VIPA System 500S

SPEED7 - CPU | 517-2AJ02 | Manual

HB145E_CPU | RE_517-2AJ02 | Rev. 10/23

June 2010

VIPA art of automation

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

Note

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. Nevertheless, the authors retain the right to modify the information. This customer document describes all the hardware units and functions known at the present time. Descriptions may be included for units which are not present at the customer site. The exact scope of delivery is described in the respective purchase contract.

CE Conformity

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

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

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

Conformity Information

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

Trademarks

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

SPEED7 is a registered trademark of profichip GmbH.

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

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

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

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

Information product support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

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

Technical support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telephone: +49 9132 744 1150/1180 (Hotline)
EMail: support@vipa.de

Contents

About this manual **1**
Safety information **2**
Chapter 1 Basics **1-1**
Safety Information for Users 1-2
Principles of Net-ID, Subnet-ID, Host-ID 1-3
Operating structure of a CPU 1-6
CPU Applications 1-7
Operands of the CPU 1-7
CPU 517S/DPM 1-9
Chapter 2 Hardware description **2-1**
Properties 2-2
Structure 2-3
Components 2-4
Technical Data 2-9
Chapter 3 Deployment CPU 517S/DPM **3-1**
Overview 3-2
Assembly 3-3
Installation of the driver 3-4
Guidelines for IP address assignment 3-5
Connect power supply 3-8
Initialization of the CPU component 3-10
Internal access to PG/OP channel 3-14
External access to PG/OP channel via routing 3-15
Access to the integrated web page 3-18
Project engineering 3-19
CPU parameterization 3-23
Project transfer 3-34
Operating modes 3-39
Overall reset 3-42
Firmware update 3-44
Factory reset 3-47
Memory expansion with MCC 3-48
Extended know-how protection 3-49
MMC-Cmd - Auto commands 3-51
VIPA specific diagnostic entries 3-53
Using test functions for control and monitoring of variables 3-57
Chapter 4 Deployment CPU with Profibus **4-1**
Overview 4-2
Project engineering CPU with integrated Profibus master 4-3
Deployment as Profibus DP slave 4-5
Project transfer 4-7
Profibus installation guidelines 4-8
Commissioning and Start-up behavior 4-11

Chapter 5 Deployment PtP communication 5-1
Fast introduction 5-2
Principle of the data transfer 5-3
Deployment of RS485 interface for PtP 5-4
Parameterization 5-6
Communication 5-9
Protocols and procedures 5-15
Modbus - Function codes 5-19
Modbus - Example communication 5-23
Chapter 6 Deployment PLC-Tool 6-1
General 6-2
Setup and run of program 6-3
PLC-Tool Operation 6-4
Deployment PLC-Tool 6-7
Chapter 7 WinPLC7 7-1
System presentation 7-2
Installation 7-3
Example project engineering 7-4
Appendix A-1
Index A-1

About this manual

This product supplement contains all information required for the deployment of the Slot-PLC 517S/DPM in your PC. The here described Slot-PLC is a SPEED7 CPU 517S/DPM with integrated Profibus-DP master. The CPU is included in the PC as Ethernet interface and can be accessed via the IP address.

Overview

Chapter 1: Basics

This Basics contain hints for the usage and information about the project engineering of a System 500S SPEED7 from VIPA.
Basic information about the structure of IP addresses can be found further.

Chapter 2: Hardware description

In this chapter the hardware components of the CPU 517S/DPM are more described here.

The chapter closes with the technical data.

Chapter 3: Deployment CPU 517S/DPM

Main topic of this chapter is the deployment of the CPU 517S/DPM from VIPA. Here information necessary for installation, start-up and project engineering can be found.

Chapter 4: Deployment CPU with Profibus

Content of this chapter is the deployment of the CPU 517S/DPM with Profibus. After a short overview the project engineering and parameterization of a CPU 575S/DPM 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.

Chapter 5: Deployment PtP Communication

In this chapter the deployment of the RS485 slot for serial PtP communication is described.
Here you'll find all information about the protocols and project engineering of the interface, which are necessary for the serial communication using the RS485 interface.

Chapter 6: Deployment PLC-Tool

This chapter contains the description of the control software *PLC-Tool* from VIPA. PLC-Tool is a component of the OPC-Server package and is installed together with the OPC server at the standard installation.

The OPC-Server package may be found at the enclosed CD *SW-ToolDemo*.

Chapter 6: WinPLC7

In this chapter the programming and simulation software WinPLC7 from VIPA is presented. WinPLC7 is suited for every with Siemens STEP®7 programmable PLC.

Besides the system presentation and installation here the basics for using the software is explained with a sample project.

More information concerning the usage of WinPLC7 may be found in the online help respectively in the online documentation of WinPLC7.

Objective and contents

This manual describes the System 500S SPEED7 517S/DPM from VIPA. It contains a description of the construction, project implementation and usage.

This manual is part of the documentation package with order number HB145E_CPU and relevant for:

Product	Order number	as of state:		
		CPU-HW	CPU-FW	DPM-FW
CPU 517S/DPM	VIPA 517-2AJ02	01	V351	V326

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 CPU is constructed and produced for:

- 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 System 500S SPEED7 from VIPA.

Basic information about the structure of IP addresses can be found further.

Content

Topic	Page
Chapter 1 Basics	**1-1**
Safety Information for Users	1-2
Principles of Net-ID, Subnet-ID, Host-ID	1-3
Operating structure of a CPU	1-6
CPU Applications	1-7
Operands of the CPU	1-7
CPU 517S/DPM	1-9

Safety Information for Users

Handling of electrostatic sensitive modules

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

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

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

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

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

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

Shipping of electrostatic sensitive modules

Modules must be shipped in the original packing material.

Measurements and alterations on electrostatic sensitive modules

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

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

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

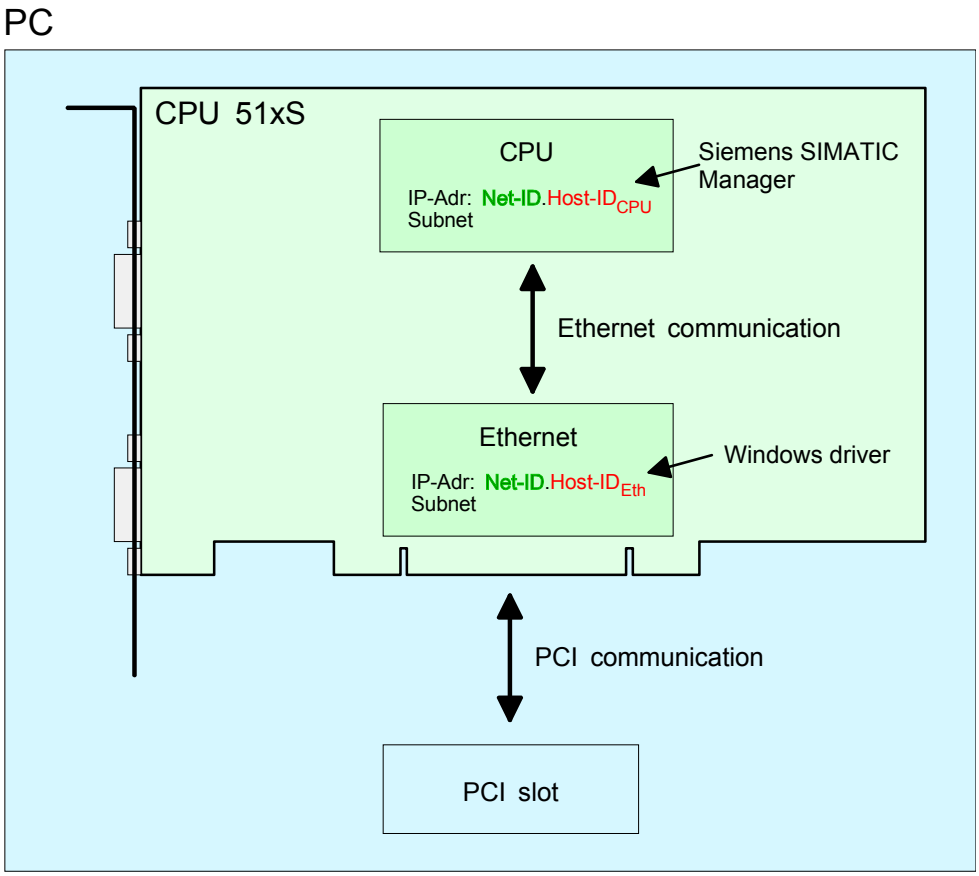
Attention!

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

Principles of Net-ID, Subnet-ID, Host-ID

Why Principles?

The CPU 51xS PC plug-in card consists of a CPU and an Ethernet portion that communicate via a TCP-based point-to-point connection. To enable this, CPU and Ethernet portion each have an alterable IP address that may only differ in the Host-ID.



If you want to install several CPU 51xS in one PC, every CPU 51xS plug-in card needs an own Net-ID.

The following text describes the approach for the assignment of IP addresses together with Net-ID and Host-ID.

Net-ID
Host-ID

Every IP address is a combination of a **Net-ID** and a **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

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 after your needs. This means e.g. that every department gets an own subnet and thus does not interfere another department.

Note!

When using the CPU 51xS in your PC, the Net-ID of the CPU 51xS must not be assigned to another device. Otherwise you have to reassign the addresses.

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+7 bit)	Host-ID (24 bit)
Class B	10	Network-ID (2+14 bit)	Host-ID (16 bit)
Class C	110	Network-ID (3+21 bit)	Host-ID (8 bit)
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 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.)

Operating structure of a CPU

General

The CPU contains a standard processor with internal program memory. In connection with the integrated SPEED7 Technology receive you efficient equipment for process automation.

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 part 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 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

Overview

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

- 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

CPU 517S/DPM

Overview

The CPU 517S/DPM is a fully adequate PLC-CPU in form of a PCI-slot card for PC-based applications. The operating systems Windows® 98, ME, NT4, 2000 and XP are supported.

The range of performance is adequate to a SPEED7 CPU from the System 300S from VIPA. The programming takes place via standard programming tools like e.g. WinPLC7 from VIPA or STEP®7 from Siemens.

For the link up to the process level there is as well a MPI as a Profibus DP master interface.
Further on, the VIPA OPC-Server is included in the delivery.

After the hardware installation, the card is linked up to the PC as COM interface. The CPU component of the CPU 517S slot card can only be operated with an external or internal DC 24V power supply. The external supply enables the operation of the card outside of a PC res. independent from the PC operation. Please consider for operation that the slot card is connected to ground via its metal cover.

Memory management

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

There is the possibility to extend the total memory to its maximum by means of a MCC memory extension card.

Integrated Ethernet-PG/OP-channel

The CPU has an Ethernet interface for PG/OP communication. Only in installed condition you have access via the PG/OP channel to your CPU. Here you may program, remote control or show the integrated web page. There are maximum 4 channels.

Integrated Profibus DP master

The CPU has an integrated Profibus DP master. Via the DP master with a data range of 1kByte for in- and output up to 124 DP slaves may be addressed. The project engineering takes place in WinPLC7 from VIPA or in the hardware configurator from Siemens. Please regard there may be a delimitation of the maximum number of configurable DP slaves by the use of the Siemens SIMATIC manager.

The Profibus part may also be used as "intelligent" DP slave. More may be found at "Deployment CPU with Profibus".

During operation the data range of the DP master is monitored at an adjustable address area of the CPU. The address area may be by hardware configuration.

MPI interface

The Slot PLC includes a MPI interface. On delivery the default MPI address is 2. This address my be changed at any time via your CPU project engineering tool.

Operating security

- External power supply of the CPU (autarkic operation)
- 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)

Environment conditions

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

Compatibility

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 CPU 517S/DPM from VIPA.

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

Power supply

The CPU component of the 517S slot card can only be operated with an external or internal DC 24V power supply. Simultaneous infeed should absolutely be avoided!

Please consider for operation that the slot card is connected to ground via its metal cover.

When connecting please consider that the internal power supply hardware conditionally does not have an EMV filter for protection against disturbances.

Operating options via PLC-Tool

For operating the CPU via the PC the program "PLC-Tool" is included in the consignment. For monitoring and operating of the CPU, your PC shows an user interface that is modeled on the schematic view on a CPU front.

Via the PLC-Tool you may request the LED state and monitor res. change the operating mode of the CPU.

Chapter 2 Hardware description

Overview

In this chapter the hardware components of the CPU 517S/DPM are more described here.

The chapter closes with the technical data.

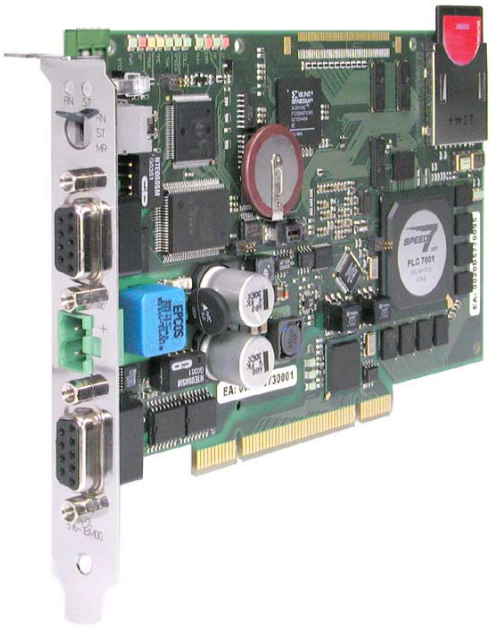
Content

Topic	Page
Chapter 2 Hardware description	**2-1**
Properties	2-2
Structure	2-3
Components	2-4
Technical Data	2-9

Properties

CPU 517S/DPM
517-2AJ02

- Integrated SPEED7 technology
- Instruction compatible to STEP®7 from Siemens
- Project engineering via the SIMATIC Manager from Siemens
- Integrated DC 24V power supply
- MPI with max. 32 PG/OP connections with up to 12Mbit/s
- Status-LEDs for operating state and diagnosis
- Battery buffer for RAM and clock
- Integrated Profibus DP master
- Integrated work memory 2Mbyte expandable to 8Mbyte (50% program - 50% data)
- Storage media slot for project engineering and firmware update
- 2048Timer, 2048 Counter, 16384 Memory-Byte

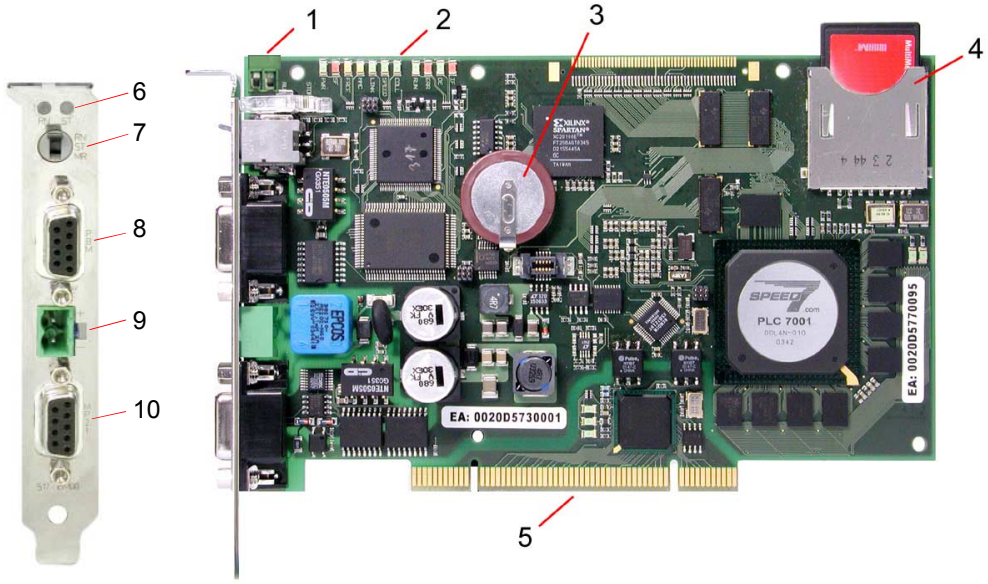


Ordering data

Type	Order number	Description
517S/DPM	VIPA 517-2AJ02	2Mbyte work memory expandable up to 8Mbyte (50% program - 50% data), external power supply, MPI, MMC slot, real time clock, Interface: Profibus DP master, 12Mbit/s, up to 124 slaves, PCI-Ethernet interface for PG/OP communication, incl. SW110A2LA OPC-Server (SW110A2LA please order separate). Incl. Drivers and SW860R OPC-Server (on the ToolDemo-CD).

Structure

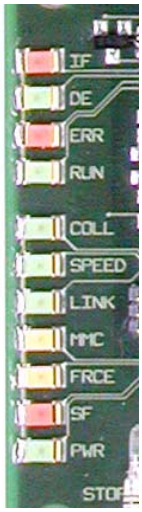
CPU 517S/DPM
517-2AJ02



[1] Clamp for internal DC 24V power supply
[2] LEDs for commissioning
[3] Lithium accumulator/battery for clock and user memory
[4] Storage media slot (here MMC is plugged)
[5] PCI bus pins
[6] RUN/STOP LEDs
[7] Operating mode switch
[8] Profibus DP master jack
[9] Plug for external DC 24V power supply
[10] MPI jack

Components

LED bar



On the plug-in module you can see a LED bar for status monitoring of the CPU and the Profibus DP master. Especially at the commissioning and the external usage of the module, the state of your CPU and your Profibus-DP master is shown.

At deployment inside a PC, you may issue the state of the LEDs on your PC via the delivered software PLC-Tool.

The usage and the according colors of the LEDs are to see in the following tables:

Master operation

RUN green	ERR red	DE green	IF red	Meaning
○	○	○	○	Master has no project, this means the DP interface is not used.
●	○	☼	○	Master is in "clear" state (safety state). The inputs of the slaves may be read. The outputs are disabled.
●	○	●	○	Master is in "operate" state (CPU RUN), 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 (state at start-up).

Slave operation

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

on: ● off: ○ flashing: ☼

... continue LEDs

Label	Color	Description
Ethernet		
COLL	green	Collision: on: total duplex operation active off: half duplex operation active blinking: Collision detected
SPEED	green	Speed: on: 100Mbit off: 10Mbit
LINK	green	Link on: physical connection detected off: no physical connection
CPU		
MMC	yellow	blinks at MMC access
FRCE	yellow	blinks as soon as variable are forced (fixed)
SF	red	blinks at system errors (hardware defect)
PWR	green	CPU section is provided internal with 5V

LEDs at connection panel

Above the operating mode lever there are 2 LEDs, showing the operating state:

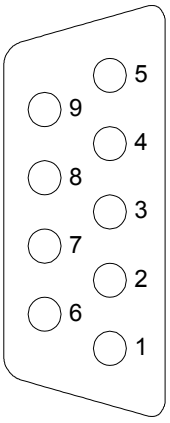
Label	Color	Description
ST	yellow	CPU is in STOP
RN	green	CPU is in RUN

Jacks and plugs

On the PC plug-in module the following jacks are led out:

Profibus DP master interface PBDP/PtP

Via the 9pin RS485 interface you link up the integrated Profibus DP master to Profibus. The RS485 interface in *PtP* operation supports the serial process connection to different source or destination systems.
The RS485 jack has the following pin occupancy:



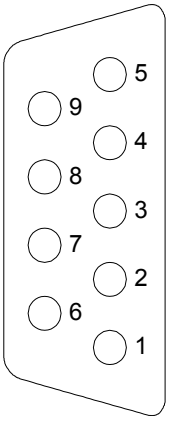
9-pin jack

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.

Note!
Please make sure to activate the terminating resistors at the bus ends!

MPI interface

MPI serves the connection to the process level. Here you may transfer programs and data between the MPI participants. On delivery the MPI address is 2.
For a serial transfer from your PC you normally need a MPI transducer.
The MPI jack has the following pin assignment:



9-pin jack

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

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 configured MMC as 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.

Memory management

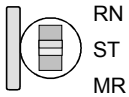
Every CPU 51xS has an integrated work memory. During program run the total memory is divided into 50% for program code and 50% for data.

There is the possibility to extend the total memory to its maximum by means of a **M**emory **C**onfiguration **C**ard called MCC.

Attention!

At deployment of a MMC, please regard, that it has to be preformatted with the FAT16 file system. The VIPA memory cards are always delivered preformatted.

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).

Power supply

After the CPU 51xS slot card is installed in the PC and the communication between PC and Ethernet component is established the DC 24V power supply may be attached.

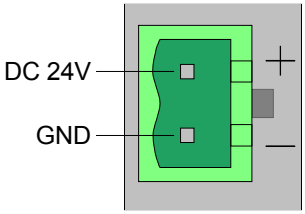
The CPU component of the CPU 51xS slot card can only be operated with an external or internal DC 24V power supply. The external supply enables the operation of the card outside of a PC res. independent from the PC operation. Please consider for operation that the slot card is connected to ground via its metal cover.

Note!

The CPU 51xS is to supply either externally or internally with DC 24V.
Simultaneous infeed should absolutely be avoided!

External power supply

For external power supply there is a plug on slot panel with the following pin assignment:



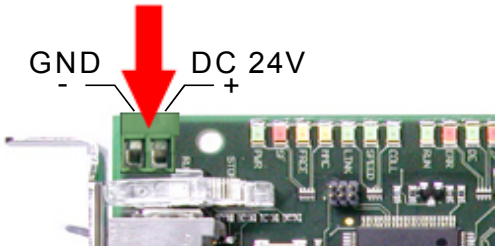
Internal power supply

For the internal power DC 24V supply there is a clamp on the top of the slot card.

Please consider when connecting a DC 24V power supply that the internal power supply hardware conditionally does not have an EMC filter for protection against disturbances (like e.g. EN 61000-4-4 [Burst], EN 61000-4-5 [Surge] or EN 61000-4-6 [conducted disturbance variable, inducted by HF fields]).

Here please use an accordingly filtered supply voltage here.

The clamp has the following pin assignment:



Battery buffer for clock and RAM

The CPU 51xS contains an internal accumulator/battery for protecting the RAM at a power break-down. Additionally the battery buffers the internal clock.

The battery is directly reloaded via the integrated voltage supply by means of a special loading electronic and guarantees a buffer of minimum 30 days.

The battery has to be error free, so that the CPU may automatically restart.

A start with a defective battery is possible, if it is manually switched to RUN or if a MMC is plugged, which contains a valid s7prog.wld project with an OB81 for battery fault.

If there is an error concerning the integrated battery, the CPU should be checked. Here please contact VIPA!

Technical Data

Order number	VIPA 517-2AJ02
Type	CPU 517S/DPM
Technical Data power supply	
Power supply (rated value)	DC 24V
Power supply (permitted range)	DC 20.4...28.8V
Reverse polarity protection	yes
Current consumption (no-load operation)	250mA
Current consumption (rated value)	1A
Inrush current	5A
Load and working memory	
Load memory, integrated	8MB
Load memory, maximum	8MB
Work memory, integrated	2MB
Work memory, maximum	8MB
Memory divided in 50% program / 50% data	yes
Memory Card Slot	MMC-Card with max. 1GB
Hardware config	
Number of DP master integrated	1
Status information, interrupts, diagnostics	
Status display	yes
Interrupts	no
Process interrupt	no
Diagnostic interrupt	no
Command processing times	
Bit instructions, min.	0.01µs
Word instructions, min.	0.01µs
Double integer arithmetic, min.	0.01µs
Floating-point arithmetic, min.	0.06µs
Timer/Counter and retentive characteristic	
Number of S7 counters	2048
Number of S7 timers	2048
Data range and retentive characteristic	
Number of flags	16384Byte
Number of data blocks	8190
Max. data blocks size	64KB
Max. local data size per execution level	510Byte
Blocks	
Number of OBs	24
Number of FBs	8191
Number of FCs	8191
Maximum nesting depth per priority class	8
Maximum nesting depth additional within an error OB	4
Time	
Real-time clock buffered	yes
Clock buffered period (min.)	6 weeks
Accuracy (max. deviation per day)	10s
Number of operating hours counter	8
Clock synchronization	yes
Synchronization via MPI	MasterSlave
Synchronization via Ethernet (NTP)	no
Address areas (I/O)	
Input I/O address area	8192Byte
Output I/O address area	8192Byte
Input processing image maximal	8192Byte

Order number	VIPA 517-2AJ02
Output processing image maximal	8192Byte
Digital inputs	65536
Digital outputs	65536
Analog inputs	4096
Analog outputs	4096
Communication functions	
PG/OP channel	yes
Global data communication	yes
Number of GD circuits, max.	16
Size of GD packets, max.	54Byte
S7 basis communication	yes
S7 basis communication, user data per job	76Byte
S7 communication	yes
S7 communication as server	yes
S7 communication, user data per job	160Byte
Number of connections, max.	32
Functionality Sub-D interfaces	
Type	X2
Type of interface	RS485
Connector	SubD, 9-pin, female
Electrically isolated	yes
MPI	yes
Type	X3
Type of interface	RS485
Connector	SubD, 9-pin, female
Electrically isolated	yes
DP master	yes
DP slave	yes
Functionality Profibus master	
PG/OP channel	yes
Routing	yes
S7 basis communication	yes
S7 communication	yes
S7 communication as Server	yes
SYNC/FREEZE	yes
Activation/Deactivation of DP slaves	yes
DPV1	yes
Transmission speed, min.	9.6kbit/s
Transmission speed max.	12Mbit/s
Number of DP slaves, max.	32
Address range inputs, max.	1KB
Address range outputs, max.	1KB
User data inputs per slave, max.	244Byte
User data outputs per slave, max.	244Byte
Functionality Profibus Slave	
PG/OP channel	yes
Routing	yes
S7 communication	yes
S7 communication as Server	yes
DPV1	yes
Transmission speed min.	9.6kbit/s
Transmission speed max.	12Mbit/s
Transfer memory inputs, max.	244Byte
Transfer memory outputs, max.	244Byte
Address areas, max.	32
User data per address area, max.	32Byte

Order number	VIPA 517-2AJ02
Functionality PCI interfaces	
Type	n/a
Type of interface	Ethernet 10/100Mbit
Connector	PCI bus
Electrically isolated	yes
PG/OP channel	yes
Mechanical Data	
Dimensions (WxHxD)	20mm x 106mm x 174mm
Weight	290g
Environment conditions	
Operating temperature	0°C to 60°C
Storage temperature	-25°C to 70°C
Certification	
UL508 certification	in preparation

Chapter 3 Deployment CPU 517S/DPM

Overview

Main topic of this chapter is the deployment of the CPU 517S/DPM from VIPA. Here information necessary for installation, start-up and project engineering may be found.

Content

Topic	Page
Chapter 3 Deployment CPU 517S/DPM	**3-1**
Overview	3-2
Assembly	3-3
Installation of the driver	3-4
Guidelines for IP address assignment	3-5
Connect power supply	3-8
Initialization of the CPU component	3-10
Internal access to PG/OP channel	3-14
External access to PG/OP channel via routing	3-15
Access to the integrated web page	3-18
Project engineering	3-19
CPU parameterization	3-23
Project transfer	3-34
Operating modes	3-39
Overall reset	3-42
Firmware update	3-44
Factory reset	3-47
Memory expansion with MCC	3-48
Extended know-how protection	3-49
MMC-Cmd - Auto commands	3-51
VIPA specific diagnostic entries	3-53
Using test functions for control and monitoring of variables	3-57

Overview

Functionality
Ethernet (LAN)
CPU component

The CPU 51xS PC plug-in card consists of an *Ethernet* (LAN) and a *CPU* component. These communicate internally over an Ethernet connection. For this both components are each be assigned to an IP address, which may differ only in the Host-ID. In this way several CPU 51xS may be operated in your PC.

The CPU component of the CPU 51xS slot card can only be operated with an external or internal DC 24V power supply. The external supply enables the operation of the card outside of a PC res. independent from the PC operation. Please consider for operation that the slot card is connected to ground via its metal cover.

IP address parameter assignment

The assignment of IP address parameters for the Ethernet component is made by the Windows operating system by the *Network environment*

The CPU component receives its IP address parameters by the Siemens SIMATIC manager by means of the *PLC functions* or a *minimal project.*

The addresses are not affected by an *overall reset.* The IP address parameters of the CPU component are deleted by a *factory reset*.

Steps of installation

- Install the CPU 51xS PC slot card at a free 32bit PCI slot (PCI version 2.2, 32bit data/address bus, 3.3V current).
- Switch PC on.
- Install driver for the Ethernet component. This can be found at the "ToolDemo-CD" SW900TOLA at driver/slotplc the according CPU 51xS.
- Set IP address and subnet mask for the Ethernet component of the slot card by means of *Properties* of the *Network environment.* Here the IP addresses may only differ in the Host-ID.
- Supply CPU component with DC 24V.
- Assign IP address parameters to the CPU component by means of Siemens SIMATIC manager. There are the following possibilities for assigning IP address parameters (Initialization):
  - PLC functions with *Edith Ethernet Node* (search PLC and assign IP address)
  - Hardware project engineering with CP (Minimal project)

If the "Net" components are installed at your Siemens SIMATIC manager the CPU 51xS is displayed as **Intel(R) 8255xER PCI Adapter** at the *PG/PC interface* area. The CPU can online be accessed by the PG/OP channel.

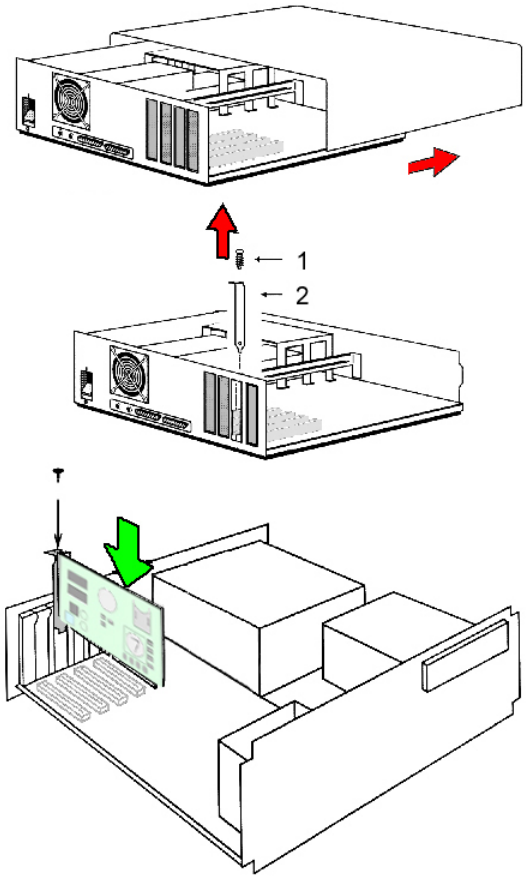
The steps are detailed described at the following pages.

Assembly

Assembly

- Eliminate possibly existing static loading, before mounting the VIPA PCI card, by affecting a grounded metal object.
- Switch your computer off and remove the power cable.
- Remove the covers from your computer according to the manufacturers instructions.
- Search a free 32bit PCI slot (usually white ore beige). Please regard that your PCI bus corresponds to the following specification:
  PCI version 2.2 , 32Bit data/address bus, 3.3V voltage.
- Remove one of the metal covers from a slot.
- Insert the slot card, bolt the slot card with the computer case and close the covers of your computer.

Please consider that the metal cover of the slot card is always connected to ground and bolted with the computer case!



[1] Screw
[2] Metal slot cover

Note!

The installation should only be accomplished by experienced technical personnel!

Slot card or PC may be damaged by an incorrect installation.

Installation of the driver

Overview

A driver is necessary for the integration of the slot card into the operating system. The slot card driver may be found at the enclosed CD SW900TOLA. For start-up the power supply of the CPU component is not for the time being necessary.

Start-up without external CPU power supply

- Turn the PC on after mounting the slot card. The slot card is recognized as a new network hardware after run-up of the PC and the appropriate driver is requested. The driver can be found on the enclosed "ToolDemo CD" SW900TOLA.
- Put in the CD and navigate via driver/slotplc to the directory from the according PLC 51xS. Here the slot card driver for each relevant operating system may be found.
- Install the slot card driver.

The CPU 51xS slot card is now specified to the operating system and is listed as an additional LAN connection with the device name "Intel(R) 8255 PCI adapters". As long as the CPU component is not supplied with DC 24V the message appears "The network cable was removed".

Assign IP address parameters to the Ethernet component

After installation of the driver IP address and Subnet mask can be assigned by the *properties* of the *network environment* at any time. Attention should be paid that the IP address of the CPU and Ethernet component differs exclusively in the Host-ID.

Note!

More information about assigning IP addresses may be found at "Guidelines for IP address assignment" at the following pages.

Please consider the guidelines, since incorrect adjusted IP address parameters may have effects on the whole firm net.

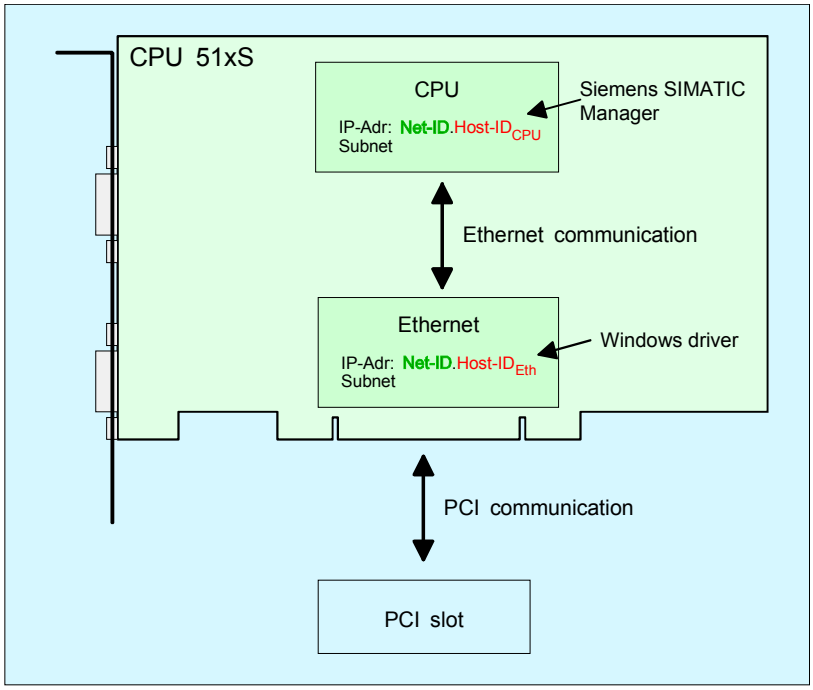
Guidelines for IP address assignment

Overview

The CPU 51xS slot card consists of a CPU and a Ethernet component that communicate via a TCP-based point-to-point connection. To enable this, CPU and Ethernet component each have an alterable IP address that may only differ in the Host-ID.

For start-up and for further comprehension, a thorough knowledge of Net-ID, Host-ID and Subnet-ID are assumed. In the following detailed information about this can be found.

PC



If you want to install several CPU 51xS in one PC, every CPU 51xS plug-in card needs an own Net-ID.

The following text describes the approach for the assignment of IP addresses together with Net-ID and Host-ID.

Net-ID
Host-ID

Every IP address is a combination of a **Net-ID** and a **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.

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.)

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

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 after your needs. This means e.g. that every department gets an own subnet and thus does not interfere another department.

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+7 bit)	Host-ID (24 bit)	
Class B	10	Network-ID (2+14 bit)		Host-ID (16 bit)
Class C	110	Network-ID (3+21 bit)		Host-ID (8 bit)
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 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.

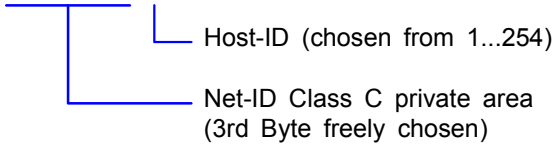
Host-ID = 0	Identifier of this network, reserved!
Host-ID = maximum (binary complete 1)	Broadcast address of this network

Example network planning for start-up

In common, your complete network consists of a PC with (at least) one network card and one or more CPU 51xS plug-in cards that are also each listed as network card with the CPU as single participant:

In this example for Ethernet and CPU component IP addresses from private class C net were selected. Using the Subnet mask 255.255.255.0 256 different networks with 254 host addresses each are available

CPU 51xS	
LAN with recomm. IP/Mask	192.168.**201**.2 / 255.255.255.0
CPU with example IP/Mask	192.168.**201**.3 / 255.255.255.0



To enable the PC to connect the CPU 51xS plug-in cards and the subordinated CPUs without using a router table, you have to assign an **individual Net-ID** for every card!

PC

Network card (IP/Mask from DHCP-Server)	
e.g.	192.168.**1**.2 / 255.255.255.0
1. CPU 51xS	
LAN with example IP/Mask	192.168.**201**.2 / 255.255.255.0
CPU with example IP/Mask	192.168.**201**.3 / 255.255.255.0
2. CPU 51xS	
LAN with example IP/Mask	192.168.**202**.2 / 255.255.255.0
CPU with example IP/Mask	192.168.**202**.3 / 255.255.255.0
3. CPU 51xS	
LAN with example IP/Mask	192.168.**203**.2 / 255.255.255.0
CPU with example IP/Mask	192.168.**203**.3 / 255.255.255.0

Connect power supply

Connect power supply

After the CPU 51xS slot card is installed in the PC and the communication between PC and Ethernet component is established the DC 24V power supply may be attached.

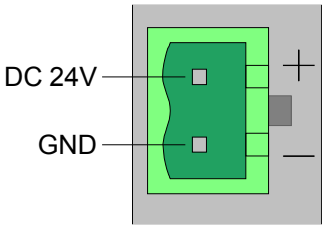
The CPU component of the CPU 51xS slot card can only be operated with an external or internal DC 24V power supply. The external supply enables the operation of the card outside of a PC res. independent from the PC operation. Please consider for operation that the slot card is connected to ground via its metal cover.

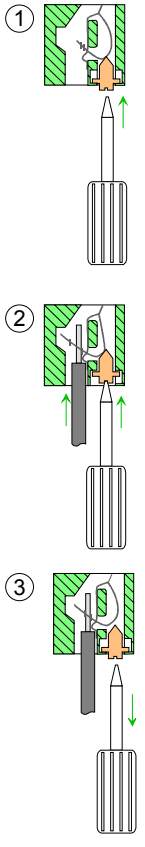
Note!

The CPU 51xS is to supply either externally or internally with DC 24V.
Simultaneous infeed should absolutely be avoided!

External power supply

For external power supply there is a plug on slot panel with the following pin assignment:

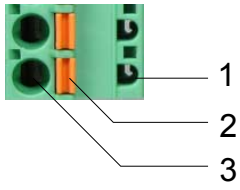




For the cabling of power supply of a CPU, a green plug with CageClamp technology is deployed.

The connection clamp is realized as plug that may be clipped off carefully if it is still wired.

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



[1] Test point for 2mm test tip
[2] Locking (orange) for screwdriver
[3] Round opening for wires

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.

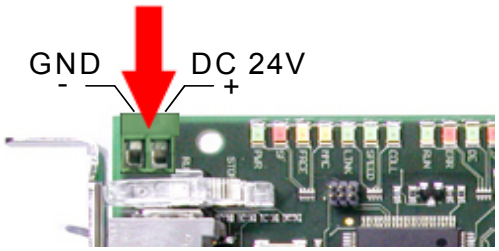
Internal power supply

For the internal power DC 24V supply there is a clamp on the top of the slot card.

Please consider when connecting a DC 24V power supply that the internal power supply hardware conditionally does not have an EMC filter for protection against disturbances (like e.g. EN 61000-4-4 [Burst], EN 61000-4-5 [Surge] or EN 61000-4-6 [conducted disturbance variable, inducted by HF fields]).

Here please use an accordingly filtered supply voltage here.

The clamp has the following pin assignment:



Initialization of the CPU component

Overview

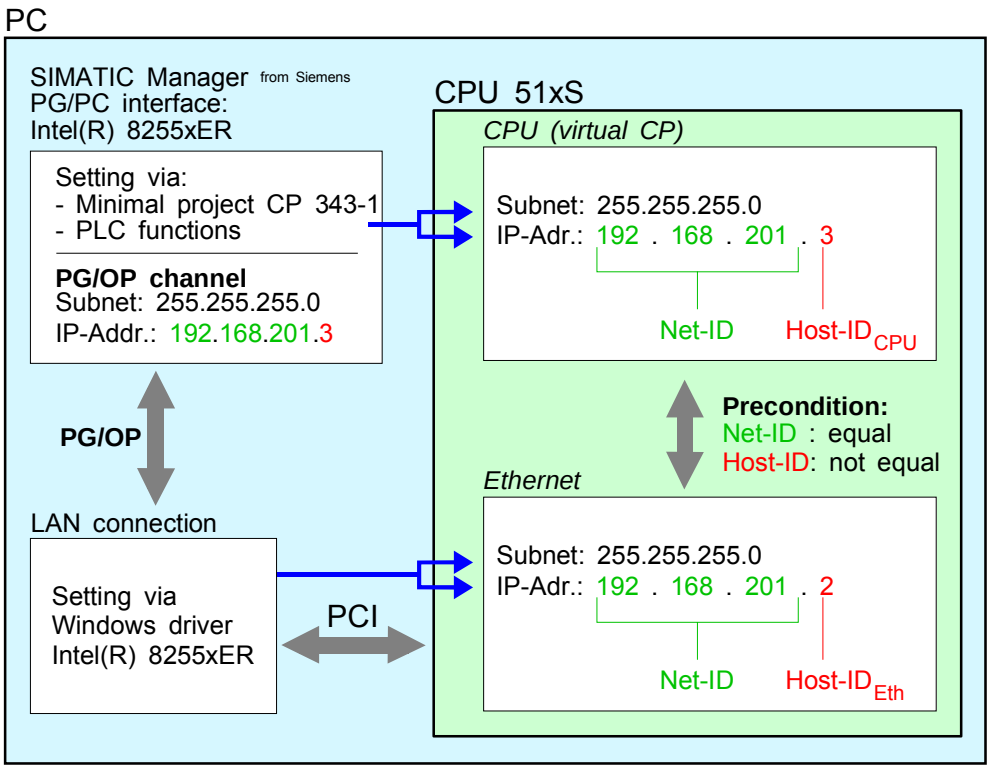
The CPU 51xS slot card consists of a CPU and an Ethernet component. For communication both components own an IP address, which may differ only in the Host-ID. For online access to the CPU component valid IP address parameters have to be assigned to this by means of the Siemens SIMATIC manager. This is called "initialization".

The initialization of the Ethernet component takes place by the properties of the network environment of your operating system, as described above.

Possibilities for Initialization

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

- PLC functions with *Edith Ethernet Node*
  (projecting tool and slot card in the same PC)
- Hardware project engineering with CP (Minimal project)



Requirements

For the hardware configuration the following software is necessary:

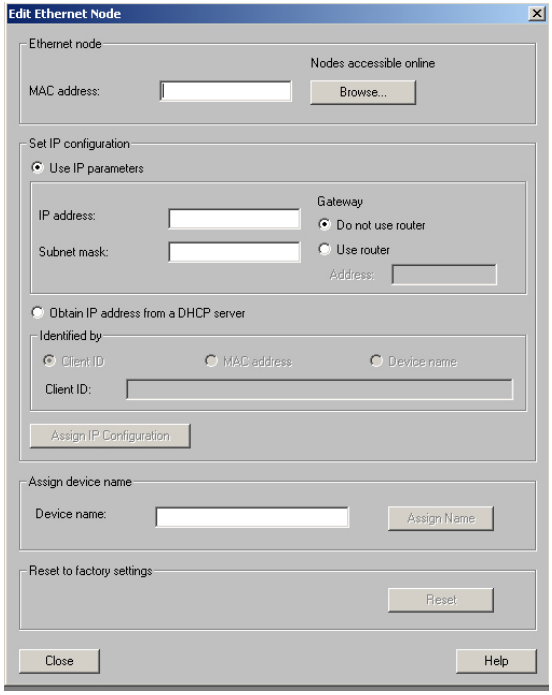
- Siemens SIMATIC manager V. 5.1 and SIMATIC NET or Siemens SIMATIC manager V. 5.2 and SP1

Initialization via PLC functions

The initialization by PLC function can only be established if Siemens SIMATIC manager and CPU 51xS slot card are at the same PC.

The initialization takes place after the following proceeding:

- Start the Siemens SIMATIC manager.
- Set via **Options** > *Set PG/PC Interface* the Access Path to "Intel(R) 8255xER".
- Open with **PLC** > *Edith Ethernet Node* the dialog window for "initialization" of a station.
- Use the [Browse] button to determine the CPU components via MAC address.



- 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 IP Configuration]

Direct after the assignment the CPU component may be reached by the Siemens SIMATIC manager by means of these IP address parameters.

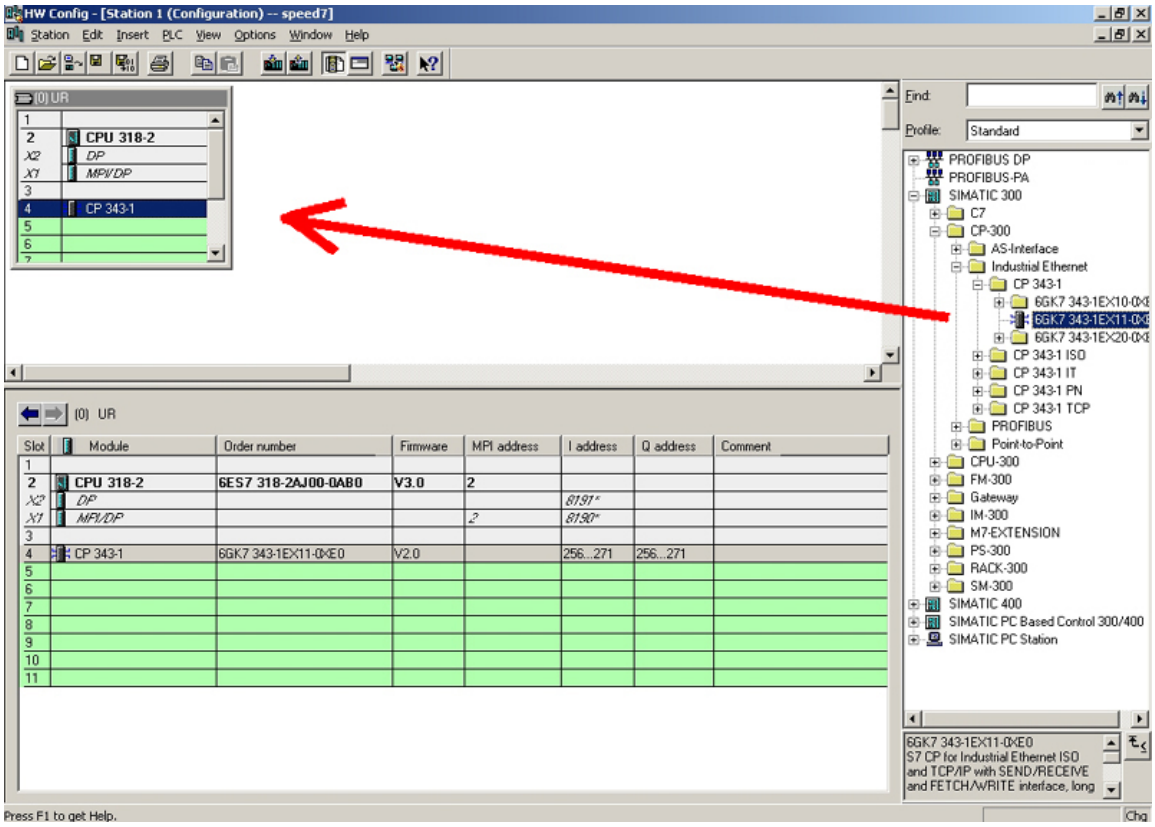
Initialization via minimal project

If the CPU 51xS slot card and the Siemens SIMATIC manager are not at the same PC, the CPU component may get its IP address parameters from a *minimal project* for CPU with CP.

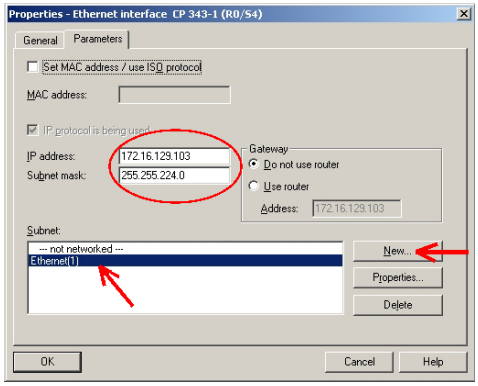
The Project may be transferred to the CPU 51xS slot card by MPI or by means of a MMC memory card.

The initialization by means of a minimal project takes place after the following proceeding:

- Start the Siemens SIMATIC manager 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.
- Project engineering Ethernet PG/OP channel as CP 343-1 (343-1EX11).
- Project engineering and networking Ethernet-CP 343 and DP master as CP 343-1 (343-1EX11) respectively CP 342-5 (342-5DA02 V5.0).



- Type the wanted IP address and subnet mask into the dialog window and connect the CP with "Ethernet".



- Save and compile your project.
- Transfer your project via MPI or MMC into your CPU.

Project transfer

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

- Transfer via MPI
- Transfer via storage card at deployment of a card reader

Transfer via MPI

- Change to the Siemens SIMATIC manager.
- Choose **Options** > *Set PG/PC interface*

  → A dialog window opens where you may configure the MPI interface you want to use.
- Choose the "PC Adapter (MPI)" from the selection list; where appropriate you have to add this first. Click on [Properties].
- Select the wanted COM port in the register "Local Port" and set the transfer rate 38400bit/s.
- Connect your PC via MPI with your CPU and transfer your project.

Transfer via MMC

As external storage medium a MMC (**Mem**ory **C**ard) is employed. The MMC is available at VIPA preformatted with the FAT16 PC file system.

- Create a new wld file via **File** > *Memory Card file* > *New* and use the mouse to drag the system data case into the window of the wld file.
- Copy the wld file with the help of a reading device to the MMC and rename the file to S7PROG.WLD.
- Insert the MMC in your CPU and execute an overall reset. This causes a transfer of the data from the MMC into the battery-buffered RAM of the CPU.

Note!

More information about transfer methods may be found in the chapter "Project transfer".

Internal access to PG/OP channel

Overview

Every CPU 51xS has an integrated Ethernet-PG/OP channel. Only in installed condition you have access via the PG/OP channel to your CPU. Here you may program, remote control or show the integrated web page. There are maximum 4 channels.

Access to PG/OP channel

If the conditions for communication are fulfilled, the slot card may be accesses by the PG/PC interface as "Intel(R) 8255xER" for project transfer and diagnostics.

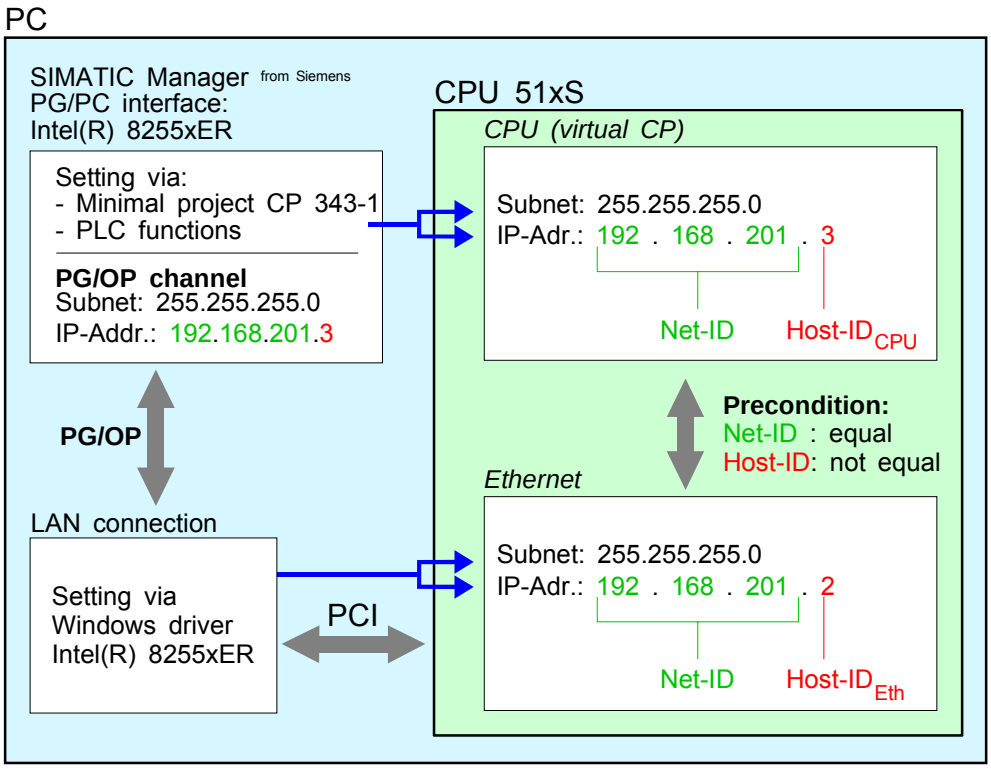
Requirements

- CPU 51xS slot card and Siemens SIMATIC manager are installed at the same PC.
- Ethernet and CPU component are assigned each to one IP address that may only differ in the Host-ID.
- Siemens SIMATIC manager starting with version V. 5.1 and SIMATIC NET respectively V. 5.2 and SP1 are installed for hardware configuration.

Approach

- Start the Siemens SIMATIC manager.
- Set **Options** > *PG/PC Interface*... to: "Intel(R) 8255xER"

The PG/OP channel may online be accessed, now.

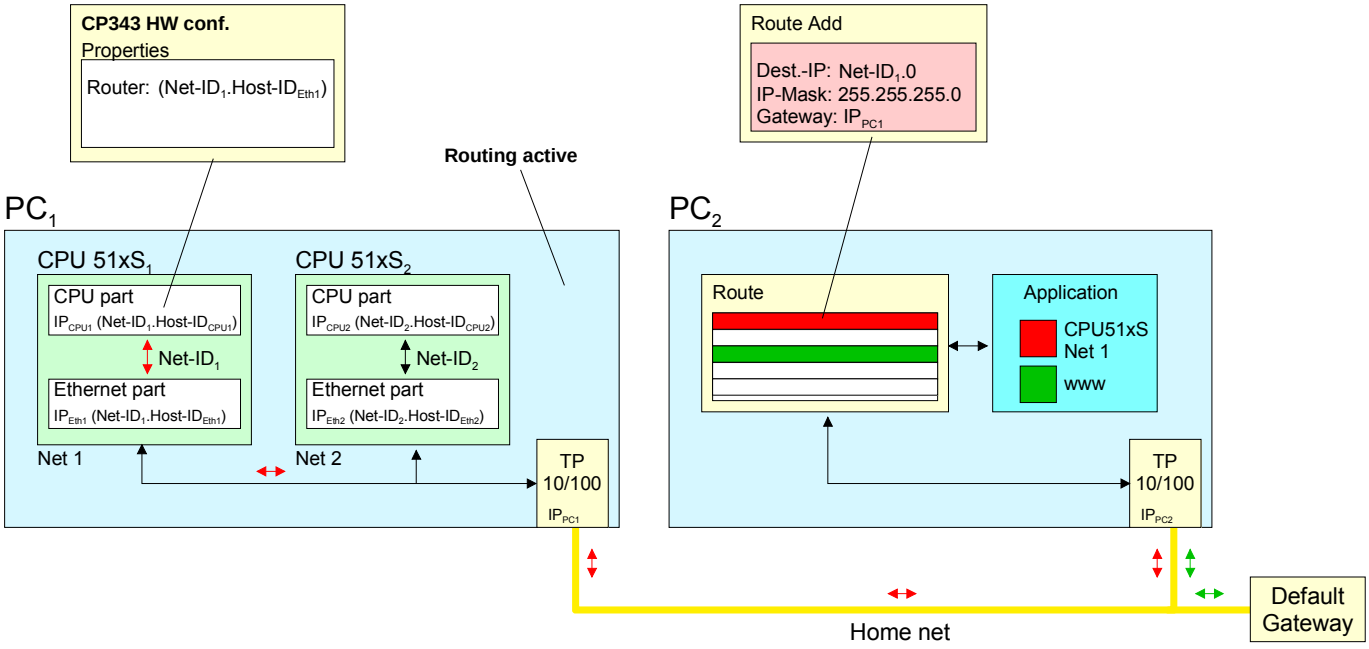


External access to PG/OP channel via routing

Overview

You may access the CPU 51xS from an external PC via Ethernet. The following preconditions must be fulfilled:

- The routing is activated on the PC with the CPU 51xS
- The route is entered at the CPU 51xS via a CP343 hardware configuration with the following parameters:
  - Destination router: IP address of the Ethernet component IP_{Eth} of the CPU 51xS
- The route is entered at the external PC with the following parameters:
  - Destination-IP: Net-ID of the Ethernet component of the CPU 51xS
  - IP-Mask: subnet mask of the CPU 51xS (default: 255.255.255.0)
  - Gateway: IP address of the PCs IP_{PC} with the CPU 51xS at the home network



Attention!

Only a trained system administrator should execute changes at the network neighborhood for this may cause conflicts with the company network!

The employment with Windows 9x res. Windows XP Home is not recommended and not described.

Activate routing

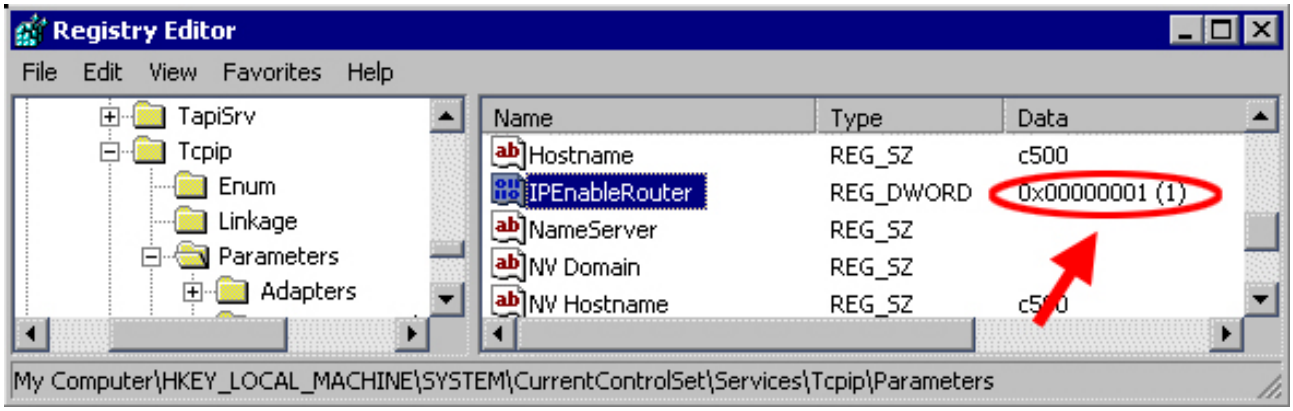
The following text describes the steps of configuration. More detailed information especially to the operating system specific routing is to find in the documentation of the operating system.

Activation at Windows NT4 / 2000$_{Server}$ */ 2003*$_{Server}$

The activation of the routing happens in the "Network neighborhood properties" at the properties of the TCP/IP protocol.

Activation at Windows XP$_{Professional}$ */ 2000*$_{Professional}$

For the activation, an entry into the registry file is required like illustrated below:



After a reboot, the routing is active.

Enter route

The entry of a route happens exclusively via the command console of the operating system by using the "route" command. The following parameters are required:

route ADD <Destination-IP> MASK <IP-Mask> <Gateway> METRIC <Metric> IF <IF>

with

ADD: Command for adding a route

Dest.-IP: IP address of the network (Net-ID) of the CPU 51xS

IP-Mask: Subnet mask of the net of the CPU 51xS

Gateway: IP address of the destination computer at the home network with the plugged CPU 51xS

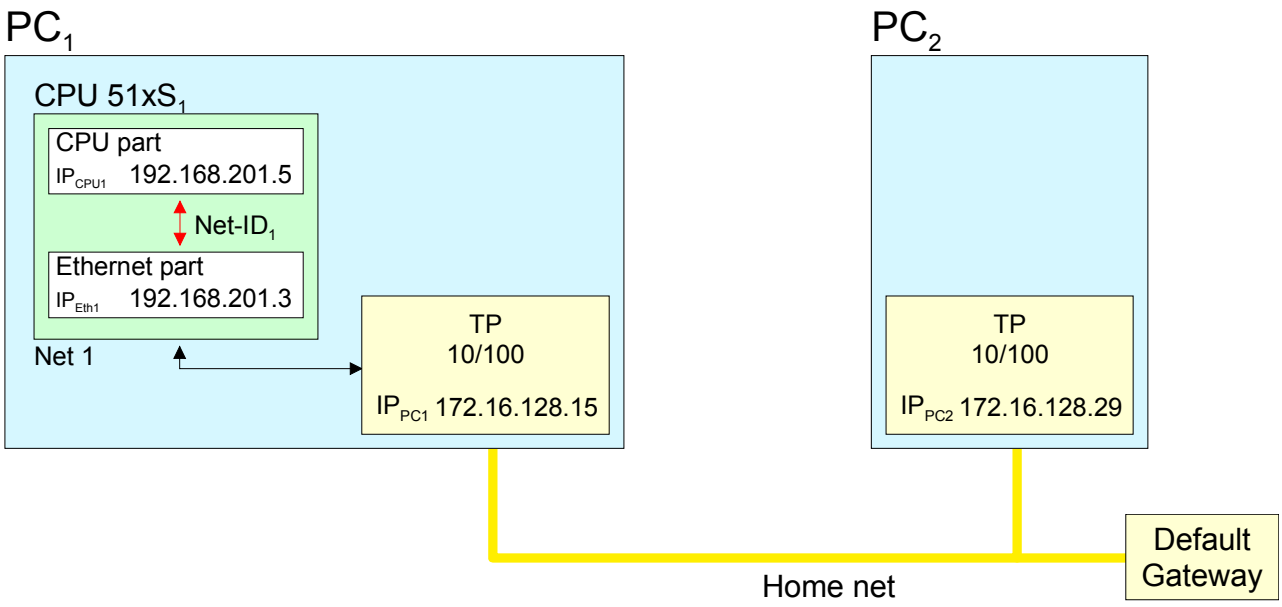
Metric: (optional) Price value for a destination

IF: (optional) Preset interface or best alternate interface

route PRINT lists all entered routes

route DELETE <Destination-IP> deletes the entry

Example: The following constellation is present and you want to access the CPU via PC:



- Activate the routing at PC1 like described above.
- Start the hardware configurator from Siemens and configure a system with CP 343.
- Enter the IP address 192.168.201.5 and the subnet mask 255.255.255.0 in "Properties Ethernet interface".
- Choose the function “Use router” at "Parameter", enter the IP address 192.168.201.3 of the Ethernet part of the CPU 51xS as "Gateway" and transfer your project.
- Start the command console at PC2 and enter the following statement:

```
route add 192.168.201.0 mask 255.255.255.0 172.16.128.15
```

Now you may access the CPU from PC2 via PC1. You may test the connection with the command `ping 192.168.201.5`.

Access to the integrated web page

Access to the web page

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

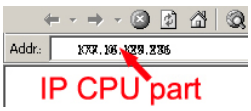
The current content of the web page is stored on MMC by means of the MMC-Cmd WEBPAGE. More information may be found at "MMC-Cmd - Auto commands".

Requirements

A PG/OP channel should be established between PC with Internet browser and CPU 51xS slot card. This may be tested by *Ping* to the IP address of the CPU component.

Web page

The access takes place via the IP address of the CPU component. The web page only serves for information output. The monitored values are not alterable.




```
CPU WITH ETHERNET-PG/OP
Slot 100
  VIPA 517-2AJ02 V3.5.1.4 Px000117.pkg,
  SERIALNUMBER 18519
  SUPPORTDATA :
  PRODUCT V3514, HARDWARE V0110, 5448D-V10 ,
  Hx000062.100 , Bx000227 V6514, Ax000086
  V1200, fx000007.wld V1140,
  FlashFileSystem : V102
  Memorysizes (Bytes): LoadMem : 8388608,
  WorkMemCode : 1048576, WorkMemData : 1048576
  OnBoardEthernet : MacAddress : 0020d5774857,
  IP-Address : 192.168.201.3, SubnetMask :
  255.255.255.0, Gateway : 192.168.201.3
  Cpu state : Stop
  FunctionRS485 X2/COM1: MPI
  FunctionRS485 X3/COM2: DPM-async
  Cycletime [microseconds]:
  min=426 cur=492 ave=522 max=912
  ArmLoad [percent] : cur=0, max=83
  PowerCycleHxRetries : 80, 0, 0, 0, 0

Slot 201
   VIPA 542-1DP00 V3.2.6 Px000119.pkg
   SUPPORTDATA : PRODUCT V3260,
   BB000554 V5260, AB000120 V4170,
   ModuleType CB2C0010
   Cycletime [microseconds] :
   min=65535000 cur=0 ave=0 max=0 cnt=0
BaudRate Read Mode1 , BaudRate Write Mode1
```

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

Ethernet PG/OP: Addresses

CPU status
RS485 function

CPU cycle time:
min= minimal
cur= current
max= maximal
Additional CPU components:

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

Standard bus

Project engineering

Overview

The project engineering of CPU and DP master takes place in the Siemens SIMATIC manager. The CPU 51xS slot card may online be accessed for parameterization by *PLC functions* via Ethernet respectively MPI/Profibus.

In addition, the project may be transferred by a MMC memory card to the PC slot card.

On delivery the CPU 51xS slot card has the MPI address 2.

Preconditions

For the hardware configuration of the CPU and the project engineering of the integrated Profibus DP master of the CPU, the following preconditions must be met:

- Siemens SIMATIC manager starting with version V. 5.1 and SIMATIC NET respectively V. 5.2 and SP1 are installed for hardware configuration.
- A communication connection to the slot card is established
- At usage of Profibus DP slaves of the Systems 100V, 200V and 300V from VIPA: GSD files are included in the hardware configurator.

For the project engineering of the CPU and the Profibus DP master, a thorough knowledge of the SIMATIC manager and the hardware configurator from Siemens are assumed!

Install hardware configurator from Siemens

The hardware configurator is part of the Siemens SIMATIC manager. It serves the project engineering. The modules that you may configure here are to find in the hardware catalog.

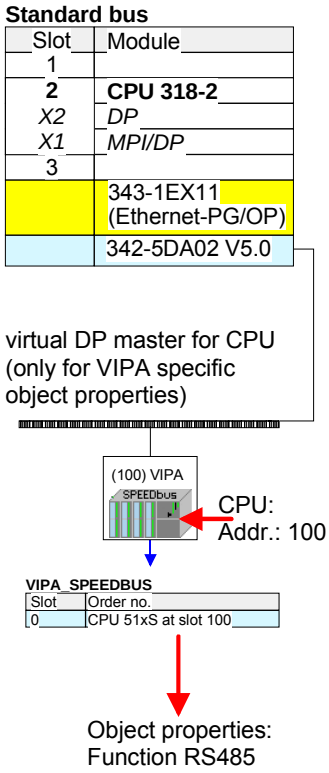
For the deployment of Profibus slaves of the Systems 100V, 200V and 300V from VIPA, the import of the modules to the hardware catalog via the GSD-files from VIPA is necessary.

Fast introduction

The project engineering of the CPU 51xS takes place at the Siemens hardware configurator and is divided into the following parts:

- Project engineering as CPU 318-2 (318-2AJ00-0AB00 V3.0)
- Project engineering Ethernet PG/OP channel as CP 343-1 (343-1EX11)
- Project engineering CPU 51xS as DP slaves in a virtual DP master CP 342-5 (342-5DA02 V5.0).

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



- Start the hardware configurator from Siemens.
- Configure the Siemens CPU 318-2 (6ES7 318-2AJ00-0AB0/V3.0). Configure the internal DP master of your CPU via the internal DP master of the CPU 318-2. Leave *MPI/DP* of the CPU 318-2 in MPI mode. The *Profibus* mode is not supported.
- For the internal Ethernet PG/OP channel you have to configure a Siemens CP 343-1 (343-1EX11).
- Configure as last module the Siemens DP master 342-5 (342-5DA02 V5.0). Link the DP master and switch it to DP master operating mode.
- To this master system you assign the CPU as "VIPA_SPEEDBUS" slave. Here the Profibus address corresponds to the slot no. Beginning with 100 for the CPU. Place at slot 0 of every slave the assigned module and alter the parameters if needed.

In the following these steps are more described.

Approach

The CPU 51xS has to be configured analog to a CPU318-2 from Siemens with Profibus DP master and a plugged Ethernet-CP CP343-1.

- Start the hardware configurator and create a new project System 300.
- Add a profile rail from the hardware catalog.
- You reach the CPU with Profibus master in the hardware catalog at: `Simatic300/CPU-300/CPU318-2DP/6ES7 318-2AJ00-0AB0`
- Insert the CPU 318-2DP (6ES7 318-2AJ00-0AB0/V3.0).
- Enter a Profibus address for the master (e.g. 2)
- Click on DP and set the operating mode “DP Master” in the *Object properties*. Confirm the settings by [OK].

Configuration of the CPU as CP 343-1

Configure representatively for the CPU component a CP 343-1.

This may be found in the hardware catalog at:

`Simatic300/CP-300/Industrial Ethernet.`

- Add the CP 343-1 (343-1EX11-0XE0) at slot 4.
- Click on the CP and enter the according IP address and subnet mask at *Object properties*. Enter the IP address parameters favored or assigned at initialization.

Please note that the IP address may differ exclusively in the host ID from the IP address of the Ethernet component.

The CPU component of the CPU 51xS slot card may be accessed by this address by means of the PLC functions.

Configure DP master system

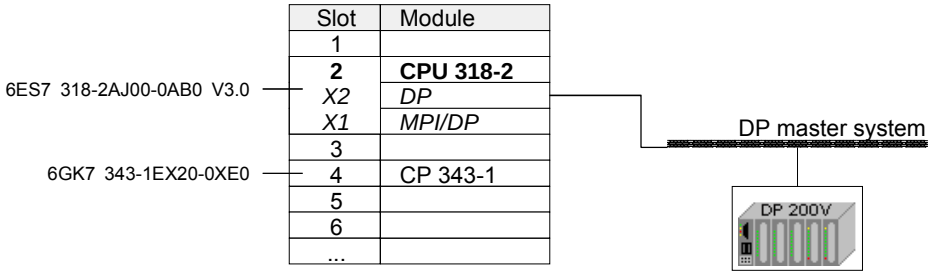
For the project engineering of the DP master system, you have to execute the following steps:

- Right click on *DP* and choose "Add Master System ".
- Create a new Profibus subnet with NEW. This subnet allows you to configure your Profibus slave modules.

Include DP slaves

- For the project engineering of Profibus DP slaves, choose the according Profibus DP slave from the *hardware catalog* and move it to the subnet of your master.
- Assign a valid Profibus address to the DP slave.
- Include the modules of the DP slave system in plugged sequence and assign valid addresses to the modules.
- Parameterize the modules if needed.

The following illustration shows the project engineering. Additionally the picture includes a VIPA Profibus DP slave as example:



Transfer project

The CPU 51xS slot card may online be accessed for parameterization by PLC functions via Ethernet respectively MPI/Profibus.

In addition, the project may be transferred by a MMC memory card to the PC slot card.

More information about transferring a project can be found at "Project transfer" below.

During start-up the Profibus project is forwarded to the Profibus master.

CPU parameterization

Overview

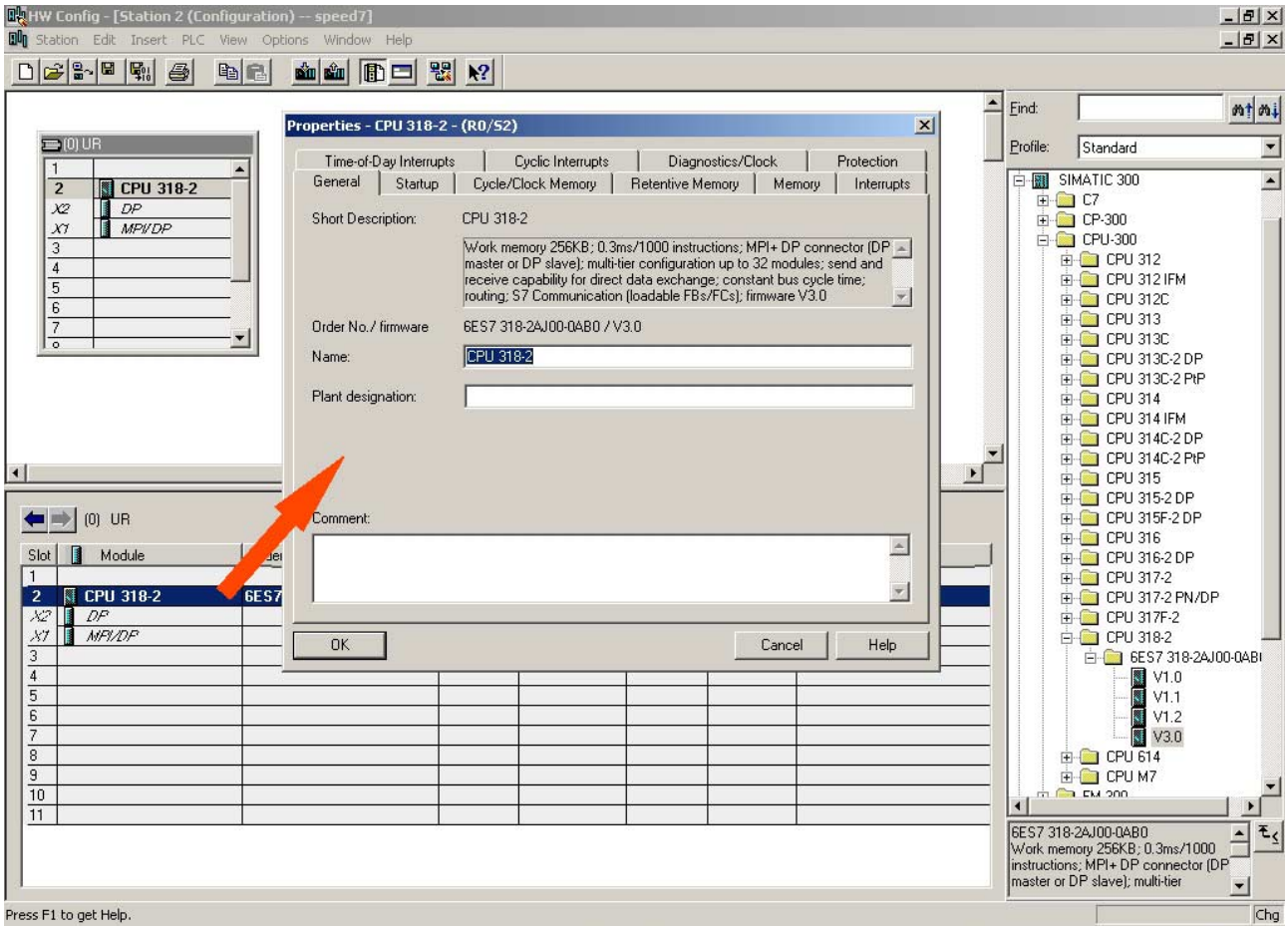
Except of the VIPA specific CPU parameters the CPU parameterization takes place in the parameter dialog of the CPU 318-2DP.

The VIPA specific CPU parameters like the RS485 interface behavior and the synchronization between CPU and DP master can 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 318-2DP the parameter window of the CPU can be archived. Using the registers you get access to all parameters of the CPU.



Setting IP address and subnet mask of the CPU

IP address and subnet mask are entered via the properties of the integrated CP 343-1.

Supported parameters

The CPU does not evaluate each parameter, which may be set at the hardware configuration.

The following parameters are supported by the CPU at this time:

General

Short description	Since every CPU 51xS from VIPA is configured as CPU 318-2AJ00 from Siemens, here the short description CPU 318-2 stands.
Order No. / Firmware	Order number and firmware are identical to the details in the "hardware catalog" window.
Name	The *Name* field provides the *short description* of the CPU. If you change the name the new name appears in the Siemens SIMATIC manager.
Plant designation	Here is the possibility to specify a plant designation for the CPU. This plant designation identifies parts of the plant according to their function. This has a hierarchical structure and confirms to IEC 1346-1.
Comment	In this field information about the module may be entered.

Startup

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

Cycle/Clock memory

Update OB1 process image cyclically

Activate the checkbox for cyclic updates of the OB1 process image.
Updating increases the cycle time.

Scan cycle monitoring time

Here the scan cycle monitoring time in milliseconds may be set. If the scan cycle time exceeds the scan cycle monitoring time, the CPU enters the STOP mode. Possible reasons for exceeding the time are:

- Communication processes
- a series of interrupt events
- an error in the CPU program

Minimum scan cycle time

The minimum scan cycle time specifies the time interval, in which the CPU program is called.
If the scan cycle time is less than the specified minimum scan cycle time, the CPU waits until the minimum scan cycle time is reached.

Scan cycle load from Communication

Using this parameter you can control the duration of communication processes, which always extend the scan cycle time so it does not exceed a specified length.
If there are no additional asynchronous events, the scan cycle time of OB1 is increased by following factor:

$$\frac{100}{100 - \text{cycle load from communication \%}}$$

If the cycle load from communication is set to 50%, the scan cycle time of OB 1 can be doubled. At the same time, the scan cycle time of OB 1 is still being influenced by asynchronous events (e.g. process interrupts) as well.

Size of the process image input/output area

Here the size of the process image max. 2048 for the input/output periphery may be fixed.

OB85 call up at I/O access error

The preset reaction of the CPU may be changed to an I/O access error that occurs during the update of the process image by the system.
Each CPU 51xS from VIPA is preset such that OB 85 is not called if an I/O access error occurs and no entry is made in the diagnostic buffer either.

Clock memory

Activate the check box if you want to use clock memory and enter the number of the memory byte.

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

Retentive Memory

Number of Memory Bytes from MB0	Enter the number of retentive memory bytes from memory byte 0 onwards.
Number of S7 Timers from T0	Enter the number of retentive *S7 timers* from T0 onwards.
Number of S7 Counters from C0	Enter the number of retentive *S7 counter* from C0 onwards.
Areas	Since with every VIPA CPU 51xS each data block is remanent stored, the settings at *Areas* are non-relevant and they are ignored.

Memory

Local data (priority classes)	In these fields you may define the number of local data (temporary data) for the priority classes 1 to 29.

Interrupts

Priority	Here the priorities may be specified according to which the hardware interrupt OBs are processed (hardware interrupt, time-delay interrupt, async. error interrupts). *Interrupts for DPV1* are not supported. With priority "0" the corresponding OB is deactivated. Please regard that this is not supported by each OB.

Time-of-day interrupts

Priority	Here the priorities may be specified according to which the time-of-day interrupt is processed. With priority "0" the corresponding OB is deactivated.
Active	Activate the check box of the time-of-day interrupt OBs if these are to be automatically started on complete restart.
Execution	Select how often the interrupts are to be triggered. Intervals ranging from every minute to yearly are available. The intervals apply to the settings made for *start date* and *time*.
Start date / time	Enter date and time of the first execution of the time-of-day interrupt.
Process image partition	This is not supported.

Cyclic interrupts

Priority	Here the priorities may be specified according to which the corresponding cyclic interrupt is processed. With priority "0" the corresponding interrupt is deactivated.
Execution	Enter the time intervals in ms, in which the watchdog interrupt OBs should be processed. The start time for the clock is when the operating mode switch is moved from STOP to RUN.
Phase offset	Enter the delay time in ms for current execution for the watch dog interrupt. This should be performed if several watchdog interrupts are enabled. Phase offset allows to distribute processing time for watchdog interrupts across the cycle.
Process image partition	Is not supported.

Diagnostics/Clock

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

Protection

Level of protection

Here 1 of 3 protection levels may be set to protect the CPU from unauthorized access.

Protection level 1 (default setting):

- No password adjustable, no restrictions

Protection level 2 with password:

- Authorized users: read and write access
- Unauthorized user: read access only

Protection level 3:

- Authorized users: read and write access
- Unauthorized user: no read and write access

Parameter for DP

The properties dialog of the Profibus part is opened via a double click to the sub module DP.

General

Short description

Here the short description "DP" for Profibus DP is specified.

Order no.

Nothing is shown here.

Name

Here "DP" is shown. If you change the name, the new name appears in the Siemens SIMATIC manager.

Interface

The Profibus address is shown here.

Properties

With this button the properties of the Profibus DP interface may be preset.

Comment

You can enter the purpose of the DP master in this box.

Addresses

Diagnostics

A diagnostics address for Profibus DP is to be preset here. In the case of an error the CPU is informed via this address.

Operating mode

Here the operating mode of the Profibus part may be preset. More may be found at chapter "Deployment CPU with Profibus".

Configuration

Within the operating mode "DP-Slave" you may configure your slave system. More may be found at chapter "Deployment CPU with Profibus".

Clock

These parameters are not supported.

Parameter for MPI/DP

The properties dialog of the MPI interface is opened via a double click to the sub module MPI/DP.

General

Short description	Here the short description "MPI/DP" for the MPI interface is specified.
Order no.	Nothing is shown here.
Name	At *Name* "MPI/DP" for the MPI interface is shown. If you change the name, the new name appears in the Siemens SIMATIC manager.
Type	Please regard only the type "MPI" is supported by the VIPA CPU 51xS.
Interface	Here the MPI address is shown.
Properties	With this button the properties of the MPI interface may be preset.
Comment	You can enter the purpose of the MPI interface in this box.

Addresses

Diagnostics	A diagnostics address for the MPI interface is to be preset here. In the case of an error the CPU is informed via this address.
Operating mode, Configuration, Clock	These parameters are not supported.

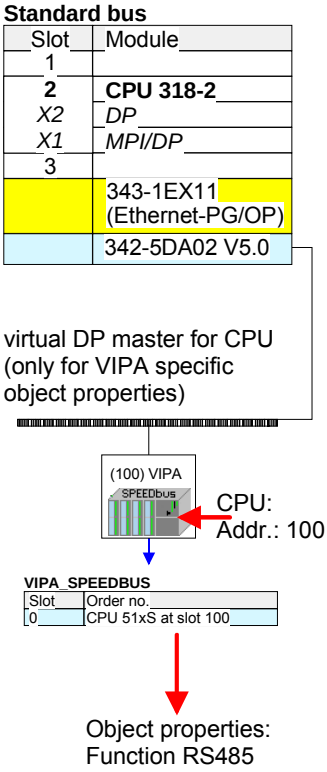
VIPA specific parameter via SPEED7 CPU

Here the following parameters may be accessed:

- Function RS485 (Synchronization DP master and CPU)
- Token Watch
- Number remanence flag
- Priority OB 28, OB 29, OB 33, OB 34
- Execution OB 33, OB 34
- Phase offset OB 33, OB 34

Via a hardware configuration you may configure the behavior of the synchronization between CPU and DP master using the parameter "Function RS485" of the *object properties*. Via a double-click on the inserted CPU 51xS at SPEED-Bus the parameter window of the SPEED7 CPU can be achieved.

Approach



Configure your SPEED7 system as shown above. After configuration the CPU 51xS must be located at slot 0 of the VIPA SPEED-Bus slave with address 100.

The *object properties* are opened via a double-click on the SPEED-Bus slave CPU 51xS.

With the parameter *Function RS485* the following adjustment possibilities are given.

Function RS485

Per default the RS485 interface is used 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 synchronization 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 PtP communication" in this manual.
Profibus-DP async	Profibus DP master operation asynchronous to CPU cycle The RS485 interface is preset at default to *Profibus-DP async*. Here CPU cycle and cycles of every VIPA Profibus DP master run independently.
Profibus-DP syncIn	The CPU is waiting for DP master input data.
Profibus-DP syncOut	The 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.

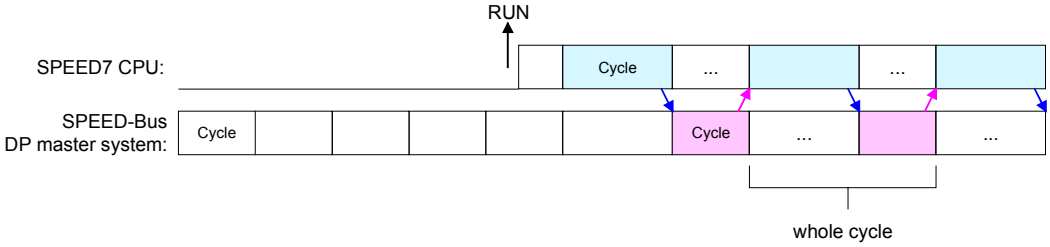
The different modes for the synchronization are in the following described.

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.

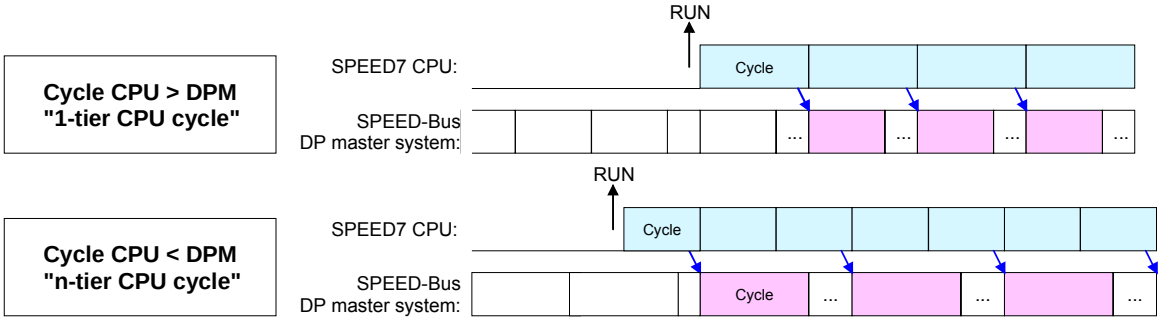
If necessary the time of the *Watchdog* of the bus parameters should be increased at this mode.



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.

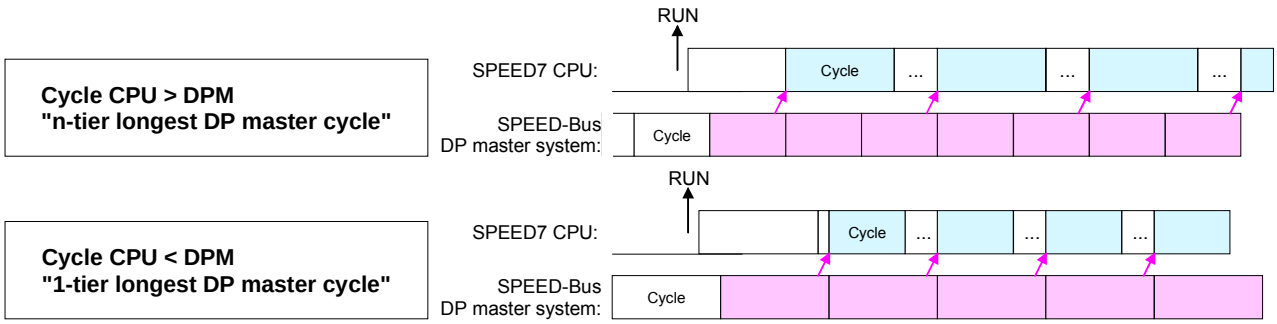
As soon as their cycle is passed they wait for the next synchronization impulse with output data of the CPU. So the response time of your system can be improved because output data were directly transmitted to the DP master system. If necessary the time of the *Watchdog* of the bus parameters should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



Token Watch

This is a VIPA internal parameter. Nothing should be changed here.

Default: On

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 318-2 is used. Otherwise the adjusted value (1 ... 16384) 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 is deactivated), 2,3,4,9,12,16,17,24

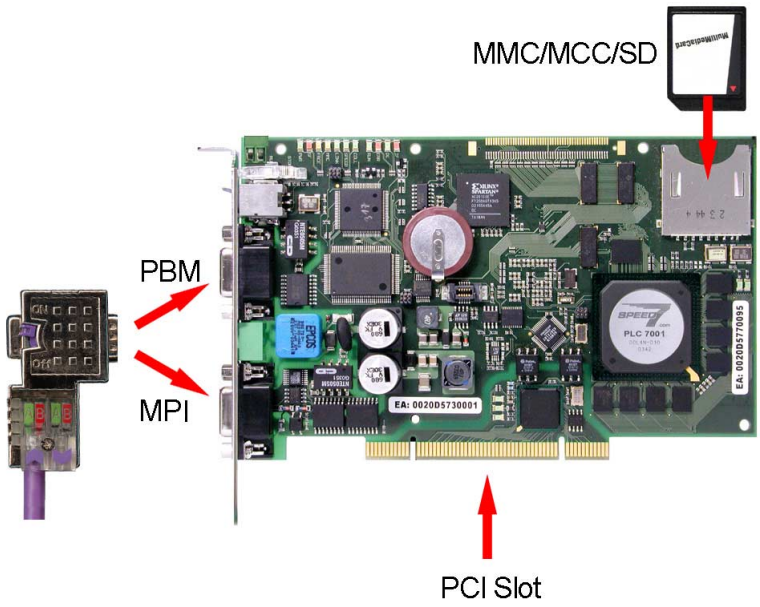
Default: 24

Project transfer

Overview

The following options are available to transfer a project into the CPU 51x slot card:

- internal via PCI slot (Ethernet connection)
- external via PC network card (routing necessary)
- external via RS485
- external via Profibus (not for first project)
- external via MMC storage card



Internal project transfer via PCI slot by an Ethernet connection

For this transfer method CPU 51xS slot card and Siemens SIMATIC manager are installed at the same PC.

As soon as the Ethernet and CPU component have been assigned to valid IP address parameters the CPU 51xS may internal be accessed with the IP address of its CPU component. Adjust here as PG/PC interface "Intel(R) 8255xER".

For the communication functions of the CPU component a hardware configuration in which beside the CPU 318-DP a CP 343-1 is configured. Set the desired IP address parameters here. Please note that the IP address may differ exclusively in the host ID from the IP address of the Ethernet component.

Within the PLC function the IP address parameters of the CP 343 may be used as target parameters.

If the target station indicated by the IP address parameters is not found during transmission, the original IP parameters for the CPU component may be entered in a hint dialog.

With confirmation of your input your project will be transferred to the original IP address. After the restart of the CPU the new IP parameters of the project are active.

External project transfer via PC network card

For this transfer method it is presupposed that CPU 51xS slot card and a network card are in the target PC and the network card is connected with the projecting PC by Ethernet.

So that the CPU 51xS slot card may be accessed by the projecting PC via the network card a "routing" is necessary. For this details may be found at "External access to PG/OP channel via routing" above.

Proceed now as described at internal project transfer. Configure a CPU 318-2DP and a CP 343-1. Set the desired IP address parameters here. Choose at "parameters" "use router" and enter as "Gateway" the IP address of the Ethernet component of the CPU 51xS slot card.

Within the PLC function the IP address parameters of the CP 343 may be used as target parameters.

If the target station indicated by the IP address parameters is not found during transmission, the original IP parameters for the CPU component may be entered in a hint dialog.

With confirmation of your input your project will be transferred to the original IP address. After the restart of the CPU the new IP parameters of the project are active.

Transfer via RS485

For transfer via RS485 there are the following 2 interfaces:

- MPI interface supports maximally 32 PG/OP channels
- PB-DP/PtP interface supports maximally 31 PG/OP channels (exclusive at Profibus DP master operation)

MPI programming cable

The MPI programming cables are available at VIPA in different variants. The deployment of the cables is identical. The cables provide a bus enabled RS485 plug for 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 an already plugged plug on the RS485 jack. Every bus participant identifies itself at the bus with an unique address, in the course of the address 0 is reserved for programming devices.

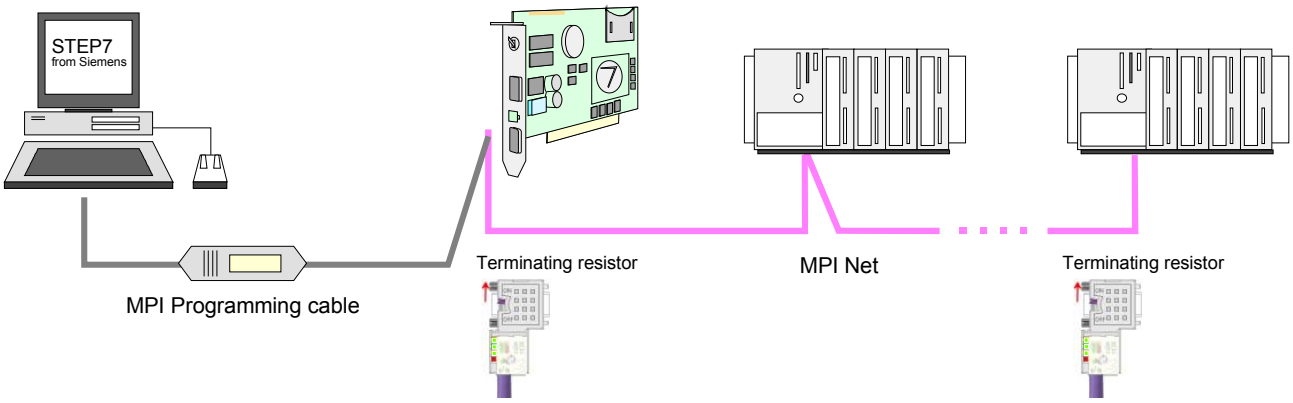
Net structure

The structure of a MPI net is in the principal identical with the structure of a 1.5Mbit/s Profibus net. This means the same rules are valid and you use the same components for the build-up. The single participants are connected with each other via bus interface plugs and Profibus cables. Per default the MPI net runs with 187.5kbit/s. 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 interface

- Connect your PC to the MPI 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 PC and the transfer rate 38400bit/s 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 interface

- Connect your PC to the DP-PB/PtP 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 38400bit/s 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.

Note!

An according project with an activated Profibus must have been loaded to the CPU before (not suited for first start-up), otherwise the CPU may not be reached after a overall reset.

Transfer via MMC

The MMC (**Me**m**o**ry **C**ard) serves as external transfer and storage medium for programs and firmware. It has the PC compatible FAT16 file system.

There may be stored several projects and sub-directories on a MMC storage module. Please regard that your current project respectively the file with the reserved file name is stored in the root directory.

With an overall reset, PowerON or CPU-STOP the MMC is automatically read. By presetting a reserved file name the functionality of the CPU may be influenced.

Reserved file names

File name	Description
S7PROG.WLD	Project file - is read after overall reset respectively may be written by CPU by an order.
AUTOLOAD.WLD	Project file - is read after PowerON.
PROTECT.WLD	Protected project file (see "Extended know-how protection").
VIPA_CMD.MMC	Command file - file is once executed during CPU-STOP by plugging a MMC or after PowerON (see "MMC-Cmd - Auto command").
*.pkg	Firmware file - is recognized after PowerON and may be installed by means of an update request (see "Firmware update").

Transfer MMC → CPU

The transfer of the application program from the MMC into the CPU takes place depending on the file name after overall reset or PowerON. The blinking of the LED "MCC" of the CPU marks the active transfer.
A transfer from CPU to MMC only happens if the size of the user memory exceeds the size of the user program. Else a memory expansion via MCC is necessary.

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 "MMC"-LED of the CPU is blinking. When the LED expires the write process is finished.

Transfer 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.

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 successful

Operating modes

Overview

The CPU has 4 operating modes:

- operating mode STOP
- operating mode START-UP
- operating mode RUN
- operating mode FLAG

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

- Processing of the application program has stopped.
- If the program was being processed before, the values of counters, timers, flags and the contents of the process image are retained during the transition to the STOP mode.
- Outputs are inhibited, i.e. all digital outputs are disabled.
- RN-LED off
- ST-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.
- RN-LED blinks
- ST-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. At 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.
- RN-LED on
- ST-LED off

Operating mode FLAG

The CPU 51xS provides the opportunity to define up to 4 break points (flags) for program diagnosis purposes. The flags are set and deleted via your programming neighborhood. As soon as a break point is reached, you may execute your application line by line and may activate in- and outputs.

Preconditions

For the usage of break points, the following preconditions must be fulfilled:

- The single step test mode is only available for STL, if needed, change the view via **View** > *STL* to STL.
- The block has to be opened online and must not be protected.
- The opened block must not have been altered in the editor.

Approach with break points

- Activate the break point bar via **View** > *Break point bar*.
- Put the cursor to the statement line where a break point is to be set.
- Set the break point with **Test** > *Set break point*. The statement line is marked with a ring.
- To activate the break point, choose **Test** > *Break point active*. The ring changes to a circle.
- Switch the CPU to RUN. When the user application reaches the break point, the CPU switches into the state FLAG, the break point is marked with an arrow and the register contents are shown.
- Now you may execute your application code step by step via **Test** > *Next command* or execute the application until the next break point with **Test** > *Continue*.
- **Test** > *Delete (all) break points* deletes (all) break points.

Behavior in operating mode FLAG

- LED RN blinks and LED ST is on.
- The code execution has been stopped. All run levels are not executed.
- All timer are frozen.
- The real-time clock is still active.
- The outputs are shut down, but may be released for test purposes.
- Passive CP communication is possible.

Note!

The usage of break points is possible at any time. A switch to the operating mode “Test operation” is not required.

If you set more than 3 break points, the single step operation is not longer available.

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 PIQ → Read PII → OB1.
	central analog outputs	The behavior of the outputs at restart can be preset.
	decentral inputs	The inputs are read constantly from the slave and the recent values are put at disposal.
RUN	general	The program execution happens cyclically and can therefore be foreseen: Read PII → OB1 → Write PIQ.

PII: = Process image inputs

PIQ: = Process image outputs

Overall reset

Overview

During the overall reset the entire user memory (RAM) and the retentive memory area is erased.

Data located on the storage medium (MMC, MCC) is not affected.

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 it.

Overall reset by means of the operating mode switch

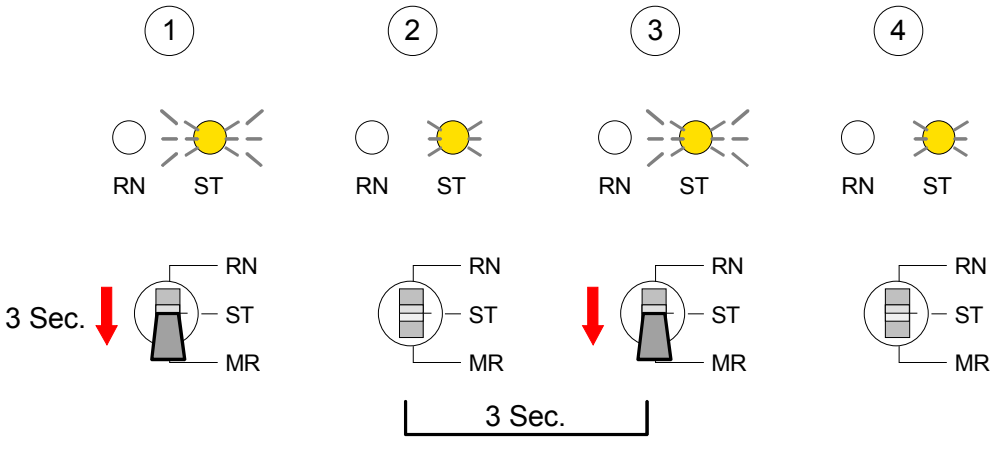
Condition

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

Overall reset

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

The following figure illustrates the above procedure:



Overall reset via PLC-Tool

At deployment of the operating software PLC-Tool you may initialize the Overall reset via the button [M-RES].

The button is available as soon as your CPU is in STOP.

Overall reset via SIMATIC manager from Siemens

Conditions

Your CPU has to be in STOP.

Via the menu command **PLC** > *Operating Mode* you switch your CPU in STOP.

Overall reset

Via the menu command **PLC** > *Clear/Reset* you request the overall reset.

In the dialog window you may switch your CPU to STOP if you didn't that yet and start the overall reset.

During the overall reset procedure the ST-LED is blinking.

When the ST-LED changes to permanently on, the overall reset has been finished.

Automatic reload

After the overall reset the CPU attempts to reload the parameters and the program from the memory card. → The MMC-LED at the board blinks.

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

Firmware update

Overview

The CPU offer you the opportunity to execute a firmware update for the CPU via MMC.

For this an accordingly prepared MMC must be in the CPU during the start-up.

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

As soon as with start-up 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 get the new firmware.

Latest Firmware at www.vipa.de

The latest 2 firmware versions may be found in the service area at www.vipa.de.

For example the following files are necessary for the firmware update of the CPU 517-2AJ02 and its components (Profibus) with hardware release 1:

- 517-2AJ02, Hardware release 1: Px000117.zip
- Profibus DP master (integrated): Px000119.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 firmware version of the SPEED7 system via Web page

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

More detailed information can be found at "Access to integrated web page".

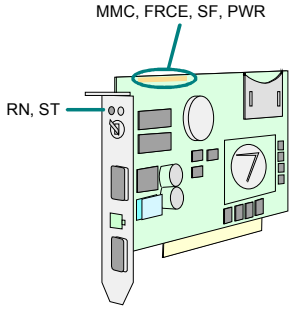
Load firmware and transfer it to storage media

- Go to www.vipa.de.
- Navigate to "Firmware".
- Click at "System 500S".
- Choose the according modules 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.

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 to CPU



1. Get the RUN-STOP lever of your CPU in position STOP. Turn off the voltage supply. In installed condition depending upon operating system now you get a message that the connected partner is no longer present. This message can be ignored. 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.
7. If the LEDs do not flash, the firmware update is ready.
8. Now a *factory reset* should be executed (see next page). After that the CPU is ready for duty.

Factory reset

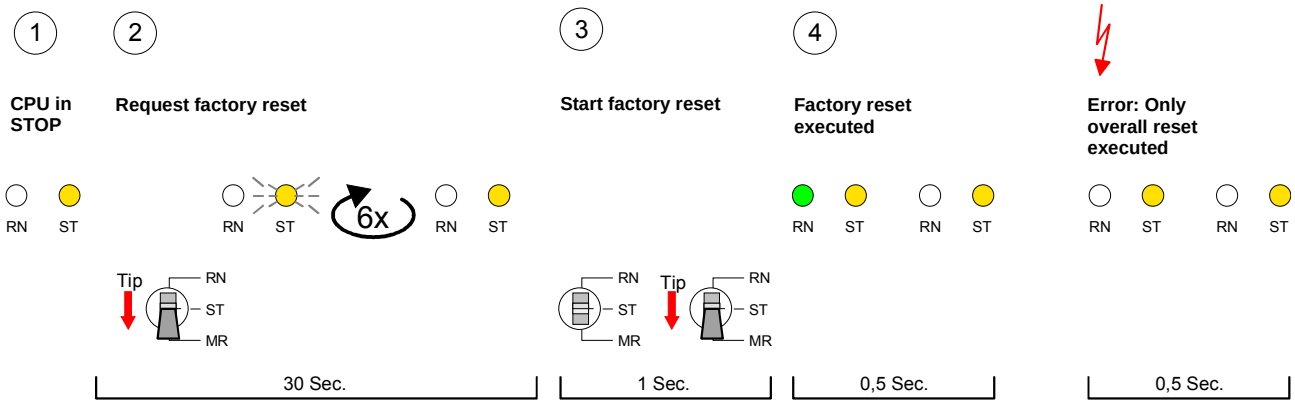
Proceeding

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 MPI address is reset to the default address 2 and the IP address of the Ethernet PG/OP channel is reset to 0.0.0.0!

A factory reset may also be executed by the 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 MR for 30s. Here the ST-LED flashes. After a few seconds the ST-LED changes to static light. Now the ST-LED changes between static light and flashing. Starting here count the static light states.
3. After the 6. static light release the operating mode switch and tip it shortly downwards to MR.
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. The end of factory reset is showing by static light of the LEDs STOP, SF, FRCE and MMC. 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 expansion with MCC

Overview

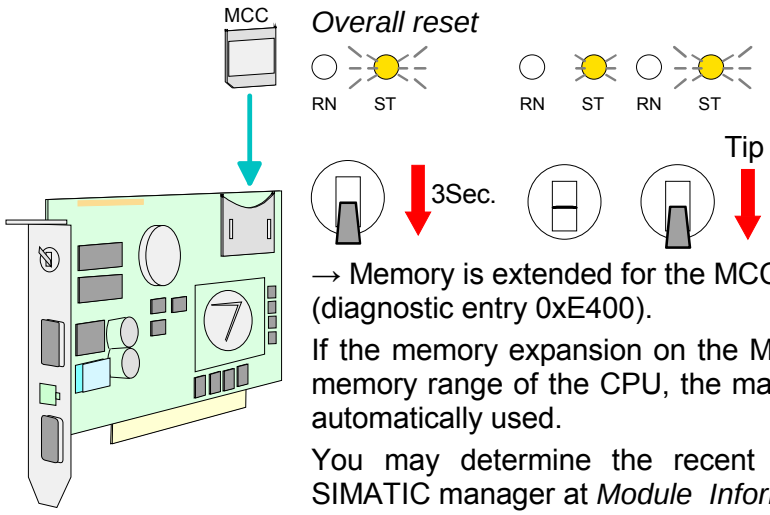
You have the option to expand the work memory of your CPU.

For this, a MCC (**M**emory **C**onfiguration **C**ard) is available at 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.



→ Memory is extended for the MCC memory configuration (diagnostic entry 0xE400).

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 72h. The MCC can not be exchanged with a MCC of the same memory configuration.

Behavior

When the MCC memory configuration has been set 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 72h 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 third 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.

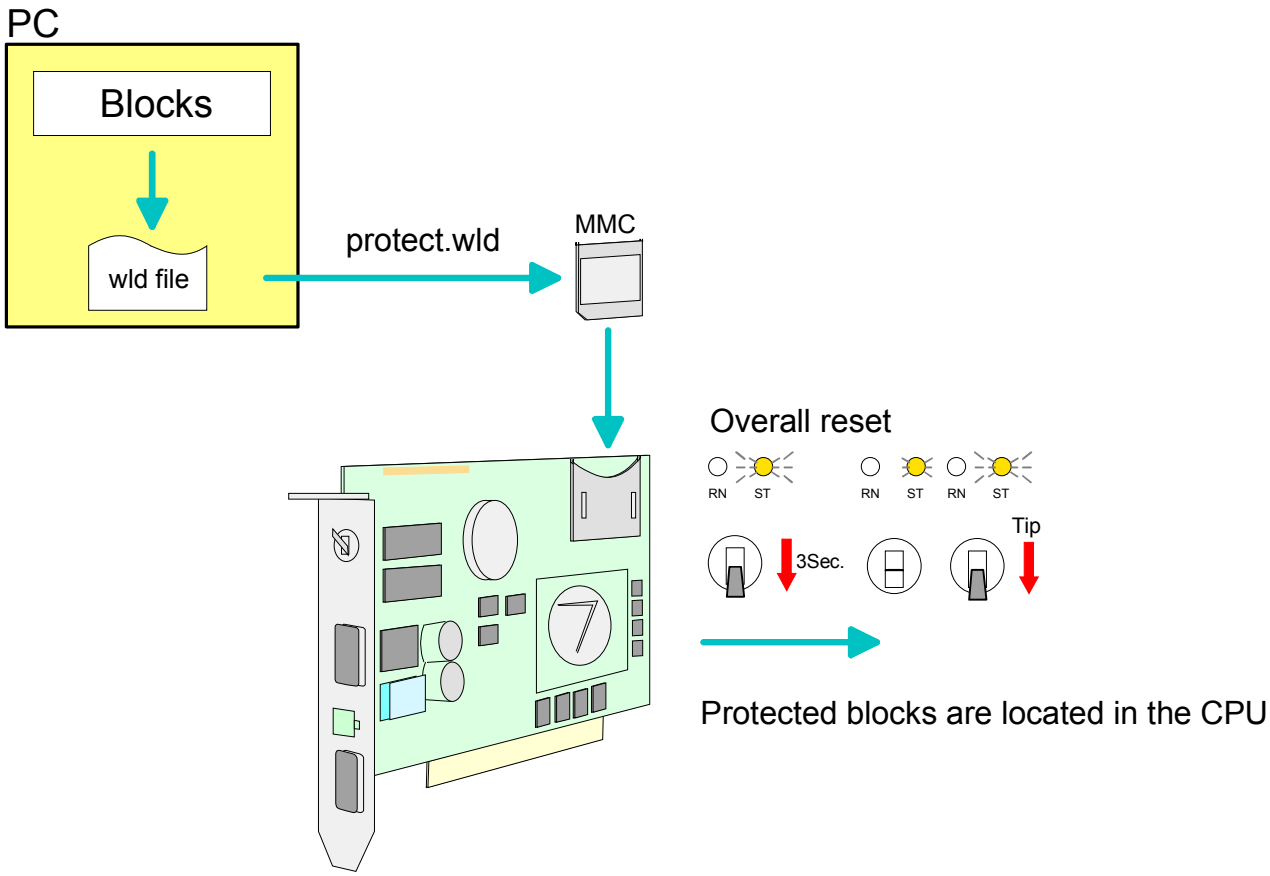
Extended protection

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 third persons.



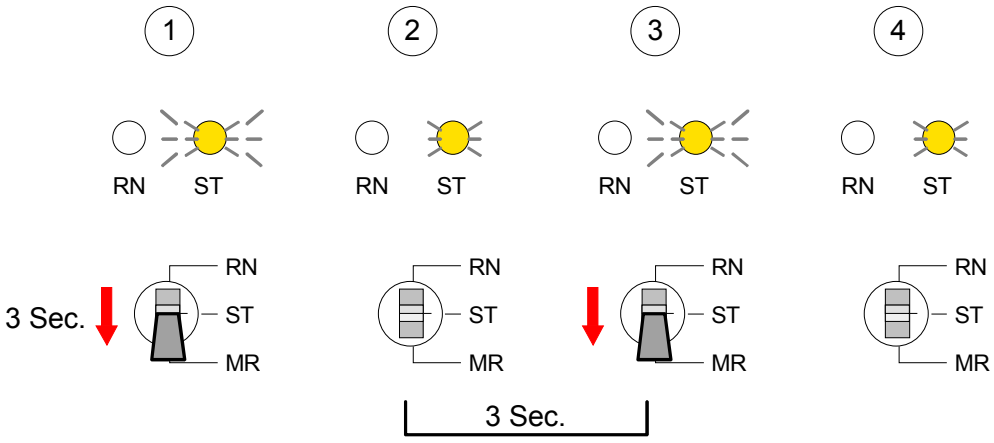
protect blocks with protect.wld

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 third persons.

Protection behavior

Protected blocks are overwritten by a new protect.wld.

Using a PG third 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

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 PowerON.

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 1. 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 [password]	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". If your project is password protected you have to add to SAVE_PROJECT your password as parameter. Example: Password = "vipa": SAVE_PROJECT vipa	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 separated by a comma. Enter the IP address if there is no gateway used.	0xE80E
CMD_END	In the last line *CMD_END* is to be located.	0xE802

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

Example 1

CMD_START	Marks the start of the command sequence (0xE801)
LOAD_PROJECT proj.wld	Execute an overall reset and load "proj.wld" (0xE805)
WAIT1SECOND	Wait ca. 1s (0xE803)
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_NETWORK 172.16.129.210,255.255.224.0,172.16.129.210	IP parameter (0xE80E)
WAIT1SECOND	Wait ca. 1s (0xE803)
WAIT1SECOND	Wait ca. 1s (0xE803)
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.

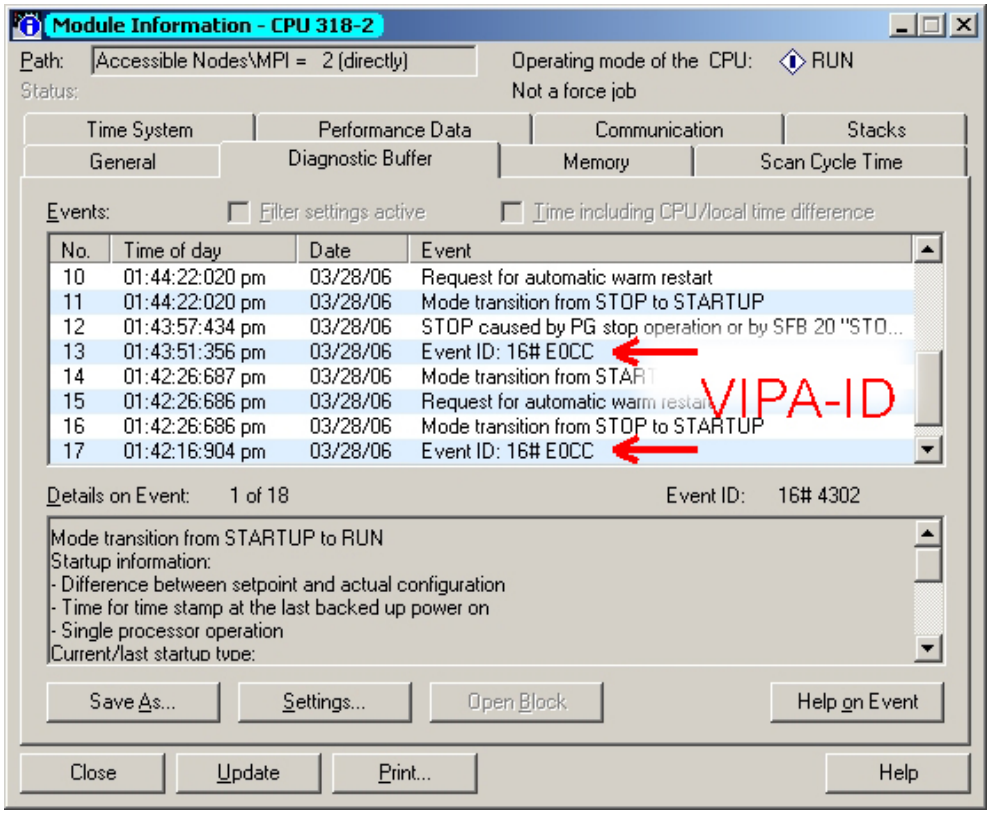
The current content of the diagnostics buffer is stored on MMC by means of the MMC-Cmd DIAGBUF. More information may be found at "MMC-Cmd - Auto commands".

Note!

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

Monitoring the diagnostic entries

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



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

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

Overview of the Event-IDs

Event-ID	Description
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 standard bus digital modules
0xE014	Error at Check_Sys
0xE015	Error at access to the master Zinfo2: Slot of the master (32=page frame master)
0xE016	Maximum block size at master transfer exceeded Zinfo1: 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 / 9bit)
0xE01B	Error - maximum number of plug-in modules exceeded
0xE030	Error of the standard bus
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
0xE0CD	Error at DPV1 job management
0xE0CE	Error: Timeout at sending of the i-slave diagnostics
0xE100	MMC access error
0xE101	MMC error file system
0xE102	MMC error FAT
0xE104	MMC error at saving
0xE200	MMC writing finished (Copy Ram2Rom)

continued ...

... continue

Event-ID	Description
0xE210	MMC reading finished (reload after overall reset)
0xE21F	MMC reading: error at reload (after overall reset), read error, out of memory
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!
0xEA02	SBUS: Internal error (internal plugged sub module not recognized) Zinfo1: Internal slot
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

continued ...

... continue

Event-ID	Description
0xEA10	SBUS: Input address outside input area Zinfo1: I/O address Zinfo2: Slot Zinfo3: Data width
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!
0xEA20	Error - RS485 interface is not set to Profibus DP master but there is a Profibus DP master configured.
0xEA21	Error - Project engineering RS485 interface X2/X3: Profibus DP master is configured but missing Zinfo2: Interface x
0xEA22	Error - RS485 interface X2 - value is out of range Zinfo: Configured value X2
0xEA23	Error - RS485 interface X3 - value is out of range Zinfo: Configured value X3
0xEA24	Error - Project engineering RS485 interface X2/X3: Interface/Protocol is missing, the default settings are used. Zinfo2: Configured value X2 Zinfo2: Configured value X3
0xEA30	Internal error - Please contact the VIPA-Hotline!
0xEA40	Internal error - Please contact the VIPA-Hotline!
0xEA41	Internal error - Please contact the VIPA-Hotline!
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 the states may be interrupted by means of jump commands or by timer and process-related alarms. At the breakpoint the CPU stops collecting data for the status display and instead of the required data it only provides the PG with data containing the value 0.

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

- the result of the logical operation 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:

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

The process image of binary and digital operands is modified independently of the operating mode of the CPU 51xS.

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

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

Chapter 4 Deployment CPU with Profibus

Overview

Content of this chapter is the deployment of the CPU 517S/DPM with Profibus. After a short overview the project engineering and parameterization of a CPU 517S/DPM 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.

Content

Topic	Page
Chapter 4 Deployment CPU with Profibus	**4-1**
Overview	4-2
Project engineering CPU with integrated Profibus master	4-3
Deployment as Profibus DP slave	4-5
Project transfer	4-7
Profibus installation guidelines	4-8
Commissioning and Start-up behavior	4-11

Overview

Profibus DP

Profibus is an international standard applicable to an open and serial field bus for building, manufacturing and process automation that can be used to create a low (sensor-/actuator level) or medium (process level) performance network of programmable logic controllers.
Profibus comprises an assortment of compatible versions. The following details refer to Profibus DP.
Profibus DP is a special protocol intended mainly for automation tasks in a manufacturing environment. DP is very fast, offers Plug'n'Play facilities and provides a cost-effective alternative to parallel cabling between PLC and remote I/O. Profibus-DP was designed for high-speed data communication on the sensor-actuator level. The data transfer referred to as "Data Exchange" is cyclical. During one bus cycle, the master reads input values from the slaves and writes output information to the 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. 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 124 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.

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

Overview

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 51xS/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 300V from VIPA: GSD Files are included into the hardware configurator.
- There is a transfer possibility between configuration tool and CPU 51xS.

Note!

For the project engineering of the CPU and the Profibus DP master a thorough knowledge of the Siemens SIMATIC manager is required!

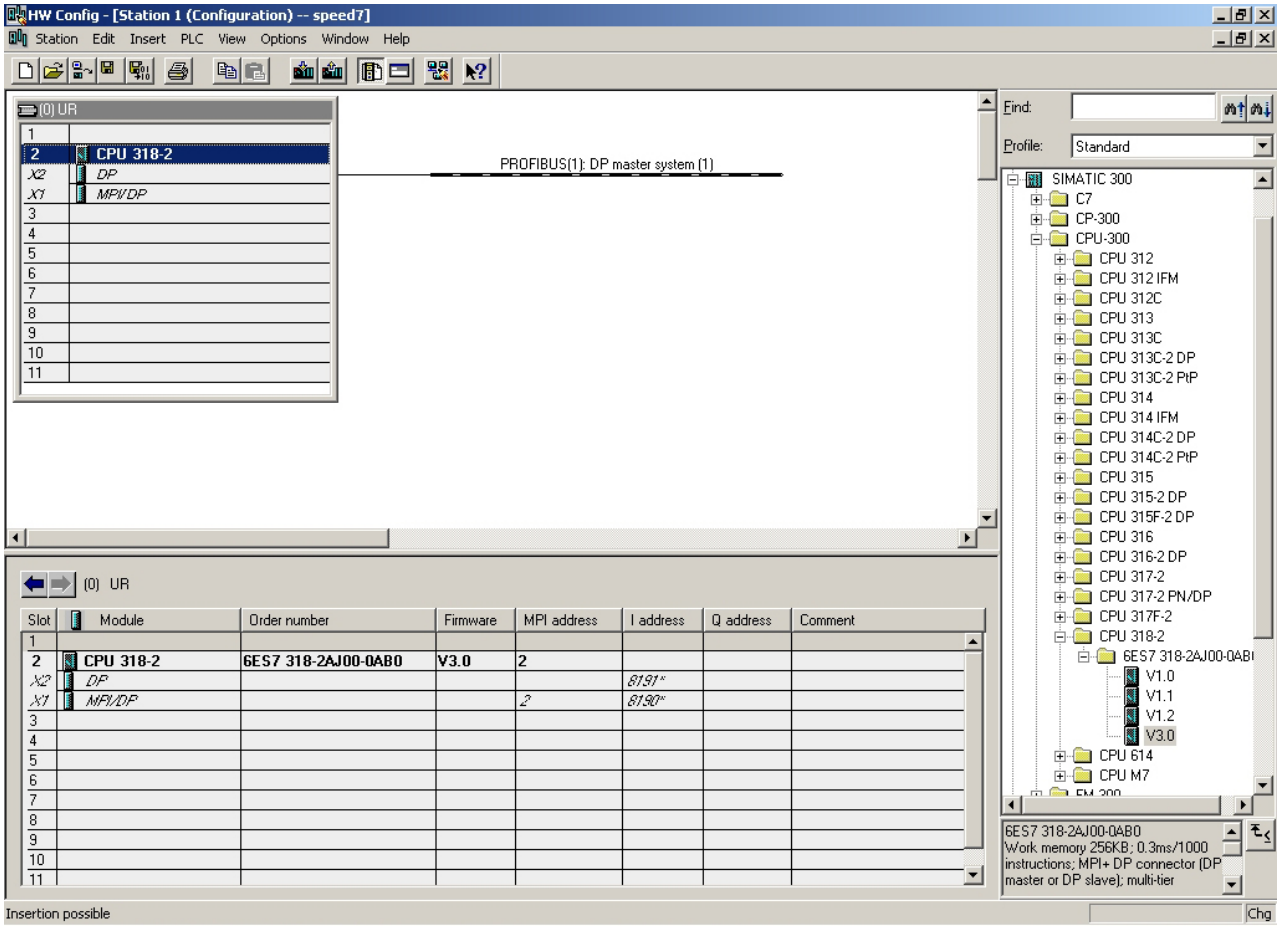
Install Siemens Hardware configurator

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.

Slave operation possible

You may deploy your Profibus part from the 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:

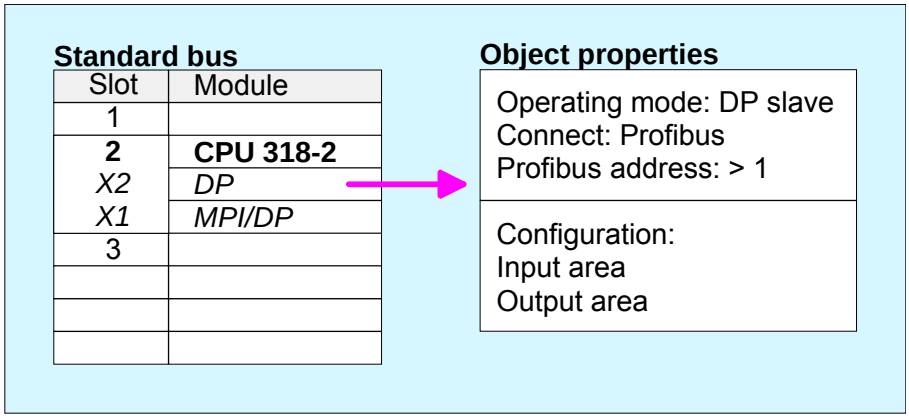
- Start the Siemens SIMATIC manager and configure a CPU 318-2DP with the operating mode *DP slave*.
- Connect to Profibus and configure the in-/output area for the slave section.
- Save and compile your project.
- Configure another station as CPU 318-2DP with operating mode *DP master.*
- Connect to *Profibus* and configure the in-/output area 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)
- 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

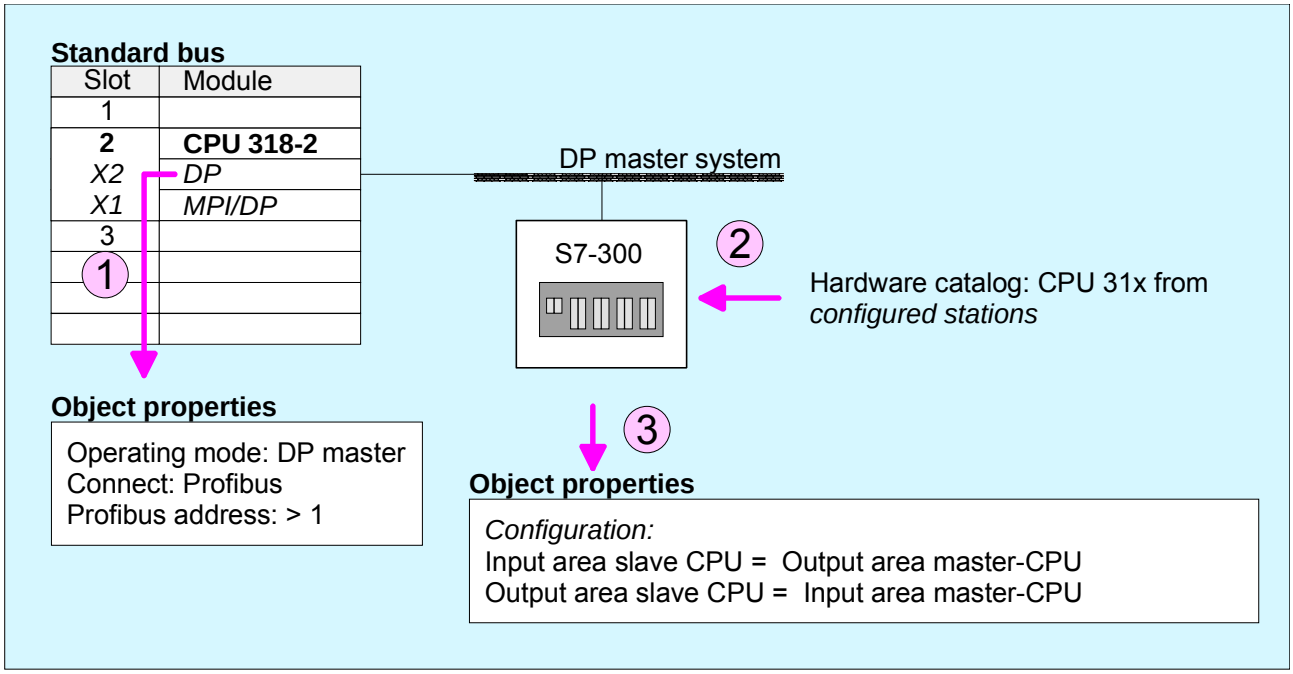
Slave section



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 and compile your project.

Master section



Note!

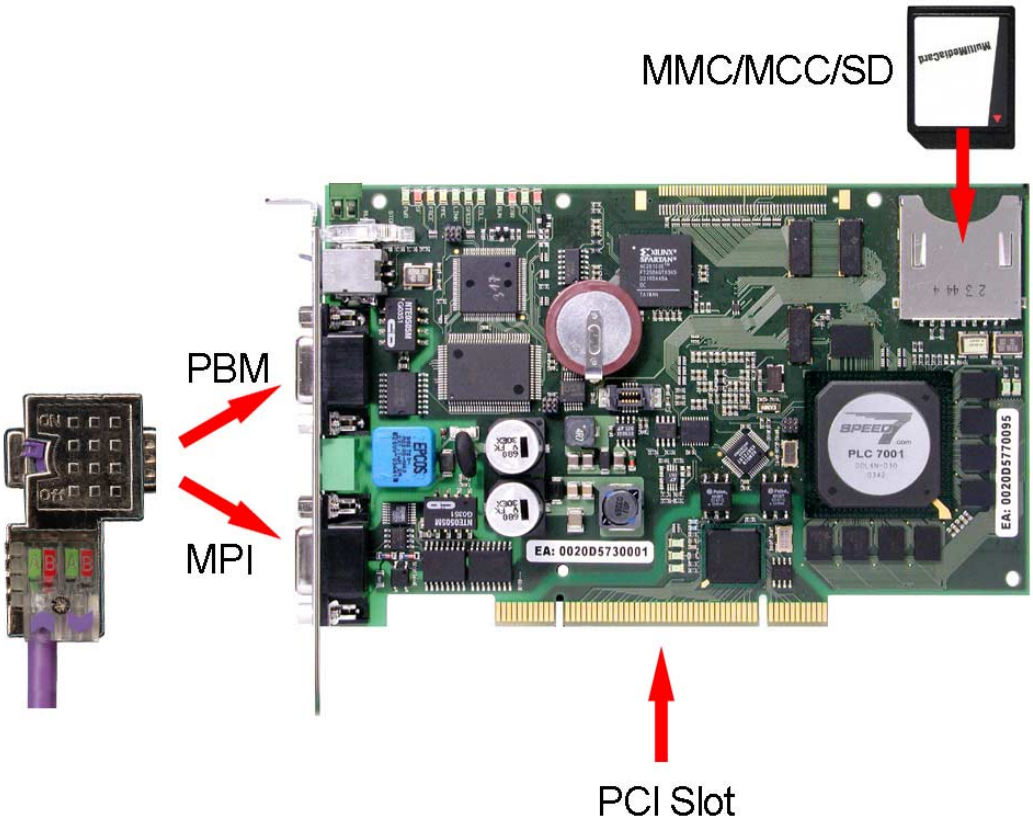
Data consistency can only be guaranteed for one *unit*! The choice "Data consistency by length" is not supported.

Project transfer

Overview

The following options are available to transfer a project into the CPU 51x slot card:

- internal via PCI slot (Ethernet connection)
- external via PC network card (routing necessary)
- external via MPI
- external via Profibus (not for first project)
- external via MMC storage card



More about the transfer methods may be found at chapter "Deployment CPU 51xS" at "Project transfer".

Profibus installation guidelines

Profibus in general

- A Profibus DP network may only be built up in linear structure.
- Profibus DP consists of minimum one segment with at least one master and one slave.
- A master has always been deployed together with a CPU.
- Profibus supports max. 124 participants.
- Per segment a max. of 32 participants is permitted.
- The max. segment length depends on the transfer rate:

9.6 ... 187.5kbit/s	→	1000m
500kbit/s	→	400m
1.5Mbit/s	→	200m
3 ... 12Mbit/s	→	100m

- Max. 10 segments may be built up. The segments are connected via repeaters. Every repeater counts for one participant.
- All participants are communicating with the same transfer rate. The slaves adjust themselves automatically on the transfer rate.
- 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 includes a 9pin slot "PBDP" where you connect the Profibus coupler into the Profibus network as a slave.

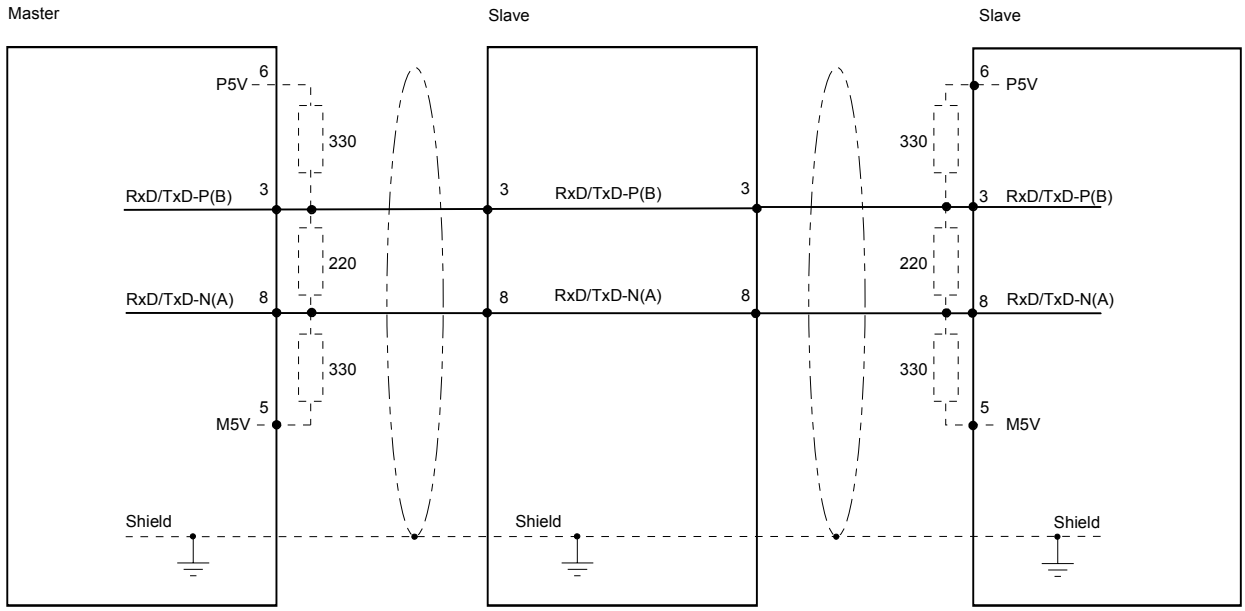
Max. 32 participants per segment are permitted. The segments are connected via repeaters. The maximum segment length depends on the transfer rate.

Profibus DP uses a transfer rate between 9.6kbit/s and 12Mbit/s, the slaves are following automatically. All participants are communicating with the same transfer rate.

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

Bus connection

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



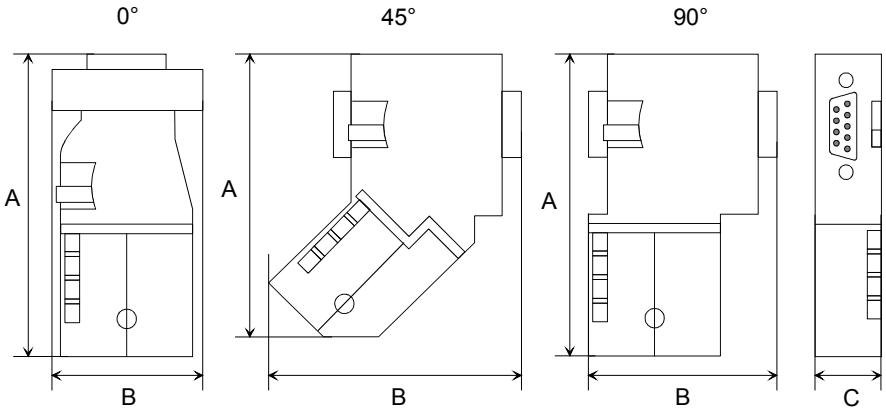
Note!

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

EasyConn bus connector

In 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.

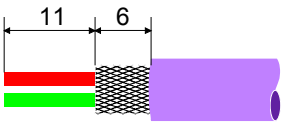


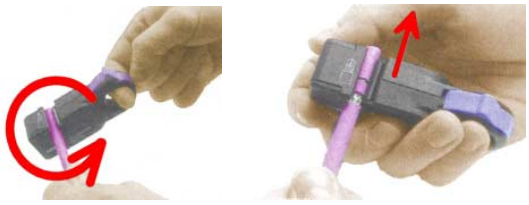
	0°	45°	90°
A	64	61	66
B	34	53	40
C	15.8	15.8	15.8

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

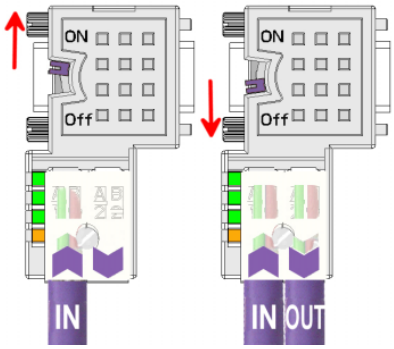




Dimensions in mm

Termination with "EasyConn"

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



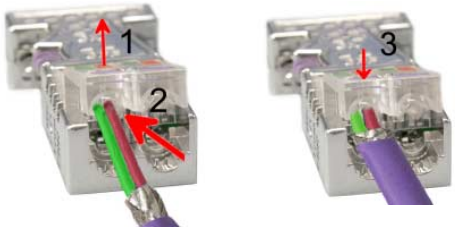
Attention!

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. A 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 RUN

- 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 5 Deployment PtP communication

Overview

Content of this chapter is the deployment of the RS485 slot for serial PtP communication.

Here you'll find all information about the protocols and project engineering of the interface, which are necessary for the serial communication using the RS485 interface.

Content

Topic	Page
Chapter 5 Deployment PtP communication	**5-1**
Fast introduction	5-2
Principle of the data transfer	5-3
Deployment of RS485 interface for PtP	5-4
Parameterization	5-6
Communication	5-9
Protocols and procedures	5-15
Modbus - Function codes	5-19
Modbus - Example communication	5-23

Fast introduction

General

Via a hardware configuration you may de-activate the Profibus part integrated to the CPU 51xS 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.

Parameterization

The parameterization of the serial interface 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.

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

Principle of the data transfer

Overview

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

Principle

Data, which are written 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.

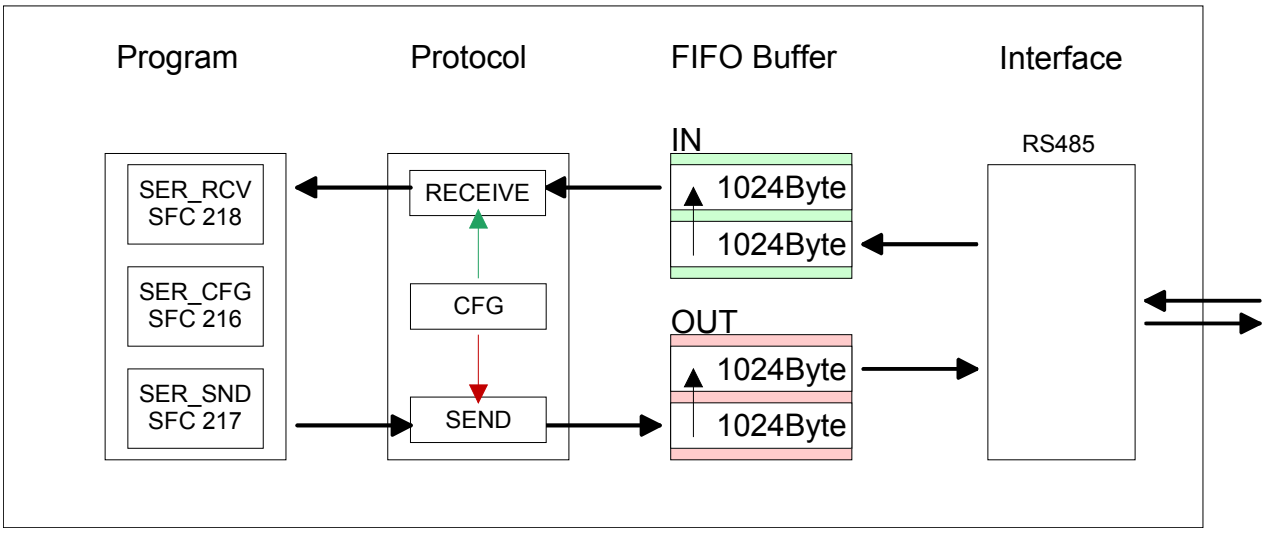
If the data is transferred via a protocol, the adoption of the data to the according protocol happens automatically.

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.

RS485 PtP communication



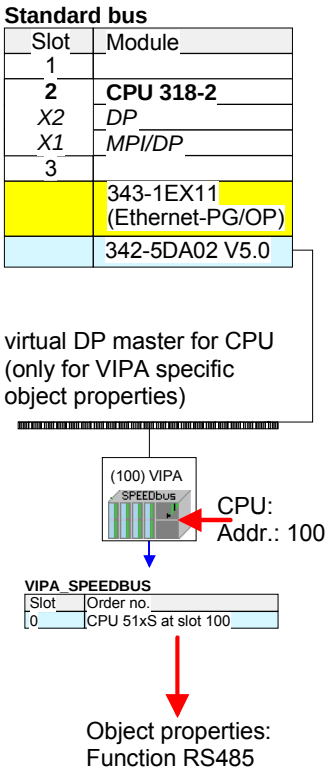
Deployment of RS485 interface for PtP

Switch to PtP operation

Per default, the RS485 interface X3 of the CPU is used for the Profibus DP master. Via hardware configuration the RS485 interfaces may be switched to point-to-point communication via the Parameter *Function RS485 X3* of the *Properties*.

Hardware configuration

The hardware configuration happens as described at "Project engineering" by means of a virtual Profibus master system with the following approach:



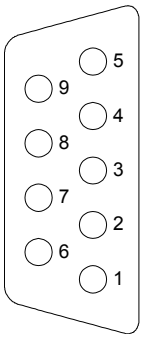
- Start the Siemens hardware configurator.
- Configure the Siemens CPU 318-2AJ00 (6ES7 318-2AJ00-0AB0/V3.0).
- Configure a Siemens CP 343-1 (343-1EX11) for the internal Ethernet PG/OP channel.
- Configure always as last module a Siemens DP master CP 342-5 (342-5DA02 V5.0). Connect and parameterize it at operation mode "DP-Master".
- Connect the slave system "VIPA_SPEEDbus".
- For the slave system set the Profibus address 100.
- Configure at slot 0 the VIPA CPU 51xS of the hardware catalog from VIPA_SPEEDbus.
- By double clicking the placed CPU 51xS the properties dialog of the CPU may be opened.
- Switch the Parameter *Function RS485 X3* to "PtP".

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

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 115.2kbit/s

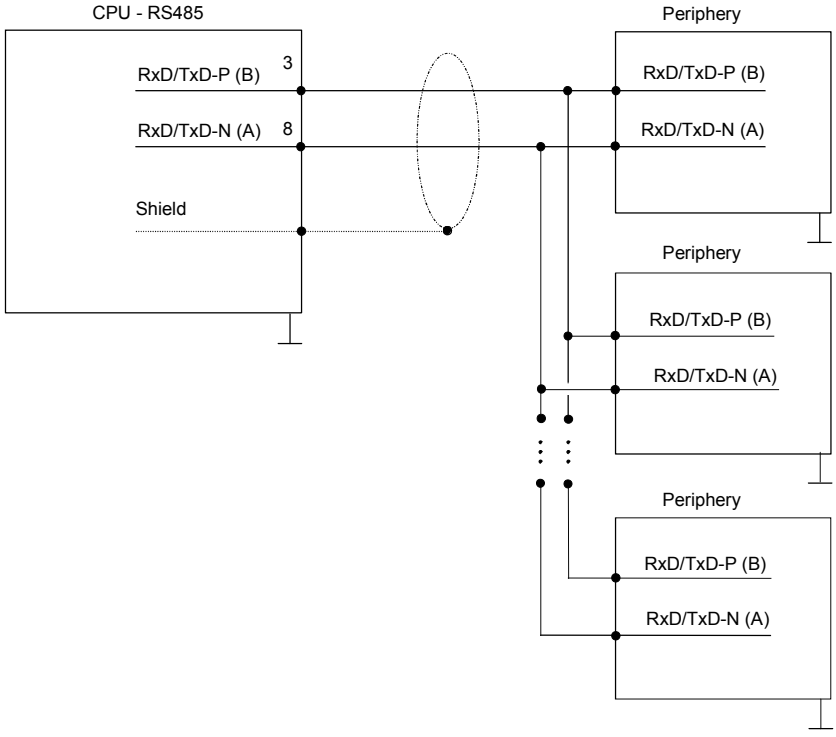
Connection RS485



9pin SubD jack

Pin	Assignment
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



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	Number of the baudrate
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	Return value (0 = OK)

Parameter description

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

Example: Wanted time 8ms at a baud rate 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!

With not used IDs please always enter FFh!

Data block at 3964R

DBB0:	Prio	BYTE	(The priority of both partners must be different)
DBB1:	ConnAttmptNr	BYTE	(Number of connection trials)
DBB2:	SendAttmptNr	BYTE	(Number of telegram retries)
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)

Baudrate

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

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

CharLen

Number of data bits where a character is mapped to.

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

Parity

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

0: NONE 1: ODD 2: EVEN

StopBits

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

1: 1Bit 2: 1.5Bit 3: 2Bit

FlowControl

The parameter FlowControl is ignored. When sending RST=1, when receiving RST=0.

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" (Parameter missing)
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 1. parameter
	2: Error 2. 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.

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	Return value (0 = OK)

DataPtr

Here you define a range of the type Pointer for the send buffer where the data that has to be sent 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)

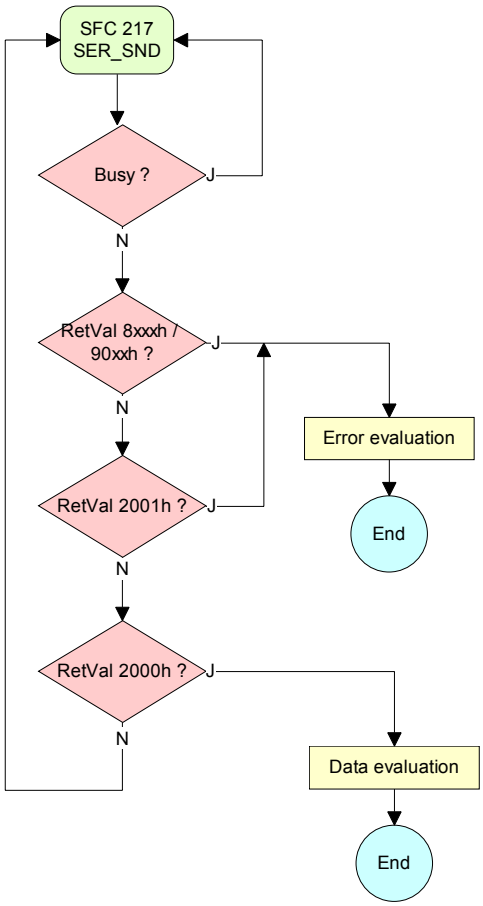
Modbus RTU/ASCII Master

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

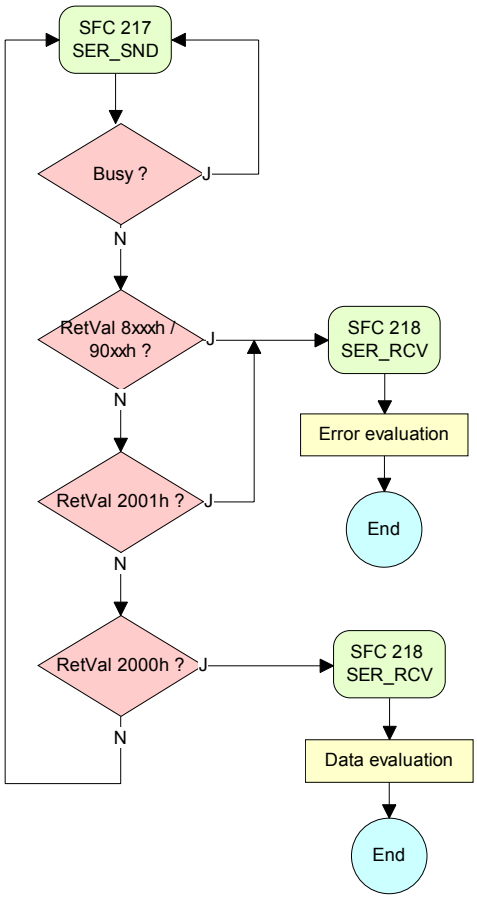
Principles of programming

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

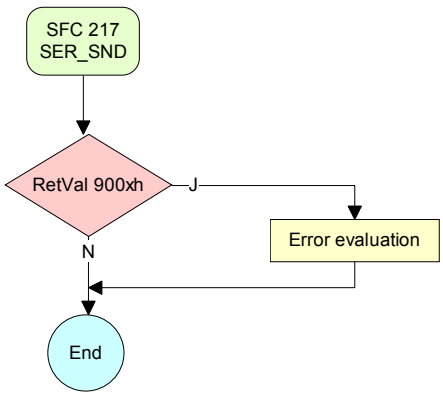
3964R



USS / Modbus



ASCII / STX/ETX



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	Return value (0 = OK)

DataPtr

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

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

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

DataLen

Word where the number of received Bytes is stored.

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

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

Error

This word gets an entry in case of an error. The following error messages may be created depending on the protocol:

ASCII

Bit	Error	Description
0	overrun	Overflow, a sign couldn't be read fast enough from the interface
1	framing error	Error that shows that a defined bit frame is not coincident, exceeds the allowed length or contains an additional Bit sequence (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.

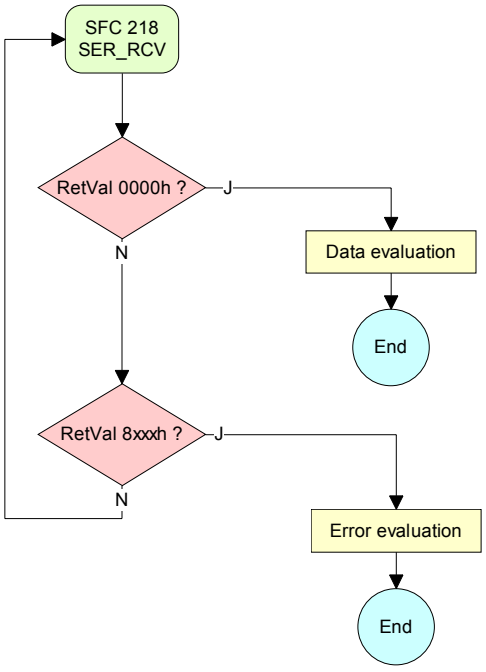
RetVal SFC 218 (Error message SER_RCV)

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.



Protocols and procedures

Overview

The CPU supports the following protocols and procedures:

- ASCII communication
- STX/ETX
- 3964R
- USS
- Modbus

ASCII

ASCII data communication is one of the simple forms of data exchange.

Incoming characters are transferred 1 to 1.

At ASCII, with every cycle the read-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.

STX/ETX

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

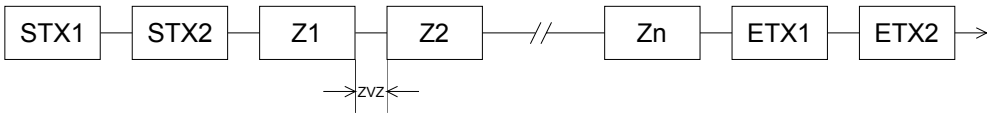
The STX/ETX procedure is suitable for the transfer of ASCII characters. It does not use block checks (BCC). Any data transferred from the periphery must be preceded by a start followed by the data characters and the end character.

Depending of the byte width the following ASCII characters can be transferred: 5Bit: not allowed: 6Bit: 20...3Fh, 7Bit: 20...7Fh, 8Bit: 20...FFh.

The effective data, which includes all the characters between Start and End are transferred to the 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).

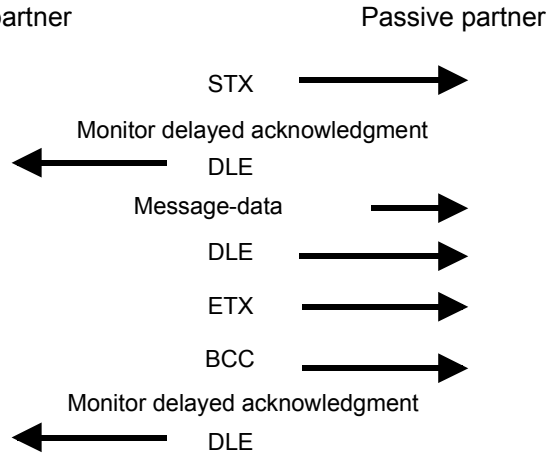
3964R

The 3964R procedure controls the data transfer of a point-to-point link between the CPU 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 **Tex**t
- DLE **D**ata **L**ink **E**scape
- ETX **E**nd of **Tex**t
- BCC **B**lock **C**heck **C**haracter
- NAK **N**egative **Ac**k**n**owledge

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

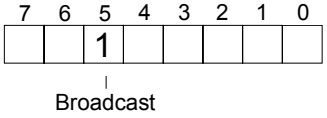
STX	LGE	ADR	PKE		IND		PWE		STW		HSW		BCC
02h			H	L	H	L	H	L	H	L	H	L	

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 sent as broadcast.

Modbus

The Modbus protocol is a communication protocol that fixes a hierarchic structure with one master and several slaves.

Physically, Modbus works with a serial half-duplex connection.

There are no bus conflicts occurring, because the master can only communicate with one slave at a time. After a request from the master, this waits for a preset delay time for an answer of the slave. During the delay time, communication with other slaves is not possible.

After a send command, the acknowledgement telegram must be read by a call of the 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 sent 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

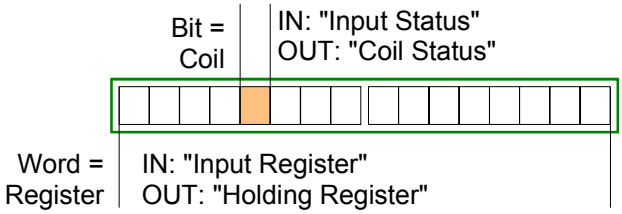
The following Modbus Protocols are supported by the RS485 interface:

- Modbus RTU Master
- Modbus ASCII Master

Modbus - Function codes

Naming convention

Modbus has some naming conventions:



- Modbus differentiates between bit and word access;
  Bits = "Coils" and Words = "Register".
- Bit inputs are referred to as "Input-Status" and Bit outputs as "Coil-Status".
- Word inputs are referred to as "Input-Register" and Word outputs as "Holding-Register".

Range definitions

Normally the access at Modbus happens by means of the ranges 0x, 1x, 3x and 4x.

0x and 1x gives you access to *digital* Bit areas and 3x and 4x to *analog* word areas.

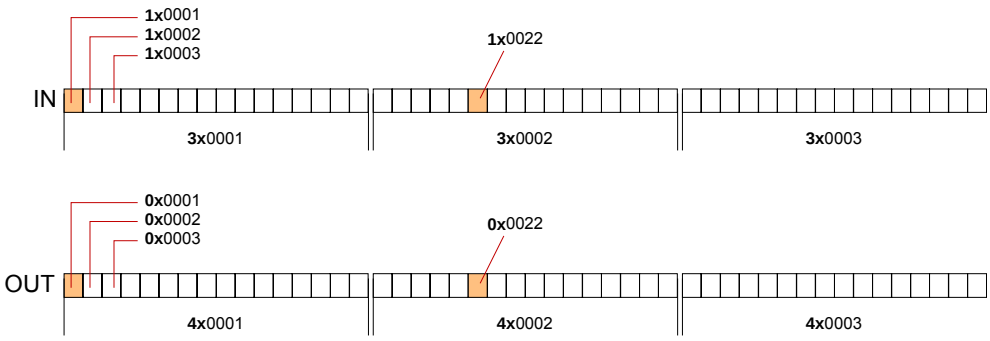
For the CPs from VIPA is not differentiating digital and analog data, the following assignment is valid:

0x: Bit area for master output data
Access via function code 01h, 05h, 0Fh

1x: Bit area for master input data
Access via function code 02h

3x: Word area for master input data
Access via function code 04h

4x: Word area for master output data
Access via function code 03h, 06h, 10h



A description of the function codes follows below.

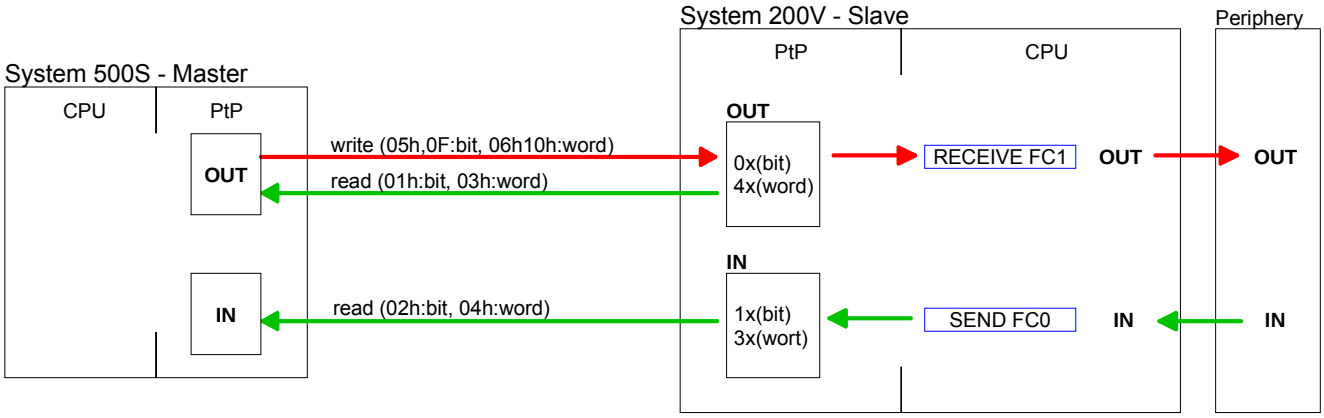
Overview

With the following Modbus function codes a Modbus master can access a Modbus slave. The description always takes place from the point of view of the master:

Code	Command	Description
01h	Read n Bits	Read n Bits of master output area 0x
02h	Read n Bits	Read n Bits of master input area 1x
03h	Read n Words	Read n Words of master output area 4x
04h	Read n Words	Read n Words master input area 3x
05h	Write 1 Bit	Write 1 Bit to master output area 0x
06h	Write 1 Word	Write 1 Word to master output area 4x
0Fh	Write n Bits	Write n Bits to master output area 0x
10h	Write n Words	Write n Words to master output area 4x

Point of View of "Input" and "Output" data

The description always takes place from the point of view of the master. Here data, which were sent from master to slave, up to their target are designated as "output" data (OUT) and contrary slave data received by the master were designated as "input" data (IN).



Respond of the slave

If the slave announces an error, the function code is send back with an "ORed" 80h. Without an error, the function code is sent back.

Slave answer: Function code OR 80h → Error
Function code → OK

Byte sequence in a Word

For the Byte sequence in a Word is always valid:

1 Word

High Byte	Low Byte

Check sum CRC, RTU, LRC

The shown check sums CRC at RTU and LRC at ASCII mode are automatically added to every telegram. They are not shown in the data block.

Read n Bits 01h, 02h

Code 01h: Read n Bits of master output area 0x
Code 02h: Read n Bits of master input area 1x

Command telegram

Slave address	Function code	Address 1. Bit	Number of Bits	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Word

Respond telegram

Slave address	Function code	Number of read Bytes	Data 1. Byte	Data 2. Byte	...	Check sum CRC/LRC
1Byte	1Byte	1Byte	1Byte	1Byte max. 250Byte		1Word

Read n Words 03h, 04h

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

Command telegram

Slave address	Function code	Address 1. Bit	Number of Words	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Word

Respond telegram

Slave address	Function code	Number of read Bytes	Data 1. Word	Data 2. Word	...	Check sum CRC/LRC
1Byte	1Byte	1Byte	1Word	1Word max. 125Words		1Word

Write 1 Bit 05h

Code 05h: Write 1 Bit to master output area 0x
A status change is via "Status Bit" with following values:

"Status Bit" = 0000h → Bit = 0
"Status Bit" = FF00h → Bit = 1

Command telegram

Slave address	Function code	Address Bit	Status Bit	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Word

Respond telegram

Slave address	Function code	Address Bit	Status Bit	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Word

Write 1 Word 06h

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

Command telegram

Slave address	Function code	Address word	Value word	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Word

Respond telegram

Slave address	Function code	Address word	Value word	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Word

Write n Bits 0Fh

Code 0Fh: Write n Bits to master output area 0x

Please regard that the number of Bits has additionally to be set in Byte.

Command telegram

Slave address	Function code	Address 1. Bit	Number of Bits	Number of Bytes	Data 1. Byte	Data 2. Byte	...	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Byte	1Byte	1Byte	1Byte	1Word
					max. 250Byte			

Respond telegram

Slave address	Function code	Address 1. Bit	Number of Bits	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Word

Write n Words 10h

Code 10h: Write n Words to master output area 4x

Command telegram

Slave address	Function code	Address 1. Word	Number of words	Number of Bytes	Data 1. Word	Data 2. Word	...	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Byte	1Word	1Word	1Word	1Word
					max. 125 Words			

Respond telegram

Slave address	Function code	Address 1. Word	Number of Words	Check sum CRC/LRC
1Byte	1Byte	1Word	1Word	1Word

Modbus - Example communication

Outline

The example establishes a communication between a master and a slave via Modbus. The following combination options are shown:

Modbus master (M)	Modbus slave (S)
CPU 51xS	CPU 21xSER-1

Components

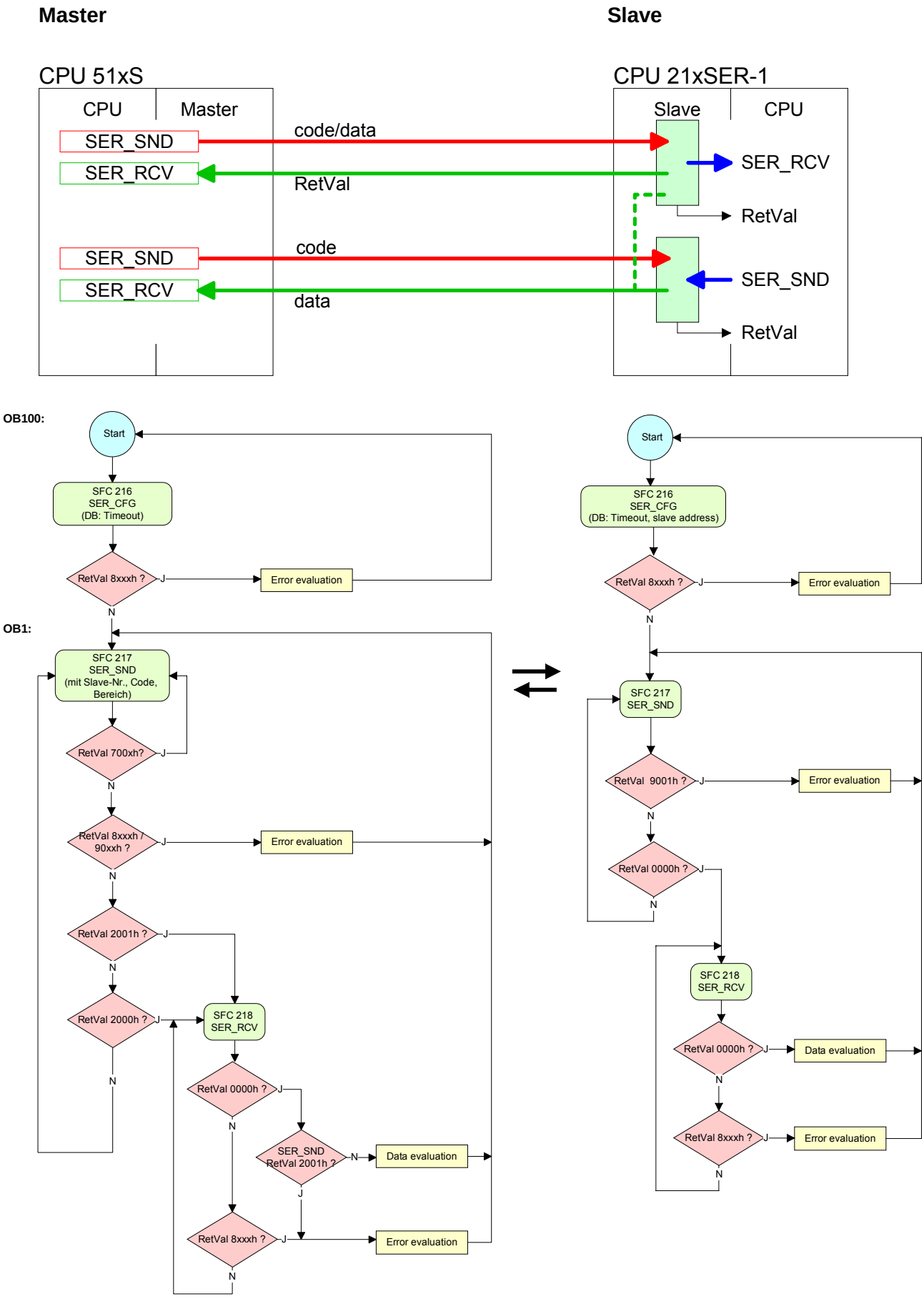
The following components are required for this example:

- CPU 51xS as Modbus RTU master
- CPU 21xSER-1 as Modbus RTU slave
- Siemens SIMATIC Manager and possibilities for the project transfer
- Modbus cable connection

Approach

- Assemble a Modbus system consisting of a CPU 51xS as Modbus master and a CPU 21xSER-1 as Modbus slave and Modbus cable.
- Execute the project engineering of the master!
  For this you create a PLC user application with the following structure:
  OB 100: Call SFC 216 (configuration as Modbus RTU master) with timeout setting and error evaluation.
  OB 1: Call SFC 217 (SER_SND) where the data is send with error evaluation. Here you have to build up the telegram according to the Modbus rules.
  Call SFC 218 (SER_RECV) where the data is received with error evaluation.
- Execute the project engineering of the slave!
  The PLC user application at the slave has the following structure:
  OB 100: Call SFC 216 (configuration as Modbus RTU slave) with timeout setting and Modbus address in the DB and error evaluation.
  OB 1: Call SFC 217 (SER_SND) for data transport from the slave CPU to the output buffer.
  Call SFC 218 (SER_RECV) for the data transport from the input buffer to the CPU. Allow an according error evaluation for both directions.

The following page shows the structure for the according PLC programs for master and slave.


Master
Slave
CPU 51xS
CPU
Master
SER_SND
SER_RCV
code/data
RetVal
SER_SND
SER_RCV
code
data
CPU 21xSER-1
Slave
CPU
SER_RCV
RetVal
SER_SND
RetVal
OB100:
Start
SFC 216
SER_CFG
(DB: Timeout)
RetVal 8xxxh ?
J
Error evaluation
N
OB1:
SFC 217
SER_SND
(mit Slave-Nr., Code,
Bereich)
RetVal 700xh?
J
N
RetVal 8xxxh /
90xxh ?
J
Error evaluation
N
RetVal 2001h ?
J
N
RetVal 2000h ?
J
SFC 218
SER_RCV
N
RetVal 0000h ?
J
N
SER_SND
RetVal 2001h ?
N
Data evaluation
J
RetVal 8xxxh ?
J
Error evaluation
N
Start
SFC 216
SER_CFG
(DB: Timeout, slave address)
RetVal 8xxxh ?
J
Error evaluation
N
SFC 217
SER_SND
RetVal 9001h ?
J
Error evaluation
N
RetVal 0000h ?
J
N
SFC 218
SER_RCV
RetVal 0000h ?
J
Data evaluation
N
RetVal 8xxxh ?
J
Error evaluation
N

Chapter 6 Deployment PLC-Tool

Overview

This chapter contains the description of the control software *PLC-Tool* from VIPA. PLC-Tool is a component of the OPC-Server package and is installed together with the OPC server at the standard installation.

The OPC-Server package can be found at the enclosed CD *SW-ToolDemo*.

Content

Topic	Page
Chapter 6 Deployment PLC-Tool	**6-1**
General	6-2
Setup and run of program	6-3
PLC-Tool Operation	6-4
Deployment PLC-Tool	6-7

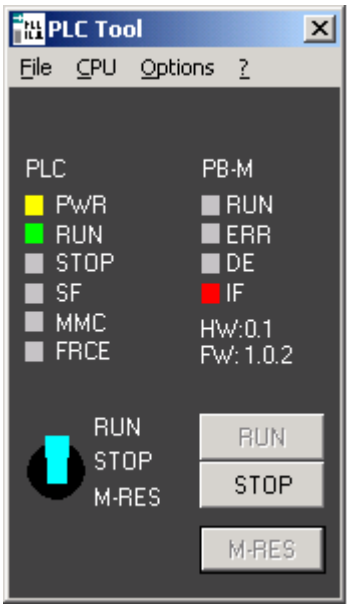
General

Overview

PLC-Tool

PLC-Tool is a program for operating the CPU 51xS.

The OPC-Server is required for communicating with the CPU. The OPC-Server has to be installed on the PC. The PLC-Tool enables you to "talk“ to external CPUs which are connected via MPI to the serial interface of the PC.



The operating surface (see figure above), which is a schematically top view of a CPU, serves for monitoring and operating the CPU. Here, the status of the LEDs on the CPU as well as the position of the mode switch are shown.

Tray-Icon

When starting the program, it installs itself also as a small icon (Tray-Icon) in the windows tool bar.

The Tray-Icon also visualizes the status of the CPU. This example here shows the CPU in run status. The program can be started repeatedly in order to simultaneously operate and monitor several CPUs. For each connection to a CPU, you have to assign an own MPI-address.

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

Setup and run of program

System requirements

For deployment of the PLC-Tool on your PC there are the following system requirements:

- PC with Windows operating system (Windows 2000 or higher, Windows XP professional or Windows NT 4.0 Service package 6)
- at least 32MB work memory (64MB are recommended)
- for installation about 10MB for OPC-Server and PLC-Tool

Requirements

The installation of the OPC-Server is required when using PLC-Tool, as the required drivers for the PLC-Tool will be installed on your PC by installing the OPC-Server.

Setup

As the PLC-Tool is a component of the OPC-Server package, the PLC-Tool will be installed together with the OPC-Server during the standard setup.

The PLC-Tool can also be installed separately. Like installing the OPC-Server, the installation of the PLC-Tool is supported by a setup-program.

Close all Windows-programs before starting the setup-program.

Insert the CD *SW-Tool/Demo*. The overview will be loaded via the auto-start function of the CD. From now on, you will be guided through the installation.

Run of program

PLC-Tool can be opened like any other PC-application. You have three options:

Start menu

In windows start menu, please click OPC! Then click on PLC-Tool!

Explorer

You can start the PLC-Tool by double clicking on file: *VPLCTool.exe* in directory C:\Programs\OPC Server.

Tool bar via Tray-Icon

As soon as PLC-Tool has been started, tray-icon (mini symbol) is shown in the start tool bar.

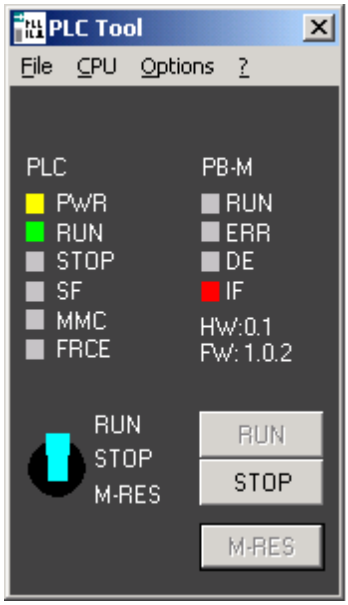
PLC-Tool can be opened by double clicking onto that tray-icon.

PLC-Tool Operation

Operating Dialog

Open operating dialog

The operating dialog will be opened after starting the program.



Main menü

Status indicator CPU

Status indicator Profibus master

Hardware release Firmware release

Operating modes CPU

Selection of operating modes

Main menu

The menu of the program consists of the following entries:

File	**CPU**	**Options**	**?(Help)**
- Minimize	*- New connection …*	*- Language*	*- Content*
- Exit	*- Connection diagnosis …*	*- Create link*	*- Help index*
	- Download WLD file	*- Always on top*	*- Use help*
			- Info

Name of the PLC system

Here, the name of your PLC system is shown. You can enter the name into the dialog box **CPU** > *New connection*.

Status indicators

The LED states of the corresponding CPU are copied into the status indicators. The set up of the status indicator depends on the CPU in use. As long as there is not a connection to the CPU, the status indicator is deactivated. Additionally, there is a status indication in the tool bar of your windows-system (tray-icon).

Operating mode switch

The push buttons which are accordingly to the operating mode either activated or deactivated, serve for adjusting the operating mode of the CPU. Additionally the physical status of the operating mode switch is shown on the desk top in form of a switch.

Structure of menu

File

Minimize

By using the command „minimize“ the operating dialog will be closed.

The program continues actively and will be stored as icon (tray-icon) on the tool bar.

Exit

Herewith, the program will be stopped and the tray-icon deleted from the tool bar.

CPU

New connection

With this command a dialog box will be opened. You can specify your connection to the CPU within this dialog box.

Connection diagnosis

When using this command a dialog box is opened which gives information regarding the effective connection.

Download WLD file

This function allows you to transfer wld files to the module.

Options

Language

When marking this command, a submenu containing a list of available languages for the surface is being opened. The active language is marked with a hook. The language on the program surface can be changed by clicking on another language.

Note!

As long as your operating system does not support languages, these languages will be shown as deactivated. The languages do exist but it is not possible to choose them.

Create link

Via setting up a link you can set up a link for your CPU connection which is momentarily active. In the dialog box you have to mention where you stored it.

Always on top

This function always puts the operating dialog onto the top level of the monitor. Therewith, the window is always visible, even then, when you are working with different applications. This function is marked with a hook, if active. By clicking anew onto that function – it can be deactivated again.

? (Help)

Content

This command opens an overview with topics of online-support.

Help Index

This command offers the option to have all catchwords regarding information for support alphabetically indicated. In that list you can go up and down by using the arrow-keys and stop at the word you are looking for to mark it. After having marked that word the appropriate text for help will be shown.

Use help

This function opens a window with standard-help for windows. Here, you can obtain information for using the help system.

Info

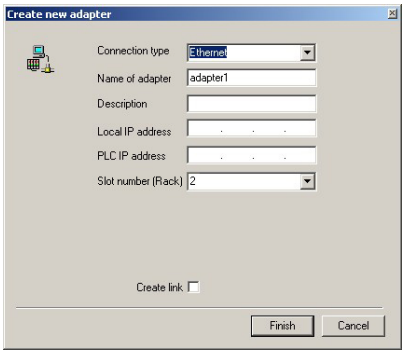
Via information you will obtain details about revision date of the PLC-Tool and copyright.

Deployment PLC-Tool

Establish connection to the CPU

With **CPU** > *New connection* the dialog window "Create new adapter" is opened.

To access the CPU 51xS set *Connection type* to "Ethernet". The following dialog window is opened.



Name of adapter Please enter here an unique name! The name should signify the PLC system in which your CPU is, e.g. "mixer".

Description Into this dialog box, you may enter an additional description, which explains your system more specifically. The assigned name in here will be given as tool-tip. If you don't assign a description – then the name of the adapter will be given as "tool-tip".

Local IP address The initialization by PLC function can only be established if Siemens SIMATIC manager and CPU 51xS slot card are at the same PC.
If the CPU 51xS slot card and PLC-Tool are at the same PC please enter here the IP address of the Ethernet part of the CPU 51xS.

If you want to access the CPU 51xS from an external PC via Ethernet, so you have to enter the IP address of the network card of the external PC. Additionally you have to set the routing to the CPU 51xS slot card in the target PC and to enter this route in the external PC. More concerning this may be found at "External access to PG/OP channel via routing".

PLC IP address Please enter here the IP address of the CPU part of the slot card

Slot number (Rack) Leave this parameter at 2.

Finish By clicking at [Finish] a connection to your CPU is established.

Note!

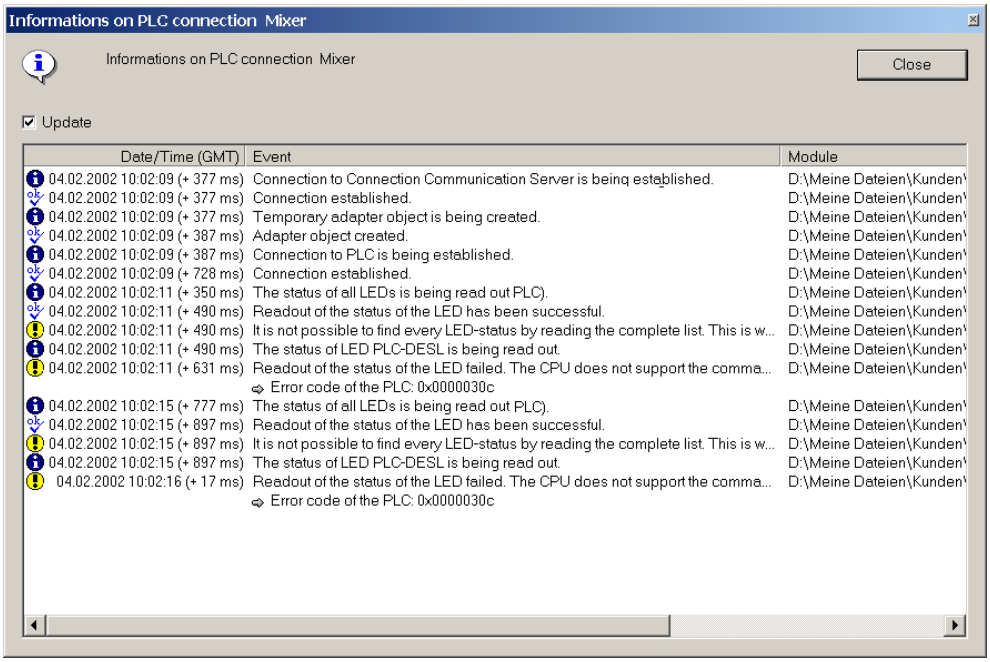
The adjustments made in the dialog box are only of temporary existence.

As soon as you close the PLC-Tool, your entries will be deleted. For securing your settings you should secure your data in form of a link via **options** > *create link*.

Connection diagnosis

Dialog box

The following dialog box will be opened under **CPU** > *connection diagnosis*:



This dialog box gives information about the effective connection.

Protocol of Procedures

Similar to the event protocol of windows, for indicating diagnosis data three procedure modes are used and are shown via a proper symbol.

The symbols have the following meaning:

A procedure was successfully finished.

A procedure is in process.

While in process, an error has occured.

Create link

Dialog box

When clicking **Options** > *create link* you reach a dialog box setting up a link. By starting the program via that link the PLC-Tool is being started and is automatically setting up the stored connection.

It's possible to enter the following inputs for the link which has to be created.

Link directory

By clicking on the arrow a list with different options is being opened – it contains various directories out of the start menu as well as the desk top. Choose the requested file for the link with your mouse.

Via entry „another folder" you can insert any other file for the link. For this, a standard dialog for a new folder is being opened.

Adapter

Via this selection-list you can see the connections already set up. This list is equivalent to the list in menu **CPU**.

In this selection-list you can find the connections already set up. You can also find this list in the connection-dialog box.

Language

In the menu **options** > *language* various languages are listed. Via selection list you have the option to choose one of these languages you prefer for the link and confirm it by mouse click.

Hidden (as icon shown in the tool bar)

By clicking this option it is reached that the program, when started via the link, will not me maximized but will only be started as icon in the tool bar.

Always on top

This option enables this program, when being started via the link, always to be atop all other programs on the monitor.

Note!
By saving your link in autostart (all users) with a setting „hidden“ the PCL-Tool is being started and secured as tray icon on the tool bar as soon your windows system is getting started.

Change of operating mode

Operating mode switch

The effective operation mode is indicated by LEDs.
The effective position of the operating mode switch on the CPU is visualized by a graphic in the PLC-Tool.

The switch has the following positions:

The CPU is in run modus.

The CPU is in stop modus.

M-Res (overall reset) – the CPU is in overall reset modus.

Push buttons

Next to the operating mode switch there are three push buttons with which the CPU can be positioned into the proper operation mode.

The following push buttons can be operated:

The CPU will be set into run modus.

The CPU will be set into run modus.

The CPU will be set into overall reset modus.

Note!
The push buttons are released or disabled for operation depending on the current operating mode (LED) and the effective position of the operating mode switch. Thereby, you can only use the push buttons which are suggestive for the current situation at a time.

Tray-Icon

Each entity of the program installs itself after starting as tray icon located in the windows-toolbar. When having finished the program, the tray icon will be deleted.

The tray icon has the following formats, according to the operating mode of the CPU:

CPU is in run modus

CPU is in start-up (changing from stop into run).

CPU is in stop modus

Status of CPU unknown (no connection).

Tool tip

When strolling over the tray icon with the mouse a small information window (tool tip) with the name of the adapter will be displayed.

The dialog is being opened by double-clicking on the symbol.

Right mouse click onto the symbol opens a menu via which the dialog can be called.

Besides, finishing of the program is offered via the menu.

Status indication

LEDs

For status indication, the PLC-Tool has LED-rows for the CPU and for the Profibus master. Use and the proper colors are to be found in below mentioned columns. For a detailed description of the LEDs, please refer to the manual for the respective CPU!

Not all LEDs listed below have to be always indicated. In fact, the PLC-Tool indicates the LEDs only which enable the PLC-Tool to read information from the CPU.

Status LEDs CPU	Description	Color	Meaning
	PWR	yellow	CPU is supplied with voltage.
	RUN	green	CPU is in run-status. If LED flashing, CPU is in start up.
	STOP	red	CPU is in STOP status.
	SF	red	Lights up when system error occurs
	MMC	red	Flashes when access to MMC
	FRCE	yellow	Lights up as soon as variables are fixed.
	DESL	yellow	Indicates Profibus slave activity as long as the integrated Profibus master is activated.

Status of LEDs Profibus master	Description	Color	Meaning
	RUN	green	Profibus master in operation. If LED flashes, Profibus master in start-up.
	ERR	red	Lights up when breakdown of slave
	DE	yellow	DE (Data exchange) indicates communication via Profibus
	IF	red	Initializing error when parameterization is faulty

Chapter 7 WinPLC7

Overview

In this chapter the programming and simulation software WinPLC7 from VIPA is presented. WinPLC7 is suited for every with Siemens STEP®7 programmable PLC.

Besides the system presentation and installation here the basics for using the software is explained with a sample project.

More information concerning the usage of WinPLC7 may be found in the online help respectively in the online documentation of WinPLC7.

Content

Topic	Page
Chapter 7 WinPLC7	**7-1**
System presentation	7-2
Installation	7-3
Example project engineering	7-4

System presentation

General

WinPLC7 is a programming and simulation software from VIPA for every PLC programmable with Siemens STEP®7.

This tool allows you to create user applications in FBD, LAD and STL.

Besides of a comfortable programming environment, WinPLC7 has an integrated simulator that enables the simulation of your user application at the PC without additional hardware.

This “Soft-PLC” is handled like a real PLC and offers the same error behavior and diagnosis options via diagnosis buffer, USTACK and BSTACK.

Note!

Detailed information and programming samples may be found at the online help respectively in the online documentation of WinPLC7.

Alternatives

There is also the possibility to use the Siemens SIMATIC manager instead of WinPLC7 from VIPA. Here the proceeding is part of this manual.

System requirements

- Pentium with 233MHz and 64Mbyte work space
- Graphics card with at least 16bit color - we recommend a screen resolution of at least 1024x768 pixel.
- Windows 98SE/ME, Windows 2000,
  Windows XP (Home and Professional), Windows Vista

Source

You may receive a *demo version* from VIPA. Without any activation with the *demo version* the CPUs 11x of the System 100V from VIPA may be configured.

To configure the SPEED7 CPUs a license for the "profi" version is necessary. This may be online received and activated.

There are the following sources to get WinPLC7:

Online

At www.vipa.de in the service area at *Downloads* a link to the current demo version and the updates of WinPLC7 may be found.

CD

Order no.	Description
SW211C1DD	WinPLC7 Single license, CD, with documentation in german
SW211C1ED	WinPLC7 Single license, CD, with documentation in english
SW900T0LA	ToolDemo VIPA software library free of charge respectively demo versions, which may be activated

Installation

Preconditions

The project engineering of a SPEED7 CPU from VIPA with WinPLC7 is only possible using an activated "Profi" version of WinPLC7.

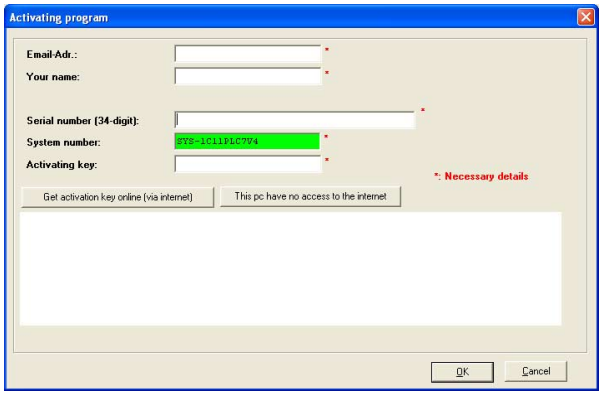
Installation WinPLC7 Demo

The installation and the registration of WinPLC7 has the following approach:

- For installation of WinPLC7 start the setup program of the corresponding CD respectively execute the online received exe file.
- Choose the according language.
- Agree to the software license contract.
- Set an installation directory and a group assignment and start the installation.

Activation of the "Profi" version

- Start WinPLC7. A "Demo" dialog is shown.
- Press the <q> key. The following dialog for activation is shown:



- Fill in the following fields:
  Email-Addr., *Your Name* und *Serial number*. The serial number may be found on a label at the CD case.
- If your computer is connected to Internet you may online request the *Activation Key* by [Get activation key via Internet]. Otherwise click at [This PC has no access to the internet] and follow the instructions.
- With successful registration the activation key is listed in the dialog window respectively is sent by email.
- Enter the activation key and click to [OK]. Now, WinPLC7 is activated as "Profi" version.

Installation of WinPCAP for station search via Ethernet

To find a station via Ethernet (accessible nodes) you have to install the WinPCAP driver. This driver may be found on your PC in the installation directory at WinPLC7-V4/WinPcap_4_0.exe.

Execute this file and follow the instructions.

Example project engineering

Job definition

In the example a FC 1 is programmed, which is cyclically called by the OB 1. By setting of 2 comparison values (*value1* and *value2*) during the FC call, an output of the PLC-System should be activated depending on the comparison result.

Project engineering for deployment in the simulator

For the output in the simulator the following should apply:
if *value1* = *value2* activate output Q 124.0
if *value1* > *value2* activate output Q 124.1
if *value1* < *value2* activate output Q 124.2

Precondition

- You have administrator rights for your PC.
- WinPLC7 is installed and activated as "Profi" version.

Hardware configuration

For the exclusive deployment in the simulator no hardware configuration is necessary.

Programming of the FC 1

The PLC programming happens by WinPLC7.

- Start WinPLC7 ("Profi" version)
- Create and open a new project by **File** > *Open/create a project*.

Creating block FC 1

- Choose **File** > *Create new block*.
- Enter "FC1" as block and confirm with [OK]. The editor for FC 1 is called.

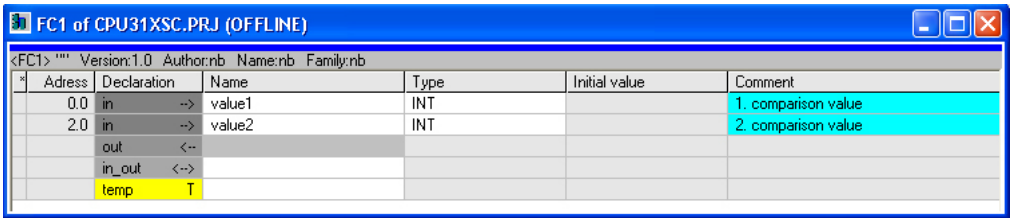
Creating parameters

In the upper part of the editor there is the *parameter table*. In this example the 2 integer values *value1* und *value2* are to be compared together. Since both values are read only by the function, these are to be defined as "in".

- Select the "in -->" row at the *parameter table* and enter at the field *Name* "value1". Press the [Return] key. The cursor jumps to the column with the data type.
- The data type may either directly be entered or be selected from a list of available data types by pressing the [Return] key. Set the data type to INT and press the [Return] key. Now the cursor jumps to the *Comment* column.
- Here enter "1. compare value" and press the [Return] key. A new "in -->" row is created and the cursor jumps to *Name*.
- Proceed for *value2* in the same way as described for *value1*.
- Save the block.

Parameter table

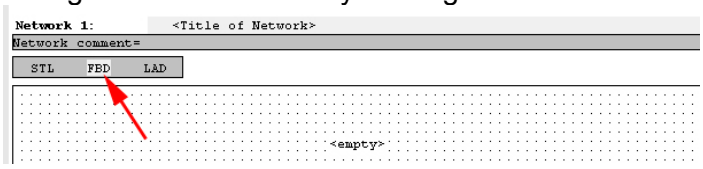
The parameter table shows the following entries, now:



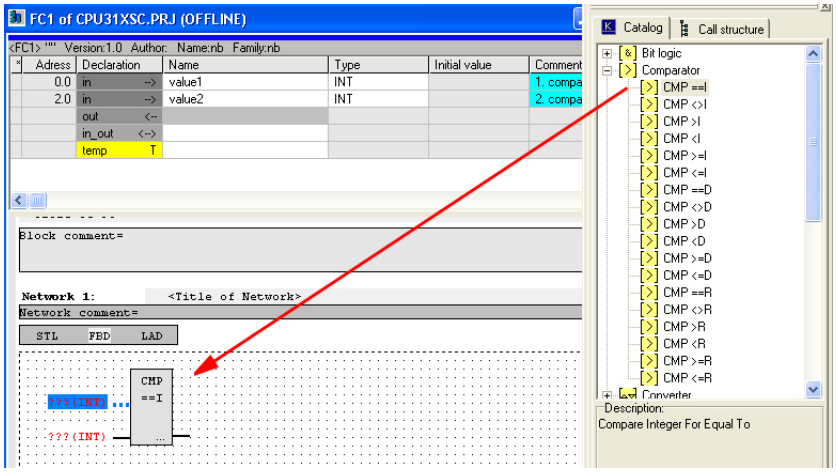
Enter the program

As requested in the job definition, the corresponding output is activated depending on the comparison of *value1* and *value2*. For each comparison operation a separate network is to be created.

- The program is to be created as FBD (function block diagram). Here change to the FBD view by clicking at FBD.



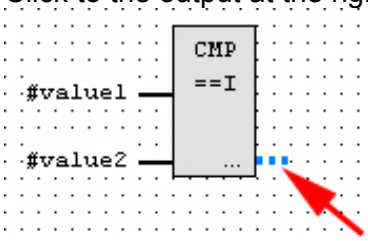
- Click to the input field designated as "empty".
  The available operations may be added to your project by drag&drop from the hardware catalog or by double click at them in the hardware catalog.
- Open in the catalog the category "Comparator" and add the operation "CMP==I" to your network.



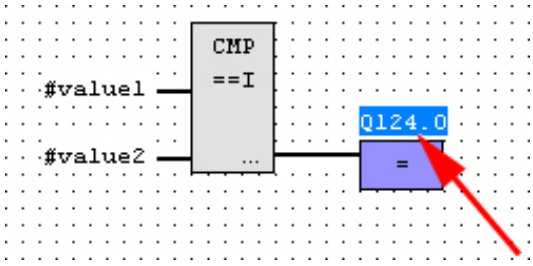
- Click to the input left above and insert *value1*. Since these are block parameters a selection list of block parameters may be viewed by entering "#".
- Type in "#" and press the [Return] key.
- Choose the corresponding parameter and confirm it with the [Return] key.
- Proceed in the same way with the parameter *value2*.

The allocation to the corresponding output, here Q 124.0, takes place with the following proceeding:

- Click to the output at the right side of the operator.



- Open in the catalog the category "Bit logic" and select the function "--[=]". The inserting of "--=" corresponds to the WinPLC7 shortcut [F7].
- Insert the output Q 124.0 by clicking to the operand.



Network1 is finished, now.

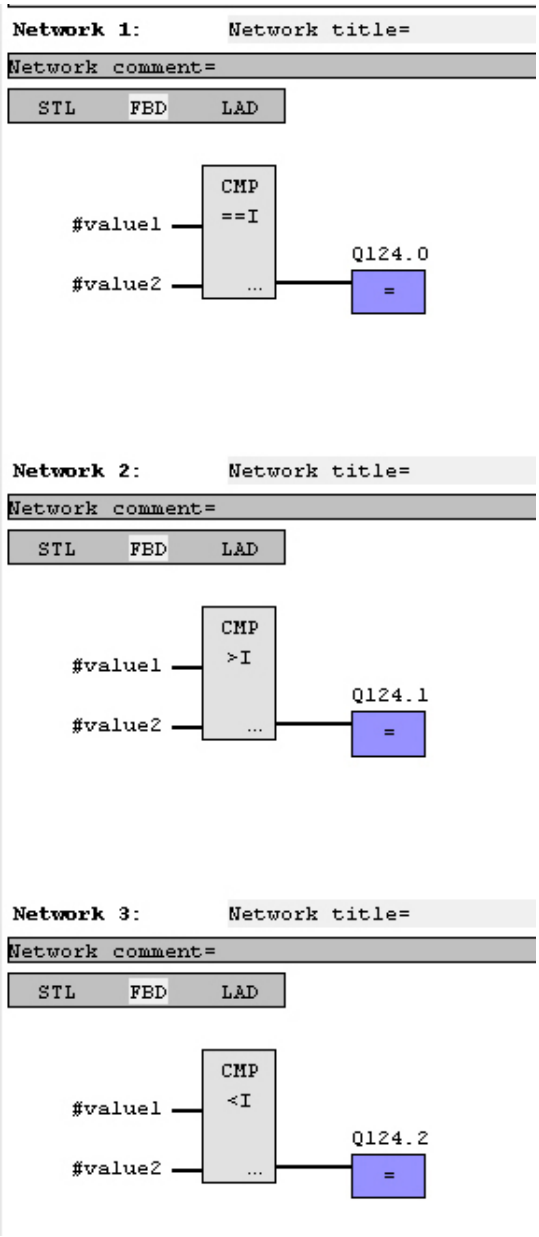
Adding a new network

For further comparisons the operations "CMP>I" at Q 124.1 and "CMP<I" at Q 124.2 are necessary. Create a network for both operations with the following proceeding:

- Move your mouse at an arbitrary position on the editor window and press the right mouse key.
- Select at the context menu "Insert new network". A dialog field is opened to enter the position and number of the networks.
- Proceed as described for "Network 1".
- Save the FC 1 with **File** > *Save content of focused window* respectively press [Strg]+[S].

FC 1

After you have programmed the still missing networks, the FC 1 has the following structure:



Creating the block OB 1

The FC 1 is to be called from the cycle OB 1.

- To create the OB 1 either you select **File** > *Create new block* or click to button [Display OB 1] and create the OB 1.
- Change to the STL view.
- Type in "Call FC 1" and press the [Return] key. The FC parameters are automatically displayed and the following parameters are assigned:

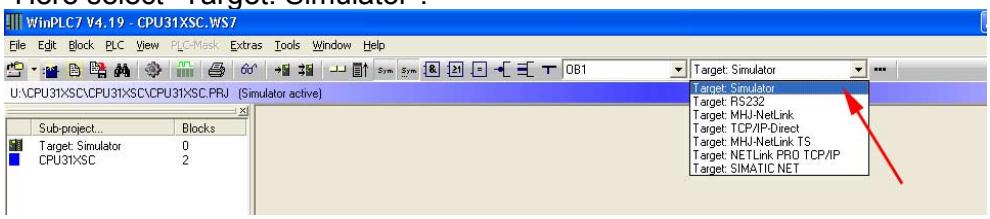
```
Network 1:        Network title=
Network comment=
  STL     FBD     LAD
  0:        CALL FC         1
  1:          value1:=10
  2:          value2:=10
```

- Save the OB 1 with **File** > *Save content of focused window* respectively *press* [Strg]+[S].

Test the PLC program in the *Simulator*

With WinPLC7 there is the possibility to test your project in a *simulator*.

- Here select "Target: Simulator".



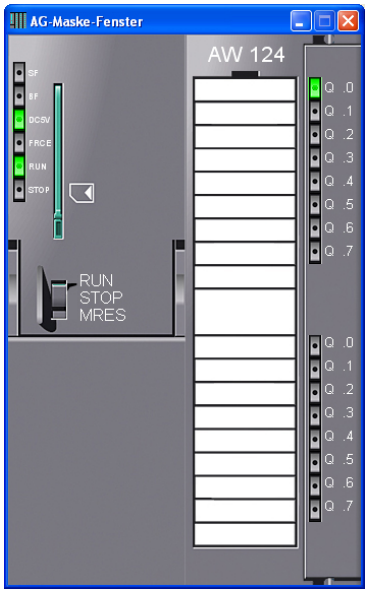
- Transfer the blocks to the simulator with **PLC** > *Send all* blocks.
- Switch the CPU to RUN, by clicking to the photo "Switch/Operating mode" and select in the dialog window the button [Warm restart]. The displayed state changes from STOP to RUN.
- To view the process image select **View** > *Display process image window*.
- Double click to the process image and enter at "Line 2" the address PQB124. Confirm with [OK]. A value marked by red color corresponds to a logical "1".
- Open the OB 1 with the button [Display OB 1].
- Change the value of one variable, save the OB 1 and transfer it to the simulator. According to your settings the process image changes immediately. The status of your blocks may be displayed with **Block** > *Monitoring On/Off*.

Visualization via PLC mask

A further component of the simulator is the *PLC mask*. Here a CPU is graphically displayed, which may be expanded by digital and analog peripheral modules.

As soon as the CPU of the simulator is switched to RUN state, inputs may be activated by mouse and outputs may be displayed.

- Open the PLC mask with **view** > *PLC mask*. A CPU is graphically displayed.
- By clicking the right mouse button within the PLC mask the context menu is opened. Choose for this example "Insert 16-port digital input module". The module is displayed at the right side of the CPU.
- Double-click to the output module, open its properties dialog and enter the *Module address* 124.
- Switch the operating mode switch to RUN by means of the mouse. Your program is executed and displayed in the simulator, now.



Project engineering for deployment in the CPU 51xS

For the output in the CPU 51xS the following should apply:
if *value1* = *value2* activate bit memory M 124.0
if *value1* > *value2* activate bit memory M 124.1
if *value1* < *value2* activate bit memory M 124.2

Precondition

- You have administrator rights for your PC.
- WinPLC7 is installed and activated as "Profi" version.
- Your CPU 51xS is installed and the PG/OP channel may be accessed via Ethernet.
- WinPCap for station search via Ethernet is installed.
- The power supply of the CPU and the I/O periphery are activated and the CPU is in STOP state.

Hardware configuration

- Start WinPLC7 ("Profi" version).
- Create and open a new project by **File** > *Open/create a project*.
- For the call of the hardware configurator it is necessary to set WinPLC7 from the *Simulator*-Mode to the *Offline*-Mode. For this and the communication via Ethernet set "Target: TCP/IP Direct".

- Start the hardware configurator with  . Please regard an object is selected with a double click at an object in the hardware configurator.
- Choose in the register *Select PLC-System* the parameter "VIPA SPEED7" and click to [Create]. A new station is created.
- Save the empty station. A station name and a comment may be entered before saving.
- By double click choose the according VIPA CPU in the hardware catalog at CPU SPEED7.
- For output place a digital output module and assign the start address 124.
- Save the hardware configuration.

Online access via Ethernet PG/OP channel

- Open the *CP343 properties* by double clicking to the CPU at slot 2 in the hardware configurator and selecting [Ethernet CP properties (PG/OP channel)] respectively by double clicking at "SPEED7 Ethernet (CP343)" slot 11 at UR3.
- Chose the register *Common Options*.
- Click to [Properties Ethernet].
- Choose the subnet "PG_OP_Ethernet".
- Enter a valid IP address-and a subnet mask. You may get this from your system administrator.
- Close every dialog window with [OK].
- Select, if not already done, "Target: External TCP/IP direct".
- Open with **Online** > *Send configuration to the CPU* a dialog with the same name.
- Click to [Accessible nodes]. Please regard to use this function it is necessary to install WinPCap before!
- Choose your network card and click to [Determining accessible nodes]. After a waiting time every accessible station is listed. Here your CPU with IP 0.0.0.0 is listed, too. To check this the according MAC address of the CPU is also listed.
- For the temporary setting of an IP address select you CPU and click to [Temporary setting of the IP parameters]. Please enter the same IP parameters, you configured in the CPU properties and click to [Write Parameters].
- Confirm the message concerning the overall reset of the CPU. The IP parameters are transferred to the CPU and the list of accessible stations is refreshed.
- Select you CPU and click to [Confirm]. Now you are back in the dialog "Send configuration".

Transfer hardware configuration

- Choose your network card and click to [Send configuration]. After a short time a message is displayed concerning the transfer of the configuration is finished.

Note!

Usually the online transfer of the hardware configuration happens within the hardware configurator.

With **File** > *Save active station in the WinPL7 sub project* there is also the possibility to store the hardware configuration as a system file in WinPLC7 to transfer it from WinPLC7 to the CPU.

The hardware configuration is finished, now and the CPU may always be accessed by the IP parameters as well by means of WinPLC7.

Programming of the FC 1

The PLC programming happens by WinPLC7. Close the hardware configurator and return to your project in WinPLC7.

The PLC program is to be created in the FC 1.

Creating block FC 1

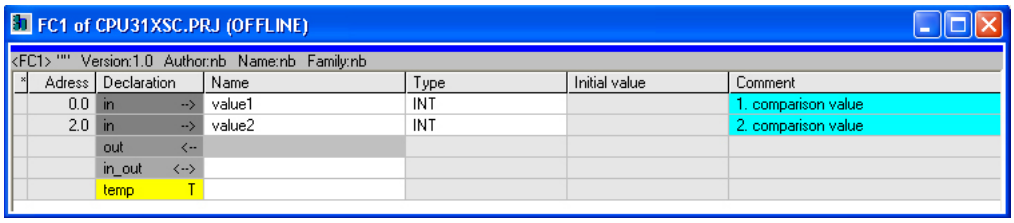
- Choose **File** > *Create new block*.
- Enter "FC1" as block and confirm with [OK]. The editor for FC 1 is called.

Creating parameters

In the upper part of the editor there is the *parameter table*. In this example the 2 integer values *value1* und *value2* are to be compared together. Since both values are read only by the function, these are to be defined as "in".

- Select the "in -->" row at the *parameter table* and enter at the field *Name* "value1". Press the [Return] key. The cursor jumps to the column with the data type.
- The data type may either directly be entered or be selected from a list of available data types by pressing the [Return] key. Set the data type to INT and press the [Return] key. Now the cursor jumps to the *Comment* column.
- Here enter "1. compare value" and press the [Return] key. A new "in -->" row is created and the cursor jumps to *Name*.
- Proceed for *value2* in the same way as described for *value1*.
- Save the block.

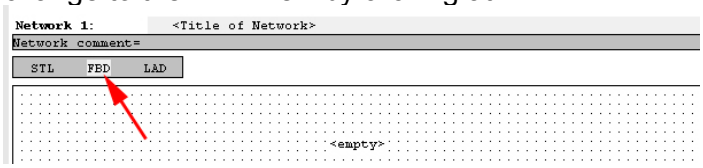
The parameter table shows the following entries, now:



Enter the program

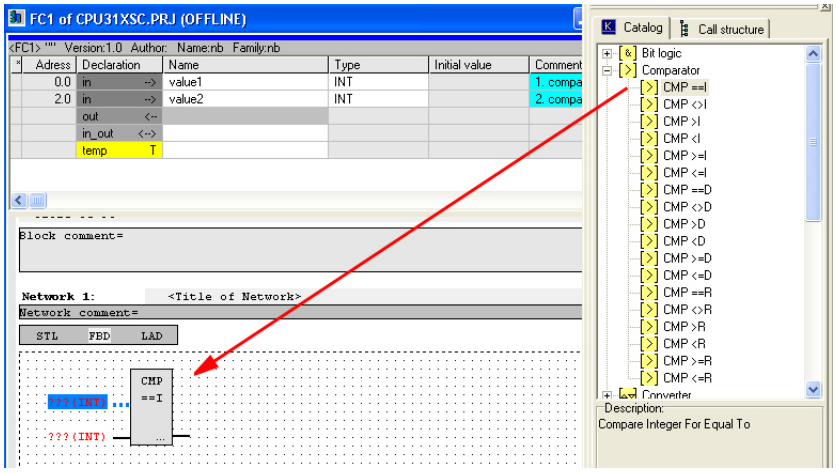
As requested in the job definition, the corresponding output is activated depending on the comparison of *value1* and *value2*. For each comparison operation a separate network is to be created.

- The program is to be created as FBD (function block diagram). Here change to the FBD view by clicking at FBD.



- Click to the input field designated as "empty".
  The available operations may be added to your project by drag&drop from the hardware catalog or by double click at them in the hardware catalog.

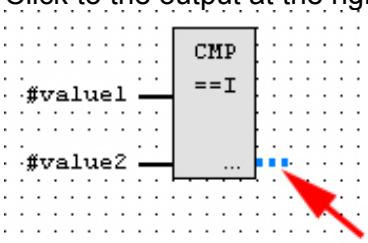
- Open in the catalog the category "Comparator" and add the operation "CMP==I" to your network.



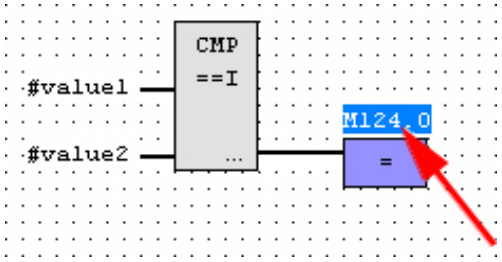
- Click to the input left above and insert *value1*. Since these are block parameters a selection list of block parameters may be viewed by entering "#".
- Type in "#" and press the [Return] key.
- Choose the corresponding parameter and confirm it with the [Return] key.
- Proceed in the same way with the parameter *value2*.

The allocation to the corresponding bit memory, here M 124.0, takes place with the following proceeding:

- Click to the output at the right side of the operator.



- Open in the catalog the category "Bit logic" and select the function "--[=]". The inserting of "--=" corresponds to the WinPLC7 shortcut [F7].
- Enter the bit memory M 124.0 by clicking to the operand.



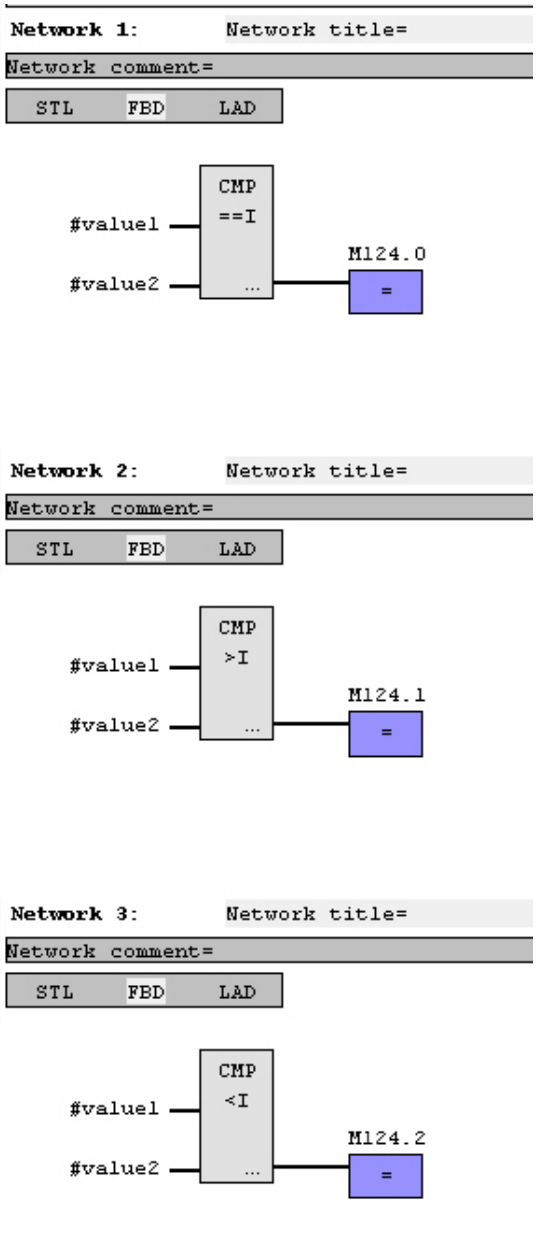
Network1 is finished, now.

Adding a new network

For further comparisons the operations "CMP>I" at M 124.1 and "CMP<I" at M 124.2 are necessary. Create a network for both operations with the following proceeding:

- Move your mouse at an arbitrary position on the editor window and press the right mouse key.
- Select at the context menu "Insert new network". A dialog field is opened to enter the position and number of the networks.
- Proceed as described for "Network 1".
- Save the FC 1 with **File** > *Save content of focused window* respectively press [Strg]+[S].

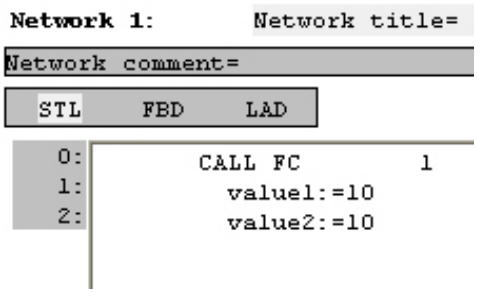
After you have programmed the still missing networks, the FC 1 has the following structure:



Creating the block OB 1

The FC 1 is to be called from the cycle OB 1.

- To create the OB 1 either you select **File** > *Create new block* or click to button [Display OB 1] and create the OB 1.
- Change to the STL view.
- Type in "Call FC 1" and press the [Return] key. The FC parameters are automatically displayed and the following parameters are assigned:



- Save the OB 1 with **File** > *Save content of focused window* respectively *press* [Strg]+[S].

Transfer PLC program to CPU and its execution

- For transfer to the CPU set the transfer mode to "Target: TCP/IP-Direct".
- For presetting the Ethernet data click to [...] and click to [Accessible nodes].
- Choose your network card and click to [Determining accessible nodes]. After a waiting time every accessible station is listed.
- Choose your CPU, which was provided with TCP/IP address parameters during the hardware configuration and click to [Confirm].
- Close the "Ethernet properties" dialog with [OK].
- Transfer the blocks to your CPU with **PLC** > *Send all blocks*.
- Switch your CPU to RUN state.
- Open the OB 1 with the button [Display OB 1].
- Change the value of one variable, save the OB 1 and transfer it to the CPU. According to your settings the process image changes immediately. The status of your blocks may be displayed with **Block** > *Monitoring On/Off*.

Appendix

A Index

3
3964R 5-16
A
Accu 2-8
Address classes 1-5, 3-6
ASCII 5-15
Assembly 3-3
B
Basics 1-1
Break points 3-40
C
Clock 2-8
Components 2-4
Cycle time surveillance 3-41
D
Deployment
CPU 517S/DPM 3-1
Profibus DP master 4-1
PtP communication 5-1
Diagnostic
Buffer 3-53
Driver 3-4
E
Environment conditions 1-10
ESD 1-10
Event-ID 3-53
F
Factory reset 3-47
Fast introduction
PtP communication 5-2
Firmware
Info by Web page 3-45
transfer 3-46
update 3-44
First start-up 3-4
H
Hardware description 2-1
Host-ID 1-3, 3-5
reserved 1-5, 3-6
I
Initialisation
Ethernet component 3-4
Initialization
CPU component 3-10
Installation 3-2
Driver 3-4
IP address assignment 3-5
Overview 3-2
Interfaces
MPI 2-6
Profibus DP master 2-6
RS485
PtP 5-1
IP address
CPU component 3-23
K
Know-how protection 3-49
L
LEDs 2-4
M
MCC 3-48
Memory
expansion 3-48
management 2-7
MMC 3-38
-Cmd - Auto commands 3-51
Project transfer 3-38
Diagnostic 3-38
Modbus 5-18
Example 5-23
Function codes 5-19
Slave respond 5-20
Telegram 5-19
MPI
Interface 1-9
MPI interface 2-6
N
Net-ID 1-3, 3-5
O
OPC-Server 6-2
Operating mode switch 2-7
Operating modes 3-39
FLAG 3-40
Overall reset 3-42
Overview 1-9
P
Parameter 3-23
via CPU 318-2DP 3-23
VIPA spezific 3-30

PG/OP channel 3-14
external access..... 3-15
internal access..... 3-14
pkg file 3-44
PLC functions 3-58
PLC-Tool 6-1
Connection diagnosis 6-8
Create link 6-9
Deployment..... 6-7
Menue structure..... 6-5
Operating dialog 6-4
Operating mode..... 6-10
Setup 6-3
Status indication 6-12
System requirements..... 6-3
Procedures 5-16
Profibus
Connectors 4-9
De-isolating lengths 4-10
Installation guidelines 4-8
Line termination 4-10
Transfer medium 4-8
Transfer rate 4-8
Profibus DP master
Commissioning 4-11
Configuration 3-22
Deployment..... 4-1
as DP slave 4-5
Include DP slave..... 3-22
Interface..... 2-6
Overview..... 4-2
Project engineering..... 4-3
Start-up behavior 4-11
Synchronization modes 3-31
Project engineering 3-19
Project transfer 3-34
via MMC..... 3-38
via PC network card 3-35
via PCI slot..... 3-34
via RS485 3-36
Properties 2-2
PtP communication 5-1
3964R..... 5-16
ASCII 5-15
Broadcast 5-17
Communication 5-9
Error messages..... 5-8, 5-10, 5-13
Fast introduction..... 5-2
Modbus..... 5-18
Parameterization 5-6
Principle data transfer 5-3
Protocols 5-15
RS485 interface..... 5-4
SFCs 5-2, 5-9
STX/ETX 5-15
USS 5-17

R

Reset
factory..... 3-47
Overall 3-42
Routing..... 3-15
Example 3-17

S

Safety Information..... 1-2
Shock resistance 1-10
Structure..... 2-3
STX/ETX..... 5-15
Subnet mask..... 1-4, 3-6
Subnet-ID..... 1-3, 3-5

T

Technical Data..... 2-9
Test functions..... 3-57
Tray-Icon..... 6-2, 6-3

U

USS..... 5-17

V

Voltage supply..... 2-7, 3-8

W

Watchdog..... 3-41
Web page 3-18
WinPLC7..... 7-1
wld files..... 3-38, 3-49

M.Stich