	Description
314ST/DPM	VIPA 314-6CF02	MPI interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master, SPEED-Bus, DI 8...16xDC24V / DO 8...0xDC24V, 0.5A, AI 4x12Bit / AO 2x12Bit / AI 1xPt100, 4 Counter

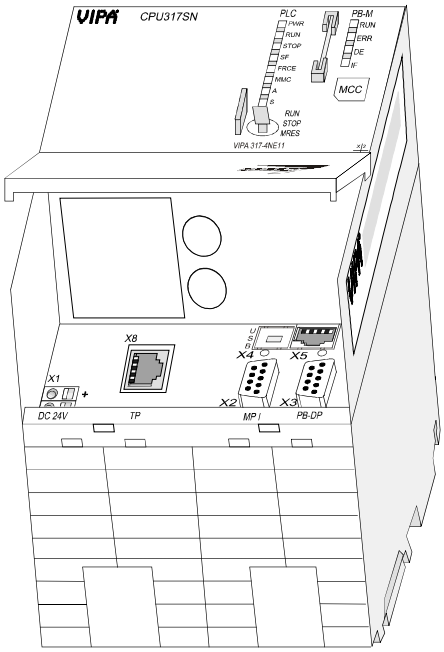
Extension Version

CPU 317SE/DPM
317-2AJ11

CPU 317SN/NET
317-4NE11

- SPEED7 technology and SPEED-Bus integrated
- 2MByte total memory (1MByte code, 1MByte data)
- Memory expandable to max. 8MByte (4MByte code, 4MByte data)
- Profibus DP master integrated (DP-V0, DP-V1)
- MP^{2}I interface
- MCC slot for external memory cards and memory extension
- Status-LEDs for operating state and diagnosis
- Real-time clock battery buffered
- Ethernet PG/OP interface integrated
- RS485 interface configurable for Profibus DP Master respectively PtP communication
- CP 343 communication processor integrated (only VIPA 317-4NE11)
- I/O address range digital/analog 8191byte
- 512 timer
- 512 counter
- 8192 flag byte





Ordering data

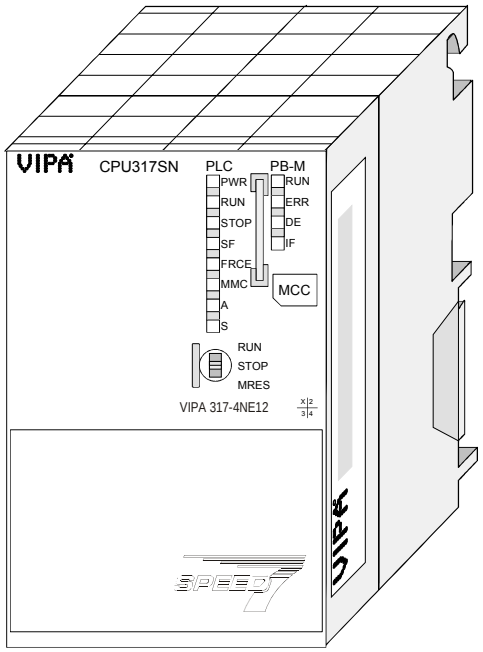
Type	Order number	Description
317SE/DPM	VIPA 317-2AJ11	MP^{2}I interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master, SPEED-Bus
317SN/NET	VIPA 317-4NE11	MP^{2}I interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master, SPEED-Bus, CP 343

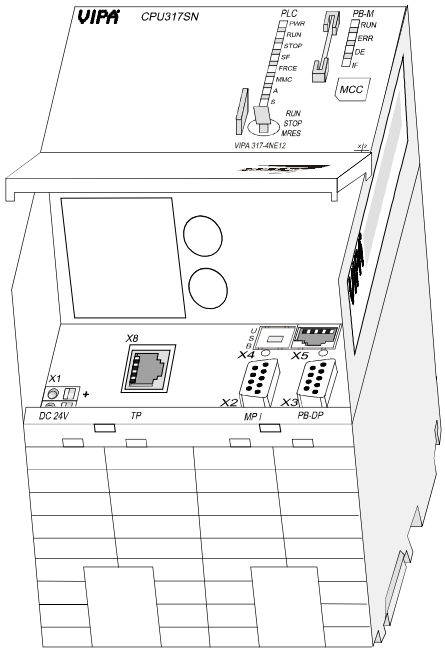
Continued Extension Version

CPU 317SE/DPM
317-2AJ12

CPU 317SN/NET
317-4NE12

- SPEED7 technology and SPEED-Bus integrated
- 2MByte total memory (1MByte code, 1MByte data)
- Memory expandable to max. 8MByte (4MByte code, 4MByte data)
- Profibus DP master integrated (DP-V0, DP-V1)
- MPI interface (connecting the Green Cable not possible)
- MCC slot for external memory cards and memory extension
- Status-LEDs for operating state and diagnosis
- Real-time clock battery buffered
- Ethernet PG/OP interface integrated
- RS485 interface configurable for Profibus DP Master respectively PtP communication
- CP 343 communication processor integrated (only VIPA 317-4NE12)
- I/O address range digital/analog 8191byte
- 2048 timer
- 2048 counter
- 16384 flag byte



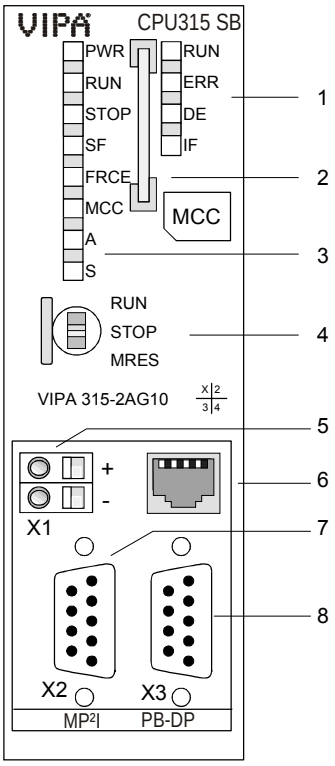


Ordering data

Type	Order number	Description
317SE/DPM	VIPA 317-2AJ12	MPI interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master, SPEED-Bus
317SN/NET	VIPA 317-4NE12	MPI interface, card slot, real time clock, Ethernet interface for PG/OP, Profibus DP master, SPEED-Bus, CP 343

Structure

CPU 315SB/DPM
315-2AG1x

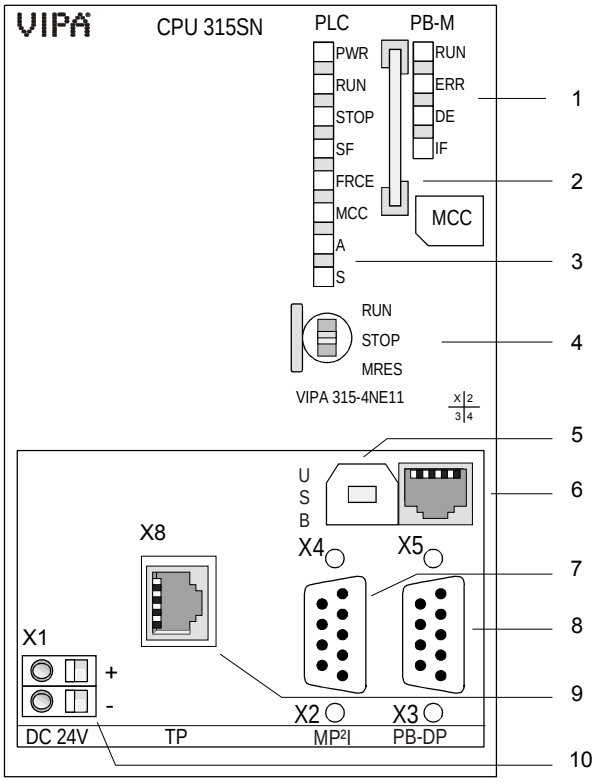


[1] LEDs of the integrated Profibus DP master
[2] Storage media slot
[3] LEDs of the CPU part
[4] Operating mode switch CPU

The following components are under the front flap

[5] Slot for DC 24V power supply
[6] Twisted pair interface for PG/OP channel
[7] 315-2AG10 MP²I interface
315-2AG12 MPI interface
[8] Profibus DP/PtP interface

CPU 315SN/NET
315-4NE1x

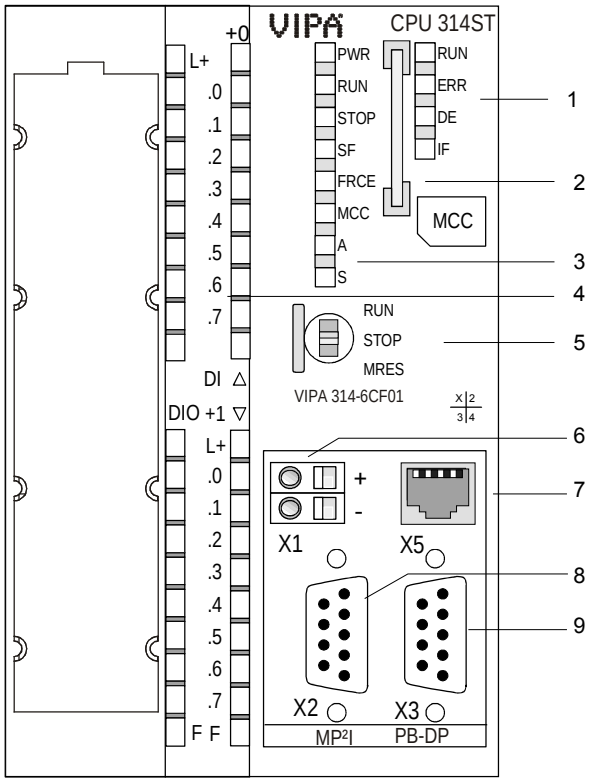


[1] LEDs of the integrated Profibus DP master
[2] Storage media slot
[3] LEDs of the CPU part
[4] Operating mode switch CPU

The following components are under the front flap

[5] USB interface
[6] Twisted pair interface for PG/OP channel
[7] 315-4NE11 MP²I interface
315-4NE12 MPI interface
[8] Profibus DP/PtP interface
[9] Twisted pair interface for CP 343
[10] Slot for DC 24V power supply

CPU 314ST/DPM
314-6CF0x

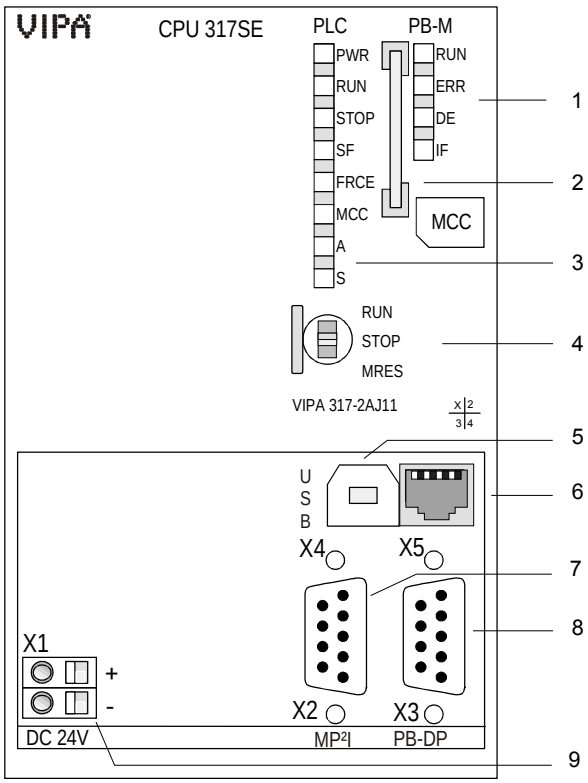


[1] LEDs of the integrated Profibus DP master
[2] Storage media slot
[3] LEDs of the CPU part
[4] LEDs of the I/O part
[5] Operating mode switch CPU

The following components are under the front flap

[6] Slot for DC 24V power supply
[7] Twisted pair interface for PG/OP channel
[8] 314-6CF01 MP^{2}I interface
314-6CF02 MPI interface
[9] Profibus DP/PtP interface

CPU 317SE/DPM
317-2AJ1x

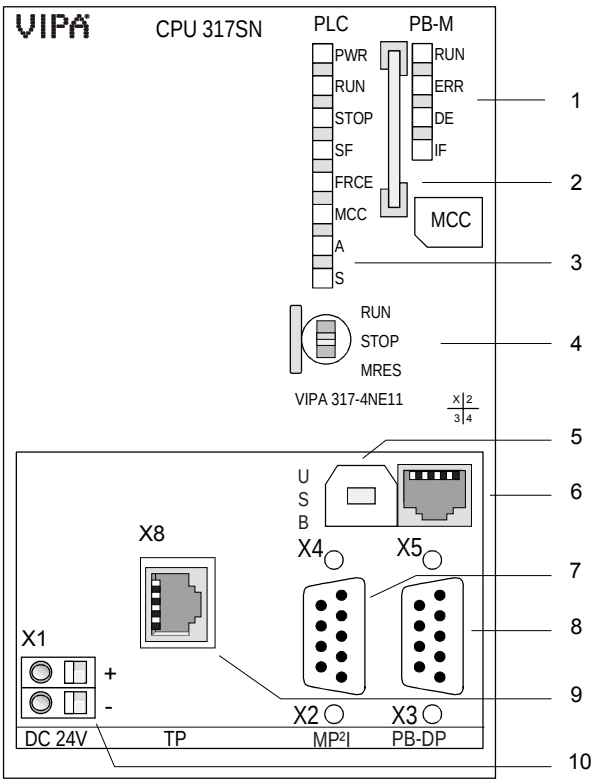


[1] LEDs of the integrated Profibus DP master
[2] Storage media slot
[3] LEDs of the CPU part
[4] Operating mode switch CPU

The following components are under the front flap

[5] USB interface
[6] Twisted pair interface for PG/OP channel
[7] 317-2AJ11 MP^{2}I interface
317-2AJ12 MPI interface
[8] Profibus DP/PtP interface
[9] Slot for DC 24V power supply

CPU 317SN/NET
317-4NE1x



[1] LEDs of the integrated Profibus DP master
[2] Storage media slot
[3] LEDs of the CPU part
[4] Operating mode switch CPU

The following components are under the front flap

[5] USB interface
[6] Ethernet PG/OP interface
[7] 317-2AJ11 MP^{2}I interface
317-2AJ12 MPI interface
[8] Profibus DP/PtP interface
[9] Twisted pair interface for CP 343
[10] Slot for DC 24V power supply

Components

CPU 31xS

The here mentioned components are part of every CPU 31xS.

LEDs CPU part

The CPU has got one row of LEDs on the front side. The following table shows you the usage of the LEDs and the according colors:

Label	Color	Meaning
PWR	green	CPU part is provided with internal 5V
RUN	green	CPU is in the operating mode RUN
STOP	yellow	CPU is in the operating mode STOP
SF	red	On at system errors (hardware defect)
FRCE	yellow	On as soon as variables are forced (fixed)
MCC	yellow	Blinks at storage media access
A	green	Activity: on: physically connected off: no physical connection blinks: shows Ethernet activity
S	green	Speed: on: 100MBit off: 10MBit

Note!

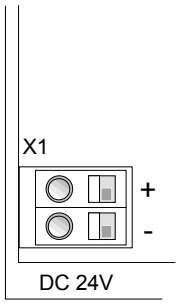
All LEDs of the CPU part are blinking three times, when accessing an invalid storage media or when it is pulled out during the reading process.

Storage media slot

As external storage medium for applications and firmware you may use a MMC storage module (**M**ulti**m**edia **c**ard) or a MCC memory extension card. The MCC can additionally be used as an external storage medium.

Both VIPA storage media are pre-formatted with the PC format FAT16 and can be accessed via a card reader. An access to the storage media always happens after an overall reset and PowerON.

Power supply



The CPU has an integrated power supply. The power supply has to be provided with DC 24V. For this serves the double DC 24V slot, that is underneath the flap.

Via the power supply not only the internal electronic is provided with voltage, but by means of the backplane bus also the connected modules. The power supply is protected against polarity inversion and overcurrent. The internal electronic is galvanically connected with the supply voltage.

Please regard that the integrated power supply may provide the backplane bus (SPEED and standard bus) with a sum of max. 5A depending on the CPU. Every SPEED-Bus bar has as option a slot for an external power supply. This allows you to raise the max. current at the backplane bus for 6A.

Operating mode switch

With the operating mode switch you may switch the CPU between STOP and RUN. The operating mode START-UP is driven automatically from the CPU between STOP and RUN.

Placing the switch to Memory Reset (MRES), you request an overall reset with following load from MMC (project or firmware update).

Memory management

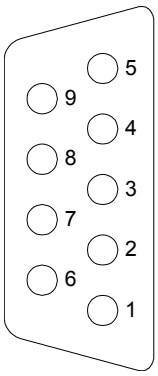
Every CPU 31xS has an integrated work memory. During program run the total memory is divided into 50% for program code and 50% for data.

Starting with CPU firmware 3.0.0 there is the possibility to extend the total memory to its maximum by means of a MCC memory extension card.

MPI interface
MP²I interface

The MP²I interface handles the data exchange between CPU and PC. Via a bus communication you may transfer applications and data with up to 12MBaud between the CPUs that are connected via MPI.

For a serial transfer from your PC you normally need a MPI transducer. The "Green Cable" may be used if your CPU has a MP²I interface. The "Green Cable" is exclusively available from VIPA with order number 950-0KB00. The "Green Cable" may only be used directly and exclusively at CPUs with MP²I interface. Please also regard the hints in the chapter "Basics"! With an MP²I interface the data transmission rate is limited to 1.5MBaud. The MPI-slot has the following pin assignment:



9pin jack

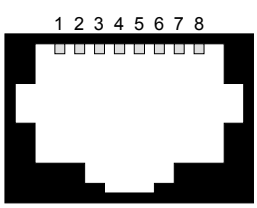
Pin	Assignment
1	reserved (must not be connected) See Hints for the deployment of the MPI interface in chapter "Basics".
2	M24V
3	RxD/TxD-P (Line B)
4	RTS
5	M5V
6	P5V
7	P24V
8	RxD/TxD-N (Line A)
9	n.c.

Ethernet PG/OP channel

The RJ45 jack serves the interface to the Ethernet PG/OP channel. This interface allows you to program res. remote control your CPU, to access the internal website or to connect a visualization via up to 2 PG/OP connections. Here a transfer rate of 10MBit at half duplex is supported.

For online access to the CPU via Ethernet PG/OP channel valid IP address parameters have to be assigned to this. More may be found at chapter "Deployment CPU 31xS" at "Initialization Ethernet PG/OP channel".

The jack has the following assignment:



8pin RJ45-slot:

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

Communication components

Additionally to the Ethernet PG/OP channel the following CPUs provide further communication components:

- CPU 315SB/DPM Profibus DP master / PtP via RS485
- CPU 315SN/NET Profibus DP master / PtP via RS485 and CP 343
- CPU 314ST/DPM Profibus DP master / PtP via RS485
- CPU 317SE/DPM Profibus DP master / PtP via RS485
- CPU 317SN/NET Profibus DP master / PtP via RS485 and CP 343

RS485 interface with configurable functionality

Every CPU 31xS has a integrated RS485 interface. The functionality of this interface can be configured by the mean of the parameter "Function RS485" of the SPEED-Bus CPUs hardware configuration.

Profibus functionality

Using the *Profibus* functionality the integrated Profibus DP master is connected to Profibus via RS485 interface. At master operation there is access to up to 125 DP slaves. For this the project engineering happens in the hardware configurator from Siemens.

For state display the CPU has a row of LEDs at its front side. Dependent on the mode of operation these give information according to the following pattern over the operating condition of the Profibus part:

Master operation

RUN green	ERR red	DE green	IF red	Meaning
○	○	○	○	Master has no project, this means the interface is deactivated respectively PtP is active.
●	○	○	○	Master has bus parameters and is in RUN without slaves.
●	○	☼	○	Master is in "clear" state (safety state). The inputs of the slaves may be read. The outputs are disabled.
●	○	●	○	Master is in "operate" state, this means data exchange between master and slaves. The outputs may be accessed.
●	●	☼	○	At least 1 slave is missing.
○	○	○	●	Initialization error at faulty parameterization.
○	●	○	●	Waiting state for start command from CPU.

Slave operation

RUN green	ERR red	DE green	IF red	Meaning
○	○	○	○	Slave has no project respectively PtP is active.
☼	○	○	○	Slave is without master.
☼	○	☼	○	Alternate flashing at configuration faults.
●	○	●	○	Slave exchanges data between Master.

on: ● off: ○ flashing: ☼

PtP functionality

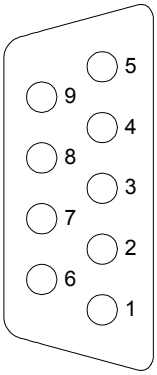
Using the *PtP* functionality the RS485 interface is allowed to connect via serial point-to-point connection to different source res. target systems. Here the following protocols are supported:

ASCII, STX/ETX, 3964R, USS and Modbus-Master (ASCII, RTU)

RS485 interface

RS485 interface of both functionalities have the same pin assignment:

9-pin Profibus SubD jack:



Pin	Assignment
1	shield
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.

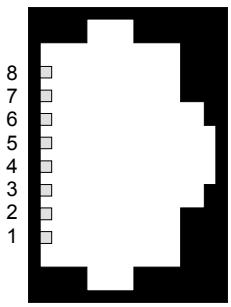
Communication processor CP 343

The CP 343 offers you a communication processor. This serves 8 PG/OP channels and 16 configurable connections (max. 8 at CPU 315-4NE11). The project engineering happens using NetPro from Siemens as CP343-1EX11.

Via the RJ45 jack you may connect the CP 343 to Twisted-Pair-Ethernet.

The slot has the following pin assignment:

8pin RJ45 plug:

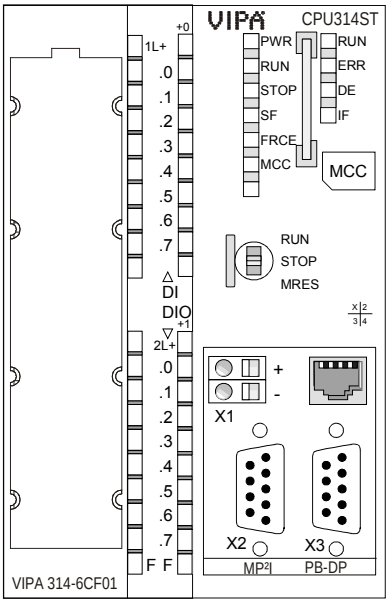


Pin	Assignment
1	Transmit +
2	Transmit -
3	Receive +
4	-
5	-
6	Receive -
7	-
8	-

In-/Output range CPU 314ST

Overview

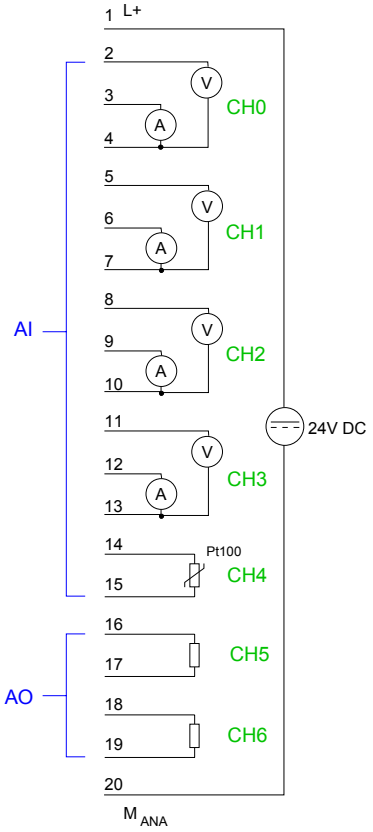
The CPU 314ST has the following integrated analog and digital in- and output ranges:



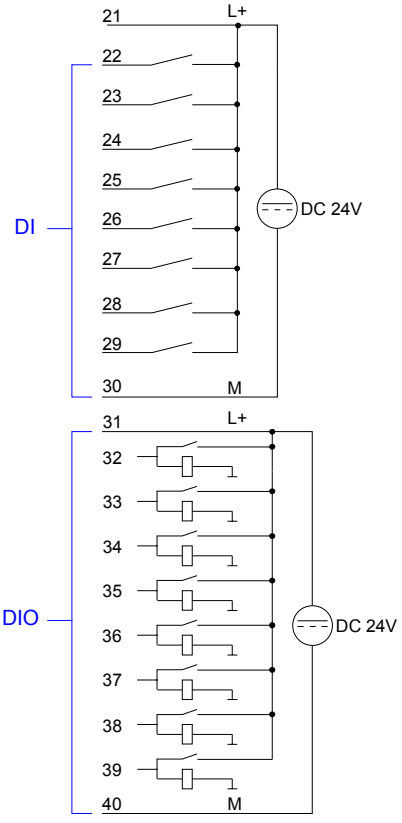
- AI 4x12Bit, 1xPt100
- AO 2x12Bit
- DI 8xDC24V alarm capable, the first 8 inputs parameterizable as 4 counters (100kHz)
- DIO 8xDC24V, 0.5A

Pin assignment

Analog part



Digital part



Analog part

The analog part consists of 4 input, 1 Pt100 and 2 output channels. 10byte for input and 4byte for output are used for the process image.

The channels of the module are galvanically separated from the SPEED-Bus via DC/DC transducer and optocouplers.

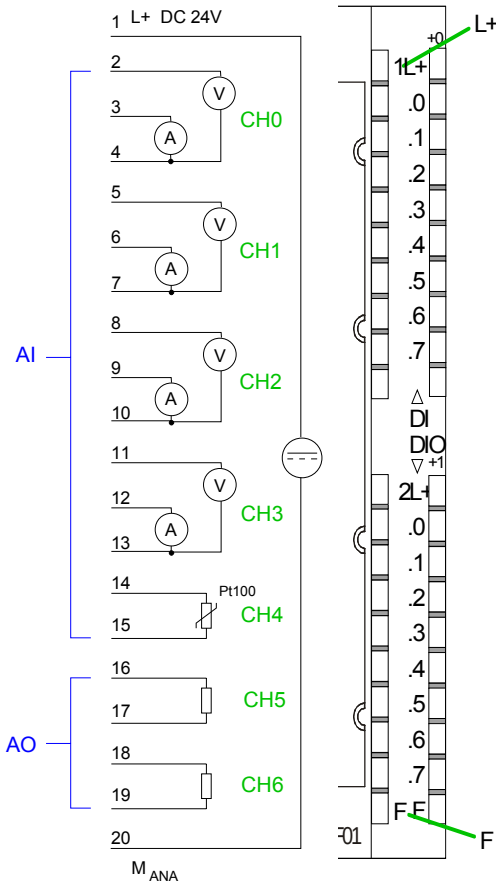
Attention!

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

Status indicator
Pin assignment

Pin	Assignment
1	Power supply DC 24V for Analog range
2	meas. voltage channel 0
3	meas. current channel 0
4	Ground channel 0
5	meas. voltage channel 1
6	meas. current channel 1
7	Ground channel 1
8	meas. voltage channel 2
9	meas. current channel 2
10	Ground channel 2
11	meas. voltage channel 3
12	meas. current channel 3
13	Ground channel 3
14	Pt 100 channel 4
15	Pt 100 channel 4
16	Output + channel 5
17	Ground output channel 5
18	Output + channel 6
19	Ground output channel 6
20	Ground power supply for analog range

Connection



LEDs

1L+	LED (green) Supply voltage available
F	LED (red) Sum error

Note!

To avoid measuring errors, you should connect only one measuring type per channel.

Digital part

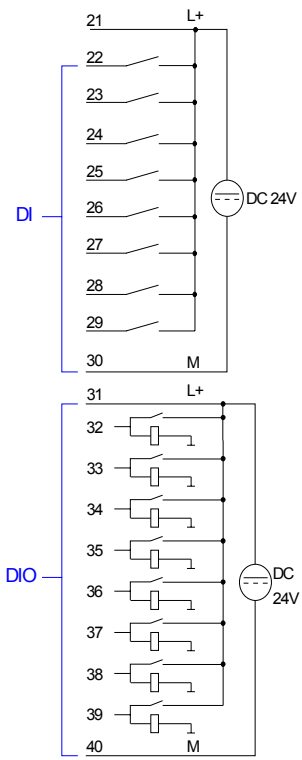
The digital part consists of 8 inputs and 8 in-/outputs. Each of this in-/outputs monitors its state via a LED. Via the parameterization you may assign alarm properties to every digital input. Additionally the digital inputs are parameterizable as counter.

The output channels provide a diagnostic function, i.e. as soon as an output is active, the according input is set to "1". At a short circuit at the load, the input is pulled to "0" and by evaluating the input, the error may be recognized. The DIO part has to be provided with external DC 24V.

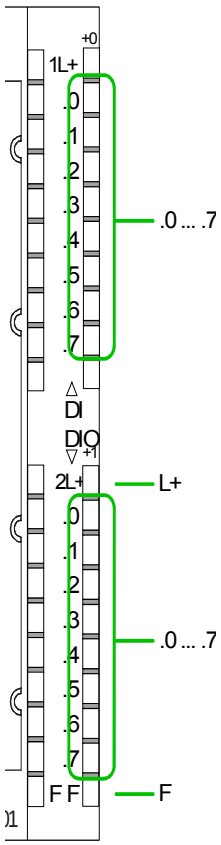
Status indicator
Pin assignment

Pin	Assignment
21	Power supply +DC 24V
22	Input I+0.0 / Counter 0(A)
23	Input I+0.1 / Counter 0(B)
24	Input I+0.2 / Gate0/Latch0/Reset0
25	Input I+0.3 / Counter 1(A)
26	Input I+0.4 / Counter 1(B)
27	Input I+0.5 / Gate1/Latch1/Reset1
28	Input I+0.6 / Counter 2(A)
29	Input I+0.7 / Counter 2(B)
30	Ground DI
31	Power supply +DC 24V
32	I/Q+1.8 / Gate2/Latch2/Reset2
33	I/Q+1.9 / Counter 3(A)
34	I/Q+1.10 / Counter 3(B)
35	I/Q+1.11 / Gate3/Latch3/Reset3
36	I/Q+1.12 / OUT0/Latch0/Reset0
37	I/Q+1.13 / OUT1/Latch1/Reset1
38	I/Q+1.14 / OUT2/Latch2/Reset2
39	I/Q+1.15 / OUT3/Latch3/Reset3
40	Ground DIO

Connection



LEDs



DI:

.0 7	LEDs (green) I+0.0 to I+0.7 (each Byte) Starting with app. 15V the signal "1" at the input is recognized and the according LED

DIO:

2L+	LED (green) Supply voltage available for DIO
.0 7	LEDs (green) I/Q+1.0 to I/Q+1.7 on at active output/input
F	LED (red) Overload or short circuit error

Attention!

Please take care that the voltage at an output channel always is ≤ the supply voltage via L+.

Further you have to regard that due to the parallel connection of in- and output channel per group a set output can be provided via a connected input signal. A thus set output remains active at connected input signal also the power supply is turned off.

Nonobservance may destroy the module.

Technical Data

CPU 315SB/DPM

Electrical Data	VIPA 315-2AG10	VIPA 315-2AG12
Supply voltage Current consumption Output current to backplane bus	DC 24V max. 1A max. 3A	
Status display (LEDs)	via LEDs at the front	
Total memory Expandable via MCC to External storage media	1MByte (512kByte code / 512kByte data) 2MByte (1MByte code / 1MByte data) MMC (Memory Card), MCC memory extension card	
Slots / Interfaces: MP^2I	MPI: 8 static / 8 dynamic connections (max. 1.5MBaud) RS232: 38.4kBaud (only via green cable from VIPA)	MPI: 8 static / 8 dynamic connections (max. 12MBaud)
RJ45 PG/OP channel	PG/OP channel via Ethernet with max. 2 connections	
RS485	Configurable functionality via project engineering: *deactivated* *Profibus DP Communication* - Transfer rate: 9.6kBaud to 12MBaud - Max. number of partners: 32 stations in every segment without repeater, with Repeater expandable to 126. - Protocol: DP-V0, DP-V1, PG/OP communication *PtP Communication* - Transfer rate: 0.15kBaud to 115.2kBaud - Max. number of partners: ASCII, RTX/ETX, 3964R: 1 Modbus: 256 Stationes, USS: 64 Stationes - Protocol: ASCII, STX/ETX, 3964R, USS_{Master}, Modbus $ASCII_{Master}/RTU_{Master}$	
Battery buffer / clock	Lithium battery, 30 days buffer / yes	
Execution time CPU for bit operation, min. for word operation, min. for fixed-point calculation, min. for floating-point calculation, min.	 0.015µs 0.015µs 0.015µs 0.090µs	 0.010µs 0.010µs 0.010µs 0.058µs
Flag byte / Timer / Counter Number of blocks	8192 / 512 / 512 FBs 2048, FCs 2048, DBs 4095	
Measurements and Weight		
Measurements (WxHxD) in mm Weight in g	40x125x120 290	

CPU 315SN/NET

Electrical Data	VIPA 315-4NE11	VIPA 315-4NE12
Supply voltage Current consumption Output current to backplane bus	DC 24V max. 1A max. 3A	
Status display (LEDs)	via LEDs at the front	
Total memory Expandable via MCC to External storage media	1MByte (512kByte code / 512kByte data) 2MByte (1MByte code / 1MByte data) MMC (Memory Card), MCC memory extension card	
Slots / Interfaces:		
MP^{2}I	MPI: 8 static / 8 dynamic connections (max. 1.5MBaud) RS232: 38.4kBaud (only via green cable from VIPA)	MPI: 8 static / 8 dynamic connections (max. 12MBaud)
RJ45 PG/OP channel	PG/OP channel via Ethernet with max. 2 connections	
RS485	Configurable functionality via project engineering: *deactivated* *Profibus DP Communication* - Transfer rate: 9.6kBaud to 12MBaud - Max. number of partners: 32 stations in every segment without repeater, with Repeater expandable to 126. - Protocol: DP-V0, DP-V1, PG/OP communication *PtP Communication* - Transfer rate: 0.15kBaud to 115.2kBaud - Max. number of partners: ASCII, RTX/ETX, 3964R: 1 Modbus: 256 Stationes, USS: 64 Stationes - Protocol: ASCII, STX/ETX, 3964R, USS_{Master}, Modbus $ASCII_{Master}$/RTU_{Master}	
RJ45-Ethernet	Twisted-Pair-Ethernet for CP Communication: - Transfer rate: 10/100MBit - Total length: max. 100m per Segment - PG/OP channel: 8 - Configurable connections: 8	
Battery buffer / clock	Lithium battery, 30 days buffer / yes	
Execution time CPU		
for bit operation, min.	0.015µs	0.010µs
for word operation, min.	0.015µs	0.010µs
for fixed-point calculation, min.	0.015µs	0.010µs
for floating-point calculation, min.	0.090µs	0.058µs
Flag byte / Timer / Counter Number of blocks	8192 / 512 / 512 FBs 2048, FCs 2048, DBs 4095	
Measurements and Weight		
Measurements (WxHxD) in mm Weight in g	80x125x120 430	

CPU 314ST/DPM
CPU 314ST/PtP

Electrical Data	VIPA 314-6CF01	VIPA 314-6CF02
Power supply Current consumption Output current to backplane bus	DC 24V max. 1.5A max. 5A (Standard + SPEED-Bus)	
Status display (LEDs)	via LEDs at the front	
Total memory Expandable via MCC to External storage media	512kByte (256kByte code / 256kByte data) 2MByte (1MByte code / 1MByte data) MMC (Memory Card), MCC memory extension card	
Slots / Interfaces: MP^2I	MPI: 8 static / 8 dynamic connections (max. 1.5MBaud) RS232: 38.4kBaud (only via green cable from VIPA)	MPI: 8 static / 8 dynamic connections (max. 12MBaud)
RJ45 PG/OP channel RS485	PG/OP channel via Ethernet with max. 2 connections Configurable functionality via project engineering: *deactivated* *Profibus DP Communication* - Transfer rate: 9.6kBaud to 12MBaud - Max. number of partners: 32 stations in every segment without repeater, with Repeater expandable to 126. - Protocol: DP-V0, DP-V1, PG/OP communication *PtP Communication* - Transfer rate: 0.15kBaud to 115.2kBaud - Max. number of partners: ASCII, RTX/ETX, 3964R: 1 Modbus: 256 Stationes, USS: 64 Stationes - Protocol: ASCII, STX/ETX, 3964R, USS_{Master}, Modbus $ASCII_{Master}/RTU_{Master}$	
SPEED-Bus - Data rate - Current consumption	 64MBaud 400mA	
Battery buffer / clock	Lithium battery, 30 days buffer / yes	
Execution time CPU for bit operation, min. for word operation, min. for fixed-point calculation, min. for floating-point calculation, min.	 0.015µs 0.015µs 0.015µs 0.090µs	 0,010µs 0,010µs 0,010µs 0,058µs
Flag byte / Timer / Counter Number of blocks	8192 / 512 / 512 FBs 2048, FCs 2048, DBs 4095	

Digital Input	DI 8 ... 16xDC24V alarm capable
Nominal input voltage Signal voltage "0" / "1" 2wire BERO permitted bias current: Input current Dissipation power Isolation	DC 24V 0 ... 5V / 15 ... 28.8V 1.5mA typ. 7mA 3.5W 500Veff (field voltage - backplane bus)

continued ...

... continue Technical Data

Digital Output	DO 8...0xDC 24V, 0.5A							
Nominal load voltage No-load current consumption at L+ Output current per channel Isolation	DC 24V from ext. power supply 30mA (all A.x=off) 0.5A short-circuit proof in groups per 8, 500Veff (field voltage - backplane bus)							
Analog In-/Output	AI 4x12Bit / AO 2x12Bit / AI 1xPt100							
Number of Current-/Voltage input Number of resistance input Number of outputs Length of cable: shielded	4 1 2 200m							
Voltages, Currents, Potentials								
Supply voltage L+ - reverse polarity protection Constant current for resistance sensor Isolation - channel / backplane (SPEED-Bus) - channel / power supply electronic - between the channels Permitted potential difference - between the inputs (U_{CM}) - between the inputs and $M_{INTERNAL}$ (U_{ISO}) Isolation tested with Current consumption - from the back plane bus - from the power supply L+ Power dissipation of the module	DC 24V yes 1.25mA yes yes no DC 11V DC 75V / AC 60V DC 500V - 85mA (without load) 2W							
Analog value calculation input	Conversion time/Resolution (per channel)							
Measuring principle	Sigma-delta							
Parameterizable	yes							
Conversion rate (Hz)	200	170	120	60	30	15	7.5	3.7
Integration time (ms)	5	6	8	17	33	67	133	270
Basic conversion time (ms)	6	7	9	18	34	68	134	268
Resolution (Bit) incl. overrange	10	12	14	15	16	16	16	16
Noise suppression for frequency f1 (Hz)	no				50 and 60Hz			
Basic execution time of the module, in ms (all channels enabled)	30	35	45	90	170	340	670	1340
Smoothing of the measured values	none							
Analog value calculation output channels								
Resolution (incl. overrange) ±10V, ±20mA 0 ... 10V, 0 ... 20mA 4 … 20mA Conversion time (per channel)	 11Bit + sign 11Bit 10Bit 1.0ms							
Settling time - impedance load - capacitive load - inductive load	 0.2ms 0.5ms 0.2ms							

continued ...

... continue Technical data

Suppression of interference, limits of error input channels		
Noise suppression for f=n x (f1 ±1%) (f1=interference frequency, n=1,2,...)		
Common-mode interference (U_{CM} < 11V)	> 80dB	
Series-mode noise (peak value of noise < nominal value of input range)	> 80dB	
Crosstalk between the inputs	> 50dB	
Operational limit (only valid to 120W/s) (in the entire temperature range, referring to input range)		
	Measuring range	Tolerance
voltage input	0 ... 10V	±0.4%
	±10V	±0,3%
current input	±20mA	±0.3%
	0 ... 20mA	±0.6%
	4 ... 20mA	±0.8%
Resistor	0 ... 600Ω	±0.4%
Resistance thermometer	Pt100, Pt1000	±0.6%
	Ni100, Ni1000	±1.0%
Basic error limit (only valid to 120W/s) (during temperature is 25°C, referring to input range)		
Voltage input	0 ... 10V	±0.3%
	±10V	±0.2%
Current input	±20mA	±0.2%
	0 ... 20mA	±0.4%
	4 ... 20mA	±0.5%
Resistors	0 ... 600Ω	±0.2%
Resistance thermometer	Pt100, Pt1000	±0.5K
	Ni100, Ni1000	±0.5K
Temperature error (with reference to the input range)		±0.005%/K
Linearity error (with reference to the input range)		±0.02%
Repeatability (in steady state at 25°C referred to the input range)		±0.05%
Suppression of interference, limits of error output channels		
Crosstalk between the outputs	> 40dB	
Operational limit (in the entire temperature range, referring to output range)		
	Measuring range	Tolerance
Voltage output	0 ... 10V	±0.8%
	±10V	±0.4%
Current output	±20mA	±0.4%[1)]
	0 ... 20mA	±0.6%[1)]
	4 ... 20mA	±0.8%[1)]

continued...

... continue Technical data

Basic error limit (during temperature is 25°C, referring to output range)		
	Measuring range	Tolerance
Voltage output	0 ... 10V	±0.6%
	±10V	±0.3%
Current output	±20mA	±0.3%[1)]
	0 ... 20mA	±0.4%[1)]
	4 ... 20mA	±0.5%[1)]
Temperature error (with reference to the output range)	±0.01%/K	
Linearity error (with reference to the output range)	±0.05%	
Repeatability (in steady state at 25°C referred to the output range)	±0.05%	
Output ripple; range 0 to 50kHz (referred to output range)	±0.05%	
States, Alarms, Diagnostic		
Diagnostic alarm	parameterizable	
Diagnostic functions		
- Sum error monitor	red LED (SF)	
- Diagnostic information readable	possible	
Substitute value can be applied	yes	
Data for choosing an encoder		
Voltage input		
±10V, 0 ... 10V	120kΩ	
Current input		
±20mA, 0 ... 20mA, 4 ... 20mA	33Ω	
Resistors		
0...600Ω	10MΩ	
Resistance thermometer		
Pt100, Pt1000, Ni100, Ni1000	10MΩ	
Maximum input voltage for voltage input (destruction limit)	25V	
Maximum input current for current input (destruction limit)	30mA	
Connection of the sensor		
- For measuring voltage	yes	
- For measuring current		
as 2wire transmitter	possible with external power supply	
as 4wire transmitter	yes	
- For measuring resistance		
with 2conductor connection	yes	
Characteristic linearization for Resistance thermometer	Pt100, Pt1000, Ni100, Ni1000	

continued ...

... continue Technical data

Data for choosing an actuator	
Output ranges (rated values)	
- Voltage	0 ... 10V, ±10V
- Current	4 ... 20mA, 0 ... 20mA, ±20mA
Load resistance	
(in nominal range of the output)	
- At voltage outputs	min. 1kΩ
capacitive load	max. 1µF
- At current output	max. 500Ω
Inductive load	max. 10mH
Voltage outputs	
- Short-circuit protection	yes
- Short-circuit current	max. 31mA
Current outputs	
- No-load voltage	max. 13V
Destruction limit against voltages/currents applied from outside	
- Voltage at outputs to M_{ANA}	max. 15V
- Current	max. 30mA
Connection of actuators	
- for voltage output	2conductor connection
- for current output	2conductor connection
Dimensions and weight	
Dimensions (WxHxD) in mm	80x125x120
Weight	480g

[1] The error limits are measured with a load of R=10Ω.

CPU 317SE/DPM

Electrical Data	VIPA 317-2AJ11	VIPA 317-2AJ12
Power supply Current consumption Output current to backplane bus	DC 24V max. 1.5A max. 5A (Standard bus + SPEED-Bus)	
Status display (LEDs)	via LEDs at the front	
Total memory Expandable via MCC to External storage media	2MByte (1MByte code / 1MByte data) 8MByte (4MByte code / 4MByte data) MMC (Memory Card), MCC memory extension card	
Slots / Interfaces:		
MP^2I	MPI: 8 static / 8 dynamic connections (max. 1.5MBaud) RS232: 38.4kBaud (only via Green Cable from VIPA)	MPI: 8 static / 8 dynamic connections (max. 12MBaud)
RJ45 PG/OP channel	PG/OP channel with max. 2 connections	
RS485	Configurable functionality via project engineering: *deactivated* *Profibus DP Communication* - Transfer rate: 9.6kBaud to 12MBaud - Max. number of partners: 32 stations in every segment without repeater, with Repeater expandable to 126. - Protocol: DP-V0, DP-V1, PG/OP communication *PtP Communication* - Transfer rate: 0.15kBaud to 115.2kBaud - Max. number of partners: ASCII, RTX/ETX, 3964R: 1 Modbus: 256 Stationes, USS: 64 Stationes - Protocol: ASCII, STX/ETX, 3964R, USS_{Master}, Modbus $ASCII_{Master}/RTU_{Master}$	
SPEED-Bus - Data rate - Current consumption	 64MBaud 400mA	
Batteriepufferung / Uhr	Lithium battery, 30 days buffer / yes	
Execution time CPU for bit operation, min. for word operation, min. for fixed-point calculation, min. for floating-point calculation, min.	 0.015µs 0.015µs 0.015µs 0.090µs	 0.010µs 0.010µs 0.010µs 0.058µs
Flag byte / Timer / Counter Number of blocks	8192 / 512 / 512 FBs 2048, FCs 2048, DBs 4095	16384 / 2048 / 2048 FBs 8192, FCs 8192, DBs 8190
Profibus Interface		
Interface Transfer rate Total length Max. number of partners Protocol	RS485 9.6kBaud to 12MBaud without repeater 100m at 12Mbaud with repeater up to 1000m 32 stations in every segment without repeater with Repeater expandable to 126. DP-V0, PG/OP communication	
Measurements and Weight		
Measurements (BxHxT) in mm Weight in g	80x125x120 420	

CPU 317SN/NET

Elektrical Data	VIPA 317-4NE11	VIPA 317-4NE12
Power supply Current consumption Output current to backplane bus	DC 24V max. 1.5A max. 5A (Standard bus + SPEED-Bus)	
Status display (LEDs)	via LEDs at the front	
Total memory Expandable via MCC to External storage media	2MByte (1MByte code / 1MByte data) 8MByte (4MByte code / 4MByte data) MMC (Memory Card), MCC memory extension card	
Slots / Interfaces:		
MP²I	MPI: 8 static / 8 dynamic connections (max. 1.5MBaud) RS232: 38.4kBaud (only via Green Cable from VIPA)	MPI: 8 static / 8 dynamic connections (max. 12MBaud)
RJ45 PG/OP channel	PG/OP channel via Ethernet with max. 2 connections	
RS485	Configurable functionality via project engineering: *deactivated* *Profibus DP Communication* - Transfer rate: 9.6kBaud to 12MBaud - max. number of partners: 32 stations in every segment without repeater, with Repeater expandable to 126. - Protocol: DP-V0, DP-V1, PG/OP communication *PtP Communication* - Transfer rate: 0.15kBaud to 115.2kBaud - max. number of partners: ASCII, RTX/ETX, 3964R: 1 Modbus: 256 Stationes, USS: 64 Stationes - Protokoll: ASCII, STX/ETX, 3964R, USS_{Master}, Modbus $ASCII_{Master}$/RTU_{Master}	
RJ45 Ethernet	Twisted-Pair-Ethernet for CP Communication: - Transfer rate: 10/100MBit - Total length: max. 100m per Segment - PG/OP channel: 8 - Configurable connections: 16	
SPEED-Bus - Data rate - Current consumption	 64MBaud 400mA	
Battery buffer / clock	Lithium battery, 30 days buffer / yes	
Execution time CPU for bit operation, min. for word operation, min. for fixed-point calculation, min. for floating-point calculation, min.	 0.015µs 0.015µs 0.015µs 0.090µs	 0.010µs 0.010µs 0.010µs 0,058µs
Flag byte / Timer / Counter Number of blocks	8192 / 512 / 512 FBs 2048, FCs 2048, DBs 4095	16384 / 2048 / 2048 FBs 8192, FCs 8192, DBs 8190
Measurements and Weight		
Measurements (WxHxD) in mm Weight in g	80x125x120 440	

Chapter 4 Deployment CPU 31xS

Overview

This chapter describes the employment of a CPU 31xS with SPEED7 technology in the System 300. The description refers directly to the CPU and to the employment in connection with peripheral modules that are mounted on a profile rail together with the CPU at SPEED-Bus respectively standard bus.

The following text describes

- Basics to assembly and operation of the CPU
- Start-up behavior and addressing
- Access to the website via Ethernet PG/OP channel
- Project engineering and parameterization
- Operating modes and overall reset
- MCC memory extension, firmware update and know how protection
- VIPA specific diagnostic entries
- test functions for controlling and monitoring variables

Content

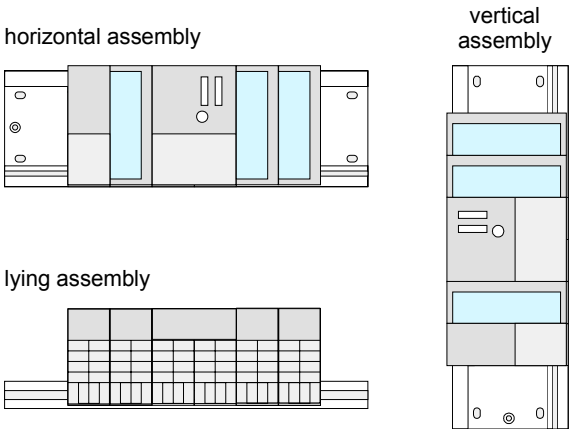
Note!

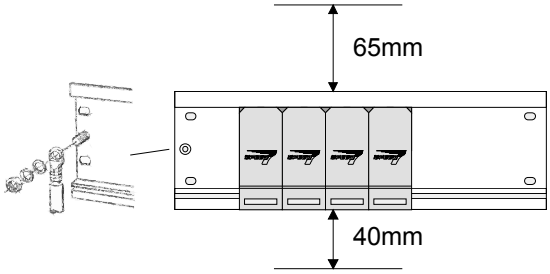
This information is valid for all the CPUs described in this manual, since the back panel communication between the CPU and the peripheral modules is the same for all models of CPU!

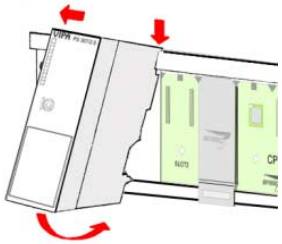
Assembly SPEED-Bus

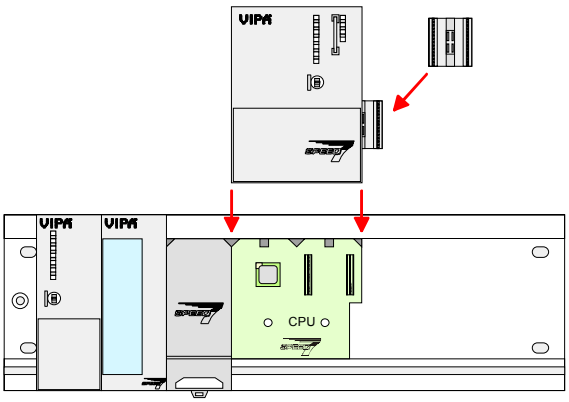
Pre-converted SPEED-Bus profile rail

For the employment of SPEED-Bus modules a pre-converted SPEED-Bus trailing socket required. This is available ready mounted on a profile rail with 2, 6, 10 or 16 extension sockets.





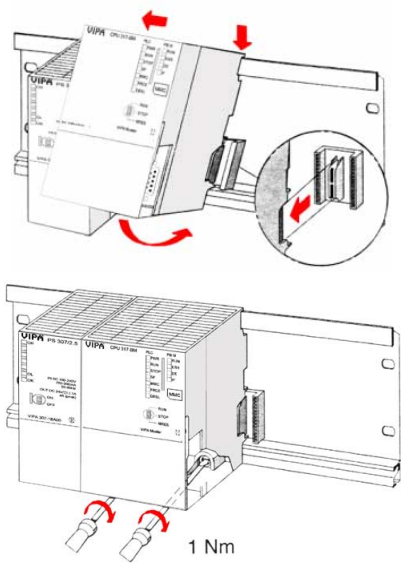




- At the assembly, please regard the permissible environment temperatures:
  - horizontal assembly: 0 to 60°C
  - vertical/lying assembly: 0 to 40°C
- Install the profile rail so that at least 65mm space are left above and 40mm below.
- Take care for a low-impedance connection between profile rail and subsoil and connect the profile rail with the ground wire via the dowel pin (min. 10mm^2) .
- Mount the power supply left of the SPEED-Bus.
- To install SPEED-Bus modules you put them between the triangular positioning helps of a slot labeled with "SLOT ..." and push them downwards.
- Only "SLOT1 DCDC" allows to mount an additional power supply instead of a SPEED-Bus module.
- If also standard modules shall be plugged, take a System 300 bus connector and stick it like shown to the CPU from the backside. When operating the SPEED7-CPU exclusively at the SPEED-Bus this is not required.
- Put the CPU like shown between the triangular positioning helps of the slot labeled with "CPU SPEED7" and push it downward.
- Repeat this procedure with the peripheral modules, by clicking a backplane bus coupler, stick the module right from the modules you've already fixed, click it downwards and connect it with the backplane bus coupler of the last module and bolt it.

Assembly without SPEED-Bus profile rail

Approach



- The assembly and grounding of the standard bus rail happens similar to that of the SPEED-Bus.
- Hang the power supply into position and push it to the left to app. 5mm before the grounding bolt of the profile rail.
- Take a bus connector and stick it to the CPU from the backside like shown.
- Mount the CPU at the right side of the power supply.
- Repeat this procedure with the peripheral modules, by clicking a backplane bus coupler, stick the module right from the modules you've already fixed, click it downwards and connect it with the backplane bus coupler of the last module and bolt it.
- Bolt all modules.

More details see chapter "Cabling and installation specifications".

Danger!

- The power supplies must be released before installation and repair tasks, i.e. before handling with the power supply or with the cabling you must disconnect current/voltage (pull plug, at fixed connection switch off the concerning fuse)!
- Installation and modifications only by properly trained personnel!

Start-up behavior

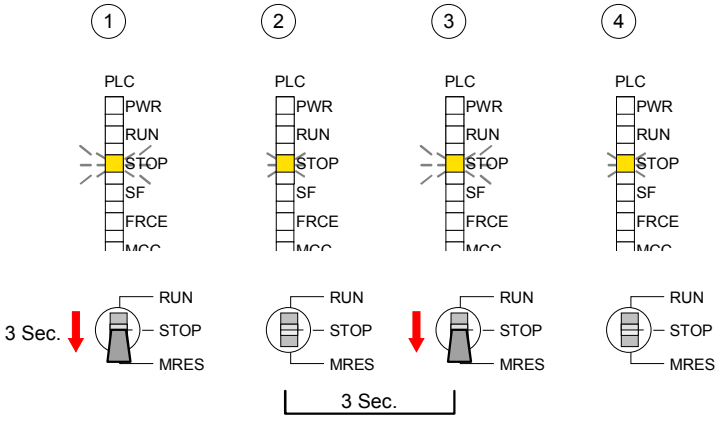
Turn on power supply

After the power supply has been switched on, the CPU changes to the operating mode the operating mode lever shows.

Now you may transfer your project to the CPU via MPI from your configuration tool res. plug in a MMC with your project and run an overall reset.

Overall reset

The following picture shows the approach once more:



Note!

The transfer of the application program 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 data in the CPU

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

Boot procedure with empty battery

The accumulator/battery is automatically loaded via the integrated power supply and guarantees a buffer for max. 30 days. If this time is exceeded, the battery may be totally discharged. This means that the battery buffered RAM is deleted.

In this state, the CPU executes an overall reset. If a MMC is plugged, program code and data blocks are transferred from the MMC into the work memory of the CPU.

If no MMC is plugged, the CPU transfers permanent stored "protected" blocks into the work memory if available.

Information about storing protected blocks in the CPU is to find in this chapter at "Extended Know-how protection".

Depending on the position of the RUN/STOP lever, the CPU switches to RUN res. remains in STOP.

This event is stored in the diagnostic buffer as: "Start overall reset automatically (unbuffered POWER_ON)".

Addressing

Overview

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

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

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

Modules at the SPEED-Bus are also taken into account at the automatic address allocation. Here the digital I/Os are stored beginning with address 128 and analog I/Os, FMs and CPs beginning with address 2048.

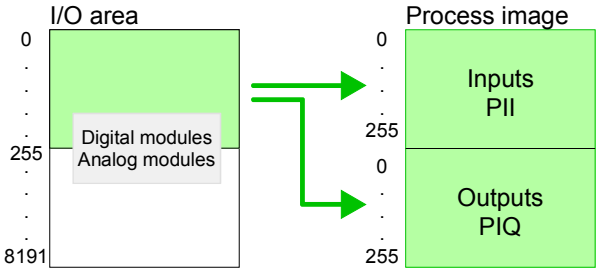
Addressing Backplane bus I/O devices

The SPEED7-CPU provides a I/O area (address 0 ... 8191) and a process image of the in- and outputs (each address 0 ... 255).

The process image stores the signal states of the lower address (0 ... 255) additionally in a separate memory area.

The process image this divided into two parts:

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



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

Up to 32 modules in one row

In the hardware configurator from Siemens you may parameterize maximum up to 8 modules per row. At employment of SPEED7-CPUs you may control up to 32 modules at the standard bus and 16 further modules at the SPEED-Bus. CPs and DP masters that are additionally virtual configured at the standard bus are taken into the count of 32 modules at the standard bus.

For the project engineering of more than 8 modules you may use virtual line interface connections. For this you set in the hardware configurator the module IM 360 from the hardware catalog to slot 3 of your 1st profile rail. Now you may extend your system with up to 3 profile rails by starting each with a IM 361 from Siemens at slot 3.

Define addresses by hardware configuration

You may access the modules with read res. write accesses to the peripheral bytes or the process image.

To define addresses a hardware configuration via a virtual Profibus system by including the SPEEDBUS.GSD may be used. For this, click on the properties of the according module and set the wanted address.

Attention!

Please take care not to configure a double address assignment at connection via external Profibus-DP masters - required for project engineering of a SPEED-Bus system! At external DP master systems, the Siemens hardware configurator does not execute an address check!

Automatic addressing

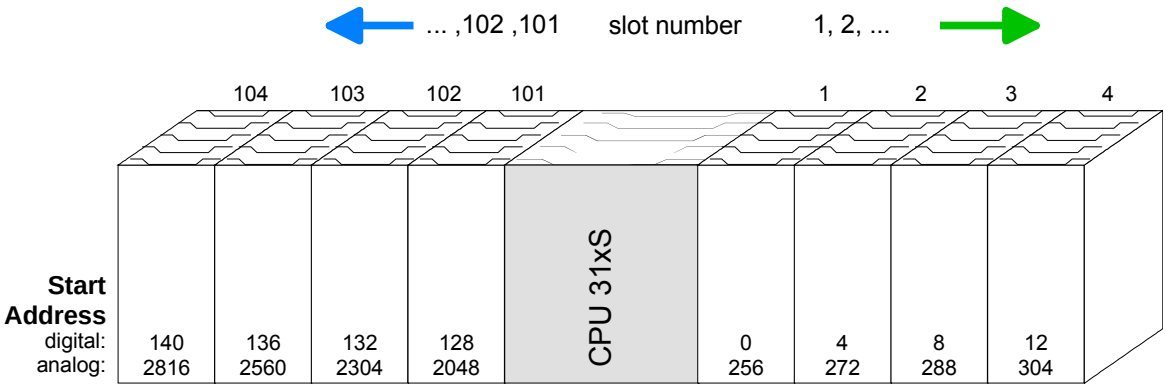
If you do not like to use a hardware configuration, an automatic addressing comes into force.

At the automatic address allocation DIOs occupy depending on the slot location always 4byte and AIOs, FMs, CPs always 16byte at the standard bus and 256byte at the SPEED-Bus.

Depending on the slot location the start address from where on the according module is stored in the address range is calculated with the following formulas:

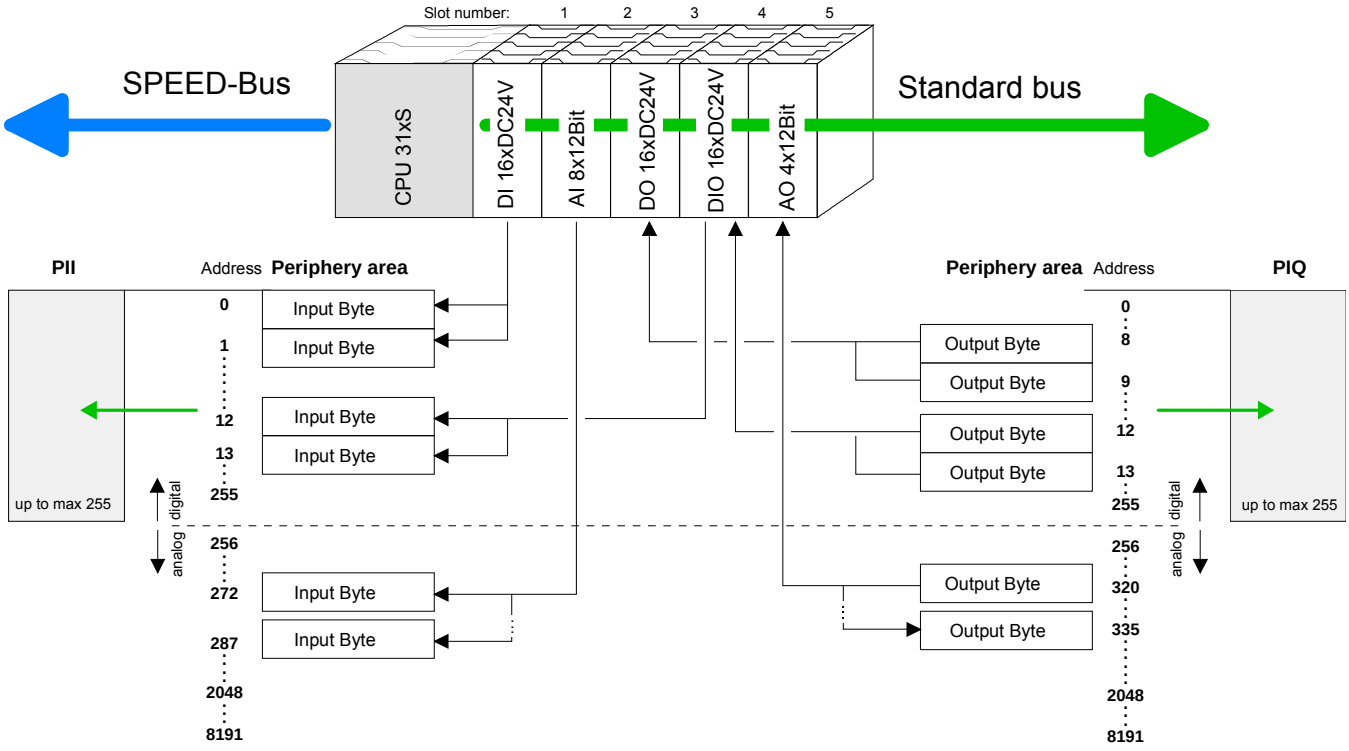
Standard bus	DIOs:	Start address = 4·(slot -1)
	AIOs, FMs, CPs:	Start address = 16·(slot -1)+256
SPEED-Bus	DIOs:	Start address = 4·(slot -101)+128
	AIOs, FMs, CPs:	Start address = 256·(slot -101)+2048

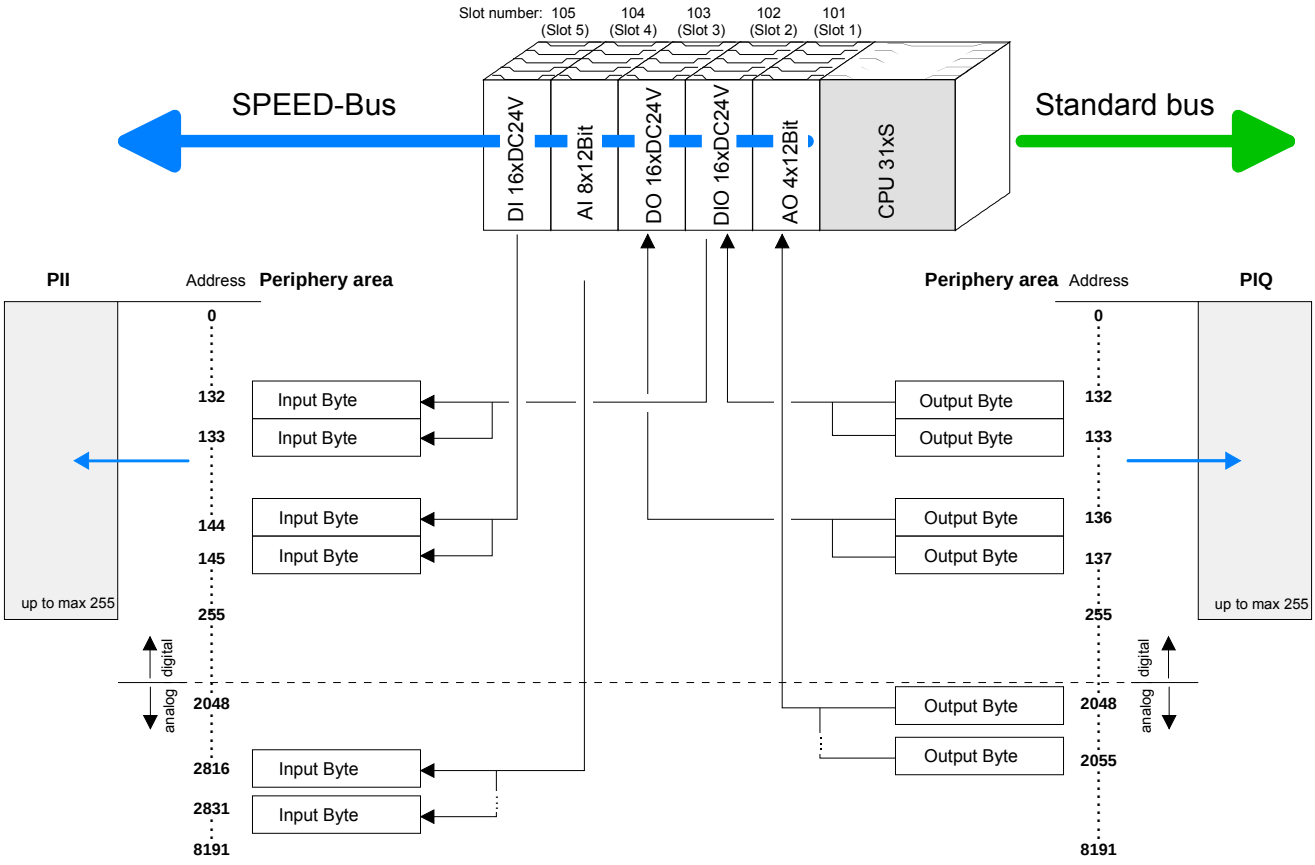
All information to this you may find in the following illustration:



Example for automatic address allocation

The following sample shows the functionality of the automatic address allocation separated for standard bus and SPEED-Bus:





Initialization Ethernet PG/OP channel

Overview

Every CPU 31xS has an integrated Ethernet PG/OP channel. This channel allows you to program and remote control your CPU with up to 2 connections.

The PG/OP channel also gives you access to the internal web page that contains information about firmware version, connected I/O devices, current cycle times etc.

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

Possibilities for Initialization

There are the following possibilities for assignment of IP address parameters (initialization):

- PLC functions with *Assign Ethernet address* (starting with firmware V. 1.6.0)
- Hardware project engineering with CP (Minimal project)

Requirements

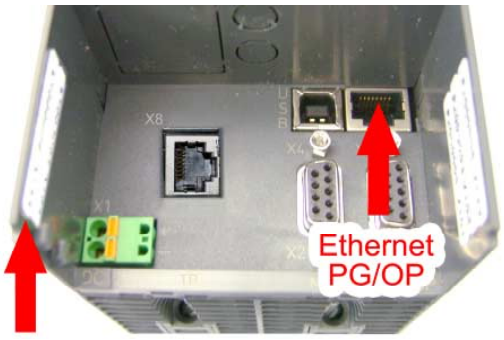
For the hardware configuration the following software is necessary:

- SIMATIC Manager from Siemens V. 5.1 or higher
- SIMATIC NET

Initialization via PLC functions

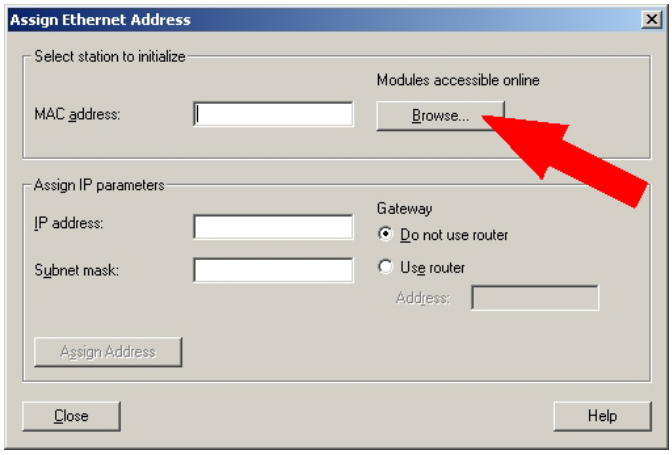
Please consider that this functionality is supported starting from the firmware version V. 1.6.0. The initialization takes place after the following proceeding:

- Determine the current Ethernet (MAC) address of your Ethernet PG/OP channel. This always may be found as 1st address under the front flap of the CPU on a sticker on the left side.

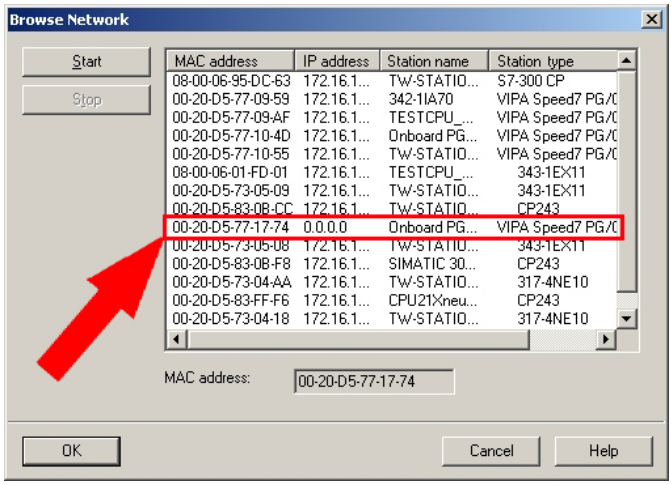


Ethernet address
1. Ethernet PG/OP
2. CP343 (optional)

- Establish a network connection between Ethernet PG/OP channel of the CPU and PC.
- Start the Siemens SIMATIC manager at the PC
- Set via **Options** > *Set PG/PC Interface* the Access Path to "TCP/IP -> Network card .... Protocol RFC 1006".
- Open with **PLC** > *Assign Ethernet Address* the dialog window for "initialization" of a station.



- Use the [Browse] button to determine the CPU components via MAC address.
  As long as the Ethernet PG/OP channel was not initialized yet, this owns the IP address 0.0.0.0 and the station name "Onboard PG/OP".

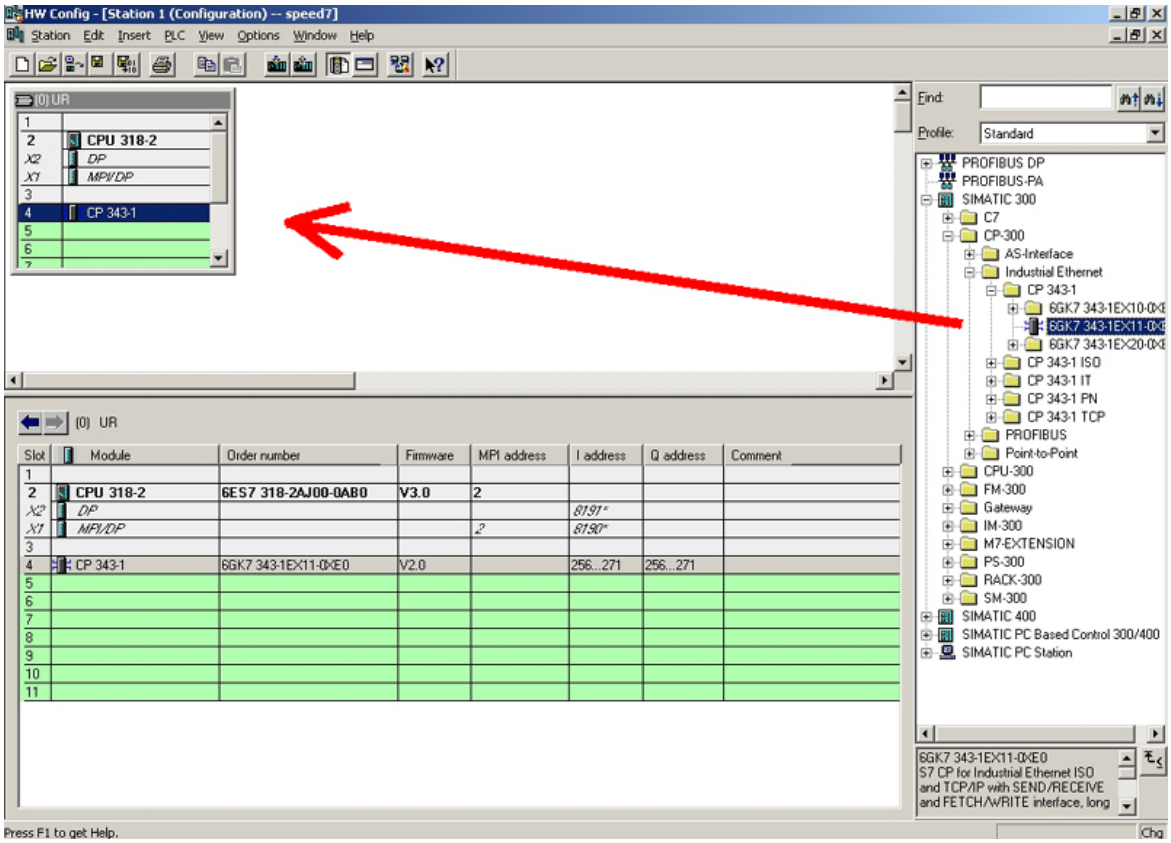


- Choose the determined module and click to [OK].
- Set the IP configuration by entering IP address, subnet mask and net transition. In addition an IP address may be received from a DHCP server. For this depending upon the selected option the MAC address, device name or the Client ID which can be entered here is to be conveyed to the DHCP server. The Client-ID is a character sequence from maximally 63 characters.
  Here the following indications may be used: Dash "-", 0-9, A-z, A-Z
- Confirm your settings by button [Assign Address]

Direct after the assignment the Ethernet PG/OP channel may be reached by the Siemens SIMATIC manager by means of these IP address parameters and the *Access Path* "TCP/IP -> Network card .... Protocol RFC 1006".

Initialization via minimal project

- Establish a network connection between Ethernet PG/OP channel of the CPU and PC.
- Start the SIMATIC Manager from Siemens and create a new project.
- Add a new System 300 station via **Insert** > *Station* > *SIMATIC 300-Station*.
- Activate the station "SIMATIC 300" and open the hardware configurator by clicking on "Hardware".
- Engineer a rack (SIMATIC 300 \ Rack-300 \ Profile rail)
- For the SPEED7-CPUs are configured as CPU 318-2, choose the CPU 318-2 with the order no. 6ES7 318-2AJ00-0AB0 V3.0 from the hardware catalog. You'll find this at SIMATIC 300 \ CPU 300 \ CPU 318-2.
- Include the CP 343-1EX11 at slot 4 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1).



- Type the wanted IP address and subnet mask into the dialog window of "Properties" of the CP 343-1 and connect the CP with "Ethernet".
- Save and compile your project.
- Transfer your project via MPI or MMC into your CPU. More information about transfer methods may be found in the chapter "Project transfer".

Direct after the assignment the Ethernet PG/OP channel may be reached by the Siemens SIMATIC manager by means of these IP address parameters and the *Access Path* "TCP/IP -> Network card .... Protocol RFC 1006".

Access to the internal Web page

Access to the web page

The Ethernet PG/OP channel provides a web page that you may access via an internet browser by its IP address. The web page contains information about firmware versions, current cycle times etc.

The current content of the web page is stored on MMC by means of the MMC-Cmd WEBPAGE. More information may be found at "MMC-Cmd - Auto commands".

Requirements

A PG/OP channel connection should be established between PC with Internet browser and CPU 31xS. This may be tested by *Ping* to the IP address of the PG/OP channel.

Web page

The access takes place via the IP address of the Ethernet PG/OP channel. The web page only serves for information output. The monitored values are not alterable.

```
CPU WITH ETHERNET PG/OP
Slot 100
  VIPA 317-4NE11 V2.0.6 Px000006.pkg,
  SERIALNUMBER 000638
  SUPPORTDATA :
  PRODUCT V2060, HARDWARE V0111, 5679B-V11,
  Bx000152 V5060, Ax000055 V1090, Ax000056
  V0200, FlashFileSystem:V102
  OnBoardEthernet : MacAddress : 0020d590001a,
  IP-Address : , SubnetMask : , Gateway :
  Cpu state : Stop
  FunctionRS485 : DPM-async
  Cycletime [microseconds] : min=0 cur=0 ave=0
  max=0

  MCC-Trial-Time: 70:23

Slot 201
  VIPA 342-1DA70 V3.0.1 Px000003.pkg,
  SUPPORTDATA :
  PRODUCT V3010, BB000154 V5010, AB000051
  V4000, AB000049 V3030 ModuleType CB2C0010,
  Cycletime [microseconds] : min=65535000
  cur=0 ave=0 max=0 cnt=0
```

Order no., firmware vers., package, serial no.
Information for support:

Ethernet PG/OP: Addresses

CPU status
RS485 function
CPU cycle time:
min= minimal
cur= current
max= maximal

Remaining time for deactivation of the expansion memory if MCC is removed.

Additional CPU components:

Slot 201 (DP master):
Name, firmware version, package
Information for support:

continued ...

... continue

```
Slot 202

  VIPA 343-1EX71 V1.8.0 Px000005.pkg,
  SUPPORTDATA :
  Bb000165 V1800, AB000060 V0320, AB000061
  V0310 PRODUCT V1800,
  ModuleType ACDB0000
  Address Input 1024...1039
  Address Output 1024...1039
```

Slot 202 (CP 343 or I/Os at the CPU 31xST):
Name, firmware version, package
Information for support:

```
SPEED-BUS
Slot 101
  VIPA 323-1BH70 V1.0.0 Px000031.pkg
  SUPPORTDATA :
  BB000191 V1002, AB000078 V1008,
  PRODUCT V1002, Hx000015 V1000,
  ModuleType 3FD20001
  Address Input 128...131
  Address Output 128...131
```

Modules at SPEED-Bus

Order no., firmware vers., package
Information for support:

```
Standard Bus 8 Bit Mode
Slot 4: ModulType:9FC3: Digital Input 32
  Baseaddress Input 0
```

Modules at standard bus
Type of module
Configured base address

Project engineering

Overview

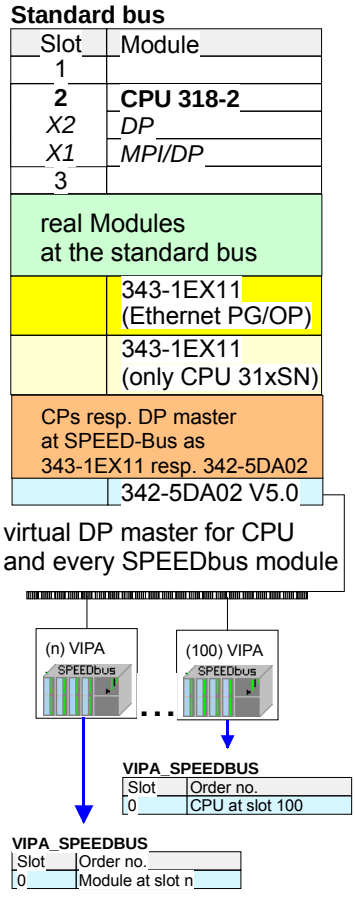
The project engineering of a SPEED-Bus system takes place at the Siemens hardware configurator and is divided into the following parts:

- Project engineering SPEED7-CPU as CPU 318-2DP (318-2AJ00-0AB00 V3.0)
- Project engineering really plugged modules
- Project engineering Ethernet PG/OP channel as CP 343-1 (343-1EX11)
- Project engineering and networking SPEED-Bus Ethernet-CP 343 and SPEED-Bus DP master as CP 343-1 (343-1EX11) respectively CP 342-5 (342-5DA02 V5.0)
- Project engineering each SPEED-Bus module as DP-Slaves (SPEEDBUS.GSD) in a virtual DP master CP 342-5 (342-5DA02 V5.0).

Fast introduction

For the employment of the System 300S modules from VIPA at the SPEED-Bus the inclusion of the System 300S modules via the GSD-file from VIPA in the hardware catalog is required.

To be compatible with the Siemens hardware configurator the following steps should be executed:



- Start the hardware configurator from Siemens and include the SPEEDBUS.GSD for SPEED7 from VIPA.
- Configure CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0) from Siemens. Configure a possibly existing internal DP master of your SPEED7-CPU via the internal DP master of the CPU 318-2DP. Leave *MPI/DP* of the CPU 318-2DP in MPI mode. The *Profibus* mode is not supported.
- Starting with slot 4, place the System 300 modules in the plugged sequence.
- For the internal Ethernet PG/OP channel that every SPEED7-CPU includes, you have to configure a Siemens CP 343-1 (343-1EX11) always as 1st module below the really plugged modules.
- If exists the integrated CP 343 of the CPU 31xSN/NET is also configured as CP 343-1 (343-1EX11) but always always as 2nd module below the before configured PG/OP channel. Else start here to configure and connect every Ethernet-CP 343 - SPEED-Bus as Siemens CP 343-1 (343-1EX11) resp. every SPEED-Bus Profibus DP master as Siemens CP 342-5DA02 V5.0.
- For the SPEED-Bus you always include, connect and parameterize to the *operating mode* DP master the DP master CP 342-5 (342-5DA02 V5.0) as last module. To this master system you assign every SPEED-Bus module as VIPA_SPEEDBUS slave. Here the Profibus address corresponds to the slot no. Beginning with 100 for the CPU. Place on slot 0 of every slave the assigned module and alter the parameters if needed.
- Let with the CPs or DP master (also virtual SPEED-Bus master) at *options* the attitude "Save configuration data on the CPU" activated!

Requirements

The hardware configurator is a part of the Siemens SIMATIC Manager. It serves the project engineering. The modules that may configured here are listed in the hardware catalog.
For the deployment of the System 300S modules at the SPEED-Bus, the inclusion via the GSD-file SPEEDBUS.GSD from VIPA in the hardware catalog is required.

Note!

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

Include the SPEED7-GSD-file

The GSD (**G**eräte-**S**tamm-**D**atei) is online available in the following language versions. Further language versions are available on inquires.

Name	Language
SPEEDBUS.GSD	german (default)
SPEEDBUS.GSG	german
SPEEDBUS.GSE	english

The GSD files can be found at www.vipa.de at the Service part and at the VIPA ftp server at ftp.vipa.de/support/profibus_gsd_files.

The integration of the SPEEDBUS.GSD takes place with the following proceeding:

- Browse to www.vipa.de.
- Click to *Service* > *Download* > *GSD- and EDS-Files* > *Profibus*.
- Download the file *Cx000023_Vxxx*.
- Extract the file to your work directory. The SPEEDBUS.GSD is stored in the directory *VIPA_System_300S*.
- Start the hardware configurator from Siemens.
- Close every project.
- Select **Options** > *Install new GSD-file.*
- Navigate to the directory *VIPA_System_300S* and select **SPEEDBUS.GSD**.

The modules of the System 300S from VIPA are now included in the hardware catalog at *Profibus-DP / Additional field devices / I/O / VIPA_SPEEDBUS*.

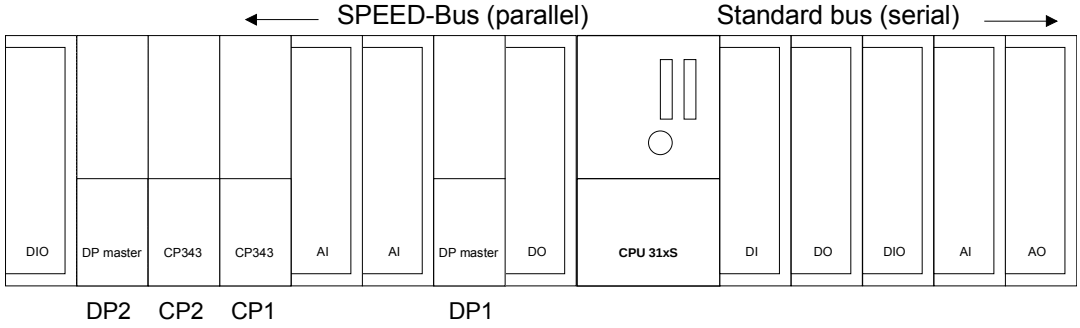
Steps of the project engineering

The following text describes the approach of the project engineering in the hardware configurator from Siemens at an abstract sample.

The project engineering is separated into 5 parts:

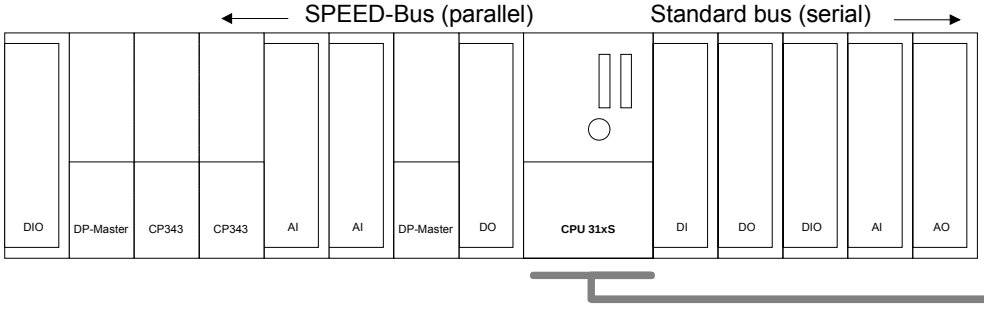
- Project engineering of the CPU
- Project engineering of the real plugged modules at the standard bus
- Project engineering of the PG/OP channel and CP343 (only CPU 31xSN/NET)
- Project engineering of all SPEED-Bus CPs and DP master
- Project engineering of the SPEED-Bus modules in a virtual master system

Hardware assembly



Project engineering of the CPU as CPU 318-2DP

- Start the hardware configurator from Siemens with a new project and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
  CPU 318-2DP (6ES7 318-2AJ00-0AB0 V3.0)
- Configure a possibly existing internal DP master of your SPEED7-CPU via the internal DP master of the CPU 318-2DP. Leave *MPI/DP* of the CPU 318-2DP in MPI mode. The *Profibus* mode is not supported.



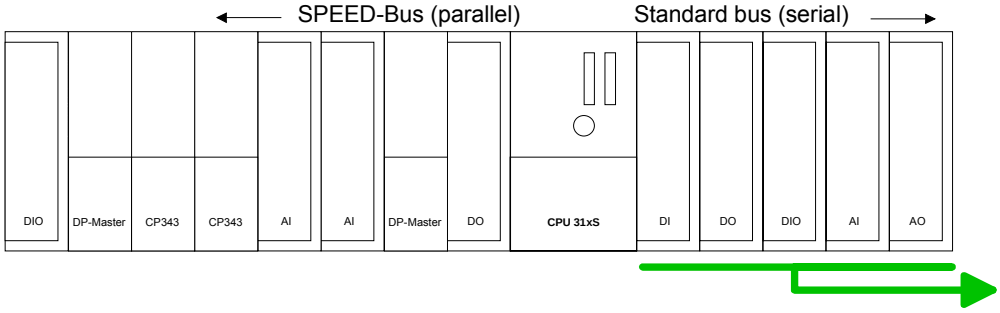
Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	

Project engineering of the modules at the Standard bus

The modules at the right side of the CPU at the Standard bus are configured with the following approach:

- Include your System 300 modules at the standard bus in the plugged sequence starting with slot 4.
- Parameterize the CPU res. the modules where appropriate. The parameter window opens by a double click on the according module.



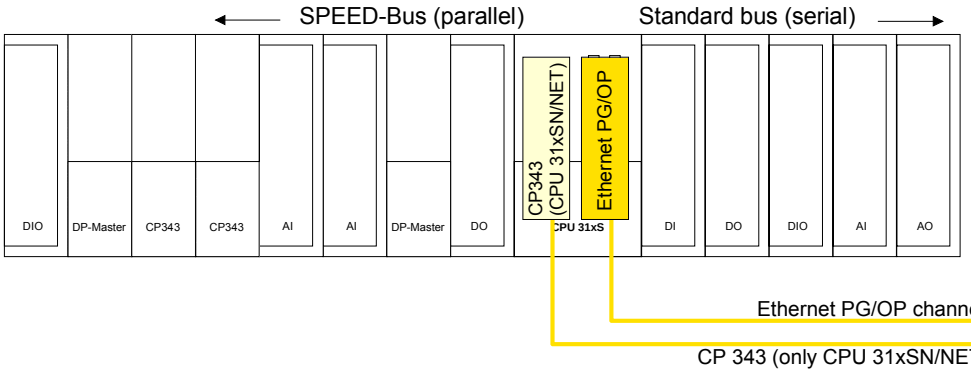
Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
4	DI
5	DO
6	DIO
7	AI
8	AO
9	
10	
11	

Project engineering of Ethernet PG/OP channel and CP 343 as 343-1EX11

For the internal Ethernet PG/OP channel that every SPEED7-CPU includes, you have to configure a Siemens CP 343-1 (343-1EX11) always as 1st module below the really plugged modules.

If exists the integrated CP 343 of the CPU 31xSN/NET is also configured and connected as CP 343-1 (343-1EX11) but always below the before configured PG/OP channel.



Standard bus

Slot.	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
4	DI
5	DO
6	DIO
7	AI
8	AO
9	343-1EX11
10	343-1EX11
11	

Set IP parameters

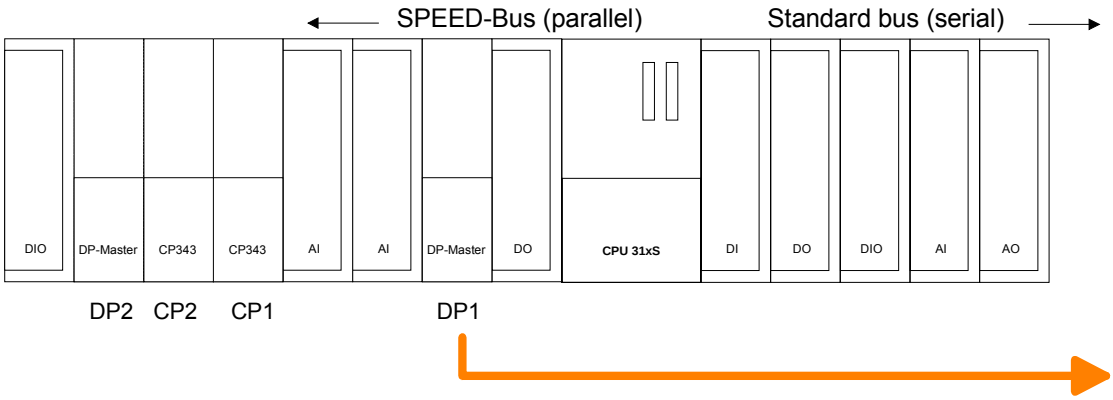
Open the property window via double-click on the CP 343-1EX11. Enter "General" and click at [Properties]. Type in the *IP address*, *subnet mask* and *gateway* for the CPs and select the wanted *subnet*.

Project engineering and networking each SPEED-Bus CP 343 and DP master at the Standard bus

Due to the fact that a Ethernet-CP 343 - SPEED-Bus and SPEED-Bus DP master is similar in project engineering and parameterization to the corresponding CP from Siemens, for each SPEED-Bus CP a corresponding Siemens CP is to be placed and networked.

Here the sequence follows the one at the SPEED-Bus from the right to the left within a function group (CP respectively DP master).

Use for each Ethernet-CP 343 - SPEED-Bus a Siemens CP 343-1 (343-1EX11) and for each SPEED-Bus Profibus DP master a Siemens CP 342-5DA02 V5.0.

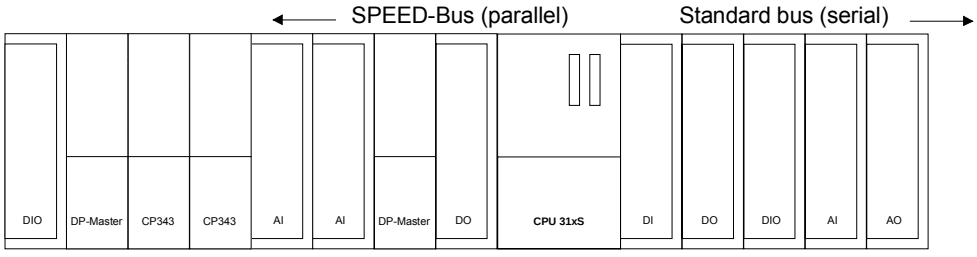


Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	IM360
4	DI
5	DO
6	DIO
7	AI
8	AO
9	CP343-1EX11
10	CP343-1EX11
11	CP342-5

Bus extension with IM 360 and IM 361

Since as many as 32 modules can be addressed by the SPEED7 CPU in one row, but only 8 modules are supported by the Siemens SIMATIC manager, the IM 360 of the hardware catalog can be used as a virtual bus extension during project engineering. Here 3 further extension racks can be virtually connected via the IM 361. Bus extensions are always placed at slot 3. Place the system expansion and project the remaining CPs.



Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	IM360
4	DI
5	DO
6	DIO
7	AI
8	AO
9	CP343-1EX11
10	CP343-1EX11
11	CP342-5

Standard bus (Extension 1)

Slot	Module
1	
2	
3	IM361
4	CP342-5
5	343-1EX11
6	343-1EX11
7	
8	
9	
10	
11	

Project engineering each SPEED-Bus module in a virtual master system

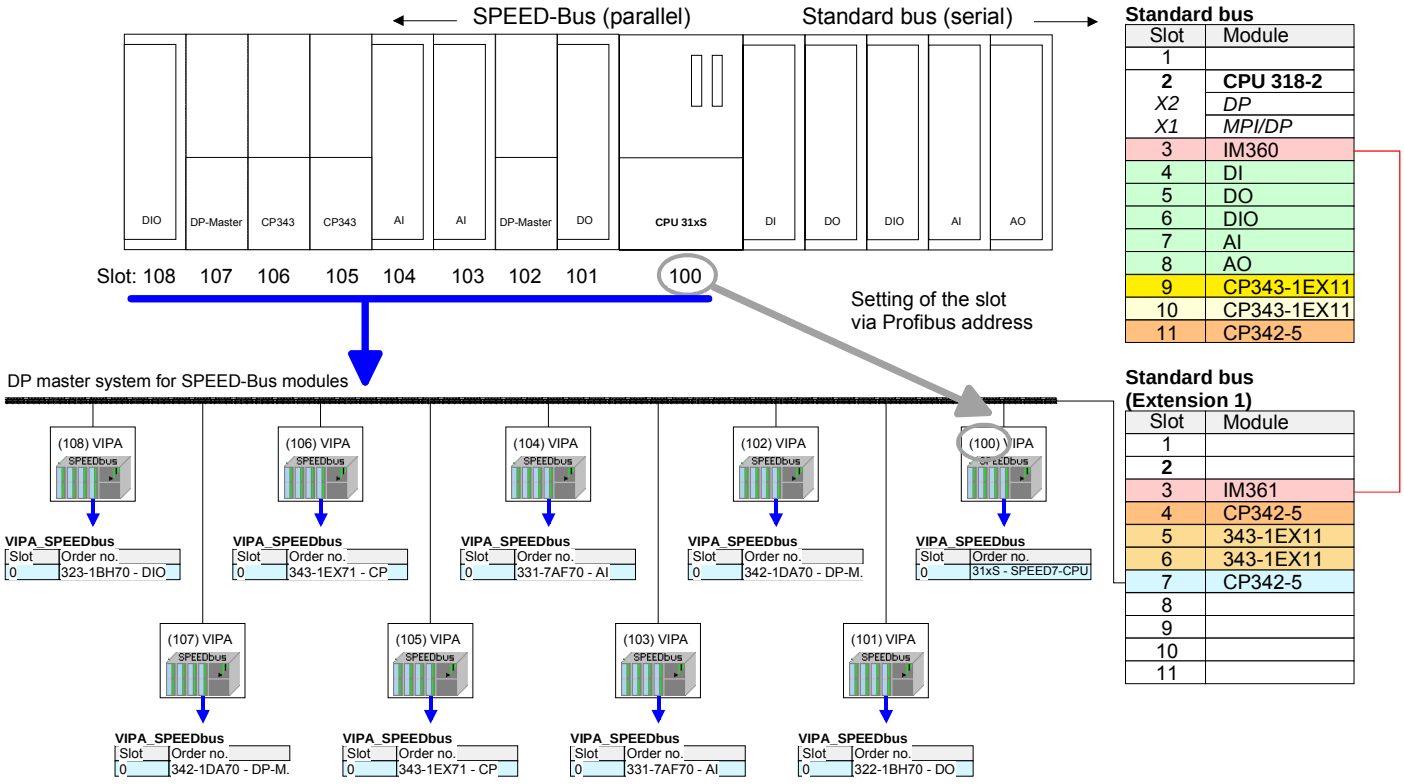
The slot assignment of the SPEED-Bus modules and the parameterization of the I/O devices happens via a virtual Profibus DP master system. For this, place as last module a DP master (342-5DA02 V5.0) with master system.

For the employment of the System 300S modules at the SPEED-Bus the inclusion of the System 300S modules into the hardware catalog via the GSD-file speedbus.gsd from VIPA is required.

After the installation of the speedbus.gsd you may locate at *Profibus DP* / *Additional field devices* / *I/O* / *VIPA_SPEEDbus* the DP slave system vipa_speedbus.

Now include for the CPU and every module at the SPEED-Bus a slave system "vipa_speedbus".

Set as Profibus address the slot no. 100 for the CPU and 101...116 for the modules and place the according CPU respectively module from the hardware catalog of VIPA_speedbus to slot 0 of the slave system.



The according module is to be taken from the hardware catalog to slot 0 of VIPA_SPEEDBUS.

Note!

Let with the CPs or DP master (also virtual SPEED-Bus master) at *options* the attitude "Save configuration data on the CPU" activated!

The following pages show information about parameterizing the CPU and modules and how to transfer the project into the CPU.

CPU parameterization

Overview

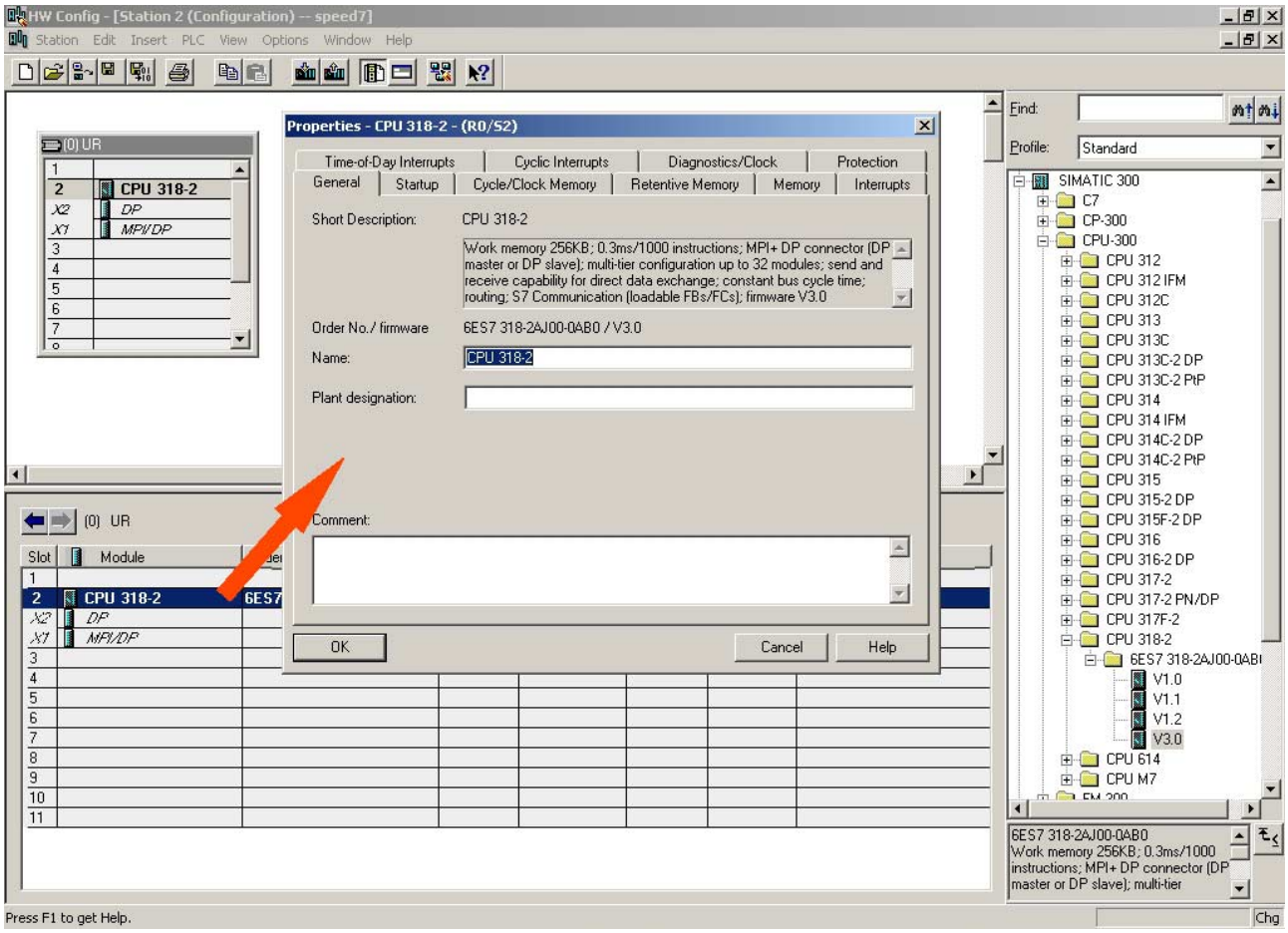
Except of the VIPA specific CPU parameters the CPU parameterization takes place in the parameter dialog of the CPU 318-2DP.

The VIPA specific CPU parameters like the RS485 interface behavior, the synchronization between CPU and DP master and the behavior of watchdog interrupt OBs (priority, execution, phase offset) may be configured in the SPEED-Bus CPU parameter dialog.

Parameterization via Siemens CPU 318-2DP

For the SPEED7-CPUs are configured in the hardware configurator from Siemens as Siemens CPU 318-2DP you may adjust the parameters for the SPEED7 CPUs at the hardware configuration at "Properties" of the CPU 318-2DP.

Via a double-click on the CPU 318-2 DP the parameter window of the CPU can be achieved. Using the registers you get access to all parameters of the CPU.



Supported parameters

The CPU does not evaluate all parameters that can be set at the hardware configuration. The following parameters are supported at this time:

General :

- Name, Plant designation, Comment
- MPI-Address of the CPU*
- Baudrate (10.2kB, 187.5kB, 1.5MB)*
- maximum MPI address*

Start-up:

- Start-up when expected/actual configuration differ
- Finished message by modules
- Transfer of the parameters to modules

Retentive memory:

- No. of memory bytes starting with MB0
- Number of S7 Timers starting withT0
- Number of S7 Counters starting with C0

Protection:

- Level of protection / Password

Memory:

- Local data
- Maximum size local stack

Time of day interrupts:

- OB10: Prio., Execution, Active Start date, Time-of-day
- OB11: Prio., Execution, Active Start date, Time-of-day

Cyclic interrupt:

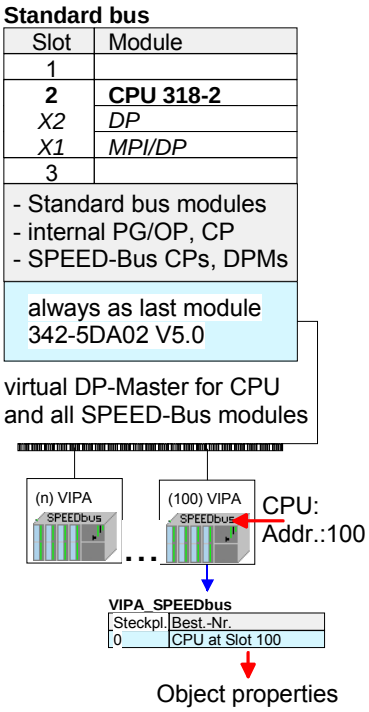
- OB32: Priority, Execution, Phase offset
- OB35: Prio., Exec., Phase offs.

Cycle / Clock memory:

- Update OB1 process image cyclically
- Scan Cycle monitoring time
- Minimum scan cycle time
- Scan Cycle load from communication (%)
- Size of the process image input area
- Size of the process image output area
- OB85 call up at I/O access error
- Clock memory with clock memory number

*) via properties MPI/DP

VIPA specific parameters via SPEED7-CPU



Via a hardware configuration the VIPA specific parameters of the SPEED7 CPU may be configured.

Via a double-click on the inserted CPU 31xS at SPEED-Bus the parameter window of the SPEED7 CPU may be achieved.

As soon as the project is transferred together with the PLC user program to the CPU, the parameters will be taken after start-up.

A description of the VIPA specific parameters of the CPU 31xS may be found at the following pages.

Function RS485

Per default, every CPU 31xS uses the RS485 interface for the Profibus DP master.

Using this parameter the RS485 interface may be switched to PtP communication (**p**oint **t**o **p**oint) respectively the synchronisation between DP master system and CPU may be set:

Deactivated	Deactivates the RS485 interface
PtP	With this operating mode the Profibus DP master is deactivated and the RS485 interface acts as an interface for serial point to point communication. Here data may be exchanged between two stations by means of protocols. More about this may be found at chapter "Deployment RS485 for PtP communication" in this manual.
Profibus-DP async	Profibus DP master operation asynchronous to CPU cycle Is there a Profibus DP integrated to your CPU 31xS, the RS485 interface is preset to *Profibus-DP async*. Here CPU cycle and cycles of every SPEED-Bus DP master run independently.
Profibus-DP syncIn	CPU is waiting for DP master input data
Profbus-DP syncOut	DP master system is waiting for CPU output data.
Profibus-DP syncInOut	CPU and DP master system are waiting on each other and form thereby a cycle.

Default: Profibus-DP async

Synchronization between master system and CPU

Normally the cycle of CPU and DP master run independently. The cycle time of the CPU is the time needed for one OB1 cycle and for reading respectively writing the inputs respectively outputs. The cycle time of a DP Master depends among others on the number of connected slaves and the baud rate, thus every plugged DP master has its own cycle time.

Due to the asynchronism of CPU and DP master the whole system gets relatively high response times.

The synchronization behavior between every SPEED-Bus Profibus DP master and the SPEED7 CPU can be configured by means of a hardware configuration as shown above.

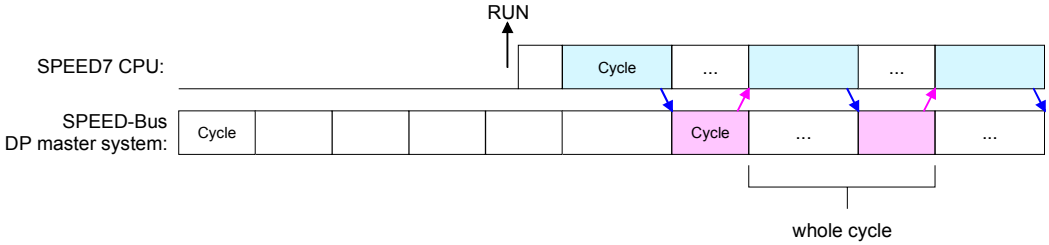
The different modes for the synchronization are in the following described.

Profibus-DP SyncInOut

In *Profibus-DP SyncInOut* mode CPU and DP-Master-System are waiting on each other and form thereby a cycle. Here the whole cycle is the sum of the longest DP master cycle and CPU cycle.

By this synchronization mode you receive global consistent in-/ output data, since within the total cycle the same input and output data are handled successively by CPU and DP master system.

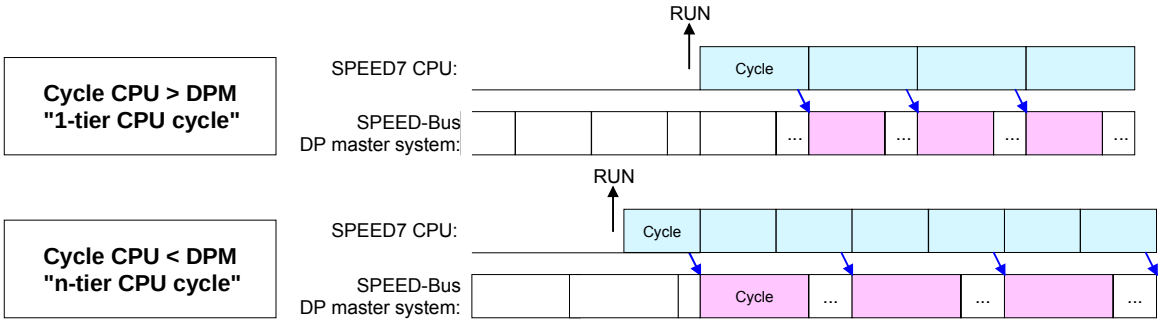
If necessary the time of the *Watchdog* of the bus parameters should be increased at this mode.



Profibus-DP SyncOut

In this operating mode the cycle time of the SPEED-Bus DP master system depends on the CPU cycle time. After CPU start-up the DP master gets synchronized.

As soon as their cycle is passed they wait for the next synchronization impulse with output data of the CPU. So the response time of your system can be improved because output data were directly transmitted to the DP master system. If necessary the time of the *Watchdog* of the bus parameters should be increased at this mode.

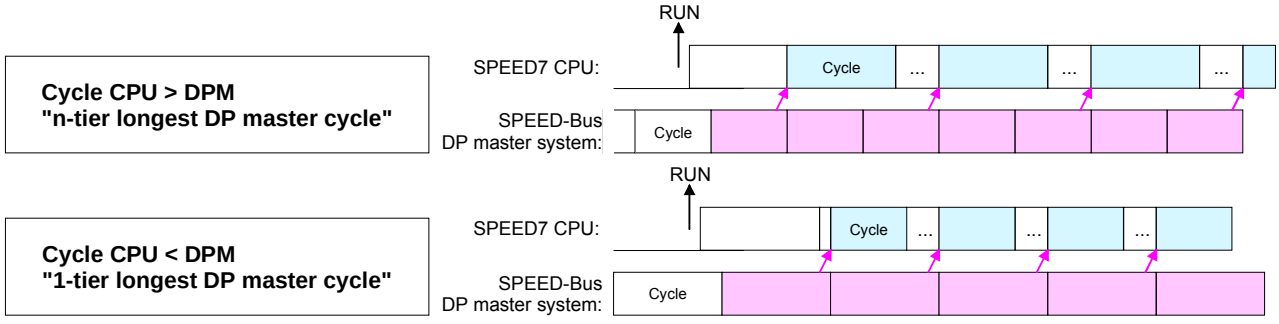


Profibus-DP SyncIn

In the operating mode *Profibus-DP SyncIn* the CPU cycle is synchronized to the cycle of the SPEED-Bus Profibus DP master system.

Here the CPU cycle depends on the speed bus DP master with the longest cycle time. If the CPU gets into RUN it is synchronized with all speed bus DP master. As soon as the CPU cycle is passed it waits for the next synchronization impulse with input data of the DP master system.

If necessary the *Scan Cycle Monitoring Time* of the CPU should be increased.



Number remanence flag

Here the number of flag bytes may be set. With 0 the value *Retentive memory > Number of memory bytes starting with MB0* set at the parameters of the Siemens CPU 318-2DP is used. Otherwise the adjusted value (1 ... 8192) is used.

Default: 0

Phase offset and execution of OB33 and OB34

The CPU offers additional cyclic interrupts which interrupt the cyclic processing in certain distances. Point of start of the time interval is the change of operating mode from STOP to RUN.

To avoid that the cyclic interrupts of different cyclic interrupt OBs receive a start request at the same time and so a time out may occur, there is the possibility to set a phase offset respectively a time of execution.

The phase offset (0 ... 60000ms) serves for distribution processing times for cyclic interrupts across the cycle.

The time intervals, in which the cyclic interrupt OB should be processed may be entered with *execution* (1 ... 60000ms).

Default: Phase offset: 0
Execution: OB33: 500ms
OB34: 200ms

Priority of OB28, OB29, OB33 and OB34

The priority fixes the order of interrupts of the corresponding interrupt OB.

Here the following priorities are supported:

0 (Interrupt-OB ist deactivated), 2,3,4,9,12,16,17, 24

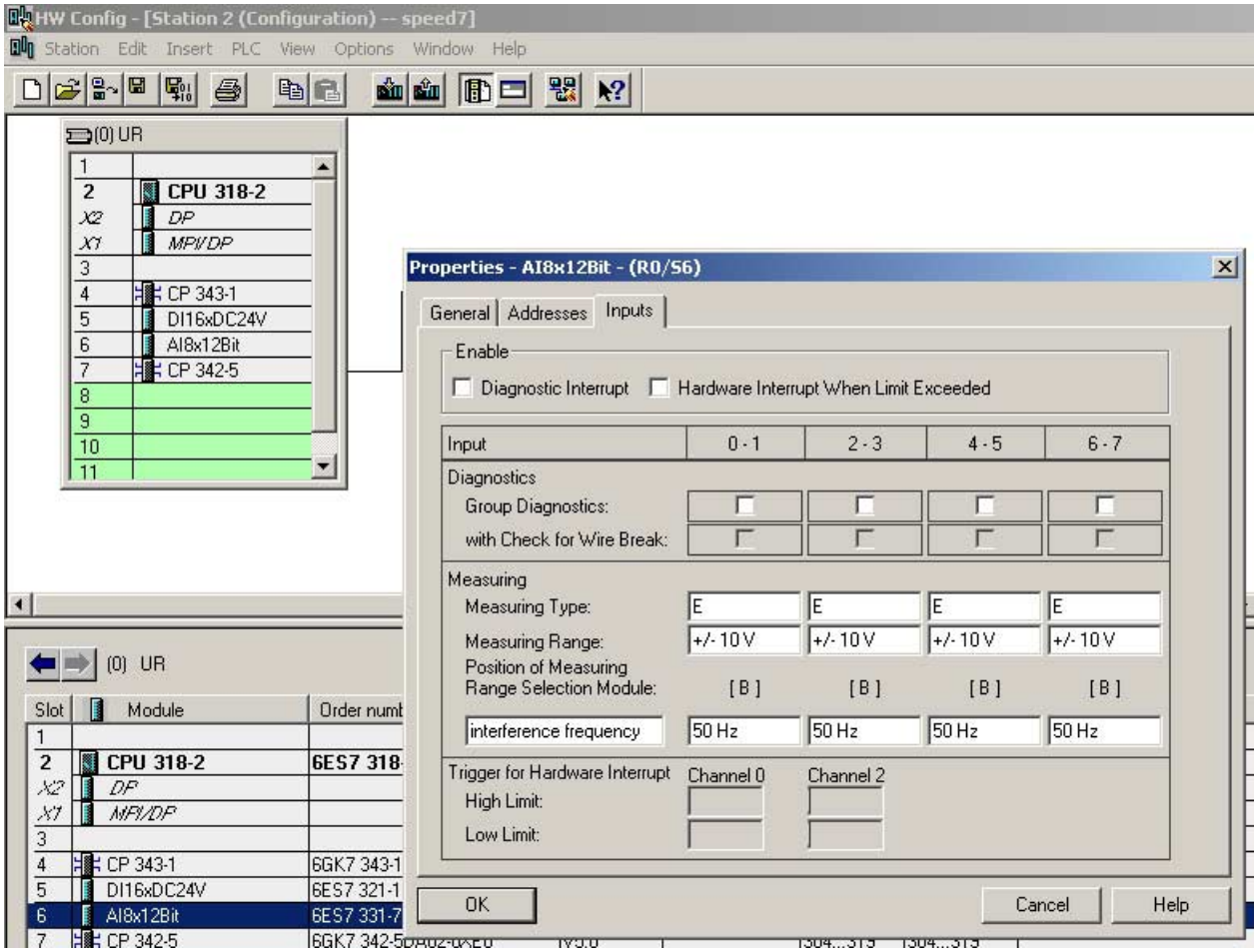
Default: 24

Parameterization of modules

Approach

By using the SIMATIC Manager from Siemens you may set parameters for configurable System 300 modules at any time.

For this, double-click during the project engineering at the slot overview on the module you want to parameterize In the appearing dialog window you may set the wanted parameters.



Parameterization during runtime

By using the SFCs 55, 56 and 57 you may alter and transfer parameters for wanted modules during runtime.

For this you have to store the module specific parameters in so called "record sets".

More detailed information about the structure of the record sets is to find in the according module description.

Project transfer

Overview

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

- Transfer via MPI respectively Profibus
- Transfer via MMC
- Transfer via integrated Ethernet PG/OP channel (Initialization necessary)

Transfer via MPI respectively Profibus

For the SPEED7-CPUs provide a MPI respectively Profibus jack you have the following transfer options:

- Transfer via MPI Programming cable (MPI/Profibus Communication)
- Only MP^{2}I jack: Transfer with VIPA Green Cable as serial communication via MP^{2}I - not Profibus

Transfer with MPI Programming cable via MPI resp. Profibus

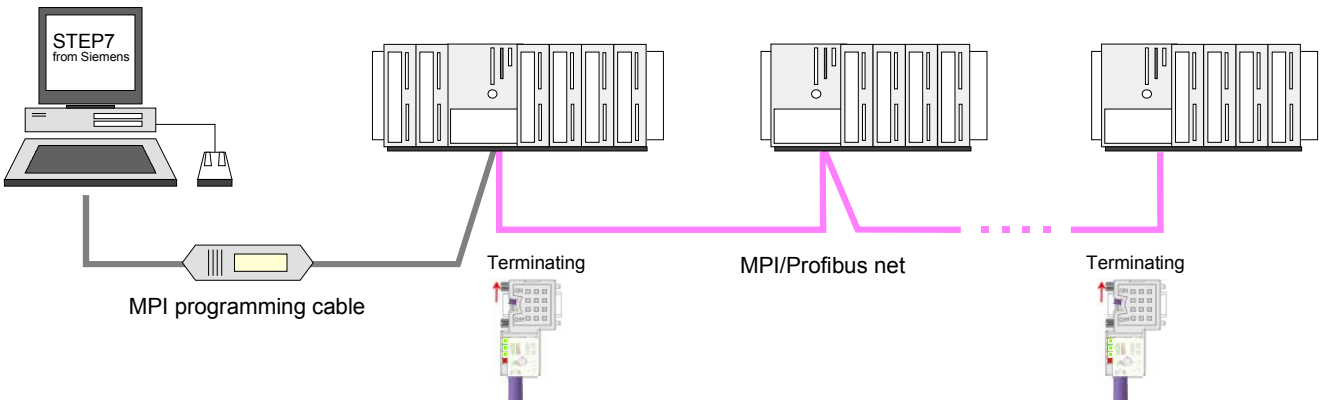
The MPI programming cables are available at VIPA in different variants. The employment of the cables is identical. The cables provide a bus enabled RS485 plug for the MP^{2}I jack of the CPU and a RS232 res. USB plug for the PC.

Due to the RS485 connection you may plug the MPI programming cables directly to a already plugged MPI plug on the MPI jack. Every bus participant identifies itself at the bus with an unique MPI address, in the course of which the address 0 is reserved for programming devices. The structure of a MPI net is in the principal identical with the structure of a 1.5MBaud Profibus net. I.e. the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and Profibus cables. Your CPU 31xS supports transfer rates of up to 1.5MBaud. Per default the MPI net runs with 187.5kBaud. VIPA CPUs are delivered with MPI address 2.

Terminating resistor

A cable has to be terminated with its surge impedance. For this you switch on the terminating resistor at the first and the last participant of a network or a segment.

Please make sure that the participants with the activated terminating resistors are always provided with voltage during start-up and operation.



Approach transfer via MPI

A maximum of 32 PG/OP connections is supported by MPI. The transfer via MPI takes place with the following proceeding:

- Connect your PC to the MPI respectively MP^2I jack of your CPU via a MPI programming cable.
- Load your project in the SIMATIC Manager from Siemens.
- Choose in the menu **Options** > *Set PG/PC interface*
- Select in the according list the "PC Adapter (MPI)"; if appropriate you have to add it first, then click on [Properties].
- Set in the register *MPI* the transfer parameters of your MPI net and type a valid *address*.
- Switch to the register *Local connection*
- Set the COM port of the PCs and the transfer rate 38400Baud for the MPI programming cable from VIPA.
- Via **PLC** > *Load to module* you may transfer your project via MPI to the CPU and save it on a MMC via **PLC** > *Copy RAM to ROM* if one is plugged.

Approach transfer via Profibus

Transfer via Profibus is only available by DP master, if projected as master and assigned with a Profibus address before. A maximum of 31 PG/OP connections is supported by Profibus. The transfer via MPI takes place with the following proceeding:

- Connect your PC to the Profibus master jack of your CPU via a MPI programming cable.
- Load your project in the Siemens SIMATIC Manager.
- Choose in the menu **Options** > *Set PG/PC interface*
- Select in the according list the "PC Adapter (Profibus)"; if appropriate you have to add it first, then click on [Properties].
- Set in the register *Profibus* the transfer parameters of your Profibus net and type a valid *Profibus address*. The *Profibus address* must be assigned to the DP master by a project before.
- Switch to the register *Local connection*
- Set the COM port of the PCs and the transfer rate 38400Baud for the MPI programming cable from VIPA.
- Via **PLC** > *Load to module* you may transfer your project via Profibus to the CPU and save it on a MMC via **PLC** > *Copy RAM to ROM* if one is plugged.

Transfer with Green Cable (possible only at MP²I jack)

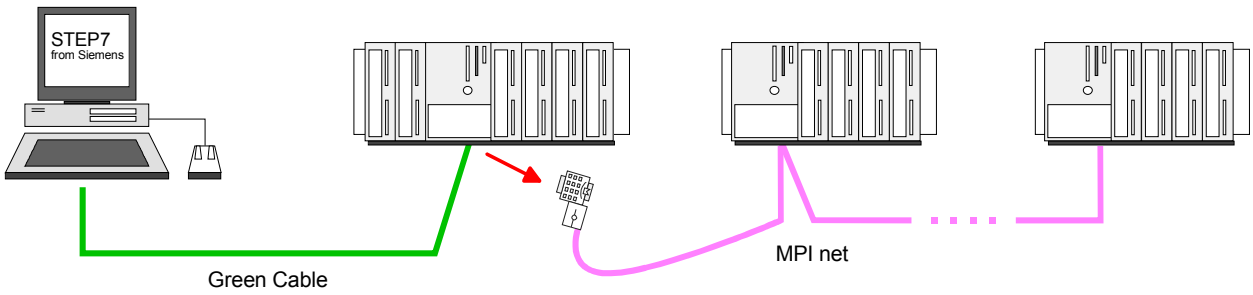
The "Green Cable" is a programming and download cable that may exclusively be plugged directly to VIPA components with MP²I jack. The usage at a "normal" MPI jack is not possible. By plugging the Green Cable to a MP²I jack you may establish a serial connection between the RS232 interface of your PC and the MP²I interface of your CPU.

Attention!

Please regard that you may plug the "Green Cable" exclusively directly and only to a **MP²I interface** of VIPA-CPUs!

Approach

- Connect the RS232 interface of the PC and the MP²I interface of the CPU with the Green Cable.
- Load your project in the SIMATIC Manager from Siemens.
- Choose in the menu **Options** > *Set PG/PC interface*
- Select in the according list the "PC Adapter (MPI)"; if appropriate you have to add it first, then click on [Properties].
- Switch to the register *Local connection*
- Set the COM port of the PCs and the transfer rate 38400Baud for the MPI programming cable from VIPA. The settings in the register *MPI* are ignored at the usage of the Green Cable.
- Via **PLC** > *Load to module* you may transfer your project to the CPU and save it on a MMC via **PLC** > *Copy RAM to ROM* if one is plugged.



Transfer via MMC

As external storage medium a MMC is deployed.

The MMC (**Me**m**o**ry **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 take care that your current project engineering is stored in the root directory. For reading from the MMC the following file names must be used:

- Read after overall reset:
  S7PROG.WLD (S7 project file)
  PROTECT.WLD (Extended Know-how protection)
- Read after PowerON: **AUTOLOAD.WLD** (S7 project file)

Transfer CPU → MMC

When the MMC has been installed, 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 "MCC"-LED of the CPU is blinking. When the LED expires the write process is finished.

Process control

After a write process on 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. At a successful write process the diagnostic buffer contains 0xE200.

When writing on the MMC, 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

Transfer MMC → CPU

The transfer of the application program from the MMC into the CPU takes always place after an overall reset. The blinking of the LED "MCC" of the CPU marks the active transfer.

An overall reset of the CPU takes place if the MMC does not contain a valid application program or if the transfer should fail. The red "STOP"-LED blinks three times.

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.

Transfer via Ethernet PG/OP channel (initialization necessary)

For the on-line access to the Ethernet PG/OP channel you have to assign IP address parameters by means of the "initialization".

After allocation the Ethernet PG/OP canal may be accessed by the IP address parameters.

Initialization

In the following the steps of initialization are described. More information may be found at "initialization" of the PG/OP channel.

- Determine the Ethernet (MAC) address of the Ethernet PG/OP channel. This always may be found as 1st address under the front flap of the CPU on a sticker on the left side.

Ethernet address
1. Ethernet PG/OP
2. CP343 (optional)

- Establish a network connection between Ethernet PG/OP channel of the CPU and PC.
- Set at Siemens SIMATIC manager via **Options** > *Set PG/PC Interface* the access path to "TCP/IP -> Network card .... Protocol RFC 1006".
- Open with **PLC** > *Assign Ethernet Address* the dialog window for "initialization" of a station.
- Determine the CPU components via MAC address and assign it to IP address parameters. As long as the Ethernet PG/OP channel was not initialized yet, this owns the IP address 0.0.0.0 and the station name "Onboard PG/OP".

Transfer

Direct after the assignment the Ethernet PG/OP channel may be reached by the Siemens SIMATIC manager by means of these IP address parameters.

The transfer happens in the following approach:

- Open your project in the Siemens SIMATIC manager.
- Set at Siemens SIMATIC manager via **Options** > *Set PG/PC Interface* the access path to "TCP/IP -> Network card .... Protocol RFC 1006".
- Click to **PLC** > Download → the dialog "Select target module" is opened. Select your target module and enter the IP address parameters of the Ethernet PG/OP channel as address for connection. Provided that no new hardware configuration is transferred to the CPU, the given Ethernet-PG/OP channel is permanently stored in the project as transfer channel.
- With [OK] the transfer is started. System dependent you get a message that the projected system differs from target system. This message may be accepted by [OK] → your project is transferred and may be executed in the CPU after transfer.

Operating modes

Overview

The CPU can be in one of 4 operating modes:

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

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

Operating mode STOP

- The application program is not processed.
- If there has been a processing before, the values of counters, timers, flags and the process image are retained during the transition to the STOP mode.
- 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

Operating mode HOLD

The CPU 31xS gives you the opportunity to define up to 4 breakpoints for program diagnosis. Setting and deletion of breakpoints happens in your programming environment. As soon as a breakpoint is reached, you may process your program step by step and in- and outputs can be activated.

Precondition

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

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

Approach for working with breakpoints

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

Behavior in operating state HOLD

- The LED RUN blinks and the LED STOP is on.
- The execution of the code is stopped. No level is further executed.
- All times are frozen.
- The real-time clock runs on.
- The outputs are closed, but may be released for test purposes.
- Passive CP communication is possible.

Note!

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

With more than 3 breakpoints, a single step execution is not possible.

Function security

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

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

Event	concerns	Effect
RUN → STOP	general	BASP (**B**efehls-**A**usgabe-**Sp**erre, i.e. command output lock) is set.
	central digital outputs	The outputs are 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, then OB 100 is called. After the execution of the OB, the BASP is reset and the cycle starts with: Delete PIO → Read PII → OB 1.
	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 → OB 1 → Write PIO.

PII = Process image inputs
PIO = Process image outputs

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

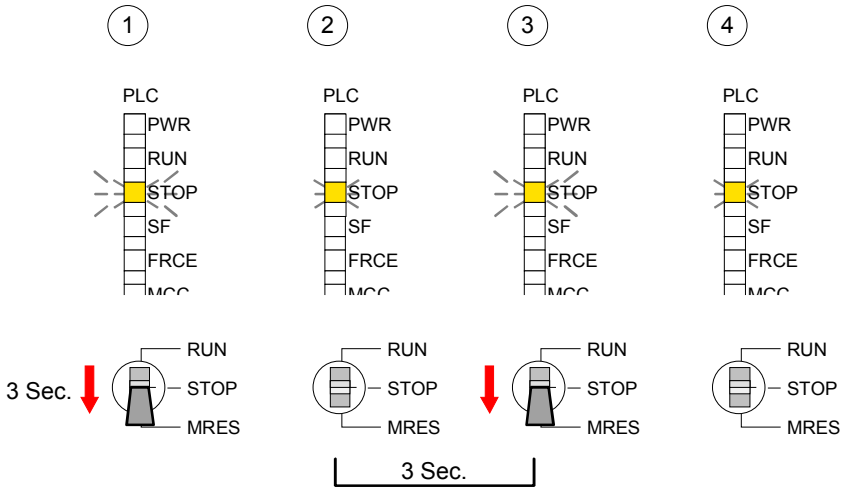
Condition

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

Overall reset

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

The following figure illustrates the above procedure:



Automatic reload

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

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

Overall reset by means of the Siemens SIMATIC Manager

Condition

The operating mode of the CPU must 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** > *Clean/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 STOP-LED blinks during the overall reset procedure.

When the STOP-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 MCC-LED blinks.

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

Set back to factory setting

The following approach deletes the internal RAM of the CPU completely and sets it back to the delivery state.

Please regard that the MPI address is also set back to default 2!

- Push down the reset lever for app. 30 seconds. The ST-LED blinks. After a few seconds the LED turns to static light. Count the number of static light phases because now the LED switches between static light and blinking.
- After the 6th static light you release the reset lever and push it down again shortly. Now the green RUN-LED is on once. This means that the RAM is totally deleted.
- Turn the power supply off and on again.

More information may be found at the part "Factory reset" further below.

Firmware update

Overview

Starting with firmware version 1.0.0 there is the opportunity to execute a firmware update for SPEED-Bus modules and CPU via MMC.

For this an accordingly prepared MMC must be in the CPU during the startup.

So a firmware files can be recognized and assigned with startup, a pkg file name is reserved for each updatable component an hardware release, which begins with "px" and differs in a number with six digits.

The pkg file name of every updateable component may be found at a label right down the front flap of the module.

As soon as with startup a pkg file is on the MMC and the firmware is more current than in the components, all the pkg file assigned components within the CPU and at the SPEED-Bus get the new firmware.

Firmware Package and Version

1. CPU 31xS
2. Profibus DP master
3. CP 343 (optional)

Latest Firmware at ftp.vipa.de

The latest 2 firmware versions are to be find in the service area at www.vipa.de and at the ftp server at ftp.vipa.de/support/firmware.

For example the following files are necessary for the firmware update of the CPU 317-4NE11 and its components (Profibus, Ethernet CP 343) with hardware release 1:

- 317-4NE11, Hardware release 1: Px000035_v142.zip
- Profibus DP-Master (integrated/SPEED-Bus): Px000009_V112.zip
- Ethernet-CP 343 (integrated/SPEED-Bus): Px000005_V179.zip

Attention!

When installing a new firmware you have to be extremely careful. Under certain circumstances you may destroy the CPU, for example if the voltage supply is interrupted during transfer or if the firmware file is defective.

In this case, please call the VIPA-Hotline!

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

Display the Firmware version of the SPEED7 system via Web Site

Every SPEED7-CPU has an integrated website that monitors information about firmware version of the SPEED7 components. The Ethernet PG/OP channel provides the access to this web site.

To activate the PG/OP channel you have to enter according IP parameters.

This can be made in Siemens SIMATIC manager either by a hardware configuration, loaded by MMC respectively MPI or via Ethernet by means of the MAC address with **PLC** > *Assign Ethernet Address*.

After that you may access the PG/OP channel with a web browser via the IP address of the project engineering. More detailed information is to find in the manual of your SPEED7 CPU, chapter "Deployment CPU31xS" at "Access to Ethernet PG/OP channel and website".

Load firmware and transfer it to MMC

- Go to www.vipa.de.
- Click on Service > Download > Firmware Updates.
- Click on "Firmware for System 300S CPUs"
- Choose the according modules (CPU, DPM, CP...) and download the firmware Px.....zip to your PC.
- Extract the zip-file and copy the extracted file to your MMC.
- Following this approach, transfer all wanted firmware files to your MMC.

Preconditions for ftp access

For the display of ftp sites in your web browser you may have to execute the following adjustments:

Internet Explorer

ftp access only with version 5.5 or higher

Options > *Internet options*, Register "Advanced" in the area "Browsing":

- activate: "Enable folder view for ftp sites"
- activate: "Use passive ftp ..."

Netscape

ftp- access only with version 6.0 or higher

No further adjustments are required

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

Attention!

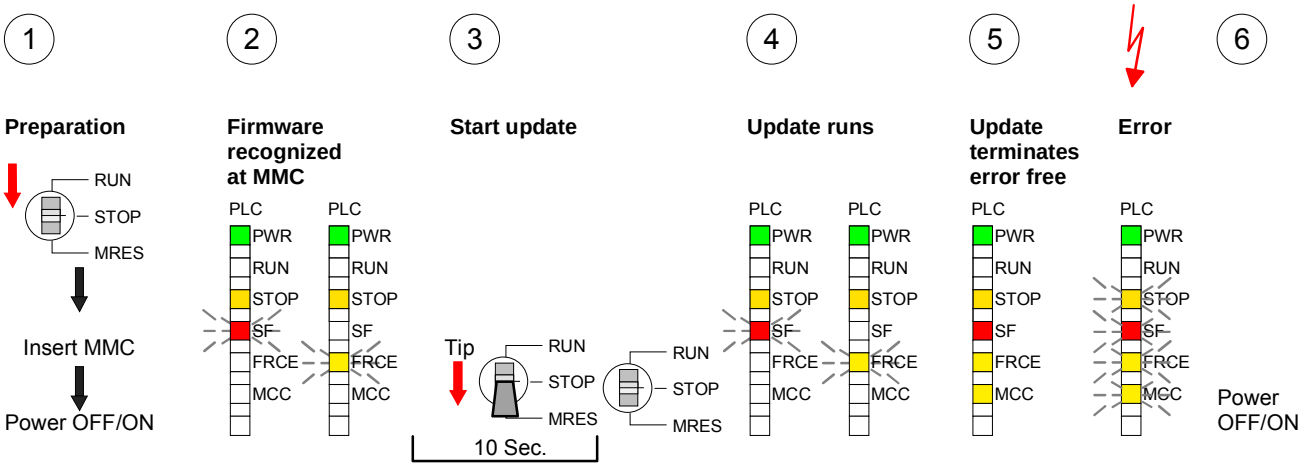
With a firmware update an overall reset is automatically executed. If your program is only available in the load memory of the CPU it is deleted! Save your program before executing a firmware update! After the firmware update you should execute a "Set back to factory settings" (see following page).

Transfer firmware from MMC into CPU

1. Get the RUN-STOP lever of your CPU in position STOP. Turn off the voltage supply. Plug the MMC with the firmware files into the CPU. Please take care of the correct plug-in direction of the MMC. Turn on the voltage supply.
2. After a short boot-up time, the alternate blinking of the LEDs SF and FRCE shows that at least a more current firmware file was found on the MMC.
3. You start the transfer of the firmware as soon as you tip the RUN/STOP lever downwards to MRES within 10s.
4. During the update process, the LEDs SF and FRCE are alternately blinking and MMC LED is on. This may last several minutes.
5. The update is successful finished when the LEDs PWR, STOP, SF, FRCE and MCC are on. If they are blinking fast, an error occurred.
6. Turn Power OFF and ON. Now it is checked by the CPU, whether further current firmware versions are available at the MMC. If so, again the LEDs SF and FRCE flash after a short start-up period. Continue with point 3.

   If the LEDs do not flash, the firmware update is ready.

   Now a *factory reset* should be executed (see next page). After that the CPU is ready for duty.



Factory reset

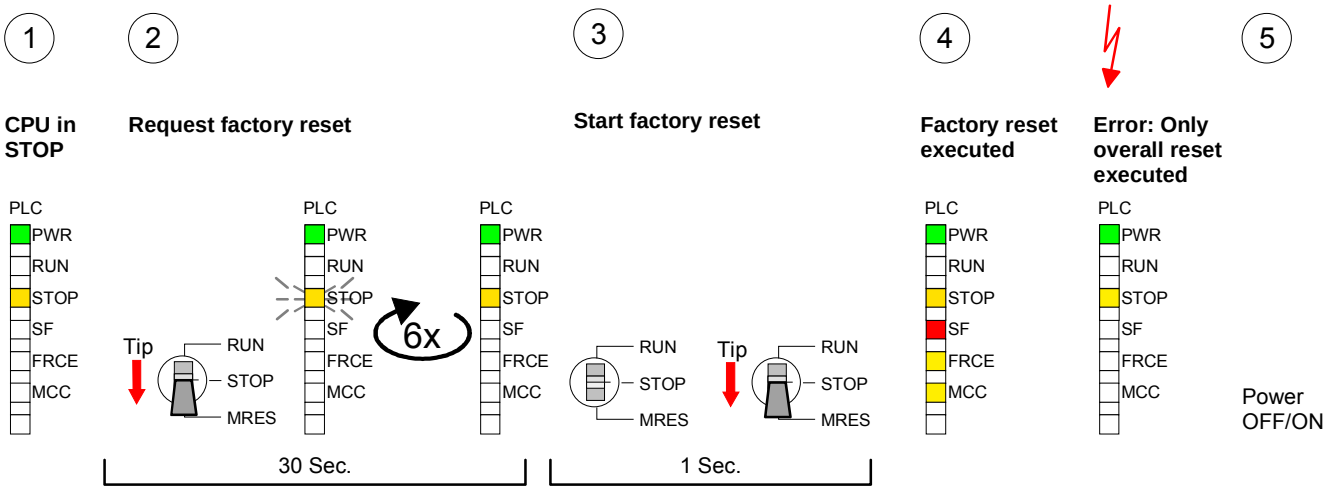
Proceeding

With the following proceeding the internal RAM of the CPU is completely deleted and the CPU is reset to delivery state. Please note that here also the IP address of the Ethernet PG/OP channel is set to 0.0.0.0 and the MPI address is reset to the address 2!

A factory reset may also be executed by the MMC-Cmd FACTORY_RESET. More information may be found at "MMC-Cmd - Auto commands".

1. Switch the CPU to STOP.
2. Push the operating switch down to position MRES for 30s. Here the STOP-LED flashes. After a few seconds the stop LED changes to static light. Now the STOP LED changes between static light and flashing. Starting here count the static light states.
3. After the 6th static light release the operating mode switch and tip it downwards to MRES within 1s.
4. For the confirmation of the resetting procedure the green run LED gets ON within 0.5s. If not the factory reset has failed and only an overall reset was executed. In this case you can repeat the procedure. An factory reset can only be executed if the stop LED has static light for exactly 6 times.
5. After factory reset switch the power supply off and on.

The proceeding is shown in the following Illustration:



Note!

After the firmware update you always should execute a *Factory reset*.

Memory extension with MCC

Overview

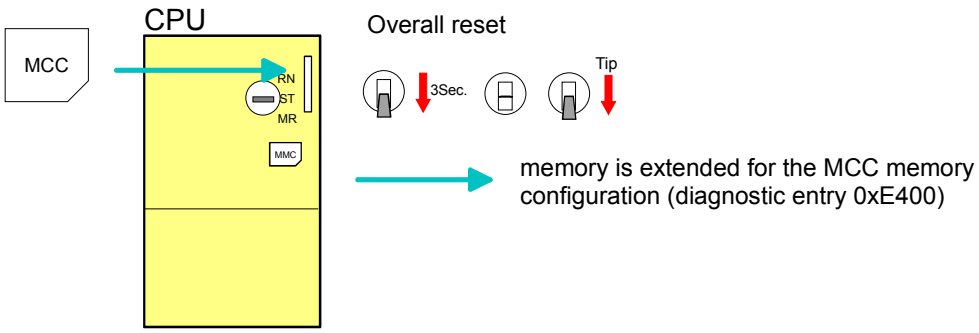
Starting with the CPU firmware version 3.0.0 you have the option to extend the work memory of your CPU.

For this, a MCC memory extension card is available from VIPA. The MCC is a specially prepared MMC (**M**ulti**m**edia **C**ard). By plugging the MCC into the MCC slot and then an overall reset the according memory expansion is released. There may only one memory expansion be activated at the time.

On the MCC there is the file *memory.key*. This file may not be altered or deleted. You may use the MCC also as "normal" MMC for storing your project.

Approach

To extend the memory, plug the MCC into the card slot at the CPU labeled with "MCC" and execute an overall reset.



If the memory expansion on the MCC exceeds the maximum extendable memory range of the CPU, the maximum possible memory of the CPU is automatically used.

You may determine the recent memory extension via the Siemens SIMATIC Manager at *Module Information* - "Memory".

Attention!

Please regard that the MCC must remain plugged when you've executed the memory expansion at the CPU. Otherwise the CPU switches to STOP after 48h. The MCC can not be exchanged with a MCC of the same memory configuration.

Behavior

When the MCC memory configuration has been taken over you may find the diagnosis entry 0xE400 in the diagnostic buffer of the CPU.

After pulling the MCC the entry 0xE401 appears in the diagnostic buffer, the SF-LED is on and after 48h the CPU switches to STOP. A reboot is only possible after plugging-in the MCC again or after an overall reset.

After re-plugging the MCC, the SF-LED extinguishes and 0xE400 is entered into the diagnostic buffer.

You may reset the memory configuration of your CPU to the initial status at any time by executing an overall reset without MCC.

Extended know-how protection

Overview

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

Standard protection

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

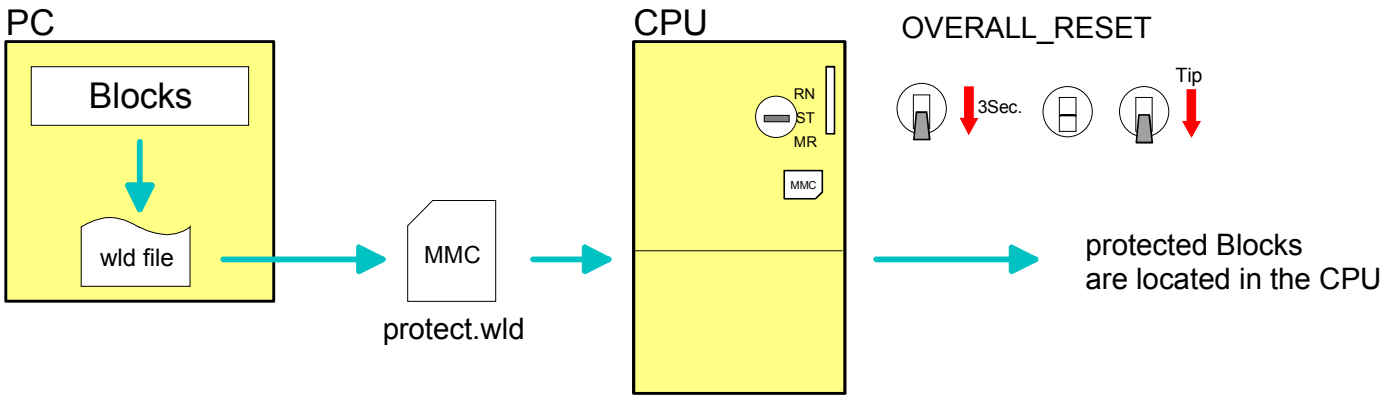
Extended protection

The "extended" know-how protection developed by VIPA offers the opportunity to store blocks permanently in the CPU.

At the "extended" protection you transfer the protected blocks into a WLD-file named protect.wld. By plugging the MMC and following overall reset, the blocks in the protect.wld are permanently stored in the CPU.

You may protect OBs, FBs and FCs.

When back-reading the protected blocks into the PG, exclusively the block header are loaded. The source remains in the CPU and is thus protected for accesses of 3rd persons.



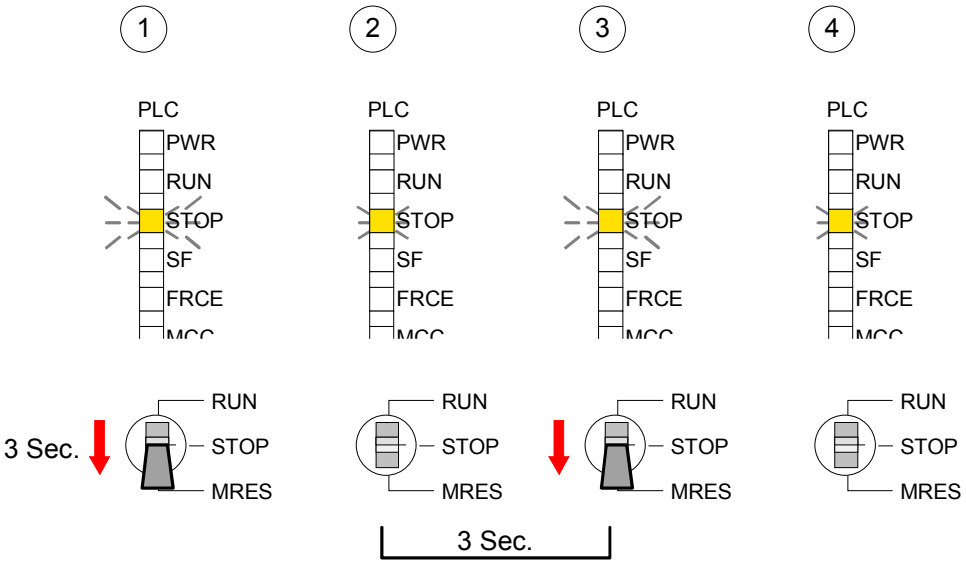
protect blocks with protect.wld

Create a new wld-file in your project engineering tool with **File** > *Memory Card file* > *New* and rename it to "protect.wld".

Transfer the according blocks into the file by dragging them with the mouse from the project to the file window of protect.wld.

Transfer protect.wld to CPU with overall reset

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



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

Protection behavior

Protected blocks are overwritten by a new protect.wld.

Using a PG 3rd persons may access protected blocks but only the block header is transferred to the PG. The block code that is to protect remains in the CPU and can not be read.

Change respectively delete protected blocks

Protected blocks in the RAM of the CPU may be substituted at any time by blocks with the same name. This change remains up to next overall reset. Protected blocks may permanently be overwritten only if these are deleted at the protect.wld before.

By transferring an empty protect.wld from the MMC you may delete all protected blocks in the CPU.

Usage of protected blocks

Due to the fact that reading of a "protected" block from the CPU monitors no symbol labels it is convenient to provide the "block covers" for the end user.

For this, create a project out of all protected blocks. Delete all networks in the blocks so that these only contain the variable definitions in the according symbolism.

MMC-Cmd - Auto commands

Overview

Since firmware version 3.0.8 a *command file at a MMC may be started a*utomatically when the MMC is stuck and the CPU is in STOP. As soon as the MMC is stuck the command file is once executed at CPU STOP up to the next Power ON.

The command file is a text file which consists of a command sequence to be stored as ***vipa_cmd.mmc*** in the root directory of the MMC.

The file has to be started by *CMD_START* as 1st command, followed by the desired commands (no other text) und must be finished by CMD_END as last command.

Text after the last command *CMD_END* e.g. comments is permissible, because this is ignored. As soon as the command file is recognized and executed each action is stored at the MMC in the log file logfile.txt. In addition for each executed command a diagnostics entry may be found in the diagnostics buffer.

Commands

In the following there is an overview of the commands. Please regard the command sequence is to be started with *CMD_START* and ended with CMD_END.

Command	Description	Diagnostics entry
CMD_START	In the first line *CMD_START* is to be located.	0xE801
	There is a diagnostic entry if *CMD_START* is missing	0xE8FE
WAIT1SECOND	Waits ca. 1 second.	0xE803
WEBPAGE	The current web page of the CPU is stored at the MMC as "webpage.htm".	0xE804
LOAD_PROJECT	The function "Overall reset and reload from MMC" is executed. The wld file located after the command is loaded else "s7prog.wld" is loaded.	0xE805
SAVE_PROJECT	The recent project (blocks and hardware configuration) is stored as "s7prog.wld" at the MMC. If the file just exists it is renamed to "s7prog.old".	0xE806
FACTORY_RESET	Executes "factory reset".	0xE807
DIAGBUF	The current diagnostics buffer of the CPU is stored as "diagbuff.txt" at the MMC.	0xE80B
SET_NETWORK	IP parameters for Ethernet PG/OP channel may be set by means of this command. The IP parameters are to be given in the order IP address, subnet mask and gateway in the format xxx.xxx.xxx.xxx each aparted by a comma. Enter the IP address if there is no gateway used.	0xE80E
CMD_END	In the last line *CMD_END* is to be located.	0xE802

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

Example 1

CMD_START	Marks the start of the command sequence (0xE801)
LOAD_PROJECT proj.wld	Execute an overall reset and load "proj.wld" (0xE805)
WAIT1SECOND	Wait ca. 1s (0xE803)
WEBPAGE	Store web page as "webpage.htm" (0xE804)
DIAGBUF	Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)
CMD_END	Marks the end of the command sequence (0xE802)
... arbitrary text ...	Text after the command CMD_END is not evaluated.

Example 2

CMD_START	Marks the start of the command sequence (0xE801)
LOAD_PROJECT proj2.wld	Execute an overall reset and load "proj2.wld" (0xE805)
WAIT1SECOND	Wait ca. 1s (0xE803)
WAIT1SECOND	Wait ca. 1s (0xE803)
SET_NETWORK172.16.129.210,255.255.224.0,172.16.129.210	IP parameter (0xE80E)
WAIT1SECOND	Wait ca. 1s (0xE803)
WAIT1SECOND	Wait ca. 1s (0xE803)
WEBPAGE	Store web page as "webpage.htm" (0xE804)
DIAGBUF	Store diagnostics buffer of the CPU as "diagbuff.txt" (0xE80B)
CMD_END	Marks the end of the command sequence (0xE802)
... arbitrary text ...	Text after the command CMD_END is not evaluated.

Note!

The parameters IP address, subnet mask and gateway may be received from the system administrator.

Enter the IP address if there is no gateway used.

VIPA specific diagnostic entries

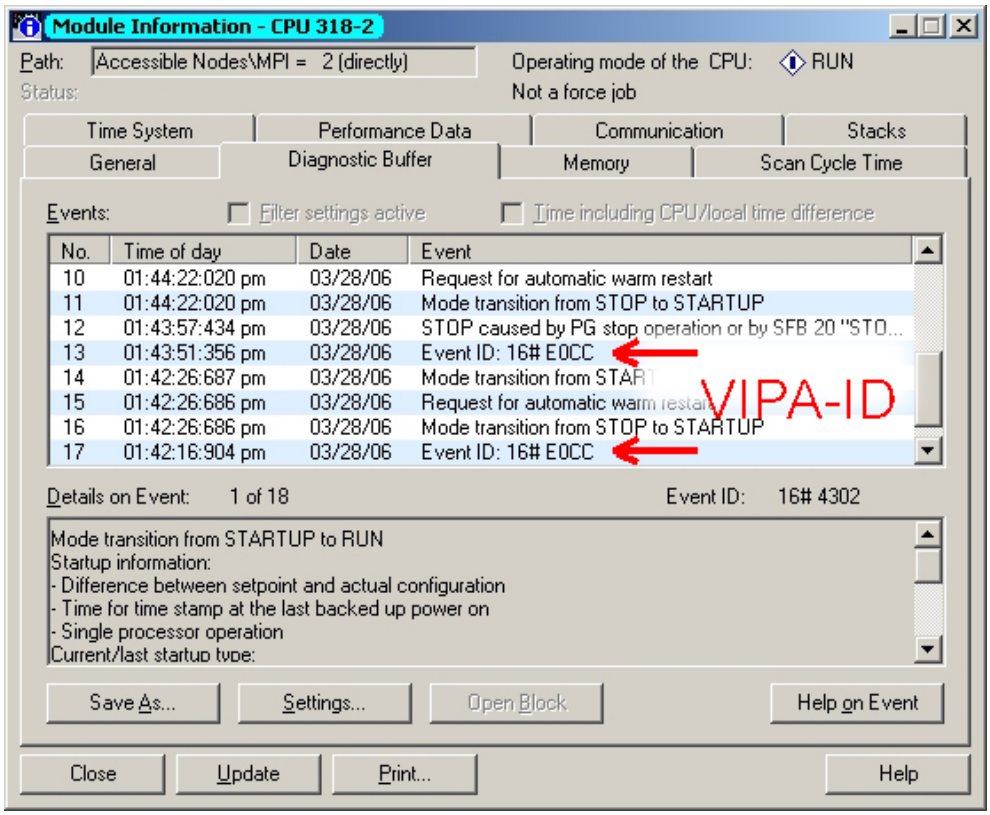
Entries in the diagnostic buffer

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

The current content of the diagnostics buffer is stored on MMC by means of the MMC-Cmd DIAGBUF. More information may be found at "MMC-Cmd - Auto commands".

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 Zinfo1: I/O address Zinfo2: Slot
0xE004	Multiple parameterization of a I/O address Zinfo1: I/O address Zinfo2: Slot
0xE005	Internal error – Please contact the VIPA-Hotline!
0xE006	Internal error – Please contact the VIPA-Hotline!
0xE007	Configured in-/output bytes do not fit into 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 Zinfo2: Slot 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 Zinfo2: Slot of the master (32=page frame master)
0xE016	Maximum block size at master transfer exceeded Zinfo1: I/O address 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)
0xE0B0	Speed7 is not stoppable (probably undefined BCD value at timer)
0xE0C0	Not enough space in work memory for storing code block (block size exceeded)
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 Ram2Rom)
0xE210	MMC reading finished (reload after overall reset)

continued ...

... continue

Event-ID	Description
0xE400	Memory expansion MCC has been plugged
0xE401	Memory expansion MCC has been removed
0xE801	MMC-Cmd: CMD_START recognized and successfully executed
0xE802	MMC-Cmd: CMD_END recognized and successfully executed
0xE803	MMC-Cmd: WAIT1SECOND recognized and successfully executed
0xE804	MMC-Cmd: WEBPAGE recognized and successfully executed
0xE805	MMC-Cmd: LOAD_PROJECT recognized and successfully executed
0xE806	MMC-Cmd: SAVE_ PROJECT recognized and successfully executed
0xE807	MMC-Cmd: FACTORY_RESET recognized and successfully executed
0xE80B	MMC-Cmd: DIAGBUF recognized and successfully executed
0xE80E	MMC-Cmd: SET_NETWORK recognized and successfully executed
0xE8FB	MMC-Cmd: Error: Initialization of the Ethernet PG/OP channel by means of SET_NETWORK is faulty.
0xE8FC	MMC-Cmd: Error: Not every IP-Parameter is set at SET_NETWORK.
0xE8FE	MMC-Cmd: Error: CMD_START was not found
0xE8FF	MMC-Cmd: Error: Reading the CMD file is faulty (MMC error)
0xE901	Check sum error
0xEA00	Internal error – Please contact the VIPA-Hotline!
0xEA01	Internal error – Please contact the VIPA-Hotline!
0xEA04	SBUS: Multiple parameterization of a I/O address Zinfo1: I/O address Zinfo2: Slot Zinfo3: Data width
0xEA05	Internal error – Please contact the VIPA-Hotline!
0xEA07	Internal error – Please contact the VIPA-Hotline!
0xEA08	SBUS: Parameterized input data width unequal to plugged input data width Zinfo1: Parameterized input data width Zinfo2: Slot Zinfo3: Input data width of the plugged module
0xEA09	SBUS: Parameterized output data width unequal to plugged output data width Zinfo1: Parameterized output data width Zinfo2: Slot Zinfo3: Output data width of the plugged module
0xEA10	SBUS: Input address outside input area Zinfo1: I/O address Zinfo2: Slot Zinfo3: Data width

continued ...

... continue

Event-ID	Description
0xEA11	SBUS: Output address outside output area Zinfo1: I/O address Zinfo2: Slot Zinfo3: Data width
0xEA12	SBUS: Error at writing record set Zinfo1: Slot Zinfo2: Record set number Zinfo3: Record set length
0xEA14	SBUS: Multiple parameterization of a I/O address (Diagnostic address) Zinfo1: I/O address Zinfo2: Slot Zinfo3: Data width
0xEA15	Internal error - Please contact the VIPA-Hotline!
0xEA18	SBUS: Error at mapping of the master I/O devices Zinfo2: Master slot
0xEA19	Internal error - Please contact the VIPA-Hotline!
0xEA98	Timeout at waiting for reboot of a SBUS module (Server)
0xEA99	Error at file reading via SBUS
0xEE00	Internal error - Please contact the VIPA-Hotline!

Using test functions for control and monitoring of 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 PLC must be in RUN mode!

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

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

- the result of the logical operation 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.

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:

E, A, M, T, Z and D.

The process image of binary and digital operands is modified independently of the operating mode of the CPU 31xS.

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 5 Deployment I/O periphery CPU 314ST

Outline

This chapter contains all information necessary for the employment of the in-/output periphery of the CPU 314ST. It describes functionality, project engineering and diagnostic of the analog and digital part.

The following text describes:

- Overview over the I/O ranges
- Employment of the analog part
- Employment of the digital part and the counter functions

Content

Overview

General

At the CPU 314ST the analog and digital in-/output channels are together in a 2tier casing.

The following components are integrated:

- Analog input: 4xU/Ix12Bit, 1xPt100
- Analog output: 2xU/Ix12Bit
- Digital input: 16(8)xDC24V with parameterizable counter function
- Digital output: 0(8)xDC24V 1A
- Counter: max. 4 counter with the operating mode endless, single or periodic count

Project engineering

The project engineering takes place in the Siemens SIMATIC manager. For this the import of the GSD speedbus.gsd is required. After the installation of the GSD you'll find the CPU in the hardware catalog in the directory VIPA_SPEEDbus with the corresponding order no..

Counter

The here used counters are endless counter where the control happens via the digital input channels. For the counter you may configure interrupts that may influence the corresponding digital output channel.

SPEED-Bus

The SPEED-Bus is a 32Bit parallel bus developed by VIPA with a max. data rate of 40MByte/s. Via the SPEED-Bus you have the opportunity to connect up to 16 SPEED-Bus modules to your CPU 31xS.

In opposite to the "standard" backplane bus where the modules are plugged at the right side of the CPU via single bus connectors, the SPEED-Bus manages the connection via a special SPEED-Bus rail at the left side of the CPU.

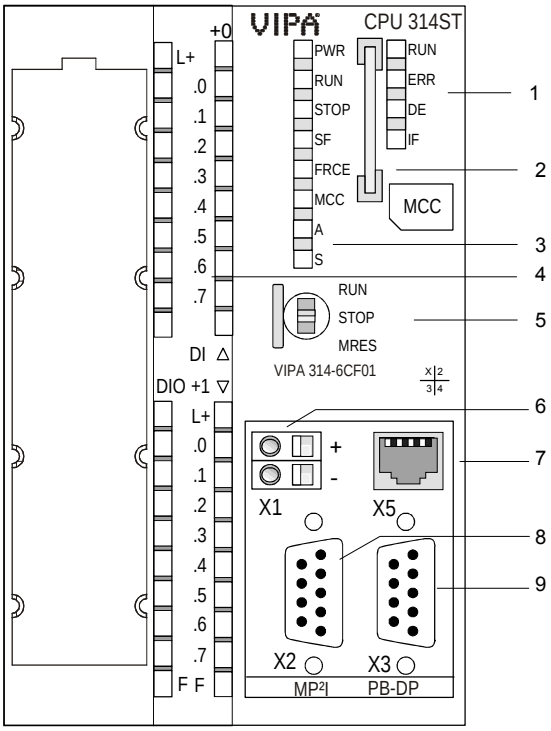
You can order profile rails at VIPA with integrated SPEED-Bus for 2, 6, 10 or 16 SPEED-Bus periphery modules in different lengths.

Ordering Data

Type	Order number	Description
314ST/DPM	VIPA 314-6CF01	MP^2I interface, MMC slot, real time clock, Ethernet Interface for PG/OP, Profibus DP master, SPEED-Bus, DI 8...16xDC24V / DO 8...0xDC24V, 0.5A, AI 4x12Bit / AO 2x12Bit / AI 1xPt100, 4 counter

In-/Output range

Construction The picture shows the CPU 314ST with open flap.



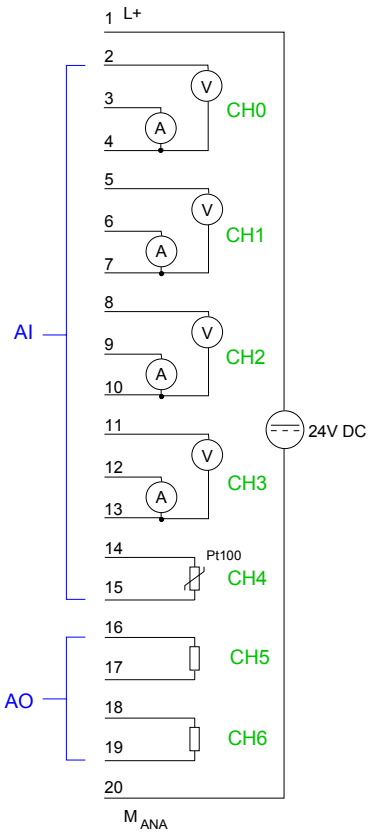
[1] LEDs of the integrated Profibus DP master (only 314-6CF01)
[2] MMC slot
[3] LEDs of the CPU part
[4] LEDs of the I/O part
[5] operating mode Switch CPU

The following components are under the front flap

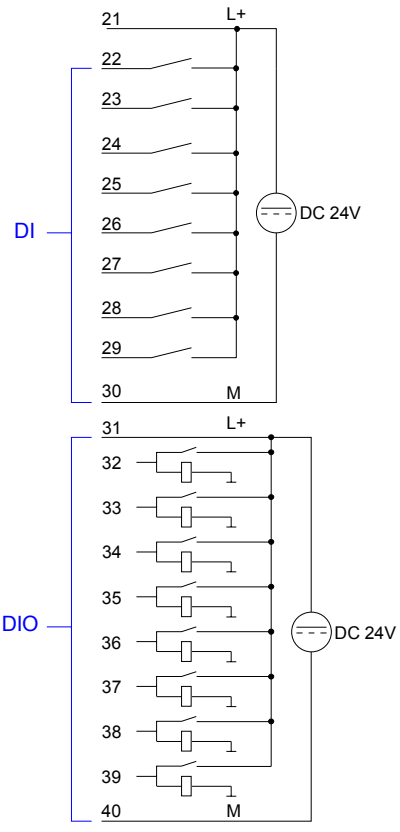
[6] Interface for DC 24V power supply
[7] Ethernet interface PG/OP
[8] MP²I interface
[9] RS485 Profibus DP/ PtP interface

Pin assignment:

Analog part



Digital part



Address assignment

By including the GSD speedbus.gsd in your hardware configurator the module is at your disposal in the hardware catalog.

After the installation of the GSD you'll find the CPU 314ST under *Additional field devices \ I/O \ VIPA_SpeedBus*.

In case there is no hardware configuration available, the in- and output areas starting at address 1024 are shown in the address range of the CPU.

For the data input a range of 48Byte and for the data output a range of 24Byte is available:

Input range

Address	Access	Assignment
+0	Byte	Digital Input I+0.0 ... I+0.7
+1	Byte	Digital Input I+1.0 ... I+1.7
+2	Word	reserved
+4	Word	Analog Input CH0
+6	Word	Analog Input CH1
+8	Word	Analog Input CH2
+10	Word	Analog Input CH3
+12	Word	Analog Input CH4
+14	Word	reserved
+16	Double word	Counter 0 / Latch 0
+20	Word	reserved
+22	Word	Status Counter 0
+24	Double word	Counter 1 / Latch 1
+28	Word	reserved
+30	Word	Status Counter 1
+32	Double word	Counter 2 / Latch 2
+36	Word	reserved
+38	Word	Status Counter 2
+40	Double word	Counter 3 / Latch 3
+44	Word	reserved
+46	Word	Status Counter 3

Output range

Address	Access	Assignment
+0	Byte	reserved
+1	Byte	Digital Output Q+1.0 ... Q+1.7
+2	Word	reserved
+4	Word	Analog Output CH0
+6	Word	Analog Output CH1
+8	Word	reserved
+10	Word	Status Counter 0
+12	Word	reserved
+14	Word	Status Counter 1
+16	Word	reserved
+18	Word	Status Counter 2
+20	Word	reserved
+22	Word	Status Counter 3

Analog part

Overview

The analog part consists of 4 input, 1 Pt100 and 2 output channels. 10Byte input and 4Byte output data of the process image are used by the analog part.

The channels of the module are galvanically separated from the SPEED-Bus via DC/DC transducer and opto couplers.

Attention!

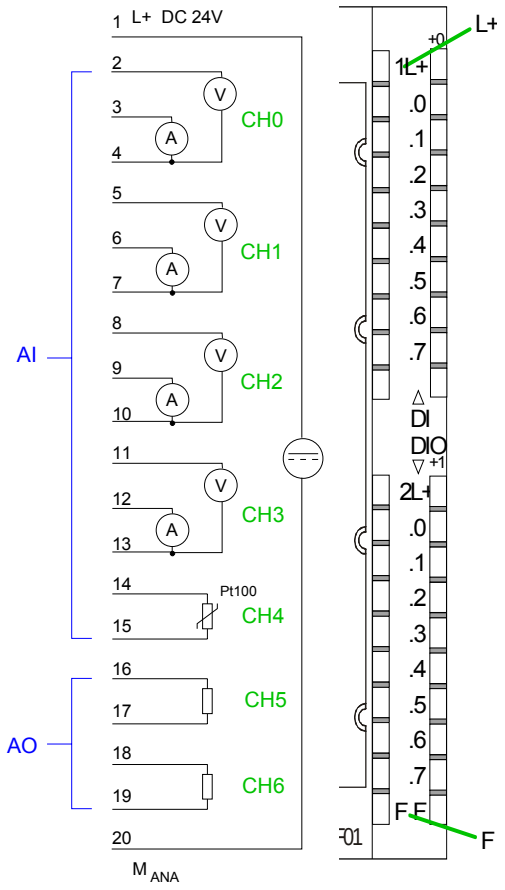
Temporarily not used analog inputs with activated channel must be connected to the concerning ground.

Status indicator

Pin assignment

Pin	Assignment
1	Power supply DC 24V for analog range
2	Meas. voltage channel 0
3	Meas. current channel 0
4	Ground channel 0
5	Meas. voltage channel 1
6	Meas. current channel 1
7	Ground channel 1
8	Meas. voltage channel 2
9	Meas. current channel 2
10	Ground channel 2
11	Meas. voltage channel 3
12	Meas. current channel 3
13	Ground channel 3
14	Pt 100 channel 4
15	Pt 100 channel 4
16	Output + channel 5
17	Ground output channel 5
18	Output + channel 6
19	Ground output channel 6
20	Ground power supply for analog range

Connection



LEDs

1L+	LED (green) Supply voltage available
F	LED (red) Sum error

Note!

To avoid measuring errors, you should connect only one measuring type per channel.

Access to the Analog part

By including the GSD speedbus.gsd into your hardware configurator the module is available at the hardware catalog.

You can find the CPU 314ST after GSD installation at *Additional filed devices \ I/O \ VIPA_SpeedBus.*

The CPU 314ST creates in the peripheral area 48Byte for data input and 24Byte for data output. Here the analog part occupies 10Byte for analog input and 4Byte for analog output. Without a hardware configuration the ranges start at address 1024.
In the following table the according areas are marked ▭ :

Input range

During the measurement, for every channel the measuring data is stored as word in the data input range.

Address	Access	Assignment
+0	Byte	Digital Input I+0.0 ... I+0.7
+1	Byte	Digital Input I+1.0 ... I+1.7
+2	Word	reserved
+4	Word	Analog Input CH0
+6	Word	Analog Input CH1
+8	Word	Analog Input CH2
+10	Word	Analog Input CH3
+12	Word	Analog Input CH4
+14	Word	reserved
+16	Double word	Counter 0 / Latch 0
+20	Word	reserved
+22	Word	Status Counter 0
+24	Double word	Counter 1 / Latch 1
+28	Word	reserved
+30	Word	Status Counter 1
+32	Double word	Counter 2 / Latch 2
+36	Word	reserved
+38	Word	Status Counter 2
+40	Double word	Counter 3 / Latch 3
+44	Word	reserved
+46	Word	Status Counter 3

Output range

For the output you enter a value as word into the data output range.

Address	Access	Assignment
+0	Byte	reserved
+1	Byte	Digital Output Q+1.0 ... Q+1.7
+2	Word	reserved
+4	Word	Analog Output CH0
+6	Word	Analog Output CH1
+8	Word	reserved
+10	Word	Status Counter 0
+12	Word	reserved
+14	Word	Status Counter 1
+16	Word	reserved
+18	Word	Status Counter 2
+20	Word	reserved
+22	Word	Status Counter 3

Numeric notation in Siemens S7 format

The analog values are represented in two's complement format.

Depending on the parameterized transformation speed the lowest value bits of the measuring value are irrelevant. With increasing sampling rate, the resolution decreases.

The following table lists the resolution in dependence of the sampling rate.

	Analog value															
	High-Byte								Low-Byte							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Resolution	sign	Measuring value														
15 Bit + sign	sign	Relevant output value (at 3.7 ... 30Hz)														
14 Bit + sign	sign	Relevant output value (at 60Hz)														X*
13 Bit + sign	sign	Relevant output value (at 120Hz)													X	X
11 Bit + sign	sign	Relevant output value (at 170Hz)											X	X	X	X
9 Bit + sign	sign	Relevant output value (at 200Hz)									X	X	X	X	X	X

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

Algebraic sign bit (sign)

Bit 15 serves as algebraic sign bit. Here is:

Bit 15 = "0" → positive value

Bit 15 = "1" → negative value

Behavior at errors

As soon as a measuring value exceeds the overdrive res. underdrive region, the following value is returned:

Measuring value > Overdrive region: 32767 (7FFFh)

Measuring value < Underdrive region: -32768 (8000h)

At wire break, parameterization error or de-activated analog part the measuring value 32767 (7FFFh) is returned.

Analog part deactivated

With this record set 9Eh you may de-activate the digital res. analog part. Please regard that in spite of the de-activation of the digital res. analog part the process image for both components remains reserved.

The record set has the following structure:

Byte	Bit 15 ... 0
0...1	*Bit 15 ... 0: Module selection* 0000h = Digital- / Analog part activated (default) 0001h = Digital part de-activated 0002h = Analog part de-activated

For detailed information see "Counter parameterization" below.

Digital/Analog conversion

In the following all measuring ranges are listed that are supported by the analog part.

The here listed formulas allow you to transform an evaluated measuring value (digital value) to a value assigned to the measuring range and vice versa.

+/- 10V

Voltage	Decimal	Hex
-10V	-27648	9400
-5V	-13824	CA00
0V	0	0
+5V	13824	3600
+10V	+27648	6C00

Formulas for the calculation:

$$Value = 27648 \cdot \frac{U}{10}, \quad U = Value \cdot \frac{10}{27648}$$

U: voltage, Value: decimal value

0...10V

Voltage	Decimal	Hex
0V	0	0
5V	13824	3600
10V	27648	6C00

Formulas for the calculation:

$$Value = 27648 \cdot \frac{U}{10}, \quad U = Value \cdot \frac{10}{27648}$$

U: voltage, Value: decimal value

0....20mA

Current	Decimal	Hex
0mA	0	0
+10mA	+13824	3600
+20mA	+27648	6C00

Formulas for the calculation:

$$Value = 27648 \cdot \frac{I}{20}, \quad I = Value \cdot \frac{20}{27648}$$

I: current, Value: decimal value

4....20mA

Current	Decimal	Hex
+4mA	0	0
+12mA	+13824	3600
+20mA	+27648	6C00

Formulas for the calculation:

$$Value = 27648 \cdot \frac{I-4}{16}, \quad I = Value \cdot \frac{16}{27648} + 4$$

I: current, Value: decimal value

+/- 20mA

Current	Decimal	Hex
-20mA	-27648	9400
-10mA	-13824	CA00
0mA	0	0
+10mA	+13824	3600
+20mA	+27648	6C00

Formulas for the calculation:

$$Value = 27648 \cdot \frac{I}{20}, \quad I = Value \cdot \frac{20}{27648}$$

I: current, Value: decimal value

Analog Part - Parameterization

Parameter data

18Byte of parameter data are available for the configuration. By using the record set B4h of the SFC 55 "WR_PARM" you may alter the parameterization in the module during runtime. The time needed until the new parameterization is valid can last up to 50ms. During this time, the measuring value output is 7FFFFh. The following table shows the structure of the parameter data:

Record set B4h

Byte	Bit 7 ... 0	Default
0	Channel 0: Wire break recognition Bit 0: 0 = off (Wire break recognition deactivated) 1 = on (Wire break recognition activated) Channel 1: Wire break recognition Bit 1: 0 = off (Wire break recognition deactivated) 1 = on (Wire break recognition activated) Channel 2: Wire break recognition Bit 2: 0 = off (Wire break recognition deactivated) 1 = on (Wire break recognition activated) Channel 3: Wire break recognition Bit 3: 0 = off (Wire break recognition deactivated) 1 = on (Wire break recognition activated) Channel 4: Wire break recognition Bit 4: 0 = off (Wire break recognition deactivated) 1 = on (Wire break recognition activated) Bit 7 ... 5: reserved	00h
1	Bit 4 ... 0: reserved Channel 5: Reaction at CPU_STOP Bit 5: 0 = Set replacement value *) 1 = Store last value Channel 6: Reaction at CPU_STOP Bit 6: 0 = Set replacement value *) 1 = Store last value Bit 7: reserved	00h
2	Channel 0: Function (see table input ranges)	19h
3	Channel 1: Function (see table input ranges)	19h
4	Channel 2: Function (see table input ranges)	19h
5	Channel 3: Function (see table input ranges)	19h
6	Channel 4: Function (see table input ranges)	00h
7	Channel 0: Measuring cycle (see table next page)	00h
8	Channel 1: Measuring cycle (see table next page)	00h
9	Channel 2: Measuring cycle (see table next page)	00h
10	Channel 3: Measuring cycle (see table next page)	00h
11	Channel 4: Measuring cycle (see table next page)	00h
12	Channel 5: Function (see table output ranges)	19h
13	Channel 6: Function (see table output ranges)	19h
14	Channel 5: High-Byte substitute value	00h
15	Channel 5: Low-Byte substitute value	00h
16	Channel 6: High-Byte substitute value	00h
17	Channel 6: Low-Byte substitute value	00h

*) If you want to get 0A res. 0V as output value at CPU-STOP, you have to set the replacement value E500h.

Parameters

Wire break recognition

The Bits 0 ... 4 of Byte 0 allow you to activate the wire break recognition for the input channels. The wire break recognition is only available for the current measuring range of 4...20mA and thermo resistance measurement. A wire break is recognized, a diagnostic entry is made and displayed by the SF LED when the current during current measuring (4 ... 20mA) sinks under 1.18mA res. at thermo resistance measurement the resistance becomes endless. If additionally a diagnostic interrupt is activated, a diagnostic message is sent to the superordinated system.

Diagnostic interrupt

The diagnostic interrupt is global released for the digital and analog part. More is to be find at "Counter - Parameterization". In case of an error like e.g. wire break, the superordinated system receives *record set 0*. For a channel specific diagnostic you may then call *record set 1* (see "Diagnostic data").

CPU-Stop reaction and substitute value

With Bit 5 and 6 of Byte 1 and Byte 14 ... 17 you may set the reaction of the module at CPU-Stop for every output channel.

Via Byte 14 ... 17 you predefine a substitute value for the output channel as soon as the CPU switches to Stop.

By setting Bit 5 res. 6, the last output value remains in the output at CPU-Stop. A reset sets the replacement value.

Function No.

Here you set the function no. of your measuring res. output function for every channel. Please see the according table above.

Measuring cycle

Here you may set the transducer velocity for every input channel. Please regard that a higher transducer velocity causes a lower resolution because of the lower integration time.

The data transfer format remains unchanged. Only the lower Bits (LSBs) are not longer relevant for the analog value.

Structure Measuring cycle Byte:

Byte	Bit 7 ... 0	Resolution	Default
7 ... 11	Bit 3 ... 0: Velocity per channel		00h
	0000 15 conversions/s	16	
	0001 30 conversions/s	16	
	0010 60 conversions/s	15	
	0011 120 conversions/s	14	
	0100 170 conversions/s	12	
	0101 200 conversions/s	10	
	0110 3.7 conversions/s	16	
	0111 7.5 conversions/s	16	
	Bit 7 ... 4: reserved		

Function no. assignment

The assignment of a function no. to a certain channel happens during parameterization. By setting 00h you may de-activate the according channel.

Input range (channel 0 ... 3)

No.	Function	Input range
19h	Voltage ±10V Siemens S7-format	±11.76V 11.76V= End overdrive region (32511) -10V...10V = nominal range (-27648...27648) -11.76 = End underdrive region (-32512) two's complement
18h	Voltage 0...10V Siemens S7-format	0...11.76V 11.76V = End overdrive region (32511) 0...10V = nominal range (0...27648) no underdrive region available
24h	Current ±20mA Siemens S7-format	±23.52mA 23.52mA = End overdrive region (32511) -20...20mA = nominal range (-27648...27648) -23.52mA = End underdrive region (-32512) two's complement
23h	Current 4...20mA Siemens S7-format	1.185...22.81mA 22.81mA = End overdrive region (32511) 4...20mA = nominal range (0...27648) 1.185mA = End underdrive region (-4864) two's complement
22h	Current 0...20mA Siemens S7-format	0...23.52mA 23.52mA = End overdrive region (32511) 0...20mA = nominal range (0...27648) no underdrive region available
00h	Channel not active (turned off)	

Input range (channel 4)

No.	Function	Measuring range / representation
82h	Pt100 in 2wire mode	-240...1000°C 1000°C = End overdrive region (10000) -200...+850°C = nominal range (-2000...8500) -240°C = End underdrive region (-2400) two's complement
85h	Pt1000 in 2wire mode	-240...600°C 600°C = End overdrive region (6000) -200...+500°C = nominal range (-2000...5000) -240°C = underdrive region (-2400) two's complement
83h	NI100 in 2wire mode	-105...295°C 295°C = Ende overdrive region (2950) -50...+250°C = nominal range (-500...2500) -105°C = Ende underdrive region (-1050) two's complement
86h	NI1000 in 2wire mode	-105...270°C 270°C = Ende overdrive region (2700) -50...+250°C = nominal range (-500...2500) -105 = Ende underdrive region (-1050) two's complement
46h	Resistance measurement 600Ohm 2wire	0...705.5Ω 705.5Ω = End overdrive region (32511) 0…600Ω = nominal range (0...27648) no underdrive region available
00h	Channel not active (turned off)	

Output range
(channel 5, channel 6)

No.	Function	Output range
19h	Voltage ±10V Siemens S7-format	±11.76V 11.76V= End overdrive region (32511) -10V...10V = nominal range (-27648...27648) -11.76 = End underdrive region (-32512) two's complement
18h	Voltage 0...10V Siemens S7-format	0...11.76V 11.76V = End overdrive region (32511) 0...10V = nominal range (0...27648) no underdrive region
24h	Current ±20mA Siemens S7-format	±23.52mA 23.52mA = End overdrive region (32511) -20...20mA = nominal range (-27648...27648) -23.52mA = End underdrive region (-32512) two's complement
23h	Current 4...20mA Siemens S7-format	0...22.81mA 22.81mA = End overdrive region (32511) 4...20mA = nominal range (0...27648) 0mA = End underdrive region (-6912) two's complement
22h	Current 0...20mA Siemens S7-format	0...23.52mA 23.52mA = End overdrive region (32511) 0...20mA = nominal range (0...27648) no underdrive region
00h	Channel not active (turned off)	

Note!
Leaving the defined range, the output is 0V res. 0A!

Analog part - Diagnostic functions

Overview

As soon as you've activated the diagnostic interrupt release in the parameterization, the following events can release a diagnostic interrupt:

- Wire break
- Parameterization error
- Measuring range overflow
- Measuring range underflow

At accumulated diagnostic the CPU interrupts the user application and branches to the OB 82 for $diagnostic_{incoming}$. This OB allows you with an according programming to monitor detailed diagnostic information via the SFCs 51 or 59 and to react to it. 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 interrupt release is still active. In the following the record sets for $diagnostic_{incoming}$ and $diagnostic_{going}$ are specified:

Record set 0
$Diagnostic_{incoming}$

Byte	Bit 7 ... Bit 0
0	Bit 0: 0 = OK 1 = Module malfunction Bit 1: 0 (fix) Bit 2: External error Bit 3: Channel error present Bit 4: External supply voltage is missing Bit 6, 5: 0 (fix) Bit 7: Wrong parameters in the module
1	Bit 3 ... 0: Module class 0101 Analog module Bit 4: Channel information present Bit 7 ... 5: 0 (fix)
2	00h (fix)
3	00h (fix)

Record set 0
$Diagnostic_{going}$

After error correction automatically a $diagnostic_{going}$ occurs if the diagnostic interrupt release is still active.

Byte	Bit 7 ... Bit 0
0	00h (fix)
1	Bit 3 ... 0: Module class 0101 Analog module Bit 4: Channel information present Bit 7 ... 5: reserved
2	00h (fix)
3	00h (fix)

Record set 1
channel specific diagnostic$_{incoming}$ (Byte 0 to 14)

The *record set 1* contains the 4Byte of record set 0 and additional 12Byte module specific diagnostic data.

The diagnostic bytes have the following assignment:

Byte	Bit 7 ... Bit 0
0 ... 3	Content record set 0 (see page before)
4	Bit 6 ... 0: Channel type 70h: Digital input 71h: Analog input 72h: Digital output 73h: Analog output 74h: Analog in-/output Bit 7: reserved
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 07h)
7	Bit 0: Channel error Channel 0 Bit 1: Channel error Channel 1 Bit 2: Channel error Channel 2 Bit 3: Channel error Channel 3 Bit 4: Channel error Channel 4 Bit 5: Channel error Channel 5 Bit 6: Channel error Channel 6 Bit 7: 0 (fix)
8	Bit 0: Parameterization error Channel 0 Bit 1: 0 (fix) Bit 2: 0 (fix) Bit 3: 0 (fix) Bit 4: Wire break Channel 0 Bit 5: 0 (fix) Bit 6: Measuring range underflow Channel 0 Bit 7: Measuring range overflow Channel 0
9	Bit 0: Parameterization error Channel 1 Bit 1: 0 (fix) Bit 2: 0 (fix) Bit 3: 0 (fix) Bit 4: Wire break Channel 1 Bit 5: 0 (fix) Bit 6: Measuring range underflow Channel 1 Bit 7: Measuring range overflow Channel 1
10	Bit 0: Parameterization error Channel 2 Bit 1: 0 (fix) Bit 2: 0 (fix) Bit 3: 0 (fix) Bit 4: Wire break Channel 2 Bit 5: 0 (fix) Bit 6: Measuring range underflow Channel 2 Bit 7: Measuring range overflow Channel 2

continued ...

... continue Record set 1

Byte	Bit 7 ... Bit 0
11	Bit 0: Parameterization error Channel 3 Bit 1: 0 (fix) Bit 2: 0 (fix) Bit 3: 0 (fix) Bit 4: Wire break Channel 3 Bit 5: 0 (fix) Bit 6: Measuring range underflow Channel 3 Bit 7: Measuring range overflow Channel 3
12	Bit 0: Parameterization error Channel 4 Bit 1: 0 (fix) Bit 2: 0 (fix) Bit 3: 0 (fix) Bit 4: Wire break Channel 4 Bit 5: 0 (fix) Bit 6: Measuring range underflow Channel 4 Bit 7: Measuring range overflow Channel 4
13	Bit 0: Parameterization error Channel 5 Bit 1: 0 (fix) Bit 2: 0 (fix) Bit 3: Short circuit Channel 5 Bit 4: Wire break Channel 5 Bit 7 ... 5: 0 (fix)
14	Bit 0: Parameterization error Channel 6 Bit 1: 0 (fix) Bit 2: 0 (fix) Bit 3: Short circuit Channel 6 Bit 4: Wire break Channel 6 Bit 7 ... 5: 0 (fix)

Digital part

Outline

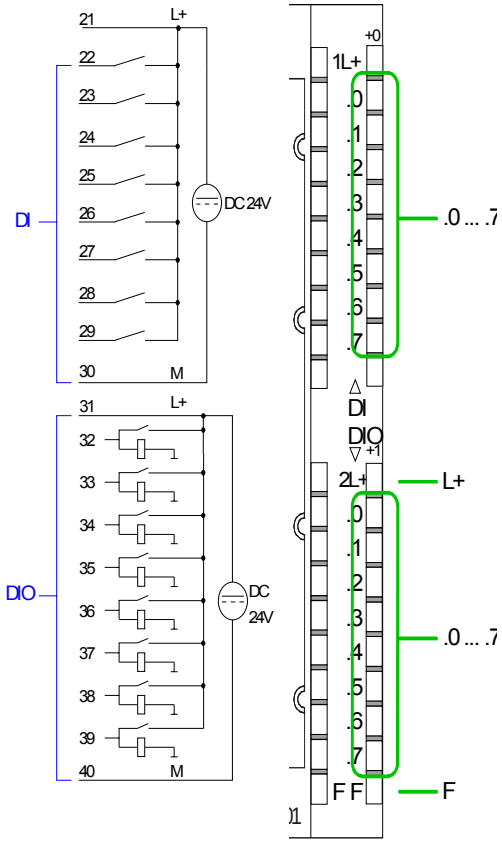
The digital part consists of 8 input and 8 in-/output channels. Each of these channels shows its state via a LED. By means of the parameterization you may assign interrupt properties to every digital input. Additionally you may parameterize the digital inputs as counter (max. 15kHz, with release 2 max. 100kHz).

The output channels provide a diagnostic function, i.e. as soon as an output is active, the concerning input is set to "1". At a short circuit at the load, the input is set to "0" and the error may be recognized by evaluating the input. The DIO area has to be provided with external DC 24V.

Pin assignment
Status monitor

Pin	Assignment
21	Supply voltage +DC 24V
22	Input I+0.0 / Counter 0(A)
23	Input I+0.1 / Counter 0(B)
24	Input I+0.2 / Gate0/Latch0/Reset0
25	Input I+0.3 / Counter 1(A)
26	Input I+0.4 / Counter 1(B)
27	Input I+0.5 / Gate1/Latch1/Reset1
28	Input I+0.6 / Counter 2(A)
29	Input I+0.7 / Counter 2(B)
30	Ground DI
31	Supply voltage +DC 24V
32	I/Q+1.0 / Gate2/Latch2/Reset2
33	I/Q+1.1 / Counter 3(A)
34	I/Q+1.2 / Counter 3(B)
35	I/Q+1.3 / Gate3/Latch3/Reset3
36	I/Q+1.4 / OUT0/Latch0/Reset0
37	I/Q+1.5 / OUT1/Latch1/Reset1
38	I/Q+1.6 / OUT2/Latch2/Reset2
39	I/Q+1.7 / OUT3/Latch3/Reset3
40	Ground DIO

Connection



LEDs

DI:

.0 ... .7 LEDs (green) I+0.0 to I+0.7 Starting at app. 15V the signal "1" is recognized at the input and the acc. LED is on

DIO:

2L+ LED (green) Supply voltage for DIO is present

.0 ... .7 LEDs (green) I/Q+1.0 to I/Q+1.7 is on at active output res. input

F LED (red) Error at overload or short circuit

Attention!

Please regard that the voltage at an output channel is always ≤ the supply voltage connected to L+.

Please regard also that due to the parallel connection of in- and output channel for each group one set output can be supplied via a connected input signal.

A thus connected output remains active even with shut down supply voltage.

Non-observance may cause damages of the module.

Access to the digital part

By including the GSD speedbus.gsd into your hardware configurator the module is monitored in the hardware catalog.

After the installation of the GSD you will find the CPU 314ST under *Additional Field devices* \ *I/O* \ *VIPA_SpeedBus*.

The CPU 314ST creates in its peripheral range 48Byte for data input and 24Byte for data output. Out of this, the digital part occupies 34Byte for digital input and 18Byte for digital output. Without a hardware configuration the ranges start at address 1024.

In the following table the according areas are marked ▭ :

Input range

Address	Access	Assignment
+0	Byte	Digital Input I+0.0 ... I+0.7
+1	Byte	Digital Input I+1.0 ... I+1.7
+2	Word	reserved
+4	Word	Analog Input CH0
+6	Word	Analog Input CH1
+8	Word	Analog Input CH2
+10	Word	Analog Input CH3
+12	Word	Analog Input CH4
+14	Word	reserved
+16	Double word	Counter 0 / Latch 0
+20	Word	reserved
+22	Word	Status Counter 0
+24	Double word	Counter 1 / Latch 1
+28	Word	reserved
+30	Word	Status Counter 1
+32	Double word	Counter 2 / Latch 2
+36	Word	reserved
+38	Word	Status Counter 2
+40	Double word	Counter 3 / Latch 3
+44	Word	reserved
+46	Word	Status Counter 3

Output range

For the output you enter a Word value into the data output range.

Address	Access	Assignment
+0	Byte	reserved
+1	Byte	Digital Output Q+1.0 ... Q+1.7
+2	Word	reserved
+4	Word	Analog Output CH0
+6	Word	Analog Output CH1
+8	Word	reserved
+10	Word	Status Counter 0
+12	Word	reserved
+14	Word	Status Counter 1
+16	Word	reserved
+18	Word	Status Counter 2
+20	Word	reserved
+22	Word	Status Counter 3

Counter - Fast introduction

Fast introduction

The CPU 314ST has 4 parameterizable counters integrated that may be controlled separately. During the count process the counter signal is recognized and evaluated. Every counter occupies one double word in the input range for the *counter register* and one word in the in- and output range for the *input* res. *output status*.

Preset res. parameterize counter

By including the speedbus.gsd you may preset all counter parameters via a hardware configuration. Here you may define among other:

- Interrupt behavior
- Assignment I/O (Gate, Latch, Reset, OUT)
- Input filter
- Counter operating mode res. behavior
- Start value for load value, end value and comparison value register

You may alter the parameters during runtime by using the SFC 55, 56, 57 and 58, except of the parameters in record set 0. Here you have to send the wanted parameters to the counter by means of the user application using the according SFC and sending the data as record set.

Control counter

The counter is controlled via the internal gate (I-gate). The I-gate is the sum of hardware- (HW) and Software-gate (SW), where the HW-gate evaluation may be deactivated via the parameterization.

HW-gate: Input at $Gate_x$-input at module

SW-gate: Open (activate): Set once output status Bit 2 in the output range

Close (deactivate): Set output status Bit 10 in the output range

The following states influence the gates:

SW-gate	HW-gate	influences I-gate
0	with positive edge	0
1	with positive edge	1
with positive edge	1	1
with positive edge	0	0
with positive edge	de-activated	1

Read counter

Depending on the status setting, the counter register contains the recent counter value (input status Bit 0=0) or the recent Latch value (input status Bit 0=1).

By setting the output status Bit 8 the recent Latch value is transferred to the counter register in the input area.

Transfer the recent counter value by setting the output status Bit 0.

Counter status word

Besides of the counter register in the input area you may find a status word for every counter in the in- res. output range. You may monitor the status or influence the counter by setting according bits like e.g. activate the SW gate.

Input status word

The status word in the input range has the following structure:

Bit	Label	Function
0	COUNT_LTCH	0: Value in input image is counter value 1: Value in input image is latch value
1	CTRL_Count_DO	Is set when the digital output is released
2	STS_SW-GATE	Status software gate (set when SW gate active)
3	reserved	reserved
4	STS_STRT	Status hardware gate (set when HW gate active)
5	STS_GATE	Status internal gate (set when internal gate active)
6	STS_DO	Status of the digital output of a counter (DO)
7	STS_C_DN	Status counter direction backwards
8	STS_C_UP	Status counter direction forward
9	STS_CMP*	Status Comparison (**Comp**are) is set when counter value = comparison value. If comparison is parameterized *never*, the bit is never set
10	STS_END*	Status set when end value is reached
11	STS_OFLW*	Status overflow
12	STS_UFLW*	Status underrun
13	STS_ZP*	Status zero run
14	STS_LTCH	Status of the Latch input of a counter
15	NEW_LTCH	Is set if value in the Latch register has changed

* The bits remains set until reset with RES (Bit 6 status word output image).

Output status word

After setting a bit in the output status word this is immediately set back. Please regard that setting and resetting of a function at the output status word takes place with different bits:

Bit	Label	Function
0	Get _Count_Val	Transfer counter value to process image
1	Set_Count_DO	Release the digital output for counter (output only available via counter)
2	Set_SW-Gate	Set software gate (not allowed at OB 100)
3	reserved	-
4	reserved	-
5	Set_Count_Val	Set counter temporarily to a value (the counter value for Z_x has to be transferred before via record set (9A+x)h
6	Reset_STS	Reset bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP
7	reserved	-
8	Get_Latch_Val	Transfer Latch value to process image
9	Reset_Count_DO	Lock digital output for counter (output available only via process image)
10	Reset_SW_Gate	Reset software gate
12	reserved	-
...	...	...
15	reserved	-

Counter inputs (Connections)

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

$Counter_x$ (A)

Pulse input for count signal res. track A of an encoder. Here you may connect encoder with 1-, 2- or 4-tier evaluation.

$Counter_x$ (B)

Direction signal res. track B of the encoder. Via the parameterization you may invert the direction signal.

The following inputs may be assigned to a pin at the module via parameterization:

$Gate_x$

This input allows you to open the HW gate with a high peek and thus start a count process.

$Latch_x$

With a positive edge at $Latch_x$ the recent counter value is stored in a memory that you may read at need.

$Reset_x$

As long as $Reset_x$ is applied with a positive level the counter is still reset to the load value.

Counter outputs

Every counter has an assigned output channel. The following behavior for the output channel can be set via parameterization:

- No comparison: output is not headed for
- Count value ≥ comparison value: output is set
- Count value ≤ comparison value: output is set
- Count value = comparison value: output is set

Maximum count frequency

At this time the maximum frequency for the release version 1 independent from the number of activated counters is 15kHz. Starting with release version 2 a max. of 100kHz is possible.

Counter - Parameterization

Overview

The parameterization takes place in the hardware configurator. Here, parameter data are transferred existing of the following components:

Byte	Record set	Description
16	0h	Counter mode C0 ... C3
4	7Fh	Diagnostic interrupt
16	80h	Edge selection for process interrupt
32	81h	Filter value I+0.0 ... I+1.7
16	82 ... 86h	C0: Comparison, Set, End value, hysteresis, pulse
16	87h	C0: Sum parameter (Comparison, Set, End value, hysteresis and pulse)
16	88 ... 8Ch	C1: Comparison, Set, End value, hysteresis, pulse
16	8Dh	C1: Sum parameter (Comparison, Set, End value, hysteresis and pulse)
16	8E ... 92h	C2: Comparison, Set, End value, hysteresis, pulse
16	93h	C2: Sum parameter (Comparison, Set, End value, hysteresis and pulse)
16	94 ... 98h	C3: Comparison, Set, End value, hysteresis, pulse
16	99h	C3: Sum parameter (Comparison, Set, End value, hysteresis and pulse)
4	9Ah	C0: Count value that is transferred to counter by setting Bit 5 in the output status word
4	9Bh	C1: Count value that is transferred to counter by setting Bit 5 in the output status word
4	9Ch	C2: Count value that is transferred to counter by setting Bit 5 in the output status word
4	9Dh	C3: Count value that is transferred to counter by setting Bit 5 in the output status word
2	9Eh	Analog-/Digital part activated or deactivated

Except of the parameter in record set 0, you may transfer the other parameters during runtime by using the SFC 55, 56, 57 and 58 to the digital part. For this you have to transfer the wanted parameters to the counter by using the according SFC in the user application.

Record set 0
Counter mode

Via the record set 0 you may preset a counter mode for every counter as double word. Please regard that the record set 0 may not be transferred during runtime. Record set 0 has the following structure:

Byte	Description
0 ... 3	Counter mode C0
4 ... 7	Counter mode C1
8 ... 11	Counter mode C2
12 ... 15	Counter mode C3

Counter mode

The double word for the counter mode has the following structure:

Byte	Bit 7 ... 0
0	*Bit 2 ... 0: Signal evaluation* 000b = Counter de-activated At de-activated counter the further parameter settings for this counter are ignored and the according I/O channel is set as "normal" output if this should be used as output. 001b = Encoder 1-tier (at counter$_x$ (A_x) and counter$_x$ (B_x)) 010b = Encoder 2-tier (at counter$_x$ (A_x) and counter$_x$ (B_x)) 011b = Encoder 4-tier (at counter$_x$ (A_x) and counter$_x$ (B_x)) 100b = Pulse/direction (pulse at counter$_x$ (A_x) and direction at counter$_x$ (B_x)) *Bit 6 ... 3: C_x input* (Function of the counter input as gate, latch or reset) 0000b = de-activated (counter starts at set SW gate) 0001b = Gate$_x$ The input of counter$_x$ serves as gate. High peek at gate activates the HW gate. The counter may only start when HW and SW gate are set. 0010b = Monoflop* 0100b = Latch$_x$ (Positive edge at input saves counter value) 1000b = Reset$_x$ (Positive level at input sets counter back) *Bit 7: Gate function (internal gate)* 0 = abort (count process starts again at load value) 1 = interrupt (count process continues with counter value)
1	*Bit 2 ... 0: Output set* (OUT$_x$ of counter$_x$ is set when condition is met) 000b = never 001b = counter value >= comparison value 010b = counter value <= comparison value 100b = counter value = comparison value *Bit 3: Count direction* 0 = count direction inverted: OFF (count direction at B_x not inverted) 1 = count direction inverted: ON (count direction at B_x inverted) *Bit 7 ... 4: reserved*

* not supported at this time

continued ...

... continue

Byte	Bit 7 ... 0
2	*Bit 5 ... 0: Counter function* 000000b = Count endless 000001b = Once: forward 000010b = Once: backwards 000100b = Once: no main direction 001000b = Periodic: forward 010000b = Periodic: backwards 100000b = Periodic: no main direction More details at "Counter - Functions" below. *Bit 7 ... 6: C_x In-/Output* (Function of the counter I/O as OUT, Latch or Reset) 00b = O: OUT_x (at comparison function) 01b = I: $Latch_x$ (rising edge saves counter value) 10b = I: $Reset_x$ (positive level resets counter)
3	*Bit 5 ... 0: Interrupt behavior* Bit 0: Proc. interrupt HW gate open Bit 1: Proc. interrupt HW gate closed Bit 2: Proc. interrupt overflow Bit 3: Proc. interrupt underrun Bit 4: Proc. interrupt comparison value Bit 5: Proc. interrupt end value By setting the Bits you may activate the wanted process interrupts. *Bit 7 ... 6: reserved*

Record set 7Fh
Diagnostic interrupt

This record set activates res. de-activates the diagnostic function.
A diagnostic interrupt occurs when during a process interrupt execution another process interrupt is initialized for the same event.

The record set has the following structure:

Byte	Bit 15 ... 0
0...1	*Bit 15 ... 0: Diagnostic interrupt* 0000h = de-activated 0001h = activated
2...3	*Bit 15 ... 0: reserved*

Record set 80h
Edge selection

Via this record set you may activate a process interrupt for I+0.0 ... I+1.7 and define for which edge type of the input signal a process interrupt is thrown.

The record set has the following structure:

Byte	Bit 7 ... 0
0	*Bit 1 ... 0: Edge selection I+0.0* 00b = de-activated 01b = Process interrupt at rising edge 10b = Process interrupt at falling edge 11b = Process interrupt at rising and falling edge *Bit 7 ... 2: reserved*
...	...
15	*Bit 1 ... 0: Edge selection I+1.7* 00b = de-activated 01b = Process interrupt at rising edge 10b = Process interrupt at falling edge 11b = Process interrupt at rising and falling edge *Bit 7 ... 2: reserved*

Record set 81h
Input filter

This record set allows you to preset an input filter in steps of 2.56µs steps for I+0.0 ... I+1.7. By preceding a filter you define how long an input signal must be present before it is recognized as "1" signal. With the help of filters you may e.g. filter signal peaks at a blurred input signal.

The entry happens as a factor of 2.56µs and is within the range 1 ... 16000 i.e. 2.56µs ... 40.96ms.

The record set has the following structure:

Byte	Bit 15 ... 0
0 ... 1	Bit 15 ... 0: Input filter I+0.0 in 2.56µs
2 ... 3	Bit 15 ... 0: Input filter I+0.1 in 2.56µs
4 ... 5	Bit 15 ... 0: Input filter I+0.2 in 2.56µs
...	...
30 ... 31	Bit 15 ... 0: Input filter I+1.7 in 2.56µs

Record set 82 ... 99h
Counter parameter

Each of the following counter parameters has an assigned record set depending on the counter number. Additionally for every counter the parameter are summoned in one record set.

The record sets have the same structure for every counter. Please refer to the following table for the structure and the according record set number assignment. The record set has the following structure:

Count. 0 87h	Count. 1 8Dh	Count. 2 93h	Count. 3 99h	Type	Function
82h	88h	8Eh	94h	Double word	Comparison value
83h	89h	8Fh	95h	Double word	Load value
84h	8Ah	90h	96h	Double word	End value
85h	8Bh	91h	97h	Word	Hysteresis
86h	8Ch	92h	98h	Word	Pulse

… Continue
Record set 82 … 99h

Comparison value

Via the parameterization you may preset a comparison value that may influence the counter output res. throw a process interrupt when compared with the recent counter value. The behavior of the output res. the process interrupt has to be set via the record set 0.

Load value, end value

You may define a main counting direction for every counter via the parameterization.

If "none" or "endless" is chosen, the complete counting range is available:

Counter limits	Valid value range
Lower count limit	- 2 147 483 648 (-2^{31})
Upper count limit	+ 2 147 483 647 (2^{31}-1)

Otherwise you may set an upper and a lower limit by setting a *load value* as start and an *end value*.

Hysteresis

The hysteresis serves the avoidance of many toggle processes of the output and the interrupt, if the counter value is in the range of the comparison value. You may set a range of 0 to 255. The settings 0 and 1 deactivate the hysteresis. The hysteresis influences zero run, comparison, over- and underflow.

Pulse (Pulse duration)

The pulse duration tells for what time the output is set when the parameterized comparison criterion is reached res. overstepped. The pulse duration can be set in steps of 2.048ms between 0 and 522.24ms.

If the pulse duration = 0, the output is set active until the comparison condition is not longer fulfilled.

Note!

More details are under "Counter – Additional functions" below!

Datensatz 9A ... 9Dh
Set counter value temporary

A register can be preset using record set (9A+x)h. The current counter value is replaced by the register value by setting bit 5 of the output status word without any influence to the load value.

Record set 9Eh
Module selection

Using this record set you can de-activate the digital res. analog part. If a part is de-activated the corresponding area of the process image is just reserved.

The record set has the following structure:

Byte	Bit 15 ... 0
0...1	*Bit 15 ... 0: Module selection* 0000h = Digital / analog part activated (default) 0001h = Digital part de-activated 0002h = Analog part de-activated

Counter - Functions

Outline

You may count forward and backwards and choose between the following counter functions:

- Count endless – e.g. distance measuring with incremental encoder
- Count once – e.g. count to a maximum limit
- Count periodic – e.g. count with repeated counter process

In the operating modes "Count once" and "Count periodic" you may define a counter range as start and end value via the parameterization.

For every counter additional parameterizable functions are available like gate function, latch function, comparison, hysteresis and process interrupt.

Main counting direction

Via the parameterization you have the opportunity to define a main counting direction for every counter.

If "none" is chosen, the complete counting range is available:

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

Main counting direction forward

Upper restriction of the count range. The counter counts 0 res. load value in positive direction until the parameterized end value –1 and jumps then back to the load value with the next following encoder pulse.

Main counting direction backwards

Lower restriction of the count range. The counter counts from the parameterized start- res. load value in negative direction to the parameterized end value +1 and jumps then back to the start value with the next following encoder pulse.

Abort - interrupt

Abort count process

The count process starts after closing and restart of the gate beginning with the load value.

Interrupt count process

The count process continuous after closing and restart of the gate beginning with the last recent counter value.

Count Continuously

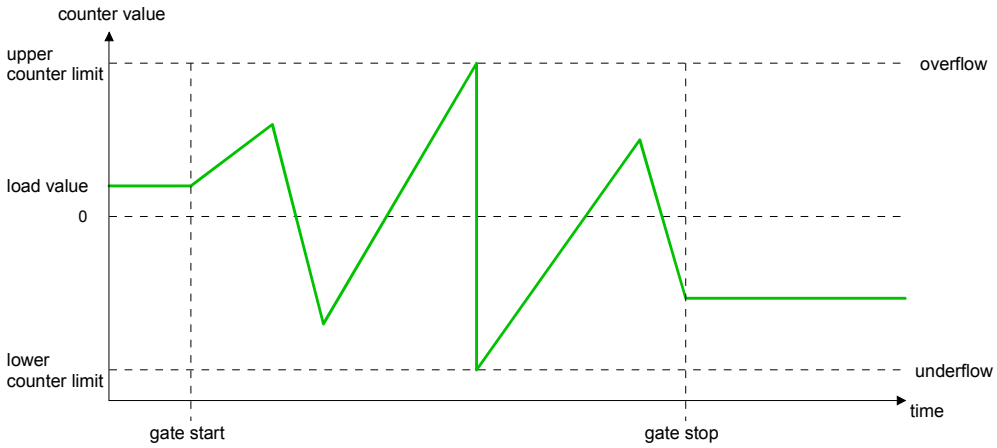
In this operating mode, the counter counts from 0 res. from the load value.

When the counter counts forward and reaches the upper count limit and another counting pulse in positive direction arrives, it jumps to the lower count limit and counts from there on.

When the counter counts backwards and reaches the lower count limit and another counting pulse in negative direction arrives, it jumps to the upper count limit and counts from there on.

The count limits are set to the maximum count range.

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



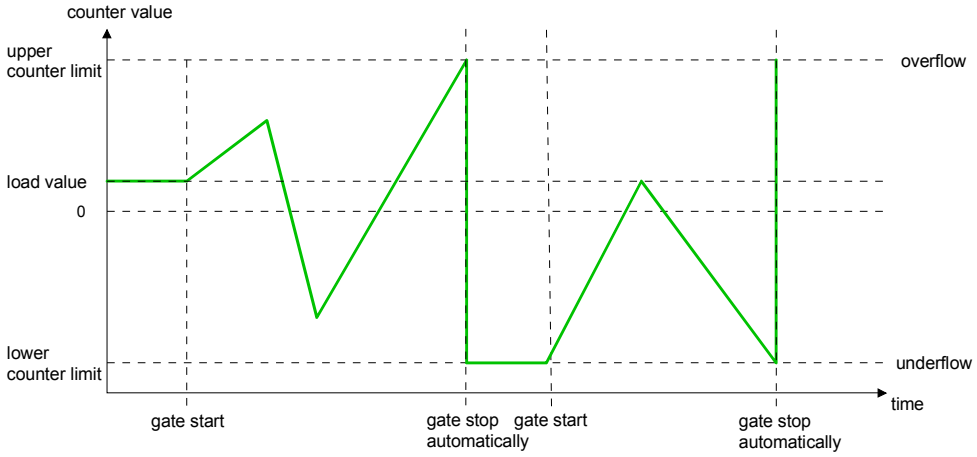
Count Once

No main counting direction

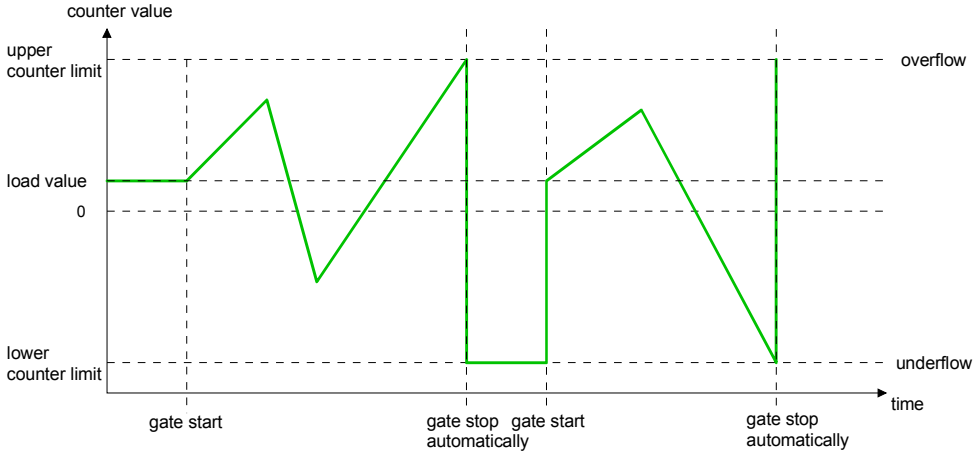
- The counter counts once starting with the load value.
- You may count forward or backwards.
- The count limits are set to the maximum count range.
- At over- or underrun at the count limits, the counter jumps to the according other count limit and the gate is automatically closed.
- To restart the count process, you must create a positive edge of the gate.
- At interrupting gate control, the count process continuous with the last recent counter value.
- At aborting gate control, the counter starts with the load value.

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

Interrupting gate control:



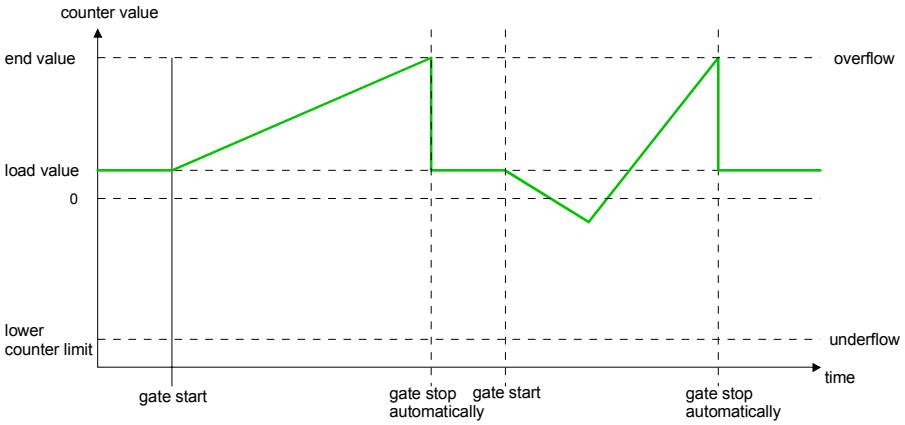
Aborting gate control:



Main counting direction forward

- The counter counts starting with the load value.
- When the counter reaches the end value –1 in positive direction, it jumps to the load value at the next positive count pulse and the gate is automatically closed.
- To restart the count process, you must create a positive edge of the gate. The counter starts with the load value.

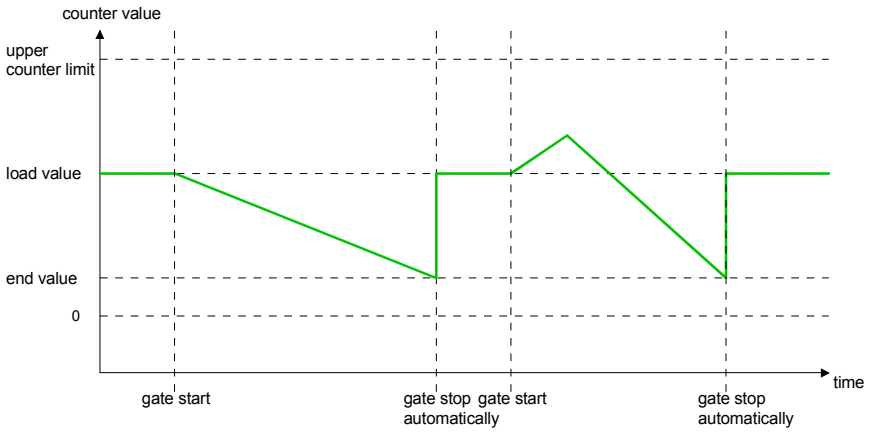
	Valid value range
Limit value	-2 147 483 646 ($-2^{31}+1$) to +2 147 483 646 ($2^{31}-1$)
Lower count limit	-2 147 483 648 (-2^{31})



Main counting direction backwards

- The counter counts backwards starting with the load value.
- When the counter reaches the end value +1 in negative direction, it jumps to the load value at the next negative count pulse and the gate is automatically closed.
- To restart the count process, you must create a positive edge of the gate. The counter starts with the load value.

	Valid value range
Limit value	-2 147 483 646 ($-2^{31}+1$) to +2 147 483 646 ($2^{31}-1$)
Upper count limit	+2 147 483 646 ($2^{31}-1$)

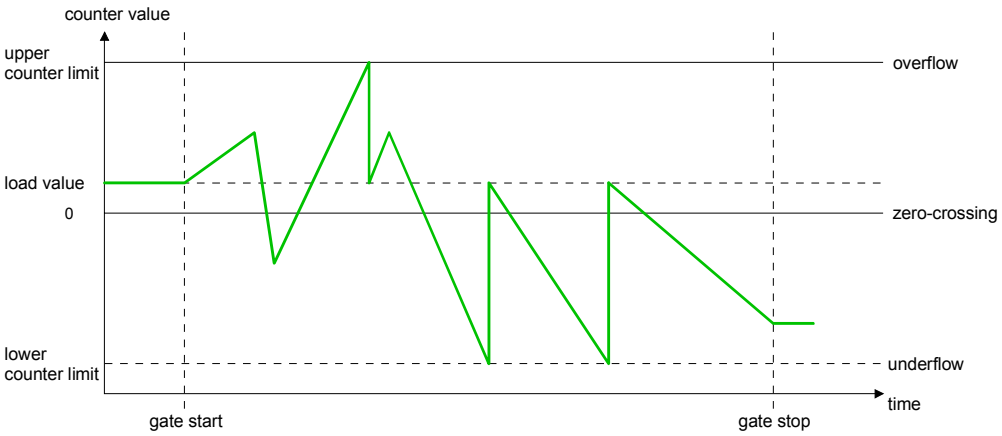


Count Periodically

No main counting direction

- The counter counts forward or backwards starting with the load value.
- At over- or underrun at the count limits, the counter jumps to the according other count limit and counts from there on.
- The count limits are set to the maximum count range.

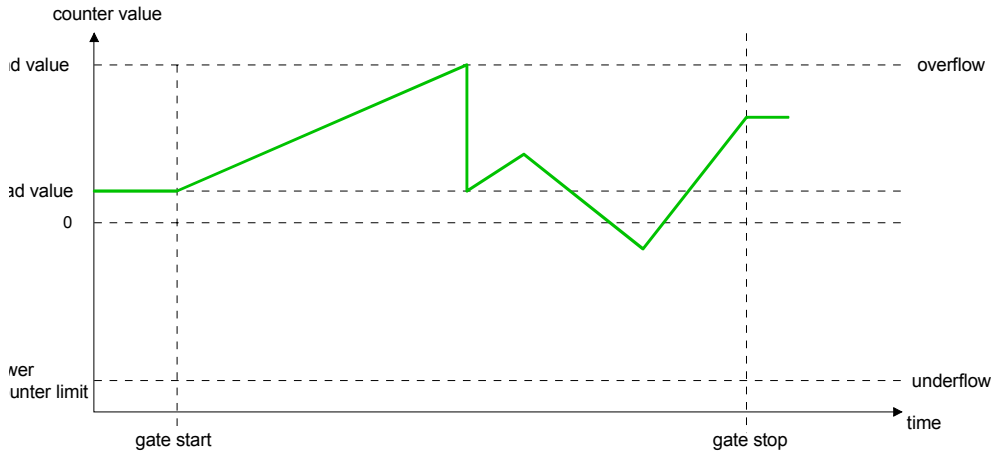
	Valid value range
Lower count limit	-2 147 483 648 (-2^{31})
Upper count limit	+2 147 483 647 (2^{31}-1)



Main counting direction forward

- The counter counts forward starting with the load value
- When the counter reaches the end value –1 in positive direction, it jumps to the load value at the next positive count pulse.

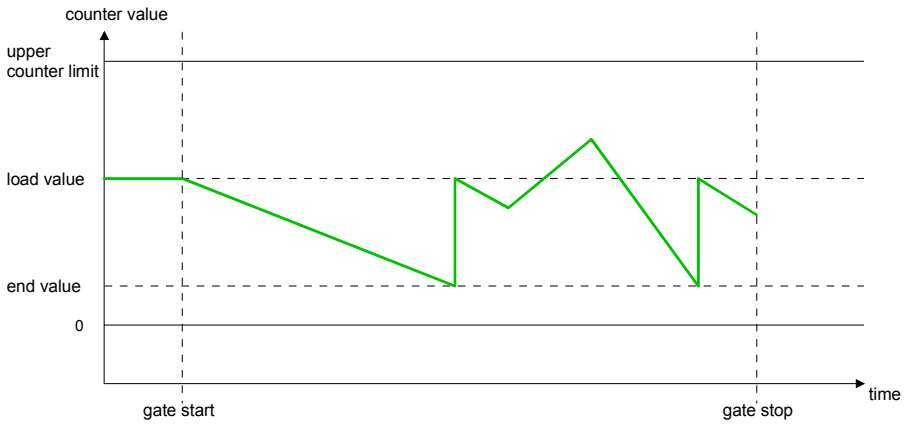
	Valid value range
Limit value	-2 147 483 647 (-2^{31}+1) to +2 147 483 647 (2^{31}-1)
Lower count limit	-2 147 483 648 (-2^{31})



Main counting direction backwards

- The counter counts backwards starting with the load value
- When the counter reaches the end value+1 in negative direction, it jumps to the load value at the next negative count pulse.
- You may exceed the upper count limit.

	Valid value range
Limit value	-2 147 483 647 ($-2^{31}+1$) to +2 147 483 647 ($2^{31}-2$)
Upper count limit	+2 147 483 647 ($2^{31}-1$)



Counter - Additional functions

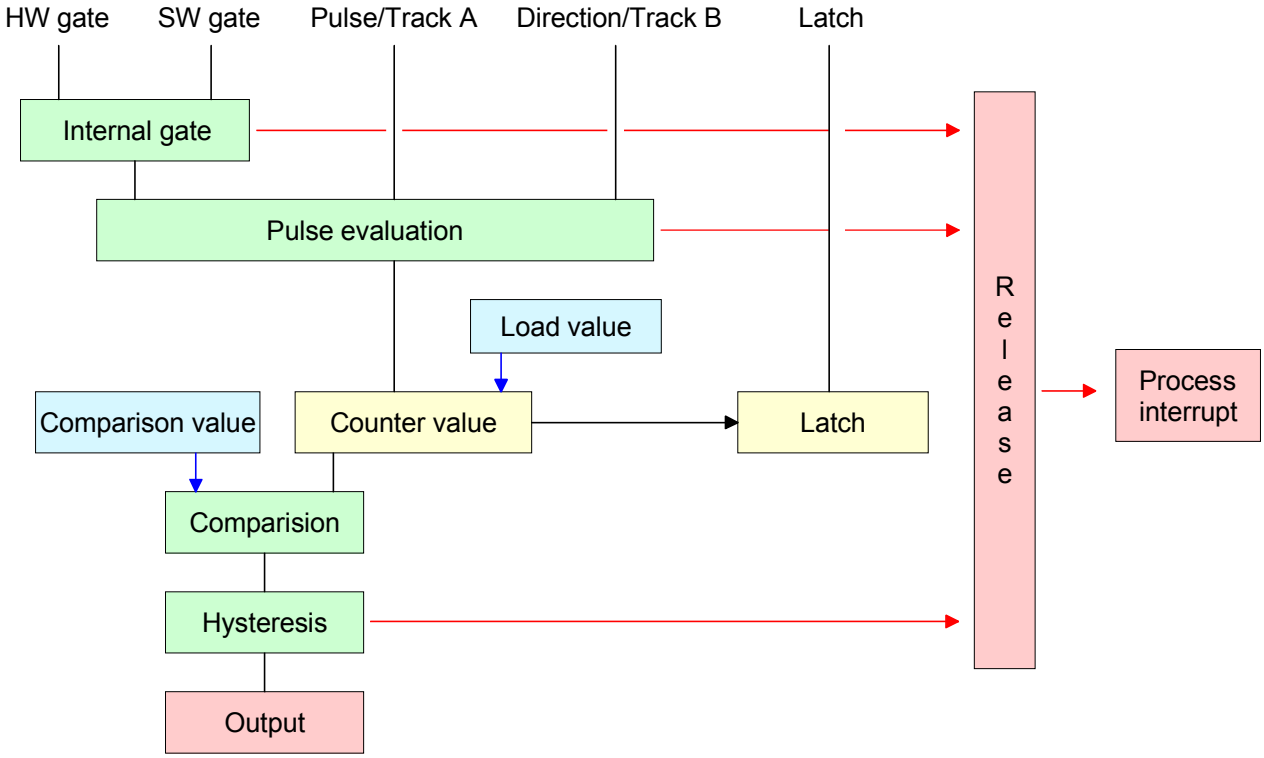
Outline

The following additional functions may be set via the parameterization for every counter:

- Gate function
  The gate function serves the start, stop and interrupt of a count function.
- Latch function
  A positive edge at the digital input "Latch" stores the recent counter value in the latch register.
- Comparison
  You may set a comparison value that activates res. de-activates a digital output res. releases a process interrupt depending on the counter value.
- Hysteresis
  The setting of a hysteresis avoids for example a high output toggling when the value of an encoder signal shifts around a comparison value.

Schematic structure

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



Gate function

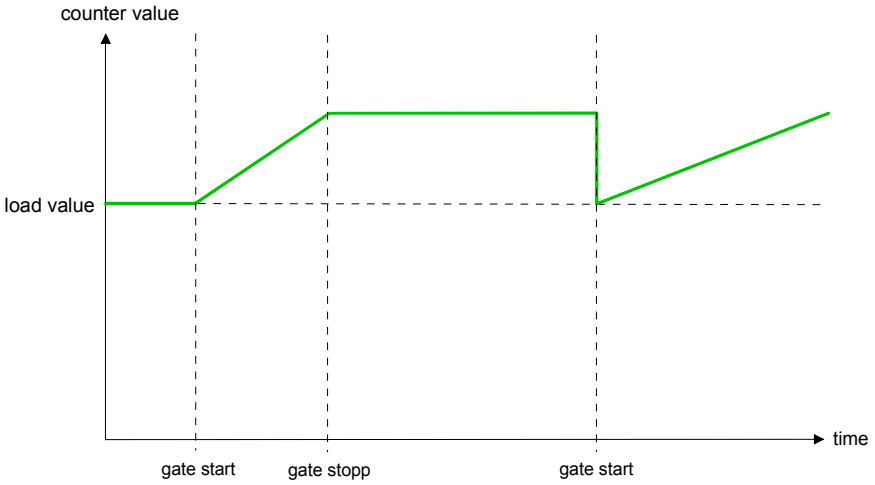
The activation res. de-activation of a counter happens via an internal gate (I-gate). The I-gate consists of a software gate (SW-gate) and a Hardware gate (HW-gate). The SW-gate is opened (activated) via your user application by setting the output status bit 2 for the according counter. The SW-gate is closed (deactivated) by setting the output status bit 10. The HW-gate is controlled via the concerning "Gate" input. The parameterization allows you to de-activate the consideration of the HW-gate so that the counter activation can take place only via the SW-gate. The following states influence the I-gate:

SW-gate	HW-gate	influences I-gate
0	with positive edge	0
1	with positive edge	1
with positive edge	1	1
with positive edge	0	0
with positive edge	de-activated	1

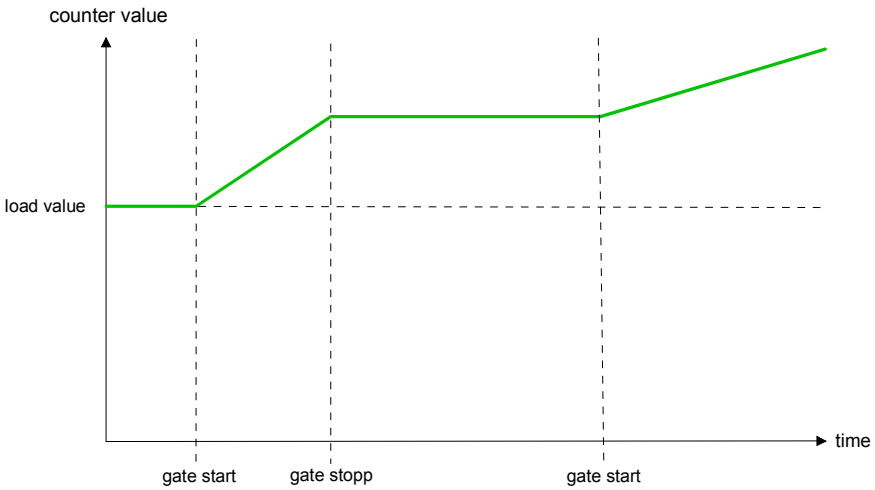
Gate function abort and interrupt

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

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



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



Gate control abort, interruption

Gate control via SW gate, aborting
(Parameterization: record set 0, Byte 0, Bit 7 ... 3 = 00000b)

SW gate	HW gate	Reaction Counter
positive edge	de-activated	Restart with load value

Gate control via SW gate, interrupting
(Parameterization: record set 0, Byte 0, Bit 7 ... 3 = 10000b)

SW gate	HW gate	Reaction Counter
positive edge	de-activated	Continue

Gate control via SW/HW gate, aborting
(Parameterization: record set 0, Byte 0, Bit 7 ... 3 = 00001b)

SW gate	HW gate	Reaction Counter
positive edge	1	Continue
1	positive edge	Restart with load value

Gate control via SW/HW gate, interrupting
(Parameterization: record set 0, Byte 0, Bit 7 ... 3 = 10001b)

SW gate	HW gate	Reaction Counter
positive edge	1	Continue
1	positive edge	Continue

Gate control "Count once"

Gate control via SW/HW gate, operating mode "Count once"

If the internal gate has been closed automatically it may only be opened again under the following conditions:

SW gate	HW gate	Reaction I gate
1	positive edge	1
positive edge (after positive edge at HW gate)	positive edge	1

Latch function

As soon as during a count process a positive edge is recognized at the "Latch" input of a counter, the recent counter value is stored in the according latch register.

You may access the latch register via the "input image". For this set Bit 15 of the output status word.

At a new latch value additionally Bit 13 is set in the input status word. By setting Bit 15 in the output status word you may read the recent latch value of the according counter and reset the Bit 13 of the input status word.

Comparison

You pre-define the behavior of the counter output via the parameterization:

- output never switches
- output switch when counter value ≥ comparison value
- output switch when counter value ≤ comparison value
- output switch at comparison value

Output never switches

The output is set as normal output.

Output switch when counter value ≥ comparison value

The output remains set as long as the counter value is higher or equal comparison value.

Output switch when counter value ≤ comparison value

The output remains set as long as the counter value is lower or equal comparison value.

Pulse at comparison value

When the counter reaches the comparison value the output is set for the parameterized pulse duration.

If the pulse duration = 0 the output is set until the comparison condition is no longer met.

When you've set a main counting direction the output is only set at reaching the comparison value from the main counting direction.

Pulse duration

The pulse duration defines how long the output is set.
it may be preset in steps of 2.048ms between 0 and 522.24ms.

The pulse duration starts with the setting of the according digital output. The inaccuracy of the pulse duration is less than 2.048ms.

There is no past triggering of the pulse duration when the comparison value has been left and reached again during pulse output.

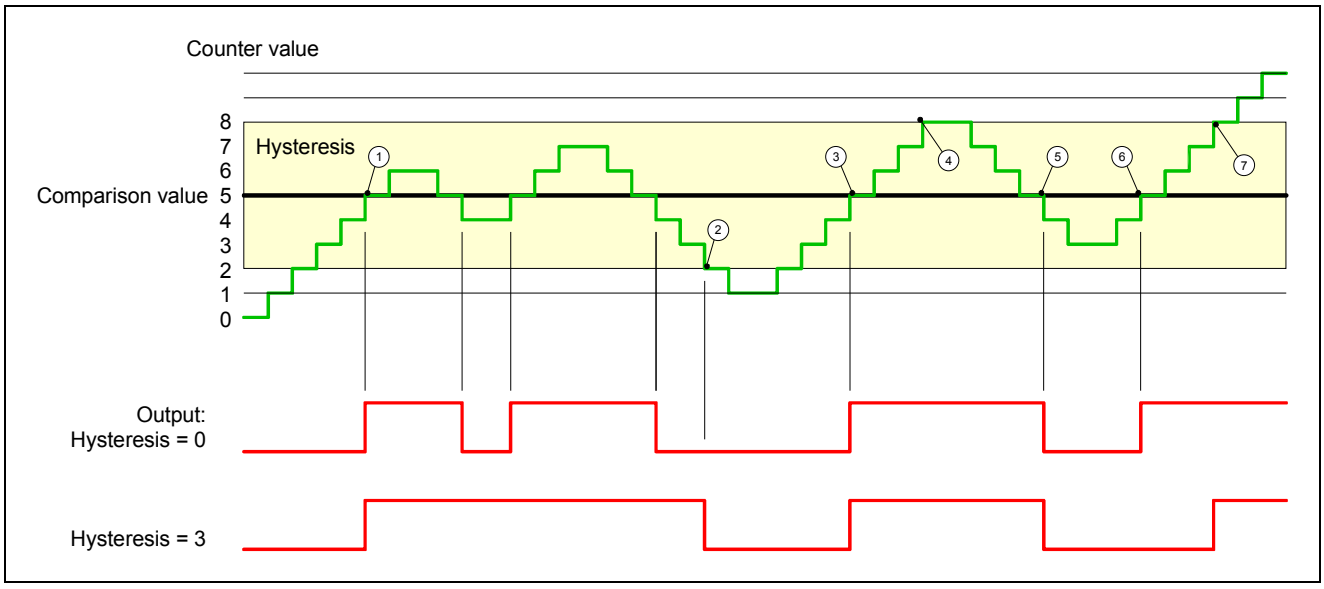
Hysteresis

The hysteresis serves e.g. the avoidance of many toggle processes of the output and the interrupt, if the counter value is in the range of the comparison value. You may set a range of 0 to 255. The settings 0 and 1 deactivate the hysteresis. The hysteresis influences the zero run, over- and underflow.

An activated hysteresis remains active after a change. The new hysteresis range is taken over at the next reach of the comparison value.

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

Effect at counter value ≥ comparison value



① Counter value ≥ comparison value → output is set and hysteresis activated

② Leave hysteresis range → output is reset

③ Counter value ≥ comparison value → output is set and hysteresis activated

④ Leave hysteresis range, output remains set for counter value ≥ comparison value

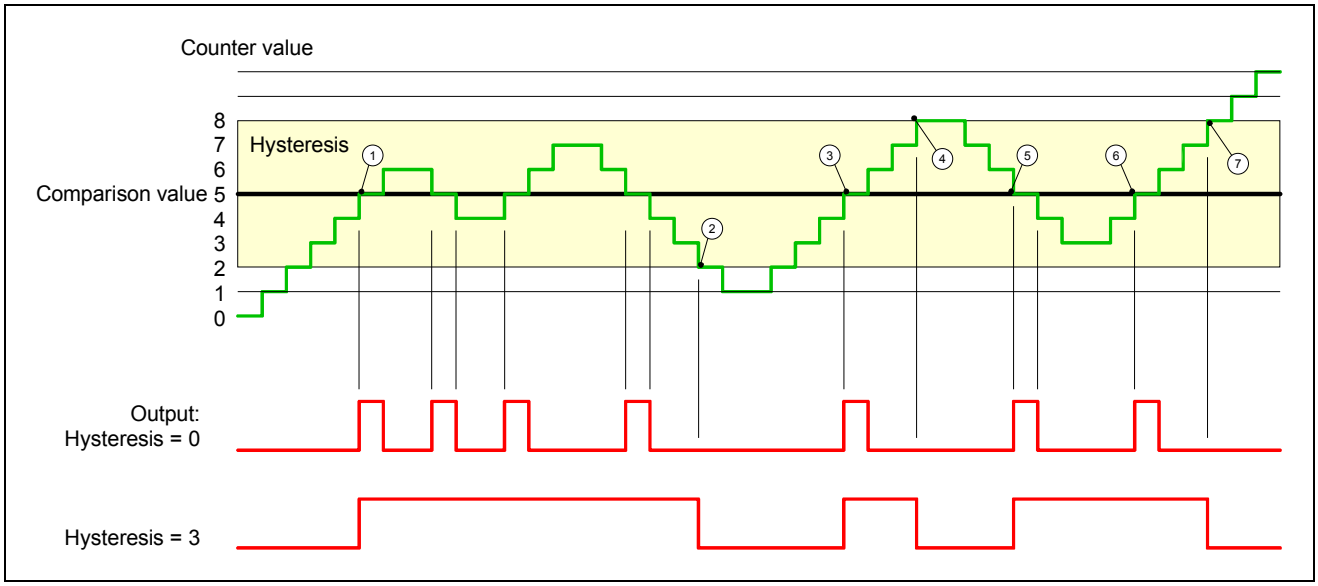
⑤ Counter value < comparison value and hysteresis active → output is reset

⑥ Counter value ≥ comparison value → output is not set for hysteresis active

⑦ Leave hysteresis range, output remains set for counter value ≥ comparison value

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

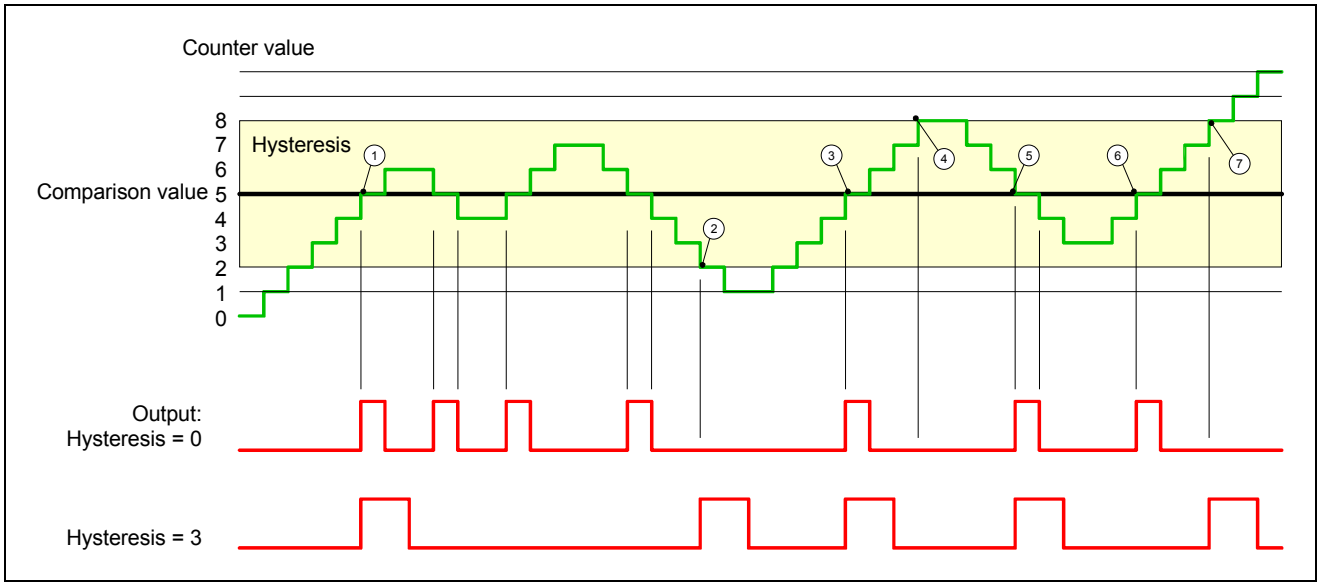
Effect at pulse at comparison value with pulse duration Zero



① Counter value = comparison value → output is set and hysteresis activated
② Leave hysteresis range → output is reset and counter value < comparison value
③ Counter value = comparison value → output is set and hysteresis activated
④ Output is reset for leaving hysteresis range and counter value > comparison value
⑤ Counter value = comparison value → output is set and hysteresis activated
⑥ Counter value = comparison value and hysteresis active → output remains set
⑦ Leave hysteresis range and counter value > comparison value → output is reset

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

Effect at pulse at comparison value with pulse duration not zero



① Counter value = comparison value → pulse of the parameterized duration is put out, the hysteresis is activated and the counting direction stored

② Leaving the hysteresis range contrary to the stored counting direction → pulse of the parameterized duration is put out, the hysteresis is de-activated

③ Counter value = comparison value → pulse of the parameterized duration is put out, the hysteresis is activated and the counting direction stored

④ Leaving the hysteresis range without changing counting direction → hysteresis is de-activated

⑤ Counter value = comparison value → pulse of the parameterized duration is put out, the hysteresis is activated and the counting direction stored

⑥ Counter value = comparison value and hysteresis active → no pulse

⑦ Leaving the hysteresis range contrary to the stored counting direction → pulse of the parameterized duration is put out, the hysteresis is de-activated

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

Counter - Diagnostic and interrupt

Outline

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

- Status changes at an input
- Status changes at the HW-gate
- Over- res. underflow
- Reaching a comparison value

Process interrupt

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

Local double word 8 of the OB 40

The *local double word 8* of the OB 40 has the following structure:

Local byte	Bit 7 ... Bit 0
8	Bit 0: Edge at I+0.0 Bit 1: Edge at I+0.1 Bit 2: Edge at I+0.2 Bit 3: Edge at I+0.3 Bit 4: Edge at I+0.4 Bit 5: Edge at I+0.5 Bit 6: Edge at I+0.6 Bit 7: Edge at I+0.7
9	Bit 0: Edge at I+1.0 Bit 1: Edge at I+1.1 Bit 2: Edge at I+1.2 Bit 3: Edge at I+1.3 Bit 4: Edge at I+1.4 Bit 5: Edge at I+1.5 Bit 6: Edge at I+1.6 Bit 7: Edge at I+1.7
10	Bit 0: Gate counter 0 open (activated) Bit 1: Gate counter 0 closed Bit 2: Over-/underflow/end value counter 0 Bit 3: Counter 0 reached comparison value Bit 4: Gate counter 1 open (activated) Bit 5: Gate counter 1 closed Bit 6: Over-/underflow/ end value counter 1 Bit 7: Counter 1 reached comparison value
11	Bit 0: Gate counter 2 open (activated) Bit 1: Gate counter 2 closed Bit 2: Over-/underflow/end value counter 2 Bit 3: Counter 2 reached comparison value Bit 4: Gate counter 3 open (activated) Bit 5: Gate counter 3 closed Bit 6: Over-/underflow/end value counter 3 Bit 7: Counter 3 reached comparison value

Diagnostic interrupt

Via the parameterization (record set 7Fh) you may activate a global diagnostic interrupt for the analog and digital part.

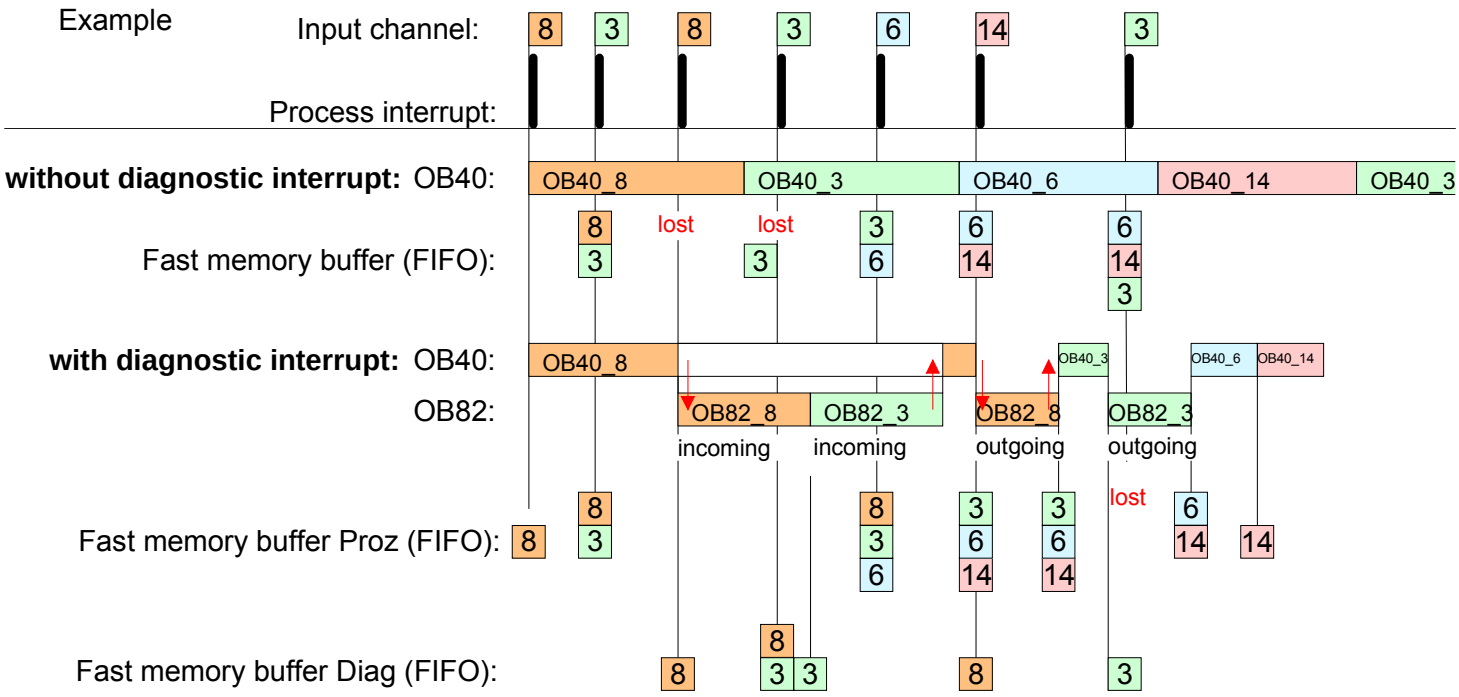
A diagnostic interrupt occurs when during a process interrupt execution in OB 40 another process interrupt is thrown for the same event. The initialization of a diagnostic interrupt interrupts the recent process interrupt execution in OB 40 and branches in OB 82 to diagnostic interrupt processing$_{incoming}$. If during the diagnostic interrupt processing other events are occurring at other channels that may also cause a process res. diagnostic interrupt, these are interim stored.

After the end of the diagnostic interrupt processing at first all interim stored diagnostic interrupts are processed in the sequence of their occurrence and then all process interrupts.

If a channel where currently a diagnostic interrupt$_{incoming}$ is processed res. interim stored initializes further process interrupts, these get lost. When a process interrupt for which a diagnostic interrupt$_{incoming}$ has been released is ready, the diagnostic interrupt processing is called again as diagnostic interrupt$_{going}$.

All events of a channel between diagnostic interrupt$_{incoming}$ and diagnostic interrupt$_{going}$ are not stored and get lost. Within this time window (1^{st} diagnostic interrupt$_{incoming}$ until last diagnostic interrupt$_{going}$) the SF-LED of the CPU is on. Additionally for every diagnostic interrupt$_{incoming/going}$ an entry in the diagnostic buffer of the CPU occurs.

Example



Diagnostic interrupt processing

Every OB 82 call causes an entry in the diagnostic buffer of the CPU containing error cause and module address.

By using the SFC 59 you may read the diagnostic bytes.

At de-activated diagnostic interrupt you have access to the last recent diagnostic event.

If you've activated the diagnostic function in your hardware configuration, the contents of record set 0 are already in the local double word 8 when calling the OB 82. The SFC 59 allows you to also read the record set 1 that contains additional information.

After leaving the OB 82 an clear assignment of the data to the last diagnostic interrupt is not longer possible.

The record sets of the diagnostic range have the following structure:

Record set 0 Diagnostic$_{incoming}$

Byte	Bit 7 ... 0
0	Bit 0: set at module failure Bit 1: 0 (fix) Bit 2: set at external error Bit 3: set at channel error Bit 4: set when external auxiliary supply is missing Bit 7 ... 5: 0 (fix)
1	Bit 3 ... 0: Module class 0101b: Analog 1111b: Digital Bit 4: Channel information present Bit 7 ... 5: 0 (fix)
2	Bit 3 ... 0: 0 (fix) Bit 4: Failure module internal supply voltage (output overload) Bit 7 ... 5: 0 (fix)
3	Bit 5 ... 0: 0 (fix) Bit 6: Process interrupt lost Bit 7: 0 (fix)

Record set 0 Diagnostic$_{going}$

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

Record set 0 (Byte 0 to 3):

Byte	Bit 7 ... 0
0	Bit 0: set at module failure Bit 1: 0 (fix) Bit 2: set at external error Bit 3: set at channel error Bit 4: set when external auxiliary supply is missing Bit 7 ... 5: 0 (fix)
1	Bit 3 ... 0: Module class 0101b: Analog module 1111b: Digital Bit 4: Channel information present Bit 7 ... 5: 0 (fix)
2	00h (fix)
3	00h (fix)

Diagnostic Record set 1 (Byte 0 ... 15)

The record set 1 contains the 4Byte of the record set 0 and additionally 12Byte module specific diagnostic data.

The diagnostic bytes have the following assignment:

Record set 1 (Byte 0 to 15):

Byte	Bit 7 ... 0
0 ... 3	Contents record set 0 (see page before)
4	Bit 6 ... 0: channel type (here 70h) 70h: Digital input 71h: Analog input 72h: Digital output 73h: Analog output 74h: Analog in-/output Bit 7: More channel types present 0: no 1: yes
5	Number of diagnostic bits per channel (here 08h)
6	Number of channels of a module (here 08h)
7	Bit 0: Error in channel group 0 (I+0.0 ... I+0.3) Bit 1: Error in channel group 1 (I+0.4 ... I+0.7) Bit 2: Error in channel group 2 (I+1.0 ... I+1.3) Bit 3: Error in channel group 3 (I+1.4 ... I+I.7) Bit 4: Error in channel group 4 (Counter 0) Bit 5: Error in channel group 5 (Counter 1) Bit 6: Error in channel group 6 (Counter 2) Bit 7: Error in channel group 7 (Counter 3)
8	Diagnostic interrupt due to process interrupt lost at... Bit 0: ... input I+0.0 Bit 1: 0 (fix) Bit 2: ... input I+0.1 Bit 3: 0 (fix) Bit 4: ... input I+0.2 Bit 5: 0 (fix) Bit 6: ... input I+0.3 Bit 7: 0 (fix)
9	Diagnostic interrupt due to process interrupt lost at... Bit 0: ... input I+0.4 Bit 1: 0 (fix) Bit 2: ... input I+0.5 Bit 3: 0 (fix) Bit 4: ... input I+0.6 Bit 5: 0 (fix) Bit 6: ... input I+0.7 Bit 7: 0 (fix)
10	Diagnostic interrupt due to process interrupt lost at... Bit 0: ... input I+1.0 Bit 1: 0 (fix) Bit 2: ... input I+1.1 Bit 3: 0 (fix) Bit 4: ... input I+1.2 Bit 5: 0 (fix) Bit 6: ... input I+1.3 Bit 7: 0 (fix)

continued ...

... continue Record set 1

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

Chapter 6 Deployment CPU 31xS with Profibus

Outline

Content of this chapter is the deployment of the CPU 31xS with Profibus. After a short overview the project engineering and parameterization of a CPU 31xS with integrated Profibus-Part from VIPA is shown.

Further you get information about usage as DP master and DP slave of the Profibus part.

The chapter is ended with notes to commissioning and start-up.

The following chapter describes:

- Overview
- Deployment as master and slave
- Installation guidelines and commissioning

Content

Overview

Profibus-DP

Profibus is an international standard applicable to an open and serial fieldbus for building, manufacturing and process automation that can be used to create a low (sensor-/actuator level) or medium (process level) performance network of programmable logic controllers.

Profibus comprises an assortment of compatible versions. The following details refer to Profibus-DP.

Profibus-DP is a special protocol intended mainly for automation tasks in a manufacturing environment. DP is very fast, offers Plug'n'Play facilities and provides a cost-effective alternative to parallel cabling between PLC and remote I/O. Profibus-DP was designed for high-speed data communication on the sensor-actuator level.

The data transfer referred to as "Data Exchange" is cyclical. During one bus cycle, the master reads input values from the slaves and writes output information to the slave.

CPU with DP master

The Profibus DP master is to configure via the hardware configurator from Siemens. Therefore you have to choose the Siemens-CPU 318-2AJ00 in the hardware configurator from Siemens.

The transmission of your project engineering into the CPU takes place by means of MPI, MMC or Ethernet PG/OP channel. This is internally passing on your project data to the Profibus master part.

During the start-up the DP master automatically includes his data areas into the address range of the CPU. A project engineering in the CPU is not required.

As external storage medium the Profibus DP master uses the MMC (**M**ulti **M**edia **C**ard) together with the CPU.

Deployment of the DP-Master with CPU

Via the Profibus DP master up to 125 Profibus DP slaves may be coupled to the CPU. The DP master communicates with the DP slaves and links up its data areas with the address area of the CPU. There may be created maximal 1024Byte Input and 1024Byte Output data.

At every POWER ON res. overall reset the CPU fetches the I/O mapping data from the master. At DP slave failure, the ER-LED is on and the OB 86 is requested. If this is not available, the CPU switches to STOP and BASP is set. As soon as the BASP signal comes from the CPU, the DP master is setting the outputs of the connected periphery to zero. The DP master remains in the operating mode RUN independent from the CPU.

Profibus address 1 is reserved

Please regard that the Profibus address 1 is reserved for the system. The address 1 should not be used!

DP slave operation

For the deployment in an super-ordinated master system you first have to project your slave system as CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0) in *Slave* operation with configured in-/output areas. Afterwards you configure your master system. Assign your slave system to your master system by dragging the "CPU 31x" from the hardware catalog at *Configured stations* onto the master system, choose your slave system and connect it.

Project engineering CPU with integrated Profibus master

Outline

For the project engineering of the Profibus DP master you have to use the hardware manager from Siemens. Your Profibus projects are transferred via MPI to the CPU by means of the "PLC" functions. The CPU passes the data on to the Profibus DP master.

Preconditions

For the project engineering of the Profibus DP master in the CPU 31xSx/DPM the following preconditions have to be fulfilled:

- Siemens SIMATIC Manager has to be installed.
- With Profibus DP slaves of the Systems 100V, 200V and 300 from VIPA: GSD Files are included into the hardware configurator.
- There is a transfer possibility between configuration tool and CPU 31xS.

Note!

For the project engineering of the CPU and the Profibus DP master a thorough knowledge of the Siemens SIMATIC manager is required!

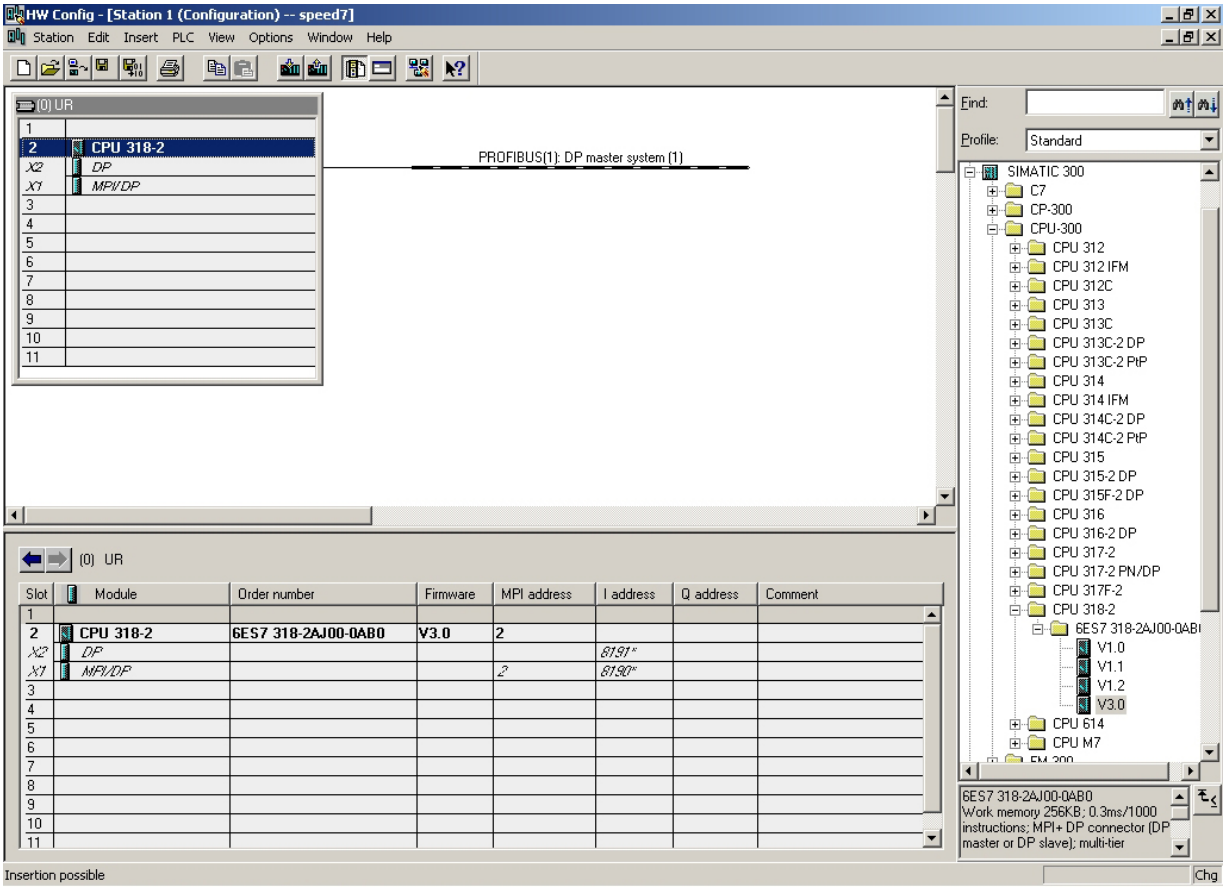
Install Siemens Hardware configurator

The hardware configurator is a part of the Siemens SIMATIC Manager. It serves the project engineering. The modules that may be configured here, 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 into the hardware catalog by means of the GSD file from VIPA.

Configure DP master

- Create a new project System 300.
- Add a profile rail from the hardware catalog.
- In the hardware catalog the CPU with Profibus master is listed as: Simatic300/CPU-300/CPU318-2DP/**6ES7 318-2AJ00-0AB0**
- Insert the **CPU 318-2DP** (**6ES7 318-2AJ00-0AB0**).
- Type the Profibus address of your master (e.g. :2).
- Click on DP, choose the operating mode "DP master" under *object properties* and confirm your entry with OK.
- Click on "DP" with the right mouse button and choose "add master system".
- Create a new Profibus subnet via NEW.



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

- For the project engineering of Profibus DP slaves you search the concerning Profibus-DP slave in the *hardware catalog* and drag&drop it in the subnet of your master.
- Assign a valid Profibus address to the DP slave.
- Link up the modules of your DP slave system in the plugged sequence and add the addresses that should be used by the modules.
- If needed, parameterize the modules.
- Save, compile and transfer your project. More detailed information about SPEED-Bus project engineering and project transfer may be found at chapter "Deployment CPU31xS".

Note!

If you deploy an IM153 from Siemens under a VIPA CPU 31xSx/DPM, please use the "compatible" DP slave modules.

These are listed in the hardware catalog under
`PROFIBUS-DP/Additional Field Devices/ Compatible Profibus-DP-Slaves.`

Slave operation possible

You may deploy your Profibus part from your SPEED7-CPU as DP slave. The approach is described on the following page.

Deployment as Profibus DP slave

Fast introduction

The deployment of the Profibus section as "intelligent" DP slave happens exclusively at master systems that may be configured in the Siemens SIMATIC manager. The following steps are required:

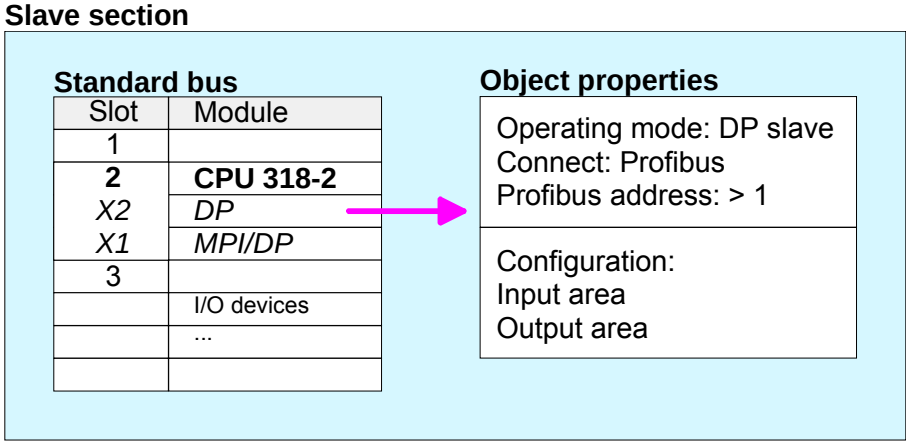
- Start the Siemens SIMATIC manager and configure a CPU 318-2DP with the operating mode *DP slave*.
- Connect to Profibus and configure the in-/output area for the slave section.
- Save and compile your project.
- Configure another station as CPU 318-2DP with operating mode *DP master.*
- Connect to *Profibus* and configure the in-/output ranges for the master section
- Save and compile your project.

In the following these steps are more detailed.

Project engineering of the slave section

- Start the Siemens SIMATIC manager with a new project.
- Insert a *SIMATIC 300 station* and name it as "...DP slave"
- Open the hardware configurator and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
  CPU 318-2DP (6ES7 318-2AJ00-0AB0 V3.0)
- Add your modules according to the real hardware assembly.
- Connect the CPU to *Profibus*, set a Profibus address >1 (preferably 3) and switch the Profibus section via *operating mode* to "slave operation".
- Via *Configuration* you define the in-/output address area of the slave CPU that shall be assigned to the DP slave.
- Save and compile your project

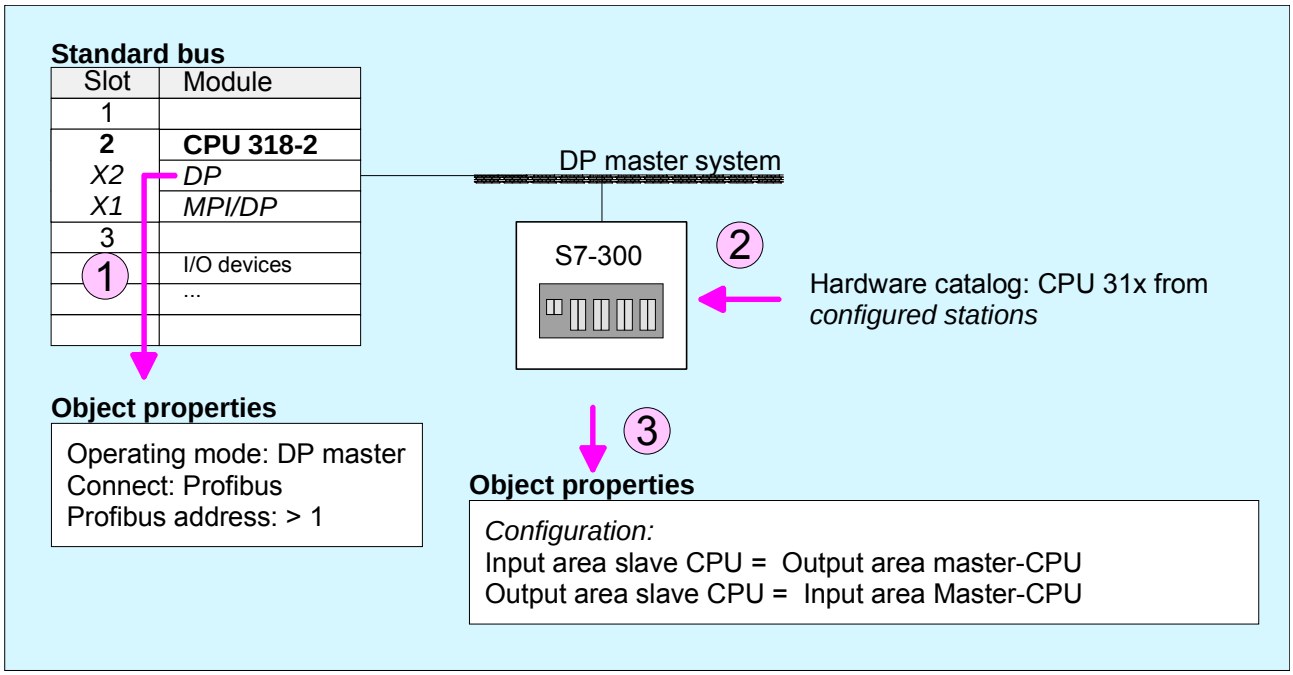
Slave section



Project engineering of the master section

- Insert another *SIMATIC 300 station* and name it as "...DP master".
- Open the hardware configurator and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
  CPU 318-2DP (6ES7 318-2AJ00-0AB0 V3.0)
- Add your modules according to the real hardware assembly.
- Connect the CPU to *Profibus*, set a Profibus address >1 (preferably 2) and switch the Profibus section via *operating mode* to "master operation"..
- Connect your slave system to the master system by dragging the "CPU 31x" from the hardware catalog at *configured stations* onto the master system and select your slave system.
- Open the *Configuration* at *Object properties* of your slave system.
- Via double click to the according configuration line you assign the according input address area on the master CPU to the slave output data and the output address area to the slave input data.
- Save, compile and transfer your project. More detailed information about SPEED-Bus project engineering and project transfer may be found at chapter "Deployment CPU31xS".

Master section



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 be deployed together with a CPU.
- Profibus supports max. 126 participants.
- Per segment a max. of 32 participants is permitted.
- The max. segment length depends on the baud 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.
- All participants are communicating with the same baudrate. The slaves adjust themselves automatically on the baudrate.
- The bus has to be terminated at both ends.
- Master and slaves are free combinable.

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.

Your VIPA CPU 31xS includes a 9pin slot where you connect the Profibus coupler into the Profibus network as a slave.

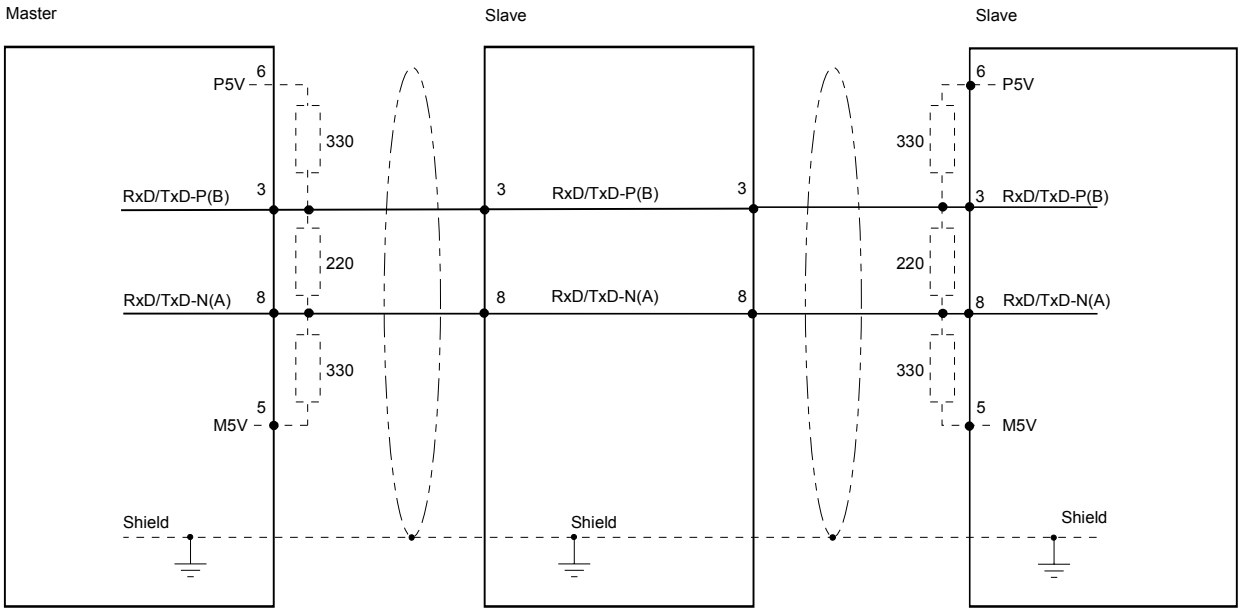
Max. 32 participants per segment are permitted. The segments are connected via repeaters. The maximum segment length depends on the transfer rate.

Profibus-DP uses a transfer rate between 9.6kBaud and 12MBaud, the slaves are following automatically. All participants are communicating with the same baudrate.

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.



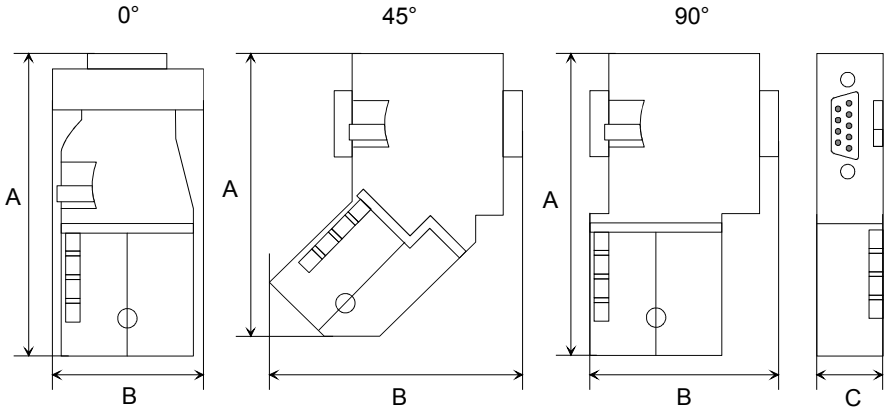
Note!

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

"EasyConn" Bus connector

In systems with more than two stations all partners are wired in parallel. For that purpose, the bus cable must be feed-through uninterrupted.

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.

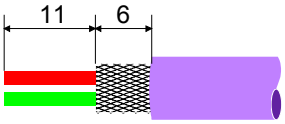


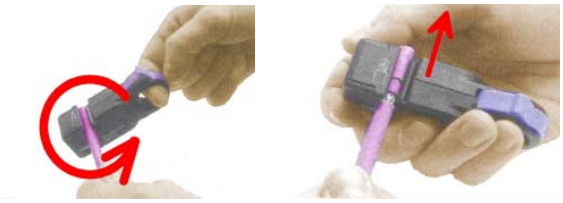
	0°	45°	90°
A	64	61	66
B	34	53	40
C	15,8	15,8	15,8

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

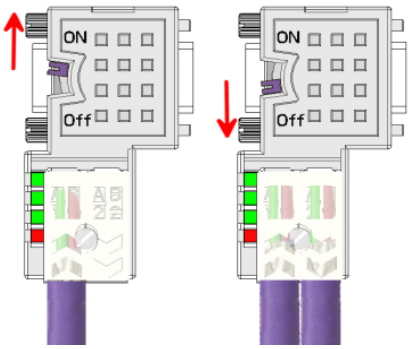




Dimensions in mm

Termination with "EasyConn"

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



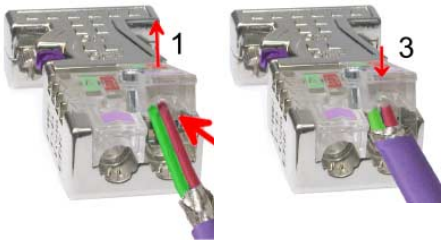
Attention!

The terminating resistor is only effective, if the connector is installed at a slave and the slave 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!

Commissioning and Start-up behavior

Start-up on delivery

In delivery the CPU is overall reset. The Profibus part is deactivated and its LEDs are off after Power ON.

Online with bus parameter without slave project

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

Slave configuration

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

CPU state controls DP master

After Power ON respectively a receipt of a new hardware configuration the configuration data and bus parameter were transferred to the DP master.

The DP master does not have any operation switch. His state is controlled by the RUN/STOP state of the CPU.

Dependent on the CPU state the following behavior is shown by the DP master:

Master behavior at CPU RUN

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

Master behavior at CPU STOP

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

Chapter 7 Deployment RS485 for PtP communication

Overview

Content of this chapter is the employment of the RS485 slot for serial PtP communication.

Here you'll find all information about the protocols, the activation and project engineering of the interface which are necessary for the serial communication using the RS485 interface.

The following description includes:

- Principles of the serial communication
- Usage of the protocols ASCII, STX/ETX, 3964R, USS and Modbus
- Switch RS485 interface to PtP usage
- Principals of the data transfer
- Project engineering of communication

Content

Fast introduction

General

You may de-activate the DP master integrated in the SPEED7-CPU via a hardware configuration and thus release the RS485 interface for PtP (**p**oint-**t**o-**p**oint) communication.

The RS485 interface supports in PtP operation the serial process connection to different source res. destination systems.

Protocols

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

Switch of RS485 for point-to-point operation

Per default, every CPU 31xS uses the RS485 interface for the Profibus-DP master. A hardware configuration allows you to switch the RS485 interface to point-to-point operation using *Object properties* and the parameter "Function RS485".

Parameterization

The parameterization happens during runtime using the SFC 216 (SER_CFG). For this you have to store the parameters in a DB for all protocols except ASCII.

Communication

The SFCs are controlling the communication. Send takes place via SFC 217 (SER_SND) and receive via SFC 218 (SER_RCV).

The repeated call of the SFC 217 SER_SND delivers a return value for 3964R, USS and Modbus via RetVal that contains, among other things, recent information about the acknowledgement of the partner station.

The protocols USS and Modbus allow to evaluate the receipt telegram by calling the SFC 218 SER_RCV after SER_SND.

The SFCs are included in the consignment of the CPU 31xS.

Overview SFCs for serial communication

The following SFCs are used for the serial communication:

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

Protocols and procedures

Overview

The CPU 31xS 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.

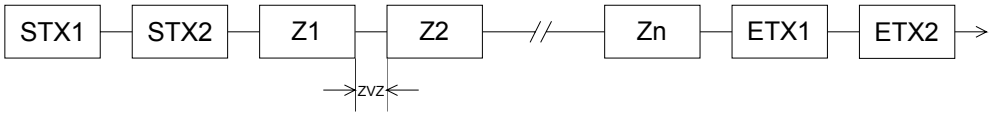
The STX/ETX procedure is suitable for the transfer of ASCII characters. It does not use block checks (BCC). Any data transferred from the periphery must be preceded by an Start followed by the data characters and the end character.

Depending of the byte width the following ASCII characters can be transfered: 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 PLC when the End has been received.

When data is send from the PLC 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-IDs.

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 PLC after a parameterizable character delay time (Timeout).

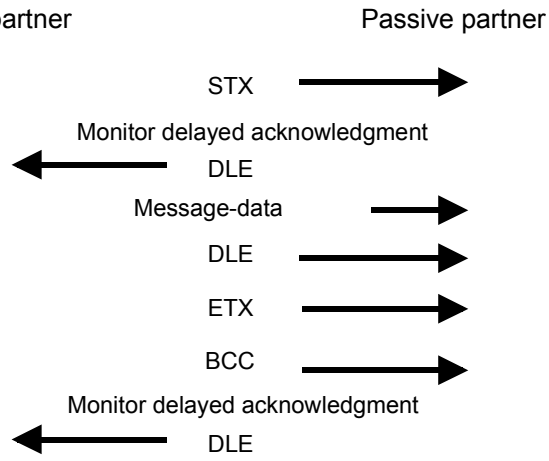
3964R

The 3964R procedure controls the data transfer of a point-to-point link between the CPU 31xS and a communication partner. The procedure adds control characters to the message data during data transfer. These control characters may be used by the communication partner to verify the complete and error free receipt.

The procedure employs the following control characters:

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

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

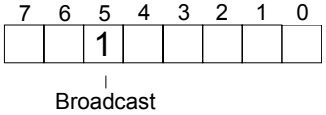
STX	LGE	ADR	PKE		IND		PWE		STW		HSW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

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

Broadcast with set Bit 5 in ADR-Byte



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

Supported Modbus protocols

The following Modbus Protocols are supported by the RS485 interface

- Modbus RTU Master
- Modbus ASCII Master

Deployment of RS485 interface for PtP

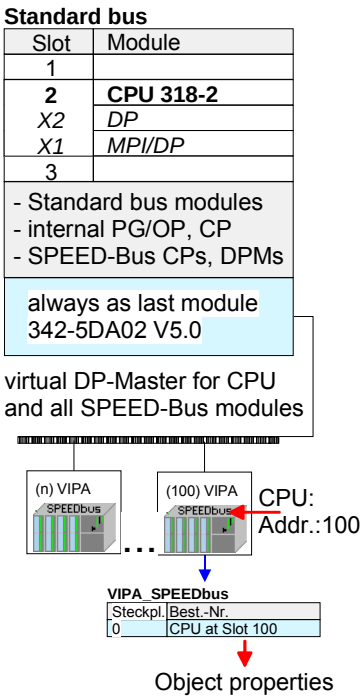
Outline

Per default, every CPU 31xS uses the RS485 interface for the Profibus-DP master. A hardware configuration allows you to switch the RS485 interface to point-to-point operation using *Object properties* and the parameter "Function RS485".

Switch to PtP operation

For the usage of the System 300S modules from VIPA at the SPEED-Bus, the inclusion of the System 300S modules in the hardware catalog is required using the GSD file from VIPA.

The switch to PtP operation has the following approach:



- Start the hardware configurator from Siemens and include the speedbus.gsd for SPEED7 from VIPA.
- Configure CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0) from Siemens.
- Starting with slot 4, place the System 300 modules at the standard bus.
- Place and connect below this modules the CPs (internal) and then SPEED-Bus-CPs and DP master.
- For the SPEED-Bus, include and connect the Siemens DP master CP 342-5 (342-5DA02 V5.0) always as last module and parameterize it to the *operating mode* DP master. For this, the Profibus address must be different from the range 100...116. This master system is used to connect every SPEED-Bus module as VIPA_SPEEDbus slave. Here, the **Profibus address** is equivalent to the **slot no.** starting with 100 for the CPU. Place the concerning module at slot 0 of every slave and alter if needed the parameters.
- Place your SPEED7-CPU with Profibus address 100 at slot 0.
- Set in *Function RS485* "PtP" in the object properties.
- Place now the further SPEED-Bus modules as VIPA_SPEEDbus slave.

After transferring your project to the CPU together with your PLC application, the RS485 interface is after the boot sequence available for PtP communication.

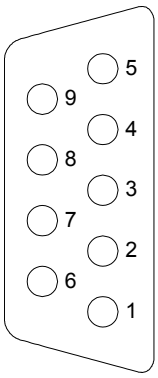
Note!

More detailed information about SPEED-Bus project engineering and project transfer is in the chapter “Employment CPU31xS"!

Properties RS485

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

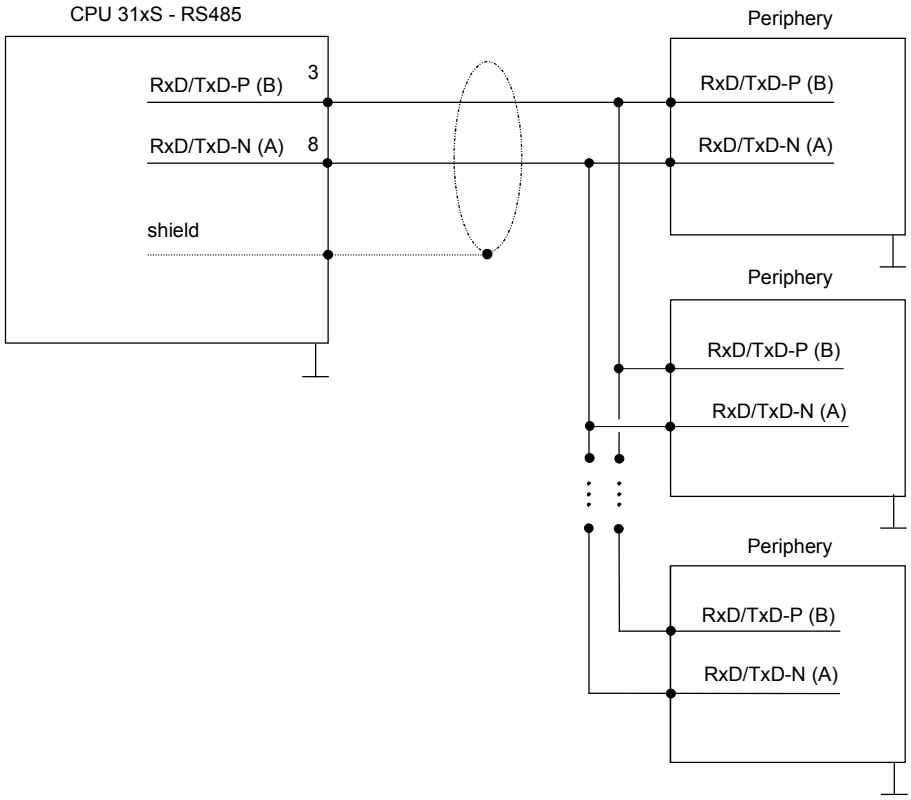
Connection RS485



9pin jack

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

Connection



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

Data that is into the according data channel by the PLC, is stored in a FIFO send buffer (**f**irst **i**n **f**irst **o**ut) with a size of 2x1024Byte and then put out via the interface.

When the interface receives data, this is stored in a FIFO receive buffer with a size of 2x1024Byte and can there be read by the PLC.

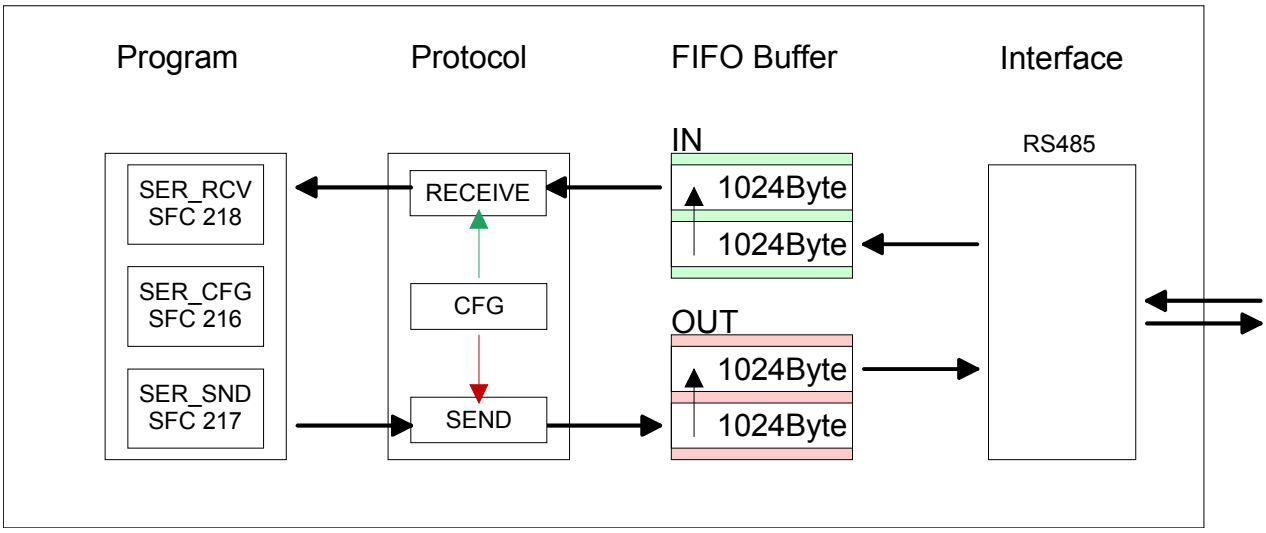
If the data is transferred via a protocol, the adoption of the data to the according protocol happens automatically.

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 31xS - RS485 PtP communication



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.

Name	Declaration	Type	Comment
Protocol	IN	BYTE	1=ASCII, 2=STX/ETX, 3=3964R
Parameter	IN	ANY	Pointer to protocol-parameters
Baudrate	IN	BYTE	Velocity of data transfer
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)

Parameter description

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

Example: Wanted time 8ms at a baudrate of 19200Baud
Calculation: 19200Bit/s x 0,008s ≈ 154Bit → (9Ah)
Hex value is 9Ah.

Protocol

Here you fix the protocol to be used. You may choose between:
1: ASCII
2: STX/ETX
3: 3964R
4: USS Master
5: Modbus RTU Master
6: Modbus ASCII Master

Parameter (as DB)

At ASCII protocol, this parameter is ignored.

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

Data block at STX/ETX

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

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	(Character delay time)
DBW6:	ConfTimeout	WORD	(Acknowledgement delay time)

Data block at USS

DBW0:	Timeout	WORD	(Delay time in)

Data block at Modbus-Master

DBW0:	Timeout	WORD	(Respond delay time)

Baud rate

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

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

CharLen

Number of data bits where a character is mapped to.

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

Parity

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

0: NONE 1: ODD 2: EVEN

StopBits

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

1: 1Bit 2: 1.5Bit 3: 2Bit

FlowControl

The parameter FlowControl is ignored. When sending RST=0, when receiving RST=1.

RetVal SFC 216 (Error message SER_CFG)

Return values send by the block:

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

Communication

Overview

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

The SFCs are included in the consignment of the CPU 31xS.

SFC 217 (SER_SND)

This block sends data via the serial interface.

The repeated call of the SFC 217 SER_SND delivers a return value for 3964R, USS and Modbus via RetVal that contains, among other things, recent information about the acknowledgement of the partner station.

The protocols USS and Modbus require to evaluate the receipt telegram by calling the SFC 218 SER_RCV after SER_SND.

Parameter

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 the sent Bytes is stored.

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.

With **STX/ETX**, **3964R**, **Modbus** and **USS** always the length set in DataPtr is stored or 0.

RetVal SFC 217 (Error message SER_SND)

Return values of the block:

Error code	Description
0000h	Send data - ready
1000h	Nothing sent (data length 0)
20xxh	Protocol executed error free with xx bit pattern for diagnosis
7001h	Data is stored in internal buffer - active (busy)
7002h	Transfer - active
80xxh	Protocol executed with errors with xx bit pattern for diagnosis (no acknowledgement by partner)
90xxh	Protocol not executed with xx bit pattern for diagnosis (no acknowledgement by partner)
8x24h	Error in SFC parameter x, where x: 1: Error in "DataPtr" 2: Error in "DataLen"
8122h	Error in parameter "DataPtr" (e.g. DB too short)
807Fh	Internal error
809Ah	Interface not found or interface is 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 (>1024Byte)
9002h	Data too short (0Byte)
9004h	Character not allowed

3964R

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

... Continue RetVal SFC 217 SER_SND

USS

Error code	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 (>1024Byte)
9002h	Data too short (<2Byte)

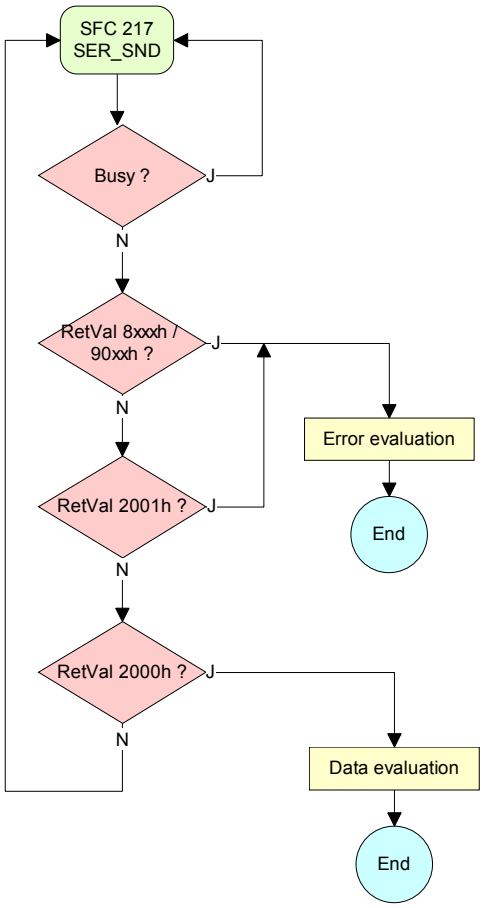
Modbus RTU/ASCII Master

Error code	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 (>1024Byte)
9002h	Data too short (<2Byte)

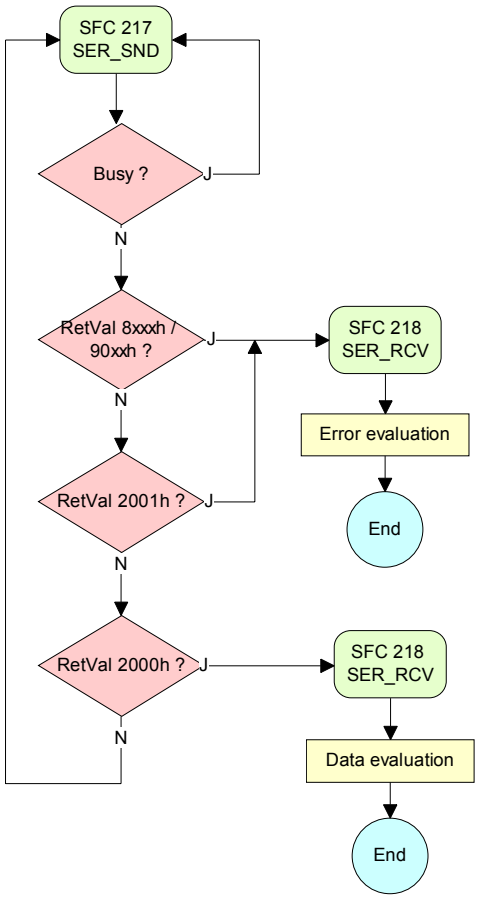
Principles of programming

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

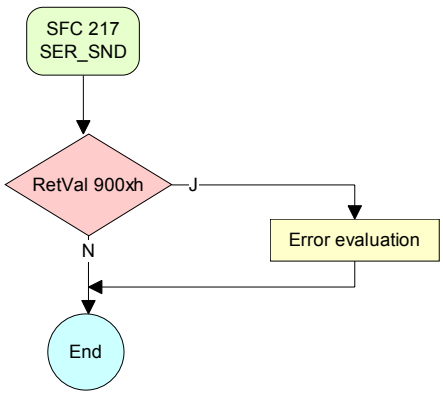
3964R



USS / Modbus



ASCII / STX/ETX



SFC 218 (SER_RCV)

This block receives data via the serial interface.

Using the SFC 218 SER_RCV after SER_SND with the protocols USS and Modbus the acknowledgement telegram can be read.

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

This word gets an entry in case of an error. The following error messages may be created depending on the protocol:

ASCII

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

STX/ETX

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

3964R / Modbus RTU/ASCII Master

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

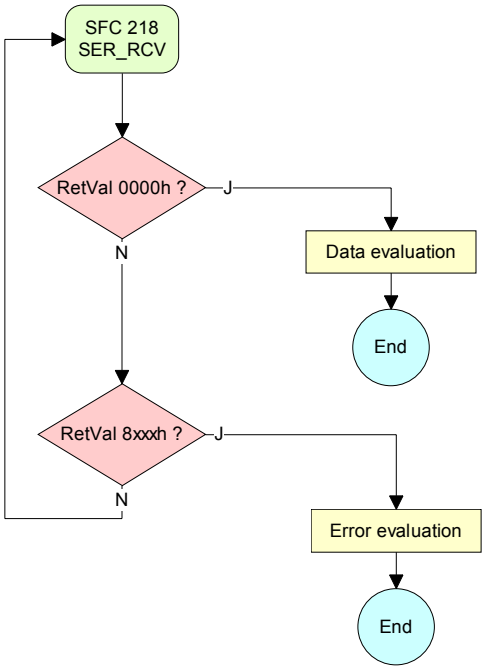
RetVal SFC 218 (Error message SER_RCV)

Return values of the block:

Error code	Description
0000h	no error
1000h	Receive buffer too small (data loss)
8x24h	Error at SFC-Parameter x, with x: 1: Error at "DataPtr" 2: Error at "DataLen" 3: Error at "Error"
8122h	Error in parameter "DataPtr" (e.g. DB too short)
809Ah	Serial interface not found res. interface is used by Profibus
809Bh	Serial interface not configured

Principles of programming

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



Chapter 8 Deployment CPU 31xS with TCP/IP

Overview

The following chapter describes the deployment of the CPU 31xSN/NET and the communication using TCP/IP. Please regard the chapter "Fast introduction" where you find all information compressed required for the project engineering of the CPU 31xS with *CP 343*. After the fast introduction, the mentioned steps are described in detail.

The following text describes:

- Basics about a Twisted-Pair network
- Access to Ethernet PG/OP channel and website
- Project engineering of a CP communication
- ORG format for the communication with other systems
- Intelligent Process communication
- Sample

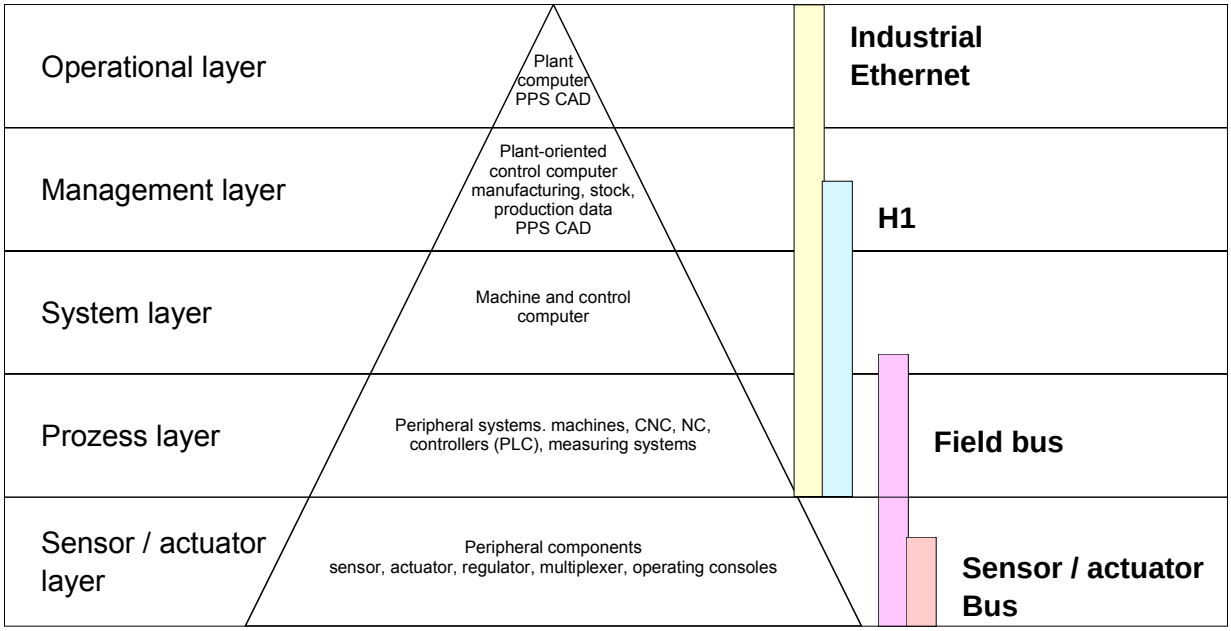
Content

Industrial Ethernet in automation

Overview

The flow of information in a company presents a vast spectrum of requirements that must be met by the communication systems. Depending on the area of business the bus system or LAN must support a different number of users, different volumes of data must be transferred and the intervals between transfers may vary, etc.

It is for this reason that different bus systems are employed depending on the respective task. These may be subdivided into different classes. The following model depicts the relationship between the different bus systems and the hierarchical structures of a company:



Industrial Ethernet

Industrial Ethernet is an electrical net based on shielded twisted pair cabling or optical net based on optical fiber.

Industrial Ethernet is defined by the international standard IEEE 802.3. The net access of Industrial Ethernet corresponds to IEEE 802.3 - CSMA/CD (**C**arrier **S**ense **M**ultiple **A**ccess/**C**ollision **D**etection) scheme: every station "listens" on the bus cable and receives communication messages that are addressed to it.

Stations will only initiate a transmission when the line is unoccupied. In the event that two participants should start transmitting simultaneously, they will detect this and stop transmitting to restart after a random delay time has expired.

Using switches there is the possibility for communication without collisions.

ISO/OSI reference model

Overview

The ISO/OSI reference model is based on a proposal that was developed by the International Standards Organization (ISO). This represents the first step towards an international standard for the different protocols. It is referred to as the ISO-OSI layer model. OSI is the abbreviation for **O**pen **S**ystem **I**nterconnection, the communication between open systems. The ISO/OSI reference model does not represent a network architecture as it does not define the services and protocols used by the different layers. The model simply specifies the tasks that the different layers must perform.
All current communication systems are based on the ISO/OSI reference model which is defined by the ISO 7498 standard. The reference model structures communication systems into 7 layers that cover different communication tasks. In this manner the complexity of the communication between different systems is divided amongst different layers to simplify the task.
The following layers have been defined:

Layer	Function
Layer 7	Application Layer
Layer 6	Presentation Layer
Layer 5	Session Layer
Layer 4	Transport Layer
Layer 3	Network Layer
Layer 2	Data Link Layer
Layer 1	Physical Layer

Depending on the complexity and the requirements of the communication mechanisms a communication system may use a subset of these layers.

Layers

Layer 1 Bit communication layer (physical layer)

The bit communication layer (physical layer) is concerned with the transfer of data bits via the communication channel. This layer is therefore responsible for the mechanical, electrical and the procedural interfaces and the physical communication medium located below the bit communication layer:

- Which voltage represents a logical 0 or a 1?
- The minimum time that the voltage be present to be recognized as a bit.
- The pin assignment of the respective interface.

Layer 2 Security layer (data link layer)

This layer performs error-checking functions for bit strings transferred between two communicating partners. This includes the recognition and correction or flagging of communication errors and flow control functions.

The security layer (data link layer) converts raw communication data into a sequence of frames. This is where frame limits are inserted on the transmitting side and where the receiving side detects them. These limits consist of special bit patterns that are inserted at the beginning and at the end of every frame. The security layer often also incorporates flow control and error detection functions.

The data security layer is divided into two sub-levels, the LLC and the MAC level.

The MAC (**M**edia **A**ccess **C**ontrol) is the lower level and controls how senders are sharing a single transmit channel.

The LLC (**L**ogical **L**ink **C**ontrol) is the upper level that establishes the connection for transferring the data frames from one device into the other.

Layer 3 Network layer

The network layer is an agency layer.

Business of this layer is to control the exchange of binary data between stations that are not directly connected. It is responsible for the logical connections of layer 2 communication. Layer 3 supports the identification of the single network addresses and the establishing and disconnecting of logical communication channels.

Additionally, layer 3 manages the prior transfer of data and the error processing of data packets. IP (**I**nternet **P**rotocol) is based on Layer 3.

Layer 4 Transport layer

Layer 4 connects the network structures with the structures of the higher levels by dividing the messages of higher layers into segments and pass them on to the network layer. Hereby, the transport layer converts the transport addresses into network addresses.

Common transport protocols are: TCP, SPX, NWLink and NetBEUI.

Layers continued...

Layer 5 Session layer

The session layer is also called the communication control layer. It relieves the communication between service deliverer and the requestor by establishing and holding the connection if the transport system has a short time fail out.

At this layer, logical users may communicate via several connections at the same time. If the transport system fails, a new connection is established if needed.

Additionally this layer provides methods for control and synchronization tasks.

Layer 6 Presentation layer

This layer manages the presentation of the messages, when different network systems are using different representations of data.

Layer 6 converts the data into a format that is acceptable for both communication partners.

Here compression/decompression and encrypting/decrypting tasks are processed.

This layer is also called interpreter. A typical use of this layer is the terminal emulation.

Layer 7 Application layer

The application layer is the link between the user application and the network. The tasks of the application layer include the network services like file, print, message, data base and application services as well as the according rules.

This layer is composed from a series of protocols that are permanently expanded following the increasing needs of the user.

Principles

Network (LAN)

A network res. LAN (**L**ocal **A**rea **N**etwork) provides a link between different stations that enables them to communicate with each other.

Network stations consist of PCs, IPCs, TCP/IP adapters, etc.

Network stations are separated by a minimum distance and connected by means of a network cable. The combination of network stations and the network cable represent a complete segment.

All the segments of a network form the Ethernet (physics of a network).

Twisted Pair

In the early days of networking the Triaxial- (yellow cable) or thin Ethernet cable (Cheapernet) was used as communication medium. This has been superseded by the twisted-pair network cable due to its immunity to interference. The CPU 31xSN/NET module has a twisted-pair connector.

The twisted-pair cable consists of 8 cores that are twisted together in pairs. Due to these twists this system is provides an increased level of immunity to electrical interference. For linking please use twisted pair cable which at least corresponds to the category 5.

Where the coaxial Ethernet networks are based on a bus topology the twisted-pair network is based on a point-to-point scheme.

The network that may be established by means of this cable has a star topology. Every station is connected to the star coupler (hub/switch) by means of a separate cable. The hub/switch provides the interface to the Ethernet.

Hub (repeater)

The hub is the central element that is required to implement a twisted-pair Ethernet network.

It is the job of the hub to regenerate and to amplify the signals in both directions. At the same time it must have the facility to detect and process segment wide collisions and to relay this information. The hub is not accessible by means of a separate network address since it is not visible to the stations on the network.

A hub has provisions to interface to Ethernet or to another hub res. switch.

Switch

A switch also is a central element for realizing Ethernet on Twisted Pair. Several stations res. hubs are connected via a switch. Afterwards they are able to communicate with each other via the switch without interfering the network. An intelligent hardware analyzes the incoming telegrams of every port of the switch and passes them collision free on to the destination stations of the switch. A switch optimizes the bandwidth in every connected segment of a network. Switches enable exclusive connections between the segments of a network changing at request.

Protocols

Overview

Protocols define a set of instructions or standards that enable computer to establish communication connections and exchange information as error free as possible. A commonly established protocol for the standardization of the complete computer communication is the so called ISO/OSI layer model, a model based upon seven layers with rules for the usage of hardware and software (see ISO/OSI reference model above).

The CPU 31xSN/NET from VIPA uses the following protocols

- TCP/IP
- UDP
- RFC1006 (ISO on TCP)

The protocols are described in the following:

TCP/IP

TCP/IP protocols are available on all major systems. At the bottom end this applies to simple PCs, through to the typical mini-computer up to mainframes.

For the wide spread of internet accesses and connections, TCP/IP is often used to assemble heterogeneous system pools.

TCP/IP, standing for **T**ransmission **C**ontrol **P**rotocol and **I**nternet **P**rotocol, collects a various range of protocols and functions.

TCP and IP are only two of the protocols required for the assembly of a complete architecture. The application layer provides programs like "FTP" and "Telnet" for the PC.

The application layer of the Ethernet part of the CPU 31xSN/NET is defined with the user application using the standard handling blocks.

These user applications use the transport layer with the protocols TCP and UDP for the data transfer which themselves communicate via the IP protocol with the internet layer.

IP

The internet protocol covers the network layer (Layer 3) of the ISO/OSI layer model.

The purpose of IP is to send data packages from on PC to another passing several other PCs. These data packages are referred to as datagrams. The IP doesn't guarantee the correct sequence of the datagrams nor the delivery at the receiver.

For the unambiguous identification between sender and receiver at *IPv4* 32Bit addresses (IP addresses) are used that are written as four octets (exactly 8Bit), e.g. 172.16.192.11.

These internet addresses are defined and assigned worldwide from the DDN network (Defense Department Network), thus every user may communicate with all other TCP/IP users.

One part of the address specifies the network, the rest serves the identification of the participants inside the network. The boarder between the network and the host part is variable and depends on the size of the network.

To save IP addresses, so called *NAT router* are used that have one official IP address and cover the network. Then the network can use any IP address.

TCP

The TCP (**T**ransmission **C**ontrol **P**rotocol) bases directly on the IP and thus covers the transport layer (layer 4) of the OSI layer model. TCP is a connection orientated end-to-end protocol and serves the logic connection between two partners.

TCP guarantees the correct sequence and reliability of the data transfer. Therefore you need a relatively large protocol overhead that slows down the transfer speed.

Every datagram gets a header of at least 20Byte. This header also contains a sequence number identifying the series. This has the consequence that the single datagrams may reach the destination on different ways through the network.

Using TCP connections, the whole data length is not transmitted. This means that the recipient has to know how many bytes belong to a message. To transfer data with variable length you may begin the user data with the length information and evaluate this at the counter station.

Properties

- Besides of the IP address ports are used for the addressing. A port address should be within the range of 2000...65535. Partner and local ports may only be identical at one connection.
- Not depending on the used protocol, the PLC needs the VIPA handling blocks AG_SEND (FC 5) and AG_RECV (FC 6) for data transfer.

UDP

The UDP (**U**ser **D**atagram **P**rotocol) is a connection free transport protocol. It has been defined in the RFC768 (Request for Comment). Compared to TCP, it has much fewer characteristics.

The addressing happens via port numbers.

UDP is a fast unsafe protocol for it doesn't care about missing data packages nor about their sequence.

ISO-on-TCP RFC1006

The TCP transport service works stream orientated. This means that data packages assembled by the user not necessarily have to receive the partner in the same packaging. Depending on the data amount, packages may though come in in the correct sequence but differently packed. This causes that the recipient may not recognize the package borders anymore. For example you may send 2x 10Byte packages but the counter station receives them as 20Byte package. But for most of the applications the correct packaging is important.

Due to this you need another protocol above TCP. This purpose is defined in the protocol RFC1006. The protocol definition describes the function of an ISO transport interface (ISO 8072) basing upon the transport interface TCP (RFC793).

The basic protocol of RFC1006 is nearly identical to TP0 (Transport Protocol, Class 0) in ISO 8073.

For RFC1006 is run as protocol for TCP, the decoding takes place in the data section of the TCP package.

Properties

- Contrary to TCP the receipt of data is confirmed by a TCP layer.
- Instead of ports TSAPs are used for the addressing besides of the IP address. The TSAP length may be 1 ... 16 characters. The entry may happen in ASCII or Hex format. Foreign and local TSAPs may only be identical at 1 connection.
- Not depending on the used protocol the VIPA handling blocks AG_SEND (FC 5) and AG_RECEIVE (FC 6) are necessary for data transfer.
- Contrary to TCP different telegram lengths can be received using RFC1006.

IP address and subnet

IP address structure

Industrial Ethernet exclusively supports IPv4. At IPv4 the IP address is a 32Bit address that must be unique within the network and consists of 4 numbers that are separated by a dot.

Every IP address is a combination of a **Net-ID** and a **Host-ID** and its structure is as follows: **XXX.XXX.XXX.XXX**

Range: 000.000.000.000 to 255.255.255.255

The network administrator also defines IP addresses.

Net-ID
Host-ID

The **Net**work-ID identifies a network res. a network controller that administrates the network.

The Host-ID marks the network connections of a participant (host) to this network.

Subnet mask

The Host-ID can be further divided into a **Subnet-ID** and a *new* **Host-ID** by using an bit for bit AND assignment with the **Subnet mask**.

The area of the original Host-ID that is overwritten by 1 of the Subnet mask becomes the Subnet-ID, the rest is the new Host-ID.

Subnet mask	binary all "1"		binary all "0"
IPv4 address	Net-ID	Host-ID	
Subnet mask and IPv4 address	Net-ID	Subnet-ID	*new* Host-ID

Subnet

A TCP-based communication via point-to-point, hub or switch connection is only possible between stations with identical Network-ID and Subnet-ID! Different area must be connected with a router.

The subnet mask allows you to sort the resources following your needs. This means e.g. that every department gets an own subnet and thus does not interfere another department.

Address at first start-up

At the first start-up of a CPU 31xSN/NET, Ethernet PG/OP channel and CP 343 part of the CPU 31xSN/NET do not have an IP address. The assignment takes place using the following possibilities:

- Using Siemens SIMATIC Manager switch PG/PC interface to "TCP/IP... RFC1006". Via "Assign Ethernet address" search the appropriate CP and assign IP parameters. After that the CP is directly assigned to the new IP parameters without any restart of the CPU.
- You may assign an IP address and a subnet mask to your CP with the help of a "minimum project" and transfer this via MMC or MPI into the CPU. After a reboot of the CPU and after switching the PG/PC interface to "TCP/IP... RFC1006" you may now configure your CPU online via the favored CP.

Address classes

For IPv4 addresses there are five address formats (class A to class E) that are all of a length of 4byte = 32bit.

Class A	0	Network-ID (1+7bit)	Host-ID (24bit)
Class B	10	Network-ID (2+14bit)	Host-ID (16bit)
Class C	110	Network-ID (3+21bit)	Host-ID (8bit)
Class D	1110	Multicast group	
Class E	11110	Reserved	

The classes A, B and C are used for individual addresses, class D for multicast addresses and class E is reserved for special purposes.

The address formats of the 3 classes A, B, C are only differing in the length of Network-ID and Host-ID.

Private IP networks

To build up private IP-Networks within the internet, RFC1597/1918 reserves the following address areas:

Network class	Start IP	End IP	Standard subnet mask
A	10.0.0.0	10.255.255.255	255.0.0.0
B	172.16.0.0	172.31.255.255	255.255.0.0
C	192.168.0.0	192.168.255.255	255.255.255.0

(The Host-ID is underlined.)

These addresses can be used as net-ID by several organizations without causing conflicts, for these IP addresses are neither assigned in the internet nor are routed in the internet.

Reserved Host-Ids

Some Host-IDs are reserved for special purposes.

Host-ID = "0"	Identifier of this network, reserved!
Host-ID = maximum (binary complete "1")	Broadcast address of this network

Note!

Never choose an IP address with Host-ID=0 or Host-ID=maximum!

(e.g. for class B with subnet mask = 255.255.0.0, the "172.16.0.0" is reserved and the "172.16.255.255" is occupied as local broadcast address for this network.)

Network planning

Standards and guidelines

The applicable rules and regulations have to be satisfied in order to establish reliable communications between the different stations.

These agreements define the form of the data protocol, the method of bus access and other principles that are important for reliable communications.

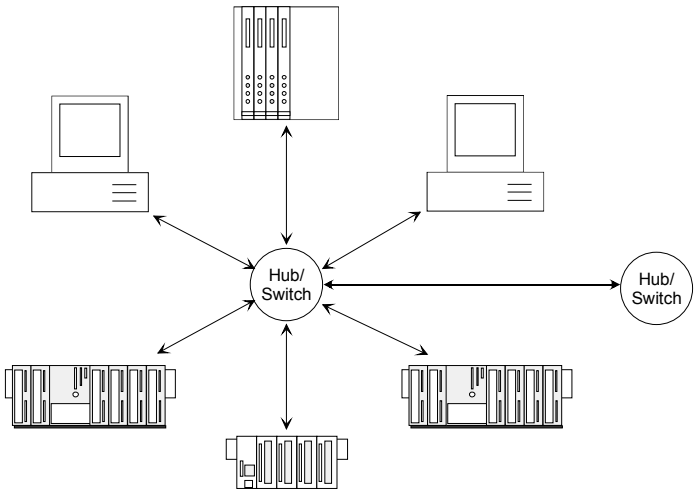
The VIPA CPU 31xSN/NET was developed in accordance with the standards defined by ISO.

International and national committees have defined the following standards and guidelines for networking technologies:

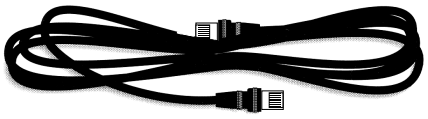
ANSI	American National Standards Institute The ANSI X3T9.5 standard currently defines the provisions for high-speed LANs (100 MB/s) based on fiber optic technology. (FDDI) Fiber Distributed Data Interface.
CCITT	Committee Consultative Internationale de Telephone et Telegraph. Amongst others, this advisory committee has produced the provisions for the connection of industrial networks (MAP) to office networks (TOP) on Wide Area Networks (WAN).
ECMA	European Computer Manufacturers Association. Has produced various MAP and TOP standards.
EIA	Electrical Industries Association (USA) This committee has issued standard definitions like RS232 (V.24) and RS511.
IEC	International Electrotechnical Commission. Defines certain special standards, e.g. for the Field Bus.
ISO	International Organization for Standardization. This association of national standards organizations developed the OSI-model (ISO/TC97/SC16). It provides the framework for the standardization of data communications. ISO standards are included in different national standards like for example UL and DIN.
IEEE	Institute of Electrical and Electronic Engineers (USA). The project-group 802 determines LAN-standards for transfer rates of 1 to 1000MB/s. IEEE standards often form the basis for ISO-standards, e.g. IEEE 802.3 = ISO 8802.3.

Overview of components

The CP is exclusively used for deployment in a Twisted-Pair network. Within a Twisted-Pair network all participating stations are connected in star topology via a Twisted-Pair cable to a hub/switch which is also able to communicate with another hub/switch. Two connected stations are building a segment where the length of the Twisted-Pair cable between two stations must be max. 100m.



Twisted Pair Cable



At twisted pair cable has 8 conductors twisted together in pairs.

The different conductors have a diameter of 0.4 to 0.6mm. For linking please use twisted pair cable which at least corresponds to the category 5.

Analyzing the requirements

- What is the size of the area that must be served by the network?
- How many network segments provide the best solution for the physical (space, interference related) conditions encountered on site?
- How many network stations (SPS, IPC, PC, transceiver, bridges if required) must be connected to the cable?
- What is the distance between the different stations on the network?
- What is the expected "growth rate" and the expected number of connections that must be catered for by the system?
- What data amount has to be handled (band width, accesses/sec.)?

Drawing a network diagram

Draw a diagram of the network. Identify every hardware item (i.e. station cable, hub, switch). Observe the applicable rules and restrictions.

Measure the distance between all components to ensure that the maximum length is not exceeded.

Linking with NetPro

Please regard that the following software packages must be installed for the project engineering:

- Siemens SIMATIC Manager V. 5.1
- For the project engineering of SPEED7 modules the vipa_speedbus.gsd is included.
- Siemens SIMATIC NET

To enable the stations to communicate with each other you have to configure the required (sub)nets in the Siemens SIMATIC Manager res. NetPro following this approach:

- Create one or more subnets of the wanted type in your project.
- Adjust the properties of the subnets.
- Connect your participants logically to the subnet.
- Establish communication connections between the single stations.

Net-Project variants

You may administrate several subnets in one project. Every station has to be created once. A station may be assigned to several subnets by assigning the CPs accordingly.

In the following typical project variants for networks are listed:

1 subnet - 1 project

The simplest case is a plant with stations that have to be connected via one subnet of the type Industrial Ethernet.

For this you create an object "Ethernet". Stations that are created in the same project refer to this object when they are configured as net knots. They may then be selected directly. Foreign devices are listed in this subnet as "Other station" during project engineering.

2 or more subnets - 1 project

Due to different tasks of the stations or due to the expansion of your plant it may be necessary to create several nets. Here you may create several subnets in one project and configure the stations easily for communication.

1 or more subnets – several part projects

At complex linked plants it is sensible to administrate plant parts in several part projects. Here it may be necessary to create project exceeding connections. For this the Siemens SIMATIC Manager starting with V. 5.2 provides the multi project function. This function allows you to split projects and join them again. A more detailed description is to be found in the manual of the Siemens SIMATIC Manager.

Subnet exceeding connections

These are connections that long into another subnet due to the complexity of the plant. The subnets are connected via a router. By setting a router address during the hardware configuration of your CP you may instruct the CP to include the according subnet via this router for communication.

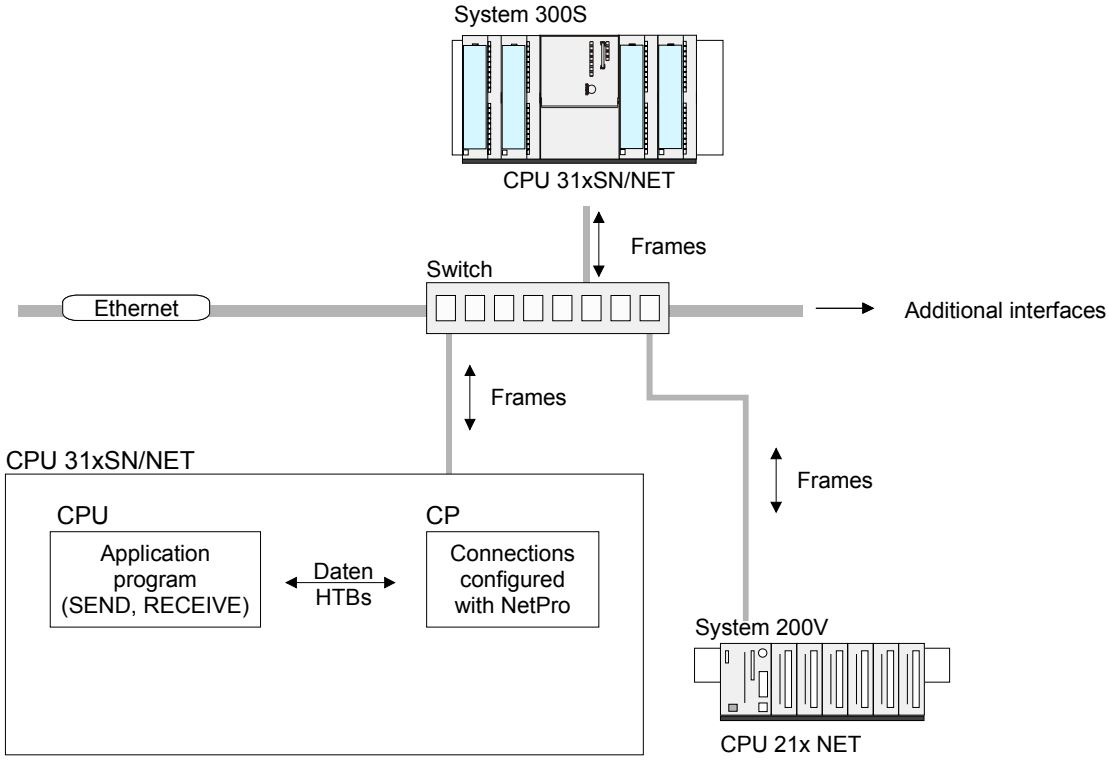
Communication possibilities of the CP

Communication between CP 343 and CPU

The internal CP 343 of the CPU 31xSN/NET is directly connected to the CPU 31xS via a Dual-Port-RAM. The CPU manages the data exchange with the VIPA handling blocks AG_SEND (FC 5) and AG_RECV (FC 6).

The communication via the according protocols are controlled by connections that are parameterized in the Siemens project engineering tool NetPro and that may be transferred into the CPU via MMC, MPI or directly via Ethernet.

For the transfer via Ethernet, your CP must be connected to Ethernet with valid IP parameters. The assignment takes place either using the corresponding menu item of the Siemens SIMATIC Manager or via a minimum project where the IP parameters are defined. This project can be transferred to the CPU via MMC or MPI.



Communication types

The CP supports the following communication types:

- PG/OP communication
- Configurable connections

PG/OP communication

The PG/OP communication serves the loading of programs and configuration data, for test and diagnostic functions as well as for operating and monitoring a plant. Here you may access the CPU online via the CP (Ethernet).

With CP firmware version 1.7.4 and up a simultaneous access of up to 32 participants is possible. Please note for each PG and OP communication 1 connection is reserved.

Configurable connections

Configurable connections are connections for the communication between PLC stations. The connections may be configured with the Siemens project engineering tool NetPro.

The following table shows the combination option with the different operating modes:

Combination options

Connection partner	Connection type	Conn. Establ.	Connection	Operating mode
Specified in NetPro (in recent project)	TCP / ISO-on-TCP	active/passive	specified	SEND/RECEIVE
	UDP	-		
Unspecified in NetPro (in recent project)	TCP / ISO-on-TCP	active	specified	SEND/RECEIVE
		passive	part specified (Port)	SEND/RECEIVE FETCH PASSIV WRITE PASSIV
			unspecified	
	UDP	-	specified	SEND/RECEIVE
Unspecified in NetPro (in "unknown project")	TCP / ISO-on-TCP	active	unspecified (connection name)	SEND/RECEIVE
		passive	unspecified (connection name)	SEND/RECEIVE FETCH PASSIV WRITE PASSIV
	UDP	-	unspecified (connection name)	SEND/RECEIVE
All Broadcast stations	UDP	-	specified (Port, Broadcast addr)	SEND
All Multicast stations	UDP	-	specified (Port, Multicast group)	SEND/RECEIVE

Connection partner

Connection partner are stations at the counter side.

Specified connection partner

Every station configured in the Siemens SIMATIC Manager is entered in the list of connection partners. By setting an IP address and a subnet mask these stations are uniquely *specified*.

Unspecified connection partner

You may also set an *unspecified* connection partner. Here the connection partner may be within the *recent project* or within an *unknown project*. Connection commands to an *unknown project* must be defined via an unique connection name that has to be used in the projects of both stations. Due to the assignment via a connection name, the connection itself remains *unspecified*.

All broadcast participants

Only with UDP connections you may here send messages to all available broadcast participants. The reception is not possible. The broadcast participants are specified via one port and one broadcast address at sender and receiver.

All multicast participants

This setting allows you to send and receive multicast telegrams between the multicast participants. By setting of one port and one multicast group for sender and receiver the multicast participants are specified.

Connection types

For the communication the following connection types are available:

- **TCP** res. **ISO-on-TCP** for the secured data transfer of related data blocks between two Ethernet participants.
- **UDP** for the unsecured data transfer of related data blocks between two Ethernet.

Connection establishment

Using configurable connections there is always one station that *actively* establishes a connection. The counter station waits *passively* for the active connection. Only then productive data can be transferred.

Connection

By setting IP address and port/TSAP of the counter station, a connection is *specified*. Active connections must always be set specified. An *unspecified* connection which is only possible for passive connection establishment, IP address and port/TSAP of the counter station are not required for telegram evaluation.

There is also an option for *part specified* connections. The part specifications happens via the setting of the port. An IP address is not required.

Operating modes

Depending on the connection, the following operating modes are available:

SEND/RECEIVE

The SEND/RECEIVE interface allows the program controlled communication to any partner station via a configured connection. Here the data transfer happens by call from your user application. The FC 5 and FC 6 that are part of the VIPA block library are serving as interface.

This enables your control to send messages depending on process events.

FETCH/WRITE PASSIVE

With the help of FETCH/WRITE services partner systems have the direct access to memory areas of the CPU. This are "passive" communication connections that have to be configured. The connections are "actively" established by the connection partner (e.g. Siemens-S5).

FETCH PASSIVE (request data)

FETCH allows a partner system to request data.

WRITE PASSIVE (write data)

This allows a partner system to write data in the data area of the CPU.

Function overview

Outline

In the following the functions are listed that are supported by the CP part of the CPU 31xSN/Net starting with CP firmware version 1.7.4:

Configurable connections

Function	Property
Maximum number of productive connections	16 (8 at CPU 315-4NE11)
TCP connections	SEND, RECEIVE, FETCH PASSIVE, WRITE PASSIVE Connection establishment active and passive, supports unspecified connection partner.
ISO-on-TCP connections (RFC1006)	SEND, RECEIVE, FETCH PASSIVE, WRITE PASSIVE Connection establishment active and passive, supports unspecified connection partner.
UDP connections	SEND and RECEIVE The transfer of the telegrams is not acknowledged, i.e. the loss of messages is not recognized by the send block.
UDP Broadcast connection	SEND
UDP Multicast connection	SEND and RECEIVE (max. 16 multicast circles)
Data block length	max. 64kByte (max. 2kByte at UDP)
VIPA handling blocks	For connection commands at the PLC: AG_SEND (FC 5) / AG_RECEIVE (FC 6) Any call without lock in all OBs.

PG connections and diagnostic

Function	Property
Maximum number of PG/OP connections	32 (each 1 connection is reserved for PG and OP)
Diagnostic	Supports NCM diagnostic via Ethernet
Search within network	Supports Siemens SIMATIC Manager search
10/100MBit	Switch happens automatically

Fast introduction

Overview

At the first start-up of a CPU 31xSN/NET, Ethernet PG/OP and CP 343 of the CPU 31xSN/NET do not have any IP address. The assignment takes place directly via the hardware configuration of the Siemens SIMATIC Manager. For the project engineering of a CPU 31xS with CP 343 please follow this approach:

- **Assembly and commissioning**
- **Hardware configuration** (Inclusion of CP in CPU)
- **CP project engineering** via NetPro (connection to Ethernet)
- **PLC programming** via user application (connection to PLC)
- **Transfer of the complete project to CPU**

Note

To be compatible to the Siemens SIMATIC Manager, the CPU 31xS from VIPA has to be configured as

CPU 318-2DP (6ES7 318-2AJ00-0AB0)!

The Ethernet PG/OP channel of the CPU 31xSN/NET is always configured virtually as 1st module after the really plugged modules at the standard bus as CP343-1 (343-1EX11) from Siemens. The CP 343 of a CPU 31xSN/NET has always to be configured below the before configured CP also as CP343-1 (343-1EX11).

Assembly and commissioning

- Install your System 300S with the CPU 31xSN/NET.
- Wire the system by connecting cables for voltage supply, signals and Ethernet. A detailed description is to be found in the chapter "Assembly and installation guidelines".
- Switch on the voltage supply. → After a short boot time, the CP is in idle.
  At the first commissioning res. after an overall reset of the CPU, Ethernet PG/OP channel and CP have no IP address. For control purposes you may now reach the CP via the MAC address. The MAC address is to be found beneath the front flap on the left side of the module at a small label on the module.

Assign IP parameters

For the assignment of the IP parameters such as IP address, Subnet mask etc. you have the following possibilities:

- Online using Siemens SIMATIC Manager via "Assign Ethernet Address" (at least CP-Firmware 1.7.4)
- with the help of a "minimum project" and transfer this via MMC or MPI into the CPU. After a reboot of the CPU and after switching the PG/PC interface to "TCP/IP... RFC1006" you may now configure your CPU online via the CP.

Address assignment with "Assign Ethernet Address"

Please regard this functionality is available with firmware version 1.7.4 and up.

- Start Siemens SIMATIC Manager
- Switch to "TCP/IP... RFC1006" using **Options** > *Set PG/PC interface*.
- The dialog for initialization of a station opens by **PLC** > *Assign Ethernet Address*.
- To get the stations and their MAC address use the [Browse] button or type in the MAC Address. The Mac address can be found at a label at the side of the CPU.
- Choose if necessary the known MAC address of the list of found stations.
- Either type in the IP configuration like IP address, subnet mask and gateway. Or your station is automatically provided with IP parameters by means of a DHCP server. Depending of the chosen option the DHCP server is to be supplied with MAC address, equipment name or client ID. The client ID is a numerical order of max. 63 characters. The following characters are allowed: "hyphen", 0-9, a-z, A-Z
- Confirm with [Assign ...]

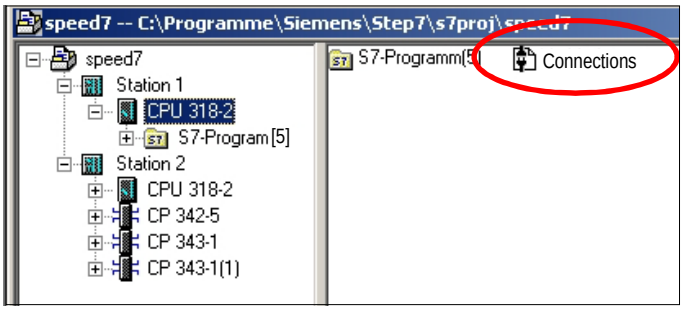
Directly after the assignment the CP is online reachable using the set IP parameters.

Address assignment with minimal project

- Start Siemens SIMATIC Manager with new project.
- Place a new System 300 station with **Insert** > *Station* > *SIMATIC 300 station*
- Activate the station "SIMATIC 300" and open the hardware configurator by clicking on "Hardware".
- Configure a rack (SIMATIC 300 \ Rack-300 \ Profile rail).
- Configure in deputy of your CPU 31xSN/NET the Siemens CPU 318-2DP with the order no. 6ES7 318-2AJ00-0AB0 V. 3.0 which is to be found at SIMATIC 300 \ CPU 300 \ CPU 318-2 \ 318-2AJ00-0AB00. If needed, parameterize the CPU 318-2DP.
- Place the System 300 modules in plugged sequence starting with plug-in location 4.
- Configure the internal PG/OP channel directly under the really plugged modules as virtual **CP 343-1 (343-1EX11)** from Siemens.
- Set IP address, subnet mask and gateway at CP properties.
- Always configure as 2nd CP the internal CP 343 as **CP 343-1 (343-1EX11)** by setting another IP address, subnet mask and gateway.
- Set IP address, subnet mask and gateway at CP properties and save and compile your project. This is the end of the *Minimal project*. After the *Minimal project* is transferred to CPU, the CP can be accessed by means of IP address and Subnet mask of the project.

Configure connections with NetPro

The link-up between the stations happens with the graphical interface NetPro. Start NetPro by clicking on a network in your project res. on connections in the CPU directory.

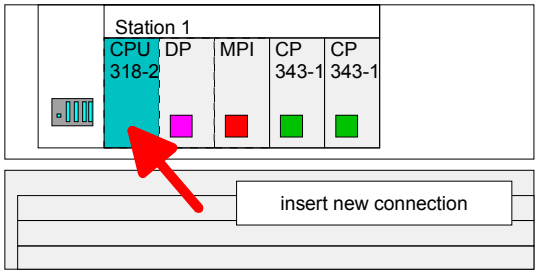


Link-up stations

For the project engineering of connections, connected stations are presumed. To link-up stations, point on the colored net mark of the according CP with the mouse and drag it to the network you want to assign. The connection is displayed graphically by a line.

Configure connections

For the project engineering of new connections click on the according CPU and choose "Insert new connection" from the context menu.



Via the dialog window you may set the parameters for a connection. The parameters ID and LADDR are required for the usage on the blocks AG_SEND res. AG_RECV.

Always use the 2nd CP from the route

Please take care to always choose the 2nd CP from route for communication. As 1st CP you will always see the Ethernet PG/OP channel that does not support configurable connections.

Save and compile connections

Save and compile your project and close NetPro.

To store the CP project engineering data in the system data, you have to activate the option "Save configuration data on the CPU" (default setting) at *object properties* area *Options* in the hardware configuration of the CP.

PLC user application

For the execution of connection commands at the PLC, your CPU requires an user application. For this, exclusively the VIPA handling blocks AG_SEND (FC 5) and AG_RECV (FC 6) are used. The blocks are part of the VIPA library that is included in the consignment as CD (SW830).

Specify the according CP via the parameters *ID* and *LADDR* by calling FC 5 res. FC 6.

Project transfer

Information about transferring a project may be found at chapter "Deployment CPU 31xS" at "Project transfer".

The following pages provide a more detailed description of the steps of the fast introduction.

Hardware configuration

Overview

For the Hardware configuration the hardware configurator from Siemens is used. Here you set amongst others the IP address of the CP and configure the hardware components of your PLC.

Due to the fact that neither the Ethernet PG/OP channel nor the CP 343 have an IP address in delivery state you may engineer the CPU exclusively via MPI or MMC.

For the access to the CPU via the Ethernet PG/OP channel res. the CP 343 it is required that the CPU has a hardware project engineering where IP address and subnet mask for Ethernet PG/OP res. CP 343 are defined.

Requirements

For the hardware configuration the following software is required:

- Siemens SIMATIC Manager V. 5.1 or higher and vipa_speedbus.gsd
- SIMATIC NET

Note!

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

Note

To be compatible to the Siemens SIMATIC Manager, the CPU 31xS from VIPA has to be configured as

CPU 318-2DP (6ES7 318-2AJ00-0AB0)!

The Ethernet PG/OP channel of the CPU 31xSN/NET is always configured virtually as 1st module after the really plugged modules at the standard bus as CP343-1 (343-1EX11) from Siemens. The CP 343 of a CPU 31xSN/NET has always to be configured below the before configured CP also as CP343-1 (343-1EX11).

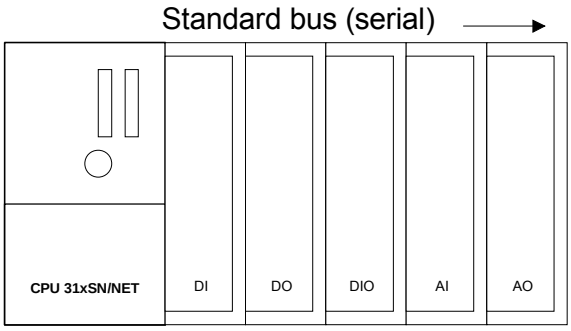
Steps of the project engineering

The following text shows the approach of the project engineering in the hardware configurator from Siemens in an abstract sample.

The project engineering is divided into 3 parts:

- Project engineering of the CPU
- Project engineering of the really plugged modules at the standard bus
- Project engineering Ethernet PG/OP channel and CP 343

Hardware assembly



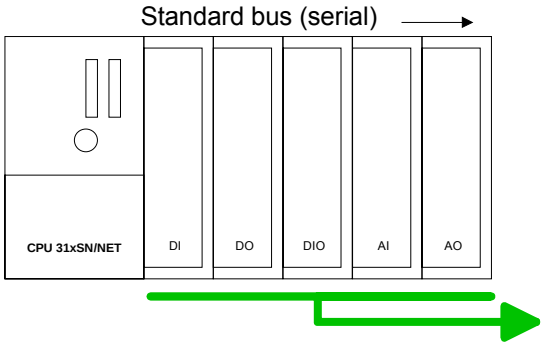
Project engineering of the CPU

- Start the hardware configurator from Siemens with a new project and insert a profile rail from the hardware catalog.
- Place the following Siemens CPU at slot 2:
  CPU 318-2DP (6ES7 318-2AJ00-0AB0 V. 3.0)

Project engineering of the modules at the standard bus

The modules at the right side of the CPU at the standard bus are configured with the following approach:

- Include your System 300V modules at the standard bus in the plugged sequence starting with slot 4.
- Parameterize the CPU res. the modules where appropriate. The parameter window opens by a double click on the according module.



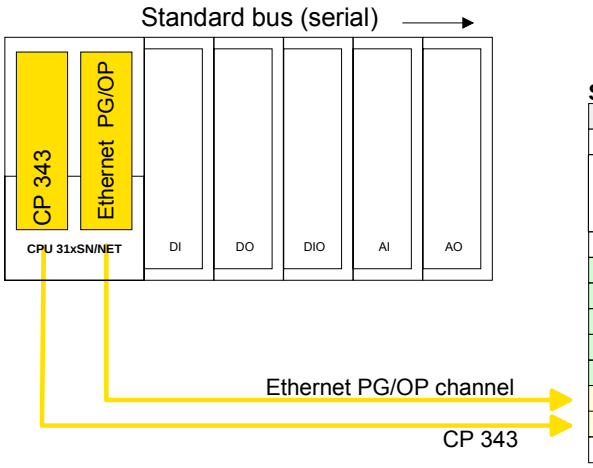
Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
4	DI
5	DO
6	DIO
7	AI
8	AO
9	
10	
11	

Project engineering Ethernet PG/OP channel and CP 343

For the internal Ethernet PG/OP channel that every SPEED7-CPU includes, you have to configure a Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \CP 343-1 \ 6GK7 343-1EX11 0XE0) always as 1st module below the really plugged modules.

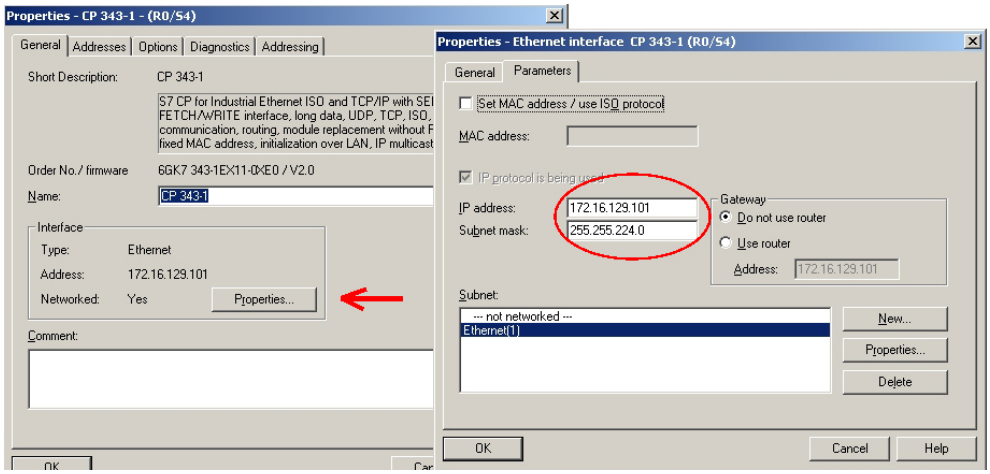
The integrated CP 343 of the CPU 31xSN/NET is also configured as **CP 343-1 (343-1EX11)** but always below the before configured CP 343-1.



Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
4	DI
5	DO
6	DIO
7	AI
8	AO
9	343-1EX11
10	343-1EX11
11	

Open the property window via double-click on the CP 343-1EX11 and enter at properties the IP address, subnet mask and gateway for the CPs and select the wanted subnet.



Bus extension with IM 360 and IM 361

To extend the bus you may use the IM 360 from Siemens where you can connect up to 3 further extension racks via the IM 361. Bus extensions are always placed at slot 3.

Project engineering SPEED-Bus and project transfer

Detailed information about project engineering of the SPEED-Bus modules and the project transfer may be found at Chapter "Deployment CPU 31xS".

Configure connections

Outline

The project engineering of connections i.e. the "link-up" between stations happens in NetPro from Siemens. NetPro is a graphical user interface for the link-up of stations.

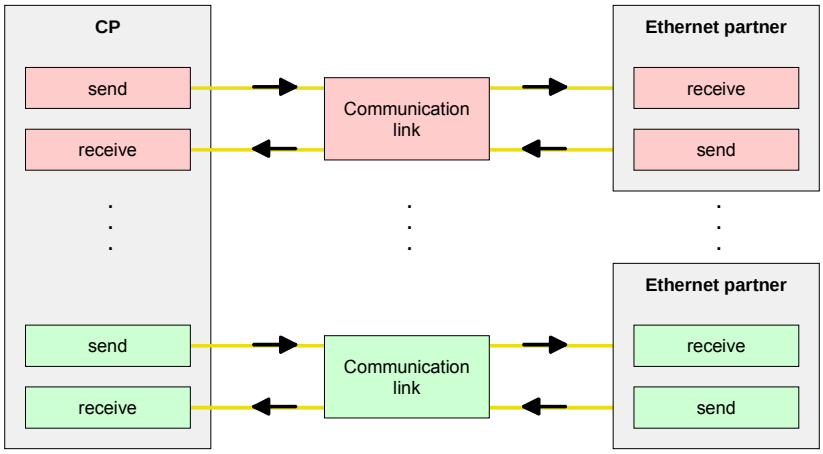
A communication connection enables the program controlled communication between two participants at the Industrial Ethernet. The communication partners may here be part of the same project or - at multi projects - separated within related part projects.

Communication connections to partners outside of a project are configured via the object "In unknown project" or via deputy objects like "Other stations" or Siemens "SIMATIC S5 Station".

Properties

The following properties are characterizing a communication connection:

- One station always executes an active connection establishment.
- Bi-directional data transfer (Send and receive on one connection)
- Both participant have equal rights, i.e. every participant may initialize the send res. receive process event controlled.
- Except of the UDP connection, at a communication connection the address of the communication partner is set via the project engineering. Here the connection is active established by one station.



Requirements

- Siemens SIMATIC Manager V. 5.1 or higher and SIMATIC NET are installed.
- The CP has been engineered at the hardware configuration, entered into the hardware configuration and linked-up to the Ethernet subnet.
- The CP as bus participant has an IP address.

Note!

All stations outside of the recent project must be configured as replacement objects like e.g. Siemens "SIMATIC S5" or "other station" or with the object "In unknown project".

When creating a connection you may also choose the partner type "unspecified" and set the required remote parameter directly in the connection dialog.

Work environment of NetPro

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

Start NetPro by clicking on a "net" in the Siemens SIMATIC Manager or on "connections" within the CPU.

The environment of NetPro has the following structure:

1 *Graphic net view*

All stations and networks are displayed in a graphic view. By clicking on the according component you may access and alter the concerning properties.

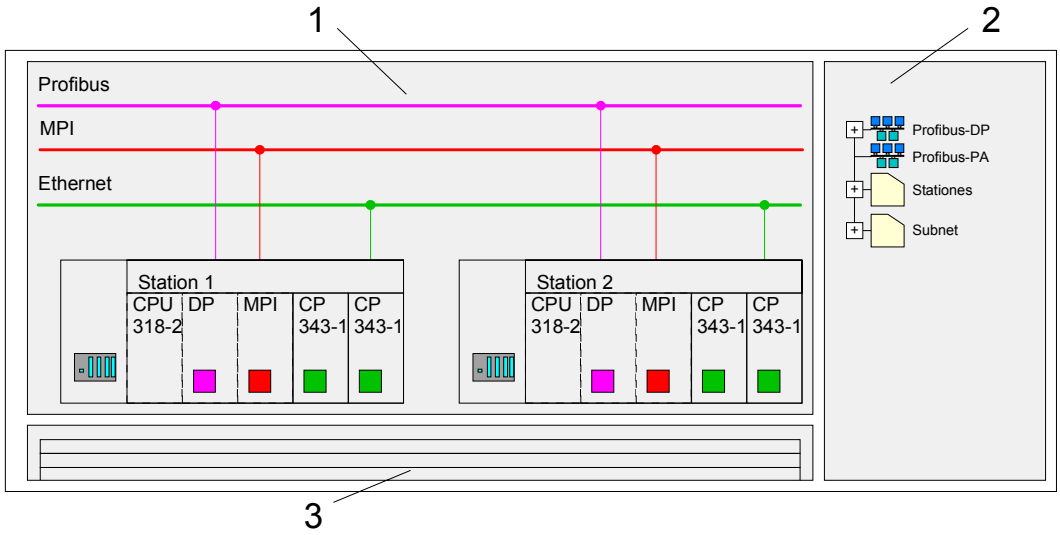
2 *Net objects*

This area displays all available net objects in a directory view. By dragging a wanted object to the net view you may include further net objects and open them in the hardware configurator.

3 *Connection table*

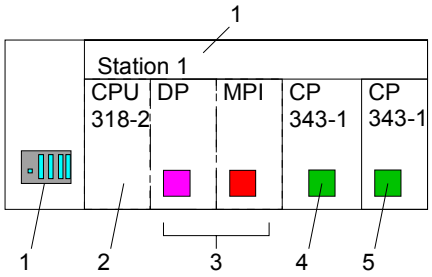
The connection table lists all connections in a table. This list is only shown when you highlighted a connectable module like e.g. a CPU.

You may insert new connections into this table with the according command.



PLC stations

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



1 *Station*

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

2 *CPU*

A click onto the CPU shows the connection table. The connection table shows all connections that are configured for the CPU.

3 *Internal communication components*

This displays the communication components that are available in your CPU. For the SPEED7-Net-CPUs are configured as CPU 318-2DP the internal components do not show the CP.

Due to this, the CPs that are included in the SPEED7-Net-CPU must be configured as external CPs behind the really plugged modules. The CPs are then also shown in NetPro as external CPs (4, 5) in the station.

4 *Ethernet PG/OP channel*

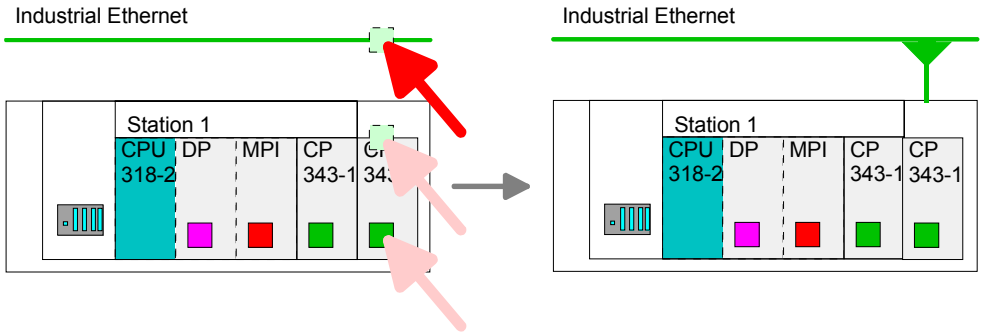
The internal *Ethernet PG/OP channel* must always be configured as 1st CP in the hardware configuration. This CP only serves the PG/OP communication. You may not configure connections.

5 *CP 343*

The internal CP 343 must always be configured as 2nd CP in the hardware configuration after the *Ethernet PG/OP channel*.

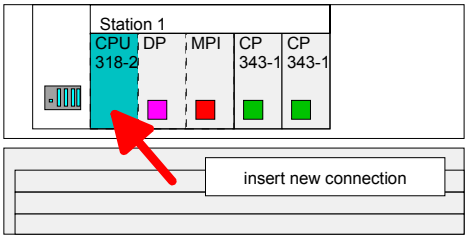
Link up stations

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



Projecting connections

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



A dialog window opens where you may choose the connection partner and the type of the connection.

Highlight the partner station to which you would like to establish a connection.

Choose at "Type" the connection type to be used.

The following connections are supported by the CP at this time:

ISO-on-TCP (SEND-RECEIVE, FETCH-WRITE PASSIVE)

TCP (SEND-RECEIVE, FETCH-WRITE PASSIVE)

UDP (SEND-RECEIVE)

General information

ID

LADDR

If activated, a properties dialog for the according connection opens. This dialog window is the link to your PLC program. Here you may adjust the *Local ID* and evaluate the *LADDR*.

Both are parameters that must be given to your PLC application when using the FC 5 and 6 (AG_SEND, AG_RECEIVE). Please do always use the VIPA FCs that are delivered with the SW830 as a library.

Note!

Please regard that a CP depending ID is assigned to the connections of the SEND/RECEIVE interface. This may cause alterations of the ID at changes of the project. In this case you also have to adjust the interface supply of AG_SEND res. AG_RECV in the user application.

If a CP is exchanged by another one, this must at least provide the same services and must at least have the same version level. Only this can guarantee the connections configured via the CP to remain consistent and useable.

Route

The *route* allows you to access the concerning CP that should be used for the connection. By using a CPU 31xSN/NET you must use the 2nd CP of the path selection for the communication via the internal CP 343. As 1st CP in the list always the integrated Ethernet PG/OP channel is shown though this only supports PG/OP communication.

Addresses

The register addresses shows the relevant local and partner address information as suggestion values. Depending on the communication type you may leave the address information unspecified.

The following table shows the combination options with the different operating modes:

Connection partner	Connection type	Conn. Establ.	Connection	Operating mode
Specified in NetPro (in recent project)	TCP / ISO-on-TCP	active/passive	specified	SEND/RECEIVE
	UDP	-		
Unspecified in NetPro (in recent project)	TCP / ISO-on-TCP	active	specified	SEND/RECEIVE
		passive	part specified (Port)	SEND/RECEIVE FETCH PASSIV WRITE PASSIV
			unspecified	
	UDP	-	specified	SEND/RECEIVE
Unspecified in NetPro (in "unknown project")	TCP / ISO-on-TCP	active	unspecified (connection name)	SEND/RECEIVE
		passive	unspecified (connection name)	SEND/RECEIVE FETCH PASSIV WRITE PASSIV
	UDP	-	unspecified (connection name)	SEND/RECEIVE
All Broadcast stations	UDP	-	specified (Port, Broadcast addr.)	SEND
All Multicast stations	UDP	-	specified (Port, Multicast group)	SEND/RECEIVE

Address parameter

A connection is specified by the *local* and *partner* connection end point. At the project engineering of connections ports/TSAPs must be congruent crosswise. Depending on the protocol the following parameters define a connection end point:

IP address Station A			IP address Station B	
partner TSAP	→	*ISO-on-TCP*	→	local TSAP
local TSAP	←	*connection*	←	partner TSAP
IP address Station A			IP address Station B	
partner Port	→	*TCP*	→	local Port
local Port	←	*connection*	←	partner Port
IP address Station A			IP address Station B	
partner Port	→	*UDP*	→	local Port
local Port	←	*connection*	←	partner Port

TSAP

ISO-on-TCP supports TSAP lengths (**T**ransport **S**ervice **A**ccess **P**oint) of 1...16Byte. You may enter the TSAP in ASCII or hexadecimal format. The calculation of the length happens automatically.

Port

Ports res. port addresses are defining the access point to the user application within the station/CPU. These must be unambiguous. A port address should be within the range of 2000...65535. Foreign and local ports may only be identical with one connection.

Save and compile connections

After you configured all connections this way, you may save and compile your project and exit NetPro.

To store the CP project engineering data in the system data, you have to activate the option "Store project data in the CPU" (default setting) at *object properties* area *Options* in the hardware configuration of the CP.

Broadcast-/ Multicast-connections

The expression "connection" is also used at UDP although there is no explicit connection establishment between the communication partners during runtime of the stations.

But during the project engineering like at with e.g. TCP the communication partners are assigned to each other and therefore logical linked-up.

Only at UDP the following options are additionally available at the selection of the connection partner:

- All broadcast stations
- All multicast stations

Broadcast stations

By selecting *All broadcast stations* as connection partner, you define that UDP telegrams are to be send to all available broadcast participants. Please regard that the CP may exclusively receive broadcast telegrams. The reception of user data via broadcast is not possible. Per default, broadcasts that are only serving the Ethernet communication, like e.g. ARP-Requests (Search MAC <> IP address), are received and accordingly processed.

For the identification of the broadcast participants within the net, you have to define a valid broadcast address as partner IP during project engineering of a broadcast connection. Additionally to the broadcast address you have to set a common port for sender and receiver.

Multicast stations

By selecting *All Multicast stations* you define that UDP telegrams have to be send res. received by all participants of a multicast group. In opposite to broadcast here a reception is possible.

For the identification of the multicast participants within the net, you have to define a valid multicast group address as partner IP during project engineering of a multicast connection. Additionally to this address you have to set a common port for sender and receiver.

The maximum number of multicast circles, which are supported by the Ethernet CP 343 - SPEED-Bus, is identical to the maximum number of connections.

SEND/RECEIVE with PLC program

Overview

For the execution of connection commands at the PLC, your CPU requires an user application. For this, exclusively the VIPA handling blocks AG_SEND (FC 5) and AG_RECV (FC 6) are used. By including these blocks into the cycle block OB 1 you may send and receive data cyclic.

The two FCs are part of the VIPA library, that is included in the consignment as CD (SW830).

Note!

Please regard that you may only use the SEND/RECV-FCs from VIPA in your user application for the communication with VIPA-CPs. At a change to VIPA-CPs in an already existing project, the present AG_SEND/ AG_LSEND res. AG_RECV/AG_LRECV may be replaced by AG_SEND res. AG_RECV from VIPA without adjustment. Due to the fact that the CP automatically adjusts itself to the length of the data to transfer, the L variant of SEND res. RECV is not required for VIPA CPs.

Communication blocks

For the communication between CPU and CP, the following FCs are available:

AG_SEND (FC 5)

This block transfers the user data from the data area given in *SEND* to the CP specified via *ID* and *LADDR*. As data area you may set a PIQ, bit memory or data block area. When the data area has been transferred without errors, "order ready without error" is returned.

AG_RECV (FC 6)

The block transfers the user data from the CP into a data area defined via *RECV*. As data area you may set a PII, bit memory or data block area. When the data area has been transferred without errors, "order ready without error" is returned.

Status displays

The CP processes send and receive commands independently from the CPU cycle and needs for this transfer time. The interface with the FC blocks to the user application is here synchronized by means of acknowledgements/receipts.

For status evaluation the communication blocks return parameters that may be evaluated directly in the user application.

These status displays are updated at every block call.

Deployment at high communication load

Do not use cyclic calls of the communication blocks in OB 1. This causes a permanent communication between CPU and CP. Program instead the communication blocks within a time OB where the cycle time is higher than the time of the OB 1 respectively event controlled.

FC call is faster than CP transfer time

If a block is called a second time in the user application before the data of the last time is already completely send res. received, the FC block interface reacts like this:

AG_SEND

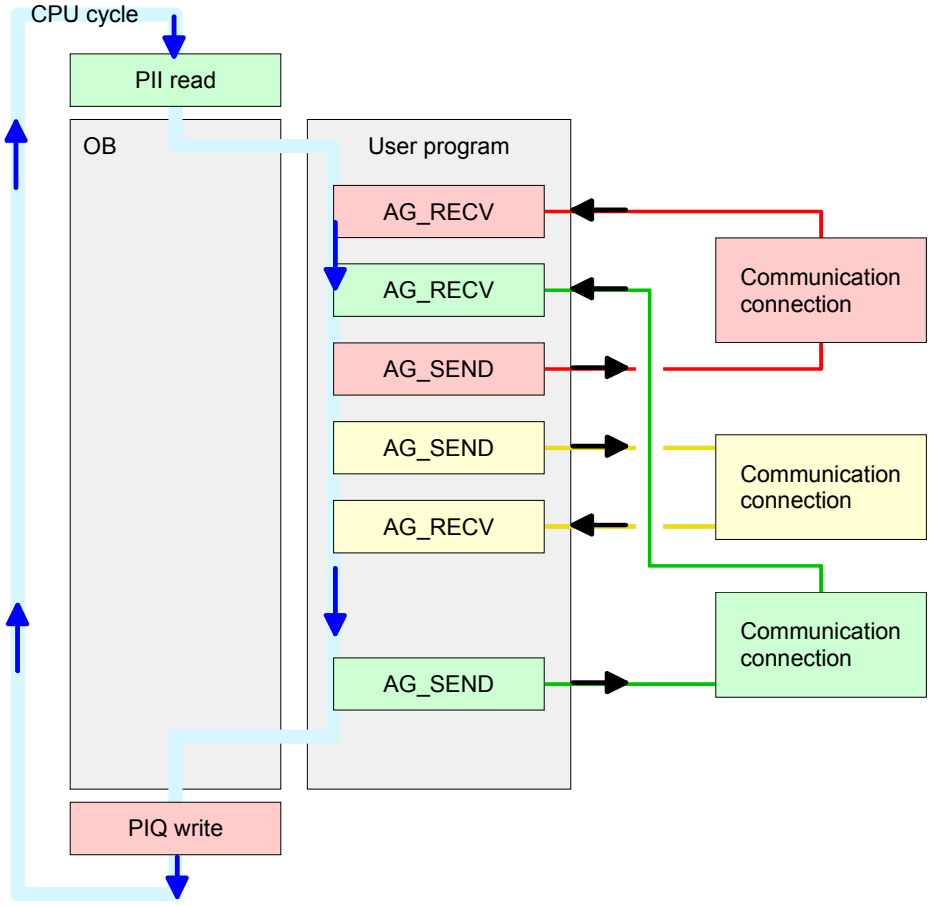
No command is accepted until the data transfer has been acknowledged from the partner via the connection. Until this you receive the message "Order running" before the CP is able to receive a new command for this connection.

AG_RECV

The order is acknowledged with the message "No data available yet" as long as the CP has not received the receive data completely.

AG_SEND, AG_RECV in the user application

The following illustration shows a possible sequence for the FC blocks together with the organizations and program blocks in the CPU cycle:



The FC blocks with concerning communication connection are summed up by color. Here you may also see that your user application may consist of any number of blocks. This allows you to send or receive data (with AG_SEND res. AG_RECV) event or program driven at any wanted point within the CPU cycle.

You may also call the blocks for **one** communication connection several times within one cycle.

AG_SEND (FC 5) By means of AG_SEND the data to send are transferred to the CP.

Parameter

Parameter	Declaration	Type	Description
ACT	Input	BOOL	Activation of the sender 0: Updates DONE, ERROR and STATUS 1: The data area defined in SEND with the length LEN is send
ID	Input	INT	Connection number 1 ... 16 (identical with ID of NetPro)
LADDR	Input	WORD	Logical basic address of the CP (identical with LADDR of NetPro)
SEND	Input	ANY	Data area
LEN	Input	INT	Number of bytes from data area to transfer
DONE	Output	BOOL	Status parameter for the order 0: Order running 1: Order ready without error
ERROR	Output	BOOL	Error message 0: Order running (at DONE = 0) 0: Order ready without error (at DONE = 1) 1: Order ready with error
STATUS	Output	WORD	Status message returned with DONE and ERROR. More details are to be found in the following table

AG_RECV (FC 6) By means of AG_RECV the data received from the CP are transferred to the CPU.

Parameter

Parameter	Declaration	Type	Description
ID	Input	INT	Connection number 1 ... 16 (identical with ID of NetPro)
LADDR	Input	WORD	Logical basic address of the CP (identical with LADDR of NetPro)
RECV	Input	ANY	Data area for the received data
NDR	Output	BOOL	Status parameter for the order 0: Order running 1: Order ready data received without error
ERROR	Output	BOOL	Error message 0: Order running (at NDR = 0) 0: Order ready without error (at NDR = 1) 1: Order ready with error
STATUS	Output	WORD	Status message returned with NDR and ERROR. More details are to be found in the following table.
LEN	Output	INT	Number of bytes that have been received

DONE, ERROR, STATUS

The following table shows all messages that can be returned by the CP after a SEND res. RECV command.

A "-" means that this message is not available for the concerning SEND res. RECV command.

DONE (SEND)	NDR (RECV)	ERROR	STATUS	Description
1	-	0	0000h	Order ready without error
-	1	0	0000h	New data received without error
0	-	0	0000h	No order present
-	0	0	8180h	No data available yet
0	0	0	8181h	Order running
0	0	1	8183h	No CP project engineering for this order
0	-	1	8184h	System error
-	0	1	8184h	System error (destination data area failure)
0	-	1	8185h	Parameter LEN exceeds source area SEND
	0	1	8185h	Destination buffer (RECV) too small
0	0	1	8186h	Parameter ID invalid (not within 1 ...16)
0	-	1	8302h	No receive resources at destination station, receive station is not able to process received data fast enough res. has no receive resources reserved.
0	-	1	8304h	The connection is not established. The send command shouldn't be send again before a delay time of >100ms.
-	0	1	8304h	The connection is not established. The receive command shouldn't be send again after a delay time of >100ms.
0	-	1	8311h	Destination station not available with the defined Ethernet address.
0	-	1	8312h	Ethernet error in the CP
0		1	8F22h	Source area invalid, e.g. when area in DB not present Parameter LEN < 0
-	0	1	8F23h	Source area invalid, e.g. when area in DB not present Parameter LEN < 0
0	-	1	8F24h	Range error at reading a parameter.
-	0	1	8F25h	Range error at writing a parameter.
0	-	1	8F28h	Orientation error at reading a parameter.
-	0	1	8F29h	Orientation error at writing a parameter.
-	0	1	8F30h	Parameter is within write protected 1st recent DB
-	0	1	8F31h	Parameter is within write protected 2nd recent DB
0	0	1	8F32h	Parameter contains oversized DB number.
0	0	1	8F33h	DB number error
0	0	1	8F3Ah	Area not loaded (DB)

continued...

... continue DONE, ERROR, STATUS

DONE (SEND)	NDR (RECV)	ERROR	STATUS	Description
0	-	1	8F42h	Acknowledgement delay at reading a parameter from peripheral area.
-	0	1	8F43h	Acknowledgement delay at writing a parameter from peripheral area.
0	-	1	8F44h	Address of the parameter to read locked in access track
-	0	1	8F45h	Address of the parameter to write locked in access track
0	0	1	8F7Fh	Internal error e.g. invalid ANY reference e.g. parameter LEN = 0 .
0	0	1	8090h	Module with this module start address not present or CPU in STOP.
0	0	1	8091h	Module start address not within double word grid.
0	0	1	8092h	ANY reference contains type setting unequal BYTE.
-	0	1	80A0h	Negative acknowledgement at reading the module
0	0	1	80A4h	reserved
0	0	1	80B0h	Module doesn't recognize record set.
0	0	1	80B1h	The length setting (in parameter LEN) is invalid.
0	0	1	80B2h	reserved
0	0	1	80C0h	Record set not readable.
0	0	1	80C1h	The set record set is still in process.
0	0	1	80C2h	Order accumulation.
0	0	1	80C3h	The operating sources (memory) of the CPU are temporarily occupied.
0	0	1	80C4h	Communication error (occurs temporarily; a repetition in the user application is reasonable.)
0	0	1	80D2h	Module start address is wrong.

Status parameter at reboot

At a reboot of the CP, the output parameter are set back as follows:

- DONE = 0
- NDR = 0
- ERROR = 8180h (at AG_RECV)
  ERROR = 8181h (at AG_SEND)

Project transfer

Information about transferring a project may be found at chapter "Deployment CPU 31xS" at "Project transfer".

NCM diagnostic – Help for error diagnostic

Check list for error search

This page shall help you with the error diagnostic. The following page lists a number of typical problems and their probable causes:

Question	Solution with "no"
CPU in Run?	Control DC 24V voltage supply. Set RUN/STOP lever in position RUN. Check PLC program and transfer it again.
AG_SEND, AG_RECV in user application?	These 2 blocks are required in the user application for the data transfer between CP and CPU. Both blocks must also be called with a passive connection.
Is CP able to connect?	Check Ethernet cable (at a point-to-point connection a crossed Ethernet cable is to be used). Check IP address.
Can data be transferred?	Check Port no. for read and write. Check source and destination areas. Check if the 2nd CP is selected in the route. Enlarge the receive res. send buffer defined via the ANY pointer.
Is the complete data block send at ISO-on-TCP?	Check the LEN parameter at AG_SEND. Set the receive res. send buffer defined via the ANY pointer to the required size.

Siemens NCM S7 diagnostic

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

The following diagnostic functions are available:

- Check operating state at Ethernet
- Read the diagnostic buffer of the CP
- Diagnostic of connections

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

Start NCM diagnostic

There are two options to start the diagnostic tool:

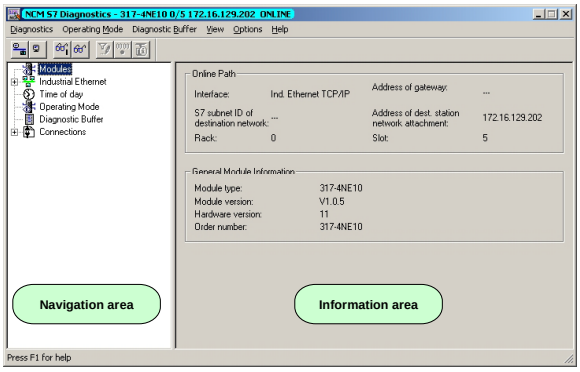
- Via *Windows-START menu > SIMATIC ... NCM S7 > Diagnostic*
- Within the project engineering res. the hardware configuration via the register "Diagnostic" in the "Property" dialog with [Execute].

Structure

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

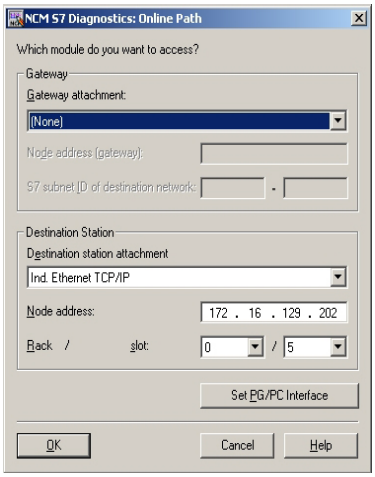
The *navigation area* at the left side contains the hierarchical listed diagnostic objects. Depending on CP type and configured connections there is an adjusted object structure in the navigation area.

The *information area* at the right side always shows the result of the navigation function you chose in the *navigation area*.



No diagnostic without connection

A diagnostic always requires a online connection to the CP you want to control. For this click on  at the symbol bar. The following dialog window appears:



Set the following parameters at *destination station*:

Connection...: Ind. Ethernet TCP/IP

Station addr.: Enter the IP address of the CP

Module rack/slot:

Enter the *module rack* and *slot* of the CP 343 that you've placed at the 2nd slot.

Set your PG/PC interface to TCP/IP...RFC1006. Via [OK] you start the online diagnostic.

Read diagnostic buffer

The CP has a diagnostic buffer. This has the architecture of a ring memory and may store up to 100 diagnostic messages. The NCM diagnostic allows you to monitor and evaluate the CP diagnostic messages via the diagnostic object *Diagnostic buffer*.

Via a double click on a diagnostic message the NCM diagnostic shows further information.

Approach for diagnostic

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

Note!

Please always control the preconditions for an operative communication using the check at the beginning of this chapter.

For the aimed diagnostic deployment the following approach is convenient:

- Start diagnostic.
- Open the dialog for the online connection with  , enter connection parameters and establish the online connection with [OK].
- Identify the CP and check the recent state of the CP via module status.
- Check the connections for particularities like:
  - Connection status
  - Receive status
  - Send status
- Control and evaluate the diagnostic buffer of the CP via *diagnostic buffer*.
- As needed, alter project engineering res. programming and restart diagnostic.

Coupling to other systems

Outline

The operating mode FETCH/WRITE supported at TCP res. ISO-on-TCP can be used for accesses of partner devices to the PLC system memory. To be able to use this access also for example for implementation in PC applications you have to know the telegram structure for orders. The specific headers for request and acknowledgement telegrams have per default a length of 16Byte and are described at the following pages.

ORG format

The organization format is the abbreviated description of a data source or a data destination in a PLC environment. The available ORG formats are listed in the following table.

The ERW-identifier is used for the addressing of data blocks. In this case the data block number is entered into this identifier. The start address and quantity provide the address for the memory area and they are stored in HIGH-/LOW- format (Motorola-formatted addresses)

Description	Type	Range
ORG identifier	BYTE	1...x
ERW identifier	BYTE	1...255
Start address	HILOWORD	0...y
Length	HILOWORD	1...z

The following table contains a list of available ORG-formats. The "length" must not be entered as -1 (FFFFh).

ORG identifier 01h-04h

CPU area	DB	MB	EB	AB
ORG identifier	01h	02h	03h	04h
Description	Source/destination data from/into data Block in main memory.	Source/destination data from/into flag memory area	Source/destination data from/into process image of the inputs (PII).	Source/destination data from/into process image of the outputs (PIQ).
ERW identifier (DBNO)	DB, from where the source data is retrieved or to where the destination data is transferred.	irrelevant	irrelevant	irrelevant
Start address significance	DBB-No., from where the data is retrieved or where the data is saved.	MB-No., from where the data is retrieved or where the data is saved.	IB-No., from where the data is retrieved or where the data is saved.	QB-No., from where the data is retrieved or where the data is saved.
Length significance	Length of the source/destination data block in words	Length of the source/destination data block in bytes	Length of the source/destination data block in bytes	Length of the source/destination data block in bytes

Note!

Information about the valid range can be found at Chapter "Hardware description of the CPU".

ORG identifier 05h-0Ah

CPU area	PB	ZB	TB
ORG identifier	05h	06h	07h
Description	source/destination data from/into peripheral modules. Input module for source data, output module for destination data.	source/destination data from/into counter cells.	Source/destination data from/into timer cells.
ERW identifier (DBNO)	irrelevant	irrelevant	irrelevant
Start address Significance	PB-No., from where the data can be retrieved or where it is saved.	ZB-No., from where the data can be retrieved or where it is saved.	TB-No., from where the data can be retrieved or where it is saved.
Length Significance	Length of the source/destination data block in bytes.	Length of the source/destination data block in words (counter cell = 1 word).	Length of the source/destination data block in words (counter cell = 1 word).

Transfer of blocks with numbers >255

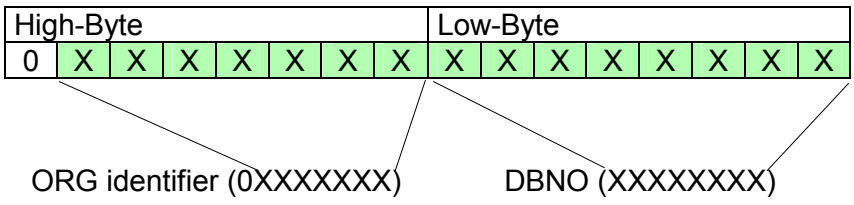
ORG identifier 81h-FFh

To transfer data blocks of the number range 256 ... 32768 you may use the ORG identifier 81h-FFh.

For the setting of a DB No. >255 needs a length of one word, the $DBNO_{new}$ is assembled from the content of the ORG identifier and the DBNO.

$DBNO_{new}$ is created as word as follows:

$DBNO_{new}$



If the highest bit of the ORG identifier is set, the Low-Byte of $DBNO_{new}$ is defined via DBNO and the High-Byte of $DBNO_{new}$ via ORG identifier, where the highest bit of the ORG identifier is eliminated.

The following formula illustrates this:

$DBNO_{new}$ =256 x (ORG-identifier AND 7Fh) + DBNO

Structure of PLC-Header

For every FETCH and WRITE the CP generates PLC header for request and acknowledgment messages. Normally the length of these headers is 16Bytes and have the following structure:

WRITE

Request telegram Remote Station

System ID	="S5"	(Word)
Length Header	=10h	(Byte)
ID OP-Code	=01h	(Byte)
Length OP-Code	=03h	(Byte)
OP-Code	**=03h**	(Byte)
ORG block	=03h	(Byte)
Length ORG block	=08h	(Byte)
*ORG identifier**		*(Byte)*
ERW identifier		*(Byte)*
Start address		*(Word)*
Length		*(Word)*
Empty block	=FFh	(Byte)
Length empty block	=02h	(Byte)
Data up to 64kByte (only if error no.=0)		

Acknowledgement telegram CP

System ID	="S5"	(Word)
Length Header	=10h	(Byte)
ID OP-Code	=01h	(Byte)
Length OP-Code	=03h	(Byte)
OP-Code	**=04h**	(Byte)
Ackn. block	=0Fh	(Byte)
Length Ack. block	=03h	(Byte)
Error no.		(Byte)
Empty block	=FFh	(Byte)
Length empty block	=07h	(Byte)
5 empty bytes attached		

FETCH

Request telegram Remote Station

System ID	="S5"	(Word)
Length Header	=10h	(Byte)
ID OP-Code	=01h	(Byte)
Length OP-Code	=03h	(Byte)
OP-Code	**=05h**	(Byte)
ORG block	=03h	(Byte)
Length ORG block	=08h	(Byte)
*ORG identifier**		*(Byte)*
ERW identifier		*(Byte)*
Start address		*(Word)*
Length		*(Word)*
Empty block	=FFh	(Byte)
Length empty block	=02h	(Byte)

Acknowledgement telegram CP

System ID	="S5"	(Word)
Length Header	=10h	(Byte)
ID OP-Code	=01h	(Byte)
Length OP-Code	=03h	(Byte)
OP-Code	**=06h**	(Byte)
Ackn. block	=0Fh	(Byte)
Length Ackn. block	=03h	(Byte)
Error no.		(Byte)
Empty block	=FFh	(Byte)
Length empty block	=07h	(Byte)
5 empty bytes attached		
Data up to 64kByte (only if error no.=0)		

*) More details to the data area is to be found at "ORG-Format" above.

Note!

Please regard that in opposite to Siemens-S5 systems, the block addressing of these CPUs takes the start address as byte number and the length as number of words.

Messages of error no.

The following messages can be returned via *error no.*:

Error no.	Message
00h	No error occurred
01h	The defined area cannot be read res. written

Example communication CPU 31xSN/NET - CPU 31xSN/NET

Overview

This chapter provides an introduction to use the TCP/IP bus system for the System 300S. The object of this chapter is to create a small communication system between two VIPA CPUs 31xSN/NET that provides a simple approach to the control of the communication processes.

Preconditions

Knowledge of the VIPA CP handling blocks AG_SEND and AG_RECV is required. CP handling blocks provide the options required to utilize the communication functions in the programs of the PLCs.

The minimum technical equipment required for the example is as follows:

Hardware

- 2 CPUs 31xSN/NET from VIPA
- 1 PC or PG with Twisted Pair Ethernet connection

Communication line

- 3 bus cables
- 1 Swich/Hub

Addresses

- 4 IP Addresses and subnet masks for each 2 CPs

Software package

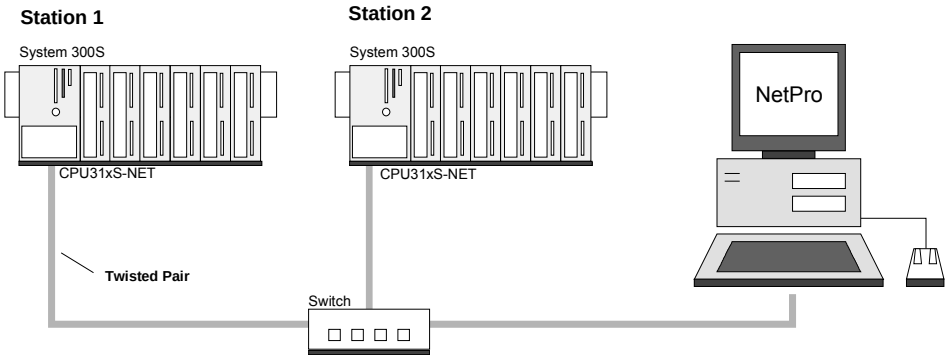
- SIMATIC Manager from Siemens V. 5.1 or higher
- SIMATIC NET

The implementation of the example requires that the two CPUs be programmed as well as the configuration of the CPs by means of NetPro from Siemens.

Note!

The complete example is to e found as zip at ftp.vipa.de/support/demofiles. You may transfer the PLC program directly to both CPUs.

Structure



Station tasks

The example for the application is based upon a communication task that is described in detail in the following passage:

Both of the CPUs contain the same PLC program, only the configuration of the CPs have to be adjusted.

Both stations are sending and receiving 16 data words per second.

- Data block DB 11 transfers the data bytes DBB 0 to DBB 32 at an interval of 1s. Data byte DBB 0 in DB 11 is used as message counter. It is only incremented if the preceding transmit command was processed correctly (completed without error). The remaining data words (DBB 2 to DBB 32) can be used for the transfer of user data.
- The receiving station stores the data in DB 12 (DBB 0 to DBB 31).
- Using NetPro an active SEND/RECEIVE connection with ID 1 is to be configured for the 2nd CP. This Connection is established at station 2 as a passive SEND/RECEIVE connection.
- The source and destination parameters must be configured directly.

At this point the purpose and the required settings have been outlined. The programs provide additional details of the configuration of the handler blocks. A detailed description follows.

Steps of project engineering

The project engineering is divided into the following steps:

- Project engineering of the CPU
- Project engineering of the really plugged modules at the standard bus
- Project engineering Ethernet PG/OP channel and *CP 343*
- Project engineering in NetPro
- PLC user application
- Transfer project

Project engineering of the CPU of Station 1

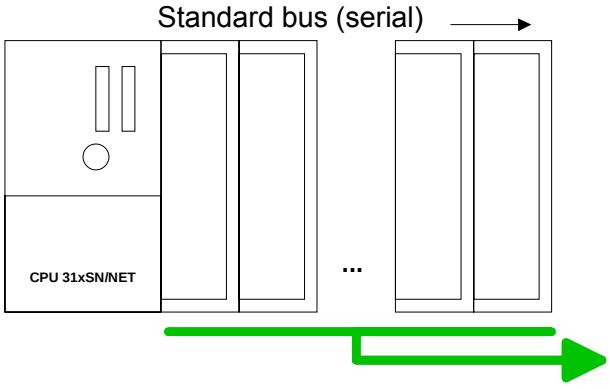
- Start the Siemens SIMATIC Manager with a new project
- Insert a "SIMATIC 300 Station" and rename it to "Station 1"
- Start the hardware configurator and add a profile rail from the hardware catalog.
- Place the following Siemens CPU at plug-in location 2:
  CPU 318-2DP (6ES7 318-2AJ00-0AB0 V. 3.0)

Project engineering of the modules at the standard bus

The modules at the standard bus at the right side of the CPU are configured like this:

- Include your System 300V modules at the standard bus in the plugged sequence starting with plug-in location 4.
- Parameterize the CPU res. the modules if needed. The parameter window appears as soon as you double click on the according module.
- For bus extension you may use the IM 360 from Siemens where you may link-up up to 3 extension racks via the IM 361. Bus extensions may only be placed at plug-in location 3.

Project engineering of the modules at the standard bus



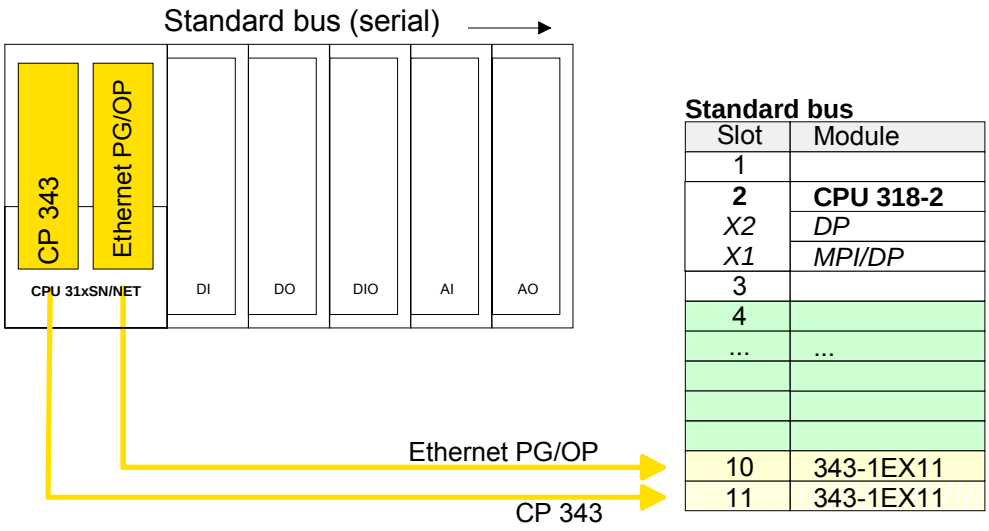
Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
4	
...	...
11	

Project engineering of the integrated CPs

For the internal Ethernet PG/OP channel that is included in every SPEED7-CPU, you have to place as 1st module below the really plugged modules a Siemens CP 343-1 (343-1EX11).

The integrated CP 343 of the CPU 31xSN/NET is also configured as CP 343-1 but always below the CP 343-1 placed before. Within "Properties" type a valid IP address, subnet mask and if needed a gateway for the CPs and connect them via "Ethernet".
Save and compile your project.



Standard bus

Slot	Module
1	
2	**CPU 318-2**
X2	*DP*
X1	*MPI/DP*
3	
4	
...	...
10	343-1EX11
11	343-1EX11

Project engineering of the CPU of Station 2

Create, following the approach above, a hardware configuration for the destination CPU and assign the name "Station 2".

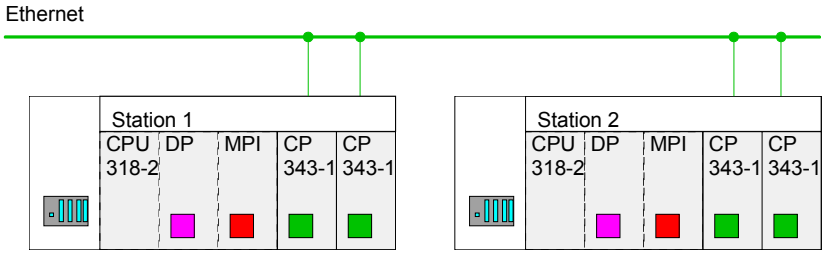
For the CPs, use the IP addresses, subnet masks and gateways assigned to Station 2.

Save and compile your project.

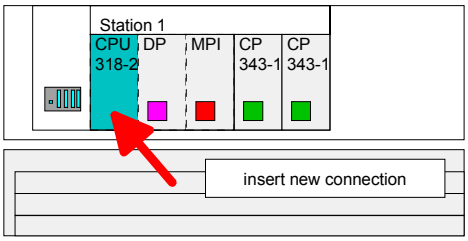
Project engineering with NetPro

Start NetPro by selecting the CPU below *Station 1* and clicking on the object "connections".

In NetPro "Station 1" and "Station 2" are listed together with Ethernet.



To configure the connection open the connection list. For this you choose the CPU of Station 1 and call *Insert new connection* via the context menu:



A dialog window appears where you can select the connection partner and the type of the connection.

Configure the following connection:

New connection

Connection: TCP connection

Connection partner: Station 2 > CPU 318-2

Properties TCP connection

ID: 1

ID and *LADDR* are parameters that you have to define in your PLC program if using FC5 (AG_SEND) and FC6 (AG_RECEIVE).

Route: The *Route* allows you to choose the CP that has to manage the connection.

As 1st CP the list always shows the integrated Ethernet PG/OP channel which only supports PG/OP communication.

For the communication via the internal CP 343 you always have to use the 2nd CP of the route. E.g. when no local modules are plugged choose "CP 343-1 - (R0/S5)".

Active connection establishment: activated

Save and compile your connection.

PLC user program

For the processing of connection commands at the PLC, a PLC user program is necessary in the concerning CPU. For this only the handling blocks AG_SEND (FC 5) and AG_RECV (FC 6) are used. By including this blocks into the cycle block OB1 with the parameters *ID* and *LADDR* you may cyclically send and receive data.

The two FCs are part of the VIPA library that is included in the consignment of the CPU as CD.

OB 1 Cycle

Via the cycle OB OB 1 the sending and receiving of the data is controlled. The OB1 that you may transfer into both CPUs has the following structure:

```
      UN    T      1              // Timer 1 triggered sending
      L     S5T#1S                // Send initiation every 1 sec
      SV    T      1
      S     M     10.0            // Init bit memory
      CALL  "AG_SEND"
      ACT   :=M10.0               // Init bit memory
      ID    :=1                   // Connection number
      LADDR :=W#16#110            // Module address
      SEND  :=P#DB11.DBX0.0 BYTE 100   // Send buffer area DB11
      LEN   :=32                  // send 32 Byte (16 Words) from DB11
      DONE  :=M10.1
      ERROR :=#Senderror          // Temporary error bit memory
      STATUS:=MW12                // Order res. connection state
      U     M     10.1            // Send ready?
      SPBN  nDon
      U     M     10.1            // Send ready?
      R     M     10.0            // Set back init
      U     #Senderror            // At send error
      SPB   nDon                  // Don't raise send counter
      L     DB11.DBW    0         // Send counter in user data (DBW0)
      L     1                     // increment for 1 and
      +I                          // store again in send buffer
      T     DB11.DBW    0

nDon: NOP   0                     // Send not ready yet

      // Cyclic call of the receive block

      CALL  "AG_RECV"
      ID    :=1                   // Connection number
      LADDR :=W#16#110            // Module address
      RECV  :=P#DB12.DBX100.0 BYTE 32  //Receive buffer
      NDR   :=#Newdata            // NewDataReceived?
      ERROR :=M0.1                // RecError
      STATUS:=MW2                 // Order res. connection state
      LEN   :=#Reclen             // Really received length
      NOP   0                     // Reclen can be at IsoOnTCP < 32
      U     #Newdata              // when new data received
      ZV    Z      1              // Increment Receive counter Counter1
      L     Z      1              // reset counter 1 at overflow
      L     999
      ==I
      R     Z      1
```

Project transfer

Information about transferring a project may be found at chapter "Deployment CPU 31xS" at "Project transfer".

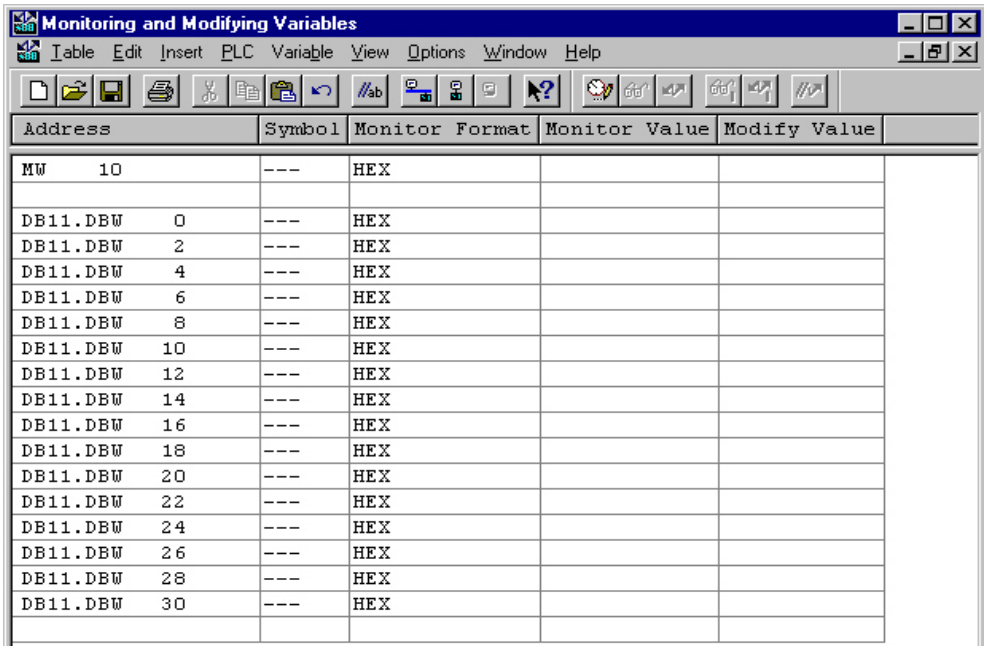
Monitoring the data transfer in the Siemens SIMATIC Manager

It is assumed, that the CPs are programmed and that an overall reset was issued to the CPUs, where the RUN/STOP switch must be located in STOP position.

Now load the above PLC programs into both CPUs and switch them into RUN.

Start the Siemens STEP®7 Manager and execute the following steps to monitor the transmit job:

- **PLC** > *Monitor/Modify Variables*
- In the column "Operand" you have to enter the respective data block number and the data word (DB11.DBB 0-31).
- Establish a connection and click "monitor"  .



Address	Symbol	Monitor Format	Monitor Value	Modify Value
MW 10	---	HEX		
DB11.DBW 0	---	HEX		
DB11.DBW 2	---	HEX		
DB11.DBW 4	---	HEX		
DB11.DBW 6	---	HEX		
DB11.DBW 8	---	HEX		
DB11.DBW 10	---	HEX		
DB11.DBW 12	---	HEX		
DB11.DBW 14	---	HEX		
DB11.DBW 16	---	HEX		
DB11.DBW 18	---	HEX		
DB11.DBW 20	---	HEX		
DB11.DBW 22	---	HEX		
DB11.DBW 24	---	HEX		
DB11.DBW 26	---	HEX		
DB11.DBW 28	---	HEX		
DB11.DBW 30	---	HEX		

Entering User data

You may enter user data starting with DBB 2. Place the cursor on *modify value* and enter the value you wish to transfer, e.g. W#16#1111.

The  button transfers the modify value in every cycle and the  button initiates a single transfer.

Appendix

A Index

M.Stich