VIPA System 200V

IM | Manual

HB97E_IM | RE_253-1NE00 | Rev. 12/44

November 2012

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.com

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 Declaration

Hereby, VIPA GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions.

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

Conformity Information

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

Trademarks

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

SPEED7 is a registered trademark of profichip GmbH.

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

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

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

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

Information product support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

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

Technical support

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

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

Contents

About this manual

This manual describes the System 200V Ethernet slave module IM 253-1NE00 from VIPA. Here you may find every information for commissioning and operation.

Overview

Chapter 1: Basics and Assembly

The focus of this chapter is on the introduction of the VIPA System 200V. Here you will find the information required to assemble and wire a controller system consisting of System 200V components.

Besides the dimensions the general technical data of System 200V will be found.

Chapter 2: Hardware description

Here the hardware components of the IM 253-1NE00 are described.
The technical data are at the end of the chapter.

Chapter 3: Deployment

Content of this chapter is the description of the Ethernet coupler IM 253NET from VIPA. It contains all information for commissioning of the Ethernet coupler.

The chapter starts with the basics. Here the basic expressions of the Ethernet communication are explained together with the guidelines for building up a network and the access to the Ethernet coupler.

The chapter ends with the used protocols and a sample for socket programming.

Objective and contents

This manual describes the System 200V Ethernet slave module IM 253-1NE00 from VIPA. It contains a description of the construction, project implementation and usage.

This manual is part of the documentation package with order number HB97E_IM and relevant for:

Product	Order number	as of state: HW
IM 253NET	VIPA 253-1NE00	01

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

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 IM 253NET is constructed and produced for:

- all VIPA System 200V components
- communication and process control
- general control and automation applications
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle

Danger!

This device is not certified for applications in

- in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

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

- Hardware modifications to the process control system should only be carried out when the system has been disconnected from power!
- Installation and hardware modification 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 and Assembly

Overview

The focus of this chapter is on the introduction of the VIPA System 200V. Here you will find the information required to assemble and wire a controller system consisting of System 200V components.

Besides the dimensions the general technical data of System 200V will be found.

Contents

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.

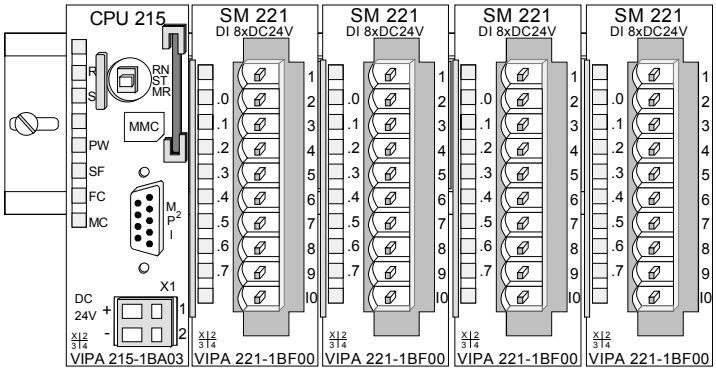
Attention!

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

System conception

Overview

The System 200V is a modular automation system for assembly on a 35mm profile rail. By means of the peripheral modules with 4, 8 and 16 channels this system may properly be adapted matching to your automation tasks.

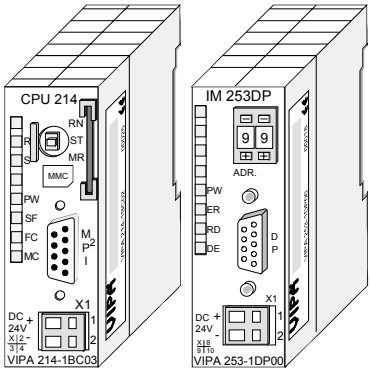


Components

The System 200V consists of the following components:

- *Head modules* like CPU and bus coupler
- *Periphery modules* like I/O, function und communication modules
- *Power supplies*
- *Extension modules*

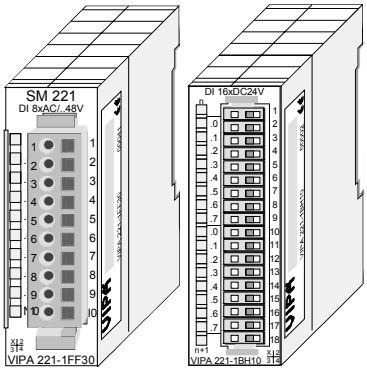
Head modules



With a head module CPU respectively bus interface and DC 24V power supply are integrated to one casing.

Via the integrated power supply the CPU respectively bus interface is power supplied as well as the electronic of the connected periphery modules.

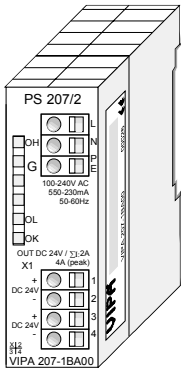
Periphery modules



The modules are direct installed on a 35mm profile rail and connected to the head module by a bus connector, which was mounted on the profile rail before.

Most of the periphery modules are equipped with a 10pin respectively 18pin connector. This connector provides the electrical interface for the signaling and supplies lines of the modules.

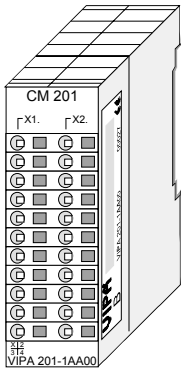
Power supplies



With the System 200V the DC 24V power supply can take place either externally or via a particularly for this developed power supply.

The power supply may be mounted on the profile rail together with the System 200V modules. It has no connector to the backplane bus.

Expansion modules



The expansion modules are complementary modules providing 2- or 3wire connection facilities.

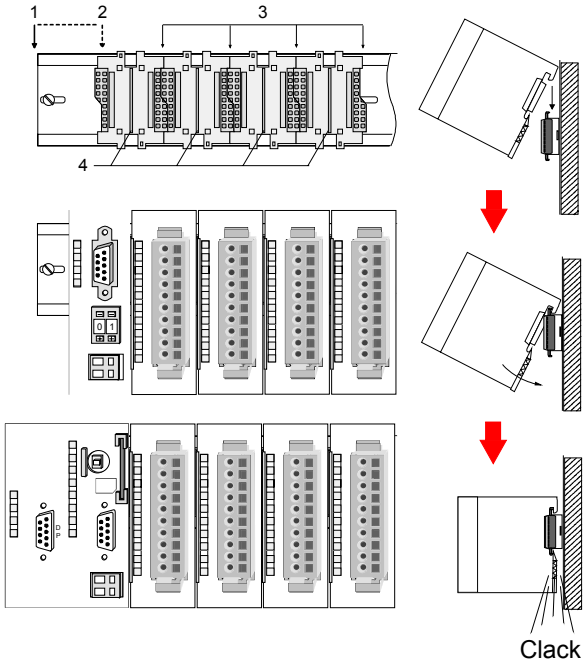
The modules are not connected to the backplane bus.

Structure/ dimensions

- Profile rail 35mm
- Dimensions of the basic enclosure:
  1tier width: (HxWxD) in mm: 76x25.4x74 in inches: 3x1x3
  2tier width: (HxWxD) in mm: 76x50.8x74 in inches: 3x2x3

Installation

Please note that you can only install head modules, like the CPU, the PC and couplers at slot 1 or 1 and 2 (for double width modules).



[1]	Head module (double width)
[2]	Head module (single width)
[3]	Periphery module
[4]	Guide rails

Note

Information about the max. number of pluggable modules and the max. current at the backplane bus can be found in the "Technical Data" of the according head module.

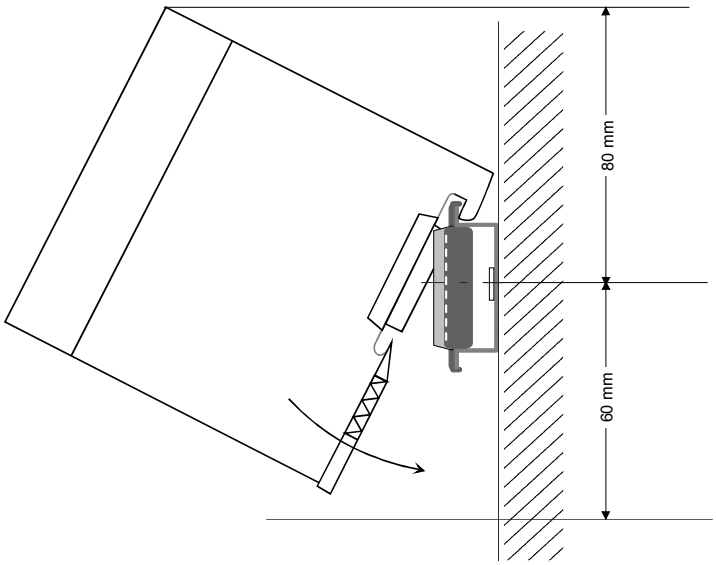
Please install modules with a high current consumption directly beside the head module.

Dimensions

Dimensions Basic enclosure

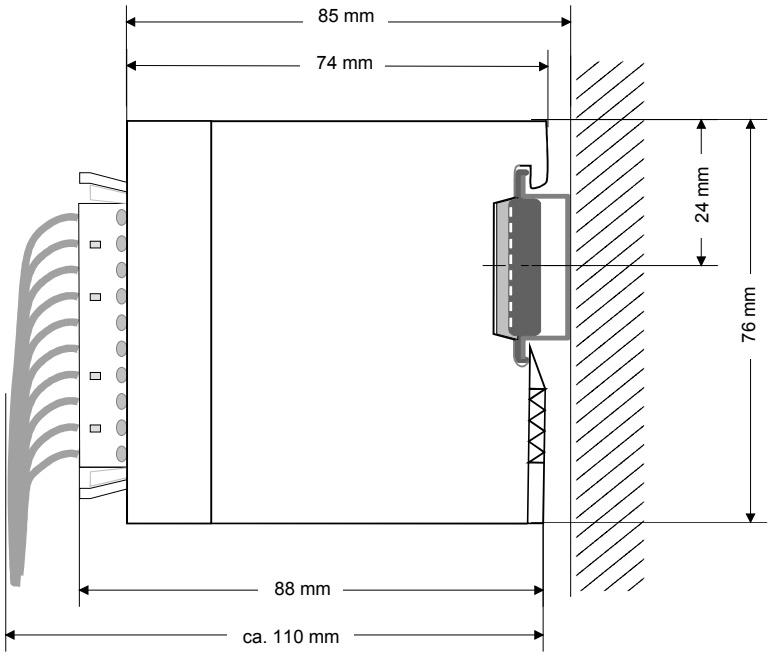
1tier width (HxWxD) in mm: 76 x 25.4 x 74
2tier width (HxWxD) in mm: 76 x 50.8 x 74

Installation dimensions

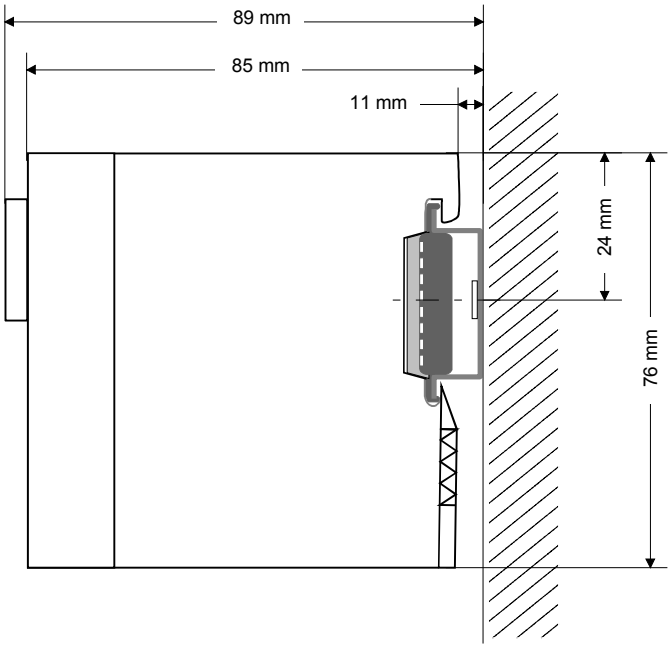


Installed and wired dimensions

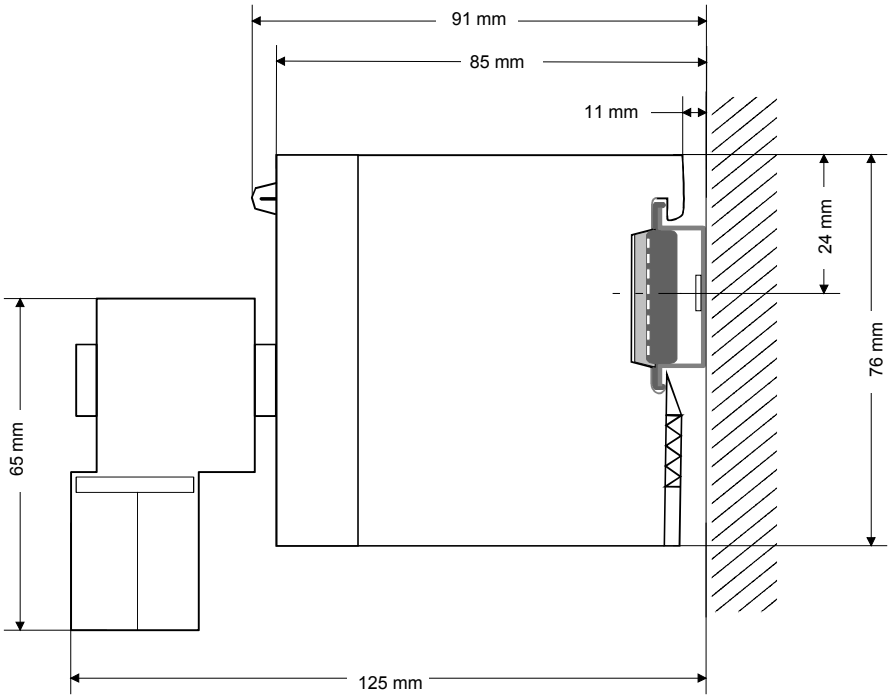
In- / Output modules



Function modules/
Extension modules



CPUs (here with
EasyConn from
VIPA)



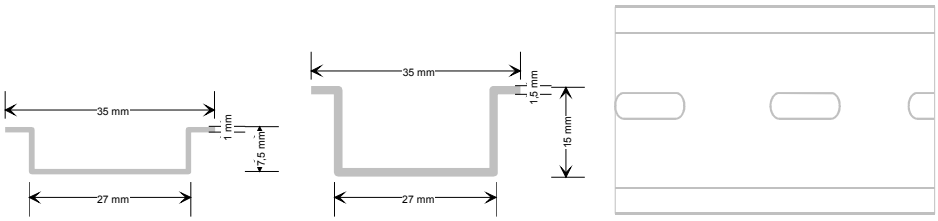
Installation

General

The modules are each installed on a 35mm profile rail and connected via a bus connector. Before installing the module the bus connector is to be placed on the profile rail before.

Profile rail

For installation the following 35mm profile rails may be used:

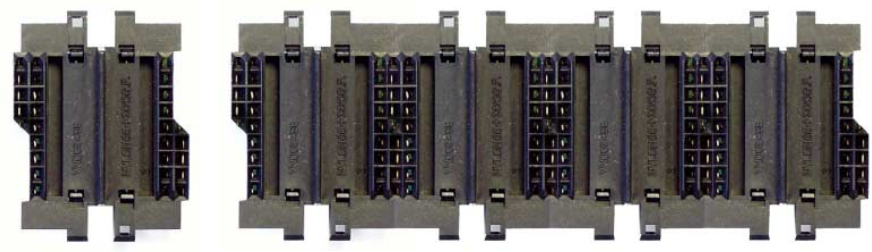


Order number	Label	Description
290-1AF00	35mm profile rail	Length 2000mm, height 15mm
290-1AF30	35mm profile rail	Length 530mm, height 15mm

Bus connector

System 200V modules communicate via a backplane bus connector. The backplane bus connector is isolated and available from VIPA in of 1-, 2-, 4- or 8tier width.

The following figure shows a 1tier connector and a 4tier connector bus:



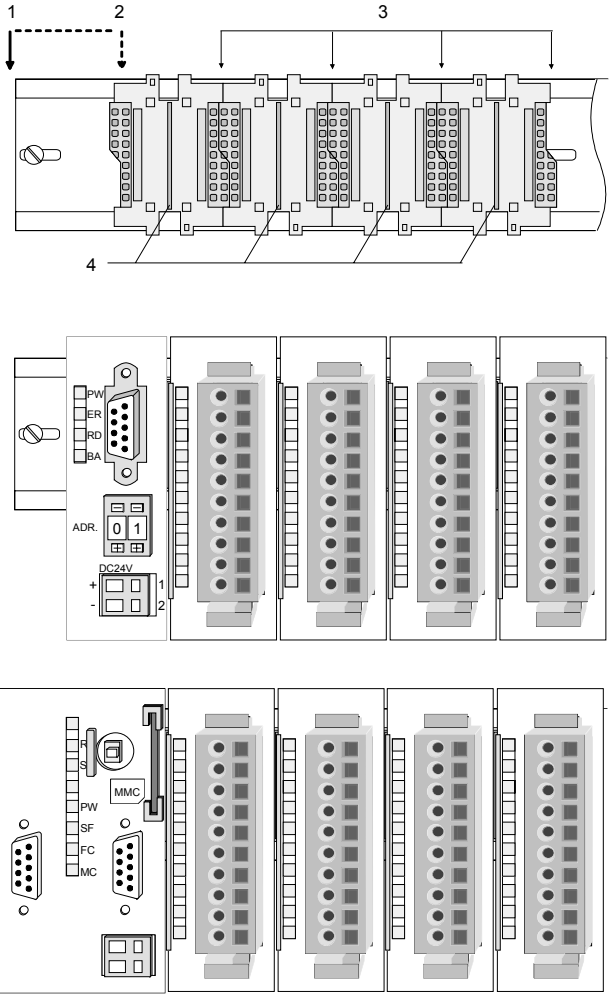
The bus connector is to be placed on the profile rail until it clips in its place and the bus connections look out from the profile rail.

Order number	Label	Description
290-0AA10	Bus connector	1tier
290-0AA20	Bus connector	2tier
290-0AA40	Bus connector	4tier
290-0AA80	Bus connector	8tier

Installation on a profile rail

The following figure shows the installation of a 4tier width bus connector in a profile rail and the slots for the modules.

The different slots are defined by guide rails.



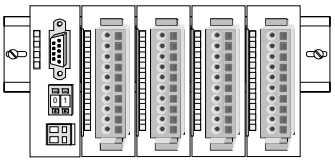
[1] Head module (double width)
[2] Head module (single width)
[3] Peripheral module
[4] Guide rails

Assembly regarding the current consumption

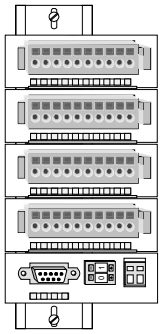
- Use bus connectors as long as possible.
- Sort the modules with a high current consumption right beside the head module. In the service area of www.vipa.com a list of current consumption of every System 200V module can be found.

Assembly possibilities

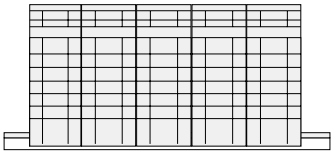
hoizontal assembly



vertical assembly



lying assembly

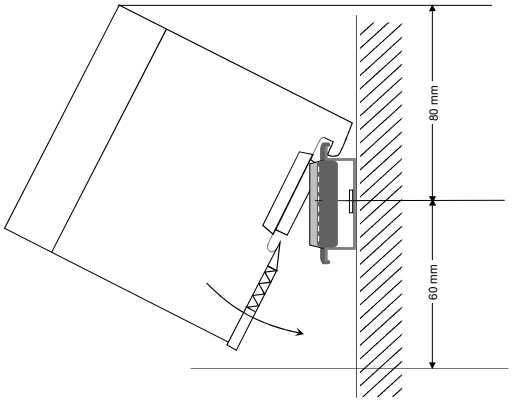


Please regard the allowed environmental temperatures:

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

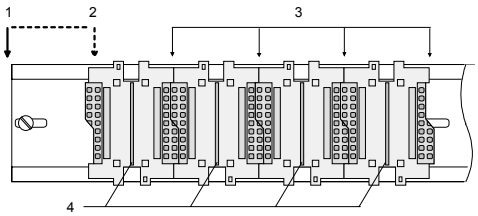
The horizontal assembly always starts at the left side with a head module, then you install the peripheral modules beside to the right.

You may install up to 32 peripheral modules.



Please follow these rules during the assembly!

- Turn off the power supply before you install or remove any modules!
- Make sure that a clearance of at least 60mm exists above and 80mm below the middle of the profile rail.



- Every row must be completed from left to right and it has to start with a head module.

  [1] Head module (double width)
  [2] Head module (single width)
  [3] Peripheral modules
  [4] Guide rails

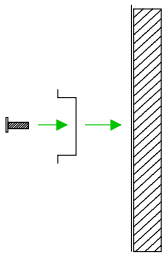
- Modules are to be installed side by side. Gaps are not permitted between the modules since this would interrupt the backplane bus.
- A module is only installed properly and connected electrically when it has clicked into place with an audible click.
- Slots after the last module may remain unoccupied.

Note!

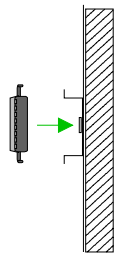
Information about the max. number of pluggable modules and the max. current at the backplane bus can be found in the "Technical Data" of the according head module.

Please install modules with a high current consumption directly beside the head module.

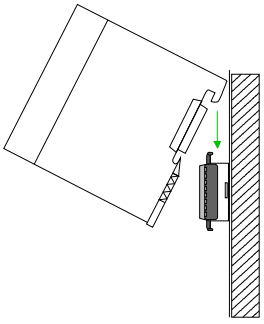
Assembly procedure



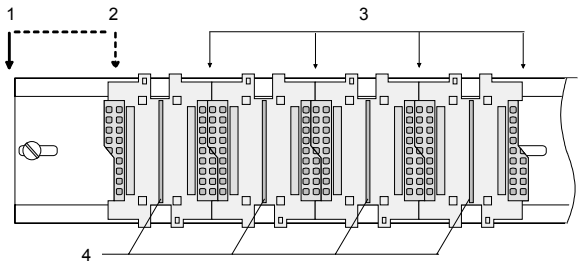
- Install the profile rail. Make sure that a clearance of at least 60mm exists above and 80mm below the middle of the profile rail.



- Press the bus connector into the profile rail until it clips securely into place and the bus-connectors look out from the profile rail. This provides the basis for the installation of your modules.



- Start at the outer left location with the installation of your head module and install the peripheral modules to the right of this.

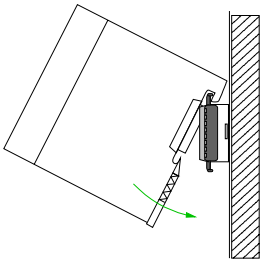


[1] Head module (double width)

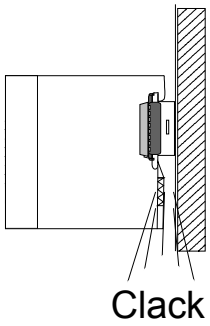
[2] Head module (single width)

[3] Peripheral module

[4] Guide rails



- Insert the module that you are installing into the profile rail at an angle of 45 degrees from the top and rotate the module into place until it clicks into the profile rail with an audible click. The proper connection to the backplane bus can only be guaranteed when the module has properly clicked into place.

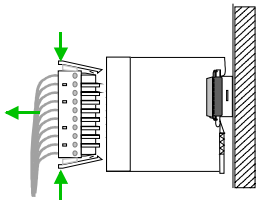


Attention!

Power must be turned off before modules are installed or removed!

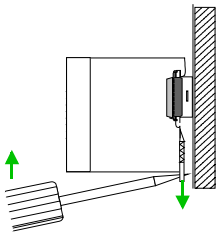
Demounting and module exchange

1



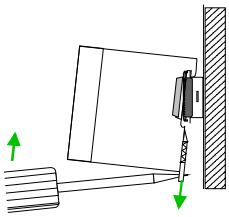
- Remove if exists the wiring to the module, by pressing both locking lever on the connector and pulling the connector.

2



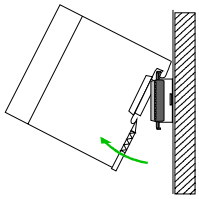
- The casing of the module has a spring loaded clip at the bottom by which the module can be removed.

3



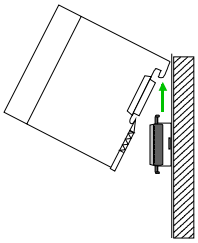
- The clip is unlocked by pressing the screwdriver in an upward direction.

4



- Withdraw the module with a slight rotation to the top.

5



Attention!

Power must be turned off before modules are installed or removed!

Please regard that the backplane bus is interrupted at the point where the module was removed!

Wiring

Overview

Most peripheral modules are equipped with a 10pole or a 18pole connector. This connector provides the electrical interface for the signaling and supply lines of the modules.

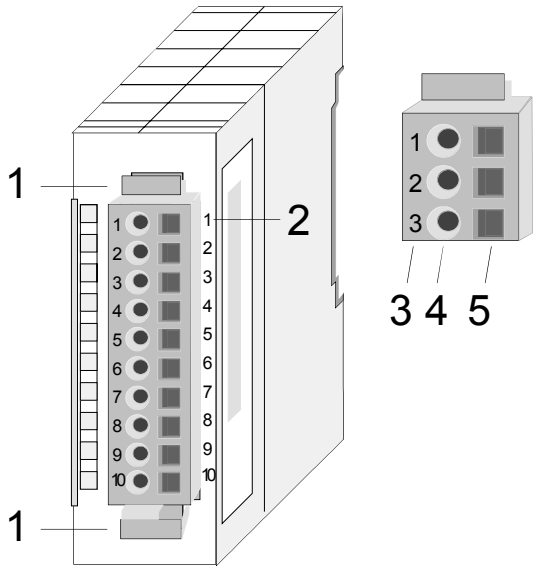
The modules carry spring-clip connectors for interconnections and wiring.

The spring-clip connector technology simplifies the wiring requirements for signaling and power cables.

In contrast to screw terminal connections, spring-clip wiring is vibration proof. The assignment of the terminals is contained in the description of the respective modules.

You may connect conductors with a diameter from 0.08mm^2 up to 2.5mm^2 (max. 1.5mm^2 for 18pole connectors).

The following figure shows a module with a 10pole connector.



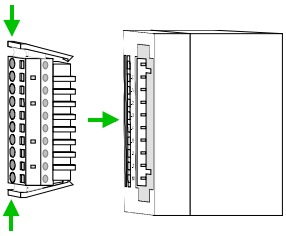
[1] Locking lever
[2] Pin no. at the module
[3] Pin no. at the connector
[4] Wiring port
[5] Opening for screwdriver

Note!

The spring-clip is destroyed if you push the screwdriver into the wire port!

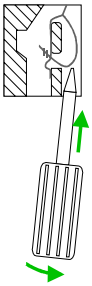
Make sure that you only insert the screwdriver into the square hole of the connector!

Wiring procedure

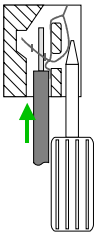


- Install the connector on the module until it locks with an audible click. For this purpose you press the two clips together as shown.
  The connector is now in a permanent position and can easily be wired.

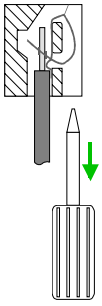
The following section shows the wiring procedure from top view.



- Insert a screwdriver at an angel into the square opening as shown.
- Press and hold the screwdriver in the opposite direction to open the contact spring.



- Insert the stripped end of the wire into the round opening. You can use wires with a diameter of 0.08mm^2 to 2.5mm^2 (1.5mm^2 for 18pole connectors).



- By removing the screwdriver the wire is connected safely with the plug connector via a spring.

Note!

Wire the power supply connections first followed by the signal cables (inputs and outputs).

Installation guidelines

General

The installation guidelines contain information about the interference free deployment of System 200V systems. There is the description of the ways, interference may occur in your control, how you can make sure the electromagnetic digestibility (EMC), and how you manage the isolation.

What means EMC?

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

All System 200V components are developed for the deployment in hard industrial environments and fulfill high demands on the EMC. Nevertheless you should project an EMC planning before installing the components and take conceivable interference causes into account.

Possible interference causes

Electromagnetic interferences may interfere your control via different ways:

- Fields
- I/O signal conductors
- Bus system
- Current supply
- Protected earth conductor

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

One differs:

- galvanic coupling
- capacitive coupling
- inductive coupling
- radiant coupling

Basic rules for EMC

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

- Take care of a correct area-wide grounding of the inactive metal parts when installing your components.
  - Install a central connection between the ground and the protected earth conductor system.
  - Connect all inactive metal extensive and impedance-low.
  - Please try not to use aluminum parts. Aluminum is easily oxidizing and is therefore less suitable for grounding.
- When cabling, take care of the correct line routing.
  - Organize your cabling in line groups (high voltage, current supply, signal and data lines).
  - Always lay your high voltage lines and signal res. data lines in separate channels or bundles.
  - Route the signal and data lines as near as possible beside ground areas (e.g. suspension bars, metal rails, tin cabinet).
- Proof the correct fixing of the lead isolation.
  - Data lines must be laid isolated.
  - Analog lines must be laid isolated. When transmitting signals with small amplitudes the one sided laying of the isolation may be favorable.
  - Lay the line isolation extensively on an isolation/protected earth conductor rail directly after the cabinet entry and fix the isolation with cable clamps.
  - Make sure that the isolation/protected earth conductor rail is connected impedance-low with the cabinet.
  - Use metallic or metalized plug cases for isolated data lines.
- In special use cases you should appoint special EMC actions.
  - Wire all inductivities with erase links.
  - Please consider luminescent lamps can influence signal lines.
- Create a homogeneous reference potential and ground all electrical operating supplies when possible.
  - Please take care for the targeted employment of the grounding actions. The grounding of the PLC is a protection and functionality activity.
  - Connect installation parts and cabinets with the System 200V in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If potential differences between installation parts and cabinets occur, lay sufficiently dimensioned potential compensation lines.

Isolation of conductors

Electrical, magnetically and electromagnetic interference fields are weakened by means of an isolation, one talks of absorption.

Via the isolation rail, that is connected conductive with the rack, interference currents are shunt via cable isolation to the ground. Hereby you have to make sure, that the connection to the protected earth conductor is impedance-low, because otherwise the interference currents may appear as interference cause.

When isolating cables you have to regard the following:

- If possible, use only cables with isolation tangle.
- The hiding power of the isolation should be higher than 80%.
- Normally you should always lay the isolation of cables on both sides. Only by means of the both-sided connection of the isolation you achieve high quality interference suppression in the higher frequency area.

  Only as exception you may also lay the isolation one-sided. Then you only achieve the absorption of the lower frequencies. A one-sided isolation connection may be convenient, if:
  - the conduction of a potential compensating line is not possible
  - analog signals (some mV res. µA) are transferred
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metalized plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.
- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to the System 200V module and **don't** lay it on there again!

Please regard at installation!

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

Remedy: Potential compensation line.

General data

Structure/ dimensions

- Profile rail 35mm
- Peripheral modules with recessed labelling
- Dimensions of the basic enclosure:
  1tier width: (HxWxD) in mm: 76x25.4x74 in inches: 3x1x3
  2tier width: (HxWxD) in mm: 76x50.8x74 in inches: 3x2x3

Reliability

- Wiring by means of spring pressure connections (CageClamps) at the front-facing connector, core cross-section 0.08 ... 2.5mm^2 or 1.5 mm^2 (18pole plug)
- Complete isolation of the wiring when modules are exchanged
- Every module is isolated from the backplane bus
- ESD/Burst acc. IEC 61000-4-2 / IEC 61000-4-4 (to level 3)
- Shock resistance acc. IEC 60068-2-6 / IEC 60068-2-27 (1G/12G)
- Class of protection IP20

Environmental conditions

- Operating temperature: 0 ... +60°C
- Storage temperature: -25 ... +70°C
- Relative humidity: 5 ... 95% without condensation
- Ventilation by means of a fan is not required

Chapter 2 Hardware description

Overview

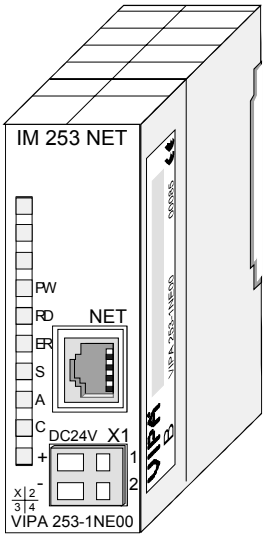
Here the hardware components of the IM 253-1NE00 are described. The technical data are at the end of the chapter.

Contents

Properties

IM 253NET
253-1NE00

- Ethernet coupler with ModbusTCP and Siemens S5 Header protocol
- max. 32 modules connectable with max. 256/256Byte in-/output data
- I/O access with both protocols via PC software like e.g. the OPC server from VIPA
- Online project engineering under WinNCS from VIPA with automatic coupler search and parameterization of modules in plain text. Here you may also fix IP address, subnet mask and coupler name and execute a firmware update.
- Extensive Alarm handling
- Integrated web server for test and diagnosis
- RJ45 jack 100BaseTX, 10BaseT
- Automatic polarity and speed recognition (auto negotiation)
- Automatic recognition of parallel or crossed cable (auto crossover)
- Network LEDs for link/activity, speed and collision
- Status-LEDs for Ready and Error

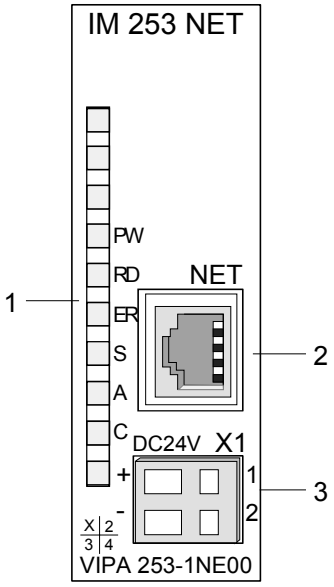


Ordering data

Type	Order number	Description
IM 253NET	VIPA 253-1NE00	Ethernet coupler

Structure

Front view
253-1NE00



[1] LED Status monitor
[2] RJ45 jack for Twisted Pair
[3] DC 24V voltage supply

Delivery default

IP address: 10.0.0.1

Password for alteration access via WinNCS: `00000000`

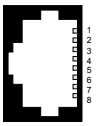
Attention!

For every Ethernet coupler is delivered with the IP address 10.0.0.1, you must not connect more than one new Ethernet coupler at one time.

First commissioning: Connect the new coupler with the network, assign a TCP/IP address. Now you may connect the next new coupler ...

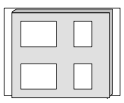
Interfaces

NET
RJ45



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

X1



\+ ① + DC 24 V
\- ② 0 V

RJ45 Ethernet connection

The RJ45 jack is the Twisted-Pair connection to Ethernet.

Power supply

The Ethernet coupler comes with an integrated power supply. The power supply has to be supplied with DC 24V via the front. By means of the supply voltage, the bus coupler electronic is supplied as well as the connected modules via backplane bus. The "max. current drain at backplane bus" can be found in the Technical Data.

The power supply is protected against inverse polarity and overcurrent, Ethernet and backplane bus are galvanically isolated.

LEDs

The Ethernet coupler has different LEDs for diagnosis and monitoring the operational state. The usage and meaning of the colors are described in the following table.

Label	Color	Description
PW	Green	Power: DC 24V voltage supply is present
RD	Green	Ready: The Ethernet coupler has booted. I/O periphery, connected to the backplane bus can be accessed.
ER	Red	Error: Shows an error like e.g. module failure or parameterization error (Details: see coupler web site)
S	Green	Speed: on: 100MBit, off: 10Mbit
A	Green	Activity: on: physically connected off: no physical connection blinking: shows bus activity
C	Green	Collision: on: full duplex operation active off: half duplex operation active blinking: collision detected

Technical data

Order number	**253-1NE00**
Type	IM 253NET, Ethernet slave
Technical data power supply	
Power supply (rated value)	DC 24 V
Power supply (permitted range)	DC 20.4...28.8 V
Reverse polarity protection	✓
Current consumption (no-load operation)	80 mA
Current consumption (rated value)	1 A
Inrush current	65 A
I^2t	0.85 A^2s
Max. current drain at backplane bus	3.5 A
Max. current drain load supply	-
Power loss	2.5 W
Status information, alarms, diagnostics	
Status display	yes
Interrupts	no
Process alarm	no
Diagnostic interrupt	no
Diagnostic functions	no
Diagnostics information read-out	possible
Supply voltage display	yes
Service Indicator	-
Group error display	red LED
Channel error display	none
Hardware configuration	
Racks, max.	1
Modules per rack, max.	32
Number of digital modules, max.	32
Number of analog modules, max.	16
Communication	
Fieldbus	Ethernet MODBUS/TCP and Siemens S5 Header
Type of interface	Ethernet 10/100 MBit
Connector	RJ45
Topology	Star topology
Electrically isolated	✓
Number of participants, max.	8
Node addresses	IP V4 address
Transmission speed, min.	10 Mbit/s
Transmission speed, max.	100 Mbit/s
Address range inputs, max.	256 Byte
Address range outputs, max.	256 Byte
Number of TxPDOs, max.	-
Number of RxPDOs, max.	-
Housing	
Material	PPE / PA 6.6
Mounting	Profile rail 35 mm
Mechanical data	
Dimensions (WxHxD)	25.4 x 76 x 78 mm
Weight	90 g
Environmental conditions	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
Certifications	
UL508 certification	yes

Additional Technical Data

Ethernet Interface	RJ45
Total length	max. 100m per segment
Online access	
Test/Diagnosis	http server integrated that graphically displays the configuration via website and supports parameterization and project engineering options for test purposes.
Project engineering	Via WinNCS with online coupler search and engineering

Chapter 3 Deployment

Overview

Content of this chapter is the description of the Ethernet coupler IM 253NET from VIPA. It contains all information for commissioning of the Ethernet coupler.

The chapter starts with the basics. Here the basic expressions of the Ethernet communication are explained together with the guidelines for building up a network and the access to the Ethernet coupler.

The chapter ends with the used protocols and a sample for socket programming.

Contents

Overview

Typical fieldbus systems are divided into master and slave systems.

Master system are CPs, coupled to a CPU, allowing remote programming res. visualization of the according CPU as well as the data transfer between several TCP/IP participants.

Slave systems on the other hand are "data collectors" that deliver the I/O data of the connected modules to the requesting master.

The Ethernet coupler described in this chapter is a slave system.

For the communication happens via TCP/IP, the slave system is referred to as server and a master as client.

The Ethernet coupler from VIPA allows you to connect up to 32 modules of your System 200V periphery via Ethernet. With each protocol up to 8 clients may communicate simultaneously with the Ethernet coupler.

Basics

Ethernet

Originally, Ethernet has been developed from DEC, Intel and Xerox (as DIX standard) for the data transfer between office devices. Nowadays it normally means the specification *IEEE 802.3 CSMA/CD*, first published in 1985. Due to the worldwide deployment and the high lot sizes, this technology is commonly available and reasonably priced. This allows the easy link-up to existing networks.

Ethernet transports Ethernet packages from one sender to one ore more receivers. This transfer happens without acknowledgement and without repetition of lost packages. For a secure data transfer, protocols like TCP/IP are used that are accompanying Ethernet.

Twisted Pair

In the early days of networking the Triaxial- (yellow cable) or thin Ethernet cable (Cheapernet) was used as communication medium. This has been superseded by the twisted pair network cable due to its immunity to interference. The IM 253NET Ethernet coupler has a twisted-pair connector.

Where the coaxial Ethernet networks are based on a bus topology the twisted pair network is based on a point-to-point scheme.

The network that may be established by means of this cable has a star topology. Every station is connected to the hub/switch by means of a separate cable. The hub/switch provides the interface to the Ethernet.

Hub

The hub is the central element that is required to implement a twisted pair Ethernet network. It is the job of the hub to regenerate and to amplify the signals in both directions. At the same time it must have the facility to detect and process segment wide collisions and to relay this information. The hub is not accessible by means of a separate network address since it is not visible to the stations on the network. A hub has provisions to interface with Ethernet or another hub.

Switch

A switch also is a central element for implementing a twisted pair Ethernet network. Several station res. hubs are connected together via a switch. These then may communicate with each other via the switch without causing network load. An intelligent hardware analyses the incoming telegrams for every port of the switch and passes them collision free on to the destination stations at the switch. A switch optimizes the band width of every connected segment of a network. Switches allow changing exclusive connections between the connected segment of a network.

Access control

Ethernet supports the principle of random bus access: every station on the network accesses the bus independently as and when required. These accesses are coordinated by a CSMA/CD (Carrier Sense Multiple Access/Collision Detection) scheme: every station "listens" on the bus cable and receives communication messages that are addressed to it.

Stations only initiate a transmission when the line is unoccupied. In the event that two participants should start transmitting simultaneously, they will detect this and stop transmitting to restart after a random delay time has expired.

Communication

The Ethernet coupler is connected with the modules via the backplane bus. It collects their data and places this as "server" (slave) at the disposal of the superordinated "client" (master system).

The communication happens via TCP/IP with leading ModbusTCP or Siemens S5 header protocol.

Vice versa, the Ethernet coupler receives the data, addressed to it by IP address and port, and transfers it to its output periphery. For project engineering, VIPA offers the configuration tool WinNCS that allows you to configure the Ethernet coupler online.

For test and diagnostic purposes the Ethernet slave provides a web server that allows the read and write access to the I/O periphery as well as the parameterization of the modules.

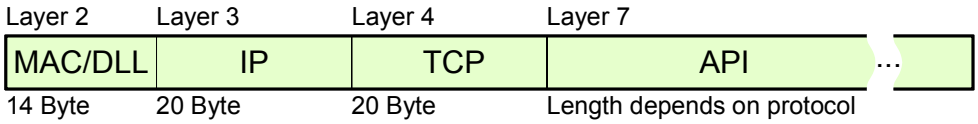
Overview Protocols

Protocols define rules or standards that enables different computers to establish communication connections and exchange data as error free as possible.

The so called ISO/OSI layer model is generally accepted for the standardization of computer communication. The layer model is based upon seven layers with guidelines for the deployment of hard- and software.

Layer	Function	Protocol
Layer 7	Application Layer (Application)	Siemens S5 Header, ModbusTCP
Layer 6	Presentation Layer (Presentation)	
Layer 5	Session Layer (Session)	
Layer 4	Transport Layer (Transport)	TCP
Layer 3	Network Layer (Network)	IP
Layer 2	Data Link Layer (Security)	
Layer 1	Physical Layer (Bit transfer)	

Telegram structure



MAC/DLL

While the Ethernet physics covers with its normed signal levels Layer 1, MAC/DLL covers the conditions of the security layer (Layer 2). With MAC (**M**edium **A**ccess **C**ontrol) / DLL (**D**ata **L**ink **L**ayer) the communication happens at the lowest Ethernet level using MAC addresses. Every Ethernet communication participant has a MAC address that must be unique at the network.

The deployment of MAC addresses specifies source and destination unambiguously.

IP

The Internet Protocol covers the network layer (layer 3) of the ISO/OSI layer model.

The main purpose of IP is to send data packages from one station to another, passing several other stations. This data packages are referred to as datagrams. The IP does neither serve the according sequence nor the deliverance at the receiver.

For the unambiguous distinction between sender and receiver, 32Bit addresses are used (IP addresses) that are normally written in four octets of each 8Bit, e.g. 172.16.192.11. One octet may represent numbers between 0 and 255.

A part of the address specifies the network, the rest identifies the single stations in the network. The proportions of network part and station part is floating and depends on the network size.

TCP

The TCP (**T**ransmission **C**ontrol **P**rotocol) puts directly upon the IP and covers therefore the transport layer (layer 4) of the ISO/OSI layer model. TCP is a connection orientated end-to-end protocol and serves the logical connection between two partners.

TCP ensures the sequential correct and reliable data transfer.

Every datagram is preceded by a header of at least 20 octets that contains, among others, the serial number for the according sequence. This causes that within a network, the single datagrams may reach their destination on different ways.

API

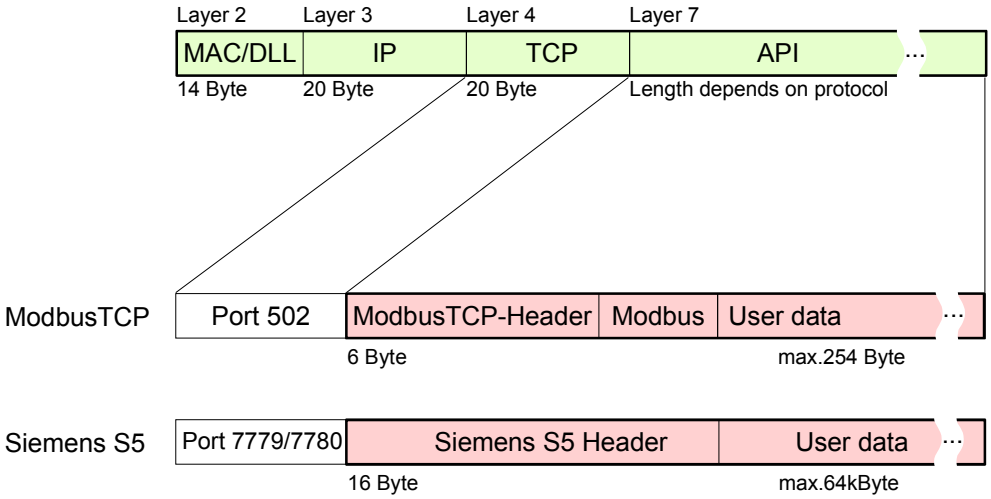
API means **A**pplication **P**rogramming **I**nterface. API covers the conditions of the Application Layer (Layer 7).

Here, the header and user data of the according protocols are stored.

The Ethernet coupler IM 253NET from VIPA uses the following protocols, described further below:

- ModbusTCP
- Siemens S5 Header

API structure



ModbusTCP

ModbusTCP is a Modbus-RTU protocol, put upon TCP/IP.

The Modbus protocol is a communication protocol supporting a hierarchic structure with one master and several slaves. ModbusTCP extends Modbus to a client server communication where several client may access a server.

For the addressing happens by means of the IP addresses, the address integrated in the Modbus telegram irrelevant. Furthermore, the check sum is not required because the sequence insurance happens via TCP/IP.

After the request of a client, this awaits the answer of the server for a configurable time.

ModbusTCP exclusively uses the RTU format.

Every Byte is transferred as one sign. This enables a higher data pass-through than the Modbus-ASCII format. The RTU time supervision is omitted for the header contains the size of the telegram length to be received.

Data that are transferred via ModbusTCP may contain bit and word information. At bit chains, the highest bit is send first, i.e. in a word it is at the most left position. At words, the highest Byte is send first.

The access to a Modbus slave happens via function codes that are described in detail in this chapter further below.

Siemens S5 Header

The Siemens S5 Header protocol serves the data transfer between PLC systems. Deploying the organization format (short ORG) integrated in the Siemens S5 Header protocol, a short description of a data source res. data destination in PLC environment is possible.

The possible ORG formats are corresponding to Siemens.

Planning a network

General

The main characteristic of a bus structure is the existence of a single physical transfer line. As physical transfer mediums are used:

- one or more electrical cables (drilled cable)
- coaxial cable (Triaxial cable)
- fiber optic transmitter.

To enable the communication between the single stations, rules and instructions have to be arranged and kept.

The appointments cover the form of the data protocol, the access procedure to the bus and more basics for communication. The Ethernet coupler IM 253NET from VIPA has been developed upon the ISO standards and norms.

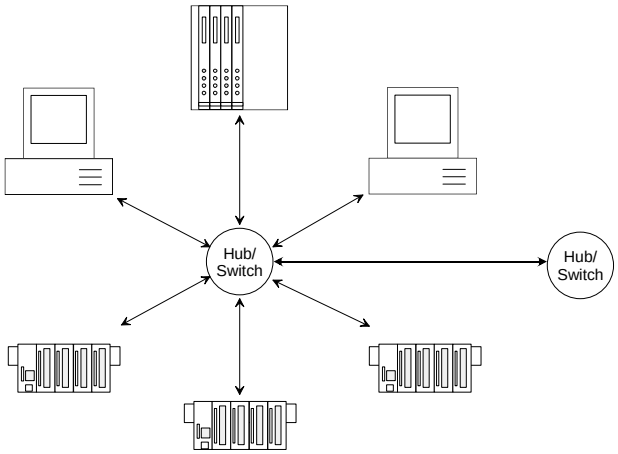
Standards and norms

The following standards and norms about network technologies have been fixed by international and national committees:

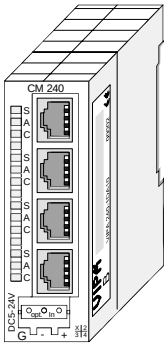
ANSI	American National Standards Institute The ANSI X3T9.5 standard currently defines the provisions for high speed LAN's (100 MB/s) based on fiber optic technology. (FDDI) Fiber Distributed Data Interface.
CCITT	Committee Consultative Internationale de Telephone et Telegraph. Amongst others, this advisory committee has produced the provisions for the connection of industrial networks (MAP) to office networks (TOP) on Wide Area networks (WAN).
ECMA	European Computer Manufacturers Association. Has produced various MAP and TOP standards.
EIA	Electrical Industries Association (USA) This committee has issued standard definitions like RS-232 (V.24) and RS-511.
IEC	International Electrotechnical Commission. Defines certain special standards, e.g. for the Field bus.
ISO	International Organization for Standardization. This association of national standards organizations developed the OSI-model (ISO/TC97/SC16). It provides the framework for the standardization of data communications. ISO standards are included in different national standards like for example UL and DIN.
IEEE	Institute of Electrical and Electronic Engineers (USA). The project-group 802 determines LAN-standards for transfer rates of 1 to 20 MB/s. IEEE standards often form the basis for ISO-standards, e.g. IEEE 802.3 = ISO 8802.3.

Overview components

A twisted pair network can only be constructed with a star topology.



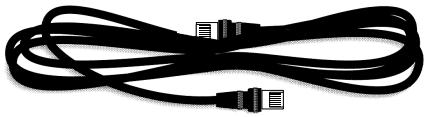
Mini-Switch CM 240



Twisted Pair Cable

A Twisted Pair cable is a cable with four cores drilled in pairs.

The single cores have a diameter of 0.4 to 0.6mm.



Restrictions

This is a summary of the restrictions and rules referring to Twisted Pair:

- Maximum number of coupler elements per segment 2
- Maximum length of a segment 100m

Analyzing the requirements

- What is the size of the area that must be served by the network?
- How many network segments provide the best solution for the physical (space, interference related) conditions encountered on site?
- How many network stations (SPS, IPC, PC, transceiver, bridges if required) must be connected to the cable?
- What is the distance between the different stations on the network?
- What is the expected ”growth rate” and the expected number of connections that must be catered for by the system?
- What is the expected data amount (Band width, accesses/sec.)?

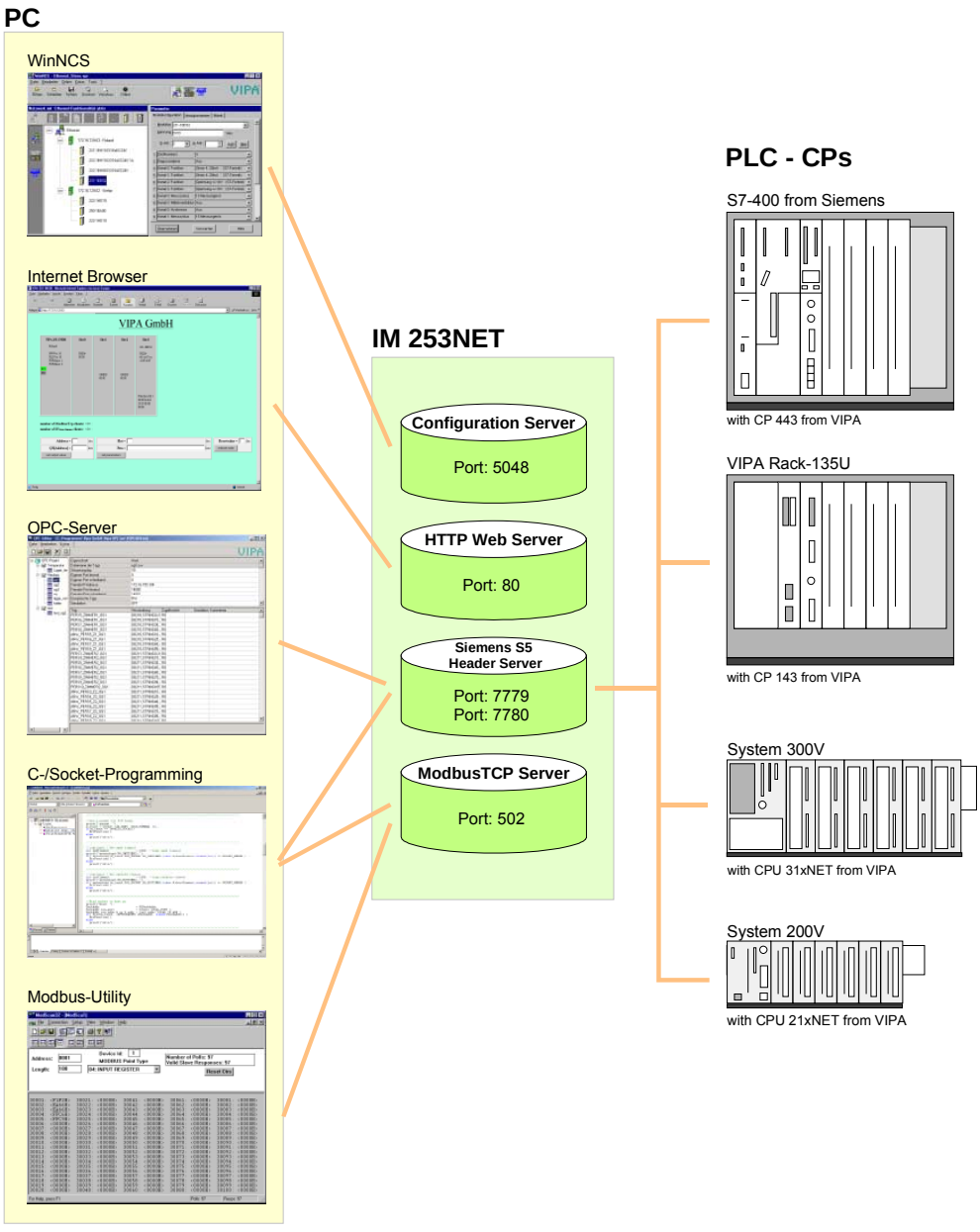
Drawing a network diagram

- Draw a diagram of the network. Identify every hardware item (i.e. station cable, Hub, switch). Observe the applicable rules and restrictions.
- Measure the distance between all components to ensure that the maximum length is not exceeded.

Access to the Ethernet coupler

Overview

The following illustration shows the Ethernet coupler IM 253NET access possibilities.



Access from PC

WinNCS for project engineering

The access happens via Port 5048 on the configuration server.

The configuration server calculates the number of plugged modules, their address and parameter ranges and puts the information under its IP address at the disposal of WinNCS.

WinNCS searches all couplers of the network via broadcast (slaves). The network to search is here until the gateway.

The collected data is used by WinNCS to model a symbolic network and is monitored in the network window.

Now you may assign real module types to the symbolic network and parameterize them.

Now you can assign an IP address to the Ethernet coupler online and update the firmware.

In WinNCS you also define the http web server properties of the Ethernet coupler.

All changing accesses are password protected. The password is requested once per session and slave.

In delivery state the password is `00000000`

Note!

Before you may access the Ethernet slave via internet browser, you have to assign an IP address according to your network. This may happen online via WinNCS.

Internet Browser for diagnosis and test

The access is via Port 80 at the HTTP web server.

The http server transfers a dynamically built web site that shows the recent configuration of the Ethernet coupler.

Besides of the firmware version and RDY/ERR-LED state, the I/O states and the parameters of the modules are shown.

The website also gives you the opportunity to send your alterations online, like accessing module outputs, change the parameters and initialize a re-boot of the Ethernet coupler.

OPC server for data transfer between coupler and PC

The access happens via the ports 7779 and 7780 on the Siemens S5 Header Server. Via these ports, fetch and write accesses via the VIPA OPC server are enabled.

The VIPA OPC server is a comfortable tool for visualization and data transfer.

C-/Socket programming for data transfer between coupler and PC

At ModbusTCP, the access is via port 502 at the ModbusTCP server and at Siemens S5 header via the ports 7779 and 7780 on the Siemens S5 Header Server.

This possibility of data transfer is for C program developers who want to create an open interface by means of socket programming.

Via simple C programs it is possible to transfer data between PC and Ethernet coupler. Depending on the program, the data is transferred via ModbusTCP or via Siemens S5 Header.

More detailed information about programming with sample sources is to find further below in this chapter.

Modbus utility

The access is via port 502 at the ModbusTCP Server. Modbus utility means all tools and programs that have a ModbusTCP interface.

For example, you may find the demo tool "ModbusScan32" from WinTech for download under www.win-tech.com.

Access from SPS res. CP

Data transfer between coupler and CP via Siemens S5 Header

The access happens via the ports 7779 and 7780 on the Siemens S5 Header Server. Via this ports, the VIPA CP, OPC server or other devices have fetch and write access.

For the communication, you need a PLC program in the CPU that serves the in-/output areas of the CP. Herefore, you have to configure fetch/write connections at the CP.

Principle of the automatic address allocation

Automatic addressing

To individually call the connected peripheral modules, certain addresses in the Ethernet coupler have to be assigned to them. For input and output area, the Ethernet coupler has an address range of each 256Byte.

The address allocation (also called Mapping) happens automatically and may not be influenced. The mapping may be seen via the website of the coupler.

In addition to the alarm processing behind the 256Byte I/O data, the "alarm information image" stored in size 520Byte.

Rules

At boot-up, the Ethernet coupler assigns automatically addresses for its in-/output periphery following this rules:

- All modules are mapped from left (Ethernet coupler) to right in ascending sequence starting with address 0.
- It is separated between in- and output area (if a module has in- and output data, these are stored at different addresses).
- There is no separation between digital and analog data. The Ethernet coupler creates cohere areas for in- and output data.

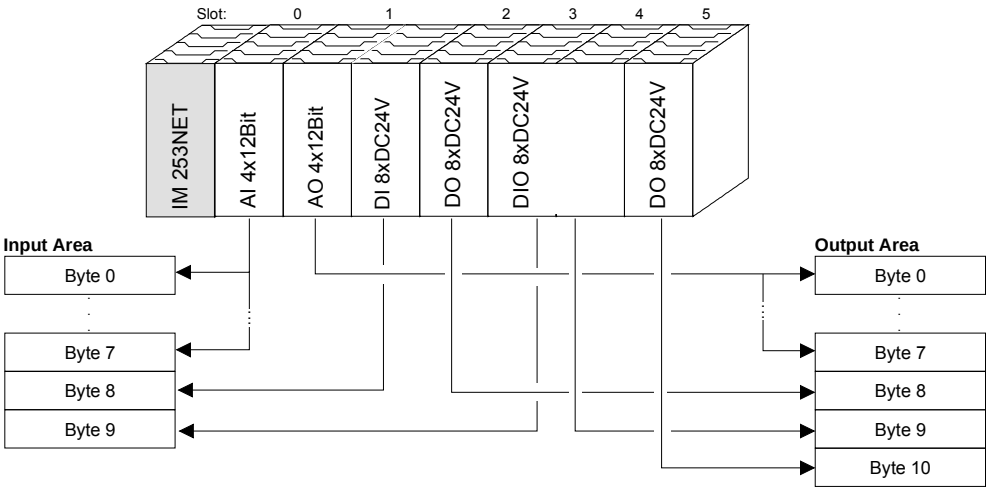
Note!

A description of the in-/output areas that are occupied by a module is to find in the concerning module description.

Please regard that modules that are occupying more than 1Byte like e.g. analog modules, are stored starting with an even address. Otherwise ModbusTCP has problems with word accesses.

Sample for the automatic address allocation

The following picture illustrates the automatic address allocation:



Project engineering under WinNCS

Preconditions

The project engineering happens via WinNCS starting with V3.09. For project engineering , the following preconditions should be met:

- Recent VIPA_ETH200V.GSD is stored in WinNCS/GSD/Englisch.

  For project engineering of the System 200V modules in WinNCS you receive the features of the VIPA components with a GSD-file.

  The GSD-file for the IM 253NET Ethernet coupler from VIPA is: VIPA_ETH200V.GSD

  Copy this GSD-file into WinNCS/GSD/Englisch.

  The latest version is to find under ftp.vipa.de/support.
- For online project engineering, the IM 253NET should be assembled with the according modules, connected to the Ethernet and supplied with voltage.

Attention!

For every Ethernet slave is delivered with the IP address 10.0.0.1, you must not install more than one new Ethernet slave at a time!

Approach online project engineering

- Start WinNCS and create a new "Ethernet" project via **File** > *Create/Open project*.
  → A parameter windows for online search of "Slaves" and "Stations" opens. [Slaves] lists all Ethernet coupler and [Stations] all CPs.
- Click at [Slaves]
  → All Ethernet coupler are searched and listed with IP address and where applicable with label.
- Via double-click at a listed slave, this is overtaken into the network window and listed with the concerning I/O periphery.
  → If there is no parameterization yet, the modules are listed as symbol (without label).
- Now you assign the according module type to the listed module symbol in the parameter window and adjust the parameters when needed. The address range that is occupied by the module in the TCP data stream is automatically preset by the Ethernet coupler.
- As soon as you click at [apply], you have to type the password. The password request happens once per session and coupler. In delivery state, the password is **00000000**.
  With correct password, the data is transferred online to the Ethernet coupler. Repeat this for all listed modules.
- Save your project.

Diagnosis and test via Internet Browser

Addressing

Type the configured IP address of your Ethernet coupler into your Internet Browser. Now you have access to a dynamically built-up website of the HTTP server.

Please regard that the website always contains the information of the last update.

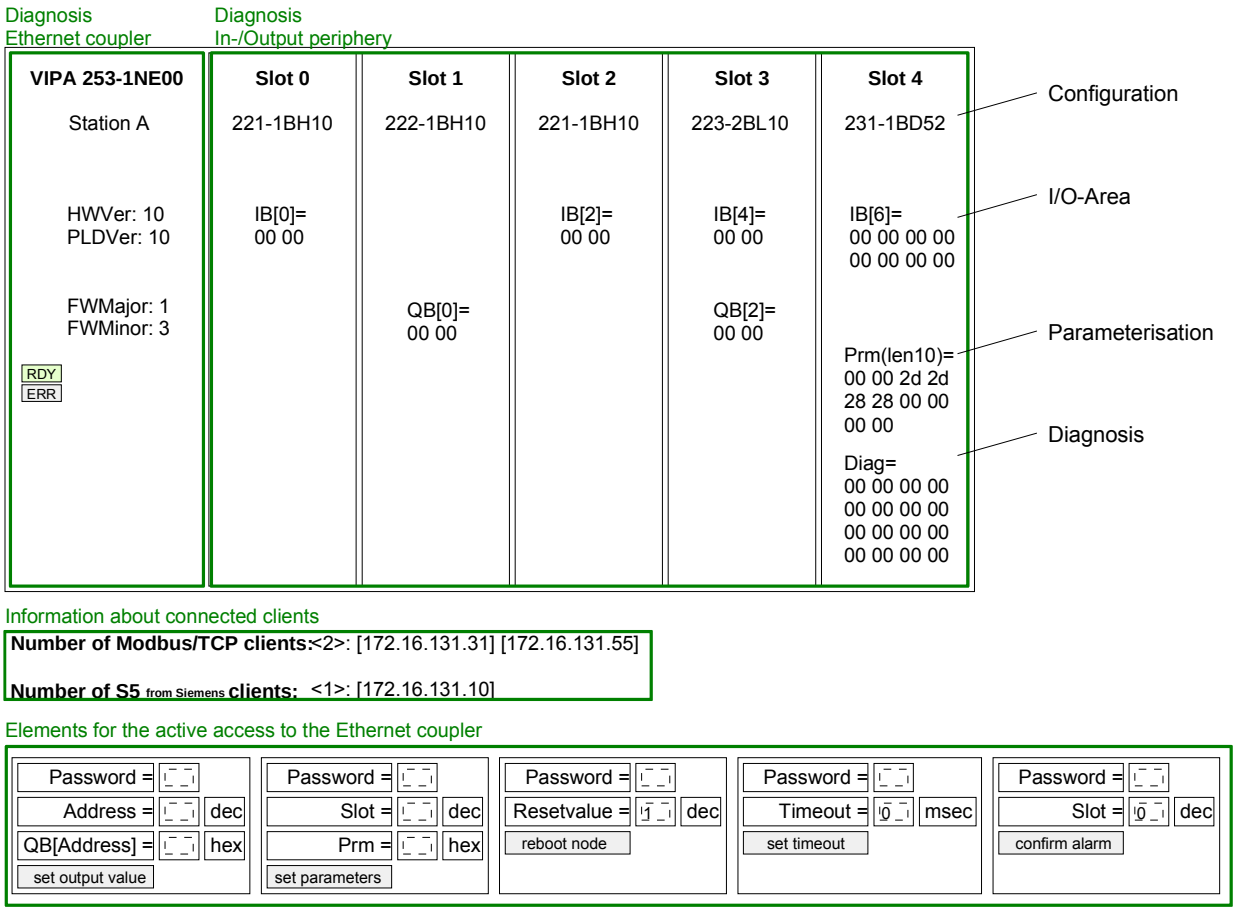
For an update, click at home in the lower left corner of the website.

Structure of Website

The website is dynamically built-up and depends on the number of the modules connected to the Ethernet coupler. The access rights to this website are in WinNCS freely configurable.

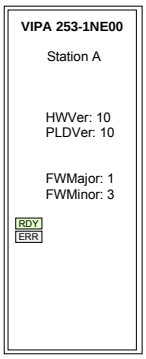
The following elements are to find on the website:

- Diagnosis Ethernet coupler
- Parameterization and diagnosis data in-/output periphery
- Information about connected clients
- Elements for active access to the Ethernet coupler



home

Diagnosis Ethernet coupler



This area shows all information about the Ethernet coupler like symbolic name, version and status monitors of the LEDs.

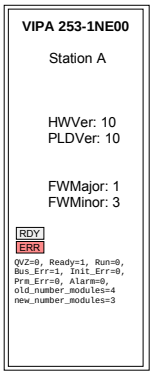
Symbolic name: Via WinNCS you may assign a symbolic name to the Ethernet coupler besides the IP address.

HWVer: This is the hardware version (electronics). The HW release (only number before comma) is also at the front side of the module.

PLDVer: The PLD (**P**rogrammable **L**ogic **D**evice) is a programmable logic block for control of the communication between backplane bus and processor.

FWMajor, FWMinor: The firmware version is divided into *FWMajor* (main version) and *FWMinor* (lower version). A lower version contains small alterations. When basic alterations are made, the main version number is increased.

Status monitor



RDY, ERR: Status monitor of the LEDs RD and ER

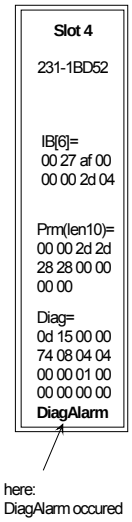
rdy (small letter): LED is blinking / RDY (capital letter): LED is on

As long as the Ethernet coupler communicates error free, the status monitor remains like shown above. In case of an error, e.g. the following message is displayed below ERR:

```
QVZ=0 Ready=1, Run=0, Bus_Err=1, Init_Err=0, Prm_Err=0, Alarm=0
old_number_modules=4, new_number_modules=3
```

This message shows that one module is defect.

Module area Slot 0 ... 31



This area shows all information about the in-/output periphery like module name, in-/output assignment, if existing parameter bytes and diagnosis data.

Module name: The order number of the module serves as module name. This allows an unambiguous identification of the module.

In-/output assignment: Here you find four information:

- Type: input area (IB), output area (QB)
- The start address of the area is in brackets
- You see the number of bytes occupied by the module
- The content of the bytes corresponds to that of the Ethernet coupler at the last website update

Example: `Slot 4`

```
IB[6]=
00 00 00 00
00 00 00 00
```

This means: the module at slot 4 has in the input area at byte 6 8 byte hexadecimal content.

The image will be little endian (Intel) output format (low byte, high byte).

The `Prm()`= Parameter bytes contain the following information:

- The length of the parameter block is in brackets with a preceding `len`.
- The content of the bytes are the parameter bytes of the according module.

`DIAG` = showes 16byte diagnosis data for the alarm handling.

Information about connected clients

This area gives you information about number and IP address of the clients that are communicating with the Ethernet coupler at the time via ModbusTCP res. Siemens S5 Header protocol. With every protocol, a max. of 8 clients may communicate simultaneously with the Ethernet slave.
The number is in <> followed by the IP addresses in [].

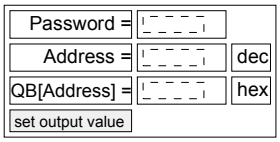
Example:
Number of ModbusTCP clients: <2>: [172.16.131.20] [172.16.140.63]
(At this time, 2 clients are communicating via ModbusTCP with the IP addresses 172.16.131.20 and 172.16.140.63.)

Elements for the active access...

Whereas the elements above are displaying information, the active access elements here allow to access the Ethernet coupler and its modules online.

Every control element is password protected. Use the password configured for your coupler (default = "00000000").
The following 5 control elements are available:

- Control outputs
- Parameterize module
- Reset the Ethernet coupler
- Configure Timeout
- Confirm alarm



Control outputs

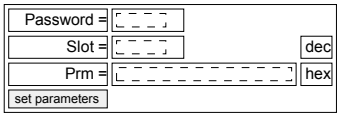
This control element allows you to set values into a wanted address area and transfer them via [set output value] to the Ethernet coupler.

Please regard that the address has to be a decimal number and the value a Hex number. You may transfer a max. of 4Byte to the address given in `Address`.

Please regard that the Bytes always have to be transferred with a leading zero. Space signs are serving as Byte separator.

Example: Address=0

```
QB[Address]= 12   →    QB[0]= 12 00
QB[Address]= 1 2  →    QB[0]= 01 02
QB[Address]= 1234 →    QB[0]= 12 34
QB[Address]= 123  →    QB[0]= 01 23
```



Parameterize module

This control element allows you to provide the module online with parameters by typing the parameter bytes into `Prm` and setting a plug-in location via `Slot`.

With [set parameters], the according parameters are transferred to the according module.

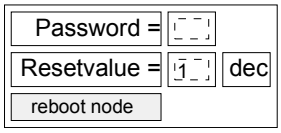
Please regard that the slot number has to be a decimal number and the parameter a Hex number.

Bytes are always transferred with a leading zero. A blank must be inserted as separator.

Note!

Always transfer the complete number of parameter bytes to a module, otherwise errors at the module may occur.

The number of parameters and their assignment is to find in the description of the concerning module.



Reset of the Ethernet coupler

Via [reboot node] a reset of the Ethernet coupler is initialized. After a re-boot, you have to update the website via *home*.

By presetting a r*eset value*, you may additionally to the re-boot of the Ethernet coupler delete the configuration or module parameters.

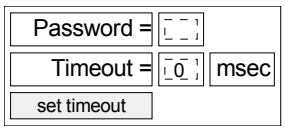
Permissible r*eset value* values are 1, 2 or 3. Other values are ignored!

Reset value= 1	Re-boot of the coupler (default setting)
Reset value= 2	Delete all module configurations (module names) and re-boot the coupler
Reset value= 3	Delete all module parameters and re-boot the coupler
Reset value= 4	Reset password (Default-Value "00000000")

Reset password

Resetting the password to its default value "00000000" is possible in a special operation mode.

- Power off the Ethernet coupler and pull it off from back plane bus.
- Power up the Ethernet coupler again
- Open the Web site of the Ethernet coupler by means of you Web browser and the IP address.
- Type in the default Password "00000000" at the parameter "reboot node"
- Set "Resetvalue =" 4 and click "reboot node".
  → Now the coupler resets its password to default value "00000000" and boots up again.



Configure Timeout

The coupler offers a connection timeout.

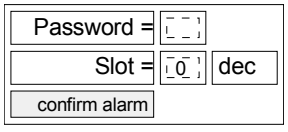
If the value 0 is transferred, this function is deactivated. (In the picture of the Ethernet coupler "Timout: off" is shown).

Note!

Choose "Timout: off" if you want to control outputs via internet browser otherwise all outputs are set to the secure state 0 after timeout.

With timeout values > 0msec, an I/O connection must read/write faster than the time value. If not, the connections are terminated and the outputs are set to the secure state 0.

The RD LED blinks and the website shows "rdy" in small letters.



Confirm alarm

Using "confirm alarm" you can acknowledge the alarm of a module.
By specifying the slot for the appropriate module with [confirm alarm] the alarm status of the module is reset.

ModbusTCP

General

ModbusTCP is a Modbus protocol put upon TCP/IP, where the IP address serves the addressing. The ModbusTCP allows a client-server-communication, several clients may be provided from one server.

Telegram structure incl. TCP/IP

The request telegrams sent by a master and the respond telegrams of the slave have the same structure:

ModbusTCP	Slave address	Function code	Data
6Byte-Header with number of following Bytes	1Byte data	1Byte data	max 254Byte

ModbusTCP-Header (6Byte)

For send and receive telegrams, ModbusTCP uses a header of 6Byte with the following structure:

ModbusTCP header

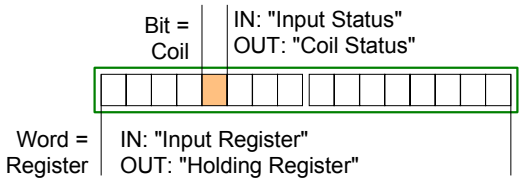
Byte	Name	Description
0	Transaction identifier (High-Byte)	Is sent back by the server (user-defined)
1	Transaction identifier (Low-Byte)	Is sent back by the server (user-defined)
2	Protocol identifier (High-Byte)	Always 0
3	Protocol identifier (Low-Byte)	Always 0
4	Length field (High-Byte)	Always 0 because messages < 256Byte
5	Length field (Low-Byte)	Number of following bytes

Normally, Byte 0 ... 4 have the value 0. You may also increase Byte 0 and 1 in the slave and thus establish an additional control.

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 under Modbus happens by means of the ranges 0x, 1x, 3x and 4x.

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

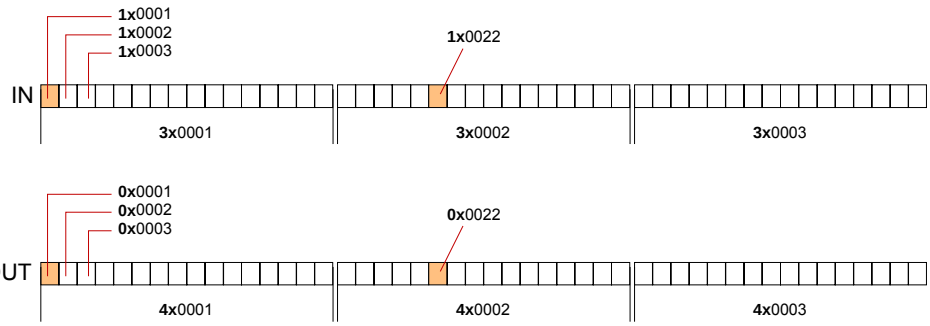
For the Ethernet coupler from VIPA is not differentiating digital and analog data, the following assignment is valid:

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

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

3x: word area for master input
Access via function code 04h, 17h

4x: word area for master output
Access via function code 03h, 06h, 10h, 17h



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 one bit	Write 1 bit to master output area 0x
06h	Write one word	Write 1 word to master output area 4x
0Fh	Write n bits	Write n bits to master area 0x
10h	Write n words	Write n words to master area 4x
17h	Write n words and Read m words	Write n words into master output area 4x and the respond contains m read words of the master input area 3x

The Ethernet coupler from VIPA does not differentiate between digital and analog data!

Note!

The byte sequence in a word always is:

1 word	
High byte	Low byte

Respond of the coupler

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

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

Read n bits 01h, 02h

This function enables the reading from a slave bit by bit.

Command telegram

ModbusTCP-Header						Slave address	Function code	Address 1. bit	Number of bits
x	x	0	0	0	6				
6byte						1byte	1byte	1word	1word

Respond telegram

ModbusTCP-Header						Slave address	Function code	Number of read bytes	Data 1. byte	Data 2. byte	...
x	x	0	0	0							
6byte						1byte max. 255byte	1byte	1byte	1byte	1byte max. 252byte	

Read n words 03h, 04h This function enables the reading from a coupler word by word.

Command telegram

ModbusTCP-Header						Slave address	Function code	Address word	Number of words
x	x	0	0	0	6				
6byte						1byte	1byte	1word	1word

Respond telegram

ModbusTCP-Header						Slave address	Function code	Number of read bytes	Data 1. word	Data 2. word	...
x	x	0	0	0							
6byte						1byte max. 255byte	1byte	1byte	1word	1word max. 126words	

Write a bit 05h This function allows to alter a bit in your coupler. A status change happens via "Status bit" with the following values:

"Status bit" = 0000h → bit = 0, " Status bit" = FF00h → bit = 1

Command telegram

ModbusTCP-Header						Slave address	Function code	Address bit	Status bit
x	x	0	0	0	6				
6byte						1byte	1byte	1word	1word

Respond telegram

ModbusTCP-Header						Slave address	Function code	Address bit	Status bit
x	x	0	0	0	6				
6byte						1byte	1byte	1word	1word

Write a word 06h

This function sends a word to the coupler. This allows to overwrite a register in the coupler.

Command telegram

ModbusTCP-Header	Slave address	Function code	Address word	Value word
x x 0 0 0 6				
6byte	1byte	1byte	1word	1word

Respond telegram

ModbusTCP-Header	Slave address	Function code	Address word	Value word
x x 0 0 0 6				
6byte	1byte	1byte	1word	1word

Write n bits 0Fh

This function writes n bits to the slave. Please regard that the number of bits has additionally given in byte.

Command telegram

ModbusTCP-Header	Slave address	Function code	Address 1. bit	Number of bits	Number of bytes	Data 1. byte	Data 2. byte	...
x x 0 0 0								
	1byte	1byte	1word	1word	1byte	1byte	1byte	1byte
	max. 255byte					max. 248byte		

Respond telegram

ModbusTCP-Header	Slave address	Function code	Address 1. bit	Number of bits
x x 0 0 0 6				
	1byte	1byte	1word	1word

Write n words 10h Via this function you may write n words to the slave.

Command telegram

ModbusTCP-Header	Slave address	Function code	Address 1. word	Number of words	Number of bytes	Data 1. word	Data 2. word	...
x x 0 0 0 .								
	1byte	1byte	1word	1word	1byte	1word	1word	1word
	max. 255byte					max. 124words		

Respond telegram

ModbusTCP-Header	Slave address	Function code	Address 1. word	Number of words
x x 0 0 0 6				
	1byte	1byte	1word	1word

Write n words and Read m words 17h This function allows to write n words and read m words with a request.

Command telegram

ModbusTCP-Header	Slave address	Functions code	Read address	Read number of words	Write address	Write No. of words	Write No. of bytes	Write Data 1. word	Write Data 2. word	...
x x 0 0 0 .										
	1byte	1byte	1word	1word	1word	1word	1byte	1word	1word	
	max. 255byte							max. 122words		

Respond telegram

ModbusTCP-Header	Slave address	Functions code	Read number of bytes	Read Data 1. word	Read Data 2. word	...
x x 0 0 0 .						
6byte	1byte	1byte	1byte	1word	1word	
	max. 255byte				max. 126words	

Siemens S5 Header Protocol

General

The Siemens S5 Header protocol serves the data exchange between PLC systems. Deploying the organization format (short ORG) that is included in the Siemens S5 Header protocol, a short description of a data source res. destination in PLC environment is possible.

ORG formats

The used ORG formats are corresponding to the Siemens specifications and are listed in the following table.

The ORG block is optional at READ and WRITE.

The ERW specification is irrelevant for the Ethernet coupler.

The start address and the number are addressing the memory area and are stored in HIGH-/LOW format (Motorola – Address format)

Description	Type	Area
ORG specification	BYTE	1...x
ERW specification	BYTE	irrelevant
Start address	HILOWORD	0...y
Number	HILOWORD	1...z

The following table lists the useable ORG formats.
The "length" may not be specified as -1 (FFFFh).

ORG specification *02h-05h*

CPU area	MB	EB	AB	PB
ORG specification	02h	03h	04h	05h
Description	Only permitted: Read MB0 with length 4. The total length of the in- and output areas is calculated and stored in MB0 ... MB3 in this format: MB0: Length In area MB1: 00 MB2: Length Out area MB3: 00	Source/destination data out/in Process image inputs (PAE).	Source/destination data out/in Process image outputs (PAA).	Source/destination data out/in peripheral module At source data input modules, at destination data output modules.
DBNR		irrelevant	irrelevant	irrelevant
Start address Meaning Permitted range:		EB-No. from where on the data is fetched resp. written. 0 ... 255	AB-No. from where on the data is fetched resp. written. 0 ... 255	PB-No. from where on the data is fetched resp. written. 0 ... 65535
Number Meaning Permitted range:		Length of the source/destination data block in Bytes. 1 ... 256	Length of the source/destination data block in Bytes. 1 ... 256	Length of the source/destination data block in Bytes. 1 ... 256

Structure PLC header

At READ and WRITE acknowledgement telegrams are created by the Ethernet coupler and request telegrams are expected with the format shown below. The headers have normally a length of 16Byte and have the following structure:

at WRITE

	Client (PLC, PC)	Server (Ethernet slave)
	Request telegram	Acknowledgement telegram
System spec.	="S"	="S"
	="5"	="5"
Length Header	=16d	=16d
Spec. OP-Code	=01	=01
Length OP-Code	=03	=03
OP-Code	**=03**	**=04**
ORG-Block	=03	Ackn. block =0Fh
Length ORG-Block	=08	Length ackn. Block =03
ORG specification		Error No. =Nr.
DBNR		Empty block =FFh
Start address	H	Length empty block =07
	L	
Length	H	free
	L	
Empty block	=FFh	
Length	=02	
Data up to 64K but only if error no. =0		

at READ

	Request telegram	Acknowledgement telegram
System spec.	="S"	="S"
	="5"	="5"
Length Header	=16d	=16d
Spec. OP-Code	=01	=01
Length OP-Code	=03	=03
OP-Code	**=05**	**=06**
ORG-Block	=03	Ackn. block =0Fh
Length ORG-Block	=08	Length ackn. Block =03
ORG specification		Error No. =Nr.
DBNR		Empty block =FFh
Start address	H	Length empty block =07
	L	
Length	H	free
	L	
Empty block	=FFh	
Length	=02	
		Data up to 64K but only if error no. =0

Possible error numbers

The following error numbers may be included in the acknowledgement telegram:

0: no error

3: Address outside the defines area

6: No valid ORG format (Specification data source/destination is wrong). Permitted: IB, QB, PIB and MB

Principle of Alarm handling

Overview

Many of the System 200V modules are able to set an alarm (all non digital modules, e.g. analog modules, function modules, fieldbus masters).

As soon as one or more modules report an alarm, the alarm data of the appropriate slot location is received and acknowledged by the Ethernet coupler.

After that the slot location assigned bit of the internal *alarm information image* is set and the diagnostic data with the length of 16Byte is stored.

In system 200V we distinguish between two types of alarms: the *process alarm* and the *diagnosis alarm.* A module will set only one of the alarm types at a time.

For differentiation, the alarm information image contains one 32bit wide field (bit 0 = slot 0 up to bit 31 = slot 31) called process alarm status and one 32bit wide field called diagnosis alarm status. After that, there follows for each slot 0 ... 31 a 16byte wide field called alarmdata.

For acknowledgement you can also access diagnostic and process alarm status writing. The 16byte alarmdata is read only.

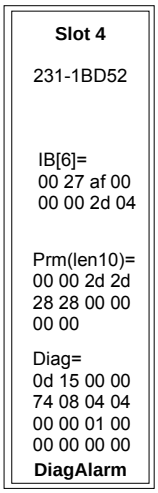
alarm information image

The *alarm information image* with a size of 520Byte is placed behind the 256Byte I/O data and has the following structure:

32Bit process alarm status (little endian format): Bit 0 = slot 0 ... Bit 31 = slot 31
32Bit diagnosis alarm status (little endian format): Bit 0 = slot 0 ... Bit 31 = slot 31
16Byte alarmdata of slot 0
16Byte alarmdata of slot 1
... etc.
16Byte alarmdata of slot 31

Output Diagnosis

Web-Server



All alarm capable modules feature the entry "Diag=" with the latest 16byte of alarmdata. With an alarm set, the message "DiagAlarm" for diagnosis alarm resp. "ProcAlarm" for process alarm is displayed.

ModbusTCP

Read starting at register 3x0129:

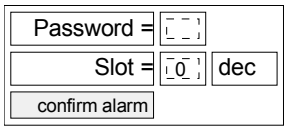
Register	Content
3x0129	process alarm status: Byte 0, Byte 1
3x0130	process alarm status: Byte 2, Byte 3
3x0131	diagnosis alarm status: Byte 0, Byte 1
3x0132	diagnosis alarm status: Byte 2, Byte 3
3x0133	Slot 0: alarmdata 16Byte
3x0141	Slot 1: alarmdata 16Byte
3x0149	Slot 2: alarmdata 16Byte
3x0157	Slot 3: alarmdata 16Byte
3x0165	Slot 4: alarmdata 16Byte
3x0173	Slot 5: alarmdata 16Byte
3x0181	Slot 6: alarmdata 16Byte
3x0189	Slot 7: alarmdata 16Byte
3x0197	Slot 8: alarmdata 16Byte
3x0205	Slot 9: alarmdata 16Byte
3x0213	Slot 10: alarmdata 16Byte
3x0221	Slot 11: alarmdata 16Byte
3x0229	Slot 12: alarmdata 16Byte
3x0237	Slot 13: alarmdata 16Byte
3x0245	Slot 14: alarmdata 16Byte
3x0253	Slot 15: alarmdata 16Byte
3x0261	Slot 16: alarmdata 16Byte
3x0269	Slot 17: alarmdata 16Byte
3x0277	Slot 18: alarmdata 16Byte
3x0285	Slot 19: alarmdata 16Byte
3x0293	Slot 20: alarmdata 16Byte
3x0301	Slot 21: alarmdata 16Byte
3x0309	Slot 22: alarmdata 16Byte
3x0317	Slot 23: alarmdata 16Byte
3x0325	Slot 24: alarmdata 16Byte
3x0333	Slot 25: alarmdata 16Byte
3x0341	Slot 26: alarmdata 16Byte
3x0349	Slot 27: alarmdata 16Byte
3x0357	Slot 28: alarmdata 16Byte
3x0365	Slot 29: alarmdata 16Byte
3x0373	Slot 30: alarmdata 16Byte
3x0381	Slot 31: alarmdata 16Byte

Siemens S5 Header

Write starting at periphery byte 256:

Byte address	Content
256	process alarm status: Byte 0, Byte 1
258	process alarm status: Byte 2, Byte 3
260	diagnosis alarm status: Byte 0, Byte 1
262	diagnosis alarm status: Byte 2, Byte 3
264	Slot 0: alarmdata 16Byte
280	Slot 1: alarmdata 16Byte
296	Slot 2: alarmdata 16Byte
312	Slot 3: alarmdata 16Byte
328	Slot 4: alarmdata 16Byte
344	Slot 5: alarmdata 16Byte
360	Slot 6: alarmdata 16Byte
376	Slot 7: alarmdata 16Byte
392	Slot 8: alarmdata 16Byte
408	Slot 9: alarmdata 16Byte
424	Slot 10: alarmdata 16Byte
440	Slot 11: alarmdata 16Byte
456	Slot 12: alarmdata 16Byte
472	Slot 13: alarmdata 16Byte
488	Slot 14: alarmdata 16Byte
504	Slot 15: alarmdata 16Byte
520	Slot 16: alarmdata 16Byte
536	Slot 17: alarmdata 16Byte
552	Slot 18: alarmdata 16Byte
568	Slot 19: alarmdata 16Byte
584	Slot 20: alarmdata 16Byte
600	Slot 21: alarmdata 16Byte
616	Slot 22: alarmdata 16Byte
632	Slot 23: alarmdata 16Byte
648	Slot 24: alarmdata 16Byte
664	Slot 25: alarmdata 16Byte
680	Slot 26: alarmdata 16Byte
696	Slot 27: alarmdata 16Byte
712	Slot 28: alarmdata 16Byte
728	Slot 29: alarmdata 16Byte
744	Slot 30: alarmdata 16Byte
760	Slot 31: alarmdata 16Byte

Confirm alarm *Web-Server*



With WinNCS (Version > V320) it will be possible to activate the web control "confirm alarm". Using that, it is possible to clear a slots alarm status bit. You need to set your couplers password and the slot (0 ... 31) where the alarm status bit shall be confirmed. Then clicking the button [confirm alarm] will clear the status bit and the "DiagAlarm" resp. "ProcAlarm" message should be deleted.

ModbusTCP

Write starting at register 4x0129:

Register	Content
4x0129	process alarm status: Byte 0, Byte 1
4x0130	process alarm status: Byte 2, Byte 3
4x0131	diagnosis alarm status: Byte 0, Byte 1
4x0132	diagnosis alarm status: Byte 2, Byte 3

Siemens S5 Header

Write starting at periphery byte 256:

Byte address	Content
256	process alarm status: Byte 0, Byte 1
258	process alarm status: Byte 2, Byte 3
260	diagnosis alarm status: Byte 0, Byte 1
262	diagnosis alarm status: Byte 2, Byte 3

Typical application A typical application watches the alarm status fields and checks their value. For "0" there is nothing to do, because no alarm has occurred. If process alarm status or diagnosis alarm status are <> "0", there have been found one or more alarms and there are updated alarmdata fields to read. Those should be evaluated (find out about modules/channels state, e.g. wire break) and then the alarm status field should be set to zero. We call that "to confirm alarms". Now continue with watching for alarms (polling).

More than one alarm from different slots:

If there was signaled an alarm from more than one slots at a time, the appropriate alarm status bit will be set to "1" and each corresponding alarmdata field will be updated. So there is no loss of information!

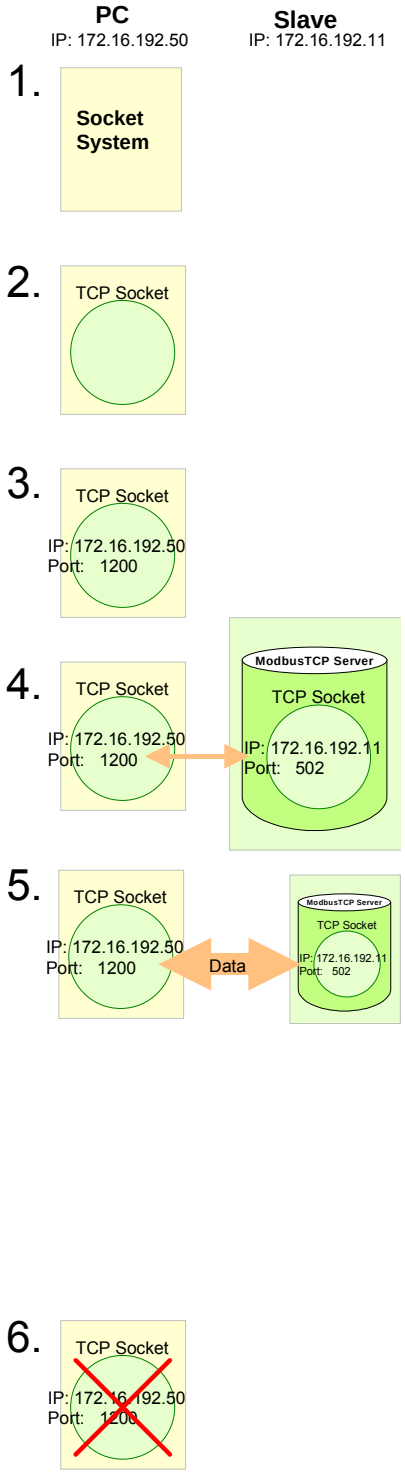
More than one alarm from one slot:

If there came more than one alarms from one slot, the slots alarm status bit will be set and keep on "1" (logical OR). In the corresponding alarmdata field the latest alarmdata may be read. The alarms history and how many alarms occurred is unknown! But at least it is assured that always the current alarm status and alarmdata is available.

Programming sample

Steps of Programming

For the deployment of the Ethernet couplers at a PC you should have a thorough knowledge in C programming, especially in socket programming. This section gives you a short overview about the programming.



to 1. Start Microsoft Socket System

```
WSAStartup (wVersionRequested, &wsaData);
```

to 2. Reserve Socket resources for TCP

```
m_lsock = socket (AF_INET, SOCK_STREAM, 0):
```

to 3. Link-up the socket to the local PC

```
SockAddr.sin_port = htons( 0 );
SockAddr.sin_addr.S_un.S_addr = inet_addr( "0.0.0.0" );
bind(m_lsock, (LPSOCKADDR) &SockAddr, sizeof(SockAddr));
```

By calling `bind` with the value 0 for port and IP address, the socket gets the PC-IP address and the next free Port.
(here: IP: 172.16.192.50, Port: 1200)

to 4. Establish connection to external device

```
SockAddr.sin_port = htons (m_wPort);
SockAddr.sin_addr.S_un.S_addr = inet_addr(m_szIpAddress);
connect(m_lsock, (LPSOCKADDR) &SockAddr, sizeof(SockAddr));
```

to 5. For write res. read access you have to build up telegrams according to the protocol and store them in sndBuf.
sndBufLen contains the number of Bytes to be sent.

Read access

Send sndBuf (Request)

```
send(m_lsock, (char *)sndBuf, sndBufLen, 0);
```

Receive telegram in rcvBuf (Response+data)

```
recv(m_lsock, (char *)rcvBuf, sizeof(rcvBuf), 0);
```

Write access

Send sndBuf (Request+data)

```
send(m_lsock, (char *)sndBuf, sndBufLen, 0);
```

Receive telegram in rcvBuf (Response)

```
recv(m_lsock, (char *)rcvBuf, sizeof(rcvBuf), 0);
```

to 6. Close socket again

```
closesocket(m_lsock);
```