SIEMENS

SIMATIC NET

S7-1500 - TeleControl
TIM 1531 IRC

Operating Instructions

02/2018

C79000-G8976-C468-02

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Division Process Industries and Drives
Postfach 48 48
90026 NÜRNBERG
GERMANY

C79000-G8976-C468-02
℗ 02/2018 Subject to change

Preface

CAUTION

To prevent injury, read the manual before use.

Validity of this manual

This manual is valid for the following product:

TIM 1531 IRC
Article number: 6GK7 543-1MX00-0XE0
Hardware product version 1
Firmware version 1.1

Communications module for SIMATIC S7-1500 / S7-400 / S7-300, SINAUT ST7 protocol, with three RJ-45 interfaces for communication via IP-based networks (WAN / LAN) and an RS-232/RS-485 interface for communication via classic WAN networks

Figure 1 TIM 1531 IRC

You will find the the article number of the device on the top right of the housing.

You will find the hardware product version of the device printed as a placeholder "X" on the right of the device (for example X 2 3 4). In this case, "X" would be the placeholder for hardware product version 1.

You will find the MAC addresses of the three Ethernet interfaces at the front in the middle of the housing.

Purpose of the manual

This manual describes the properties of this device and shows application examples.

The manual supports you when installing, connecting up and commissioning the device.

The required configuration steps for the device are described.

You will also find instructions for operation and information about the diagnostics options of the device.

New in this release

- New firmware version with the following new functions:
  - Use of the TD7 variant "TD7onCPU" via program blocks on the CPU
  - Options for expanding SINAUT projects configured in STEP 7 V5 to STEP 7 Professional
  - PG routing
- Editorial revision

Replaced manual edition

Edition 05/2017

Required experience

To install, commission and operate the device, you require experience in the following areas:

- Setting up industrial networks with security functions
- Data transfer via WAN networks
- SIMATIC STEP 7 Professional

Abbreviations/acronyms

This manual often uses the following abbreviations/acronyms:

- **TIM / submodule / module**

  The names are used instead of the full product name "TIM 1531 IRC" of the device.
- **ST7**

  Short form for the telecontrol protocol "SINAUT ST7"
- **WBM**

  "WBM" is the acronym for the "Web Based Management" of the pages of the TIM Web server for configuration and diagnostics data.

Cross references

In this manual there are often cross references to other sections.

To be able to return to the initial page after jumping to a cross reference, some PDF readers support the command <Alt>+<left arrow>.

Current manual release on the Internet

You will also find the current version of this manual on the Internet pages of Siemens Industry Online Support:

Link: (https://support.industry.siemens.com/cs/ww/en/ps/24710/man)

Sources of information and other documentation

You can find an overview of further reading and references in the Appendix of this manual.

License conditions

Note

Open source software

The product contains open source software. Read the license conditions for open source software carefully before using the product.

You will find the license conditions on the supplied data medium:

- OSS_TIM1531IRC_99.pdf
- OSS_TIM1531IRC-TI_76.pdf

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For additional information on industrial security measures that may be implemented, please visit
Link: (http://www.siemens.com/industrialsecurity)

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customers' exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under
Link: (http://www.siemens.com/industrialsecurity)

Firmware

The firmware is signed and encrypted. This ensures that only firmware created by Siemens can be downloaded to the device.

Device defective

If a fault develops, please send the device to your SIEMENS service center for repair. Repairs on-site are not possible.

Recycling and disposal

The product is low in pollutants, can be recycled and meets the requirements of the WEEE directive 2012/19/EU "Waste Electrical and Electronic Equipment".

Do not dispose of the product at public disposal sites. For environmentally friendly recycling and the disposal of your old device contact a certified disposal company for electronic scrap or your Siemens contact.

Keep to the local regulations.

You will find information on returning the product on the Internet pages of Siemens Industry Online Support:
Link: (https://support.industry.siemens.com/cs/ww/en/view/109479891)

Training, Service & Support

You will find information on training, service and support in the multilanguage document "DC_support_99.pdf" on the Internet pages of Siemens Industry Online Support:

Link: (https://support.industry.siemens.com/cs/ww/en/view/38652101)

SIMATIC NET glossary

Explanations of many of the specialist terms used in this documentation can be found in the SIMATIC NET glossary.

You will find the SIMATIC NET glossary on the Internet at the following address:

Link: (https://support.industry.siemens.com/cs/ww/en/view/50305045)

Table of contents

1 Application and functions

1.1 Properties of the TIM

Application

The TIM 1531 IRC is a telecontrol communications module for the following SIMATIC automation systems:

- S7-300
- S7-400
- S7-1500

The TIM serves as the connection of the SIMATIC stations via public or private infrastructures to a telecontrol master station. Precisely one local CPU is assigned to the TIM.

As a standalone device it is not dependent on the S7 device family or rack type of the assigned CPU.

For the telecontrol communication the TIM uses the "SINAUT ST7“ protocol.

Due to the capability of the TIM to establish redundant transmission paths between stations and to the master station and the option of storing process values the communication has a high degree of reliability.

Master station types

As the connection partner of the TIM in a telecontrol master station, the following computer assisted applications are supported:

- SINAUT ST7cc
- SINAUT ST7sc
- SIMATIC WinCC OA

If no master station PC is used, the TIM can be used as a master station. For visualization, for example, an HMI panel can be used via the CPU.

Interfaces of the TIM

The TIM is equipped with the following interfaces:

- **3 x Ethernet interface RJ-45**

  Connection to three different subnets

  Telecontrol communication via IP-based WAN/LAN networks
- **1 x serial interface**

  Default can be switched over by configuration: RS-232 ⇔ RS-485

  Telecontrol communication via the following WAN networks:
  - Classic WAN networks: Dedicated line, dialup network
  - IP-based networks (can be switched over by the configuration)

You will find the supported transmission protocols and network types in the section Communications services (Page 14).

Network node types

With its four interfaces, the TIM can used at the following positions of a WAN network:

- Master station
- Node station
- Station

Expansion of existing SINAUT systems

In existing SINAUT systems with SIMATIC S7-300/400 stations and TIM modules for the telecontrol functions, the TIM 1531 IRC can be used universally for expansions.

1.2 Communications services

The following communications services are supported:

Telecontrol communication

Network types

The TIM makes telecontrol communication possible via the following network types:

- Industrial Ethernet
- Dedicated line / wireless network
- Analog dial-up network, ISDN network

- Mobile wireless networks
  - GSM / GPRS (2G) with MODEM MD720
  - UMTS (3G) / HSPA+ with SCALANCE M router
  - LTE (4G) with SCALANCE M router
- IP-based wireless networks

You will find an overview of the transmission paths and network types in the section Overview: Connection to LAN / WAN (Page 59).

The "SINAUT ST7" protocol

For telecontrol communication via telecontrol networks the TIM uses the ST7 protocol on the application layer (OSI layer 7).

The protocol supports the following functions and services

- **Communication with the control center**

  The TIM communicates via LAN or WAN with an application in the master station.

  You will find the supported master station types in the section Properties of the TIM (Page 13).

- **Direct communication**

  In dial-up networks, mobile wireless networks and Ethernet networks, there is direct communication between the subscribers.

- **Inter-station communication**

  In dedicated line networks and with communication via the Internet with a mobile wireless network (GSM/MSC), the TIM supports inter-station communication between S7 stations via the master station.

  With inter-station communication, the TIM establishes a connection to the master station. The master station forwards the messages to the destination station.

- **Messages: SMS / e-mail**

  When configurable events occur, the TIM can send SMS messages to mobile telephones and e-mails to PCs with an Internet connection.

  - SMS messages can be sent if the TIM is connected to a mobile wireless network via the serial interface and a GSM or GPRS module (MODEM MD720).
  - If the TIM is connected, e-mails can be sent via the Ethernet interface.

    You configure the e-mail protocol to be used and the server access in STEP 7 in the parameter group "E-mail configuration".

  You configure the messages in the message editor, see section Messages (Page 127). The use of program blocks is not necessary for this.

You will find information about addressing in ST7 and configuring in the section Communication types (Page 56).

Protocols with security functions

- **MSC**

  For secure telecontrol communication the transmission protocol "MSC" (OSI layer 3) is available. MSC can be used for communication between two TIM modules (not between the TIM and a master station application).

  MSC is IP-based and can be used in the following networks:

  – Ethernet
  – Internet (DSL)
  – Mobile wireless network (GSM) and Internet

  The following variants of the protocol are available:

  – MSC

    Simple Internet communication via the Internet (DSL)

  – MSCsec

    Secure Internet communication when security requirements are higher.

- **IPsec / VPN (via router SCALANCE M)**

  VPN stands fro highly secure communication via mobile wireless and the Internet (DSL) using a SCALANCE M mobile wireless router.

For a description of the protocols, refer to the section Security functions (Page 18).

S7 communication

For reading / writing data from and to the local CPU via S7 connections, the following services are supported.

- **PG communication**

  Communication with an engineering station

- **PG routing**

  S7 communication across subnetwork boundaries

- **PUT/GET**

  To exchange data with a local S7-300/400 CPU the TIM supports communication using the the program blocks PUT/GET as client and server.

- **READ/WRITE**

  To exchange data with a local S7-1500 CPU the TIM supports communication using the the program blocks READ/WRITE as client and server.

The Ethernet interfaces of the TIM must be configured with the network type "Neutral" for the S7 communications services.

You will find information on the communications functions and the program blocks in the STEP 7 information system.

Web server connections (WBM)

- **HTTP/HTTPS**

  The TIM has an integrated Web server with which you can connect from a PC via HTTP/HTTPS. Connections are possible via LAN or WAN

  The Web pages display diagnostics information and configuration data.

  You will find information on the content and operation in the section The Web server (WBM) (Page 263).

1.3 PG routing

PG routing between telecontrol modules

PG routing is supported between the modules listed in the table and via the specified media.

A requirement for the CPs is that the options "S7 communication" and "Online functions" are enabled in the "Communication types" parameter group.

Module Medium (protocol)	TIM 1531 IRC	TIM 4R-IE	TIM 3V-IE / TIM 3V-IE Advanced	CP 1243-8 IRC	CP 1542SP-1 IRC
TIM 1531 IRC	Ethernet (S7) Ethernet (MSC) RS-232	Ethernet (S7) Ethernet (MSC) -	Ethernet (S7) Ethernet (MSC) -	Ethernet (S7)	Ethernet (S7)
TIM 4R-IE	Ethernet (S7) Ethernet (MSC) -	Ethernet (S7) Ethernet (MSC) RS-232	Ethernet (S7) Ethernet (MSC) RS-232	-	-
TIM 3V-IE	Ethernet (S7) Ethernet (MSC) -	Ethernet (S7) Ethernet (MSC) RS-232	Ethernet (S7) Ethernet (MSC) RS-232	-	-
CP 1243-8 IRC	Ethernet (S7)	-	-	-	-
CP 1542SP-1 IRC	Ethernet (S7)	-	-	-	-

"RS-232" means communication via dedicated line or dialup network.

See the performance data and the configuration limits for the number of supported connections.

1.4 Security functions

Security functions of the transmission protocols

The transmission protocols that can be used for telecontrol communication support the following security functions:

- **MSC**

  The MSC protocol supports authentication of the communications partners and simple encryption of data. A user name and a password are included in the encryption. An MSC tunnel is established between the MSC station and MSC master station.

- **MSCsec**

  MSCsec supports authentication of the communications partners and data encryption with a user name and password.

  In addition to this, the shared automatically generated key is renewed between the communications partners at a configurable Key exchange interval.

Further security functions of the TIM

The TIM also supports the following security functions:

- **NTP (secure)**

  For secure transfer during time-of-day synchronization

- **STARTTLS / SSL/TLS**

  For the secure transfer of e-mails

- **HTTPS**

  For secure access to the Web server of the TIM

- **SNMPv3**

  For secure transmission of network analysis information safe from eavesdropping

---

Note

Plants with security requirements - recommendation

Use the following options:

- If you have systems with high security requirements, use the secure protocols for example HTTPS or SNMPv3.
- If you connect to public networks, you should use security modules with a firewall, see below.

Refer to the information in the section Security recommendations (Page 51).

Additional protection be using security modules

With Industrial Ethernet Security, individual devices, automation cells or network segments of an Ethernet network can be protected. The following security modules are suitable for connecting the TIM to public networks:

- SCALANCE M800

  Routers for IP-based data transfer via DSL or mobile networks of the standards GPRS, EGPRS, UMTS, LTE

- SCALANCE S

  Security modules for connection to Ethernet networks

The data transfer of the TIM along with a security module can be protected from the following attacks by a combination of different security measures:

- Data espionage
- Data manipulation
- Unwanted access

Secure underlying networks can be operated via additional Ethernet interfaces of the TIM or CPU.

Using the security modules mentioned above SCALANCE M / SCALANCE S the following additional security functions can be used:

- **Firewall**
  - IP firewall with stateful packet inspection (layer 3 and 4)
  - Firewall also for "non-IP" Ethernet frames according to IEEE 802.3 (layer 2)
  - Limitation of the transmission speed to restrict flooding and DoS attacks ("Define IP packet filter rules")
  - Global firewall rule sets
- **Protection for devices and network segments**

  The protection provided by the firewall can cover individual devices, several devices or even entire network segments.

- **Communication made secure by IPsec tunnels (VPN)**

  VPN tunnel communication allows the establishment of secure IPsec tunnels for communication with one or more security modules.

  VPN can be used for communication via mobile wireless and the Internet (DSL) along with a SCALANCE M router. The SCALANCE M800 product line includes various VPN routers with encryption software and a firewall.

  The router can be put together with other modules to form VPN groups during configuration. IPsec tunnels (VPN) are created between all security modules of a VPN group. All internal nodes of these security modules can communicate securely with each other through these tunnels.

- **Logging**

  To allow monitoring, events can be stored in log files that can be read out using the configuration tool or can be sent automatically to a Syslog server.

You will find further information on the functionality and configuration of the security functions in the information system of STEP 7 and in the manual /9/ (Page 321).

1.5 Other services and properties

Other services and properties

- **Data point configuration**

  Due to the data point configuration in STEP 7, programming program blocks in order to transfer the process data is unnecessary. The process data is configured as individual data points and transferred one-to-one to the communications partner.

- **Saving messages**

  Data messages that cannot be transferred to the communications partner due to connection disruptions are saved.

  – In the message memory up to 100000 messages can be buffered.

  – If an SD card is used, messages can also be saved on the SD card.

- **IP configuration**

  The TIM supports IP addresses according to IPv4.

  Address assignment:

  – The IP address, the subnet mask and the address of a gateway can be set manually in the configuration.

  – As an alternative, the IP address can be obtained from a DHCP server or by other means outside the configuration.

- **Time-of-day synchronization**

  On every interface the TIM provides the option of receiving and forwarding the time of day.

  Synchronization method NTP (Network Time Protocol) via Industrial Ethernet:

  – NTP

  – NTP (secure)

  You can find information in the section Time-of-day synchronization (Page 85).

  For information on the format of the time stamp of the messages, refer to the section Time stamp (Page 89).

- **Access to the Web server of the TIM**

  With the aid of the Web server, you can display module and diagnostics data.

- **Storage and event-driven transfer of process data**

  The TIM can store events of different classes and transfer the corresponding process values individually or bundled together to the communications partner. The transfer can be triggered by various triggers.

- **Analog value processing**

  Analog values can be preprocessed on the TIM according to various methods, see section Analog value preprocessing (Page 119).

- **Online functions**

  From an engineering station (ES) on which STEP 7 is installed, you can use the online functions of STEP 7 via an Ethernet interface of the TIM to access the S7 CPU if the station is located in the same IP subnet.

  The following online functions are available:

  – Downloading project or program data from the STEP 7 project to the station

  – Querying diagnostics data on the station

  – Downloading firmware files to the TIM

- **SNMP**

  As an SNMP agent, the TIM supports data queries using SNMPv1 and SNMPv3 (Simple Network Management Protocol).

  For more detailed information, refer to section SNMP (Page 288).

1.6 Configuration examples

Below, you will find configuration examples for the TIM.

Sending SMS messages

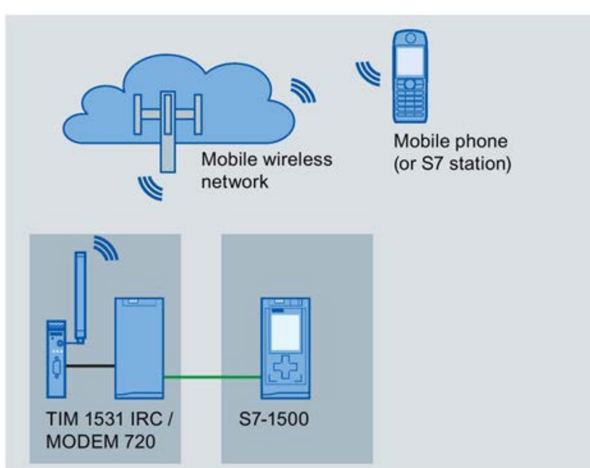


Figure 1-1 Sending messages by SMS

The TIM can send SMS messages to a mobile phone. SMS messages are generated and sent due to events.

Communication via Ethernet / Internet, sending e-mails

In the sample configuration shown, S7 stations communicate with a redundant master station via the Ethernet interfaces of the TIM.

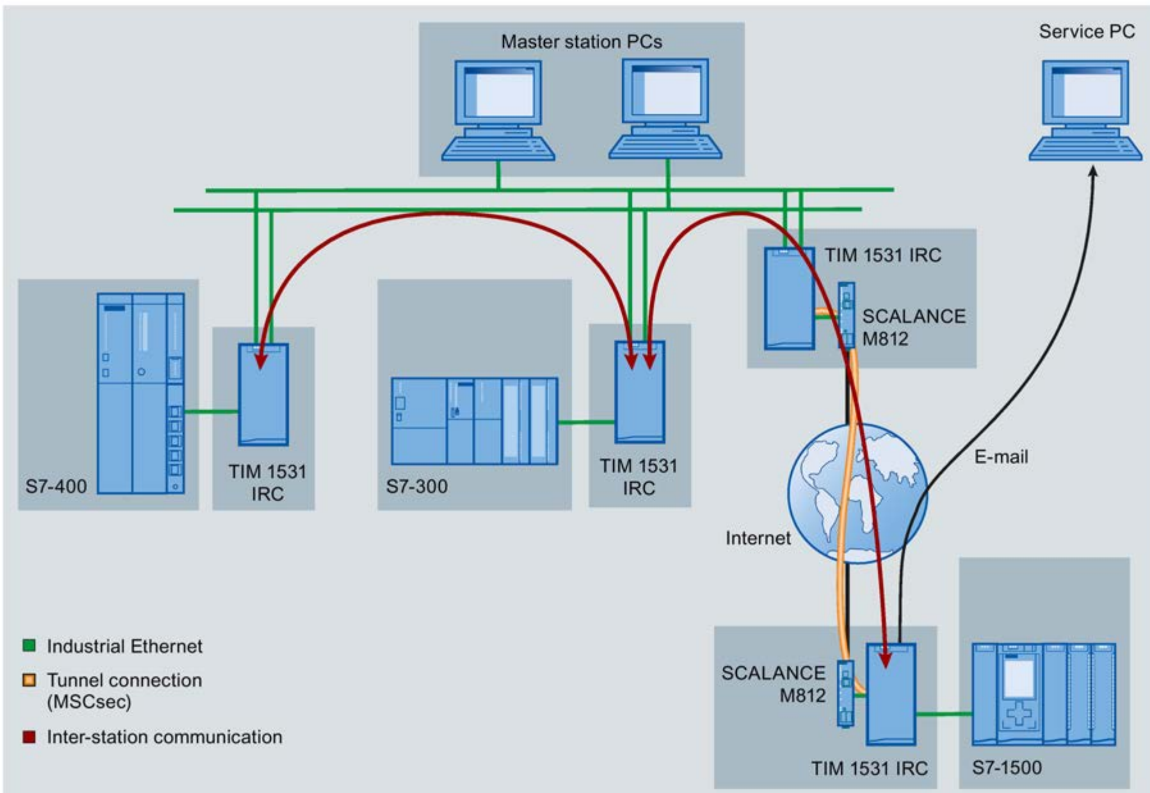


Figure 1-2 Communication via Ethernet / Internet

E-mails

The TIM can generate and send e-mails due to events. The following recipients are possible:

- PCs with an Internet connection
- Mobile phones
- SIMATIC stations with the appropriate program blocks

Inter-station communication

Inter-station communication between S7 stations with a TIM is possible via IP-based and WAN networks.

Inter-station communication via mobile wireless runs via a master station that forwards the messages to a target subscriber.

Path redundancy using the serial interface

In the following example in addition to the Ethernet interface, the serial interface of the TIM is also used. This allows redundant transmission paths to be set up.

The figure shows two examples in which the following interfaces are used:

- Ethernet interface for communication via Ethernet / Internet

  The TIM modules use the transmission protocol MSC or MSCsec.
- Serial interface for communication via a WAN network (dedicated line or dialup network)

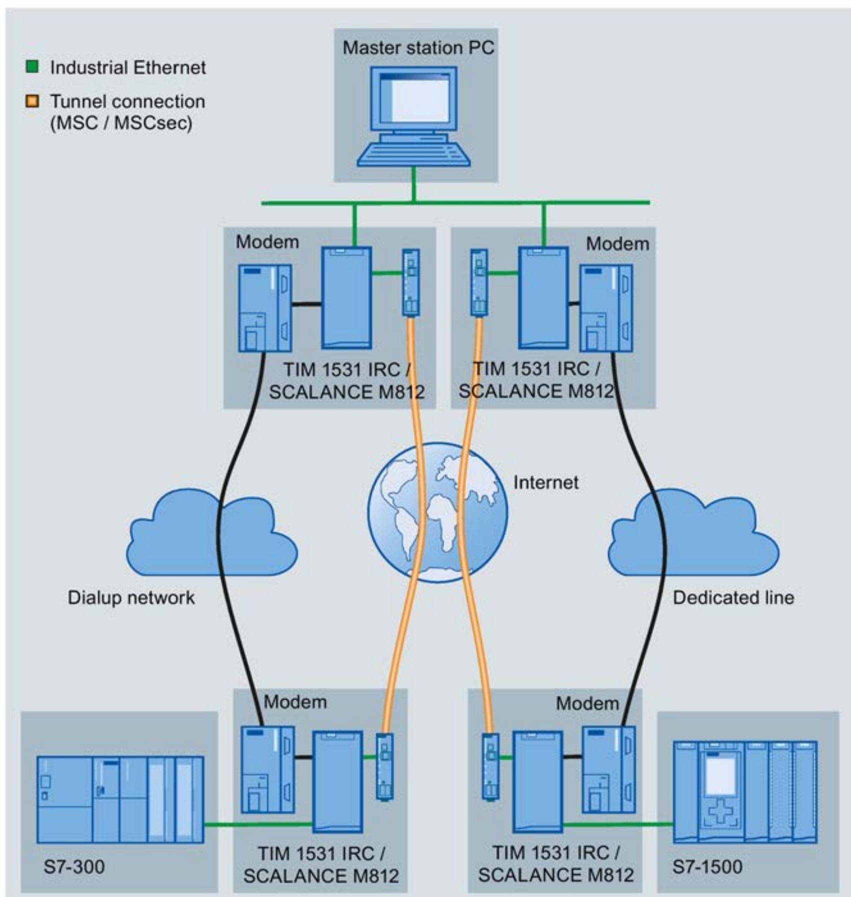


Figure 1-3 Communication via redundant paths

Telecontrol communication via mobile wireless and wireless networks

Mobile wireless

In these examples the TIM modules of the master communicate with the station or node station TIMs via mobile wireless:

- Left: Use of the Ethernet interface with VPN
- Right: Use of the serial interface and the transmission protocol MSCsec

Private wireless networks

The left station TIM is configured as a node station. Underlying this an analog wireless network with several stations is connected via the serial interface. An IP-based wireless network could also be connected via the Ethernet interface.

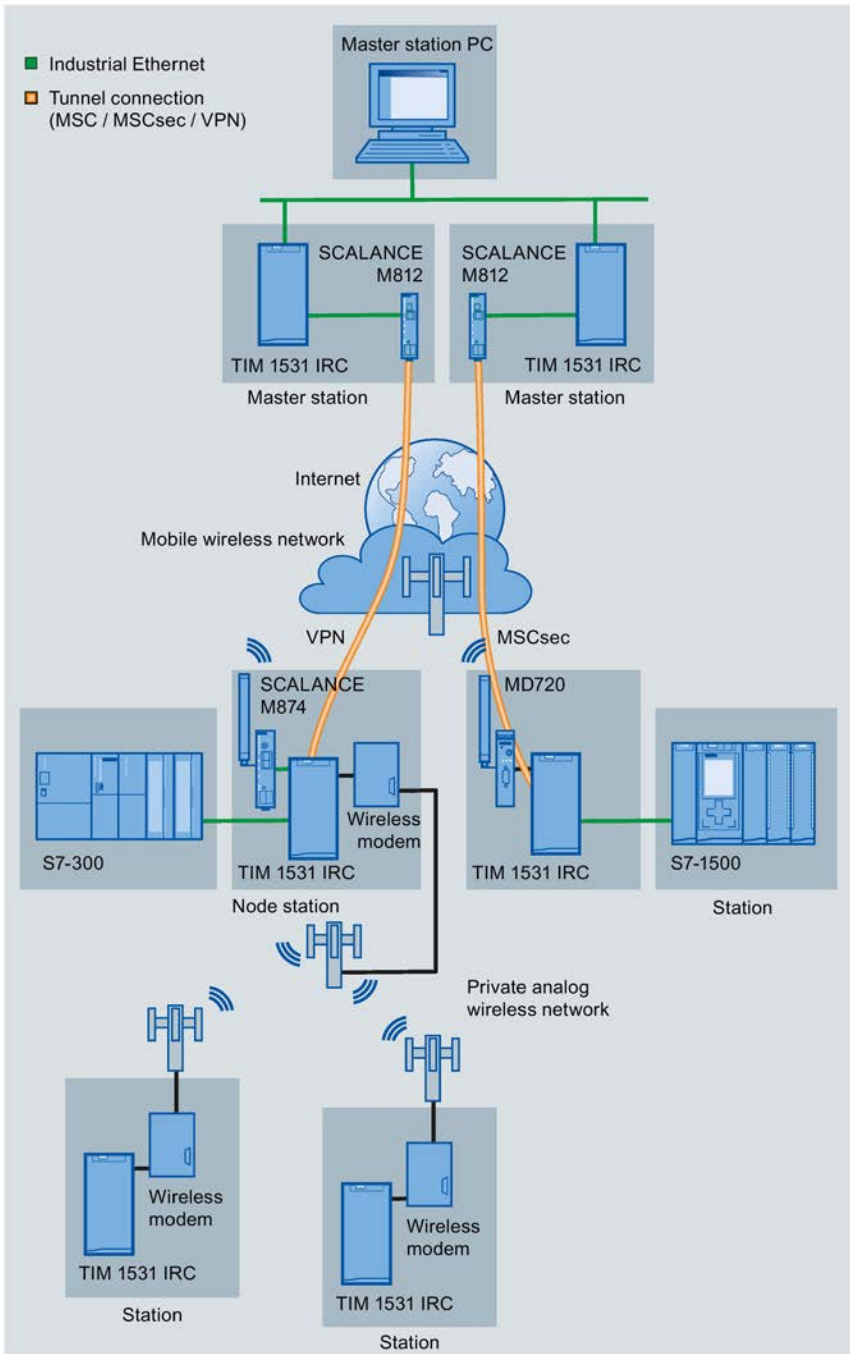


Figure 1-4 Communication using mobile wireless (at the top) and wireless (at the bottom)

1.7 Performance data and configuration limits

Connection resources

- **Telecontrol connections**

  The number of connections or communications partners is limited for the two interface types and every individual interface.

  The maximum number of connections or communications partners depends on the network node type with which the interface is set.

<table>
<tr><th>Network node type</th><th>Interface</th><th>Max. number per interface type</th><th>Network type</th><th>Max. number of connections / partners per interface</th></tr>
<tr><td rowspan="5">Master station</td><td>Ethernet X1</td><td rowspan="3">128</td><td rowspan="3">TCP</td><td>• 62 S7 connections
or
• 127 MSC connections</td></tr>
<tr><td>Ethernet X2</td><td>62 S7 connections</td></tr>
<tr><td>Ethernet X3</td><td>62 S7 connections</td></tr>
<tr><td rowspan="2">Serial X4</td><td rowspan="2">32</td><td>Dedicated line</td><td>32 partners</td></tr>
<tr><td>Dialup network</td><td>32 partners</td></tr>
<tr><td rowspan="5">Node / station</td><td>Ethernet X1</td><td rowspan="3">32</td><td rowspan="3">TCP</td><td>• 16 S7 connections
or
• 1 MSC connection</td></tr>
<tr><td>Ethernet X2</td><td>16 S7 connections</td></tr>
<tr><td>Ethernet X3</td><td>16 S7 connections</td></tr>
<tr><td rowspan="2">Serial X4</td><td rowspan="2">16</td><td>Dedicated line</td><td>16 partners</td></tr>
<tr><td>Dialup network</td><td>16 partners</td></tr>
</table>

- **E-mail / SMS**

  At runtime, a connection can be established to send an e-mail or SMS message.

- **S7 connections**

  – When using ST7 communication:

    66 connection resources in total incl. telecontrol connections (see above) and max. 4 PG/OP connections (see below)

  – Without using ST7 communication:

    4 connection resources only for PG/OP connections (see below)

- **PG/OP connections**

  4 connection resources for connections to the engineering station or HMI devices

  (included in the configuration limits of the S7 connections, see above)

- **PG routing**

  Max. 4 simultaneous connections
- **Online functions**

  See PG/OP connections
- **HTTP/HTTPS**

  Max. 2 connections per Ethernet interface
- **Connectable control center applications**

  Number of SINAUT control centers connectable via Ethernet ST7cc / ST7sc Max. 4

Number of data points for the data point configuration

The maximum number of configurable data points is 1000.

Message memory: Send buffer / SD card

The TIM has a message memory (send buffer) for the values of data points configured as an event.

The send buffer has a maximum size of 100000 messages. The size of the message memory is divided equally among all configured communications partners.

You will find details of how the send buffer works (storing and sending events) as well as the options for transferring data in the section Process image, type of transmission, event classes (Page 113).

You can set saving of frames on an optional SD card in the configuration, see section Configuration (Page 58).

Messages: E-mail / SMS

Up to 10 messages which the TIM can send as e-mails or SMS messages can be configured in STEP 7.

- Number of characters per SMS message

  Maximum number of characters that can be transferred per SMS message: 160 ASCII characters including any value sent at the same time
- Number of characters per e-mail

  Maximum number of characters that can be transferred per e-mail: 256 ASCII characters including any value sent at the same time

1.8 Requirements and compatibility

1.8.1 Software requirements

Software for configuration and online functions

To configure the TIM, the following configuration tool is required:

- STEP 7 Professional V15.0

The following configuration tool is required to configure the proxy module in STEP 7 V5 to expand SINAUT systems:

- STEP 7 V5.6

  together with

- SINAUT engineering software V5.5 + SP3

1.8.2 Usable CPUs

Compatible CPUs

The following can be used as the local CPU of the TIM:

- S7-1500

  All CPUs that can be configured in STEP 7 as of firmware version V2.1
- S7-300

  All CPUs that can be configured in STEP 7 with a PROFINET interface
- S7-400

  All CPUs that can be configured in STEP 7

1.9 Expansion of SINAUT projects

1.9.1 Modules for new SINAUT projects and those to be expanded

New SINAUT projects in the TIA Portal

For new SINAUT projects, the following modules can be configured in STEP 7 Basic / Professional V15 (TIA Portal) without pre-configuration in STEP 7 V5.

Table 1- 1 Configuring modules in STEP 7 Basic / Professional

Module name (firmware version)	Usable modules in STEP 7 Basic / Professional as of V15	
	Catalog module	STEP 7 version
TIM 3V-IE (V2.6)	TIM 3V-IE	STEP 7 Professional
TIM 3V-IE Advanced (V2.6)	TIM 3V-IE Advanced	STEP 7 Professional
TIM 4R-IE (V2.6)	• TIM 4R-IE • TIM 4R-IE Stand-alone	STEP 7 Professional
CP 1243-8 IRC (V3.0)	CP 1243-8 IRC	STEP 7 Basic
TIM 1531 IRC (V1.0)	TIM 1531 IRC	STEP 7 Professional
CP 1542SP-1 IRC (V2.0)	CP 1542SP-1 IRC	STEP 7 Professional

Extension of existing SINAUT projects in the TIA Portal

SINAUT projects with TIM modules for the SIMATIC S7-300 and S7-400 series, which were configured in STEP 7 V5, can be extended with communication modules of the S7-1200/1500 series which are configured in STEP 7 Basic or STEP 7 Professional in the TIA Portal.

The following modules are available as communication modules for expanding existing SINAUT systems:

- CP 1243-8 IRC

  As of STEP 7 Basic V13.0 SP1
- TIM 1531 IRC

  As of STEP 7 Professional V15

To avoid having to create, configure and program the entire STEP 7 V5 project in STEP 7 Professional, the STEP 7 V5 project can be expanded by S7-1200/1500 stations with compatible communication modules.

The procedure for configuration of a communication module for the expansion is as follows:

1. Configuration of a placeholder (proxy) for an S7-1200/1500 module in the STEP 7 V5 project

   The proxy receives the SINAUT-specific communication, connection and address parameters.
2. Export the configuration data (SDBs) of the proxy from STEP 7 V5 as a text file.
3. Import the configuration data of the proxy into a compatible module in STEP 7 Basic / Professional.

   The new module adopts the SINAUT-specific communication, connection and address parameters from STEP 7 V5.
4. Complete the configuration of the new module in STEP 7 Basic / Professional.

This procedure is supported by the following modules:

Table 1- 2 Module migration from STEP 7 V5 to STEP 7 Basic / Professional (TIA Portal)

Module function for STEP 7 V5 project expansion			**S7-1200/1500 module in STEP 7 Basic / Professional**	
TIM (function) for expansion	**Proxy to be used in the catalog**		**Compatible module**	**Required STEP 7 version**
TIM 3V-IE Advanced	PROXY CP1243-8 IRC	⇒	CP 1243-8 IRC	STEP 7 Basic
TIM 4R-IE	PROXY TIM 1531 IRC	⇒	TIM 1531 IRC	STEP 7 Professional
TIM 4R-IE Stand-alone	PROXY TIM 1531 IRC	⇒	TIM 1531 IRC	STEP 7 Professional

---

Note

TIM 4R-IE Stand-alone for S7-400 becomes TIM 1531 IRC

A TIM 4R-IE Stand-alone required in STEP 7 V5 that is assigned to a CPU-400 must be replaced by a TIM 1531 IRC for the expansion of classic SINAUT projects in STEP 7 Professional.

A TIM 4R-IE Stand-alone can only be created in new projects that are configured exclusively in STEP 7 Professional.

You can find details on configuration in the section STEP 7 V5 configuration of the proxy (Page 313).

1.9.2 Requirements for the expansion

Furthermore, the following requirements apply to importing configuration data from STEP 7 V5 to STEP 7 Basic / Professional.

Additional modules that are not listed here are available for new projects in STEP 7 Basic / Professional.

Requirements: Software versions

The above-mentioned configuration tools are required in the specified versions for expansion of SINAUT projects.

- **STEP 7 V5 project**

  A configuration file of the communication module from a consistent STEP 7 V5 project is required.

Requirements: Firmware versions/module update

- **Firmware of the communication module**

  The following firmware versions of the communication modules used are required:

  – **TIM 1531 IRC - V1.1**

    A TIM 1531 IRC V1.0 can be exchanged for a TIM 1531 IRC V1.1 but not the other way round.

    The mode of the "Telecontrol configuration" ("Basic settings") is set to "Configure". The existing data is adopted.

  – **CP 1243-8 IRC - V2.1**

    Adoption of the data from STEP 7 V5 without the option of making changes

    A physical V2.1 CP can be exchanged for a V3.1 CP but not the other way round. The CP used must have hardware product version 2. The existing data is adopted.

Requirements: TIM 1531 IRC

- **Communication partner of the TIM 1531 IRC**

  Requirements for importing ST7 connections between a TIM 1531 IRC and a CPU 1500:

  – The CPU 1500 must have at least the firmware version V2.5.

  – TIM and CPU must be networked in STEP 7 Professional before the import.

  – When using TD7onCPU, the CPU must not yet be assigned to the TIM in STEP 7 Professional.

  Requirements for ST7 connections between a TIM 1531 IRC and a PC application:

  – Prior to import, the TIM and the PC station must be networked in STEP 7 Professional.

  IP addresses of a CPU 1500 or a PC station as ST7 communication partner of the TIM 1531 IRC:

  – The IP address in the STEP 7 Professional project must be identical to the address of the respective CPU or PC station configured in STEP 7 V5.

LEDs, connectors, switches, card slot

2

2.1 Overview: LEDs, connectors, switches, card slot

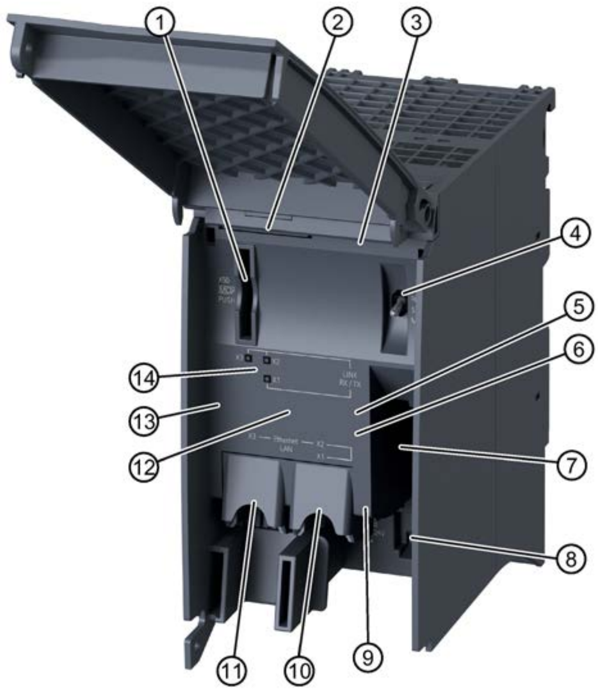


Figure 2-1 Location of the elements LEDs, connectors, switches, SD card slot

Table 2- 1 Meaning of the number symbols

①	X50: Receptacle for SD card	⑧	X80: Power supply (socket)
②	Status and diagnostics LEDs (covered): RUN, ERROR, CONNECT	⑨	X1: Ethernet interface (Gigabit Ethernet)
③	Article number (covered)	⑩	X2: Ethernet interface (Fast Ethernet)
④	Switch	⑪	X3: Ethernet interface (Fast Ethernet)
⑤	MAC address Ethernet interface X2	⑫	Serial number (2D matrix code, alphanumeric)
⑥	MAC address Ethernet interface X1	⑬	MAC address Ethernet interface X3
⑦	X4: Serial interface, plug (RS-232/RS-485)	⑭	LEDs for Ethernet communication status: X1, X2, X3

2.2 LEDs

At the top on the front of the TIM there are the status and diagnostics LEDs that can also be seen when the front cover is closed.

Position	LED name	Meaning
Left	**RUN**	Operating mode
Middle	**ERROR**	Error
Right	**CONNECT**	Connection status (to the CPU / to the communications partner)

Below the front cover there are LEDs for the communication status of the Ethernet interfaces:

LED name	Meaning
X1 / X2 / X3	Communication status of Ethernet interface, X1, X2 and X3

The LED symbols have the following meaning:

Table 2- 2 Legend for the tables

Symbol	🟢 🟡 🔴	⚪	✴🟢 ✴🟡 ✴🔴	-
Meaning / LED status	ON (LED lit)	OFF	LED flashes	Any

Status and diagnostics LEDs

Table 2- 3 Meaning of the LED patterns in productive operation *

RUN (yel-low/green)	ERROR (red)	CONNECT (green)	Meaning
⚪	⚪	⚪	No supply voltage or supply voltage too low
🟢	-	-	TIM in RUN mode
🟡	-	-	TIM in STOP mode
⚪	-	-	TIM in maintenance status
-	-	🟢	All configured connections established **
-	-	⚪	No configured connections established **
-	-	✴🟢	At least one configured connection aborted **
-	🔴	-	Missing or incorrect configuration
-	✴🔴	-	Duplicate IP address detected

***** During startup and during the firmware update the TIM shows deviant LED patterns.

****** Relates to connections both to the CPU and to the communications partner.

- **Startup**

  For the LED pattern see section Startup - LED pattern (Page 49).
- **Firmware update**

  For the LED pattern see section Update firmware (Page 284).

LEDs of the Ethernet interfaces

Every interface has an LED that informs about the connection status with Ethernet and the message traffic of the port.

Table 2- 4 Meaning of the LED statuses

X1 / X2 / X3 (yellow/green)	Meaning
	No connection to the Ethernet network
	Ethernet connection without data transfer
	Ethernet connection with data transfer

2.3 Electrical connectors

2.3.1 Ethernet interfaces (X1, X2, X3)

Ethernet interfaces

The Ethernet connectors are located behind the cover of the enclosure. The interfaces are RJ-45 jacks according to IEEE 802.3.

Note

Connection to subnets

The three Ethernet interfaces are not designed as a switch, but are intended for connection to different networks. Operation in the same physical network is not permitted.

The pin assignment of the Ethernet interfaces and other data can be found in the section Technical specifications (Page 293).

2.3.2 Serial interface X4

Serial interface X4

The serial interface is designed as 9-pin D-sub miniature male connector.

The interface can be operated in the two following standards:

- RS-232

  or

- RS-485

You specify the standard to be used with the configuration in STEP 7, see section Advanced options (Page 67).

You will find the pinout of the interface in the section Pin assignment of the serial interface (Page 296).

2.3.3 X80: External power supply

External power supply

The connector X80 (socket) for the external 24 VDC power supply is located on the front of the TIM.

The power supply is connected to the TIM with the supplied plug-in terminal block. The plug-in terminal block is designed so that it can only be plugged in in one position in the X80 socket of the TIM.

The connector X80 has electronic reverse polarity protection.

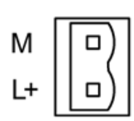


Figure 2-2 Connector X80 for the power supply

Table 2- 5 Pin assignment of the socket for the external power supply

Labeling	Function
M	Reference ground
L+	24 VDC

For information on the connector, refer to the section "Wiring (Page 45)".

You will find further data on the power supply in section Technical specifications (Page 293).

2.4 Switch

Operating status switch

⚠ WARNING
EXPLOSION HAZARD
Do not press the button if there is a potentially explosive atmosphere.

The switch is a combined element with switch and button functions. It has three settings:

- **RUN**

  RUN mode

  This is the basic setting for productive operation.

- **STOP**

  STOP mode

  Set the TIM to STOP before disconnecting the power supply. Setting to STO may also be necessary if an error/fault occurs to be able to restart the TIM afterwards.

  In STO mode, the following functions are disabled:

  – Telecontrol operation of the TIM

  – Time-of-day synchronization via Telecontrol connections

  – S7 connections are terminated.

  The following functions remain enabled:

  – Access to the WBM of the TIM

  – Time-of-day synchronization with NTP

  – Diagnostic functions of the TIM

- **MRES**

  Maintenance status

  The setting "MRES" is adopted by pressing and holding down the switch in this position (button function).

  By pressing the switch for at least 5 seconds, the functions described below are triggered. If you release the switch before the 5 seconds have elapsed, the TIM shows no reaction.

  Functions:

  – Restart

    When the switch is pressed in STOP mode

    The TIM restarts.

  – Reset to factory settings

    Only when the switch is pressed during a restart

    The TIM is reset to the factory settings. Note the effects of the reset, particularly the deletion of the configuration data.

    For information on the effects of resetting, refer to the section Resetting to factory settings (Page 287).

Functions in the setting "MRES"

Functions when holding the switch in the "MRES" setting for at least 5 seconds:

Table 2- 6 Button functions

Function	Previous setting or action	Effect
Restart	Pressing the switch from the "STOP" setting	The TIM restarts. Release the switch at the beginning of the restart.
Reset	Pressing the switch after triggering a restart	The TIM is reset to the factory settings. All the configuration data is deleted.

For the relevant LED patterns see the following section:

- Startup - LED pattern (Page 49)
- Resetting to factory settings (Page 287)

2.5 SD card slot

"X50": Slot for an optional SD card

You have the option of using an SD card as an exchangeable storage medium for storing important data.

- Configuration data
- Process data

  Values of data points configured as an event.

An SD card does not ship with the TIM.

Compatible cards

You will find a list of compatible SD cards in the appendix SD cards (Page 303).

Use one of the SD cards listed there.

The minimum size is 24 MB. If you use an SD card with less storage space, it is possible that not all process data (see below) can be saved retentively.

Card errors / diagnostics

Card errors are indicated by entries in the diagnostics buffer.

Inserting the card

Inserting the SD is described in the section Inserting the SD card (Page 47).

Retentive storage of important data on the SD card

The SD card is an exchangeable storage medium for storing the following data safe from power failure.

- **Configuration data**

  The configuration data is backed up on the SD card following every change in a configuration file. Storing configuration data on the SD card serves the following purpose:

  – Device replacement without engineering station

    If the TIM needs to be replaced for maintenance purposes or another location, by transferring the SD card from the previous to the new TIM, the configuration data can be made available to the new TIM. In this case, you do not need an engineering station with a STEP 7 project.

  The configuration data is saved on the SD card when the TIM starts up.

  For information on device replacement refer to section Module replacement (Page 292).

- **Process data**

  You have the option of storing values of data points configured as an event. This serves the following purposes:

  – Storage when there are connection disruptions

    If the TIM cannot send the data to its communications partner due to connection disruptions, this remains stored until the connection is restored.

  – Storage if there is a power failure

  You can specify whether values of events are stored globally in the configuration of the TIM. There are also options for storing on the SD card for every data point created as an event.

Installation, connecting up, commissioning 3

3.1 Important notes on using the device

Safety notices on the use of the device

Note the following safety notices when setting up and operating the device and during all associated work such as installation, connecting up or replacing the device.

WARNING

Accessories: Personal injury and property damage can occur.

The installation of expansions that are not approved for SIMATIC NET products or their target systems may violate the requirements and regulations for safety and electromagnetic compatibility.

Only use expansions that are approved for the system.

WARNING

If the device is installed in a cabinet, the inner temperature of the cabinet corresponds to the ambient temperature of the device.

Overvoltage protection

NOTICE

Protection of the external power supply

If power is supplied to the module or station over longer power cables or networks, the coupling in of strong electromagnetic pulses onto the power supply cables is possible. This can be caused, for example by lightning strikes or switching of higher loads.

The connector of the external power supply is not protected from strong electromagnetic pulses. To protect it, an external overvoltage protection module is necessary. The requirements of EN61000-4-5, surge immunity tests on power supply lines, are met only when a suitable protective element is used. A suitable device is, for example, the Dehn Blitzductor BVT AVD 24, article number 918 422 or a comparable protective element.

Manufacturer:
DEHN+SOEHNE GmbH+Co.KG Hans Dehn Str.1 Postfach 1640 D-92306 Neumarkt, Germany

3.1.1 Notes on use in hazardous areas

WARNING
EXPLOSION HAZARD Do not open the device when the supply voltage is turned on.

WARNING
EXPLOSION HAZARD Replacing components may impair suitability for Class 1, Division 2 or Zone 2.

WARNING
The device may only be operated in an environment with pollution degree 1 or 2 (see IEC 60664-1).

WARNING
EXPLOSION HAZARD Do not connect or disconnect cables to or from the device when a flammable or combustible atmosphere is present.

WARNING
When used in hazardous environments corresponding to Class I, Division 2 or Class I, Zone 2, the device must be installed in a cabinet or a suitable enclosure.

WARNING
If a device is operated in an ambient temperature of more than 60 to 70 °C, the temperature of the device housing may be higher than 70 °C. The device must therefore be installed so that it is only accessible to service personnel or users that are aware of the reason for restricted access and the required safety measures at an ambient temperature higher than 60 °C.

3.1.2 Notes on use in hazardous areas according to ATEX / IECEx

WARNING

DIN rail

In the ATEX and IECEx area of application only a Siemens DIN rail 6ES7590-1Axx0-0AB0 may be used to mount the TIM.

WARNING

Requirements for the cabinet

To comply with EC Directive 2014/34 EU (ATEX) or the conditions of IECEx, this enclosure or cabinet must meet the requirements of at least IP54 in compliance with EN 60529.

WARNING

Cable

If the cable or conduit entry point exceeds 70 °C or the branching point of conductors exceeds 80 °C, special precautions must be taken. If the equipment is operated in an air ambient in excess of 50 °C, only use cables with admitted maximum operating temperature of at least 80 °C.

WARNING

Take measures to prevent transient voltage surges of more than 40% of the rated voltage. This is the case if you only operate devices with SELV (safety extra-low voltage).

WARNING

LAN connection (Local Area Network)

A LAN or LAN segment with all the interconnected devices should be contained completely in a single low voltage power distribution system in a building. The LAN is designed either for “Environment A” according to IEEE802.3 or "Environment 0" according to IEC TR 62102.

Do not connect any electrical connectors directly to the telephone network (Telephone Network Voltage) or a WAN (Wide Area Network).

WARNING

EXPLOSION HAZARD

Do not press the button if there is a potentially explosive atmosphere.

3.1.3 Notes on use in hazardous areas according to UL HazLoc and FM

This equipment is suitable for use in Class I, Division 2, Groups A, B, C and D or non-hazardous locations only.

This equipment is suitable for use in Class I, Zone 2, Group IIC or non-hazardous locations only.

WARNING

EXPLOSION HAZARD

Do not connect or disconnect while the circuit is live or unless the area is known to be free of ignitible concentrations.

WARNING

Do not remove or replace while circuit is live when a flammable or combustible atmosphere is present.

WARNING

Explosion hazard

Do not disconnect equipment when a flammable or combustible atmosphere is present.

WARNING

EXPLOSION HAZARD

The equipment is intended to be installed within an ultimate enclosure. The inner service temperature of the enclosure corresponds to the ambient temperature of the module. Use installation wiring connections with admitted maximum operating temperature of at least 30 ºC higher than maximum ambient temperature.

WARNING

Wall mounting is only permitted if the requirements for the housing, the installation regulations, the clearance and separating regulations for the control cabinets or housings are adhered to. The control cabinet cover or housing must be secured so that it can only be opened with a tool. An appropriate strain-relief assembly for the cable must be used.

WARNING

Substitution of components may impair suitability for Division 2.

WARNING

Substitution of components may impair suitability of the equipment.

3.2 Installation

WARNING

Open equipment

The device is "open equipment" acc. to the standard UL 61010-2-201. To fulfill requirements for safe operation with regard to mechanical stability, flame retardation, stability, and protection against contact, the following alternative types of installation are specified:

- Installation in a suitable cabinet.
- Installation in a suitable enclosure.
- Installation in a suitably equipped, enclosed control room.

WARNING

Cable temperatures

If the cable or housing socket exceeds 70 °C or the branching point of the cables exceeds 60 °C, special precautions must be taken. If the equipment is operated in an ambient environment in excess of 40 °C, only use cables with permitted maximum operating temperature of at least 80 °C.

NOTICE

Install and remove the TIM only when the power is off

Switch off the power supply of the TIM before you install or remove the TIM. Installing and removing modules with the power supply on can lead to damage to the modules and to loss of data.

NOTICE

Installation location - Dependency of the temperature range

Note the dependency of the permitted temperature range of the installation location.

- Horizontal installation of the rack (DIN rail) means a vertical position of the modules.
- Vertical installation of the rack (DIN rail) means a horizontal position of the modules.

You will find the permitted temperature ranges in the section Technical specifications (Page 293).

Minimum clearances

Mount the TIM so that its upper and lower ventilation slits are not covered, allowing adequate ventilation as protection from overheating.

Keep to the following minimum clearances for the circulation of air when the rack is installed horizontally:

- Above the TIM: At least 33 mm
- Below the TIM: At least 25 mm

Installation of the rack	Installation position of the modules
Horizontal installation of the rack	
Vertical installation of the rack	

Installation on a DIN rail

Note

Protecting the modules from slipping on the DIN rail

If you install the modules in an area with mechanical load, use suitable clamping devices at both ends of the device group to secure the modules on the DIN rail, e.g. Siemens and retainer 8WA1808.

The end retainers prevent the modules separating under mechanical load.

The TIM is suitable for mounting on a standard rail of the S7-1500 (article number 6ES7590-1Axx0-0AB0).

1. Hang the TIM on the mounting rail.
2. Tilt the TIM towards the back.
3. Screw the TIM at the bottom using the securing screws (tightening torque 1.5 Nm).
4. Ground the mounting rail, see next section.

Grounding

For reasons of electrical safety, the DIN rail must be connected to the protective conductor system (PE) of the electrical system.

You will find details on grounding and installation in the SIMATIC S7-1500/ET 200MP Manual Collection, see /2/ (Page 319).

Removal from the DIN rail

Follow the steps below to remove the TIM from the DIN rail:

1. Switch the TIM to STOP.
2. Turn off the power supply to the TIM.
3. Pull off the Ethernet cables and the serial bus.
4. Release the securing screws of the TIM.
5. Tilt the TIM out of the standard rail.

3.3 Wiring

WARNING

Power supply

The device is designed for operation with a directly connectable safety extra-low voltage (SELV) and protective extra-low voltage (PELV) according to IEC 60364-4-41.

Note

Protective ground

A PELV circuit contains a connection to protective ground. Without a connection to protective ground, or in case there is a fault in the connection to the protective ground, the voltage for the circuit is not stabilized.

NOTICE
Suitable fusing for the power supply cable
The current at the connecting terminals must not exceed 4 A. Use a fuse for the power supply that protects against currents > 4 A. The fuse has to be designed for protection of DC power supply circuits as well as for the following requirements. • In areas used according to NEC or CEC: – Suitable for DC (min. 60 V / max. 4 A) – Cut-off voltage min. 10 kA – UL/CSA listet (UL 248-1 / CSA 22.2 No. 248.1) – Classes R, J, L, T or CC • In other areas: – Suitable for DC (min. 60 V / max. 4 A) – Cut-off voltage min. 10 kA – Approved for power supply circuits (branch circuits) according to local regulations (e.g. IEC 60127-1, EN 60947-1) – Breaking characteristics: B or C circuit breakers and fuses You do not need a fuse for the power supply cable if you use a voltage source according NEC Class 2 or a power supply from the range of accessories, see attachment Power supply (Page 303).

Order of the work

NOTICE
Connection only with power off
Only connect the TIM with the power switched off.

The device can be disconnected from the power supply with the terminal block.

Requirement: The TIM is mounted.

1. Connect the external power supply to the terminal block of connector X80.
2. Connect the lines of the (different) Ethernet networks to the desired ports of the TIM.

   Note the permitted combinations of the Ethernet connectors in the section Ethernet interfaces (X1, X2, X3) (Page 33).
3. Plug the cable to the modem in the serial connector of the TIM.

   When connecting a star shaped network, remember to connect the terminating resistor for RS-485 operation of the interface. See also the section below.

NOTICE
Contacting the shield of the cable on the plug
The shield of the cable must be contacted. To do this, strip the insulation from the end of the cable and connect the shield to functional earth.

For the network connection products from the Siemens accessories program are recommended, see Appendix Cables, connecting cables (Page 310)

Turn the power supply on only after the TIM has been completely wired and connected.

The further procedure is described in the section Commissioning (Page 48).

Power supply at connector X80

Recommendation: Use the same power supply as the CPU if this is in the vicinity of the TIM.

The 2-terminal plug-in terminal block with polarity reversal protection has the following pin assignment for the X80 socket:

Terminal	Assignment
L+	24 VDC
M	Ground

You will find information about the connectable cable cross sections, power consumption and further technical details in section Technical specifications (Page 293).

RS-485: Connection of the terminating resistor

If you connect a star-shaped network with several dedicated line or dialup network modems to the serial interface of the TIM then operate the serial interface with the RS-485 standard.

You set the RS-485 standard in the configuration of the TIM. With a network operating according to RS-485 you also need to activate the cable terminating resistor of the bus cable in the configuration.

3.4 Inserting the SD card

Figure 3-1 Slot for the SD card (yellow frame)

You will find the SD cards supported by the TIM in the appendix SD cards (Page 303).

NOTICE
Only remove / insert the SD card when the power is off
You can only remove or insert the SD card when the TIM is not supplied with power.
If you remove or insert the SD card during operation, data on the card can be damaged.

Inserting the SD card

1. Insert the SD card into the compartment until you can feel the card lock in place.

Removing the SD card

1. By pressing, unlock the card.
2. After unlocking it takes the card out of the slot.

3.5 Commissioning

Requirement

Requirements for commissioning the TIM are as follows:

- The TIM is mounted and connected up.
- The TIM is fully configured in STEP 7, refer to the section Configuration (Page 51).
- The STEP 7 project with the TIM is open on the engineering station.
- For loading, the engineering station is connected to an Ethernet interface of the TIM.

Downloading the configuration data

To start productive operation the TIM requires the STEP 7 configuration data.

To download the configuration data, follow the steps outlined below:

1. Change the switch of the TIM to the RUN setting.
2. Turn on the power supply to the TIM.

   The TIM starts up with the factory defaults and remains standing with a yellow flashing RUN LED.

   You will find information on the LED displays during startup in the section Startup - LED pattern (Page 49).
3. Start by downloading the configuration data.

You will find more detailed information on loading in the following sections of the STEP 7 information system:

- “Loading project data"
- “Using online and diagnostics functions"

On completion of the download close the front cover of the TIM.

3.6 Startup - LED pattern

During startup the TIM shows different LED patterns than at runtime. You will find the LED patterns at runtime in the section LEDs (Page 32).

LED patterns at startup / restart

The LED symbols have the following meaning:

Table 3- 1 Legend for the tables

Symbol	green, yellow, red (lit)	(unlit)	green, yellow, red (flashing)	-
Meaning / LED status	ON (LED lit)	OFF	LED flashes	Any

The table below describes the LED patterns in the individual startup phases of the TIM.

Table 3- 2 Meaning of the LED patterns at startup

Startup phase	RUN (yellow/green)	ERROR (red)	CONNECT (green)	Meaning
1	ON (yellow)	ON (red)	ON (green)	Hardware initialization and downloading the operating system
2	flashes (yellow)	OFF	OFF	Downloading the firmware
3	flashes (green)	OFF	OFF	Downloading the configuration data
4	ON (green)	OFF	OFF	Startup successful / start of productive operation
Startup aborted				
4	flashes (green)	flashes (red)	flashes (green)	The startup was incorrect and was aborted. *

* If the startup is incorrect, note which phase (1, 2, 3) is reached and when it is aborted. You may be able to localize the error.

Configuration 4

4.1 Security recommendations

Keep to the following security recommendations to prevent unauthorized access to the system.

General

- You should make regular checks to make sure that the device meets these recommendations and other internal security guidelines if applicable.
- Evaluate your plant as a whole in terms of security. Use a cell protection concept with suitable products.
- Do not connect the device directly to the Internet. Operate the device within a protected network area.
- Keep the firmware up to date. Check regularly for security updates of the firmware and use them.
- Check regularly for new features on the Siemens Internet pages.
  - You can find information on Industrial Security here:
    Link: (http://www.siemens.com/industrialsecurity)
  - You can find information on security in industrial communication here:
    Link: (http://w3.siemens.com/mcms/industrial-communication/en/ie/industrial-ethernet-security/Seiten/industrial-security.aspx)
  - You can find a publication on the topic of network security (6ZB5530-1AP02-0BA5) here:
    Link:
    (http://w3app.siemens.com/mcms/infocenter/content/en/Pages/order_form.aspx?nodeKey=key_518693&infotype=brochures)
    Enter the following filter: 6ZB5530

Physical access

Restrict physical access to the device to qualified personnel.

Network attachment

Do not connect the TIM directly to the Internet. If a connection of the TIM to the Internet is required, use the security variants of the telecontol protocols or use protection mechanisms in front of the TIM. Protection mechanisms are for example a SCALANCE M router or a SCALANCE S security module with firewall.

Security functions of the product

Use the options for security settings in the configuration of the product. These includes among others:

- Protection levels

  Configure a protection level of the CPU.

  You will find information on this in the information system of STEP 7.

- Security function of the communication
  - Using the security functions of the telecontrol protocols.
  - Use the secure protocol variants for example NTP (secure) or SNMPv3.
  - Leave access to the Web server deactivated.

Passwords

- Define rules for the use of devices and assignment of passwords.
- Regularly update the passwords to increase security.
- Only use passwords with a high password strength. Avoid weak passwords for example "password1", "123456789" or similar.
- Make sure that all passwords are protected and inaccessible to unauthorized personnel.

  See also the preceding section for information on this.

- Do not use one password for different users and systems.

Protocols

Secure and non-secure protocols

- Only activate protocols that you require to use the system.
- Use secure protocols when access to the device is not prevented by physical protection measures.

  The NTP protocol provides a secure alternative with NTP (secure).

  The HTTP protocol provides a secure alternative with HTTPS when accessing the Web server.

Table: Meaning of the column titles and entries

The following table provides you with an overview of the open ports on this device.

- **Protocol / function**

  Protocols that the device supports.

- **Port number (protocol)**

  Port number assigned to the protocol.

- **Default of the port**
  - Open

    The port is open at the start of the configuration.
  - Closed

    The port is closed at the start of the configuration.
- **Port status**
  - Open

    The port is always open and cannot be closed.
  - Open after configuration

    The port is open if it has been configured.
  - Open (login, when configured)

    As default the port is open. After configuring the port, the communications partner needs to log in.
  - Closed after configuration

    The port is closed because the TIM is always client for this service.
- **Authentication**

  Specifies whether or not the protocol authenticates the communications partner during access.

Protocol / function	Port number (protocol)	Default of the port	Port status	Authentication
S7 and online connections	102 (TCP)	Open	Open	No
HTTP	80 (TCP)	Closed	Open after configuration	No
HTTPS	443 (TCP)	Closed	Open after configuration	Yes
SNMP	161 (UDP)	Open	Open after configuration	Yes (with SNMPv3)

Ports of communication partners and routers

Make sure that you enable the required client ports in the corresponding firewall on the communications partners and in intermediary routers.

These can be:

- MSC / 26382 (TCP) - configurable with a central interface: 1024 .. 65535
- NTP / 123 (UDP)
- DNS / 53 (UDP)

4.2 Configuration in STEP 7

Configuration in STEP 7

You configure the modules and networks in STEP 7 Professional. You will find the required version in the section Software requirements (Page 27).

The following sections describe the STEP 7 configuration. If individual parameters are not described here, you will find help help in the STEP 7 information system.

Components of a rack

The TIM is configured as a standalone device. No further devices may be configured in the station of the TIM.

The assigned CPU with which the TIM exchanges data is configured in a separate rack. You will find the compatible CPUs in the section Usable CPUs (Page 27).

Configuration - overview

Configuration involves the following steps:

1. Create a STEP 7 project with security functions activated.

   Note

   Activated security functions

   If you have not yet activated any security functions or have logged on in "Global security settings", you first need to log on as a security user.

   The configuration of the TIM is only possible with activated security functions.

   Also for access to the Web server of the TIM, the users must be created in "Global security settings".

2. Insert the required TIM modules and SIMATIC stations with CPU, input/output modules and the other required modules and configure the modules.
3. Create PLC tags for the data to be transferred in the CPU.

   The input and output data of the station is not addressed directly by the TIM but via PLC tags. You require these for the data points of the TIM (see below).
4. Create the necessary networks for networking the interfaces.

   Refer to the information in the section Networking a TIM (Ethernet addresses / WAN settings) (Page 61).
5. Complete the configuration of the modules and networks.

6. Create telecontrol connections for the telecontrol communication using ST7.

   For telecontrol connections that use the S7 protocol (Network type "Neutral") you do not need to create any separate S7 connections.

   You will find the description in the section Creating telecontrol connections for the ST7 communication (Page 99).
7. Create the required data points and messages.
8. Save the project.

For complete commissioning of the TIM, you need to download the configuration data to the relevant TIM. The project data is stored in the work memory of the TIM.

When using an optional SD card, the data is stored on the SD card of the TIM, see section Commissioning (Page 48).

You will find information on loading in the STEP 7 information system.

Copy functions for larger projects

The copying functions of STEP 7 reduce configuration effort. Use the copying functions in particular with large projects with several identical or similar stations.

- Copying devices and stations

  If you use several similar TIM modules or CPUs in a project, it is advisable to configure the first device first and then to copy the entire configuration data. This can include the following objects:

  – The user program of CPUs
  – PLC tags of CPUs
  – Data points of TIM modules

  After copying the address parameters and other various parameters need to be adapted.

  Recommendation:
  Network the interfaces only after copying the devices.
- Copying PLC tags

  If you use several similar items of input and output data, you can copy PLC tags in the tag table.

  After copying you only need to adapt the address parameters.
- Copying data points

  In much the same way as with PLC tags, you can also copy data points.

  For a detailed description of the functions, refer to the section Data point configuration (Page 104).

4.3 Communication types

4.3.1 Communication types

"Communication types"

In this parameter group you specify the protocol for the telecontrol communication.

- **Enable telecontrol communication**

  The communication with the communications partners via telecontrol protocols is activated generally.

- **Protocol type**
  - **ST7**

    The protocol is preallocated.

4.3.2 Communication via the SINAUT ST7 protocol

The ST7 protocol

Network types

ST7 is a protocol for telecontrol communication via WANs (Wide Area Network).

With ST7 it is possible to communicate via the following network types:

- Classic WAN networks
- IP-based WAN networks

Subscriber types

The following subscriber types that support the ST7 protocol are possible communications partners:

- SIMATIC stations with TIM modules or a CP capable of ST7
- Control center applications capable of ST7

Connections

Communication with the ST7 protocol runs via configurable connections. You make the required settings in the STEP 7 configuration.

Communication via single or redundant telecontrol connections

The ST7 communication between two subscribers runs via configurable paths. To do this, telecontrol connections are created in the "Network data" editor in the "Telecontrol" tab, see section Creating telecontrol connections for the ST7 communication (Page 99).

When different networks between two subscribers have been created in the network configuration, you can create redundant telecontrol connections. In the configuration these are displayed in the "Network data" editor. There the main and the substitute path are specified.

To check the reachability of a communications partner of the module, you can write the connection status to "PLC tags for partner status / path status", see section Communication with the CPU (Page 81).

For redundant telecontrol connections, the behavior is as follows:

- **Main path**

  If the connection between the two subscribers is undisturbed, the main path is always used.

  The status of the main path is rechecked each time there is communication and saved in the PLC tag.

- **Substitute path**

  If the connection via the main path is disturbed, communication is switched to the substitute path.

  The status of the substitute path, on the other hand, is only registered when communication is established.

  If you configure a dialup network connection as the substitute path via which generally a connection is seldom established, you can test the reachability via this path for example by synchronizing the time of day once daily via this path. To do this a connection must be established once daily via the dialup network. The reachability is then updated once daily in the PLC tag.

Addressing communications partners and network nodes

Since the communication between two ST7 subscribers can run via different paths and subnets, precise addressing of the individual subscribers in the ST7 network is necessary. The following two parameters serve this purpose:

- **Subscriber number**

  The subscriber number is unique for every subscriber in a STEP 7 project. The following subscribers require a subscriber number:

  – **TIM**

    Communication module for the ST7 protocol

  – **CP 1243-8 IRC, CP 1542SP-1 IRC**

    Communication module for the ST7 protocol

  – **CPU**

    The local CPU that is assigned to the communication module receives a subscriber number as the end point of an ST7 connection.

    With a CP and a classic TIM in the S7-300 rack, the CPU is assigned automatically to the communication module via the rack.

    With the TIM 1531 IRC and a TIM 4R-IE Stand-alone, you must assign the local CPU manually.

– **Application of the master station PC**

The application that you configure in the PC station of the master station.

For the TIM and the assigned CPU, the subscriber number is configured in the "Subscriber numbers" parameter group of the TIM.

For the application of the master station PC, the subscriber number is configured in the telecontrol connection: "Network data" editor > "Telecontrol" tab

- **WAN address**

  For every networked serial interface in a classic ST7 network a WAN address is assigned. It is unique in the respective WAN network.

  Since telecontrol connections can run via several network nodes and node stations, unique addressing of the interfaces involved is necessary.

  The WAN address is configured in the parameter group "WAN parameters" of the serial interface.

4.4 Basic settings

4.4.1 Configuration

Telecontrol configuration

This is where you specify whether you want to freely configure the module or whether you want to import the configuration data of a proxy module to expand a STEP 7 V5 project.

- Configure

  The subscriber data and SINAUT connections are configured in the parameter groups of the module and in the connections editor of STEP 7 Professional.

- Import

  The configuration data of the proxy module fro, STEP 7 V5 is imported into the module in STEP 7 Professional.

  You can perform the import yourself in the parameter group Partner stations (Page 80).

  To expand STEP 7 V5 projects via proxy modules, see section Expansion of SINAUT projects (Page 28).

Retentive saving of events

If you use an optional SD card in the TIM, in this parameter group you set the conditions for saving the values of messages whose data points are configured as an event.

For the general reaction, the following parameters are available:

- Activate retentive saving

  When there are connection disruptions activates the retentive saving of events on the SD card.
- Number of events before saving

  Saving events on the SD card starts when the number of events in the send buffer configured here is reached after a connection failure.
- Interruption time before saving

  Saving events on the SD card starts when the time of the connection interruption configured here is reached.

Whether or not the values of the individual data points are saved is specified in the configuration of the individual data point in the "Trigger" tab, see section Data points (Page 104).

For the maximum number of savable messages see section Performance data and configuration limits (Page 25).

4.5 Overview: Connection to LAN / WAN

Transfer options

The interfaces of the TIM support the following network types and protocols:

Ethernet interfaces

- **IP-based**

  Communication via LAN (copper / FO cable), Internet and IP-based wireless networks
  - Neutral (via S7 connection)
  - MSC / MSCsec

    For information on the protocol variants, refer to the section Communications services (Page 14).
  - Mobile wireless with VPN

    IP-based mobile phone communication (with gateway to the Internet) only with SCALANCE M router

Serial interface

- **Classic WAN**
  - Dedicated line (incl. analog wireless network)
  - Dial-up network (analog, ISDN mobile phone)
- **IP-based**
  - MSC / MSCsec via GPRS

Connection combinations of the interfaces

The following table provides an overview of the various connection options of the TIM interfaces and the devices required for them (modems, routers, switches).

The table contains the information for the interfaces of the TIM and for the connection of the communication partner. The protocols or services used are listed.

Network type / transmission path	Connection TIM		Standard, protocol, service	Connection of the partner	Partner type *
	Serial interface	Ethernet interface			
Ethernet	-	SCALANCE M / SCALANCE S	VPN	SCALANCE M / SCALANCE S	• PC • TIM
	-	SCALANCE X / W		SCALANCE X / W	• PC • TIM
Mobile phone + Internet	-	DSL router	MSC / MSCsec	DSL router	TIM
	-	SCALANCE M	MSCsec	SCALANCE M	TIM
	-	SCALANCE M	VPN	SCALANCE M	TIM
IP wireless network	-	IP wireless modem	IP	IP wireless modem	TIM
Dedicated line	Dedicated line modem	-	RS-232 / RS-485	Dedicated line modem	TIM
	Analog wireless modem			Analog wireless modem	
Analog dial-up network	Dial-up network modem	-	V.32bis/V.34bis	Dial-up network modem	TIM
ISDN dial-up network	ISDN modem	-	• ISDN • ISDN + GSM/CSD	• ISDN modem • MODEM MD720	TIM
Mobile network	MODEM MD720	-	GSM/CSD	• DSL router • MODEM MD720	TIM
			GSM + MSC/MSCsec	• DSL router	

***** PC: Control center computer with ST7-capable application, for example, SINAUT ST7cc/ST7sc.

You will find information on the accessories in the following sections or literature sections:

- Router SCALANCE M (Page 307)
- Dedicated line and dialup network modems (Page 305)
- MODEM MD720 (Page 306)
- Connecting cables between TIM and modem: Cables, connecting cables (Page 310)

  To connect a modem to the TIM a cable must be ordered.
- SCALANCE S: /10/ (Page 321)

4.6 Networking a TIM (Ethernet addresses / WAN settings)

Interfaces of the TIM

The arrangement of the interface in the STEP 7 device symbol (network view) corresponds to the structure of the device. The interfaces are arranged as follows:

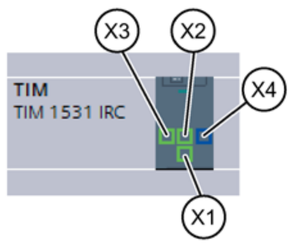


Figure 4-1 Device symbol of the TIM with interface numbers

Networking interfaces

To network an interface depending on the initial situation you have different options:

- Creating a subnet
- Connecting two target devices via a new subnet
- Connecting devices to existing subnet
- Selecting an existing subnet from the "Subnet" list

You will find the description of the individual methods in the STEP 7 information system.

Interface networking with the TIM

You will find the transmission options of the TIM depending on the interface in the section Overview: Connection to LAN / WAN (Page 59).

Recommendation networking:

To network the interfaces with a WAN network, the following procedure is recommended:

1. Network the WAN networks in the network view of STEP 7.

   In the graphic network view, you have an overview of the subnets of the entire system in the project.
2. First configure the interface parameters described in the section WAN settings of the interfaces (Page 63):
   - WAN type
   - Network type
   - Network node type
   - Modem type

3. Select the relevant interface to create a new WAN network. Alternatively:

   In the parameter group "Network interface with" of the interface:

   – Using the "Add new subnet" button

   On the interface in the device symbol of the TIM

   – Using the shortcut menu "Create subnet"

   – Graphically by dragging (holding the mouse pointer pressed) to the interface symbol of the communications partner

   A new WAN network is created that adopts the network type from the connected interface.

Network representation of a classic WAN

A classic WAN network is displayed in blue.

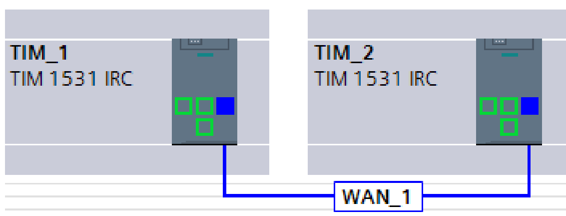


Figure 4-2 TIM modules, serial interfaces networked via classic WAN.

MSC connections

Note the options of using the "MSC" protocol via the interfaces of the TIM:

Note

MSC via TIM interfaces:

- **Ethernet interface X1**

  Of the three Ethernet interfaces, MSC is supported only by interface X1.

- **Serial interface**

  For the serial interface you need to set the WAN type "IP-based" and the Network type "MSC".

MSC connections are represented as follows:

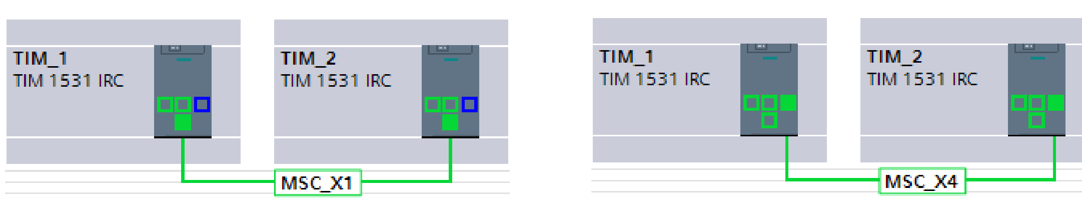


Figure 4-3 MSC connection via Ethernet interface X1 (left) and via serial interface (right)

The color of the serial changes from blue to green if the WAN type is changed from "Classic WAN" to "IP-based".

4.7 WAN settings of the interfaces

WAN parameters of interfaces

In this parameter group of the interfaces you will find parameters that decide the properties of the connected WAN network: First, configure the respective interface of the TIM. The subsequently connected WAN network adopts the most important settings (see below).

- **WAN type**

  Selection of the WAN type of the interface:

  - IP-based

    Default setting of the Ethernet interface

  - Classic WAN

    Default setting of the serial interface

    To use MSC via GPRS, the serial interface can be changed to "IP-based".

- **Network type**

  For classic WAN:

  - Dedicated line
  - Dialup network

  For IP-based WAN:

  - Neutral
  - Mobile wireless with VPN
  - MSC

- **Connection mode**

  Decides the Network node type of the interface:

  - Master station
  - Node station

    For TIM modules that act as node stations, the following applies:

    The interface in the direction of the master station is configured as a "node station".

    The interface in the direction of a lower-level station is configured as a "master station".

  - Station

With the classic network types dedicated line and dialup network, a modem needs to be connected to the serial interface of the TIM. You select the suitable modem type using the following parameters (serial interface only):

- **Modem type**
  - MD2

    Dedicated line modem (network type "Dedicated line")
  - MD3

    Modem for analog dialup networks (network type "Dialup network")
  - MD4

    ISDN modem (network type "Dialup network")
  - MD720

    Siemens-GSM modem MD720 (network type "Dialup network")
  - Third-party modem

    Any compatible modem for the network types "Dedicated line" or "Dial-up network" (analog / ISDN)

    If you configure "Dialup network" and want to transmit via mobile wireless, you need to select "MD720" as the modem type.

4.8 Parameters of WAN networks

Parameters of classic WAN networks

First configure the parameter group "WAN settings" of the TIM interfaces. The subsequently connected WAN networks adopt the most important settings of the respective interface.

The classic WAN networks, shown in blue in STEP 7, have the following parameter groups.

General

Like for any other network, this is where you configure the name and the S7 subnet ID.

Network settings

Network configuration

- **WAN type**

  The WAN type is taken from the network type of the connected interface:

  For classic WAN:
  - Dedicated line
  - Dialup network

  For IP-based WAN:
  - Neutral
  - Mobile wireless with VPN
  - MSC

Access method

- **Access method**

  If required, set the default access method for the respective WAN type here:

  – Dedicated line: Polling / Polling with time slots

  – Dial-up network: ST7 protocol (spontaneous)

Frame parameters

- **Frame format**

  – FT1.2

  – FT2

- **Acknowledgment type**

  – Short acknowledgment (1 byte)

  – Long acknowledgment (5 bytes)

- **Repetition factor**

  The repetition factor decides how often a data frame that has not been acknowledged positively is repeated.

- **Max. frame length**

  The maximum frame length depends on the largest length (240 bytes) of an ST7 data frame within the network.

  From this parameter time values (e.g. send repetition time) are derived for monitoring functions.

Network settings

- **Dependence on direction**

  Direction dependency of the network

  – Duplex

  – Half duplex

- **Transmission speed**

  Speed at which the CP and modem communicate.

Dial-up network options

- **Cancel parameter**

  Number of dialing attempts with unavailable partner until the attempts to establish the connection are finally ended. (0 .. 127)

  – 1 ... 127: Final abort when attempted calls failed n times (no connection or no data transfer).

  – 0: Final abort when a connection was established 127 times without data transfer.

- **Redial**

  Number of attempted calls until a disruption is reported.

- **Network password**

  Formerly "Customer identification"

  The parameter ensures that connections in dial-up networks are made only to configured partners.

  Range of values: 1 ... 127

  Default setting: 0

Time-of-day synchronization

- Enable time-of-day synchronization for WAN

  When the parameter is enabled, you specify whether the time for the time-of-day synchronization of the connected stations should be transmitted via the WAN network.

You specify the synchronization cycle if the parameter is enabled.

Note

Transfer of the setting by stations

The connected TIM modules adopt the settings made here on the network.

Station list

This is where you can find an overview table of the stations connected to the network with their most important parameters.

4.9 Ethernet interface

4.9.1 Ethernet interface > address parameters

Ethernet interface > Ethernet addresses > IP protocol ...

In the following parameter groups configure the IP address parameters of the Ethernet interface.

You will find information on configuration in the STEP 7 information system.

Ethernet interface > Port [Xn P1]

You will find information on configuration in the STEP 7 information system.

4.9.2 Advanced options

4.9.2.1 MSC protocol settings

The following descriptions apply to the MSC-specific parameter groups of the two interface types of the TIM.

For information on each of the following parameter groups "TCP connection monitoring" and "Port" see section Ethernet interface (Page 67).

MSC protocol settings - Ethernet: MSC master station

The description applies to the parameter group of the Ethernet interface with the Network node type "Master station".

Access to MSC master station

First, specify whether the MSC center or the router should be available via DNS, a fixed or a dynamic IP address.

The MSC master station is connected to the Internet via a router.

If the router can be reached via a dynamic IP address, there must be at least one available DNS server in the network.

- **MSC port of the router**

  The port number is preset. If necessary, you can change the port number. Note that the port number of both partners must be identical. The port number canot be configured for the station.

  Default setting: 26382. Permitted range: 1024...65535

- **Host name**

  If the router should be available using a host name enter it here.

- **IP address DNS server**

  Here, enter the addresses of the DNS servers in the subnet.

MSC monitoring settings

- **MSC monitoring time**

  Monitoring time (minutes) for the MSC connection

  – Behavior of the master station:

    If there is no data traffic between the MSC master and station within the monitoring time, the MSC master sends a sign-of-life frame to the station.

    If the station does not acknowledge the sign-of-life frame, the master station sets the station to 'Faulty'.

  – Behavior of the station:

    If there is no data traffic with the MSC master station within the monitoring time, the station sets the center to 'Faulty'.

  Default setting: 10. Permitted range: 0...65535

  If you enter 0 (zero), the function is disabled.

- **Collect data volume**

  If the option is enabled, the MSC master passes the monthly transferred data volume of the stations to online diagnostics.

MSC protocol settings - Ethernet: MSC station/node station

The description applies to the parameter group of the Ethernet interface with the Network node type "Station" / "Node station".

MSCsec protocol

The use of the secure protocol variant is configured only in the stations. You can also enable use of MSCsec for individual stations of an MSC network.

- **Activate security protocol MSCsec**

  If the option is enabled, the secure version of the MSC protocol "MSCsec" is used.

- **Key exchange interval**

  Time (h) after which the secret key is renewed between the MSC station and master station.

  Default setting: 0. Permitted range: 0...255

MSC protocol settings - serial interface: MSC station

The description applies to the parameter group of the serial interface regardless of the setting of the "Network node type" parameter.

GPRS access

Here you configure the data of the access to the mobile wireless network.

- **Activate PIN**

  Activates the PIN of the installed SIM card. If the contract with your network provider does not include a PIN, then disable this function.

- **PIN / Repeat PIN**

  Entry of the PIN of the SIM card inserted in the modem. Not required for mobile wireless contracts without a PIN.

- **Extra transmission time**

  Time offset (s) with slow networks to prevent premature stopping of connection establishment.

  Default setting: 10. Permitted range: 0...65535. If you enter 0 (zero), the function is disabled.

  The following values are usual:

  – Mobile wireless networks: 10

  – Wireless or satellite transmission: 0 .. 1

- **Max. allowed disruption time**

  Tolerance time (s) for a detected connection disruption.

  If there is still a disruption on the connection when the set time has elapsed, the disruption is signaled to all accessible connection partners of the disrupted station.

  Default setting: 0. Permitted range: 0...255. If you enter 0 (zero), the function is disabled.

MSCsec protocol

- **Activate security protocol MSCsec**

  If the option is enabled, the secure version of the MSC protocol "MSCsec" is used.

  Can only be configured if the setting of the Network node type of the interface is "Station" or "Node station".

- **Key exchange interval**

  Time (h) after which the secret key is renewed between the MSC station and master station.

  Default setting: 0. Permitted range: 0...255

APN settings

Here you configure the access point (APN) from the mobile wireless network to the Internet.

- **APN - country**

  Country of the network provider that provides the APN.

  Select the country and one of the preset network providers or select the entry "User-defined" and configure the APN manually.

- **APN - provider**

  name o the network provider that provides the APN. Select the name from the drop-down list.

- **APN**

  Access Point Name

  Name of the access point. You obtain this information from your network provider.

- **APN user name**

  Entry of the user name for Internet access via the mobile wireless network. You obtain this information from your network provider.

- **APN password**

  Entry of the password for Internet access via the mobile wireless network. You obtain this information from your network provider.

4.9.2.2 TCP connection monitoring

Ethernet interface > Advanced options > TCP connection monitoring

- **TCP connection monitoring time**

  Function: If there is no data traffic within the TCP connection monitoring time, the TIM sends a keepalive to the communications partner.

  Default setting: 10 s. Permitted range: 1...65535 s

- **TCP keepalive monitoring time**

  After sending a keepalive, the TIM expects a reply from the communications partner within the keepalive monitoring time. If the TIM does not receive a reply within the configured time, it terminates the connection.

  Default setting: 1 s. Permitted range: 1...65535 s

4.9.2.3 Transmission settings

Transmission settings

- **Send conditional spontaneous frames as blocks**

  Indicates the form of transfer of data frames with the transmission mode "Conditional spontaneous".

  – Option enabled

    Several data items are collected and sent only when a frame is filled to the maximum.

  – Option disabled

    The data is transmitted spontaneously as a single frame.

  The transmission mode "Conditional spontaneous" is set differently for the TD7 versions:

  – TD7onTIM

    In the "Trigger" tab of the data point (data point editor)

  – TD7onCPU

    Via the "Unconditional" parameter of the data point typicals (TD7 blocks)

- **Max. allowed disruption time**

  Tolerance time (s) for a detected connection disruption.

  If there is still a disruption on the connection when the set time has elapsed, the disruption is signaled to all accessible connection partners of the disrupted station.

  Default setting: 0. Permitted range: 0...255. If you enter 0 (zero), the function is disabled.

4.9.3 Web server access

Access to the Web server

You can activate access to the Web server of the TIM via HTTP/HTTPS for each individual Ethernet interface.

As default access is disabled. Refer to the explanations in section Security recommendations (Page 51).

You enable the Web server and make further settings in the parameter group "Web server", see section Web server (Page 79). There you can also enable or disable access.

For access to the Web server you need to enable access on the Ethernet interface ("Access to the Web server") and enable the Web server itself "Web server".

4.10 Serial interface > Advanced options

4.10.1 MSC settings

The parameter group becomes visible if you use the serial interface as an Ethernet interface (WAN type = IP-based).

For a description of the parameters, see section MSC protocol settings (Page 67).

4.10.2 Dedicated line

Settings dedicated line

Only the configurable parameters are explained.

Settings serial interface

- **Interface standard**

  Standard of the serial interface: RS232 / RS485

  Select the following value:

  – RS232

    When a modem with an RS-232 interface is connected to the interface of the TIM

  – RS485

    When a modem with an RS-485 interface is connected

    With parallel connection of several modems to the interface of the TIM (star-shaped network)

- **RS-485 termination**

  Enable the option when connecting a terminating resistor for the RS-485 bus when a star-shaped network is connected.

Number of spontaneous frames

- **Number of spontaneous frames**

  Only for interfaces with the Network node type "Station" / "Node station".

  The parameter decides after how many frames the master station has the opportunity of transferring its pending frames to the station. As maximum, the configured number of spontaneous frames that are pending at the time of the first call frame are transferred.

  Default setting: 200. Permitted range: 0...255

  If 0 (zero) is set, all spontaneous frames are sent at the call.

Connection establishment

- **Extra transmission time**

  Time offset (s) with slow networks to prevent premature stopping of connection establishment.

  Default setting: 10. Permitted range: 0...65535. If you enter 0 (zero), the function is disabled.

  Usually in dedicated line networks the value is 0:

Time options

- **RTS/CTS delay time**

  The RTS/CTS delay (ms) can be required when a modem is connected via an RS-485 interface.

  Default setting: 0. Permitted range: 0...65535

  – Value 0

    After setting the RTS signal, (turn transmitter on) transmission only starts when the CTS signal (Clear To Send) was set by the modem.

  – Value > 0

    Transmission is not delayed until the CTS signal of the modem.

    After the RTS signal has been set, transmission is delayed for the configured time and then started immediately.

- **Send delay time**

  The send delay time (ms) is used only when the ready to send (CTS signal) comes from the modem (RTS/CTS delay time = 0).

  As soon as the CTS signal comes from the modem, the send delay time is started. Sending starts after the time has elapsed.

  Default setting: 0. Permitted range: 0...65535

- **Polling monitoring time**

  Only for interfaces with the Network node type "Station" / "Node station".

  Time (s) after which the TIM expects to be polled. After this time elapses, the TIM sends a message to its local CPU indicating that the master station is disrupted.

  Default setting: 0. Permitted range: 0...65535

  – Parameters configured:

    After the configured time elapses, the TIM sends a message to its local CPU.

  – Parameter not configured at a transmission speed of 9600 bps:

    After 4 seconds without message traffic the TIM outputs a message.

  – Parameter not configured at a transmission speed of 1200 bps:

    After 32 seconds without message traffic the TIM outputs a message.

- **Max. allowed disruption time**

  Tolerance time (s) for a detected connection disruption.

  If there is still a disruption on the connection when the set time has elapsed, the disruption is signaled to all connection partners of the disrupted station.

  Default setting: 0. Permitted range: 0...255. If you enter 0 (zero), the function is disabled.

Polling parameters

- **Ratio "Polling / Spontaneous"**

  Max. number of spontaneous frames that can be sent by a master station between two polls.

  Only for interfaces with the Network node type "Master station".

  Default setting: 0. Permitted range: 0...255. If you enter 0 (zero), the function is disabled.

Event transmission

- **Type of transmission**

  Specifies the form in which data messages are sent.

  – As single messages

  – Several messages as a block

4.10.3 Dialup network

Settings dialup network

Only the configurable parameters are explained.

Settings serial interface

- **Interface standard**

  Standard of the serial interface: RS232 / RS485 - can be switched over

  Select one of the following values:

  – RS232

    When a modem is connected to the interface of the TIM

  – RS485

    Connection of the internal terminating resistor of the TIM

    With parallel connection of several modems to the interface of the TIM (star-shaped network)

- **RS-485 termination**

  Enable the option when connecting a terminating resistor for the RS-485 bus when a star-shaped network is connected.

Number of spontaneous frames

- **Number of spontaneous frames**

  Only for interfaces with the Network node type "Station" / "Node station".

  The parameter decides after how many messages the master station has the opportunity of transferring its pending messages to the station. As maximum, the configured number of spontaneous frames that are pending at the time of the first call frame are transferred.

  Default setting: 200. Permitted range: 0...255

  If 0 (zero) is set, all spontaneous frames are sent at the call.

- **Limit for locked frames**

  Only for interfaces with the Network node type "Station" / "Node station".

  Messages are marked as locked when due to communications problems they cannot be transferred to the communications partner.

  The parameter specifies the maximum quota of locked messages in the send buffer. If the value is exceeded, the image procedure is used for new messages to prevent the send buffer from overflowing.

  Default setting: 50. Permitted range: 0...90

  If 0 (zero) is set, all spontaneous frames are sent at the call.

Connection establishment

- **Extra transmission time**

  Time offset (s) with slow networks to prevent premature stopping of connection establishment.

  Default setting: 10. Permitted range: 0...65535. If you enter 0 (zero), the function is disabled.

  The following values are usual:

  – Mobile wireless networks: 10

  – Wireless or satellite transmission: 0 .. 1

Call parameters

- **Dialing command**

  Dialing command for the local modem

  Possible values:

  – D (AT command)

  – DP (AT command, pulse dialing)

  – DT (AT command, tone dialing)

  When possible use the dialing command "D".

- **Dialing prefix**

  Access number (outside line) for a private branch exchange (typical entry 0 or 9) or for an alternative telephone provider.

  Permitted range: Max. 12 digits

  With direct connection to the dial-up network and without an alternative telephone provider, this parameter can remain empty.

- **Own phone number**

  Entry of your own telephone number for the network node including the area code.

  Permitted values:

  – Digits 0 ... 9

  – Plus character (+) as placeholder for the trunk prefix (usually 00 or 09) before the international area code

  Example: +1230123456789

AT initialization

- **User-defined**

  If the option is enabled the AT initialization string for the basic settings of the modem can be assigned manually.

  If the option is disabled, the AT initialization string is preset for the specific modem:

  – MD3 : ATS45=3\N0F0&W

  – MD4 : ATS45=83$P1\N0&W

  – MD720 : For information on the initialization string, refer to the manual /5/ (Page 320).

- **AT string**

  Input box for the AT initialization string

Transmission setting for conditionally spontaneous messages

- **Transmission criterion**

  Only for interfaces with the Network node type "Station" / "Node station".

  The transmission criterion controls connection establishment for the transmission of conditional spontaneous messages. This reduces the number of connection retries.

  Range of values:

  – Standard conditions

    No connection will be established due to the existence of conditionally spontaneous messages .

    Only in the following cases will a connection be established to send conditional spontaneous messages.

    - Threatened overflow of the send buffer

    - Connection establishment by the communications partner

  – Degree of filling

    The TIM only establishes a connection when the configured fill level of the send buffer for conditional spontaneous messages is exceeded.

    In the input box, enter the fill level (%) of the send buffer at which when exceeded the TIM establishes a connection.

  – Time

    The TIM sends conditional spontaneous messages at a configured time of day, configurable with "Hours" / "Minutes".

  – Time scheme

    The TIM sends the conditional spontaneous messages cyclically at a configurable interval, configurable with "Hours" / "Minutes".

Time options

- **Call acceptance delay**

  Wait time (s) before accepting an incoming call

  Due to the wait time acceptance of the call is possible for the TIM when a telephone is connected to a shared telephone connection to the TIM.

  Default setting: 0. Permitted range: 0...30

- **Dial test interval**

  The test interval (min) is started when no connection could be established by the TIM after 3 attempted repetitions. When the test interval elapses, the TIM re-establishes the connection again.

  If it fails to establish the connection again, the test interval is restarted.

  If a new message is pending for transfer during the test interval in a master station TIM, the TIM attempts to establish a connection immediately.

  Default setting: 5. Permitted range: 0...255

- **Minimum connection duration**

  Only for interfaces with the Network node type "Master station".

  Minimum connection duration (s) for a dial-up connection.

  The minimum connection duration may be required in fast dial-up networks to be able to wait for the reply during a general request before the connection is terminated.

  Default setting: 5. Permitted range: 0...65535

- **Abort delay time**

  Duration (s) a dial-up connection is retained when the send buffer of the TIM is full and the TIM can send no further messages or data to the CPU.

  During the abort delay time, received messages are acknowledged negatively and the communications partner repeats sent and negatively acknowledged messages.

  When the abort delay time elapses, the connection is terminated.

  Default setting: 20. Permitted range: 0...255

Event transmission

- **Type of transmission**

  Specifies the form in which data messages are sent.

  – As single messages

  – Several messages as a block

Mobile wireless settings

- **PIN**

  PIN of the SIM card of the modem MD720

4.11 Web server

The Web server of the TIM

The TIM provides you with the functionality of a Web server for access using a Web browser. The following functions are available via the Web server:

- Read access
  - A selection of diagnostics data
  - A selection of configuration data
- Write access
  - Setting the time
  - Firmware update
  - Module restart
  - Reset to factory settings
  - Recording of statistics values of the Ethernet interfaces

For a description of the content, refer to the section The Web server (WBM) (Page 263).

Access rights via "Global security settings"

The rights for access to the Web server are configured in STEP 7 in the Global security settings. Only users created there can log on with the Web server using HTTP/HTTPS.

The following preset roles are relevant for Web server access:

- Admin
- Standard
- Diagnostics

The rights of the preset roles mean the following for access to the Web server of the TIM:

- Web: Read system configuration

  Reading out configuration and diagnostics data from the Web server of the TIM.
- Web: Write application parameters

  Writing configuration data
- Web: Update firmware

  Loading firmware files via the Web server
- Web: Access Web diagnostics and the file system of the module

  Read and write access to all data in the Web server

You will find further help on the roles and rights of users in the STEP 7 information system.

Releasing access to the Web server

To be able to connect to the Web server of the TIM, access to the Web server must be enabled for every Ethernet interface, see also section Web server access (Page 72). As default access is disabled.

"Web server" parameter group

General

- **Enable Web server on this module**

  Enables data processing in the Web server of the TIM and allows access to this data.
- **Allow access only using HTTPS**

  Allows access to the Web server only with the secure protocol HTTPS.

Automatic update

- **Enable automatic update**

  Enables automatic updating of the displayed values.

  If the option is disabled, only the values at the time of connecting to the Web server are displayed.
- **Update interval**

  Select the interval at which you require an update of the displayed values.

  Default setting: 30. Permitted range: 5...999

Overview of the interfaces

Here you can see the releasing access to the Web server via the Ethernet interfaces of the TIM.

You can activate access to the Web server of the TIM via HTTP/HTTPS for each individual Ethernet interface.

The settings for activation in the parameter groups "Web server access" and "Web server" are adopted reciprocally in the other parameter group.

4.12 Partner stations

The parameter group is only visible when the option "Import" is activated under "Basic settings" > Configuration (Page 58).

Importing the configuration data from STEP 7 V5

Follow the steps below to import the configuration data from STEP 7 V5

1. In your STEP 7 Professional project, select the TIM into which you want to import the SDBs from the STEP 7 V5 project.
2. Select the "Partner stations" parameter group.

3. Click the "Import partner configuration" button.

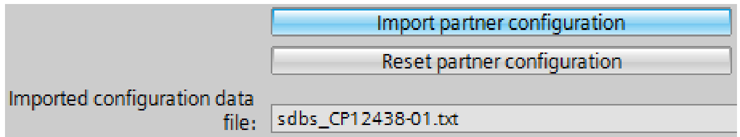


Figure 4-4 Importing the configuration data

The dialog for selecting the file with the configuration data opens

4. From the file system of the engineering station open the text file that you exported from the proxy from the STEP 7 V5 project.

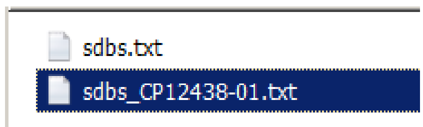


Figure 4-5 Selection of the file "sdbs_xxx.txt" with the configuration data from STEP 7 V5.

You can see the result of the import of partner data in the table "Partner stations".

In the "Imported configuration data file" output field, the file name of the imported SDBs is displayed.

With the "Reset partner configuration" button you can delete the imported data again. The partner-specific configuration data are then deleted.

Table "Partner stations"

This is where the data on the imported partner data is shown.

4.13 Communication with the CPU

Communication with the CPU

Using the first three parameters you specify the CPU access by the TIM in the CPU scan cycle. You will find the structure of the CPU scan cycle in the section Read cycle (Page 115).

The fourth parameter "Frame memory size" decides the size of the send buffer of the TIM for frames of data points that are configured as an event.

- **Cycle idle time**

  Wait time between two scan cycles of the CPU memory area

- **Max. number of write jobs**

  Maximum number of write jobs to the CPU memory area within a CPU scan cycle

- **Max. number of read jobs**

  Maximum number of low-priority read jobs from the CPU memory area within a CPU scan cycle.

- **Frame memory size**

  Here, you set the size of the frame memory for events (send buffer).

  The size of the frame memory is divided equally among all configured communications partners. You will find the size of the frame memory in the section Performance data and configuration limits (Page 25).

  You will find details of how the send buffer works (storing and sending events) as well as the options for transferring data in the section Process image, type of transmission, event classes (Page 113).

Watchdog bit

- **TIM monitoring**

  Via the watchdog bit the CPU can be informed of the status of the telecontrol communication of the TIM.

PLC tags for partner status / path status

Via the PLC tag that can be configured here, you can monitor the following information about the reachability of the communications partners:

- **Partner status**

  Reachability of the remote communications partner

- **Path status**

  Status of the connection path or the redundant connection paths to the remote communications partner

  For the communication response of the two connection paths see section Communication via the SINAUT ST7 protocol (Page 56).

For every configured communications partner to which a single or redundant telecontrol connection is created, you can create a PLC tag of the type Word.

Assignment of the PLC tags for partner status / path status

In the two bytes of the PLC tag of the data type Word (DB, memory bit, output) the following information is output:

- **Byte 0: Partner status**
- **Byte 1: Path status**

Byte 0 "Partner status"

Byte 0 codes information on the reachability of the communications partner, on the existing connections and connection paths and on the status of the send buffer of the TIM.

Table 4- 1 Assignment of byte 0: Meaning of the bit statuses

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Path redundancy	**Connection mode**	**Temporary connection ****	*(Reserved)*	**Message memory ***		**Path status**	**Partner status**
0: No redundancy 1: Redundancy exists	0: Permanent 1: Temporary	0: Partner not reachable 1: Partner accessible **	-	0: Send buffer OK 1: Memory allocation > 90% 3: overflow (memory allocation 100%)		0: Not all paths reachable 1: All paths reachable	0: Partner not reachable 1: Partner accessible

***** For information on the behavior of the frame memory, refer to the paragraph "Send buffer" in the section:

****** Partners that support temporary connections are set to "accessible" if the partner itself terminates the connection and there is no connection established.

Process image, type of transmission, event classes (Page 113)

Byte 1 "Path status"

Byte 1 shows the status of the connection path (configured connection) to the partner from the point of view of the local TIM.

A maximum of 2 paths (main and substitute path) to a partner can be configured, see section Creating telecontrol connections for the ST7 communication (Page 99).

Both connection paths must start or end on a local TIM.

The byte shows the following:

- The paths via which the partner can be reached.
- The path currently being used
- The TIM interface via which the main path was configured.
- The TIM interface via which the substitute path was configured.

The path of a connection is specified as a combination of the used interfaces of the TIM and the status of the path.

Byte assignment

Byte 1 is assigned as follows:

- Two bits for the interface of the main path
- Two bits for the interface of the substitute path
- Two bits for the status of the main path
- Two bits for the status of the substitute path

Table 4- 2 Assignment of byte 1:

Bits 6 + 7	Bits 4 + 5	Bits 2 + 3	Bits 0 + 1
Configured interface		**Path status**	
Coding for substitute path	Coding for main path	Substitute path (2nd path)	Main path (1st path)

- **Configured interface**

  The TIM interfaces "Ethernet 1" (IE1), "Ethernet 2" (IE2), "Ethernet 3" (IE3) and WAN1 are numbered through from 0 . 3 (decimal):

  – 0 = Ethernet interface IE1 (X1)

  – 1 = Ethernet interface IE2 (X2)

  – 2 = Ethernet interface IE3 (X3)

  – 3 = Serial interface WAN1 (X4)

Status of bit 5 (7)	Status of bit 4 (6)	Meaning
0	0	Coding for Ethernet interface X1 (decimal: No. 0)
0	1	Coding for Ethernet interface X2 (decimal: no. 1)
1	0	Coding for Ethernet interface X3 (decimal: no. 2)
1	1	Coding for serial interface X4 (decimal: no. 3)

- **Path status**

  – Main path = 1. Path (bits 0 + 1)

  – Substitute path = 2nd path (bits 2 + 3)

Status of bit 1 (3)	Status of bit 0 (2)	Meaning bit 1	Meaning bit 0
0	0	Bit 1: Path not current	Bit 0: Subscriber not reachable
0	1	Bit 1: Path not current	Bit 0: Subscriber reachable
1	0	Bit 1: Path current	Bit 0: Subscriber not reachable
1	1	Bit 1: Path current	Bit 0: Subscriber reachable

Coding options of byte 1

Same coding of the configured interface for the main and the substitute path means that there is no path redundancy (only one interface configured). The path status is output via the bits of the main path (1st path).

Table 4- 3 Coding options for the path status

Configured interface		Path status	
Coding for substitute path	**Coding for main path**	**Substitute path (2nd path)**	**Main path (1st path)**
0 0	0 0 (Coding for IE1)	Irrelevant (not redundant)	Status IE1
0 0	0 1 (Coding for IE2)	Status IE1	Status IE2
0 0	1 0 (Coding for IE3)	Status IE1	Status IE3
0 0	1 1 (Coding for WAN1)	Status IE1	Status WAN1
0 1	0 0	Status IE2	Status IE1
0 1	0 1	Irrelevant (not redundant)	Status IE2
0 1	1 0	Status IE2	Status IE3
0 1	1 1	Status IE2	Status WAN1
1 0	0 0	Status IE3	Status IE1
1 0	0 1	Status IE3	Status IE2
1 0	1 0	Irrelevant (not redundant)	Status IE3
1 0	1 1	Status IE3	Status WAN1
1 1	0 0	Status WAN1	Status IE1
1 1	0 1	Status WAN1	Status IE2
1 1	1 0	Status WAN1	Status IE3
1 1	1 1	Irrelevant (not redundant)	Status WAN1

4.14 Time-of-day synchronization

Basics of time-of-day synchronization

With telecontrol application that require time-of-day synchronization, you need to synchronize the time of day of the TIM regularly. If you do not synchronize the time of day, there may be deviations of several seconds per day in the time information of the stations.

The TIM can obtain the time of day from an external source (for the methods see below) and forward the time of day to the station or the connected WAN networks.

When using an external time source, the connected S7 station can obtain the current time of day both via the CPU as well as via the TIM or a CP.

Note

Recommendation: Time-of-day synchronization only by 1 module

Only have the time of day of the station from an external time source synchronized by a single module so that a consistent time of day is maintained within the station.

When the CPU takes the time from the TIM or from a CP, disable time-of-day synchronization of the CPU.

If you have the time synchronized on the TIM and the CPU via NTP, when possible use the same NTP server to maintain a consistent time of day within the station.

Synchronization method of the TIM

The TIM supports the following methods of time-of-day synchronization:

- **NTP / NTP (secure)**

  Network Time Protocol

  Time-of-day synchronization only via Ethernet

  The secure method NTP (secure) uses authentication with symmetrical keys according to the hash algorithms MD5 or SHA-1. In the global security settings, you can create and manage NTP servers of the type NTP (secure).

  Recommendation wit NTP:
  Synchronization with an external clock at intervals of approximately 10 seconds is recommended. This achieves as small a deviation as possible between the internal time and the UTC time.

  You can also change the time-of-day synchronization via NTP using the WBM, see section NTP (Page 267).

- **Time of day from subscriber in the network**

  In this case the TIM adopts the time of day from a subscriber in the connected network. Time-of-day masters can for example be:

  – A synchronized CPU

  – A subscriber with a time receiver

  – A master station PC (e.g. ST7cc/ST7sc) connected to the Ethernet network.

- **Setting the time of day manually using the WBM**

  If you have configured a time source for the TIM, you can also set the time via the WBM, see section System time (Page 267).

Forwarding time of day by the TIM

The TIM can forward its time of day as follows:

- **To connected networks**

  Configuration with "Time of day synchronization" > "Send time" or "Receive time"

  The procedure for configuration differs in Ethernet and classic WAN Networks, see below.

- **On the assigned CPU**

  – Configuration with "Time of day synchronization" > "Send time"

  – Configuration with "Communication with the CPU" > "Time to CPU"

    With this method the time-of-day is made available to the CPU via a PLC tag.

  Decide on one of the two methods for forwarding to the CPU and disable the other.

General information on configuriation

Parameter groups for time-of-day synchronization

For configuring the time-of-day synchronization the following parameter groups are available:

- **TIM**

  – **Receive time**

    Here you specify via which of the connected networks the TIM will receive the time of day. You configure this parameter group for the TIM modules with the network type "Node station" and "Station".

    Here is where you also configure the NTP servers if the TIM is to be synchronized directly via NTP. This is usually only one TIM that functions as time master in the network.

  – **Send time**

    Here you specify the networks on which the TIM will forward the time of day.

    You configure this parameter group for TIM modules with the network type "Master station" in other word on the TIM that functions as time master in the network.

- **Classic WAN network**

  For classic networks the "Time-of-day synchronization" is enabled in the parameter group of the same name. You also specify the synchronization cycle.

  The settings for synchronization are then adopted by all connected TIM modules.

  The send direction of the time-of-day messages is derived automatically from the node type of the connected interfaces:
  Master staion ⇒ Node station ⇒ Station

The settings for the network are not necessary with time-of-day synchronization via Ethernet.

Time-of-day concept

Before configuring time-of-day synchronization specify the following:

- Specify the time source in the network.
- Specify the time master in the network.
- Specify the network via which the time of day will be forwarded by the time master to the time slaves.

Configuring the synchronization via Ethernet

Time master

1. In the parameter group "Receive time" of the TIM to be time master configure the time source with one of the following options:
   - From NTP server
   - From local station

     (take the time from the assigned CPU)
   - Receive time from WAN

     (take the time from a network)
2. Configure the interface of the TIM via which time messages will be forwarded in the parameter group "WAN settings" as network type "Master station".

   The function is supported for the Ethernet interface with the MSC protocol and for the serial interface with the setting "network type" = "Neutral".
3. In the parameter group "Send time" for the interface from step 2 enable the option "Forward time to WAN".

   Via the connected network the time messages are forwarded in the network.
4. If necessary, enable the "To local station" option in the "Send time" parameter group if the assigned CPU should also be synchronized.

Time slaves

1. Configure the interfaces of the other TIM modules that will be time slaves in the parameter group "WAN settings" as network node type "Node station" or "Station".

   The function is supported for the Ethernet interface with the MSC protocol and for the serial interface with the setting "network type" = "Neutral".
2. Network the interfaces of the TIM modules involved with each other and with the interface of the time master.
3. For the stations set the parameters of time-of-day synchronization in the parameter group "Receive time".
4. If necessary, enable the "To local station" option in the "Send time" parameter group if the assigned CPU should also be synchronized.

Configuring the synchronization via classic WAN networks

TIM modules (time master and slaves)

1. In the parameter group "Receive time" of the TIM to be time master configure the time source with one of the following options:
   - From NTP server
   - From local station

     (take the time from the assigned CPU)
   - Receive time from WAN

     (take the time from a network)
2. Configure the interface of the master TIM as network node type "Master station".
3. Configure the interfaces of the other TIM modules (time slaves) as network node type "Node station" or "Station".
4. If necessary, enable the "To local station" option in the "Send time" parameter group of the stations if the assigned CPU should also be synchronized.

WAN network

1. In the parameter group "Time-of-day synchronization" of the network enable the option "Enable time-of-day synchronization for WAN".
2. There configure the required synchronization cycle.
3. Network the interfaces of all TIM modules involved with the WAN network.

   The settings configured for the WAN network are adopted in the following parameter groups of the connected TIM modules:
   - For the time master (Master station): "Send time" parameter group
   - For the time slaves (node stations / station) "Receive time" parameter group

4.15 Time stamp

Time stamp of the ST7 frames

The values of the data points are transferred by the communications moduel with a time stamp, see also section Creating telecontrol connections for the ST7 communication (Page 99), subsection "Connections to redundant control centers and other properties".

Format of the time stamp

The ST7 time stamp is coded in BCD format. The exception is the half byte (nibble) with the time status.

The structure of 8 bytes is assigned as follows:

Table 4- 4 Assignment of the structure of the time stamp

Byte no.	Content	
	High nibble	**Low nibble**
0	Year * 10	Year * 1
1	Month * 10	Month * 1
2	Day * 10	Day * 1
3	Hour * 10	Hour * 1
4	Minute * 10	Minute * 1
5	Second * 10	Second * 1
6	Millisecond * 100	Millisecond * 10
7	Millisecond * 1	Time status

Table 4- 5 The assignment of the half byte "time status" (low nibble of byte 7)

Bit No.	Value	Meaning
0	0	Time is invalid
	1	Time is valid
1	0	Standard time
	1	Daylight saving time
2		*Not used*
3		*Not used*

4.16 MSC authentication

MSC authentication on the different interfaces

When using the secure MSC protocol (MSC / MSCsec), the authentication data for the interfaces involved must be configured for the TIM as MSC station and as MSC node station.

Parameter groups

- MSC authentication - Ethernet interface

  Relevant when using MSC on the Ethernet interface X1
- MSC authentication - MD720

  Relevant on the serial interface when using MSC via mobile wireless

Parameter

- **User name**

  User name for communication using MSC
- **Password**

  Password for communication using MSC

The user name and password must be different for every MSC station in the network.

4.17 SMSC

Enabling SMSC

If you want to use SMS, enable the SMSC (Short Message Service Center).

Enter the phone number of the SMSC. You will receive the number from your service provider.

4.18 E-mail configuration

Configuring e-mails in STEP 7

In the "E-mail configuration" entry, you configure the protocol to be used and the data for access to the e-mail server.

In the message editor ("Messages" entry in STEP 7), you configure the individual e-mails, see section Messages (Page 127).

E-mail configuration

If you want to use the secure transfer of e-mails, the module must have the current date and the current time of day.

With the default setting of the SMTP port 25, the module transfers unencrypted e-mails.

If your e-mail service provider only supports encrypted transfer, use one of the following options:

- Port no. 587

  By using STARTTLS, the module sends encrypted e-mails to the SMTP server of your e-mail service provider.

  Recommendation: If your e-mail provider offers both options (STARTTLS / SSL/TLS), you should use STARTTLS with port 587.

- Port no. 465

  By using SSL/TLS (SMTPS), the module sends encrypted e-mails to the SMTP server of your e-mail service provider.

Ask your e.mail service provider which option is supported.

Importing the certificate with encrypted transfer

To be able to use encrypted transfer, you need to load the certificate of your e-mail account in the certificate manager of STEP 7. You obtain the certificate from your e-mail service provider.

Use the certificate by taking the following steps:

1. Save the certificate of your e-mail service provider in the file system of the engineering station.
2. Import the certificate into your STEP 7 project with "Global security settings > Certificate manager".
3. Use the imported certificate with every module that uses encrypted e-mails via the "Certificate manager" table in the local "Security“ parameter group.

For the procedure, refer to the section Certificate manager (Page 94).

4.19 SNMP

SNMP

You will find the range of functions of the device for SNMP in the section SNMP (Page 288).

If the security functions are enabled, you have the following selection and setting options.

SNMP

- **"Enable SNMP"**

  If the option is enabled, communication via SNMP is released on the device. As default, SNMPv1 is enabled.

  If the option is disabled, queries from SNMP clients are not replied to either via SNMPv1 or via SNMPv3.
- **"Use SNMPv1"**

  Enables the use of SNMPv1 for the device. For information on the configuration of the required community strings see below (SNMPv1).
- **"Use SNMPv3"**

  Enables the use of SNMPv3 for the device. For information on the configuration of the required algorithms see below (SNMPv3).

SNMPv1

The community strings need to be sent along with queries to the device via SNMPv1.

Note the use of lowercase letters with the preset community strings!

- **"Reading community string"**

  The string is required for read access.

  Leave the preset string "public" or configure a string.

- **"Allow write access"**

  If the option is enabled write access to the device is released and the corresponding community string can be edited.

- **"Writing community string"**

  The string is required for write access and can also be used for read access.

  Leave the preset string "private" or configure a string.

Note

Security of the access

For security reasons, change the preset and generally known strings "public" and "private".

SNMPv3

The algorithms need to be configured for encrypted access to the device via SNMPv3.

- **"Authentication algorithm"**

  Select the authentication method to be used from the drop-down list.

- **"Encryption algorithm"**

  Select the encryption method to be used from the drop-down list.

User management

In the user management that you will find in the global security settings, assign the various users their role.

Below the properties of the roles you can see the rights list of the particular role, for example the various types of access using SNMP. For new roles, you can freely configure individual rights.

You will find information on users, roles and the password policy in the information system of STEP 7.

4.20 Security and protection

4.20.1 Certificate manager

Assignment of certificates

If you use communication with authentication for the module, for example SSL/TLS for secure transfer of e-mails, certificates are required. You need to import certificates of non-Siemens communications partners into the STEP 7 project and download them to the module with the configuration data:

1. Import the certificates of the communications partners using the certificate manager in the global security settings.
2. Then assign the imported certificates to the module in the table below the local security settings of the module.

For a description of the procedure, refer to the section Handling certificates (Page 94).

You will find further information in the STEP 7 information system.

4.20.2 Handling certificates

Certificate for authentication

If you have configured secure communication with authentication for the module, own certificates and certificates of the communications partner will be required for communication to take place.

All nodes of a STEP 7 project with enabled security functions are supplied with certificates. The STEP 7 project is the certification authority.

For the secure transfer of e-mails via SSL/TLS and SSL certificate is created for the module. It is visible in STEP 7 in "Global security settings > Certificate manager > Device certificates".

The table "Device certificates" shows the issuer, validity, use of a certificate (service/application) and the use of a key. You can call up further information about a certificate by selecting the certificate in the table and selecting the shortcut menu "Show".

The table also shows all other certificates generated by STEP 7 and all imported certificates.

If the module communicates with non-Siemens partners when the security functions are enabled, the relevant certificates of the communications partners must be exchanged. To do this, follow the steps below:

1. Importing third-party certificates from communications partners

   ⇒ Global security settings of the project (certificate manager)

2. Assigning certificates locally

   ⇒ Local security settings of the module ("Certificate manager" table)

These two steps are described in the next two sections.

Importing third-party certificates from communications partners

Import the certificates of the communications partners of third-party vendors using the certificate manager in the global security settings of the STEP 7 project. Follow the steps outlined below:

1. Save the third-party certificate in the file system of the PC of the connected engineering station.
2. In the STEP 7 project open the global certificate manager:

   Global security settings > Certificate manager
3. Open the "Trusted certificates and root certification authorities" tab.
4. Click in a row of the table can select the shortcut menu "Import".
5. In the dialog that opens, import the certificate from the file system of the engineering station into the STEP 7 project.

Assigning certificates locally

To be able to use an imported certificate for the TIM, you need to specify it in the "Security" parameter group of the TIM. Follow the steps outlined below:

1. In the STEP 7 project select the module.
2. Navigate to the parameter group "Security > Certificate manager".
3. In the table, double-click on the cell with the entry "<Add new>".

   The "Certificate manager" table of the Global security settings is displayed.
4. In the table. select the required third-party certificate and to adopt it click the green check mark below the table.

   The selected certificate is displayed in the local table of the module.

Only now will the third-party certificate be used for the module.

Exporting certificates for applications of third-party vendors

For communication with applications of third-party vendors, the third-party application generally also requires the certificate of the module.

You export the certificate of the module for communications partners from third-party vendors in much the same way as when importing (see above). Follow the steps outlined below:

1. In the STEP 7 project open the global certificate manager:

   Global security settings > Certificate manager
2. Open the "Device certificates" tab.

3. In the table select the row with the required certificate and select the shortcut menu "Export".
4. Save the certificate in the file system of the PC of the connected engineering station.

Now you can transfer the exported certificate of the module to the system of the third-party vendor.

Change certificate: Subject Alternative Name

STEP 7 adopts the properties "DNS name", "IP address", and "URI" from the parameter "Subject Alternative Name" (Windows: "Alternative applicant name") from the STEP 7 configuration data.

You can change this parameter of a certificate inn the certificate manager of the global security settings. To do this, select the a certificate in the table of device certificates and call the shortcut menu "Renew". Properties of the parameter "Alternative name of the certificate owner" changed in STEP 7 are not adopted by the STEP 7 project.

4.20.3 Protection

Protection functions

The module provides various access levels to restrict access to certain functions.

NOTICE
Configuring an access level does not replace the know-how protection Configuring access levels prevents unauthorized changes to the module by restricting the download rights. This does not, however, provide write or read protection for blocks on a memory card. Use the know-how protection to protect the code of blocks on the memory card.

The table of access levels

You configure the access levels in the table. The green check mark in the columns on the right of the particular access level indicate the maximum possible operations without knowing the password for this access level.

The default access level is "Full access (no protection)". Every user can read and modify the configuration. No password has been configured and no password is required for online access.

You can configure the following access levels:

- **Full access (no protection)**

  The configuration and the blocks can be read and modified by anybody.

- **Read access**

  With this access level, without entering the password, only read access to the hardware configuration and the blocks is possible; in other words, you cannot download the blocks or hardware configuration to the TIM without entering the password. Without the password, writing test functions and firmware updates are also not possible.

- **No access (complete protection)**

  If the module is completely protected, neither read nor write access to the hardware configuration and blocks is possible.

If you want to use the functions of the unmarked access levels, you will need to enter a password.

With the legitimization provided by using the password, you once again have full access to the module.

Behavior of a password-protected module during operation

Protection of the module is effective after the settings have been loaded on the module.

Before an online function is executed, a check is made to establish whether or not it is permitted. If there is password protection, you will be prompted to enter the password.

Example:
The module was configured for read access and you want to use the "Modify tags" function. Since this is write access, the configured password must be entered before the function can be executed.

The functions protected by the password can only be executed by one PG/PC at any one time. Another PG/PC cannot log on.

The access rights to the protected data apply for the duration of the online connection or until the access rights are canceled again with "Online > Delete access rights".

Each access level allows unrestricted access to certain functions even without entering a password, for example identification using the "Accessible devices" function.

4.20.4 Configuring access protection

Configuration

You can enter several passwords setting up different access rights for different user groups.

The passwords are entered in the table so that precisely one access level is assigned to each password.

The "Access level" column shows how the password takes effect.

Example:

You select the access level "No access (complete protection)" for the module and enter your own password for each of the access levels higher up the table.

For users that do not know any of the passwords, the module is completely protected.

For users who know one of the set passwords the effect depends on the table row in which the password is located:

- The effect of the password in row 1 "Full access (no protection)" is as if the CP was unprotected. Users that know this password have unrestricted access to the module.
- The effect of the password in row 2 "Read access" is as if the module was write-protected. Despite knowing the password, users that know this password only have read access to the module.
- The effect of the password in row 3 "No access (complete protection)" is as if the CP was write and read protected. Users that know this password only have read access to the module.

Procedure

Follow the steps below to set the parameters for the access levels of the module:

1. Open the module properties in the Inspector window.
2. Open the "Protection" entry in the navigation panel.
   A table with the possible access levels is displayed in the Inspector window.
3. Select the required access level in the first column of the table. The green check mark in the columns on the right of the particular access level indicate which operations are still possible without entering the password.
4. If you have selected an access level other than "Full access":
   – Assign a password for full access in the "Password" column in the first row (full access).
   – Repeat the selected password in the "Confirm password" column to protect against incorrect entries.
   – Make sure that the password is adequately secure; in other words, that it does not include a pattern that can be machine read!
   – The entry of the password in the first row "Full access (no protection)" is obligatory and allows a user who knows the password unrestricted access to the module, regardless of the selected access level.
5. As necessary, assign other passwords to the required access levels if the selected access level permits this.
6. Download the hardware configuration so that the access level takes effect.

Result

The hardware configuration and the blocks are protected from unauthorized online access according to the set access level. If an operation cannot be executed without a password due to the set access level, a dialog appears prompting entry of a password.

4.21 Creating telecontrol connections for the ST7 communication

Single and redundant connections

For ST7 communication of a communication module, you need to create at least one telecontrol connection in the "Network data" editor.

If two subscribers can be reached via different networks, to establish path redundancy for modules with multiple ST7 capable interfaces you create a maximum of two connections between the two subscribers. You specify redundant connections in the "Connection path" table (see below).

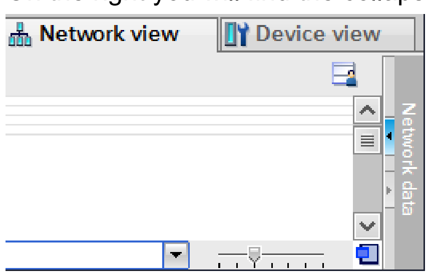
Not all communication modules have serial interfaces or only one Ethernet port with ST7 capability. These modules cannot use redundant paths.

Opening the editor "Network data" > "Telecontrol" tab

To open the editor, follow the steps below:

1. Open the network view of the project.

   On the right you will find the collapsed "Network data" editor.

2. Open the "Network data" editor using the arrow symbol.

   The editor is displayed with several tabs, on the left the "Network overview" tab.

3. Expand the editor until the "Telecontrol" tab appears.

   In this tab you configure the telecontrol connections.

Creating telecontrol connections for the ST7 communication of the module

For communication between two ST7 subscribers, you require a telecontrol connection. To create the connection, follow the steps below:

1. In the "Starting point" box, click on a free row in the "Telecontrol" tab.

   The drop-down list shows the configurable starting points that can use the ST7 protocol:

   – S7 stations: Assigned CPU

   – PC stations: Application (e.g. ST7cc / ST7sc)

   For connections between an application on the master station PC and an S7 station, always select the application as the starting point because it can only be selected in the list of starting points.

2. Select the starting point of the connection by double-clicking.

   The connection will then be included in the list of telecontrol connections.

   The name of the connection is preset, but you can change it.

3. Now in the "Endpoint" box select the connection partner from the drop-down list.

Boxes with a missing or bad configuration are shown on a red background, in the example the representation "TC_Connection_3".

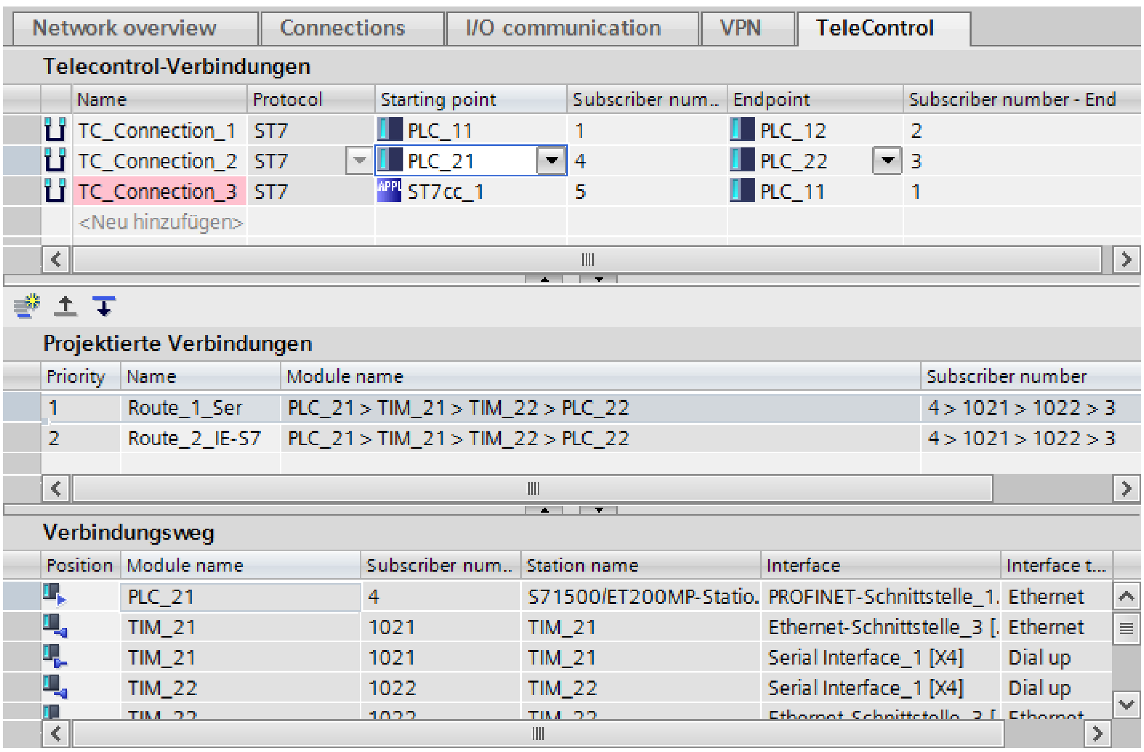


Figure 4-6 "Network data" editor, "Telecontrol" tab

Specification of the connection sequence

Then specify the connection sequence of the created telecontrol connection.

1. Above the second table "Configured connection paths" click the paths symbol .

   The dialog "Add connection paths" opens and the possible connection path are searched for.

2. Select one of the possible connection paths in the top table.

   The connection path(s) are shown with details in the second table.

   – If several connections are displayed in the upper table, you can select one or two redundant connections.

   – If no connection is displayed in the upper table, there is a configuration error in the corresponding stations or networks.

     In this case, close the dialog with the "Close" button and complete the configration.

3. If you want to choose the selected connection path as a "configured connection" click on "Add".

   "Information" in the lower part of the dialog shows whether the connection was added or it is already configured.

4. Close the dialog with the "Close" button if the configured connections correspond to the demands of the project.

   The connection details of every configured connection are displayed in the lower table "Connection path".

The "Configured connection paths" and "Connection path" tables

"Configured connection paths" table

This table shows the configured connections (connection paths) of a telecontrol connection.

Projektierte Verbindungen

Priority	Name	Module name	Subscriber number	Networks
1	Route_1	ST7cc_1 > TIM_11 > PLC_11	5 > 1001 > 1	PN > PN/IE_1
2	Route_2	ST7cc_1 > TIM_12 > TIM_11 > PLC_11	5 > 1002 > 1001 > 1	PN > > WAN_1 > PN/IE_1

Verbindungsweg

Position	Module name	Subscriber num..	Station name	Interface	Interface type	Mode	Network
	ST7cc_1	5	PC_1	CP 1616_1, PROFINET int..	Ethernet	None	PN
	TIM_12	1002	TIM_12	Ethernet-Schnittstelle_1 [..	MSC	Station	
	TIM_12	1002	TIM_12	Serial Interface_1 [X4]	Dedicated line	Station	WAN_1
	TIM_11	1001	TIM_11	Serial Interface_1 [X4]	Dedicated line	Master	WAN_1
	TIM_11	1001	TIM_11	Ethernet-Schnittstelle_3 [..	Ethernet	Neutral	PN/IE_1
	PLC_11	1	S71500/ET200MP-Station_1	PROFINET-Schnittstelle_1..	Ethernet	Neutral	PN/IE_1

Figure 4-7 Redundant connection with incorrectly configured connection path (Route_2)

Connections shown on a red background are invalid.

In the example the PC station was connected via the PC CP to the MSC network between TIM_11 and TIM_12 which is not allowed.

"Connection path" table

When checking the configured connections. the "Connection path" table supports you.

For every configured connection, the detailed connection path is shown here.

The "Position" column shows a station symbol with a symbol for the connection point (starting point, node, endpoint). Based on the color of the symbols for the connection endpoint you can recognize connections that are not permitted.

- Blue symbols for the connection point: Valid connections
- Red symbols for the connection point: Invalid connections

Bad connections

In the example the connection path "Route_2" was originally created from the application "ST7cc_1" (PC_1) via the MSC network to TIM_12. This continued via the dedicated line network from TIM_12 to TIM_11.

A connection path from the application via an MSC network or via modules with ST7 capability is not permitted.

The connection of the PC station to the MSC network was deleted in the meantime, so that the connection points "ST7cc_1" and "TIM_12" are invalid and their symbols are red.

Redundant telecontrol connections

When different networks between two subscribers have been created in the network configuration the different connection paths are shown in the "Add connection paths" table.

If you only want to use one connection, select only this connection in the table and click "Add".

If you only want to establish a redundant connection, select two connection paths in the table and click "Add". You specify the main and substitute paths between the subscribers after closing the dialog using the "Configured connections" table.

Main and substitute path of a redundant connection

When you have created two connection paths for a telecontrol connection, the first of the two is the main path and the second the substitute path. This is displayed in the "Priority" column of the "Configured connections" table.

You can change the priority of the two connection paths later, i.e. the specification of the main and substitute path. The arrow keys serve this purpose:

Delete invalid or redundant connections

If you have non-permitted or unwanted redundant connections you need to delete one connection. Follow the steps outlined below:

1. To delete a connection in the table "Configured connections", select the connection you do not need.
2. Delete the connection using the shortcut menu "Delete".

Rules for connection configuration

Note the following rules for connection configuration:

- A connection via an inconsistent network is invalid. Examples:
  – Subscriber with incompatible modems
  – Incompatible settings of two modems in a connection
  – Incompatible settings between modem and network parameters
  – Connections via nodes that are not configured as node stations.
- Two connections to one partner (endpoint) via the same interface of a module (start point) are not permitted.

- An MSC connection can only run via the first Ethernet interface [X1] of a TIM or via the serial interface of a module with the setting "IP-based".
- Connections between a PC application and a module can only be of the Network type "Neutral" (S7 connections).
- For connections between a redundant PC application and a module, you need to create two connections.

  For a redundant ST7cc control center, in STEP 7 create two PC stations each with an ST7cc application (see below).

  Only in this way can you configure a connection from each application.
- A connection via a network in which two MD3 modems communicate with each other and that are configured with 1200 bps / half duplex / AT mode is invalid.

Connections to redundant control centers and other properties

If you select a connection in the table "Telecontrol connections" at the top in the "Network data" editor, three parameter groups are displayed in the "Properties" tab of the Inspector window. Here you can check and, if necessary, correct the connection configuration and configure the following properties:

- **General**

  Here, you will find the basic properties of a telecontrol connection.

  You can change the subscriber number of the connection starting point and endpoint. This affects the configuration data of the subscriber.
- **Redundant ST7cc/sc**

  If you use a redundant control center in your project (e.g. ST7cc or ST7sc), you must first create two PC stations with CP and application and connect these to the communication module via Ethernet.

  Then you create a telecontrol connection from the application of a PC station (start point) and the communication module (end point).

  Only afterwards do you create the connection to the second application in the redundancy group in this telecontrol connection.

  – Add redundant ST7cc/cc partner

    Activate the option for the selected telecontrol connection between application and module.

  – Redundant application

    Select the second application from the "Redundant application" drop-down list.

- **Destination subscriber properties**

  The parameter group relates to the destination subscriber that was configured as the endpoint of the connection.

  – **General request supervision time**

    This is the maximum time that may be required by a destination subscriber (communication module capable of ST7) to respond fully to a general request (GR).

    If the GR response has not arrived completely at the requesting partner when the supervision time has expired, a message is entered in the diagnostics buffer of the module.

    The general request supervision time is 900 seconds.

  – **Frames to destination subscriber with time stamp**

    Regardless of the configuration all frames are sent to the destination subscriber with a time stamp.

4.22 Data points

4.22.1 Data point configuration

Data point-related communication with the CPU

No program blocks need to be programmed for telecontrol modules with data point configuration to transfer user data between the station and communications partner.

The data areas in the memory of the CPU intended for communication with the communications partner are configured data point-related on the module. Each data point is linked to a PLC tag or the tag of a data block.

Requirement: Created PLC tags and/or data blocks (DBs)

PLC tags or DBs must first be created on the CPU to allow configuration of the data points.

The PLC tags for data point configuration can be created in the standard tag table or in a user-defined tag table. All PLC tags intended to be used for data point configuration must have the attribute "Visible in HMI".

Address areas of the PLC tags are input, output or bit memory areas on the CPU.

Note

Number of PLC tags

Remember the maximum possible number of PLC tags the can be used for data point configuration in the section Performance data and configuration limits (Page 25).

The formats and S7 data types of the PLC tags that are compatible with the protocol-specific data point types of the module can be found in the section Datapoint types (Page 111).

Access to the memory areas of the CPU

The values of the PLC tags or DBs referenced by the data points are read and transferred to the communications partner by the module.

Data received from the communications partner is written by the module to the CPU via the PLC tags or DBs.

Configuring the data points and messages in STEP 7

You configure the data points in STEP 7 in the data point and message editor. You can open both editors alternatively as follows:

- Selecting the communication module

  Shortcut menu "Open the data point and messages editor"

- Via the project navigation:

  Project > directory of the relevant station > Local modules > required communication module

  By double-clicking on the entry, the data point or message editor opens.

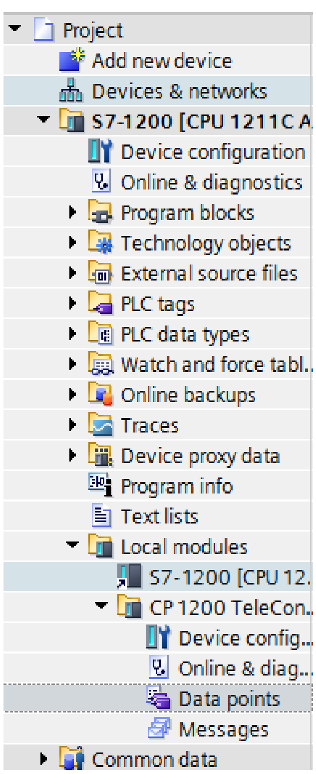


Figure 4-8 Configuring data points and messages

After opening the editor window using the two entries to the right above the table, you can switch over between the data point and message editor.

Figure 4-9 Switching over between the two editors

Creating obects

With the data point or message editor open, create a new object (data point / message) by double clicking "<Add object>" in the first table row with the grayed out entry.

A preset name is written in the cell. You can change the name to suit your purposes but it must be unique within the module.

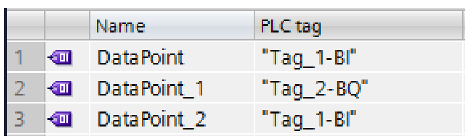


Figure 4-10 Data point table

You configure the remaining properties of every object using the drop-down lists of the other table columns and using the parameter boxes shown at the bottom of the screen.

Assigning data points to their data source

After creating it, you assign a new data point to its data source. Depending on the data type of the data point a PLC tag can serve as the data source.

For the assignment you have the following options:

- Click on the table symbol in the cell of the "PLC tag" column.

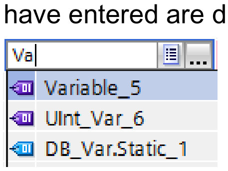
  All configured PLC tags and the tags of the created data blocks are displayed. Select the required data source with the mouse or keyboard.

- Click the symbol .

  A selection list of the configured PLC Tags and the blocks is displayed. From the relevant table, select the required data source.

- In the name box of the PLC tag, enter part of the name of the required data source.

  All configured PLC tags and tags of the data blocks whose names contain the letters you have entered are displayed.

  Select the required data source.

Note

Assignment of parameter values to PLC tags

The mechanisms described here also apply when you need to assign the value of a parameter to a PLC tag. The input boxes fro the PLC tag (e.g.: PLC tag for partner status support the functions described here for selecting the PLC tag.

Arranging and copying objects

As with many other programs in the data point or message editor you can also arrange the columns, sort the table according to your requirements and copy and insert objects.

- Arrange columns

  If you click on a column header with the left mouse button pressed, you can move the column.

- Sorting objects

  If you click briefly with the left mouse button on a column header, you can sort the objects of the table in ascending or descending order according to the entries in this column. The sorting is indicated by an arrow in the column header.

  After sorting in descending order of a column the sorting can be turned off by clicking on the column header again.

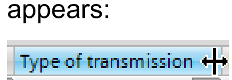
- Adapting the column width

  You can reach this function with the following actions:

  – Using the shortcut menu that opens when you click on a column header with the right mouse key.

    "Optimize width", "Optimize width of all columns"

  – If you move the cursor close to the limit of a column header, the following symbol appears:

    When it does, click immediately on the column header. The column width adapts itself to the broadest entry in this column.

- Showing / hiding columns

  You call this function using the shortcut menu that opens when you click on a column header with the right mouse key.

- Copying, pasting, cutting and deleting objects

  If you click in a parameter box of an object in the table with the right mouse key, you can use the functions named with the shortcut menu (copy, paste, cut, delete).

  You can paste cut or copied objects within the table or in the first free row below the table.

Exporting and importing data points

To simplify the engineering of larger plants, you can export the data points of a configured module and import them into other modules in the project. This is an advantage particularly in projects with many identical or similar stations or data point modules.

The export / import function is available when you select the module for example in the network or device view and select the relevant shortcut menu.

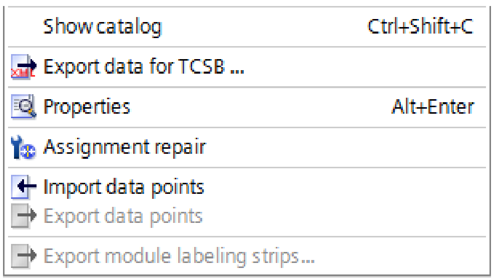


Figure 4-11 Shortcut menu of the module

When it is exported the data point information of a module is written to a CSV file.

Export

When you call the export function, the export dialog opens. Here, you select the module or modules of the project whose data point information needs to be exported. When necessary, you can export the data points of all modules of the project together.

In the export dialog, you can select the storage location in the file directory. When you export the data of a module you can also change the preset file name.

When you export from several modules, the files are formed with preset names made up of the station name and module name.

The file itself contains the following information in addition to the data point information:

- Module name
- Module type
- CPU name
- CPU type

Editing the export files

You can edit the data point information in an exported CSV file. This allows you to use this file as a configuration template for many other stations.

If you have a project with many stations of the same type, you can copy the CSV file with the data points of a fully configured module for other as yet unconfigured stations and adapt individual parameters to the particular station. This saves you having to configure the data points for every module in STEP 7. Instead, you simply import the copied and adapted CSV file to the other modules of the same type. When you import this file into another module, the changed parameter values of the CSV file are adopted in the data point configuration of this module.

The lines of the CSV file have the following content:

- Line 1: ,Name,Type,

  This line must not be changed.
- Line 2: PLC,<CPU name>, <CPU type>,

  Meaning: PLC (designation of the station class), CPU name, CPU type

  Only the elements <CPU name> and <CPU type> may be changed.

  The CPU type must correspond exactly to the name of the CPU in the catalog.
- Line 3: Module,<module name>,<module type>,

  Meaning: Module (Designation of the module class), module type, module name

  Only the elements <module name> and <module type> may be changed.

  Be careful when changing the module names if you want to import data points into several modules (see below).

  The module type must correspond exactly to the name of the module in the catalog.
- Line 4: Parameter names (English) of the data points

  This line must not be changed.
- Lines 5..n: Values of the parameters according to line 4 of the individual data points

  You can change the parameter values for the particular station.

Importing into a module

Before importing the data points make sure that the PLC tags required for the data points have been created.

Note that when you import a CSV file all the data points existing on the module will be deleted and replaced by the imported data points.

Select a module and select the import function from the shortcut menu of the module. The import dialog opens in which you select the required CSV file in the file directory.

If the information on the assignment of the individual data points to the relevant PLC tags matches the assignment in the original module, the data points will be assigned to the corresponding PLC tags.

When you import data points into a module, but some required PLC tags have not yet been created in the CPU, the corresponding data point information cannot be assigned. In this case, you can subsequently create missing PLC tags and them assign them the imported data point information. The "Assignment repair" function is available for this (see below).

If the names of the PLC tags in the module into which the import is made have different names than in the module that exported, the corresponding data points cannot be assigned to your PLC tags.

Importing into several modules

You can import the data points from several modules into the modules of a different project. To do this in the import dialog select all the required CSV files with the control key.

Before importing the data points, make sure that the respective stations have been created with CPUs of the same name, modules of the same name and PLC tags of the same name.

When you import the corresponding stations of the project are searched for based on the module names in the CSV files. If a target station does not exist in the project or the module has a different name, the import of the particular CSV file will be ignored.

Restrictions for the import of data points

In the following situations the import of data points will be aborted:

- An attribute required by the module is missing in the CSV file to be imported.

  Example: If a data point to be imported uses a time trigger, the import will be aborted if no time-of-day synchronization was configured for the module.
- The telecontrol protocol used by the module differs from that of the original module.

Only when importing into several modules:

- The import is aborted when a module or CPU name is different from the data in the CSV file.

Note:
Modules with the same telecontrol protocol are compatible with each other:

- TeleControl Basic

  All SIMATIC NET modules with the TeleControl Basic protocol:

  CP 1243-1, CP 1242-7 GPRS V2, CP 1243-7 LTE, CP 1542SP-1 IRC
- ST7

  CP 1243-8 IRC, CP 1542SP-1 IRC, ST7 TIM
- DNP3

  CP 1243-1, CP 1243-8 IRC, TIM modules capable of DNP3
- IEC

  CP 1243-1, CP 1243-8 IRC

Data points can be imported and exported between compatible modules.

Assignment repair

If you have named the PLC tags in a station into which you want to import differently from the station from which the CSV file was exported, the assignment between data point and PLC tag is lost when you import.

You then have the option to either rename the existing PLC tags appropriately or add missing PLC tags. You can then repair the assignment between unassigned data points and PLC tags. This function is available either via the shortcut menu of the module (see above) or with the following icon to the upper left in the data point editor: 

If a PLC tag with a matching name is found for a data point by the repair function, the assignment is restored. However the data type of the tag is not checked.

After the assignment repair make sure that you check whether the newly assigned PLC tags are correct.

4.22.2 "General" tab

General

You will also the most important parameters in the first tab of the data point editor in the data point table.

- **Data source**
  - For the assignment of the PLC tabs, see section Data point configuration (Page 104).
  - For the data point type, see Datapoint types (Page 111).
- **Send parameters**

  See following sections
  - For the type of transmission, see section Process image, type of transmission, event classes (Page 113).
  - For the read cycle see section Read cycle (Page 115).
  - High priority

    If the option is enabled, the transfer has higher priority (important process data). The values of the data point are transferred before those of lower priority data points.

    In dial-up networks the high priority leads to connection establishment only when the value of the data point is transferred with the transfer mode “Spontaneous (direct transfer)“ (see “Trigger” tab).
- **Object**
  - Object number

    The object number of the data point must be unique. The maximum permitted number of data points per SINAUT object must not be exceeded.

    Refer to the column “Number of data points per SINAUT object” in the section Datapoint types (Page 111).
  - Object channel

    Channel number of the SINAUT object
  - Partner object number

    Object number of the data point on the partner.

    Per object number, you need to specify a unique object number of the communications partner. This results in unique pairs of "Object number" and "Partner object number".

    See also section Partner stations (Page 127).

4.22.3 Datapoint types

During the configuration of the user data to be transferred by the module, each data point is assigned a protocol-specific data point type. The data point types supported by the module along with the compatible S7 data types are listed below. They are grouped according to format (memory requirements).

Note

Effect of the change of arrays for data points

If an array is modified later, the data point must be recreated.

Data point types

The direction of the data transfer is specified in the "Data point type" column.

- Input = Monitoring direction

  The objects (right column) for the monitoring direction have the abbreviation "_S".
- Output = Control direction

  The objects (right column) for the control direction have the abbreviation "_R".

Table 4- 6 Supported data point types and compatible S7 data types

Format (memory requirements)	Data point type	S7 data type	Operand area	Number of data points per object	Object
Bit	Digital input	Bool	I, Q, M, DB	8	Bin08X_S
	Digital output	Bool	Q, M, DB	8	Bin08X_R
Byte	Digital input	Byte, USInt	I, Q, M, DB	4	Bin04B_S
	Digital output	Byte, USInt	Q, M, DB	4	Bin04B_R
	Command output	Byte, USInt	Q, M, DB	1	Cmd01B_R
	Command input [1)]	Byte, USInt	I, Q, M, DB	1	Cmd01B_S [1)]
Integer with sign (16 bits)	Analog input	Int	I, M, DB	4	Ana04W_S
	Mean value input	Int	I, M, DB	4	Mean04W_S
	Analog output	Int	Q, M, DB	4	Ana04W_R
	Mean value output	Int	Q, M, DB	4	Mean04W_R
	Setpoint output [2)]	Int	Q, M, DB	1	Set01W_R
	Setpoint input [1)]	Int	I, Q, M, DB	1	Set01W_S [1)]
Counter (16 bits)	Counter input	UInt, Word	I, Q, M, DB	1	Cnt01D_S
	Counter input	UInt, Word	I, Q, M, DB	4	Cnt04D_S
	Counter output [1)]	UDInt, DWord	Q, M, DB	1	Cnt01D_R [1)]
	Counter output [1)]	UDInt, DWord	Q, M, DB	4	Cnt04D_R [1)]
Floating-point number (32 bits)	Analog input	Real	M, DB	4	Ana04R_S
	Analog output	Real	Q, M, DB	4	Ana04R_R
Data block (4 .. 48 bytes)	Data input	ARRAY [1...12] of DInt / UDInt / DWord / Real [3)]	DB	12	Dat12D_S
	Data output		DB	12	Dat12D_R
	Parameter output [2)]		DB	12	Par12D_R
	Parameter input [1)]		DB	12	Par12D_S [1)]

[1)] Only ST7-TIM as master station or node station

[2)] See below, section "1: Mirroring back”

[3)] See below, section "2: Data block"

Mirroring

The mirroring back function using the "Value monitoring" parameter can be configured for the following data point types:

- Setpoint output
- Parameter output

The local values of the data points of this type can be monitored for change and the changes transferred to the master with the Value monitoring function.

Changing a local value can, for example, be caused by manual operator input on site.

To allow the value resulting from local events or interventions to be transferred to the master station, a mirroring back channel is generated for the relevant data point with the "Value monitoring" function via which the locally changed value is mirrored back.

Remember that to use the mirror back function, you need to interconnect the local values in the controller with the relevant PLC tag of the data point.

Arrays

With the ARRAY data type, blocks of data from contiguous memory areas up to a size of 4 .. 48 bytes can be transferred.

Compatible components of ARRAY are DInt, UDInt, DWord or Real. The components within an array must be of the same type.

4.22.4 Process image, type of transmission, event classes

Storage of values

As a rulethe values of all data points are stored in the image memory of the module. Values in the image memory are transferred only after being called by master station TIM.

Events are also stored in the send buffer and can be transferred unsolicited.

The image memory, the process image of the module

The image memory is the process image of the TIM. All the current values of the configured data points are stored in the image memory. New values of a data point overwrite the last stored value in the image memory.

The values are sent only after a query by the communications partner - see below "Transfer after call" in the "Types of transmission" section - or along with a message from the send buffer that needs to be transferred immediately.

The send buffer

The send buffer of the TIM is the memory for the individual values of data points that are configured as an event. You will find the size of the send buffer in the manual of the relevant module.

The capacity of the send buffer is divided up equally for all enabled partners.

If the connection to a communications partner is interrupted, the individual values of the events are retained in the buffer. When the connection returns, the buffered values are sent. The frame memory operates chronologically; in other words, the oldest frames are sent first (FIFO principle).

If a frame was transferred to the communications partner, the transferred value is deleted from the send buffer.

If frames cannot be transferred for a longer period of time and the send buffer is threatening to overflow, the "Forced image mode" applies.

- If the send buffer reaches a fill level of 90%, the module changes to the forced image mode. New values from data points configured as an event are no longer added to the send buffer but rather they overwrite older existing values in the image memory.
- When the connection to the communications partner returns, the TIM changes back to the send buffer mode if the fill level of the send buffer has fallen below 50%.

Saving the data point values

As a rule, the values of data points are stored in the image memory of the moodule and transferred only when queried by the communications partner.

Events are also stored in the send buffer and can be transferred unsolicited.

Data points are configured as a static value or as an event using the "Type of transmission" parameter (see below):

- **Static value (no event)**

  Static values are entered in the image memory (process image).

  Static values correspond the following type of transmission "Transfer after call (class 0)".

- **Event**

  The values of data points configured as an event (triggered type of transmission) are also entered in the image memory of the module. The values are also entered in the send buffer.

Types of transmission and event classes

The following types of transmission are possible:

- **Transfer after call (class 0)**

  The current value of the data point is entered in the image memory. New values of a data point overwrite the last stored value in the image memory.

  After being called by the communications partner, the current value at this time is transferred.

- **Triggered**

  Data points are configured as an event using a triggered type of transmissopn. The values of these data points are entered in the image memory and also in the send buffer.

  The values of an event are saved as soon as the configured trigger conditions are met.

  The following event classes are available:

  - **Every value triggered**

    Each value change is entered in the send buffer in chronological order.

  - **Current value triggered**

    Only the last, current value is entered in the send buffer. It overwrites the value stored there previously.

  For information on the different trigger types, see section "Trigger“ tab (Page 116).

4.22.5 Read cycle

Input data points are assigned to the read cycle of the CPU in the data point configuration in the "General > Read cycle" tab.

Structure of the CPU scan cycle

The cycle with which the transferring module (TIM) scans the memory area of the CPU is made up of the following phases:

- **High-priority read jobs**

  (Fast cycle)

  For all data points with the assignment "Fast cycle" the PLC tags are read in every scan cycle.

  As a rule, it is sufficient to assign only data to be acquired quickly, such as alarms and contact changeover messages as well as command, setpoint and parameter objects for the 1oon check, to the fast cycle.

  For information on the 1oon check, see the Glossary.

- **Write jobs**

  In every cycle, the values of a certain number of unsolicited write jobs are written to the CPU.

  The number of tags written per cycle is specified for the transferring module in the "Communication with the CPU" parameter group with the "Max. number of write jobs" parameter. The tags whose number exceeds this value are then written in the next or one of the following cycles.

- **Low-priority read jobs - proportion**

  (Normal cycle)

  For data points with the assignment "Normal cycle", a proportion of the values of their PLC tags is read in every scan cycle.

  The number of tags read per cycle is specified for the transferring module in the "Communication with the CPU" parameter group with the "Max. number of read jobs" parameter. The tags that exceed this value and can therefore not be read in one cycle are then read in the next or one of the following cycles.

- **Cycle idle time**

  This waiting time between two scan cycles is used to reserve adequate time for other processes that access the CPU.

4.22.6 "Trigger“ tab

Trigger

Data points are configured as a static value or as an event using the "Type of transmission" parameter:

Saving the value of a data point configured as an event

Saving the value of a data point configured as an event in the send buffer (message memory) can be triggered by various trigger types:

- **Threshold value trigger**

  The value of the data point is saved when this reaches a certain threshold. The threshold is calculated as the difference compared with the last stored value, refer to the section Threshold value trigger (Page 118).

- **Time trigger**

  The value of the data point is saved at configurable intervals or at a specific time of day.

- **Event trigger (Trigger tag)**

  The value of the data point is saved when a configurable trigger signal is fired. For the trigger signal, the edge change (0 → 1) of a trigger tag is evaluated that is set by the user program. When necessary, a separate trigger tag can be configured for each data point.

  Resetting the trigger tag in the bit memory area / DB:

  If the memory area of a trigger tag is in the bit memory or in a data block, the module resets the trigger tag itself to 0 (zero) as soon as the value of the data point has been transferred. This can take up to 500 milliseconds.

Note

Fast setting of triggers

Triggers must not be set faster than a minimum interval of 500 milliseconds. This also applies to hardware triggers (input area).

Note

Hardware trigger

You need to reset hardware triggers via the user program

Transferring the value of a data point configured as an event

You specify whether the value of a data point is transferred to the communications partner immediately after the trigger fires or after a delay in the "Transmission mode" parameter.

Transmission mode

The transmission mode of a frame is set in the "Trigger" tab of the data point. With the option, you specify whether messages of events are sent immediately or following a delay:

- Spontaneous (unsolicited - direct transfer)

  The value is transferred immediately.
- Conditional spontaneous (buffered transfer)

  The value is transferred only when one of the following conditions is fulfilled:
  - The communications partner queries the station.
  - The value of another event with the transmission mode "Spontaneous" is transferred.
  - The fill level of the send buffer has reached 90% of its maximum capacity.

Enable archiving

Allows the retentive saving of events on the SD card of the S7-1500 CPU when there are connection disruptions.

Archiving can be enabled under the following conditions:

- The transmission mode of the data point is "Every value triggered".
- The "Enable retentive saving" option is selected in the parameter group "Basic settings > Retentive saving of events" of the TIM.

For information on the options and mechanisms of the parameter, see section Configuration (Page 58).

4.22.7 Threshold value trigger

Note

Threshold value trigger: Calculation only after "Analog value preprocessing"

Note that the analog value preprocessing is performed before the check for a configured threshold value and before calculating the threshold value.

This affects the value that is configured for the threshold value trigger.

Note

No Threshold value trigger if Mean value generation is configured

If mean value generation is configured, no threshold value trigger can be configured for the analog value event involved.

For the time sequence of the analog value preprocessing refer to the section Analog value preprocessing (Page 119).

Threshold value trigger

Function

If the process value deviates by the amount of the threshold value, the process value is saved.

Two methods are used to calculate the threshold value deviation:

- **Absolute method**

  With binary and counter values as well as with analog values with configured mean value generation, the absolute method is used to calculate the threshold value deviation.

- **Integrative method**

  With analog values without configured mean value generation, the integrating method is used to calculate the threshold value deviation.

  In the integrating threshold value calculation, it is not the absolute value of the deviation of the process value from the last stored value that is evaluated, but rather the integrated deviation.

Absolute method

There is a check for each binary value to determine whether the current (possibly smoothed) value is outside the threshold value band. The current threshold value band results from the last saved value and the amount of the configured threshold value:

- Upper limit of the threshold value band: Last saved value + threshold value
- Lower limit of the threshold value band: Last saved value - threshold value

As soon as the process value reaches the upper or lower limit of the threshold value band, the value is saved. The newly saved value serves as the basis for calculating the new threshold value band.

Integrative method

The integration threshold value calculation works with a cyclic comparison of the integrated current value with the last stored value. The calculation cycle in which the two values are compared is 500 milliseconds.
(Note: The calculation cycle must not be confused with the scan cycle of the CPU memory areas).

The deviations of the current process value are totaled in each calculation cycle. The trigger is set only when the totaled value reaches the configured value of the threshold value trigger and a new process value is entered in the send buffer.

The method is explained based on the following example in which a threshold value of 2.0 is configured.

Table 4- 7 Example of the integration calculation of a threshold value configured with 2.0

Time [s] (calculation cycle)	Process value stored in the send buffer	Current process value	Absolute deviation from the stored value	Integrated deviation
0	**20.0**	**20.0**	0	0
0.5		20.3	+0.3	0.3
1.0		19.8	-0.2	0.1
1.5		20.2	+0.2	0.3
2.0		20.5	+0.5	0.8
2.5		20.3	+0.3	1.1
3.0		20.4	+0.4	1.5
3.5	**20.5**	**20.5**	+0.5	**2.0**
4.0		20.4	-0.1	-0.1
4.5		20.1	-0.4	-0.5
5.0		19.9	-0.6	-1.1
5.5		20.1	-0.4	-1.5
6.0	**19.9**	**19.9**	-0.6	**-2.1**

With the change in the process value shown in the example, the threshold value trigger configured with 2.0 is initiated twice:

- At the time 3.5 s: The value of the integrated deviation is at 2.0. The new process value stored in the send buffer is 20.5.
- At the time 6.0 s: The value of the integrated deviation is at 2.1. The new process value stored in the send buffer is 19.9.

In this example, if a deviation of the process value of approximately 0.5 should fire the trigger, then with the behavior of the process value shown here a threshold value of approximately 1.5 ... 2.5 would need to be configured.

4.22.8 Analog value preprocessing

The TIM supports analog value preprocessing. For analog value data points, some or all of the functions described below can be configured.

Requirements and restrictions

You will find the requirements for the configuration of the preprocessing options and restrictions in the section relating to the particular function.

Note

Restrictions due to configured triggers

The analog value preprocessing options "Error suppression time", "Limit value calculation" and "smoothing" are not performed if no threshold value trigger is configured for the relevant data point.. In these cases, the read process value of the data point is entered in the image memory and transferred transparently before the preprocessing cycle of the threshold value calculation (500 ms) elapses.

Sequence of the analog value preprocessing options

The values of analog inputs configured as an event are processed on the TIM according to the following scheme:

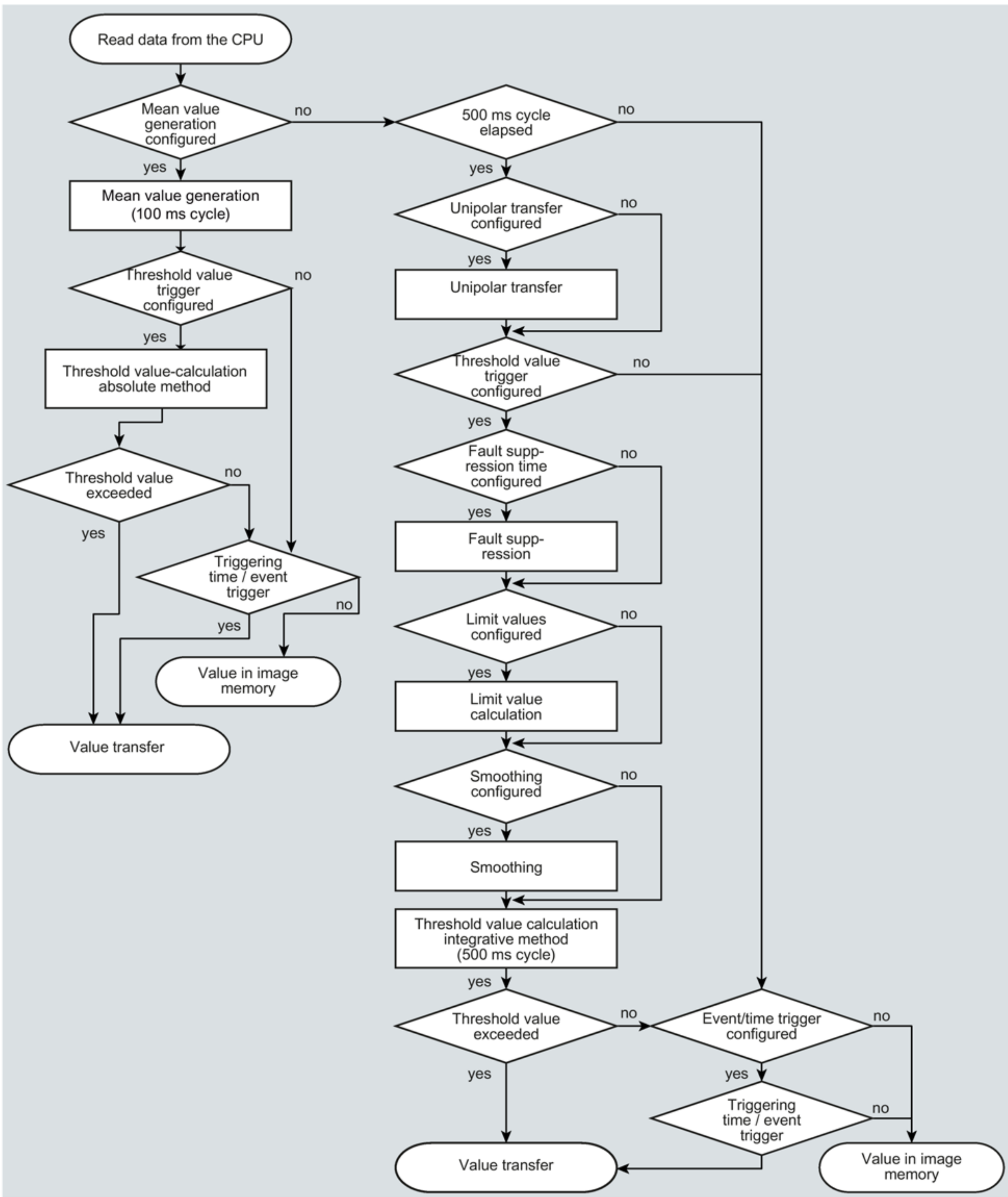


Figure 4-12 Sequence of the analog value preprocessing

The 500 millisecond cycle is started by the integrative threshold value calculation. In this cycle, the values are saved even when the following preprocessing options are enabled:

- Unipolar transfer
- Fault suppression time
- Limit value calculation
- Smoothing

Mean value generation

Note

Restricted preprocessing options if mean value generation is configured

If you configure mean value generation for an analog value event, the following preprocessing options are not available:

- Unipolar transfer
- Fault suppression time
- Smoothing

Function

With this parameter, acquired analog values are transferred as mean values.

If mean value generation is active, it makes sense to configure a time trigger..

The current values of an analog data point are read in a 100 millisecond cycle and totaled. The number of read values per time unit depends on the read cycle of the CPU and the CPU sampling cycle of the TIM.

The mean value is calculated from the accumulated values as soon as the transfer is triggered by a trigger. Following this, the accumulation starts again so that the next mean value can be calculated.

The mean value can also be calculated if the transmission of the analog value message is triggered by a request from the communications partner. The duration of the mean value calculation period is then the time from the last transmission (for example triggered by the trigger) to the time of the request. Once again, the accumulation restarts so that the next mean value can be calculated.

Input modules: Overflow range / underflow range

As soon as a value is acquired in the overflow or underflow range, mean value generation is stopped. The value 32767 / $7FFF_h$ or -32768 / 8000_h is saved as an invalid mean value for the current mean value calculation period and sent with the next message.

The calculation of a new mean value is then started. If the analog value remains in the overflow or underflow range, one of the two values named is again saved as an invalid mean value and sent when the next message is triggered.

Note

Fault suppression time > 0 configured

If you have configured an error suppression time and then enable mean value generation, the value of the error suppression time is grayed out but no longer used. If mean value generation is enabled, the error suppression time is set to 0 (zero) internally.

Unipolar transfer

Restrictions

Unipolar transfer cannot be configured at the same time as mean value generation. Enabling unipolar transfer has no effect when mean value generation is activated.

Function

With unipolar transfer, negative values are corrected to zero. This can be desirable if values from the underrange should not be transferred as real measured values.

Exception: With process data from input modules, the value -32768 / 8000_h for wire break of a live zero input is transferred.

With a software input, on the other hand, all values lower than zero are corrected to zero.

Fault suppression time

Requirements for the function

Configuration of the threshold trigger for this data point

Restrictions

The fault suppression time cannot be configured at the same time as mean value generation. A configured value has no effect when mean value generation is activated.

Function

A typical use case for this parameter is the suppression of peak current values when starting up powerful motors that would otherwise be signaled to the control center as a disruption.

The transmission of an analog value in the overflow ($7FFF_h$) or underflow range (8000_h) is suppressed for the specified time. The value $7FFF_H$ or 8000_H is only sent after the fault suppression time has elapsed, if it is still pending.

If the value returns to the measuring range before the fault suppression time elapses, the current value is transferred.

Input modules

The suppression is adjusted to analog values that are acquired directly by the S7 analog input modules as raw values. These modules return the specified values for the overflow or underflow range for all input ranges (also for live zero inputs).

An analog value in the overflow range (32767 / $7FFF_h$) or underflow range (-32768 / 8000_h) is not transferred for the duration of the fault suppression time. This also applies to live zero inputs. The value in the overflow/underflow range is only sent after the fault suppression time has elapsed, if it is still pending.

Recommendation for finished values that were preprocessed by the CPU:

If the CPU makes preprocessed finished values available in bit memory or in a data block, suppression is only possible or useful if these finished values also adopt the values listed above 32767 / $7FFF_h$ or -32768 / 8000_h in the overflow or underflow range. If this is not the case, the parameter should not be configured for preprocessed values.

For finished values preprocess in the CPU, the limits for the overflow and underflow can be freely assigned.

Smoothing factor

Requirements for the function

Configuration of the threshold trigger for this data point

Restrictions

The smoothing factor cannot be configured at the same time as mean value generation. A configured value has no effect when mean value generation is activated.

Function

Analog values that fluctuate quickly can be evened out using the smoothing function.

The smoothing factors are calculated according to the following formula as with S7 analog input modules.

$$y_n = \frac{x_n + (k-1)\ y_{n-1}}{k}$$

where
y_n = smoothed value in the current cycle n
y_{n-1} = smoothed value in the previous cycle n-1
x_n = value acquired in the current cycle n
k = smoothing factor

The following values can be configured for the module as the smoothing factor.

- 1 = No smoothing
- 4 = Weak smoothing
- 32 = Medium smoothing
- 64 = Strong smoothing

Set limit value 'low' / Set limit value 'high'

Requirements for the function

- Configuration of the threshold trigger for this data point
- PLC tag in the bit memory operand area or data area

  The analog value data point must be linked to a PLC tag in the bit memory or data area (data block).

  For PLC tags for analog input modules (input operand area) limit value configuration is not possible. With these analog values the limit values of the following table are used automatically.

The configuration of limit values is pointless for measured values that have already been preprocessed on the CPU.

Function

In these two input boxes, you can set a limit value in the direction of the start of the measuring range or in the direction of the end of the measuring range. You can also evaluate the limit values, for example as the start or end of the measuring range.

Configuration of the limit value

The limit value is configured as a whole decimal number. The range of values is based on the range of values of the raw value of analog input modules.

Range	Raw value (16 bits) of the PLC tag		Module output [mA]			Measuring range [%]
	Decimal	Hexadecimal	0 .. 20 (unipolar)	-20 .. +20 (bipolar)	4 .. 20 (life zero)	
Overflow	32767	7FFF	> 23.515	> 23.515	> 22.810	> 117.593
Overrange	32511 ... 27649	7EFF ... 6C01	23.515 ... 20.001	23.515 ... 20.001	22.810 ... 20.001	117.593 ... 100.004
Nominal range (unipolar / life zero)	27648 ... 0	6C00 ... 0000	20 ... 0		20 ... 4	100 ... 0
Nominal range (bipolar)	27648 ... 0 ... -27648	6C00 ... 0000 ... 9400		20 ... 0 ... -20		100 ... 0 ... -100
Underrange (unipolar / life zero)	-1 ... -4864	FFFF ... ED00	-0.001 ... -3.518		3.999 ... 1.185	-0.004 ... -17.59
Underrange (bipolar)	-27649 ... -32512	93FF ... 8100		-20.001 ... -23.516		-100.004 ... -117.593
Undershoot / wire break	-32768	8000	< -3.518		< 1.185	< -17.593

Please note: The entry of the value 0 (zero) is interpreted as a deactivated limit value.

Note

Evaluation of the value even when the option is disabled

If you enable one or both options and configure a value and then disable the option later, the grayed out value is nevertheless evaluated.

To disable the two options, delete the previously configured values limit values from the input boxes and then disable the relevant option.

Recommendation for quickly fluctuating analog values:

If the analog value fluctuates quickly, it may be useful to smooth the analog value first if limit values are configured.

4.22.9 Command output

Parameters for the data point type "Command output (Cmd01B_R)"

The following output options can alternatively be assigned to a data point of the type "Command output (Cmd01B_R)":

- **LATCH_ON/OFF**

  The function permanently latches the command output to the value 1.

  Note:
  The latched value is only canceled by a new command. Alternatively, the command can be reset by the user program.
- **PULSE_ON/OFF**

  The function evaluates the "Command output time" for the command output of the data point.

Parameter

Name:	**Max. number of pulses**
Range of values:	1
Explanation:	Only a single received pulse is evaluated. If the number of pulses received exceeds the value 1, the command is discarded.

Name:	**Command output time (s)**
Range of values:	0 ... 65535
Default:	0
Explanation:	Period for which the command is output. Only for the "PULSE_ON/OFF" option. When the time expires, the output is automatically reset. With zero, a set command output is not automatically reset. In this case, you need to reset the output via the user program.

4.22.10 Partner stations

Enabling the partner for the ST7 data point

Enable all partners to which a telecontrol connection was configured and with which the selected data point will exchange data. To do this select the relevant " " check box.

You can configure or change the partners in the overview table or in the "Partner stations" tab of the module. The data will be adopted at the other point.

4.23 Messages

Configuring e-mails

If important events occur, the TIM can send e-mails to a communications partner.

You configure the e-mail in STEP 7 in the editor for the data point and message configuration. You can find this using the project tree:
Project > Directory of the relevant TIM

For the view in STEP 7, refer to the section Data point configuration (Page 104).

Requirements and necessary information

Remember the following requirements in the configuration for the transfer of messages:

- Enabling telecontrol communication ("Communication types") parameter group
- SMS: Configuration of the SMSC in the "SMSC" parameter group
- E-mails: Configuration of the "E-mail configuration" parameter group

  To do this, you require the following information:

  – Access data of the SMTP server: Address, port number, user name, password
  – TIM's own e-mail address
  – Email address of the recipient

"Message parameter"

Here you configure the recipient, the subject and the text of the message.

"Trigger"

In the "Trigger" parameter group you configure triggering for sending the message and other parameters.

- **E-mail trigger / SMS trigger**

  Specifies the event for which the sending of the message is triggered.

  – Use PLC tag

    For the trigger signal to send the message, the edge change (0 → 1) of the trigger bit "PLC tag for trigger" is evaluated that is set by the user program. When necessary, a separate trigger bit can be configured for each e-mail. For information on the trigger bit, see below.

– CPU changes to STOP

– CPU changes to RUN

– Connection to a partner interrupted

  Triggers the sending of the message when the connection to a partner is interrupted.

– Connection to a partner established

  Triggers the sending of the message when the connection returns.

– Weak mobile wireless network

  Only with SMS

  Triggers the sending of an SMS when the network is weak and possibly no dialup connection can be established via mobile wireless.

- **PLC tag for trigger**

  PLC tag for the message trigger "Use PLC tag"

  If the memory area of the trigger bit is in the bit memory or in a data block, the trigger bit is reset to zero when the message is sent.

- **Enable identifier for processing status**

  If the option is enabled, every attempt to send returns a status with information about the processing status of the sent message.

  The status is written to the "PLC tag for processing status". If there are problems delivering messages, you can determine the status via the Web server (diagnostics status) or read it out from the PLC tag.

  For the significance of the status output in hexadecimal, refer to the section Processing status of the messages (SMS / e-mail) (Page 290).

- **PLC tag for processing status**

  PLC tag of the type DWORD for the processing status

- **Include value**

  If you enable the option, the TIM sends a value for the placeholder $$ from the memory area of the CPU in the message. To do this enter "$$" as a placeholder for the value to be sent in the message text.

  Select a PLC tag whose value will be integrated in the message. The value is entered in the message text instead of the placeholder $$.

  $$ can be a placeholder for data point types with a simple data type up to a size of 32 bits.

- **PLC tag for value**

  PLC tag in which the value to be sent is written.

5 Block library TeleControl ST7

5.1 Creating a library

Delivery form of the block library

The program blocks of the TD7onCPU library are used as a global library for STEP 7.

The library is archived and compressed on the Internet at the following address:
Link: (https://support.industry.siemens.com/cs/ww/en/view/109755374)

The file archive name has the following form: TD7_<version>.zip

Creating the library

To use the library in STEP 7, follow these steps:

1. Create a suitable directory in the STEP 7 installation directory of your engineering station, for example, one named "Telecontrol ST7".
2. Save the compressed and archived library (*.zip) in this directory.
3. Unpack the file archive using the utilities of the operating system.

   The library is now available as an archived global library under the name "Telecontrol ST7.zal15".
4. Open the STEP 7 project.
5. In the "Options" menu, select the command "Global libraries > Retrieve library".

   The "Retrieve archived global library" dialog opens.
6. Select the archive file.
7. Recommendation: Select the check box "Open the project read-only" to load the global library read-only.
8. Click Open.

   The "Find folder" dialog opens. Select the destination directory in which the archived global library is to be unpacked.
9. Click "OK".

   The library is unpacked and opened.

You can find the library in the STEP 7 project in the library view (task card) in the "Global libraries" pane under the name "Telecontrol ST7_<Version>".

5.2 Structure of the library

Structure of the global library

The opened library contains the following directories:

- **Types**
  - PLC data types (UDT)

    Here you will find the PLC data types (UDTs) used by the libraries.
  - System data types (SDT)

    Here you will find the system data types used by the libraries.
- **Master copies**

  You can find the following program blocks here:
  - **CPU300/400**

    Program blocks for the S7-300/400 CPU
  - **CPU1500**

    Program blocks for the S7-1500 CPU

Master copies

The two libraries under "Master copies" for S7-300/400 and S7-1500 are divided into the following folders:

- Data point typicals

  Data point typicals (blocks) for transferring data
- Optional blocks

  Optional blocks that you can use when necessary.
- System blocks

  Blocks that are used for basic communication.

  Depending on the SIMATIC product series used, you can find the following auxiliary blocks for communication between TIM and CPU in separate folders:
  - BCom

    Blocks for processing the communication mailbox of the S7-400 and S7-1500
  - PCom

    Blocks for processing the communication mailbox of an S7-300 with P bus (CPU without partyline)
  - XCom

    Blocks for processing the communication mailbox of an S7-300 with X connections

5.3 Generating, using and calling the blocks

Generating TD7onCPU blocks

Once you have opened the TD7 library in the pane "Global libraries" under the name "Telecontrol ST7", you can generate the blocks for basic communication for one or more CPUs.

Requirements

The following requirements must be met before generating the blocks:

- The communication modules must be networked with each other.
- A TIM 1531 IRC must be networked with a CPU via Ethernet.
- A TIM 1531 IRC cannot be assigned to a CPU.

  In the "Subscriber numbers" parameter group, the "Selected CPU" box must be empty.
- No data point can be created in the TIM whose CPU is to use TD7onCPU.
- No message can have been created already in the message editor of the TIM.
- Valid telecontrol connections must be created between the communication partners involved.
- The "Telecontrol ST7" library must be open in the "Global libraries" task card.

Different requirements for modules with imported configuration data

Note the other requirements for communications modules whose SINAUT configuration was imported from a STEP 7 V5 project. With these modules, the option "Telecontrol configuration" is set to "Import" in the parameter group "Basic settings > Configuration".

These modules have different requirements for generating the blocks:

- No ST7 telecontrol connections need to be created between the communication partners involved.

  The relevant information is stored in the imported configuration data.
- The CPU of the communication module must have the same IP address as in the STEP 7 V5 project. The CPU is assigned to the communication module via its IP address.

Procedure for generating

Generate the blocks for basic communication as follows:

1. Select the desired TIM.
2. Select the shortcut menu command (right mouse button) "Generate TD7onCPU blocks".
3. In the dialog that follows, select one or more CPUs and click "Generate..."

The TD7onCPU blocks are created and compiled in the selected CPUs under the program blocks in a new "TD7onCPU" directory.

Using blocks

- **System blocks**

  The required blocks for basic communication are created automatically during the generation in the CPU.

  You do not need to use these blocks manually in the CPU.

  The system blocks handle central tasks such as startup, monitoring of the connections, reachability of the connection partners, entering data in the send mailbox or taking data from the receive mailbox of the communication DBs, general request, time keeping, handling communication etc.

  Communication DBs:

  – S7-300/400

    For each connection a separate communication DB is created automatically: It contains the send and receive mailbox and all the data required for controlling and monitoring this connection.

  – S7-1500

    For the CPU 1500, there is only one communication DB "BConnectData". A connection instance is stored in this DB in the "BConnection" array for each local connection.

- **Data point typicals**

  You need to use data point typicals on the relevant CPU according to the communications tasks.

  Drag the individual blocks with the mouse from the global library to the user program of the CPU. The corresponding instance data block is created automatically when using a data point typical.

  – Dat256D_S / Dat256D_R

    Copy also the UDT "TransmitBlock" for these typicals from the library (Types > PLC data types (UDT)) into the "PLC data types"directory of the CPU. The UDT is not automatically referenced by the typical.

- **Optional blocks**

  You should also use the optional blocks on the CPU depending on the communications tasks of the station.

  Drag the individual blocks with the mouse from the global library to the user program of the CPU.

Call in the user program

- **System blocks**

  Only the two following FCs need to be called in the user program:

  – StartUp

    Call the FC in the startup OB. The block has no parameters.

    To be able to call the FC StartUp, you must first create the startup OB (OB100).

  – BasicTask

    Call the FC in the cyclic OB.

    The DB BasicData is created automatically. The DB contains all the centrally required data including the records of all communication partners and the connections to be managed.

    For information on the various "BasicTask" FCs, see section Note on the "BasicTask_*" FCs (Page 133).

- **Data point typicals**

  Depending on the communication tasks, call up all required data point typicals in the user program.

- **Optional blocks**

  Depending on the communication tasks, call up all required optional blocks in the user program.

See also section The cyclic SINAUT program (Page 134) for information on calling the blocks.

Note

Error display for library conformance

In the "Compilation" parameter group of blocks, an error can be displayed under the entry "Library conformance" (call of single instances). This is caused by the call of global data blocks, for example, DB BasicData.

Since the "Telecontrol ST7" library has no "know-how protection", you can ignore these error notifications.

5.4 Note on the "BasicTask_*" FCs

Versions of the "BasicTask" FC

There are different versions of the "BasicTask" FC in the TD7 block library "Telecontrol ST7" for STEP 7 Professional.

Depending on the SIMATIC product series and the backplane bus, the FCs have a different suffix at the end of the block name:

- **BasicTask_B**

  For S7-1500 CPU
- **BasicTask_B**

  For S7-400 CPU
- **BasicTask_X**

  For S7-300 PU with partyline
- **BasicTask_P**

  For S7-300 CPU without partyline

For information on "Partyline", see Glossary.

For the block description, see section BasicTask_* FC (Page 255).

"BasicTask" label for three FCs

References to the respective FC are made in many places in the following sections. Since the description of the different blocks usually applies to several SIMATIC product series, the three FCs are referred to hereafter as "BasicTask". The FC version for the relevant CPU type is meant in each context.

5.5 Structure of user program for TD7onCPU

5.5.1 The cyclic SINAUT program

Introduction

The structure of the TD7 blocks in the user program (OB1) is explained below.

The TD7 blocks must be proceeded in every OB1 cycle. Keep to the call sequence of the blocks unless instructed otherwise.

Other user-specific parts of the program can be linked in before or after the TD7 blocks in OB1 or, if practical, within the TD7 blocks.

You can structure the SINAUT-specific part in OB1 by calling it in lower-level FCs.

All data point typicals are FBs. An instance DB must be specified when an FB is called. The number of this instance DB is identical to the object number of the data point object.

The structure of the TD7 blocks in OB1

Cyclic OB1	
BasicTask	The respective FC BasicTask must always be called at the start. It handles basic SINAUT tasks that are always required.
Optional SINAUT basic functions	After FC BasicTask, call the following optional blocks (FCs) if you need them: • ListGenerator Creation of address lists for received frames with incomplete destination address. • TimeTask Provides the time of day.
	If necessary, call additional optional FCs in the subsequent program execution, for example: • Trigger Scheduled start of user programs and data frames • PartnerStatus Display the status of partners • PartnerMonitor Extended subscriber-specific display and control options. • PulseCounter Count pulse acquisition • PathStatus Display the status of the connection path to a partner **Note:** The following two blocks may not be called explicitly, since they are activated via an internal interface. • TestCopy • TestcopyDB
Data point typicals	In the subsequent program execution, call the data point typicals for sending and receiving data. The sequence of the individual typicals is unimportant. The number of typicals to call and the required types depend on the amount and type of data to be sent and received. **Note:** The instance DBs of these FBs are referenced using the partner object number of the communication partner.

Example of a TD7 program for a station

Cyclic OB1	
BasicTask	The respective FC BasicTask must always be called at the start of the cyclic program. Generally, data is ready for further processing after it has been received. The FC has only one parameter, "UserFC". Normally 0 can be specified. If you require user-specific processing for received data, specify here the number of an FC containing the program for this processing. You can also copy received data chronologically to an archive memory via this interface.
TimeTask	As an option, you can call the FC TimeTask immediately after FC BasicTask. The FC has no parameters. FC TimeTask TimeTask must be included if you need the time of day. This allows data frames to be time-stamped. However, you can also use the time of day to start program components at a specific point in time or to schedule the transmission of data frames. FC Trigger, described below, is then required. For this FC Time Task to be used, the CPU must be provided with the time from a local TIM module. You specify this in the configuration.
Trigger	FC Trigger can be included as an option. The FC sets its output for the duration of one OB1 cycle when the point in time or the time interval set for the FC has been reached. The FC can be inserted several times if several times or various time intervals are required. Using the FC requires that FC TimeTask must be called first in the OB1 program (see above) and the CPU time must have already been set once.
PartnerStatus	FC PartnerStatus can be included as an option. The FC shows the reachability for a maximum of 8 communications partners.
ListGenerator	• ListGenerator300 for S7-300 CPU • ListGenerator400 for S7-400 CPU • ListGenerator1500 for S7-1500 CPU The FC can be installed as an option. The FC is required if the station receives frames containing no destination address or an incomplete destination address. This happens if the configuration of the destination address has been completely or partially omitted in a partner for data point typicals. ("PartnerNo" and "PartnerObjectNo" were not specified, so transmission to all known destination subscribers takes place.)
PulseCounter **PathStatus** **PartnerMonitor**	When necessary

Cyclic OB1	
Data point typicals	Following the FCs for SINAUT basic tasks, data point typicals for sending and receiving data are called. The sequence of the individual typicals is unimportant. The number of typicals to call and the required types depend on the amount and type of data to be sent and received. The following typically applies to a station: • The following are sent: – Binary information, such as status messages and alarms – Analog values – Counted values • The following are received: – Commands – Setpoints, limit values, parameters
Bin04B_S	One or more FBs for the acquisition and transfer of binary information, such as messages, alarms, etc.
Ana04W_S	One or more FBs for the acquisition and transfer of analog values
Cnt01D_S / Cnt04D_S	One or more FBs for the acquisition and transfer of counted values A requirement for the use of the FBs is that FC PulseCounter is included in a cyclic interrupt OB, e.g. OB35. This FC is responsible for the time-driven acquisition of counted pulses in the background.
Cmd01B_R	One or more FBs for receipt and output of commands.
Set01W_R / Par12D_R	One or more FBs for receipt and output of setpoints, limit values or parameters..
Dat12D_S / Dat256D_S	One or more FBs for the acquisition and transfer of 12 data double words with any content. There is no data-specific processing and change control for these typicals. The user program is responsible for this. As an option change control can be activated that triggers a transfer with each bit change.

TD7 programs for master stations and node stations

In principle, the TD7 programs of master stations and node stations look the same as for a station. The corresponding receive typicals are used in a master station for the send typicals of a station.

In a node station, both send typicals and the corresponding receive typicals are used according to the transmission direction.

In the master station it is practical to structure the OB1 program according to stations, in other words, all data typicals belonging to the same station are called in one FC. The best overview is provided when the number of the FC is identical to the subscriber number of the station.

5.5.2 Cyclic interrupt OB

Introduction

Only include time-controlled TD7 blocks in a CPU if fast count pulses must be detected in this CPU which could not be reliably detected within an OB1 cycle due to an excessively long cycle time.

The count pulses are acquired via any digital input module. To acquire the pulses reliably, the digital inputs use must be queried for change at a fixed time interval. The time interval is based on the duration of the shortest count pulse. The minimum count pulse duration may be 50 ms. The same applies to the duration of the pause. This results in a maximum count frequency of 10 Hz.

The time interval in which the count pulse acquisition is performed must be approximately half the count pulse duration, i.e. at 50 ms at an interval of approximately 25 ms.

For this time-controlled count pulse acquisition, OB35 needs to be configured for an S7-300 CPU, one of the available cyclic interrupt OBs OB30 to OB38 for an S7-400 CPU, and a cyclic interrupt OB with a number from 30..38 / ≥ 123 for an S7-1500 CPU.

All cyclic interrupt OBs have a preset time interval, for OB35 for example 100 ms. This can be changed in steps of 1 ms. This makes it possible to set a cyclic interrupt OB, for example, to 25 ms.

You change the time interval for a cyclic interrupt OB in the Properties dialog (S7-400/1500), or in the Properties dialog of the CPU with S7-300.

Table 5- 1 Calling FC PulseCounter in the cyclic interrupt OB

OB35	
Pulse Counter	One or more FC-PulseCounters can be included for the acquisition of count pulses. FC PulseCounter processes up to 8 pulse inputs of any digital input. The acquired count pulses are added together in programmable SIMATIC counters. These access the function blocks that put together the count value frames (FB-Cnt0x_S). User-specific cyclic interrupt functions required independently of the SINAUT program can be included at any point in the cyclic interrupt OB.

5.5.3 Programming error OB

Validity

Only for S7-300/400

Function

When a block that does not exist is called in an S7-300/400 CPU, the CPU normally changes to STOP. The diagnostics buffer indicates which FB, FC or DB was missing. You can then reload the missing block and restart the CPU.

If, however, you do not want the CPU to change to STOP if there is a missing block or only changes to STOP when certain block types or block numbers are missing, you can specify the reaction you require in OB121.

Even if you have loaded OB121 as an empty block on the CPU, this is enough to have the CPU continue running if a block is missing. If you want to decide more selectively when the CPU should continue running or change to STOP, include OB121 in the user program.

In conjunction with SINAUT ST7 it is possible that a CPU changes to STOP when it receives data from another CPU that it does not (or not yet) know. This is, for example, the case when you add a data point typical to a station and give it a complete destination address (destination subscriber no. plus destination object no.). The specified destination object no. can lead to a stop on the destination subscriber in the following situation:

- As soon as you have installed a new data point typical in a station, the data is transferred to the destination.
- If there is no corresponding receive typical installed on the destination CPU, the destination object no. (= instance DB of the received typical) does not exist.

The result is that the CPU changes to STOP as soon as the data is received.

To avoid this it is advisable to call FC ST7ObjectTest in OB121.

Table 5- 2 Call of FC ST7ObjectTest in the programming error OB

OB121	
ST7ObjectTest	Calling FC ST7ObjectTest in OB121 prevents a CPU STOP, if the CPU receives data with an unknown destination object no. Other calls can be included at any point in OB121 regardless of the FC ST7ObjectTest call. FC ST7ObjectTest has a parameter "StopInOtherCases". Here you can specify what happens in other situations (STOP or continue running) if OB121 was called because another data block or an FB or FC is missing.

5.6 Types

5.6.1 PLC data types (UDT)

5.6.1.1 UDT

UDTs used

The following PLC data type (UDT) is used by the block library:

- TransmitBlock

  UDT for the data transfer between the CPU and TIM

  When using the "Dat256D_S / Dat256D_R" typical, copy the UDT into the "PLC data types" directory of the respective CPU. The UDT is not automatically referenced by the typical.

  The UDT does not require any parameter assignment.

5.6.2 System data types (SDT)

5.6.2.1 SDTs

SDTs used

The following system data types (SDTs) are used by the DB BasicData and stored automatically in the directory "PLC data types" of the CPU when generating/compiling the TD7 blocks:

- ConnectionDescription

  SDT is part of the "BasicData" DB and has an "Array[1..1] of ConnectionDescription" data type. It contains the following details for the connection description:

  – Number of the configuration DB

  – Connection type

  – CFB number in the case of an S7 connection

  – Subscriber number of the local TIM.

- SubscriberObject

  SDT is part of the "BasicData" DB and has a "Array[1..1] of SubscriberObject" data type. It contains the following information about the communication partner:

  – Subscriber number of the partner

  – Array index of the connection description

  – Subscriber type

  – Information whether the partner is local or remote.

5.7 Master copies

Validity

The following program blocks are located in the master copies of the global library:

- **CPU300/400**

  Program blocks for the S7-300/400 CPU

- **CPU1500**

  Program blocks for the S7-1500 CPU

The blocks are largely identical. Unless otherwise stated, the description applies equally to the blocks of the S7-300/400 and S7-1500.

If individual blocks cannot be used for all SIMATIC product series mentioned or if functions for the SIMATIC product series differ, this is expressly indicated.

5.7.1 Data point typicals

5.7.1.1 Types and overview of the data point typicals

Types of the data point typicals

Data point typicals process one or more data points of the same information type, e.g. 4 bytes of binary information, 4 analog values or 1 byte commands, etc.

When a data point typical (FB) is called the corresponding instance DB needs to be specified in which the data will be written or from which the data to be transferred will be read.

Transfer direction of the typicals

A data point typical for a specific type or amount of information always come in two versions:

- A typical for acquiring and sending
- A typical for receiving and outputting

When using data point typicals a distinction is therefore made according to the transfer direction:

- **Sending typicals**

  Sending typicals process data and send it to the remote partner.

  They have the ending "_S" Examples: Bin04B_S, Ana04W_S

- **Receiving typicals**

  Receiving typicals receive the data from their remote partner.

  They have the ending "_R" Examples: Bin04B_R, Ana04W_R

When transferring data on the two communicating partners there is always a corresponding pair of typicals involved.

Examples:

- Monitoring direction
  - The station sends binary data with the typical Bin04B_S.
  - The central station receives the data with the typical Bin04B_R.
- Control direction
  - The central station sends a command with the typical Cmd01B_S.
  - The station receives the command with the typical Cmd01B_R.

Structure of typical names

Note

No SMS blocks

Unlike the block library for STEP 7 V5, there are no blocks for sending SMS in the global library "Telecontrol ST7".

You configure the sending of e-mails and/or SMS in STEP 7 Professional in the TIA Portal via the message editor, see section Messages (Page 127).

The 8-character names of the data point typicals are assigned according to the following scheme:

Table 5- 3 Structure of 8-character typical names

<table>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td colspan="3">Data point type</td><td colspan="2" rowspan="8">Amount of data
(number of the type according to characters 6)
Exception:
The amount of data with "Dat256D_x" occupies three columns.</td><td rowspan="8">Data format
X = bit
B = byte
W = word
D = double word
R = real</td><td rowspan="8">Not used
(underscore)</td><td rowspan="8">S
Send function
R
Receive function</td></tr>
<tr><td colspan="3">Bin = Binary information</td></tr>
<tr><td colspan="3">Ana = Analog value</td></tr>
<tr><td colspan="3">Cnt = Count value</td></tr>
<tr><td colspan="3">Cmd = Command</td></tr>
<tr><td colspan="3">Set = Setpoint, parameter</td></tr>
<tr><td colspan="3">Par = Parameter</td></tr>
<tr><td colspan="3">Dat = Data (any mixture of information types)</td></tr>
</table>

Overview of the data point typicals

The following table provides an overview of the data point typicals.

Table 5- 4 Overview of the data point typicals

Data format	Symbolic block name	Function
Binary value typicals		
Byte	Bin04B_S	Send 4 byte binary values
Byte	Bin04B_R	Receive 4 byte binary values
Analog value typicals		
Int	Ana04W_S	Send 4 analog values
Int	Ana04W_R	Receive 4 analog values
Real	Ana04R_S	Send 4 analog values as floating point number (CPU 1500 only)
Real	Ana04R_R	Receive 4 analog values as floating point number (CPU 1500 only)
Counted value typicals		
UInt, Word	Cnt01D_S	Send 1 counted value
UInt, Word	Cnt01D_R	Receive 1 counted value
UInt, Word	Cnt04D_S	Send 4 counted values
UInt, Word	Cnt04D_R	Receive 4 counted values
Command typicals		
Byte, USInt	Cmd01B_S	Send 1 byte commands (1-out-of-8)
Byte, USInt	Cmd01B_R	Receive 1 byte commands (1-out-of-8)
Setpoint and parameter typicals		
Word	Set01W_S	Send 1 setpoint and receive current local setpoint
Word	Set01W_R	Receive 1 setpoint and send current local setpoint
ARRAY [1...12] of DInt / UDInt / DWord / Real	Par12D_S	Send max. 12 double words with parameters and receive current local parameters
ARRAY [1...12] of DInt / UDInt / DWord / Real	Par12D_R	Receive max. 12 double words with parameters and send current local parameters
Typicals for variable data types and amounts		
ARRAY [1...12] of DInt / UDInt / DWord / Real	Dat12D_S	Send maximum of 12 double words with any data
ARRAY [1...12] of DInt / UDInt / DWord / Real	Dat12D_R	Receive maximum of 12 double words with any data
DWord	Dat256D_S	Send maximum of 256 double words with any data
DWord	Dat256D_R	Receive maximum of 256 double words with any data

5.7.1.2 Reoccurring parameters

Description of the typical parameters

In the following description data point typicals that are identical except for the number of data points to be processed are described together.

In the tables of the data point typicals you will find the following information about the individual parameters:

- Parameter

  Name of the parameter
- Declaration

  Parameter type
  - INPUT

    Input parameter
  - OUTPUT

    Output parameter
  - IN_OUT

    In-out parameter
- Data type

  Data type supported for this parameter
- Range of values
- Default

  Preset value of the parameter

  If you do not program individual parameters of a data point typical the preset value will be used.
- Explanation

  Description of the function and the typical-specific properties of the parameter

Parameters used by many typicals

The following parameters are used by many of the typicals of the TD7onCPU library. These parameters are described once here and the description is not repeated in the following sections for the individual data point typicals.

Depending on the use of the typicals, some parameters are programmed differently. Note the typical use described below.

PartnerNo

Parameter:	**PartnerNo**
Declaration:	INPUT
Data type:	INT
Range of values:	0 / 1 ... 32000
Default:	0
Explanation:	Subscriber number of the partner

The subscriber number of the partner with which the block communicates must be specified..

- With a process typical this is normally the subscriber number of the master station or the application of the control center (e.g. ST7cc).
- With an operator typical this is normally the subscriber number of a station.

Effects of the value 0 (zero) on various typical classes

- **Sending process typicals**

  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S)

  With the value 0 the data is sent to all subscribers to which an ST7 connection is configured. In this case, the parameter "PartnerObjectNo" is automatically sent with the value zero by the process typical.

  If PartnerNo is not found in the administration (DB BasicData), an entry to this effect is made in the diagnostics buffer (event ID B101). The CPU does not change to STOP. The block is then no longer processed, however, until the error in the programming has been corrected.

  Notes

  – If the"PartnerObjectNo" is missing, there must be a list on the partner CPU from which the missing object number can be recognized.

  – Use of the block in a node station

    If the CPU of the node station maintains both connections to higher-level subscribers as well as to lower-level stations, a data frame with PartnerNo = 0 is transferred to all subscribers in the direction master station and in the direction stations.

- **Receiving operator typicals**

  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Dat12D_R, Dat256D_R)

  The value 0 is not permitted!

  If PartnerNo is < 1 or > 32000, an error message is entered in the diagnostic buffer (event ID B100).

  If the programmed value is permitted and correct but "PartnerNo" is not found in the administration (DB BasicData), an entry is also made in the diagnostics buffer (event ID B101). The CPU does not change to STOP. The block is then no longer processed, however, until the error in the programming has been corrected.

  If the CPU receives a data frame for this typical, a check is made to establish whether source subscriber number in the data frame is identical to the "PartnerNo" programmed here. If the two subscriber numbers are different, the received information is discarded and an error message is entered in the diagnostic buffer (Event ID B130).
- **Sending operator typicals**

  (Cmd01B_S, Set01W_S, Par12D_S)

  The value 0 is not permitted!

  If "PartnerNo" is < 1 or > 32000, an error message is entered in the diagnostic buffer (event ID B100).

  If the programmed value is permitted and correct but "PartnerNo" is not found in the administration (DB BasicData), an entry is also made in the diagnostics buffer (event ID B101). The CPU does not change to STOP. The block is then no longer processed, however, until the error in the programming has been corrected.

- **Receiving process typicals**

  (Cmd01B_R, Set01W_R, Par12D_R)

  Program the value 0 if the typical is to receive data from more than one partner, e.g. if it is to receive data from several control centers.

  If the CPU receives data for this typical and "PartnerNo" is > 0, a check is made to establish whether the source subscriber number in the data frame is identical to the "PartnerNo" programmed here. If the two are different, the received information is discarded and an error message is entered in the diagnostic buffer (Event ID B130).

  This check is not made if "PartnerNo" is = 0. Regardless of the sender, each data frame addressed to the typical is passed on to the typical.

  If "PartnerNo" is > 0 and this number is not found in the administration (in DB-BasicData), an entry is made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The block is then no longer processed, however, until the error in the programming has been corrected.

  Notes

  – If "PartnerNo = 0" is set, make sure that each partner sends the data with a complete destination address (target subscriber no. and target object no.).

  – Use of the block in a node station

    If the CPU of the node station maintains both connections to higher-level subscribers as well as to lower-level stations, a data frame with PartnerNo = 0 is transferred to all subscribers in the direction master station and in the direction stations.

PartnerObjectNo

Parameter:	**PartnerObjectNo**
Declaration:	INPUT
Data type:	INT
Range of values:	0 / 1 ... 32000
Default:	0
Explanation:	Object number of the partner The number of the object (= DB number) on the partner with which the block communicates.

Effects of the value 0 (zero) on various typical classes

- **Sending process typicals**

  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S)

  Programming 0 is useful, if PartnerNo = 0 was programmed for the previous parameter. If the"PartnerObjectNo" is missing, there must be a list on the partner CPU from which the missing object number can be recognized (see FC-ListGenerator).

  If the partner is an ST7cc control center specifying the "PartnerObjectNo" for this block can be omitted because in ST7cc there are no DBs as target objects. ST7cc decodes its data solely based on the source address in the data frame.
- **Receiving operator typicals**

  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Dat12D_R, Dat256D_R)

  The value 0 is not permitted!

  If the programming is incorrect (< 1 or > 32000), an error message is entered in the diagnostic buffer (event ID B102). The CPU does not change to STOP. The block is then no longer processed, however, until the error in the programming has been corrected.

  If the CPU receives data for the object programmed here, there is a check to establish whether the source object number in the data frame is identical to the "PartnerObjectNo" programmed here. If they are different, the received information is discarded. An error message is entered in the diagnostic buffer (event ID B131).
- **Sending operator typicals**

  (Cmd01B_S, Set01W_S, Par12D_S)

  The value 0 is not permitted!

  If the programming is incorrect (< 1 or > 32000), an error message is entered in the diagnostic buffer (event ID B102). The CPU does not change to STOP. The block is then no longer processed, however, until the error in the programming has been corrected.

- **Receiving process typicals**

  (Cmd01B_R, Set01W_R, Par12D_R)

  Programming 0 is necessary in the following situations:
  - The partner is not an S7 CPU, i.e. there is no DB number as object. This is, for example, the case when the partner is an ST7cc control center.
  - There is more than one partner (PartnerNo = 0) from which the typical is to receive data. The corresponding objects of these partners will then generally have different numbers; in other words, no unique number can be specified here.

  If the CPU receives data for the object programmed here and "PartnerObjectNo" is > 0, there is a check to establish whether the source object number in the data frame is identical to the "PartnerObjectNo" programmed here. If they are different, the received information is discarded. An error message is entered in the diagnostic buffer (event ID B131).

  This check is not made if "PartnerObjectNo" is = 0. Regardless of the sender object, each data frame addressed to the object is also passed on to the receiving object.

Enabled

Parameter:	**Enabled**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	TRUE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Enable block processing	

- If processing is enabled, all the functions of the block execute.
- The response is different if processing has not been enabled:

Processing not enabled

- **With sending process typicals**

  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S,
  Set01W_R, Par12D_R,
  Dat12D_S, Dat256D_S)

  If processing is not enabled, the block can only communicate at the organizational level; in other words, Org frames can be sent and received.

  Please note: Although a query from an operator typical is answered, however the reply frame contains the data valid at the time the function was disabled.

  Note
  The response described here does not apply to Cmd01B_R, (see Command typical Cmd01B_R (Page 173))!

- **With receiving operator typicals**

  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R,
  Set01W_S, Par12D_S,
  Dat12D_R, Dat256D_R)

  If processing is not enabled, the block can only communicate at the organizational level; in other words, Org frames can be sent and received.

  Please note: A request can still be sent and the answer received, the received information is, however, not output to the outputs. You will find the relevant outputs in the description of the appropriate data point typicals.

  Note
  The response described here does not apply to Cmd01B_S, see Command typical Cmd01B_S (Page 171)!

ImageMemory

Parameter:	**ImageMemory**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	TRUE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Using in typicals	Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S, Dat256D_S	

Explanation:	Image memory principle for spontaneous data transmission • TRUE The data is transferred according to the image memory principle. The image memory principle reduces the memory required for storing data frames and produces as little data traffic on the WAN as possible. The default TRUE is the correct choice in most cases. • FALSE The data is transferred according to the send buffer principle. The send buffer principle is only required with data points whose individual data changes will be saved and transferred to the partner, for example alarms with time stamps.	

Conditional

Parameter:	**Conditional**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	TRUE	
Address range:	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Using in typicals	Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S, Dat256D_S	
Explanation:	Conditional spontaneous data transfer You will information on this below with the parameter "Unconditional".	

Unconditional

Parameter:	**Unconditional**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Using in typicals	Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S, Dat256D_S	

Explanation: Unconditional spontaneous data transfer

With the two parameters "Conditional" and "Unconditional" you decide whether a data frame is transferred immediately by the module if the value changes (unconditional spontaneous) or at a later point in time (conditional spontaneous).

The two parameters are programmed as follows:

- **Conditional spontaneous transfer (not necessarily immediately)**
  - Conditional = TRUE
  - Unconditional = FALSE
- **Unconditional spontaneous transfer (immediately)**
  - Conditional = FALSE
  - Unconditional = TRUE

The default of the two parameters was chosen so that a data frame is not transmitted immediately.

The decision whether a data frame is transferred immediately or later only relates to dial-up networks.

In a dial-up network you must decide from case to case whether a change to the value of the data point requires immediate transfer and therefore the immediate establishment of a connection. This can, for example, be required for data points with alarms.

On a dedicated line, the transmission is always immediate even if the combination of "Conditional" and "Unconditional" is set to "not immediately". On dedicated lines, you do not need to make changes to the two parameter settings.

Permanent

Parameter: **Permanent**

Declaration: INPUT

Data type: BOOL

Default: FALSE

Explanation: Permanent data transfer

This parameter has no significance and is always = FALSE.

Note

The "Permanent" parameter is no longer supported, it has been retained to ensure compatibility.

TimeStamp

Parameter:	**TimeStamp**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Using in typicals	Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S, Set01W_R, Par12D_R
Explanation:	Time stamp

- TRUE

  The data frame will be transferred with a time stamp.

  The prerequisite is that the time provided by the local TIM is available in the CPU. For more detailed information, refer to the description of FC TimeTask.

  Note
  With Ana04W_S remember the dependency of the time stamp on the parameter "MeanValueGeneration", see Analog value typical Ana04W_S (Page 160).
- FALSE

  The data frame is transferred without a time stamp.

For information on the format of the time stamp, refer to the section Time stamp (Page 154).

NewData

Parameter:	**NewData**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Output	Q 0.0 ... Q n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Using in typicals	Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Cmd01B_R, Set01W_S, Set01W_R, Par12D_S, Par12D_R, Dat12D_R	

Explanation: Receive new data

The NewData output is intended for user-specific further processing, for example to react in a specific way to receipt of new data.

Whenever the block has received new data and has output it to the outputs for the specific typical, NewData is set to TRUE for one OB1 cycle.

You will find the specific outputs in the description of the individual data point typicals.

With the data point typicals "Set01W_R" and "Par12D_R", NewData is also set to TRUE for one OB1 cycle if there is a new local value entered in the Local = 1 state.

If you do not require the parameter, simply leave it open.

5.7.1.3 Time stamp

Format of the SINAUT time stamp

For many data point typicals you can use the TimeStamp parameter to instruct that the data of the object should be transferred with a time stamp.

However for the receiving data point typicals there is no output parameter with which to output the received time stamp. The time stamp is only saved in the instance DB which you have specified when calling the respective receive typical. To further process the time stamp, the data must be read out of the DB by the user program.

The time stamp is saved in two data double words that have the same name in all object DBs:

Name of the double word	Contents
RecTimeStamp_1	Year, month, day and hour
RecTimeStamp_2	Minute, second, millisecond and time status

With the exception of the half byte with the time status, the date and time are coded in BCD format.

Table 5- 5 Assignment of the structure of the time stamp

Name of the double word	Byte no.	Contents	
		High nibble	Low nibble
RecTimeStamp_1	0	Year * 10	Year * 1
	1	Month * 10	Month * 1
	2	Day * 10	Day * 1
	3	Hour * 10	Hour * 1
RecTimeStamp_2	0	Minute * 10	Minute * 1
	1	Second * 10	Second * 1
	2	Millisecond * 100	Millisecond * 10
	3	Millisecond * 1	Time status

Table 5- 6 The bit assignment of the half byte "time status" (low nibble of byte 3 of RecTimeStamp_2)

Bit no.	Value	Meaning
0	0	Time is invalid
	1	Time is valid
1	0	Standard time
	1	Daylight saving time
2		Not used
3		Not used

The time double words occupy different addresses depending on the typical. Look in the instance DB or in the declaration header of the FB to find the absolute address of both double words.

It is more convenient to give the instance DBs symbolic names. You can then use the symbolic addresses to read out the information. In this case, you do not need to worry about the actual absolute addresses. These are used automatically by STEP 7. The following example clarifies this procedure.

Example

Symbolic name of instance DB: `ObjectDB27`

The STEP 7 program for reading the date and time of day and for saving in DB20 beginning with data byte 100 may appear as follows programmed in STL:

```
L "ObjectDB27".RecTimeStamp_1

T DB 20.DBD 100

L "ObjectDB27".RecTimeStamp_2

T DB 20.DBD 104
```

5.7.1.4 Analog value typical Ana04R_S

Validity

S7-1500 only

Function

Send 4 analog values as 32-bit floating-point number

Ana04R_S transfers the 4 analog values as instantaneous values. At the time of the transfer, the currently pending analog value is acquired and transferred to the partner.

Note

Common processing of the four analog values

The processing parameters such as threshold, smoothing factor etc. exist only once in a typical. These parameters apply to all 4 analog values in common; in other words, it is not possible to set the parameters for the individual analog values. For this reason, each typical should only acquire analog values that should be processed in the same way.

Parameter

Parameter:	**PartnerNo** **PartnerObjectNo** **Enabled** **ImageMemory** **Conditional** **Unconditional** **TimeStamp**

For a description, see section Reoccurring parameters (Page 144).

Parameter:	**ThresholdIntegration**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Threshold processing according to the integration principle With this parameter, you can specify whether the integration principle is used in threshold value processing. With the default FALSE the threshold value is calculated without integration. In this case, you can expect less data traffic on the telecontrol line and locally between CPU and TIM.

Parameter:	**ZeroLimitation**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	TRUE
Explanation:	Zero limitation If this parameter is activated negative values are be suppressed and replaced with the value 0.

Parameter:	**TriggerInput**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation:	Trigger input

With the edge change 0 → 1 of the "TriggerInput" input the transfer of the analog value frame can be triggered at a required time.

Example:
Time-driven analog value transfer with time stamp for supplying an analog value archive in the control center.

Make sure that you set the "ImageMemory" parameter to FALSE to prevent these data frames with time stamps from being overwritten when saving on the station TIM.

The FC Trigger block can be used for time-driven triggering of a transmission over "TriggerInput".

If you do not require the parameter, simply leave it open. The transfer should then be triggered based on the "ThresholdValue" and "ThresholdIntegration" threshold parameters.

"TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. With dedicated lines or wireless networks the transfer is immediate. With a dial-up connection, it is possible that the data frame is saved first on the TIM and sent at a later point in time. The reason can, for example, be that the data frame is marked as "Conditional spontaneous", see parameter "Conditional".

Parameter:	**AnalogInput_1 ... _4**	
Declaration:	INPUT	
Data type:	REAL	
Range of values:	See address range	
Default:	0.0	
Address range:	Bit memory address	MD0 ... MDn
	Data address	LD0 ... LDn DBm.DBD0 ... DBDn
Explanation:	Analog input word	

For each analog value to be transmitted in the data frame, you can specify from where the FB will take the analog information. Any combination of a data block and the bit memory address area can be used.

If you do not require parameters, simply leave them open. The value 0 is transferred for these analog inputs in the data frame.

Parameter:	**SamplingPeriod**
Declaration:	INPUT
Data type:	INT
Range of values:	0 ... 32767 [ms]

Default: 500

Explanation: Acquisition interval for analog inputs in milliseconds

The acquisition interval is required for the following parameters:

- Threshold formation according to the integration principle (ThresholdIntegration)
- Smoothing the analog input value (SmoothingFactor)

The value must be selected high enough so that it is certain that a new value was acquired within the encryption time of the analog input. The interval to be specified has to be at least as long as the encoding time of the analog input module being used at the selected resolution (8 ... 15 bits).

The value must also be selected generously so that analog values are acquired even with the highest resolution and with analog modules with the highest number of inputs.

Specifying a "SamplingPeriod" that is too short may lead to an overflow of the internal accumulation counter.

Parameter: **ThresholdValue**

Declaration: INPUT

Data type: REAL

Range of values: +1.175495e-38 ... +3.402823e+38

Default: 1.0

Explanation: Threshold value

Without configuration, the default value 1.0 is used.

Point to note with "ThresholdValue" = 0: Changes are not checked based on the threshold value. The analog value frame will only be sent in the following situations:

- When there is a trigger via the "TriggerInput" input, typically a time-driven or event-driven trigger.
- When there is a general request to the station or a single request for the data frame.

Parameter: **SmoothingFactor**

Declaration: INPUT

Data type: REAL

Range of values: 1.0 (no smoothing)

4.0 (weak smoothing)

32.0 (medium smoothing)

64.0 (strong smoothing)

Default: 4.0

Explanation:	Smoothing factor
	Using the smoothing factor, quickly fluctuating analog values can be smoothed to a greater or lesser extent depending on the factor. This may allow a narrower threshold band to be selected (ThresholdValue).
	The smoothing factors are identical to the smoothing factors that are configured for some S7 analog input modules. The smoothing is calculated using the same formula as on the input module:
	$y_n = \frac{x_n + (k-1)\ y_{n-1}}{k}$
	y_n = smoothed value in the current cycle n
	y_{n-1} = smoothed value in previous cycle n-1
	x_n = acquired value in the current cycle n
	k = smoothing factor

5.7.1.5 Analog value typical Ana04R_R

Validity

S7-1500 only

Function

Receive 4 analog values as 32-bit floating-point number

Parameter

Parameter:	**PartnerNo**
	PartnerObjectNo
	Enabled
	For a description, see section Reoccurring parameters (Page 144).

Parameter:	**AnalogOutput_1 … _4**	
Declaration:	OUTPUT	
Data type	REAL	
Range of values:	See address range	
Default:	0.0	
Address range:	Bit memory address	MD0 ... MDn
	Data address	LD0 ... LDn DBm.DBD0 ... DBDn
Explanation:	You can select where the individual analog values received by the FB are output. Addresses of a data block and in the bit memory address area can be mixed as required.	
	If you do not require parameters, simply leave them open.	

Parameter:	**NewData** For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).
Typical-specific response:	Whenever the FB has received new data and has output it to the outputs "AnalogOutput_1" to "AnalogOutput_4", the "NewData" output is set to TRUE for one OB1 cycle.

5.7.1.6 Analog value typical Ana04W_S

Function

Send 4 analog values as 16 bit values

Ana04W_S transfers the 4 analog values alternatively:

- As instantaneous values

  At the time of the transfer, the currently pending analog value is acquired and transferred to the partner.
- As mean values

  The pending analog value is accumulated at selectable intervals. At the time of the transmission, a mean value is formed from the total value and transferred to the partner.

Note

Common processing of the four analog values

The processing parameters such as threshold, smoothing factor etc. exist only once in a typical. These parameters apply to all 4 analog values in common; in other words, it is not possible to set the parameters for the individual analog values. For this reason, each typical should only acquire analog values that should be processed in the same way.

Parameter

Parameter: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional

For a description, see section Reoccurring parameters (Page 144).

Parameter: **TimeStamp**

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).

Typical-specific functions	With the setting TRUE the time stamp depends on the setting of the parameter "MeanValueGeneration": • MeanValueGeneration = FALSE Instantaneous values are transferred in the data frame. The time stamp in the data frame is identical to the time of acquisition of the instantaneous values contained in the data frame. • MeanValueGeneration = TRUE The data frame contains mean values. The time stamp is identical to the time at which the mean value calculation period was completed. The start of the mean value calculation period is not included in the data frame. This is, however, identical to the time stamp of the previously transferred mean value frame.

Parameter:	**ThresholdIntegration**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Threshold processing according to the integration principle With this parameter, you can specify whether the integration principle is used in threshold value processing. With the default FALSE the threshold value is calculated without integration. In this case, you can expect less data traffic on the telecontrol line and locally between CPU and TIM. When MeanValueGeneration = TRUE, (analog values are sent as mean values), the "ThresholdIntegration" parameter has no meaning.

Parameter:	**ZeroLimitation**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	TRUE
Explanation:	Zero limitation If this parameter is activated negative values are be suppressed and replaced with the value 0.

Parameter:	**TriggerInput**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Trigger input	

With the edge change 0 → 1 of the "TriggerInput" input the transfer of the analog value frame can be triggered at a required time.

Example:
Time-driven analog value transfer with time stamp for supplying an analog value archive in the control center.

Make sure that you set the "ImageMemory" parameter to FALSE to prevent these data frames with time stamps from being overwritten when saving on the station TIM.

If the block calculates mean values, the duration of the calculation period is decided by the "TriggerInput" input. The current period is ended and a new period begun each time a transmission is triggered by this input. The interval between triggering a data frame twice determines the duration of the mean value calculation period.

The FC Trigger block can be used for time-driven triggering of a transmission over "TriggerInput".

If you do not require the parameter, simply leave it open. The transfer should then be triggered based on the "ThresholdValue" and "ThresholdIntegration" threshold parameters.

"TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values/mean values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. With dedicated lines or wireless networks the transfer is immediate. With a dial-up connection, it is possible that the data frame is saved first on the TIM and sent at a later point in time. The reason can, for example, be that the data frame is marked as "Conditional spontaneous", see parameter "Conditional".

Parameter:	**MeanValueGeneration**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE

Default: FALSE

Explanation: Mean value generation

If the parameter is enabled the analog values to be acquired are transferred as mean values.

If you select mean value generation, the currently pending analog value is acquired cyclically and accumulated. The acquisition cycle depends on the "SamplingPeriod" parameter (for example 500 ms, see also the description of this parameter). The mean value is calculated from the accumulated values as soon as a transmission is triggered via the "TriggerInput" input. Following this, the accumulation starts again so that the next mean value can be calculated.

The mean value can also be calculated if the transmission of the analog value frame is triggered by a general or single request. The duration of the mean value calculation period is then the time from the last transmission (for example triggered via TriggerInput) to the time of the general or single request. Once again, the accumulation restarts so that the next mean value can be calculated.

If the acquired analog value is above or below the permitted range (7FFFH bzw. 8000H), this value can either be taken into account immediately in the calculation of the mean value or it can be suppressed for a specific period for the calculation of the mean value. The required response can be decided with the "FaultSuppressionTime" parameter:

- **FaultSuppressionTime = 0**

  Acquisition of a value above or below the over- or underrange results in immediate cancellation of the mean value calculation. The value $7FFF_H$ or 8000_H is saved as an invalid mean value for the current mean value calculation period and sent when the next analog value frame is triggered. The calculation of a new mean value is then started. If the analog value remains in the overshoot or undershoot range, this new value is again saved immediately as an invalid mean value and sent when the next frame is triggered.

- **FaultSuppressionTime > 0**

  If the acquired analog value is in the overshoot or undershoot range, the bad values are excluded from the calculation of the mean value for a maximum duration as defined by the FaultSuppressionTime. If this time is exceeded, the value $7FFF_H$ or 8000_H is saved as an invalid mean value and sent when the next analog value frame is triggered. The procedure is identical in each new mean value calculation period averaging period; in other words, bad values are again suppressed for the duration of the "FaultSuppressionTime".

  The duration of the "FaultSuppressionTime" also indirectly decides the proportion of invalid values per mean value calculation period. For example, if the mean value is calculated every 15 minutes and "FaultSuppressionTime" is set to 5 minutes, the mean value is only sent as invalid when more than 1/3 of the analog values acquired are above or below the overshoot or undershoot range in the current mean value calculation period.

Parameter:	**AnalogInput_1 ... _4**	
Declaration:	INPUT	
Data type:	WORD	
Range of values:	See address range	
Default:	0 (W#16#0)	
Address range:	I/O words	PIW0 ... PIWn
	Memory words	MW0 ... MWn
	Data words	LW0 ... LWn DBm.DBW0 ... n
Explanation:	Analog input word For each analog value to be transmitted in the data frame, you can specify from where the FB will take the analog information. I/O words from analog input modules, data words from a data block and memory words can be mixed as required. If you do not require parameters, simply leave them open. The value 0 is transferred for these analog inputs in the data frame.	

Parameter: **SamplingPeriod**
Declaration: INPUT
Data type: INT
Range of values: 0 ... 32767 [ms]
Default: 500
Explanation: Acquisition interval for analog inputs in milliseconds

The acquisition interval is required for the following parameters:

- Threshold formation according to the integration principle (Threshold Integration)
- Smoothing of the analog input value (Smoothing Factor)
- Mean value generation

The value must be selected high enough so that it is certain that a new value was acquired within the encryption time of the analog input. The interval to be specified has to be at least as long as the encoding time of the analog input module being used at the selected resolution (8 ... 15 bits).

The value must also be selected generously so that analog values are acquired even with the highest resolution and with analog modules with the highest number of inputs.

If mean values are calculated, SamplingPeriod should not be less than 500 ms. If mean values are calculated over very long periods, the time must be increased as follows:

- Mean value calculation period 12 h: SamplingPeriod = 1000 [ms]
- Mean value calculation period 24 h: SamplingPeriod = 2000 [ms]

Specifying a "SamplingPeriod" that is too short may lead to an overflow of the internal accumulation counter. The maximum value of 2 147 483 647 of a double integer must not be exceeded. When an overflow is detected, the invalid mean value of 8000_H is transferred for the current mean value calculation period.

Parameter:	**ThresholdValue**
Declaration:	INPUT
Data type:	INT
Range of values:	0 / 1 ... 32767
Default:	270
Explanation:	Threshold value

When specifying the threshold value, take the encryption range of the analog values into account. Raw values from S7 analog inputs are always encoded in the range from 0 ... 27648 (= 0 ... 100 %) or + 27648 (= + 100%). Depending on the resolution of the analog input, the value jumps by 128 (at 8-bit resolution) or 1 (at 15-bit resolution). If the acquired analog values have a different encoding range, specify a threshold value oriented toward this.

If the parameter is not configured, the default value of 270 is used. This corresponds to approximately 1% of the normal S7 analog raw value range.

Point to note with "ThresholdValue" = 0

Changes are not checked based on the threshold value. The analog value frame will only be sent in the following situations:

- When there is a trigger via the "TriggerInput" input, typically a time-driven or event-driven trigger.
- When there is a general request to the station or a single request for the data frame.
- When the analog value moves into the overshoot or undershoot range ($7FFF_H$ or 8000_H) (possibly after the suppression time set for "FaultSuppressionTime" has elapsed).

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the "ThresholdValue" parameter has no meaning.

Parameter:	**SmoothingFactor**
Declaration:	INPUT
Data type:	INT
Range of values:	1 (no smoothing) 4 (weak smoothing) 32 (medium smoothing) 64 (strong smoothing)
Default:	1

Explanation: Smoothing factor

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the "ThresholdValue" parameter has no meaning.

Using the smoothing factor, quickly fluctuating analog values can be smoothed to a greater or lesser extent depending on the factor. This may allow a narrower threshold band to be selected (ThresholdValue).

The smoothing factors are identical to the smoothing factors that are configured for some S7 analog input modules. The smoothing is calculated using the same formula as on the input module:

$$y_n = \frac{x_n + (k - 1)\ y_{n-1}}{k}$$

y_n = smoothed value in the current cycle n

y_{n-1} = smoothed value in previous cycle n-1

x_n = acquired value in the current cycle n

k = smoothing factor

Parameter:	**FaultSuppressionTime**
Declaration:	INPUT
Data type:	INT
Range of values:	0 ... 32767
Default:	0

Explanation: Fault suppression time in seconds.

Transmission of an analog value located in the overshoot or undershoot range ($7FFF_H$ or 8000_H) is suppressed for the time period specified here. The value $7FFF_H$ or 8000_H is only sent after this time has elapsed, if it is still pending. If the value returns to below $7FFF_H$ or above 8000_H again before this time elapses, it is immediately sent again as normal. The suppression time is started again for the full duration the next time $7FFF_H$ or 8000_H is acquired.

This is typically used for temporary suppression of current values that may occur when powerful pumps and motors are started. The analog input may exceed the maximum range several times under some circumstances. Suppression prevents these values from being signaled as faults in the control center system.

The suppression is adjusted to analog values that are acquired by the S7 analog input modules as raw values. These modules return the specified values for the overflow or underflow range for all input ranges (also for life-zero inputs). With provided ready values, fault suppression is only possible if these also adopt the values $7FFF_H$ or 8000_H when there is overshoot or undershoot. If this is not the case, the parameter does not need to have a value entered.

The parameter can also be used in combination with the mean value calculation for temporary suppression of the values $7FFF_H$ or 8000_H (see parameter MeanValueGeneration).

If no parameter is specified, the default of 0 seconds applies. An acquired value of $7FFF_H$ or 8000_H is then sent immediately when it is first detected or, with mean value calculation, as an invalid mean value for the current mean value calculation period.

5.7.1.7 Analog value typical Ana04W_R

Function

Receive 4 analog values as 16-bit values

Parameter

Parameter: **PartnerNo**

PartnerObjectNo

Enabled

For a description, see section Reoccurring parameters (Page 144).

Parameter: **AnalogOutput_1 … _4**

Declaration: OUTPUT

Data type WORD

Default: TRUE

Explanation 0 (W#16#0)

Range of values:	I/O words	PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

You can select where the individual analog values received by the FB are output. I/O words from analog output modules, data words from a data block and memory words can be mixed as required.

If you do not require parameters, simply leave them open.

Parameter: **NewData**

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).

Typical-specific response: Whenever the FB has received new data and has output it to the outputs "AnalogOutput_1" to "AnalogOutput_4", the "NewData" output is set to TRUE for one OB1 cycle.

5.7.1.8 Binary value typical Bin04B_S

Function

Send 4 bytes of binary information

Parameter

Parameter: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional
TimeStamp

For a description, see section Reoccurring parameters (Page 144).

Parameter:	**InputByte_1 … _4**
Declaration:	INPUT
Data type:	BYTE
Range of values:	See address range
Default:	0 (B#16#0)

Address range:	Input bytes	IB0 ... IBn PIB0 ... PIBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n

Explanation: Input byte

Specify the memory area (1 to 4 bytes) of the binary information to be transferred by the FB.

Input bytes from the process input image, I/O bytes directly from digital input modules, data bytes from a data block and memory bytes can be mixed.

If you do not require parameters, simply leave them open.

For unconfigured bytes, the value 0 (zero) is transferred.

Parameter: **DisableMask**

Declaration: INPUT

Data type: DWORD

Range of values: 0 ... 2147483647

- As 32 bit binary number

  2#0 ... 2#11111111_11111111_11111111_11111111
- As 32 bit hexadecimal number

  DW#16#0 ... DW#16#FFFF_FFFF

Default: 0 (2#0)

Explanation: Disable mask

- For every input to be blocked enter a 1 at the relevant position in the bit pattern.
- For the other inputs enter a 0.

A disabled input always has the value 0 (zero) during the transfer.

For the assignment of the 32 inputs from "InputByte_1" to "InputByte_4" to the 32 bits of the blocking mask, see the following table.

	InputByte_1								InputByte_2								InputByte_3								InputByte_4							
Bit	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0
2#	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
DW#16#	_				_				_				_				_				_				_				_			

Parameter: **InversionMask**

Declaration: INPUT

Data type: DWORD

Range of values: 0 ... 2147483647

- As 32 bit binary number

  2#0 ... 2#11111111_11111111_11111111_11111111
- As 32 bit hexadecimal number

  DW#16#0 ... DW#16#FFFF_FFFF

Default: 0 (2#0)

Explanation: Inversion mask

The inversion of input signals can, for example, be useful when using a mixture of sensors operating on the open and closed circuit principle.

- For every input to be inverted enter a 1 at the relevant position in the bit pattern.
- For the other inputs enter a 0.

For the assignment of the 32 inputs from "InputByte_1" to "InputByte_4" to the 32 bits of the inversion mask, see the following table.

	InputByte_1								InputByte_2								InputByte_3								InputByte_4							
Bit	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0
2#	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
DW#16#	_				_				_				_				_				_				_				_			

5.7.1.9 Binary value typical Bin04B_R

Function

Receive 4 bytes of binary information

Parameter

Parameter: **PartnerNo**

PartnerObjectNo

Enabled

For a description, see section Reoccurring parameters (Page 144).

Parameter: **OutputByte_1 … _4**

Declaration: OUTPUT

Data type: BYTE

Range of values: See address range.

Default: 0 (B#16#0)

Address range:	Output bytes	QB0 ... QBn PQB0 ... PQBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n

Explanation: Output byte

Specify the memory area (1 to 4 bytes) of the binary information to be output in the binary information.

Output bytes from the process image output, I/O bytes directly to digital output modules, data bytes from a data block and memory bytes can be mixed.

If you do not require parameters, simply leave them open.

For information on reading out the time stamp received with the data using the user program, see the section Time stamp (Page 154).

Parameter: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).

The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the output bytes "OutputByte_1" to "OutputByte_4"

5.7.1.10 Command typical Cmd01B_S

Function

Send 1 byte commands with 1-out-of-8 check

The 1-out-of-8 check is performed by the data point typical.

The 1-out-of-n check is performed by FC Safe.

Note

FC Safe required

With Cmd01B_S, data can only be transmitted when FC Safe is linked in cyclic program.

Parameter

Parameter: **PartnerNo**

PartnerObjectNo

For a description, see section Reoccurring parameters (Page 144).

Parameter:	**Enabled**
	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).
Typical-specific response:	Enable block processing If processing is disabled the FB only checks to see if the disabled status has been canceled. The block cannot communicate at the organizational level in this status because it cannot send any Org. frames.

Parameter:	**CommandInputByte_HW**	
Declaration:	INPUT	
Data type:	BYTE	
Range of values:	See address range	
Default:	0 (B#16#0)	
Address range:	Input bytes	IB0 ... IBn PIB0 ... PIBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n

Explanation: Command input byte for hardware input.

This command input byte is specially designed for entering commands using hardware, i.e. via digital inputs. Input using memory or data bytes is also possible, but you must then make sure that the command pending at the input byte is reset, which occurs during hardware input when the command button is released.

When input is detected, if no error is detected during the 1-out-of-8 and 1-out-of-n check, and if the central enable memory bit is set, the command is transferred. This is automatically set by FC Safe following a selected time delay set there (see FC Safe, "InputDelayTime" parameter).

If a 1-out-of-8 or 1-out-of-n error is detected, the entered command is no longer processed. A new command is only read in again if previously for the time of one OB1 cycle no hardware command was detected in the CPU at this or another command input block with a hardware input.

The FB enters a detected 1- out-of-8- or 1-out-of-n error in the diagnostic buffer (event ID B171 or B172). The error status is also indicated via the "InputError" output of FC Safe (see FC Safe, "InputError" parameter) and continues to be indicated as long as the error remains.

Name:	**CommandInputByte_SW**
Declaration:	IN_OUT
Data type	BYTE
Range of values:	See address range

Default:	0 (B#16#0)	
Address range:	Memory bytes	MB0 ... MBn
	Data bytes	DBm.DBB0 ... n

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Explanation: Command input byte for software input

This command input byte is specially designed for entering commands using software, i.e. by the user program or an operator panel (OP). When an input is detected and if no error is detected during the 1-out-of-8 and 1-out-of-n check, the command id reset at the input byte and transferred. The central enable memory bit is ignored here because it is only intended for command input over hardware (see "CommandInputByte_HW").

If a 1-out-of-8 or 1-out-of-n error is detected, the entered command is no longer processed. A new command is only read in again if previously for the time of one OB1 cycle no software command was detected in the CPU at this or another command input block with a software input.

The FB enters a detected 1- out-of-8- or 1-out-of-n error in the diagnostic buffer (event ID B171 or B172). Appropriate error bits are also set in the central data block BasicData where they can be queried by the software. For further details, refer to the description of FC Safe.

In principle it is possible to create a new command at "CommandInputByte_SW" in every OB1 cycle. However, only one command per OB1 cycle is allowed and this applies to all command input blocks with software input (1-out-of-n check). An ’empty cycle’ between two consecutive software commands is therefore not necessary.

Note

CommandInputByte_HW / CommandInputByte_SW

If the same commands need to be entered over the hardware and software, the two command inputs "CommandInputByte_HW" and "CommandInputByte_SW" can also be used at the same time.

If a command is entered at the same time over both input bytes, this is only accepted when exactly the same command is entered over the hardware as well as the software input. The hardware input is then processed further.

In all other cases the input is rejected and an error message is entered in the diagnostic buffer (Event ID B170). The error status is also indicated via the InputError output of FC Safe. Appropriate error bits are set in the central data block BasicData where they can be queried by the software (see FC Safe).

5.7.1.11 Command typical Cmd01B_R

Function

Receive 1 byte commands (1-out-of-8 format)

Parameter

Parameter:	**PartnerNo**
	PartnerObjectNo
	For a description, see section Reoccurring parameters (Page 144).

Parameter:	**Enabled**
	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).
Typical-specific response:	Enable block processing If processing is disabled the FB only checks to see if the disabled status has been canceled. Any commands that are still received are not output. The FB cannot communicate at the organizational level in this status because Cmd01B_R cannot send or receive Org. frames. If the "Enabled" input should be made operable by a switch, this local disable means received commands are no longer output. Since the block is, however, not capable of sending Org. frames, it cannot report this local block back to the partner itself. This must be done with another typical, e.g. Bin04B_S.

Parameter:	**MultipleOutput**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Simultaneous output of multiple commands permitted With this parameter, you can specify whether or not several (consecutively received) commands can be output simultaneously; in other words, you specify how the block reacts when a new command is received and the previously received command still needs to be output. Requirement: The command output time has not yet elapsed and the user program has not yet reset this command. • FALSE Multiple output is not permitted. The newly received command overwrites the output byte. Any command still pending is therefore reset to 0 unless the new command is identical to the old one. • TRUE Multiple output is permitted. A newly received command is written to the current output byte. The command output time is restarted and applies to all pending commands.

Parameter:	**CommandOutputTime**
Declaration:	INPUT

Data type:	INT
Range of values:	0 ... 500
Default:	500
Explanation:	Command output time for command outputs in milliseconds The specified time applies to all command outputs. If more than one output can be set at the same time (MultipleOutput = TRUE), the output time is restarted with each newly received command. This means retriggering for already pending commands. All the command outputs are reset at the same time onöy when the output time elapses. With the value 0 a set command output is not reset by the command typical. You need to do this via the user program.

Parameter:	**NewData**
Explanation:	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144). The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the output byte "CommandOuputByte".

Parameter:	**CommandOutputByte**	
Declaration:	IN_OUT	
Data type:	BYTE	
Range of values:	See address range	
Default:	0 (B#16#0)	
Address range:	(process image) output bytes	QB0 ... QBn
	Memory bytes	MB0 ... MBn
	Data bytes	DBm.DBB0 ... n
	Since the parameter is an IN_OUT parameter, direct I/O output of the command byte to PQB0 ... PQBn is not permitted. It is also difficult to specify local bit memory with this parameter type and this should not be used.	
Explanation:	Command output byte To allow the command outputs to be reset both by the command typical itself as well as by the user program (when output time = 0), the parameter was declared as an IN_OUT parameter.	

5.7.1.12 Counted value typicals Cnt01D_S / Cnt04D_S

Function

- Cnt01D_S: Send 1 counted value (32 bits).
- Cnt04D_S: Send 4 counted values (32 bits).

  Note that the parameter "DifferenceValue" for forming the difference value can only be activated at the same time for all four counted values.

Parameter

Parameter: **PartnerNo**

PartnerObjectNo

Enabled

ImageMemory

Conditional

Unconditional

TimeStamp

For a description, see section Reoccurring parameters (Page 144).

Parameter:	**GeneralTriggerCommand**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Restore collective command

The restore collective command as a central system memory bit belongs to the organizational SINAUT system commands.

Set the parameter to TRUE if the counted value transfer is to be triggered by a restore collective command.

- When the destination subscriber no. (PartnerNo) = 0 (transfer to all), the restore collective command is taken into account.

  When the restore collective command is detected, the currently accumulated counted value is transferred regardless of other triggers for transfer. The restore bit is inverted in this counted value.
- If in the typical an explicit destination subscriber no. (PartnerNo > 0) is configured, the restore collective command is evaluated in the corresponding subscriber object in the central administration.

You can use the parameters "GeneralTriggerCommand" and "TriggerInput" at the same time. In this case the transfer is triggered both by an edge change 0 → 1 at "TriggerInput" and when a restore collective command is received.

Parameter:	**TriggerInput**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Trigger input With the edge change 0 → 1 of the "TriggerInput" input the triggered transfer can be triggered at a required time regardless of other criteria for a transfer.. The currently accumulated counted value is transmitted. The restore bit (see above) is inverted in this counted value. Example: Time-driven transmission with time stamp for supplying an archive in the control center. You can use the parameters "GeneralTriggerCommand" and "TriggerInput" at the same time. In this case the transfer is triggered both by an edge change 0 → 1 at "TriggerInput" and when a restore collective command is received. With the setting FALSE no restoring and no transfer via the "TriggerInput" input is triggered.	

Parameter:	**Counter_1 (Cnt01D_S)** **Counter_1 ... _4 (Cnt04D_S)**
Declaration:	INPUT
Data type:	COUNTER
Range of values:	0 ... 32767 • Z0 as placeholder or • Z1 ... Zn n depends on the CPU type.
Default:	-
Explanation:	Number of the SIMATIC counter Here, you specify the SIMATIC counter in which the pulses were counted time-driven. This counting takes place in the background using FC PulseCounter that is called in a cyclic interrupt OB (for example in OB35). See also section FC PulseCounter (Page 239) and Cyclic interrupt OB (Page 138). The COUNTER data type cannot be preassigned a value. If you configure the Z0 placeholder, the corresponding counted value is not processed.

Parameter:	**DifferenceValue**
Declaration:	INPUT
Data type:	INT
Range of values:	0 ... 31767
Default:	0
Explanation:	Difference value

- When a value between 1 and 31767 is configured, the counted value is transferred as soon as the difference between the current and most recently transferred counted value reaches the value specified here.
- If the default value 0 is configured, a counted value is transferred only in the following situations:
  - On an edge change 0 → 1 at the "TriggerInput" input
  - On receipt of a restore command when "GeneralTriggerCommand" = TRUE.

Select the difference value dependent on the maximum pulse rate per second.

Do not select a value that is too low so that the counted value is not constantly transferred to the TIM. This would put load on the communications path to the CPU and the send queue of the CPU.

Note on Cnt04D_S
In the typical this processing parameter for forming the difference value only exists once. It applies to all 4 counted values together. It is not possible to set the parameter for the individual counted values. When using this parameter, each typical should therefore only acquire counted values that can be processed identically.

5.7.1.13 Counted value typicals Cnt01D_R / Cnt04D_R

Function

- Cnt01D_S: Receive 1 counted value (32 bits)
- Cnt04D_S: Receive 4 counted values (32 bits)

Parameter

Parameter: **PartnerNo**
PartnerObjectNo
Enabled

For a description, see section Reoccurring parameters (Page 144).

Parameter:	**BCD_Format**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	TRUE
Explanation:	Counted value output in BCD format • If the parameter is activated the received counted value is output as a positive BCD value at the "CountedValueOutput_n" output. • If the parameter is deactivated the counted value is output as a positive 32-bit integer value. For the different value ranges of the two formats, see the parameter "CountedValueOutput_n". If the maximum counted value that can be represented is exceeded, the counted value starts again at 0 and counting continues in the positive numeric range. **Note on Cnt04D_R** The parameter exists only one in the typical and it applies to all 4 counted values together. It is not possible to make an individual setting per counted value. When using this parameter, per typical only counted values with an identical output format should be output.

Parameter:	**CntValInvalid**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Output	Q 0.0 ... Q n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation:	Counted value invalid When evaluating "CntValInvalid" remember that the bit might only be set for one OB1 cycle. The CntValInvalid output indicates whether the last received counted value was invalid. . With "Cnt04D_R", this counts as a group display for all 4 counted values, see the note below. The output shows the validity status of the most recently received counted value in inverted form. The output serves the following purposes: • Error display • Signal for user-specific further processing You can, for example, react to the lack of up-to-dateness, by correcting the counted value output at "CountedValueOutput_n" with possibly lost counting pulses. If you do not require the parameter, simply leave it open. **Note on Cnt04D_R** Although all 4 counted values have their own status bit in the data frame, for the status at the output "CntValInvalid" only the bit of the first counted value in the previously received data frame is evaluated. This status, however, applies to all 4 counted values. (All counted values in the data frame always have the same status.)	

Parameter:	**RestoreStatus**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Output	Q 0.0 ... Q n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation:	Status of the restore bit in the received counted value. The "RestoreStatus" output indicates the current status of the restore bit from the last received counted value frame. You can use the output for user-specific further processing. Example: You can only access the information at "CountedValueOutput_n" when a change has been detected at the "RestoreStatus" output; in other words, when the counted value has been received due to a restore, such as a local time-driven restore. If you do not require the parameter, simply leave it open. **Note on Cnt04D_R** Although all 4 counted values have their own restore bit in the data frame, for the status at the "RestoreStatus" output only the restore bit of the first counted value in the previously received data frame is evaluated. This status, however, applies to all 4 counted values. (All counted values in the data frame always have the same status.)

Parameter:	**NewData**
Explanation:	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144). The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the output / outputs "CountedValueOutput_1" to "CountedValueOutput_4"

Parameter:	**CountedValueOutput_1 (Cnt01D_S)** **CountedValueOutput_1 ... _4 (Cnt04D_S)**	
Declaration:	IN_OUT	
Data type:	DWord, UDInt	
Range of values:	• Integer:	0 ... 2 147 483 647
	• BCD:	0 ... 9 999 999
Default:	0	
Address range:	Output (DWORD)	QD0 … QDn
	Bit memory (DWORD)	MD0 ... MDn
	Data (DWORD)	DBm.DBB0 … n
	Since the parameter is an in-out parameter (declaration IN_OUT), direct I/O output of the counted value to PQD0 ...PQDn is not permitted. It is also difficult to specify local bit memory with this parameter type and this should not be used.	

Explanation: Counted value output

The counted value typical always adds the newly formed difference value (difference between the new and last received counted value) to the value currently output at the counted value output.

The counted value output is a double word in which the counted value is stored in BCD format or as a 32-bit integer value (depending on the "BCD_Format" parameter, see above).

The counted value id always output as a positive number. If the maximum counted value that can be represented is exceeded, the counted value starts again at 0 and counting continues in the positive numeric range.

Since the parameter is an in-out parameter (IN_OUT), the value can be reset to 0 or another value at the counted value output by the user program at any time.

5.7.1.14 Data typical Dat12D_S

Function

Send maximum of 12 double words with any data content.

The content of each double word may be a value in double word (DWORD, DINT, REAL) format, it can also be a mixture of other data types which together form a double word, for example:

- 4 bytes
- 2 words
- 2 bytes + 1 word

Sending the data area can be triggered in two ways:

- By a change check

  The data is transferred as soon as a bit changes ("SendOnChange" = TRUE).
- By the user program

  The transfer can be triggered by an edge change 0 → 1 at the "TriggerInput" input

  For time-driven transfer FC Trigger can be used.

With "SendAll" you can also specify whether the transfer always includes all data or only the data double words that have changed.

Note

Remember double word boundaries

When changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 12 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameter

Parameter:	**PartnerNo** **PartnerObjectNo** **Enabled** **ImageMemory** **Conditional** **Unconditional** **TimeStamp** For a description, see section Reoccurring parameters (Page 144).

Parameter:	**SendOnChange**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Send on change With the setting TRUE, the block runs a change check within the acquired data area "DataInput". The block checks whether at least one bit has changed. If a change is detected, a transfer of the data area is started automatically. You specify whether the entire area is transferred or only the changed part with the "SendAll" parameter. If the setting is FALSE, you need to trigger the transfer via the input parameter "TriggerInput".

Parameter:	**TriggerInput**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Trigger input	

With the edge change 0 → 1 of the "TriggerInput" input, the transfer of the data frame can be triggered at a required time.

Example:
Time-driven analog value transfer with time stamp for supplying an analog value archive in the control center.

Make sure that you set the "ImageMemory" parameter to FALSE to prevent this data with time stamps from being overwritten when saving on the station TIM.

The FC Trigger block can be used for time-driven triggering of a transmission over "TriggerInput".

"TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. With dedicated lines or wireless networks the transfer is immediate. With a dial-up connection, it is possible that the data frame is saved first on the TIM and sent at a later point in time. The reason can, for example, be that the data frame is marked as "Conditional spontaneous", see parameter "Conditional".

If you do not require the parameter, simply leave it open. You should, however, then set the "SendOnChange" parameter to TRUE so that the data is transmitted automatically at every change.

For the triggering you can also select a combination of "SendOnChange" plus "TriggerInput". This means that a transfer is triggered both when a change is detected and at every edge change from 0 to 1 at the "TriggerInput" input.

If you use neither "SendOnChange" nor "TriggerInput" to trigger data transfer, the data will only be transferred when there is a single request for this data object or within the framework of a general request.

Parameter:	**SendAll**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE

Default: TRUE

Explanation: Send all data with every transfer

With the parameter, you specify whether the block will always transfer all data of the area specified with "DataInput" or only changed data. The transfer can be triggered by the activated change check (SendOnChange = TRUE) or by "TriggerInput".

- SendAll = TRUE

  Always send all data
- SendAll = FALSE

  Send only changed data

  Exception:
  If "SendAll" is set to = FALSE, the transfer is triggered by "TriggerInput" and if no data has changed at this time, the complete area will be transferred. For this exception this corresponds to "SendAll" = TRUE.

  When only the changed data area is transferred ("SendAll" = FALSE), this area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

If there is a single request for this data object or within the framework of a general request, all data words of the area specified by "DataInput" are always transferred.

Parameter: **DataInput**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

Address range: P#DBxx.DBX yy.0 DWORD zz

- xx: Data block number 1...32767
- yy: Byte number
- zz: Number of double words 1...12 starting at byte number yy

Example:
`P#DB20.DBX 100.0 DWORD 4`

Remember the periods and spaces when entering the pointer!

Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation: Data input area

The ANY pointer addresses the data area in which the data to be acquired is located. This data area must be within a data block and its length can vary between 1 and 12 data double words.

For information on the content and formats, refer to the section "Function" above.

If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

5.7.1.15 Data typical Dat12D_R

Function

Receive maximum of 12 double words with any data content.

The content of each double word may be a value in double word (DWORD, DINT, REAL) format, it can also be a mixture of other data types which together form a double word, for example:

- 4 bytes
- 2 words
- 2 bytes + 1 word

Dat12D_R stores the received data without further processing in the data area specified by "DataOutput". You need to evaluate and process the received data with the user program.

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameter

Parameter: **PartnerNo**

PartnerObjectNo

Enabled

For a description, see section Reoccurring parameters (Page 144).

Parameter: **DataOutput**

Declaration: INPUT

Data type: ANY

Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)
Address range:	P#DBxx.DBX yy.0 DWORD zz • xx: Data block number 1...32767 • yy: Byte number • zz: Number of double words 1...12 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 4` Remember the periods and spaces when entering the pointer! Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.
Explanation:	Data output area The ANY pointer addresses the data area in which the received data is saved. This data area must be within a data block and its length can vary between 1 and 12 double words. For information on the content and formats, refer to the section "Function" above. Dat12D_R stores the received data without further processing in the data area specified by "DataOutput". You need to evaluate and process the received data with the user program. When only changed data is sent by the partner object Dat12D_S, it is possible that only part of the data output area is newly written. this is the area in which the changes were detected at the acquisition end If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Parameter:	**NewData**
Explanation:	Receive new data For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144). Whenever the block has received new datar values from the partner object and has output them to the output field "DataOutput", the "NewData" output is set to TRUE for one OB1 cycle.

5.7.1.16 Data typical Dat256D_S

Function

Send a maximum of 256 double words with any data content

The content of each double word can be a value in double word format (DINT, REAL etc.). A combination of other formats is permitted that together result in a double word again, for example

- 32 Bool
- 4 bytes
- 2 words
- Any combination such as 2 bytes plus 1 word etc.

Note

Remember double word boundaries

When changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 256 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Sending the data area can be triggered in two ways:

- By a change check

  The data is transferred as soon as a bit changes ("SendOnChange" = TRUE).
- By the user program

  The transfer can be triggered by an edge change 0 → 1 at the "TriggerInput" input

  For time-driven transfer FC Trigger can be used.

With "SendAll" you can also specify whether the transfer always includes all data or only the data double words that have changed.

With S7-300 CPUs with X communication, the maximum length of a data frame is 76 bytes. 1024 bytes of net data are transferred using a serial transfer process consisting of a sequence of at least 22 data frames (segments). Each data frame apart from the last contains a segment of 48 bytes of net data of the input data area.

To ensure data consistency when the "SendAll" parameter is activated or during a general or single request, the data is transferred in consecutive segments. During the transfer process, the status is indicated by "SendAllBusy". On the recipient, the status is indicated at the "DataStatus" output.

Note

TriggerInput - SendAllBusy

If "TriggerInput" is triggered when "SendAllBusy" = TRUE, this leads to the "DataLoss" error message (status in the frame header) if the transfer is triggered again.

Only when "SendAllBusy" = FALSE is set is the edge change 0 → 1 triggered at "TriggerInput".

If the transfer is interrupted, "SendAllError" is indicated. An entry is also made in the diagnostics buffer with the event ID B14DTD7_Diagnostics.

If the transfer is incomplete, the data status at the recipient is also "invalid". This is indicated on the recipient in the DataStatus parameter. Apart from this an entry with the event ID B13BTD7_Diagnostics is written to the diagnostics buffer.

Note

Availability of the partner

If the status of the partner changes from "available" to "unavailable", the transfer of all data is stopped immediately. All object data is deleted from the TIM buffer. This can lead to loss of data.

As soon as the partner is available again, the automatic general request ensures that the data of the partner is up-to-date again for the next transfer.

Note

Dat256D_S and Dat256D_R require the UDT "TransmitBlock".

When using the typical, copy the UDT from the global library into the "PLC data types" directory of the CPU. The UDT is automatically referenced by the typical from the block dictionary of the CPU, but not from the global library.

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameter

Parameter:	**PartnerNo**
	PartnerObjectNo
	Enabled
	Unconditional
	TimeStamp
	For a description, see section Reoccurring parameters (Page 144).

Parameter:	**SendOnChange**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE

Explanation:	Send on change With the setting TRUE, the block runs a change check within the acquired data area "DataInput". The block checks whether at least one bit has changed. If a change is detected, a transfer of the data area is started automatically. You specify whether the entire area is transferred or only the changed part with the "SendAll" parameter. If the setting is FALSE, you need to trigger the transfer via the input parameter "TriggerInput".

Parameter:	**TriggerInput**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Trigger input With the edge change 0 → 1 of the "TriggerInput" input, the transfer of the data frame can be triggered at a required time. Example: Time-driven analog value transfer with time stamp for supplying an analog value archive in the control center. The FC Trigger block can be used for time-driven triggering of a transmission over "TriggerInput". "TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. With dedicated lines or wireless networks the transfer is immediate. With a dial-up connection, it is possible that the data frame is saved first on the TIM and sent at a later point in time. The reason can, for example, be that the data frame is marked as "Conditional spontaneous", see parameter "Conditional".	

Select suitable trigger points so that the data on the TIM is not overwritten by buffer overflow (intervals too long).

If you do not require the parameter, simply leave it open. You should, however, then set the "SendOnChange" parameter to TRUE so that the data is transmitted automatically at every change.

For the triggering you can also select a combination of "SendOnChange" plus "TriggerInput". This means that a transfer is triggered both when a change is detected and at every edge change from 0 to 1 at the "TriggerInput" input.

If you use neither "SendOnChange" nor "TriggerInput" to trigger data transfer, the data will only be transferred when there is a single request for this data object or within the framework of a general request.

Do not transfer any analog values for which the "SendOnChange" parameter = TRUE is set without first preprocessing the process data. You will find more detailed information on this with the analog value typical Ana04W_S, parameter "ThresholdValue".

Parameter:	**SendAll**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	TRUE
Explanation:	Send all data with every transfer

With the parameter, you specify whether the block will always transfer all data of the area specified with "DataInput" or only changed data. The transfer can be triggered by the activated change check (SendOnChange = TRUE) or by "TriggerInput".

- SendAll = TRUE

  Always send all data
- SendAll = FALSE

  Send only changed data

  Exception:
  If "SendAll" is set to = FALSE, the transfer is triggered by "TriggerInput" and if no data has changed at this time, the complete area will be transferred. For this exception this corresponds to "SendAll" = TRUE.

If there is a single request for this data object or within the framework of a general request, all data words of the area specified by "DataInput" are always transferred.

Parameter:	**DataInput**
Declaration:	INPUT
Data type:	ANY

Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)
Address range:	P#DBxx.DBX yy.0 DWORD zz • xx: Data block number 1...32767 • yy: Byte number • zz: Number of double words 1...256 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 200` Remember the periods and spaces when entering the pointer! Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.
Explanation:	Data input area The ANY pointer addresses the data area in which the data to be acquired is located. This data area must be within a data block and its length can vary between 1 and 256 data double words. For information on the possible double word formats, refer to the section "Function" above. If the parameter assignment is incorrect (null pointer, length > 256, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected. **Data consistency;** If a data segment to be transferred consists of a maximum of 48 bytes, data consistency during the transfer is assured.

Parameter:	**SendAllBusy**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Block being processed with "SendAll" = TRUE This output indicates that the block is currently transferring the data specified by "DataInput". The procedure is activated either by a remote single or general request or by a local internal or external trigger. If "SendAll" is set to TRUE, the transfer of all data is triggered either by internal change control (SendOnChange = TRUE) or by the external "TriggerInput" (edge change 0 → 1). The edge change 0 → 1 has no effect whatsoever with an external "TriggerInput" as long as "SendAllBusy" indicates TRUE. The edge change 0 → 1 of "TriggerInput" only takes effect when "SendAllBusy" = FALSE.

Parameter:	**SendAllError**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Error when processing "SendAll" "SendAllError" is set to TRUE when the connection is interrupted during processing of "SendAll". Apart from this an entry is written to the diagnostics buffer. "SendAllError" remains set to TRUE until it is reset by the user program or by the next CPU restart.

5.7.1.17 Data typical Dat256D_R

Function

Receive maximum of 256 double words with any data content.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 32 Bool
- 4 bytes
- 2 words
- Any combination such as 2 bytes plus 1 word etc.

Note

Remember double word boundaries

When changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 256 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Dat256D_R stores the received data without further processing in the data area defined by "DataOutput". Evaluate the received data with the user program.

With S7-300 CPUs with X communication, the maximum length of a data frame is 76 bytes (net 48 bytes). 1024 bytes of net data are transferred using a serial transfer process consisting of a sequence of at least 22 data frames (segments). Each data frame apart from the last contains a segment of 48 bytes of net data of the output data area.

Each time a detected data segment is received, this is indicated by a corresponding status (bit 1 to 22) of the "NewData" output parameter.

If a change was detected in the data segment, the status bit 0 is also set to TRUE in "NewData" for one CPU cycle. This makes it possible to recognize which segment of the output data area has changed.

Note

When receiving a sequence of several data segments (data frames), the status bits 1 to 22 in the "NewData" parameter are set to TRUE one after the other and remain set to TRUE until the last segment has been received.

If a data segment (data frame) is not part of a received sequence (SendAll = FALSE), the status remains set to TRUE for only one CPU cycle.

To ensure data consistency when "SendAll"= TRUE or during a general or single request, the data area is updated in consecutive individual segments.

During receipt, the status is indicated by the "DataStatus" output byte ("SequenceState" status). If the receive sequence was completed successfully, the data output area is up-to-date and the output data is consistent. This is indicated by "DataStatus" ("DataValid" = TRUE status).

Note

The consistency of the data segments or the limit segments cannot be guaranteed if the "SendAll" parameter was set to FALSE on the sender.

Receipt of a sequence can be disrupted by the following causes:

- The receive sequence was interrupted when communication to the partner fails during an active sequence (event ID B13BTD7_Diagnostics).
- The monitoring time was exceeded. Not all segments could be received within the time set for the "MonitoringTime" parameter (event ID B13CTD7_Diagnostics).
- Other receiving errors occur (event ID B13DTD7_Diagnostics), for example:
  - A new receive sequence is registered during an active, error-free sequence.
  - A spontaneous segment (data frame) is received during an active sequence.

Note

Dat256D_S and Dat256D_R require the UDT "TransmitBlock".

When using the typical, copy the UDT from the global library into the "PLC data types" directory of the CPU. The UDT is automatically referenced by the typical from the block dictionary of the CPU, but not from the global library.

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameter

Parameter:	**PartnerNo** **PartnerObjectNo** **Enabled** For a description, see section Reoccurring parameters (Page 144).	

Parameter:	**SingleRequest**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	TRUE	
Address range:	Input	I 0.0 … I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: A single request is sent to the partner object.

If the partner is available, you can send a single request to the partner object. If a reply is returned, the information is forwarded to the data area specified in "DataOutput".

In terms of transfer sequences, there are priorities:

- Lowest priority: TriggerInput

  An active transfer triggered, for example, by "TriggerInput" on the sender can be interrupted by a single request or a general request.
- Medium priority: Single request

  An active transfer triggered, for example, by "TriggerInput" on the sender can be interrupted by a single request or a general request.

  The interrupted or restarted request leads to a restart of the active sequences without an error message.
- Highest priority: General request

  A general request can interrupt itself or a single request.

  The interrupted or restarted request leads to a restart of the active sequences without an error message.

The request interrupted or restarted by a single or general request leads to a restart of the active sequences without an error message. If a sequence was completed successfully, the "DataValid" status of the "DataStatus" output byte remains set to TRUE.

The time taken for the response to the single request is evaluated by the "MonitoringTime" parameter.

Note:
Consistency of the data output area across segments is only assured if the receive sequence was completed successfully.

Parameter:	**MonitoringTime**
Declaration:	INPUT
Data type:	INT
Range of values:	0 (no limit) / 1 … 32000 (seconds)
Default:	0
Explanation:	Maximum time for a complete response to a single request Each time a single request starts (see SingleRequest parameter), the time specified here is started in "SingleRequest". If a value higher than 0 is entered and the time for the response sequence is exceeded, an error is indicated via the "DataStatus" output byte (status bits "SequenceState"). Apart from this an entry is written to the diagnostics buffer (event ID B13CTD7_Diagnostics). Each time a single request starts, "MonitoringTime" is reactivated.

Parameter:	**DataOutput**
Declaration:	INTPUT
Data type:	ANY
Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)
Address range:	P#DBxx.DBX yy.0 DWORD zz • xx: Data block number 1...32767 • yy: Byte number • zz: Number of double words 1...256 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 200` Remember the periods and spaces when entering the pointer! Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation:	Data output area	

The ANY pointer addresses the data area in which the received data is saved. This data area must be within a data block and its length can vary between 1 and 256 data double words. For information on the possible double word formats, refer to the section "Function" above.

Dat256D_R stores the received data without further processing in the data area specified by "DataOutput". You need to evaluate and process the received data with the user program.

When only changed data is sent by the partner object Dat256D_S, it is possible that only part of the data output area is newly written. this is the area in which the changes were detected at the acquisition end

If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Parameter:	**NewData**	
Declaration:	OUTPUT	
Data type:	DWORD	
Range of values:	See address range	
Default:	0 (DW#16#0)	
Address range:	Output (DWORD)	QD0 … QDn PQD0 … PQDn
	Bit memory (DWORD)	MD0 … MDn LD0 … LDn
	Data (DWORD)	DBm.DBB0 … n
Explanation:	Receive new data	

Whenever the block receives new data segments, the display of status bits 1 to 22 according to the received data segments 1 to 22 appears in "NewData".

If at least one data segment of the received data contains changes, bit 0 of "NewData" is set to TRUE for one OB1 cycle.

When receiving a sequence of several data segments (data frames), the status bits 1 to 22 in the "NewData" parameter are set to TRUE one after the other and remain set to TRUE until the last segment has been received.

If a data segment (data frame) is not part of a received sequence ("SendAll" = FALSE), the status remains set to TRUE for only one CPU cycle.

The output is intended for user-specific further processing, for example to react in a specific way to receipt of new data.

If you do not require the parameter, simply leave it open.

Parameter:	**DataStatus**
Declaration:	OUTPUT
Data type:	BYTE

Range of values:	See address range	
Default:	0 (B#16#0)	
Address range:	Output	QB0 … QBn PQB0 … PQBn
	Memory bit	MB0 … MBn LB0 … LBn
	Data	DBm.DBB0 … n

Explanation: Currentness status of a received data segment

During receipt of a sequence, the current status is indicated by the "DataStatus" output byte:

- If the receipt of the sequence was completed successfully, the "DataOutput" data output area is up-to-date. The status bit "DataValid" is set to TRUE.

  If "SendAll" is set to TRUE on the sender, the data is consistent.
- If the receipt of a sequence is disrupted, the "DataOutput" data output area is not up-to-date and "DataStatus" indicates an error.
  - The status bit "DataValid" is set to FALSE and an entry is written to the diagnostics buffer.
  - The "SequenceState" status shows the error (see table).

Table 5- 7 Bit assignment of "DataStatus"

Bit	Name	Value	Meaning
0	**DataValid**	FALSE	Data invalid
		TRUE	Data valid
1 ... 5	Reserved	- (FALSE)	Not used
7, 6	**SequenceState**	0	No data being received or receipt completed without error.
		1	First segment of a sequence received
		2	Second or higher segment of a sequence received
		3	Errors: • Transmission sequence aborted • Monitoring time exceeded • Other error receiving

5.7.1.18 Parameter typical Par12D_S

Function

Send 1 to 12 parameter values (each 1 double word) and receive the current, locally valid parameter values from the partner.

The 1-out-of-n check is performed by FC Safe.

Note

FC Safe required

With Par12D_S, data can only be transmitted when FC Safe is linked in the end of cyclic program, see Section FC Safe (Page 214).

The content of each double word may be a value in double word (DWORD, DINT, REAL) format, it can also be a mixture of other data types which together form a double word, for example:

- 4 bytes
- 2 words
- 2 bytes + 1 word

See also the note "Note word boundaries" below.

The data area to be transferred is specified for the "ParameterInput" parameter in the form of an Any pointer. This data area must be within a data block and its length can vary between 1 and 12 double words. The data area sent to the partner or the parameter values entered locally at the partner are returned from there and output here at the "ReturnedParameter" parameter. This output area (Any pointer) must also be within a data block and its length must match that specified for "ParameterInput".

Separate data areas are normally specified for "ParameterInput" and "ReturnedParameter". This makes it easy to recognize what was most recently entered and what is locally valid. However, it is also possible to specify the same data area for both parameters. The two areas then overlap 100% and therefore always match. In this case, you can no longer distinguish the difference between what has been entered most recently and what is locally valid. When mirrored back values are not needed, there is no need to configure "ReturnedParameter".

Even when separate areas are specified for "ParameterInput" and "ReturnedParameter", it is still possible to ensure that the "ParameterInput" input area is always synchronized with the mirrored back values of "ReturnedParameter". This can be done manually from case to case with the "ApplyRemoteParamMan" input or automatically by setting the "ApplyRemoteParamAuto" parameter to TRUE.

A parameter can also be set locally at the partner object that receives the parameter. The partner object must then be set to ’local’ at the "Local" input parameter (see block Par12D_R). The current status of the "Local" input parameter is reported by the partner object and indicated here at the "LocalOperation" output. As long as the partner object is set to 'local', no parameters are accepted there from other nodes.

The sending of the data area defined by "ParameterInput" can be triggered via the following parameters:

- EnterInput

  You should use this input parameter when the data area defined at "ParameterInput" is entered over hardware (digital and analog input modules). "EnterInput" must then be connected to a button on a console or panel via a digital input. The transmission of the entered values is then triggered by pressing this button.

  The entire data area specified by "ParameterInput" is transferred.

- ContinuousEnterFunct

  Set the parameter to TRUE if you enter the parameters using software, for example via an operator panel (OP). There is a constant check for changes. When a change is detected in the data area defined with "ParameterInput", the data double words that have changed since the last transfer are transferred.

  Only the changed data is transferred (see note "Changed data areas" below).

- Release

  Use this input parameter if you enter the parameters using software, for example via an OP. The "Release" input should then be set using a function key on the OP. Changes are checked when a 1 signal is detected at the "Release" input. The data double words from the data area specified with "ParameterInput" that have changed since the last transfer are transferred.

  Only the changed data is transferred (see note "Changed data areas" below).

- RetransmitAll

  Use this input parameter if you enter the parameters using software, for example via an OP. The "RetransmitAll" input should then be set using a function key on the OP. When a 1 signal is detected at the "RetransmitAll" input, the data area configured in "ParameterInput" is transferred without checking for changes.

  The entire data area specified by "ParameterInput" is transferred.

Note

Changed data areas

When only the changed data area is transferred, this area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read in is 10 double words long. In this case, changes were detected in the second, fifth and eighth double words. The transferred area is therefore from the 2nd to the 8th double word.

Remember word boundaries

When only changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 12 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameter

Parameter:	**PartnerNo** **PartnerObjectNo** **Enabled** For a description, see section Reoccurring parameters (Page 144).

Parameter:	**EnterInput**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Enter input

The transfer of the parameter values at the "ParameterInput" input can be triggered vis this input by a signal edge change.

A signal change at "EnterInput" is only taken into account when "ContinuousEnterFunct" = FALSE. If this condition is fulfilled, an edge change 0 → 1 causes the parameter values entered in "ParameterInput" to be adopted and transferred. There is no change check. The entire data area specified in "ParameterInput" is always transferred.

This method of triggering transfer is suitable for input vis hardware, for example via a console or control panel. For further information and related parameters, refer to the section "Function" above.

If you do not require the parameter, simply leave it open.

Data checks:

- The parameters that are read in are then transferred f no error is detected during the 1-out-of-n check, and if the central enable memory bit is set. This is automatically set by FC Safe following a selected time delay set there (see FC Safe, "InputDelayTime" parameter). The input area is then only read in again by the FB when a 0 signal was detected at EnterInput for at least one OB1 cycle.
- When a 1-out-of-n error is detected at the hardware input, the entered parameters are no longer processed. New parameters are read in again only when previously for the length of an OB1 cycle, no hardware input via a command, setpoint or parameter block was detected.

  The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172). As long as the error remains, the error status is indicated via the "InputError" output of FC Safe (see FC Safe, "InputError" parameter).

Parameter:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Continuous change check

With this parameter, you can decide whether the parameter values at the "ParameterInput" input should be continuously read in and checked for changes. The change check is made by comparison with the last values that were transferred. Only changed values are sent. If more than one change is detected, the block sends the data area in which all changed parameter values are located.

The changed data area that is transferred consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

A new transfer of the parameter values can be triggered via the "RetransmitAll" input (see below) even when the parameter entries have not changed.

This method of transmission triggering is suitable when the parameter values are entered in the ParameterInput area by software, but can also be used for entering the parameters from an operator panel (OP).

For further information refer to the section "Function" above.

If you do not require the parameter, simply leave it open.

Data checks:

The parameters read in are only transmitted if no error is detected during the 1-out-of-n check.

- While for hardware input (see EnterInput) an empty cycle must be detected before new parameter values can be transferred from the block.
- For software input new parameter values can be transferred in every OB1 cycle. This assumes that there is no other software entry by another block pending in this cycle. Otherwise a 1-out-of-n error is detected.

  When a 1-out-of-n error is detected during the software input, the entered parameters are no longer processed. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172).

  New parameters are read in again only when previously for the length of an OB1 cycle in the CPU, no software input via a command, setpoint or parameter block was detected.

Parameter:	**ApplyRemoteParamAuto**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE

Default:	FALSE
Explanation:	Automatic synchronization of the input area with the returned area. With the parameter all the parameter values from the "ReturnedParameter" mirror back area are then copied to the "ParameterInput" area. In addition to this the mirrored back parameter values are written the send mailbox of the communications DB. Automatic synchronization is then always performed when new data is received from the partner object (Par12D_R). If you do not require the parameter, simply leave it open.

Parameter:	**ParameterInput**
Declaration:	INPUT
Data type:	ANY
Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)
Address range:	P#DBxx.DBX yy.0 DWORD zz • xx: Data block number 1...32767 • yy: Byte number • zz: Number of double words 1...12 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 4` Remember the periods and spaces when entering the pointer! Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.
Explanation:	Parameter input area. The ANY pointer addresses the data area in which the parameter values to be acquired are located. This data area must be within a data block and its length can vary between 1 and 12 data double words. For information on the content and formats, refer to the section "Function" above. If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected. How the parameters at ParameterInput are processed depends on whether they are hardware or software entries and how the transfer of this data area is triggered. You will find more information on this in the section "Function" above.

Parameter:	**ReturnedParameter**
Declaration:	OUTPUT
Data type:	ANY
Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)
Address range:	P#DBxx.DBX yy.0 DWORD zz • xx: Data block number 1...32767 • yy: Byte number • zz: Number of double words 1...12 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 4` Remember the periods and spaces when entering the pointer! Note that the preset value (null pointer) is not permitted. A pointer with a real address must be specified.
Explanation:	Parameter output area The partner object receiving the parameter values reports back the valid parameter values there. These values are displayed at the "ReturnedParameter" output. If the partner object is set to 'local' and a new input is made there, the parameters changed locally are indicated here by "ReturnedParameter". The ANY pointer defines the data area in which the received parameter values are output. This data area must be within a data block and its length can vary between 1 and 12 data double words. The length must be identical with the length set for ParameterInput. After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the current, local, valid parameters are indicated at "ReturnedParameter". If you do not require the parameter, simply leave it open. If the parameter setting is incorrect (data area not a data block, length greater than 12 or length different from the length set for ParameterInput), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The FB is then no longer processed, however, until the error has been corrected.

Parameter:	**LocalOperation**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Output	Q 0.0 ... Q n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation:	Return message from partner object: Object is set to local operation.
	At the partner object that receives the parameters a local setpoint entry can also be made.. The partner object Par12D_R must then be set to 'local' at the "Local" input parameter. The current status of the "Local" input parameter is reported by the partner object and indicated here at the "LocalOperation" output.
	After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the local currently valid status is displayed at "LocalOperation".

Parameter:	**NewData**
Explanation:	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).
	Whenever the block has received new data and has output it to the outputs "ReturnedSetpoint" or "LocalOperation", the "NewData" output is set to TRUE for one OB1 cycle.

Parameter:	**Release**	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
	This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.	

Explanation:	Trigger input for sending the currently pending parameter values You can use this input parameter when the parameter is entered by software, for example at an operator panel (OP). "Release" should then be set using a function key on the OP. You can then enter several parameters initially on the OP. The parameters are transferred only when the Release function key is activated. A change check is performed only with signal 1 at the Release input. The data double words from the data area configured with "ParameterInput" that have changed since the last transfer are transferred. If you always want to transfer the entire data area specified with "Parameter-Input" and not only the changed parameter values, you should use the "Re-transmitAll" input parameter instead of "Release". The "Release" input is reset automatically. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input. Data checks: The same safety checks are carried out as with ContinuousEn-terFunct, see above. If you do not require the parameter, simply leave it open.

Parameter:	**RetransmitAll**	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	TRUE	
Address range:	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
	This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.	

Explanation:	Trigger input for retransferring the entire data area specified by "ParameterInput". You can use this input parameter when the parameter is entered by software, for example at an operator panel (OP). "RetransmitAll" should then be set using a function key on the OP. When a 1 signal is detected at the "RetransmitAll" input, the entire data area specified by "ParameterInput" is transferred. A change check is not performed The "RetransmitAll" input is reset automatically. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input. Since there is no change check, this would lead to continuous transfer of all parameter values as long as the input has a 1 signal. "RetransmitAll" can also be used as an option in addition to "Release" or "ContinuousEnterFunct" when new parameter values were entered but could not be transferred to the partner (disrupted connection or the partner object is set to 'local'). In this case you can then trigger repeated transfer of the entire data area specified by "ParameterInput" using "RetransmitAll". All changes that were previously entered but are not yet available at the partner are consistently included. "RetransmitAll" can also be used as an independent transfer trigger when you always want to send all entries and not just those that have changed. In this case you can use "RetransmitAll" instead of "Release" that only sends the changed parameter values. Data checks: The same safety checks are carried out as with "ContinuousEnterFunct", see above. If you do not require the parameter, simply leave it open.

Parameter:	**ApplyRemoteParamMan**
Declaration:	IN_OUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Address range:	This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.
Explanation:	Trigger input for synchronization of the input area with the returned area. The input triggers a one-time synchronization of the ParameterInput input area with the "ReturnedParameter" mirror back area. All the parameter values from the "ReturnedParameter" mirror back area are then copied to the "ParameterInput" input area. The send mailbox of the communications DB is also synchronized with the returned parameter values. "ApplyRemoteParamMan" is reset automatically. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input. The result would be a constant synchronization as long as the parameter has a 1 signal. If you do not require the parameter, simply leave it open.

5.7.1.19 Parameter typical Par12D_S

Function

Receive 1 to 12 parameter values (each 1 double word) or enter locally and send back the current, locally valid parameter values to the partner.

The content of each double word may be a value in double word (DWORD, DINT, REAL) format, it can also be a mixture of other data types which together form a double word, for example:

- 4 bytes
- 2 words
- 2 bytes + 1 word

See also the note "Note word boundaries" below.

The data area to be transferred is specified for the "ParameterOutput" parameter in the form of an Any pointer. This data area must be within a data block and its length can vary between 1 and 12 double words.

You can also use the block to enter the parameter values locally. The input area for this is specified as an Any pointer with the "LocalParameterInput" parameter. It must be located within a data block and its length must be identical to the length configured in the "ParameterOutput" parameter.

The block only processes the changed data area. In response to a general or single request, on the other hand, the entire parameter set is transferred or mirrored back..

Bumpless switchover between the "Local" and "Remote" operating modes is guaranteed.

Note

Changed data areas

The changed data area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read in is 10 double words long. In this case, changes were detected in the second, fifth and eighth double words. The transferred area is therefore from the 2nd to the 8th double word.

Remember word boundaries

When only changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 12 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameter

Parameter:	**PartnerNo** **PartnerObjectNo** **Enabled** **ImageMemory** **Conditional** **Unconditional** For a description, see section Reoccurring parameters (Page 144).	

Parameter:	**Local**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Local parameter input released

Release of the local parameter input of the data area specified by "LocalParameterInput".

A setpoint sent by the partner (master station) is not accepted by the object as long as "Local" = TRUE.

The current status of the "Local" input is transferred to the partner.

Bumpless switchover:

- When there is a switchover from "Local" = 0 to 1, the last values output at the "ParameterOutput" output are retained until new parameter values are entered via the local input area "LocalParameterInput".
- When there is a switchback from Local = 1 to 0, the last values at the "ParameterOutput" output are retained until the block receives new parameter values from the partner.

Read the note on the "ContinuousEnterFunct" parameter.

Special situation:

You can also enter the parameter values during local input directly in the output area specified by "ParameterOutput". Either you do not specify an input area for "LocalParameterInput" or you specify the same data area both for "LocalParameterInput" and "ParameterOutput".

This type of parameter input cannot be prevented by the "Local" input. Regardless of the status of the "Local" parameter, the values entered in the output area are sent immediately by the block to the partner.

Local parameter entries can therefore be made regardless of the status of the "Local" input. "Local" only influences the acceptance of parameters sent by the partner:

- Local = 0

  The parameters sent by the partner are accepted and output to the "ParameterOutput" data area.
- Local = 1

  The parameters sent by the partner are rejected.

In this special situation, the "Release" and "ContinuousEnterFunct" have no function.

A status change of the "Local" parameter is always transferred by the module according to the send buffer principle, even when the parameter "ImageMemory" = TRUE. This ensures that the optional synchronization of the input and output area on the partner is always performed correctly (see Par12D_S, parameters "ApplyRemoteParamMan" and "ApplyRemoteParamAuto").

Parameter: **ContinuousEnterFunct**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default:	FALSE
Explanation:	Continuous local parameter acquisition.

With this parameter, you can decide whether the values in the "LocalParameterInput" input area should be continuously read in and checked for changes. The change is checked by comparing the current values at the "ParameterOutput" output.

Changes in the input area are copied immediately to the output area and transferred to the partner object. Only changed values are sent. If there is more than one change, the block sends the data area in which all changed parameter values are located.

The "ContinuousEnterFunct" = TRUE setting only takes effect when the following conditions are met:

- An input area is defined by "LocalParameterInput" and this is not identical to the output area set for "ParameterOutput".

  and
- there is a 1 signal at the "Local" input (= TRUE).

With the setting "ContinuousEnterFunct" = TRUE, when the signal 1 is detected at the "Local" input, the values pending at "LocalparameterInput" are adopted immediately and output at the "ParameterOutput" output. The condition is that the local input values differ from the currently output values at this time.

This method of acquisition of local values is suitable when the values are entered in the "LocalParameterInput" area by software.

If you do not require the parameter, simply leave it open.

Parameter:	**LocalParameterInput**
Declaration:	INPUT
Data type:	ANY
Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)

Address range:	P#DBxx.DBX yy.0 DWORD zz • xx: Data block number 1...32767 • yy: Byte number • zz: Number of double words 1...12 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 4` Remember the periods and spaces when entering the pointer! Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.
Explanation:	Local parameter input area The ANY pointer addresses the data area in which the parameter values to be acquired are located. This data area must be within a data block and its length can vary between 1 and 12 data double words. The length must be identical to the length specified for" ParameterOutput". For information on the content and formats, refer to the section "Function" above. If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected. If you do not require the parameter, simply leave it open.

Parameter:	**ParameterOutput**
Declaration:	INPUT
Data type:	ANY
Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)
Address range:	P#DBxx.DBX yy.0 DWORD zz • xx: Data block number 1...32767 • yy: Byte number • zz: Number of double words 1...12 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 4` Remember the periods and spaces when entering the pointer! Note that the preset value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation:	Parameter output area

The ANY pointer addresses the data area in which the locally entered parameter values or those received from the partner are output. This data area must be within a data block and its length can vary between 1 and 12 double words.

For information on the content and formats, refer to the section "Function" above.

Par12D saves the received data without further processing in the data area specified by "ParameterOutput". You need to evaluate and process the received data with the user program.

When only changed data is sent by the partner object Par12D_S, it is possible that only part of the data output area is newly written. this is the area in which the changes were detected at the acquisition end

If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Parameter:	**NewData**
Explanation:	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).

Whenever the block has received new parameter values from the partner object and has output them to the output field "ParameterOutput", the "NewData" output is set to TRUE for one OB1 cycle. This also applies when there is new local input in the status "Local" = 1.

Parameter:	**Release**	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Explanation: Input for the acceptance of local parameter input.

The acceptance of the parameter values at the "LocalParameterInput" parameter input can be triggered via this input by a signal edge change.

A change from 0 to 1 at the "Release" input is taken into account only when the following conditions are met:

- An input area is specified by the "LocalParameterInput" parameter and this is not identical to the output area specified by "ParameterOutput"

  and
- The "Local" input is set to TRUE.

You can use "Release" for parameter input via software, e.g. via an operator panel (OP). The "Release" input should then be set using a function key on the OP. You can then enter several parameters initially on the OP. The parameter values are read in and checked for changes only when the Release function key is activated.

The change is checked by comparing the current parameter values at the "ParameterOutput" output. Changes in the input area are then copied immediately to the output area and transferred to the partner.

Only changed values are sent. If there is more than one change, the block sends the data area in which all changed parameter values are located.

The "Release" input is reset automatically. Instead of a memory bit or data bit, a digital input can also be specified as the input. The automatic reset would not work with a digital input. This does not, however, have negative effects. Acquisition via "Release" is edge triggered, in other words only once.

If you do not require the parameter, simply leave it open.

5.7.1.20 FC Safe

Function

The FC Safe ensures reliable input of commands and setpoints by using the 1-out-of-n check.

The FC Safe checks th einput for the following data point typicals:

- Cmd01B_S
- Set01W_S
- Par12D_S

If an input is pending, the FC checks whether there is only one entry pending in the current OB1 cycle and then enables the block reading in. As soon as two (or more) entries are pending in an OB1 cycle, both entries are ignored.

In every CPU in which commands and/or setpoints are acquired, the FC Safe needs to be called in OB1 to complete of all command and setpoint FBs.

The FC has separate monitoring for entries via different system memory areas. They are divided into hardware entries and software entries:

- Hardware entries
  – Input modules (I)
  – Peripheral inputs (PI)
- Software entries
  – Bit memory (M)
  – Data blocks (DB)
  – Local data (L)
  – Operator panels

The two types of entry "hardware entries" and "Software entries" are checked separately by the block. The FC enables the hardware and software entries separately. For each type of entry, only one command or setpoint entry may be detected.

If a single hardware entry and a single software entry are pending at the same time both entries are enabled.

For hardware entries, the following additional condition applies: The entry must be pending constantly for the duration of "InputDelayTime". Only when an entered command or setpoint is pending unchanged for this time and during this time no other command or setpoint entry is detected is the block processing enabled.

The actual putting together of the command or setpoint frame is handled by the block that read in the command or setpoint.

For the hardware entry the FC Safe provides the following two display bits:

- InputOK

  Display of the enabling of hardware commands and setpoints
- InputError

  Display of entry errors with hardware commands and setpoints

FC-Safe shows a command output error detected in a station via the following output:

- GlobalCmdOutputError

  Group message: Command output error in a station

Parameter

Name:	**InputDelayTime**
Declaration:	INPUT
Data type:	INT
Range of values:	• 0 Value for unrequired parameter • 1 ... 32000 Range of values for delay time
Explanation:	Delay time in ms for commands and setpoints that are input via hardware. A delay time of at least 1000 ms is recommended.

Name:	**MaxInputTime**	
Declaration:	INPUT	
Data type:	INT	
Range of values:	• 0 Value for unrequired parameter • 1 ... 32000 Range of values for monitoring time	
Explanation:	Monitoring time in ms for commands and setpoints that are input via hardware. A monitoring time of at least 30 s is recommended. If you do not require the parameter, specify 0 (zero).	

Name:	**ResetError**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Input for resetting the output GlobalCmdOutputError. If you do not require the parameter, specify a memory or data bit that always has signal 0.	

Name:	**InputOK**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	Output	Q 0.0 ... Q n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Display of the enabling of hardware commands and setpoints Has a 1 signal as soon as the current entry was enabled, i.e. when the hardware command or setpoint was entered correctly. The display bit goes off when the entry is reset, in other words when the command key is released or the setpoint entry key "EnterInput" input is released. If you do not require the parameter, specify a memory or data bit in the memory area of the local data.	

Name:	**InputError**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	Output	Q 0.0 ... Q n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Display of entry errors with hardware commands and setpoints The output has a 1 signal when one of the following hardware entry errors is detected within the monitoring time "MaxInputTime": • Two or more command and/or setpoint entries were detected at the same time. • At one of the inputs a 1 signal was detected over a longer period of time (for example when the input is defective). If you do not require the parameter, specify a memory or data bit in the memory area of the local data.	

Name:	**GlobalCmdOutputError**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	Output	Q 0.0 ... Q n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Group message: In a station a command output error was detected. A command output error can occur at the receive end only in the following cases: • The content of the two command bytes in the received frame is not identical. • More than one bit is set in the command byte (1-out-of-8 error). If such an error is detected, this is sent by the station with an organizational frame to the subscriber that sent the commands. FC Safe on the sending subscriber then indicates the disruption at the output GlobalCmdOutputError. When an error is detected, the output remains set to the 1 signal until you request the group signal to be reset via the "ResetError" input. If you do not require the parameter, specify a memory or data bit in the memory area of the local data.	

5.7.1.21 Setpoint typical Set01W_S

Function

Send 1 setpoint as a 16 bit value and receive local setpoint from partner

The 1-out-of-n check is performed by FC Safe.

Note

FC Safe required

With Set01W_S, data can only be transferred when the block FC Safe is linked in the end of the cyclic program, see section FC Safe (Page 214).

Parameter

Parameter:	**PartnerNo** **PartnerObjectNo** **Enabled** For a description, see section Reoccurring parameters (Page 144).	

Parameter:	**EnterInput**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Enter input for hardware setpoint The adoption of a setpoint at the "SetpointInput" can be triggered by a signal edge change. A signal change at "EnterInput" is only taken into account when "ContinuousEnterFunct" = FALSE. If this condition is fulfilled, an edge change 0 → 1 causes the setpoint entered in "SetpointInput" to be adopted and transferred. This also applies when the newly entered setpoint is identical to the last setpoint transferred. This method of setpoint adoption is suitable for input vis hardware, for example via a console or control panel. It can also be used for entering setpoints at an operator panel (OP). In this the adoption must be triggered by a function key on the OP. If you do not require the parameter, simply leave it open.	

Parameter:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	TRUE
Explanation:	Apply setpoint continuously for software setpoint With this parameter, you specify whether the setpoint at "SetpointInput" should be continuously read in and checked for changes. Changes are checked by comparison with the last setpoint that was transferred. This method of setpoint adoption is suitable for input via suitable software. It can, however, also be used for entering setpoints at an operator panel (OP) if the OP does not have a separate function key that can be used to trigger the input. If you do not require the parameter, simply leave it open.

Parameter:	**SetpointInput**	
Declaration:	INPUT	
Data type:	WORD	
Range of values:	See address range.	
Default:	0 (W#16#0)	
Address range:	Input words	IW0 ... IWn PIW0 ... PIWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

Explanation: Setpoint input

How a setpoint available at SetpointInput is processed depends on whether it is a hardware or software input. You specify the type of input with the "ContinuousEnterFunct" parameter:

- **ContinuousEnterFunct = FALSE (hardware input)**

  A setpoint at "SetpointInput" is only read in as long as the signal is 1 at "EnterInput". The setpoint that is read in is then transferred if no error is detected during the 1-out-of-n check, and if the central enable memory bit is set. This is automatically set by FC Safe following a selected time delay set there (see FC Safe, "InputDelayTime" parameter).

  A further setpoint is then only read in again by the FB when a 0 signal was detected at EnterInput for at least one OB1 cycle.

  When a 1-out-of-n error is detected at the hardware input, the entered setpoint is no longer processed. A new setpoint is read in again only when previously for the length of an OB1 cycle in the CPU, no hardware input via a command, setpoint or parameter block was detected.

  The FB enters the detected 1-out-of-n error in the diagnostics buffer (event ID B172). As long as the error remains, the error status is indicated via the "InputError" output of FC Safe (see FC Safe, "InputError" parameter).

- **ContinuousEnterFunct = TRUE (software input)**

  A setpoint at SetpointInput is read in continuously and checked for changes. Changes are checked by comparison with the last setpoint that was transferred. The setpoint is sent immediately every time a change occurs unless the 1-out-of-n check detects an error.

  – With hardware input (see EnterInput) an empty cycle must be detected before a new setpoint can be transferred by the block.

  – With software input a new setpoint can be transferred in every OB1 cycle. This assumes that there is no other software entry by another block pending in this cycle. Otherwise a 1-out-of-n error is detected.

    A new transfer of the software setpoint can be triggered via the "SendSoftSetpoint" input even when the software input has not changed (see below).

    When a 1-out-of-n error is detected at the software input, the entered setpoint is no longer processed.

    A new setpoint is read in again only when previously for the length of an OB1 cycle in the CPU, no software input (command or setpoint) was detected. The block enters detected 1-out-of-n errors in the diagnostics buffer (event ID B172). Appropriate error bits are also set in the central data block BasicData where they can be queried by the software. For further details, refer to the description of FC Safe.

Parameter: **ReturnedSetpoint**

Declaration: OUTPUT

Data type: WORD

Range of values:	See address range.	
Default:	0 (W#16#0)	
Address range:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n
Explanation:	Output for a returned setpoint The partner object receiving the setpoint reports back the currently valid setpoint there. This value is displayed at the "ReturnedSetpoint" output. If the partner object is set to 'local' and a new input is made there, the setpoint changed locally is indicated here by "ReturnedSetpoint". After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the locally valid setpoint is indicated by "ReturnedSetpoint". If you do not require the parameter, simply leave it open.	

Parameter:	**LocalOperation**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Output	Q 0.0 ... Q n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Return message from the partner object: The object is set to local operation. A setpoint can also be set locally at the partner object that receives the setpoint. The partner object Set01W_R must then be set to 'local' at the "Local" input parameter. The current status of the "Local" input parameter is reported by the partner object and indicated here at the "LocalOperation" output. After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the local currently valid status is displayed at "LocalOperation". If you do not require the parameter, simply leave it open.	

Parameter:	**NewData**
Explanation:	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144). Whenever the block has received new data and has output it to the outputs "ReturnedSetpoint" or "LocalOperation", the "NewData" output is set to TRUE for one OB1 cycle.

Parameter:	**SendSoftSetpoint**	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
	This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.	
Explanation:	Trigger input for resending the last software setpoint. For further details, refer to the "SetpointInput" parameter. If you do not require the parameter, simply leave it open.	

5.7.1.22 Setpoint typical Set01W_R

Function

Receive 1 setpoint in the station as a 16 bit value or enter locally and return the local setpoint to the master station

Parameter

Parameter:	**PartnerNo** **PartnerObjectNo** **Enabled** **ImageMemory** **Conditional** **Unconditional** **TimeStamp** For a description, see section Reoccurring parameters (Page 144).

Parameter:	**Local**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Release of the local setpoint input of the data area specified by "LocalSetpointInput".

A setpoint sent by the partner (master station) is not accepted by the object as long as "Local" = TRUE.

The current status of the "Local" input is transmitted to the partner together with a copy of the setpoint which is currently being output at "SetpointOutput" (setpoint mirroring).

Bumpless switchover:

- When there is a switchover from "Local" = 0 to 1, the last values output at the "SetpointOutput" output are retained until a new setpoint is entered via the local input area "LocalSetpointInput".
- When there is a switch back from "Local" = 1 to 0, the last value output at the "SetpointOutput" output is retained until the block receives a new setpoint from the partner.

Read the note on the "ContinuousEnterFunct" parameter.

Parameter:	**EnterInput**	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation:	Enter input for local setpoint input. Edge triggered, the parameter triggers adoption of the setpoint at the setpoint input "LocalSetpointInput". A signal change at "EnterInput" is only taken into account when the value TRUE is set at the "Local" input parameter and "ContinuousEnterFunct" = FALSE. If these conditions are fulfilled, a signal change from 0 to 1 at the causes the setpoint entered at "LocalSetpointInput" to be adopted and output at "SetpointOutput". This method of setpoint adoption is suitable for hardware input, for example via a console, control panel or an operator panel (OP). With an OP, the adoption must be triggered by a function key. If you do not require the parameter, simply leave it open.

Parameter:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE
Explanation:	Continuous local setpoint acquisition When "Local" = TRUE and "ContinuousEnterFunct" = TRUE the setpoint at "LocalSetpointInput" is continuously read in and checked for changes. The setpoint read in at "LocalSetpointInput" is output at the "SetpointOutput" output if at this time the local input value differs from the last returned setpoint. If an array is detected at the setpoint input "LocalSetpointInput" this is output immediately at "SetpointOutput" (without a change check). This method of setpoint adoption is suitable for software input. It can, however, also be used for entering setpoints at an operator panel (OP) if the OP does not have a function key. If you do not require the parameter, simply leave it open.

Parameter:	**LocalSetpointInput**	
Declaration:	INPUT	
Data type:	WORD	
Range of values:	See address range	
Default:	0 (W#16#0)	
Address range:	Input words	IW0 ... IWn PIW0 ... PIWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

Explanation: Local setpoint input

A value at "LocalSetpointInput" is only adopted if "Local" = TRUE. If this condition is met, how a pending setpoint is processed depends on whether it is a hardware or software input. The type of input is decided by the "ContinuousEnterFunct" parameter:

- ContinuousEnterFunct = FALSE

  Hardware input

  A setpoint at "LocalSetpointInput" is only read in when a signal change 0 → 1 is detected at "EnterInput". The setpoint entered locally is output via the "SetpointOutput" output and transferred to the partner.

  A further setpoint is then only read in again by the block when a 0 signal was detected at "EnterInput" for at least one OB1 cycle.
- ContinuousEnterFunct = TRUE

  Software input

  A setpoint at "LocalSetpointInput" is read in continuously and checked for changes. The change check is achieved by comparison with the last valid setpoint; in other words, the value stored as the returned setpoint. With each change, the setpoint is output via the output set at "SetpointOutput" and transferred to the partner.

  With software input a new setpoint can be entered in every OB1 cycle.

  With hardware input an empty cycle must be detected before a new setpoint can be read in by the block.

If you do not require the parameter, simply leave it open.

Parameter:	**SetpointOutput**	
Declaration:	OUTPUT	
Data type:	WORD	
Range of values:	See address range	
Default:	0 (W#16#0)	
Address range:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

Explanation: Setpoint output word.

The setpoint sent by the partner object or entered locally at "LocalSetpointInput" is output at the "SetpointOutput" output.

Parameter: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 144).

Whenever the block has received a new setpoint from the partner object and has output it to "SetpointOutput", the "NewData" output is set to TRUE for one OB1 cycle. This also applies when there is a new local setpoint input (Local = 1).

5.7.2 Optional blocks

5.7.2.1 ListGenerator1500/300/400 FC

The ListGenerator FC is available in three versions with the following symbolic names:

- For S7-300

  ListGenerator300
- For S7-400

  ListGenerator400
- For S7-1500

  ListGenerator1500

Function

The FC ListGenerator is required in a CPU that receives data containing either an incomplete destination address or no destination address at all. The lack of the destination object number is the most important factor here because this points to the instance DB in which the received information should be stored.

Missing or incomplete destination addresses can occur when no or incomplete parameters are set for them in the station. This is permitted for typicals that send binary information, analog values or counted values. If these typicals send data to more than one destination, no destination address is set for them. Due to the missing destination information, the send frame is automatically transmitted to all destinations for which a connection was set up during configuration. This data is therefore received without a destination address at the various destinations.

Note

Supplement to destination subscriber no.

Data frames to be sent without a destination address have the destination subscriber number added by the sending TIM, and sometimes several if there are several destinations.

The TIM enters 0 in the address field for the destination object number, since the TIM does not have the relevant information. The only destination subscribers it knows are those to which it has a configured connection.

At the receiving end, the data frame therefore contains the destination subscriber number but the destination object number is 0.

If the destination object number is not contained in the received data frame, FC Distribute, which is responsible for distributing the received frames, references an object reference list.

Using the source address (source subscriber no. + source object no.) contained in every data frame, FC Distribute searches through the list for an entry that specifies the missing destination object number for the given source address; in other words, it searches for the number of the local instance DB.

This object reference list is created by FC ListGenerator. The FC has no parameters. It is linked into the cyclic user program (OB1) following the FC BasicTask.

When creating the list, FC ListGenerator uses the addresses set in the parameters for the receiving typicals. Specification of "PartnerNo" and "PartnerObjectNo" is mandatory for these typicals. These parameters are identical to the source address in the corresponding receive frame. Since the typical also knows the number of its instance DB, it therefore knows all the addresses required for an entry in the reference list.

During startup, FC ListGenerator arranges that all receiving typicals enter the addresses from their parameter assignment with the number of the instance DB in the reference list. The object reference list therefore does not require special parameter settings, it is simply created from the existing parameters of the receive typicals and is therefore always consistent.

How it works

FC ListGenerator creates the lists after startup in three consecutive OB1 cycles:

- In the first cycle, it determines how many entries will be required in the first and, if applicable, in the second object reference list. The typicals involved only increment a counter during this run.
- In the second cycle, FC ListGenerator generates the data block for the first and, if applicable, the second object reference list with the required length and enters 0 in all the data words. During the same cycle, all typicals involved enter their addresses and the number of the corresponding instance DB in the list.
- In the third and final cycle, FC ListGenerator sorts all the entries in ascending order. Sorting speeds up the search in the list during actual operation.

When generating the data block, FC ListGenerator does the following:

If no list has yet been created, a free DB number is searched for. Starting with the number of DB BasicData, the number of the next lowest free DB is taken.

If a list already exists, FC ListGenerator checks to see whether the existing DB is long enough for the currently required number of references. If the length is adequate, 0 is entered as the content and the addresses are written again and sorted.

If the existing data block is too short, different procedures are used for the various SIMATIC product series:

- For S7-300

  A new DB is generated. The old DB remains in memory because S7-300 has no delete function for data blocks.

  With CPU 300, you have to delete the old DBs with the programming device.

  Note:
  Without generating a new DB, you would need to delete this DB with the programming device. If there is not enough memory on the CPU to be able to generate a new DB, you need to delete the existing DB before restarting.

- For S7-400

  The existing DB is deleted, the memory is compressed and the DB is then generated with the same number in the new length.

  With the CPU 400, you may be able to compress the memory manually or reload the CPU.

- For S7-1500

  The existing DB is deleted and generated in the new length with the same number.

  With the CPU 1500, the memory is automatically compressed, as with the CPU 300.

If the ListGenerator FC can no longer generate a DB, an error message is written to the diagnostic buffer of the CPU:

- 0xB107 "Error generating the object reference list"
  - DB[Info1]

    DB[Info1] cannot be created.
  - Cause: [Info2].

    In Info2, the return value of the SFC Create_DB function is output, refer to the description there.

5.7.2.2 FC PartnerMonitor

Function

FC PartnerMonitor has the following functions:

- It displays important status information about a SINAUT subscriber (see "PartnerStatus" parameter).
- The FC can also be used to trigger a general request of the subscriber, except to a control center (e.g. ST7cc/ST7sc).
- It can also be used to establish and terminate a permanent connection to the subscriber.

The FC can be called at any point in the cyclic user program (in OB1).

If you want to monitor and control more than one subscriber, include an appropriate number of FC PartnerMonitor in the user program.

A SINAUT subscriber (partner) can only be an ST7 CPU or an ST7cc to which a connection was configured. TIMs cannot be monitored or controlled by FC PartnerMonitor.

Note

FC PartnerMonitor in a station

FC PartnerMonitor can also be used in a station. However, the control inputs for establishing and terminating a permanent connection can then no longer be used. This only works in the master station when the local TIM is a master station TIM.

Parameter

Name:	**PartnerNo**
Declaration:	INPUT
Data type:	INT
Range of values:	1 ... 32000 [ms]
Explanation	SINAUT subscriber no. of the subscriber to be monitored and controlled. If the set "PartnerNo" is not found in the administration (DB BasicData), then (only during startup) an entry is written to the diagnostics buffer (event ID B101). The CPU does not change to STOP. The status of a subscriber with correct parameter settings is indicated in the "PartnerStatus" output word and the control inputs are processed. An unknown subscriber is not processed until the error is eliminated. The "PartnerStatus" output word remains set to 0 during this time.

Name:	**MaxConnectTime**
Declaration:	INPUT
Data type:	INT
Range of values:	• 0 (no limitation) • 1 … 480 [minutes]
Explanation	Maximum duration of a permanent connection The time specified here (> 0) is activated at the start of a permanent connection (see "PermanentCall_On" parameter). If the time elapses before the permanent connection is reset, it is automatically disconnected. The time is retriggered as long as the signal 1 is present at the "PermanentCall_On" input. The time specified here applies to a permanent connection in a dial-up network as well as to a permanent connection (continuous polling) on a dedicated line.

Name:	**PartnerStatus**	
Declaration:	OUTPUT	
Data type:	WORD	
Range of values:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n
Explanation	Output word to indicate the status of the subscriber to be monitored. If you do not require the parameter, simply leave it open.	

The meaning of the status bits in the "PartnerStatus" output word:

Bit	.0	Status of the subscriber
	0	0 = Subscriber disrupted
	1	1 = Subscriber OK

Bit	.1	Status of the redundant connection
	0	0 = Redundant connection is disrupted
	1	1 = All connections OK.

Bit	.3	.2	Status of the general request (GR)
	0	0	0 = GR complete without error
	0	1	1 = GR started
	1	0	2 = GR start received
	1	1	3 = GR finished with error (GR incomplete or cannot be executed, e.g. due to fault at subscriber)

Bit	.6	.5	.4	Status of the dial-up connection
	0	0	0	0 = No connection
	0	0	1	1 = Outgoing call activated
	0	1	0	2 = Incoming call established
	0	1	1	3 = Outgoing call established
	1	0	0	4 = Permanent connection registered
	1	0	1	5 = Permanent connection established
	1	1	0	6 = Permanent connection disconnected

Bit	.7	Status of the dial-up connection
	0	0 = No dial-up connection check in background
	1	1 = Dial-up connection check in background is activated

Bit	.8	Status of continuous polling (on dedicated line)
	0	0 = No continuous polling
	1	1 = Continuous polling activated

Bit	.9	Status of the WAN connection resources
*)	0	0 = Sufficient resources on partner
	1	1 = Insufficient resources on partner

Bit	.10	Time status
	0	0 = Date/time not available / not OK on partner
	1	1 = Date/time OK on partner

Bit	.11	Time-of-day synchronization
	0	0 = The partner CPU received a plausible time during the last synchronization or no synchronization time has been received since startup.
	1	1 = The partner CPU has received an implausible synchronization time; the last valid time will continue to be used. (Display only with TimeMask > V1.6)

Note

In-out parameters

The following parameters are in/out parameters (declaration IN_OUT):

- GeneralRequest
- PermanentCall_On
- PermanentCall_Off

It is difficult to specify local bit memory with this parameter type and this should not be used.

Name:	**GeneralRequest**	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation	Input for triggering a general request to the subscriber specified with PartnerNo. A general request to the subscriber is triggered with a 1 signal at this input if no request is active for this subscriber at this time. The input is then automatically reset by the FC. If an input of a digital input is specified (I 0.0 ... I n.7), you are responsible for resetting the signal at the input. Reset the signal before ending the currently running general request so that another general request is not triggered immediately.	

Name:	**PermanentCall_On**	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation	Input for triggering a permanent connection to the subscriber specified with PartnerNo.

A permanent connection to the subscriber is triggered with a 1 signal at this input if there is currently no permanent connection to this subscriber. The input is then automatically reset by the FC. If an input of a digital input is specified (I 0.0 ... I n.7), you are responsible for resetting the signal at the input at the latest before the termination of the existing permanent connection.

A 1 signal at the input "PermanentCall_On" also activates the time specified with "MaxConnectTime" if it is greater than 0.

Depending on whether the subscriber can be reached over a dial-up connection or a dedicated line, the command to establish the permanent connection is processed as follows and indicated at the "PartnerStatus" output:

- **For a dial-up connection:**

  A dial-up connection is established by the master TIM to the appropriate subscriber and, regardless of the data traffic, maintained until the terminate command is sent.
  The current status of the permanent connection is indicated in the PartnerStatus output word with the bits 4 ... 6 (see PartnerStatus parameter).
- **For a dedicated line:**

  In this case the master TIM operates in polling mode with the stations. A permanent connection is implemented in this case by ’continuous polling’ of the subscriber. This is actually an intermittent poll to the subscriber; in other words, the other subscribers on the dedicated line network are still polled but the preferred subscriber is polled again after every poll to a ’normal’ subscriber .
  The current status of the continuous polling is indicated by bit 8 in the "PartnerStatus" output parameter.

Special feature with stations:

A permanent connection cannot be established from a station. This control input cannot therefore be used when FC PartnerMonitor is used in a station.

Name:	**PermanentCall_Off**	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	Input	I 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation

Terminating an existing permanent connection

The input serves to trigger termination of an existing permanent connection to the subscriber specified with "PartnerNo".

A permanent connection to the subscriber is terminated with a 1 signal at this input if there is currently a permanent connection to this subscriber. The input is then automatically reset by the FC. If an input of a digital input is specified (I 0.0 ... I n.7), you are responsible for resetting the signal at the input via the user program. This should be done at the latest before establishing a permanent connection again.

Depending on whether the subscriber can be reached over a dial-up connection or a dedicated line, the command to terminate the permanent connection is processed as follows and indicated at the "PartnerStatus" output:

- **For a dial-up connection:**

  The existing dial-up connection is terminated by the master TIM but only after any pending data has been sent.
  The current status of the permanent connection is indicated in the PartnerStatus output word with the bits 4 ... 6 (see PartnerStatus parameter).
- **For a dedicated line:**

  The master TIM deletes the registration for continuous polling of the corresponding subscriber. The polling cycle for all connected subscribers continues in normal mode.
  The current status of the continuous polling is indicated by bit 8 in the "PartnerStatus" output parameter.

  Continuous polling can also be canceled on a dedicated line by instructing the master TIM to start continuous polling of another subscriber. The existing job is then replaced by the new one.

Special feature with stations:

A permanent connection cannot be terminated by a station. This control input cannot therefore be used when FC PartnerMonitor is used in a station.

5.7.2.3 FC PartnerStatus

Function

The FC PartnerStatus can show the current status ’disrupted’ or ’OK’ for a maximum of 8 SINAUT subscribers.

The FC can be called at any point in the cyclic user program (in OB1).

If you want to monitor more than 8 subscribers, include the appropriate number of FCs PartnerStatus in the user program.

The partner can be an ST7 CPU or an ST7cc to which a connection was configured, or a local TIM.

One bit per subscriber is reserved in the "PartnerStatus" output byte to indicate the status of the respective subscriber:

- FALSE (0):
  - Subscriber disrupted
  - Corresponding input parameter not used (= 0)
  - Subscriber unknown
- TRUE (1): Subscriber OK

Parameter

Name:	**Partner1 ... Partner8**
Declaration:	INPUT
Data type:	INT
Range of values:	• 0 Value for unrequired parameter • 1 ... 32000 Number of the subscriber to be monitored
Explanation	SINAUT subscriber number of the subscriber to be monitored If a set subscriber number is not found in the administration (DB BasicData), then (only during startup) an entry is written to the diagnostics buffer (event ID B101). The CPU does not change to STOP. The status of a subscriber with a correct parameter assignment is indicated in the "PartnerStatus" output byte. An unknown subscriber is not processed until the error is eliminated. Their status bits are set to 0.

Name:	**PartnerStatus**	
Declaration:	OUTPUT	
Data type:	BYTE	
Range of values:	Output bytes	QB0 ... QBn PQB0 ... PQBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n
Explanation:	Indication of the status of the subscriber to be monitored. Assignment of the status bits in the "PartnerStatus" output byte depending the parameters Partner1 ... "Partner8":	

Bit	.7	.6	.5	.4	.3	.2	.1	.0
Partner	8	7	6	5	4	3	2	1

Value:

- 0 = partner disrupted, not set in the parameters or unknown
- 1 = partner OK

5.7.2.4 FC PathStatus

Function

The block (FC) shows the status of the path to a partner from the perspective of the local TIM.

A maximum of 2 paths (main and substitute path) to a partner can be configured. Both paths must begin or end on a local TIM.

The block shows the following:

- The paths via which the partner can be reached.
- The path currently being used
- The TIM interface via which the main path was configured.
- The TIM interface via which the substitute path was configured.

The path of a connection is specified as a combination of the used interfaces of the TIM and the status of the path.

In the output byte PathStatus the following bits are reserved:

- Two bits for the interface of the main path
- Two bits for the interface of the substitute path
- Two bits for the status of the main path
- Two bits for the status of the substitute path

The FC can be called at any point in the cyclic user program (OB1) after calling the FC BasicTask.

If the status of the path to more than one subscriber is to be shown, a corresponding number of FC calls need to be programmed in the user program program.

Parameter

Name:	**Partner**
Declaration:	INPUT
Data type:	INT
Range of values:	1 ... 32000 (subscriber number)
Explanation:	SINAUT subscriber no. of the partner

Name:	**PathStatus**
Declaration:	OUTPUT
Data type:	BYTE

Range of values:	Output bytes	QB0 ... QBn PQB0 ... PQBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n
Explanation:	Display of the path status of the connection to the partner	

PathStatus - Coding

Table 5- 8 Coding of the status bits in the "PathStatus" output byte

Bits 6 + 7	Bits 4 + 5	Bits 2 + 3	Bits 0 + 1
Configured interface		Path status	
No. for substitute path	No. for main path	Substitute path (2nd path)	Main path (1st path)

Configured interface (bits 4..7)

The TIM interfaces in the block are consecutively numbered decimally from 0 to 3, see "No." column in the table. The table shows the coding of the bits for the different TIM types.

Table 5- 9 Coding of bits 4 + 5 (interface of the main path) or bits 6 + 7 (interface of the substitute path)

Status bit 5 (7)	Status bit 4 (6)	Interfaces			
		No.	TIM 3V-IE	TIM 4R-IE	TIM 1531 IRC
0	0	0	Ethernet "IE1" (X2)	Ethernet "IE1" (X3)	Ethernet "IE1" (X1)
0	1	1	-	Ethernet "IE2" (X4)	Ethernet "IE2" (X2)
1	0	2	Serial "WAN1" (X1)	Serial "WAN1" (X1)	Ethernet "IE3" (X3)
1	1	3	-	Serial "WAN2" (X2)	Serial "WAN" (X4)

Path status (bits 0..3)

- Main path = 1. Path (bits 0 + 1)
- Substitute path = 2nd path (bits 2 + 3)

Table 5- 10 Status table: Coding of bits 0 + 1 or bits 2 + 3

Status bit 1 (3)	Status bit 0 (2)	Meaning bit 1	Meaning bit 0
0	0	Path not current	Subscriber not reachable
0	1	Path not current	Subscriber reachable
1	0	Path current	Subscriber not reachable
1	1	Path current	Subscriber reachable

Coding options - TIM 1531 IRC

Same coding of the configured interface for the main and the substitute path means that there is no path redundancy (only 1 interface configured). The path status is output via the bits of the main path (1st path).

Table 5- 11 Coding options for the output byte "PathStatus"

Configured interface		**Path status**	
Coding for substitute path	**Coding for main path**	**Substitute path (2nd path)**	**Main path (1st path)**
0 0	0 0 (Coding for IE1)	Irrelevant (not redundant)	Status IE1
0 0	0 1 (Coding for IE2)	Status IE1	Status IE2
0 0	1 0 (Coding for IE3)	Status IE1	Status IE3
0 0	1 1 (Coding for WAN1)	Status IE1	Status WAN1
0 1	0 0	Status IE2	Status IE1
0 1	0 1	Irrelevant (not redundant)	Status IE2
0 1	1 0	Status IE2	Status IE3
0 1	1 1	Status IE2	Status WAN1
1 0	0 0	Status IE3	Status IE1
1 0	0 1	Status IE3	Status IE2
1 0	1 0	Irrelevant (not redundant)	Status IE3
1 0	1 1	Status IE3	Status WAN1
1 1	0 0	Status WAN1	Status IE1
1 1	0 1	Status WAN1	Status IE2
1 1	1 0	Status WAN1	Status IE3
1 1	1 1	Irrelevant (not redundant)	Status WAN1

Coding options - TIM 4R-IE

Same coding of the configured interface for the main and the substitute path means that there is no path redundancy (only 1 interface configured). The path status is output via the bits of the main path (1st path).

Table 5- 12 Coding options for the output byte "PathStatus"

Configured interface		**Path status**	
No. for substitute path	**No. for main path**	**Substitute path (2nd path)**	**Main path (1st path)**
0 0	0 0 (Coding for IE1)	Irrelevant (not redundant)	Status IE1
0 0	0 1 (Coding for IE2)	Status IE1	Status IE2
0 0	1 0 (Coding for WAN1)	Status IE1	Status WAN1
0 0	1 1 (Coding for WAN2)	Status IE1	Status WAN2
0 1	0 0	Status IE2	Status IE1

0 1	0 1	Irrelevant (not redundant)	Status IE2
0 1	1 0	Status IE2	Status WAN1
0 1	1 1	Status IE2	Status WAN2
1 0	0 0	Status WAN1	Status IE1
1 0	0 1	Status WAN1	Status IE2
1 0	1 0	Irrelevant (not redundant)	Status WAN1
1 0	1 1	Status WAN1	Status WAN2
1 1	0 0	Status WAN2	Status IE1
1 1	0 1	Status WAN2	Status IE2
1 1	1 0	Status WAN2	Status WAN1
1 1	1 1	Irrelevant (not redundant)	Status WAN2

5.7.2.5 FC PulseCounter

Function

The FC PulseCounter is responsible for count pulse acquisition.

A maximum of 8 pulse strings are detected via digital inputs and fed to the function blocks with the aid of SIMATIC counters that put together the counted value frames (Cnt01D_S, Cnt04D_S).

The acquisition of the count pulses is time-controlled To do this the FC PulseCounter must be included in a cyclic interrupt OB, e.g. OB35. The call interval of the cyclic interrupt OB must be matched to the pulse duration of the count pulses. You will find more information on count pulse acquisition with the cyclic interrupt OB in the section Cyclic interrupt OB (Page 138).

Parameter

Name:	**InByte**	
Declaration:	INPUT	
Data type:	BYTE	
Range of values:	Input bytes	PEB0 ... PEBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n
	If an input byte of a digital input is specified, this must be the address of the I/O byte (PIB) directly from the digital input modules. The current status of the count input can only be detected reliably by direct access. When reading out from the process image of the inputs (PII) count pulses could remain undetected.	
Explanation:	Input byte for count pulses. The parameters for inputs for count pulse acquisition can can be set in bytes.	

Name:	**EnableMask**
Declaration:	INPUT
Data type:	BYTE
Range of values:	B#16#00 ... B#16#FF
Explanation	Enable mask for the counting inputs

With this parameter "EnableInMask" it is possible to specify in the form of a bit mask at which inputs in the input byte count pulses are actually connected. The following applies to every bit in the bit mask:

- 0 = Input bit blocked for acquisition
- 1 = Input bit enabled for acquisition

The input of the mask is only permitted in hexadecimal format B#16#00 to B#16#FF.

Input as an 8-bit binary number from 2#0 to 2#1111 1111 is not possible with the data type BYTE.

The assignment of the bits in the mask to the inputs in the input byte "InByte":

InByte	.7	.6	.5	.4	.3	.2	.1	.0
EnableMask B#16#		0 … F				0 … F		

Example: EnableMask B#16#83

Enabled are: Inputs .7, .1 and .0

Blocked are: Inputs .6 to .2

Name:	**CntIn_0 ... CntIn_7**
Declaration:	INPUT
Data type:	COUNTER
Range of values:	C0 or C1 ... Cn (n depends on the CPU)
Explanation	Pulse counter

For each of the enabled counting inputs a SIMATIC counter needs to be specified for the corresponding parameter CntIn_0 ... . "CntIn_7". With each pulse acquired, the SIMATIC counter is incremented.

The counters set here must be specified as input counters (parameter Counter_1 ... _4) in the actual counted value function blocks "Cnt01D_S" and "Cnt04D_S". These function blocks out read the assigned counter and then reset it.

As the placeholder for parameters that are not required, it is recommended to specify the counter C0.

Example of the parameter assignment of "CntIn_0" ... "CntIn_7" starting at "EnableMask" = B#16#83:

CntIn_0 : = Z10
CntIn_1 : = Z11
CntIn_2 : = Z0
CntIn_3 : = Z0
CntIn_4 : = Z0
CntIn_5 : = Z0
CntIn_6 : = Z0
CntIn_7 : = Z12

5.7.2.6 FC ST7ObjectTest

Validity

S7-300/400 CPU

Function

Calling FC ST7ObjectTest in programming error OB121 prevents a CPU STOP, if the CPU receives data with an unknown destination object no.

FC ST7ObjectTest checks why OB121 was called, i.e. which block type is missing.

- If the missing block is a data block and this data block is an instance DB of a SINAUT object, the CPU does not change to STOP.
- If no SINAUT instance DB is missing but rather another block (DB, FB, FC), you can specify the reaction with the parameter "StopInOtherCases".
  - CPU changes to STOP.
  - CPU continues to run.

For more information on the programming error OB121 and background information relating to the use of FC ST7ObjectTest, see section Programming error OB (Page 138).

Parameter

Name:	**StopInOtherCases**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Explanation:	The CPU should change to STOP in other error situations. For details of the parameter see above, section "Function". • TRUE: CPU changes to STOP. • FALSE: CPU continues to run.

5.7.2.7 FC TestCopy

Function

Using FC TestCopy, extracts of the data traffic between ST7 subscribers can be recorded or the entire traffic can be recorded. With search masks to be set in the control field of DB TestCopyData individual frame types can be filtered out and then copied from the send or receive buffer for further evaluation in the DB TestCopyData. For details, see below.

Send and receive frames are all stored in the same data block DB TestCopyData. This makes it simple to track the chronological order of the copied send and receive frames.

Requirements

To use the TestCopy function, the user program must meet the following conditions:

- The FC TestCopy function must be available on the CPU.
- DB TestCopyData must be present on the CPU and have a sufficient length.
  - To achieve this, copy DB TestCopyData (DB99) from the TD7 library to your CPU.
  - If necessary, change the length of the buffer area in the DB by increasing or decreasing the size of the array "TestCopyBuffer" in DW40, which has the default length of [0..240] WORD.
- Make the following entries in the respective communication DB of the CPU 300/400 (BComData / XComData / PComData) whose send and/or receive frames you want to write:
  - In DW32 (TestCopyDBNo) of the communication DB, enter the number of the DB TestCopyData.
  - In DW34 (TestCopyFCNo) of the communication DB, enter the number of the FC TestCopy.

  Proceed in the same way for the communication DB of CPU 1500. The tags in DB BConnectData have the same names.

Linking FC TestCopy into the user program

If the above-mentioned conditions are fulfilled, the test function is processed cyclically by the respective communication FB of the CPU.

FC TestCopy cannot be called in the user program.

Monitoring the written data

Use the ready-made watch table "TestCopyMonitor", which you can copy from the TD7 library into the "Watch and force tables" directory of the CPU.

If the settings are to be retained even after CPU startup, they can also be stored directly in the start values of the individual BConnection instances of the DB BConnectData.

5.7.2.8 DB TestCopyData

Structure of the TestCopyData DB

Areas of the TestCopyData DB

The DB for the TestCopy function is divided into the following areas (after Offset in the DB):

- **0 ... 27: User interface**

  Interface for setting the TestCopy mode and functions. The area is divided into:

  – 1 ... 13
  Filter settings for RecvCopy function and number of counted received frames

  – 15 ... 25
  Filter settings for SendCopy function and number of counted sent frames

- **28: Error display**
- **31 ... 39: Internal management pointer**
- **40 ... (Default: 523): Buffer range**

  Buffer area for storing frames that match the filter criteria.

  The buffer area must be configured as an array [0...xxxx] of WORD.

The following table shows the structure of the DB TestCopyData:

Data type / (offset)		Tag name	Format	Explanation
User interface				
DBB	0	OperationMode	BYTE	Mode
DBW	12	Recv_TgramCounter	INT	Number of copied receive frames
DBW	26	Send_TgramCounter	INT	Number of copied send frames
RecvCopy function				
DBB	1	Recv_TgrmType	BYTE	Receive filter: Message type (MT)
DBW	2	Recv_DestSubscr	INT	Receive filter: Destination subscriber no.
DBW	4	Recv_DestObject	INT	Receive filter: Destination object no.
DBW	6	Recv_SourceSubscriber	INT	Receive filter: Source subscriber no.
DBW	8	Recv_SourceObject	INT	Receive filter: Source object no.
DBW	10	Recv_StartIndex	INT	Receive filter: Start index no.
DBB	14	SpareDBB14	BYTE	*Reserve*
SendCopy function				
DBB	15	Send_TgrmType	BYTE	Send filter: Message type (MT)
DBW	16	Send_DestSubscr	INT	Send filter: Destination subscriber no.
DBW	18	Send_DestObject	INT	Send filter: Destination object no.
DBW	20	Send_SourceSubscriber	INT	Send filter: Source subscriber no.

Data type / (off-set)		Tag name	Format	Explanation
DBW	22	Send_SourceObject	INT	Send filter: Source object no.
DBW	24	Send_StartIndex	INT	Send filter: Start index no.
Error display				
DBB	28	FC_RetVal	BYTE	Error information: 0 = No error 1 = DB TestCopyData too short 10 = Unknown mode
DBB	29	SpareDBB29	BYTE	*Reserve*
DBB	30	SpareDBB30	BYTE	*Reserve*
Internal management pointer				
DBB	31	TestCopyStatus	BYTE	Status byte for TestCopy operation
DBB	32	TestCopyCmdByte	BYTE	Command byte for TestCopy operation
DBB	33	TestCopyDelCount	BYTE	Loop counter for TestCopy delete function
DBW	34	NextFreeCopyByte	INT	Address of the next free TestCopyBuffer byte
DBD	36	StartTimeSFC64	DINT	SFC64 time at the start of the copy procedure
Buffer range				
DBB	40	TestCopyBuffer[0]	BYTE	Copy area, byte 0
DBB	41	TestCopyBuffer[1]	BYTE	Copy area, byte 1
DBB	42	TestCopyBuffer[2]	BYTE	Copy area, byte 2
DBB	43	TestCopyBuffer[3]	BYTE	Copy area, byte 3
DBB	n	TestCopyBuffer[n]	BYTE	Copy area, byte n

Structure of a copied message block

A frame block can contain several frames. The frames are saved in the DB TestCopyData according to the following rules:

1. The first entry indicates the time difference in milliseconds (7 decade BCD plus sign) since the last selection of an operating mode > 0.
2. This is followed by a separation sign AAAA for sent messages, EEEE for received messages.
3. Storage of the first message from the frame block.
4. Separation identifier AAAA, or EEEE:
5. Storage of the last frame from the message block.
6. Block end identifier FFFF.

Example

- All received frames will be stored in DB TestCopyData.
- Communication via X blocks, i.e. max. 76 bytes per receive block.
- The receive buffer of the DB XComData is the source for FC TestCopy.
- The current receive block contains 3 messages.

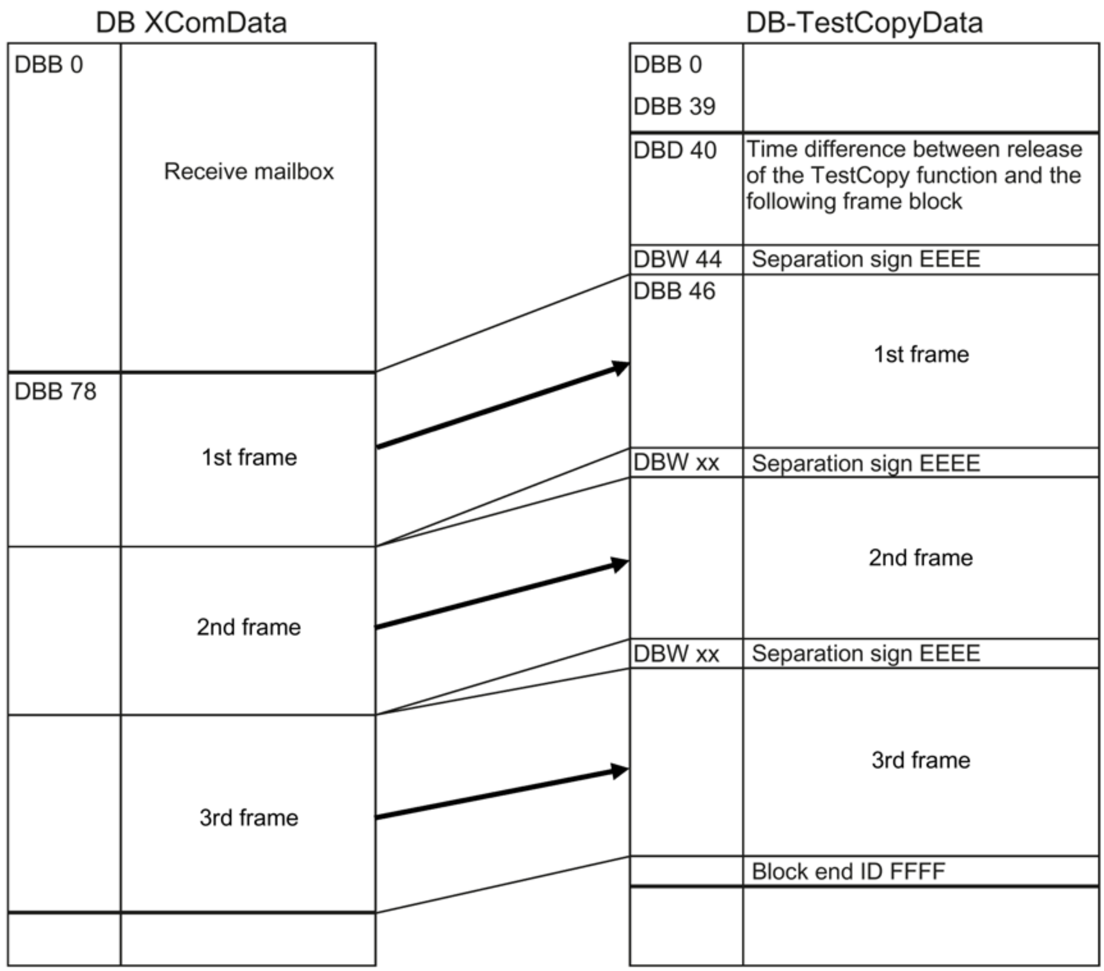


Figure 5-1 Example of filling the DB TestCopyData

Length calculation

FC TestCopy uses the following parameters for determining the minimum length for the DB TestCopyData:

Parameter	Parameter name	Length
Length of communication buffer	$Len_{ComBuffer}$	76 or 202 bytes *****
Minimum frame length	$Len_{MinTgrm}$	14 bytes
Offset management area	Offset	40 bytes

Parameter	Parameter name	Length
Length of the time difference	Len_{dt}	4 bytes
Length of the block separators	$Len_{Trenner}$	2 bytes

***** 76 with X communication, 202 with B communication

The formula used for the actual calculation is the same for X communication and B communication. The results differ only due to different lengths for the communication buffer for X and B communication:

- **Length with X communication**

Len_{Min_Xcom}	$= Len_{ComBuffer} + Offset + Len_{dt} + (Len_{ComBuffer} / Len_{MinTgrm} + 1) * Len_{Trenner}$
	= 76 + 40 + 4 + (76 / 14 + 1) * 2
	= 120 + 12 = 132 bytes minimum

- **Length with B communication**

Len_{Min_Bcom}	$= Len_{ComBuffer} + Offset + Len_{dt} + (Len_{ComBuffer} / Len_{MinTgrm} + 1) * Len_{Trenner}$
	= 202 + 40 + 4 + (202 / 14 + 1) * 2
	= 236 + 40 = 276 bytes minimum

If FC TestCopy determines that the DB TestCopyData does not have the calculated minimum length an error message to this effect is entered in data byte DBB28.

5.7.2.9 Operation of TestCopyMonitor

Operating modes and transfer directions of DB TestCopyData

In byte 0 (OperationMode) of the DB TestCopyData, the mode of the FC is coded. The values 0 ... 3 identify the mode:

- Mode 0

  Function blocked
- Mode 1

  Frame entry as of the start of DB TestCopyData
- Mode 2

  Write to DB TestCopyData endlessly as circulating buffer
- Mode 3

  Fill DB TestCopyData once, then set mode 0.

You will find the detailed coding in the table below.

Along with the mode the coding of DBB0 also includes the transfer direction of the logged data. The following assignment applies for the coding:

- Bit 0...3

  Modes (0, 1, 2, 3) for die direction "RecvCopy" (copy receive frames)
- Bit 4...7

  Modes (0, 1, 2, 3) for the direction "SendCopy" (copy send frames)

Bit	.7	.6	.5	.4	.3	.2	.1	.0
Direction	SendCopy				RecvCopy			

The functions "SendCopy" and "RecvCopy" can be activated individually or at the same time, however only a common mode is possible for both communication directions.

When a mode > 0 is set in the less significant half byte, this always applies to both communication directions.

Only if mode 0 is coded in bit 0...3 (RecvCopy), does the value in bit 4...7 apply (SendCopy).

Exception:
To delete the DB TestCopyData, DBB0 must be written with FF, 0F is not enough.

Examples of coding of DBB0:

- 00_h

  No TestCopy function enabled
- 03_h

  Only RecvCopy function (OperationMode = 3), no SendCopy function.
- 30_h

  Only SendCopy function (OperationMode = 3), no RecvCopy function.
- 33_h

  RecvCopy function and SendCopy function (OperationMode = 3)
- FF_h

  Delete content of DB TestCopyData

Filter settings via TestCopyMonitor

Run FC TestCopy via a watch table which is available in finished form as "TestCopyMonitor" in the TD7 library.

The following settings are possible using the watch table in the DB TestCopy:

Name (symbol)	**Permitted values (hex.)**	**Meaning**
OperationMode	00	Mode 0: Function blocked
	11	Mode 1: Frame entry always as of the start of DB TestCopyData
	22	Mode 2: Write to DB TestCopyData endlessly as circulating buffer
	02 20	• Write for SendCopy function • Write for RecvCopy function
	33	Mode 3: Fill DB TestCopyData once, then set mode 0.
	03 30	• Fill for SendCopy function • Fill for RecvCopy function
	FF	Delete the entire DB TestCopyData and reset to default

As the output value, FC TestCopy returns a count value in DBW12 (Recv_TgramCounter) of the DB TestCopyData for the frames received since setting mode 1, 2 or 3 that match the filter criteria.

The number of send frames is entered in DBW26 (Sendcv_TgramCounter).

In DBB28, you receive a return value that indicates the errors that occurred during processing of the FC:

- FC_RetVal = 0

  No error
- FC_RetVal = 1

  The specified DB TestCopyData is too short.
- FC_RetVal = 10d

  The mode entered in DBB0 is not defined.

In addition, the following special filters are available for the RecvCopy or SendCopy function:

Name (symbol) *	Permitted values (hex.)	Meaning
Xxxx_TgrmType	FF	Copy all frame types (TA) to DB TestCopyData
	00	Copy only spontaneous Org. frames (TA = 0)
	11	Copy only queried Org. frames (TA = 1).
	22	Copy only spontaneous data frames (TA = 2).
	33	Copy only queried data frames (TA = 3).
	01	Copy Org. frames with TA = 0 or 1
	23	Copy data frames with TA = 2 or 3
	Any combination	Copy any 0, 1, 2, 3 combinations.
Xxxx_DestSubscr	All permitted subscribers	Filter for the dest. subscriber no. in the frame
	-1	Copy all frames regardless of the dest. subscriber no.
Xxxx_DestObject	All permitted objects	Filter for the dest. object no. in the data frame.
	-1	Copy all frames regardless of the dest. object no.
Xxxx_SourceSubscr	All permitted subscribers	Filter for the source subscriber no. in the frame
	-1	Copy all frames regardless of the source subscriber no.
Xxxx_SourceObject	All permitted objects	Filter for the source object no. in the data frame
	-1	Copy all data frames regardless of the source object no.
Xxxx_StartIndex	All permitted indexes	Filter for the start index no. in the data frame
	-1	Copy all data frames regardless of the start index no.

***** "Xxxx": Placeholder for "Recv" or "Send"

The desired filter tags can be easily added using the tag names of the TestCopy DB.

Notes on operator input

When changing from one mode to the next, the content of DB TestCopyData is not deleted. Only internal pointers and frame counters in the management area of the DB TestCopyData are reset. When there is a mode change it is therefore recommended to use the delete function "FF" to preset the frame buffer area with 0. The makes the copied frame blocks easier to read.

If send and receive frames are to be copied, in the left half byte of the "OperationMode" parameter, the same mode must be entered as in the right half byte.

The following scheme applies to mode 0, 1, 2 and 3:

- If the buffer is deleted, FF_h must always be entered.
- Separate deletion of the receive and send frames is not possible.

5.7.2.10 FC TimeTask

Function

The FC TimeTask keeps a continuous date and time on a CPU The FC has no parameters.

Link the FC into the cyclic user program (in OB1) following the FC BasicTask.

FC TimeTask can only be used when the CPU is synchronized by a local TIM. Activate the time-of-day synchronization for the relevant TIM modules.

After the CPU has started up, the TIM supplies the date and current time the first time with an Org. frame. After this, time-of-day synchronization is performed at the interval specified in the configuration of the TIM. For time-of-day synchronization on MPI/partyline an interval of one minute is recommended. FC TimeTask sets the clock of the CPU with the time provided by the TIM.

The FC reads out the time in every OB1 cycle. The read time is entered in the first two double words of the DB BasicData and marked as valid or invalid and with an indication whether it is daylight saving or standard time.

From DB BasicData all blocks take the current time if they need it. For example the data point typicals do this to time stamp their data or FC Trigger to check whether a time set for the FC has been reached or a preset interval has elapsed. This time of day is also available to the user program.

Table 5- 13 Assignment of the data words with date, time and time status

CurrentDate	Data byte 0	Year * 10	Year * 1
	Data byte 1	Month * 10	Month * 1
	Data byte 2	Day * 10	Day * 1
	Data byte 3	Hour * 10	Hour * 1
CurrentTime	Data byte 4	Minute * 10	Minute * 1
	Data byte 5	Second * 10	Second * 1
	Data byte 6	Milliseconds * 100	Milliseconds * 10
	Data byte 7	Milliseconds * 1	Time status

Table 5- 14 Assignment of the half byte "time status"

0	0 = date/time invalid 1 = date/time valid
1	0 = standard time 1 = daylight saving time
2	(not used)
3	(not used)

Apart from the time status, whether or not the date/time is valid can also be determined based on data bit 16.1 "CpuClockOk". As soon as the time on the CPU is valid, this bit is set to 1 by FC TimeTask. In the user program this bit can be queried directly under the symbolic name "BasicData.CpuClockOk".

5.7.2.11 FC Trigger

Function

The FC sets an output (memory bit, data bit or digital output) at a time that can be configured by the user or at a preset time interval.

The FC resets this output after one OB1 cycle.

The FC can be called at any point in the cyclic user program (OB1) also more than once.

If the running of a program section or a software function is to be triggered using FC Trigger we recommend that FC Trigger is called directly before execution of this function. Applications for triggering functions due to the memory bit set by FC Trigger are, for example, as follows:

- Running through a function
- Calling a block
- Triggering transfer of a counted value every 2 hours

If several functions need to be activated at the same time this can be implemented by one FC Trigger block if all functions query the same memory bit set by the FC. This works, however, only when the triggered functions do not reset this memory bit themselves.

Remedy if triggered blocks reset the memory bit:

- You call FC Trigger often, every time with the same time but a different output memory bit.
- After calling FC Trigger you reproduce the set output memory bit in an appropriate number of further memory bits.

The FC accesses the SINAUT time of day in the first two data double words of the DB BasicData. These are constantly supplied if an FC TimeTask is included in the user program and this is synchronized at regular intervals by a local TIM. FC Trigger compares the time set for it with the current time only when the time in DB BasicData, data byte 7 (time status byte, bit 0 = 1) is marked as being valid.

The accuracy with which FC Trigger operates depends on the accuracy of the time and on the OB1 cycle time.

If the OB1 cycle time is lower than 1 second (this is normal) the output is set exactly at the programmed second value with the inaccuracy of the OB1 cycle time of less than 1 s.

If the OB1 cycle time is higher than 1 seconds, the FC works with a tolerance of 4 seconds. If the FC is processed too late but still within 4 seconds of the configured time, the output is still set.

The edge memory bit "Flag" configured for the FC is set at the same time as the output and reset 5 seconds after the configured time.

No placeholder parameter may be used for the edge memory bit and it must not be reset by the user program.

You can find examples of the parameter assignment for FC Trigger further below.

Parameter

Name:	**IntervalMode**
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Explanation	Point in time / time interval • FALSE = point in time • TRUE = time interval You will find examples of the parameter assignment for a time or time interval after this explanation of the parameters.

Name:	**Hour_Minute**
Declaration:	INPUT
Data type:	WORD
Explanation	Specifies the values for hours and minutes. Further explanation: Refer to the parameter "Month_Year".

Name:	**Second_Day**
Declaration:	INPUT
Data type:	WORD
Explanation	Specifies the values for seconds and day. Further explanation: Refer to the parameter "Month_Year".

Name:	**Month_Year**
Declaration:	INPUT
Data type:	WORD

Range of values:	• 00 ... 99 • FF
Explanation	Specifies the values for month and year Each parameter is in two parts. Each of the two values per parameter is specified with two digits as a BCD-coded value. • The first two digits specify the value for hours, seconds or month. • The two other digits specify the value for minutes, day or year. For parts of the parameter that are not needed FF is entered. Which parameters are permitted depends on the particular parameter and on the "IntervalMode" parameter. You will find further information following the explanation of the parameters.

Name:	**TriggerOutput**	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	Output	Q 0.0 ... I n.7
	Memory bit	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation	Trigger output The output is set for the duration of one OB1 cycle when the time or time interval set for "Hour" to "Year" is reached.	

Name:	**Flag**	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
	This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter and this should not be used.	
Explanation	Edge memory bit for the "TriggerOutput" output. No placeholder memory bit may be specified! The edge memory bit must not be reset by the user program.	

Examples of the parameter assignment of a time or time interval

IntervalMode = FALSE (or 0)

The FC operates according the time principle. When the time set is reached, the output "TriggerOutput" is set for one OB1 cycle.

All time parameters can be used to specify the point in time. Parameters not required should be assigned FF. They are ignored during the check for reaching the time.

Even when "IntervalMode" = 0, time intervals can be set in certain ranges, see the following examples.

Permitted values for the time parameters:

Hours	00-23	Day	01-31
Minutes	00-59	Month	01-12
Seconds	00-59	Year	00-99

Examples:

- IntervalMode = FALSE

  The output "TriggerOutput" is set once on 04.02.91 at 06:45:12:

  – Hour_Minute : W#16#0645
  – Second_Day : W#16#1204
  – Month_Year : W#16#0291
- IntervalMode = FALSE

  The output "TriggerOutput" is set every day at 06:00:00:

  – Hour_Minute : W#16#0600
  – Second_Day : W#16#00FF
  – Month_Year : W#16#FFFF
- IntervalMode = FALSE

  The output "TriggerOutput "is set on the 1st of every month at 06:00:00:

  – Hour_Minute : W#16#0600
  – Second_Day : W#16#0001
  – Month_Year : W#16#FFFF
- IntervalMode = FALSE

  The output "TriggerOutput" is set every year on October 1, at 06:00:00:

  – Hour_Minute : W#16#0600
  – Second_Day : W#16#0001
  – Month_Year : W#16#10FF

IntervalMode = TRUE (or 1)

The FC operates according the time interval principle. When the time value set or a multiple of it is reached, the output "TriggerOutput" is set for one OB1 cycle.

Only the specifications for hours, minutes and seconds are relevant. The date parameters are ignored. A time interval can also only be set in hours or in minutes or in seconds. Time parameters not required should be assigned FF.

The following time intervals are permitted:

- Hours: 01, 02, 03, 04, 06, 08, 12, 24
- Minutes: 01, 02, 03, 04, 05, 06, 10, 12, 15, 20, 30, 60
- Seconds: 10, 12, 15, 20, 30, 60

Examples:

- IntervalMode : TRUE

  The output "TriggerOutput" is set as follows:

  – Hour_Minute : W#16#06FF (every 6 hours)

  – Second_Day : W#16#FFFF (at 00:00:00, 06:00:00, 12:00:00 o'clock and ...)

  – Month_Year : W#16#FFFF (... at 18:00:00 o'clock)

- IntervalMode : TRUE

  The output "TriggerOutput" is set as follows:

  – Hour_Minute : W#16#FF30

  – Second_Day : W#16#FFFF (at 00:00:00, 00:30:00, 01:00:00 o'clock and ...)

  – Month_Year : W#16#FFFF (... at 01:30:00, 02:00:00, 02:30:00 o'clock etc.)

Error message during startup

The FC checks the parameters Hour_Minute, Second_Day und Month_Year in every cycle to ensure that they keep to the permitted range of values. What is permitted is also dependent on the "IntervalMode" parameter.

If the parameter assignment is incorrect, an error message is entered in the diagnostics buffer (event ID B113) only during startup. The CPU does not change to STOP. Afterwards, the FC checks the parameters without outputting error messages until the error has been eliminated.

The diagnostics message provides a precise identification of the incorrect parameter (continuous number of the parameter, i.e. 2, 3 or 4). The causes of the diagnostics message can be depend on the parameter "IntervalMode".

IntervalMode = FALSE (or 0)

The permitted ranges of values for the parameters Hours, Minutes, Seconds, Day, Month and Year were not kept to. Apart from FF, the following can be configured:

Hours	00-23	Day	01-31
Minutes	00-59	Month	01-12
Seconds	00-59	Year	00-99

"IntervalMode" = TRUE (or 1)

In this case the error can have two different causes:

- The permitted ranges of values for the parameters Hours, Minutes and Seconds were not kept to. Apart from FF, the following can be configured:
  - Hours: 01, 02, 03, 04, 06, 08, 12, 24
  - Minutes: 01, 02, 03, 04, 05, 06, 10, 12, 15, 20, 30, 60
  - Seconds: 10, 12, 15, 20, 30, 60
- A time interval can only be set in hours or in minutes or in seconds. The two unused parameters must have FF written to them. Even if FF was entered for all three named parameters, an error exists.

5.7.3 System blocks

5.7.3.1 BasicTask_* FC

Block versions

The block is available in the following versions for the different CPU types:

- **BasicTask_B**

  For S7-1500 CPU

  The block for the S7-1500 CPU cannot be used for S7-400 CPU.
- **BasicTask_B**

  For S7-400 CPU

  The block for the S7-400 CPU cannot be used for S7-1500 CPU.
- **BasicTask_X**

  For S7-300 PU with partyline
- **BasicTask_P**

  For S7-300 CPU without partyline

For information on "Partyline", see Glossary.

The tasks of the various "BasicTask_*" FCs for handling basic communication tasks are the same in the respective CPU type. The only parameter of the FC is identical in each case.

When the TD7onCPU is generated, the correct FC is automatically created in the respective CPU.

Function

The block is required in every CPU. It handles the following tasks:

- Central tasks during startup
- The processing of all communication mailboxes
- Central organizational tasks such as starting, monitoring and answering general requests.

Call FC BasicTask as the first block in OB1.

Parameter

Name:	**UserFC**
Declaration:	INPUT
Data type:	INT
Range of values:	• 1 ... 32000 Number of the FC The maximum possible number depends on the CPU. • 0 Substitute value if there is no FC present for the specified purpose.
Explanation:	Number of a user FC for specific further processing of the received data. If an FC is specified, this FC is called automatically by the user program with all received data. At the time of the call, the receive frame is still in the receive mailbox of the communication DB. The program in the user FC can read the receive frame from the receive mailbox and process it further in any way required, e.g. write the received data to an intermediate buffer. Using the user program, the required information on the communication DB can be read from the DB BasicData via the "CurrentComDB" tag in DW60. CurrentComDB contains the following information: • S7-1500 Index of the current BConnection instance in the DB BConnectData • S7-300/400 Number of the current communication DB In the opened communication DB, the start of the current receive frame in the receive mailbox can be found via the "CurrentReceivedMessage" tag in DW10.

5.7.3.2 DB BasicData

This data block provides the central data management. It contains information that must be maintained centrally for all blocks. Among other things, the data block contains the subscriber records and the connection descriptions.

DB BasicData is automatically generated in the required length preset with the subscriber and connection specific data and then saved in the block directory of the CPU.

DB BasicData exists once on every CPU.

Note

Number of DB BasicData

In the TD7 library, DB BasicData has the number DB127 and is also saved there under this number when the DB is generated for the various CPUs In principle it would be possible to change the number, but this involves a lot of work and can cause errors in the further creation of the user program.

It is therefore recommended to keep the DB number 127 free for the DB BasicData.

5.7.3.3 FC Create

Auxiliary block for putting together the data to be sent and its entry in the relevant instances of the send mailbox(es) (SendBuffer array) in the respective communication DB. These are the following DBs:

- **S7-1500**

  DB BConnectData > BConnection[n]
- **S7-400**

  DB BComDataXX
- **S7-300**
  - DB XComDataXX

    or
  - DB PComDataXX

XX is a placeholder for the sequential number (01, 02, etc.).

FC Create is required by the data point typicals for data and organizational frames and by FC BasicTask for organizational frames only.

5.7.3.4 Diagnose / Diagnostics FC

Name of the block:

- For S7-300: FC Diagnose
- For S7-1500: FC Diagnosis

Auxiliary block for entering SINAUT system messages in the diagnostics buffer of the CPU.

5.7.3.5 FC Distribute

Auxiliary block for distribution of the data located in the receive mailbox to the data point typicals responsible or to the node objects in the subscriber records.

5.7.3.6 FC Search

Auxiliary block for the following search tasks:

- Search for the initial address of a subscriber object within the subscriber records
- Search for the local object no. (Instance DB) from one of the two object reference lists for a received message with an incomplete destination address

The auxiliary block is required by almost all blocks.

5.7.3.7 FC Startup

The block is required in every CPU. It must be linked into the startup program OB100.

The block has the task of setting the startup memory bit in the DB BasicData and to reset the corresponding edge memory bit if this is still set.

The block has no parameters.

5.7.4 Communication blocks BCom (CPU1500)

5.7.4.1 FB BCom

Validity

- S7-1500
- S7-400

The block for the S7-400 CPU cannot be used for S7-1500 CPU and vice versa. The tasks of the two FCs are the same.

Function

Auxiliary block for FC BasicTask_B for processing a communication mailbox of the type DB BConnectData (S7-1500) or DB BComData (S7-400).

Via the DB, a configured PBK (programmed block communication) connection is handled using the SFBs "BSEND" and "BRCV".

FB BCom also ensures that received frames are distributed immediately to the receive objects responsible in the CPU. To do this FB BCom calls FC Distribute as an auxiliary block.

5.7.4.2 BConnect FB

Validity

S7-1500

Function

Auxiliary block for FC BasicTask for processing connections to the communication partners.

5.7.4.3 DB BConnectData

Validity

S7-1500

Function

Instance data block for the communication block FB BCom. The instance DB represents the communication mailbox and contains the following, among other things:

- The receive mailbox (ReceiveBuffer)
- A send mailbox (SendBuffer)

It also contains central data required to control and manage the PBK connection that runs via this mailbox.

The data block is required in every CPU in which FB BCom is used. If the CPU has several PBK connections the DB is required more than once.

DB BComData is automatically generated in the required length preset with the connection specific data and then saved in the block directory of the CPU.

5.7.5 Communication blocks BCom (CPU400)

5.7.5.1 FB BCom

Validity

- S7-1500
- S7-400

The block for the S7-400 CPU cannot be used for S7-1500 CPU and vice versa. The tasks of the two FCs are the same.

Function

Auxiliary block for FC BasicTask_B for processing a communication mailbox of the type DB BConnectData (S7-1500) or DB BComData (S7-400).

Via the DB, a configured PBK (programmed block communication) connection is handled using the SFBs "BSEND" and "BRCV".

FB BCom also ensures that received frames are distributed immediately to the receive objects responsible in the CPU. To do this FB BCom calls FC Distribute as an auxiliary block.

5.7.5.2 DB BComData

Validity

S7-400

Function

Instance data block for the communication block FB BCom. The instance DB represents the communication mailbox and contains the following, among other things:

- The receive mailbox (ReceiveBuffer)
- A send mailbox (SendBuffer)

It also contains central data required to control and manage the PBK connection that runs via this mailbox.

The data block is required in every CPU in which FB BCom is used. If the CPU has several PBK connections the DB is required more than once.

DB BComData is automatically generated in the required length preset with the connection specific data and then saved in the block directory of the CPU.

5.7.6 Communication blocks PCom (CPU300)

5.7.6.1 FB PCom

Validity

S7-300 CPU without partyline

FB PCom is used only with communication via the P bus. This affects the communication between a TIM and a CPU with P bus.

Function

Auxiliary block for FC BasicTask_P for processing a communication mailbox of the type DB PComData using the SFCs WR_REC and RD_REC.

Received frames are also distributed immediately to the receive objects responsible in the CPU. To do this FB PCom calls FC Distribute as an auxiliary block.

5.7.6.2 DB PComData

Instance data block for the communication block FB PCom. The instance DB makes the communication mailbox available and contains the following:

- The receive mailbox (ReceiveBuffer)
- A send mailbox (SendBuffer)

It also contains central data required to control and manage the connection that runs via this mailbox.

The data block is required in every CPU in which FB PCom is used. If the CPU has several corresponding connections the DB is required more than once.

5.7.7 Communication blocks XCom (CPU300)

5.7.7.1 FB XCom

Validity

S7-300 CPU with partyline

Function

Auxiliary block for FC BasicTask_X for processing a communication mailbox of the type DB XComData.

Via DB BComData a configured connection (PBK connection) is handled using the SFCs "X_SEND" und "X_RCV".

The FB XCom also ensures that received frames are distributed immediately to the receiving objects responsible in the CPU. To do this the FB XCom calls the FC Distribute as an auxiliary block.

5.7.7.2 DB XComData

Instance data block for the communication block FB XCom. The instance DB makes the communication mailbox available and contains the following:

- The receive mailbox (ReceiveBuffer)
- A send mailbox (SendBuffer)

It also contains central data required to control and manage the X connection that runs via this mailbox.

The data block is required in every CPU in which FB XCom is used. If the CPU has several X connections the DB is required more than once.

DB XComData is automatically generated in the required length preset with the connection specific data and then saved in the block directory of the CPU.

6 The Web server (WBM)

6.1 Supported Web browsers

Web browser

For secure access to the Web server of the TIM the following Web browsers are suitable:

- Internet Explorer (version 11)
- Google Chrome (version 56)
- Firefox (version 51)

You will find these Web browsers, information and any necessary addons on the Internet.

6.2 Establishing a connection to the WBM of the TIM

Possible connections

You can establish a connection between a PC and the TIM using the HTTP/HTTPS protocol:

- LAN connection

  With a local connection from the PC to the TIM you can connect directly.
- Connection via WAN (Internet/mobile wireless)

  The TIM must be reachable via a fixed IP address.

  With connections via the Internet / mobile wireless network you need to use the security protocol "HTTPS".

Requirements

The condition for access to the TIM is that the PC is located in the same subnet and that the TIM can be reached.

Connection to the Web server of the TIM

Follow the steps below to connect the PC to the Web server of the TIM:

1. Open the Web browser.
2. Enter the address (IP address / host name) of the TIM (or the router) in the address line of the Web browser either via the HTTP or HTTPS protocol:
   - http://<Address>
   - https://<Address>

   When selecting the protocol, make sure that it is released in the configuration of the TIM ("Web server" tab).

With HTTPS connections via the Internet when you log in the first time, a warning can appear that the Web page is not secure or that the certificate is not trustworthy. If you are sure that you have entered the correct address, ignore the message. If necessary add the connection to the exceptions (depending on the Web browser).

The logon window of the TIM opens.

3. In the "User name" input box, enter the name of a user or administrator configured in STEP 7.

   The rights assigned in "Global security settings" of the STEP 7 project apply.

4. Enter the corresponding password in the "Password" input box.
5. Click the "Log in" button.

   The Web server opens with the start page:

6.3 General functions of the WBM

Displays and symbols in the title bar

The displays and symbols in the WBM title bar have the following meaning:

Symbol	Function
User: 1	Name of the currently logged in user
Log out	User logout
Number of active sessions: 1	Number of connections to a PC
2015-01-28 14:30:37	Date and time of the last page update of the WBM in local time of the TIM (yyyy-mm-dd hh:mm:ss)
English	Displays the set WBM language You set the WBM language with the setting of the browser being used. The following languages are supported: • German • English
	The automatic update of the WBM display is enabled. The data is fetched at the interval configured under "System > Web server".
	The automatic update of the WBM display is disabled.
Turn on	Switches on the automatic update of the WBM display.
Turn off	Switches off the automatic update of the WBM display.
	Prints out the current WBM page

6.4 Start page

After logging in to the WBM, the start page appears.

On the left you will find the navigation area with the main levels of the WBM.

Navigation in the WBM

By clicking on an entry in the navigation area on the left open the WBM page you want for further information or on which you want to configure or program.

The WBM opens the first tab of the entry.

On other pages with several tabs change to the relevant tab by clicking on the tab name.

Start page

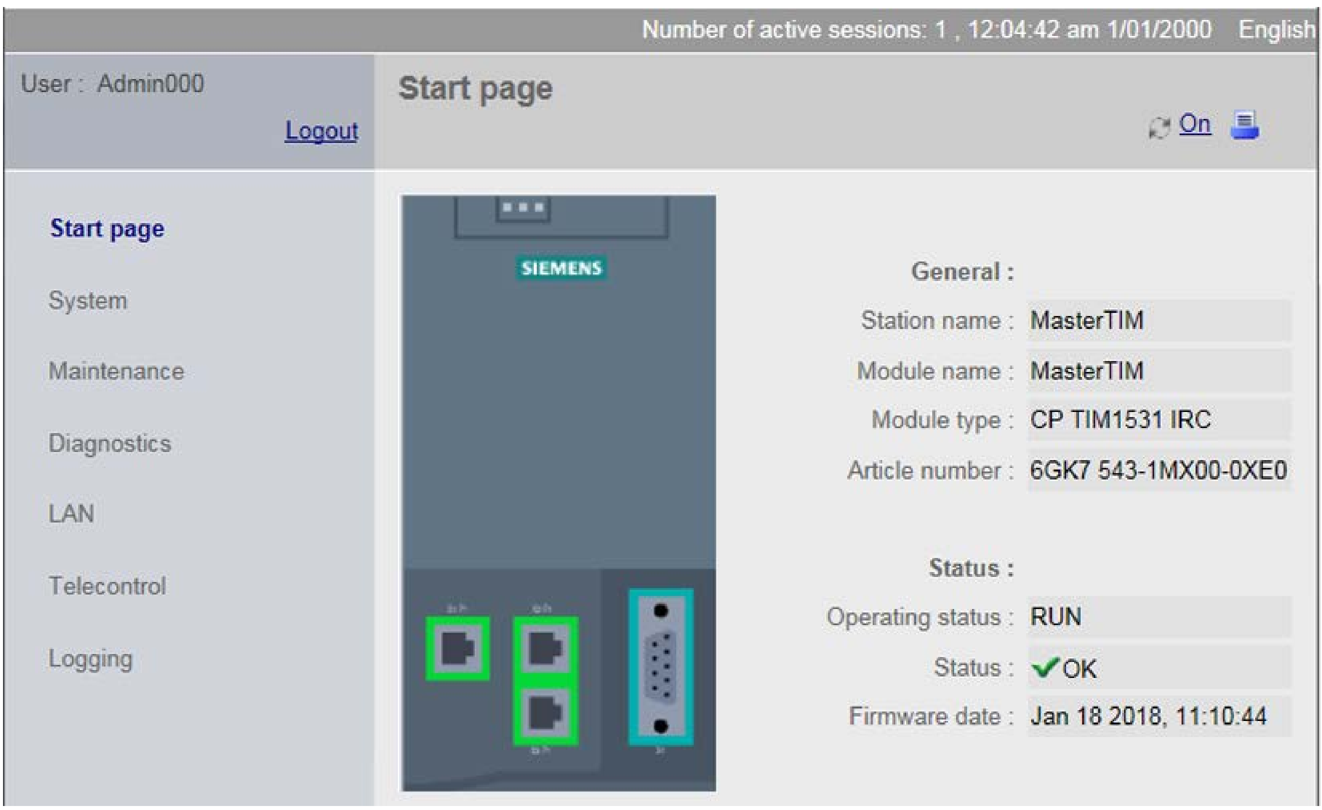


Figure 6-1 Start page of the WBM

The page shows general data of the module.

General

- **Station name**

  Parameter configured in STEP 7
- **Module name**

  Parameter configured in STEP 7

- **Module type**
- **Article number**

Status

- **Operating status**

  Current operating status of the TIM
- **Status**

  Status of the firmware startup of the TIM:
  - TIM started up free of errors
  - Startup aborted with error
- **Firmware date**

  Date the firmware currently being used was generted

  Format: MMM DD YY, hh:mm:ss

6.5 System

6.5.1 Device info

Module

- Short designation

  Parameter configured in STEP 7
- Article number
- Hardware product version
- Firmware version
- Rack
- Slot

Module information

- Device name

  - not relevant -
- Module name

  Parameter configured in STEP 7

Vendor information

- Vendor
- Serial number

  Serial number of the device

6.5.2 SD card

SD card

SD card

- **SD card inserted**

  yes / no
- **Free memory space / total**

  Display of the free memory space still available and the total usable memory capacity
- **Content**

  Display of the messages and files saved on the SD card

6.5.3 System time

System time

- **Display of the current time of day of the TIM**

  Format: YYYY-MM DD hh:mm:ss
- **New time of day**

  Using the button you can set the time of day of the TIM manually.

  When making your entry, keep to the specified format.

  Month, day, and hour can also be entered as single digits. Example: March is accepted as "03" or as "3".

  If you click the "New time of day" button, the TIM adopts the entered time of day.
- **Adopt PC time**

  If you click the "Adopt PC time" button, the TIM adopts the time of day from the connected PC.

6.5.4 NTP

NTP

- **NTP server list**

  Shows the addresses of the configured NTP servers.

6.5.5 Web server

Web server

- **Disable Web server**

  Disables the Web server of the TIM. The setting is adopted in the configuration data of the TIM.

  Note

  No HTTP/HTTPS connection to the TIM

  If you disable the Web server of the TIM, you lose the possibility to access the TIM via HTTP/HTTPS.

  Access is only possible again after loading the configuration data (with enabled Web server access).

- **Automatic update**

  Enable the option if the contents of the Web pages are to be updated automatically.

  If the option is disabled, the pages are updated at the interval you configured in STEP 7.

- **Update interval**

  Here, the update interval configured in STEP 7 is displayed.

  If the option is enabled (above) you can enter the desired update interval manually.

- **Save**

  Applies the update interval entered manually.

6.5.6 DNS configuration

DNS server list

- **List of configured DNS servers**

  Servers configured in STEP 7

6.6 Maintenance

6.6.1 Firmware

Firmware

This page displays the most important version data of the firmware currently being used.

If a new firmware version is available for the TIM, you will find this on the Internet pages of Siemens Industry Online Support:
Link: (https://support.industry.siemens.com/cs/ww/en/ps/21764/dl)

If a new firmware version is available, you can download the firmware file from the PC to the TIM via this WBM page.

Note

Digitally signed and encrypted firmware prevents manipulation by third parties

To be able to check the authenticity of the firmware, the firmware is digitally signed by Siemens. This allows manipulation by third parties to be detected and prevented.

Note

Do not operate during the update

During the update of the firmware until the TIM restarts, the WBM is not blocked.

Do not perform any operations during this time (e.g. no restart).

Note

Do not switch off the power supply

During activation of the firmware do not switch off the power supply. This avoids the occurrence of inconsistent statuses.

Firmware

The following information is shown:

- **Firmware version**

  Version of the firmware currently being used by the TIM.

- **Date**

  Date the firmware was generated

Firmware update

Download the firmware file to the file system of your connected PC.

- **File**

  After selecting a firmware file stored on the PC using the "Browse" button, the file name is displayed here.

- **Browse**

  Searches the file system of the PC for a firmware file saved there that is intended to be loaded on the TIM.

- **Load on device**

  By clicking the button, download the selected firmware file to the TIM.

  Note that updating the firmware can take a while. You can recognize the current status of the firmware download based on the LED pattern, see section Update firmware (Page 284).

After updating the firmware the TIM starts up again automatically.

6.6.2 Operating status

Operating statuses

Apart from using this WBM page you can also execute the functions described below using the switch of the TIM, see section Switch (Page 35).

The buttons have the following functions:

- **Run a restart**

  When restarting, existing telecontrol connections are interrupted and cyclic processing stops. The TIM restarts.

- **Reset to factory settings**

  Resets the TIM to the factory settings. During this all parameters are reset to the initial statuses as shipped and the TIM restarts.

  If you use an SD card and you want to reset the TIM to the factory settings, you must pull the SD card before resetting. If the SD card remains inserted, the TIM starts up again with the configuration data on the SD card.

  When pulling the SIM card, remember the information in section Resetting to factory settings (Page 287).

  Note

  Data loss: Note the effects of resetting

  Before you reset, note the effects, see section Inserting the SD card (Page 47). There, you will also find the LED patterns during the reset.

6.7 Diagnostics

6.7.1 Diagnostics messages

Diagnostics messages

Table

The table lists the last diagnostics events to occur on the TIM with the following information:

- **Number**

  Consecutive number

- **Time**

  Time of the diagnostics event

- **Date**

  Date of the diagnostics event
- **Event type**

  The diagnostics messages are classified as follows:
  - INFO

    Information about a special event
  - WARNING

    Warning of a possibly unwanted event
  - ERROR

    Internal error. The TIM starts up.
  - FATAL

    Serious error that impairs or interrupts the operation of the TIM.
- **Event**

  Plain text of the diagnostics event

Copy of the diagnostics buffer

Using the button, you save the content of the diagnostics buffer on the PC.

The diagnostics buffer

The diagnostics buffer receives diagnostics messages for internal events and errors. It can hold a maximum of 200 entries. When the maximum number is exceeded, the oldest entries are overwritten.

The entries in the diagnostics buffer contain a consecutive number, a classification, a time stamp and the message text.

Below you will find several examples of events that are entered in the diagnostics buffer:

- TIM startup
- Change to the configuration
- Establishment/abort of the communications connection
- Time-of-day synchronization
- Power failure

6.7.2 Notifications

Messages

Table

The table lists the last messages of the TIM with the following information:

- **Number**

  Consecutive number
- **Time**

  Time of sending
- **Trigger**

  Trigger that fired generation of the message.
- **Recipient**

  Configured recipient of the message
- **Message**

  Message text
- **Processing status**

  Status of the sending of the message

  You will find an overview of the possible statuses in the section Processing status of the messages (SMS / e-mail) (Page 290).
- **Type**

  Type of the message

6.8 LAN

6.8.1 Ethernet interface [Xn]

- The three Ethernet interfaces of the TIM are selected via the upper tabs.
  – X1 ... X3
- The parameters of the selected interface are shown in the lower series of tabs:
  – IPv4 parameters
  – IPv6 parameters
  – Statistics

IPv4 parameters

Network attachment

- **MAC address**

IP parameters

- **IP address**

  Current IP address
- **Subnet mask**

  Default or last configured subnet mask.
- **Default router**

  Configured default router
- **Address assignment**

  Shows how obtaining the IP address is configured in STEP 7:
  - Set IP address in the project
  - IP address from DHCP server
  - Set IP address on the device

    The IP address obtained using other services outside the configuration

Ports

- **Port number**

  Port of the interface
- **Connection status**
  - OK: Existing connection to the network
  - Not OK: No connection
- **Settings**

  Behavior of the network setting:
  - Automatic
  - Manual setting for transmission speed and direction dependency
- **Mode**

  Used transmission speed and direction dependency (duplex/half duplex)
- **Connection medium**

  Connected medium (copper / optical)

IPv6 parameters

- **IPv6 address**

  Currently used IPv6 address
- **Manual configuration**

  Display of manually configured IPv6 addresses

Statistics

Statistics

The following statistical data of the interface since the TIM last started up is displayed.

- **Bytes received**
- **Received frames discarded**

  Number of messages that were discarded on receipt due to address. protocol or data errors.
- **Error on receipt**

  Number of internal errors on receipt
- **Frames with unknown protocol**

  Number of messages with the wrong protocol
- **Bytes sent**
- **Sent unicast frames**
- **Dropped frames**

  Number of frames that were discarded due to errors when sending.
- **Error sending**

  Number of internal errors when sending
- **Frames in the send mailbox**

  Number of unsent frames waiting for transfer.

6.9 Telecontrol

6.9.1 Partner information

The tab shows you information on the communications partners and the connection status of the TIM.

Area top left

Here you will find the connection overview with information on the connection partners of the TIM.

⇒ Select a partner in the "Partner type" list e with the mouse.

Area top right

Here the information on the selected partner and the connection paths is displayed.

Area bottom right

Here information on the frame status of the TIM is displayed.

Connection overview

- **Connection status**

  The status of the connections to the assigned CPU and to the remote partners is shown as follows:

  - **Green: Connected**

    All connections are established.
  - **Yellow: Connected**

    Some of the possible connections are established.
  - **Red: Disconnected**

    None of the possible connections is established
- **Partner type**

  Possible partner types:

  - CPU (local CPU)

    The CPU assigned to the TIM in the configuration.
  - Application

    (e.g. WinCC)
  - TIM

    TIM of the remote station
  - CPU

    CPU of the remote station
  - CP ...

    CP of the remote station CP 1243-8 IRC / CP 1542SP-1 IRC)
- **Subscriber number**

  Subscriber number of the partner

Partner: Local CPU

Local CPU

- **Status**

  Operating status of the local CPU

- **Number of connections**

  Number of connections between the TIM and local CPU

Transmission path 1

- **Interface no.**

  Number of the Ethernet interface of the TIM: 1 (X1) / 2 (X2) / 3 (X3) / 4 (X4)

- **CPU type**

  Type of the local CPU

- **Connection status**

  – Connected

  – Not connected

- **CFB reference**

  Local ID (decimal) of the S7 connection

- **Local TSAP**

  Local TSAP of the S7 connection

- **Remote TSAP**

  Remote TSAP of the S7 connection

- **IP address**

  IP address of the CPU

Partner: Remote partner

Partner

Information on the partner selected on the left of the WBM page

- **Partner type**
  - Application (e.g. WinCC)
  - CPU

    CPU of the remote station
  - TIM
  - CP ...
- **Subscriber number**

  Subscriber number of the partner
- **Connection status**
  - Connected
  - Not connected

  For the meaning of the colors, see above (Connection overview).

Transmission paths

- **Number of transmission paths**

  Number of configured telecontrol connections
- **Status transmission paths**
  - All transmission paths OK
  - 1 transmission path OK
  - No transmission path OK
- **Security options**

  Display of the active access level (protection):

  ON / OFF
- **Time master**

  Display of the active option:

  Yes / No

Transmission path 1 / 2

- **Address**

  IP address or WAN address of the interface of the TIM
- **Interface**

  Interface of the TIM

- **CFB reference**

  Local ID (decimal) of the S7 connection
- **Connection type**

  Display of several of the following connection properties:
  - PBK connection

    Configured S7 connection
  - ST7

    ST7 connection via classic WAN
  - MSC connection

    Connection of the MSC protocol for which no S7 connection is required.
  - CR connection

    Read/write connection to the local CPU that does not require an S7 connection.
  - X connection

    Unconfigured S7 connection that uses the SFCs "X_SEND" and "X_RCV".
  - Permanent / temporary

    Permanent or temporary telecontrol connection
  - GPRS / no GPRS

    GPRS connection or no GPRS connection
  - Local/remote

    connection to a local or remote partner
- **Connection status**
  - Connected
  - Disconnected

Frame memory status

Display of the frame memory status of the TIM:

- **Normal operation**

  The send buffer is working normally. The memory space allocation is between 10 and 90%.
- **90% limit reached**

  When the send buffer is 90% full, the TIM switches to the forced image mode.

  See also section Process image, type of transmission, event classes (Page 113).
- **Overflow**

  Send buffer 100% full

6.9.2 Data points

The tab shows you information on the configures data points of the TIM.

Data points

- **Data point number**
- **Name & type**

  Name and type of the data point
- **Type identifier**

  Object group/Type of data point

  Only relevant for DNP3/IEC
- **Object number**

  Object number of the ST7 data point
- **Status**

  Status of the assignment (connection) of the data point to a communication partner
- **Current value**

  Currently saved value
- **Historical value**

  Previous value
- **Time stamp**

  Time stamp of the currently saved value

6.10 Logging

Functions of logging

On this page, you can log the data traffic of the TIM using PCAP functionality for diagnostics purposes.

If an error occurs or if the TIM behaves in an unwanted manner, the communication behavior of the TIM can be recorded. The frame traffic of the TIM is recorded for a defined time or for a configurable number of frames.

The log files are stored as PCAP files on the connected PC and can be evaluated with the Wireshark program, for example.

Options:

- **Ethernet Interface X1 / X2 / X3**

  Enable the interfaces for which you want to record data.

- **Data volume (kB)**

  Via the input box, you specify the overall size of the logging file.

  Maximum file size: 10000 kB

- **Recording acc. to time**

  If the option is enabled, the recording is made for a configurable time.

  Via the input box, you specify the recording time in seconds.

  Max. recording duration: 600 s

- **Recording acc. to frames**

  If the option is enabled, the recording is made for a configurable number of frames.

  Via the input box, you specify the number of frames.

  Max. number of frames: 500 s

- **Start**

  With this button you start the logging.

- **Stop**

  With this button you stop the logging.

Diagnostics and upkeep

7

7.1 Diagnostics options

The following diagnostics options are available.

LEDs of the module

For information on the LED displays, refer to the section LEDs (Page 32).

STEP 7: The "Diagnostics" tab in the Inspector window

If your engineering station is connected to the TIM via Ethernet, here you will receive information about the selected module:

- Connection status of the engineering station with the TIM

STEP 7: Diagnostics functions in the "Online > Online and diagnostics" menu

Using the online functions, you can read various diagnostics information from the TIM from an engineering station on which the STEP 7 project is stored and perform maintenance functions.

You will find additional information on the diagnostics functions of STEP 7 in the STEP 7 information system.

Online access

This is where you establish the online connection to the module.

For the procedure, refer to the section Online functions (Page 283).

Diagnostics

Here, you can obtain the following static information on the selected module:

- **General**

  General information on the module

- **Diagnostics status**

  Information on the diagnostics status

- **Ethernet interface[X1/2/3]**

  Address and statistical information

- **Industrial Remote Communication**

  Here, you obtain WAN-specific information on the TIM module:

  – **Partner**

    Here you will find address and configuration data of the partners, a connection statistic and further diagnostics information. Click on a subscriber to display further information.

    You will also find information on the partners in the WBM.

  – **Data point list**

    Information on the data points such as configuration data, value, connection status etc.

  – **Protocol diagnostics**

    With this function you can enable the logging of ST7 frames of the TIM and evaluate it using the SINAUT engineering software.

    With the function "Enable protocol trace" the frames received and sent by the module are copied for several seconds.

    With the function "Disable protocol trace", the logging is stopped and the data is written to a logging file.

    With the function "Save", you can save the log file on the engineering station.

    To evaluate the file you nedd to rename it to the format "*.7dt". You can decode and analyze the renamed file using the TIM frame monitor of the SINAUT diagnostics tool.

  – **Device-specific events**

    Here you will find diagnostics buffer entries of the TIM and an overview of the sent messages (SMS messages / e-mails).

- **Time**

  Specification of the current time in the module and the time source

Functions

You can run the following functions here:

- **Firmware update**

  For a description, see section Update firmware (Page 284).

- **Assign IP address**
- **Assign PROFINET device name**
- **Reset to factory settings**

  For a description, see section Resetting to factory settings (Page 287).

Web server (WBM) of the TIM

From a PC you can use HTTP/HTTPS to access the Web pages (WBM) of the TIM. The WBM returns a variety of information.

For access to the content, refer to the section The Web server (WBM) (Page 263).

Partner status and connection status in the WBM

You will see the configured partners and the status of the connections to the local and remote communications partners of the TIM on the page "Telecontrol" > "Partner information" of the WBM. For details, see section Partner information (Page 274).

Partner and connection information to the CPU

The TIM can signal the status of the connection and the connection paths to the communications partner to its local CPU via a PLC tag. For information on the configuration, refer to the section Communication with the CPU (Page 81).

SNMP

For information on the functions, refer to the section SNMP (Page 288).

7.2 Online functions

Online functions

Along with STEP 7 on the engineering station (ES) the TIM provides various diagnostics and maintenance functions. The requirement is that the ES and the TIM are located in the same subnet.

Connection establishment to use the online functions via Ethernet

Procedure:

1. Connect the ES to the network.
2. Open the relevant STEP 7 project on the ES.
3. Select the TIM that you want to update with new firmware.
4. Enable the online functions using the "Connect online" icon.
5. In the "Connect online" dialog, go to the Choose the entry "TeleService via telecontrol" in the "Type of PG/PC interface" drop-down list.
6. In the "PG/PC interface" drop-down list select the entry "TeleService board".
7. In the table select the TIM if it is not already selected.

   The path both via the TIM or also via the CPU is possible.
8. Click on the icon next to the "PG/PC interface" drop-down list.

   The "Establish remote connection via telecontrol" dialog box opens.
9. Make the necessary entries in this dialog (see below) and click on "Connect".

Terminate online connection

On completion of the online session, terminate the online connection again using the "Disconnect" button.

7.3 Update firmware

New firmware versions of the TIM

If a new firmware version is available for the TIM, you will find this on the Internet pages of Siemens Industry Online Support:

Link: (https://support.industry.siemens.com/cs/ww/en/ps/21764/cert)

Firmware files have the file format *.upd.

Save the firmware file on your PC.

There are different ways of loading a new firmware file on the TIM:

- Loading the firmware via the WBM
- Loading the firmware with the online functions of STEP 7 via Ethernet / Internet
- Loading the firmware from an SD card

Note

SD card only for firmware file

For the firmware file you require a separate SD card. This must meet the requirements of the TIM, see section SD cards (Page 303).

You cannot use the SD card with the configuration data.

Note

Duration of the firmware update

Downloading a new firmware file can take several minutes.

Always wait until the completion of the firmware update can be recognized from the LEDs (see below).

Loading the firmware via the WBM

Requirements:

The new firmware file is stored on your engineering station.

Procedure:

For a description of the procedure, refer to the section Firmware (Page 268).

Loading the firmware with the online functions of STEP 7 via Ethernet / Internet

Requirements:

- The TIM can be reached using an IP address.
- The engineering station and the TIM are located in the same subnet.
- The new firmware file is stored on your engineering station.
- The engineering station is connected to the network.
- The relevant STEP 7 project is open on the engineering station.

Procedure:

1. Select the TIM that you want to update with new firmware.
2. Enable the online functions using the "Connect online" icon.
3. In the "Connect online" dialog, select the Ethernet interface in the "Type of PG/PC interface" list box.
4. Select the TIM.
5. Click on "Start search" to search for the module in the network and to specify the connection path.

   When the module is found it is displayed in the table.
6. Connect using the "Connect" button.

   The "Connect online" wizard guides you through the remaining steps in installation.
7. In the network view, select the TIM and select the "Online & Diagnostics" shortcut menu (right mouse button).
8. In the navigation panel of the Online & Diagnostics view select the entry "Functions > Firmware update".
9. Using the "Browse" button (parameter group "Firmware loader") search for the new firmware file in the file system of the engineering station.
10. Start to download the firmware with the "Start update" button when the correct version of the signed firmware is displayed in the "Status" output box.

You will find further information on the online functions in the STEP 7 information system.

Loading the firmware via the SD card

Requirements:

- You have copied the new firmware file from your PC to the SD card using a suitable card reader.
- You have saved a backup file of the firmware file currently being used.

Procedure:

1. Turn off the power supply to the TIM.
2. If you use an optional SD card for the configuration data, take the SD card out of the card slot of the TIM.

   See section Switch (Page 35) for information on this.
3. Insert the SD card with the firmware file in the card slot of the TIM.
4. Turn on the power supply to the TIM.

   The TIM starts up with the new firmware from the SD card.

   Wait until the CONNECT LED flashes.
5. Turn off the power supply to the TIM.
6. Remove the SD card with the firmware file from the card slot of the TIM.

   If you use an optional SD card for the configuration data, plug this into the card slot of the TIM.
7. Turn on the power supply to the TIM again.

   The TIM starts up with the new firmware and the configuration data on the SD card.

   The LED pattern shows the end of the startup, see below.

LED patterns when updating the firmware

The table below describes the LED patterns when transferring a firmware file to the TIM.

Table 7- 1 Meaning of the LED patterns

RUN	ERROR	CONNECT	LED pattern / meaning
			The firmware file is transferred. • "RUN" flashes alternately green and yellow. • "ERROR" flashes. • "CONNECT" flashes.
			Startup successfully completed, start of the firmware

For the LED pattern during the restart, refer to the section Startup - LED pattern (Page 49).

7.4 Resetting to factory settings

Resetting to factory settings: Effect

Note

Configuration data is deleted

With the functions for resetting to factory settings described here, all configuration data on the TIM is deleted!

- **Deleted data**

  The following data is deleted by resetting to factory settings:

  – Configured IP addresses of the LAN interfaces X1 X2 and X3

  – All other configuration data in the work memory of the TIM

- **Data not deleted**

  The following data is not deleted by resetting to factory settings:

  – MAC addresses of the LAN interfaces

Note

Configuration data when using an SD card

If you use an SD card and you want to reset the TIM to the factory settings, you must pull the SD card before resetting. If the SD card remains inserted, the TIM starts up again with the configuration data on the SD card.

When pulling the SIM card, remember the information in section Inserting the SD card (Page 47).

Executing the "Reset to factory settings" function

To reset to factory settings using the switch, follow the steps outlined below:

1. Put the switch to the position "MRES" and hold it in the "MRES" position for at least 5 seconds.

   The TIM restarts.

2. During the restart, continue to hold the switch in the "MRES" position.

   If the LED pattern of step 2 occurs (see below) the TIM is reset.

For information on the switch settings, refer to the section Switch (Page 35).

LED patterns when resetting

The table below describes the LED patterns when resetting the TIM.

Table 7- 2 Meaning of the LED patterns

Step	LED pattern			Meaning
	RUN	ERROR	CONNECT	
1				After holding the switch in the "MRES" position for 5 seconds, chaser lights of the three LEDs appear (flashing alternately). The TIM restarts. Continue to hold switch.
2				As soon as the three LEDs flash three times synchronously the TIM is reset. It adopts the factory settings. You can release the button:

Startup after resetting to factory settings

After resetting return the switch of the TIM back to the "RUN" position.

The TIM restarts again. For the LED pattern see section Startup - LED pattern (Page 49).

When resetting using the WBM, the TIM starts up again automatically.

The remaining behavior depends on the use of an optional SD card:

- **Startup without SD card**

  If you do not use an SD card, the TIM starts up without configuration data with an error.

  You need to reload the configuration data.

  The TIM can be reached via its Ethernet interfaces with default IP address (see above) set in the factory.

  Generally, the defaults apply during the first commissioning, refer to the section Commissioning (Page 48).

- **Startup with SD card**

  If the SD card remains inserted, the TIM starts up again with the configuration data on the SD card.

7.5 SNMP

SNMP (Simple Network Management Protocol)

SNMP is a protocol for management and diagnostics of networks and nodes in the network. To transmit data, SNMP uses the connectionless UDP protocol.

The information on the properties of SNMP-compliant devices is entered in MIB files (MIB = Management Information Base).

Scope of performance of the TIM as an SNMP agent

The TIM supports data queries in the following SNMP versions:

- SNMPv1 (standard)
- SNMPv3 (Security)

It returns the contents of MIB objects of the standard MIB II according to RFC 1213 and the Siemens Automation MIB.

- **MIB II**

  The TIM supports the following groups of MIB objects:
  - System
  - Interfaces

    The "Interfaces" MIB object provides status information about the TIM interfaces.
  - IP
  - ICMP
  - TCP
  - UDP
  - SNMP

  The following groups of the MIB II standard are not supported:
  - Adress Translation (AT)
  - EGP
  - Transmission
- **Siemens Automation MIB**

  The following exceptions / restrictions apply to the TIM.

  Write access is permitted only for the following MIB objects of the system group:
  - sysContact
  - sysLocation
  - sysName

    A set sysName is sent as the host name using DHCP option 12 to the DHCP server to register with a DNS server.

  For all other MIB objects / MIB object groups, only read access is possible for security reasons.

Traps are not supported by the TIM.

For more detailed information about the MIB files and SNMP, refer to the manual /11/ (Page 321).

Configuration

For information on the configuration, refer to the section SNMP (Page 92).

7.6 Processing status of the messages (SMS / e-mail)

Processing status of messages

If this option is enabled in the "Trigger" tab of the message configuration of STEP 7, a status is output on the CP that provides information about the processing status of the sent message. The status is written to a PLC tag of the type DWORD. Select this tag via the "PLC tag for processing status" box.

Processing status of the telecontrol e-mails

The meaning of the statuses is as follows:

Table 7- 3 SMS: Meaning of the status ID output in hexadecimal format

Status	Meaning
0000	Transfer completed free of errors
0001	Error in the transfer, possible causes: • SIM card invalid • No network • Wrong destination phone number (number not reachable)

Table 7- 4 E-mail: Meaning of the status ID output in hexadecimal format

Status	Meaning
0000	Transfer completed free of errors
82xx	Other error message from the e-mail server Apart from the leading "8", the message corresponds to the three-digit error number of the SMTP protocol.
8401	No channel available. Possible cause: There is already an e-mail connection via the module. A second connection cannot be set up at the same time.
8403	No TCP/IP connection could be established to the SMTP server.
8405	The SMTP server has denied the login request.
8406	An internal SSL error or a problem with the structure of the certificate was detected by the SMTP client.
8407	Request to use SSL was denied.
8408	The client could not obtain a socket for creating a TCP/IP connection to the mail server.
8409	It is not possible to write via the connection. Possible cause: The communications partner reset the connection or the connection aborted.
8410	It is not possible to read via the connection. Possible cause: The communications partner terminated the connection or the connection was aborted.
8411	Sending the e-mail failed. Cause: There was not enough memory space for sending.
8412	The configured DNS server could not resolve specified domain name.
8413	Due to an internal error in the DNS subsystem, the domain name could not be resolved.
8414	An empty character string was specified as the domain name.

Status	Meaning
8415	An internal error occurred in the cURL module. Execution was aborted.
8416	An internal error occurred in the SMTP module. Execution was aborted.
8417	Requests to SMTP on a channel already being used or invalid channel ID. Execution was aborted.
8418	Sending the e-mail was aborted. Possible cause: Execution time exceeded.
8419	The channel was interrupted and cannot be used before the connection is terminated.
8420	Certificate chain from the server could not be verified with the root certificate of the module.
8421	Internal error occurred. Execution was stopped.
8450	Action not executed: Mailbox not available / unreachable. Try again later.
84xx	Other error message from the e-mail server Apart from the leading "8", the message corresponds to the three-digit error number of the SMTP protocol.
8500	Syntax error: Command unknown. This also includes the error of having a command chain that is too long. The cause may be that the e-mail server does not support the LOGIN authentication method. Try sending e-mails without authentication (no user name).
8501	Syntax error. Check the following configuration data: Alarm configuration > E-mail data (Content): • Recipient address ("To" or "Cc").
8502	Syntax error. Check the following configuration data: Alarm configuration > E-mail data (Content): • Email address (sender)
8535	SMTP authentication incomplete. Check the "User name" and "Password" parameters in the configuration.
8550	SMTP server cannot be reached. You have no access rights. Check the following configuration data: • Module configuration > E-mail configuration: – User name – Password – Email address (sender) • Alarm configuration > E-mail data (Content): – Recipient address ("To" or "Cc").
8554	Transfer failed
85xx	Other error message from the e-mail server Apart from the leading "8", the message corresponds to the three-digit error number of the SMTP protocol.

7.7 Module replacement

Startup with configuration data from the SD card

Depending on the use of an optional SD card, the STEP 7 configuration data of the TIM are stored differently.

- No use of an SD card

  The configuration data of the TIM is stored in the work memory of the TIM.

- Use of an SD card

  If an SD card is used, the configuration data is stored on the SC card and read from there each time the TIM starts up.

  If the TIM needs to be replaced, this allows simple replacement of the TIM without needing to download the configuration data again.

  If you insert the SD card of the TIM being replaced in the new TIM, when it restarts it reads the configuration data from the SD card of the replaced TIM.

8 Technical specifications

8.1 Technical specifications

Table 8- 1 Technical specifications - TIM 1531 IRC

Technical specifications		
Article number	6GK7 543-1MX00-0XE0	
Attachment to Industrial Ethernet		
Quantity	• 1 x gigabit interface (X1) • 2x Fast Ethernet interface (X2, X3)	
Design	RJ-45 jack, galvanically isolated	
Properties	Half duplex/full duplex, autocrossover, autonegotiation	
Standard / transmission speed • Standard • Transmission speeds	Gigabit interface (X1) • 1000BASE-T, IEEE 802.3ab • 10 / 100 / 1000 Mbps	Fast Ethernet interfaces (X2 / X3) • 100BASE-TX, IEEE 802.3-2005 • 10 / 100 Mbps
Permitted cable lengths (Ethernet)	**(Alternative combinations per length range) ***	
0 ... 55 m	• Max. 55 m IE TP Torsion Cable with IE FC RJ45 Plug 180 • Max. 45 m IE TP Torsion Cable with IE FC RJ45 + 10 m TP Cord via IE FC RJ45 Outlet	
0 ... 85 m	• Max. 85 m IE FC TP Marine/Trailing/Flexible/FRNC/Festoon/Food Cable with IE FC RJ45 Plug 180 • Max. 75 m IE FC TP Marine/Trailing/Flexible/FRNC/Festoon/Food Cable + 10 m TP Cord via IE FC RJ45 Outlet	
0 ... 100 m	• Max. 100 m IE FC TP Standard Cable with IE FC RJ45 Plug 180 • Max. 90 m IE FC TP Standard Cable + 10 m TP Cord via IE FC RJ45 Outlet	
Serial interface for connection to the transmission device		
Quantity	1 x serial interface (X4)	
Design	9-pin D-sub male connector, isolated	
Standards	RS-232 / RS-485 (can be changed in the configuration)	
Transmission speeds	300 ... 115 200 Bps (depending on the connected modem)	
Power supply		
Design	Socket (X80) with terminal block	1 two-pole terminal block with polarity protection
Power supply	• Type of voltage • Permitted low limit • Permitted high limit	• 24 VDC • 19.2 V • 28.8 V

Technical specifications		
Cable cross-section connectable to the terminal block	• Without wire end ferrule • With wire end ferrule • With TWIN wire end ferrule	• 0.2 .. 2.5 mm² / AWG 24 .. 13 • 0.25 .. 1.5 mm² / AWG 24 .. 16 • 0.5 .. 1.0 mm² / AWG 20 .. 17
Further electrical data		
Details for a TIM with all interfaces connected:		
Current consumption (typical)	160 mA	
Effective power loss (typical)	4 W	
Overvoltage category according to IEC / EN 60664-1	Category II	
Permitted ambient conditions		
Ambient temperature	During operation with the rack in-stalled horizontally	0 °C ... +70 °C
	During operation with the rack in-stalled vertically	0 °C ... +50 °C
	During storage	-40 °C to +70 °C
	During transportation	-40 °C to +70 °C
Relative humidity	During operation	≤ 60 % at 25 °C, no condensation
Permitted contaminant concentration	Corrosive gas test according to ISA-S71.04 severity level G1, G2, G3	
	• SO_2 • H_2S	• < 0.5 ppm • < 0.1 ppm
Design, dimensions and weight		
Module format	Compact module S7-1500	
Degree of protection	IP20	
Weight	525 g	
Dimensions (W x H x D)	70 x 147 x 129 mm	
Installation options	DIN rail for SIMATIC S7-1500 (article numbers 6ES7590-1Axx0-0AB0)	
Product functions **		

***** For details, refer to the IK PI catalog, cabling technology
****** You will find further characteristics and performance data in the section Application and functions (Page 13).

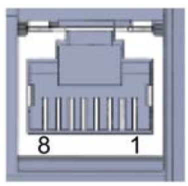
8.2 Pinout of the Ethernet interfaces

Pinout of the Ethernet interfaces

The tables below show the pin assignment of the Ethernet interfaces.

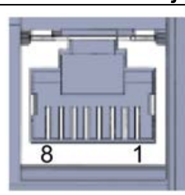
- **X1**

  Gigabit interface

View of the RJ-45 jack	Pin	Signal name	Assignment
	1	D1+	D1+ bidirectional
	2	D1-	D1- bidirectional
	3	D2+	D2+ bidirectional
	4	D3+	D3+ bidirectional
	5	D3-	D3- bidirectional
	6	D2-	D2- bidirectional
	7	D4+	D4+ bidirectional
	8	D4-	D4- bidirectional

- **X2 / X3**

  Fast Ethernet interfaces

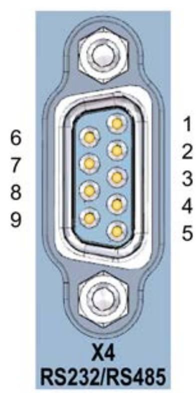
View of an RJ-45 jack	Pin	Signal name	Assignment
	1	Tx+	Transmit Data +
	2	Tx-	Transmit Data -
	3	Rx+	Receive Data +
	4	-	
	5	-	
	6	Rx-	Receive Data -
	7	-	
	8	-	

8.3 Pin assignment of the serial interface

Pin assignment of the serial interface X4 (RS-232 / RS-485)

The table below shows the pin assignment of the 9-pin D-sub miniature plug of the serial interface. The interface corresponds to the connector assignment of a standardized PC connector.

Table 8- 2 Pinout of the plug of the serial interface

Illustration	Pin no.	Signal name	Meaning / remarks	Signal direction
	1	DCD	Received signal level The DCE reports the input of data to be sent to the DTE (connection establishment). [1], [2]	Input
	2	RxD	Received data (DCE → DTE) Switchover to RS-485 by configuration [3]	Input
	3	TxD	Send data (DTE → DCE) Switchover to RS-485 by configuration [3]	Output
	4	DTR	DTE signals readiness to sent to DCE.	Output
	5	GND	Reference mass of the interface	
	6	DSR	DCE reports readiness for operation to DTE.	Input
	7	RTS	Turn on transmitter The DTE requests the DCE to send data on the data cable. The DTE waits for confirmation of the readiness to to send (CTS) of the DCE.	Output
	8	CTS	Ready to send The DCE can transfer the data coming from the DTE.	Input
	9	-	-	-
	Schirm	-	On connector housing	-
	;[1] DCE = data communication equipment (connected modem) ;[2] DTE = data terminal equipment (TIM / CP) [3] Fore the switchover RS-232 ↔ RS-485 see section .			

9 Approvals

Approvals issued

Note

Issued approvals on the type plate of the device

The specified approvals apply only when the corresponding mark is printed on the product. You can check which of the following approvals have been granted for your product by the markings on the type plate.

Scope of validity of the approvals

The approvals listed below are only valid for the TIM module.

The products of the accessories program have their own approvals, that are not listed here.

EC declaration of conformity

The product meets the requirements and safety objectives of the following EC directives and it complies with the harmonized European standards (EN) for programmable logic controllers which are published in the official documentation of the European Union.

- **2014/34/EU (ATEX explosion protection directive)**

  Directive of the European Parliament and the Council of 26 February 2014 on the approximation of the laws of the Member States concerning equipment and protective systems intended for use in potentially explosive atmospheres, official journal of the EU L96, 29/03/2014, pages. 309-356

- **2014/30/EU (EMC)**

  EMC directive of the European Parliament and of the Council of February 26, 2014 on the approximation of the laws of the member states relating to electromagnetic compatibility.; official journal of the EU L96, 29/03/2014, pages. 79-106

- **2011/65/EU (RoHS)**

  Directive of the European Parliament and of the Council of 8 June 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment, official journal of the EC L174, 01/07/2011, page 88-110

The EC Declaration of Conformity is available for all responsible authorities at:

Siemens Aktiengesellschaft
Division Process Industries and Drives
Process Automation
DE-76181 Karlsruhe
Germany

You will find the EC Declaration of Conformity on the Internet at the following address:
Link: (https://support.industry.siemens.com/cs/ww/en/ps/21764/cert)
> Ceificate type: "EC declaration of conformity"

IECEx

The product meets the requirements of explosion protection according to IECEx.

IECEx classification: Ex nA IIC T4 Gc

IECEx certificate: IECEx DEK 14.0025X

The product meets the requirements of the following standards:

- IEC 60079-0

  Hazardous areas - Part 0: Equipment - General requirements

- EN 60079-15

  Explosive atmospheres - Part 15: Equipment protection by type of protection 'n'

You can see the current versions of the standards in the IECEx certificate that you will find on the Internet at the following address:
Link: (https://support.industry.siemens.com/cs/ww/en/ps/21764/cert)

Note the conditions for the safe deployment of the product according to the section Notes on use in hazardous areas according to ATEX / IECEx (Page 41).

You should also note the information in the document "Use of subassemblies/modules in a Zone 2 Hazardous Area" that you will find on the Internet at the following address:
Link: (https://support.industry.siemens.com/cs/ww/en/view/78381013)

ATEX

The product meets the requirements of the EC directive:2014/34/EC "Equipment and Protective Devices for Use in Potentially Explosive Atmospheres".

ATEX approval: II 3 G Ex nA IIC T4 Gc

Type Examination Certificate: KEMA 07ATEX0145 X

Applied standards:

- EN 60079-0

  Hazardous areas - Part 0: Equipment - General requirements

- EN 60079-15

  Explosive atmospheres - Part 15: Equipment protection by type of protection 'n'

The current versions of the standards can be seen in the EC Declaration of Conformity, see above.

The conditions must be met for the safe deployment of the product according to the section Notes on use in hazardous areas according to ATEX / IECEx (Page 41).

You should also note the information in the document "Use of subassemblies/modules in a Zone 2 Hazardous Area" that you will find on the Internet at the following address:
Link: (https://support.industry.siemens.com/cs/ww/en/view/78381013)

EMC

The product meets the requirements of the EC Directive 2014/30/EU "Electromagnetic Compatibility" (EMC directive).

Applied standards:

- EN 61000-6-4

  Electromagnetic compatibility (EMC) - Part 6-4: Generic standards - Emission standard for industrial environments

- EN 61000-6-2

  Electromagnetic compatibility (EMC) - Part 6-2: Generic standards - Immunity for industrial environments

RoHS

The product meets the requirements of the EC directive 2011/65/EU on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Applied standard:

- EN 50581:2012

c(UL)us

Applied standards:

- Underwriters Laboratories, Inc.: UL 61010-1 (Safety Requirements for Electrical Equipment for Measurement, Control, and Laboratory Use - Part 1: General Requirements)
- IEC/UL 61010-2-201 (Safety requirements for electrical equipment for measurement, control and laboratory use. Particular requirements for control equipment)
- Canadian Standards Association: CSA C22.2 No. 142 (Process Control Equipment)

Certificate Number: 20130830-E85972

cULus Hazardous (Classified) Locations

Underwriters Laboratories, Inc.: CULUS Listed E223122 IND. CONT. EQ. FOR HAZ. LOC.

Applied standards:

- ANSI ISA 12.12.01
- CSA C22.2 No. 213-M1987

APPROVED for Use in:

- Cl. 1, Div. 2, GP. A, B, C, D T4
- Cl. 1, Zone 2, GP. IIC T4

Ta: Refer to the temperature class on the type plate

Note the conditions for the safe deployment of the product according to the section Notes on use in hazardous areas according to UL HazLoc and FM (Page 42).

FM

Factory Mutual Approval Standard Class Number 3600, 3611, 3810

Equipment rating:
Class I, Division 2, Group A, B, C, D, Temperature Class T4, Ta = 70 °C
Class I, Zone 2, Group IIC, Temperature Class T4, Ta = 70 °C

Ta: Refer to the temperature class on the type plate

Note the conditions for the safe deployment of the product according to the section Notes on use in hazardous areas according to UL HazLoc and FM (Page 42).

Australia - RCM

The product meets the requirements of the AS/NZS 2064 standards (Class A).

Marking for the customs union

EAC (Eurasian Conformity)

Customs union of Russia, Belarus and Kazakhstan

Declaration of the conformity according to the technical regulations of the customs union (TR CU)

MSIP 요구사항 - For Korea only

Registration Number: MSIP REI S7M

A급 기기(업무용 방송통신기자재)

이 기기는 업무용(A급) 전자파 적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정 외의 지역에서 사용하는것을 목적으로 합니다.

Current approvals

SIMATIC NET products are regularly submitted to the relevant authorities and approval centers for approvals relating to specific markets and applications.

If you require a list of the current approvals for individual devices, consult your Siemens contact or check the Internet pages of Siemens Industry Online Support:

Link: (https://support.industry.siemens.com/cs/ww/en/ps/21764/cert)

Dimension drawings

A

All dimensions in the dimension drawings are in millimeters.

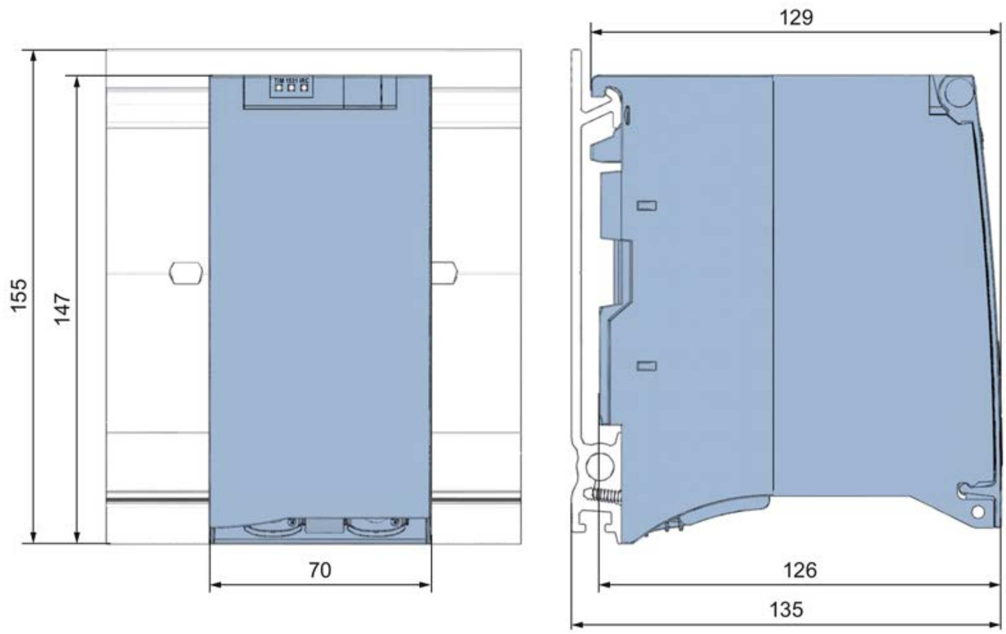


Figure A-1 TIM 1531 IRC: Front view and side view

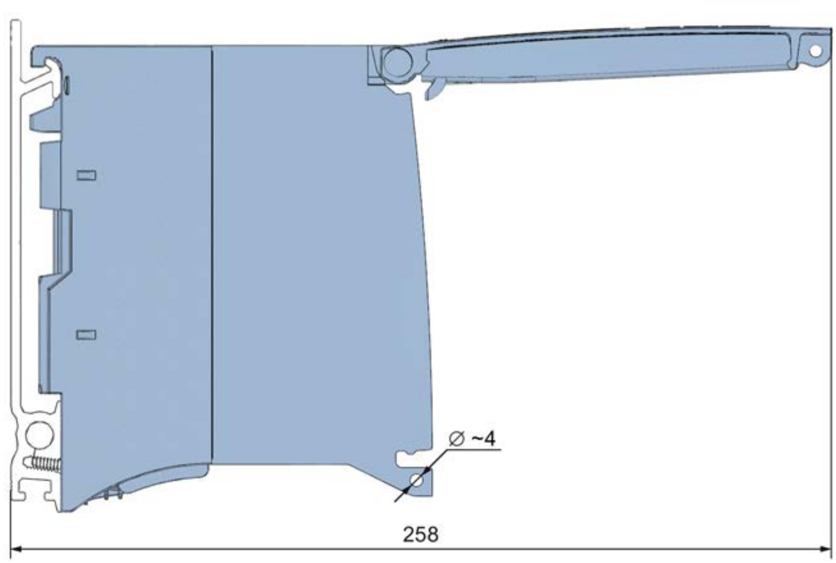


Figure A-2 TIM 1531 IRC: Side view with cover open

Accessories

B

B.1 Power supply

Power supplies for the TIM

Excerpt from the Siemens program for power supplies SITOP and S7-1500:

- SITOP PSU100C

  24 V / 0.6 A stabilized power supply, input: AC 120/230 V, output: DC 24 V / 0.6 A

  Article number: 6EP1331-5BA00
- SIMATIC PM 1507 24 V / 3 A

  Stabilized power supply for SIMATIC S7-1500, input: AC 120/230 V, output: DC 24 V / 3 A

  Article number: 6EP1332-4BA00
- SIMATIC PM 1507 24 V / 8 A

  Stabilized power supply for SIMATIC S7-1500, input: AC 120/230 V, output: DC 24 V / 8 A

  Article number: 6EP1333-4BA00

B.2 SD cards

Compatible SD cards

To store configuration data and firmware files you have the option of using an SD card. To achieve the number of frames named in the section Performance data and configuration limits (Page 25), the card should have a minimum size of 24 MB.

Note

Temperature range of the SD card

When using an SD card, make sure that this is suitable for the industrial temperature range from -40 ... +85 °C.

The following card formats can be used:

- Size: 24 x 32 mm
- Standard and capacity:
  - SDSC (SD 1.0 / SD 1.1)

    24 MB ... 2 GB
  - SDHC

    4 ... 32 GB
- Speed classes: Class 0, 2, 4, 6, 10
- File system: FAT32

You will find information on using the SD card in the section SD card slot (Page 37).

SIMATIC S7 - Memory Card (SMC)

The SMC is available with various capacities. The recommended memory capacity is in the range between 24 MB and 32 GB. SMCs up to 32 GB memory capacity are supported.

Article numbers: 6ES7954-8Lx02-0AA0
x is a placeholder for: F / L / P / T

You will find the cards on the pages of the Siemens Industry Mall using the search term "6ES7954".

B.3 Routers, modems, antennas

B.3.1 Dedicated line and dialup network modems

Modems for dedicated line and dialup networks

Note

Discontinuation of modules

The following products have the product status "type discontinued" but if they exist can be operated with the communications moduel:

- Modem MD2

  Dedicated line modems

  Product notification on the Internet:
  Link: (https://support.industry.siemens.com/cs/ww/en/view/109740149)
- Modem MD3

  Modems for analog dialup networks

  Product notification on the Internet:
  Link: (https://support.industry.siemens.com/cs/ww/en/view/109740148)
- Modem MD4

  Modems for ISDN networks

  Product notification on the Internet:
  Link: (https://support.industry.siemens.com/cs/ww/en/view/67637816)

When using the serial interface for dedicated line and dialup networks, use suitable products of other vendors.

B.3.2 MODEM MD720

MODEM MD720

Article number: 6NH9720-3AA01-0XX0

Use in SIMATIC S7 stations that are part of a telecontrol or remote maintenance system and for communication with other stations in the network or an OPC server in the master station.

The MD720 supports the following types of communication:

- IP-based communication with the control center using GPRS and the MSC protocol or the MSCsec secure protocol
- SMS messages from or to a mobile telephone
- CSD communication for maintenance and for data connections

Technical specifications (excerpt)

Connection to Industrial Ethernet		
X1 interface	Number:	1
	Implementation:	D-sub 9-pin, female
	Characteristics:	RS-232 Control using AT commands
	Transmission speed:	19200 bps Permitted range: 300 ... 57600 bps
Wireless interface		
Antenna connector	Number:	1
	Implementation:	SMA socket
	Impedance:	50 Ω nominal
Frequency bands	GPRS / CSD:	Quad band: 850, 900, 1800, 1900 MHz

GPRS	Characteristics:	Maximum of 5 time slots at the same time, of which: • Up to 2 uplinks • Up to 4 downlinks
	Transmission speed • Uplink (modem → Internet) • Downlink (Internet → modem)	Gross values: • Max. 42 kbps • Max. 54 kbps The net values (user data) are approximately 30% lower.
CSD	Characteristics:	MTC (Mobile Terminated Call)
	Transmission speed:	9600 bps
SMS (TX)	Characteristics:	Text mode

B.3.3 Router SCALANCE M

Routers for IP-based communication

To connect a communications module to IP-based infrastructure networks, the following routers are available:

- SCALANCE M812

  ADSL router for wired IP communication via the Internet, VPN, firewall, NAT, 1 RJ-45 Ethernet interface,1 digital input, 1 digital output, ADSL2T or ADSL2+

  – ADSL2T (analog phone connection - Annex A)
    Article number: 6GK5812-1AA00-2AA2

  – ADSL2+ (ISDN connection - Annex B)
    Article number: 6GK5812-1BA00-2AA2

- SCALANCE M816

  ADSL router for wired IP communication via the Internet, VPN, firewall, NAT, 1 RJ-45 Ethernet interface with 4-port switch,1 digital input, 1 digital output, ADSL2T or ADSL2+

  – ADSL2T (analog phone connection - Annex A)
    Article number: 6GK5816-1AA00-2AA2

  – ADSL2+ (ISDN connection - Annex B)
    Article number: 6GK5816-1BA00-2AA2

- SCALANCE M826-2

  SHDSL router for IP communication via 2- and 4-wire cables, ITU-T standard G.991.2 / SHDSL.biz, SHDSL topology: Point-to point, bonding, line bridge mode; routing mode with VPN, firewall, NAT, 1 Ethernet interface with 4-port switch, 1 digital input, 1 digital output
  Article number 6GK5826-2AB00-2AB2

- SCALANCE M874-2

  2.5G router for wireless IP communication via 2.5G mobile phone, VPN, firewall, NAT, 1 RJ45 Ethernet interface with 2-port switch, SMA antenna connector, 1 digital input, 1 digital output
  Article number: 6GK5874-2AA00-2AA2
- SCALANCE M874-3

  3G router for wireless IP communication via 3G mobile phone HSPA+, VPN, firewall, NAT, 1 RJ45 Ethernet interface with 2-port switch, SMA antenna connector, 1 digital input, 1 digital output
  Article number: 6GK5874-3AA00-2AA2
- SCALANCE M876-3

  3G router for wireless IP communication via 3G mobile phone HSPA+/EV-DO, VPN, firewall, NAT, 1 RJ45 Ethernet interface with 4-port switch, SMA antenna connector, antenna diversity,1 digital input, 1 digital output
  Note network provider approvals!
  – International version
    Article number: 6GK5876-3AA02-2BA2
  – Version for Korea
    Article number: 6GK5876-3AA02-2EA2
- SCALANCE M876-4

  4G router for wireless IP communication via LTE mobile phone, VPN, firewall, NAT, 1 RJ45 Ethernet interface with 4-port switch, 2 SMA antenna connectors, MIMO technology, 1 digital input, 1 digital output
  – Version for Europe
    Article number: 6GK5876-4AA00-2BA2
  – Version for North America
    Article number: 6GK5876-4AA00-2DA2

Information on the devices can be found on the Internet pages of Siemens Industry Online Support:
Link: (https://support.industry.siemens.com/cs/ww/en/ps/15982)

B.3.4 Mobile wireless antennas

GSM/GPRS antennas

The following antennas are available for use in GSM/GPRS networks and can be installed both indoors and outdoors. The antennas must be ordered separately.

Quadband antenna ANT794-4MR

You will find detailed information in the device manual. You will find this on the Internet on the pages of Siemens Industry Online Support:
Link: (https://support.industry.siemens.com/cs/ww/en/view/23119005)

Figure B-1 ANT794-4MR GSM/GPRS antenna

Short name	Article number	Explanation
ANT794-4MR	6NH9 860-1AA00	Quadband antenna (900, 1800/1900 MHz, UMTS); weatherproof for indoor and outdoor areas; 5 m connecting cable connected permanently to the antenna; SMA connector, including installation bracket, screws, wall plugs

Flat antenna ANT794-3M

Figure B-2 Flat antenna ANT794-3M

Short name	Article number	Explanation
ANT794-3M	6NH9 870-1AA00	Flat antenna (900, 1800/1900 MHz); weatherproof for indoor and outdoor areas; 1.2 m connecting cable connected permanently to the antenna; SMA connector, including adhesive pad, screws mounting possible

You will find detailed information in the device manual. You will find this on the Internet on the pages of Siemens Industry Online Support:
Link: (https://support.industry.siemens.com/cs/ww/en/view/48729835)

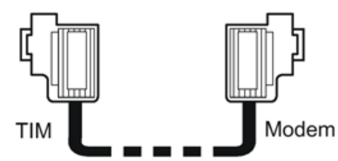
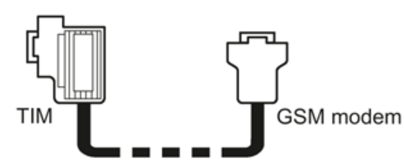
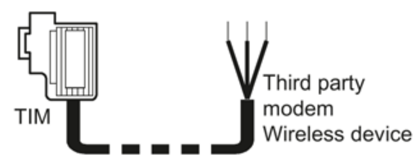
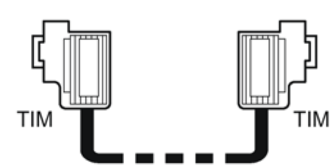
B.4 Cables, connecting cables

B.4.1 Connecting cables for connecting modems

Standard connecting cables (RS-232)

The following connecting cables do not ship with the TIM.

Table B- 1 Standard connecting cables for connecting TIM and modem

Article number	Description	Illustration
6NH7701-4AL	Connecting cable for connecting the serial interface of the TIM (RS-232) with a modem D-sub male connector Cable length 1.5 m Suitable for the connection of the following modems: • Modem for a dedicated line (MD2) • Modem for analog dialup network (MD3) • Modem for analog dialup network (MD4) • GSM mobile wireless modem MODEM MD720 (can be used for MD720 with the supplied gender changer)	
6NH7701-5AN	Connecting cable for connecting the serial interface of the TIM (RS-232) with a GSM mobile wireless modem 1 D-sub male connector, 1 D-sub female connector Cable length 2.5 m Suitable for the connection of the following modems: • GSM mobile wireless modem MODEM MD720 (can be used without gender changer) • Third-party modems with RS-232 connector • Wireless devices with RS-232 connector	
6NH7701-4BN	Connecting cable open at one end for connecting the serial interface of the TIM (RS-232) with a third-party modem Cable length 2.5 m	
6NH7701-0AR	Test cable Crossover connecting cable for connection of two TIM modules via their RS-232 interface (without the interposition of a null modem) Cable length 6 m	

Plug pin assignment of the standard connecting cables

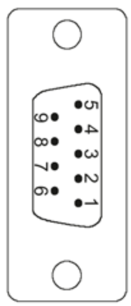
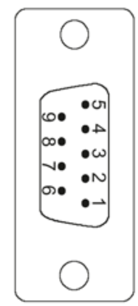
TIM (RS-232)	Pin	Interconnection	Pin	Modem (RS-232)
 D-sub female connector 9-pin	Housing shield		Housing shield	 D-sub female connector 9-pin
	1	DCD	1	
	2	RxD	2	
	3	TxD	3	
	4	DTR	4	
	5	GND	5	
	6	DSR	6	
	7	RTS	7	
	8	CTS	8	
	9	RI	9	

Figure B-3 6NH7701-4AL

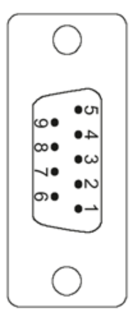
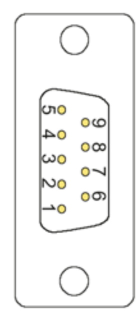
TIM (RS-232)	Pin	Interconnection	Pin	Modem (RS-232)
 D-sub female connector 9-pin	Housing shield		Housing shield	 D-sub male connector 9-pin
	1	DCD	1	
	2	RxD	2	
	3	TxD	3	
	4	DTR	4	
	5	GND	5	
	6	DSR	6	
	7	RTS	7	
	8	CTS	8	
	9	RI	9	

Figure B-4 6NH7701-5AN

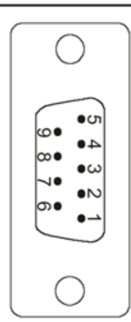
TIM (RS-232)	Pin	Interconnection	Pin	Color code (wires)
 D-sub female connector 9-pin	Housing shield		Housing shield	
	1	DCD	1	white
	2	RxD	2	brown
	3	TxD	3	green
	4	DTR	4	yellow
	5	GND	5	gray
	6	DSR	6	pink
	7	RTS	7	blue
	8	CTS	8	red
	9	RI	9	black

Figure B-5 6NH7701-4BN

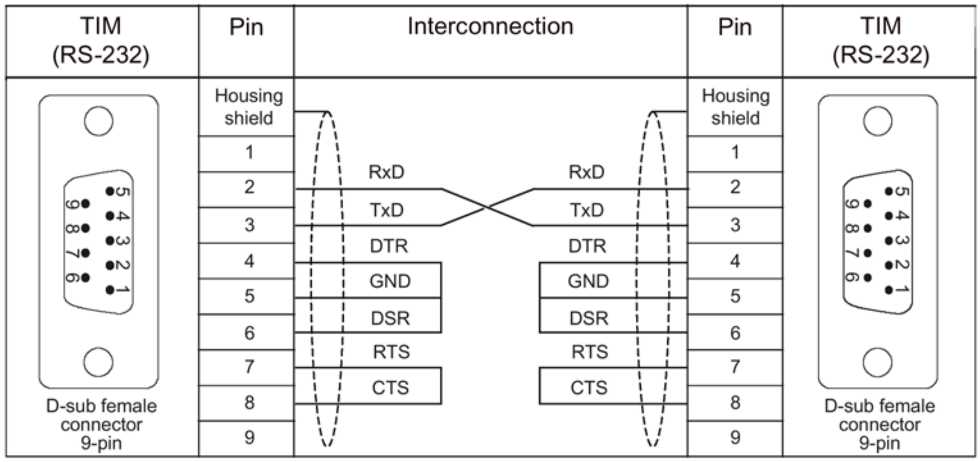


Figure B-6 6NH7701-0AR

B.4.2 Cable for RS-485 connection

Accessories for RS-485 operation of the serial interface

Excerpt from the Siemens accessories program PROFIBUS or RS-485 operation

- **Cable**
  - PROFIBUS FC standard cable GP, bus cable 2-wire, shielded, special design for fast installation, sold by the meter

    02YSY (ST) CY, 1x2x0.64 / 2.55-150 VI KF 40 FR

    Article number: 6XV1830-0EH10
- **Terminating resistor**

  In a network in RS-485 operation the terminating resistor of the bus cable is turned on or off by the STEP 7 configuration.

The connecting cable for the serial interface in RS-485 operation must be assigned as follows:

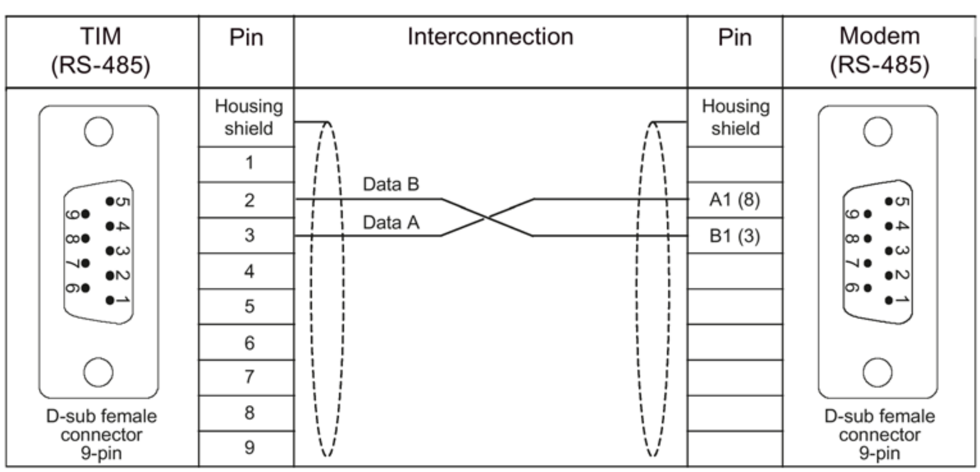


Figure B-7 Assignment of the cable for the RS-485 interface

STEP 7 V5 configuration of the proxy

C

Note

Use of the TIM

The information on configuration in STEP 7 V5 applies only in situations where you want to use the TIM as a communication module with the functions of a TIM 4R-IE in existing SINAUT ST7 systems.

See also

Expansion of SINAUT projects (Page 28)

C.1 Configuration in STEP 7 V5

Configuration of a proxy in STEP 7 V5

1. Open the STEP 7 V5 project that you wish to expand with S7-1200/1500 modules.
2. Create an S7-300 station with CPU and the new necessary networks, if applicable.

   The CPU only serves as placeholder to receive the SINAUT subscriber number.
3. Insert the "proxy" in the station.
4. Perform the configuration in STEP 7.

   Deviations from the normal configuration are described below.
5. Perform the configuration in the connection configuration of the SINAUT engineering software.
6. Export the configuration data of the proxy via the SINAUT diagnostics and service tool.

   For information on this, refer to section Exporting configuration data (Page 316).

   The data exported from STEP 7 V5 is adopted later in the STEP 7 Basic project for the S7-1200/1500 communication module.

C.2 Proxies in HW Config

PROXY modules in HW Config

You can find the proxy modules as representatives of S7-1200/1500 communication modules in the HW Config catalog under:

SIMATIC 300 > SINAUT ST7 > TIM IE

They have the following names:

- PROXY CP1243-8 IRC
- PROXY TIM 1531 IRC

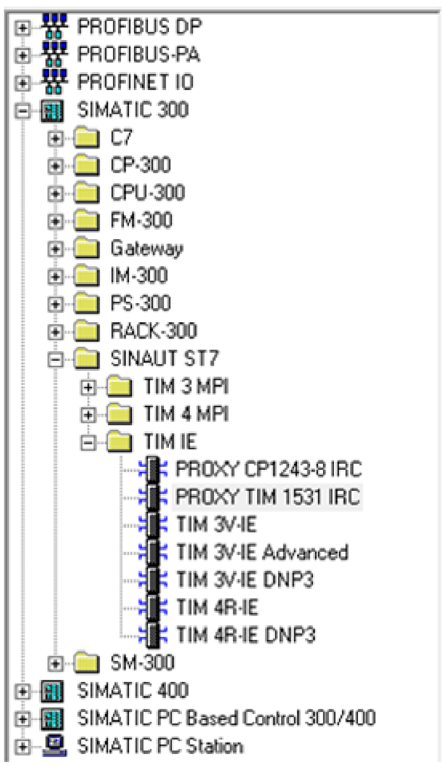


Figure C-1 PROXY modules in the catalog

C.3 Special features of the PROXY TIM 1531 IRC

Properties of the TIM or the "PROXY TIM 1531 IRC"

Compared to TIM 4R-IE, the TIM 1531 IRC and PROXY TIM 1531 IRC have the following differences:

- **Interfaces**

  Similar to the TIM, the proxy has the following interfaces:

  – 3 Ethernet interfaces

    Ethernet interface X1 can be configured for using the MSC protocol.

  – 1 serial WAN interface

    The WAN interface can also be configured as an Ethernet interface for using the MSC protocol.

You perform the configuration in the "Interfaces" tab as for TIM 4R-IE. The "Modem type" and "Connection mode" parameters correspond to those of the TIM 4R-IE.

C.4 SINAUT configuration

Connection configuration with SINAUT Engineering Software

Configure the connection of a proxy in the SINAUT configuration tool in the same way as for a TIM.

In the subscriber administration of the configuration tool, a proxy appears as follows:

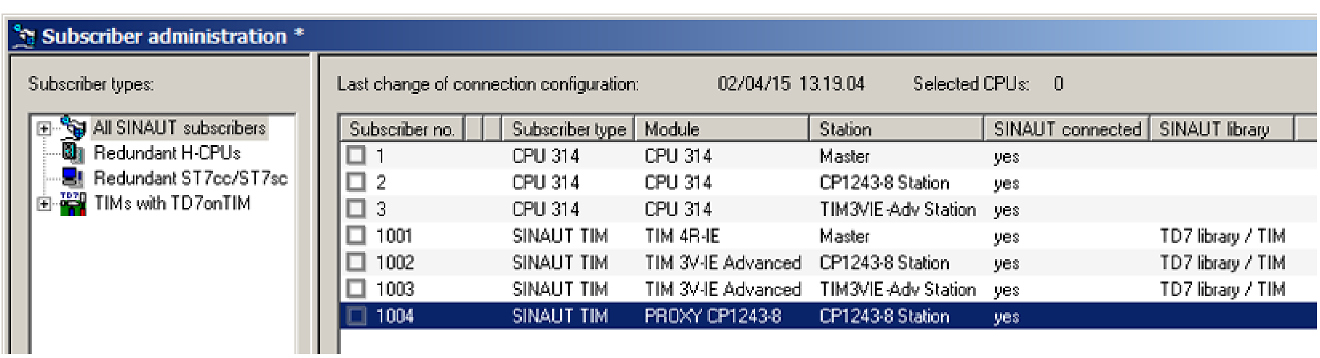


Figure C-2 View of a proxy in the SINAUT subscriber administration

If you double-click on the selected subscriber (proxy) in the subscriber list, you will find the corresponding entry for the expanded type in the properties dialog of the subscriber:

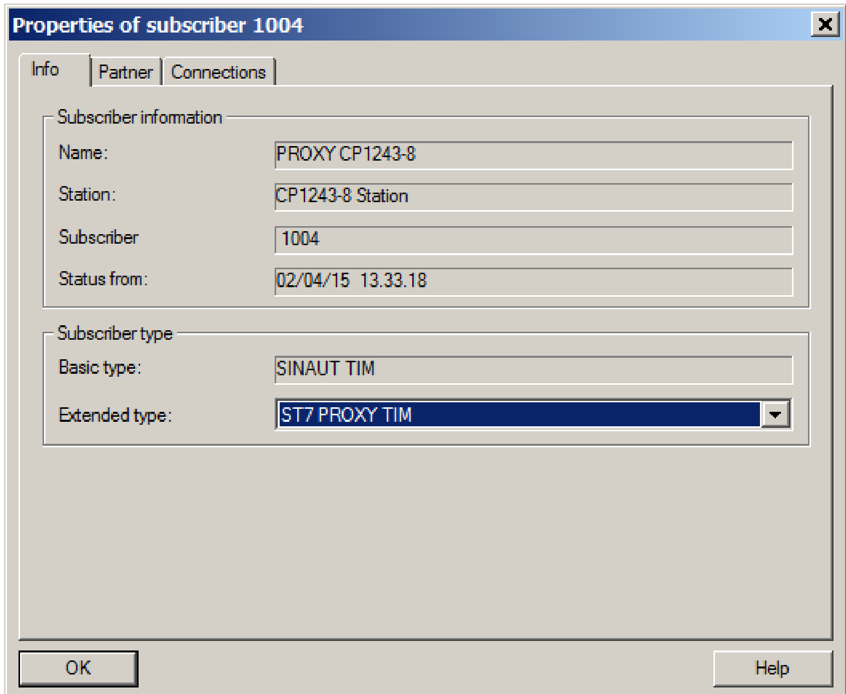


Figure C-3 Expanded type of the proxy in the Properties dialog of the subscriber

Note that neither TD7onCPU nor TD7onTIM is configured for the proxy.

C.5 Exporting configuration data

Exporting the configuration data using SDB text files

After completing the configuration of the proxy in STEP 7 V5 and in the SINAUT configuration tool, the specific configuration data for the telecontrol communication of the proxy is stored in system data blocks (SDBs) just as with TIM modules.

Follow the steps below to export the configuration data of the proxy:

1. Open the SINAUT diagnostics and service tool with the relevant project.
2. Select the proxy.

3. Open the menu "SINAUT" > "SDB display".

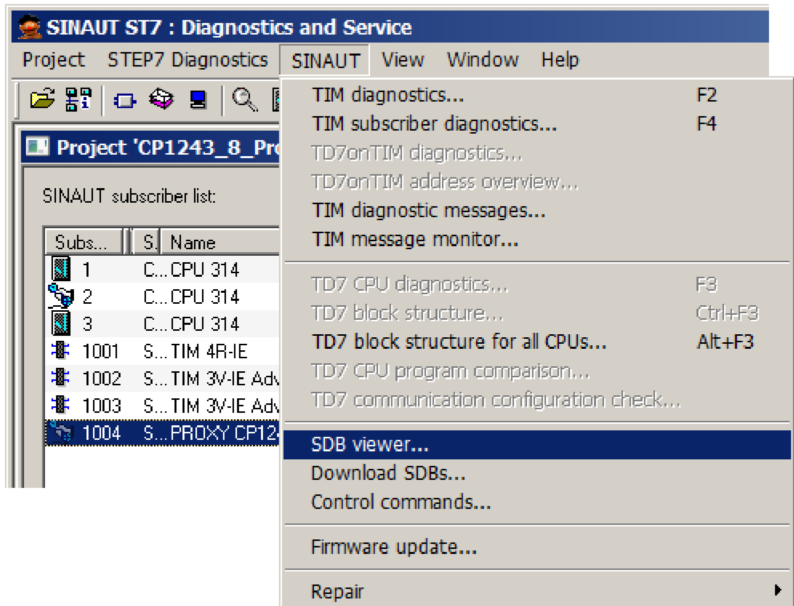


Figure C-4 Opening the "SDB display" dialog

The "SDB display" dialog opens.

With the drop-down list box "System data blocks" you can display the contents of the individual SDBs. This is however not relevant for exporting the configuration data.

4. Click the "Save" button.

   The "Save as" dialog opens.

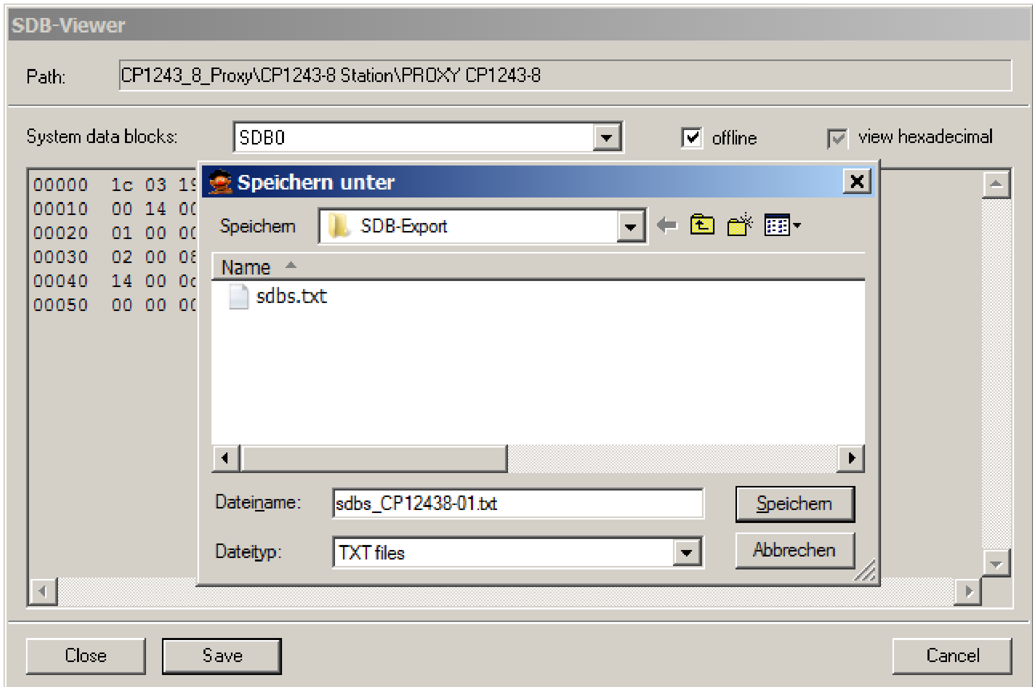


Figure C-5 Export of the configuration data of a proxy in the example as the file "sdbs_CP12438-01.txt".

5. In the file directory of the configuration PC/PG, select a suitable directory for the file of the configuration data.
6. Select a unique name for the file of the configuration data of this proxy.

   Retain the default file type TXT file.
7. Click "Save".

   A text file with the data of all SINAUT SDBs is saved in the file directory.

Note

TXT file for STEP 7 in the TIA Portal.

You then need the exported text file to import it into the respective successor module in STEP 7 Basic / Professional.

Documentation references

D

Where to find Siemens documentation

- Article numbers

  You will find the article numbers for the Siemens products of relevance here in the following catalogs:

  – SIMATIC NET - Industrial Communication / Industrial Identification, catalog IK PI

  – SIMATIC - Products for Totally Integrated Automation and Micro Automation, catalog ST 70

  You can request the catalogs and additional information from your Siemens representative. You will also find the product information in the Siemens Industry Mall at the following address:

  Link: (https://mall.industry.siemens.com)

- Manuals on the Internet

  You will find SIMATIC NET manuals on the Internet pages of Siemens Industry Online Support:

  Link: (https://support.industry.siemens.com/cs/ww/en/ps/15247/man)

  Go to the required product in the product tree and make the following settings:

  Entry type “Manuals”

- Manuals on the data medium

  You will find manuals of SIMATIC NET products on the data medium that ships with many of the SIMATIC NET products.

/1/

SIMATIC NET
TIM 1531 IRC
Operating instructions
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/24710/man)

/2/

SIMATIC
S7-1500/ET 200MP Manual Collection
Reference work
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/view/86140384)

/3/

SIMATIC NET
SCALANCE M812, M816
Operating Instructions
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/15984)

/4/

SIMATIC NET
Mobilfunkrouter SCALANCE M870
(M873 / M874 / M875 / M876)
Operating Instructions
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/15987/man)

/5/

SIMATIC NET
MODEM MD720
Operating Instructions
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/15923/man)

/6/

SIMATIC NET
MODEM MD2
Operating Instructions
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/view/17163799)

/7/

SIMATIC NET
MODEM MD3
Operating Instructions
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/view/17164329)

/8/

SIMATIC NET
MODEM MD4
Operating Instructions
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/view/17165032)

/9/

SIMATIC NET
Industrial Ethernet Security
Security basics and applications
Configuration manual
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/15326/man)

/10/

SIMATIC NET
Industrial Ethernet Security
SCALANCE S
Commissioning and Installation Manual
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/15327/man)

/11/

SIMATIC NET
Diagnostics and configuration with SNMP
Diagnostics manual
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/15392/man)

/12/

SIMATIC NET
CP 1243-8
Operating Instructions
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/21162/man)

/13/

SIMATIC NET
SINAUT ST7
System Manual
- Volume 1: Station control system
- Band 2: Software (STEP 7 V5)
- Volume 3: Software (STEP 7 Professional)
Siemens AG
Link: (https://support.industry.siemens.com/cs/ww/en/ps/21771/man)

/13/

Glossary

1-out-of-8 check

Mechanism for interlocking multiple simultaneous commands.

When entering commands, there is a check to determine if only one command is pending at the time of acquisition. Transmission of the command byte is only triggered if there is a single modified command bit in the command byte compared to the last cycle. If several bits within the command byte have been changed, errors are detected and the command byte is not sent.

The function is performed by the data point typical "Cmd01B_S" of the TD7onCPU block library. The "FC Safe" block is also required.

1-out-of-n check

Mechanism for interlocking multiple simultaneous commands.

When entering commands, there is a check to determine if only one command is pending at the time of acquisition. Transmission of the command to the communication partner is only triggered when a single command is pending. If several commands are pending at the same time, errors are detected and the command is not sent or not issued by the receiver.

The function is executed by the "FC Safe" block of the TD7onCPU block library. The function is supported by the data point typicals "Cmd01B_S", "Par12D_S" and "Set01W_S".

APN

Access Point Name

DNS host name of the access point for an external network (in this case: access point in the GPRS network to the Internet).

Conditional spontaneous frame

→ *Spontaneous / conditional spontaneous / unconditional spontaneous frame*

CP

Communications processor

Module for expanded communications tasks that provides the CPU with additional interface types or communications options.

CPU

Central Processing Unit

Main processor of a SIMATIC controller

CSD

Circuit Switched Data

Service for transferring data in the GSM network. Possible are dial-in connections of GSM modems to GSM/ISDN/analog modems and other devices with modems. The transmission speed is 14400 bps full duplex for non-secure transmission and 9600 bps for secure transmission.

CTS

Clear to send

Signal in the data flow control

Data frame

Data unit transferred between communication partners. Meaning:

- Data unit transferred on the application layer (OSI layer 7)
- General term for a transferred data unit regardless of the relevant OSI layer.

Data frames contain the data of an ST7 object to be transmitted. Depending on the object type, a message can contain either all data of an ST7 object or a contiguous subarea of the object data.

See also "Organizational frame"

Direct communication

With direct communication, the S7 stations communicate directly with each other without the frames needing to be forwarded by a master station or station.

See also "Inter-station communication"

Context: Telecontrol / SINAUT

DNP

Distributed Network Protocol

DNP3 TIM

→ *TIM*

DSL

Digital Subscriber Line

Standards for transmission of telephone and Internet data with transmission speeds up to 1000 Mbps.

EGPRS

Enhanced GPRS

Packet-oriented service for IP-based data transmission in GSM networks. By using an additional modulation procedure (EDGE technology), a higher transmission speed is achieved compared with GPRS.

Engineering station

PC with the STEP 7 Professional project (TIA Portal)

Ethernet TIM

→ *TIM*

Frame

→ *Data frame*

General request

With a general request (GR), a central station requests the current process image from the connected nodes.

A general request is started automatically when a disrupted connection has been restored or when a failed partner reports a restart. Apart from the automatic general request, a general request can also be triggered by the user program or from the control center.

TD7onTIM does not support general requests.

GPRS

General Packet Radio Service

Packet-oriented service for IP-based data transmission in GSM networks. The data is transmitted using the Internet protocols TCP/IP or UDP/IP.

GSM

Global System for Mobile Communication

Worldwide standard for mobile communication (2G)

Image memory

Memory area for the process image in a telecontrol module

Each data frame is saved exactly one time in the image memory. New values of a data point overwrite the existing value in the image memory.

See also send buffer

Context: TeleControl

Image memory / send buffer principle

- **Image memory principle**

  A fixed position is reserved in the image memory for each data message transferred to the TIM for transmission. Each newly transferred message always overwrites the old message in the image memory.

  If a send message is entered using the image memory principle, only a reference to the location of the message in the message image memory is entered. If the TIM has not yet been able to transmit the message when the same message is transferred to it again, the message is not entered in the send buffer a second time, but rather the image is simply updated.

  At the time of transmission, the message is sent with its up-to-date content from the image memory. Only then can the message be entered in the send buffer again.

  Transmission using the image memory principle achieves the following:

  – There is less load on transmission link, fewer messages are transmitted.

  – There is less load on the send buffer of the TIM; an image memory message is entered a maximum of once in the send buffer.

- **Send buffer principle**

  If a data frame is transmitted using the send buffer principle, it is entered completely in the send buffer each time it is transferred to the TIM. If such a message cannot or should not be transmitted immediately, it may exist more than once in the send buffer.

  When it is sent, the message is taken completely from the send buffer and transmitted.

Inter-station communication

Communication between two stations, which is mediated by a Telecontrol/SINAUT master station.

For dial-up networks, a direct connection between the two stations is established in ST7 networks.

See also "Direct communication".

IRC

Industrial Remote Communication

SIMATIC NET product group for telecontrol.

ISDN

Integrated Services Digital Network

Standard for a digital transmission network for telephone, telefax, telex, teletext and datex-J/L/P services. The data of various services can be transferred simultaneously. Telephone connections normally operate at transmission speeds of 56 to 64 Kbps.

LAN

Local Area Network

Local network, usually "Industrial Ethernet".

Local CPU

CPU which is assigned to a TIM module.

Local TIM

A TIM connected to a PC (ST7cc, ST7sc) or an S7 CPU over the MPI bus, Industrial Ethernet or an IP-based network.

LTOP

Line Transformer with Overvoltage Protection

Overvoltage protection module of the SINAUT device program - discontinued

Main cycle / sub-cycle

The sequence of the polling cycle can be structured on the master TIM by assigning individual polling stations to the main cycle or the subcycle.

The subcycle is always activated at the end of the main cycle once all stations from the main cycle have been polled. A configurable number of stations is called in a subcycle.

Following this, all the stations in the main cycle are polled again. This is followed by a subcycle in which further stations that are assigned to the subcycle are called.

Master station

Station in the top hierarchy of a telecontrol network with ST7 protocol. It is connected to the control system and the substations or node stations.

The WAN interfaces of the TIM are set to the network node type "Master station".

MCC - Mobile Country Code

→ *PLMN*

Messages

Emails and SMS in the TeleControl context

See also Data frame.

MNC - Mobile Network Code

→ *PLMN*

MPI

Multi Point Interface

MPI is the programming device interface of the SIMATIC S7-300/400. Devices such as the TIM can communicate with each other via the MPI interface.

See also Partyline.

MSC

The MSC transmission protocol is a proprietary protocol on OSI layer 3 for the secure communication via Ethernet, landline or mobile wireless networks in SINAUT ST7. The MSC protocol provides an authentication mechanism and simple encryption of data.

The protocol is available in the MSC and MSCsec versions (with cyclic key exchange).

MSCsec

→ *MSC*

Multi-master polling with time slots

When stations need to communicate with more than one master station in dedicated line or wireless operation, the multimaster polling with time slots mode is used. Each of the connected master stations is assigned one or more defined time slots per minute for polling the stations. The master stations then have their turn to poll in every minute.

Node station

A node station is a station located between the master station and stations in the hierarchy of a telecontrol network. One or more subordinate stations are connected to a node station. The data traffic between these stations and the master station is handled via the node station. Direct data exchange between the node station and the subordinate stations is also possible. Multiple node station levels are possible in a SINAUT network.

Organizational frame

Organizational frames are used to execute organizational system functions, for example:

- General requests
- Time-of-day synchronization
- Counted value storage
- Coordinated connection establishment and termination in a dial-up network
- Message indicating station startup and station failure
- Requests for and transmission of subscriber records

Party line

Setup of the communications bus of the smaller S7-300 CPUs. The communications bus id physically wired through to the MPI interface.

Party line CPUs are: CPU 312.. / 313.. / 314.. / 315.. to CPU 315-2 DP and C7 devices.

With party line CPUs you can use every type of TIM. You will find details on the Internet at the following address:
https://support.industry.siemens.com/cs/ww/en/view/24059469

With non party line CPUs (CPU 315-2 PN/DP to CPU 319-3 PN/DP) the MPI interface and communications bus are separate.

Permanent call

A permanent call does not interrupt the normal polling cycle; it is always executed alternating with the standard poll from the normal polling cycle.

PG

Programming device

Allows access by the STEP 7 configuration software to the SIMATIC CPU.

PG routing

Using PG routing, it is possible to access programmable modules or modules with diagnostics capability beyond network boundaries from a programming device (PG) or computer (PC).

PLMN

Public Land Mobile Network

Worldwide unique identifier of mobile networks. The PLMN is made up of the three-digit Mobile Country Code (MCC) and the two-or three-digit Mobile Network Code (MNC) of the network provider.

Polling

→ *Polling mode*

Polling mode

The polling mode is a method of data transmission in which a central instance controls the data exchange with the communication partners.

Using a polling message, the master TIM instructs the connected station TIMs one after the other to transmit their stored data frames to the master TIM. If a polled station has no stored data, it responds with an acknowledgment message and the polling cycle then continues by polling the next station.

A station that has stored data sends a single data frame or, if block transfer was configured, several data frames in a block.

If the TIM has stored additional data, it indicates this in the response frame. In this case, the station is then immediately called up again until the stored data has been transferred.

Polling with time slots

The polling with time slots mode is used in a wireless network in which the use of the radio frequency assigned by the registration authorities must be shared with other users. Each user typically has 6 seconds per minute to exchange data with its stations. The frequency must then be released for other operators. During the allocated time slot, this pooling variant functions like a normal polling system.

Context: SINAUT ST7

Protocol

A protocol is a set of rules for controlled transfer of data. Protocols, for example, specify the data structure, the structure of data packets and the coding. Protocols can also specify a control mechanisms and hardware and software requirements.

Requested message

Polled data frames are data frames of a station or node TIM with a special identifier indicating that they were sent in response to a general request from the master station.

RS-232

RS-232 is a standard for serial (bit-by bit) data transmission with +12 V and -12 V signals. RS-232 is a Recommended Standard of the Electronic Industries Association. 9-pin and 25-pin connections with D-sub connectors (subminiature connector with D-shaped surface area) are normal for the RS-232 interface.

RS-485

RS-485 is a standard for data transmission with 5 V differential signals. The RS-485 interface uses only one pair of wires and is operated in half duplex. The connection is multipoint-compliant; in other words, up to 32 subscribers can be connected.

RTS

Request to send

Signal in the data flow control

S0 interface

Basic interface of ISDN for connecting end devices

Send buffer principle

→ *Image memory / send buffer principle*

SIM card

SIM - Subscriber Identity Module

The SIM card is an identification card for a subscriber of a mobile wireless service.

Simple Internet communication

In SINAUT ST7, simple Internet communication means data exchange between TCP/IP-compliant devices in Ethernet, landline or mobile networks using the MSC protocol.

SINAUT

Siemens Network Automation

Station control system or telecontrol system based on SIMATIC S7.

SINAUT ST7 works with the SINAUT ST7 telecontrol protocol.

SINAUT object

A SINAUT object contains the data of one or more process variables such as analog values, commands, calculated values, status information on motors, sliders etc. An ST7 object has type-specific processing functions and change checks assigned to it to minimize the communication traffic in the WAN. Type-specific processing functions include, for example, threshold checks or mean value calculation with the object type for analog values. The change check is designed so that a message is generated only when the object data has changed compared with the last time its value was transferred or when the type-specific processing enables generation of a message because the object data is "worth" transferring.

SINAUT ST7

Proprietary telecontrol protocol for SIMATIC NET telecontrol modules

SINAUT ST7cc

Control center system based on SIMATIC WinCC for SINAUT ST7.

SINAUT ST7sc

System for networking SIMATIC stations with a control station via WAN. The control center can also be a SIMATIC station or a PC-based control center, for example, WinCC with the SINAUT ST7cc add-on.

SINAUT TD7 Library

Software for control of ST7 communication of telecontrol modules. The TD7 software in the stations allows change-controlled transmission of process data between the individual CPUs and the control center, for example ST7cc. Failure of connections, CPUs, or the control center are displayed. Once a problem has been corrected or the CPUs or control center have started up, data is updated automatically. Data frames can be given a time stamp, if required.

The following variants of the TD7 software exist:

- **TD7onCPU**

  Program blocks in the CPU user program

  The SINAUT TD7 library consists of program blocks for the CPU. They are available in the following versions:

  – Library for STEP 7 V5

    The blocks are executable on S7-300- and S7-400-CPUs (except CPU 400H). There are only a few blocks intended specifically for the S7-300 or S7-400.

  – Libraries for STEP 7 Professional

    There is a global library with two versions for STEP 7 projects in the TIA Portal:

    - Blocks for S7-1500

    - Blocks for S7-300 and S7-400

  If the DNP3 protocol is used, TD7onCPU is not supported.

- **TD7onTIM**

  Configurable part of the firmware of the communication module

  TD7onTIM can be used as an alternative to TD7onCPU for an Ethernet TIM. TD7onTIM runs on the TIM. It is configured as follows:

  - STEP 7 V5: In the SINAUT engineering software

  - STEP 7 Professional: Via data points of the TIM

TD7onCPU and TD7onTIM cannot be used simultaneously in a station.

SMS

Short Message Service

The short message service in the GSM standard is used to transfer short text messages to mobile phone users.

When the short messages are transferred, they are first transferred to the SMS center (SMSC) using a store-and-forward technique. They are buffered there and then forwarded to the recipient. The sender can query the status of the message in the SMS center or can request acknowledgment of delivery.

SMSC

Short Message Service Center

When sending an SMS message, the message is first sent to the SMSC, buffered there and then forwarded to the recipient.

Spontaneous / conditional spontaneous / unconditional spontaneous frame

- **Spontaneous frame**

  In SINAUT networks, data frames are always transmitted spontaneously; in other words, data are created and transmitted only when changes to process values occur or event-controlled. These messages are known as spontaneous messages.

- **Conditional spontaneous frame**

  In the dial-up network, you can specify whether or not a change causes a "conditional spontaneous" or "unconditional spontaneous" transmission for each individual message.

  Conditional spontaneous messages are initially only entered in the send buffer of the TIM. They are only transmitted when a connection is established to the partner for whatever reason, for example because an unconditional spontaneous message needs to be transmitted or because the partner calls.

  Even when using pay by volume transmission in a GPRS network, message prioritization "conditional spontaneous" can also be used. Such a message is not transmitted immediately, but is first buffered. In a GPRS network, the TIM stores "conditional spontaneous" messages in the following situations:

  – When the collected frames reach or exceed a size of 202 bytes.

  – An important frame ("unconditional spontaneous" or "spontaneous" priority) should be transmitted immediately.

  – The collected frames have not yet reached a volume of 202 bytes, but the TCP/IP keep-alive interval expires.

  – The fill level of the send buffer has reached 90% of its maximum capacity.

- **Unconditional spontaneous frame**

  In the dial-up network, you can specify whether or not a change causes an "conditional spontaneous" or "unconditional spontaneous" transmission for each individual message.

  Unconditional spontaneous messages cause the connection to be established immediately. Even with pay-by-volume/time transmission in a GPRS network, you can use the message prioritization "unconditional spontaneous"; in other words, in contrast to a "conditional spontaneous" message, a message is transmitted immediately.

Spontaneous mode

The spontaneous mode is a method of SINAUT data transmission in which subscribers can exchange data among themselves directly. Here, no central entity is necessary in the form of a master TIM as in polling mode (see "Polling mode"). The spontaneous mode is intended for data transmission in dial-up networks and for communication via IP-based networks.

For transmission in a dial-up network and in pay-by-volume/time IP-based networks (for example GPRS), the data to be sent is assigned different priorities during parameter assignment ("high" or "normal", with TD7onTIM also "alarm"). When data with high or alarm priority are ready for transmission, the dial-up connection is established immediately. If the data has "normal" priority, it is first stored on the TIM. This data is then sent the next time a connection is established to the partner for whatever reason. This can, for example, be the situation when information with higher priority needs to be sent or when the partner establishes a connection to exchange data.

ST7 protocol

→ *SINAUT ST7*

ST7cc

→ *SINAUT ST7cc*

ST7sc

→ *SINAUT ST7sc*

ST7-TIM:

→ *TIM*

Station

- Hardware

  SIMATIC controller with the required components for acquisition, processing and communication, consisting of: CPU, I/O modules, communication module, modem, etc.
- Network node type

  Setting a WAN interface of the TIM. An interface of the "Station" network node type communicates at the lowest hierarchy level in a SINAUT network.

Subcycle

→ *Main cycle / sub-cycle*

TD7 software

→ *SINAUT TD7 Library*

TIM

Telecontrol Interface Module

Communication module that handles all data transmission functions provided by the SINAUT system independently.

- Ethernet TIM

  Compared to a classic TIM, an Ethernet TIM (ST7-TIM) has additional LAN interfaces for IP-based communication.
- ST7-TIM:

  An ST7-TIM uses the SINAUT ST7 telecontrol protocol.
  - The TIM 3V-IE versions have an RS-232 and an Ethernet interface.
  - The TIM 4R-IE has two RS-232 / RS-485 and two Ethernet interfaces.
  - The TIM 1531 IRC has two RS-232 / RS-485 and two Ethernet interfaces.
- DNP3 TIM

  A DNP3 TIM uses the DNP3 telecontrol protocol.
  - TIM 3V-IE DNP3 has an RS-232 interface and an Ethernet interface.
  - TIM 4R-IE DNP3 has four WAN connections: Two RS-232/RS-485 interfaces and two Ethernet interfaces

Unconditional spontaneous frame

→ *Spontaneous / conditional spontaneous / unconditional spontaneous frame*

VPN

Virtual Private Network

Technology for secure transportation of confidential data in public IP networks, for example the Internet.

WAN

Wide Area Network

Data network with a large geographical span, such as the Internet. telephone or enterprise networks. The following WANs can be used with SINAUT:

- **WAN, classic**

  Includes SINAUT communication via dedicated lines (private or leased), private wireless networks, analog telephone network, digital ISDN network and mobile networks (without Internet).

  A classic WAN is connected via suitable transmission device (modem) to a serial interface of the TIM.
- **WAN, IP-based**

  Includes IP-based telecontrol communication via wireless, fiber-optic cables, public networks and the Internet using services such as DSL, GPRS or UMTS or via broadband systems such as OTN and PCM30.

  An IP-based WAN is normally connected to an RJ-45 interface of an Ethernet TIM via an Ethernet-compliant module.

Index