SIEMENS

SIMATIC NET

Industrial Remote Communication - TeleControl
SINAUT ST7 software

System Manual

Volume 2: Software

09/2016

C79000-G8976-C222-10

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-C222-10
Ⓟ 10/2016 Subject to change

Preface

Note

Discontinuation of modules

As of 01.10.2016 the following products have the status "discontinued":

- Modem MD2
- Modem MD3
- LTOP1 / LTOP2
- Accessories:

  Overvoltage protection module as replacement part for LTOP1 and LTOP2, DIN rail adapter, various plug-in cables

  Observe the product notifications on the Internet:
  Link: (https://support.industry.siemens.com/cs/ww/en/ps/15915/pm)
- EGPRS router MD741-1

  Observe the following product notification:
  Link: (https://support.industry.siemens.com/cs/ww/en/view/62607452)

If there are successors to the discontinued devices, you will find these in the notifications listed above on the Internet.

What's new in SINAUT ST7?

SINAUT ST7 provides the following new functions:

- **Engineering software**

  New version of the SINAUT ST7 configuration and diagnostics software V5.5 SP2

  After release of the version V5.4 of the engineering software small functional improvements were made. In addition to this, the following functions are new:

  – Configuration of a proxy TIM 3V-IE Advanced as substitute for a CP 1243-8 IRC

    The 1243-8 IRC can be configured in STEP 7 Basic as of version V13.0 SP1.

    You will find the description in Volume 2 in section 3.2, PROXY CP1243-8 IRC (Page 48).

  – Improvement of the selective connection configuration by selecting individual subscribers

  – Passing of the key exchange interval to the MODEM MD720 when using the MSC protocol

  – Errors corrected in the time-of-day synchronization of a TIM by the CPU

    You will find the description in Volume 2 in section 3.5, Synchronization of an Ethernet TIM by the CPU (Page 120).

To upgrade to this version, a full version of the SINAUT engineering software V5.0 must be installed.

The engineering software V5.5 SP2 can be used with STEP 7 as of version V5.4 SP4.

To use the engineering software, a Windows operating system is required that is compatible with the installed STEP 7 version.

You will find details on the ordering data and operating systems of the engineering software in volume 2 of the system manual in the section SINAUT ST7 software: Ordering data and compatible operating systems (Page 17).

- **Block library TD7onCPU**

  New version of the block library TD7onCPU V2.2 SP4 + Hotfix 1

  – New block "FC-PathStatus" to display the main and substitute path to the remote communications partner.

    The block can be used as of version V5.5 SP1 of the engineering software and as of TIM firmware V2.5.4.

  – The block library can now be used both in a standalone TIM in an S7-400 and an S7-400H also with only one single CPU (single mode).

- **TIM firmware**

  The new functions of the SINAUT engineering software named above are supported by firmware version V2.5.4 for the following TIM modules:
  TIM 3V-IE, TIM 3V-IE Advanced, TIM 4R-IE

Replaced documentation

This manual replaces the manual release 08/2011 (C79000-G89xx-C222-07).

For older releases of the manual, see below, section Version history (Page 7).

Purpose of the manual

The SINAUT ST7 system manual is split into two complementary volumes.

- **Volume 1: System & Hardware**

  This documentation will support you on your way to successful application of SINAUT ST7. This introduces you to the topic in clear and straightforward steps and provides you with an overview of the hardware components of the SINAUT ST7 station control system.

  You will be supported during the planning of network structures and topologies and will see how to install and commission SINAUT components based on the installation guidelines.

- **Volume 2: Software**

  This documentation provides you with an overview of the software components of the SINAUT ST7 station control system. You will see how individual components are configured. Diagnostics and service options are also explained.

Note

Documentation for the SINAUT ST1 system and older modules

This release of the manual "SINAUT ST7" (volume 1 + 2) no longer has detailed information on the SINAUT ST1 system and the following older modules:

- All previous TIM 3 modules: TIM 3V, TIM 32, TIM 33, TIM 34
- TIM 4V, TIM 4VD, TIM 42, TIM 42D, TIM 43, TIM 43D, TIM 44, TIM 44D

If you require information on these modules or on SINAUT ST1, refer to release 05/2007 of this manual. You will find release 05/2007 on the Internet pages of Siemens Industry Online Support at the following address:

Link: (https://support.industry.siemens.com/cs/ww/en/ps/15931/man)

The documents have the following IDs.

- Volume 1: 24621696
- Volume 2: 24619519

Validity of this manual

This manual applies to the following ST7 modules and software versions:

- TIM 3V-IE, TIM 3V-IE Advanced, TIM 4R-IE, TIM 4R / 4RD
- SINAUT ST7 configuration and diagnostics software for the PG V5.4
- SINAUT TD7 library for the CPU V2.2 SP2
- SINAUT TIM firmware V4.4.0 for the TIM 4
- SINAUT TIM firmware V2.5 for the TIM 3V-IE variants
- SINAUT TIM firmware V2.5 for the TIM 4R-IE

Further information on the Internet

You will find further information on the TeleControl ST7 products on the Internet at the following address:

Link: (https://support.industry.siemens.com/cs/ww/en/ps/15931/man)

There select the required information under "Entry type" (for example "Updates", "Manuals", "FAQs" etc.).

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 only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

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

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's 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).

License conditions

Note

Open source software

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

You will find license conditions in the following documents on the supplied data medium:

- OSS_TIM-ST7_86.pdf
- OSS_SINAUT-ES_86.pdf

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 here:

- SIMATIC NET Manual Collection or product DVD

  The DVD ships with certain SIMATIC NET products.
- On the Internet under the following address:

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

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)

Version history

The previous versions of the manual described the innovations and versions listed below.

Release 09/2014 of the manual (C79000-G89xx-C222-09)

New functions:

- New version of the SINAUT ST7 configuration and diagnostics software V5.4
  - Time-of-day synchronization of the TIM 4R-IE using NTP

    You will find the description in Volume 2 in section 3 (Configuration in STEP 7 > Configuration of TIM modules > "NTP" tab).
  - Synchronization of the TIM time of day by the CPU

    You will find the description in Volume 2 in the section 3 (Configuration in STEP 7 > Configuration of the time-of-day synchronization).
  - MSCsec protocol: Secure transfer, authentication with key exchange

    You will find the description in Volume 2 in section 2 (Configuration - Overview > GPRS/Internet Communication).
- New firmware version V2.5 for the TIM modules TIM 3V-IE, TIM 3V-IE Advanced, TIM 4R-IE

  The functions named above among the innovations of the configuration software are new.

Validity of the manual:

- TIM 3V-IE, TIM 3V-IE Advanced, TIM 4R-IE, TIM 4R / 4RD
- SINAUT ST7 configuration and diagnostics software for the PG V5.4
- SINAUT TD7 library for the CPU V2.2 SP2
- SINAUT TIM firmware V4.4.0 for the TIM 4
- SINAUT TIM firmware V2.5 for the TIM 3V-IE variants
- SINAUT TIM firmware V2.5 for the TIM 4R-IE

Release 08/2011 of the manual (C79000-G89xx-C222-08)

New functions:

- New version of "SINAUT ST7 configuration and diagnostics software" V5.2
- Version "SINAUT ST7 configuration and diagnostics software" V5.1
- New SINAUT TD7 library V2.2 SP2 for the CPU

Validity of the manual:

- SINAUT ST7 configuration and diagnostics software for the PG V5.2
- SINAUT TD7 library for the CPU V2.2 SP2
- SINAUT TIM firmware V4.4.0 for the TIM 4

- SINAUT TIM firmware V2.3 for the TIM 3V-IE variants
- SINAUT TIM firmware V2.3 for the TIM 4R-IE

Release 07/2009 of the manual (C79000-G89xx-C222-07)

New functions:

- New version "SINAUT ST7 configuration software for the PG/PC" V5.0
  - The Ethernet TIMs can be configured for communication via the MSC protocol. This allows the use of the GPRS/GSM modem SINAUT MD720-3 even in SINAUT in Internet/GPRS networks. An encrypted connection can be established from an Ethernet TIM to the Internet via a DSL modem.
  - The time slot method can now also be configured with the "SINAUT ST7 configuration software for the PG/PC" as of V5.0 for a master TIM without DCF7 receiver, if a TIM 4R-IE with an Ethernet connection to an ST7cc/ST7sc PC is used as the master TIM.

  The configuration software Version V5.0 can be used with STEP 7 as of Version V5.4 Service Pack 4.

  The configuration software version V5.0 is supported by the following operating systems:
  - Windows XP Professional SP2
  - Windows Server 2003 SP2
  - Windows Vista 32 Bit Ultimate and Business (with or without SP1)
- New firmware version V2.0 for all Ethernet TIM modules

  The new firmware supports the MSC protocol.

Validity of the manual:

- SINAUT ST7 configuration software for the PG/PC V5.0
- SINAUT TD7 library for the CPU V2.2
- SINAUT TIM firmware V4.3.9 for the TIM 4
- SINAUT TIM firmware V2.0 for the TIM 3V-IE variants
- SINAUT TIM firmware V2.0 for the TIM 4R-IE

Release 05/2007 of the manual (C79000-G89xx-C222-06)

New functions:

- New product "TIM 4R-IE" for connecting SINAUT via WAN and Ethernet
- New product version "SINAUT ST7 configuration software for the PG/PC" V4.1

Validity of the manual:

- SINAUT ST7 configuration software for the PG/PC V4.1
- SINAUT TD7 library for the CPU V2.2
- SINAUT TIM firmware V4.3.7 for the TIM 3 / TIM 4

- SINAUT TIM firmware V1.2 for the TIM 3V-IE variants
- SINAUT TIM firmware V1.0 for the TIM 4R-IE

Release 10/2006 of the manual (C79000-G89xx-C222-05)

New functions:

- New product versions
  - SINAUT ST7 configuration software for the PG/PC V4.0
  - SINAUT TD7 library for the CPU V2.2 with new blocks for communication via P-bus
- New hardware for GSM and GPRS
  - GPRS modem MD740-1 for secure packet-oriented communication via GSM mobile wireless (GPRS)
  - GSM modem MD720-3 as replacement for the discontinued GSM modem MC45 for establishing dial-up connections via the GSM mobile wireless network; possible as of firmware V1.7.3 of the MD720-3

Validity of the manual:

- SINAUT ST7 configuration software for the PG/PC V4.0
- SINAUT TD7 library for the CPU V2.2
- SINAUT TIM firmware V4.3.7 for the TIM 3 / TIM 4
- SINAUT TIM firmware V1.2 for the TIM 3V-IE variants

Table of contents

1 The SINAUT engineering software

The SINAUT engineering software (SINAUT ST7 standard software package) is supplied on CD-ROM and is made up of two parts:

- SINAUT ST7 / DNP3 configuration and diagnostics software

  It is installed on the PG/PC and serves the following purposes:

  - Configuration of the telecontrol-specific data of the TIM modules as an expansion of the STEP 7 functions
  - Diagnostics of the TIM modules

- SINAUT TD7 Library

  This contains program block for the S7-CPU.

- The CD-ROM also contains the system manual SINAUT ST7 and the manual TIM DNP3 as a PDF (German/English).

The software package can be used for any number of SINAUT projects.

1.1 SINAUT ST7 software: Ordering data and compatible operating systems

Table 1- 1 Overview for ordering SINAUT ST7 software

Article number	Product name / explanation
6NH7997-0CA54-0AA0	SINAUT Engineering Software version V5.4, consisting of: • SINAUT TD7 library with blocks for the CPU • SINAUT ST7 configuration and diagnostics software for the programming device • Electronic manual (German/English)
6NH7997-0CA54-0GA0	Upgrade of SINAUT Engineering Software version V5.0, V5.1, V5.2, V5.3

Information on the Internet

Information on the SINAUT Engineering Software can be found on the Internet pages of Siemens Industry Online Support under the following entry ID:

16627207 (https://support.industry.siemens.com/cs/ww/en/ps/15928)

Information on the upgrade package can be found on the Internet pages of Siemens Industry Online Support under the following entry ID:

87795994 (https://support.industry.siemens.com/cs/ww/en/ps/6NH7997-0CA54-0GA0)

Compatible operating systems:

- MS Windows 7 Professional, Ultimate (32-bit)
- MS Windows 7 Professional, Ultimate (64-bit)
- MS Windows Server 2008 R2 Standard Edition (64-bit)
- MS Windows XP SP3
- MS Windows Server 2003 R2 SP2

1.2 SINAUT ST7 / DNP3 configuration and diagnostics software

The configuration software includes the following components:

- A module manager that expands the *HW Config* STEP 7 tool. It handles the display and parameter assignment of the TIM modules in *HW Config*.
- A WAN manager that expands the *NetPro* STEP 7 tool. This is responsible for the display and parameter assignment of the SINAUT WAN networks SINAUT network nodes in *NetPro*.
- The SINAUT ST 7 configuration tool is used for project-wide functions such as SINAUT connection configuration and SINAUT subscriber management.
- The SINAUT ST7 diagnostics and service tool. In addition to the diagnostic functions familiar from STEP 7. Using the service tool, it is, for example, possible to load new software on a TIM.

The module manager for SINAUT ST7

The SINAUT module manager expands the catalog of the STEP 7 hardware configuration tool HW Config. The *SIMATIC 300* folder was extended by the *SINAUT ST7* folder. This lists all available TIM modules.

You can select the required TIM module from the *SINAUT ST7* folder and install it in the S7 rack.

You can then open a properties dialog and assign the required parameters to the module. This dialog is also available in the *NetPro* network configuration tool.

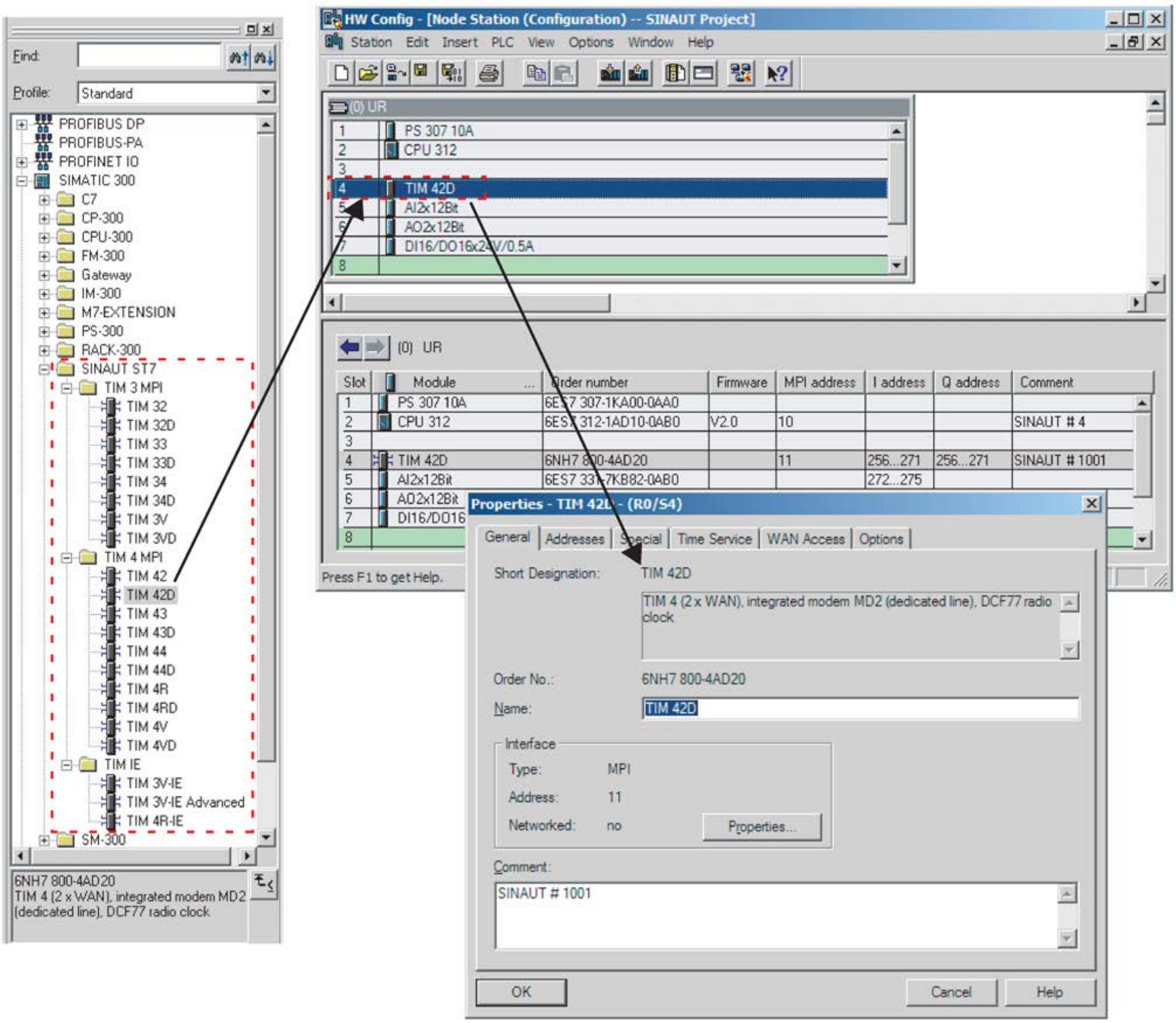


Figure 1-1 Module manager for SINAUT ST7

The WAN manager for SINAUT ST7

The SINAUT WAN manager expands the catalog for the STEP 7 *NetPro* network configuration tool. The SINAUT networks have been added to the *Subnets* directory. You can select the SINAUT networks you require from this directory and install them in the STEP 7 network window.

The TIM modules are interconnected with the networks in the STEP 7 network window using the mouse. Errors in the interconnections are rejected immediately.

A properties dialog can be opened in which you can specify the generally valid parameters for a network, for example, ST7 protocol, transmission speed etc.

In a further dialog, you can specify individual properties for the network nodes, when connecting to a telephone network, for example, the phone number.

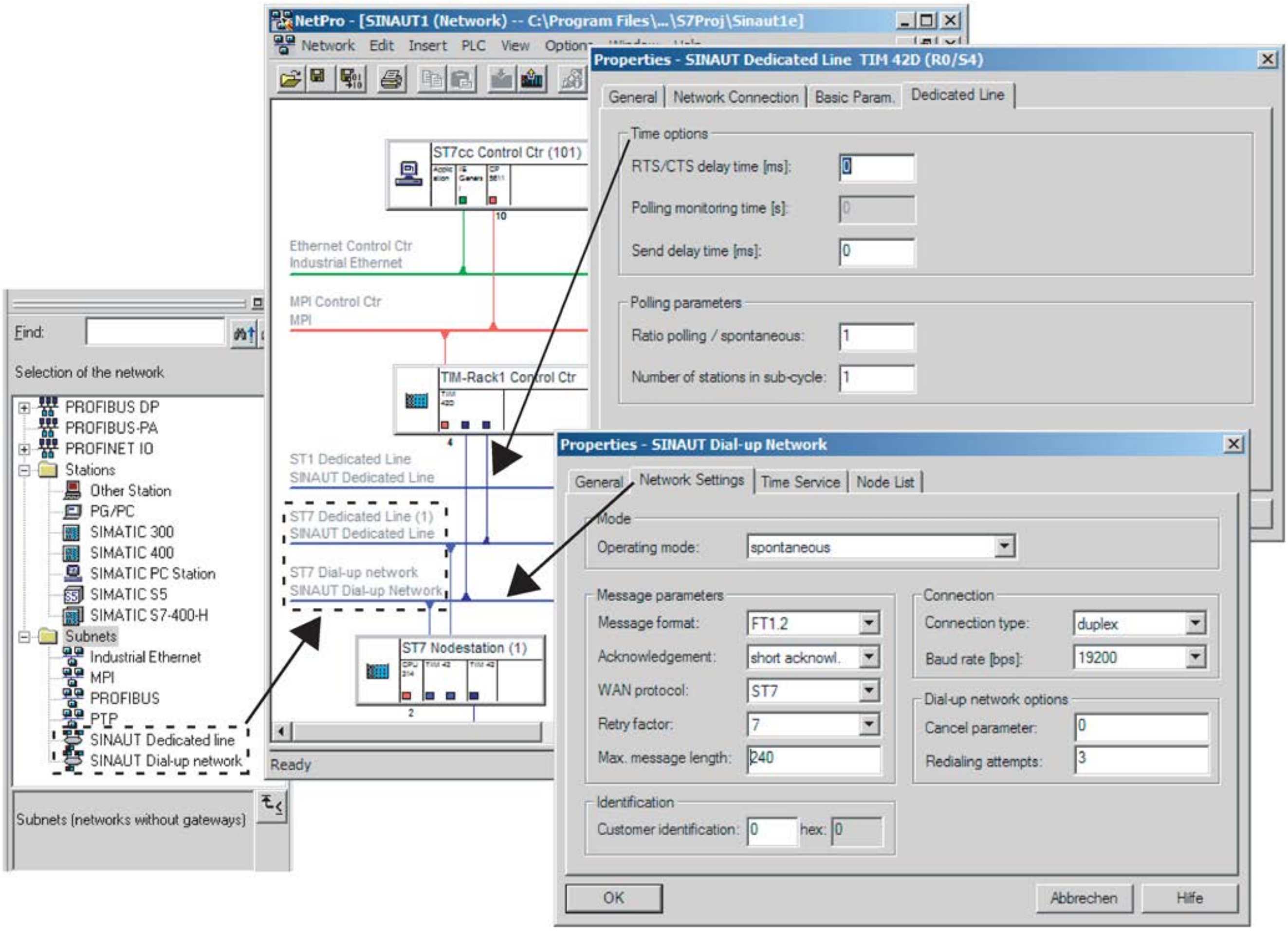


Figure 1-2 The WAN manager for SINAUT ST7

When necessary, the properties dialog of a TIM module can also be opened in *NetPro*. You have the same configuration options as in *HW Config*.

The SINAUT ST7 configuration tool

Unlike the two managers mentioned up to now, the SINAUT ST7 configuration tool does not expand one of the standard STEP 7 tools but is a separate configuration tool embedded in the STEP 7 environment. This allows you to perform the remaining SINAUT-specific configuration tasks. It consists of the following:

- Connection configuration
- Subscriber administration
- SINAUT ST1 - Configuration Overview

The first step is to use the *Connection configuration* tool to specify the SINAUT subscribers (ST7-CPU, ST7cc/ST7sc) between which a connection is necessary. The tool provides you with a selection of all theoretically possible connections in the right-hand pane of the split window. These are determined automatically by the tool based on the network configured with NetPro (see WAN manager). The user then selects the actually required connections and copies them to the right to the left-hand pane using the context menu.

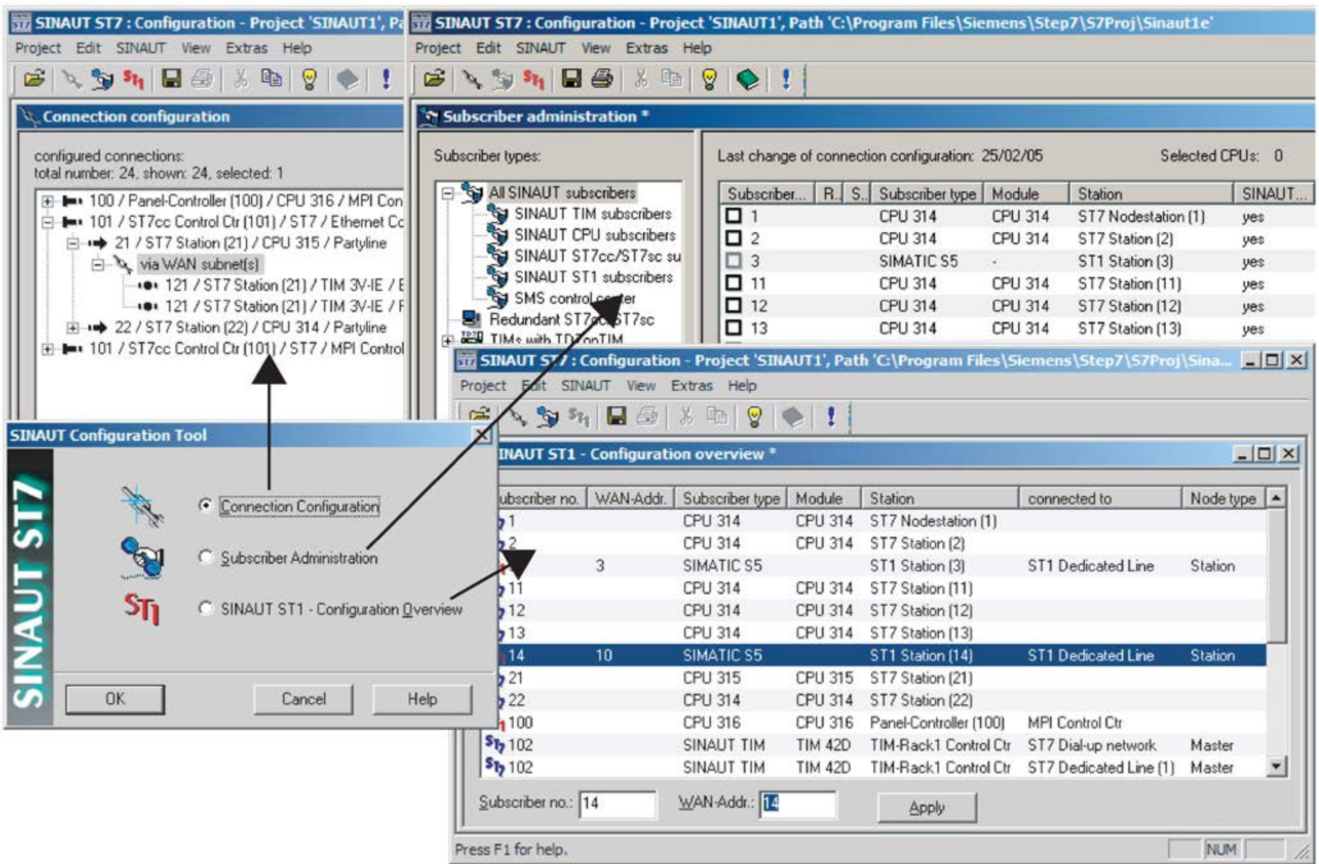


Figure 1-3 The SINAUT ST7 configuration tool

The *Subscriber administration* tool then provides the user with a list of all available SINAUT subscribers. Where necessary, you can make subscriber-specific adaptations, for example, changing the SINAUT subscriber number of the individual devices or the configuration of message texts to be sent as an SMS. Subscriber management also involves the configuration of the data messages to be sent and received if the message generation and evaluation is to be performed by the TIM (only possible for TIMs with TD7onTIM functionality). Based on the configuration data, the tool generates the system data blocks (SDBs) for the CPUs and TIMs. If the SINAUT TD7 software for the CPU is used, the tool also prepares records and communication data blocks for the CPUs and enters them in the block directories of the CPUs along with the blocks (FBs, FCs) required by the individual CPUs for SINAUT communication.

The third tool *SINAUT ST1 - configuration overview* is required only for configuring systems which include SINAUT ST1 devices. This is a convenient utility with which addresses can be synchronized for SINAUT ST1 when necessary.

The following figure illustrates how SMS messages can be configured conveniently in subscriber administration. The configuration results in the automatic generation of data blocks containing texts and the corresponding function blocks that are saved in the block directory on the relevant CPUs.

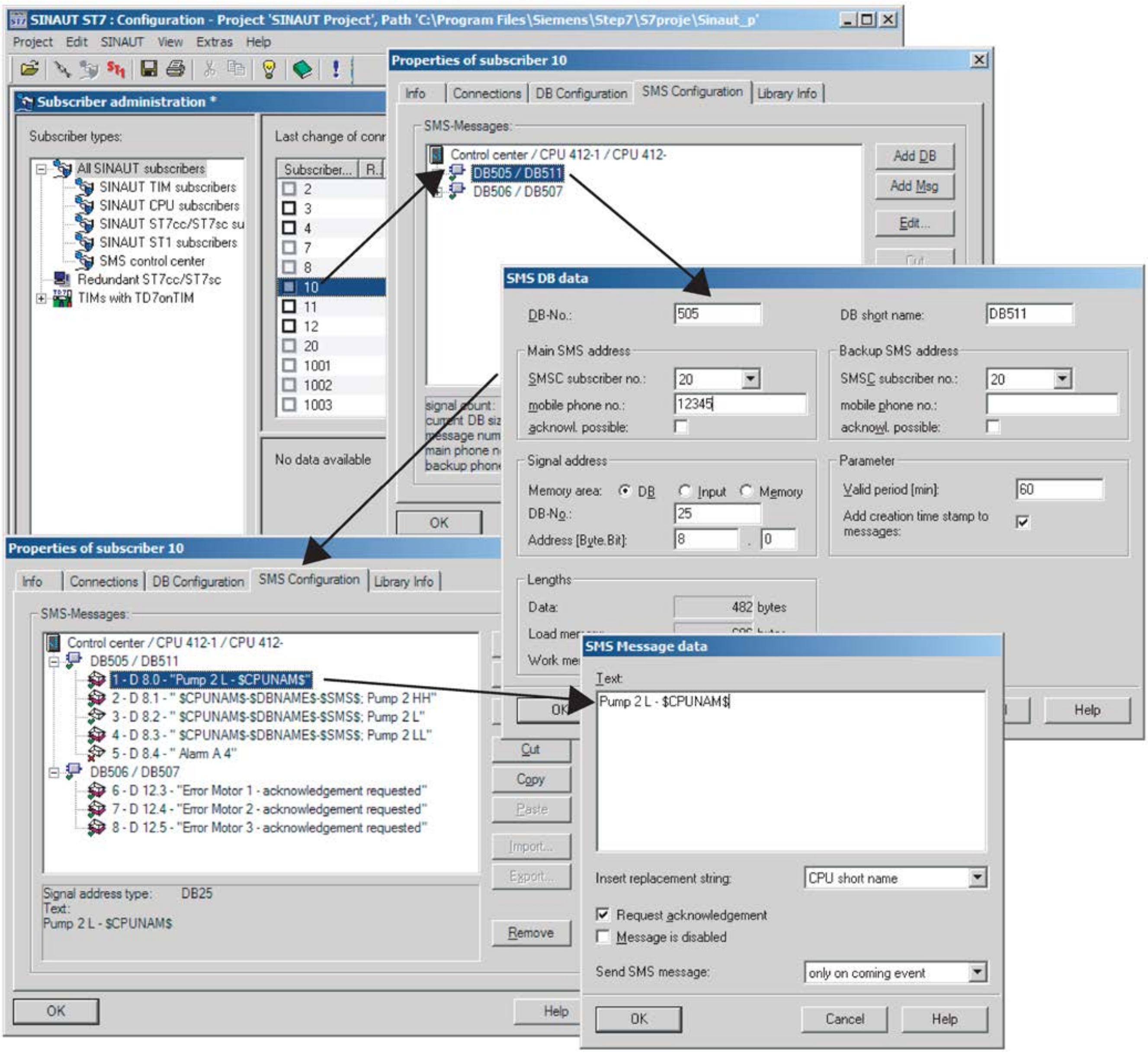


Figure 1-4 Configuring SMS messages

The SINAUT ST7 diagnostics and service tool

In addition to the diagnostic functions familiar from STEP 7, you also have access here to SINAUT-specific diagnostic information. Using the service tool, it is, for example, possible to load new software on a TIM.

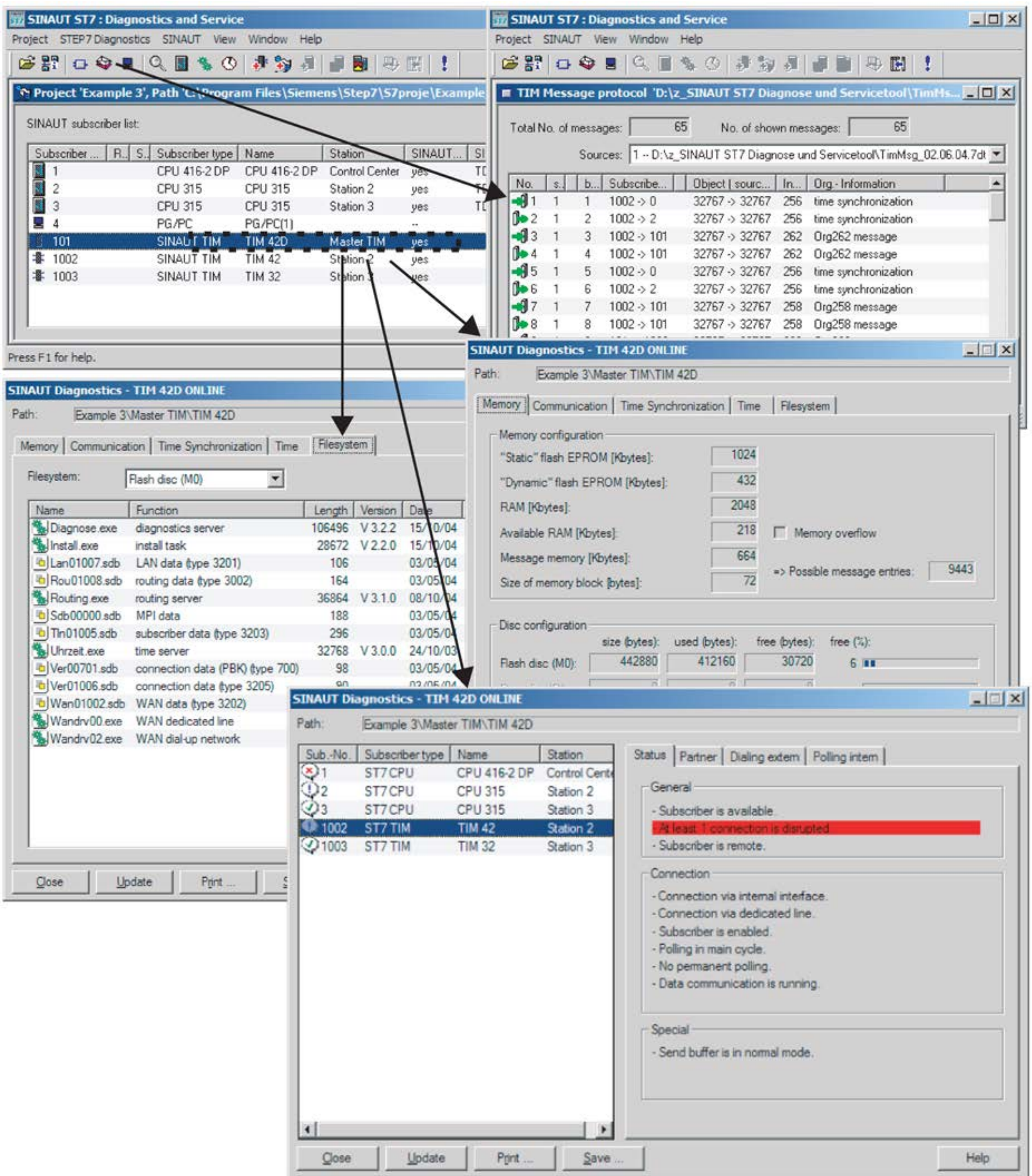


Figure 1-5 Examples of SINAUT ST7 diagnostics

1.3 SINAUT TD7 library (TD7onCPU), blocks for the CPU

The SINAUT TD7 library is a software package with blocks for the CPU. The package was designed so that it can run both on an S7-400 and on and S7-300 CPU. There are only a few blocks intended specifically for the S7-300 or S7-400 CPUs.

The SINAUT TD7 software in the stations allows change-driven 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 has started up, data is updated automatically. When necessary, data messages can be given a time stamp.

Note

Data communication from the CPU to other CPUs or to ST7cc/ ST7sc over a WAN connection can only be implemented with the SINAUT TD7 software. This is not possible with the S7 communication SFBs/SFCs for configured and unconfigured connections. These are suitable only for local communication without a gateway.

Note

With the Ethernet TIMs, you can choose between 2 alternatives:

- As described above, the SINAUT program is created with the TD7 blocks and therefore runs on the CPU and takes up a corresponding amount of work memory (TD7onCPU)
- The SINAUT program runs on the TIM (TD7onTIM). In an ideal situation, no program memory on the CPU is required for SINAUT.

The essential components of the TD7onCPU package are as follows:

- **Basic and auxiliary blocks**
  Most of these blocks are always required on the CPU. A few are purely optional. The basic blocks handle central tasks such as startup, monitoring of connections and connection partners, general requests, time management, handling communication etc. The auxiliary blocks enter messages in the send buffer or fetch them from the receive buffer, handle send and receive jobs for specific connections or provide information as a result of searches.
- **Data point typicals**
  These blocks are included in the CPU program depending on the type and amount of data to be transferred. They put messages together when data changes and output received process data.

To operate correctly, the TD7 package requires some data blocks that are generated by the SINAUT ST7 configuration tool. These are as follows:

- **Central records DB**
  This contains all the centrally required data including the records of all communication partners and the connections to be managed.
- **Communication DBs**
  A separate communication DB is created for each connection. This DB contains a send and receive buffer and all the data required for controlling and monitoring the connection.

The package is completed by blocks for diagnostics and alarms:

- TestCopy
  This block can be used to log which messages are sent and/or received within the CPU.
- SMS_Control
  When certain user-defined events occur, this block puts together appropriate text messages that can be transmitted as SMS messages to a specified mobile phone.

1.4 Installation of the SINAUT ST7 standard software package

The following requirements must be met to install any or all components of the software package:

- A compatible operating system: See section SINAUT ST7 software: Ordering data and compatible operating systems (Page 17).
- A compatible version of the STEP 7 software: Refer to Preface (Page 3).

If the *Autorun* feature is activated on your PC/PG, the SINAUT ST7 setup program starts automatically when you insert the CD. Otherwise, use the Explorer to open the root directory on the CD and click on the *Setup.exe* application.

The setup program checks which components on the CD can be installed and then displays a list from which you can make your selection. Setups are currently available for the following components.

- SINAUT ST7 configuration: Basic version
- SINAUT TD7 Library

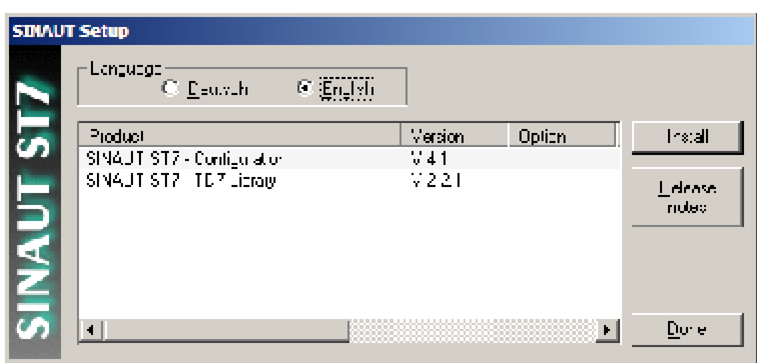


Figure 1-6 Master setup dialog

Select the components you want to install (you can select several packages at once using shift-click) and start the installation by clicking the *Install* button.

Note

To configure SINAUT systems, you must install the two packages *TD7 Library* and *Manager*.

During installation, the setup program will prompt you for any necessary information.

The installation programs perform the following individual operations:

- The programs and data supplied with SINAUT ST7 are copied to the correct locations in the STEP 7 system.
- A new submenu *SINAUT ST7* containing the SINAUT tools is integrated in the *SIMATIC* menu.
- A *SINAUT ST7 Configuration* icon is created on your desktop.

Note

SINAUT ST7 registers in the Microsoft Windows system files. Do not use Microsoft Windows tools such as the Explorer to move or rename SINAUT ST 7 files or folders or to modify STEP 7 data in the Microsoft Windows registry. Following such modifications, your program may no longer functioning correctly.

1.5 Uninstalling SINAUT ST7 software packages

Uninstalling scene of software packages must be in conformity with the operating system you are using. Open the Windows Control Panel and select *Add/Remove Programs*.

In the dialog that opens, select *Change or Remove Programs* and select the required SINAUT ST7 component from the list and then select the *Remove* button. Following a prompt for confirmation, the SINAUT ST7 component is then removed from your system.

Note

When the component is uninstalled, the entries made in the *WINSTART.BAT/CONFIG.SYS/AUTOEXEC.BAT* files by the system are not cleared. Files created dynamically by STEP 7 may also remain on your hard disk after uninstalling a component. This is detected by the uninstall tool and indicated by the message "Some elements could not be removed. You should manually remove items related to the application." However, these dynamically created files do not need to be deleted.

Configuration - overview

2

2.1 The SINAUT ST7 configuration software in the SIMATIC world

The SINAUT ST 7 configuration software represents the user interface for parameter assignment of SINAUT telecontrol systems. With this software, the user can implement and set the parameters for the telecontrol components in a STEP 7 project.

The following figure shows where the SINAUT ST7 configuration software fits into the overall system of the SIMATIC world. The areas with the "cloud" behind them are covered by the SINAUT ST 7 configuration software.

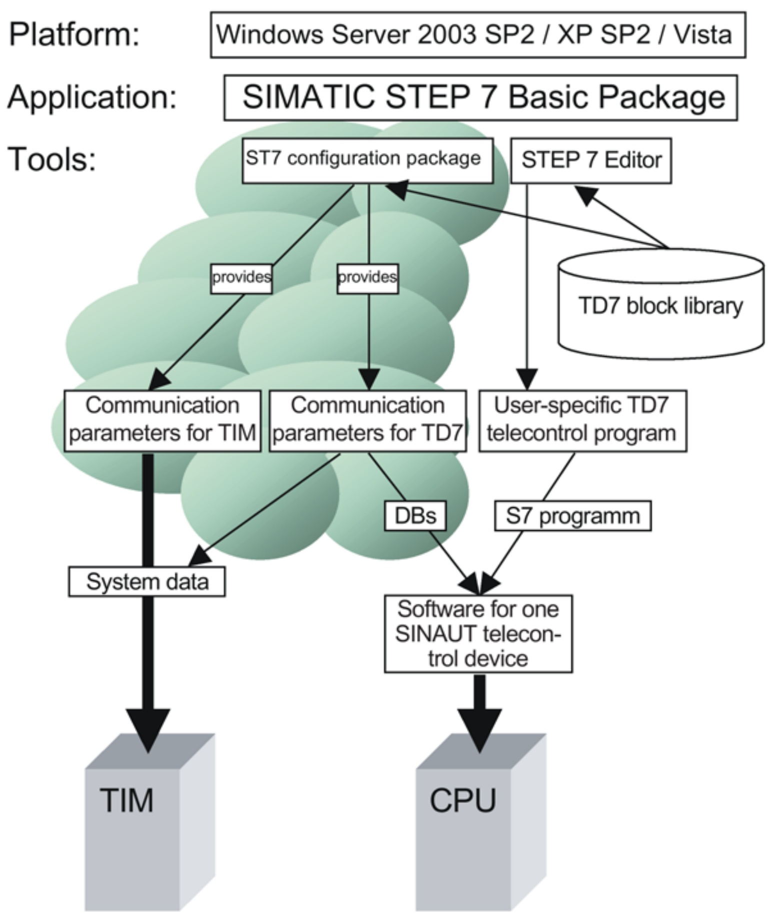


Figure 2-1 The SINAUT ST 7 configuration software within the overall system

The STEP 7 package provides the tools for configuring LANs

The SINAUT ST7 software also allows the configuration of:

- SINAUT networks and WAN network nodes
- SINAUT TIM modules

- SINAUT connections
- Simple Internet communication

To help the user to become familiar with the SINAUT software as simply as possible, the SINAUT tools are always integrated wherever possible in the STEP 7 software. This applies in particular to the parameter assignment of the TIM modules, the SINAUT networks and WAN network nodes.

2.2 Sequence of configuration of a telecontrol system

The configuration of a SINAUT telecontrol system is demonstrated below step-by-step. The dialogs of the SINAUT configuration tool are also explained.

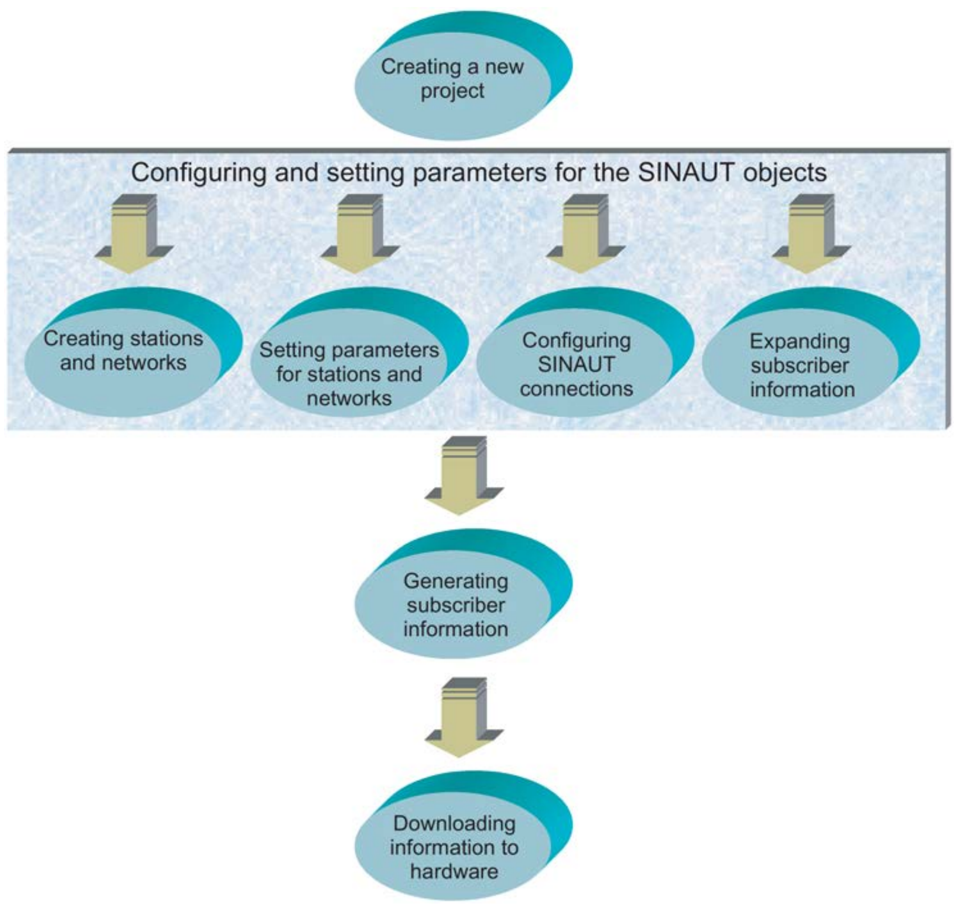


Figure 2-2 Sequence of configuration of a telecontrol system

When configuring a new SINAUT telecontrol system, the SINAUT configuration tool must be started after the network has been configured to allow configuration of the connections and then the data of the subscribers. Following each step, the configured data must be saved. Finally, the system data blocks for TIM and CPU modules and the SINAUT TD7 software blocks must be generated and then downloaded to the relevant modules.

Note

The language for all STEP 7 applications including the configuration tools can be changed in the SIMATIC Manager in the *Options / Customize / Language* menu.

2.3 Use and variants of the SINAUT TD7 software

Basic functions of the TD7 software

SINAUT communication between the CPU modules or between CPU modules and a control center is implemented with the aid of TIM modules. The organization of SINAUT communication is handled by the "SINAUT TD7" software.

TD7 handles the sending and receiving of process data for the local CPU. Data to be sent is read via the backplane bus from the CPU, received data is written to the CPU.

The TD7 variants

In a SINAUT station, the TD7 software must be available either on the CPU or on a TIM. For these two situations, the SINAUT TD7 software is available in two variants:

- **TD7onCPU**

  The software blocks TD7onCPU are used on the CPU and are necessary in all stations with TIM modules that do not have TD7onTIM.

  If fast data acquisition is required or if there is a lot of data traffic in a master station, the variant TD7onCPU should be preferred.

  Configuration is performed in the STEP 7 editor in STL, FBD or LAD.

- **TD7onTIM**

  The TD7onTIM software is available only for Ethernet TIMs and is part of the TIM firmware.

  TD7onTIM is suitable primarily for use in remote S7 stations without time-critical processes. When using TD7onTIM, no or only very little work memory is required on the CPU.

  TD7onTIM cannot be used if the TIM sends or receives ST1 messages. In this case, you must use TD7onCPU

  You configure in the subscriber administration of the SINAUT configuration tool.

The following section provides an overview of the combinations in which the variants of the TD7 software can be used.

Use of the TD7 software

The variants of the TD7 software can be used in the configurations listed below.

		Master station type, TIM type, TD7 variant												
		CPU 300				**CPU 400**			**CPU 400H single**			**CPU 400H red.**	**ST7cc / ST7sc**	**PCS7 Tele-Control ***
		3V-IE Advanced		4R-IE in the rack		3V-IE Advanced	4R-IE standalone		3V-IE Advanced	4R-IE standalone		4R-IE standalone	4R-IE standalone	4R-IE standalone
		TD7 on CPU	TD7 on TIM	TD7 on CPU	TD7 on TIM	TD7 on CPU	TD7 on CPU	TD7 on TIM **	TD7 on CPU	TD7 on CPU	TD7 on TIM **	TD7 on TIM **	-	-
Communications partner, station type, TIM type	**TD7 variant**													
CPU 300														
3V-IE Advanced	TD7on CPU	+	+	+	+	+	+	+	+	+	+	+	+	+
	TD7on TIM	+	+	+	+	+	+	+	+	+	+	+	+	+
4R-IE in the rack	TD7on CPU	+	+	+	+	+	+	+	+	+	+	+	+	+
	TD7on TIM	+	+	+	+	+	+	+	+	+	+	+	+	+
CPU 400														
4R-IE standalone	TD7on CPU	+	+	+	+	+	+	+	+	+	+	+	+	+
	TD7on TIM	+	+	+	+	+	+	+	+	+	+	+	+	+ **
CPU 400H single														
4R-IE standalone	TD7on CPU	+	+	+	+	+	+	+	+	+	+	+	+	+ **
	TD7on TIM	+	+	+	+	+	+	+	+	+	+	+	+	+ **
CPU 400H red.														
4R-IE standalone	TD7on CPU	-	-	-	-	-	-	-	-	-	-	-	-	-
	TD7on TIM	+	+	+	+	+	+	+	+	+	+	+	+	+ **

***** Only single TIM. No communication via two redundant TIM 4R-IE.

****** New with engineering software V5.5 SP2

2.4 Overview of the configuration of a TIM

Configuring the TIM

The following sections contain a detailed description of configuring the TIM. The next sections provide you with a brief summary of this topic and explain which data generated during commissioning needs to be loaded on the TIM to make it operational.

Hardware configuration

Just as with all other S7 modules, the TIM is configured and its parameters assigned in the *HW Config* STEP 7 program. The various TIM modules are available in the hardware catalog of HW Config in the *SIMATIC 300 / SINAUT ST7* folder.

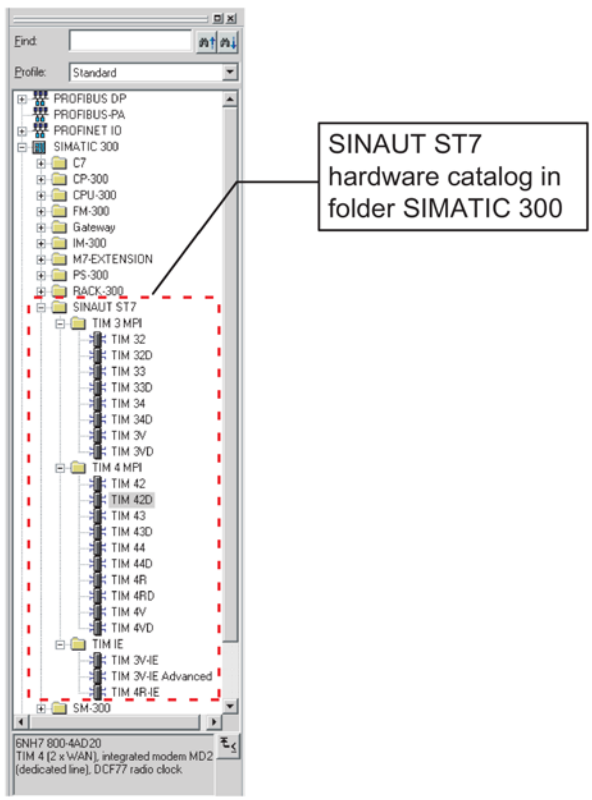


Figure 2-3 The catalog window *Hardware Catalog* with the SINAUT ST7 folder open

You can select the TIM you want to install in a rack from this catalog. The following figure shows an S7-300 station with a TIM and various other modules installed.

Configuring the TIM

If you double-click on the TIM module, you open the properties dialog for the TIM. You can then specify the required properties in a series of tabs.

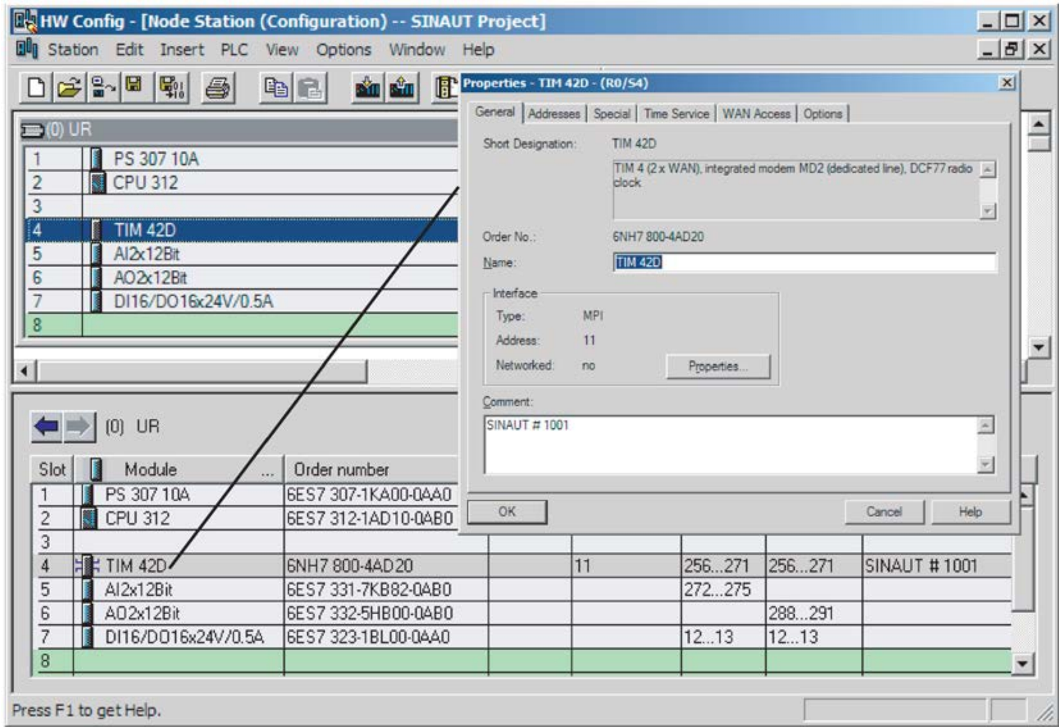


Figure 2-4 S7-300 station with installed TIM and TIM parameter assignment dialog

Configuring the networks and network nodes

The next step is to connect the various stations in a WAN. This part of the configuration is done with the *NetPro* STEP 7 program.

If you are using an IP-based WAN such as DSL or GPRS, take a normal Industrial Ethernet from the NetPro catalog. The classic SINAUT WANs "dedicated line" and "dial-up network" can also be taken from the NetPro catalog. These SINAUT networks have been added to the "Subnets" folder.

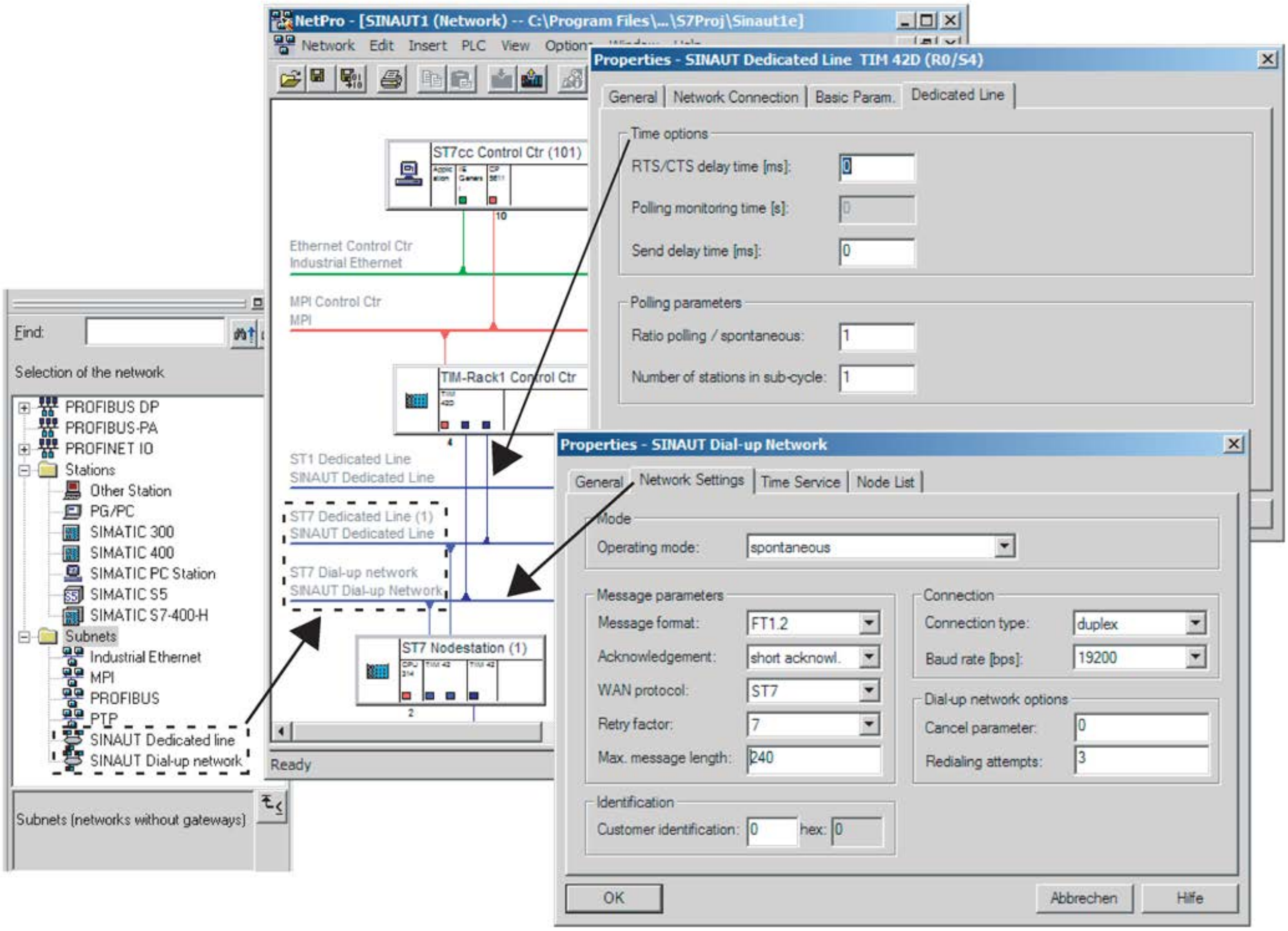


Figure 2-5 Networked system with parameter assignment dialogs for network and network nodes opened

The properties of the two classic WANs are specified in a parameter assignment dialog for dedicated lines and for dial-up networks. The relevant dialog can be opened simply by double-clicking on the corresponding network. To set the parameters for the various network nodes, you also double-click on the connecting line between the network and the node to open the relevant parameter assignment dialog.

Configuring the SINAUT connections

Once the system is networked and the properties for the networks or network nodes have been specified, you can start the SINAUT ST7 configuration tool to complete parameter assignment. This tool is used, on the one hand, to automatically identify which connections are possible from the PC of the control center to the CPU and from CPU to CPU and to make them available for selection. on the other hand, all CPUs, PC(s) of the control center and all TIMs are also assigned a SINAUT subscriber number that is unique throughout the system.

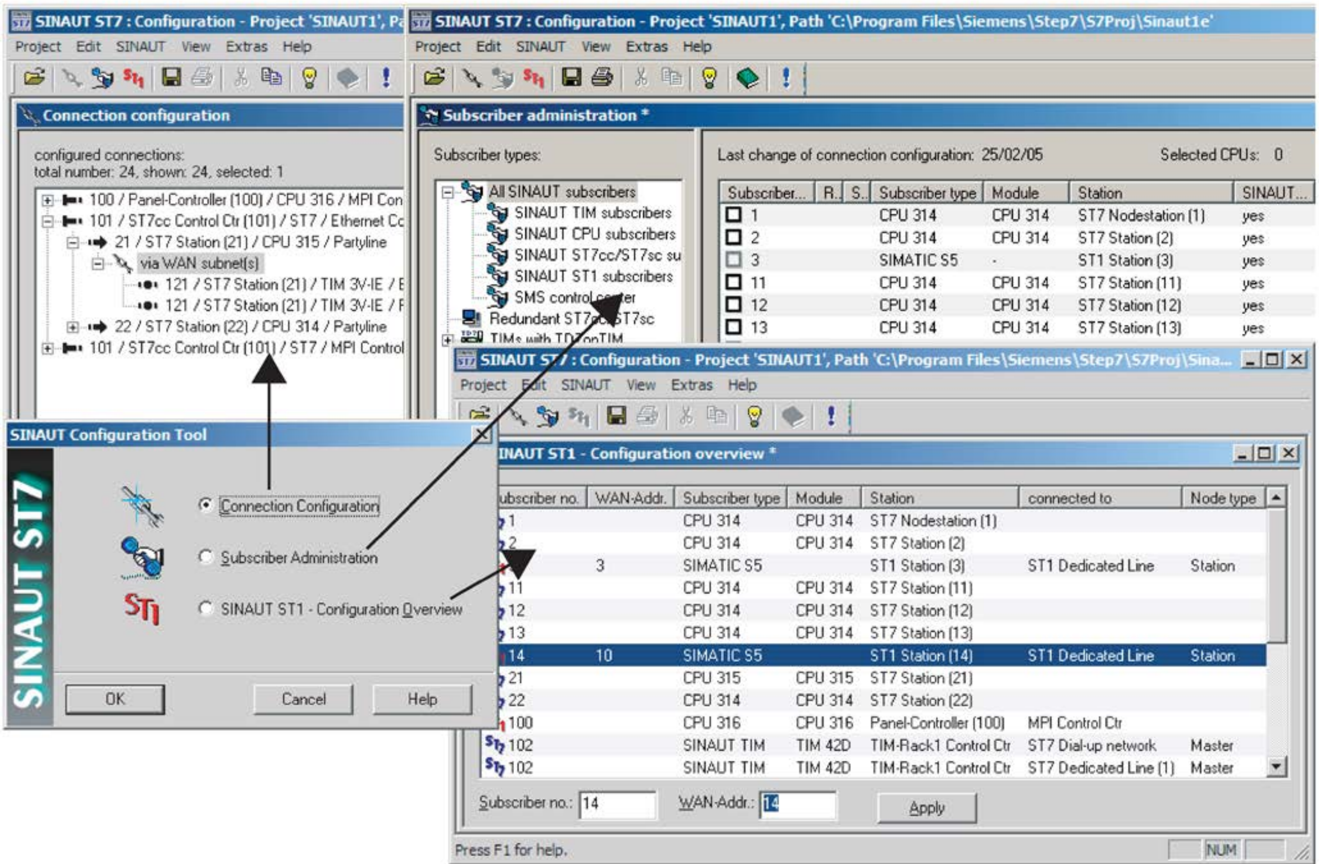


Figure 2-6 SINAUT ST7 configuration tool for connection configuration and subscriber administration

Select the required connections from the right-hand pane of the connection configuration by transferring them to the left-hand window.

After saving the selected connections, you then open Subscriber administration. In the simplest case, no changes need to be made here. The open Subscriber administration can then be saved immediately. The configuration tool then generates the system data blocks (SDBs) resulting from the configuration for all TIMs, the SDBs for the CPUs and some additional data blocks required by the SINAUT TD7 (TD7onCPU) software package.

If the project also includes ST1 devices, it may be necessary to synchronize the addresses for SINAUT ST1. This can be carried out in a further tool SINAUT ST1 - Configuration Overview.

Transferring the SDBs to a TIM

The SDBs and DBs are saved automatically in the program directory of the CPU or the TIM by the configuration tool. From there, transfer them to the CPU and TIM. The following figure shows the SDB folder that was saved in the program directory of the TIM. This must be transferred to the TIM 42 in TIM rack 2 of the control center. This completes the parameter assignment for the TIM that you can now put into operation.

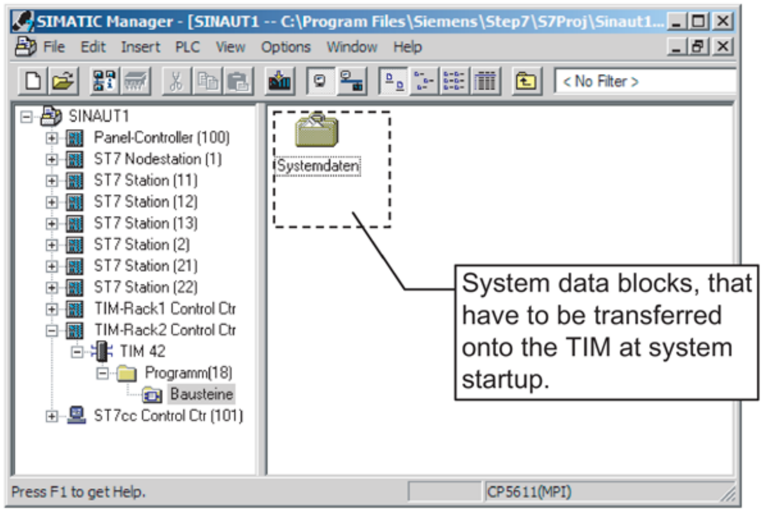


Figure 2-7 System data blocks (SDBs) in the program directory of the TIM

The SDBs are transferred to a TIM as follows:

- Over the MPI interface of the TIM 4 if this is installed as a stand-alone device.
- With a TIM 3 without MPI interface and with a TIM 4 installed as a CP in an S7-300, use the MPI interface of the CPU via which the TIM is accessible on the S7-300 backplane bus and can be reached to load the SDBs.
- With the TIM 3V-IE, TIM 3V-IE Advanced and TIM 4R-IE via the MPI interface of the CPU or via the Ethernet interface.

The SDBs may only be transferred after the TIM has completed its startup. The TIM's LEDs indicate when startup is completed. For more detailed information, refer to the section *Startup activities of the TIM*.

Note

With the TIM 3V-IE variants and the TIM 4R-IE, the SDBs can be stored on the TIM or the S7-300 CPU. This is specified in the parameter assignment of the module. If you decide to store the SDBs on the TIM, the SDBs must be transferred to the module as described above. If the SDBs are stored on the CPU, it is not necessary to transfer the SDBs separately. The TIM receives the SDBs from the CPU during the common startup of the modules in the S7-300 rack.

2.5 GPRS / Internet communication: Overview of configuration

Alternative configurations

The following options are available for communication via the Internet and, if required the GSM network (using GPRS):

- **Highly secure communication**

  Highly secure communication using GPRS and Internet with the SCALANCE M874-2 (or in existing systems with MD741-1)

- **Secure communication**

  Secure communication via GPRS + Internet or only via Internet/DSL with theMSCsec protocol

- **Simple communication**

  Simple communication via GPRS + Internet or only via Internet/DSL with the MSC protocol

Highly secure communication with SCALANCE M874-2

Communication with GPRS and a GSM network combined with the Internet is possible with the Ethernet TIMs and the 2.5G router SCALANCE M874-2. The SCALANCE M874-2 is a VPN router with IPsec and encryption software and its own firewall.

Depending on the network node type, the Ethernet interface of the TIM is configured either as:

- GPRS master station
- GPRS node station
- GPRS station

See section "Interfaces" tab (Page 105) for information on this.

Following the SINAUT configuration, the other communications parameters are configured on the SCALANCE M874-2.

You will find the manuals of the SCALANCE M874-2 on the Internet pages of Siemens Industry Online Support.

(https://support.industry.siemens.com/cs/ww/en/ps/15987/man)

Simple or secure Internet communication with the MSC or MSCsec protocol

The protocols MSC andMSCsec

- **MSC**

  The MSC protocol is used for "simple Internet communication".

  The MSC protocol supports authentication of the communications partner 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**

  The MSCsec protocol is used for "secure Internet communication".

  MSCsec also supports authentication of the communications partner 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 configurable intervals (rekeying interval).

Network node types and usable devices

The following devices can be used depending on the network node type:

- MSC master station

  Only a TIM 4R-IE can be used as the MSC master station.

  Ethernet interface 1 of the TIM is connected to the Internet via a DSL router.

- MSC node station

  TIM 3V-IE Advanced / TIM 4R-IE

- MSC station

  TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE

Connection options for stations/node stations

For the connection between the master station and MSC station/node station there are two alternative options for connection to the network:

- **Internet (direct connection)**

  To connect the MSC station/node station to the Internet directly, a DSL modem is connected to the Ethernet interface of the TIM.

  In this case, a TIM 3V-IE Advanced or a TIM 4R-IE can be used.

- **GSM network and Internet**

  To connect the MSC station/node station to the GSM network via a modem MD720 and then to the Internet, when configuring the interfaces, the switchable WAN interface of the TIM is configured as an Ethernet interface (with a TIM 4R-IE, WAN 1).

  Only the MODEM MD720 can be used as the modem. The MD720 operates in terminal mode and as default is controlled by the application of the connected TIM.

You will find the system manual MODEM MD720 on the Internet pages of Siemens Industry Online Support:

(https://support.industry.siemens.com/cs/ww/en/ps/15923/man)

Simple or secure Internet communication with the MSC or MSCsec protocol

1. Make the settings for networking the interfaces in STEP 7 in the properties dialog of the TIM: "Interfaces" tab (Page 105)

   With a TIM 3V-IE Advanced , use the switchable WAN interface since you can only activate the MSC protocol for one WAN interface configured as an Ethernet interface.
2. In the "Interfaces" tab, set the "Connection mode" for each interface (MSC master station/node station/station).

   With a WAN interface configured as an Ethernet interface, you also do this in the "Interfaces" tab.
3. Create the connection from the master station to the station in the connection configuration of the SINAUT configuration tool.

   For stations that you are connecting directly to the Internet, continue configuration at step 5.
4. When connecting the MSC stations to the Internet via the GSM network, first configure the GSM network providers in the subscriber administration of the SINAUT configuration tool: Configuring the GSM providers for MSC stations (Page 139)
5. You make the settings of the gateways and for stations connected to the GSM network and the assignment of the stations to the previously configured GSM network providers in the subscriber administration of the SINAUT configuration tool in the "Properties of subscriber" > "MSC station list" dialog and the other dialogs that can be called from it: "MSC Station List" tab (Page 160)

   You configure use of the MSCsec protocol for each individual connection in the "Properties MSC station" that is called from the "MSC station list".
6. You configure the interval for the common automatically generated key with the “Rekeying interval", see section "MSC Station List" tab (Page 160).

2.6 SMS

Options for sending SMS messages from a SINAUT station

SMS messages can be sent from a SINAUT station via two network types:

- SMS messages via the GSM network with GPRS
- SMS via the dial-up network

SMS messages via the GSM network with GPRS

The sending of SMS messages via the GSM network with GPRS is supported by Ethernet TIMs using the TD7onTIM TD7 software.

Note

SMS messages via GPRS also with TD7onCPU

The sending of SMS messages via GPRS with TD7onTIM can also be used in conjunction with TD7onCPU. In this case, you can, however, only configure the SMS function for TD7onTIM.

Depending on the interfaces of the TIM modules, the following modems or routers can be used for the GSM connection:

Table 2- 1 SMS messages via GPRS: TIM interfaces and modems / routers

Interface of the TIM	Modem / router
TIM 3V-IE / TIM 3V-IE Advanced	
Ethernet interface	SCALANCE M874-2 (MD741-1)
WAN interface	MD720
TIM 4R-IE	
Ethernet interface 1	SCALANCE M874-2 (MD741-1)
Ethernet interface 2	SCALANCE M874-2 (MD741-1)
WAN interface 1	MD720
WAN interface 2	No connection possible

- Response of the MD720

  Note

  Connection interrupted

  To be able to send an SMS message, the MD720 must be switched over from the GPRS mode to the terminal mode. This requires the modem to be restarted. An existing MSC connection is then terminated.

  The SMS message is transferred by TD7onTIM to the MD720 using an AT command and the MD720 sends the SMS message to the SMS center.

  When the MD720 has accepted the SMS message from TD7onTIM and sent it to the SMSC, a positive acknowledgement is returned to TD7onTIM.

  After successful transfer of the SMS message, the MD720 must be switched back to GPRS mode. The total interruption of the MSC connection typically lasts 3 to 4 minutes.

- Response of the SCALANCE M874-2 / MD741-1.

  The sending of SMS messages must be configured on the router using its Web interface.

  The router does not return an acknowledgement to TD7onTIM about the sending of the SMS message to the SMSC. Only the receipt of the SMS message by TD7onTIM is acknowledged.

You require the following for the TD7onTIM configuration:

- The system object "SmServiceCenter" for configuring the data to access the SMSC
- Data objects of the type "Sms01_S" for the individual SMS messages.

For a description of the configuration, refer to the section The TD7onTIM software (Page 168).

SMS via the dial-up network

The sending of SMS messages via the dial-up network is supported by the TD7onCPU TD7 software.

You require the following for the configuration:

- In STEP 7 / HW Config, an "other station" for the SMS center (SMSC)
- Configuration of the WAN network nodes > "Call Parameters" tab
- DBs for the SMS messages
- Configuration in the SINAUT configuration tool > "SMS Configuration" tab
- The blocks of the SINAUT TD7 library required for the CPU

See also

SINAUT TD7onCPU software package (Page 239)

2.7 Generating system data after changing the configuration of an existing system

Generating system data after changing the configuration of an existing system

Changes to an existing system made in the network configuration *NetPro* or in the hardware configuration *HW Config* are saved there. The new system data blocks, on the other hand, can be generated in various configuration tools.

In the following situations, after changes have been made to the project configuration, the SINAUT configuration tool should be started to generate the system data blocks (SDB) in the *Subscriber Administration*:

- After changing the configuration of a TIM module
- After changing the configuration of a configured connection to a SINAUT subscriber
- After changing SINAUT parameters of a PC station

After changing or adding new subscribers or connections, the connection configuration and then the subscriber administration must be called in the SINAUT configuration tool to generate the SDBs there.

The activities required following a configuration change are summarized in the section Change matrix (Page 233).

3 Configuration in STEP 7

3.1 Creating a project, the stations and networks

3.1.1 Creating a project in the SIMATIC Manager

The first step in configuring a new installation is to create a new project in the STEP 7 *SIMATIC Manager*. This project serves as a directory for all the configuration data of the installation. You create the project in the SIMATIC Manager by selecting the *File / New...* menu and entering the name of the project. After creating the project, the SIMATIC Manager shows the following picture:

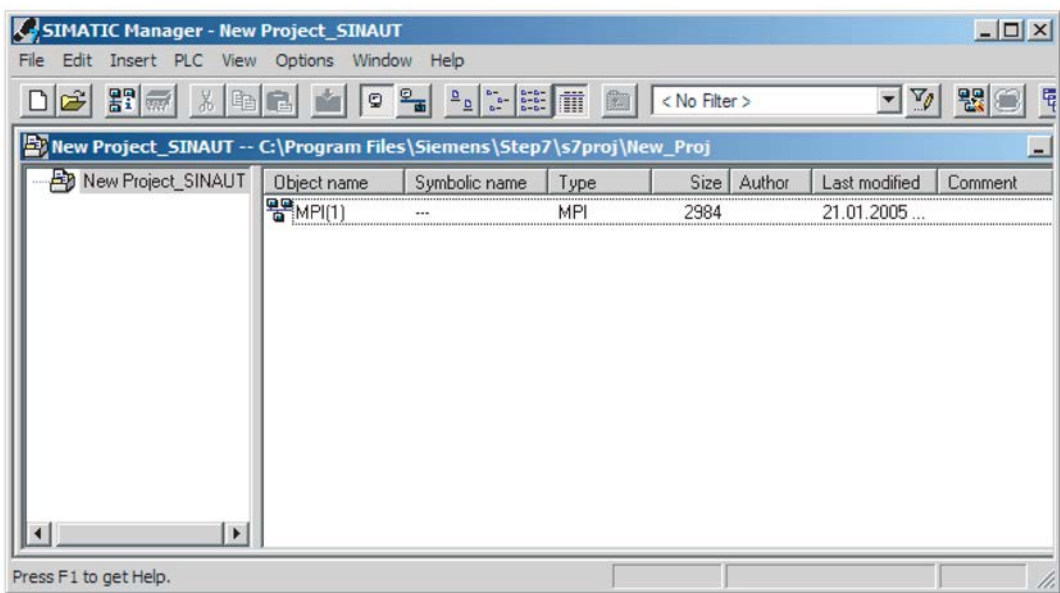


Figure 3-1 Open the project in the "SIMATIC Manager” in the *Details* view.

The generated project is empty apart from an MPI network During parameter assignment, the project will fill up successively with further stations and networks. By double-clicking on the MPI network, the network configuration tool *NetPro* is started.

The STEP 7 *NetPro* tool is used for graphic configuration of network topologies. During network configuration, networks and stations are added to a new project, given parameter settings, and interconnected. The various network types and stations are available in the network and station catalog. The basic functions and possible settings are described in the STEP 7 documentation.

3.1.2 The network and station catalog

The catalog of the network configuration contains:

- PROFIBUS DP objects
- PROFIBUS PA objects
- PROFINET IO objects
- the possible station types
- the known network types including the SINAUT networks

If an object is selected, a brief explanation appears in the lower area of the catalog window. The following figure shows an example of the catalog window.

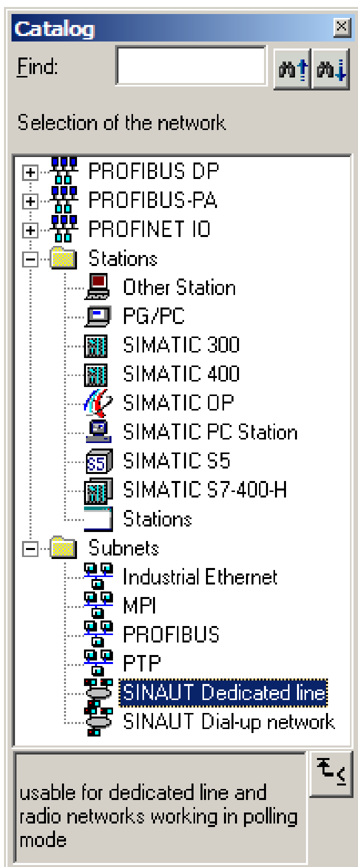


Figure 3-2 Catalog in the network configuration *NetPro*

3.1.3 Creating networks and stations

Networks and stations can be generated in the network configuration in two ways:

- Double-clicking on a catalog entry generates the required object at a free position on the project area.
- Dragging a catalog entry to the project area generates the object immediately at the required position.

The position of installed objects can be changed at any time by moving them with the mouse.

The following figure shows a possible picture after installing several stations and networks.

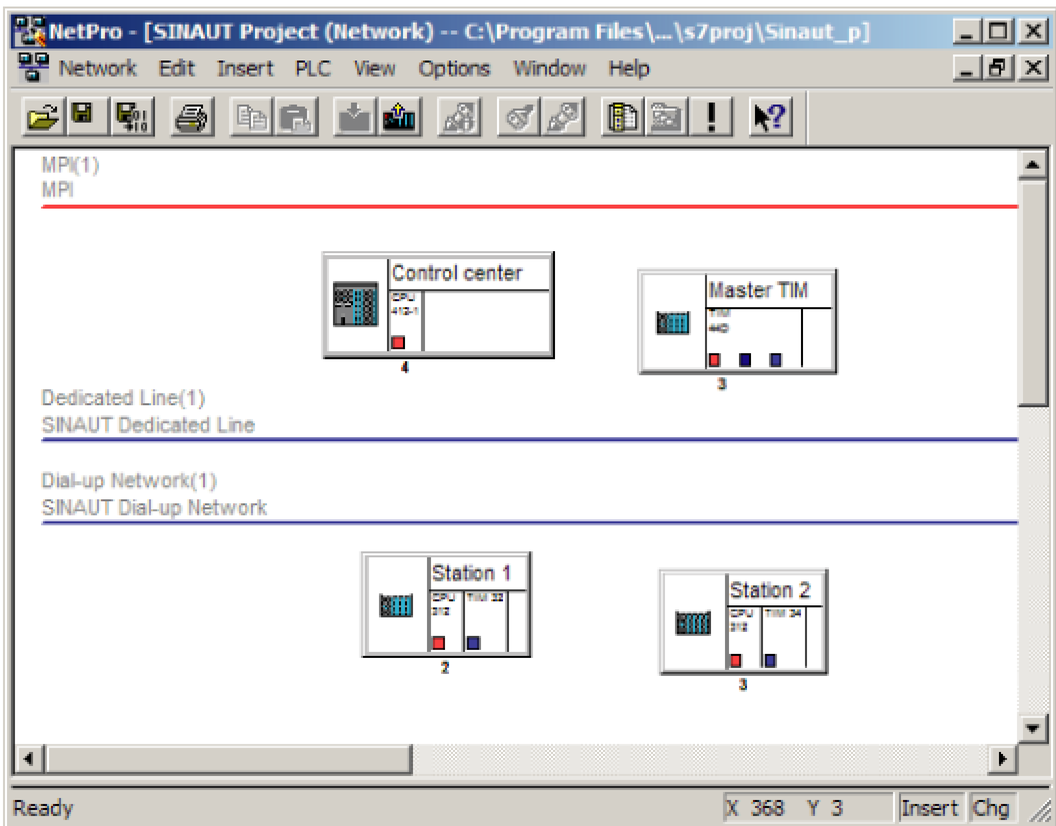


Figure 3-3 Project window in the network configuration *NetPro*

In this example 4 stations, a dedicated line network and a dial-up network were created in addition to the already existing MPI network. The stations include 2 telecontrol stations (*Station 1* and *Station 2*) and a telecontrol master station formed from the S7-400 station *master station* and the S7-300 as a rack for accommodating the *master TIM* (stand-alone).

By selecting an object with the right mouse button, a shortcut menu for the possible operations on this object is displayed.. The following functions are available for objects in the shortcut menu::

- *Open object* (only for stations):
  Starts the hardware configuration *HW Config* for this station..
- *Copy / Paste*:
  Copies and inserts objects.
- *Delete*:
  Deletes the selected object after confirmation in a dialog that opens.
- *Object properties*:
  Opens a specific properties dialog for parameter assignment of the relevant object.. Here, you can already assign a name and comments to the object. If interfaces exist, they are displayed. With networks, network-wide valid parameters are set that affect subsequent configuration.

3.1.4 Creating non-STEP 7 stations

Creating non-STEP 7 stations

SINAUT ST7 supports the configuration of different types of non-STEP 7 stations. These include SINAUT ST1 devices, a SINAUT ST7cc control center or an SMS center. As with the STEP 7 stations, they are created by double-clicking on the icon in the station catalog or by dragging them to the project window. The following objects must be selected for these subscribers:

- For a SINAUT ST7cc/ST7sc control center: *SIMATIC PC station*
- For an SMS center: *Other station*

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

Note

The non-STEP 7 stations listed above are used as placeholders in the network configuration. Hardware configuration of a non-STEP 7 station of the type *SMS center* is not possible since its content is unknown to STEP 7.

To attach created stations to the networks of the project, communication-compliant modules are first configured for the STEP 7 stations in *HW Config* so that the required interfaces can then be configured.

With non-STEP 7 stations, the interfaces can be created immediately.

Creating the interfaces for non-STEP 7 stations

- **SMS center:**

  An SMS center (SMSC) configured as an *other station* is configured in the *Properties* dialog available over the context menu with exactly **1** SINAUT dial-up network node.

- **ST7cc / ST7sc control center:**

  A SINAUT ST7cc or ST7sc control center configured as a *SIMATIC PC station* is equipped with the suitable communications module in hardware configuration and attached to the master TIM over the MPI network. The station is then recognized as a SINAUT master station PC.

  For detailed information on configuring a single or redundant ST7cc/ST7sc control center, refer to the SINAUT ST7cc or ST7sc documentation.

3.2 Configuring stations in STEP 7 / HW Config

HW Config

The hardware configuration program *HW Config* is used to install hardware components in stations. The *HW Config* program is opened by double-clicking on one of the station icons configured in the network configuration.

The module catalog on the right of the hardware configuration window contains the available objects. For SINAUT networks, these are:

- Racks
- Power supplies
- CPU modules
- SINAUT ST7 modules
- Other modules
- Applications for control centers in the SIMATIC PC station directory

The devices are installed in the station from the module catalog.

Slot rules

Possible slot restrictions are checked and reported immediately during configuration of the object. This makes an incorrect hardware configuration impossible. The installation rules include, for example:

- S7-300 + S7-400: Power supply permitted only in slot 1
- S7-300: CPU permitted only in slot 2
- S7-300: IM module permitted only in slot 3
- S7-300: Function modules (CPs, I/O, FMs, TIMs) permitted in slots 4 - 11
- S7-300: There must be no gaps between the modules inserted in slots 4 - 11

The installation rules for function modules are different in the expansion racks depending on the interface module (IM) with which the expansion rack is connected to the basic rack.

The hardware catalog

The catalog of the hardware configuration contains hardware from the following system families:

- PROFIBUS DP
- PROFIBUS PA
- PROFINET IO
- SIMATIC 300

  In this group you will find the TIM modules in the "SINAUT ST7" folder.
- SIMATIC 400
- SIMATIC PC Based Control 300/400
- SIMATIC PC Station

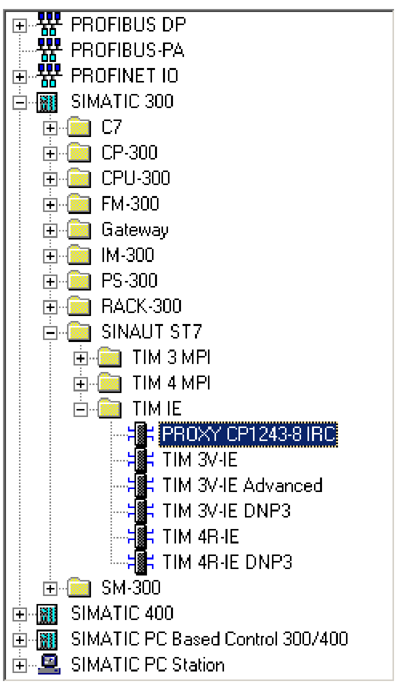


Figure 3-4 Section of the hardware catalog in HW Config

In the der figure the "PROXY CP1243-8 IRC" is selected. This is a TIM 3V-IE Advanced, that serves as a substitute for the CP 1243-8 IRC. The CP receives its basic configuration via the TIM in STEP 7 V5 and is then configured further STEP 7 Basic as of V13.0 SP1. You will find the special configuration rules for this module in the section PROXY CP1243-8 IRC (Page 48).

3.2.1 Installing racks and configuration of the modules

After creating the stations perform the remaining configuration of the SINAUT modules in HW Config as normal in STEP 7 V5 for all other modules.

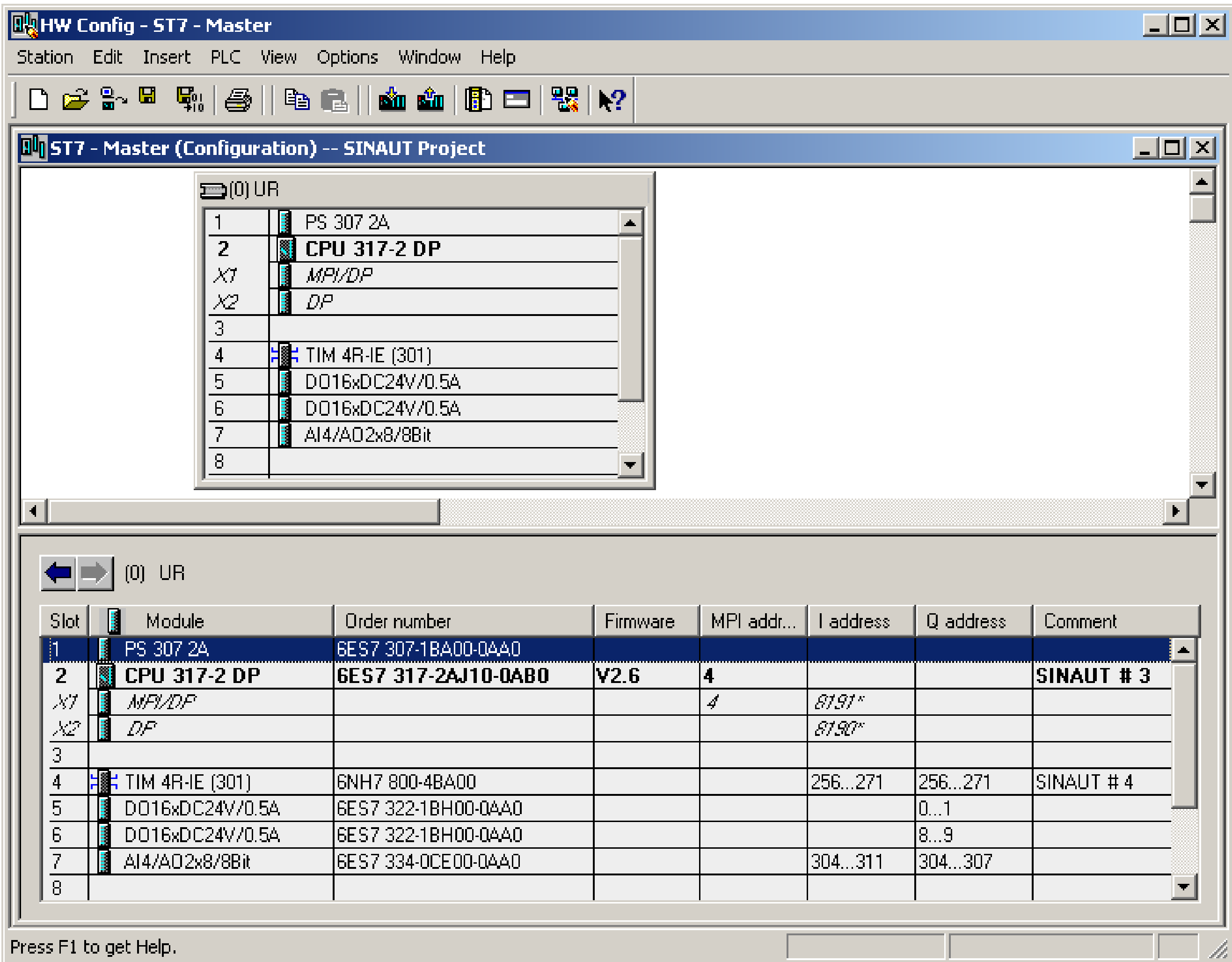


Figure 3-5 HW Config with SIMATIC 300 rack and various modules

Configuration

By double-clicking on a module in HW Config, you open the Properties dialog for parameter assignment of the module. The content of the dialog depends on the module type.

As an alternative, you can also open the same Properties dialog in NetPro.

Note

At least all the TIM modules of the project need to be configured in the Properties dialog, for example to create the interfaces.

3.2.2 Cycle monitoring time of the CPU

Malfunctions of theTIM 3V-xxx.

Note

No cycle monitoring time of the CPU

If the cycle monitoring time of the CPU is set to very low values, for example 1 ms, this can lead to malfunctions on theTIM 3V-xxx. In this case, increase the cycle monitoring time of the CPU or use a TIM 4R-xxx.

3.2.3 PROXY CP1243-8 IRC

"PROXY CP1243-8 IRC" as CP substitute in HW Config

The CP 1243-8 IRC is used to expand SINAUT systems set up based on S7-300 stations and TIM modules with modules from the product range of the S7-1200. For this, the CP 1243-8 IRC was introduced first.

This telecontrol CP supports the functions of a TIM 3V-IE Advanced and further functions of the telecontrol CPs.

The configuration if the CP is divided into two parts:

- Basic configuration with telecontrol functions in STEP 7 V5

  Points to note when configuring are described below.
- Further configuration in STEP 7 Basic as of V13.0 SP1

  Here further functions are enabled and set, such as data point configuration.

Overview of the configuration in STEP 7 V5.5

1. In STEP 7 V5 insert any S7-300 CPU in an S7-300 station.

   The CPU serves as a placeholder for the SINAUT subscriber number.
2. Insert the proxy as the substitute for the CP.

   You will find the module in the catalog of HW Config under:

   SIMATIC 300 > SINAUT ST7 > TIM IE > "PROXY CP1243-8 IRC"
3. Configure the proxy as a TIM 3V-IE Advanced.

Special properties of the proxy module "PROXY CP1243-8"

Like a TIM 3V-IE Advanced the proxy supports parallel connection to two WAN networks via the serial and the Ethernet interface.

Compared with the TIM, the proxy has the following differences:

- **Only ST7**

  Only ST7 is supported as the telecontrol protocol. The use of the CP in systems that use the older ST1 protocol is not possible.

- **Interface configuration**

  Both interfaces of the proxy can be configured.

- **Station**

  The proxy can only be configured in a station. Use in a note station or master station is not possible.

- **No TD7 software**

  For the proxy, no TD7 software is configured, neither TD7onCPU nor TD7onTIM.

  The required data is configured in STEP 7 Basic in the data point configuration.

- **Time slave**

  The proxy can only be configured as a time slave. The function of time master is not supported.

  Synchronization via MPI is not possible.

- **"Mode" of SINAUT-dial-up networks**

  In the STEP 7 properties dialog of dial-up networks for connected stations with a proxy only the “spontaneous“ mode is supported.

- **Connection mode (Ethernet)**

  Configuration in the "Interfaces“ tab

  – MSC station

    The communication is handled using the MSC protocol. The proxy is configured to connect to the Internet as an MSC station via a DSL router.

  – GPRS station

    The proxy is configured to connect to the Internet as a GPRS station via an MD720.

  – Neutral

    On the Ethernet interface, there is normal TCP/IP communication with the ST7 protocol. The proxy is not connected via GPRS and the MSC protocol is not activated.

- **Connection mode (RS232)**

  Configuration in the "Interfaces“ tab

  – MSC station

    The serial interface is configured as an Ethernet interface. The communication is handled using the MSC protocol. The proxy is configured to connect to the Internet as an MSC station via a DSL router.

  – Neutral

    The serial WAN interface of the proxy is set to the connection mode "Neutral“ if communication is handled via a dial-up network or dedicated line.

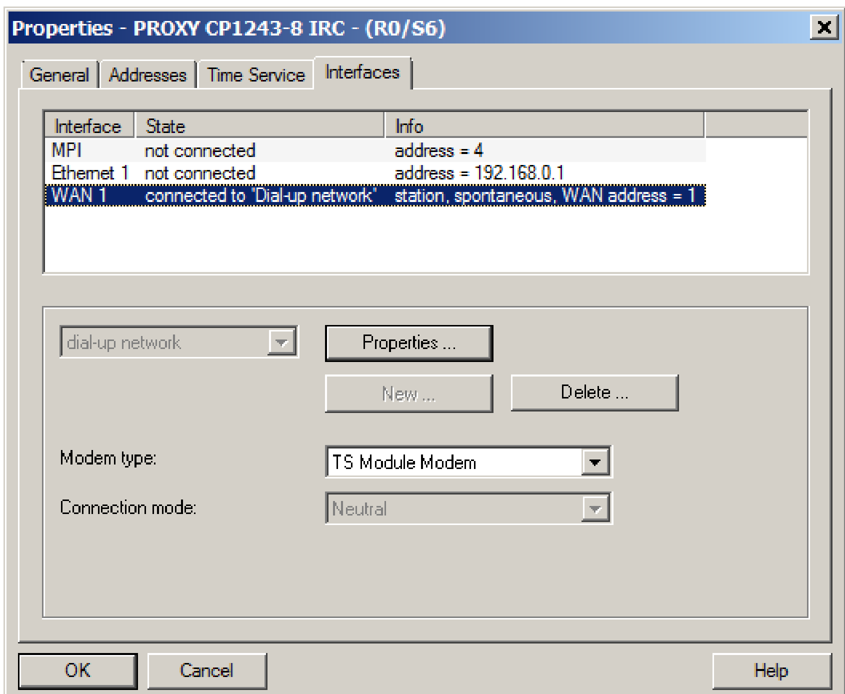


Figure 3-6 Properties dialog, "Interfaces" tab with WAN interface to which a dial-up network is connected.

- **Modems for dial-up networks**

  In addition to the TIM for communication of the CP via dial-up networks, the required TS modules for the serial WAN interface of the proxy can be selected in theInterfaces tab, see below.

- **AT strings for TS modules**

  AT strings were added for the TS modules, see below.

- **No SMS configuration**

  The configuration of the sending of SMS messages is not possible for the proxy in STEP 7.

  You can arrange for the sending of SMS messages in STEP 7 Basic for every data point.
- **No loading of firmware**

  The loading of firmware files on a CP 1243-8 IRC is not supported.

TS modules in the "Interfaces“ tab

TS modules are used to connect the serial interface of the CP to dial-up networks that are also configured in STEP 7 V5 in the “Interfaces“ tab of the Properties dialog of the proxy. The following figure shows drop-down list for selecting the modem type.

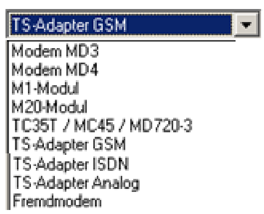


Figure 3-7 Selectable modem types of a WAN interface with a connection to a dial-up network

The following entries are available for the TS modules:

- TS Module GSM

  Connection to a GSM network
- TS Module ISDN

  Connection to an ISDN network
- TS Module Modem

  Connection to an analog dial-up network
- Third-party modem

  Select the "3rd party modem" entry in the following cases:.

  – Connection to a dedicated line network

    For the connection to a dedicated line network via a dedicated line or analog wireless modem, a TS module RS-232 is connected to the CP.

  – Connection to a dial-up network

    For the connection to a dial-up network via a third-party modem, a TS module RS-23 is connected to the CP.

  The TS module RS-232 cannot be configured in STEP 7 V5.

AT strings for TS modules

If the CP is connected to a dial-up network via a TS module, the suitable AT string must be configured in the properties dialog of the network node in STEP 7 V5 for the network node of the proxy.

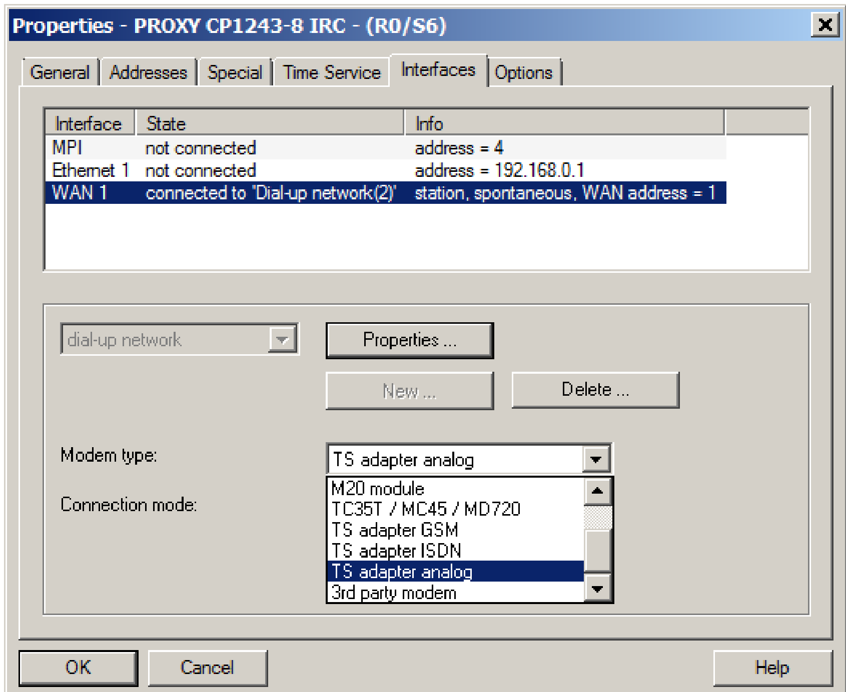


Figure 3-8 Properties of a dial-up network node "AT initialization " tab

For dial-up network connections via TS modules, AT strings are configured in the AT Initialization tab of the Properties dialog of the dial-up network node. Depending on the TS module being used, the following AT strings are preassigned:

- CP with TS Module GSM

  `ATE0S0=1&D2+CBST=7,0,1;+CRC=1;&W+IPR=115200`

- CP with TS Module Modem

  `AT&FE0&M0&Q6S0=1x3&w0`

- CP with TS Module ISDN **-** partner (master station) with GSM connection:

  AT string of the TS Module ISDN in the station:

  `AT&FE0\N1`

Only when transferring via an ISDN network with the following configuration do the AT strings need to be adapted manually in the "Initialization string" input box.

- CP with TS Module ISDN **-** partner (master station) with Modem MD4:
  - AT string for the TS Module ISDN in the station:

    `AT&FE0\N2`

  - AT string for the Modem MD4 in the master station:

    `ATS45=85$P5\N0&W$M=1`

Configuring with the SINAUT Engineering Software

Configure the connection of the proxy in the SINAUT configuration tool as usual for a TIM.

Note that for the proxy, neither TD7onCPU nor TD7onTIM is configured.

In the subscriber management of the configuration to, the proxy appears as follows:

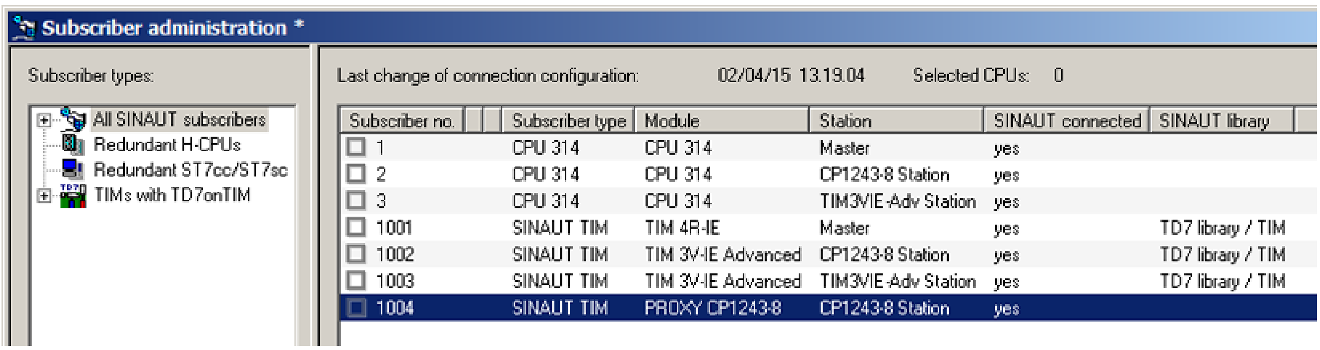


Figure 3-9 View of a proxy in the SINAUT subscriber administration

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:

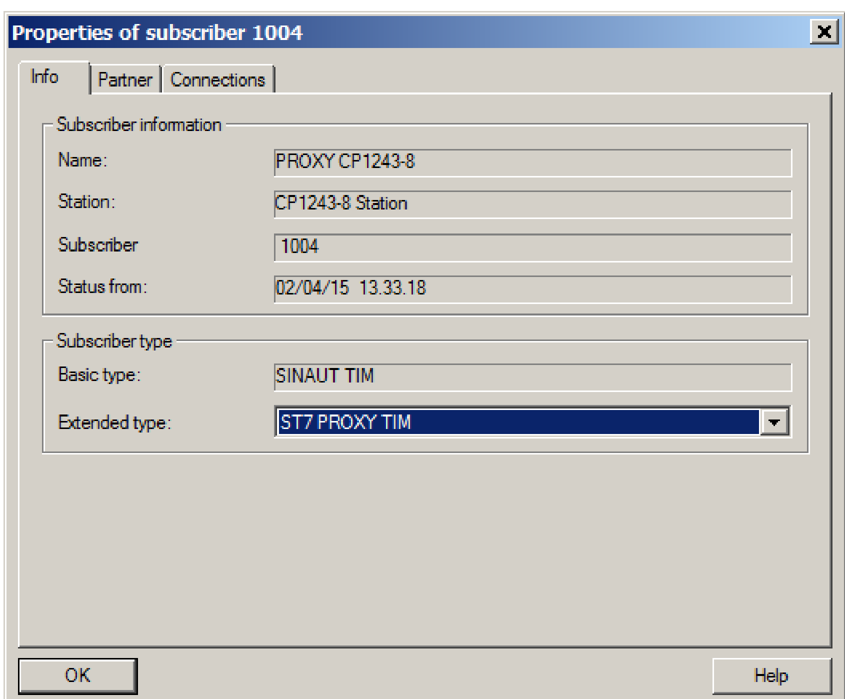


Figure 3-10 Expanded type of the proxy in the Properties dialog of the subscriber

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".

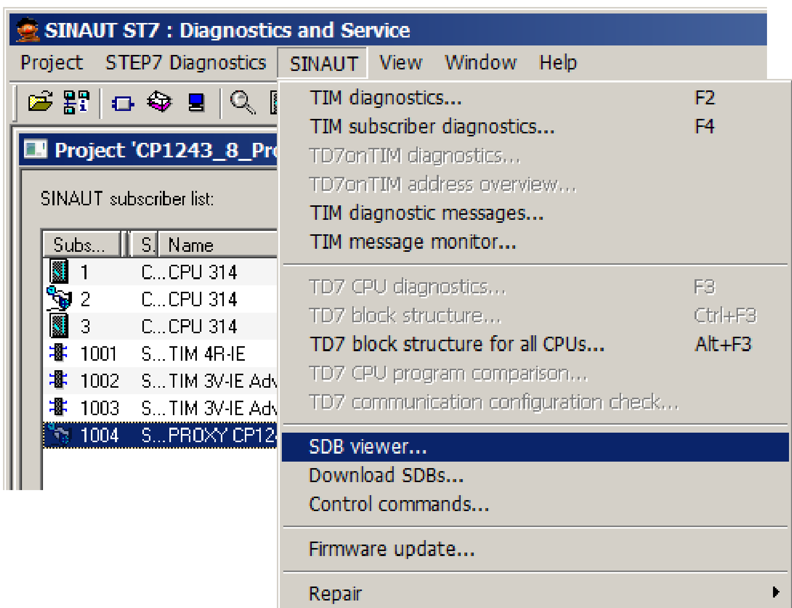


Figure 3-11 Opening the "SDB display" dialog

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.

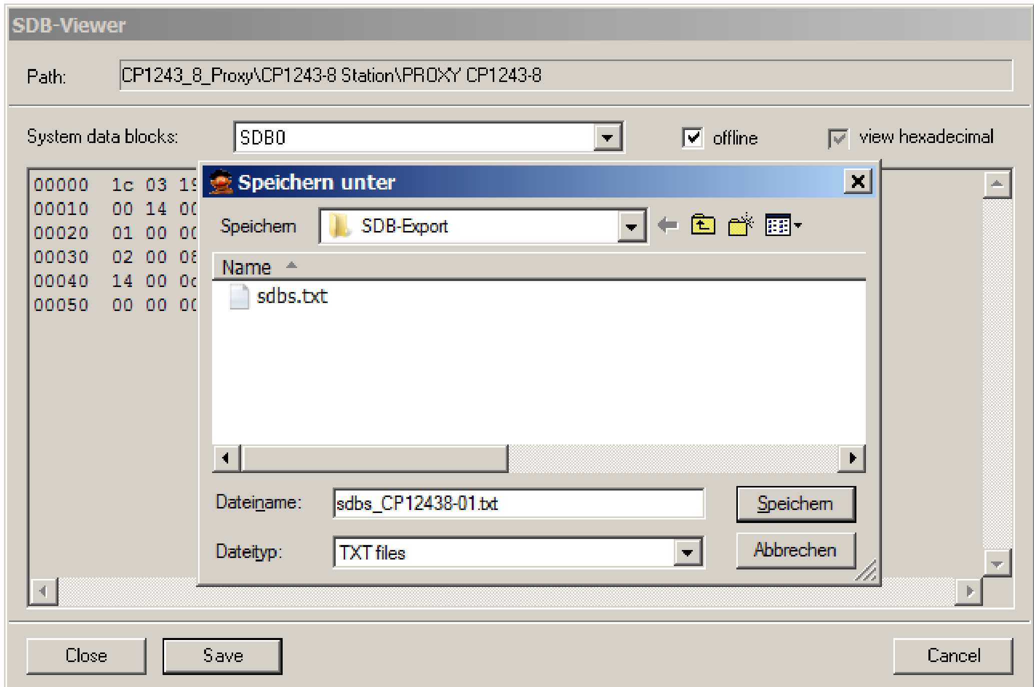


Figure 3-12 Export of the configuration data of a proxy in the example as the file "sdbs_CP12438-01.txt".

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 Basic

You require this exported text file later for the configuration of the CP in STEP 7 Basic.

3.3 Configuring networks and network nodes in STEP 7 / NetPro

Sequence

A fully configured network is required for the remaining steps in configuration:

- Configuration of SINAUT connections
- Configuration of the SINAUT subscriber data
- Configuration of SINAUT objects of an Ethernet TIM
- Generation of system data blocks (SDBs) and data blocks (DBs)

During the initial network configuration, the following is achieved:

- Connection of the modules with network capability to the networks
- Creation of a graphic view of the network consisting of one or more subnets
- Specification of the required properties and parameters for each subnet and each networked module
- Documentation of the network configuration

Starting the parameter assignment dialogs for networks and network nodes

If you double-click on a network or network node icon or select the Object Properties menu in the shortcut menu (right-hand mouse button), the Properties dialog opens to allow you to set parameters. Here, you can connect modules with networking capability with the networks and adapt the properties of the relevant object to your requirements. All parameters have default settings that simplify parameter assignment.

First the network attachments are made.

3.3.1 Generating network attachments

To network a project, the communication-compliant modules (for example CPU or TIM) must be connected to suitable networks. The modules in the station icons in the project window of the network configuration include interface and network node icons displayed in different colors according to the network type.

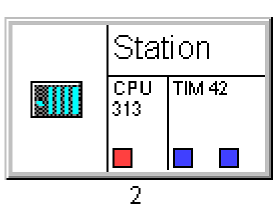


Figure 3-13 Icon of a master TIM station in the project window of the network configuration containing a TIM module and three network nodes

The station icon shows a master TIM containing a communication-compliant TIM 44D module. This module has three network nodes, visible as small squares in the module icon.

If these network nodes are not connected to a network as in the example, the relevant network node is not networked.

You connect network nodes with the networks using the mouse by dragging the network node icons to the line of the required network.

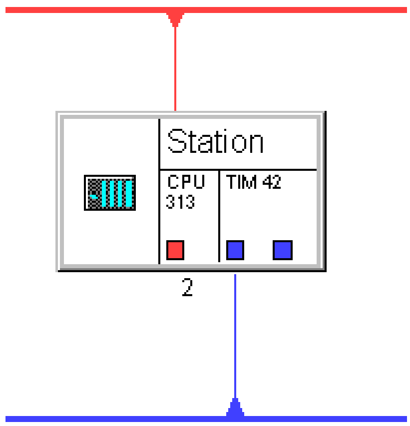


Figure 3-14 Station with three network nodes, two of which are networked

As an alternative you can attach to a network in the *Properties* dialog of the module available with the shortcut menu (right-hand mouse button) and selecting the *Object Properties...* menu. The dialog is described in the section dealing with parameter assignment of network nodes.

With the simpler technique of dragging network node icons to the network line, only suitable partners can be networked; in other words an MPI node can only be connected to an MPI network. This makes incorrect attachment impossible.

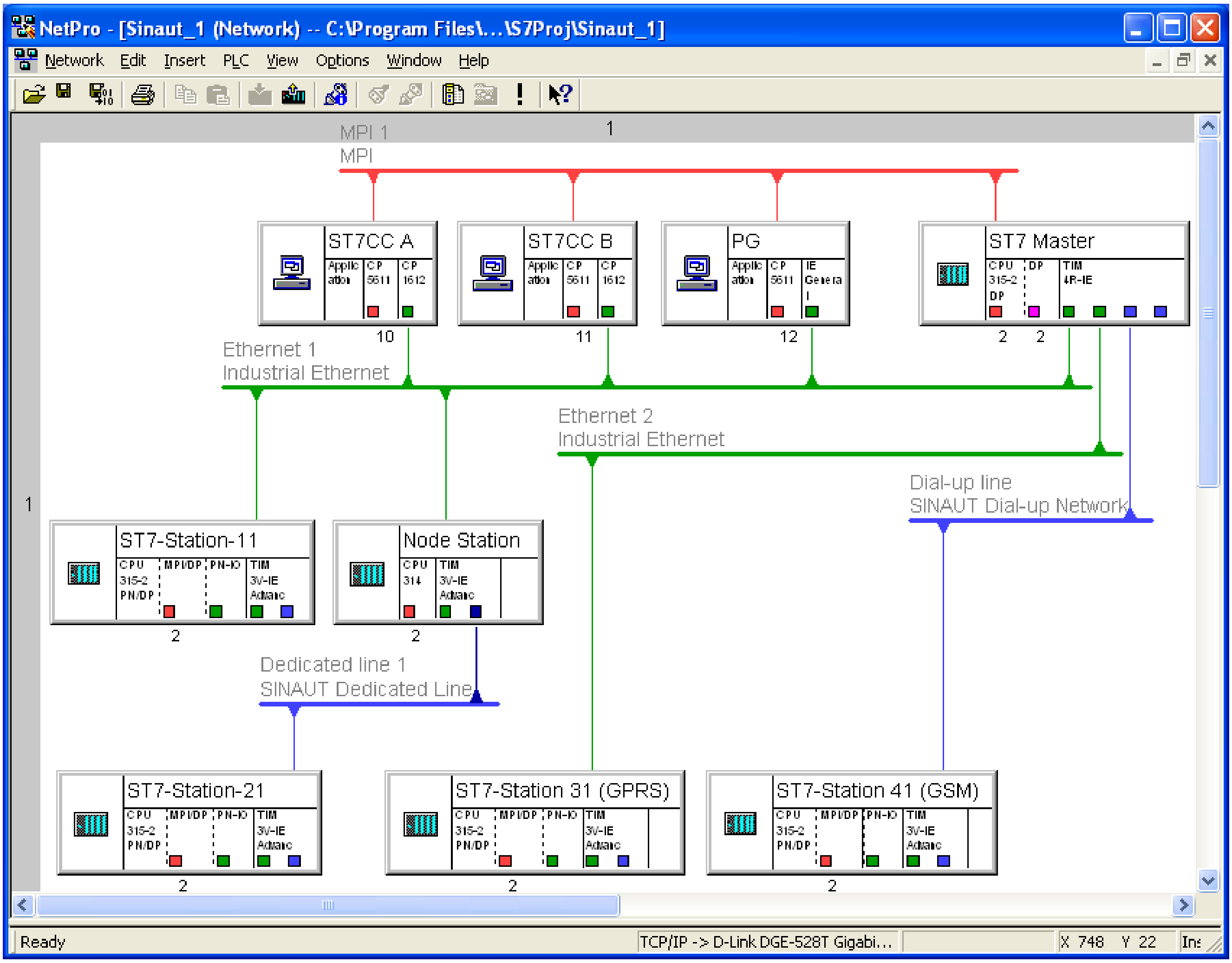


Figure 3-15 Networked sample project

The following network attachments were made in the sample project:

- The master station with the CPU 317-2 DP, the redundant ST7cc control center created as PC stations and the PG were connected to the MPI network and to Ethernet 1.
- The master TIM 4R-IE was further connected with Ethernet 2, dedicated line 2 and the dial-up network.
- The TIM 3V-IE in station 1 was connected to Ethernet 1.
- The TIM 3V-IE Advanced of the node station was connected to Ethernet 1 and dedicated line 1.
- The TIM 3V-IE in station 21 was connected to dedicated line 1.
- The TIM 3V-IE in station 31 (GPRS station) was configured for the MSC protocol and connected to Ethernet 2.
  Special feature: With this configuration, the WAN node to the right of the TIM is displayed green and is connected to the Ethernet network.
- The TIM 3V-IE Advanced in station 41 (GSM station) was connected to dial-up network 1.

The blue WAN connection of the master TIM and the connection of the node station to dedicated line 1 are shown in dark blue and indicate a connection in which the connected interface for the relevant subnet was configured as a master station. This is explained in the description of the network node parameter assignment.

Note

After changing connections, even if these are re-established again later, the SINAUT configuration tool with the *connection configuration* and the*subscriber administration* must be called.

Note

Further information

For the network configuration note the information in the following sections:

- On use of the variants of the TD7 software: Section Use and variants of the SINAUT TD7 software (Page 29)
- On configuration of the Ethernet interface Section "Interfaces" tab (Page 105)

Printing network information

The project can be printed and documented as a graphic or as text using the *Network / Print* menu.

3.3.2 Configuration of MPI networks

To assign parameters for MPI networks, the *Properties - MPI* dialog is opened by double-clicking on the MPI network or using the shortcut menu *Object Properties ...*.

General tab

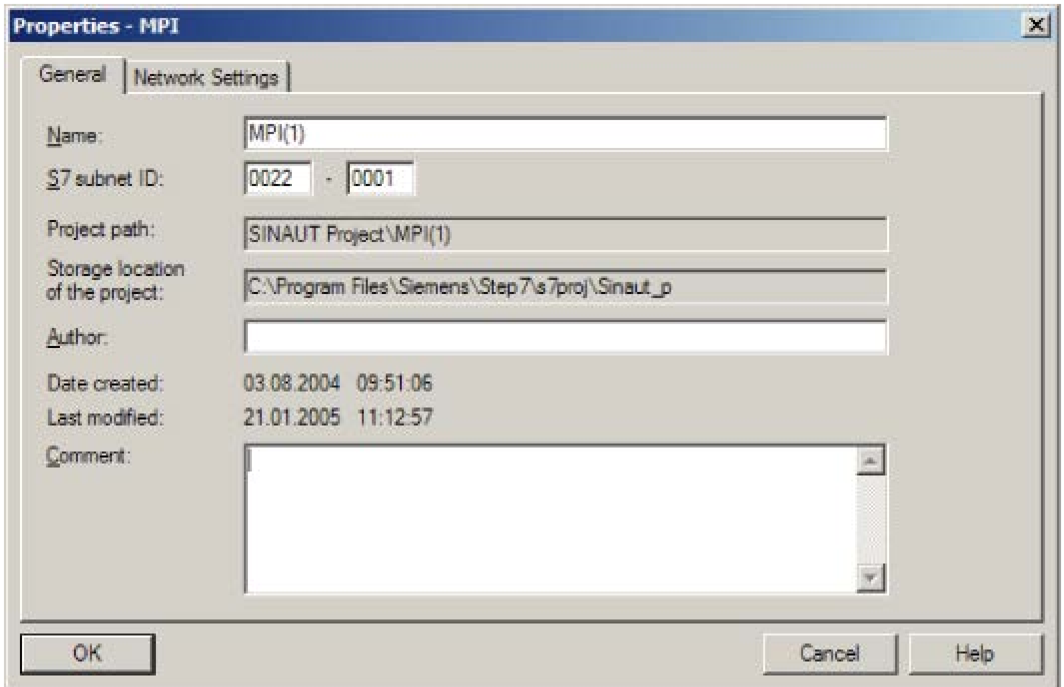


Figure 3-16 *Properties - MPI* dialog, *General* tab

The following parameters are available in this tab:

- *Name:*
  As default in the *Name* input box is the standard designation of the network. You can change this to suit your purposes. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- *S7 subnet ID*:
  The subnet ID is made up of two numbers separated by a hyphen:
  - The number for the project
  - The number for the subnet

  If you want to go online with a PG without a consistent project, you must know the subnet ID. The subnet ID is also printed out when you print the network configuration.
- The *Project path* is displayed.
- The *Storage location* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Network Settings tab

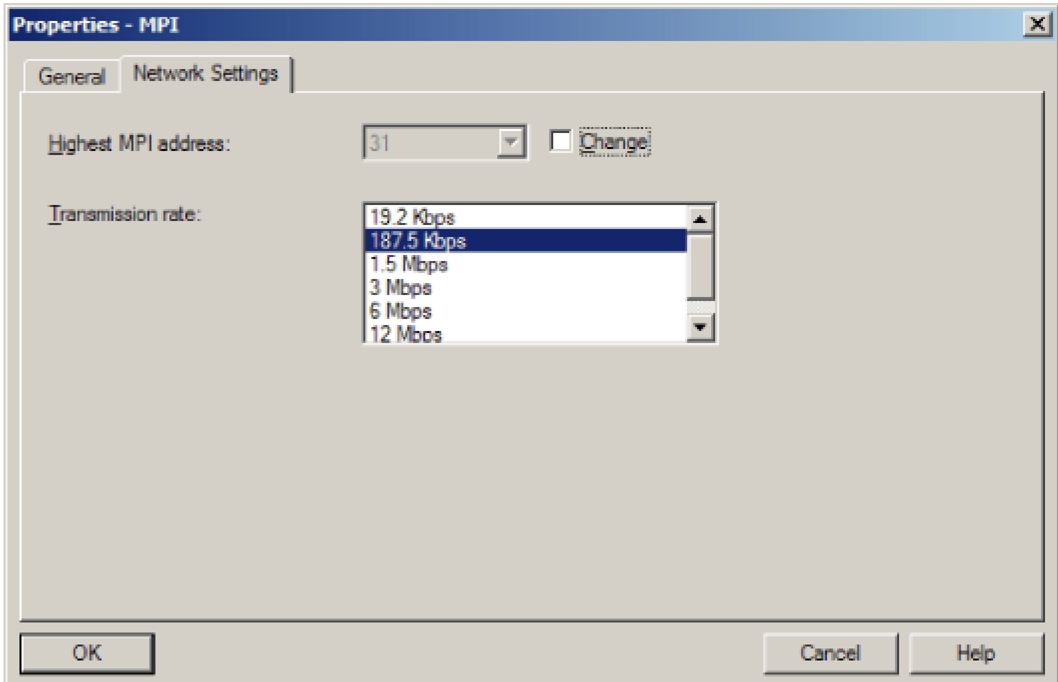


Figure 3-17 *Properties - MPI* dialog, *Network Settings* tab

The following parameters are available:

- The *Highest MPI address* is displayed.
  This serves to optimize the MPI network.. It is recommended that you retain the highest MPI address set by STEP 7.
- The *Change* option:
  By enabling the option, the highest MPI address can be changed..
- *Transmission speed*:
  The setting for the transmission speed of the MPI network depends on the properties of the MPI subscriber being used and must not be higher than that of the slowest subscriber. The default value can generally be used.

3.3.3 Configuring MPI network nodes

General tab

The *General* tab informs you about general parameters of the MPI interface.

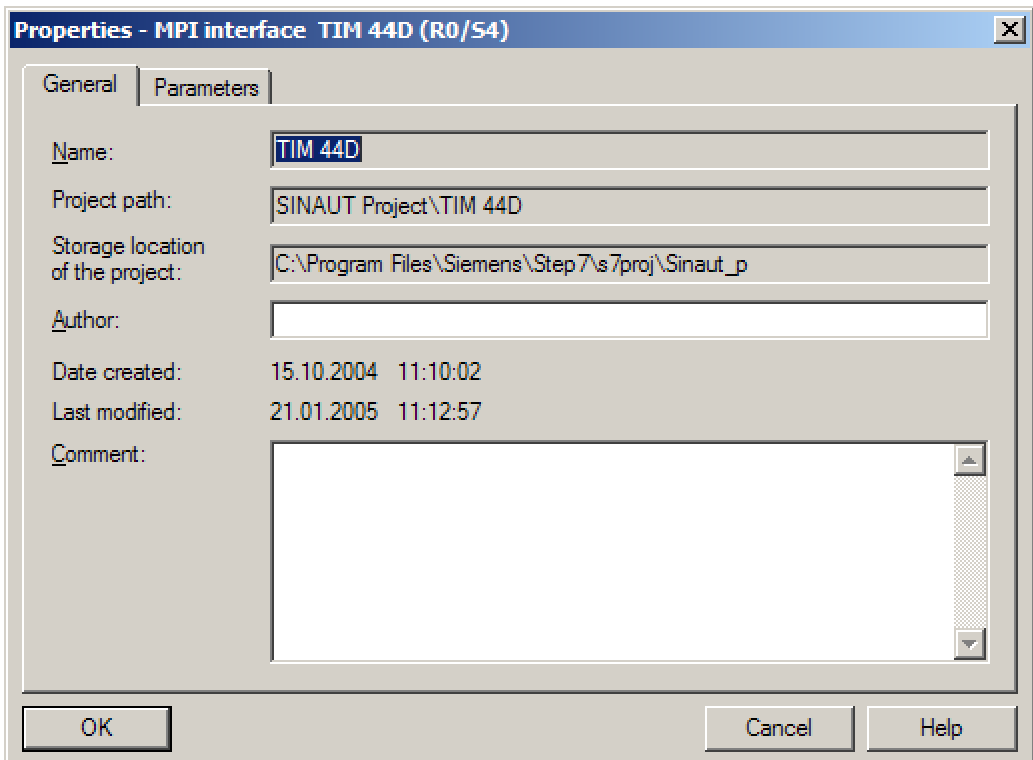


Figure 3-18 *Properties - MPI interface* dialog, *General* tab

The following parameters are available:

- Name:
  The *Name* box displays the name of the module in SIMATIC stations. You can only change the default interface name in SIMATIC PC stations and other stations. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Parameters tab

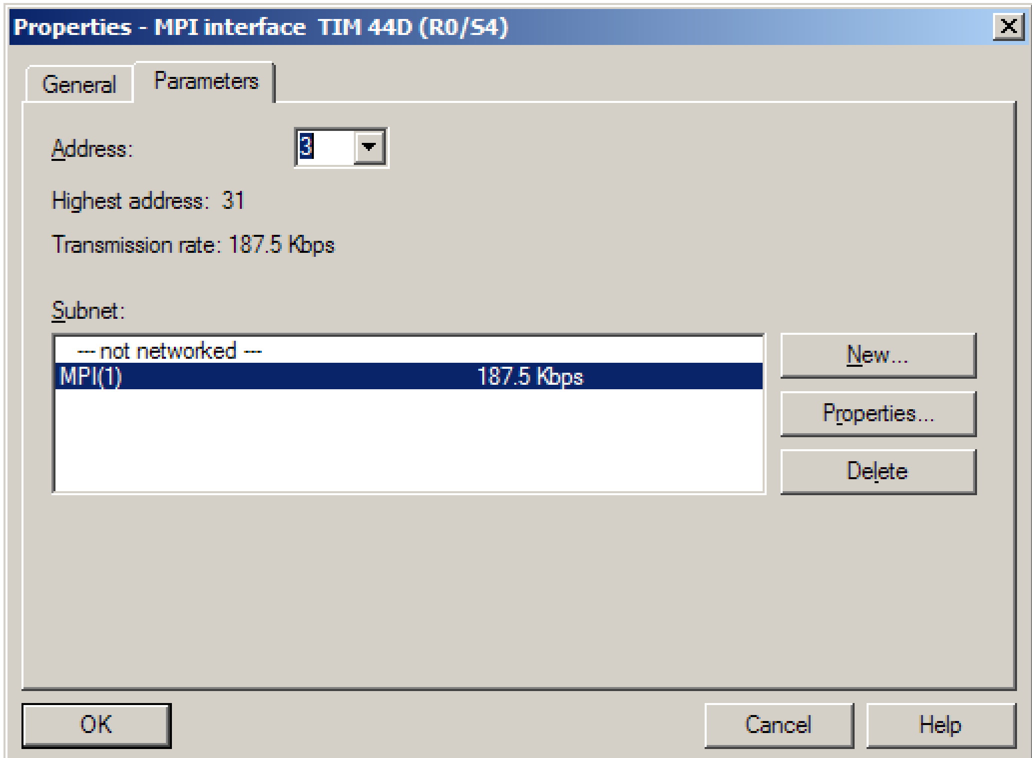


Figure 3-19 *Properties - MPI interface* dialog, *Parameters* tab

The *Parameters* tab provides the following parameters:

- The network *address* in the local MPI network.
  The address of the network node can be modified. There is a consistency check which blocks network addresses that have already been assigned.
- The *highest MPI address* in the network is displayed.
- The *transmission rate*:
  Just like the highest MPI address, this cannot be modified here but only in the parameter assignment of the MPI network.
- The *Subnet*:
  This lists all the networks of this type in the project. If the subscriber is not connected, the row *----not networked----* is shown as selected in the *Subnet* list.
  - If the current subscriber is connected, the row of the relevant network is shown as selected. You can set parameters for the current network with the *Properties* button.
  - If no connection exists, a network connection can be set up using the *New* button.
  - An existing connection can be deleted with the *Delete* button.

3.3.4 Configuring Industrial Ethernet

General tab

The parameter assignment of the Industrial Ethernet is made in the *Properties - Industrial Ethernet* dialog.

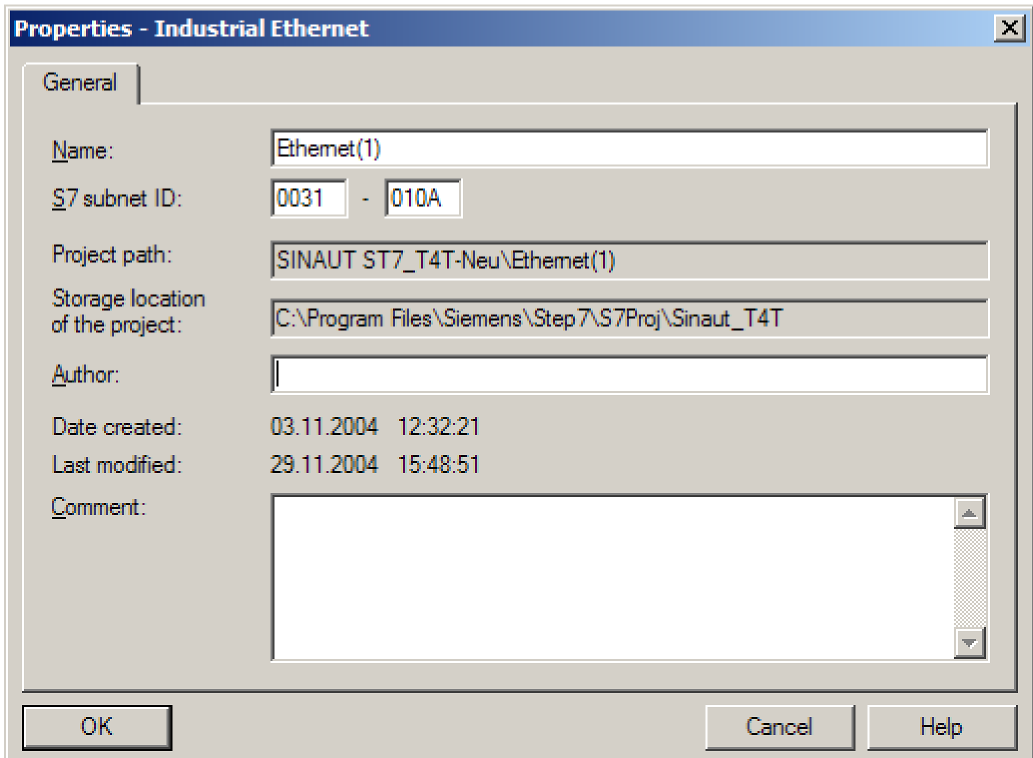


Figure 3-20 *Properties - Industrial Ethernet dialog*, *General* tab

The following parameters are available in this tab:

- *Name:*
  As default in the *Name* input box is the standard designation of the network. You can change this to suit your purposes. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- *S7 subnet ID*:
  The subnet ID is made up of two numbers separated by a hyphen:
  - The number for the project
  - The number for the subnet

  If you want to go online with a PG without a consistent project, you must know the subnet ID. The subnet ID is also printed out when you print the network configuration.
- The *Project path* is displayed.
- The *Storage location* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.

- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

3.3.5 Configuring Ethernet nodes

This type of interface is available among the SINAUT subscribers only for the Ethernet TIMs.

General tab

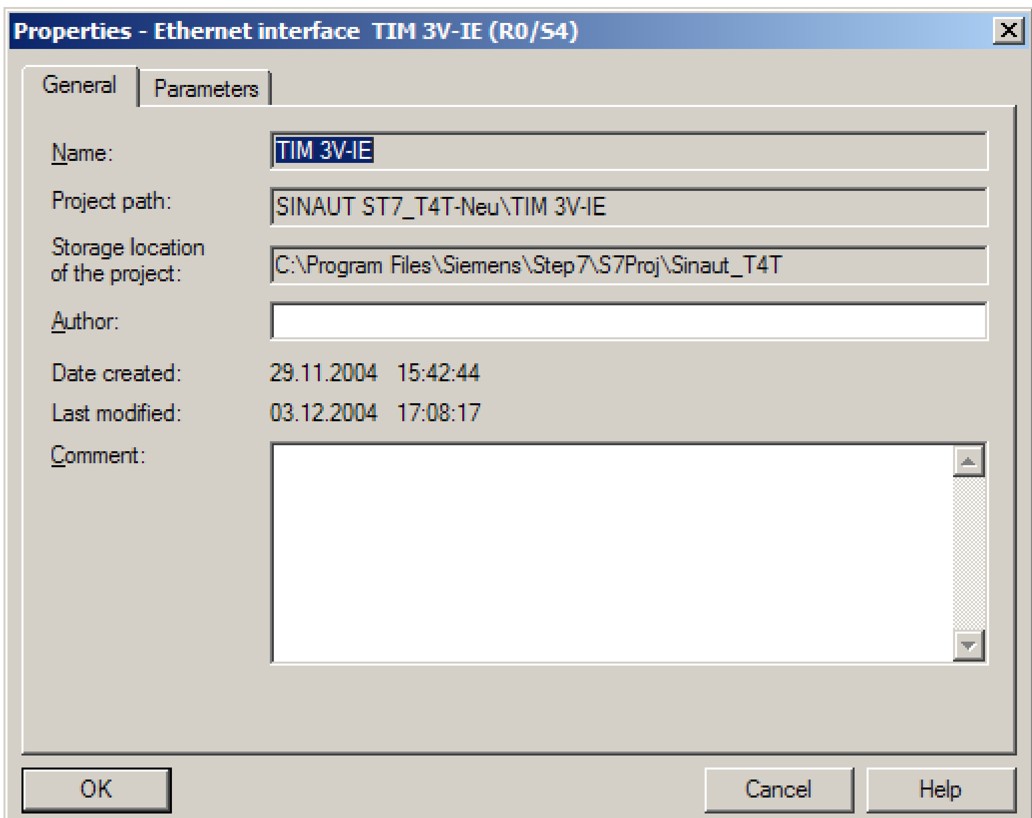


Figure 3-21 *Properties - Ethernet interface* dialog, *General* tab

The *General* tab informs you about the following general parameters of the Ethernet interface.

- Name:
  The *Name* box displays the name of the module.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.

- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Parameters tab

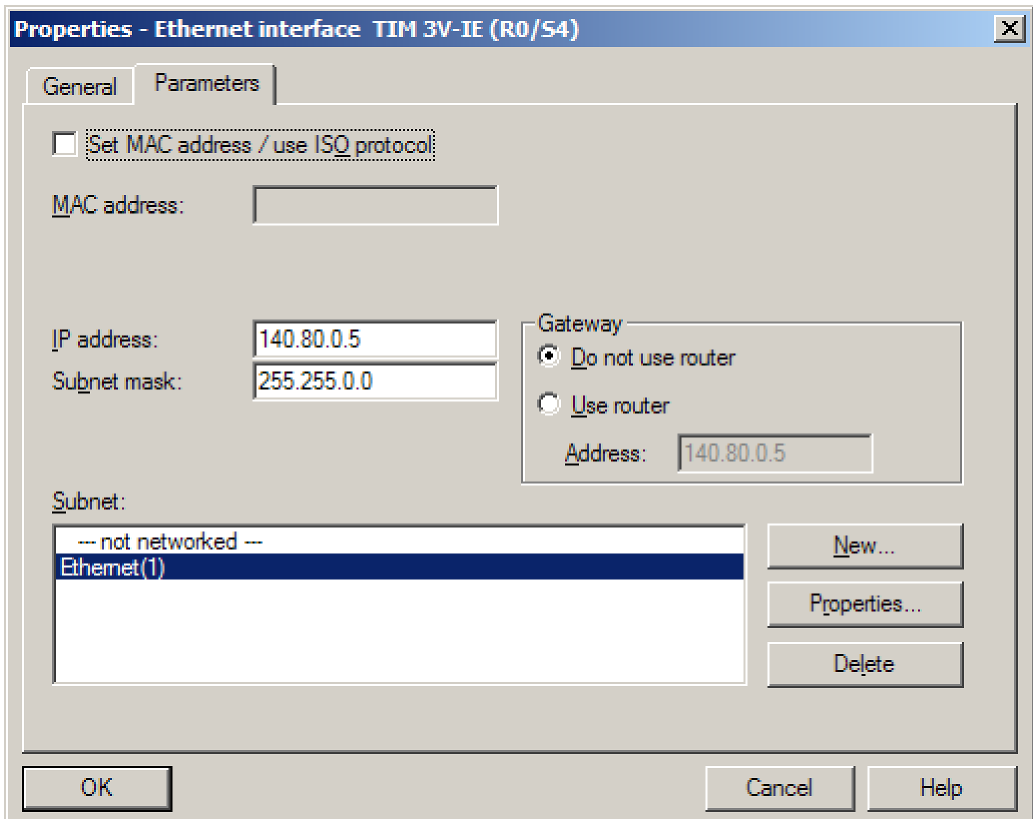


Figure 3-22 *Properties - Ethernet interface* dialog, *Parameters* tab

The *Parameters* tab provides the following parameters:

- *Set MAC address / use ISO protocol*:
  Since an Ethernet TIM does not use the ISO protocol but TCP/IP, this option remains disabled.
- The *IP address*:
  This cannot be changed here.
- The *Subnet mask*:
  This has the default value 255.255.0.0 and cannot be changed, in other words restricted here.

- In the *Gateway* box, you have the option of specifying whether data transmission is over a router.
  If a router exists, the IP address of the router is entered in the *Address* box.
- The *Subnet*:
  This lists all the networks of this type in the project. If the subscriber is not connected, the row *----not networked----* is shown as selected in the *Subnet* list.
  - If the current subscriber is connected, the row of the relevant network is shown as selected. You can set parameters for the current network with the *Properties* button.
  - If no connection exists, a network connection can be set up using the *New* button.
  - An existing connection can be deleted with the *Delete* button.

3.3.6 Configuration of SINAUT networks

You set parameters for SINAUT networks (WANs) in the *Properties - SINAUT Dedicated Line* or in the *Properties - SINAUT Dial-up Network* dialog. The parameters to be set in the following tabs always apply to the entire network and are identical for all attached network nodes or communications partners:

- *General*
  tab with general information and for modifying the module name or adding comments
- *Network settings*
  for setting the communications parameters of the current SINAUT network
- *Time Service*
  tab for setting parameters for time synchronization on the SINAUT network
- *Node List*
  tab with the list of all subscribers on the current SINAUT network
- *Time Slots* tab (only for the corresponding polling mode)
  to specify the time slots for polling

General tab

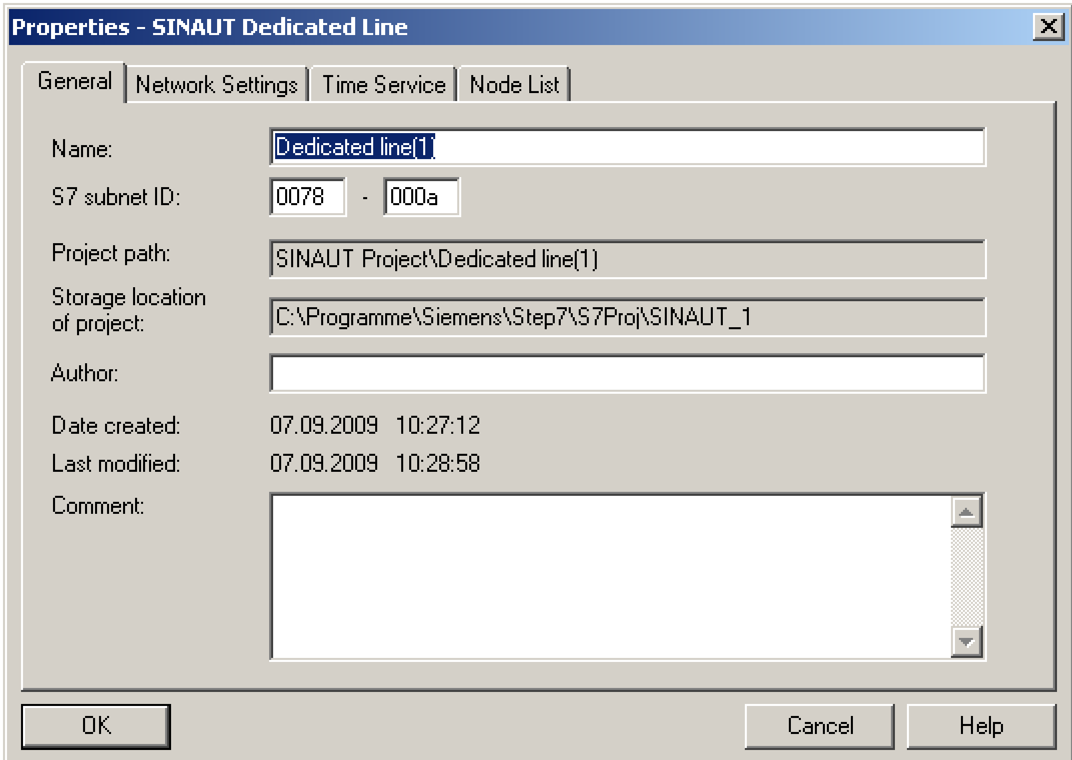


Figure 3-23 *Properties - SINAUT Dedicated Line* dialog, *General* tab

The following parameters are available in this tab:

- *Name*:
  The default entry in the *Name* input box is the default name of the network. You can change this to suit your purposes. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- *S7 subnet ID*:
  The S7 subnet ID is made up of two numbers, one for the project and one for the subnet separated by a dash.
  If you want to go online with a PG without a consistent project, you must know the subnet ID. The subnet ID is also printed out when you print the network configuration.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Network Settings tab

The *Network Settings* tab specifies the basic communications parameters for the current network.

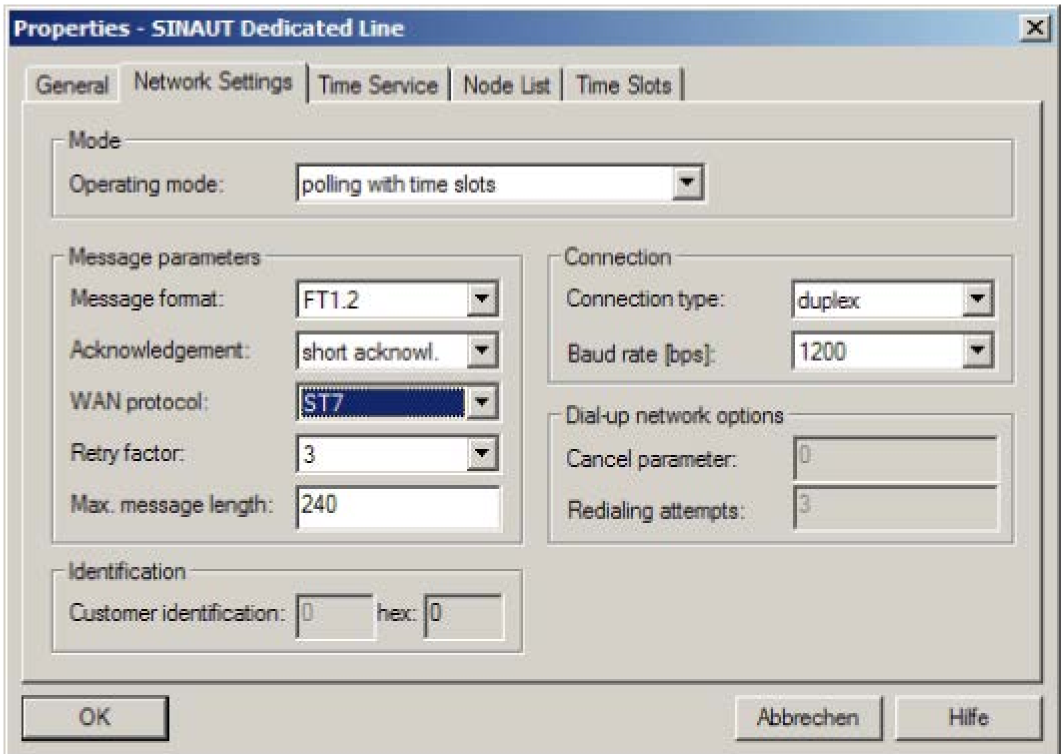


Figure 3-24 *Properties - SINAUT Dedicated Line* dialog, *Network Settings* tab

The following parameters are available:

- The *Operating mode* list box in the *Mode* area with the options:
  - Polling (dedicated lines only):
    In polling mode that is used with dedicated lines, the data exchange is controlled by the master TIM. This polls the connected stations and node stations one after the other. Stations with data to transmit send it as soon as they are polled. Stations that do not currently have any data acknowledge the poll. Only data to be sent from the master TIM to the stations can be transferred at any time between two individual polls.
  - Polling with time slots (dedicated lines only):
    This mode differs from the polling mode because time slots are defined in which the polls take place. Each minute is divided into a selectable number of time slots. When using external wireless networks, the number of time slots per minute is generally prescribed by the relevant regulatory bodies. You configure the time slots to be used in the *Time Slots* tab.
  - Multi-master polling with time slots (dedicated lines are only):
    In this mode, the polls originate from several masters, once again in time slots. Here, various masters can be assigned different time slots for their polling.
    Multi-master polling with time slots is available only for ST7 networks with the FT2

message format with long acknowledgment. You configure the time slots in the *Time Slots* tab.

– Spontaneous (dial-up networks only):
Spontaneous mode is intended for data exchange in the public telephone network, the ISDN network, or the GSM network. Only the TIM with an important data change transfers its data spontaneously and waits for an acknowledgment from the partner. Prior to the actual data transfer, the TIM must first establish a dial-up connection to the partner. Following successful transmission of the data, the TIM waits for the acknowledgment. Following this, the dial-up connection is terminated immediately again if the partner does not use the existing connection to transfer any existing data.

In the *Message parameters* area, you can set the following parameters:

- *Message format*:
  The message format corresponds to IEC 870-5-1. The selection FT1.2 or FT2 depends on the modem. The standard modems MD2 - MD4 can handle both message formats, the GSM modules M20, TC35 and MC45 only FT2.
  Default: FT1.2

  – FT1.2 (8E1):
  Character format 8 data bits,
  even parity, 1 start bit, 1 stop bit
  Modem setting: Data format 11 bits

  – FT2 (8N1):
  Character format 8 data bits,
  no parity, 1 start bit, 1 stop bit
  Modem setting: Data format 10 bits

  Note

  For more detailed information on setting the message format, refer to the section on installing and putting a SINAUT modem into operation in the description of the displays and the connectors accessible from above.

- *Acknowledgment*:
  The type of an acknowledgment does not depend on the modem used. It is set dependent on the quality of the transmission line.
  Default: short acknowl.

  – short acknowl.:
  consists of one byte.

  – long acknowl.:
  consists of 5 bytes. The long acknowledgment is advisable in applications when interference produces spurious characters on the transmission line that could be interpreted by the TIM as a short acknowledgment.
  example: Bad wireless link

- *WAN protocol*:
  The WAN protocol is configured for the specific network.
  Default: ST7
  - ST7:
    The ST7 protocol is used as the WAN protocol. They should be the setting in all purely ST7 networks of an ST7 installation.
  - ST1:
    The ST1 protocol is used as the WAN protocol. This is required for communication in networks with old ST1 installations and in networks in which both ST7 subscribers and ST1 subscribers are connected.

  In a SINAUT dedicated line and dial-up network, either the ST1 or the ST7 protocol can be used. The selected network protocol applies to all subscribers connected to the relevant network or subnet.

  The following combinations of WAN protocol and mode with the dependent frame format are permitted:

Table 3- 1 Overview of the permitted WAN protocols with various modes (ST7 only)

Network type	Mode	WAN protocol	Asynchronous characters	Message format
Dedicated line / wireless network	Polling	ST7	11 or 10 bits	FT1.2 or FT2
Dedicated line / wireless network	Polling with time slots	ST7	11 or 10 bits	FT1.2 or FT2
Dedicated line / wireless network	Multi-master polling with time slots	ST7	10 bits	FT2
Dial-up network	Spontaneous	ST7	11 or 10 bits	FT1.2 or FT2

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

- *Retry factor*:
  This value decides how often a message that has not been acknowledged positively is repeated.
  Range of values: 0 .. 15
  Default value in dedicated lines and in the spontaneous network: 3
  Default in the dial-up network: 7
- *Max. Message length* (in bytes):
  The maximum message length is based on the longest ST7 message length within a network. Time values (for example sender retry time) for internal monitoring functions are derived from this information.
  Range of values: 40 .. 240
  Default: 240

Note

The default of 240 for the maximum message length should, whenever possible, not be changed since PG routing will not work with a maximum message length less than 240.

- *Connection type*:
  Range of values: Half duplex, duplex
  Default: Duplex
- *Baud rate*:
  This is the speed at which the TIM and modem communicate. The baud rate in the current network is decided by the modem and is normally identical to that of the modem. If you want to operate the modem at a speed that is not one of the default speeds, set the next higher speed here (for example 19200 bauds with the modem operating at 14400 bauds).
  – Range of values on dedicated lines and in the spontaneous network:
    300, 600, 1200, 2400, 4800, 9600, 19200, 38400 bauds

    - TIM 4: Also 50, 100, 150, 200 bauds

    - TIM 4R-IE: Also 115200 bauds
  – Range of values in the dial-up network:
    1200, 2400, 4800, 9600, 19200, 38400 bauds
  – Default: 1200 bauds
- *Cancel parameter* (with dial-up networks only):
  This is the number of attempted dialing attempts until the attempt is finally aborted.
  Range of values: 0 ... 127
  Default: 0
  – Cancel parameter = 0:
    The call attempts are finally aborted when a connection was established 127 times in a row but no data could be transferred.
  – Cancel parameter = 1 ... 127:
    The call attempts are finally aborted when a connection was unsuccessful n times in a row, regardless of whether a connection could be established at all or whether data could not be transferred on an established connection.
- *Redialing attempts* (with dial-up networks only):
  This is the number of attempted calls until a disruption is reported.
  Range of values: 1 ... 127
  Default: 3
- *Customer identification* (with dial-up networks only):
  The customer identification is used to specify whether connections can only be established to partners permitted for the network. The customer identification has the function of password protection in the relevant network.
  Range of values: 0 ... 65535
  Default: 0

  For more information on SINAUT ST1, refer to release 05/2007 of the manual.

Time Service tab

The *Time Service* tab specifies the extent to which time synchronization services will be executed by the master or in the case of a subnet by the node station in this network. The time synchronization for TIM modules is described in the tab with this name in the properties dialog of the TIM, see section Configuring TIM modules (Page 92).

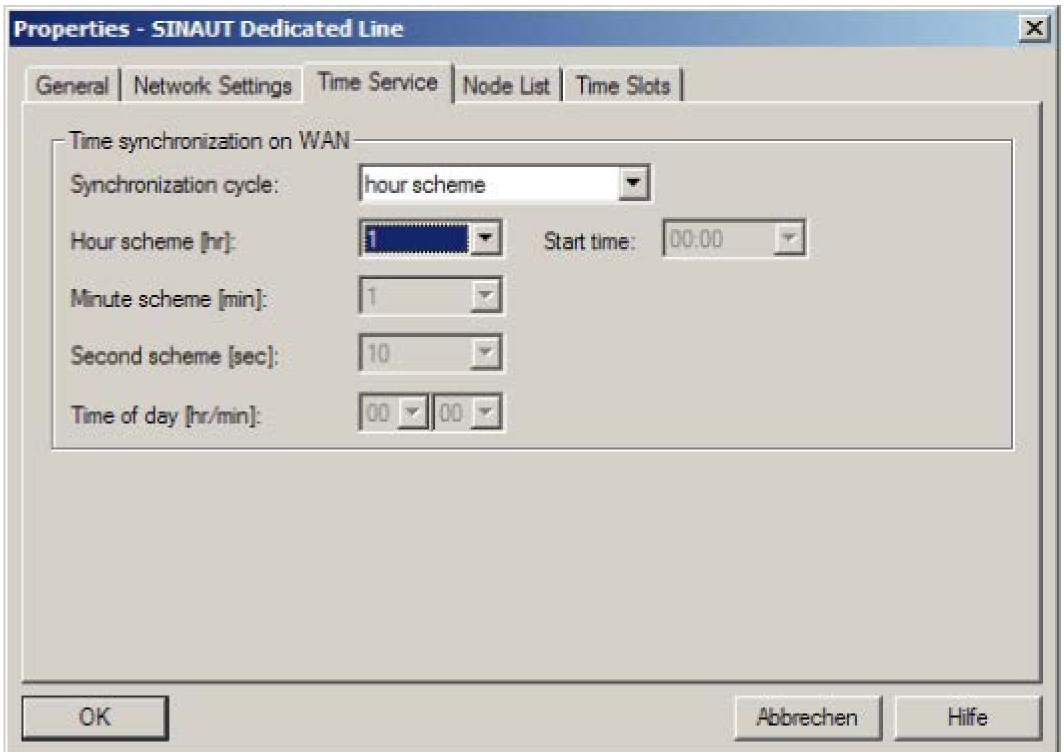


Figure 3-25 *Properties - SINAUT Dedicated Line* dialog, *Time Service* tab

You can set the following options for the *Synchronization cycle* parameter:

- No synchronization:
  There is no time synchronization on the relevant network.
- Hour scheme:
  The number of hours between synchronization activities can be set in the "Hour scheme" drop-down list box.
  - Start time:
    If the cycle for time synchronization is longer than 1 hour, you can set a start time for time synchronization in the "Start time" drop-down list box.
- Minute scheme:
  The number of minutes between synchronization activities can be set in the "Minute scheme" drop-down list box.

- Second scheme:
  The number of seconds between synchronization activities can be set in the "Second scheme" drop-down list box.
- Time of day:
  Synchronization takes place once a day. Set the time of day for the synchronization in the "Time of day" drop-down list box (for example 01:00).

On dedicated lines, a synchronization cycle of 1 hour is recommended and in dial-up networks, once a day, for example at 01:00 a.m.

In dial-up networks, time synchronization can also be used to check the availability of a subscriber and to fetch data.

Which TIM is synchronization master on this WAN is decided automatically based on the configuration of the network attachments in the properties dialog of the network node.

Node List tab

The node list displays all the communications subscribers connected to the current network; in other words, TIM modules. It also lists the station name, the WAN address and the configured node type making it easy to check these parameters throughout the network.

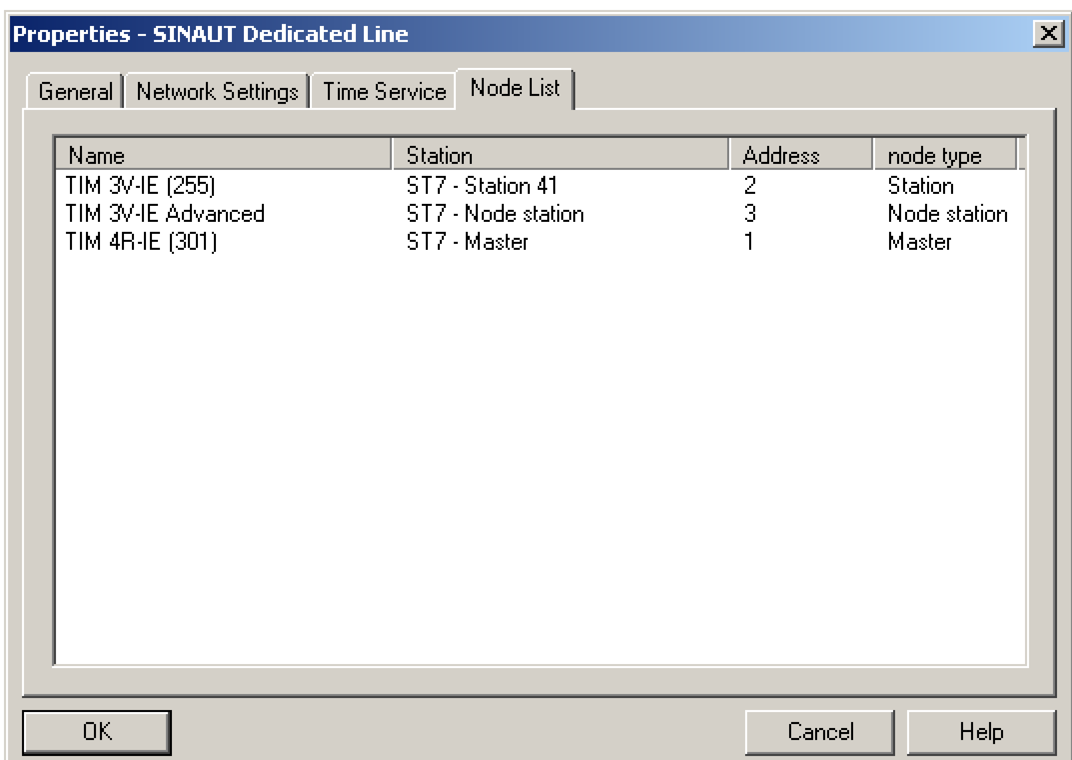


Figure 3-26 *Properties - SINAUT Dedicated Line* dialog, *Node List* tab

Note

If there is an asterisk after one or two addresses, the WAN address is not unique and must be changed.

Time Slots - Only in networks with a suitable polling mechanism

In the *Time Slots* tab, you can specify which time slots are used for transmission in the *Polling with time slots* and *Multi-master polling with time slots* modes.

To allow communication, at least one time slot must be selected since the master can only poll within the selected time range.

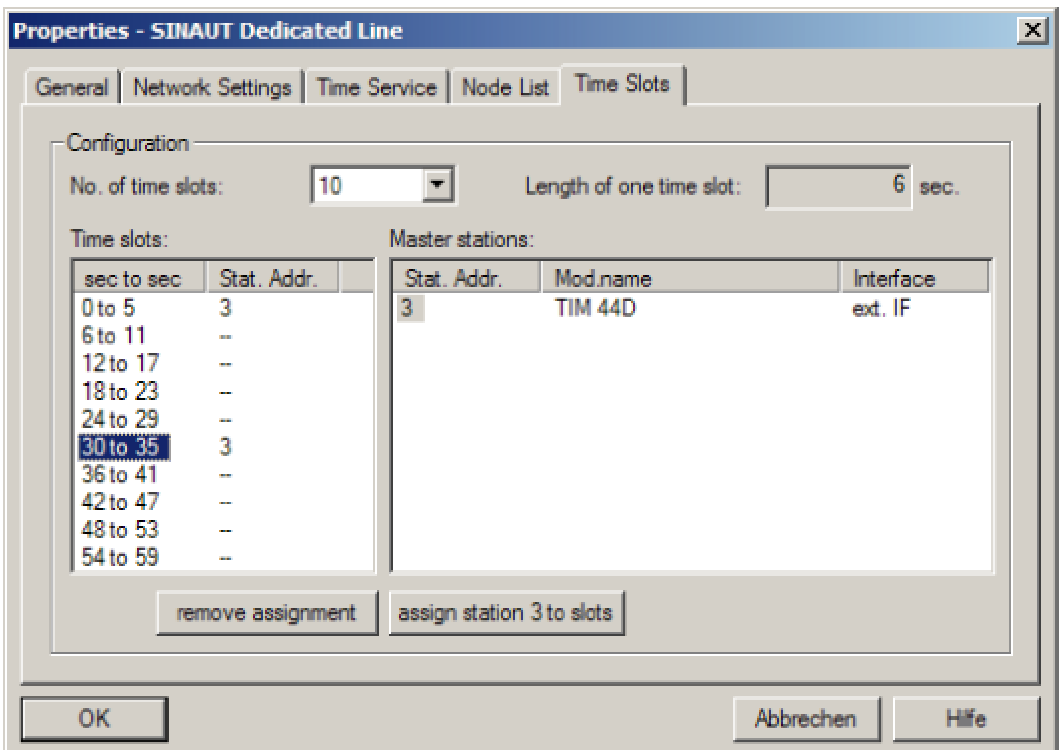


Figure 3-27 *Properties - SINAUT Dedicated Line* dialog, *Time Slots* tab

When setting the parameters for time slots, follow the steps below:

1. Specify the *No. of time slots* per minute. The *Length of one time slot* is calculated from this.
2. In the *Master stations* list, in the *Stat. Addr. column*, select the station address of the master station of the subnet.
3. In the *Time slots* list, in the *sec to sec* column, select the required time slot for calling this master station.
4. Click the *assign station # to slots* to assign the selected time slot to the selected master station.

Repeat this procedure if you want to use more than one time slot per minute.

In the multi-master polling with time slots mode, this must be done for each of the polling masters with different time slots.

You can delete a selected time slot again with the *remove assignment* button.

Note

Master TIM

The time slot method can now also be configured with the "SINAUT ST7 configuration software for the PG/PC" as of Version V4.2 for a master TIM without DCF7 receiver when a TIM 4R-IE with an Ethernet connection to an ST7cc-/ST7sc PC is used as the master TIM.

3.3.7 Configuring WAN network nodes

You set parameters for SINAUT WAN networks in the *Properties - SINAUT Dedicated Line* or in the *Properties - SINAUT Dial-up Network* dialog in the following tabs:

- *General*
  tab with general information on the network node and entry of comments
- *Network Connection*
  tab for setting the most important network properties
- *Basic Param.*
  tab for setting the basic communications parameters
- *Dedicated Line*
  tab with parameters specifically for dedicated lines
- *Dial-up Network*
  tab with parameters specifically for dial-up networks
- *Call Parameters*
  tab with parameters specifically for call numbers
- *AT Initialization*
  tab for setting special AT strings when they are required

The relevant tabs are displayed depending on the network type.

Note

When setting parameters for network nodes, only the parameters that are practicable for the particular combination can be modified. This depends on the following:

- Network node type specified in the *Properties Network Node* dialog, *Network Connection* tab
  and the
- Operating mode specified in the *Properties Network* dialog, *Network Settings* tab.

General tab

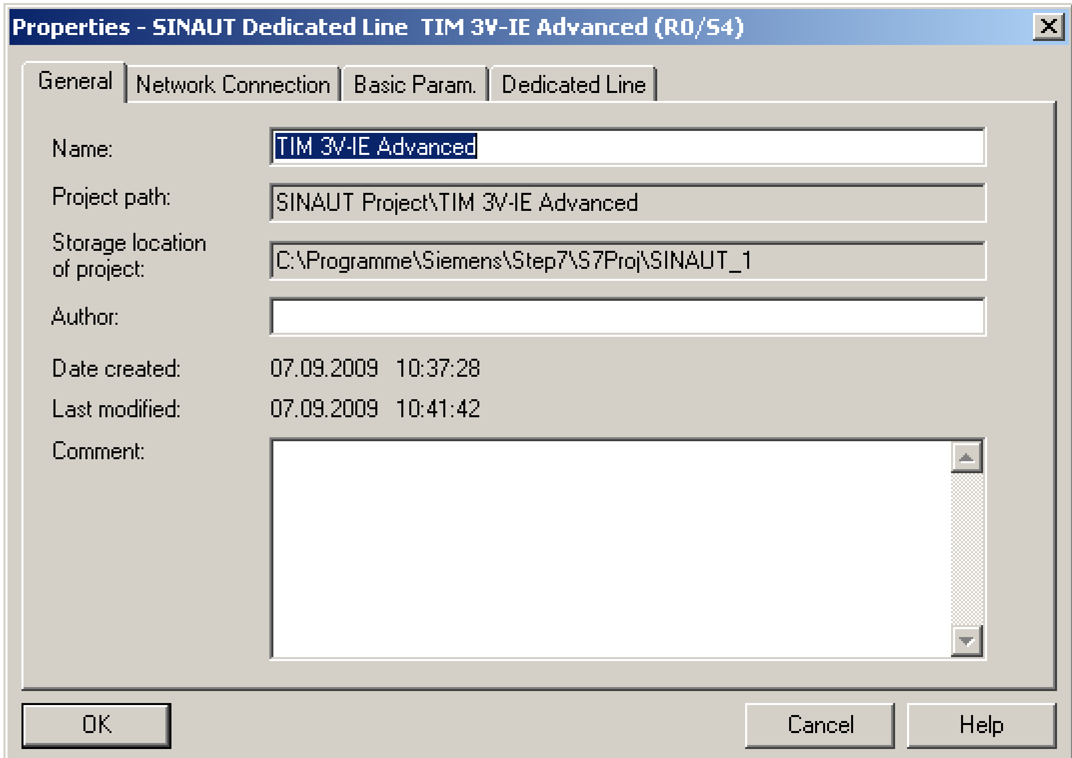


Figure 3-28 *Properties - SINAUT Dedicated Line TIM* dialog, *General* tab

The following parameters are available in the *General* tab:

- The *Name* box displays the name of the module in SIMATIC stations. You can only change the default interface name in SIMATIC PC stations and other stations. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Network Connection tab

The *Network Connection* tab allows you to set the most important networking properties of the WAN network node.

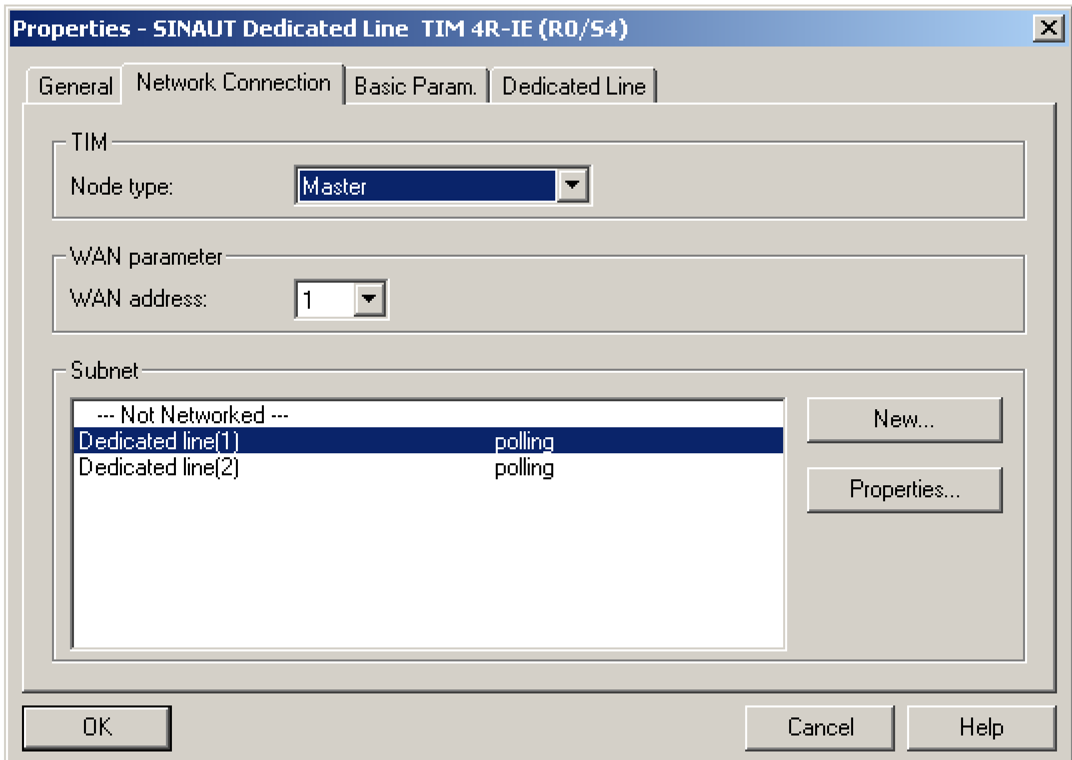


Figure 3-29 *Properties - SINAUT Dedicated Line TIM* dialog, *Network Connection* tab

The parameters here are:

- The *node type* selected in the list box:
  - The master station is the highest hierarchic level in the network. It generally collects information from the underlying network nodes and specifies settings for the nodes in the field.
  - A node station is at a hierarchically lower level than the master station or another node station and is at a higher level than one or more other stations.
  - A station is at a level close to the field and hierarchically below a master station or node station.
- The *WAN address* in the network.
  The unique WAN address of the node can be modified. A consistency check ensures that WAN addresses that have already been assigned cannot be selected.

  For more information on SINAUT ST1, refer to release 05/2007 of the manual.

- The *Subnet*:
  This lists all the networks of this type in the project. If the subscriber is not connected, the row *----not networked----* is shown as selected in the *Subnet* list.
  - If the current subscriber is connected, the row of the relevant network is shown as selected. You can set parameters for the current network with the *Properties* button.
  - If no connection exists, a network connection can be set up using the *New* button.
  - An existing connection can be deleted with the *Delete* button.

Basic Param. tab

The *Basic Param.* tab contains the communications parameters for the selected WAN node.

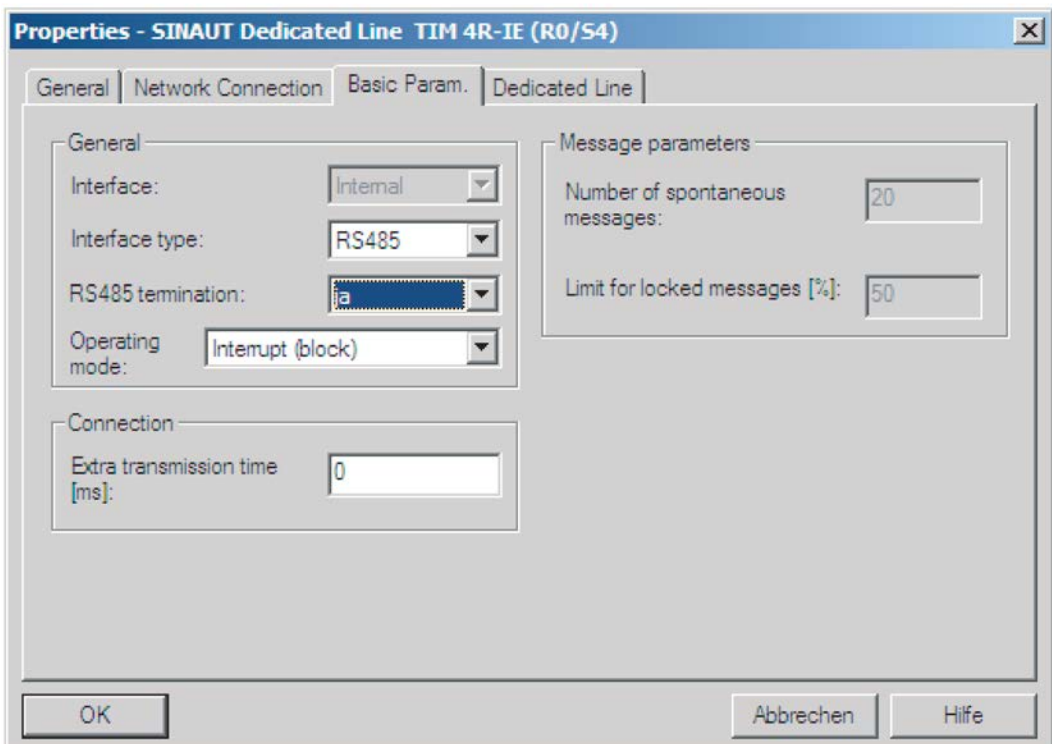


Figure 3-30 *Properties - SINAUT Dedicated Line TIM* dialog, *Basic Param.* tab

The following parameters are available:

- *Interface*:
  This output box indicates whether the current node is operated on the internal or the external WAN interface of the TIM.
- *Interface type* (TIM 4R-IE only):
  Here, you set the type of interface: RS-232 or RS-485 mode
- *RS-485 termination* (TIM 4R-IE only):
  Activation of the terminating resistor for the RS-485 bus

Note

If the interface is set to "RS-485", you will need to set the internal terminating resistor of the module for the RS-485 bus.

If the TIM 4R-IE is the first subscriber on the RS-485 bus, which is normally the case, select the setting "yes".

If the TIM 4R-IE is not the first or last subscriber on the RS-485 bus, select "no".

- *Operating mode*:
  This setting specifies whether the interface connected to the current node will be operated in interrupt or in DMA mode. Only one of the two interfaces of a TIM module may be operated in DMA mode.
  Range of values: Interrupt (block), DMA, Interrupt (single characters)
  Default: Interrupt (block)
  - Operating mode = Interrupt (block)
    This operating mode applies to the transmit and receive direction.
    The default mode Interrupt (block) is suitable for all connections. Four characters are transferred per block. Following this, there is an interrupt. The received characters are checked only after a complete message has arrived.
  - Operating mode = DMA
    This operating mode applies to the transmit and receive direction.
    The DMA mode should be used for connections with a high baud rate or heavy message traffic, however not for GSM networks.
    Only one of the two interfaces of a TIM module may be operated in DMA mode.
  - Operating mode = Interrupt (single characters)
    This operating mode is used only in the receive direction. In the transmit direction, the block mode continues to the used.
    This interrupt mode is suitable for extremely bad lines. An interrupt is triggered per transmitted character and each character this analyzes immediately after it is received allowing extremely good diagnostics of transmission errors. This mode is more reliable than the block mode but is slower.
- *Extra transmission time*:
  This is an offset added to the transmit retry time. The send retry time is calculated automatically on the TIM.
  From the *Extra transmission time* parameter, the character delay time can also be calculated (character delay time = extra transmission time divided by 5).
  An offset time should be entered in the *Extra transmission time* input box, for example when the send retry time cannot be calculated completely as is the case with satellite transmissions or wireless links over repeaters.
  Range of values: 0 .. 65535 ms
  Default when using the MD720 or TS modules: 1000 ms
  Default when using M1 or M20 modules: 400 ms
  Default when using the other modems: 0 ms

- *Number of spontaneous messages*:
  This function is available only for the station and node station node types.
  Range of values: 0 .. 255
  Default for dedicated lines: 20
  Default for dial-up networks: 200
  - Number = 0 in polling mode:
    At the time of the first Polling message, all spontaneous messages pending are transferred.
  - Number = 1 .. 255 in polling mode:
    Maximum 1-255 spontaneous messages pending at the time of the first Polling message are transferred.

  In a dial-up station, the *Number of spontaneous messages* parameter decides after how many messages the master station has the opportunity of transferring its pending messages to the station.
- *Limit for locked messages*:
  This parameter can only be set for dial-up networks and specifies the maximum percentage of locked messages in the send buffer. If this percentage is exceeded, the image method will be used for all new locked messages arriving. This prevents an overflow of the send buffer.
  Messages are marked as locked if they can no longer be transferred to the addressees due to communications problems (known as the data brake).
  Range of values: 0 ... 90% (If 0 is entered, the default setting is used)
  Default: 50%

Dedicated Line tab

The *Dedicated Line* tab contains special parameters required only when using dedicated lines.

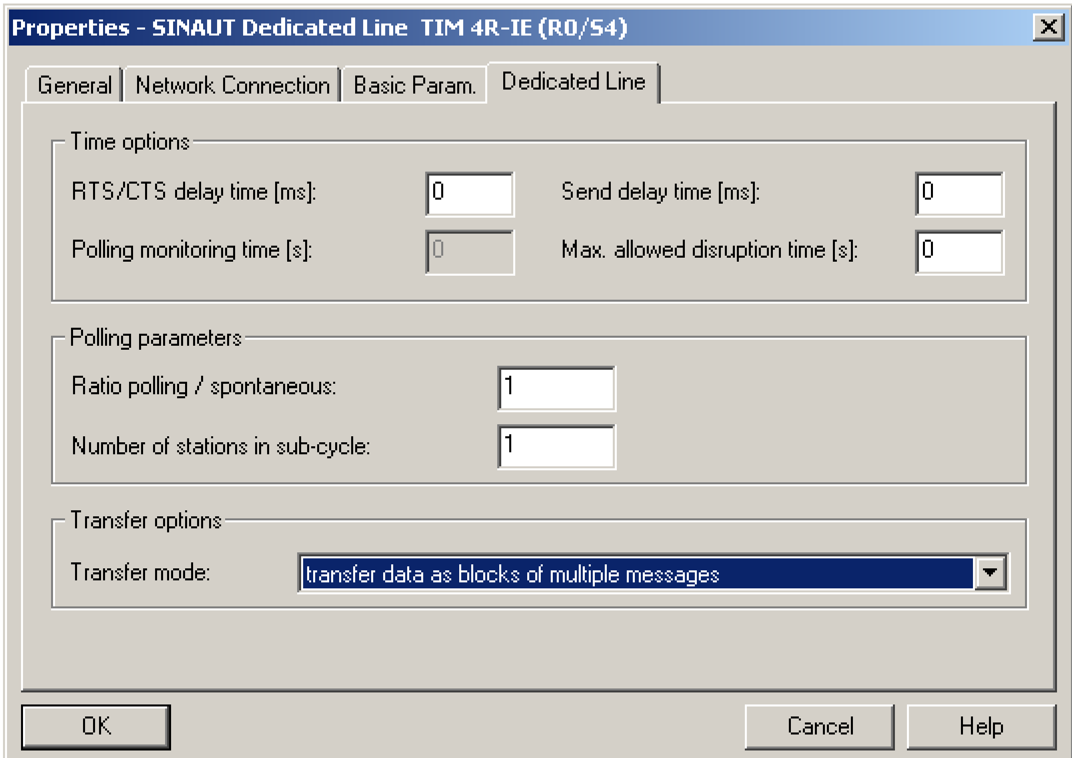


Figure 3-31 *Properties - SINAUT Dedicated Line TIM* dialog, *Dedicated Line* tab

The parameters for dedicated lines include:

- *RTS/CTS delay time*:
  Setting the RTS/CTS delay time is required, for example when connecting a modem to the RS-485 interface of the TIM module. The values necessary for the RTS/CTS delay time can be found in the descriptions of the modems.
  Range of values: 0 .. 65535 ms
  Default: 0
  - RTS/CTS delay time = 0:
    After setting the RTS signal, transmission only starts when the CTS signal was set by the modem.
  - RTS/CTS delay time > 0:
    Transmission is not delayed until the CTS signal of the modem. After the RTS signal has been set, transmission is delayed for the selected time and then started immediately.
- *Polling monitoring time*:
  Specifies the latest time after which a station or node station TIM expects to be polled. If the TIM is not called after this time, it sends a message to its local CPU indicating that the master station is disrupted.

Range of values: 0 .. 65535 s
Default: 0 (0 means no monitoring)

Even if no monitoring is set here, the TIM module registers the message traffic over the WAN and automatically sends the same fault message to its CPU if it does not register any message traffic for several seconds. If a maximum message length of 240 bytes is set, a with a retry factor of 3 and a transmission rate of 9,600 Bd, the message is sent after approximately 4 seconds without message traffic, and at a transmission rate of 1,200 Bd, after approximately 32 seconds.

- *Send delay time*:
  The send delay time is used only when the CTS signal comes from the modem (RTS/CTS time delay parameter = 0). As soon as the CTS signal comes from the modem, the send delay time is started. Data transmission is started only after this time elapses.
  This parameter is required, for example, when additional offset times are required to allow repeaters to start up on wireless links prior to starting data transmission.
  If 0 is entered, no send delay time is used.
  Range of values: 0 .. 65535 ms
  Default: 0
- *Max. allowed disruption time* (Ethernet TIMs only):
  Here, you can enter the tolerance time for a connection disruption detected by the TIM. 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.
  Range of values: 0 ... 255 seconds
  Default: 0

  If disruptions occur frequently in networks (for example in some wireless networks), it may be helpful to increase the allowed disruption time without increasing the repetition factor for messages (see also properties dialog *Dedicated line*, *Network settings* tab). Increasing the allowed disruption time delays signaling of station failures and so reduces the number of organizational messages when stations return.
- *Ratio polling / spontaneous*:
  This output box displays the number of spontaneous messages that can be sent by a master station between two polls.
  Range of values: 0 .. 255
  Default: 1
- *Number of stations in sub-cycle*:
  This output box displays how many stations in the sub-cycle should be polled per main cycle.
  Range of values: 0 .. 250
  Default: 0

  The schematic shows a configuration with stations in the main and sub-cycle and the resulting polling order if 1 is set for the *Number of stations in sub-cycle* parameter.

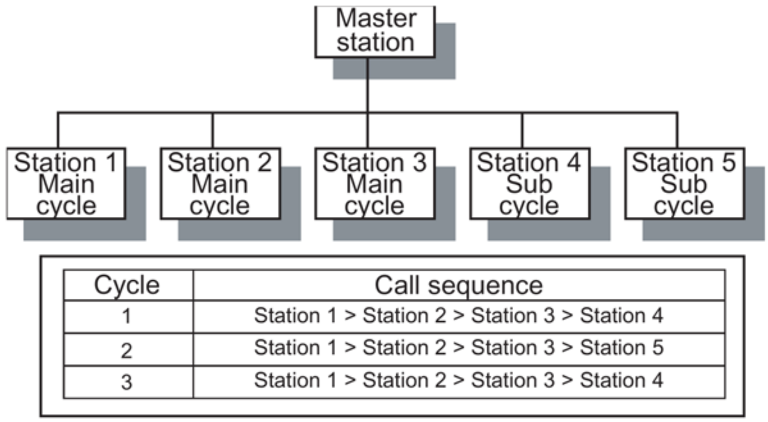


Figure 3-32 Main cycle - sub-cycle

- *Transmission mode:*:
  This parameter specifies the form in which ST7 data messages are sent when using the ST7 protocol.
  Range of values:
  - transfer data messages of single messages
  - transfer multiple data messages as a block
  Default: Single messages

Dial-up Network tab

The *Dial-up Network* tab contains special parameters required only when using dial-up networks lines.

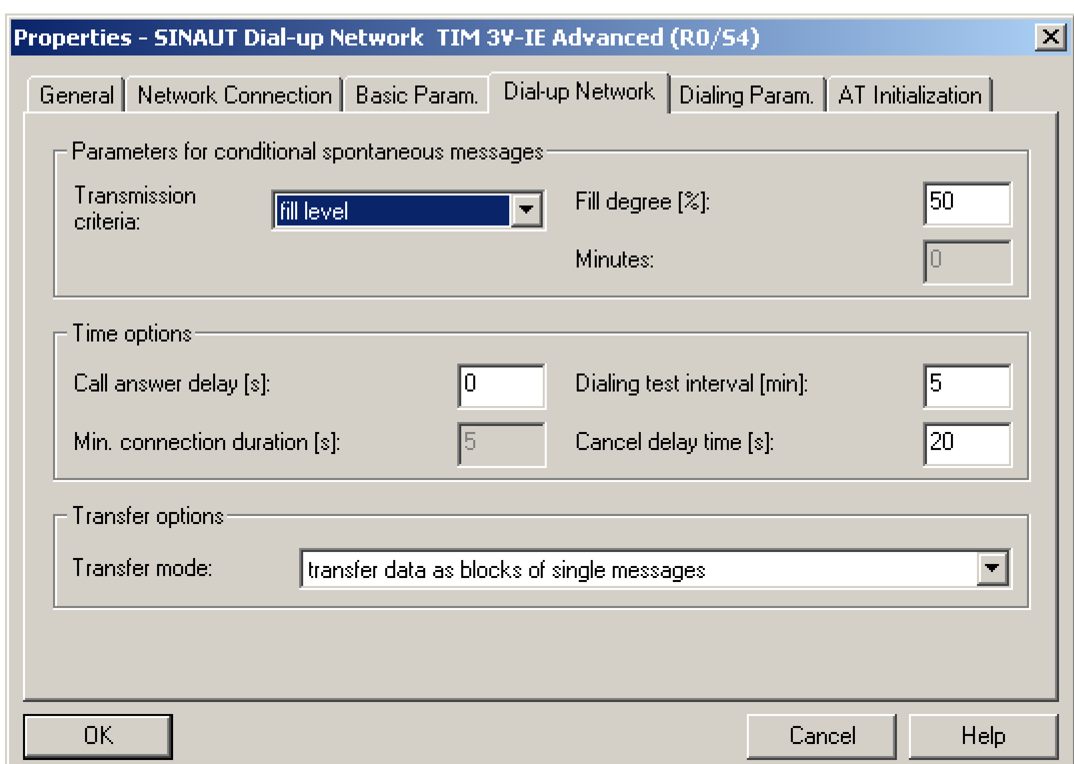


Figure 3-33 *Properties - SINAUT Dedicated Line TIM* dialog, *Dial-up Network* tab

The dial-up network parameters include:

- *Transmission criteria*:
  This setting controls connection establishment for the transmission of conditional spontaneous messages. The transmission criterion for conditional spontaneous messages can only be set for stations and node stations.
  Range of values: Standard conditions, fill level, time of day, time scheme
  Default: Standard conditions
  - Standard conditions:
    A connection is not established for existing conditional spontaneous messages. The conditional spontaneous messages are sent only when a connection is established for sending unconditional spontaneous messages or a buffer overflow is threatening or the connection is established by the other end.
  - Fill level:
    When the send buffer is filled to the specified level with conditional spontaneous messages, the TIM module automatically attempts to establish a connection and to transmit the messages.
    - Input box *%*: Entry of the send buffer fill level as a percentage (default: 50%)
  - Time of day:
    A connection is automatically established and the messages sent at the specified time of day. The time must be entered:
    - Input box *Hours*: Entry of the time (hour)
    - Input box *Minutes*: Entry of the time (minute)
  - Time scheme:
    A connection is automatically established and the messages sent at the specified time intervals. The interval must be entered:
    - Input box *Hours*: Entry of the hour value for the send interval
    - Input box *Minutes*: Entry of the minute value for the send interval
- *Call answer delay*:
  This sets the time that the WAN driver waits before answering an incoming call. This allows time to answer a telephone call if a telephone is attached parallel with the TIM on a shared telephone connection. The value 0 means there is no call answer delay.
  Range of values: 0 ... 60 s
  Default: 0 s
- *Min. connection duration*:
  Here, the minimum connection duration of a dial-up connection can be set. This may be required in fast dial-up networks to be able to wait for the response of subscribers during a GR before the connection is terminated.
  Range of values: 0 ... 65535 s
  Default: 5 s
  0 means that there is no minimum connection duration.
- *Dialing test interval*:
  This specifies the time in minutes for a test interval. A test interval is started when no connection to a particular subscriber could be established from a master TIM after the specified number of retries.
  Following the test interval, the WAN driver automatically starts to to establish a connection to the specified subscriber. If a connection cannot be established, the test interval starts again. If the test interval is running and the WAN driver gets a new message to be sent to the disrupted subscriber, it does not wait for the test interval to end

but attempts to establish a connection immediately and to send the message.
Range of values: 0 ... 255 minutes
Default: 5 minutes

- *Cancel delay time*:
  This parameter specifies how long a dial-up connection is retained when the send buffers of the TIM module are full and it can send no further messages or data to the CPU.
  All messages received over the WAN interface are acknowledged negatively until this time has elapsed. Due to the negative acknowledgment of the previously sent message, the communications partner will repeat the message after the send retry time.
  If the cancel delay time has elapsed, the connection is terminated.
  Range of values: 0 ...255 seconds
  Default: 0
- *Transmission mode:*:
  This parameter specifies the form in which ST7 data messages are sent when using the ST7 protocol.
  Range of values:
  - transfer data messages of single messages
  - transfer multiple data messages as a block
  Default: Single messages

Dialing Param. tab

The *Dialing Param.* tab appears only with dial-up network nodes and includes all parameters specific to call-numbers.

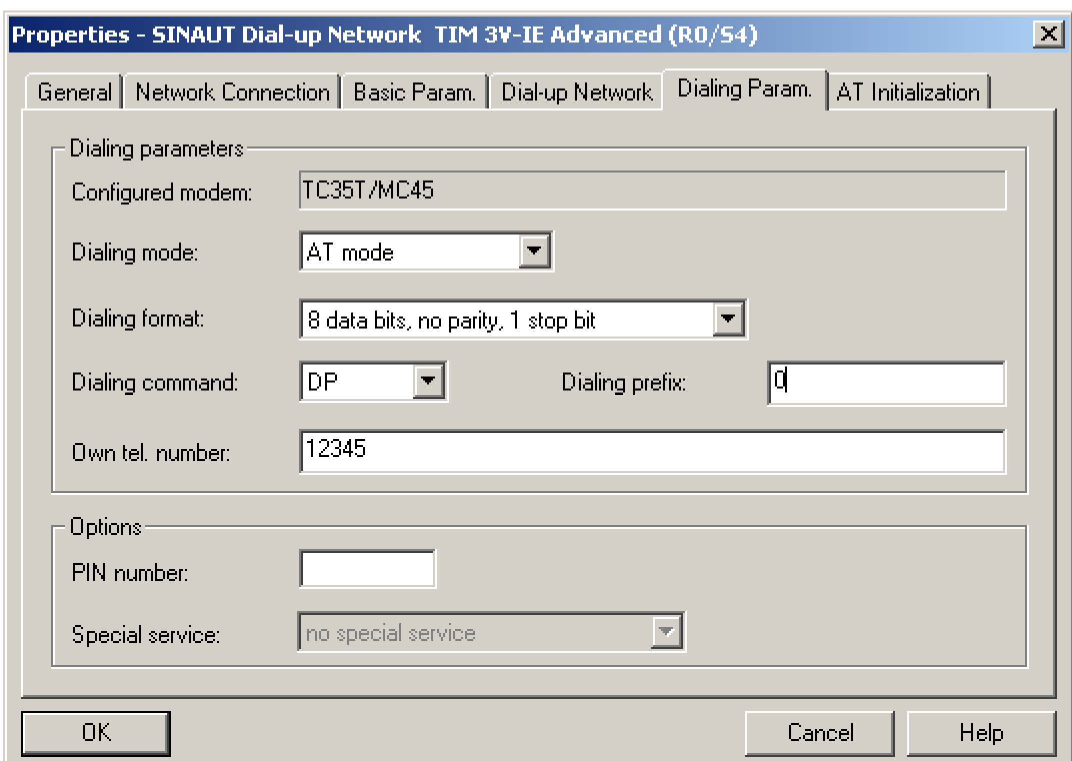


Figure 3-34 *Properties - SINAUT Dedicated Line TIM* dialog, *Dialing Param.* tab

The parameters here are:

- *Configured modem*:
  The modem specified in the hardware configuration is displayed.
- *Dialing mode*:
  Specifies how the modem is controlled. Note that only the AT mode can be used on the internal interface of the TIM. A choice between AT mode and V.25bis is possible only for the external WAN interface.
  Range of values: AT mode, V.25bis
  Default: AT mode
- *Dialing format*:
  The data format of the dial-up phase depends on the type of modem. The following settings are possible:
  8 data bits, no parity, 1 stop bit
  8 data bits, odd parity, 1 stop bit
  8 data bits, even parity, 1 stop bit
  8 data bits, no parity, 2 stop bits
  8 data bits, odd parity, 2 stop bits
  8 data bits, even parity, 2 stop bits
  7 data bits, no parity, 1 stop bit
  7 data bits, odd parity, 1 stop bit
  7 data bits, even parity, 1 stop bit
  7 data bits, no parity, 2 stop bits
  7 data bits, odd parity, 2 stop bits
  7 data bits, even parity, 2 stop bits

  Default:
  - with AT mode: 8 data bits, no parity, 1 stop bit
  - with V.25bis: 7 data bits, even parity, 1 stop bit

  Note

  The data listed above applies only to the dialing phase. The data format in the data phase is set at switch 5 on the MD3 modem.

  It is only necessary to set the dialing format on older modems that do not support "Autoband". With "Autoband", the modem can determine the character and data format in the dialing phase automatically based on the first AT string.

- *Dialing command*:
  This is the dialing command for the local modem. Possible dialing commands are:
  – D (AT command)
  – DP (AT command, pulse dialing)
  – DT (AT command, tone dialing)
  – CRN (V.25bis)
  – CRNP (V.25bis, pulse dialing)
  – CRNT (V.25bis, tone dialing)

  Default: D. This default modem dialing command should be used where possible.

- *Dialing prefix*:
  This is the access number (outside line) for a private branch exchange (typical entry 0 or 9) or for an alternative telephone provider. A number up to 12 digits long can be specified. With direct connection to the dial-up network and without an alternative telephone provider, this parameter can remain empty.
  The dialing prefix can be changed again in the *Properties of subscriber* dialog.
- *Own tel. number*:
  Here, you enter your own telephone number for the network node including the area code. This telephone number can no longer be changed later in the *Properties of subscriber* dialog.

Note

In dial-up networks, in which another subscriber within the same local network cannot be dialed with the local area code, it is advisable to enter your own telephone number in the *Own tel. number* box (without area code) and to specify the area code in *Dialing prefix*.

- *PIN number*:

  For a GSM module, the 4 to 8 digit PIN number must be entered here so that this can be transferred from the TIM module to the module.

  If you have a contract without a PIN, leave the box empty.

Note

If an incorrect PIN is entered, the SIM card in the module might be disabled. If the fault LED lights up during connection establishment, the diagnostics buffer of the TIM must be checked because an entry for a bad PIN is generated here.

- *Special service*:
  The following SMS special services are available:
  - No special service

    The node does not use any SMS service.
  - SMS via fixed network (TAP)

    Sending of SMS messages to an SMS center that uses the TAP protocol
  - SMS via fixed network (UCP)

    Sending of SMS messages to an SMS center that uses the UCP protocol
  - SMS via GSM network

    Sending SMS messages to an SMS center with digital access

Note

If two MD3 modems communicate with each other, they must not be operated in the *1200 baud, half-duplex, AT mode* mode.

AT Initialization tab

The *AT Initialization* tab appears only in dial-up network nodes and when the *AT mode* is selected as *Dialing mode* in the *Dialing parameters* tab. The string stored here is formed automatically from the previously set dial-up network parameters and the SINAUT dial-up modem selected for the network node (MD3, MD4 or GSM modem TC35, MC45, MD720).

If "third-part modem" is set for the network node, no initialization string is displayed. You will then need to enter the correct string for the modem you are using. The initialization string set here is transferred to the modem operating in AT mode only when the TIM starts up.

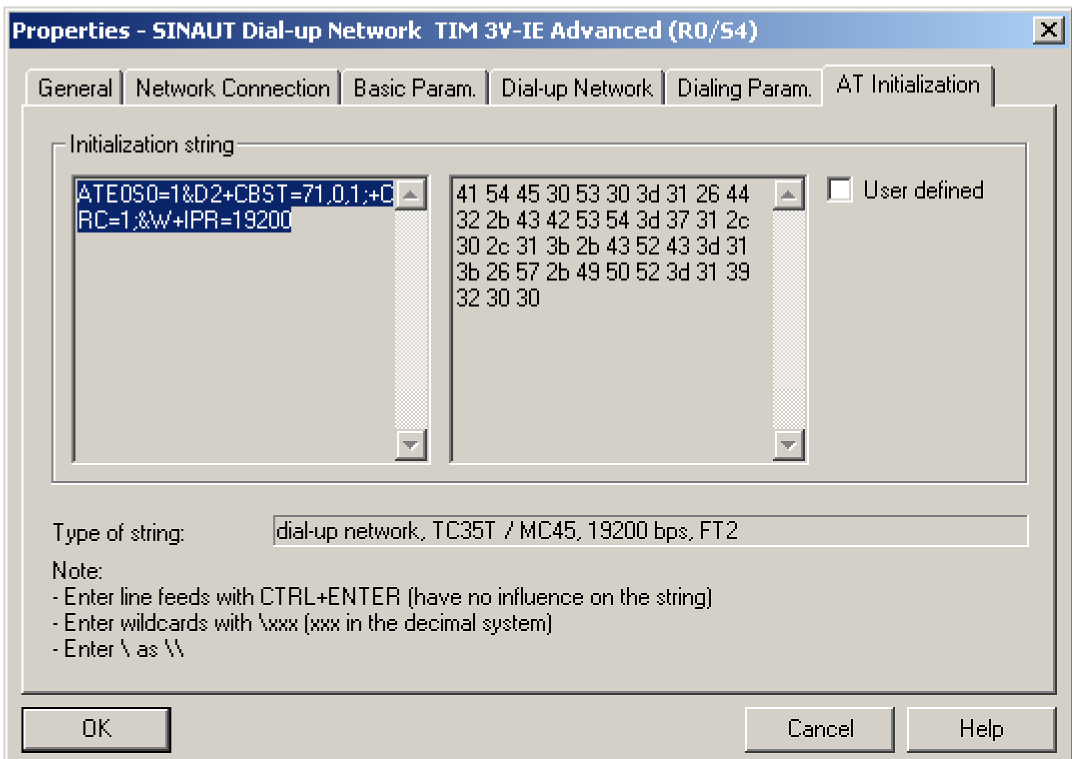


Figure 3-35 *Properties - SINAUT Dedicated Line TIM* dialog, *AT Initialization*

- *Initialization string* input box:
  In the left input box, the AT string is displayed as text and in the right-hand box in hexadecimal notation. No entries can be made in the right-hand box. In the left-hand box, a string can only be entered when either no entry can be found for the current combination of modem and network parameters in the SINAUT modem database or the *User defined* option was set. Otherwise the valid string is taken from the database.
- *User defined* option:
  This allows the manual entry of AT Initialization strings for the basic settings of the modem.
- *Type of string* output box:
  This box displays the content of the current string.

Special features of the MD4 modem

On the MD4 modem, the standard string is *ATS45=83$P1\N0&W$M=n*.
The part string *$P1* defines the V.110 mode at a transmission rate of 9,600 Bd.
The *n* character at the end of the string is the placeholder for the MSN (last digit of the telephone number) of the subscriber.

The stored standard string allows only transmission rates of 9,600 and 19,200 Bd, no other rates are supported. This standard setting does, however, allow communication with the following subscribers:

- Old ISDN modems installed with SINAUT ST1 devices
- GSM stations
- SMS centers

When using MD4 modems in ISDN networks, it is only possible to operate at a transmission speed of 9,600 Bd with the standard setting and not at the maximum speed of 38,000 Bd.

If the maximum speed of 38,000 Bd is to be used with MD4 modems in pure ISDN networks, the *$P1* entry in the standard string must be replaced by *$P5*.
This changes from the V.110 transmission mode to the X.75 mode allowing a transmission rate up to 38,000 Bd.

When using MD4 modems, note the following if you change the telephone number later:

Note

Since the MSN number is automatically added to the generated AT string with the MD4 modem, if you change the telephone number later, remember that the MSN number in the AT string for the MD4 modem may need to be modified manually.

Special features of the TS modules

For dial-up connections via TS modules, AT strings are configured in the "AT Initialization" tab of the Properties dialog of the dial-up network node. Depending on the TS module you are using, the following AT strings are the defaults:

- CP with TS Module GSM

  ```
  ATE0S0=1&D2+CBST=7,0,1;+CRC=1;&W+IPR=115200
  ```

- CP with TS Module Modem

  ```
  AT&FE0&M0&Q6S0=1x3&w0
  ```

- CP with TS Module ISDN - partner (master station) with GSM connection:

  AT string of the TS Module ISDN in the station:

  ```
  AT&FE0\N1
  ```

Only when transferring via an ISDN network with the following configuration do the AT strings need to be adapted manually in the "Initialization string" input box.

- CP with TS Module ISDN - partner (master station) withModem MD4:
  - At string for the TS Module ISDN in the station:

    `AT&FE0\N2`
  - At string for the Modem MD4 in the master station:

    `ATS45=85$P5\N0&W$M=1`

3.3.8 Plausibility check of the network configuration

A plausibility check of the configured network is performed either when the network is stored with the *Save and Compile...* function or when the consistency check is started directly from the *Network / Check Consistency* menu. The following is reported:

- Subscribers not connected to a any subnet
- Subnets with only one subscriber
- Inconsistent connections, for example due to the wrong WAN protocol

The following are also checked for SINAUT networks:

- Compatibility of the connected modem types with each other
- Compatibility of the connected modem types with the network parameters

The following is checked for SINAUT dedicated line networks:

- The parameter assignment of a master station for the dedicated line network
- The existence of more than one master station for the dedicated line network in polling mode

---

Note

If SIMATIC S7-300 stations are connected only via WAN, in older STEP 7 versions (lower than V5), the following warning may occur for the non-connected MPI nodes of a CPU: "CPU... (Station ..): The subscriber (...) is not connected to a network."

This warning can be ignored.

On completion of the network configuration, the configured status needs to be saved using the *Network Save* menu to allow the other STEP 7 and SINAUT applications access to the configured data.

Configuration continues by calling the SINAUT configuration tool.

Note

From the network configuration, not only the *Save* but also the *Save and Compile...* function can be called that generates the system data blocks (SDB) after saving the configuration.

To acquire all the configured parameters of the TD7 software when generating the SDBs both in new projects or when making modifications to the configuration of existing SINAUT installations, the generation of SDBs for SINAUT networks should only be performed in the *Subscriber Administration* of the SINAUT configuration tool.

3.4 Configuring TIM modules

3.4.1 Overview of the tabs of the properties dialog

The parameters for a TIM module are divided among various tabs of the *Properties - TIM* dialog. The following tabs are available:

- *General*
  tab with general information and for modifying the module name or adding comments
- *Addresses*
  tab with information on I/O address areas of the CPU
- *Special*
  tab for setting parameters for an ST1 master in a dial-up network and for the diagnostic buffer
- *Time Service* tab
  for assigning parameters for time synchronization of a TIM module on the MPI bus or an Ethernet TIM (TIM 3V-IE variants, TIM 4R-IE) on Ethernet
- *Interfaces* tab
  for configuring the Ethernet and WAN interface(s)
  This tab exists only with Ethernet TIMs
- *WAN Access* tab
  for creating WAN interfaces of the TIM modules of type TIM 3/TIM 4
- *Options* tab
  with options for assigning parameters for the message memory and the message indicating a failed local subscriber to substations over a dial-up network

---

Note

Communication-specific parameters are entered in the *Properties* dialogs for network and network node parameter assignment. These are explained in the relevant sections.

3.4.2 "General" tab

General tab

The *General* tab informs you about the general properties of a TIM module.

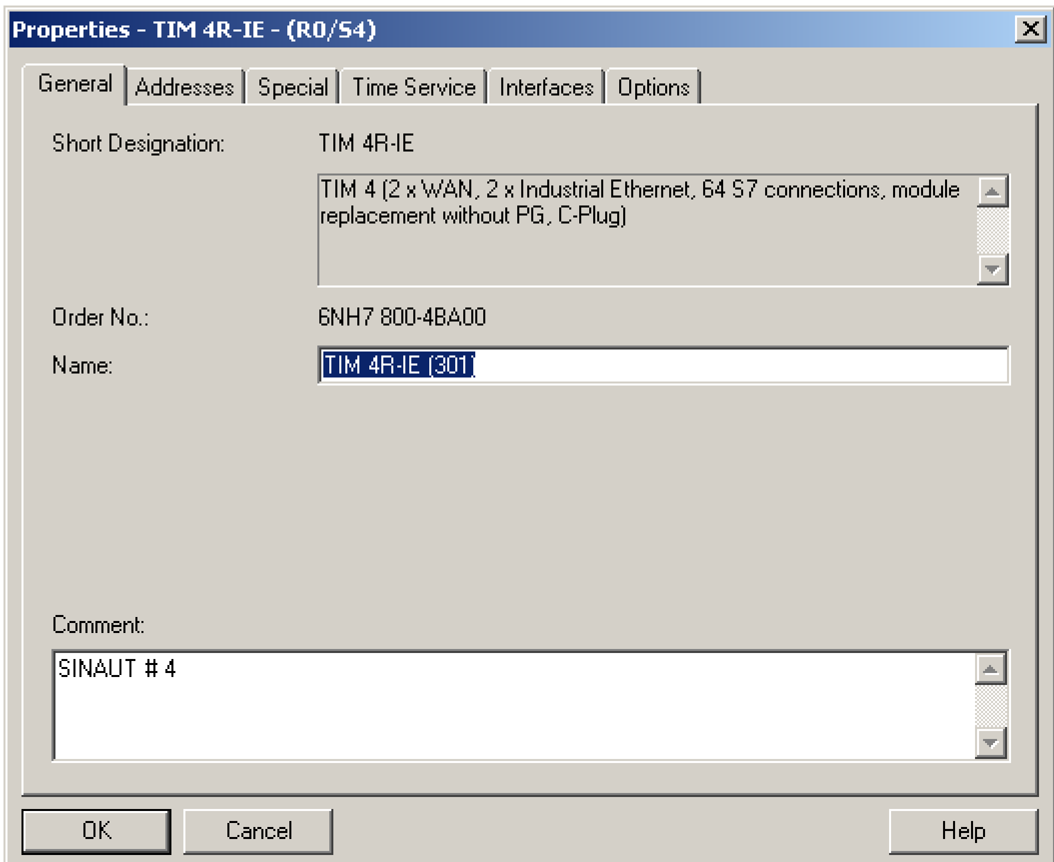


Figure 3-36 *Properties - TIM* dialog, *General* tab

This tab contains the following output boxes and parameter assignment options:

- *Short Designation:* output area This displays the module type and a brief outline of the hardware configuration.
- The *Order No.* output box displays the order number of the module.
- The *Name:* input box allows you to change the name of the module.
- As default, the *Interface* area shows the address and the networking status of the MPI interface. With the *Properties* button, you can open a dialog for setting parameters of the MPI node of the module. This is described in detail in the section on network node parameter assignment.
  Modules of the type series TIM 3 are also assigned an MPI address as default, even if this does not physically exist. If you click the *Properties* button, the parameters of the MPI interface are not available for these modules in the next dialog.

- The *Comment:* input box allows you to enter comments, for example on the purpose of the module.
  The SINAUT subscriber number of the TIM module that can be generated as a comment in the *Subscriber Administration* of the SINAUT configuration software is then displayed in this comment box.

3.4.3 "Addresses" tab

Addresses tab

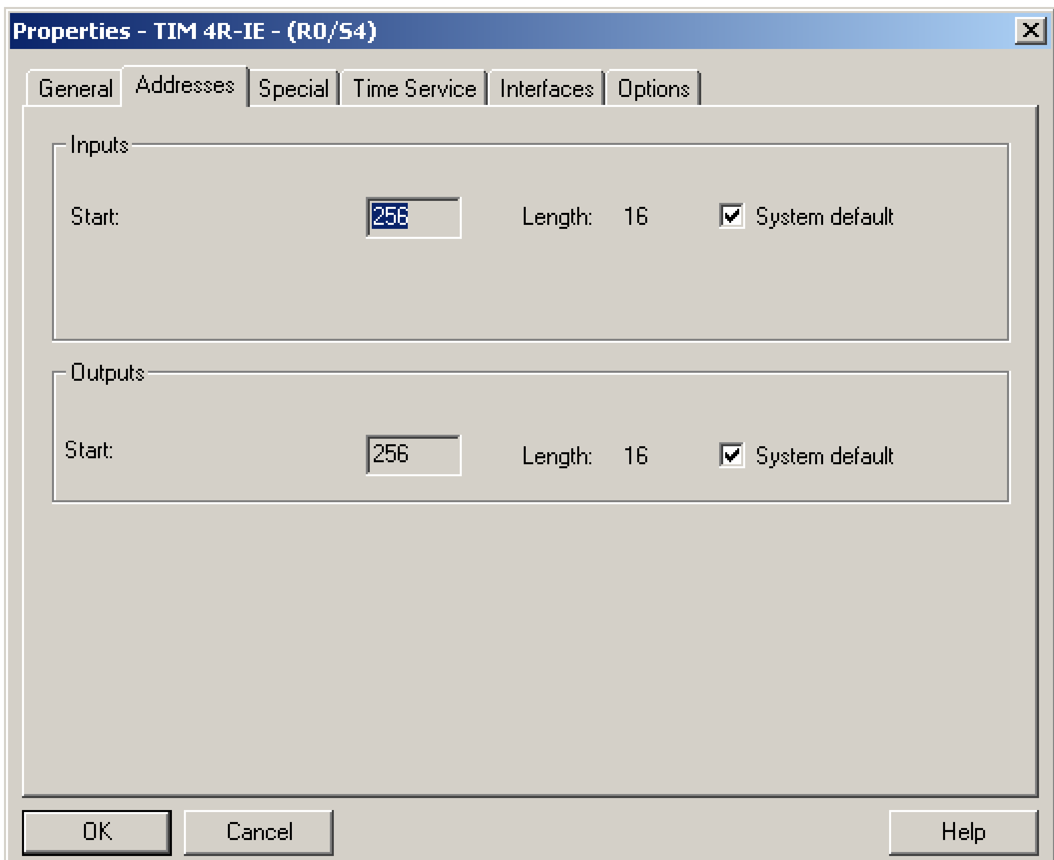


Figure 3-37 *Properties - TIM* dialog, *Addresses* tab

The Addresses tab provides information on the address areas occupied by the TIM module in the I/O from the perspective of the CPU.

These addresses are only relevant to you when the SINAUT program is configured on the TIM (TD7onTIM, possible with Ethernet TIMs) and when the CPU is supplied with the date and time by the TIM. In this case, the TIM supplies the time data to the inputs specified here. This is described in detail elsewhere (refer to the section: Time-of-day synchronization of the S7-300 CPU with TD7onTIM (Page 118)

The start address and length of the address ranges are assigned by the system. As an alternative, you can change the inputs and outputs by disabling the system selection option and entering the start address in the input box manually. Since the addresses are always set consistently by the system and are not generally used, it is not normally necessary to make a change.

Note

No writing of I/O addresses of the TIM from the user program

If you use TD7onCPU a CPU with a party line, the I/O addresses of the TIM must not be written from the user program. This can lead to failure of the communication..

On the party line, see Glossary.

3.4.4 "Special" tab

Special tab

The *Special* tab shows the SINAUT subscriber number of the TIM module and you can set the size and configuration of the diagnostic buffer.

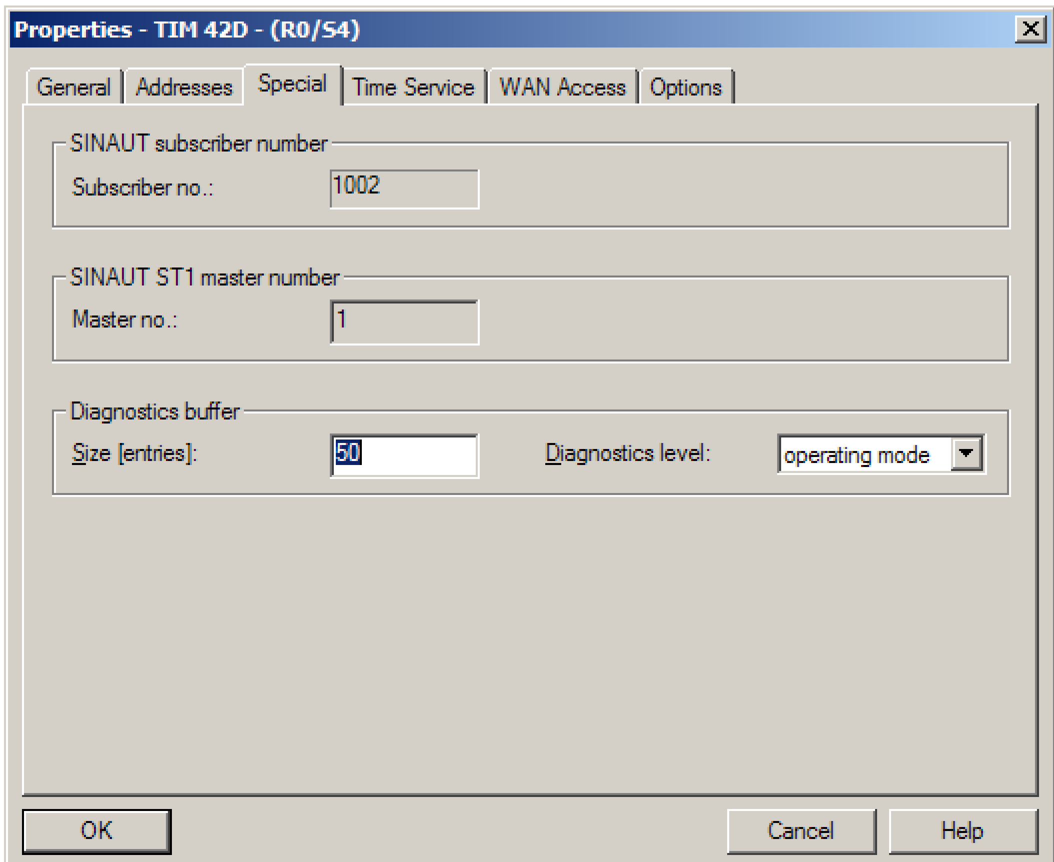


Figure 3-38 *Properties - TIM* dialog, *Special* tab

This tab contains the following parameter assignment options:

- *SINAUT subscriber number* box:
  The project-wide unique SINAUT subscriber number is displayed here. For more data of information, refer to the configuration of the *subscriber administration* in the SINAUT configuration tool.
- *SINAUT ST1 master number* box:
  In ST1 dial-up networks, the SINAUT ST1 master number must be specified for the TIM modules. For more information on SINAUT ST1, refer to release 05/2007 of the manual.

- *Diagnostics buffer size* box:
  The diagnostics buffer is organized as a circulating buffer and can hold the specified number of messages.
  Range of values: 10 ... U00
  default value: 50
- *Diagnostics level*:
  The diagnostics messages required for normal operation are generated in the *Operating mode*. In *Service mode*, additional diagnostics messages are generated.

3.4.5 "Time Service" tab

Time Service tab

Note

Here, it is not possible to set the time synchronization of the network attachments to a SINAUT dedicated line or a SINAUT dial-up network (RS-232/RS-485 port on the TIM). You make these settings in the properties dialog of the relevant dedicated line or dial-up network, refer to the "Time Service" tab in the section Configuring WAN network nodes (Page 76). There is no setting per network node in this tab because the hierarchical distribution of the time is specified automatically during parameter assignment (master/node station/station).

In the *Time Service* tab of a TIM module, you decide how the TIM will react to time synchronization on its interfaces:

- Time synchronization on the Ethernet interface(s)
- Time synchronization on the S7-300 backplane bus when the TIM is inserted in an S7-300
- Time synchronization on the MPI bus (with the TIM 4)

The figure below shows the tab for an Ethernet TIM (TIM 3V-IE variants and TIM 4R-IE).

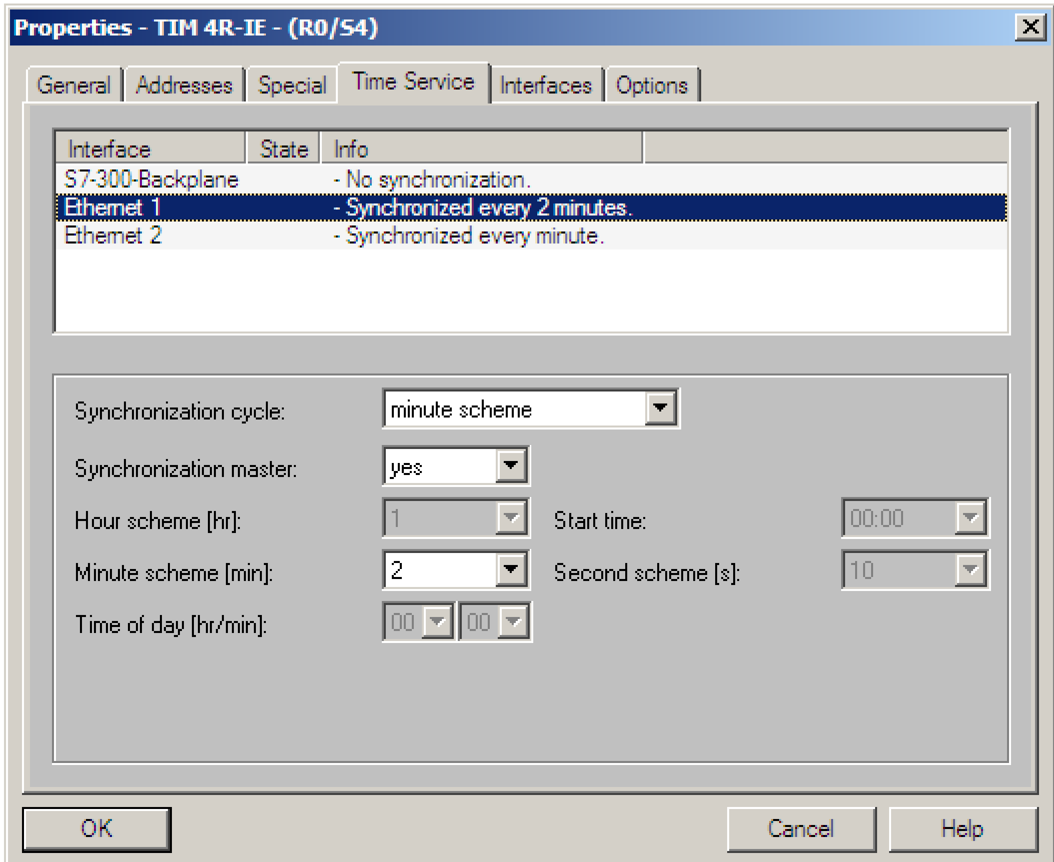


Figure 3-39 *Properties - TIM* dialog, *Time Service* tab

The interfaces for which you can set parameters are listed in an overview box. If you click on one of the interfaces, the parameters for time synchronization appear below the box for the interface.

Note

In the overview box of the TIM 4R-IE, the "S7-300 backplane bus" interface is always displayed. Parameters can, however, only be set for this interface if the TIM 4R-IE is inserted in an S7-300 as a CP.

The overview box is not available for modules without an Ethernet interface. Here, you can only set parameters for time synchronization on the MPI bus or S7-300 backplane bus. The same parameters are displayed as for an Ethernet TIM if the "S7-300 backplane bus" is selected there.

If a TIM is inserted in an S7-300 as a CP, time synchronization on the "S7-300 backplane bus" specifies when the time synchronization of the local S7-300 CPU is performed. If other TIMs are inserted in the same S7 rack, these are also synchronized at the same intervals as set here.

For a TIM 4 with MPI interface, the parameter settings for synchronization of the SINAUT nodes attached to the MPI bus apply (PCs, S7-300 and S7-400 CPUs and any other TIM 4 modules connected to the MPI bus).

The following rule applies if there are several TIMs in the S7-300 rack or several TIMs on the MPI bus:

1. Time synchronization must be enabled for all TIMs and set to the same time interval.

   After startup, only one of the TIMs will actually behave as the time master. This is negotiated automatically by the TIMs. The TIM acting as time master synchronizes all the local SINAUT nodes known to it. The other TIMs act as slaves and allow themselves to be synchronized by the current master. If the TIM acting as master fails, one of the other TIMs automatically takes over the time master function until the failed master TIM is available again.

While the "time master" or "time slave" roles are negotiated automatically on the S7-300 backplane bus or MPI, with the Ethernet interfaces of the TIM, the role of master or slave must be specified explicitly. The following rules apply:

1. If the Ethernet port of the TIM is connected to an Ethernet on which there is also an ST7cc or ST7sc PC, the PC in this network is always time master, in other words, the relevant Ethernet port of the TIM must be set to the "slave" function.
2. In an Ethernet network without ST7cc or ST7sc PC, the Ethernet port of one of the TIMs must be set to master and all others to slave. If the master function is set for more than one TIM on the Ethernet network, an error message is generated during the verification performed in the SINAUT node management.
3. Each Ethernet port to be synchronized by a master must be enabled as a slave. Otherwise synchronization is not accepted on this port. The setting of the synchronization interval or time of a slave should be identical to that of the master on the Ethernet network because the slave monitors whether or not the synchronization takes place at the specified intervals or at the specified time. Setting a shorter interval or a different time would lead to error messages in the diagnostics buffer of the TIM.

Note

The error message is not exactly coordinated with the interval or the point in time.

- With an interval, the error message comes after 2.5 times the set interval. Example: At an interval of 2 hours, the error message is entered only after 5 hours.
- If a specific time is selected, a tolerance of 2.5 hours is allowed before the error is signaled.

In the following example, meant to illustrate the two sections of a SINAUT project, shows where and which time synchronization setting must be made.

Example of time synchronization

It is assumed that all SINAUT nodes in the network need to be synchronized.

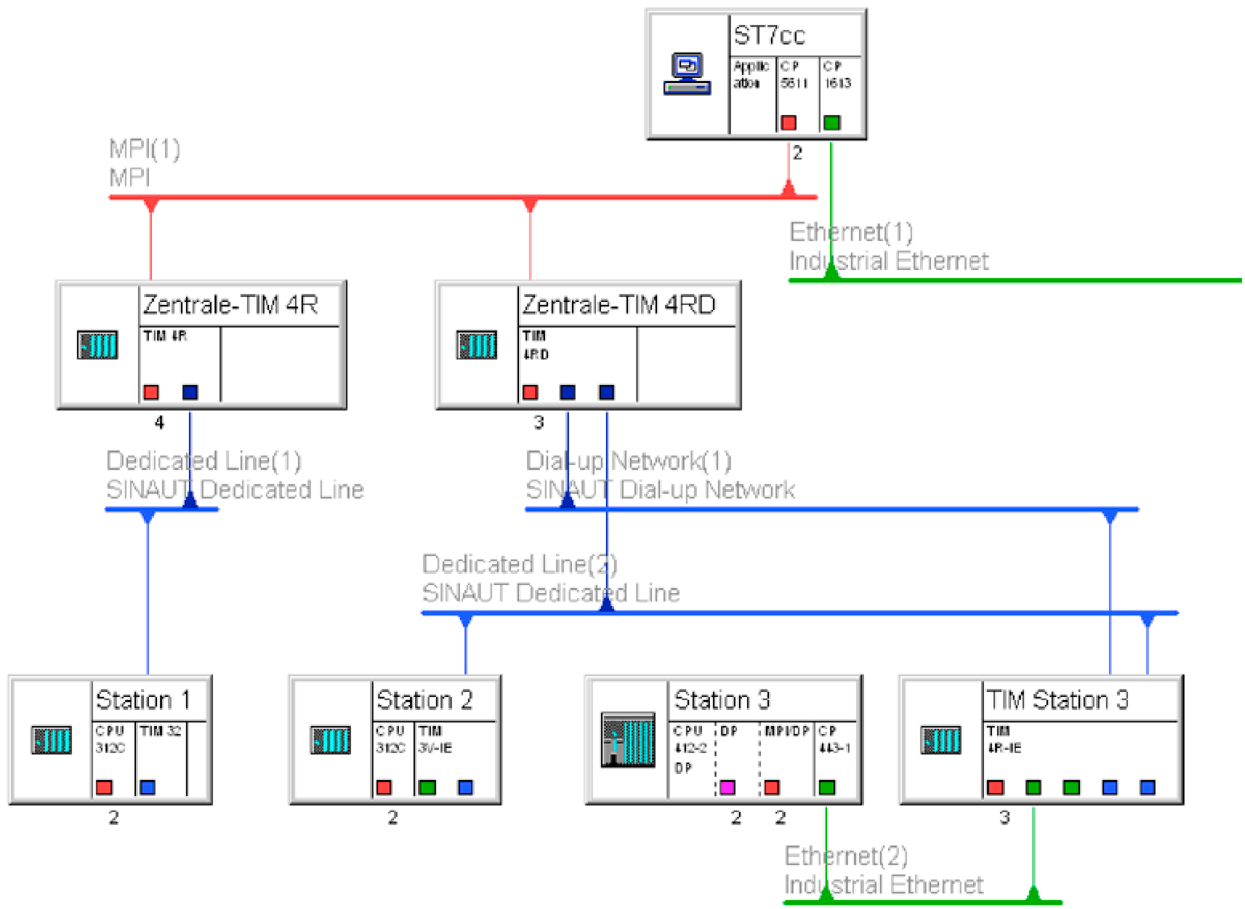


Figure 3-40 Example of time synchronization: Network section 1 - MPI / classic WAN

- **ST7cc**
  No settings are necessary here.
- **Master TIM 4R / master TIM 4RD**
  Time synchronization on the MPI bus must be enabled for both TIMs. An interval of 1 minute is recommended.

Note

ST7cc and ST7sc expect time synchronization on the MPI bus at intervals of 1 minute or less. Longer intervals cause error messages.

In this example, the "master TIM 4RD" takes over the master function after startup because it has a DCF77 receiver. Once this TIM has received a valid time of day, the ST7cc PC and the "master TIM 4R" are synchronized over MPI. If the TIM with a DCF77 receiver fails, the "master TIM 4R" can take over the master function.

Both TIMs synchronize the stations connected over a dedicated line or dial-up network by synchronizing the TIMs in the stations that, in turn, supply their own CPU. In these networks, you do not need to make settings for the TIMs. These TIMs obtain their parameters from the time parameter settings made centrally for the particular SINAUT network (dedicated line or dial-up network), refer to the "Time Service" tab in the section Configuring WAN network nodes (Page 76).

- **Station1, TIM 32**
  Time synchronization on the S7-300 backplane bus must be enabled for this TIM; in other words, the TIM then supplies the S7-300 CPU with the current time over the backplane bus. An interval of 1 minute is recommended.

  Note

  If a TIM synchronizes an S7-300 CPU over the backplane bus, no synchronization settings are necessary for the S7-300 CPU in HW Config. The SINAUT software on the CPU (TD7onCPU, FC TimeTask) handles the synchronization by using the synchronization message of the TIM to set the CPU clock.

- **Station 2, TIM 3V-IE**
  Here, there are two situations to be taken into account:
  - A SINAUT program (TD7onCPU) is running on the CPU:
    You set the time synchronization on the S7-300 backplane for the TIM.
  - The SINAUT program is configured on the TIM 3V-IE (TD7onTIM):
    You do not need to enable time synchronization on the TIM.
    Although there is no SINAUT program on the CPU, the CPU can nevertheless by supplied with the time of day when necessary. In this case, the TIM supplies the time data to its inputs. This is explained elsewhere, see section "Addresses" tab (Page 94).
- **TIM Station 3**
  This TIM handles the SINAUT communication for the S7-400 "Station 3". Here, you will need to enable time synchronization on the relevant Ethernet interface of the TIM as master. An interval of 1 minute is recommended.
- **Station 3**
  No settings need to be made for the S7-400 CPU nor for the CP 443 in HW Config. The SINAUT software on the CPU (TD7onCPU, FC TimeTask) handles the synchronization by using the synchronization message of the TIM to set the CPU clock.

The next figure shows a further excerpt from the sample project which will be used to explain further details on synchronization, particularly in an Ethernet network.

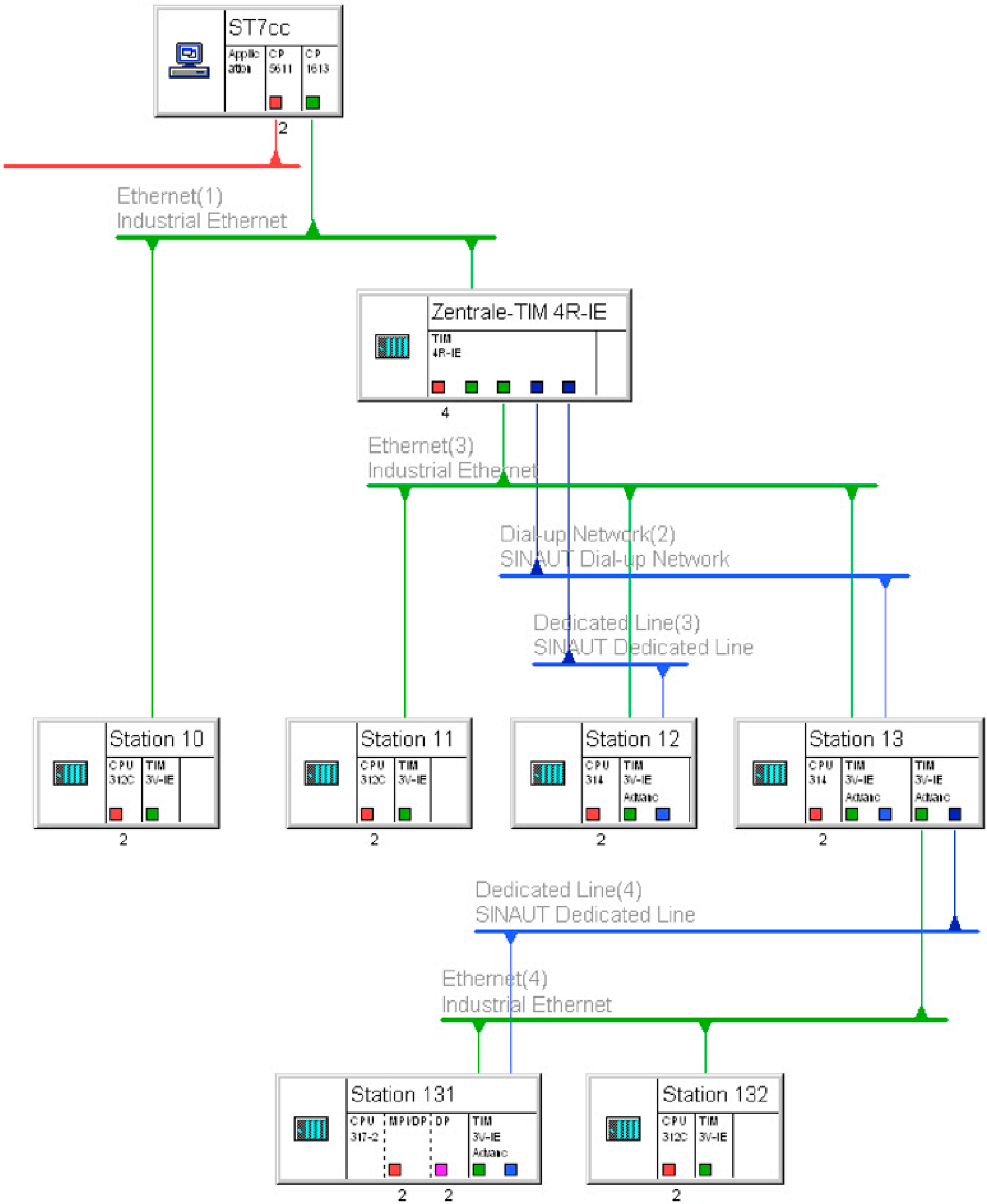


Figure 3-41 Example of time synchronization: Network section 2 - Ethernet

- **ST7cc**
  No settings need to be made in an Ethernet network in which ST7cc (or ST7sc) is always time master.
- **Station 10, TIM 3V-IE**
  This TIM is connected directly to ST7cc, the time master, over Ethernet. You will therefore need to enable the Ethernet port of the TIM as slave for time synchronization. An interval of 1 minute is recommended.

  For the CPU in station 10, you may want to enable synchronization on the S7-300 backplane bus. Refer to the notes above in the section "Station 2, TIM 3V-IE".

  Note

  If the station is connected to an Ethernet network for which fees are charged, for example via GPRS, it may be more economic to set an interval longer than 1 minute.

- **Master TIM 4R-IE**
  This TIM has two networked Ethernet accesses. Make the following settings.
  – On Ethernet(1):
    There is an ST7cc computer (= time master) on this Ethernet network. Enabling the interface as time slave. An interval of 1 minute is recommended.
  – On Ethernet(3):
    Enable the interface of the TIM as time master on this Ethernet network. Here, you can set an interval different from the interval for the slave interface on Ethernet(1).

  Apart from synchronizing the stations connected to Ethernet(3), the TIM also supplies the stations in the dedicated line or dial-up network by synchronizing the TIMs in these stations that, in turn, supply their CPUs. In these networks, you do not need to make settings for the TIMs. These TIMs obtain their parameters from the time parameter settings made centrally for the particular SINAUT network (dedicated line or dial-up network), refer to the "Time Service" tab in the section Configuring WAN network nodes (Page 76).
- **Station 11, TIM 3V-IE / station 12, TIM 3V-IE Advanced**
  The TIMs in both stations are attached to Ethernet(3) in which the "master TIM 4R-IE" is enabled as time master. This means that you will need to enable both TIMs as time slaves. The interval should be identical to that on the time master on Ethernet(3).

  For the CPU in station 11 or 12, you may want to enable synchronization on the S7-300 backplane bus. Refer to the notes above in the section "Station 2, TIM 3V-IE".
- **Station 13, two TIM 3V-IE Advanced modules**
  This station functions as a node station. Each of the two TIMs has a network access to Ethernet that you enable as time slave for the TIM on Ethernet(3) and as time master on the other TIM on Ethernet(4).

  To allow the TIM connected to Ethernet(4) to adopt the role of time master, it must be synchronized by the TIM connected to Ethernet(3). The TIMs are synchronized over the S7-300 backplane bus. To allow this, you will need to enable time synchronization over the S7-300 backplane bus on both TIMs.

  The hierarchical structure of the time synchronization network is continued here. If necessary, you can set an interval for the master interface (right-hand TIM) that differs from the interval for the slave interface (left-hand TIM).

Parameter settings for time synchronization

You can set the following options for the *Synchronization cycle* parameter:

- No synchronization:
  There is no time synchronization on the relevant network.
- Hour scheme:
  The number of hours between synchronization activities can be set in the "Hour scheme" drop-down list box.
  - Start time:
    If the cycle for time synchronization is longer than 1 hour, you can set a start time for time synchronization in the "Start time" drop-down list box.
- Minute scheme:
  The number of minutes between synchronization activities can be set in the "Minute scheme" drop-down list box.
- Second scheme:
  The number of seconds between synchronization activities can be set in the "Second scheme" drop-down list box.
- Time of day:
  Synchronization takes place once a day. Set the time of day for the synchronization in the "Time of day" drop-down list box (for example 01:00).
- Synchronization master (only for Ethernet port)
  Here, you can decide whether the TIM module adopts the master role for time synchronization (setting "yes") or not (setting "no"). If "no" is set, the TIM is a time slave.

  You will find more information on setting the time master or slave in the explanations above.

3.4.6 "Interfaces" tab

"Interfaces" tab

The “Interfaces" tab is available only for Ethernet TIMs. It displays a list of interfaces of the TIM module.

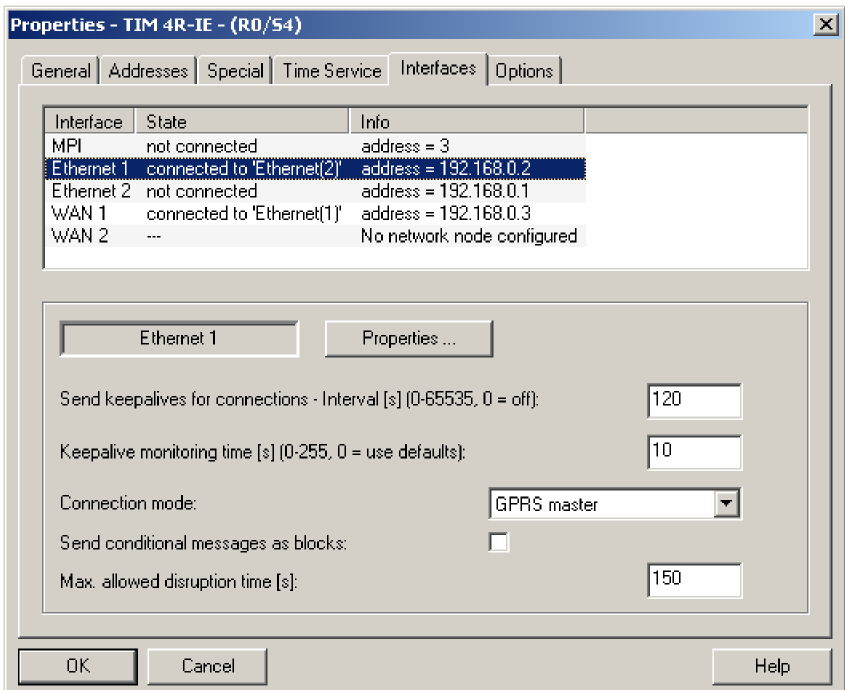


Figure 3-42 "Properties - TIM" dialog, "Interfaces" tab for an Ethernet TIM

If you click on an interface, the box for configuring the relevant interface is displayed below the list..

MPI

Ethernet TIMs do not have an MPI interface. If a TIM 4 is installed in an S7-300 as a CP, it is displayed in the list of the "MPI" interface. If you select this by clicking on it with the mouse, you can configure the internal station MPI address of the TIM using the "Properties" button.

Note

The MPI interface is displayed only when one of the following CPU types is inserted:

- All variants of the CPUs 312, 312C, 313C, 314 and 314C
- The CPUs 315-2 DP and 315F-2 DP

WAN 1 / 2

Parameter assignment as classic WAN interface:

The parameters of the WAN interfaces correspond to those of the TIM modules of the type TIM 3/TIM 4. You will find the description in the section "WAN Access" tab (Page 110).

WAN 1

Parameter assignment of the switchable serial interface as Ethernet interface for simple Internet communication via GPRS.
An Ethernet TIM that you want to connect as an MSC station (or node station) to a GPRS network via the GSM MD720 modem is connected to an Ethernet network via its serial interface.

With a TIM 4R-IE, only the switchable WAN 1 supports the function of an Ethernet interface. When networking interfaces in NetPro, remember that the left interface of 2 interfaces of the same type is not necessarily interface 1.

Follow the steps below to configure the interface:

1. Select the serial interface (WAN 1 with a TIM 4R-IE).
2. From the drop-down list in the lower part of the dialog, select the network type "Ethernet" and assign it to the WAN interface by clicking the "New" button.
   The drop-down list entry "Ethernet" is grayed out.
3. Open the properties dialog of the network node by clicking the "Properties" button, then connect the WAN network nodes with an Ethernet network and configure the IP parameters. Close the dialog with "OK".

   Back in the "Interfaces" tab, the MD720 is automatically assigned as "Modem type" and the "connection mode" is enabled. The WAN interface now has parameter settings as an Ethernet interface.
4. As the "connection mode", select either "MSC station" or "MSC node station".

The remaining MSC configuration is done in the SINAUT configuration tool. You will find an overview of MSC configuration in the section GPRS / Internet communication: Overview of configuration (Page 36).

TS module for the proxy module "PROXY CP1243-8" of the CP1243-8

To connect the serial interface of the CP to dial-up networks, TS modules are you must that are also configured in STEP 7 V5 in the "Interfaces“ tab of the properties dialog of the proxy. The following figure shows the drop-down list for selecting the modem type.

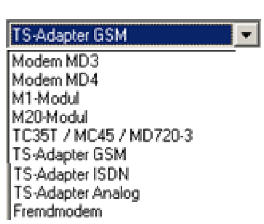


Figure 3-43 Selectable modem types of a WAN interface with connection to a dial-up network

The following entries are available for the TS modules:

- TS Module GSM

  Connection to GSM network
- TS Module ISDN

  Connection to ISDN network
- TS Module Modem

  Connection to an analog dial-up network
- 3rd party modem

  Select the "3rd party modem“ in the following situations:

  – Connection to a dedicated line network

    For the connection to a dedicated line network via a dedicated line or analog wireless modem, a TS module RS-232 is connected to the CP.

  – Connection to a dial-up network

    For the connection to a dial-up network via a 3rd party modem, a TS module RS-232 is connected to the CP.

  The TS module RS-232 cannot be configured in STEP 7 V5.

Ethernet 1 / 2

The parameters of the Ethernet interface are displayed below the list.

In the list box for the (static) Ethernet interface, "Ethernet 1" (or "Ethernet 2") is selected as default. You set the IP parameters of the Ethernet interface by clicking the "Properties..." button, see Configuring Ethernet nodes (Page 65).

Note

Ethernet interfaces of the TIM 4R-IE

The two Ethernet interfaces of a TIM 4R-IE are not designed as a switch, but are intended for connection to two different networks. Operation in the same Ethernet network is not permitted.

If this is ignored, it will not be possible to generate SDBs for the TIM. This is detected during the verification in SINAUT node management and signaled.

The IP addresses of the two interfaces must therefore differ in at least one of the three leftmost decimal (separated by a period) numbers (applies to the usual subnet mask 255.255.255.0).

Note

Ethernet interfaces of the TIM 4R-IE with S7-400

If you use a TIM 4R-IE as a standalone TIM in an S7-400, when using TD7onTIM the assigned CPU is always placed on Ethernet interface 1. The connection of a second CPU to Ethernet interface 1 of the TIM is not permitted.

If you use the "MSC" protocol, you need to configure this on the first Ethernet interface of the TIM. Once again, no further CPU may be connected to Ethernet interface 1. Only in this case can you also use TD7onTIM on the second Ethernet interface.

You will find an overview of the TD7 variants with the various combinations of master station and station types in the section Use and variants of the SINAUT TD7 software (Page 29).

When an Ethernet interface is selected, you have the option of setting the following parameters:

- **Send Keepalives for Connections** - Interval [s] (0-65535, 0 = off):

  This parameter specifies the interval in seconds at which keepalives are sent if there is no data traffic. If the value is set to 0, no keepalive messages are sent.

  For GPRS connections, a value of 120 seconds is recommended. The value can also be selected here depending on the period in which the "conditional spontaneous" frames stored on the TIM must be sent (see parameter Send conditional messages as blocks). The keepalive interval should always be shorter than the interval for "dead peer detection" (DPD) of the router (SCALANCE M874-2 / MD741-1). Note the default DPD interval of the router (MD741-1: 150 seconds).

- **Keepalive timeout** [s] (0-255, 0 = default):

  The parameter specifies the monitoring time in seconds after sending a keepalive. The acknowledgment of the message just sent must arrive within the monitoring time defined here. If the value is set to 0, the internal TIM default value is used (1 second).

  In GSM networks, a message is usually acknowledged within 1 to 2 seconds. This may take longer depending on the load on the GSM network. Experience has shown that a value of 10 seconds is practical in GSM networks.

- **Connection mode**

  You can either set parameters for the Ethernet interface for IP communication according to the S7 protocol ("neutral" setting) or for data transmission by GPRS. If you are connected to a GPRS network, you have the alternative of transmission using the GPRS router or for simple Internet communication with the MSC protocol using the GPRS modem MD720.

  An Ethernet interface cannot operate at the same time with the S7 protocol and the MSC protocol.

  In contrast to a normal "flat" Ethernet network in which every connected subscriber can communicate with every other subscriber, there are only point-to-point connections between station and master in GPRS networks. A direct connection from station to station is not strictly possible. Messages must be forwarded in GPRS networks via the master.

  To allow this, an Ethernet TIM can be used in the master that handles the routing of data messages between stations. To be able to establish the correct connection paths in the

GPRS network during configuration of the SINAUT connections, you will need to assign one of the following options to the Ethernet interface of the TIM on the GPRS network:

– Neutral

  On the Ethernet interface of the TIM, there is normal TCP/IP communication with the ST7 protocol; in other words, no subscriber is connected via GPRS and the MSC protocol is not enabled.

– GPRS master station

  In the role of "GPRS master", the Ethernet interface represents the highest level in the GPRS network hierarchy. Messages from TIMs in the role of "GPRS station" or "GPRS node station" to other stations in the network can then be routed over this interface.

– GPRS node station

  In the role of "GPRS node station", the Ethernet interface is subordinate to the "GPRS master". This setting is normally selected for a TIM located in a node station; in other words, in a station to which other stations are connected over a different network. Messages to be sent from this node station interface to other stations in the network are forwarded via the TIM with the interface role "GPRS master".

– GPRS station

  In the role of "GPRS station", the Ethernet interface is subordinate to the "GPRS master". Messages to be sent from this station interface to other stations in the network are forwarded via the TIM with the interface role "GPRS master".

You can still enable the MSC protocol. This makes simple Internet communication with GPRS via DSL/Internet or via a GPRS network possible. On the TIM 4R-IE, this function is supported only by Ethernet interface 1.

With the "MSC master", "MSC node station" or "MSC station" options, you specify the direction of the data transmission in SINAUT networks.

– MSC master (only TIM 4R-IE)

  You use this option for the interface of a TIM 4R-IE in the master or for the interface of a node station connected to a lower level network with stations.

– MSC node station

  You use this option for the interface of an Ethernet TIM in a node station connected to the master via a higher level network.

– MSC station

  You use this option for the TIM 3V-IE in a station.

The remaining MSC configuration is done in the SINAUT configuration tool. You will find an overview of MSC configuration in the section GPRS / Internet communication: Overview of configuration (Page 36).

- **Send conditional messages as blocks**

  Data transmission over a GPRS network is subject to fees depending on the amount of data transmitted. To minimize costs, smaller data packets can be collected and transferred in larger blocks if these messages are assigned the "conditional" priority; in other words, they do not need to be sent immediately, refer to the SINAUT software

TD7onTIM, section Basic parameters of the data objects (Page 194) or the description "SINAUT TD7 software package for the CPU" > "Data point typicals".

If the "Send conditional messages as blocks" option is enabled, the TIM transmits "conditional" messages in the following situations:

– When the collected messages reach or exceed a size of 202 bytes.

– If an important message needs to be transmitted immediately, "conditional spontaneous" messages already in memory are transmitted along with it.

– If the collected messages have not reached a size of 202 bytes, but the TCP/IP keepalive interval has elapsed, the stored messages are sent instead of the keepalive.

– If messages are sent via an MSC connection, conditional spontaneous messages are collected until 1 300 bytes have been reached or until an unconditional spontaneous message is transmitted.

- **Max. allowed disruption time**

  With this parameter, you can enter the tolerance time for connection disruption detected by the TIM. This can be useful in wireless networks that are often disrupted.

  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.

  Range of values: 0 ... M5535 seconds, default value: 0

3.4.7 "WAN Access" tab

WAN Access tab

In the *WAN Access* tab, you can configure the following WAN interfaces of the TIM modules of types TIM 3 and TIM 4. The parameters correspond to the WAN parameters of the *Interfaces* tab for the Ethernet TIMs:

- The internal interface that is normally assigned to a modem installed on the module
- The external interface that must be set up by the user by selecting the required network type and clicking on the *New...* button

Behind each interface that is to be connected with a network, there must be a network node of the corresponding type and this can be recognized by the "*Properties*" button being available for selection.

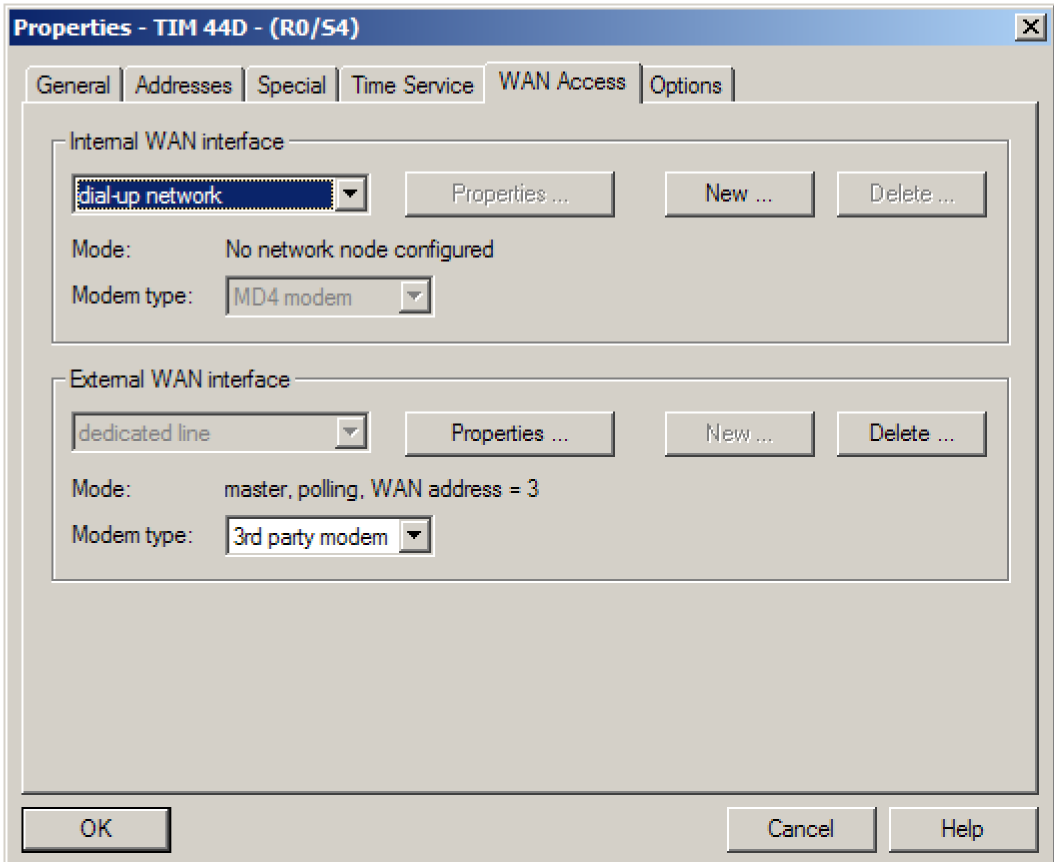


Figure 3-44 *Properties - TIM* dialog, *WAN Access* tab

For each of the two WAN accesses, you can make the following entries depending on the display in the *internal/external interface* list:

- Interface list box disabled with *Dedicated line*, *spontaneous network* or *dial-up network*:
  There is already a network node for this interface. Using the *Properties...* button, you can branch to the *Properties* dialog of the network node to make parameter settings. With the *Delete...* button, you can remove the network node.
  The *Properties* dialog of the interface is written when you set parameters for the network node.
- Interface list box selectable when *dedicated line*, *spontaneous network* or *dial-up network* is displayed:
  There is not yet a network node for this interface. To create a network node, you must select the corresponding interface type in the *Internal/external WAN interface* list box.
  With the *New...* button, you create a network node of the type displayed in the interface list box.
- Interface list box disabled with the display *Not available*:
  This interface cannot be operated by the current TIM module, no further parameter assignment possible.

The type of connected modem is displayed for each interface in "*Modem type*" if a modem exists on the TIM module or can be connected externally. This type cannot be changed for the internal interface. The type of a modem on the external interface must be configured over the list box.

The modem type is checked during the plausibility checks to establish whether or not it is compatible with the current network parameters. In addition to this, a default AT string for a SINAUT dial-up modem or GSM module is derived from the mode type and the network parameters *Baud rate*, *Message format* and *Connection type* (duplex, half duplex).

3.4.8 "Options" tab

Options tab

In the *Options* tab, you can set parameters for the message memory.

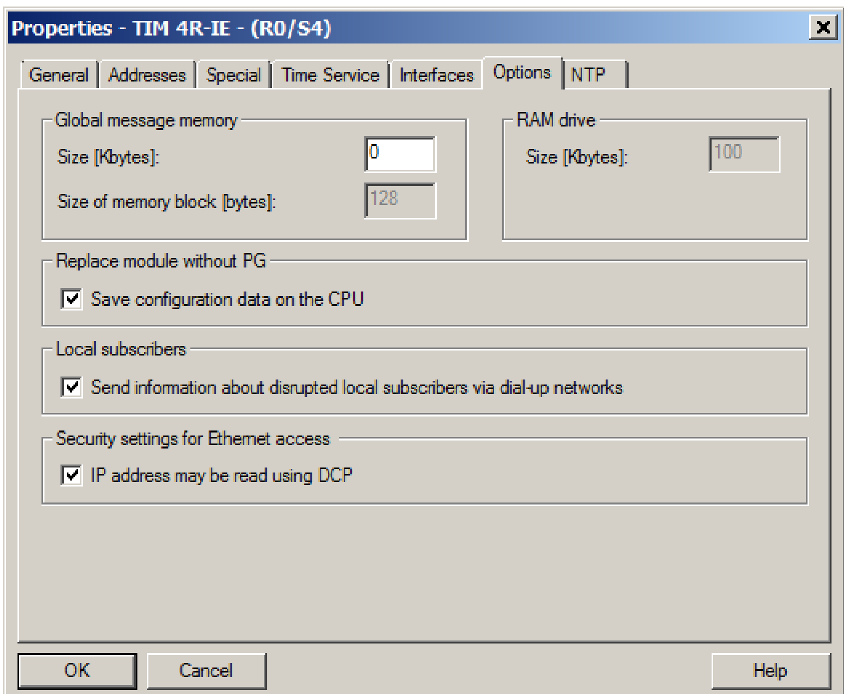


Figure 3-45 *Properties - TIM* dialog, *Options* tab

The following parameters can be set in the *Global message memory* area:

- *Size*:
  This is the size of the memory in which the messages to be transmitted are stored for all configured WAN drivers. If the value 0 is entered here, the entire free memory following startup is used.
  Range of values: 0 ... 1024 KB
  default value: 0
- *Size of memory block*:
  This is the size of the blocks into which the global message memory is segmented. Each message to be transmitted occupies at least this space in memory. The size should be matched to the size of the messages most commonly transmitted.

If the size is set too small, longer messages must be distributed over several blocks. If the size is set too large, memory space is wasted with many of the messages.
Range of values: 48 ... 65535 bytes
Default value: 64
With the Ethernet TIM, the value is set permanently to 128.

- *RAM drive*:
  In this input box, you set the size of the RAM drive.
  The RAM drive is a restricted area in the main memory of the TIM module that can be created for special test purposes. The memory cannot be used for normal operation and is deleted again when the TIM module is restarted.
  On the TIM 3V-IE, the RAM drive is already set up at 100 KB, the value 0 is displayed here and cannot be modified.
  On the TIM 3V-IE Advanced and the TIM 4R-IE, the RAM drive (100 KB) can be modified.
  Range of values: 0 .. 1024 KB
  Default: 0
  With the Ethernet TIM, the value is set permanently to 100.

In the *Replace module without PG* box, you can enable the following option for the Ethernet TIMs:

- *Save configuration data on the CPU* (Ethernet TIMs only):
  If you enable this option, the system data blocks (SDBs) of the TIM module are stored on the CPU. If the TIM module fails, the defective TIM can be replaced by a TIM of the same type without leading to download the SDBs to the TIM using a PG. The TIM module obtains its SDBs from its local CPU during startup.
  If the TIM is configured as a standalone TIM without a CPU in the rack, this function is not available.

Note

If there is no C-PLUG inserted in a TIM 4R-IE, the configuration data is stored in flash memory. If there is a C-PLUG inserted in the TIM 4R-IE, the configuration data is stored automatically on the C-PLUG when it is downloaded. If you replace the module, you can insert the C-PLUG with the configuration data in the new module.

In the *Local Subscribers* box, you can enable the following option for the TIM 4R-IE:

- *Send information about disrupted local subscribers vial dial-up networks*:
  If this option is enabled, the TIM signals the failure of local subscribers over connected dial-up networks to the substations.

  To reduce costs resulting from the automatic connection establishment in dial-up networks when subscribers drop out, this option can be disabled.

"Security settings for Ethernet access" box

- IP address may be read using DCP

  If the option is enabled, the TIM replies to read queries for the IP address using the DCP protocol (Discovery and Basic Configuration Protocol). The address of the TIM cannot be modified using DCP.

  DCP is used by a PG with the aid of the Primary Setup Tool (PST) or the addressing function of the SIMATIC Manager.

3.4.9 "NTP" tab

"NTP" tab

Here, you make the settings for a TIM 4R-IE in a master station for time-of-day synchronization by up to two NTP servers.

In NTP mode, UTC (Universal Time Coordinated) is transmitted. This corresponds to GMT (Greenwich Mean Time).

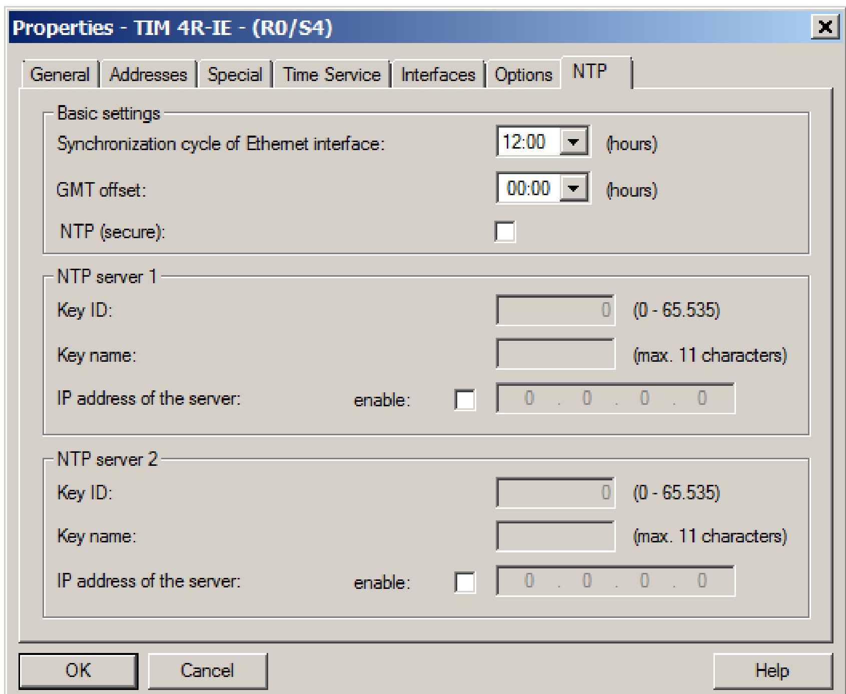


Figure 3-46 "NTP" tab

"Basic settings" box

- **Synchronization cycle**

  You specify the synchronization cycle using the drop-down list.

- **GMT offset (hours)**

  Here, select the offset of your local time zone to Greenwich Mean Time.

- **NTP (secure)**

  – If the option is disabled, NTP without authentication is used.

  – If the option is enabled, the secure method NTP (secure) is used.

    NTP (secure) uses authentication with symmetrical keys according to the hash algorithms MD5 or SHA-1.

    You then make the settings for the parameters of the NTP servers.

"NTP server 1" / "NTP server 2" boxes

- **Key ID**

  Here, enter the key ID agreed with the provider of the NTP server.

  The hash function MD5 is supported for the key.

  Possible values: 0...65535

- **Key name**

  Here, enter the key name that helps you to identify the key with the key ID entered above.

  Range of values: 5 ... 11 ASCII characters

- **IP address of the server**

  If you want to use the NTP server, first select the "enable" check box.

  Then enter the IP address of the NTP server in decimal numbers.

3.4.10 Consistency check and saving

Saving and consistency check

Once you have completed the parameter assignment in the hardware configuration, the current version must be saved with the *Station / Save* menu. You can generate system data blocks (SDBs) using the *Save and compile* menu later since there are still other configuration steps necessary before the SDBs can be generated completely. When you close hardware configuration, a dialog opens automatically prompting you to save your entries.

When you select the *Save and compile...* function, a consistency check is run and a message is displayed if configuration errors are detected.

Note

To be able to acquire all the configured parameters for the TD7 software following changes in the hardware configuration of existing SINAUT installations when the system data is generated, the SINAUT configuration tool must first be started with connection configuration and then with subscriber administration. The project should be saved there, and the SDBs should be generated exclusively in *subscriber administration*.

Printing module information

The information on all configured modules in the current rack or for a selected module can be printed out using the *Station / Print...* menu.

3.5 Configuring time-of-day synchronization

3.5.1 Configuring time-of-day synchronization - overview

Configuring time-of-day synchronization

You configure time-of-day synchronization in STEP 7.

Note

Avoiding time-of-day inconsistencies

Note the following points relating to time-of-day synchronization:

- Make sure that only one subscriber ever operates as the time-of-day master in a telecontrol network.
- Make sure that only one of the methods of time-of-day synchronization listed below is used.

Initially, specify precisely one time master in the network. This obtains its time of day from a PC or an NTP server.

- You make the settings for synchronization by a PC in the "Time Service" tab of the TIM, see below.
- You make the settings for synchronization of a TIM 4-IE by an NTP server in the following tabs:"NTP" tab (Page 114)

  To connect the TIM to the network / Internet in which the NTP server is located, you do not need to specify an interface in the configuration. Simply enter the IP address of the NTP server in the "NTP" tab mentioned above.

  It is advisable to separate the two interfaces of the TIM 4R IE into the internal data network (your system) and external network (connection to NTP server).

The procedure that follows differs depending on the S7 device family (S7-300 / S7-400) and on the network type.

Time-of-day synchronization via the TIM in an S7-300

You specify day direction of the time-of-day synchronization with the type of connected interfaces of the TIM modules. The configuration is slightly different for the various network types:

- Ethernet

  The direction of time-of-day transfer is made here by setting the "connection mode" of the TIM interfaces. Set the connection mode in the properties dialogs of the TIM in the "Interfaces" tab; for the master station set "GPRS master station" and for the other TIM modules set "GPRS station" or "GPRS node station".

- Dedicated line / dial-up network

  Here, the direction of the time of day transfer is decided by the setting of the "Network node type" of the connected TIM modules (properties dialog of the TIM > "Interfaces" tab > WAN interface X > Properties > Node type) The time of day is sent by the master station to the connected station TIMs or node stations. For information on configuration, refer to the properties dialog of the WAN network, section Configuring WAN network nodes (Page 76)

You configure the time-of-day synchronization of the TIM modules in the properties dialog of the TIM > "Time Service" tab (Page 97).

The availability of the time on the TIM depends on the variant of the TD7 software being used.

- TD7onCPU

  The time is made available in the block "FC-TimeTask". The block then synchronizes the CPU time.

- TD7onTIM

  The station TIM makes the time of day available to the local CPU in its I/O addresses. This makes the time of day available to the assigned S7 CPU. You will find details in the section Time-of-day synchronization of the S7-300 CPU with TD7onTIM (Page 118).

Time-of-day synchronization via the TIM in an S7-400

You will find a description of the procedure for a TIM as CP of the S7-300 and for the S7-400 in the section Time-of-day synchronization of an S7-400 CPU (Page 119).

Time-of-day synchronization using the CPU

Apart from the time of day synchronization of the station using the TIM, you can also synchronize the time of day of the TIM with that of the CPU.

You will find a description of the procedure for an S7-300 and for an S7-400 in the section Synchronization of the Ethernet TIM by the CPU (Page 120).

3.5.2 Time-of-day synchronization of the S7-300 CPU with TD7onTIM

Time information provided by the TIM

The synchronization of the time on stations is selected in HW Config in the properties dialog of the TIM, *Time Service* tab, refer to the section Overview of the tabs of the properties dialog (Page 92).

If the "TD7onCPU" TD7 software is running on an Ethernet TIM (TIM 3V-IE variant or TIM 4R-IE) and the TIM's time is synchronized by a master computer, the TIM acts as time master and synchronizes its local CPU (using FC TimeTask) over the backplane bus.

If the "TD7onTIM" TD7 software is running on an Ethernet TIM and the time on the TIM is synchronized by a master computer, the TIM makes the time available to its local CPU at the I/O addresses. The CPU user program can then read and evaluate the time there.

The I/O addresses of the CPU available for the date and time information from the TIM are set in HW Config in the properties dialog of the TIM, *Addresses* tab, refer to the section Overview of the tabs of the properties dialog (Page 92).

The time information of a time-synchronized TIM module with TD7onTIM is stored in 8 bytes of the peripheral "inputs". The time information has an offset of 8 bytes from the base value of the inputs. The following table shows the assignment of the 8 bytes of time information.

Time-of-day format

Table 3- 2 Format of the time information in the inputs of the I/O addresses of the TIM

Byte No.	Offset [bytes]	Meaning	High nibble		Low nibble	
			Value	Range of values (decimal)	Value	Range of values (decimal)
1	+8	**Year**	tens	0...9	ones	0...9
2	+9	**Month**	tens	0...9	ones	0...9
3	+10	**Tag**	tens	0...9	ones	0...9
4	+11	**Hour**	tens	0...9	ones	0...9
5	+12	**Minute**	tens	0...9	ones	0...9
6	+13	**Second**	tens	0...9	ones	0...9
7	+14	**Millisecond**	hundreds	0...9	tens	0...9
8	+15	**Millisecond**	ones	0...9	**Status**	**Bit coded**

Meaning of the entries in the table:

- Offset: Offset to the base value of the peripheral inputs in bytes
- High nibble: Bits 4 -7:
- Low nibble: Bits 0-3

- Value: Position of the relevant number
  - Year, month, day, hour, minute and second are two-digit (tens + ones)
  - Milliseconds are three-digit (hundreds + tens + ones)
- Status: Status of the time information

Format of "Status"

The status of the time information is available in the four bits of the low nibble of byte 8. The following table shows the meaning and the values of the status.

Table 3- 3 Status bits of the time (low nibble of byte no. 8)

Bit No.	3	2	1	0
Meaning	Prewarning bit	*Not defined*	Daylight saving time (DS), standard time (ST)	Validity of the time
Value	0 = - 1 = prewarning: Changeover at the next full hour (DS -> ST or ST -> DS)		0 = ST 1 = DS	0 = invalid 1 = time valid

Initial setting of the time

The user program that reads out the time on the CPU should only do this when the validity bit is set. This is the case as soon as the TIM is synchronized the first time, either by the time master in the SINAUT network or by the PG.

Note

If the time on the TIM was set from a PG, this is always indicated as standard time (status bit 1 has the value "0").

3.5.3 Time-of-day synchronization of an S7-400 CPU

Time information provided by the TIM 4R-IE

If a TIM 4R-IE in stand-alone mode is connected to a master station via Ethernet and an S7-400 CPU is connected to the second Ethernet interface, the TIM can synchronize the time of day of the CPU. The TIM synchronizes the S7-400 CPU using SNAP messages.

The parameters for time-of-day synchronization are set in the properties dialog of the TIM in the "Time Service" parameter group:

- The synchronization cycle is set on the Ethernet interface connected to the master station. The TIM automatically obtains the time-of-day from the master station computer.
- The second Ethernet interface that is connected to the S7-400 CPU is set to "Synchronization master".

The S7-400 CPU must be configured in the "Clock" parameter group as a slave.

3.5.4 Synchronization of the Ethernet TIM by the CPU

Apart from the option of synchronizing the TIM by a master station or using NTP, an Ethernet TIM can also be synchronized by its local CPU.

In this case, the CPU must receive the time from a suitable time master.

Note

Changing the time

Note a possible changeover between standard and daylight saving time on the master station when evaluating the messages from the station if the TIM is connected to a master station.

Constraints

The time-of-day synchronization of the TIM by a CPU is subject to the following conditions:

- Ethernet TIM

  The synchronization of a TIM by the local CPU is possible only with the S7 Ethernet TIM not with the DNP3 TIM.
- S7-300

  If you want the TIM synchronized by the CPU, there must be no classic TIM (TIM 3 / TIM 4) plugged in beside the Ethernet TIM.
- S7-400

  The time of day is received via a CP of the station and forwarded to the station.

  The TIM receives time-of-day frames via the PROFINET interface of the CPU or via an Ethernet CP of the S7-400 CPU.

  If you use the TD7 variant TD7onTIM in the TIM in an S7-400, the TIM cannot write the time to the process image of the inputs (PII) of the CPU.

Configuration with S7-300

Configuration of the CPU

1. In the properties dialog of the CPU, open the "Diagnostics / Clock" tab.
2. In the "Clock" box, select the option "As master" in the "On PLC" drop-down list and select the time interval.
3. Close the properties dialog of the CPU.

Configuration of the TIM

1. Open the properties dialog of the TIM with the "Time Service" tab.
2. Select the backplane bus and set the synchronization cycle to the same setting as configured on the CPU.
3. Close the dialog and save.

Configuration with S7-400

Configuration of the CPU

1. In the properties dialog of the CPU, open the "Diagnostics / Clock" tab.
2. In the "Clock" box, select the option "As master" in the "On PLC" drop-down list and select the time interval.
3. Close the properties dialog of the CPU.

Configuration of the CP

1. Open the properties dialog of the Ethernet CP and switch on time-of-day synchronization in the "Time-of-Day Synchronization" tab.
2. In the selected synchronization method enable the forwarding of the time to the station.
3. Close the properties dialog of the CP.

Configuration of the TIM

1. Open the properties dialog of the TIM with the "Time Service" tab.
2. Select the Ethernet interface with which the TIM is connected to the station.
3. Set the synchronization cycle the same as configured on the CPU and set the "Synchronization master" option to "No".
4. Close the dialog and save.

Use of FC-TimeTask with TD7onCPU

- **Use of FC-TimeTask**

  If you want time-dependent processing such as use of FC-Trigger, SMS_Control or for averaging (e.g. Ana044W_S), to synchronize the CPU you should not set this itself, but write the time in the DB BasicData to the data double words LastSyncDate and LastSyncTime.

  With this, FC-TimeTask sets and reads the CPU clock and the messages receive a time stamp.

  No use of FC-TimeTask

  If you work without FC-TimeTask and the CPU is synchronized directly, no time-dependent processing is possible. In this case, the TIM handles the time stamping of the messages configured with time. In this case, the messages receive the time stamp somewhat later.

4 Configuration in the SINAUT configuration tool

4.1 Introduction

4.1.1 Working with the SINAUT ST7 configuration tool

The SINAUT ST 7 configuration software fits into the familiar Windows sequences. This means that functions such as

- Window technology
- Menu bar
- Toolbar
- Online help for the dialogs of the configuration software
- Online help for the TD7 blocks
- Printing

are integrated according to the Windows and STEP 7 standards. Working with the configuration tool is explained in the individual sections of the chapter.

General information on working with the tool

You select a menu or a graphic object by clicking once with the left mouse button.

Further functions for this object are then available over the menu bar, over the buttons of the toolbar or often over a context menu that opens when you select an object with the right mouse button.

The buttons for the properties dialogs available for configuration have the following functions:

- *OK:* Confirms the entries made and closes the dialog.
- *Cancel:* Entries made are ignored and the dialog is closed.
  The *Cancel* button is not available in the dialogs for subscriber administration in the SINAUT configuration tool. In this case, the dialog is closed without entering changes by clicking on the close button [x] in the right-hand top corner of the header line of the dialog.
- *Help:* Opens the online help.

4.1.2 Starting the SINAUT Configuration Tool

Introduction

Start the SINAUT ST7 configuration tool using the menu "Start > SIMATIC > SINAUT ST7 > Configuration".

At the start of configuration, you open a project with the *Project / Recently Used* menu or the *Open Project* button in the toolbar.

After selecting the project, the *SINAUT ST7: Configuration* dialog opens making the three following main functions of the SINAUT Configuration Tool available:

- *Connection Configuration* for SINAUT connections
- *Subscriber Administration* for SINAUT subscribers
- *SINAUT ST1 - Configuration Overview*

  For more information on SINAUT ST1, refer to release 05/2007 of the manual.

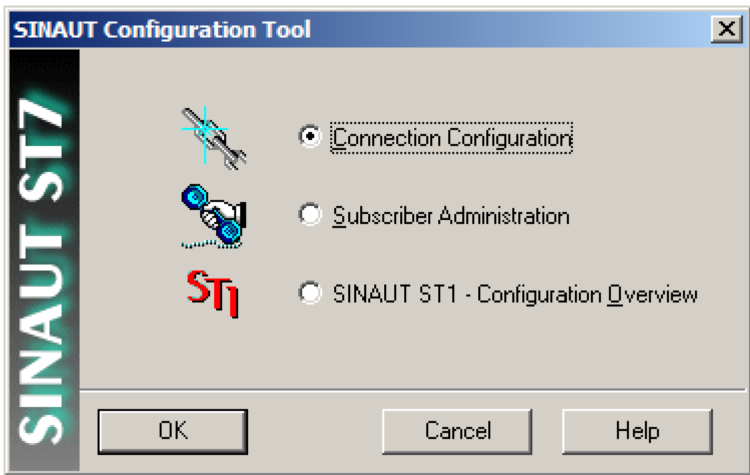


Figure 4-1 Selection dialog of the SINAUT Configuration Tool

To continue configuration, first select the *Connection Configuration* option.

When the SINAUT Configuration Tool is open, you can change between the three functions using:

- The *SINAUT / ...* menu,
- The corresponding buttons in the toolbar or
- the following function keys:
  - F3 for *Connection Configuration*
  - F4 for *Subscriber Administration*

Program information and SINAUT Internet pages

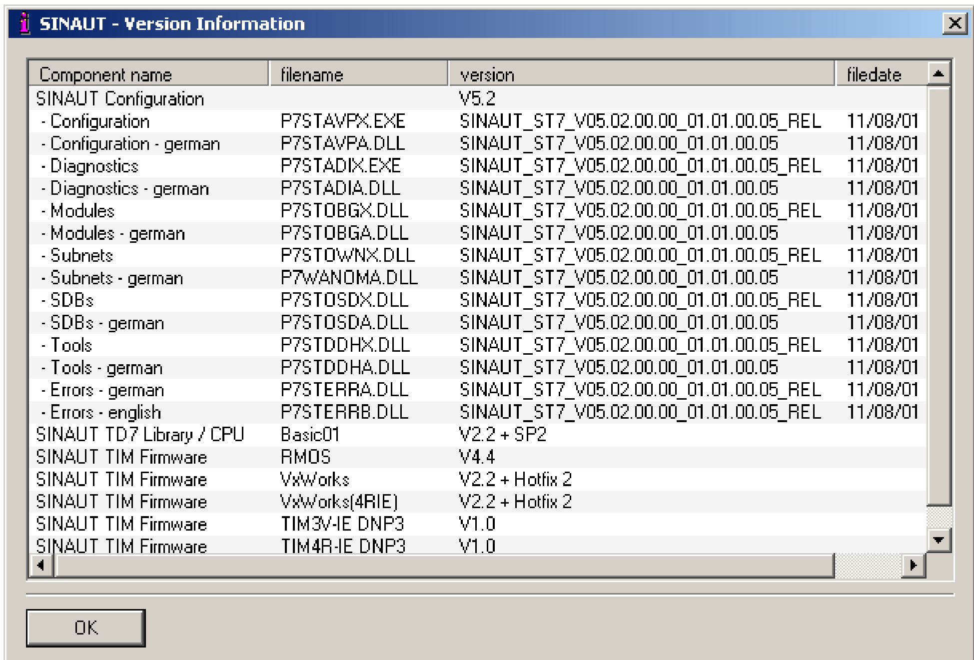
To display the program version, open the Info dialog of the SINAUT configuration tool using the "Help >Info" menu.

You can access the SINAUT homepage at the Internet address http://www.sinaut.de (http://www.sinaut.de).

4.1.3 Version information

The version information wizard displays the currency installed version and compilation time of the most important components of the SINAUT configuration software. The installed versions of the SINAUT TD7 library and the SINAUT TIM firmware are also shown.

This function is started from the Windows start menu *SIMATIC / SINAUT ST7 / Information*.



Component name	filename	version	filedate
SINAUT Configuration		V5.2	
- Configuration	P7STAVPX.EXE	SINAUT_ST7_V05.02.00.00_01.01.00.05_REL	11/08/01
- Configuration - german	P7STAVPA.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05	11/08/01
- Diagnostics	P7STADIX.EXE	SINAUT_ST7_V05.02.00.00_01.01.00.05_REL	11/08/01
- Diagnostics - german	P7STADIA.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05	11/08/01
- Modules	P7STOBGX.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05_REL	11/08/01
- Modules - german	P7STOBGA.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05	11/08/01
- Subnets	P7STOWNX.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05_REL	11/08/01
- Subnets - german	P7WANOMA.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05	11/08/01
- SDBs	P7STOSDX.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05_REL	11/08/01
- SDBs - german	P7STOSDA.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05	11/08/01
- Tools	P7STDDHX.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05_REL	11/08/01
- Tools - german	P7STDDHA.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05	11/08/01
- Errors - german	P7STERRA.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05_REL	11/08/01
- Errors - english	P7STERRB.DLL	SINAUT_ST7_V05.02.00.00_01.01.00.05_REL	11/08/01
SINAUT TD7 Library / CPU	Basic01	V2.2 + SP2	
SINAUT TIM Firmware	RMOS	V4.4	
SINAUT TIM Firmware	VxWorks	V2.2 + Hotfix 2	
SINAUT TIM Firmware	VxWorks(4RIE)	V2.2 + Hotfix 2	
SINAUT TIM Firmware	TIM3V-IE DNP3	V1.0	
SINAUT TIM Firmware	TIM4R-IE DNP3	V1.0	

Figure 4-2 SINAUT ST7 version information

4.2 Configuring connections in the SINAUT Configuration Tool

4.2.1 Configuring SINAUT connections

The SINAUT ST7 connections

In SINAUT ST7 installations, connections are always configured between the following subscribers:

- From CPU to CPU
- From a CPU to a SINAUT ST7cc/ST7sc control center

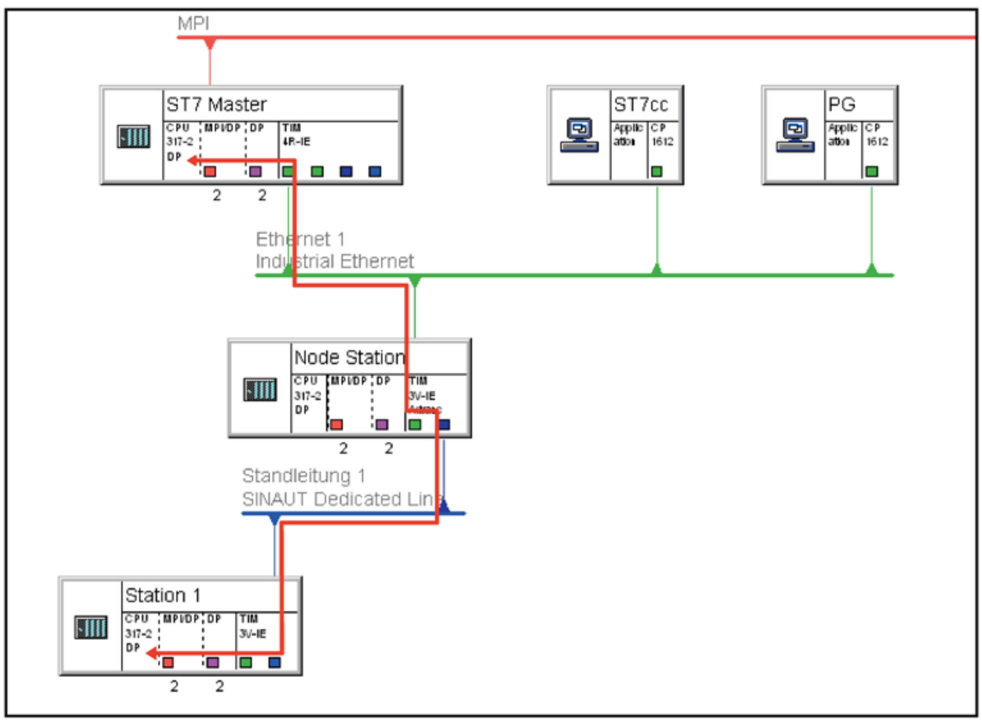


Figure 4-3 Example of a SINAUT ST7 connection from the master station to station 1

The connection shown in dark red from the master station to station 1 is not shown in this form in the network configuration with NetPro.

In the connection configuration, on the other hand, you can configure connections as follows:

- You can configure connections without knowing the exact connection path.
- You can view the exact connection path with a list of used network nodes.

  The degree of detail of the connection information can be set.

View of the *Connection Configuration* window

After opening the connection configuration, the configuration window opens.

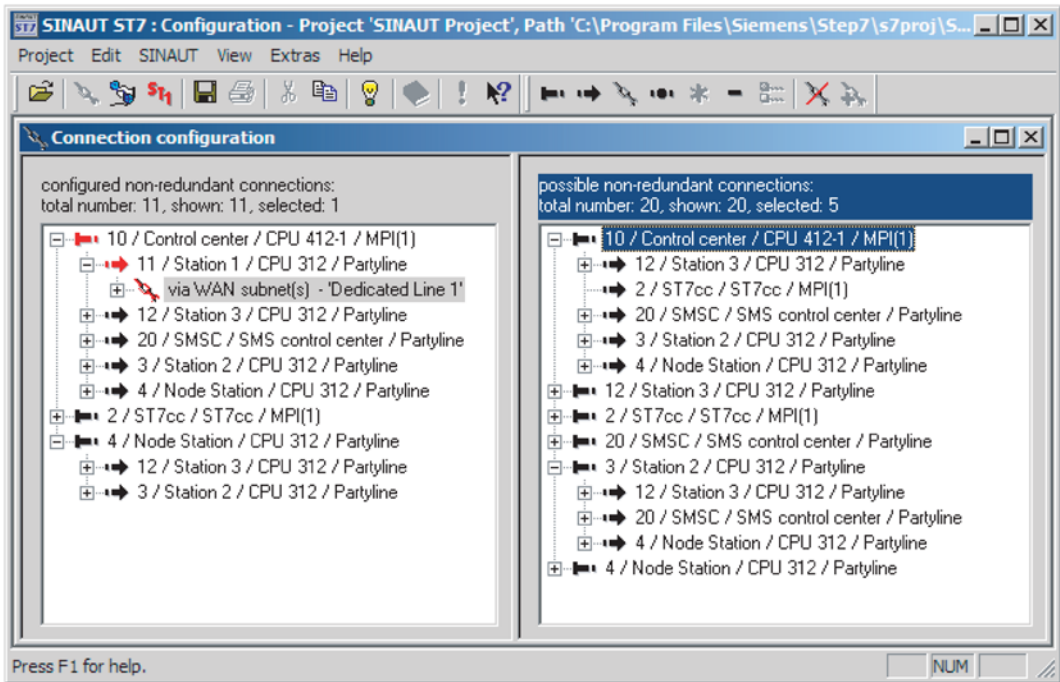


Figure 4-4 The *Connection Configuration* window of the SINAUT Configuration Tool

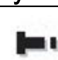
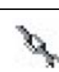
In the right-half of the window, the *possible connections* are listed in a tree structure resulting from the network configuration in NetPro. The connections actually required and used for communication in the SINAUT installation must be transferred from the right-hand to the left-hand window *Configured connections*.

Above the two lists the number of connections involved is displayed.

Setting the view:
The view, in other words the amount of information shown such as names of subscribers, nodes or interfaces etc. can be set in the "Connection configuration" dialog. You open this using the "Extras" > "Options" menu. See also section Options for connection configuration (Page 133).

The entries in the connection tables must be interpreted as follows:

Table 4- 1 Symbols in the connection list of the connection configuration

Level	Symbol	Meaning
1		Connection starting point
2		Connection end point
3		Alternative path

Level	Symbol	Meaning
4		Connection node over which the connection runs
1		Invalid connection

Invalid connections

Invalid connections are displayed in red as shown in the example of a connection that no longer exists due to changing the configuration. A connection can be invalid due to the following conditions:

- The connection was configured earlier but has been deleted again in the meantime.
- The subscriber number of a communications partner has changed (for example after replacing a CPU).
- The connection is invalid due to a parameter (for example baud rate).

Labeling

The labeling of the individual connection point in the basic setting describes the relevant subscriber with the following properties:

Subscriber number / Station name / Module / Interface.

Example: *5 / Station 3 / CPU 312 / MPI (2)*

The representation can be set to meet individual requirements using the *Extras / Options* menu.

Functions of connection configuration

To make configuration of the required connections as simple as possible for you, the SINAUT Configuration Tool uses the following strategy:

- The entire currently configured network is analyzed. All potential communication subscribers from the SINAUT perspective are assigned a subscriber number if they do not already have one. The subscriber numbers for CPU modules and third-party stations are assigned starting at no. 1, for TIM modules there are assigned starting at no. 1001.
- A tracking algorithm detects ALL connections in the current network. These connections can also extend over several LANs and SINAUT networks. The connections permitted based on specified rules are represented on the right as *possible connections* in a tree structure.
- SINAUT connections that have already been configured are displayed in the left-hand window for *configured connections*. Each of the connections loaded there is then checked to establish whether its configured parameters match the current network and hardware configuration. If this is not the case, an error message indicating incorrect connections is displayed as soon as the connection configuration is opened and the bad connection is displayed in red in the *configured connections* window.

If a station of the type *other station* or *SIMATIC S5* was configured in NetPro, connections from the stations to stations of the type ST7, PG/PC, or an ST7cc control center are not displayed. This does not, however, mean that these connections do not exist. These connections are in fact displayed in the opposite direction; this means, for example, from an ST7 station to a station of the type *other station* or *SIMATIC S5*.

As a general rule, a connection displayed in the *configured connections* in only one direction, works in both directions.

Selecting the required CPU-CPU connections

If no connections are displayed in the left-hand window, the required connections must be transferred from the right-hand window. Follow the steps outlined below:

1. Expand the tree structure by clicking on the branch symbol (+) or by double-clicking on the connection group. The tree structure opens.
2. Select a *possible connection* in the right-hand window.
3. Enter the possible connection as a *configured connection* in the left-hand window either by
   – selecting the *Edit / Apply* menu or
   – pressing the right mouse button and selecting *Apply* in the displayed context menu.

If alternative communication paths exist and you want to use them, expand the possible connection structure in the *possible connections* by double-clicking on it, select the connection and apply it.

Connections that are not required can be removed from the list of configured connections at any time. This is achieved by selecting the connection in the *configured connection* and either:

- selecting the *Edit / Delete* menu or
- selecting *Delete* in the displayed shortcut menu (right mouse button)

Bad connections displayed in the *configured connections* window are shown in red and can result from bad configuration or subsequent changes to a configuration. If there is an incorrect connection between two nodes, the old and no longer valid connection must be deleted from the *configured connections* and the current connection must be taken from the list of *possible connections* again.

Note

If a connection configured in the network configuration is not included in the list of *possible connections* because it is invalid, this is displayed in the "Invalid Connections (Page 130)" list.

Redundant connections

With redundant connections, for example those of a redundant ST7cc-/ST7sc control center, the higher connection in the tree structure is the preferred path and the lower connection is the substitute path. The connection with the preferred path should therefore be applied first.

Saving the connection configuration

Once any invalid connections have been checked and removed and all required connections configured so that they appear in the left-hand window, the connection configuration must be saved with the *SINAUT / Save* menu or the *Save* button in the toolbar. Saving is necessary to store the connections permanently.

If a message is displayed during saving indicating that a connection between two stations configured in NetPro could not be found, the connection must be checked in NetPro and reconfigured.

After saving the connection configuration, open *Subscriber Administration* of the SINAUT Configuration Tool to configure the subscriber data and to generate the system data blocks there.

Changing the connection configuration

By opening the connection configuration again, you can change the scope of the configured connections at any time. By changing the configuration, it is possible that a previously configured connection no longer exists. This then appears in the *Recover lost connections* list that is described separately.

After changing the connection configuration, this must be saved, *Subscriber Administration* must be called and the generation of the system data blocks started.

4.2.2 Invalid Connections

With its algorithms, connection configuration finds all the possible connections in the current project. Connections that do not meet certain rules are displayed for the user in the *Invalid Connections* dialog. The connections contained here are then not included in the list of *possible connections*.

If the connection configuration contains a connection with a red symbol, but the list of invalid connections is empty, this means the following: The connection with the red symbol was configured earlier but has been removed in the meantime.

The *Invalid Connections* dialog is displayed using the *SINAUT / Show Invalid Connections* menu or the *Show Invalid Connections* button in the toolbar.

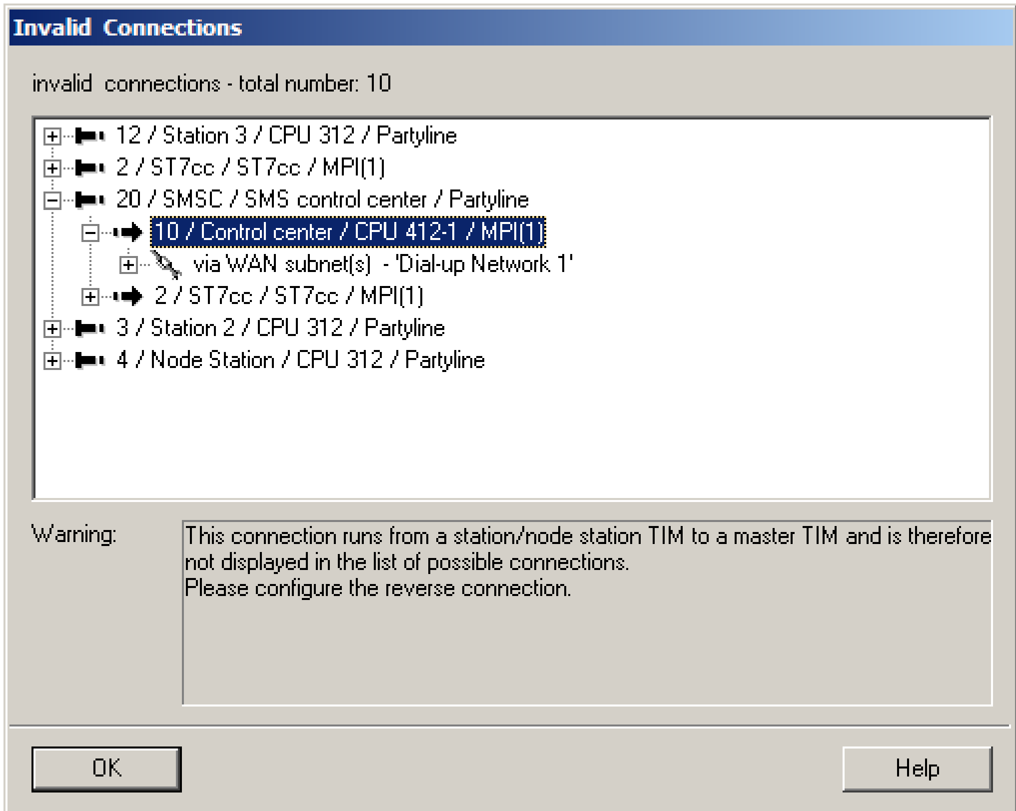


Figure 4-5 *Invalid Connections* dialog

If you expand a connection structure in the list of *invalid connections* by double-clicking on it and if you then select a single connection, the reason for the invalidity and a note on how to remedy the situation are displayed in the lower part of the dialog. The note might, for example, inform you that the connection should be configured in the reverse direction.

Within SINAUT ST7, permitted connections must adhere to the following rules:

- A connection must not run through an inconsistent network. Examples are described along with the plausibility check in the network configuration.
- A connection must not run through a WAN sub-connection on which two MD3 modems communication each other using 1200 bauds / half duplex / AT mode.
- A connection should not run from a station or node station TIM to a master TIM. The reverse direction is preferable.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

Note

To avoid including large numbers of connections unnecessarily in the list of possible and configured connections, some connections used between two subscribers in both directions are shown only in one direction.

A connection displayed in the configuration tool is always valid in both directions.

4.2.3 Recovering lost connections

If changes are made to the connections in the project, it is recommended that you then open the window *Recovering lost connections*. You do this with:

- The *SINAUT / Recovering lost connections* menu or
- The *Recovering lost connections* icon on the right in the toolbar

If changes to the connection configuration mean that previously existing connections are cleared and either closed again, linked to other objects or completely deleted the connection configuration finds all lost connections in the project with its algorithms.

Deleted connections are shown in red in the left-hand part of the *Recovering lost connections* window. Connections that are similar to the deleted connections, are listed in the right-hand part of the window. This allows you to check whether a substitute or successor exists for a lost connection.

If you no longer require the lost connections in the left-hand part of the window, you can delete these by selecting the connections and clicking the *Delete lost connections* button.

If connections that are similar to the last connections are displayed in the right-hand part of the window, you can insert these again if you have accidentally deleted them and still require them. To do this, select the connection in the right-hand part of the window and click the *Recover connections* button.

4.2.4 Selective connection configuration

Selecting subscribers in large projects

The "Selective connection configuration" function was introduced for larger projects.

The selection of individual subscribers increases the clarity of the number of displayed connections. Particularly with large projects, this reduces the time for opening the connection window and reduces the system memory required.

Opening the "Selective connection configuration" dialog

The "Selective connection configuration" dialog opens only if the "Selective connection configuration" option is enabled.

Selective connection configuration is enabled using the "Extras" > "Options" menu in the Options for connection configuration (Page 133) tab.

When you enable the option and close the dialog with "OK", close the connection configuration (for example by calling subscriber administration) and then start connection configuration again, the dialog "Selective connection configuration" opens.

The "Selective connection configuration" dialog

The "Selective connection configuration" dialog shows all subscribers that are connected to at least one other subscriber via SINAUT connections.

Select two (or more) subscribers with the mouse (holding down the <Ctrl> key) and click on "Start".

The connection configuration opens and of the possible connections shows only the ones in which the previously selected subscribers are involved.

Connection symbols

- Black symbols

  The connection is valid and its plausibility has been checked.
- Blue symbols

  A connection with a blue symbol includes a subscriber that was not selected in the "Selective connection configuration" dialog.

Plausibility check: Notes on selective connection configuration

---

Note

Plausibility check

The plausibility of connections with blue symbols is not checked.

- Plausibility check

  To ensure that all configured connections are checked before saving and downloading the STEP 7 project, you should select all subscribers with SINAUT connections at least once in the selective connection configuration.
- In most projects it should be adequate to select the following subscribers for selective configuration at least once so that they are checked for plausibility:
  - The master station

    Most stations generally have a connection to the master station.
  - Stations with inter-station communication

    With connections for inter-station communication between to S7 stations, you only need to select one of the two communications partners in the "Selective connection configuration" dialog.

4.2.5 Options for connection configuration

Printing

To document the configured SINAUT subscribers, you can print out the connection lists in 2 formats. You start a printout with the "Project" > "Print" menu.

With the "Project" > "Print preview" menu you can display a print preview before you print.

Setting the print options:
Make the settings for printing in the "Connection configuration" dialog. You display this with the "Extras" > "Options" menu.

Format of the connection display

Here you select the representation of the connections in the two boxes of the connection configuration and for printing.

By deleting or adding placeholders for key boxes in the three categories Start point, End point and Support point (nodes) you reduce or increase the information density.

Via the key box "&a", for example, the WAN address (station address) or the IP address of subscribers can be displayed.

Configuration mode

- **Selective connection configuration**
  - Option enabled

    If the option is enabled, only the possible connections of the subscribers selected in the "Selective connection configuration" dialog are displayed.

    The selection of individual subscribers increases the clarity of the number of possible connections. Particularly with large projects, this reduces the time for opening the connection window and reduces the system memory required.

    For more detailed information, refer to section Selective connection configuration (Page 132).
  - Option disabled

    If the function is disabled, all possible connections within the project are displayed.
- **Displaying invalid connections**

  If the option is enabled, the invalid connections are also displayed.

  If the "Selective connection configuration" option is enabled, only the invalid connections of the subscribers selected in the "Selective connection configuration" dialog are displayed.

  With larger projects, this saves memory and increases the speed of the display.

4.3 Subscriber administration in the SINAUT configuration tool

Once the SINAUT ST7 connections have been configured, all the requirements are met for the subscriber data for the ST7 communication subscribers

- can be generated,
- displayed,

- processed and
- packed in a form that is understandable for the hardware components; in other words, in data blocks (DBs) or system data blocks (SDBs).

Processing is always necessary when data needs to be acquired that is connection-related; in other words cannot be assigned to a particular subscriber.
Processing of subscriber data is also necessary when SINAUT data is stored for non-SINAUT objects, for example the DB configuration for the CPU modules.
The assignment of parameters for this data is made in the subscriber Administration of the SINAUT configuration tool and is explained below.

After starting subscriber administration, first the previously known data of the subscribers is loaded from the data management and then updated. The following data is updated:

- Subscriber information: Which subscribers exist?
- Networking information: Who communicates with whom over which connections?
- DB configuration information: Which data blocks are generated for a CPU?

The subscriber list always shows the latest situation in the SINAUT subscriber world.

4.3.1 Subscriber list

The subscriber list

In the left-hand pane, the subscriber administration contains a directory tree (structure) of the subscriber types of the project and at the top of the right-hand pane the subscriber list itself.

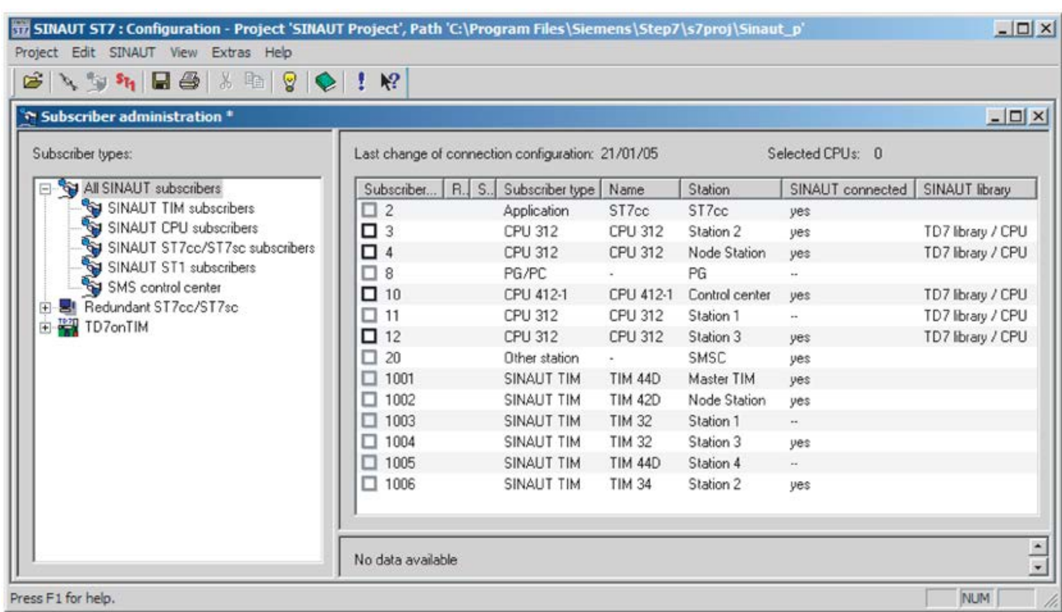


Figure 4-6 Windows of subscriber administration

By making a selection in the directory tree in the left-hand window, only certain subscriber types can be displayed in the subscriber list. The *TD7 on TIM* folder in the directory tree is used to configure the TD7 software for an Ethernet TIM and is not relevant for the subscriber parameter assignment described here.

In the subscriber list on the right, you will see the following entries for the SINAUT ST7 communication subscribers:

- **Subscriber no.**: The subscriber number of the SINAUT subscriber that is unique throughout the project and is required for WAN communication. This has an extra check box to allow the CPU modules to be selected for system data generation.
- **Red. Subscriber no.**: The *Redundant subscriber number* parameter is used only when there is a redundant partner for the subscriber in question. The number specifies the common subscriber number under which the redundant system can be addressed by other subscribers.
- **Subscriber no. of red. Partner**: The *Subscriber number of the redundant partner* parameter is used only when there is a redundant partner for this subscriber. The parameter specifies which of the subscribers belong to a redundant relationship.
- **Subscriber type**: The *subscriber type* specifies the class of subscriber involved. This cannot be changed by the user.
- **Module**: The module, application or PC/PG name. This can be changed in the configuration. As default, this is the name of the module type or the application as specified in the configuration.
- **Station**: Name of the station assigned in the network configuration in *NetPro*.
- **SINAUT connected**: Specifies whether a SINAUT connection was configured for the subscriber.
- **SINAUT library**: Name of the SINAUT software block library for CPU and TD7onTIM-compliant TIM modules.

The setting of the subscriber-related properties is made in the *Properties of subscriber* dialog that is opened when you double-click on one of the subscribers in the list.

Marking for selective system data generation

In the subscriber administration, it is possible to select any CPU subscriber for later selective generation of the software blocks. To select a subscriber, click in the check box in front of the subscriber number (a check mark is set). The number of selected CPU modules is displayed in the text box *Selected CPUs*: above the list. You can remove the selection again by clicking with the mouse.

Changing the subscriber number

To change the subscriber number, click on the required subscriber. By clicking again on the subscriber number, by typing *Alt+Return*, by pressing function key F2 or by selecting the menu *Edit / Change Subscriber*, the *Subscriber number* field becomes editable. The user can then enter any unassigned subscriber number as required.

TD7onTIM dialogs below the subscriber list

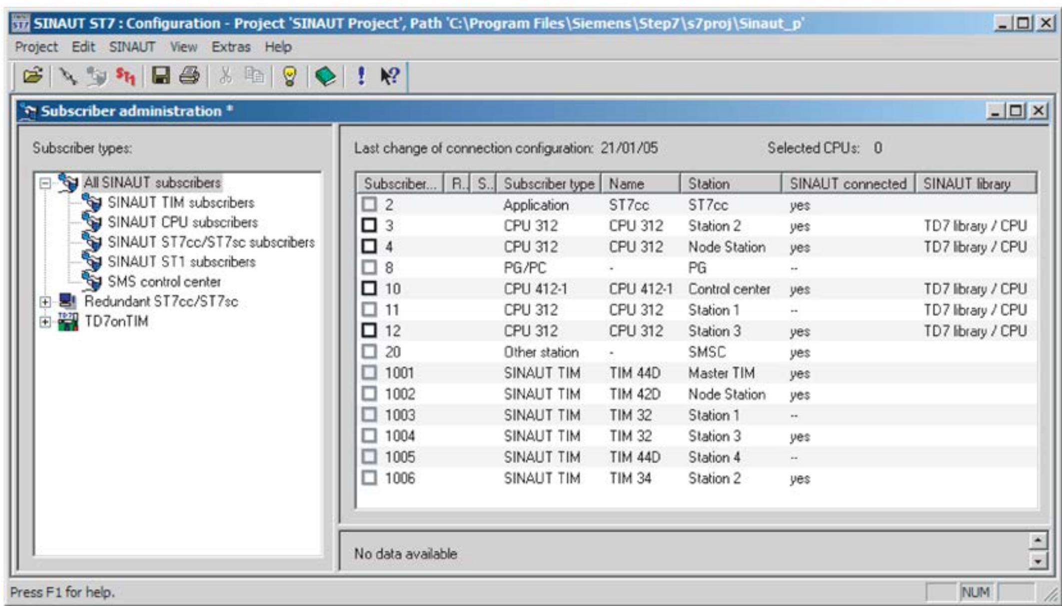
If you use modules that are capable of TD7onTIM in your project, the entry "TIMs with TD7onTIM" appears in the directory tree. The corresponding dialogs of the relevant TIM modules are displayed below the subscriber list and are used to configure TD7 software for the relevant TIM. These dialogs are not available for other subscribers. The TD7onTIM is configured following this in the subscriber configuration.

Standalone TIM with TD7onTIM - Display of the assigned CPU

If you select a standalone TIM with TD7onTIM in the list "TIMs with TD7onTIM", you will find the assigned CPU in the subscriber administration.

In the configuration area below the subscriber list, the following information is displayed:

- Above the "Partner" box. Station name (in the case of a standalone TIM: station name of the CPU)
- Above the "Object" box. Subscriber number of the CPU



4.3.2 Redundant ST7cc/sc control centers

Generating a redundant ST7cc/sc control center

If two SIMATIC PC stations were configured for a redundant SINAUT ST7cc/sc control center, make the assignment of the two redundant partners at this point in the subscriber administration. To do this, select the entry "Redundant ST7cc/ST7sc server" in the directory tree and select "Add redundant ST7cc/ST7sc..." from the shortcut menu (right mouse button). The "Properties - ST7cc/ST7sc redundant" dialog - opens

"Properties - ST7cc/ST7sc redundant" dialog

This dialog allows you to generate a redundant ST7cc/ST7sc network. To do this, an ST7cc/ST7sc subscriber must be selected in each of the "ST7cc / ST7sc 1" and "ST7cc / ST7sc 2" list boxes.

The subscriber number of the redundant grouping can then be set. This is always identical to one of the two subscriber numbers of the selected subscribers.

A name for the redundant grouping can also be specified.

After you have confirmed the generation of the grouping with "OK", it appears in the tree on the left as "Redundant ST7cc/ST7sc server".

Deleting a redundancy grouping

You can delete a redundancy grouping again by selecting the entry of the redundancy grouping in the directory tree and then selecting the "Delete redundant object" shortcut menu command. This deletes only the data of the grouping but not the underlying subscribers.

For detailed information on configuring a single or redundant ST7cc/ST7sc control center, refer to the SINAUT ST7cc or ST7sc documentation.

Communication with redundant ST7cc/sc control centers

The communication of a TIM module with redundant ST7cc/sc control centers depends on the type of TIM:

- Classic TIM modules(TIM 3, TIM 4)

  If the connection between a classic TIM and the two control center PCs is established, the TIM sends every message to both PCs of the redundant control center.

  If the connection is established to only one control center PC, the TIM sends the message only to the connected PC. When the acknowledgment is received from the connected control center PC, the sent message is deleted from the send buffer.

  When the connection to the second PC is established again, the frames that were sent to the partner PC during the downtime are no longer sent to the reconnected PC. The archives of the two PCs differ following connection failures.

- Ethernet TIMs

  As of firmware version 2.1 of the Ethernet TIMs, an Ethernet TIM behaves as follows:

  The Ethernet TIM sends the messages to both PCs of the redundant control center only when one of the two following conditions is met:

  – The connections between the Ethernet TIM and both control center PCs are established.

    or

  – Only one connection is established and a wait time that corresponds to the keepalive timeout has elapsed. This wait time is started as soon as the first connection is established and is stopped as soon as the connection to the second PC is established.

  The effect of this behavior is that in particular brief connection failures do not lead to archive differences on the two control center PCs.

4.3.3 Configuring the GSM providers for MSC stations

Configuring GPRS providers

If you want to use the GPRS service with the MSC protocol, you will need to configure at least one GSM network provider for the MSC subscribers. To avoid having to call up the dialogs for configuring the MSC subscribers more than once in the SINAUT configuration tool, we recommend that you configure the providers first. Click on the globe icon ( ).

The "List of GPRS Providers" dialog opens.

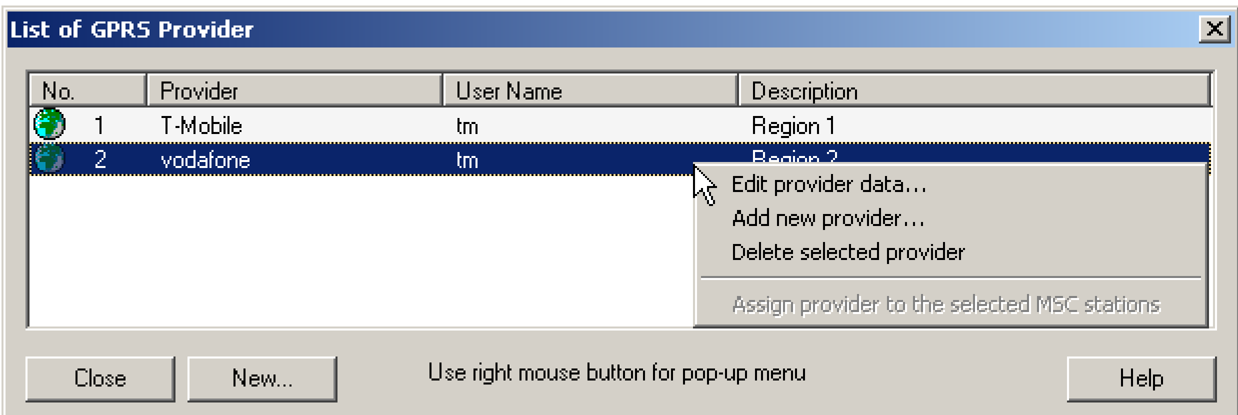


Figure 4-7 "List of GPRS Providers" dialog box

Creating GSM network providers in the "List of GPRS Providers" dialog

This dialog displays the configured GSM network providers with their names, the user name and a description.

Apart from the "New..." button for creating a new provider (see below), you can also use the following four shortcut menu commands in the dialog:

- **"Edit provider data..."**

  The "Provider Properties" dialog opens. Here, you can configure the properties required for the GPRS provider (see below: The "Provider Properties" dialog).

- **"Add new provider..."**

  The "Provider Properties" dialog opens. Here, you can configure a new GPRS provider with the required properties (see below: The "GPRS Provider Properties" dialog).

- **"Delete selected provider"**

  The provider you have selected is deleted from the list.

- **"Assign provider to the selected MSC stations"**

  See below for a description.

Meaning of the globe icon

The globe icon in this dialog has a the following meaning depending on its color:

- Green ( )

  All the required properties of the provider are configured.

- Black and white ( )

  The configured data of the provider is not complete.

The "GPRS Provider Properties" dialog

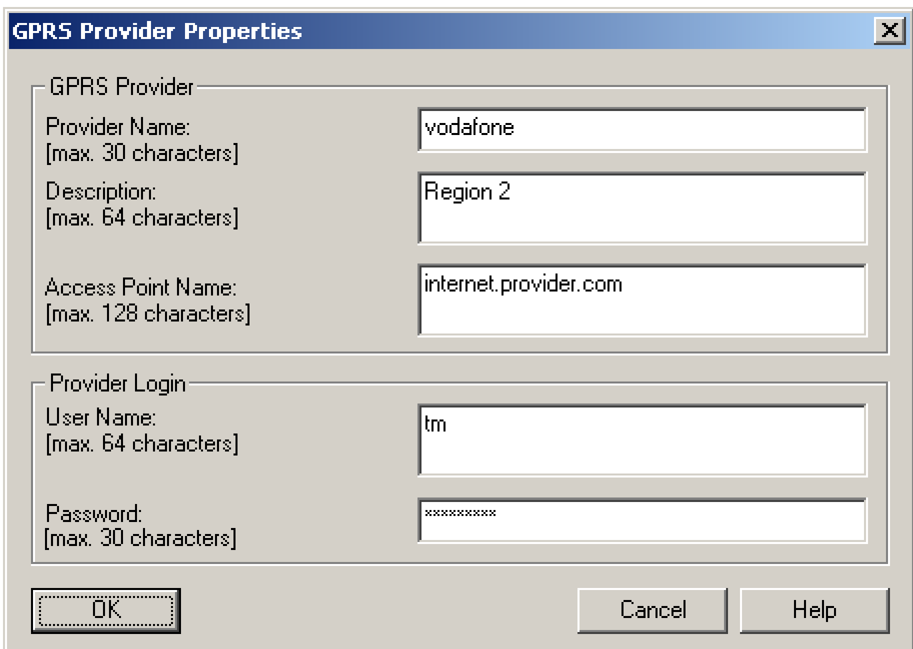


Figure 4-8 The "GPRS Provider Properties" dialog

Configure the following properties for the GSM network provider:

- Provider name

  Here, you need to assign a name for the provider. You can choose any name.

- Description

  Here, you can enter a description of the provider if required. This could, for example, be a assignment to the stations being served.

- Name of the Internet access

  Here, enter the name of the Internet access (APN) of your provider in the GPRS network. The provider will give you this name.

- User name *)

  Here, enter the user name with which the connection to the provider will be established. The provider will give you this name.

- Password *)

  Here, enter the password required to log on with the provider. If the provider does not demand a password, you could, for example, enter the word "guest" here.

*) There must be no blanks in the user name or password.

"Delete selected provider"

If you select a provider in the "List of GPRS Providers" dialog, and then select the "Delete selected provider" shortcut menu command, the provider will be deleted from the list.

Deleting is only fully complete after you exit the dialog with "OK".

"Assign provider to the selected MSC stations"

You can only assign a GPRS provider to one or more stations when the properties dialog of this station is open in the SINAUT configuration tool at the "MSC Station List" and you have selected the relevant menu command there.

The provider assignment is described in the following section:

"MSC Station List" tab (Page 160)

4.3.4 The "Properties of subscriber" dialog

4.3.4.1 Parameter overview

Opening the dialog

Open the "Properties of subscriber" dialog as follows:

- Double-click on a subscriber in the subscriber list of the subscriber administration
- Selection of a subscriber in the subscriber list and selection of the "Edit / Properties" menu

Overview of the tabs of the "Properties of subscriber" dialog

The content and number of tabs of the dialog depend on the subscriber type.

Table 4- 2 Overview of the Properties dialog tabs according to subscriber type

Tab	CPU	TIM	Other station, SIMATIC S5, PC station, PG	Names of follow-up dialogs
Info	yes	yes	yes	-
Partner	yes	yes	-	-
Connections	yes	yes	-	Properties - Local Connection
Polling List	-	Master TIM	-	Properties - Poll list entry
MSC station list	-	Master TIM (MSC master)	-	Properties - MSC master Properties - MSC station Provider assignment
Telephone Directory	-	When necessary	-	Properties - Telephone number
DB Configuration	yes	-	-	-
SMS Configuration	When SMS center configured	-	-	SMS CPU Configuration, SMS DB Data, SMS Message Data
Library Information	yes	-	-	-

4.3.4.2 "Info" tab

Info tab

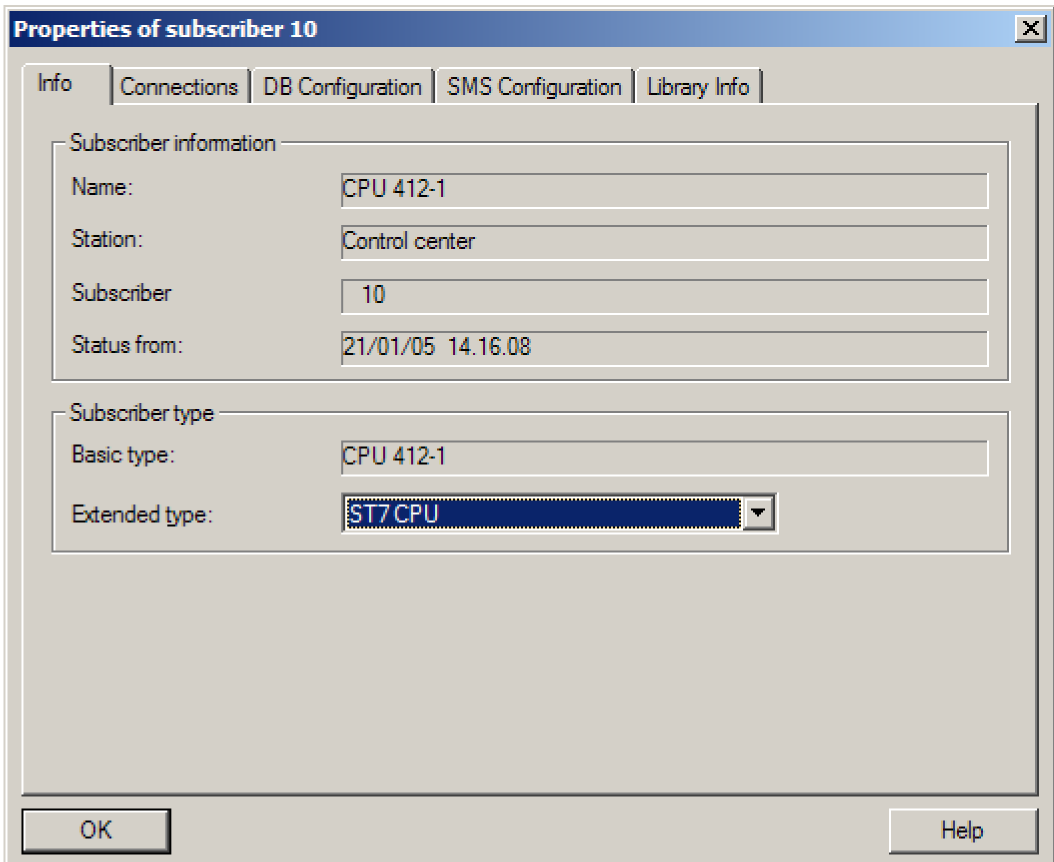


Figure 4-9 *Properties of subscriber* dialog (CPU), *Info* tab

The *Info* tab displays the following information on the selected subscriber:

- *Name* shows the default name of the module or the name assigned in network configuration
- *Station* displays the set network node type
- The *subscriber number* is displayed
- *Status from* displays the date of the last configuration
- The *date created* is displayed.
- *Basic type* displays the network object type from the network configuration
- *Extended type* displays the extended network object type adopted in the network configuration (for example ST7-CPU, ST7-TIM, ST7cc or SMS center)

4.3.4.3 "Partners" tab

Partners tab

This tab lists the connection partners of the selected module with subscriber number, module type, name and station name.

4.3.4.4 "Connections" tab

Connections tab

In this tab, all configured local connections via LAN with their most important properties are listed for the selected subscriber:

- X connections

  Unconfigured S7 connections that use the SFCs "X_SEND" and "X_RCV".
- PBC connections

  Configured S7 connections that use the SFBs "BSEND" and "BRCV".
- CR connections

  Read/write connections of the TD7 software "TD7onTIM" to the local CPU that do not require S7 connections.
- MSC connections

  Connections for the MSC protocol that do not require S7 connections.

Note

During the analysis of the subscriber data, if the configuration software detects that STEP 7 homogeneous connections are necessary for processing a SINAUT ST7 connection, these are created automatically. These are connections from the S7-400 CPU to a TIM module and from TIM to TIM module over the MPI bus and over communication block connections.
As an alternative, the user can create these connections manually during network configuration. Existing connections are automatically used by the SINAUT configuration tool.

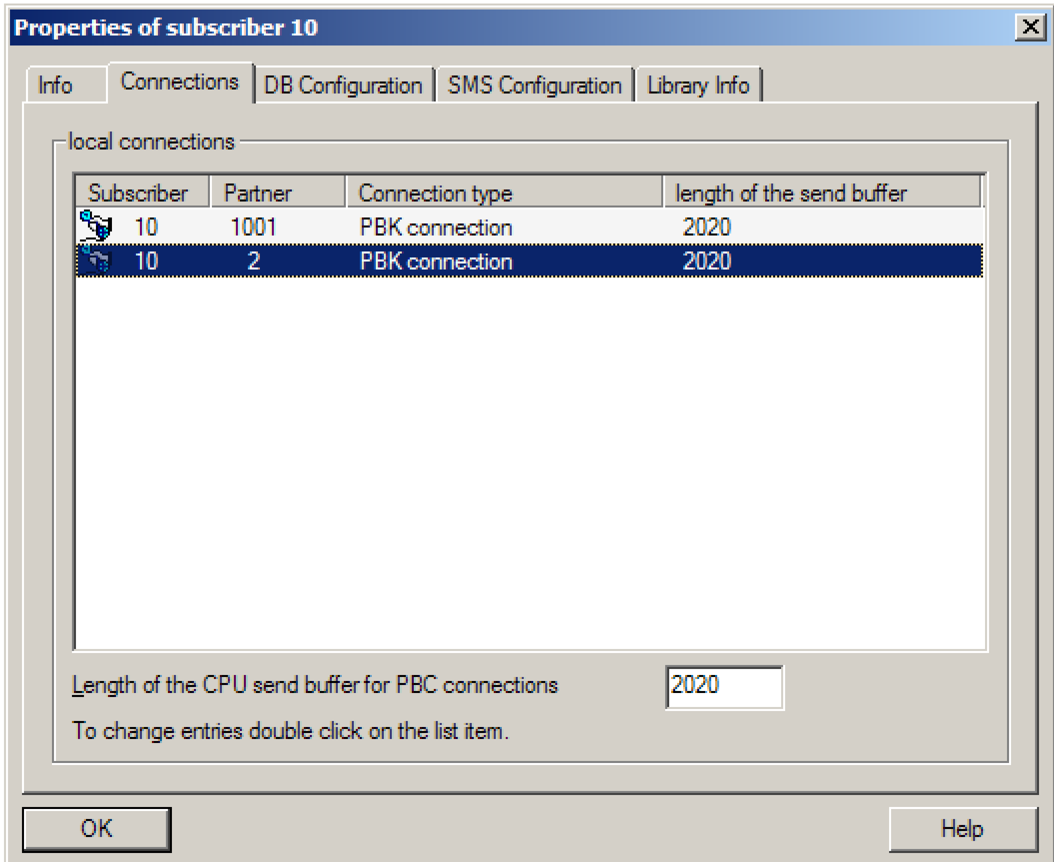


Figure 4-10 "Properties of subscriber" dialog (here an S7-400 CPU), "Connections" tab

Here, you can configure the following:

- The length of the CPU send buffer for any existing communication function block connections of a CPU. This is the same for all communication function block connections of the current CPU.
  Range of values: 202 .. 65208 bytes
  Default: 2020 bytes

By double-clicking on a subscriber row in the *local connections* output box, you open the *Properties - Local Connection* dialog for this connection.

Properties - Local Connection dialog for the TIM

This dialog displays the properties of the local connection of a TIM module.

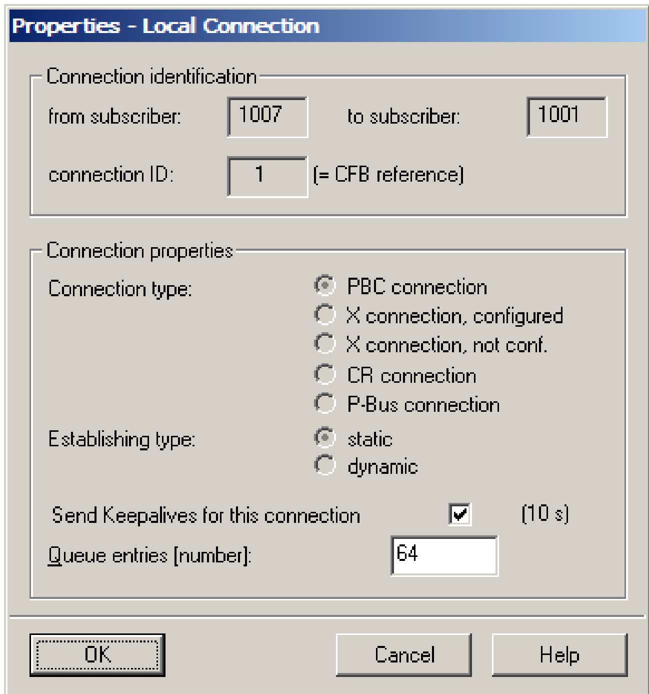


Figure 4-11 "Properties - Local Connection" dialog (here connection TIM to TIM)

For the local connection of the TIM module, you can set the following here:

- *Send Keepalives for this connection:*
  If this option is enabled, keepalive messages are sent on this connection at the intervals set in the network configuration for the TIM.
- *Queue entries [number]*:
  Range of values: 10 .. 256
  Default: 64

  The number of queue entries is the number of messages that the TIM module can buffer prior to transmission. In general, you do not need to change the default setting. It may be useful to increase the value for a master TIM to relieve message traffic if the TIM has a heavy load at certain times due to the transfer of large amounts of data, for example archive data.

Properties - Local Connection dialog for the CPU

This dialog visualizes the properties of a local LAN connection of a CPU and allows the following properties to the selected:

- CPU modules with X connections / P bus connections:
  - The *length of the send buffer [bytes]* for these connections
    Range of values: 76 ... 65382 bytes

Default: 760 bytes
This parameter is not relevant for PBC connections.

– The *Connection monitoring time [s]*; in other words, the time that must elapse before dummy messages are sent to check the connection.
Range of values: 1 ... 32 s
Default: 5 s

- CPU modules with PBC connections:

– The *Connection monitoring time [s],*
Range of values: 1 ... 32 s
Default: 5 s

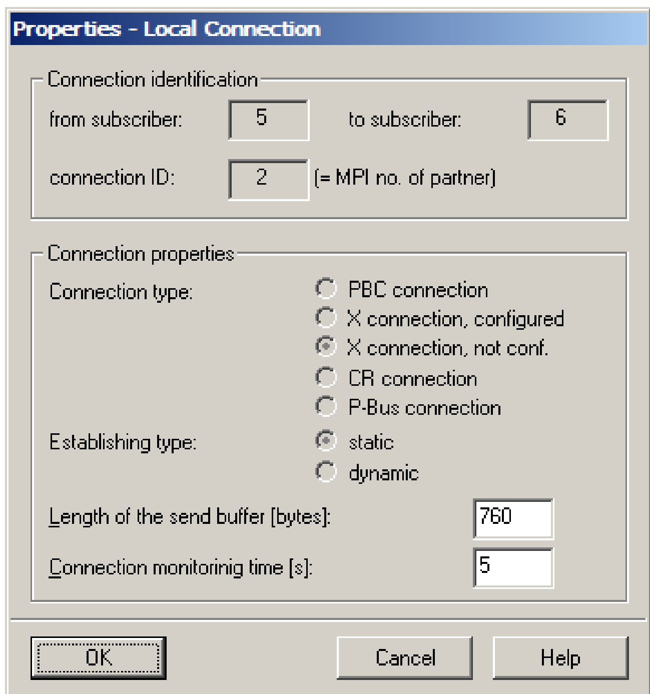


Figure 4-12 *Properties - Local Connection* dialog (CPU)

4.3.4.5 "Polling List" tab

Polling List tab

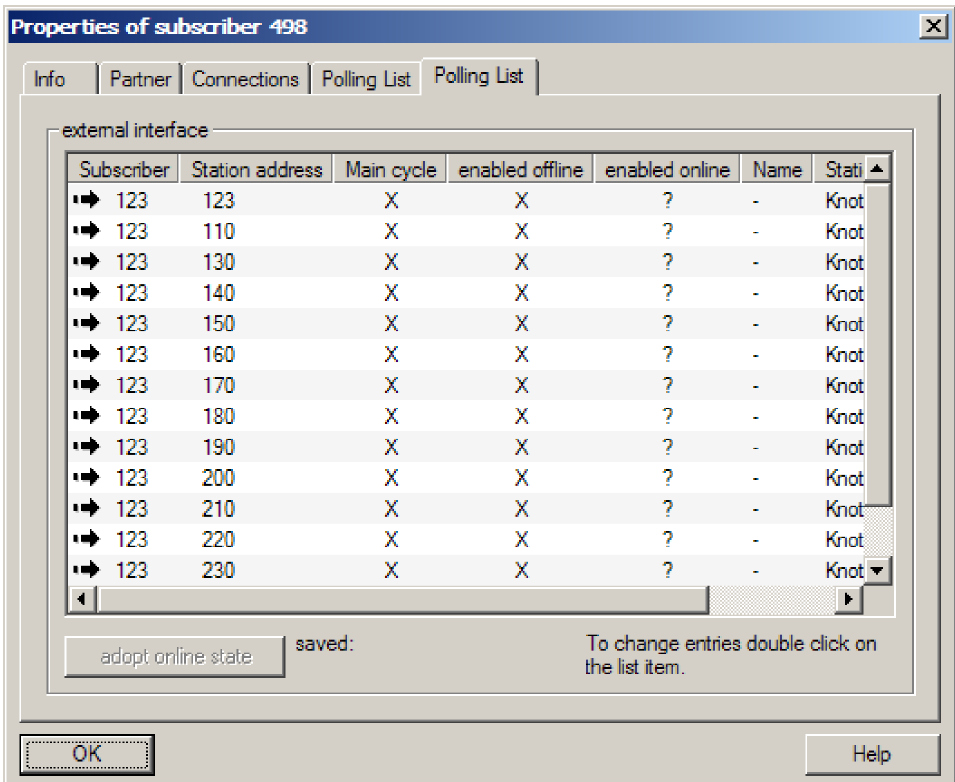


Figure 4-13 *Properties of subscriber* dialog (TIM), *Polling List* tab

This tab for a TIM module displays the TIM modules that can be polled by this station in polling mode with the subscriber name, station address, ID for main cycle poll, enable status offline, enable status online, module name, and station name.

By clicking *adopt online state*, the entire online status is adopted in the configuration. This function is available only for the TIM 4R-IE.

By double-clicking on a subscriber role in the *internal interface* or *external interface* output box, the *Properties - Poll list entry* dialog for this connection opens.

Properties- Poll list entry dialog

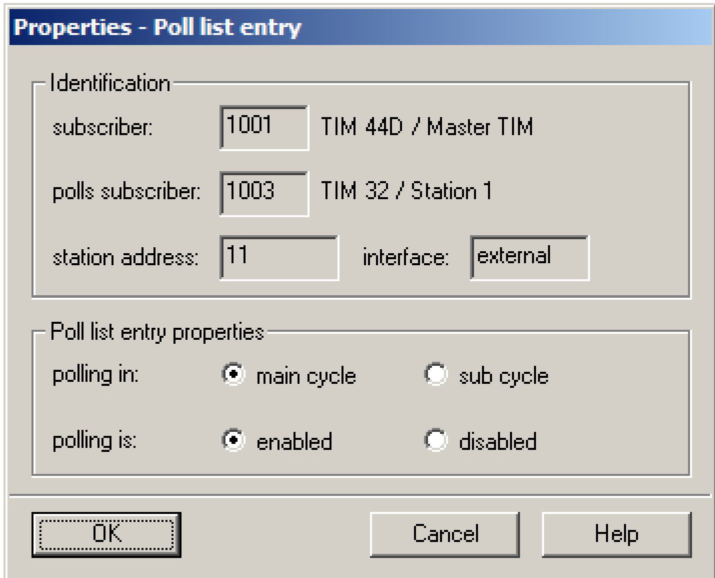


Figure 4-14 *Properties- Poll list entry* dialog

The *Identification* area displays the following properties:

- *subscriber*:
  The *subscriber* output box displays the polling subscriber (master TIM) with its subscriber number, module name, and station name.
- *polls subscriber*:
  The *polls subscriber* output box displays the polled subscriber with its subscriber number, module name, and station name.
- *station address*:
  The *station address* output box displays the station address of the polled subscriber.
- *interface*:
  The *interface* output box displays the interface type (internal/external) of the polled subscriber.

The *Properties* box displays the following options:

- *polling in*:
  Options: *main cycle*, *sub cycle*
  Depending on the selected option, the CPU module is polled in the main or sub cycle.
- *polling is*:
  Options: *enabled*, *disabled*
  If the *enabled* option is selected, polling the CPU module in polling mode is enabled. Otherwise polling is disabled.

4.3.4.6 "Telephone Directory" tab

Telephone Directory tab

This tab of a TIM module displays the subscribers of a dial-up network with the subscriber number, station address, dial string, enable status offline, enable status online (TIM 4R-IE only), module name and station name. If the TIM module is connected to two dial-up networks as is the case with a node station, the subscribers of both dial-up networks are listed.

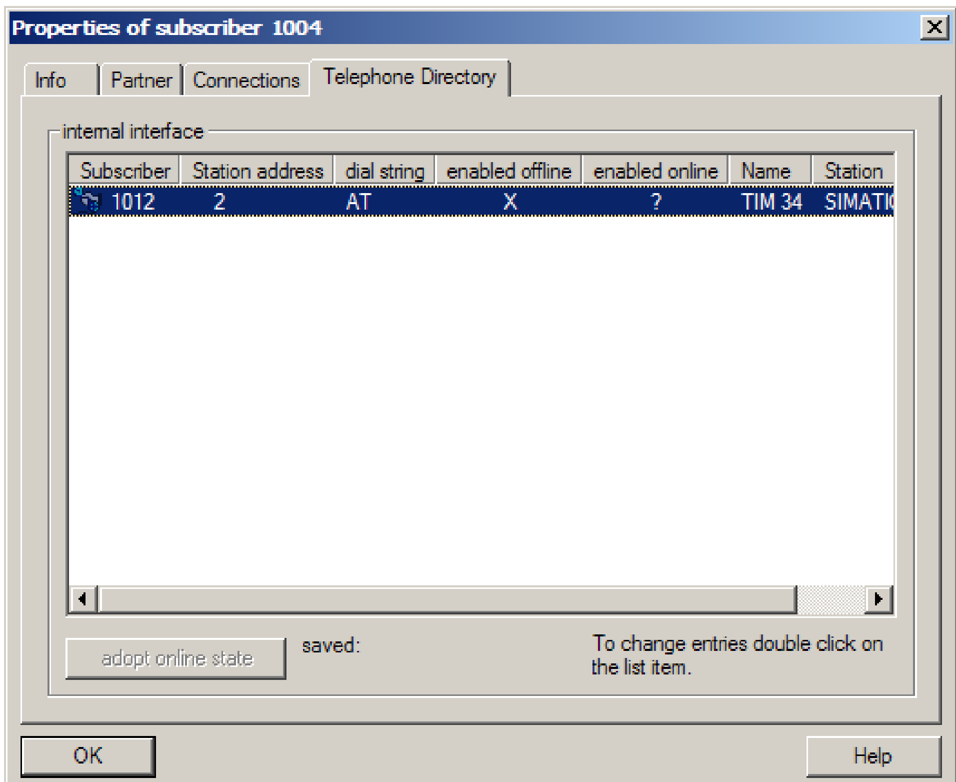


Figure 4-15 *Properties of subscriber* dialog (TIM), *Telephone Directory* tab

By clicking adopt online state, the entire online status is adopted in the configuration. This function is available only for the TIM 4R-IE.

Double-clicking on a subscriber in the *internal interface* output box opens the *Properties - Telephone Number* dialog for the connection of the subscriber selected in subscriber administration to the subscriber selected here in the *Telephone Directory* tab.

Properties- Telephone Number dialog

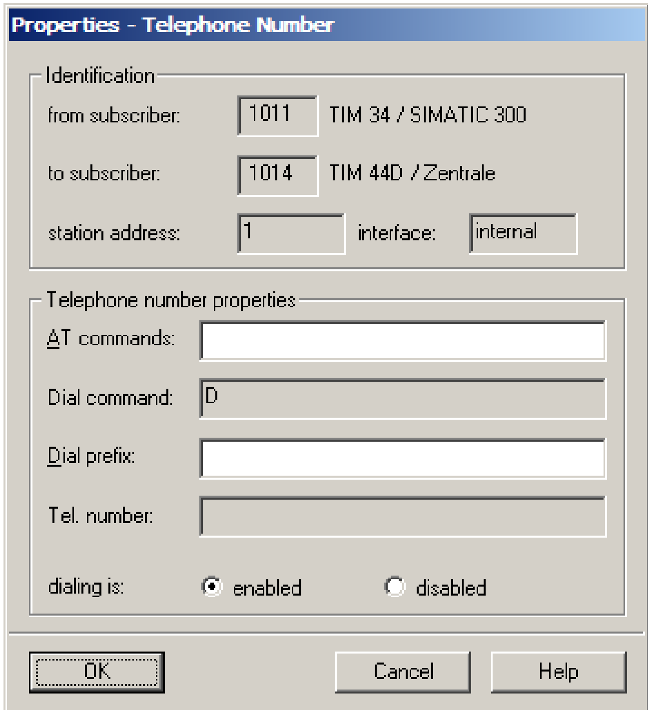


Figure 4-16 *Properties- Telephone Number* dialog

Double-clicking on a subscriber in the list in the *Telephone Directory* tab opens the *Properties - Telephone Number* dialog. This dialog is used when connection-specific modifications to the telephone number are necessary. Examples might be the unlocking of telephone numbers using AT commands or the use of different telephone service providers for connections.

The *Identification* area at the top displays the following information:

- *from subscriber*:
  Source subscriber of the connection
- *to subscriber*:
  Destination subscriber of the connection
- *station address*:
  Station address of the destination subscriber
- *interface*:
  Interface type (internal/external) of the destination subscriber

In the lower *Properties* box, you set the following properties:

- *AT commands*:
  Here, you can set the connection-specific initialization of the modem. The specified AT command is output before the dial command for this number. The AT commands entered must be specified without the *AT* string.
- *Dial command*:
  Displays the configured dial command.

- *Dial prefix*:
  Displays the configured dial prefix. This can be modified.
- *Tel. number*:
  Displays the telephone number of the destination subscriber entered in the network configuration.
- *dialing is:*
  Options: *enabled*, *disabled*
  if the *enabled* option is selected, the dial-up connection is enabled. Otherwise, the dial-up connection is disabled. This function is available only for the TIM 4R-IE.

4.3.4.7 "DB Configuration" tab

DB Configuration tab

This tab displays the type and number of data blocks required for this CPU type.

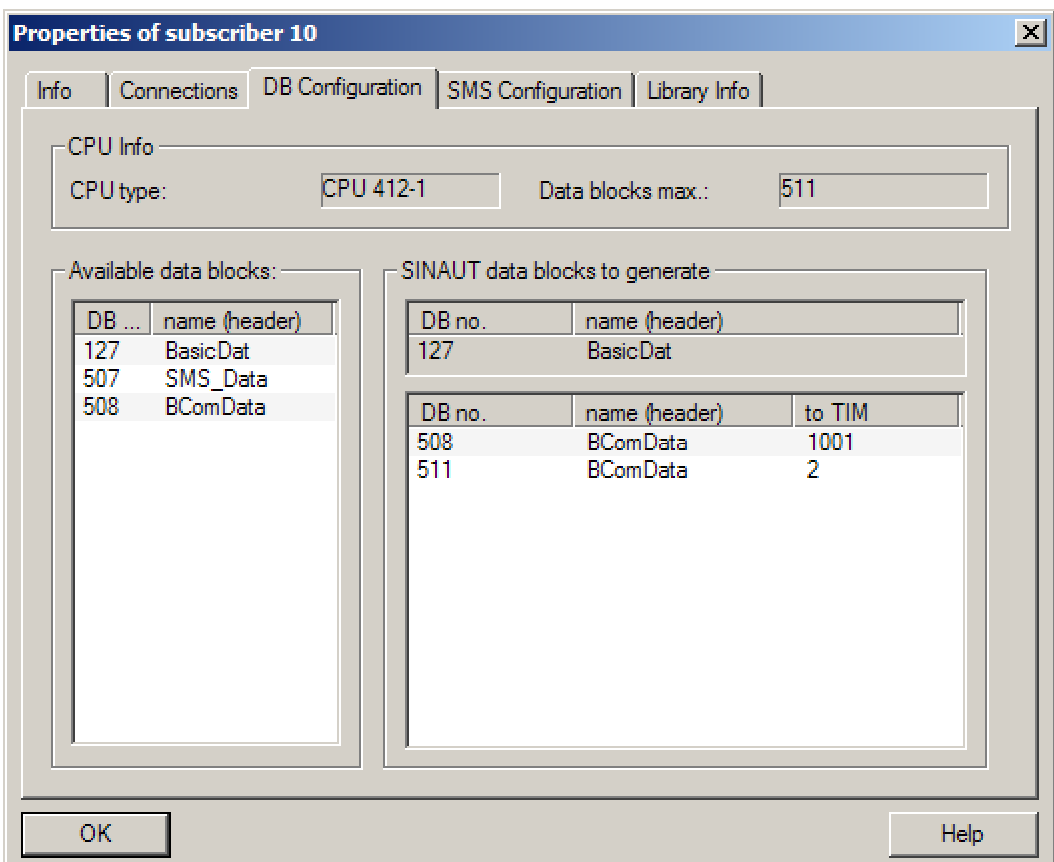


Figure 4-17 *Properties of subscriber* dialog (CPU), *DB Configuration* tab

The following is displayed:

- *CPU type*:
  Type of the current CPU
- *Data blocks max.*:
  Maximum number of data blocks for this CPU (highest DB no.)
- *Available data blocks*:
  DB no. and *name (header)* of the existing data blocks in the offline data management of this CPU. These DB numbers can no longer be assigned.
- *SINAUT data blocks to generate*:
  - *DB no.* and *name (header)* of the DB BasicData:
    This number is either read from the symbol table or is assigned the default 127. The number can only be modified if the complete SINAUT program with the supplied sources is recompiled.
  - *DB no.* and *name (header)* of the communication data blocks and in the column *to TIM* the subscriber number of the partner TIM.
    Based on the existing blocks and the maximum value, the program proposes the numbers for all required data blocks. The user can change these numbers when necessary.

4.3.4.8 "SMS Configuration" tab

Validity of the tab

This tab is only relevant for SMS messages sent using TD7onCPU.

- SMS messages sent via a connected dial-up network.
- SMS messages sent by a station connected via a dedicated line, dial-up network or Ethernet and that has a connection to the SMSC.

  Requirements are:
  - At the end of the connection an SMSC must be configured.
  - An S7 connection to the SMSC must be configured.
  - Within the S7 connection, the ST7 protocol is used (not MSC).

SMS Configuration tab

In the *SMS Configuration* tab, you configure all the SMS messages required for the current CPU. This is only possible if the corresponding DB SMS_Control was created on the CPU. Several SMS data blocks can be defined per CPU and these can in turn contain several SMS messages.

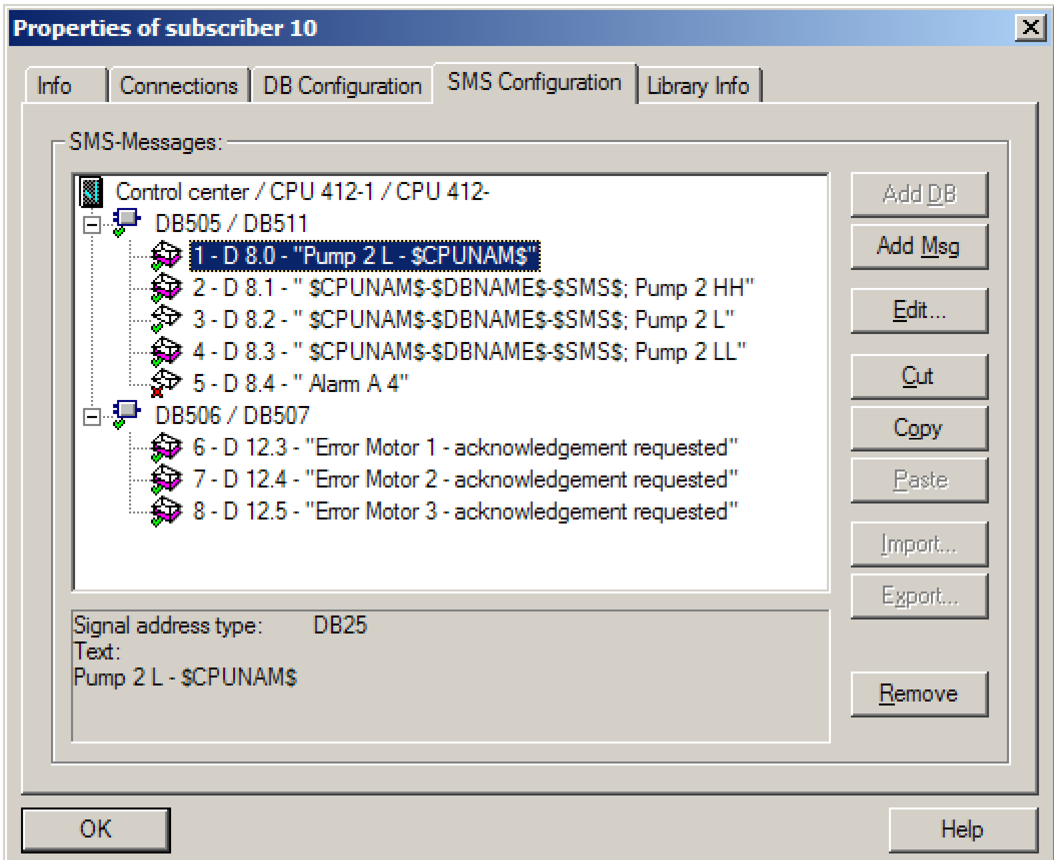


Figure 4-18 *Properties of subscriber* dialog (CPU), *SMS Configuration* tab

In the tree display on the left-hand side, you can see the configured SMS messages in a tree structure with the following information:

- The highest level of the tree displays the current CPU with the text:
  STEP 7 name of the CPU / type of the CPU / CPU short name
- The second level contains the configured data blocks with the text:
  Block number / DB short name
- The third level lists the messages of the relevant DB with the text:
  Message number - signal address (byte.bit) - "Message text"

In the tree display, the status of the object is indicated by the type of symbol used.

Table 4- 3 Symbols of SMS Configuration

	Correctly configured DB, data will be generated
	Incompletely configured DB, data will not be generated
	activated SMS message
	deactivated or empty SMS message
	activated SMS message with coming/going status
	deactivated or empty SMS message with coming/going status

Below the tree display there is an information area in which the following characteristic data of the object selected in the list is displayed:

- Selection of a *CPU*:
  subscriber number and current size of an SMS message
- Selection of a *DB*:
  number of signals, current DB size, message numbers in this DB, main mobile wireless number and backup mobile number
- Selection of a *message*.
  Signal address type and message text

To the right of the tree display of the messages, there are buttons that trigger the available processing functions. The available functions can also be started using the shortcut menu (right mouse button) when an object is selected. All functions are only possible when a CPU, a DB, or a message was selected in the *SMS messages* list.

The range of active functions depends on the selected object. The functions available in SMS configuration are as follows:

- *+ DB* (Add DB) :
  A new data block is added. A free number is searched for as the DB number, starting at the maximum number for the current CPU.
- *+ Message* (Add Msg) :
  A new empty message is added to the current DB. New messages are deactivated as default and must be activated before they can be sent.
- *Edit...* :
  The CPU, a DB or an SMS can be edited. Depending on the selected object, one of the following dialogs opens when the *Edit* function is activated:
  – Edit CPU: The *SMS CPU Configuration* dialog opens.
  – Edit DB: The *SMS DB Configuration* dialog opens.
  – Edit SMS: The *SMS Message Configuration* dialog opens.
- *Cut*:
  The currently selected message is cut and can be inserted again later with the *Paste* function.

- *Copy*:
  The selected message is copied.
- *Paste*:
  The last message to be copied or cut is inserted.
- *Import*:
  A file created with the export function is imported into the selected DB.
- *Export*:
  The data of the current DB is exported to a file. The file format used is the Excel-compatible CSV format; in other words, the exported data can be edited in Excel.
- *Remove*:
  The current object is deleted. With data blocks, you are prompted for confirmation; messages are deleted without a prompt for confirmation.

SMS CPU configuration dialog

If you select a station in the *SMS Configuration* tab and click the *Edit* button, the *SMS CPU configuration* dialog opens. Here, you can see the data of the current CPU that are relevant for SMS configuration.

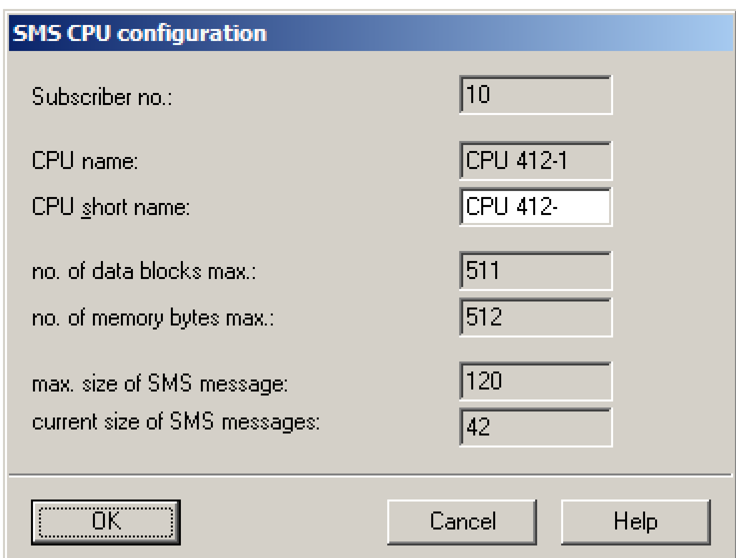


Figure 4-19 *SMS CPU configuration* dialog

For the CPU, you can specify a name with up to eight characters (*CPU short name*). This short name can be inserted with a placeholder string in SMS messages later during SMS *message configuration*. As default, the short name has the first eight characters of the CPU name.

SMS DB data dialog

If you select an SMS DB in the *SMS Configuration* tab and click on the Edit button, the *SMS DB data* dialog opens. This dialog displays and allows you to configure the data of the selected SMS DB.

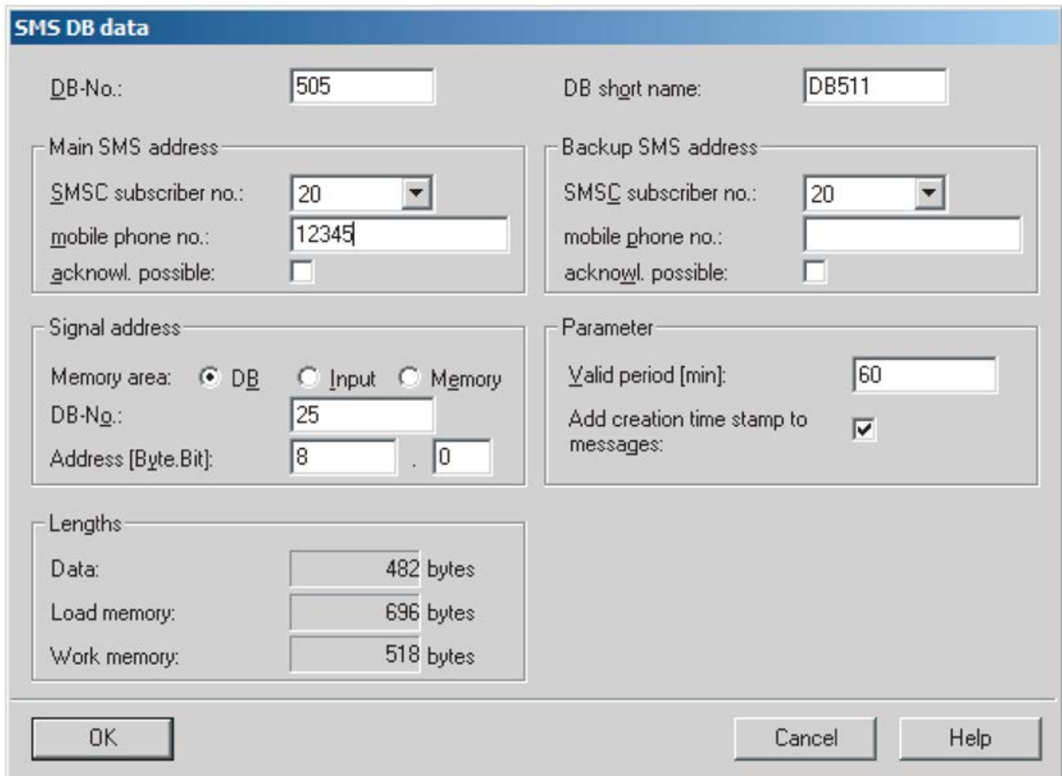


Figure 4-20 *SMS DB data* dialog

The following parameters are available:

- *DB No.*:
  The number used for the SMS DB. The number can be modified, the availability of the specified number on the current CPU is checked. If the number is already assigned by a DB of the CPU user program, an error message is displayed. SMS DBs on the other hand can be overwritten.
- *DB short name*:
  As default, the SMS DB short name has the first eight characters of the CPU name. For the SMS DB, you can specify a DB Short name with up to eight characters. This DB short name can be inserted with a placeholder string in the message text of SMS messages later during *SMS message configuration*.
- In the field *Main SMS address*:
  – *SMSC subscriber no.*: The subscriber number of the SMSC configured in subscriber administration
  – *mobile phone no.*: The mobile phone number to be used for the SMS recipient
  – *acknowl. possible*: The recipient can acknowledge (option selected) or not

- In the field *Backup SMS address* (optional):
  - *SMSC subscriber no.*: The subscriber number of the SMSC configured in subscriber administration
  - *mobile phone no.*: The mobile phone number to be used for the SMS recipient
  - *acknowl. possible*: The recipient can acknowledge (option selected) or not
- In the field *Signal address*:
  The signal address triggers the SMS message. As signal address, bits of a data block (DB), inputs or memory bits can be used. Per SMS data block, the data type of the signals (*address type*), the *DB no.* and the start address within the data field (*address*) must be specified.
  The signals must be located in a contiguous data field, for example in a data block. The first bit triggers the first SMS message, the nth bit triggers the nth SMS message.
  - Address type (DB, Input, Memory)
  - DB no. (only with an addressed type *DB*)
  - Address (Byte.Bit)
- In the field *Parameter*:
  - *Valid period*:
    The period of validity specifies the time within which an SMS message should be delivered and, if set in the *SMS Message Configuration* dialog, must also be acknowledged. If this has not taken place when the time expires, a diagnostic message is entered in the diagnostic buffer of the CPU.
  - *Add creation time stamp to messages* option:
    In addition to the time added by the SMS center, the creation time of the triggering event can also be sent if this option is selected.
- In the *Lengths* field:
  Here, the length of the DB to be created is calculated based on the currently available *data* and the resulting *Load memory* and the *Work memory requirements*.
  There is no check to make sure that the DB can actually be loaded on the CPU. This is the responsibility of the user.

SMS Message data dialog

Note

Blocking special characters

When configuring SMS messages, a function is available with which you can check the SMS character set. When entering text for SMS messages, this function blocks certain special characters that can cause problems with some SMS providers.

The function is enabled in the "Options" dialog ("Extras" > "Options" menu "Subscriber administration" > "SMS Configuration" tab.

Before you configure SMS messages for a subscriber, you should therefore check whether or not this function should be activated. When this function is active, only the following characters can be entered:

- All digits
- All letters (except umlauts ä, ü etc.)
- The space and the following special characters:
  ! # % & / ( ) ? * + - . , : ; < = >

All other special characters and umlauts (ö, Ö, ü, Ü, ä, Ä) are blocked.

If you select a message in the *SMS Configuration* tab and click the *Edit* button, the *SMS message configuration* dialog opens. You configure an SMS message in this dialog.

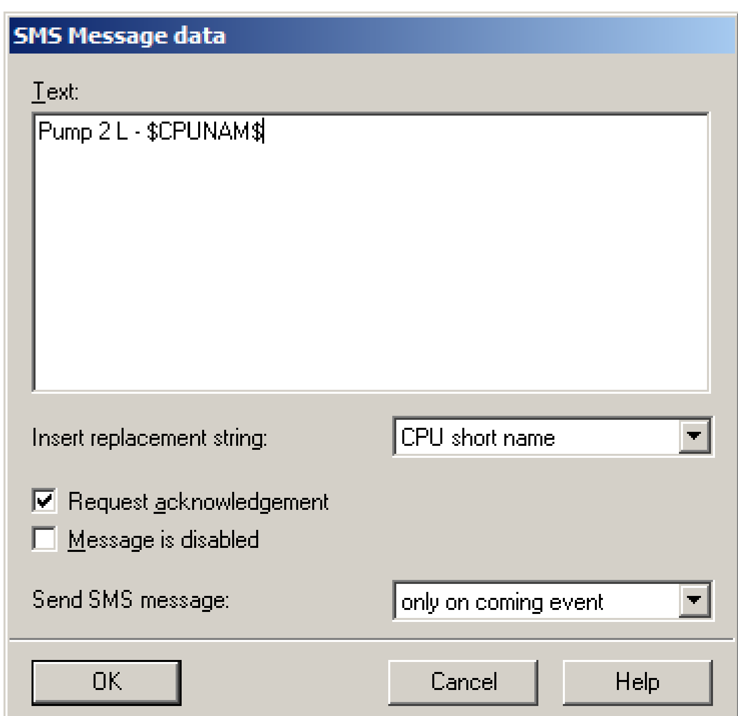


Figure 4-21 *SMS Message data* dialog

The configuration options are available:

- *Text*:
  The message text of the SMS message is entered in the *Text* input box. 120 characters can be entered.
- *Insert replacement string*:
  With this list box, you can insert replacement strings as keywords in the text of the SMS. The following replacement strings can be inserted in the text:
  - CPU short name ($CPUNAM$):
    The CPU short name is configured in the *SMS CPU configuration* dialog
  - DB short name ($DBNAME$):
    The DB short name is configured in the *SMS DB data* dialog
  - Message no. (SMS):
    The message number is a unique number per CPU that is assigned by the SINAUT configuration tool. It can be seen in the tree display of the *SMS messages* list of the *Properties of subscriber* dialog.

  The replacement strings are replaced by the actual values when the SMS is generated. By specifying the 3 strings, every SMS message in an S7 project can be uniquely identified.
- *Request acknowledgment* option:
  You can specify whether or not an acknowledgment is required for this message.
- *Message is disabled* option:
  If message output is activated, this option must be disabled (no check mark).
- *Send SMS message*:
  The list box indicates whether the message is sent on an event entering state (coming) or an event leaving state (going).

4.3.4.9 "MSC Station List" tab

Requirement for configuring the MSC subscribers

Before you configure the MSC subscribers in the properties dialog of the MSC master, you have already performed the steps listed below:

- Networking of the interface of the TIM module and assignment of the network node type MSC master, MSC node station or MSC station
- Configuring the GSM network provider

You will find an configuration overview of "simple Internet communication" in the section GPRS / Internet communication: Overview of configuration (Page 36).

Information in the "MSC Station List" tab

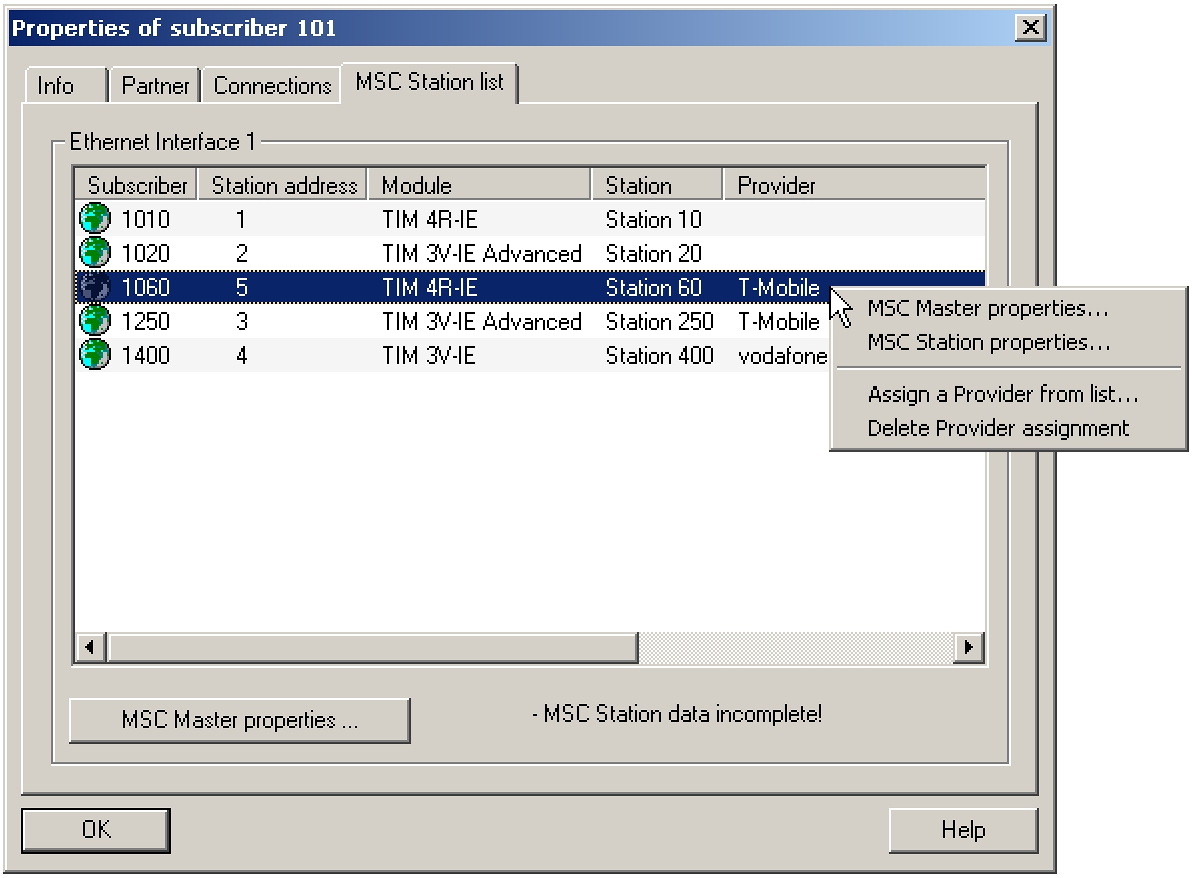


Figure 4-22 "Properties of Subscriber" dialog of the MSC master, "MSC Station List" tab

The "Ethernet interface 1" table shows all MSC stations of the project and specifies the subscriber number, station address, module type, station name (from NetPro) and provider name. The globe icon at the start of the row shows the status of the GPRS provider.

The globe icon in this dialog has a the following meaning depending on its color:

- Green ()

  All the required properties of the station are configured.

- Black and white ()

  The configured data of the station is not complete.

  If you select subscribers that have not been fully configured, the incomplete data is shown in the lower part of the dialog.

Configuration in the "MSC Station List" tab

Follow the steps below to configure the MSC subscribers:

1. Configure the data of the MSC master by clicking the "MSC Master Properties..." button.
2. Configure the data of the MSC stations by double-clicking on a station in the list or using the "MSC Station Properties" shortcut menu command.
3. Assign a GSM network provider to the MSC stations (with MD720) by selecting a station in the list and then the shortcut menu command "Assign Provider...". You can select one or more stations at the same time.

   You can delete the provider assignment again with the shortcut menu command "Delete Provider Assignment".

   Here, you cannot assign a provider to stations with a DSL connection. You make the assignment in the configuration of the relevant DSL modem.
4. Close the properties dialog with "OK" when you have completed the MSC configuration.

Configuring the MSC master

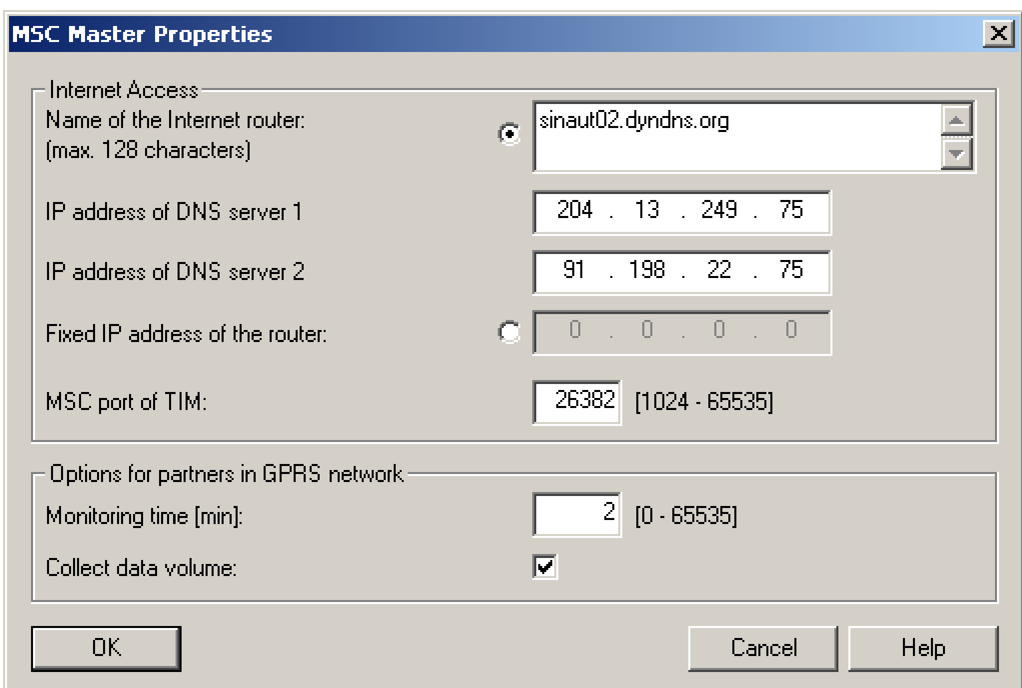


Figure 4-23 "MSC Master Properties" dialog

In the "Internet Access" box, you configure the address of the Internet router via which the MSC master is connected to the Internet and the port of the MSC master used for MSC communication. You can specify the Internet router either with a DNS name or a fixed IP address.

- Name of the Internet router:

  Enable this option if you enter the DNS name of the router.

  - Enter the name of the router in the input box.

    Max. 128 characters. The following characters are permitted: a...z, A...Z, 0...9, comma (,), period (.), hyphen (-), underscore (_) and the blank

  - Enter the IP address of at least one DNS server.

- Fixed IP address of the router:

  Enable this option if you configure the router with a fixed IP address.

  Enter the IP address of the TIM interface that is connected to the Internet or the interface of the router if you have configured a router (option "Use router" in the "Parameters" tab of the properties dialog of the Ethernet network node of the TIM).

  As default, the IP address is displayed from the interface parameter assignment of the TIM.

- MSC port of TIM

  If necessary, assign the number of the MSC port (no. 1 024 ... 65 535 can be selected). As default, port 26 382 is used.

In the "Options for partners in GPRS network" box, you can make the settings for GPRS stations (with MD720).

- Monitoring time [min]

  The monitoring time in minutes is used to monitor the MSC tunnel in the GPRS network. When using a GPRS station, if there is no data traffic between the MSC master and GPRS station within the monitoring time, the MSC master sends a monitoring message to the GPRS station. If there is no sign of life from the MSC master during the monitoring time, the master station and all subscribers that can be reached via the MSC master are indicated as disrupted in the GPRS station. En entry is made in the diagnostics buffer of the TIM in the station and the MD720 is reinitialized.

  The entry "0" means: No monitoring.

- Collect data volume

  If this option is enabled (checkmark), the master TIM makes the transferred data of all GPRS stations available.

  The data is available in the SINAUT diagnostics and service tool, see TIM Diagnostics (Page 424).

Configuring an MSC station/node station

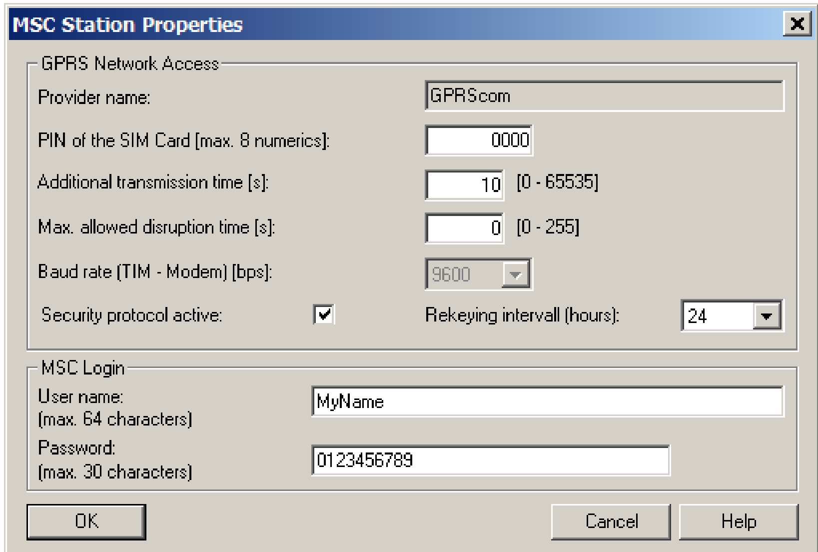


Figure 4-24 "MSC Station Properties" dialog

In this dialog, you configure the access data for the connected network and the type of protocol used.

"GPRS Network Access" box

In this box, you configure the Internet access data for GPRS stations (with MD720). For stations with a DSL connection, you configure the relevant data on the DSL router.

- Provider name

  If you have not yet assigned a GSM network provider, the first provider from the list of GPRS providers is entered as default. You can assign the provider in the MSC station list, see section "MSC Station List" tab (Page 160).

- PIN of the SIM card

  Here, you enter the PIN of the SIM card for the MD720.

- Additional transmission time [s]

  This is an offset added to the transmit retry time. The transmit retry time is calculated automatically on the TIM. The character delay time is also calculated from the "additional transmission time" parameter (character delay time = additional transmission time divided by 5).

  For GPRS networks, a value of 10 seconds as in the default setting is normally adequate.

- Max. allowed disruption time [s]

  Here, you can enter the tolerance time for connection disruption detected by the TIM. 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.

  If networks are regularly disrupted, it may be useful to increase the default value of 10 seconds. Increasing the allowed disruption time delays signaling of station failures and so reduces the number of organizational messages when stations return.

- Baud rate (TIM - modem) [bps]

  This is the speed at which the TIM and modem communicate. The default baud rate is 9 600 bps and cannot be changed.

- Enabling the security protocol

  – If the option is activated, the MSCsec protocol is used for this connection.

  – If the option is disabled, the MSC protocol is used for this connection.

  For information on the functionality of the two protocols, refer to the section GPRS / Internet communication: Overview of configuration (Page 36).

- Rekeying interval [hours]

  If the MSCsec protocol is enabled, set the key exchange interval between the station and master station here.

"MSC Login" box

Here, you configure the two mandatory boxes "User name" and "Password" for all MSC stations/node stations with GPRS or DSL connection. The data is required for establishing the connection and logging on with the MSC master.

- User name

  Enter the user name here with a maximum of 64 characters.

  The following characters are permitted: a...z, A...Z, 0...9, comma (,), period (.), hyphen (-)

- Password

  Enter the password here with a maximum of 30 characters.

  The following characters are permitted: a...z, A...Z, 0...9, comma (,), period (.), hyphen (-)

The "Provider Assignment" dialog

You open the "Provider Assignment" dialog from the properties dialog of the MSC station in the SINAUT configuration tool. You select an MSC station (with MD720) in the "MSC Station List" tab and then select the "Assign provider..." shortcut menu command (right mouse button). The "List of GPRS Providers" dialog opens.

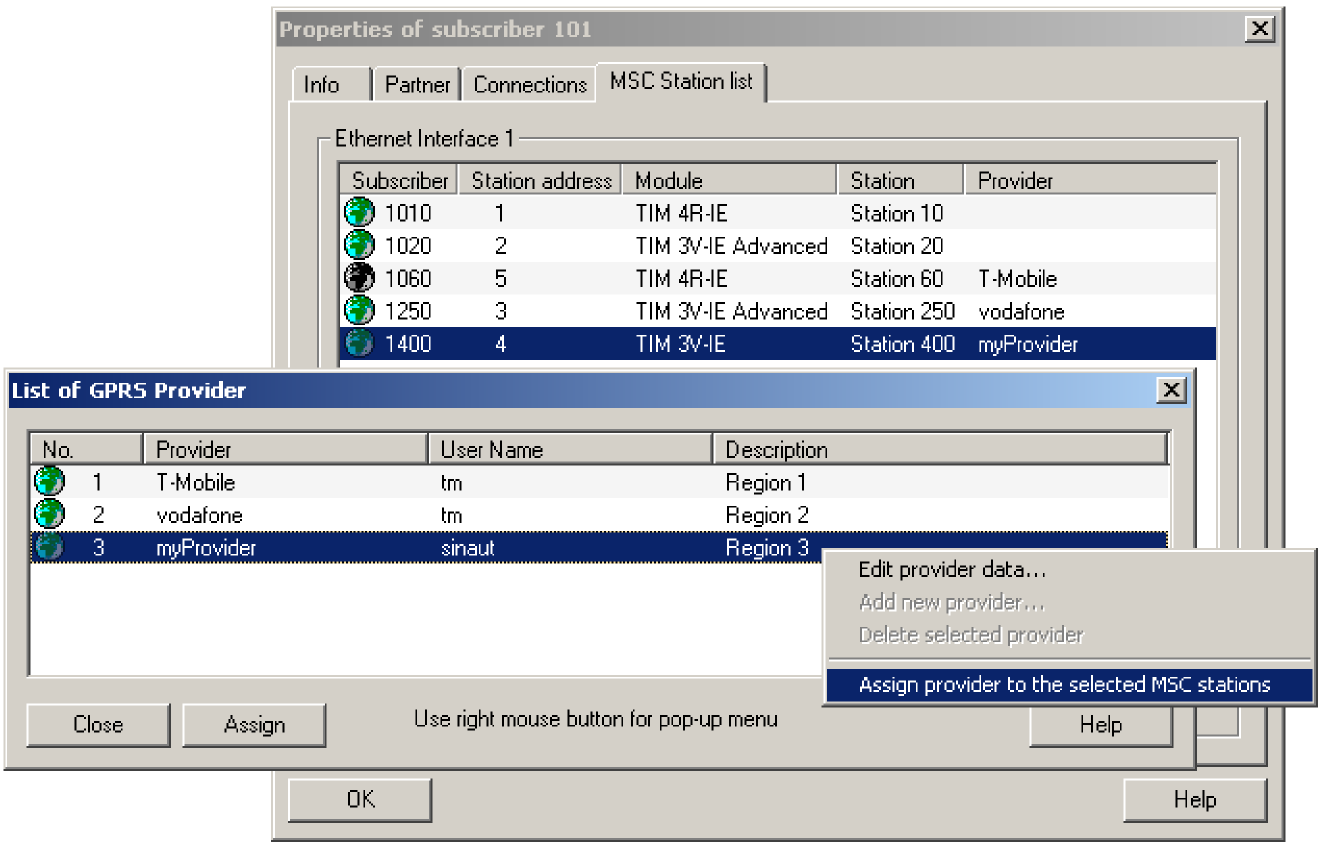


Figure 4-25 "List of GPRS Providers" dialog opened from the "MSC Station List" tab

The dialog box lists the GSM network providers configured for MSC communication showing the user name, a description and the globe icon.

The globe icon in this dialog has a the following meaning depending on its color:

- Green ( )

  All the required properties of the provider are configured.

- Black and white ( )

  The configured data of the provider is not complete.

You cannot assign incompletely configured providers.

Assigning providers and saving the assignment

Select a provider from the list and then select the "Assign provider to the selected MSC stations" shortcut command (right mouse button).

The assignment of the provider is saved when you exit the properties dialog in the SINAUT configuration tool ("MSC Station List" tab) with "OK".

Configuring the provider data

You cannot edit the provider data at this point in the "List of GPRS Providers" dialog. To edit the data, close both open dialogs and reopen the "List of GPRS Providers" dialog again by clicking on the globe icon ( ) in the toolbar in the window of the SINAUT configuration tool.

4.3.4.10 "Library Info" tab

Library Info tab

This tab shows the name, path, version, date of creation and source information for the German and English version of the *SINAUT TD7 library* with which system data for the current CPU will be generated.

4.3.5 Options for subscriber administration

Printing

To document the configured SINAUT subscribers, you can print out the list of subscribers in two formats. You start a printout with the "Project" > "Print" menu.

With the "Project" > "Print preview" menu you can display a print preview before you print.

Setting the print options:
Make the settings for printing in the "Subscriber administration" dialog. You display this with the "Extras" > "Options" menu.

Generate / compilation options

For information on the various options, refer to the section Saving subscriber data (Page 227).

SMS configuration

If the option is enabled, the configured message texts are checked for special characters.

You will find information about the permitted characters and the blocking of special characters in the section "SMS Configuration" tab (Page 153).

4.4 The TD7onTIM software

4.4.1 Components of TD7onTIM

Use and variants of the SINAUT TD7 software

The TD7 software is available in the two variants "TD7onTIM" and "TD7onCPU". You will find details of the differences and uses of the two variants in the section Use and variants of the SINAUT TD7 software (Page 29).

The configuration of TD7onTIM is described below.

Components of TD7onTIM

SINAUT communication makes use of SINAUT objects: Which data is to be sent or received can be configured over standardized data objects. These are collected in the TD7onTIM standard library. This library also contains the system objects with which system information is displayed and with which system functions are activated and set.

The parameter assignment for TD7onTIM is made in the subscriber administration of the SINAUT ST7 configuration tool in the following steps:

- Setting basic parameters for TD7onTIM
- Specifying the parameters specific to the destination subscribers
- Inserting system objects and assigning parameters to them
- Inserting data objects including their send and receive channels and assigning parameters to them

The basic settings for TIM subscribers with TD7onTIM

For each TIM with TD7onTIM, several settings must be made that are always required when working with TD7onTIM, for example specifying the read/write cycle. The settings are made in the basic settings for TIM subscribers with TD7onTIM.

The parameters specific to the destination subscribers

Each TIM with TD7onTIM can exchange data with one or more partners, known as destination subscribers. Which subscribers in the project are suitable as destination subscribers depends on the connection configuration. Here, you specify which TIM with TD7onTIM will have a connection with which SINAUT subscriber in the network.

Several settings are required for each of the possible destination subscribers that apply to data traffic between the TD7onTIM of a project and this subscriber, for example whether the subscriber expects data messages with a time stamp. This information is specified in the parameters specific to destination subscribers.

The system objects

The system objects provide system-relevant information for the CPU user program. Configuration of the system objects is optional.

- WatchDog

  The WatchDog indicates to the CPU program whether the communication between CPU and local TIM is still working; in other words, whether TD7onTIM is still reading from and writing to the memory areas of the CPU.

- PartnerStatus

  The PartnerStatus indicates to the CPU program whether communication with its partners (other ST7 CPUs or ST7cc/sc control centers) is OK or disrupted.

- OpInputMonitor

  OpInputMonitor indicates the status of operator input to the CPU program (with command, setpoint, and parameter input).

- SMSC (Short Message Service Centre):
  To send SMS messages via the GSM network using an MD720 or GPRS router, the connection data for the SMCS is configured in this system object.

The data objects

The sending and receiving of process data or SMS texts is configured with the aid of standardized data objects. According to the two transmission directions, these are divided into:

- Data objects for acquiring and sending data

  Their names have the ending "_S" for send.

- The data objects for receiving and outputting data

  Their names have the ending "_R" for receive.

In terms of the names, the data objects of TD7onTIM are identical with those of the data objects of TD7onCPU. In terms of functionality, they are compatible with each other; in other words, communication between data objects of TD7onTIM and the corresponding data objects of TD7onCPU is guaranteed.

The data objects are available in a standard library "TD7onTIM" and are inserted from the library into the TD7onTIM configuration. An example of a data object is "Ana04W_S" that organizes the transmission of 4 analog values.

The send and receive channels

The send and receive channels of the data objects are responsible for the processing of an individual process value, for example for processing and sending an analog value or receiving and outputting a message byte.

Each data object contains one or more send or receive channels. The number and type of send and receive channels per data object cannot be modified. The data object "Ana04W_S", for example, has 4 send channels of the type "send analog value".

If objects have multiple channels of the same type, not all channels need to be activated if they are not required.

4.4.2 Parameter assignment dialogs for TD7onTIM

Calling and appearance of the parameter assignment dialogs

The displayed information and parameter assignment boxes of TD7onTIM are divided into three areas in subscriber administration:

- Directory tree in the left-hand part of subscriber administration
- Subscriber list top right
- Parameter area below the subscriber list

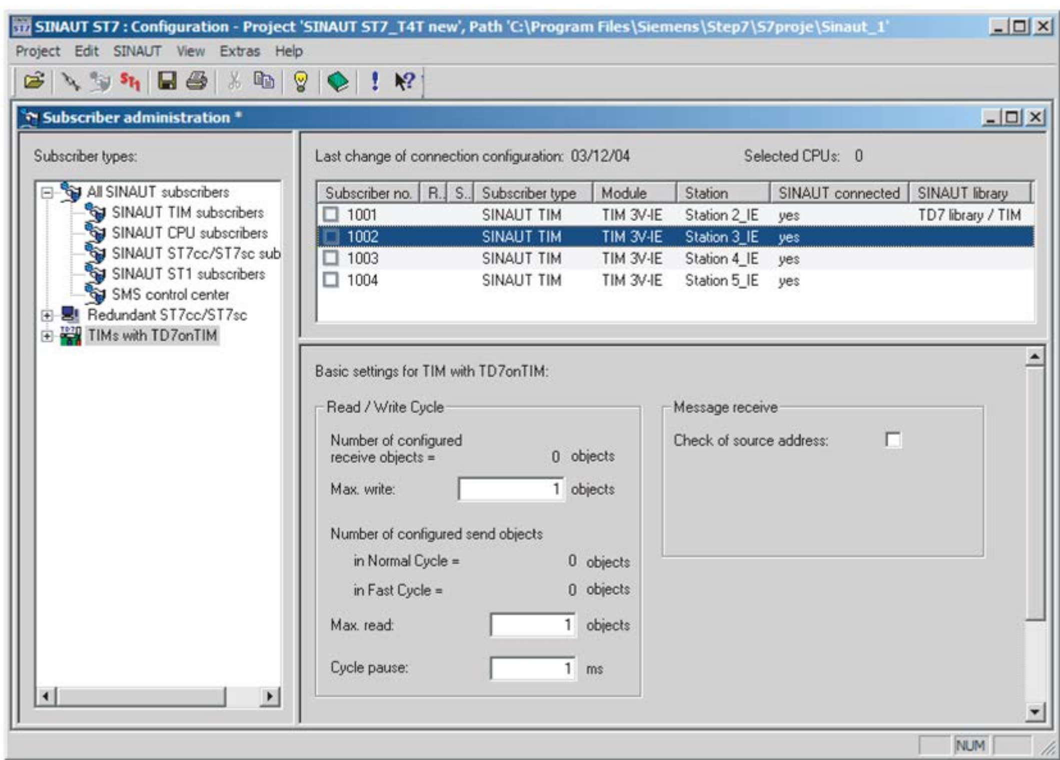


Figure 4-26 Subscriber administration of the SINAUT Configuration Tool

The "TIMs with TD7onTIM" directory is selected in the directory tree.

The subscriber list shows the subscribers with TD7onTIM.

The parameter area for these subscribers shows the "Basic settings for TIM with TD7onTIM".

You can change the size of the three windows in subscriber administration. The vertical and horizontal divisions between the windows can be moved with the mouse.

The directory tree

The TD7-compliant TIM modules of a project are shown in the directory tree in the *TIMs with TD7onTIM* directory. If you expand the directory with the (+) symbol or double-click on the directory, the following contents are displayed:

- The *All Destination Subscribers* directory
- The directories of all TIMs with TD7onTIM

If you expand a single TIM directory, the *data objects* already configured on this TIM are displayed.

The list box

At the top right of the subscriber administration there is a list box that lists certain subscriber types, objects, or send/receive channels depending on what is selected in the directory tree. By successively expanding the *TIMs with TD7onTIM* directory, the list box displays the following content:

- The TD7-compliant TIMs of the project
- The destination subscribers with which the TD7-compliant TIMs can communicate
- The system and data objects of a TD7-compliant TIM
- The channels of a data object

The SINAUT objects are displayed with the following symbols:

- Blue symbols: System objects
- Yellow symbols: Data objects

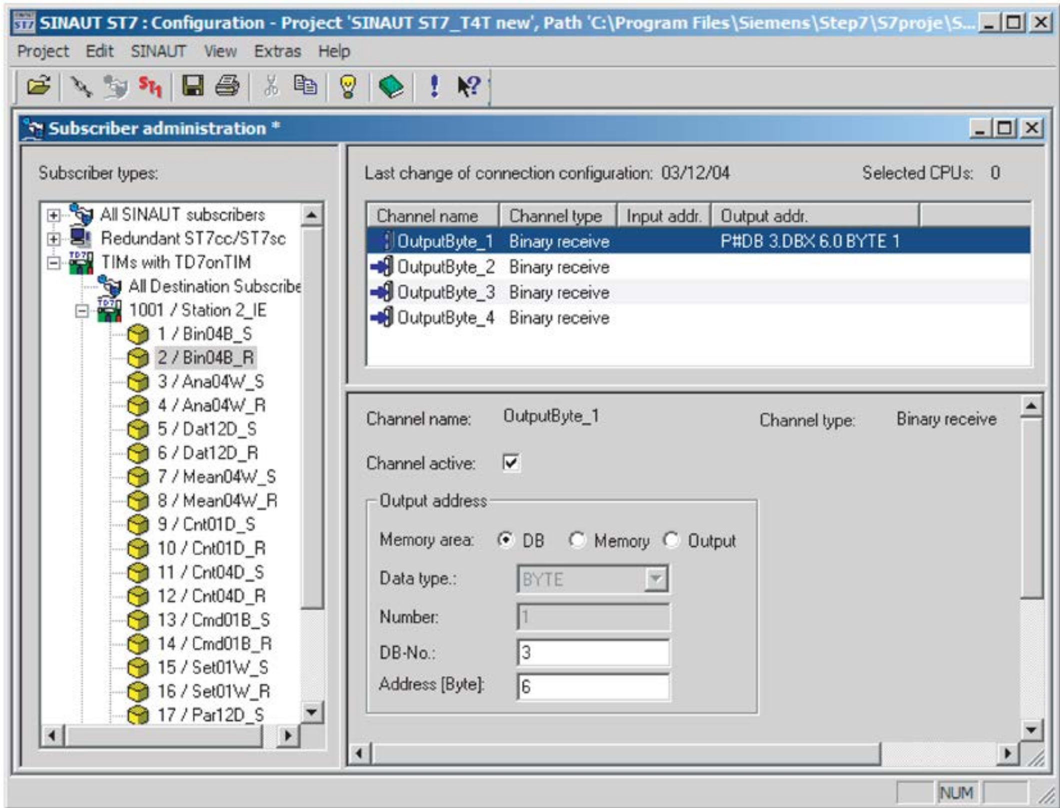


Figure 4-27 Selected data object *Bin04B_R* with selected channel in the list box and the parameter assignment window of a receive channel

The parameter assignment windows

In the parameter assignment windows below the list box, you set the parameters for the *TD7onTIM*-relevant subscribers, objects and channels. Depending on what is selected in the directory tree or in the list box, the following dialogs are displayed:

- Parameter assignment dialog of the basic parameters for a TD7-compliant TIM
- Parameter assignment dialog of a destination subscriber
- Parameter assignment dialog of a system or data object
- Parameter assignment dialog of a send or receive channel
  - Send channels are displayed with an outgoing arrow.
  - Receive channels are displayed with an incoming arrow.

To open a parameter assignment dialog, select the relevant subscriber, object or channel in the *list box*.

If a subscriber or object is selected in the *directory tree*, the parameter assignment dialog of the first object of the subscriber or the first channel of the data object is displayed.

Parameter entries

To simplify data entry, the parameter assignment dialogs are not opened or closed using separate buttons or menus but are displayed automatically when a subscriber, object or channel is selected with the mouse in the list box above.

The entries in the parameter assignment dialogs are not applied using a separate button, but immediately:

- When you activate or deactivate an option
- After entering data when you exit the input box with the mouse or tab key

The entries are applied permanently using the *Save* function.

4.4.3 Basic settings for TIM subscribers with TD7onTIM

To make the basic settings for the TD7 software of the individual TIM subscribers, you first select the *TIMs with TD7onTIM* directory in the directory tree. The list box then displays all the TD7-compliant TIM modules of the project.

Below the list box, the parameter assignment dialog appears for the TIM selected in the list window either automatically or with the mouse.

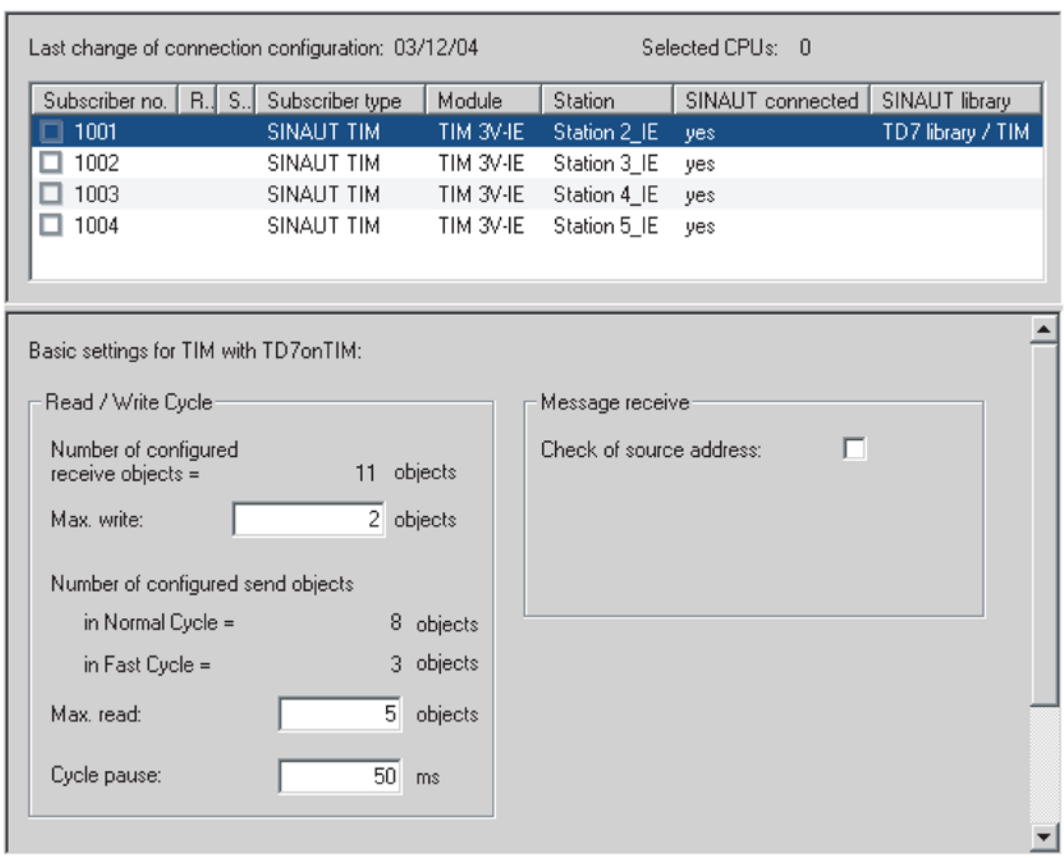


Figure 4-28 Parameter assignment dialog for basic settings for TIM subscribers with TD7onTIM

The parameters of the basic settings for TIM subscribers with TD7onTIM relate to

- The configuration of the read/write cycle of the TIM and
- Checking the source address when a message is received

The write/read cycle

Data to be sent by TD7onTIM, is read by the TIM over the backplane bus of the CPU and received data is written to the CPU.

The TIM also writes system information to the CPU (see system objects *Watchdog*, *PartnerStatus* and *OpInputMonitor*) and certain read data is reset; in other words, 0 is written. In the latter situation, this involves send trigger and command information that was read from the memory bit area or data blocks. TD7onTIM ensures that these are reset to 0 automatically after they have been acquired. All of these procedures take place within a defined and selectable write/read cycle.

The writing and reading of data takes place in consecutive write/read cycles. A basic cycle of the write/read cycle of TD7onTIM is made up as follows:

1. Write all pending system information (see system objects) to the CPU and reset all currently acquired send triggers and commanded entries. If no such data is currently pending, there is no write procedure in the basic cycle.
2. Read *all* data of the send objects that were assigned to the *fast cycle*. If no objects were configured for the *fast cycle*, this read procedure is omitted in the basic cycle.
3. Read data from some of the send objects that were assigned to the *normal cycle*. How many objects per basic cycle will be read can be set by the user. Refer to the *Max. read* parameter below.
4. Write data of some of the currently pending receive objects.
   How many objects this can be as a maximum per basic cycle can be selected by the user. Refer to the Max. write parameter below.
   If less received data is currently pending than is permitted as maximum per basic cycle, only this subset is written in the basic cycle.
   If there is currently no received data from the remote partners, this write procedure is omitted in the basic cycle.
5. Cycle pause (optional) to relieve the TIM and backplane bus communication.

With the *Max. read* and *Max. write* parameters and by specifying how many objects are assigned to the fast cycle, the user can set the duration of a basic cycle. Essentially, this specifies how fast the fast cycle really is: It is identical to the duration of the basic cycle. With the default 1 for the *Max. read* and *Max. write* parameters, the basic cycle has the shortest possible duration.

It must also be taken into account that the make-up of the basic cycle decides how long TD7onTIM requires to read all the data of the objects assigned to the normal cycle once. If, for example, 12 objects are assigned to the *normal cycle* and if *Max. read* is set to 2 objects per basic cycle, it takes 6 basic cycles until all the data of the 12 objects have been read once completely from the memory areas of the CPU.

Parameters in the *read/write cycle* box

Name:	**Max. write**
Range of values:	1 ... 32000
Default:	1
Explanation:	This is the maximum number of (different) data objects whose data is written to the CPU per basic cycle. If there are several messages of the same receive object in the buffer, only the data of one message of this object is written per basic cycle. As information, the number of receive objects configured for the TIM in total by the user is displayed above the input box beside *Number of configured receive objects*.

Name:	**Max. read**
Range of values:	0 ... 32000
Default:	1
Explanation:	This is the maximum number of data objects whose data is read from the CPU per basic cycle. As information, above the input box beside *Number of configured send objects*, you can see how many send objects the user configured for the TIM - in the *normal cycle* and - in the *fast cycle* .

Name:	**Cycle pause**
Range of values:	0 ... 32000 [ms]
Default:	1
Explanation:	This parameter specifies the duration of an optional pause between 2 basic cycles. A pause may be necessary if communication of other modules on the backplane bus is disrupted too much by the write and read jobs between the TIM and CPU. This also applies to subscribers on the MPI bus (further CPUs or PG) if the backplane bus is implemented as a party line. By setting a suitable time for the pause, the other bus subscribers have time for their communication. Specifying a cycle pause may also be necessary to relieve the TIM itself; in other words, when it becomes clear that it has too little time for other tasks due to the fast write/read cycle.

Parameters in the *Message receive* area

Name:	**Check of source address**
Range of values:	Function active, function deactivated
Default:	Function active
Explanation:	With this parameter, you specify whether or not the source address of the sending subscriber is checked prior to accepting data from a received message. If the function is activated, all messages that do not originate from the configured partner are discarded. **Note**: If a data object receives messages from several partners, the check of the source address must be deactivated.

Copying basic settings to other TIMs

Once the basic settings of a subscriber with TD7onTIM have been made, the settings can also be transferred to other TIMs that require the same parameters. Follow the steps outlined below:

1. Select a TIM for which you have already set the parameters in the list box.
2. Select *Copy* in the shortcut menu (right mouse button).
3. Select a second TIM.
4. Transfer the parameters to this second TIM using *Paste* in the shortcut menu.

4.4.4 Subscriber-specific parameters of TD7onTIM

Each TIM with TD7onTIM can exchange data with one or more partners, known as destination subscribers. Which subscribers in the project are suitable as destination subscribers depends on the connection configuration. Here, you specify which TIM with TD7onTIM will have a connection with which SINAUT subscriber in the network.

Several settings are required for each of the possible destination subscribers that apply to data traffic between the TD7onTIM of a project and this subscriber, for example whether the subscriber expects data messages with a time stamp.

To set the parameters specific to destination subscribers, open the *TD7 on TIM* directory and select the *All Destination Subscribers* directory. The list displays all potential destination subscribers of the TD7-compliant TIM modules. These are:

- SIMATIC S7 CPU modules
- SINAUT ST7cc/sc control centers

Below the list box, the parameter assignment dialog of the destination subscriber (selected automatically or with the mouse) opens.

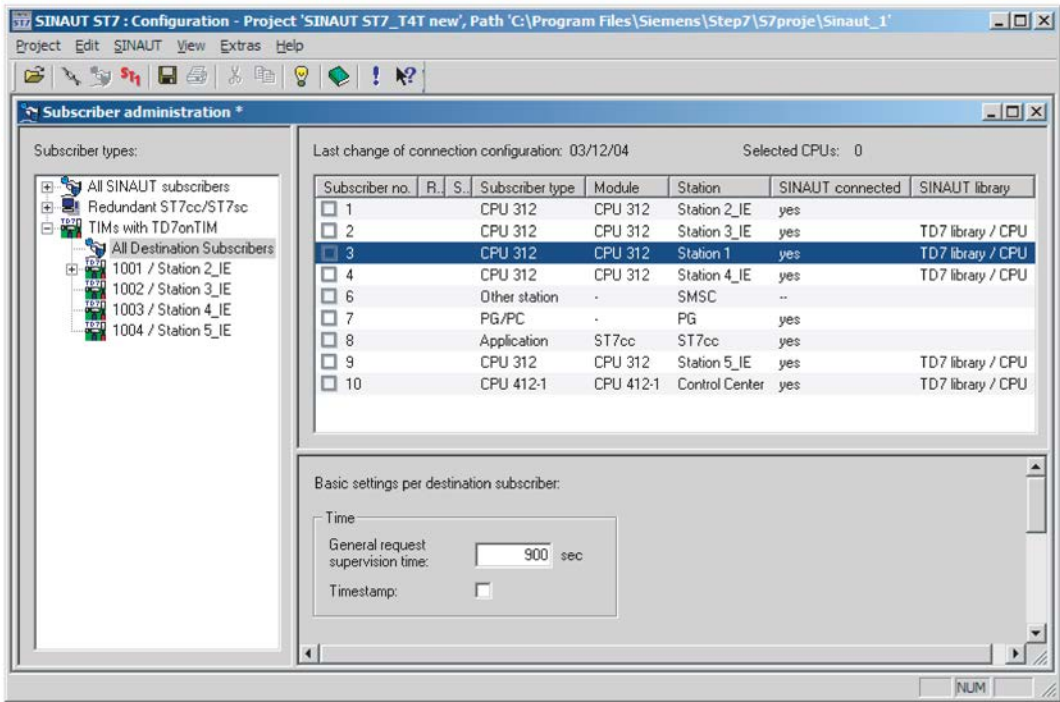


Figure 4-29 Directory tree, list box, and parameter assignment dialog of the destination subscriber-specific parameters of TD7onTIM for a destination subscriber

The parameters to be entered here are valid in terms of communication with the configured communications partners (destination subscribers) for all TD7-compliant TIM modules of the project.

The parameter assignment dialog for the destination subscriber-specific parameters is opened and the parameters are set for every configured destination subscriber.

The following destination subscriber-specific parameters are available:

Name:	**General request supervision time**
Range of values:	10 ... 32000 s

Default:	900 s
Explanation:	The *General request supervision time* is the maximum time required by a destination subscriber to respond fully to a general request (GR). If the GR response has not arrived completely at the requesting TIM module when the supervision time has expired, a message is entered in the diagnostics buffer of this TIM module and an ID is entered in the objects of the TD7onTIM affected. A TIM can only send requests to SINAUT stations with TD7 software; in other words, a CPU with TD7onCPU or a CPU with a local TIM with TD7onTIM. The time should be set generously. In particular with dial-up connections, it should be remembered that the time for connection establishment is included in the supervision time. In addition to this, with dial-up stations the messages stored in the send buffer further delay processing of the GR because the queried messages are entered in the send buffer behind all other messages.

Name:	**Time stamp**
Range of values:	Function active, function deactivated
Default:	Function deactivated
Explanation:	This parameter specifies whether or not messages with a time stamp are sent to this destination subscriber. If this is the case, the *Timestamp* option must be activated. TD7onTIM can send either all messages with or all messages without a time stamp to a destination subscriber. Mixing messages to a destination subscriber with and without time stamps is not possible.

Copying parameters to other destination subscribers

Once the destination subscriber-specific parameters have been set for a destination subscriber, they can be transferred to other destination subscribers that require the same parameters. Follow the steps outlined below:

1. Select a destination subscriber for which you have already set the parameters in the list box.
2. Select *Copy* in the shortcut menu (right mouse button).
3. Select a second destination subscriber in the list box.
4. Transfer the parameters to this second destination subscriber using *Paste* in the shortcut menu.

4.4.5 Overview of the SINAUT objects

Overview of the SINAUT objects

The communications properties of the TD7onTIM software are configured using objects. The objects are divided into two groups:

- **System objects**

The system objects provide system-relevant information for the CPU user program. Configuration of the system objects is optional.

- **Data objects**

  The special communications parameters such as subscriber, data etc. are configured in the data objects.

The following objects are available for TD7onTIM:

Table 4- 4 System objects of TD7onTIM

Name	Explanation
System objects	
WatchDog	Monitoring of the CPU-TIM connection
PartnerStatus	Displays connection status for up to 8 SINAUT subscribers
PathStatus	Show path status to a partner
OpInputMonitor	Signals detection of hardware entries
SmServiceCenter *	Access data for access to the Short Message Service Center (SMSC)

* Sending SMS messages via TD7onTIM is supported only in conjunction with an S7-300 CPU with party line. On the party line, see Glossary.

Table 4- 5 Data objects of TD7onTIM

Object type	Name	Explanation
Message objects		
Message object for send direction	**Bin04B_S**	Send 4 bytes of messages/binary information
Message object for re-ceive direction	**Bin04B_R**	Receive 4 bytes of messages/binary information
Analog value and mean value objects		
Analog value object for send direction	**Ana04W_S**	Send 4 analog values (16-bit value in the INT format)
Analog value object for receive direction	**Ana04W_R**	Receive 4 analog values (16-bit value in the INT for-mat)
Mean value object for send direction	**Mean04W_S**	Send 4 mean values (16-bit value in the INT format)
Mean value object for receive direction	**Mean04W_R**	Receive 4 mean values (16-bit value in the INT for-mat)
Counted value objects		
Counted value object for send direction	**Cnt01D_S**	Send 1 counted value (32-bit SINAUT format).

Object type	Name	Explanation
Counted value object for receive direction	Cnt01D_R	Receive 1 counted value (32-bit SINAUT format)
Counted value object for send direction	Cnt04D_S	Send 4 counted values (32-bit SINAUT format)
Counted value object for receive direction	Cnt04D_R	Receive 4 counted values (32-bit SINAUT format)
Command objects		
Command object for send direction	Cmd01B_S	Send 1 byte commands (1-out-of-8 SINAUT format)
Command object for receive direction	Cmd01B_R	Receive 1 byte commands (1-out-of-8 SINAUT format)
Setpoint and parameter objects		
Setpoint object for send direction	Set01W_S	Send 1 setpoint (16 bits), object with 3 channels : - operating mode status *local* - returned value - setpoint entry
Setpoint object for receive direction	Set01W_R	Receive 1 setpoint (16 bits), object with 3 channels : - operating mode *local* - local setpoint entry - setpoint output
Parameter object for send direction	Par12D_S	Send max. 12 double words with parameters or setpoints, object with 3 channels: - operating mode status *local* - returned parameters - parameter entry
Parameter object for receive direction	Par12D_R	Receive max. 12 double words with parameters or setpoints, object with 3 channels: - operating mode *local* - local parameter entry - parameter output
Parameter object for receive direction	Par12x1D_R	max. 12 double words for receiving individual parameters or setpoints of ST7cc/sc, object with 14 channels: - 1 x operating mode *local* - 1 x local parameter entry - output 12 parameter individually
Objects for any data		
Data object for send direction	Dat12D_S	Send max. 12 double words (at least 1 double word) with any information
Data object for receive direction	Dat12D_R	Receive max. 12 double words (at least 1 double word) with any information
Data object for receive direction	Dat12x1D_R	Receive 12 individual double words from ST7cc/ST7sc and output them individually
Object for sending SMS messages		
Data object for SMS messages to be sent	Sms01_S *	Sending SMS messages via the connected modem MD720 or GSM router

* Sending SMS messages via TD7onTIM is supported only in conjunction with an S7-300 CPU with party line. On the party line, see Glossary.

The endings "_S" and "_R" in the object names mean Send or Receive.

4.4.6 Inserting and deleting SINAUT objects

Open in the standard library of TD7onTIM

After setting the basic parameters of TD7onTIM and the destination subscriber-specific parameters, the SINAUT objects of TD7onTIM are configured.

If no SINAUT objects have yet been configured for the TIM (as is the case in a new project), the directory of the TIM cannot be expanded any further and the list box is empty.

Inserting objects in the project

Every TIM module with TD7onTIM has the required SINAUT objects added from the standard library.

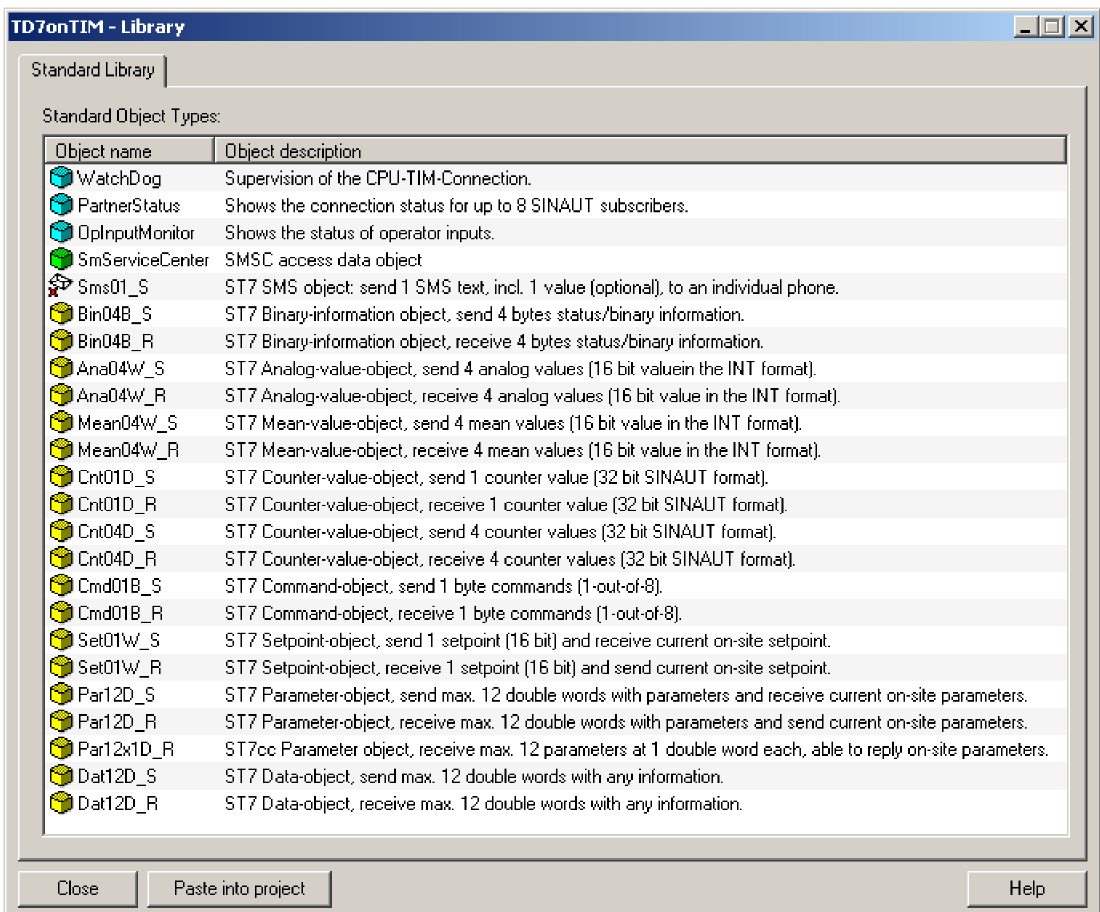


Figure 4-30 Window of the standard library "TD7onTIM" of the SINAUT objects for TD7onTIM

To insert new SINAUT objects in the TD7onTIM of a subscriber, follow the steps outlined below:

1. Go to the directory tree and select the TIM for which you want to configure the SINAUT objects.
2. Open the standard library in one of the following ways:
   - The *Standard library* button in the toolbar
   - The *SINAUT / Standard Library for the TIM* menu
   - The F7 function key

   The library is opened in a separate window. The objects are listed with the object name and a brief object description.
3. Select an object with the mouse in the window of the standard library.

   To insert several objects at the same time, follow the steps below:
   - Select an object and while holding down the Shift key move up or down with the arrow up or arrow down key. You select several adjacent objects.

     or
   - While holding down the Ctrl key, select several distributed objects one after the other.
4. In the standard library window, click on the *Paste* button or select *Paste* in the context menu (right mouse button).

   The selected objects are then added to TD7onTIM.
5. Close the standard library with the *Close* button when you no longer require it.

All the inserted objects are displayed in the list window.

In the directory tree, only the data objects are displayed below the selected TIM.

Note

100 objects can be configured per TD7onTIM.

Deleting objects

You can delete an object you do not require from the TIM directory by selecting it in the directory tree or in the list box and then selecting "Delete" in the context menu (right mouse button).

Copying objects to other TIMs

Once all the SINAUT objects for a TIM have been configured and have had then parameters assigned, you can copy the objects and the parameter assignments to another TIM in the project that requires the same or similar objects.

Follow the steps outlined below:

1. Select the objects in the list box while pressing the Shift key and then press the arrow up or down key or select the objects while pressing the Ctrl key.
2. Select *Copy* in the context menu (right mouse button).
3. Select another TIM in the *TIMs with TD7onTIM* directory.
4. Add the selected objects to this TIM with *Paste* in the context menu.

   The objects along with their entire parameter assignment are adopted by the TD7onTIM of the other TIM.

In the copied object and its channels, it may be necessary to adapt the subscriber-specific parameter assignment (for example the input and output addresses).

If the complete parameter assignment of TD7onTIM is required for other TIM modules, you can also transfer the entire SINAUT objects to a different TIM. Follow the steps outlined below:

1. Select a TIM in the *TIMs with TD7onTIM* directory.
2. Select *Copy* in the context menu (right mouse button).
3. Select another TIM in the *TIMs with TD7onTIM* directory.
4. Add all objects to the other TIM with *Paste* in the context menu.

4.4.7 System objects

4.4.7.1 Setting parameters for system objects

Procedure

After the SINAUT objects for the individual subscribers have been configured, you set the parameters for the system objects. Follow the steps outlined below:

1. Select a TD7onTIM subscriber in the directory tree.
2. Select the required system object in the list.

   The appropriate parameter assignment dialog is displayed below the object list.
3. You set the parameters in this dialog.

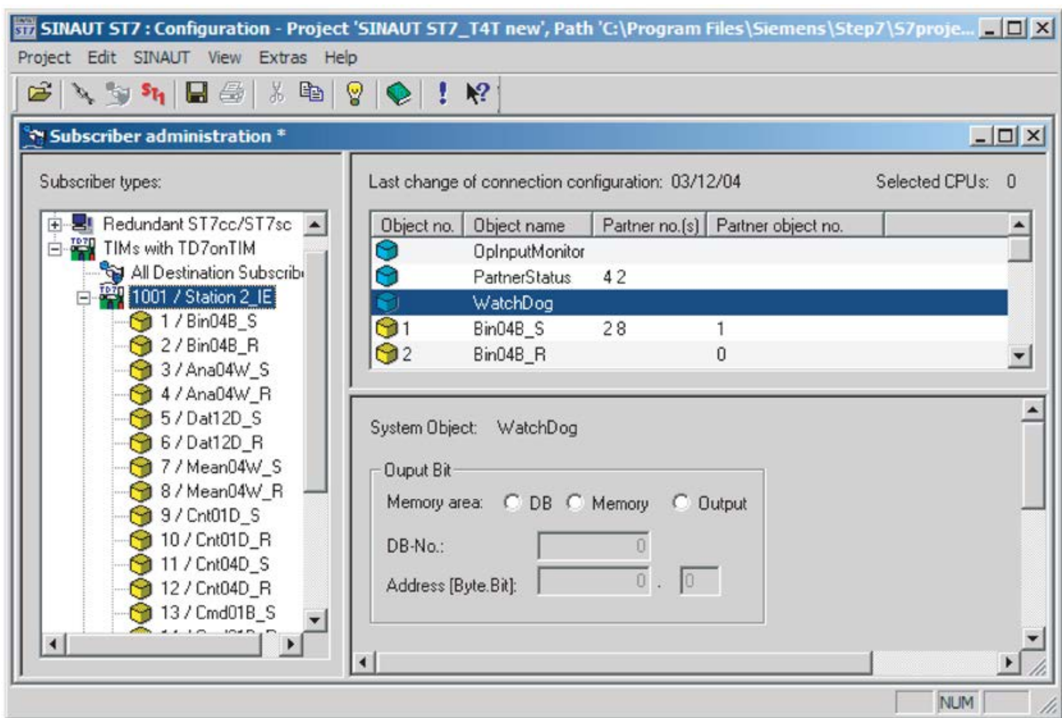


Figure 4-31 TIM with selected system object "WatchDog" and the corresponding parameter assignment dialog

4.4.7.2 WatchDog

The "WatchDog" system object

The *WatchDog* system object can be included as an option. The WatchDog indicates to the CPU program whether the communication between CPU and local TIM is still working; in other words, whether TD7onTIM is still reading from and writing to the memory areas of the CPU. As long as communication is functioning correctly, a selectable output bit changes state at five second intervals. The constant status change can be evaluated by the CPU user program.

Parameters in the *Output Bit* area

Name:	**Output bit**
Memory area:	The following options are available: - DB: Data block - Memory bit: Memory area - Output: Process output image (PIQ)
DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected
Address [Byte.Bit]:	Input fields for the byte and bit number in the selected memory area

4.4.7.3 PartnerStatus

The "PartnerStatus" system object

The optional *PartnerStatus* object can be used to monitor the availability of up to 8 communication partners. A partner can be an ST7 CPU or an ST7cc/sc control center to which a connection was configured. TIM modules cannot be monitored with the PartnerStatus object. The status is made available to the CPU user program in an output byte. One bit per communication partner is reserved in the output byte to indicate the status of the respective partner:

- Status 0: Problem on partner or corresponding bit not assigned to any partner
- Status 1: Partner OK

If TD7onTIM has a connection to more than 8 partners whose status needs to be monitored, the *PartnerStatus* system object is configured more than once.

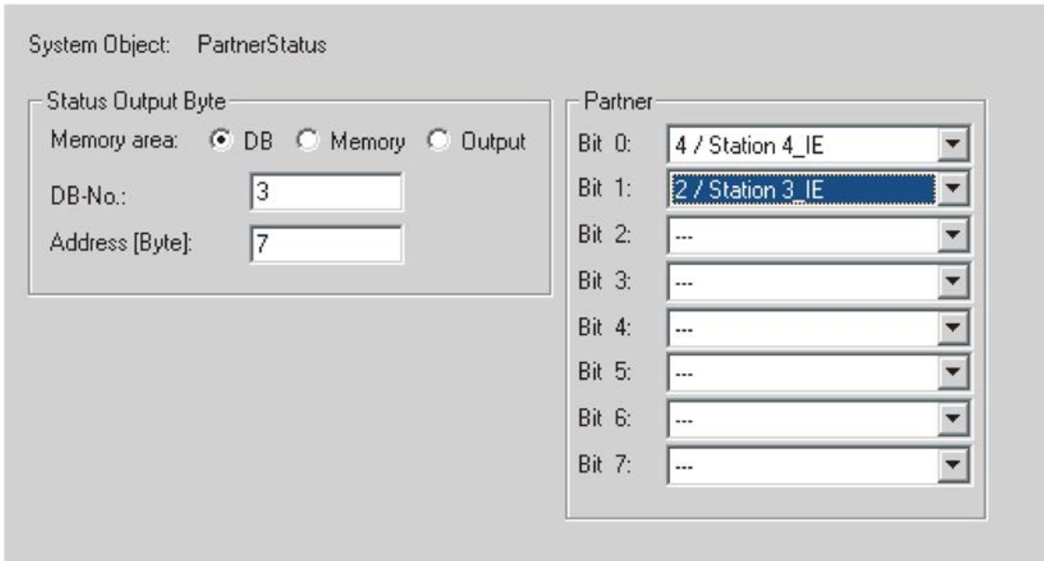


Figure 4-32 Parameter assignment dialog of the "PartnerStatus" system object

Name: **Status output byte**

Memory area: The following options are available:

- DB Data block
- Bit memory: Memory bit area
- Output: Process output image (PIQ)

DB No.: Specifies the DB number in the CPU if the data block memory area (*DB*) was selected

Address [Byte]: Input field for the byte number in the selected memory area

Name: **Partner**

Explanation: In the list boxes, the 8 status bits of the output byte are assigned to the communication partners to be monitored. The partners can be selected in the 8 list boxes. The list boxes display only the partners with which the TIM can actually communicate; in other words, to which a connection was configured.

Bit status: Status = 0: Problem on partner or bit not assigned
Status = 1: Partner OK

4.4.7.4 PathStatus

The "PathStatus" system object

The optional object "PathStatus" shows the status of the path to a partner from the perspective of the local TIM. The use of the system object is practical when more than two paths are configured to this partner.

A maximum of 2 paths (main and substitute path) to a partner can be configured. Both paths must begin or end on the local TIM.

The system object 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 Status output byte per path 2 bits are reserved for the configured interface and 2 bits for the status.

If the status of the path to more than one partner is to be shown, you need to create a system object per partner.

Parameters

Name:	**Status output byte**
Explanation:	Memory area for the output byte of the path status
Memory area:	The following options are available: • DB Data block • Bit memory: Memory bit area • Output: Process output image (PIQ)
Address [Byte]:	Input box for the byte number in the selected memory area

Name:	**Path status**
Explanation:	From the drop-down list, select the partner to which the paths are to be shown. The subscribers that can be selected result from the configuration. The following is displayed for every partner: "SINAUT subscriber no. / station name"
Bit assignment:	• Bit 0 + 1: Path status main path (1st path) • Bit 2 + 3 Path status substitute path (2nd path) • Bit 4 + 5: Configured interface for main path • Bit 6 + 7 Configured interface for substitute path

Explanation of the status bits in the path status output byte

Table 4- 6 Bit assignment of the output byte path status

Bit 6 + 7	Bit 4 + 5	Bit 2 + 3	Bit 0 + 1
Configured interface	Configured interface	Path status	Path status
No. for substitute path	No. for main path	Substitute path (2nd path)	Main path (1st path)

Status table: Configured interface

The TIM interfaces "Ethernet 1" (IE1), "Ethernet 2" (IE2), WAN1 and WAN2 are numbered through from 0 to 3 .

The TIM interfaces are numbered through from 0 to 3 .

- 0 = "Ethernet 1" (IE1)
- 1 = "Ethernet 2" (IE2)
- 2 = WAN1
- 3 = WAN2

Table 4- 7 Coding of bits 4 + 5 or bits 6 + 7

Status of bit 5 (7)	Status of bit 4 (6)	Meaning
0	0	Coding for interface IE1 (decimal): No. 0)
0	1	Coding for interface IE2 (decimal: no. 1)
1	0	Coding for interface WAN1 (decimal: no. 2)
1	1	Coding for interface WAN2 (decimal: no. 3)

Status table: Path status

- Main path = 1. Path (bits 0 + 1)
- Substitute path = 2nd path (bits 2 + 3)

Table 4- 8 Coding of bits 0 + 1 or bits 2 + 3

Status of bit 1 (3)	Status of 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

Example of coding options

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

Table 4- 9 Coding example for the output byte path status

Configured interface		Path status	
Bit 6 + 7 **No. for substitute path**	**Bit 4 + 5** **No. for main path**	**Bit 2 + 3** **Substitute path (2nd path)**	**Bit 0 + 1** **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

4.4.7.5 OpInputMonitor

The "OpInputMonitor" system object

The OpInputMonitor indicates the status of operator input to the CPU (with command, setpoint, and parameter input). The current status can be displayed to the operator for each user program in a suitable form, for example using the LEDs, on an operator panel etc.

Note

The "OpInputMonitor" system object can only be included once per TD7onTIM.

TD7onTIM should therefore have the "OpInputMonitor" system object added when one or more of the following objects is used with this TD7onTIM:

- Cmd01B_S (Command object for send direction)
- Set01W_S (Setpoint object for send direction)
- Par12D_S (Parameter object for send direction)

The OpInputMonitor is recommended particularly when commands are entered over digital inputs, for example using buttons connected to them. This also applies to the situation when setpoint and parameter entries are transmitted as the result of the send trigger "Triggersignal" and when this triggering is over a digital input, for example, once again using a button.

Using OpInputMonitor reduces the risk of incorrect input when the entries are made over digital inputs. For these inputs, a "Minimum input time" can be specified for OpInputMonitor, in other words, the button must be pressed for the minimum time. The accidental activation of a button does not then lead to unwanted command, setpoint or parameter transfer. When the minimum input time has elapsed and the button can be released, OpInputMonitor indicates this with its "status byte for operator input" in the "Input OK" bit (see below).

Apart from the minimum duration, a "maximum input time" can also be set for digital inputs. This allows a button that is sticking or defective digital inputs that supply a permanent 1 signal to be detected in good time. Such errors are once again indicated in the "status byte for operator input" of OpInputMonitor, in this case in the "Input error" bit.

The two times and the code bits mentioned above are relevant only for operator input over digital inputs.

For all types of operator input, in other words both for input over digital inputs as well as input over the memory or data blocks, OpInputMonitor also returns the error status "1 out-of-n error". This is set when TD7onTIM has detected one of the following input errors:

- More than 1 bit was set in the input byte of the command object "Cmd01B_S". To increase reliability of command input, only one bit may ever be set with this object. If two or more bits are set at the same time, the command input is rejected
- If increased reliability is required for the input of commands, setpoints and parameters, all objects with which this data is sent should be assigned to the fast cycle. All command, setpoint and parameter objects in the fast cycle are subjected to a 1-out-of-n check; in other words, at the end of the fast cycle there is a check to make sure that there is a command, setpoint or parameter entry for only one of the acquired objects. Only then is the corresponding entry processed and transferred. If there is more than one entry, the entries are rejected. A new command, setpoint or parameter is processed only when previously no entry was acquired in at least one fast cycle.

Note

If commands are entered over a memory or data byte, or a setpoint or parameter entry is enabled by a memory or data bit (over the send trigger "Triggersignal"), the set command bit or trigger signal is automatically reset to zero by TD7onTIM.

If, however, a 1-out-of-n error is detected, these bits are not automatically reset. They must then be reset by the user or the user program.

Parameter assignment in the "Status byte for operator input" box:

Name: **Status byte for operator input**

Memory area: The following options are available:

- DB Data block
- Bit memory: Memory bit area
- Output: Process output image (PIQ)

DB No.: Specifies the DB number in the CPU if the data block memory area (DB) was selected

Address [Byte]: Input field for the byte number in the selected memory area

Explanation: The status byte for operator input is the output byte of the OpInputMonitor system object.

In the status byte for operator input, the next 3 bits are assigned (explanation see above).

Bit assignment of the status byte								
Bit:	.7	.6	.5	.4	.3	.2	.1	.0
Status:	8	7	6	5	4	1-out-of-n error	Input error	Input OK
For value:	0	0	0	0	0	1	1	1
Unused bits are set to 0								

Parameter settings in the "Tolerance time for hardware input" box:

Name: **Max. Input Time**

Range of values: Enter value x 1 [s] (10 corresponds to 10 seconds)

Default: 0

Explanation: Monitoring time for commands entered over hardware inputs, or setpoints and parameters whose transmission is triggered over a hardware input. If the 1 signal is set at these hardware inputs for longer than defined in Max. Input Time, then the "input error" bit is set in the status byte for operator input. Further hardware entries are not processed as long as the "input error" bit is set.

The "Max. Input Time" is specified in seconds.
A time of at least 30 seconds is recommended (entry: 30)
0 (zero) can be entered if the parameter is not required.

Name: **Min. Input Time**

Range of values: Enter value x 0.1 [s] (10 corresponds to 1 second)

Default:	0
Explanation:	Delay time for commands entered over hardware inputs, or setpoints and parameters whose transmission is triggered over a hardware input. The message is entered in the send buffer of the TIM only if the currently entered command, setpoint for parameter remains unchanged for the specified delay time and no other command or setpoint input is detected during this time. The Min. Input Time is specified in tenths of seconds. A time of at least 1 second is recommended (entry: 10). 0 (zero) can be entered if the parameter is not required.

4.4.7.6 SmServiceCenter

The "SmServiceCenter" system object

Sending SMS messages via TD7onTIM is supported only in conjunction with an S7-300 CPU with party line. On the party line, see Glossary.

This object is only relevant for SMS messages sent with TD7onTIM via GPRS.

To send SMS messages via the GSM network using an MD720 or SCALANCE M874-2, the data for accessing the Short Message Service Center (SMSC) is configured in this system object.

A maximum of 3 objects of this type can be configured.

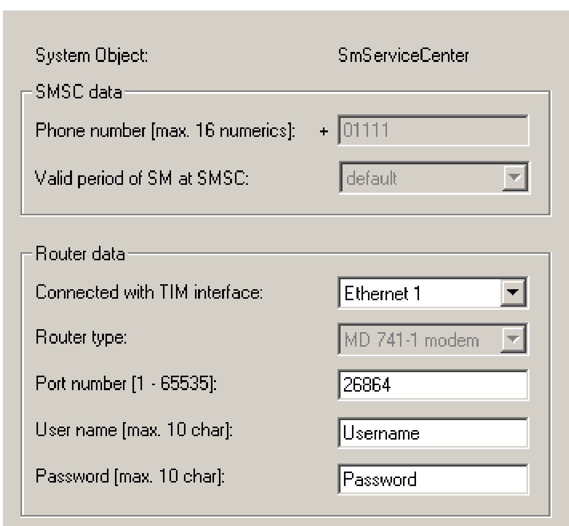


Figure 4-33 Parameter assignment dialog of the "SmServiceCenter" system object

Configuration in the "SMSC data" box:

Name:	**Telephone number**
Range of values:	Max. 16 digits, no plus character (+)

Default: 0

Explanation:
- With connected MD720:

  Entry of the call number of the SMSC (Short Message Service Center)

  The phone number is entered without the leading zeros of the country code (trunk prefix), in other words only the country ID number.
- With connected SCALANCE M874-2:

  When using the SMS function via a SCALANCE M874-2, the box is grayed out. The telephone number of the SMSC is stored on the SIM card and in this case, can only be changed via the Web interface of the SCALANCE M874-2.

Name: **Valid period of the SMS**

Range of values:
- With connected SCALANCE M874-2:
  - Default

    The value cannot be set.
- With connected MD720, the following can be set:
  - Default
  - 1 hour
  - 1 day
  - 1 week

Default: The value of the GSM network provider stored on the SIM card is adopted. The value depends on the particular provider.

Explanation: Time during which the GPRS router attempts to send the SMS message to the target subscriber.

Configuration in the “Router data" box:

Name: **Connection to TIM interface**

Range of values:
- With connected MD720:
  - WAN interface 1 (configured as a virtual Ethernet interface)
- With connected SCALANCE M874-2:
  - Ethernet interface 1
  - Ethernet interface 2

Explanation: Interface of the TIM to which the GPRS router is connected and via which SMS messages will be sent.

Name: **Router type**

Range of values:
- SCALANCE M874-2
- MD720

Explanation: Depending on the selected interface of the TIM, the router type is entered automatically.

Name:	**Port number**
Range of values:	1...65535
Default:	26864
Explanation:	Port number of the Ethernet interface (SCALANCE M874-2 only); must match the port number used on the SCALANCE M874-2.

Name:	**User name**
Range of values:	Max. 10 characters: Letters, digits, special characters; no spaces
Default:	User
Explanation:	User name for the SIM card of the SCALANCE M874-2 (refer to the contract documents of the GSM network provider)

Name:	**Password**
Range of values:	Max. 10 characters: Letters, digits, special characters; no spaces
Default:	Password
Explanation:	Password for the SIM card of the SCALANCE M874-2 (refer to the contract documents of the GSM network provider)

4.4.8 Data objects

4.4.8.1 Basic parameters of the data objects

Configuration of the objects and channels

The parameters for data objects are set in two phases:

1. Setting of the basic parameters of the data objects (for example partners to which the data of the object is sent or from which it is received)

   This section first describes setting the basic parameters of the data objects.

2. Setting of the channel-specific properties for the individual send and receive channels of the data objects

   The parameter assignment of the channels is described in the following sections and is divided up as follows:

   – Obligatory parameter of the channels

     These are channels that exist with all data objects and always require parameter assignment.

   – Object-specific parameters of the channels

     These are the specific channels of an object type and they are described later with the individual object types.

Configuration of the basic parameters of the data objects

Follow the steps outlined below:

1. Select a TIM with TD7onTIM in the directory tree.
2. Select the required data object in the list box.

   The corresponding parameter assignment dialog opens.
3. You set the parameters in this dialog.

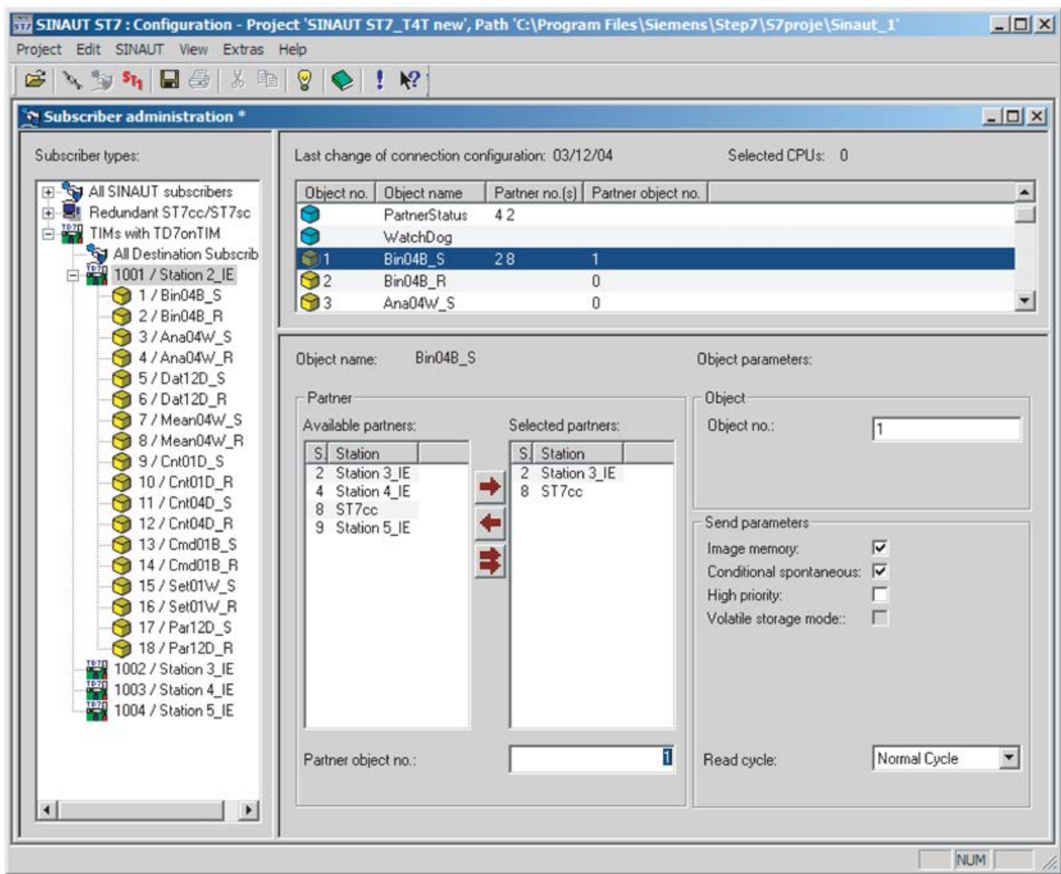


Figure 4-34 Subscriber administration with the parameter assignment dialog of the basic parameters of a data object

Parameters in the *Object* area

Name:	**Object no.**
Range of values:	1 ... 32000
Explanation:	The source object number of this TIM module is set in the input box. The configuration tool proposes a consecutive number that can be modified. An inconsistent duplicate assignment of numbers is prevented.

Parameters in the *Partner* area

The *Available partners* list box shows all the partners configured for the TIM along with their subscriber numbers and station names. The communication partners for the relevant data object are selected from this list and added to the *Selected partners* list.

Note

All partners from which data of the object is received or to which it is sent must be added to the *Selected partners* list. If no partner is entered in the *Selected partners* list, the object is not processed.

Name:	**Selected partners**
Explanation:	The *Selected partners* list displays the communication partners for the relevant data object. These are selected from the *Available partners* list box. Partners are entered in the *Selected partners* list by selecting one or more (using the Ctrl key) partners in the *Available partners* list and clicking on the button with the arrow pointing to the right. The selected partners are then entered in the *Selected partners* list. The button with the double arrow is is used to enter all available partners. Partners are removed from the *Selected partners* list using the button with the arrow pointing left. Up to 15 partners can be configured per data object.

Name:	**Partner object no.**
Range of values:	0 ... 32000

Default: 0

Explanation: In this input box, you assign the data object to the corresponding partner object of *all* selected communication partners when the partner object number is identical for all partners. Otherwise set 0 (zero).

Sms01_S:
For the "Sms01_S" object, the object number of the system object "SmServiceCenter" is entered here. Only 1 SMSC can be assigned to an Sms01_S.

ST7cc/sc:
Note: There is no object no. for objects of an ST7cc/sc control center!
For a send object of TD7onTIM that transfers data to ST7cc/sc, a partner object no. = 0 can be set since the specified partner object no. is not evaluated by ST7cc/sc.
For a receive object of TD7onTIM that receives data from ST7cc/sc, the partner object no. = 0 must be set.

Partner object no. = 0:

- After object no. = 0 for *send objects*:
  The partner object no. = 0 must be set if the object data will be sent to several partners and the receiving objects on these partners have different object numbers.
  If partner object no. = 0 is specified and if *TD7onCPU* is used on the partner, the ListGenerator must be installed on the partner. Only then can the corresponding receive object to be recognized on this partner.
- Partner object no. = 0 for *receive objects*:
  The partner object no. = 0 must be set if the object receives data from several partners and the sending objects of these partners have different object numbers.
  Note: If partner object no. = 0 is specified for at least one receive object, the option *Check of source address* must be deactivated in the basic settings of this TD7onTIM. Otherwise, messages intended for these receive objects will be rejected.

Note

With data objects of the type *Command snd*, *Setpoint send* and *Parameter send*, partner object no. 0 (zero) is not permitted.

The *Send parameters* area:

Name: **Image memory**

Range of values: Function active: Send using the image memory principle
Function deactivated: Send using the send buffer principle

Default:	In general: Function active With Cmd01B_S, Set01W_S and Par12D_S: Function deactivated and setting cannot be changed
Explanation:	The basic setting for saving messages in the send buffer of the TIM prior to transmission is the *image memory method* set as default in the *Image memory* parameter. As long as a message has not been sent, the process data in the message waiting to be sent is updated by current process data if this changes. This setting is practical for most data. It ensures efficient use of memory for storing messages on the TIM and produces as little message traffic on the WAN link as possible. Generally, the default setting for the *Image memory* parameter only needs to be changed to the *Send buffer method* with a few objects whose data changes must be stored individually on the TIM and sent individually to the partner, for example alarms with a time stamp or analog values with a time stamp for entry in archives. With the *Bin04B_S* object, selected individual binary inputs can be transmitted using the send buffer method even if the *Image memory* function is active. You make this setting during the parameter assignment of the channel type *Binary send* in the *Send buffer principle mask* parameter.

Name:	**Conditional spontaneous**
Range of values:	Function active: Transmission is conditional spontaneous Function deactivated: The transmission is unconditional spontaneous
Default:	Function active
Explanation:	If the function is activated (transmission is conditional spontaneous), the message does not trigger connection establishment in dial-up networks. When using a dedicated lines or Ethernet, this parameter has no significance since transmission is then always immediate. If the function is deactivated (transmission is unconditional spontaneous), the message triggers connection establishment immediately in dial-up networks. In the *Bin04B_S* object, even if the *Conditional spontaneous* function is activated, selected individual binary inputs can trigger *unconditional spontaneous* transmission. You make this setting during the parameter assignment of the channel type *Send binary value* in the *Alarm mask* parameter.

Name:	**High priority**
Range of values:	Function active, function deactivated

Default:	Function deactivated
Explanation:	Important process data can be given higher priority than the messages waiting in the send buffer. If you activate the *High priority* function, the messages of this object are given a higher priority and sent before the other buffered messages. **Note**: In dial-up networks, the *High priority* function does not necessarily lead to immediate connection establishment. This happens only when the *Conditional spontaneous* function has been deactivated for this object.

Volatile storage mode output box:
The *Volatile storage mode* output box indicates that a message already stored in the send buffer of the TIM will be deleted if the subscriber is not available. As long as a subscriber is disrupted, no new non-retentive messages for this subscriber can be entered in the send buffer of the TIM. The *volatile storage mode* applies only to the send messages of the following objects:
-*Cmd01B_S*
-*Par12D_S*

By deleting command, setpoint and parameter messages, you avoid out-of-date commands, setpoints all parameters being sent to the destination subscriber when a connection is re-established following a failure.

The messages of all other send objects are not deleted when there is a connection failure and further messages can be entered in the send buffer of the TIM during a disruption.

Name:	**Read cycle**
Range of values:	1: *Normal cycle* 2:*Fast cycle*
Default:	Normal cycle
Explanation:	Each data object that sends data must be assigned to one of the two read cycles. The *normal cycle* is the most suitable selection for most process data. Data that must be acquired quickly such as alarms and pulse messages are suitable for assignment to the *fast cycle*. Command, setpoint and parameter objects for which a 1-out-of-n check is required, must be assigned to the *fast cycle*. If these objects are acquired in the normal cycle, they are not included in the 1-out-of-n check. All send channels of a data object are included in the same read cycle. The parameters of the read/write cycle themselves are set in the basic settings of TD7onTIM.

4.4.8.2 Channel overview and functions of channel parameter assignment

Overview of the channel types

Each data object has a defined number of channels with default parameters. The number of channels and the data type cannot be changed for an individual data object. Apart from the setpoint and parameter objects, all data objects have one or more channels of the same type.

The channel types differ in the transmission direction relative to the communication partner and fall into the two classes *Send* and *Receive* channels:

- **Send channels** for the *send data* function:
  - Binary send
  - Analog send
  - Data send
  - Mean value send
  - Counted value send
  - Command send
  - Setpoint send
- **Receive channels** for the *receive data* function:
  - Binary receive
  - Analog receive
  - Data receive
  - Mean value receive
  - Counted value receive
  - Command receive
  - Setpoint receive

Dialogs for setting channel parameters

To set the parameters for the channels, you select the data object of a TIM with TD7onTIM in the directory tree. The list box displays the channels of this object with the channel name and channel type. If channels have already had parameters set, the input or output address is displayed in the list.

Below the list box, there is a parameter assignment dialog for the channel selected automatically or with the mouse in the list box.

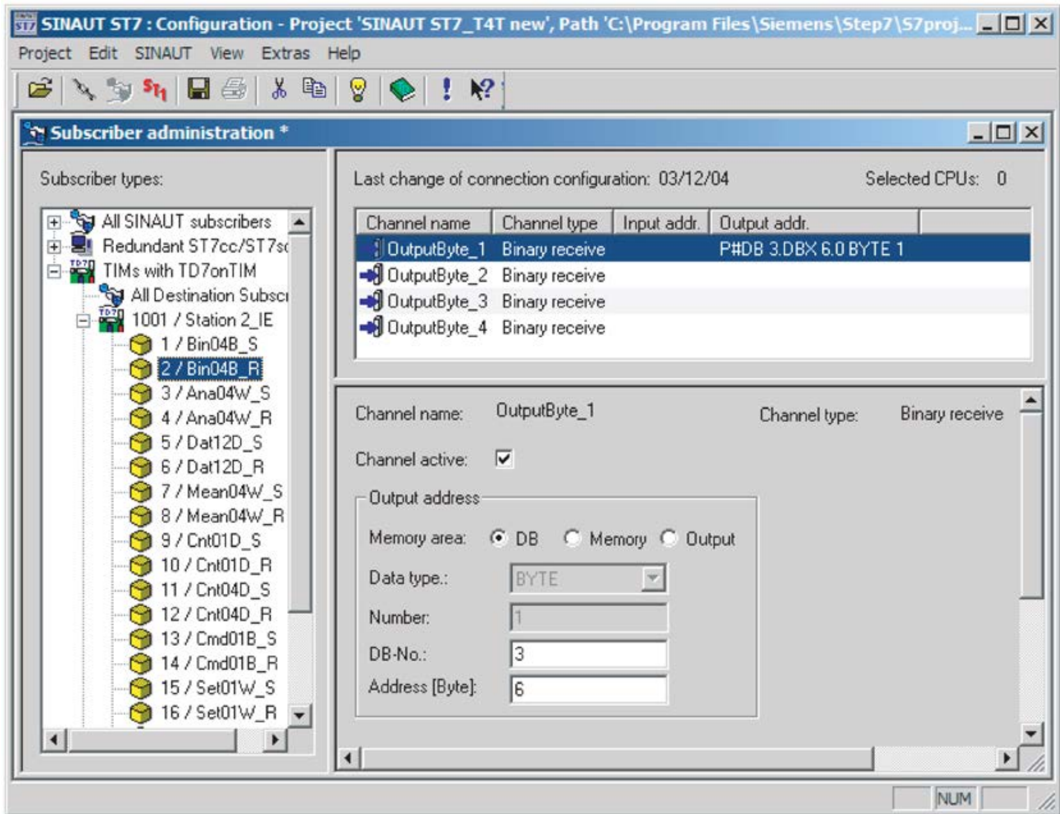


Figure 4-35 Selected object *Bin04B_R* with *Binary receive* channel selected in the list box and its parameter assignment dialog

Activating channels

Prior to parameter assignment, the send and receive channels are not yet active, the input boxes in the dialog are disabled.

Each required channel must be enabled in its parameter assignment dialog using the *Channel active* option.

If certain channels of a data object are not required, they can be ignored.

If users are not sure whether they actually require a channel that has already had parameters set, or when they want to disable a channel later (perhaps temporarily), they can deactivate each individual channel here without losing the parameter settings.

Copying channels

To simplify parameter assignment, channels along with their parameter assignment can be copied. If a data object requires more than one channel with the same parameter assignment (except for the input or output address) and if the parameter settings have been completed for one channel, this channel along with its parameters can be copied.

Follow the steps outlined below:

1. Select a channel in the list box.
2. Select *Copy* in the context menu (right mouse button).
3. Select a different channel of the same object or the channel of another data object of the same type.
4. Insert the channel with its parameters using the *Paste* context menu (right mouse button). The selected channel is overwritten by the channel and its parameters.

Note

When you copy channels, the channel you are copying is not added to the existing channels since the number of channels is fixed per data object. The channel selected prior to paste is overwritten by the copied channel and its parameters.

4.4.8.3 Mandatory parameters of the send channels

During the parameter assignment of each send channel, you must first set the mandatory parameters "Input address" and "Send trigger".

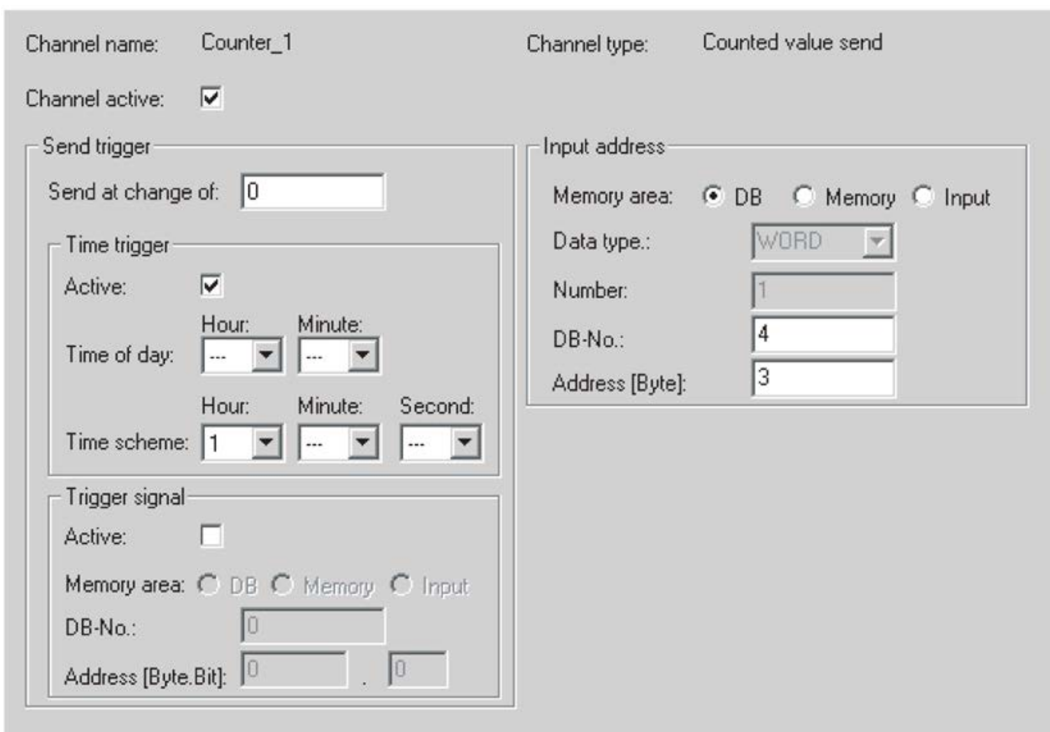


Figure 4-36 Parameter assignment dialog of a send channel based on the example of "Counted value send"

The input address

Data transferred from TD7onTIM to a communication partner is first read from the work memory of the CPU. For each send channel, the source address (input address) must be specified in the relevant memory area of work memory on the local CPU module from which the data will be read.

Name:	**Input address**
Memory area:	The following options are available for the source address: • DB Data block • Bit memory: Memory bit area • Input: Process input image (PII)
Data type:	The output box displays the default data format of the corresponding channel type: • Binary send: BYTE • Analog send: WORD • Send data: DWORD (double word) Send data (with "SmsDataInput" channel"): CHAR • Mean value send: WORD • Counted value send: WORD • Command send: BYTE • Setpoint send (object Set01W_S): WORD • Setpoint send (object Par12D_S): DWORD
DB No.:	Specifies the DB number in the CPU if the data block memory area (DB) was selected
Address [Byte]:	Input field for the byte number in the selected memory area. For data types involving more than one byte (WORD, DWORD), the least significant byte number must be entered as in STEP 7.
Number:	The number of double words included in the array (maximum 12). The parameter is available only for the following channel types: • Data send with object Dat12D_S • Setpoint send with object Par12D_S The parameter is explained along with these object-specific channels.

Note

Only the specified address areas can be read by TD7onTIM. Data from other areas, for example analog values acquired over peripheral input words (PIW) must be mapped to the bit memory or data block area by the user program.

The send trigger

The second mandatory parameter that must be set for send channels specifies when the data will be sent. This setting is made in the "Send trigger" area.

Four options are available for the send trigger. For each channel, you can configure a single option or a combination of different options.

Note

The send trigger can be set individually for each channel. If an object has more than one channel, remember that the activation of the send trigger of one of the object channels will transmit all channels of the data object.

Since the various options of the send trigger must be activated alternatively or additionally with most channel types, the parameter setting should selected to produce practical results.

Name:	**Send at change of**
Range of values:	In general: 0 ... 32767 • Channel type Analog send (object Ana04W_S): 0 ... 32767 • Channel type Command send (object Cmd01B_S): 0 or 1 • Channel type Data send (object Dat12D_S): 0 or 1 • Channel type Setpoint send (object Set01W_S or Par12D_S) 0 or 1
Default:	• For binary value, counted value, data and command channels: 1 • For mean value and setpoint channels: 0 • For analog value channels: 270 (1% of the raw value of S7 analog input modules, 27648 = 100%)
Explanation:	The value must be entered (as an integer) in the input field by which the process value must change so that it is transferred again. If you enter 0 (zero), the function is deactivated.

The "Time trigger" area provides two alternatives "Time of day" or "Time scheme". A combination of both is not possible.

The "Time trigger" send trigger is activated with the "Active" option.

Name:	**Time of day**
Range of values:	Time of day (hour and minute)
Default:	Function not active
Explanation:	If the send trigger is triggered by a time of day, the data is read out at the selected time of day and a message is sent.

Name:	**Time scheme**
Range of values:	Time scheme (hour, minute or second)

Default:	Function not active with the exception of "Mean value send" (Mean04W_S): Function active, time scheme: 15 minutes
Explanation:	If the send trigger is triggered by a time scheme, the data is read out at the selected time interval and a message is sent.

If the time trigger is, for example, initially set to time of day and then to time scheme, the previously set times are replaced by dashes (- - -).

The "Trigger signal" area includes the option of an external send trigger that can be activated with the "Active" option.

Name:	**Trigger signal**
Memory area:	The following options are available for the source address: • DB Data block • Bit memory: Memory bit area • Input: Process input image (PII)
Default:	Function deactivated
DB No.:	Specifies the DB number in the CPU if the data block memory area (DB) was selected
Address [Byte]:	Input field for the byte number in the selected memory area
Explanation:	If a trigger signal is specified, the data of the object is transferred when the trigger signal changes from 0 to 1. If the trigger signal is a memory or data bit, it is automatically reset after it has been read. The reset can, if necessary, be evaluated by the user program, for example to display that the message was triggered.

4.4.8.4 Mandatory parameters of the receive channels

When you set parameters for each receive channel, the Output address parameter must be set as a mandatory parameter. This is the address in the memory area of the local CPU to which the received data will be written by TD7onTIM.

The parameter assignment dialog of the "Binary receive" channel type illustrates the "Output address" area.

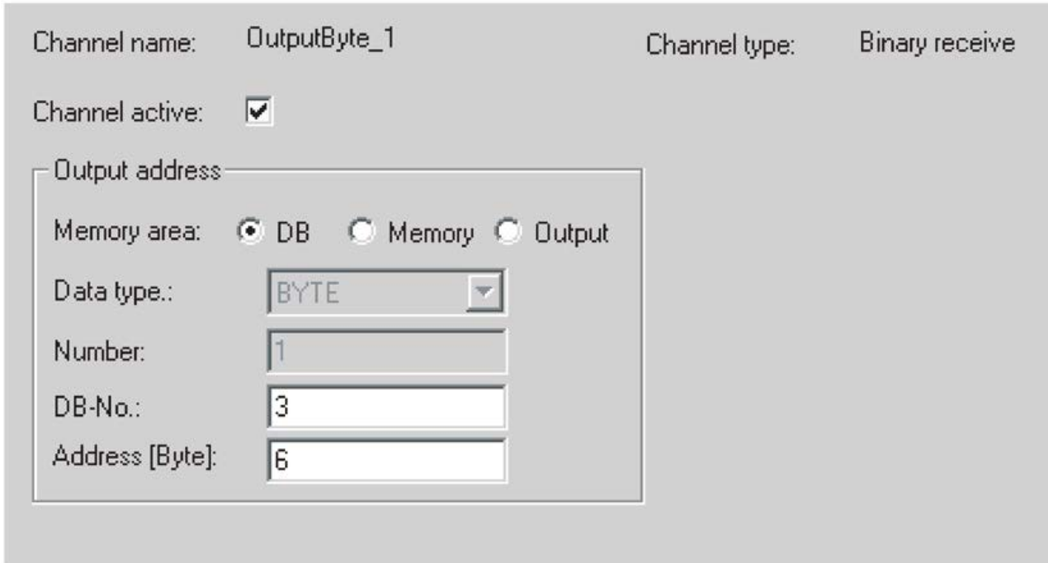


Figure 4-37 Parameter assignment dialog of the "Binary receive" receive channel

The output address

The destination address in the memory area of the CPU is set in the "Output address" area by specifying the following information:

Name: **Output address**

Memory area: The following options are available for the destination address:

- DB Data block
- Bit memory: Memory bit area
- Output: Process output image (PIQ)

Data type: The output box displays the default data format of the corresponding channel type:

- Binary receive: BYTE
- Analog receive: WORD
- Receive data: DWORD (double word)
- Mean value receive: WORD
- Counted value receive: DWORD
- Command receive: BYTE
- Setpoint receive (object Set01W_R): WORD
- Setpoint receive (object Par12D_R): DWORD

DB No.: Specifies the DB number in the CPU if the data block memory area (DB) was selected

Address [Byte]: Input field for the byte number in the selected memory area. For data types involving more than one byte (WORD, DWORD), the least significant byte number must be entered as in STEP 7.

Number: The number of double words included in the array (maximum 12). The parameter is available only for the following channel types:

- Data receive with object Dat12D_R
- Setpoint receive with object Par12D_R

The parameter is explained along with these object-specific channels.

Note

Only the specified address areas can be written by TD7onTIM. Data for other areas, for example analog values output over peripheral output words (PQW) must be mapped over the bit memory or data block area by the user program.

Apart from these generally valid channel parameters, most send and receive channels have specific parameters that are described below based on the individual data objects.

4.4.8.5 Bin04B_S/_R

The parameters of the obligatory channels are described in the following sections:

Mandatory parameters of the send channels (Page 202)

Mandatory parameters of the receive channels (Page 205)

Specific parameters of the data object "Bin04B_S"

- Channel type: *Binary send*:
  The *Masks* field provides three options for transmitting binary value messages. You can specify bit-by-bit whether certain bits do not trigger message transmission or which bits trigger a different type of transmission than was specified in the basic parameters of the object. This setting is made in the *Masks* area.

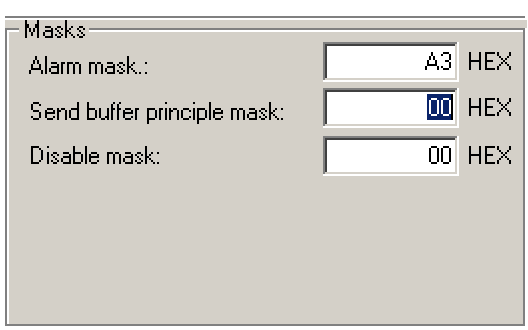


Figure 4-38 The *Masks* area in the parameter assignment dialog of the *Binary send* channel type

Name:	**Alarm mask**
Range of values:	Mask in hexadecimal format
Default:	00 (hex)
Explanation:	Changes in masked bits in the byte of the *Binary send* channel cause an unconditional spontaneous message transmission. Changes from 0 to 1 and from 1 to 0 are evaluated. The corresponding bits are masked in hexadecimal format. The *alarm mask* is only practical when the object is transmitted over a dial-up network and the option *Conditional spontaneous* was activated in the basic parameters of the object.

Name:	**Send buffer principle mask**
Range of values:	Mask in hexadecimal format
Default:	00 (hex)
Explanation:	Changes in masked bits in the byte of the *Binary send* channel cause a message transmission according to the send buffer principle. Changes from 0 to 1 and from 1 to 0 are evaluated. The corresponding bits are masked in hexadecimal format. The*Send buffer principle mask* is only practical when the option *Image memory* was activated in the basic parameters of the object.

Name:	**Disable mask**
Range of values:	Mask in hexadecimal format
Default:	00 (hex)
Explanation:	Masked bits in the byte of the *Binary send* channel are ignored when changes are checked. This means that changes to the masked bits for this channel do not trigger message transmission. A masked bit always has the value 0 in the message. The corresponding bits are masked in hexadecimal format.

The bits are masked as shown in the following example in which the hexadecimal value *A3* is entered in the input field of the parameter assignment dialog. The bits with the value 1 are masked; in other words bits no. 0, 1, 5 and 7 cause the relevant function in the described masks.

Byte assignment								
Bit:	.7	.6	.5	.4	.3	.2	.1	.0
Masked	1	0	1	0	0	0	1	1
Hex mask	A				3			

Specific parameters of the data object "Bin04B_R"

- Channel type: *Binary receive*:
  This channel type has no specific parameters.

4.4.8.6 Ana04W_S/_R

The parameters of the obligatory channels are described in the following sections:

Mandatory parameters of the send channels (Page 202)

Mandatory parameters of the receive channels (Page 205)

Specific parameters of the data object "Ana04W_S"

- Channel type: *Analog send*:
  The following parameters are available in the *Processing parameters* area:

Name:	**Unipolar analog value**
Range of values:	Function active, function deactivated
Default:	Function active
Explanation:	If the function is activated, negative analog values are corrected to the value zero. The error ID 8000h (-32768), that is displayed, for example, if there is a wire break in *life zero* inputs, is nevertheless transmitted.

Name:	**Smoothing factor**
Range of values:	1 = none, 4 = weak, 32 = medium, 64 = strong
Default:	1
Explanation:	Using the smoothing factor, quickly fluctuating analog values can be smoothed to a greater or lesser extent depending on the parameter setting. It may then be possible to set a lower value for the send trigger *Send at change of*. The smoothing factors are identical to the smoothing factors that are configured for some S7 analog input modules. The smoothing in the channel functions according to the same formula as on an input module:

$$y_n = \frac{x_n + (k-1)\,y_{n-1}}{k}$$

where

y_n = smoothed value in the current cycle n

y_n = acquired value in the current cycle n

k = smoothing factor

Name:	**Fault suppression time**
Range of values:	0 ... 32767 [s]
Default:	0
Explanation:	Transmission of an analog value located in the overflow or underflow range (7FFFh or 8000h) is suppressed for the time period specified here. The value 7FFFh or 8000h is only sent after this time has elapsed, if it is still pending. If the value returns to below 7FFFh or above 8000h 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 7FFFh or 8000h is received. This is typically used for temporary suppression of current values that may occur when powerful motors are started. The analog input may exceed several times the maximum range 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). When the user provides specific values, fault suppression is only possible if these also adopt the values 7FFFh or 8000h when the permitted ranges are exceeded. If this is not the case, the parameter does not need to have a value entered.

Specific parameters of the data object "Ana04W_R"

- Channel type: *Analog receive*:
  This channel type has no specific parameters.

4.4.8.7 Dat12D_S/_R

The parameters of the obligatory channels are described in the following sections:

Mandatory parameters of the send channels (Page 202)

Mandatory parameters of the receive channels (Page 205)

Specific parameters of the data object "Dat12D_S"

- Channel type: *Data send*:
  With the channel type *Data send*, a data field of 1 to a maximum of 12 double words can be sent. This setting is made with the *Number* parameter in the *Input address* area.

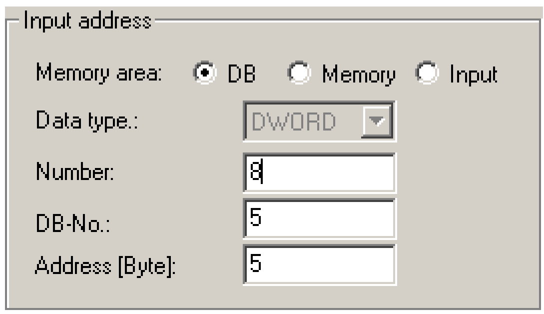


Figure 4-39 Section of the parameter assignment dialog of the channel type *Data send* with the *Number* parameter

Name:	**Number**
Range of values:	1 ... 12
Default:	12
Explanation:	The parameter decides the size of the data field in double words. This allows the message length to be reduced to the length actually required. This saves transmission time.

Specific parameters of the data object "Dat12D_R"

- Channel type: *Data receive*:
  With the channel type *Data receive*, a data field of 1 to a maximum of 12 double words can be received. This setting is made with the *Number* parameter in the *Output address* area.

Name:	**Number**
Range of values:	1 ... 12
Default:	12
Explanation:	The parameter specifies the size of the data field in double words that must be identical to the data field size of the sending partner object *Dat12D_S*.

4.4.8.8 Mean04W_S/_R

The parameters of the obligatory channels are described in the following sections:

Mandatory parameters of the send channels (Page 202)

Mandatory parameters of the receive channels (Page 205)

Specific parameters of the data object "Mean04W_S"

- Channel type: *Mean value receive*:
  This channel type has no specific parameters. The duration of the interval for forming an individual mean value is determined by the *Time trigger* specified for the *Mean value send* channel.
  If the mean value is to be entered in an archive in the control center, the *Mean04W_S* object should be transmitted according to the send buffer principle.

Specific parameters of the data object "Mean04W_R"

- Channel type: *Mean value receive*:
  This channel type has no specific parameters.

4.4.8.9 Cnt01D_S/_R, Cnt04D_S/_R

The parameters of the obligatory channels are described in the following sections:

Mandatory parameters of the send channels (Page 202)

Mandatory parameters of the receive channels (Page 205)

Specific parameters of the data objects "Cnt01D_S" and "Cnt04D_S"

The object type *Cnt01D_S* sends a counted value, *Cnt04D_S* sends four counted values.

- Channel type: *Counted value send*:
  This channel type has no specific parameters.
  The counted value read in from the CPU must be made available by a software counter of the CPU in the WORD format.
  In TD7onTIM, the value originating from the CPU counter is compared with the value last read from the object and the difference is added to the SINAUT counted value maintained internally in the *Cnt01D_S* or *Cnt04D_S* object. An overflow of the CPU counter at 65535 is detected by the counted value object and taken into account.
  The internally formed SINAUT counted value is stored by TD7onTIM in DWORD format and transmitted.

Specific parameters of the data objects "Cnt01D_R" and "Cnt04D_R"

The object type *Cnt01D_R* receives a counted value, *Cnt04D_R* receives four counted values.

- Channel type: *Counted value receive*:
  The SINAUT counted value received from the partner object is compared with the last received counted value and the difference is added to the value at *CountedValueOutput*.
  The value is output in DWORD format so that the maximum displayable counted value is 2,147,483,647. If the maximum value that can be represented is exceeded, the counted value starts again at 0 and counting continues in the positive numeric range.

  Reset:
  If the counted value at *CountedValueOutput* is reset to zero, when the next counted value is received, the difference is added to zero. To reset, a bit in a memory area of the CPU is defined in the *Reset* field. The reset takes place on a signal edge change from 0 to 1.

Name:	**Reset**
Memory area:	The following options are available for the address: - DB: Data block - Memory bit: Memory area - Input: Process input image (PII)
Default:	Function inactive
DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected
Address [Byte.Bit]:	Input fields for the byte and bit number in the selected memory area

4.4.8.10 Cmd01B_S/_R

The parameters of the obligatory channels are described in the following sections:

Mandatory parameters of the send channels (Page 202)

Mandatory parameters of the receive channels (Page 205)

Specific parameters of the data object "Cmd01B_S"

If increased reliability is required for the input of commands, setpoints and parameters, all objects with which this data is sent should be assigned to the fast cycle. All command, setpoint and parameter objects in the fast cycle are subjected to a 1-out-of-n check; in other words, at the end of the fast cycle there is a check to make sure that there is a command, setpoint or parameter entry for only *one* of the acquired objects. Only then is the corresponding entry processed and transferred. If there is more than one entry, the entries are rejected. A new command, setpoint or parameter is processed only when previously no entry was acquired in one fast cycle. The error status is indicated in the output byte of the *OpInputMonitor* system object using the 1-out-of-n error bit.

- Channel type: *Command send*:
  Only one input may be set at the same time for command input. This is verified by the 1-out-of-8 check. If a 1-out-of-8 error is detected, the entered commands are not processed further. A new command is only processed again when there was previously no command pending for one read cycle. The error status is displayed in the output byte of the *OpInputMonitor* system object in bit 2 (*1-out-of-n error*). The bit remains set until the error is corrected.

  If the command is entered over *digital inputs*, for example using a button connected to them, the button must remain pressed until it is acquired by TD7onTIM. The output byte of the OpInputMonitor system object indicates when the command has been acquired in the Input OK bit. OpInputMonitor also takes into account any minimum input time that has been set for it; in other words the button must be pressed at least as long as this selected time. Only then is *Input OK* indicated.
  If the commands are input over *memory* or *data* bits, the set bit is automatically reset by TD7onTIM as soon as it is acquired. Here, there is *no Input OK* display. Successful acquisition can, however, be recognized indirectly because the command bit was reset.

Special feature of send trigger:
For the *Send at change of* send trigger, only the values 0 and 1 can be set. Values higher than 1 are meaningless for command input.

Disable mask:
For the *Command send* channel type, individual bits can be masked for command acquisition. You do this in the *Disable mask*:

Name:	**Disable mask**
Range of values:	Mask in hexadecimal format
Default:	00 (hex)
Explanation:	Masked bits in the byte of the *Command send* channel are ignored when changes are checked. This means that changes to the masked bits for this channel do not trigger message transmission. A masked bit always has the value 0 in the message. The corresponding bits are masked in hexadecimal format.

The bits are masked as shown in the following example in which the hexadecimal value *A3* is entered in the input field of the parameter assignment dialog. The bits with the value 1 are masked; in other words bits no. 0, 1, 5 and 7 are ignored in command acquisition.

Byte assignment								
Bit:	.7	.6	.5	.4	.3	.2	.1	.0
Masked	1	0	1	0	0	0	1	1
Hex mask	A				3			

Specific parameters of the data object "Cmd01B_S"

- Channel type: *Command receive*:

Name:	**Command output time**
Range of values:	0 ... 50 [s x 0.1] (5 = 0.5 seconds)
Default:	5
Explanation:	This is the command output time for the command outputs of the channel. When the set time has elapsed, the command output is reset again by TD7onTIM. The *command output time* applies to all 8 command outputs of the *Command receive* channel. If the *command output time* is set to zero, a set command output is not reset by TD7onTIM. Resetting the command output to zero must then be implemented in the user program.

4.4.8.11 Set01W_S/_R

The parameters of the obligatory channels are described in the following sections:

Mandatory parameters of the send channels (Page 202)

Mandatory parameters of the receive channels (Page 205)

Specific parameters of the data object "Set01W_S"

The object type *Set01W_S* sends 1 setpoint. The setpoint assignment status 'local' and the locally valid setpoint can also be indicated by this object.

If increased reliability is required for the input of commands, setpoints and parameters, all objects with which this data is sent should be assigned to the fast cycle. All command, setpoint and parameter objects in the fast cycle are subjected to a 1-out-of-n check; in other words, at the end of the fast cycle there is a check to make sure that there is a command, setpoint or parameter entry for only *one* of the acquired objects. Only then is the corresponding entry processed and transferred. If there is more than one entry, the entries are rejected. A new command, setpoint or parameter is processed only when previously no entries were acquired in one fast cycle. The error status is indicated in the output byte of the OpInputMonitor system object using the 1-out-of-n error bit.

If a setpoint entry is transmitted as a result of the *Trigger signal* send trigger, and if this is triggered over a digital input, for example by a connected button, the button must remain activated until the signal has been acquired by TD7onTIM. The status byte for operator input of the *OpInputMonitor* system object indicates whether the entry has been acquired using the *Input OK* bit.
OpInputMonitor takes into account any minimum input time; in other words, the button must be kept pressed for at least as long as the time set with this parameter. Only then is *Input OK* indicated.
If the trigger signal is a *memory* or *data* bit, the bit is automatically reset by TD7onTIM as soon as it is acquired. Successful acquisition can be recognized indirectly because the trigger bit was reset.

- Channel name: *LocalOperation* - Setpoint assignment status 'local'
  (Channel type: *Binary receive*):
  The return message from the partner object that the local object is set to 'local operation' is sent over this channel. The *LocalOperation* channel is used only for signaling. It can but does not necessarily need to be used.
  A setpoint can also be set locally at the partner object that receives the setpoint. As information, the input parameter *Local* can then be set to 'local' locally on the partner object. The current status of the *Local* input parameter is reported by the partner object and displayed here at the *LocalOperation* output.
  An interlock of the remote and local setpoint assignment must be implemented by the user program; in contrast to TD7onCPU, in TD7onTIM, this parameter (or the *Local* parameter of the partner object *Set01W_R*) does not cause an interlock with local setpoint assignment.
  After the local or the partner CPU has started up or after the return of the connection, a general request makes sure that the current valid status of the partner is indicated in *LocalOperation*.

  If you do not require the parameter, leave it inactive.
  This channel type has no specific parameters.
- Channel name: *ReturnedSetpoint* - returned setpoint
  (channel type: *Data receive*):
  In TD7onTIM, the partner object receiving the setpoint reports back the currently valid local setpoint when the *LocalSetpointInput* channel is set there. This returned value is

displayed at the *ReturnedSetpoint* output. If the partner object is set to 'local' and if a new entry is made there, the setpoint changed there is indicated at *ReturnedSetpoint* if the *Local* channel is set on the partner object.
After the local or partner CPU has started up or after return of the connection, an automatic general request makes sure that the currently valid local setpoint is indicated at *ReturnedSetpoint*.

If you do not require the parameter, leave it inactive.
This channel type has no specific parameters.

- Channel name: *SetpointInput* - Setpoint input
(channel type: *Setpoint send*):
The setpoint to be sent to the partner object is entered over this channel.

  This channel type has no specific parameters.

Specific parameters of the data object "Set01W_R"

The object type *Set01W_R* receives 1 setpoint. The object also has an input over which the locally valid setpoint can be returned.

- Channel name: *Local* - Setpoint assignment 'local'
(Channel type: *Message send*):
The information as to whether the locally set setpoint or the remote setpoint is valid can be signaled over this input. The current status of the *Local* input and a copy of the current local setpoint is returned (mirrored) to the partner. A setpoint sent by the remote partner (for example, master station) can also be accepted when *Local* is set.
In contrast to TD7onCPU, the *Local* cannel here is only for information. An interlock with the remote setpoint assignment must be implemented in the user program.

  If you do not require the parameter, leave it inactive.
  This channel type has no specific parameters.

- Channel name: *LocalSetpointInput* - Local setpoint input
(channel type: *Data send*):
A locally active setpoint can be returned over the *LocalSetpointInput* input to the send block.

  If you do not require the parameter, leave it inactive.
  This channel type has no specific parameters.

- Channel name: *SetpointOutput* - Setpoint output
(channel type: *Setpoint receive*):
The setpoint sent by the partner object or entered locally at *LocalSetpointInput* is output at *SetpointOutput*.

  This channel type has no specific parameters.

4.4.8.12 Par12D_S/_R, Par12x1D_R

The parameters of the obligatory channels are described in the following sections:

Mandatory parameters of the send channels (Page 202)

Mandatory parameters of the receive channels (Page 205)

Specific parameters of the data object "Par12D_S"

The object type "Par12D_S" sends 1 to 12 parameters. The setpoint assignment status 'local' and the locally valid parameters can also be indicated by this object.

If increased reliability is required for the input of commands, setpoints and parameters, all objects with which this data is sent should be assigned to the fast cycle. All command, setpoint and parameter objects in the fast cycle are subjected to a 1-out-of-n check; in other words, at the end of the fast cycle there is a check to make sure that there is a command, setpoint or parameter entry for only one single acquired object. Only then is the corresponding entry processed and transferred. If there is more than one entry, the entries are rejected. A new command, setpoint or parameter is processed only when previously no entry was acquired in one fast cycle. The error status is indicated in the output byte of the "OpInputMonitor" system object using the "1-out-of-n error" bit.

If a parameter entry is transmitted as a result of the "Trigger signal" send trigger, and if this is triggered over a digital input, for example by a connected button, the button must remain activated until the signal was acquired by TD7onTIM. The status byte for operator input of the "OpInputMonitor" system object indicates whether acquisition was successful in the "Input OK" bit.

OpInputMonitor also takes into account any minimum input time that has been set for it; in other words the button must be pressed at least as long as this selected time. Only then is "Input OK" indicated.
If the trigger signal is a memory or data bit, the bit is automatically reset by TD7onTIM as soon as it is acquired. Successful acquisition can be recognized indirectly because the trigger bit was reset.

- **LocalOperation**

  Meaning: Setpoint assignment status 'local'

  Channel type: Binary receive

  The return message from the partner object that the local object is set to 'local operation' is sent over this channel. The "LocalOperation" channel is used only for signaling. It can, but does not need to be modified.

  A parameter can also be set locally at the partner object that receives the parameter. As information, the input channel "Local" can then be set to 'local' locally on the partner object. The current status of the "Local" input channel is reported by the partner object and displayed here at the "LocalOperation" output.

  An interlock of the remote and local setpoint assignment must be implemented in the user program; in contrast to TD7onCPU, in TD7onTIM, this parameter (or the "Local" parameter of the partner object "Par12D_R") does not cause an interlock of the remote and local setpoint assignment.

  After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the current, valid status of the partner is displayed at "LocalOperation".

  If you do not require the parameter, leave it disabled.

  This channel type has no specific parameters.

- **ReturnedParameter**

  Meaning: Returned parameter

  Channel type: Receive data

  In TD7onTIM, the partner object receiving the parameter values reports back the currently valid local parameter values when the "LocalParameterInput" channel is set there. These returned values are indicated at the "ReturnedParameter" output. If the partner object is set to 'local' and if new input is made there, the parameters changed there are indicated at "ReturnedParameter" if the "Local" channel is set for the partner object.

  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, leave it disabled.

  This channel type has no specific parameters.

- **ParameterInput**

  Meaning: Parameter input

  Channel type: Setpoint send

  The parameters to be sent to the partner object are entered over this channel. The content of each double word may be a value in double word (DWORD) format, it can also be a mixture of other formats which together form a double word, for example:

  – 4 bytes

  – 2 words

  – 2 bytes plus 1 word

  The number of required double words, in other words the size of the data field is set in the *Number* input box.

Name:	**Number**
Range of values:	1 ... 12
Default:	12
Explanation:	The parameter decides the size of the data field in double words. This allows the message length to be reduced to the length actually required. This saves transmission time.

Specific parameters of the data object "Par12D_R"

The "Par12D_R" object type receives 1 to 12 parameters, for example setpoints. The object also has an input channel over which the locally valid parameters can be returned.

- **Local**

  Meaning: Parameter assignment 'local'

  Channel type: Binary send

  The information as to whether the locally set parameters or the remote parameters are valid can be signaled over this input. The current status of the "Local" input and a copy of the local parameters is returned (mirrored) to the partner. A parameter set sent by the remote partner (for example, master station) can also be accepted when "Local" is set.

  In contrast to TD7onCPU, the channel is only used for information. An interlock with the remote setpoint assignment must be implemented in the user program.

  If you do not require the parameter, leave it disabled.

  This channel type has no specific parameters.

- **LocalParameterInput**

  Meaning: Local parameter input

  Channel type: Send data

  Local active parameters can be returned to the send block via the "LocalParameterInput" input.

  The data area corresponds to the data area set for "ParameterOutput" of the same object.

  If you do not require the parameter, leave it disabled.

  This channel type has no specific parameters.

- **ParameterOutput**

  Meaning: Parameter output

  Channel type: Setpoint receive

  The parameters sent by the partner object or entered locally at "LocalParameterInput" are output at "ParameterOutput".

  The data area can vary in length between 1 and 12 double words. The content of each double word may be a value in double word format, it can also be a mixture of other formats which together form a double word, for example:

  – 4 bytes

  – 2 words

  – 2 bytes plus 1 word

Name:	**Number**
Range of values:	1 ... 12
Default:	12

Explanation: The parameter decides the size of the data field "ParameterOutput" in double words. This must be identical to the data field size of the sending partner object "Par12D_S".

Specific parameters of the data object "Par12x1D_R"

The object type "Par12x1D_R" is used for the selective receipt and selective output of a maximum of 12 individual parameter values each with 1 double word per parameter.

In contrast to the typical "Par12D_R", at any one time "Par12x1D_R" can only receive 1 of the maximum 12 parameter values and write it to the CPU. For this reason, "Par12x1D_R" may only be configured in conjunction with a PC station of the type SINAUT ST7cc or SINAUT ST7sc. ST7cc/sc can also not send more than 1 parameter at any one time.

Note

No "Par12D_S" in conjunction with "Par12x1D_R"

The use of "Par12D_S" (TD7onTIM or TD7onCPU) in conjunction with "Par12x1D_R" is not permitted.

With SINAUT ST7cc or ST7sc as communications partner, the use of Par12x1D_R avoids all 12 parameter values being written to the CPU by TD7onTIM when a parameter is received When receiving a parameter with Par12x1D_R, this means that the other 11 parameters that might otherwise be modified on the CPU remain untouched.

The parameters can be written to a data block, a bit memory area or directly to the outputs.

Just like Par12D_R, Par12x1D_R also has an additional input channel via which the locally valid parameters can be returned.

Note

Mapping of the data of the output area necessary for returning

Parameters that were written to the memory area for outputs, cannot be returned directly but must first be mapped by the user program from the output area to a data block or bit memory area before they can be returned.

- **Local**

  Meaning: Parameter assignment 'local'

  Channel type: Binary send

  The information as to whether the locally set parameters or the remote parameters are valid can be signaled over this input. The current status of the "Local" input and a copy of the local parameters is returned (mirrored) to the partner. The parameters sent by the remote partner (ST7cc/sc) can also be accepted by the object when "Local" is set.

  In contrast to TD7onCPU, the channel is only used for information. An interlock with the remote setpoint assignment must be implemented in the user program.

  If you do not require the parameter, leave it disabled.

  This channel type has no specific parameters.

- **LocalParameterInput**

  Meaning: Local parameter input

  Channel type: Send data

  Local active parameters can be returned to the send block via the "LocalParameterInput" input.

  The length of the data area is set automatically and corresponds to the sum of the activated "ParameterOutput" channels that have a length of 1 double word. This means that all set output channels are always returned.

  If you do not require the parameter, leave it disabled.

  This channel type has no specific parameters.

- **ParameterOutput**

  Meaning: Parameter output (max. 12 parameters)

  Channel type: Setpoint receive

  The parameters sent by the partner object or entered locally at "LocalParameterInput" are output at the "ParameterOutput" outputs.

  For each parameter used, 1 channel must be activated. The memory space of channel 1 is permanently occupied (see configuration example below).

  The data area (parameter "Number" in the "Output address" field) has a fixed length of 1 double word for each "ParameterOutput" channel. 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,

  – 4 bytes

  – 2 words

  – 2 bytes plus 1 word

  Configuration example:

  – Specify the memory area (DB, bit memory or output) and the start address in channel 1.

  – If, for example, you require all 7 channels, activate channel 1 and channel 7.

    The channels 2 ... 6 in between are activated automatically.

4.4.8.13 SMS01_S

Functions and channels of the data object "Sms01_S"

Sending SMS messages via TD7onTIM is supported only in conjunction with an S7-300 CPU with party line. On the party line, see Glossary.

An "Sms01_S" object is used to configure an SMS message.

Until an "Sms01_S" object is assigned to an SMSC, it is marked with a red cross in the navigation area and in the object list. After it has been assigned to an SMSC, the "Sms01_S" object is marked by a green check mark.

Configuring the phone number of the mobile wireless recipient on the CPU

The phone number of the mobile wireless recipient is configured on the CPU and read out from there. The storage location is specified in the "**Input address of SMS phone number**" box.

The format of the phone number is ASCII; it must be available on the CPU as a numeric string of the type character with a maximum of 40 characters. Exception: It may be preceded by the plus character (+).

If the phone number on the CPU is shorter than the length specified in the SINAUT configuration tool, the phone number on the CPU must be completed with a null byte (0x00).

The phone number can be changed at any time on the CPU online without reconfiguration.

SmsPhoneNumber

- **Input address of SMS phone number**

Name:	**Input address of SMS phone number**
Memory area:	The following options are available for the source address: • DB Data block • Bit memory: Memory bit area • Input: Process input image (PII)
Data type:	The output box displays the default data format of the corresponding channel type (here "CHAR").
DB No.:	Specifies the DB number in the CPU if the data block memory area (DB) was selected.
Address [Byte]:	Input field for the byte number in the selected memory area. Analogous to STEP 7, the least significant byte number needs to be entered.
Number:	The number of bytes. Note the information in the section "Configuring the phone number of the mobile wireless recipient on the CPU".

Note

Only the specified address areas can be read by TD7onTIM. Data from other areas, for example analog values acquired over peripheral input words (PIW) must be mapped to the bit memory or data block area by the user program.

- **SMS send trigger**

  The SMS send trigger specifies the reason for the data being sent.

  In the drop-down list, there are three options for triggering an SMS transmission:

  – **Send on coming event**

    Sending of the SMS message with each incoming event at the "**SMS trigger signal**" input

    An "**SMS time trigger**" can only be specified with this option.

  – **Send when the CPU changes to STOP**

    Send the SMS message when the CPU changes to STOP.

  – **Send whenever the CPU changes mode**

    Send the SMS message each time the CPU changes mode

- **SMS time trigger**

  The SMS time trigger is activated with the "Active" option.

  The "SMS Time trigger" area provides two alternatives "Time of day" or "Time scheme". A combination of both is not possible.

Name:	**Time of day**
Range of values:	Time of day (hour and minute)
Default:	Function not active
Explanation:	If the send trigger is triggered by a time of day, the data is read out at the selected time of day and a message is sent.

Name:	**Time scheme**
Range of values:	Time scheme (hour, minute or second)
Default:	Function not active
Explanation:	If the send trigger is triggered by a time scheme, the data is read out at the selected time interval and a message is sent.

- **SMS trigger signal**

The "SMS trigger signal" area includes the option of an external send trigger that can be activated with the "Active" option.

Name:	**SMS trigger signal**
Memory area:	The following options are available for the source address: • DB Data block • Bit memory: Memory bit area If the phone number is configured in memory bytes, remember that the CPU sets the memory bytes to 0x00 at each restart which means that no SMS messages can be sent when the CPU starts up. • Input: Process input image (PII)
Default:	Function deactivated
DB No.:	Specifies the DB number in the CPU if the data block memory area (DB) was selected
Address [Byte]:	Input field for the byte number in the selected memory area
Explanation:	If an SMS trigger signal is specified, the data of the object is transferred when the trigger signal changes from 0 to 1. If the trigger signal is a memory or data bit, it is automatically reset after it has been read. The reset can, if necessary, be evaluated by the user program, for example to display that the message was triggered.

SmsText

The text of the SMS message is configured with this channel. The following options are available:

- **Text:**

  The message text of the SMS message is entered in the "Text" input box. A maximum of 155 characters including the replacement string can be transferred.

- **Insert replacement string:**

  With this list box, you can insert replacement strings as keywords in the text of the SMS. The following replacement strings can be inserted once in the text:

  – Acquisition time

    Time at which the transmitted process value was acquired

  – Station name

    The name of the sending station configured in STEP 7

  – SMS object no.

    Object no. of the SINAUT object of the sent value

  – Process value

    Address (byte.bit) of the transfer value in the process image of the CPU

  – CPU status

    Mode of the CPU (RUN, STOP)

  The acquisition time stamp and the process variable are inserted in the SMS text during runtime.

  The station name and the SMS object number are set during configuration and are used after it is saved when generating the SDB. By specifying the station name and the SMS object no. every SMS message in an S7 project can be uniquely identified.

Note

Blocking special characters

When configuring SMS messages, a function is available with which you can check the SMS character set. When entering text for SMS messages, this function blocks certain special characters that can cause problems with some SMS providers. Illegal special characters are blocked directly when they are entered.

The following characters can be entered:

- All digits
- All letters (except umlauts ä, ü etc.)
- The space and the following special characters:
  . , ! % ? & / ( ) * + - =

All other special characters, accents and umlauts are blocked.

SmsDataInput

You will find the description of the obligatory parameters of this channel in the section Mandatory parameters of the send channels (Page 202).

Note the following special feature of the "Input address" for SmsDataInput:

Since the data value of SmsDataInput is inserted in the static SMS text, the data value must be specified in ASCII format (data type CHAR).

The CPU has various functions available for converting to the CHAR data format, for example I_STRING or DI_STRING for integer values (see block container "IEC Function Blocks" of the standard library).

If there are invalid characters in the address area of the data value, these are replaced by spaces in the SMS text.

SmsStateOutput

You will find the description of the obligatory parameters of this channel in the section Mandatory parameters of the receive channels (Page 205).

Using "SmsStateOutput" (double word), TD7onTIM sends various statuses to the CPU. The first half of the double word always contains the object number of the SMS typical. This ensures fast and simple assignment of the value to the object.

Table 4- 10 Status

Value (hex)	Meaning
Object no. + 7000	Ready to send SMS: • Start value (no SMS sent yet) or • Following an outgoing error, when the send trigger event has been reset by the CPU.
Object no. + 7001	SMS currently being sent.
Object no. + 7002	SMS sent successfully.
Object no. + 8001	Errors: SMS could not be sent.
Object no. + 8002	Errors: Phone number wrong
Object no. + 8003	Errors: New send trigger event detected while the SMS message is being sent.

- If an SMS message has never been sent, and TD7onTIM is ready to send, the CPU receives the value 7000.
- If a send trigger event is detected, the phone number is read in and checked for plausibility:
  - If the phone number is plausible, the status 7001 is sent and the message prepared for the interface.
  - If an invalid phone number is detected, the CPU is informed of this with the value 8002. Each further send trigger event is also discarded until the phone number has been corrected.

    If the phone number is valid, the status 7000 is set and a new SMS message can be sent.
- If a going and coming of the send trigger event is detected while the TIM is sending a message, it is blocked with the value 8003. The previous send job is not aborted.

  If TD7onTIM receives a positive or negative acknowledgement for this job from the interface, either the value 7002 or 8001 is sent to the CPU.

4.5 Saving and generating system data

4.5.1 Saving subscriber data

You save the data from subscriber administration using the *Save* button in the toolbar or with the *SINAUT / Save* menu. The *Options* dialog opens in which you specify the scope of the system data to be saved, processed and prepared.

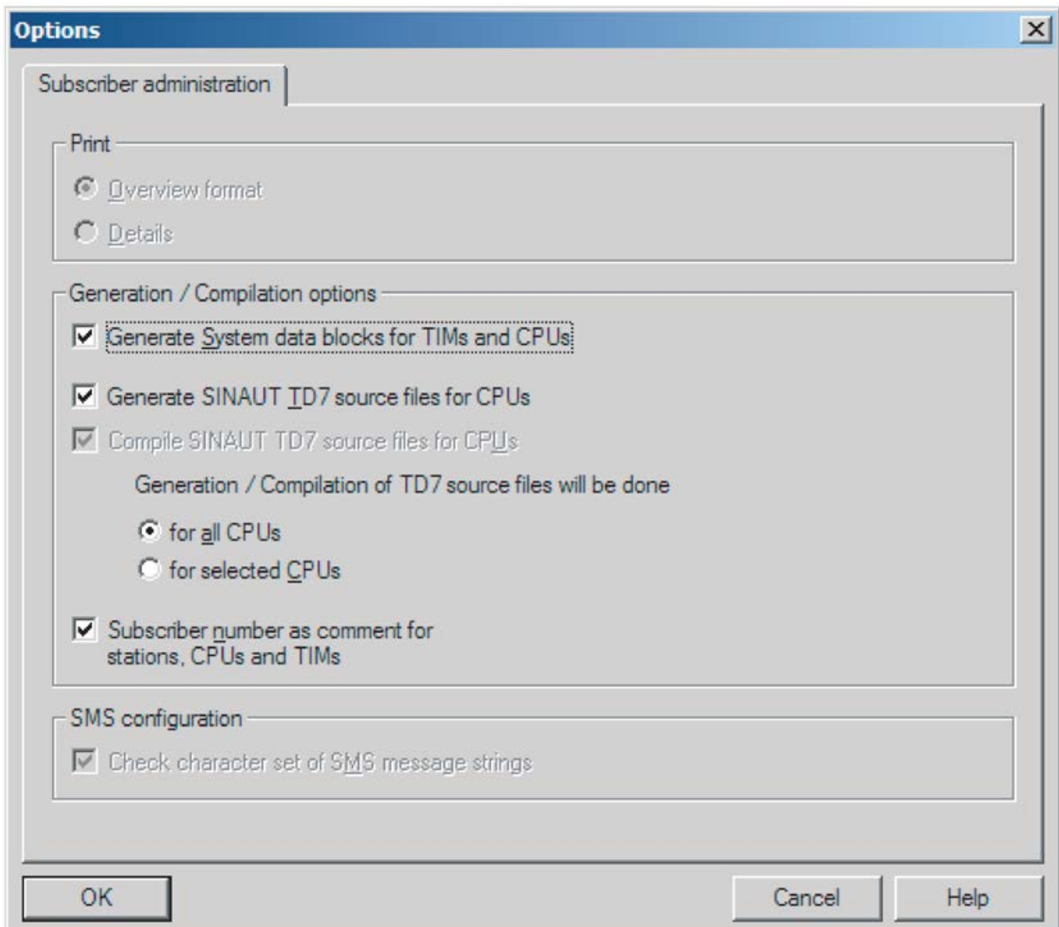


Figure 4-40 The *Options* dialog after selecting the Save function

The *Options* dialog provides the following convenient functions in the *Generation / Compilation options* box:

- Generate System data blocks for TIMs and CPUs
- Generate SINAUT TD7 source files for CPUs

The other options relate to the scope of system data generation

- for all CPUs or
- for selected CPUs (selected in the subscriber list)

The third option relates to generating the

- Subscriber number as a comment for stations, CPUs and TIMs

The convenient functions described below represent the automation of several operator control steps in the STEP 7 system. These functions always relate to all SINAUT networked subscribers; in other words, to subscribers involved in at least one SINAUT connection.

Regardless of the selected generation options, the internal data is saved in this case and a consistent version is always available later.

After saving the internal data, a consistency check determines whether the user data is free of errors. If this is the case, the required functions are executed.

If problems are detected during the consistency check, the convenient functions are *not* executed. An error list is displayed as well as a message indicating the functions that have not been executed.

As soon as the generation has been completed successfully, this is indicated by a status dialog.

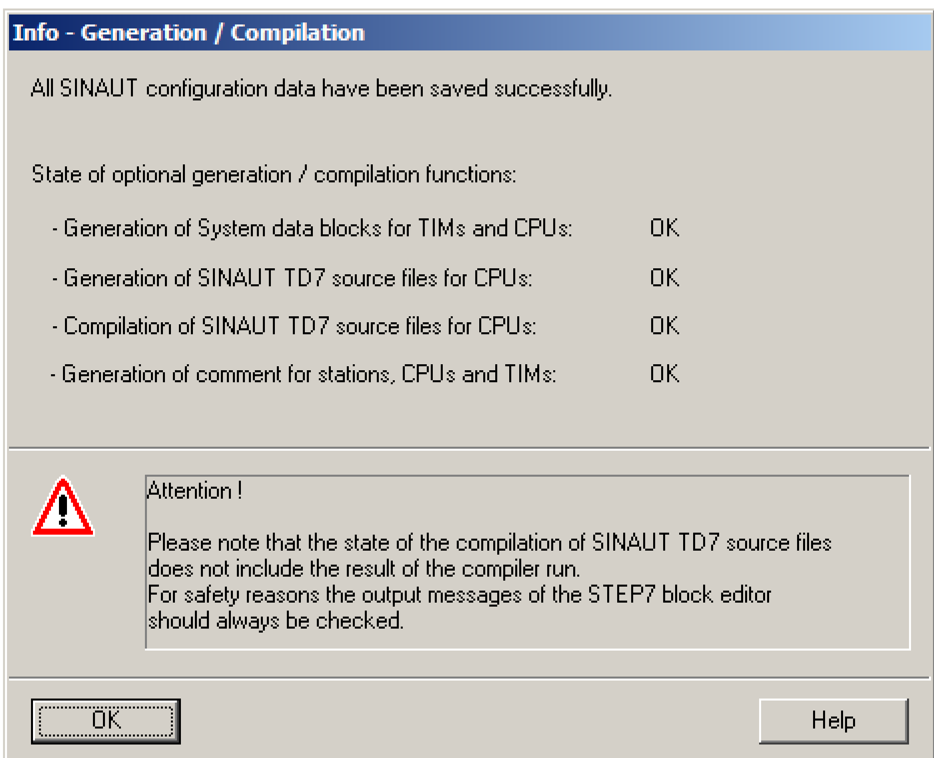


Figure 4-41 Status dialog after saving and generating the system data

4.5.2 Generating system data blocks

All the parameters of the TIM module from the hardware setting to information on communication partners or local connections are packed in system data blocks (SDBs). SDBs with numbers starting at 1000 are used.

If S7-homogeneous connections (communication block connections) are used for communication between TIM and TIM or between TIM and CPU, their data is packed in SDBs starting at no. 700.

If the *Generate System data blocks for TIMS and CPUs* option is selected in the *Options* dialog, this system data is saved for all the SINAUT-networked TIM and CPU modules in the subscriber list and saved in the offline data management.

Note

The SDBs must be transferred to the modules either in the SIMATIC Manager or using the SINAUT Diagnostics and Service tool.

4.5.3 Compiling SINAUT TD7 blocks for the CPU

The following information on the TD7 software relate only to the TD7 software for the CPU.

If when saving in the *Options* dialog, the option *Generate SINAUT TD7 source files for CPUs* is enabled, several generate functions are called up for all SINAUT networked CPU modules or only for the CPU modules selected in the subscriber list. These generate functions are as follows:

- Entry of the dynamic data for the SINAUT TD7 software in the form of data blocks in the source file and generation of symbolic entries for the communications DBs
- Entry of the non-existent SINAUT TD7 basic blocks in the source file or of blocks that exist in a version lower than/equal to the library version
- Entry of the CPU-specific SINAUT TD7 blocks in the source file according to the user selection (CPU-specific blocks)
- Adoption of all entries for SFCs or SFBs from the symbol table of the SINAUT TD7 library in all CPU modules involved if they do not already exist there.
- Comparison of the time stamp of the compiled blocks with those of the library blocks to avoid time stamp conflicts within the TD7 part.

The listed generate functions generate the corresponding SINAUT ST7 blocks as a source for and store the source file in the *Sources* directory of the relevant CPU. The source file generated in this way is then compiled so that the blocks are stored in the block directory.

Note

Only for systems with
- *a SINAUT TD7 library* lower than version 2.0
- *SINAUT configuration software* lower than version 2.0:

If you want to change the STEP 7 user interface to English (in the SIMATIC manager, menu *Options/Customize*, the option *Mnemonic* must not be changed to IEC! This must remain set to SIMATIC. Otherwise compilation errors will occur when compiling the SINAUT program!

Dynamic data

All the dynamic data required by the SINAUT TD7 software for the CPU is stored in the following data blocks:

- 1 DB BasicData
- n communications DBs
- m SMS data DBs.

The quantity and the numbers of the DBs are set by the system. The numbers of the communications DBs and the quantity and numbers of the SMS data DBs can be changed in the *DB configuration* or *SMS configuration* tabs of the *Properties of subscriber* dialog that can be opened with the shortcut menu *Properties* of the CPU modules in the subscriber list.

Basic blocks

When generating for a CPU, a check is made to establish whether all basic blocks exist that are required to run the SINAUT ST7 software. The basic blocks are:

Table 4- 11 Basic blocks of TD7onCPU

Block name	Block type	Default block number
SubscriberObject	UDT	UDT127
ConnectionDescription	UDT	UDT126
BCom	FB	FB127
XCom	FB	FB126
Pcom	FB	FB125
SMS_Ctrl *	FB	FB124
BasicTask	FC	FC127
Search	FC	FC126
Diagnose	FC	FC125
Distribute	FC	FC124
Create	FC	FC123
Startup	FC	FC122

* Only if SMS messages were configured.

For each of these blocks a check is made whether

- a symbol table entry exists for it and if yes
- whether the block exists in the user program

If no symbol table entry exists, a free block number for the relevant type is searched for, a corresponding entry is made in the symbol table and the block is also entered in the source file for the current CPU.

If a symbol table entry already exists, however the corresponding block does not exist or exists in a version lower than/equal to the library version, only the entry in the source file is made.

In both cases, the compilation produces an executable TD7 program in the block directory. Deselected sequence also guarantees that blocks that may be removed are always replaced again.

CPU-specific blocks

For all blocks required depending on the configuration, the user has the option of copying these automatically from the SINAUT master source file to the relevant CPU.

This function is controlled by the symbol table in the CPU. There, the user enters the required SINAUT TD7 blocks. If during generation it is established that

- a symbol table entry exists for a SINAUT block,
- this block, however, does not exist in the user program of the CPU,

this block is entered in the generated source file.

The user can use the symbol table more or less as a wish list. There is therefore no longer any need to put together the CPU program by copying source files or blocks.

The user can gain an overview of the blocks that can be generated in this way either in the SINAUT TD7 documentation, in the SINAUT TD master source file or in the symbol table of the SINAUT TD7 library.

Further user activities

The user now only needs to call the FC BasicTask in OB1 and call the FC Startup in OB100 to enable the basic functionality of SINAUT TD7.

Note

You will find more detailed information on this and on the parameter assignment of the data messages in the description of the software package *TD7onCPU*.

In addition to this, with the generate function, the user has the option of deleting all SINAUT blocks in the user program at any time and regenerating the source file to generate the SINAUT program completely with the block numbers stored in the symbol table.

4.5.4 Creating SINAUT subscriber numbers as comments

To make the subscriber numbers that are important for SINAUT communication visible in the SIMATIC Manager or in the hardware configuration, subscriber administration allows you to enter the subscriber number of SINAUT networked components in the comment field of the properties.

The function is activated by selecting the option *Subscriber number as comment for stations, CPUs and TIMs* in the *Options* dialog when you call the *Save* function in the *Options* dialog.

With the following SINAUT networked subscribers, the subscriber number is entered in the comment bar of the SIMATIC Manager:

- Stations:
- CPU modules
- TIM modules
- Third-party stations

The comments are visible in the SIMATIC Manager when you select the "*View / Details*" menu. The TIM modules are visible after expanding the tree structure and selecting a station.

Note

Creating the subscriber number when generating the system data overwrites comments previously entered in the network configuration without any possibility of restoring them.

4.5.5 Consistency check

The consistency check is always started automatically before you use the generate/compile functions in subscriber administration to prevent SDBs or DBs being created with inconsistent data.

The consistency check can also be started as a separate function using the *SINAUT / Check consistency...* menu in subscriber administration.

Errors detected during the consistency check are displayed to the user in an error list.

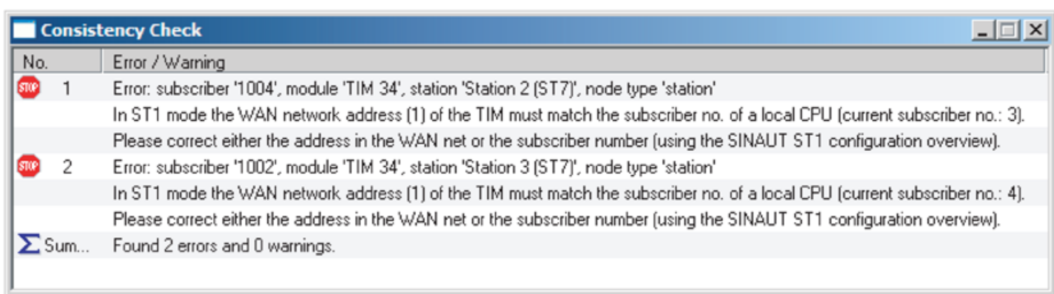


Figure 4-42 Example of an error list after running the consistency check

If inconsistent connections are found, the error list indicates that cause will be diagnosed in the connection configuration in the *Invalid connections* dialog.

After eliminating an error, you must save prior to the next consistency check otherwise be eliminated error will still be reported.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

4.6 Change matrix

The change matrix describes the necessary follow-up activities of the user following typical actions in the SINAUT configuration tool.

Table 4- 12 Change matrix

Object affected	Operator activity in the SINAUT configuration tool	Necessary follow-up action
Station	Adding a station	-
Station	Renaming a station	-
Station	Changing the parameter assignment of a station	-
Station	Deleting a station	All SINAUT connections running over a module in this station are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
TIM module	Adding a TIM module	-
TIM module	Renaming a TIM module	-
TIM module	Changing the parameter assignment of a TIM module	If the parameters of a TIM module are changed, the SDBs must be regenerated in subscriber administration only for this TIM. Exception: If parameters are changed in the "WAN access" tab, this affects all SINAUT connections running over the modified WAN driver.
TIM module	Deleting a TIM module	All SINAUT connections running over this module are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
Network	Adding a network	-
Network	Renaming a network	-
Network	Changing the parameter assignment of a network	The SDBs of all the modules connected to this network must be regenerated in the subscriber administration. In WAN networks, these are only TIM modules, in LANs all connected modules.
Network	Deleting a network	All SINAUT connections running over this network are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
Network nodes	Adding a network node	-
Network nodes	Renaming a network node	-
Network nodes	Changing the parameter assignment of a network node	The SDBs of the module containing this network node must be regenerated. Exception: When changing the telephone number, the SDBs of all TIMs in the dial-up network must be regenerated.
Network nodes	Networking a network node	-

Object affected	Operator activity in the SINAUT configuration tool	Necessary follow-up action
Network nodes	Deleting a network node	All SINAUT connections running over this network node are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
Network nodes	Canceling the networking of a network node	All SINAUT connections running over this network node are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
SINAUT connection	Adding a SINAUT connection	The SDBs or DBs of all subscribers over which this connection runs must be regenerated in the subscriber administration.
SINAUT connection	Deleting a SINAUT connection	The SDBs or DBs of all subscribers over which the deleted connection runs must be regenerated in the subscriber administration.
SINAUT subscriber	Changing the parameter assignment of a SINAUT subscriber	The SDBs or DBs of all connections that run over this subscriber must be regenerated in the subscriber administration.
Destination subscriber properties of TD7onTIM	Changing the parameter assignment of a destination subscriber of TD7onTIM	The SDBs of all TIMs with TD7onTIM that communicate with this destination subscriber must be regenerated in subscriber administration.

4.7 Configuration practice

4.7.1 Downloading data blocks to the CPU

To download data blocks to the CPU module, you use the STEP 7 SIMATIC Manager standard tool. This allows you to copy blocks using drag-and-drop or a menu either in an online window (configured online access) or in the window with the accessible nodes (non-configured online access).

WARNING

When you download blocks to the automation system by dragging and dropping, you yourself are responsible for ensuring that the blocks are copied to the correct online object (in other words, the object with the correct MPI address). The STEP 7 tool does not check this.

For more detailed information on these activities, refer to the online help of the SIMATIC Manager.

4.7.2 Downloading system data blocks to the TIM

You should only download system data blocks (SDBs) to the TIM in the SIMATIC Manager or in the SINAUT Diagnostics and Service tool.

Note

When downloading system data blocks in *hardware configuration*, make sure that *no* connection SDBs (SDB7xx) are downloaded. If SDBs of this type need to be downloaded to the TIM module, you must use the SIMATIC Manager or the SINAUT Diagnostics and Service tool.

SDBs can also be downloaded in *network configuration*. Creating SDBs during network configuration is a different procedure from that in the SIMATIC Manager and hardware configuration and is not suitable for the TIM module. Copying SDBs to TIM modules should therefore only be done in the SIMATIC Manager or SINAUT Diagnostics and Service tool.

In the SIMATIC Manager, all SDBs of a module are indicated by a symbol with the name *System data*. This means that you can only ever manipulate all SDBs of a module as a single unit. Otherwise, the same applies as for data blocks.

In the hardware configuration, it is possible to download the SDBs of individual modules or entire stations.

In both cases, the function is followed by a dialog in which you are asked whether you want to restart the TIM module. This dialog must be exited with *Yes* to restart the TIM and activate the new SDBs.

4.7.3 Uploading stations with the Upload Station to PG function

The STEP 7 function *Upload PLC/Station to PG* allows the configuration of a connected station to be adopted. In conjunction with TIM modules, this function can only be used with certain restrictions.

- If the station to be uploaded is a TIM rack; in other words if the rack contains only stand-alone TIM modules, it is not possible to upload the station. The *Upload Station to PG* function can only be used in racks with CPU modules.
- If there is a 300 series CPU in the rack, the configuration can be uploaded, the TIM module represented in the rack is, however, not fully initialized and is not suitable for further configuration. This must be replaced by a new module from the hardware catalog.

WARNING

If the TIM modules uploaded in this way are further configured, problems can arise particularly in SINAUT communication and when handling the relevant module.

4.7.4 Changing the MPI address of the CPU

In hardware configuration, it is possible to change the MPI address of the CPU. If TIM modules are installed in the same rack, when downloading the SDBs ,**it is necessary** that the download is performed in two steps.

1. Download SDBs only to the CPU not to the TIM. Once the CPU module has received its new MPI address, the TIM modules go through a reset.
2. Download the SDBs to the TIM modules when they have completed the restart.

4.7.5 Copying and reorganizing STEP 7 projects

Copying projects

In the SIMATIC Manager, you can copy entire projects by saving them under a different name ("File" > "Save As..." menu). In the following dialog "Save project as", you enter the name and storage path of the new project.

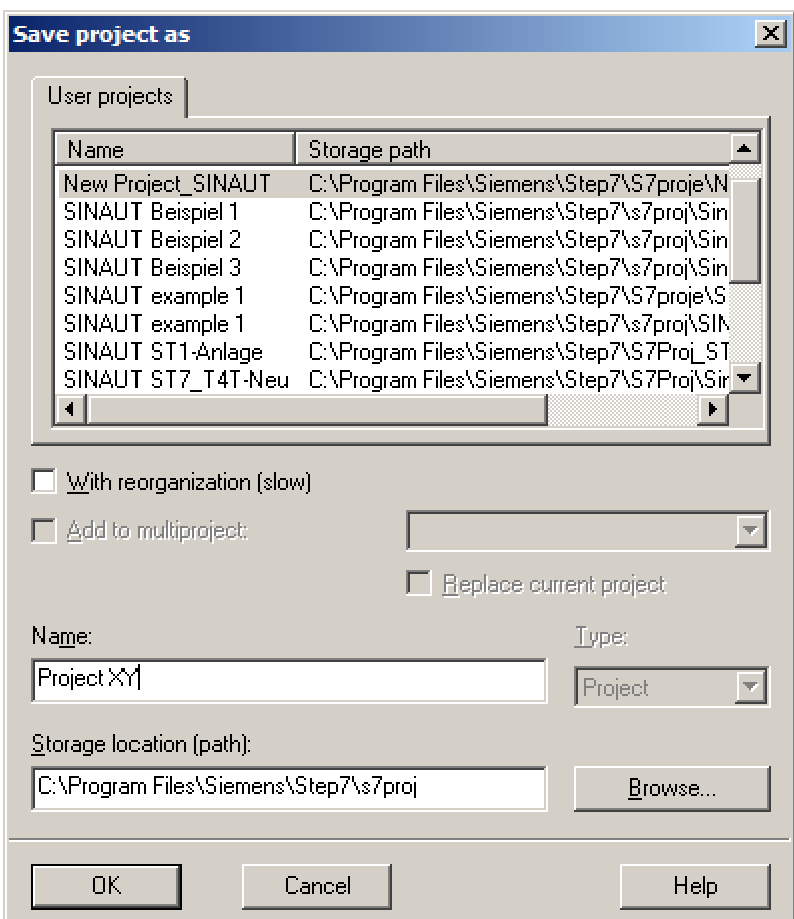


Figure 4-43 "Save project as" dialog for copying objects

Reorganizing projects

When you save, you can also select the "With reorganization" copy option.

Note

Loss of configuration data when reorganizing

When copying projects with reorganization, the following data is not copied:

- SINAUT subscriber numbers
- ST7 connection data

This data must be reconfigured following the copy function.

Swapping over the interfaces of the TIM 4R-IE during reorganization

When you copy projects with reorganization, the two Ethernet interfaces of the TIM4R-IE are swapped over. These must be reconfigured following the copy function.

If you copy projects without reorganization, the SINAUT data is also copied. This variant does not have any disadvantages.

SINAUT users should normally use the "without reorganization" variant.

4.7.6 Avoiding time stamp conflicts

Each block has an interface time stamp that provides information about when the interface visible to other blocks was last changed.

These time stamps are compared by the block editor when a block is opened. If inconsistencies are detected, the inconsistent calls are opened up; in other words, instead of a CALL command, the user finds the code generated by the system as a substitute for the CALL. This status is reported as a *time stamp conflict*. The substitute code must then be deleted by the user and the CALL parameters set again.

Inconsistencies of this type can occur between:

- OB and FBs / FCs
- FBs /FCs and FBs /FCs
- FBs and their instance DBs

Recommended procedure

- When using the configuration tool, this makes sure that the interface time stamp of all blocks regenerated by the tool (including instance data blocks) are synchronized with the interface time stamps of the blocks in the TD7 library.
- The time stamp of the block interfaces (supplied version / updates) are frozen until there is an actual change to an interface; in other words changes or corrections in the program code do not affect the interface time stamp.

- This procedure means that an update is possible at any time without causing a time stamp conflict.
- Directly inserting blocks by dragging them from the SINAUT library is permitted as long as the standard ST7 block number range (compare symbol table of the library) is not changed. This procedure is recommended

Not recommended procedure

- If the *TD7_UserSource* provided by the tool is compiled manually into a CPU block directory in which there are not yet any SINAUT TD7 blocks or in which some are missing, these blocks are given an interface time stamp that differs from that in the SINAUT ST7 library.
- Any SINAUT ST7 user programs already stored there then develop time stamp conflicts or they occur at the latest after the first update.
- Since SINAUT ST7 support is oriented mainly on the interface time stamp of the blocks due to version maintenance and management of upgrades, this procedure is generally not advisable.

Restrictions

The time stamps of the blocks that call SINAUT blocks are not synchronized. Here, time stamp conflicts can continue to occur.

The following sequence is therefore advisable for the user:

1. Before starting generation in the SINAUT ST7 configuration tool, a source file with all blocks that call SINAUT blocks is created.
2. The generation is run.
3. The previously generated source file is compiled.

SINAUT TD7onCPU software package 5

5.1 Overview

Introduction

For SINAUT ST7 there is the software package SINAUT TD7 for transferring process data between SINAUT subscribers via WAN, MPI bus and Ethernet. The software package exists in two versions:

- The software package *SINAUT TD7 for the CPU*, known below as TD7onCPU is a software package that is assigned parameters on the CPU and runs on the CPU. It is used in all SINAUT stations in which TIM modules of the type TIM 3 or TIM 4 are configured.
- The software package *SINAUT TD7 for the TIM*, known below as TD7onTIM is a software package that is configured on Ethernet TIMs (e.g. TIM 3V-IE) and also runs there. It can be used as an alternative to the software package TD7onCPU when an Ethernet TIM is configured in a SINAUT station.
  TD7onTIM is described in the section *Configuration software for SINAUT ST7*.

In this section only the software package TD7onCPU is described.

The TD7 software package for the CPU contains specific blocks for CPU modules. The package was designed so that it can run both on an S7-400 and on and S7-300 CPU. Exceptions to this rule will be pointed out explicitly.

With the aid of the SINAUT TD7 software users create a program for their CPUs. This program allows change-driven transmission of process data between the individual CPUs and the control center, e.g. 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.

Apart from process data transfer via WAN, the package is also suitable for local communication between CPUs, if they are connected via MPI. Here, as well, local connections and CPUs are constantly monitored and there is an automatic data update after starting up od eliminating disturbances.

Note

Data communication from CPU to CPU can only be implemented via a WAN connection with the aid of the SINAUT TD7 software. This is not possible with the S7 communication SFBs/SFCs for configured or unconfigured connections. These are suitable only for local communication without a gateway.

The content of the SINAUT TD7 software package

The essential components of the package are as follows:

- Basic and auxiliary blocks
  Most of these blocks are always required on the CPU. A few can be installed as options. The basic blocks handle central tasks such as startup, monitoring of connections and reachability of connection partners, general requests, time management, handling communication etc. The auxiliary blocks enter messages in the send buffer or fetch them from the receive buffer, handle send and receive jobs for specific connections or provide information as a result of searches, etc.
- Data point typicals
  These blocks are included in the CPU program depending on the type and amount of data to be transferred. They put messages together when data changes or on request and output received process data.

Blocks that contribute to the SINAUT ST7 configuration tool

To operate correctly, the TD7 package requires some data blocks per CPU. These DBs are generated automatically by the SINAUT ST7 configuration tool during configuration of the SINAUT connections and stored in the program directory of the relevant CPU. These are as follows:

- Central records DB
  This contains all the centrally required data including the records of all communication partners and the connections to be managed.
- Communication DBs
  A separate communication DB is created for each connection. This DB contains a send and receive buffer and all the data required for controlling and monitoring the connection.

Along with the data blocks named above, the SINAUT ST7 configuration tool also stores the basic and auxiliary blocks that are absolutely necessary for a program to be able to run in the CPU program directory. The users only need to copy the data point typicals they require from the SINAUT TD7 library to the CPU program directory and can then create the SINAUT user program.

Note

The blocks stored by the SINAUT ST7 configuration tool in the CPU program directory exist there once as individual blocks in the "Blocks" directory and in the "Sources" directory as an STL source file with the name 'TD7_UserSource'.

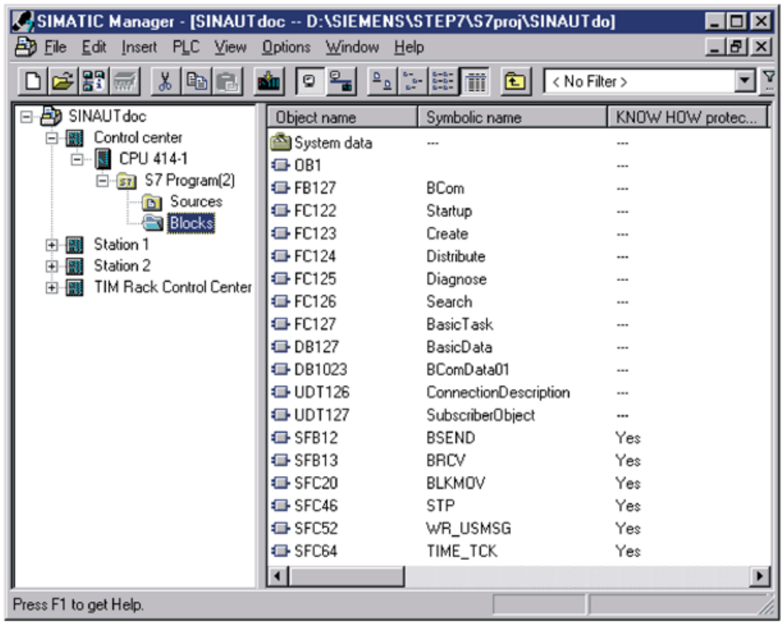


Figure 5-1 Basic and auxiliary blocks in the Blocks program directory

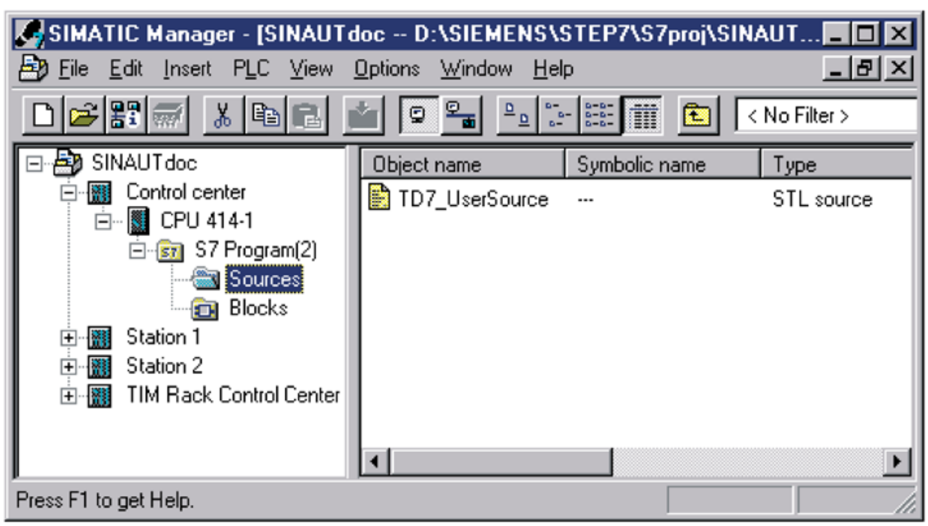


Figure 5-2 TD7_UserSource in the Sources program directory

Note

If you want to change the STEP 7 user interface to English (in the SIMATIC Manager, menu Options/Customize) the following apples depending on the version of the SINAUT configuration tool:

- SINAUT configuration tool version < 2.0
  The option *Mnemonic* must not be changed to IEC or English! This must remain set to SIMATIC. Otherwise compilation errors will occur when compiling the SINAUT program!
- SINAUT configuration tool version 2.0
  The option *Mnemonic* may be changed to IEC or English! The configuration tool checks the Mnemonic set and then ensures correct generation of the SINAUT program.

Basic structure of the user program

The SINAUT user program consists of the following:

- Startup program OB100
  Here only the ST7 startup block FC-Startup needs to be called. It does not require any parameters.
- Cyclic program OB1
  In the simplest case this program consists of the basic block FC-BasicTask. Then the users call up all their data point typicals they require on the CPU and assigns parameters to these according to their requirements.
- Time-driven program OB35 (or another cyclic interrupt OB)
  This program is only needed when count pulses need to be detected on the CPU. In the cyclic interrupt the FC-PulseCounter is called once or several times.

5.1.1 SINAUT TD7 Library

Introduction

When the SINAUT software is installed a library is also created with the SINAUT TD7 software package. As with all libraries available with STEP 7 this is accessible, i.e. from the SIMATIC Manager a list of all available libraries can be displayed using the menu *Open* in the *Libraries* tab. The SINAUT library can be found there under the name *SINAUT TD7 Library*.

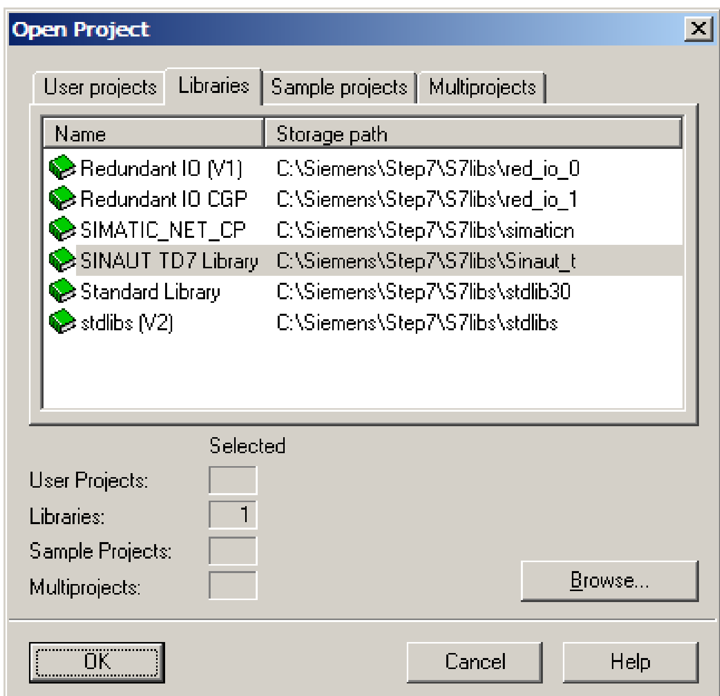


Figure 5-3 Opening the SINAUT library *SINAUT TD7 Library*

Composition of the library

In the *SINAUT TD7 Library* there is the directory *Basic01* that is made up of the following sections:

- Source
  Contains the two STL source files Basic01_Source_de (blocks with German mnemonics) and Basic01_Source_en (blocks with English mnemonics). These contain all blocks of the TD7 library in STL source format.
- Blocks
  Contains all blocks of the TD7 library in block format. Here you will find all SINAUT blocks of the type FB, FC, DB, UDT and VAT and the SIMATIC system blocks SFC and SFB that are used by the SINAUT software.
- Symbols
  Contains a symbol table of the SINAUT TD7 library.

Note

In the SINAUT TD7 library never change the content *Source*, *Blocks* or *Symbols*! Renaming or moving the library to a different directory is also not permitted.

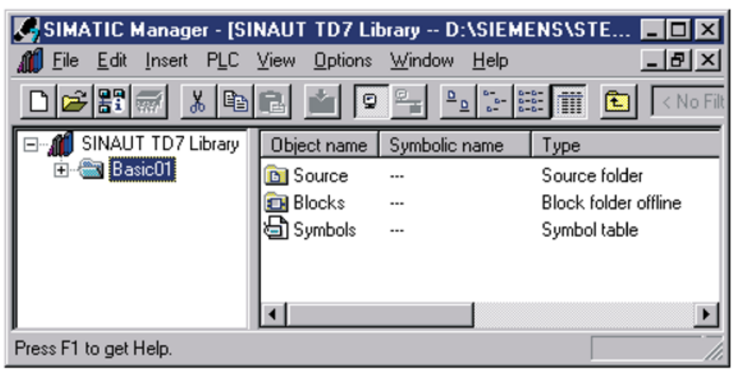


Figure 5-4 Composition of the SINAUT TD7 library

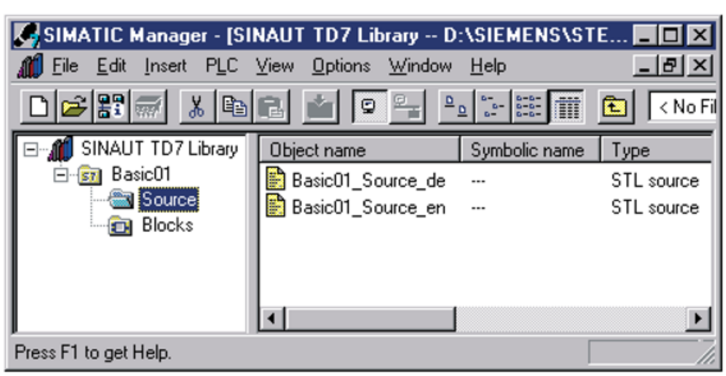


Figure 5-5 Source - STL source files *Basic01_Source_de/_en* in the SINAUT TD7 library

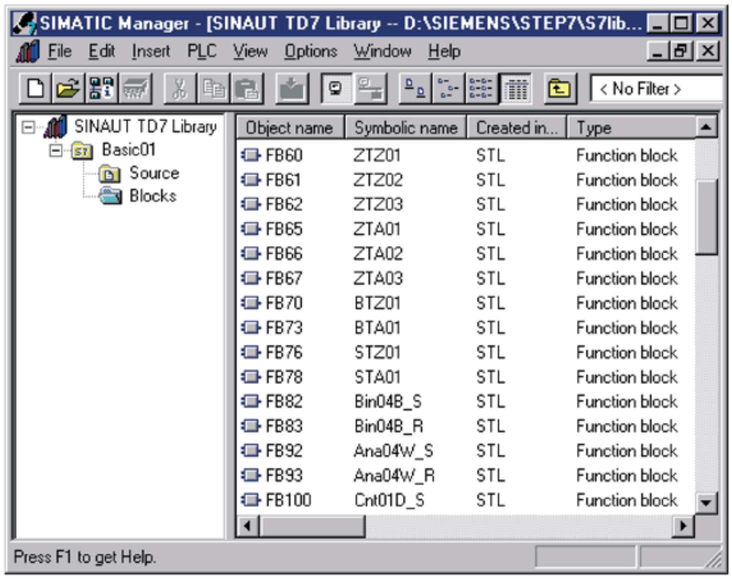


Figure 5-6 Blocks - blocks of the SINAUT TD7 Library in block format

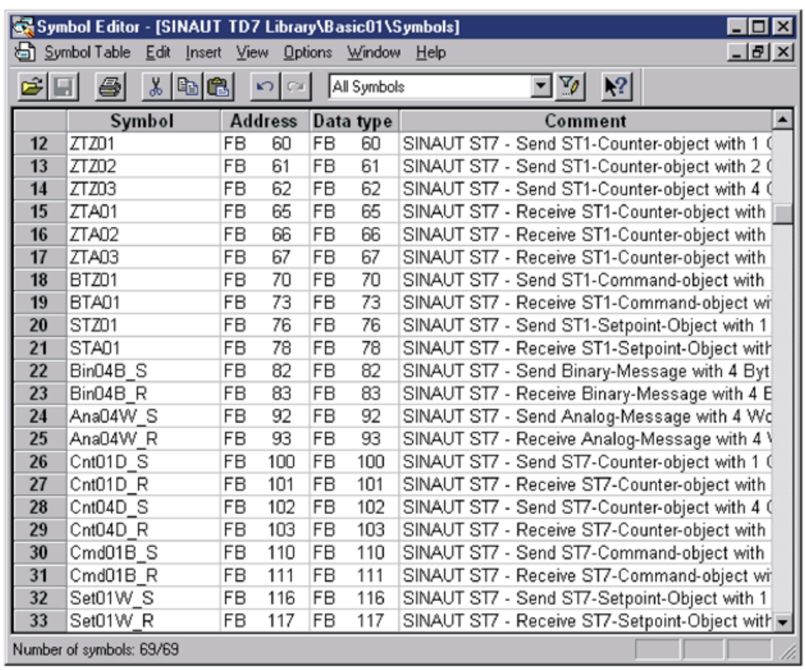


Figure 5-7 The symbols table *Symbols* in the SINAUT TD7 library

Note

If you want to change the numbers of individual SINAUT blocks to avoid a collision with blocks already being used in the user program, a symbol table needs to be created for the CPU module(s) involved that lists all the SINAUT blocks required per CPU. This includes not only the blocks whose number will be changed in the symbol table but also those whose number remains unchanged. You will find further information in the section *Changing SINAUT block numbers*.

5.1.2 Block overview

Introduction

The following table lists all the blocks contained in the TD7 library. For each block, the table provides:

- The standard block number under which the lock is available in the TD7 library. This number can be changed when necessary.
- The symbolic name of the block. You can call the relevant block in the user program using this name.
- A note indicating whether this block is an auxiliary block that is called indirectly. This is important in case the standard number of the auxiliary block needs to be changed.
- A list of the auxiliary blocks required by the relevant block. Only the SINAUT auxiliary blocks are specified since only these blocks need to be transferred from the TD7 library into the user program directory (normally performed automatically by the ST7 configuration tool). The information on the auxiliary blocks is also important if one or more of the listed SINAUT auxiliary blocks are given a different block number. The block must then be recompiled.
- A brief explanation of the function of the block.

You will find the blocks for SINAUT ST1 in release 05/2007 of the manual.

Table 5- 1 SINAUT TD7 Library: Block overview

Block no.	Symbolic name	Aux. block	Required (SINAUT) auxiliary blocks	Explanation
Function blocks FB				
FB82	Bin04B_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 status message object, send 4 bytes of status/binary information.
FB83	Bin04B_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 status message object, receive 4 bytes of status/binary information.
FB84	Dat12D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 data object, send max. 12 double words with any information.
FB85	Dat12D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 data object, receive max. 12 double words with any information.
FB86	Dat256D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 data object, send max. 256 double words with any information.
FB87	Dat256D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 data object, receive max. 256 double words with any information.
FB92	Ana04W_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 analog value object, send 4 analog values (16-bit value in the INT format).
FB93	Ana04W_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 analog value object, receive 4 analog values (16-bit value in the INT format).
FB100	Cnt01D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 counted value object, send 1 counted value (32-bit ST1 format).

Block no.	Symbolic name	Aux. block	Required (SINAUT) auxiliary blocks	Explanation
FB101	Cnt01D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 counted value object, receive 1 counted value (32-bit ST1 format).
FB102	Cnt04D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 counted value object, send 4 counted values (32-bit ST1 format).
FB103	Cnt04D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 counted value object, receive 4 counted values (32-bit ST1 format).
FB110	Cmd01B_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 command object, send 1 byte commands (1-out-of-8 ST1 format).
FB111	Cmd01B_R	-	DB127 BasicData, FC126 Search	ST7 command object, receive 1 byte commands (1-out-of-8 ST1 format).
FB116	Set01W_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 setpoint object, send 1 setpoint (16 bits) and receive current local setpoint.
FB118	Par12D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 parameter object, send max. 12 double words with parameters and receive current local parameters.
FB119	Par12D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 parameter object, receive max. 12 double words with parameters and send current local parameters.
FB124	SMS_Control	-	DB127 BasicData, FC123 Create, FC125 Diagnose, FC126 Search	Block for sending SMS messages.
FB125	PCom	Yes	DB127 BasicData, FC124 Distribute, FC125 Diagnose, FC126 Search	Block for communication over a peripheral bus connection (SFCs WR_REC and RD_REC are used).
FB126	XCom	Yes	DB127 BasicData, FC124 Distribute, FC125 Diagnose, FC126 Search	Communication block for an unconfigured X connection (SFCs X_SEND and X_RCV are used).
FB127	BCom	Yes	DB127 BasicData, FC124 Distribute, FC125 Diagnose, FC126 Search	Communication block for a configured communication block connection (SFBs BSEND and BRCV are used).
Functions FC				
FC113	PartnerMonitor	-	DB127 BasicData, FC125 Diagnose, FC126 Search	Allows the display of important status information and control for a SINAUT subscriber.
FC114	Trigger	-	DB127 BasicData, FC125 Diagnose	Sets an output at a defined time or at a defined interval.
FC115	PartnerStatus	-	DB127 BasicData, FC125 Diagnose, FC126 Search	Displays the connection status for up to 8 SINAUT subscribers.
FC116	Safe	-	DB127 BasicData	Block for saving command and setpoint input.
FC117	PulseCounter	-	DB127 BasicData	Block for acquiring up to 8 counter inputs
FC118	TestCopy	-	DB127 BasicData	Test block for logging received and/or sent messages.
FC119	ListGenerator300	-	DB127 BasicData	Block for generating an object list. Version for S7-300.

Block no.	Symbolic name	Aux. block	Required (SINAUT) auxiliary blocks	Explanation
FC120	ListGenerator400	-	DB127 BasicData	Block for generating an object list. Version for S7-400.
FC121	TimeTask	-	DB127 BasicData, FC125 Diagnose	Block for keeping the date and time on a CPU
FC122	Startup	-	DB127 BasicData	Startup block.
FC123	Create	Yes	DB127 BasicData, FC125 Diagnose, FC126 Search	Block for creating messages and entering them in the send buffer.
FC124	Distribute	Yes	DB127 BasicData, FC125 Diagnose, FC126 Search	Block for distributing received messages to the local destination objects.
FC125	Diagnose	Yes	DB127 BasicData	Block for entering system messages in the diagnostic buffer of the CPU.
FC126	Search	Yes	-	Block for handling searches.
FC127	BasicTask	-	DB127 BasicData, FC123 Create, FC125 Diagnose, FB125 PCom, FB126 XCom, FB127 BCom	Block for handling all SINAUT basic tasks on the CPU.
Data blocks DB				
DB99	TestCopyData	-	-	Data block for the test block FC TestCopy.
DB125	SMS_Data	-	UDT125 ShortMessageObject	Data block for FB SMS_Control for entry, for example of SMS texts.
DB127	BasicData	Yes	UDT126 ConnectionDescription, UDT127 SubscriberObject	Data block for entry of SINAUT basic information.
User-defined data types UDT				
UDT125	ShortMessage object	Yes	-	SMS object (for DB SMS_Data).
UDT126	Connection description	Yes	-	Connection description (for DB BasicData).
UDT127	SubscriberObject	Yes	-	Subscriber object (for DB BasicData).
Variable table VAT				
VAT99	VAT_TestCopy	-	-	Variable table for the test block FC TestCopy.
System function blocks SFB				
SFB12	BSEND	Yes	-	Block-oriented sending of data over a configured connection.
SFB13	BRCV	Yes	-	Block-oriented reception of data over a configured connection.
System functions SFC				
SFC0	SET_CLK	Yes	-	Set CPU clock.
SFC1	READ_CLK	Yes	-	Read CPU clock.
SFC20	BLKMOV	Yes	-	Copy variables.
SFC22	CREAT_DB	Yes	-	Create data block.
SFC23	DEL_DB	Yes	-	Delete data block.
SFC24	TEST_DB	Yes	-	Test data block.
SFC25	COMPRESS	Yes	-	Compress user memory.

Block no.	Symbolic name	Aux. block	Required (SINAUT) auxiliary blocks	Explanation
SFC46	STP	Yes	-	Set CPU to STOP.
SFC52	WR_USMSG	Yes	-	Write user diagnostic message to the diagnostic buffer.
SFC64	TIME_TCK	Yes	-	Read system time.
SFC65	X_SEND	Yes	-	Send data over an unconfigured con-nection.
SFC66	X_RCV	Yes	-	Receive data over an unconfigured connection.

5.1.3 Changing SINAUT block numbers

Introduction

The blocks in the SINAUT library have fixed block numbers that you should, whenever possible, use with these numbers in your user program. This applies to all SFB an SFC system blocks whose numbers cannot be modified. The numbers of SINAUT FBs, FCs, DBs and UDTs can, however, be adapted when necessary. The effort required for the changes varies from case to case. The three following modification stages can be distinguished:

1. Only numbers of SINAUT blocks are changed that do not belong to the category of auxiliary blocks (there is a dash in the 'Aux. block' column in the table),
   and
   The standard numbers of the SINAUT auxiliary blocks were not changed in the relevant project.
2. Only the numbers of the SINAUT UDTs are changed.
3. One or more SINAUT auxiliary blocks will be given a different number (in the table, there is a 'yes' in the 'Aux. block' column),
   or
   you want to copy for the SINAUT blocks into a project in which numbers of SINAUT auxiliary blocks have already been changed (it does not matter whether these are auxiliary blocks or not).

Depending on the change level, you should follow the steps outlined below:

Note

With each change to a block number made by the user, make sure that the number change is also entered in the symbol table of the CPU. Otherwise, errors will occur when you save in the 'Subscriber Administration' SINAUT tool.

Changing the numbers of blocks other than auxiliary blocks

The prerequisite for using the instructions here is as follows:

- You only want to change numbers of SINAUT blocks that do not belong to the auxiliary block category,
  and
- The standard numbers of the SINAUT auxiliary blocks have not been changed in the relevant project.

Initial situation

You have configured your SINAUT installation as already described in the section 'Configuration software for SINAUT ST7'; in other words, you have completed the connection configuration with the appropriate SINAUT configuration tool and have started to save in the 'Subscriber Administration' SINAUT tool. The result is that all the necessary SINAUT blocks already exist in all the program directories of the CPU in addition to the SDB directory and OB1 as shown in the following figure.

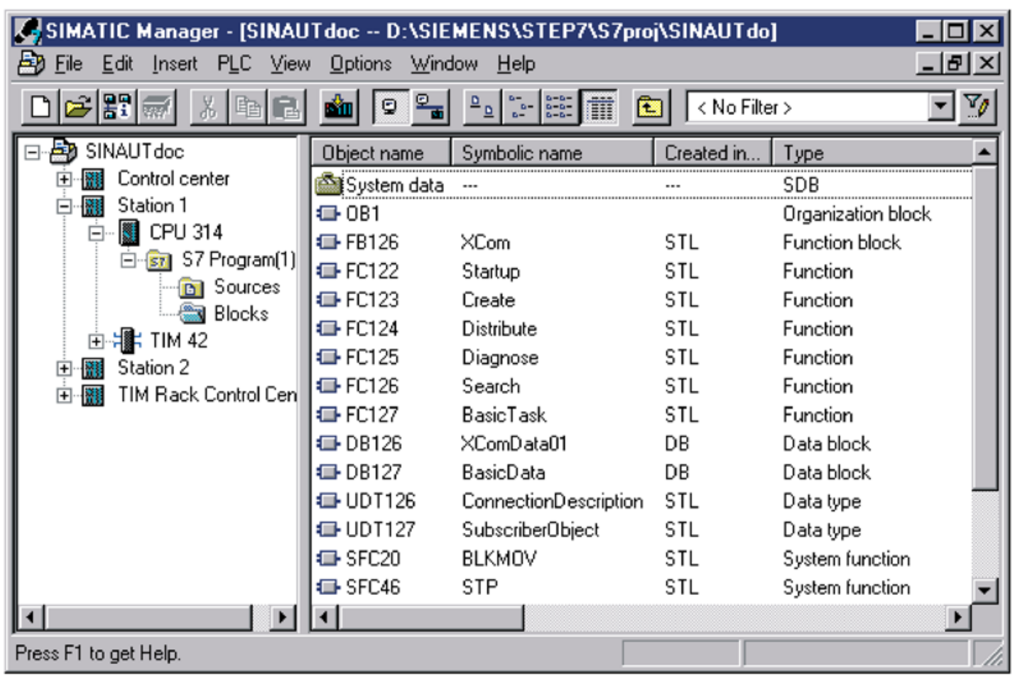


Figure 5-8 Example of a project (SINAUT basic blocks already present)

Making the change

Copy the additional blocks that you require directly from the TD7 library to the program directory of the relevant CPU. If the number of one of the copied blocks is identical to an existing block, a dialog is displayed automatically indicating the number conflict. You will be asked whether or not you want to remain the block. After clicking on *Rename...*, a further dialog appears in which you can enter the new number, for example FB82 is renamed to FB8 as shown in the following figure.

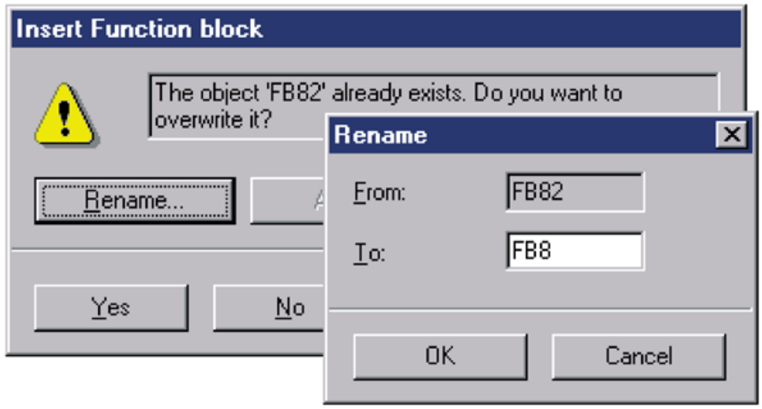


Figure 5-9 Renaming blocks in the *Find* dialog

If the block number is changed using this dialog, the number change is automatically included in the symbol table. This can be recognized because the renamed block in the block directory is displayed immediately with the corresponding name from the TD7 library.

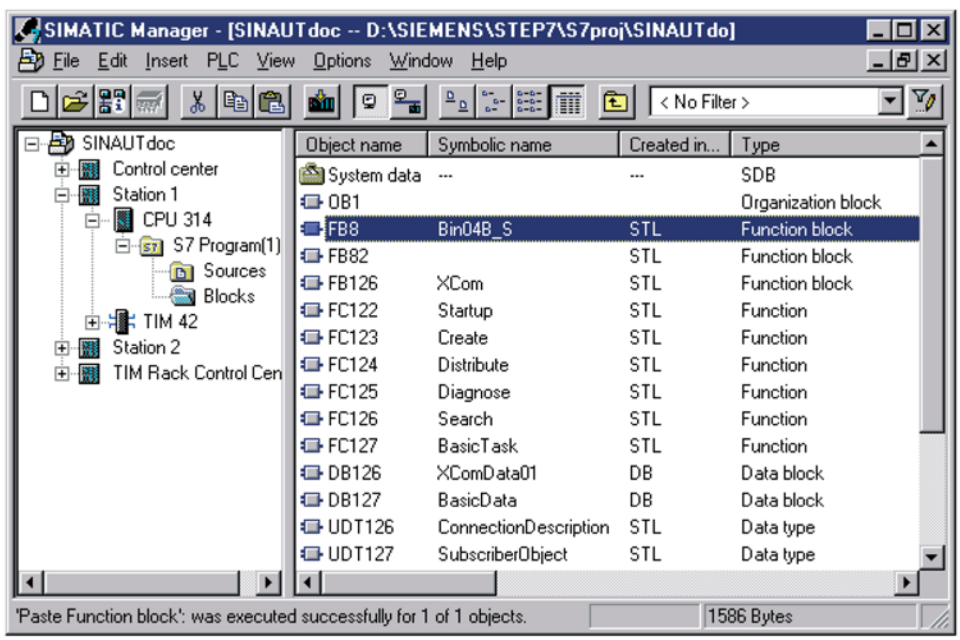


Figure 5-10 Renaming with automatic symbol assignment

If you copy several blocks at one time, the dialog is displayed for each block whose number is already being used. You can therefore copy all the required blocks per CPU at one time and then adapt the block numbers and entries in the symbol table as necessary.

If the dialog does not appear when you copy blocks, there is no conflict with existing blocks. If you nevertheless want to give blocks a different number, you must change these numbers directly in the block directory and also adapt the relevant entries in the symbol table (although the symbolic name of a SINAUT block was entered automatically in the symbol table of the destination CPU when the blocks were copied from the TD7 library, the new numbers are not taken into account since they were not changed initially during copying).

If you want to change one of the blocks already stored by the SINAUT tool (here, this can only involve the blocks FC122 Startup and FC127 BasicTask because all other blocks belong to the auxiliary block category), the numbers of these blocks can be changed directly in the block directory. Do not forget, however, to make the same number change in the symbol table of the CPU.

Note

Never use this method to change the numbers of the SINAUT data blocks. These are the data blocks with the symbolic names *BasicData*, *XComData01*, *XComData02* etc., *PComData01*, *PComData02* etc. and *BComData01*, *BComData02* etc.

If you want to make the same changes for more than one CPU, the most efficient method is to make the changes first in the block directory and the symbol table of one CPU. You can then copy the modified blocks from the block directory of the finished CPU to the other CPUs and then do the same for the symbol tables.

Changing the numbers of UDTs

The prerequisite for using the instructions here is as follows:

- You only want to change the numbers of SINAUT UDTs.

Initial situation

You have completed the configuration of your SINAUT installation including the connection configuration with the appropriate SINAUT configuration tool. You have not yet saved in the 'Subscriber Administration' SINAUT tool. The result is that there are not yet any SINAUT blocks in the program directories of the CPUs. In the block directory of the CPU, for example, there is only the system data and OB1 as shown in the following figure.

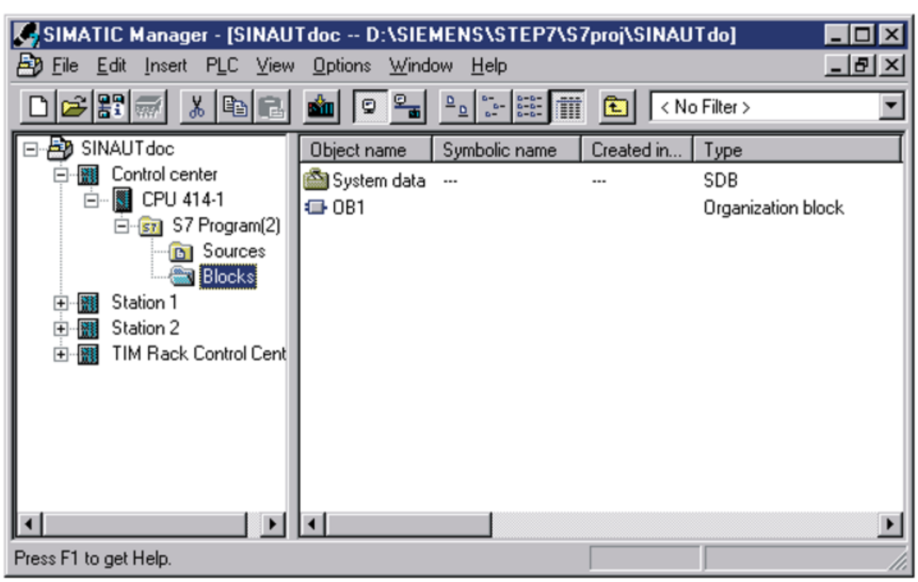


Figure 5-11 Example of a project (still without user or SINAUT program)

Making the change

The change to the UDT numbers is made indirectly using the symbol table of the CPU with the aid of the 'Subscriber Administration' SINAUT configuration tool.

Step 1

- Copy the UDTs that you want to change in your program from the symbol table of the TD7 library to the symbol table of the first CPU.
- Change the numbers of the SINAUT UDTs to the numbers you require in the symbol table.
- Save the symbol table.
- Then copy the rows with the changed UDT numbers to the symbol tables of all other CPUs of your project. Do not forget to save all the modified symbol tables.

Step 2

- Once all the symbol tables have the required values and have been saved, change to the 'Subscriber Administration' SINAUT ST7 Configuration Tool.
- Click the 'Save' button. An extra dialog 'Properties' opens (see following figure).
- Make sure that a check mark is entered in front of the following options in this dialog:
  - 'System data blocks for TIMS and CPUs'
    and
  - 'SINAUT TD7 blocks for CPUs'
  - and a dot in front of 'for all CPUs'. Then click on 'OK'.

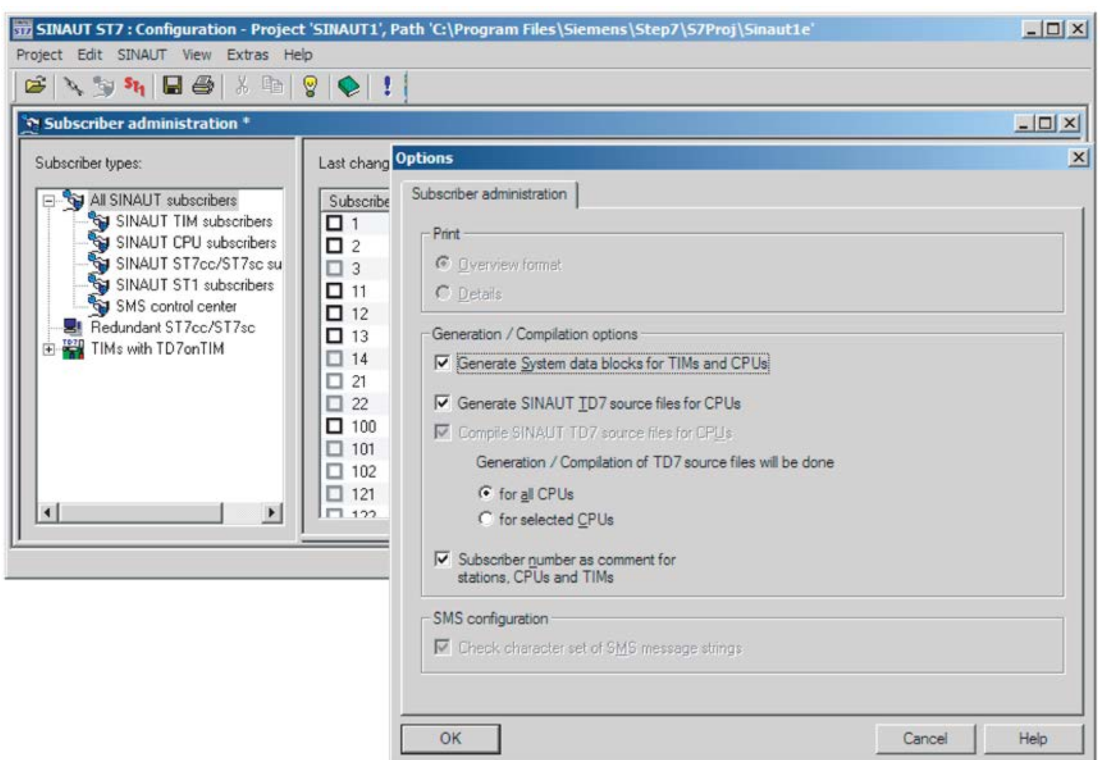


Figure 5-12 Dialog for triggering compilation of the SINAUT TD7 blocks

After you have saved, all the SINAUT blocks basically required are entered in the block directories of all CPUs, and among other things also the UDTs with the new numbers you have selected. The SINAUT UDTs are required to generate the BasicData data block. This central administrative block also exists in the block directory and is generated taking into account the new UDT numbers.

Changing the numbers of auxiliary blocks

The prerequisite for using the instructions here is as follows:

- You want to change the numbers of SINAUT auxiliary blocks in a new project (see below, initial situation 1),
  or
- You want to copy further SINAUT blocks into a project in which numbers of SINAUT auxiliary blocks have already been changed; it does not matter whether these are auxiliary blocks or not (see below, initial situation 2).

Initial situation 1

You configure your SINAUT installation and as described in the section 'Configuration software for SINAUT ST7' and have configure the connections with the appropriate SINAUT configuration tool. You have not yet started the 'Subscriber Administration' SINAUT tool.

Or the S7 CPUs and TIM modules exist in your project. The block directory of the CPU contains only the system data and OB1.

Making the change

The change to the numbers is made indirectly using the symbol table of the CPU with the aid of the 'Subscriber Administration' SINAUT configuration tool.

Step 1

Fill the block directories of all S7 CPUs with the user blocks (FBs, FCs, DBs and UDTs) whose numbers you want to retain. If you have assigned symbolic names to your blocks, enter these in the symbol tables of the CPUs.

Step 2

Then check which SINAUT blocks you require for your program. The following table shows which blocks are always required by the SINAUT program (depending on the CPU type, there are slight variations). The SMSxxx blocks are required only when the SMS function is configured.

Table 5- 2 SINAUT blocks that are always required

S7-300	S7-400
FB126 XCom	FB127 BCom
FB125 PCom	
FB124 SMS_Control	
FC122 Startup	
FC123 Create	
FC124 Distribute	
FC125 Diagnose	

S7-300	S7-400
FC126 Search	
FC127 BasicTask	
DBxxx SMS_Data	
DB127 BasicData	
UDT126 ConnectionDescription	
UDT127 SubscribeObject	

Based on the table, you can see which other blocks may be required. At least one or more blocks for data acquisition and output are required. These are the blocks in the range from FB40 ... FB117. Further optional basic functions may also be required that are available in the library under FC114 ...FC121 (you will find information on these optional basic functions below in the section 'SINAUT startup program in OB100').

When putting together the required SINAUT blocks, does not matter whether or not these blocks have numbers that need to be changed. You should identify all the SINAUT blocks required per CPU.

Copy the rows from the symbol table of the SINAUT TD7 library with the required SINAUT blocks to the symbol tables of the individual CPUs. In the symbol tables of the CPUs, change the SINAUT block numbers to the required numbers.

Note

If a change is the same for several CPUs, the most efficient method is to adapt the symbol table first for one of the CPUs and then to copy these entries to the symbol tables of all other CPUs.

Step 3

Once all the symbol tables have the required values, change to the 'Subscriber Administration' SINAUT ST7 Configuration Tool. Click the 'Save' button. An additional dialog "Properties" opens. In this dialog, make sure that there is a check mark in front of 'System data blocks for TIMs and CPUs' and in front of 'SINAUT TD7 block for CPUS' and a dot in front of 'for all CPUs'. Then click on 'OK'.

After the save is completed, the SINAUT blocks listed in the symbol table have been added to the content of the block directory and one or more communication DBs have been added (recognizable by their symbolic names, for example XComData01 or BComData01). The SINAUT blocks now have their new numbers. If numbers of SINAUT auxiliary blocks have been changed, all SINAUT blocks that call these auxiliary blocks have been recompiled; in other words, they now call these auxiliary blocks internally with the new numbers.

Initial situation 2

You want to copy further SINAUT blocks to a CPU on which the numbers of SINAUT auxiliary blocks have already been changed. It does not matter whether or not these are auxiliary blocks or not and whether you also want to give these blocks new numbers.

Copying later

The blocks to be added to the existing, changed SINAUT auxiliary blocks are adapted and any number changes to these blocks made indirectly using the symbol table of the CPU and with the aid of the 'Subscriber Administration' SINAUT configuration tool.

Step 1

Copy the rows from the symbol table of the SINAUT TD7 library with the SINAUT blocks you still require to the symbol table of the CPU. If necessary, change the numbers of these blocks in the symbol table of the CPU.

If you want to add the same blocks to further CPUs, the most efficient method is to copy the new entries from the symbol table of the first CPU to the symbol tables of the other CPUs, particularly if you have assigned different numbers to the new blocks. Changes to the numbers of these blocks are then adopted directly on the other CPUs.

Step 2

Change to the 'Subscriber Administration' SINAUT configuration tool and save again as explained in Step 3.

When the save is complete, the blocks newly added to the symbol table are stored in the block directory of the CPU, if applicable with new block numbers. They have also been recompiled and therefore adapted to their local SINAUT environment; in other words internally, they call the SINAUT auxiliary blocks under their new numbers.

Note

If a SINAUT block that you have added already exists on one of the other CPUs of the project, and if the same number changes have been made for the SINAUT auxiliary blocks, the block can also be copied from the program of the relevant CPU. In this case, however, no new entry is made in the symbol table of the target CPU as is the case when copying from the TD7 library *). You should therefore modify the symbol name manually to avoid errors during subsequent compilations.

*) This applies only to versions older than STEP 7 Version 5.1

5.1.4 Copying programs

Introduction

If several stations of a SINAUT project should receive the same or practically the same program, it is advisable to finish this program initially for one of the stations. Afterwards it is copied to all other stations and adapted to the local situation. A safe method for copying programs in SINAUT projects is described below.

Initial situation

You have configured your SINAUT system as already described in the section 'Configuration software for SINAUT ST7'; in other words, you have completed the connection configuration with the appropriate SINAUT configuration tool and have started to save in the 'Subscriber Administration' SINAUT tool. The result is that all the necessary SINAUT blocks already exist in all the program directories of the CPUs in addition to the SDB directory and OB1. You have finished the program for one of the CPUs that you want to copy to other CPUs.

Performing the copying.

With the method described below, the compete content of the S7 program is copied from the source CPU to the target CPU(s). Then on the target CPUs a correction of the CPU SDBs and the SINAUT data blocks (BasicData, XComData, PComData, BComData) is necessary. This is done by triggering a save procedure in the SINAUT configuration tool "Subscriber administration".

Step 1

- Open the directory S7 program(..)mon the CPU whose program you have finished as the model.
- In the right sub-window, select all subdirectories Sources, Symbols and Blocks (see figure).
- Copy the selected elements to the clipboard: either using the "Copy" function in the "Edit" menu or with the Windows shortcut *Ctrl + C*.

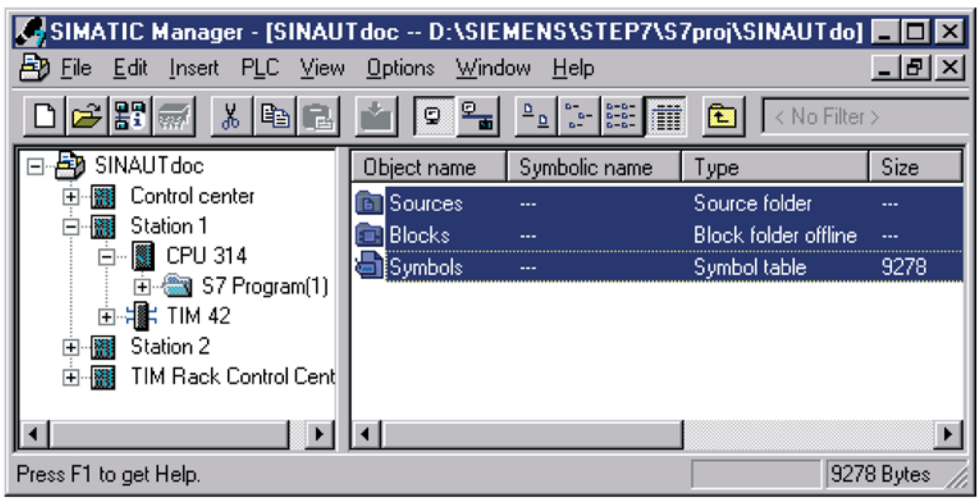


Figure 5-13 Selecting the program of the source CPU

Step 2

- Now open the directory S7 program(..) of the CPU in which you want to insert the program from the clipboard.
- In the right sub-window, select the 3 subdirectories Sources, Symbols and Blocks again.
- Now trigger the "Paste" function: either using the "Paste" function in the "Edit" menu or with the Windows shortcut *Ctrl + V*.

Repeat step 2 for all other CPUs that are to receive the same program.

Step 3

- Change to the 'Subscriber Administration' SINAUT configuration tool and start a save procedure there.
- Make sure that in the displayed dialog "Properties" the check marks before
  - 'System data blocks for TIMS and CPUs'
    and
  - 'SINAUT TD7 blocks for CPUs'

are selected.

- Then click on 'OK'.

On completion of the save it is guaranteed that the SINAUT-specific data has been adapted to the local situation on all CPUs and the SDBs of the CPUs once again have the correct content.

5.1.5 Using online help

Introduction

This section "Software package SINAUT TD7 for the CPU" is on the PG as a help file. This means you can access this section online. How you access the online help is explained in the following sections.

Access to the online help of the SINAUT TD7 library

To change to the online help from the SINAUT TD7 library, simply select a block from the 'Blocks' directory, e.g. FB82. Then press the F1 key. A description of the selected block is displayed immediately.

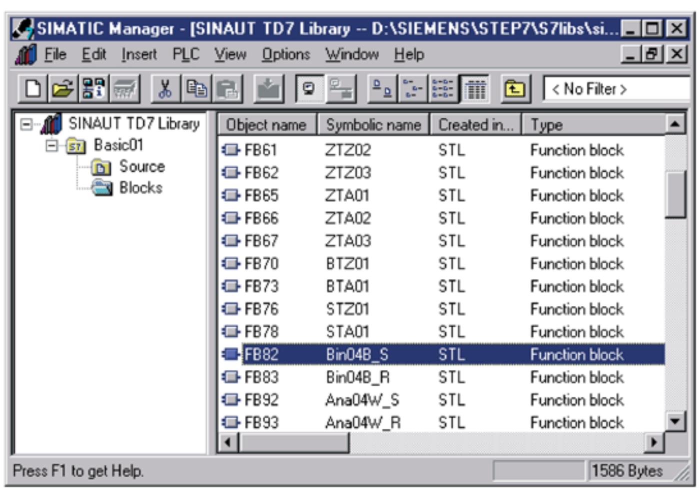


Figure 5-14 Access to the online help of the SINAUT TD7 library

Access to the online help from the blocks directory of the user program.

Similar to the SINAUT TD7 Library, you can also change to the online help from the blocks directory of the user program: the required block is selected and then the F1 key pressed.

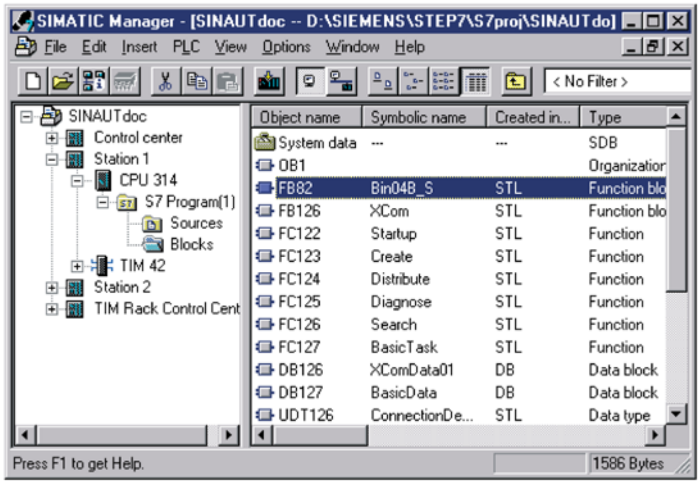


Figure 5-15 Access to the online help from the blocks directory of the user program.

Access to the online help when creating the program

The online help can also be called when programming. To do this the cursor must be in the line with the call of the block for which help is required. In the following figure, foe example, the cursor is in the line with the call for FB82 'Bin04B_S'. If the F1 key is pressed now, the help text for this block appears.

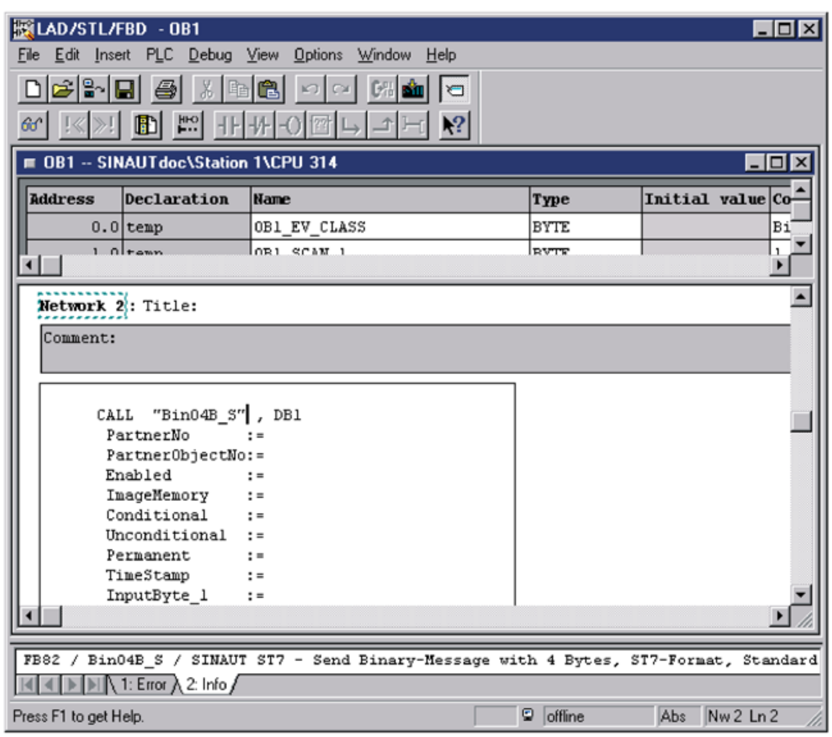


Figure 5-16 Access to the online help when creating the program

Example of a help text

The following figure shows an excerpt from the help text when the F1 key is pressed in the preceding examples.

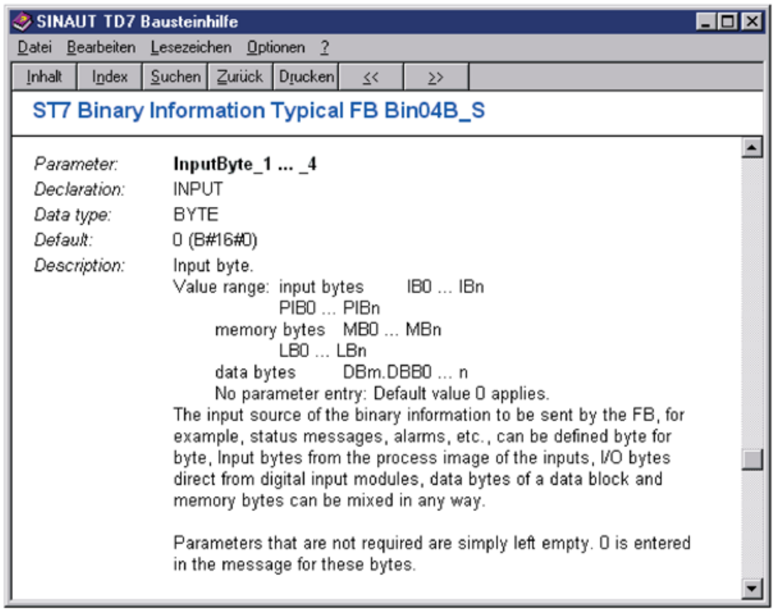


Figure 5-17 Example of a help text

Selecting help topics

From every help text for a block, the user can access to total content of the help file. After double clicking on 'Content', the box 'Help topics: SINAUT TD7 block help (see following figure). From here, it is possible to branch to every section of this chapter 'Software package SINAUT TD7 for the CPU' either via the 'Content', 'Index' or 'Find' tab.

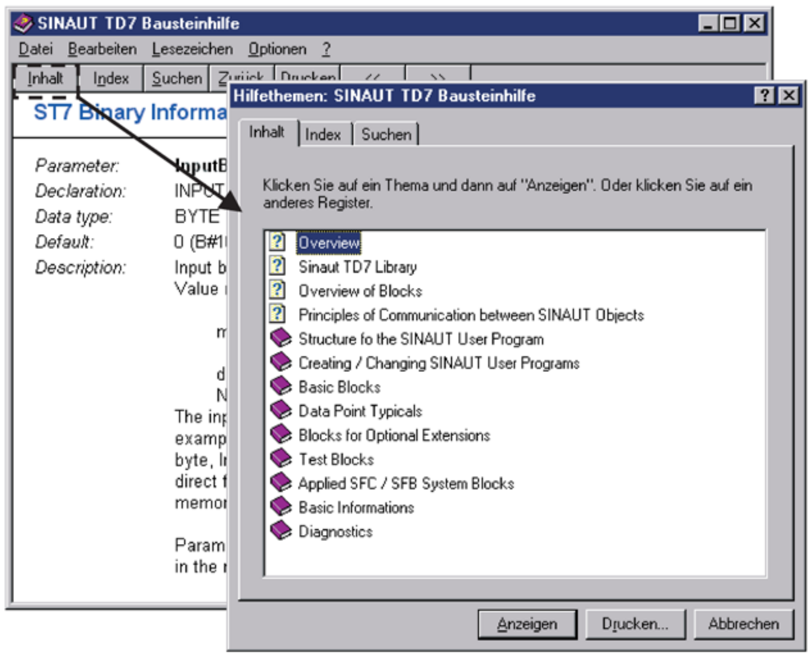


Figure 5-18 Selecting help topics

5.2 Principle of communication between SINAUT objects

Introduction

In SINAUT ST7 the term "object" refers to the representation and handling of process variables such as messages, analog values, commands, motors, valves, controllers etc. The TD7 software always processes "objects". An object is always made up of a process part and an operator control and monitoring part (simply called operator part below) that operate in different PLCs (nodes) in the SINAUT network. To be able to process their predetermined function the two parts must communicate with each other. The operator side sends setpoints, parameters, commands and organizational instructions to the process object, this returns process data, alarm and status messages and organizational information.

An object in the TD7 world both at the process end and at the operator end consists of a processing instruction - below also called typical - in the form of an S7 function block (FB) and of the assigned object data record that is implemented as an instance DB. The following figure shows the principle of the data exchange between the process and the operator part of an object.

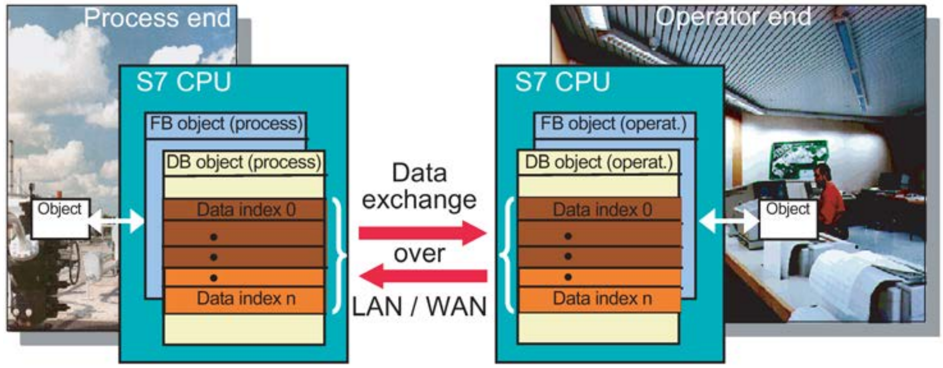


Figure 5-19 The principle of object communication

Essentially the data that describes the object is exchanged. This is located in the object data record in the data indexes 0...n. The extent and composition of this data area depend on the typical involved. It can consist of several identical or a combination of different data types. In the object data record at the process and operator end, the data structure of two typicals that belong together is identical

The data exchange does not necessarily run in both directions. There are also simpler objects with which there is only one-sided data transfer, e.g. only process data to the operator part.

After the process and operator data has been exchanged, organizational information is transferred between the two ends. This data flow and the data areas in the object data record intended for this (the so-called Org. indexes) are not shown in the figure above.

The principle of object addressing

Each programmable controller with SINAUT TD7 software (Also called PLC or CPU below) receives a SINAUT subscriber number that can be assigned in the range from 1 to 32000 and is unique throughout the network. Each typical that is called in one of these CPUs has an instance DB whose number is identical to the SINAUT object number. With the aid of the SINAUT subscriber number and the SINAUT object number, explicit addressing for the communication between the typicals that belong together can be implemented.

To specify the communications relation, every typical as the two parameters

- PartnerNo
  Subscriber number of the partner in the communication.
- PartnerObjectNo
  Object number (= instance DB no.) of this partner.

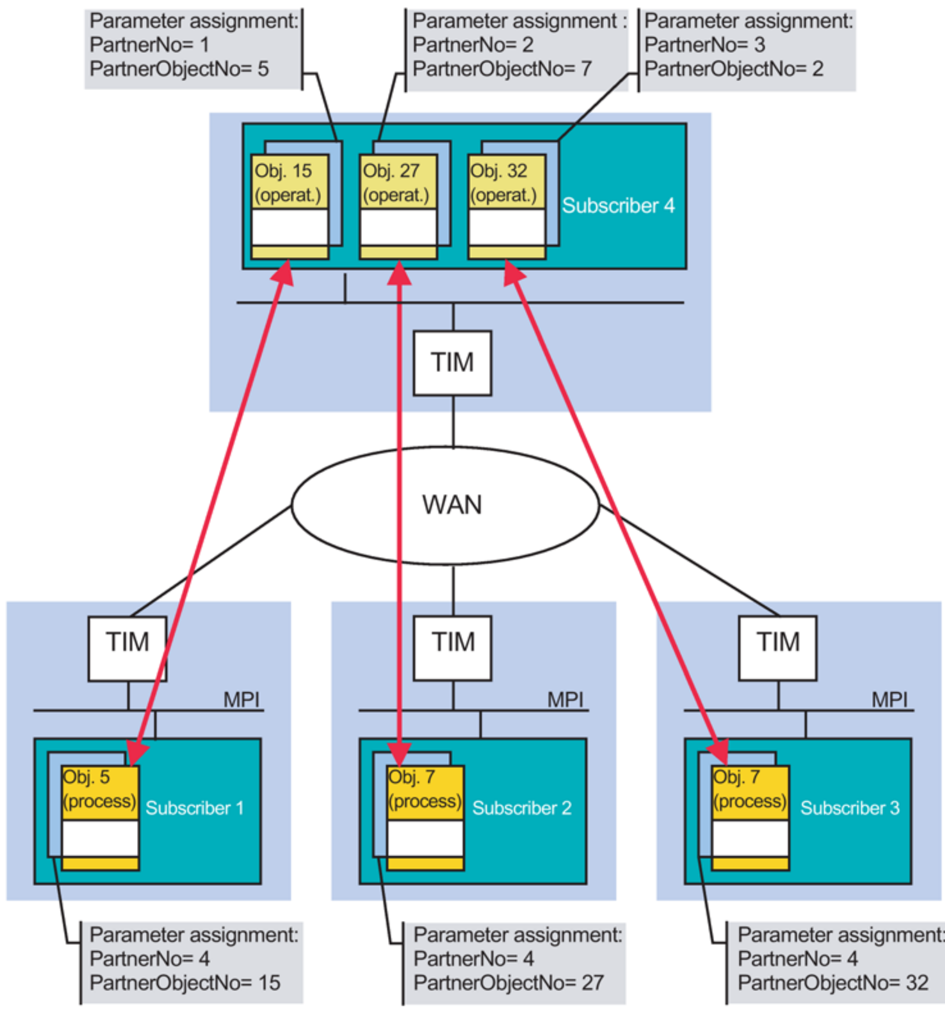


Figure 5-20 Object communication in the WAN area

The figure shows several examples of addressing for objects that exchange data with their partner objects via WAN.

Explanation based on the example of object communication between subscribers 1 and 4:

In subscriber 1 there is the process end object no. 5 that is to communicate with the operator end object no. 15 on subscriber 4. At the typical of the object 5, the following therefore needs to be specified as the partner addressing:

PartnerNo = 4 (Subscriber number of the operator partner)

PartnerObjectNo = 15 (Number of the relevant object on this partner)

In the other direction the partner addressing for object 15 is as follows:

PartnerNo = 1 (Subscriber number of the process partner)

PartnerObjectNo = 5 (Number of the relevant object on this partner)

When object 5 wants to transfer data to its partner object 15, a data message is put together in which the set two-level partner address is entered as the destination address. Due to the destination subscriber 4 (= PartnerNo) the SINAUT TD7 software and the TIM modules in the SINAUT network ensure that this message is delivered to the specified destination subscriber 4.

If the message has arrived at destination subscriber 4, based on the destination object number 15 (= PartnerObjectNo) the TD7 software there can determine that the information in the message should be store in the local DB15. For the storage, the index number also included in the message is taken into account. If this is, for example, X, the information is stored starting at the data index X in DB15. In addition to this the fact that new data has arrived and which data indexes were renewed is also entered in this object DB. In the next program cycle the operator typical that processes this object will recognize the new data, process it as performed specifically by this typical and output it to the data outputs set for the typical.

In the opposite direction, the operator object 15 will send its data that is intended for process object 5 in a message in which the subscriber no. 1 and object no. 5 are entered as the destination address. Via the TD7 software and TIMs this message will finally arrive at subscriber 1. The information contained in the message will be entered in the object DB 5 depending on the destination object no. 5 and the data index. In the next program cycle the process typical will prepare the newly received data and output it in a suitable form to the set outputs.

Object communication via LAN

Apart from the data transfer between SINAUT objects via WAN, with the aid of the SINAUT TD7 software local communication via LAN is possible (the current software currently only supports the MPI bus as the LAN).

The following figure shows an assumed system structure in which several PLCs are intended in one station. Each PLC handles a certain automation task (in the example the subscribers 10, 11 and 12). At a higher level there is a further PLC (subscriber 13) in this station which contains a user control that controls the automation in the lower-level PLCs.

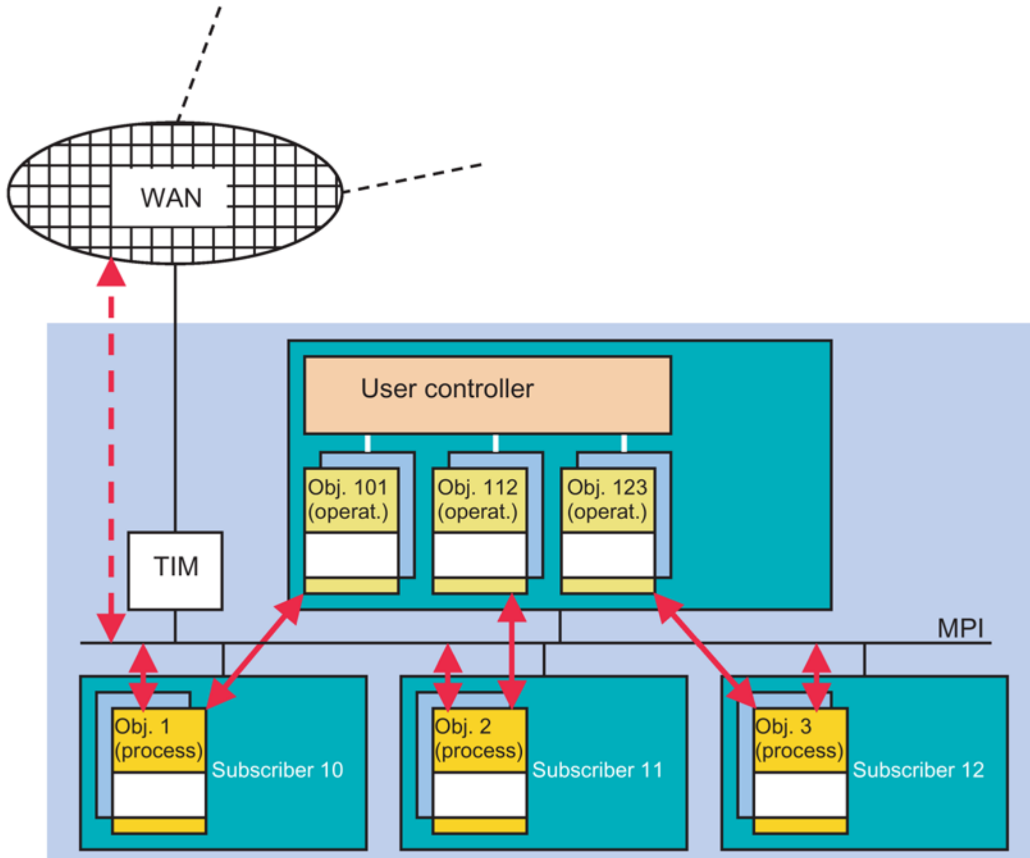


Figure 5-21 Object communication in the LAN (MPI)

For this use case, the TD7 software with its process and operator typicals could be used as shown in the figure. In the lower-level PLCs, the process typicals handle the local automation and transfer data changes via the MPI bus to the operator objects in the central control device subscriber 13. The user control reads in the process information output to the operator objects and processes this accordingly. If due to the current situation commands or setpoints need to be transferred to the process objects, these are entered by the user control via the corresponding inputs of the operator typicals. These then take care of the immediate transfer to the process objects that take into account this information once again specifically for the typical for their automation task.

The process objects can not only transfer their data to the relevant operator object in subscriber 13 and receive control instructions from there. It is also possible to transfer the process information additionally to a further partner, e.g. a control center that can be reached via WAN and also to receive any commands or setpoints from there. How communication with more than one partner can be implemented is explained in the next section.

Object communication with multiple operator subscribers

Apart from the communication between a process typical and an operator typical as shown in the figure above the process typical can also exchange data with more than one operator partner. The following figure illustrates the principle for the situation where on local subscriber 2 and on the subscribers 3 and 4 reachable via WAN an operator typical is provided for the process object in subscriber 1.

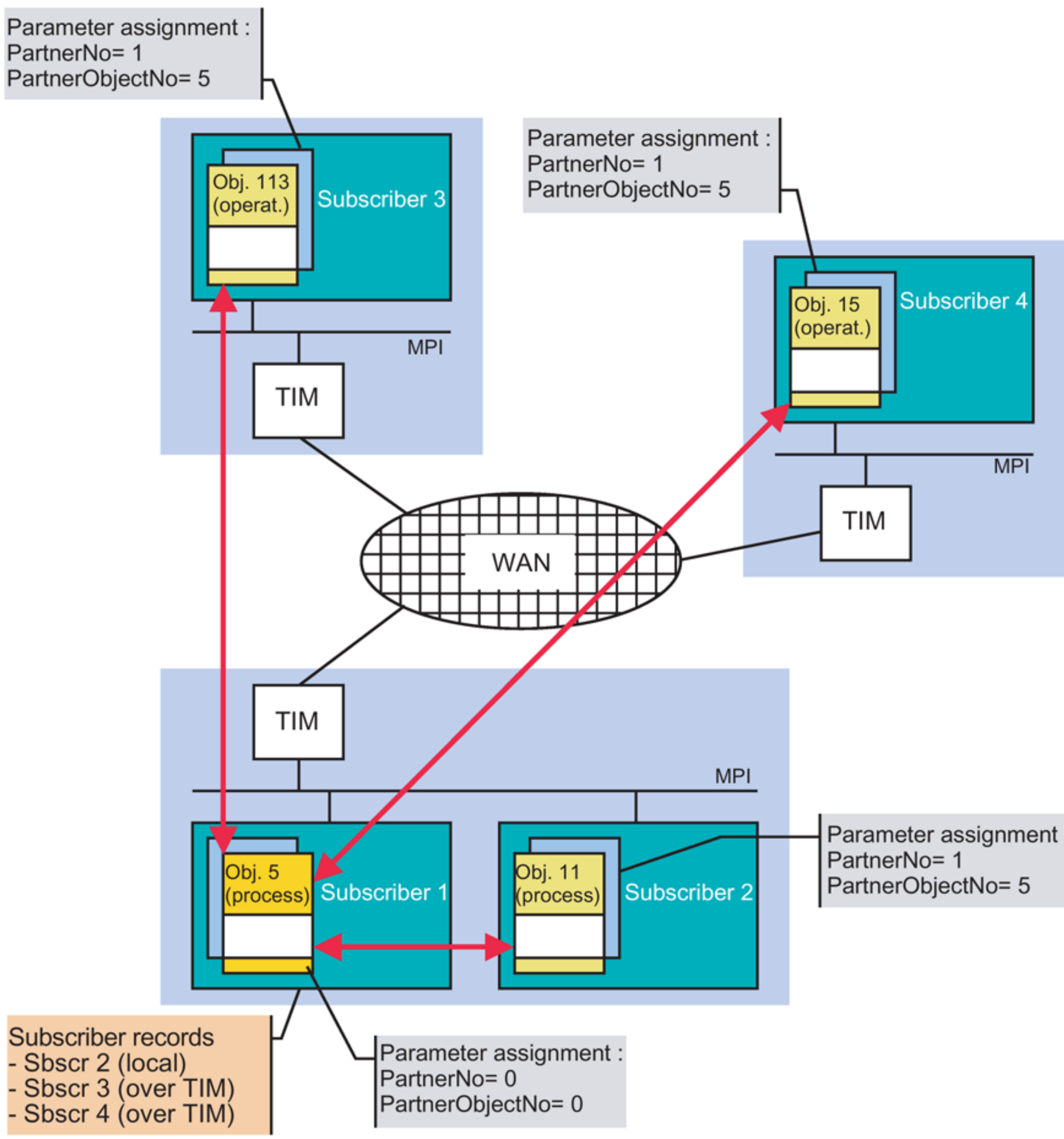


Figure 5-22 Object communication with multiple operator subscribers

From the processing end, the data acquired there is transferred at the same time to all three operator partners. From the operator end, operator data can be sent to the process object at any time independently. Feedback from the process caused by an operator instruction of a subscriber, is automatically transferred to all three operator objects. Each operator end therefore always has the latest information. With a general or single request, the process object also transfers the requested data not only to the partner that made the request but also to the two other partner objects.

For the process typical, in this case, the subscribers 2, 3 and 4 must either be set explicitly as partners (the typical then also needs at least three parameters for partners which is currently not the case) or no partner addressing specified for the typical. In the last case, the TD7 software automatically ensures transfer to all subscribers to which a connection was configured (with the aid of the SINAUT connection configuration). In the example for subscriber 1, both a connection to subscriber 2 and subscriber 3 and 4 must have been configured.

Due to this connection configuration it is known in the subscriber record keeping on subscriber 1 that a local connection to subscriber 2 exists as well as a WAN connection to subscriber 3 and 4 that runs for both via the local TIM module. The TD7 then puts two messages together: one message that is transferred via the MPI bus to the locally reachable partner 2 and one message that is transferred to the TIM for forwarding. The TIM itself checks which partners are known to it in conjunction with subscriber 1. From the record-keeping it identifies these as subscriber 3 and subscriber 4.

Assuming that the SINAUT network is a dial-up network, the TIM will duplicate the message: once for subscriber 3 and once for subscriber 4. The TIM adds the destination subscriber number 3 or 4 that has been missing up to now to both messages. Following this the TIM establishes the connection to subscriber 3 and transfers the message intended for it. Afterwards, subscriber 4 is addressed and receives the copy of the message created for it.

If the SINAUT network is a dedicated line network and the TIM of subscriber 3 is the master for the call-up operation (the so-called central TIM), the TIM of subscriber 1 cannot make a direct transfer to 3 and 4 as in the dial-up network. In the current situation, the message intended for subscriber 4 must be transferred indirectly via the central TIM of subscriber 3. The message to be sent is not now duplicated but has two destination addresses added, the 3 and the 4. This message is then transferred during the next call to the central TIM. Due to the two destination addresses contained and based on its subscriber record-keeping, this determines that one of them can be reached by it locally and the other destination via the dedicated line network. The central TIM therefore makes a copy of the message. Destination address 4 is removed from the original and the message is forwarded to the remaining locally reachable subscriber 3 with the remaining destination address 3. The copy only has the destination address 4 and is transferred via the dedicated line to the TIM of subscriber 4 and is then transferred to the subscriber 4.

Interaction of the blocks

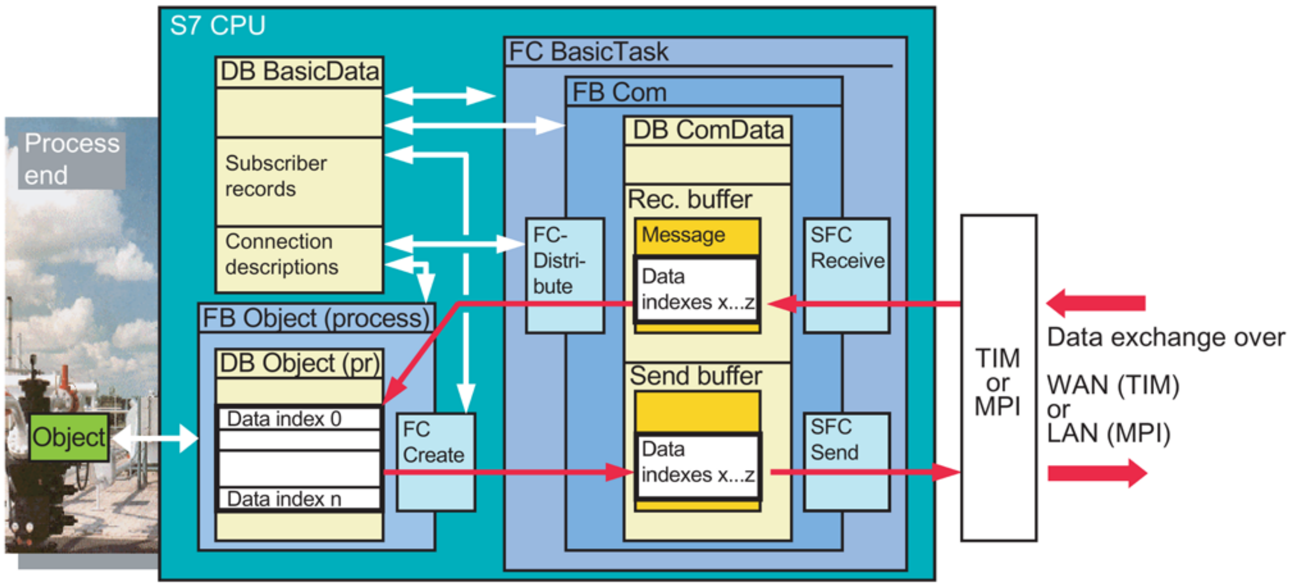


Figure 5-23 Interaction of the blocks (based on the example of the process end)

5.3 Structure of the SINAUT user program

Introduction

The SINAUT user program is contained in the following organization blocks:

- SINAUT startup program in OB100
- Cyclic SINAUT program in OB1
- Time-driven SINAUT program in a cyclic interrupt OB, for example OB35
- SINAUT test routine in the programming error OB121

Note

In the following description of the SINAUT program structure, the tasks of the individual blocks are only outlined briefly. For more detailed information on the functions and parameters, refer to the descriptions of the blocks mentioned.

5.3.1 SINAUT startup program in OB100

The structure of the SINAUT program in the startup OB100 is as follows:

Startup OB100	
Startup	• The only task required here for the SINAUT program is to integrate the FC Startup call in the startup OB100. The FC has no parameters.
	• User-specific startup functions that are required independent of the SINAUT program can be included before or after FC Startup in the startup OB.

Note

Startup OB101, which is intended for S7-400 restarts, may not be used!

5.3.2 Cyclic SINAUT program in OB1

Introduction

The basic structure of the cyclic SINAUT program in OB1 is described below. In later sections, you will find a detailed description of OB1 for a station and a master station.

Note

Unless indicated otherwise, the call sequence of the blocks must be adhered to exactly!

The entire cyclic SINAUT program must be processed in every OB1 cycle.

Basic structure of the cyclic SINAUT program in OB1

Cyclic OB1		
BasicTask	• FC BasicTask must always be called at the start of the cyclic SINAUT program. It handles basic SINAUT tasks that are always required.	
Optional SINAUT basic functions	• Directly following FC BasicTask, additional blocks can be called to perform optional basic functions, for example:	
	• - FC TimeTask	Provides the SINAUT time.
	• - FC Trigger	Scheduled starts for user programs and data messages.
	• - FC PartnerStatus	Displays subscriber OK/disrupted
	• - FC PartnerMonitor	Extended subscriber-specific display and control features.

Cyclic OB1		
	• - FC ListGenerator	Creation of address lists for received messages with incomplete destination addresses.
Data point typicals	• Following the FCs shown above 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. You can see which data point typicals are currently available by referring to the SINAUT TD7 library installed on the programming device. 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 datapoint object (this addressing rule does not apply to the ST1 versions of the data point typicals). • The user-specific cyclic program that is required independent of the SINAUT program can be included before or after the SINAUT program in OB1 or, if suitable, within the SINAUT program itself.	

Note

Users, of course, is free to structure the SINAUT program in OB1 according to their preferences by ’packing’ the SINAUT program in one or more FCs.

The cyclic OB1 program for a station

Cyclic OB1	
BasicTask	• FC BasicTask must always be called at the start of the cyclic SINAUT program. The FC has only one parameter, namely UserFC. Normally 0 can be specified. However, if you require user-specific processing for received messages, you will need to specify the number of an FC containing the user program for this processing.
TimeTask	• As an option, you can call FC TimeTask immediately after FC BasicTask. The FC has no parameters. FC TimeTask must be included if you need the SINAUT time. This enables SINAUT messages to be time-stamped. However, you can also use the SINAUT time to start program components at a specific point in time or to schedule the transmission of data messages. FC Trigger, described below, is then required. For this FC to be used, the PLC must be provided with the SINAUT time from a local TIM module. This can be specified during the parameter assignment of the TIM in HW Config in the ’Properties’ dialog, ’Time service’ tab. See chapter ’Configuration software for SINAUT ST7’.
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. Requirement for the use of the FC: FC TimeTask must be called first in the OB1 program (see above).

Cyclic OB1	
PartnerStatus	• FC PartnerStatus can be included as an option. The FC shows the current 'disrupted' or 'OK' status for a maximum of 8 SINAUT subscribers (communication partners).
ListGenerator	• FC ListGenerator300 (for S7-300 CPU) or FC ListGenerator400 (for S7-400 CPU) can be included as options. The FC is required if the station receives messages containing no destination address or an incomplete destination address. This can occur in the following situations: – When the station receives ST1 messages. – When the configuration of the destination address is omitted in one or more data point typicals in an ST7 device with which the station communicates (the parameters PartnerNo and PartnerObjectNo were not specified; there is therefore a transmission to all known destination subscribers). Following the FCs shown above 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: • Send – Binary information, such as status messages and alarms – Analog values – Counted values • Receive – Commands – Setpoints (including limit values, parameters, etc.) 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 datapoint object (this addressing rule does not apply to the ST1 versions of the data point typicals). A data point object consists of one or more data points of the same type, e.g. 4 bytes of binary information, or 4 analog values, or 1 byte commands, etc.
ST7 format	
Bin..._S	• The following should be inserted for acquiring and transmitting of binary information, such as status information, alarms, etc.: For ST7: one or more FB-Bin…_S
Ana…_S	• The following should be inserted for acquiring and transmitting analog values: For ST7: one or more FB-Ana…_S
Cnt…_S	• The following should be inserted for acquiring and transmitting counted values: For ST7: one or more FB-Cnt…_S A requirement for the use of the FBs mentioned is that FC PulseCounter is included in a cyclic interrupt OB, e.g. OB35. This FC is responsible for the actual (time-driven) acquisition of counted pulses in the background.
CMD..._R	• The following should be included for receiving and outputting commands: For ST7: one or more FB-Cmd…_R

Cyclic OB1	
Set..._R or Par…_R	• The following should be included for receiving and outputting setpoints, limits, parameters, etc.: For ST7: one or more FB-Set…_R or FB-Par..._R
	• The user-specific cyclic program for the station that is required independent of the SINAUT program can be included before or after the SINAUT program in OB1 or, if suitable, within the SINAUT program itself.

In the above OB1 program structure for a station, only data point typicals which process data of the same type are listed. There are additional data point typicals that can send and receive any combination of data types:

For ST7:

- FB-Dat12D_S for sending or
- FB-Dat12D_R for receiving 12 data double-words with any content

These typicals can also be included in the cyclic SINAUT program in place of or in addition to the data point typicals in the program structure here.

Note

There is no data-specific processing and change control for these typicals for any combination of information types. The user program is responsible for this. The only optional change control that can be activated is the triggering of a transmission at each bit change.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

The cyclic OB1 program for a master station

Cyclic OB1	
BasicTask	• FC BasicTask must always be called at the start of the cyclic SINAUT program. The FC has only one parameter, namely UserFC. Normally 0 can be specified. However, if you require user-specific processing for received messages, you will need to specify the number of an FC containing the user program for this processing.
TimeTask	• As an option, you can call FC TimeTask immediately after FC BasicTask. The FC has no parameters. FC TimeTask must be included if you need the SINAUT time. This enables SINAUT messages to be time-stamped. You can also use the SINAUT time to start program sections according to a schedule. FC Trigger, described below, is then required. For this FC to be used, the PLC must be provided with the SINAUT time from a local TIM module. This can be specified during the parameter assignment of the TIM in HW Config in the ’Properties’ dialog, ’Time service’ tab. See chapter ’Configuration software for SINAUT ST7’.

Cyclic OB1	
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 more than once if several times or various time intervals are required. Requirement for the use of the FC: FC TimeTask must be called first in the OB1 program (see above).
PartnerStatus	• FC PartnerStatus can be included as an option. The FC indicates the current 'disrupted' or 'OK' status for a maximum of 8 SINAUT subscribers (communication partners). If you want to display the status for more than 8 subscribers, a corresponding number of PartnerStatus FCs must be included. The block is practical for monitoring the connections with local TIMs.
PartnerMonitor	• FC PartnerMonitor can be included as an option. This FC displays important status information about a SINAUT subscriber (communication partner). The FC can also be used to trigger a general request to the subscriber and to establish and disconnect a permanent connection with the subscriber. FC PartnerMonitor must be included once for each subscriber requiring the extended display and control features. FC PartnerStatus can be omitted for these subscribers. The block is practical for monitoring and controlling the connections with ST7 stations.

Cyclic OB1	
ListGenerator	FC ListGenerator300 (for S7-300 CPU) or FC ListGenerator400 (for S7-400 CPU) can be included as options. The FC is required if the master station receives messages containing no destination address or an incomplete destination address. This can occur in the following situations: • When the configuration of the destination address is omitted in one or more data point typicals in an ST7 station (the parameters PartnerNo and PartnerObjectNo were not specified; there is therefore a transmission to all known destination subscribers). • When ST1 stations are connected to the master station. Following the FCs shown above 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 master station: • Send – Commands – Setpoints (including limit values, parameters, etc.) • Receive – Binary information, such as status messages and alarms – Analog values – Counted values 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 datapoint object (this addressing rule does not apply to the ST1 versions of the data point typicals). A data point object consists of one or more data points of the same type, e.g. 4 bytes of binary information, or 4 analog values, or 1 byte commands, etc.
ST7 format	
Bin..._R	• The following should be inserted for receiving and outputting binary information, such as status information, alarms, etc.: For ST7: one or more FB-Bin…_R
Ana…_R	• The following should be included for receiving and outputting analog values: For ST7: one or more FB-Ana…_R
Cnt…_R	• The following should be included for receiving and outputting counted values: For ST7: one or more FB-Cnt…_R
CMD..._S	• The following should be inserted for acquiring and transmitting commands: For ST7: one or more FB-Cmd…_S A requirement for the use of the FBs mentioned is that FC Safe is included at the end of all data point typicals. This FC is responsible for reliable input of commands and setpoints.

Cyclic OB1	
Set..._S or Par…_S	• The following should be included for acquiring and transmitting setpoints, limits, parameters, etc.: For ST7: one or more FB-Set…_S or FB-Par..._S A requirement for the use of the FBs mentioned is that FC Safe is included at the end of all data point typicals. This FC is responsible for reliable input of commands and setpoints.
Safe	• FC Safe must be called once at the end of all data point typicals when send blocks for commands (Cmd..._S, BTZ..) or setpoints (Set..._S, Par..._S, STA..) are called. • The user-specific cyclic program for the master station that is required independent of the SINAUT program can be included before or after the SINAUT program in OB1 or, if suitable, within the SINAUT program itself.

In the above OB1 program structure for a master station, only data point typicals which process data of the same type are listed. There are additional data point typicals that can send and receive any combination of data types:

For ST7:

- FB-Dat12D_S for sending or
- FB-Dat12D_R for receiving 12 data double-words with any content

These typicals can also be included in the cyclic SINAUT program in place of or in addition to the data point typicals in the program structure here.

Note

There is no data-specific processing and change control for these typicals for any combination of information types. The user program is responsible for this. The only optional change control that can be activated is the triggering of a transmission at each bit change.

Note

In the master station it is practical to structure the OB1 program according to stations, in other words, all send and receive data typicals belonging to the same station are packaged in one FC. The best overview is provided when the number of the FC is identical to the subscriber number of the station.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

5.3.3 Time-driven SINAUT program in a cyclic interrupt OB

Introduction

A time-controlled SINAUT program only needs to be installed in a CPU if count pulses need to be detected in this CPU.

The count pulse acquisition is 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, with an S7-300 CPU OB35 needs to be programmed and with an S7-400 CPU one of the available cyclic interrupt OBs OB30 to OB38. Although all cyclic interrupt OBs have a preset time interval (with OB35 e.g. 100 ms), this can be changed in 1 ms steps. This makes it possible to set a cyclic interrupt OB, for example, to 25 ms.

The following figure shows how the change to the time interval for a cyclic interrupt OB is made in HW Config in the Properties dialog of the CPU.

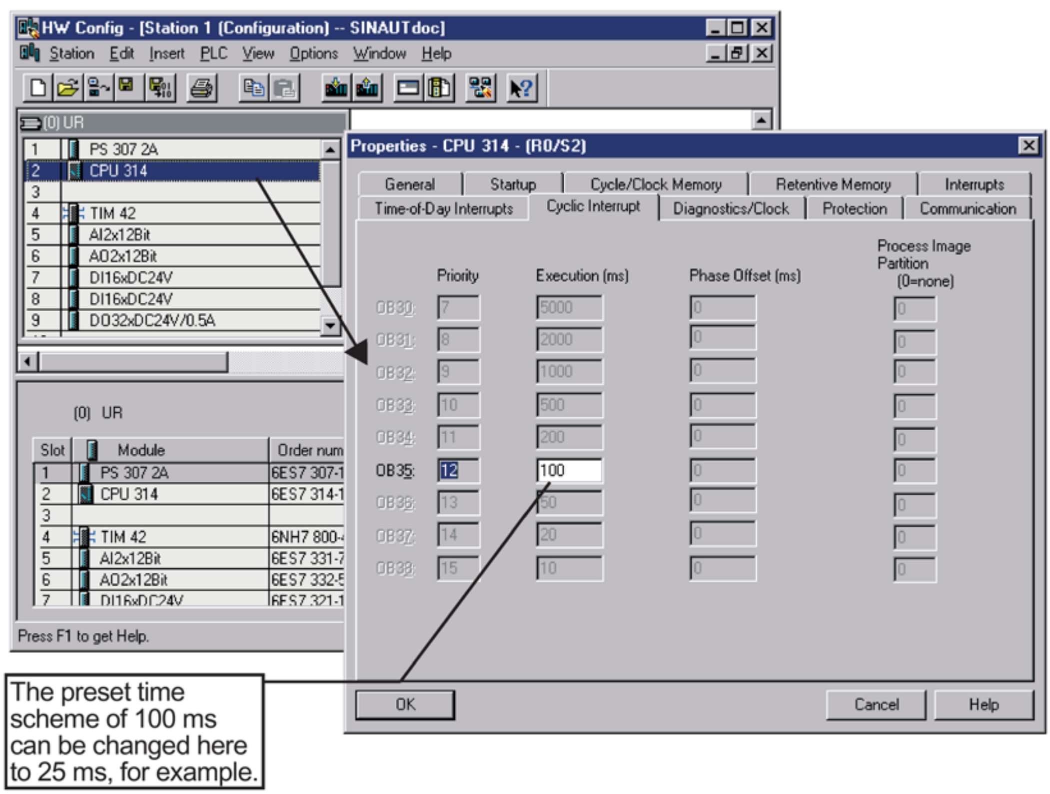


Figure 5-24 Changing the time interval for the cyclic interrupt OB35

The program structure in the cyclic interrupt OB

The structure of the SINAUT program in the cyclic interrupt OB appears as follows:

Cyclic interrupt OB	
Pulse counter	• One or more FC-PulseCounters can be included for the acquisition of count pulses. The FC-PulseCounter processes up to 8 pulse inputs of any digital input. The acquired count pulses are then added in SIMATIC counters that can have parameters assigned and which are accessed by the function blocks that put together the count value messages (FB-Cnt..._S, FB-ZTZ..). See the explanations of the structure of the SINAUT program in OB1. • User-specific cyclic interrupt functions required independently of the SINAUT program can be included at any point in the cyclic interrupt OB.

5.3.4 SINAUT test routine in the programming error OB121

Introduction

When a block that does not exist is called in a CPU, the CPU normally changes to STOP and the diagnostics buffer indicates which block (FBxx, FCyy or DBzz) is 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 with the user program. 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, however, to decide more selectively when it continues to run and when it changes to STOP, you need to include a suitable use a program in OB121.

In conjunction with SINAUT ST7 it is possible that a CPU changes to STOP when it receives a message from another CPU that it does not (or not yet) know. For example 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 set destination object no. can then lead to a stop of the destination subscriber because as soon as you install the new data point typical in the station, the message is transferred to the destination. If, however, there is no corresponding received typical installed on the destination CPU, the destination object no. (= Instance DB of the received typical to be installed) does not exist. The result is then that the CPU changes to stop as soon as this message is received unless you have made sure that this does not happen in OB121. For SINAUT ST7 CPUs it is recommended that you call the FC-ST7ObjectTest function in OB121. The CPU does not then change to STOP if a SINAUT object DB is not present as in the example shown above.

FC-ST7ObjectTest has one 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.

The program structure in the programming error OB121

The structure of the SINAUT program in the programming error OB121 appears as follows:

OB121	
ST7ObjectTest	• Calling FC-ST7ObjectTest in OB121 prevents a CPU stop, if the CPU receives a message with an unknown destination object no. 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. • A user-specific program that is required independently of the SINAUT program can be included at any point in OB121.

5.4 Basic blocks

These blocks are needed for organizational tasks within a CPU and for controlling and monitoring all the transmission channels.

5.4.1 FC Startup

This block is required in every CPU. It must be included in the startup program OB100. Its only task is to set the startup memory bit in the DB BasicData and reset the corresponding edge memory bit if it is still set.

The block has no parameters.

In a normally configured SINAUT installation, FC Startup is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

For more detailed information on the SINAUT startup program, refer to the section 'SINAUT startup program in OB100’.

5.4.2 FC BasicTask

Function

This block is required in every CPU. It handles

- the central tasks to be performed during startup
- the processing of all communication mailboxes
- the central organizational tasks such as starting and monitoring general requests, answering general requests etc.

In a normally configured SINAUT installation, FC-BasicTask is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool 'Subscriber Administration' and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

Note

On principle, FC-BasicTask should be called as the first block in the cyclic SINAUT program (in OB1).

Explanation of the parameters

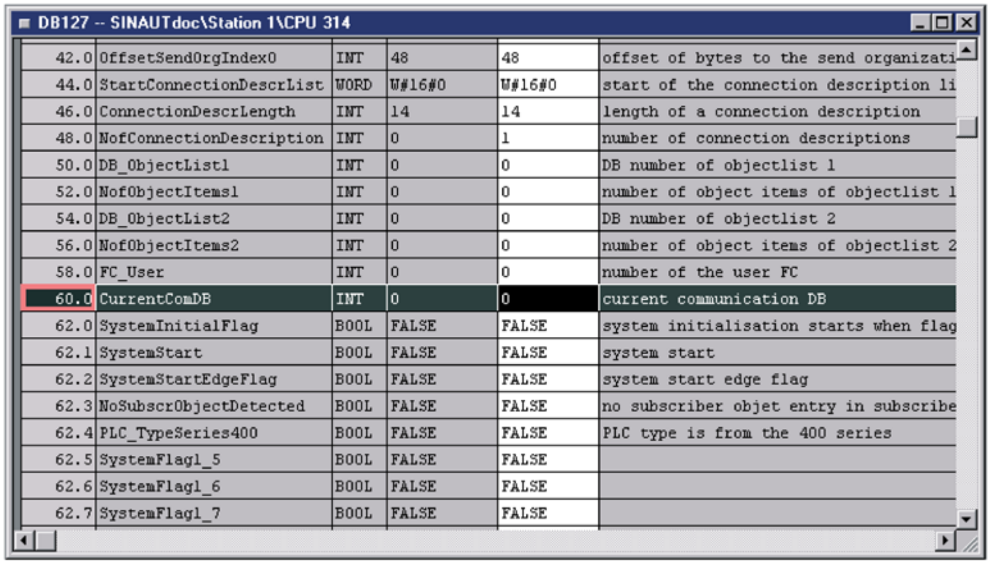
Name:	**UserFC**		
Declaration:	INPUT		
Data type:	INT		
Explanation:	Number of a user FC for user-specific further processing of the received SINAUT messages.		
	Range of values:	0 or 1 ... 32000	
		0 =	Dummy value if there is no user FC present for the specified purpose.
		1 ... 32000 =	Number of the user FC. The maximum possible number depends on the CPU in which the SINAUT program will run.

If a user FC is specified, this FC is called automatically by the SINAUT program with each received message. At the time of the call, the received message is still in the receive mailbox of the communication DB. The user program in the user FC can read the received message from the receive mailbox and process it further in any way required, e.g. write it to an intermediate buffer.

The number of the current communication DB can be read out of DW60 of the DB-BasicData by the user program (symbolic address: CurrentComDB).

As soon as the communication DB is open, the start of the current received message can be found in the receive mailbox via the pointer in DW10 (symbolic address: CurrentReceivedMessage).

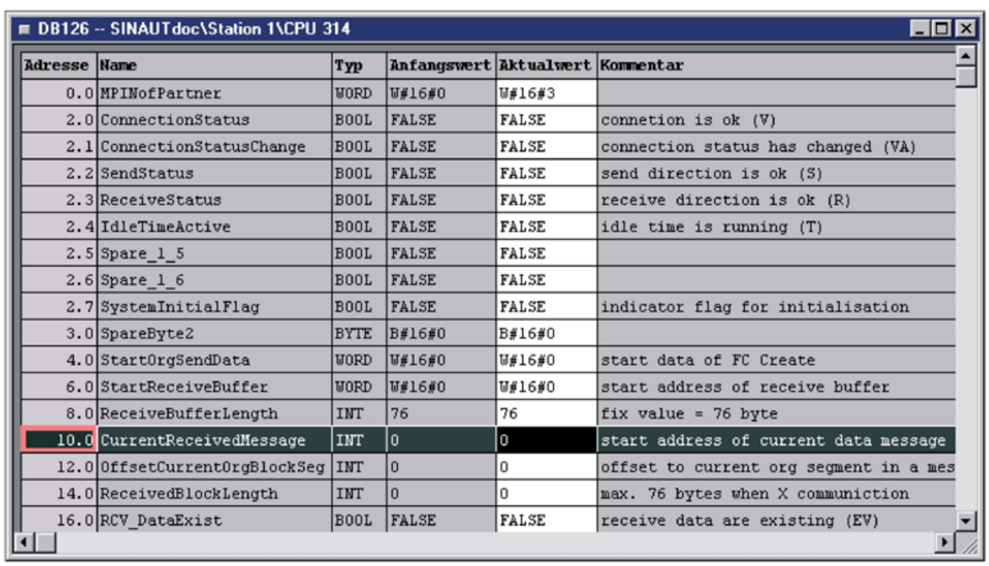
How a frame located in the receive mailbox is structured can be found in a separate description.



DB127 -- SINAUTdoc\Station 1\CPU 314

42.0	OffsetSendOrgIndex0	INT	48	48	offset of bytes to the send organizati
44.0	StartConnectionDescrList	WORD	W#16#0	W#16#0	start of the connection description li
46.0	ConnectionDescrLength	INT	14	14	length of a connection description
48.0	NofConnectionDescription	INT	0	1	number of connection descriptions
50.0	DB_ObjectList1	INT	0	0	DB number of objectlist 1
52.0	NofObjectItems1	INT	0	0	number of object items of objectlist 1
54.0	DB_ObjectList2	INT	0	0	DB number of objectlist 2
56.0	NofObjectItems2	INT	0	0	number of object items of objectlist 2
58.0	FC_User	INT	0	0	number of the user FC
60.0	CurrentComDB	INT	0	0	current communication DB
62.0	SystemInitialFlag	BOOL	FALSE	FALSE	system initialisation starts when flag
62.1	SystemStart	BOOL	FALSE	FALSE	system start
62.2	SystemStartEdgeFlag	BOOL	FALSE	FALSE	system start edge flag
62.3	NoSubscrObjectDetected	BOOL	FALSE	FALSE	no subscriber objet entry in subscribe
62.4	PLC_TypeSeries400	BOOL	FALSE	FALSE	PLC type is from the 400 series
62.5	SystemFlag1_5	BOOL	FALSE	FALSE	
62.6	SystemFlag1_6	BOOL	FALSE	FALSE	
62.7	SystemFlag1_7	BOOL	FALSE	FALSE	

Figure 5-25 DB-BasicData, DW60 CurrentComDB number of the current communication DB



DB126 -- SINAUTdoc\Station 1\CPU 314

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	MPINofPartner	WORD	W#16#0	W#16#3	
2.0	ConnectionStatus	BOOL	FALSE	FALSE	connetion is ok (V)
2.1	ConnectionStatusChange	BOOL	FALSE	FALSE	connection status has changed (VA)
2.2	SendStatus	BOOL	FALSE	FALSE	send direction is ok (S)
2.3	ReceiveStatus	BOOL	FALSE	FALSE	receive direction is ok (R)
2.4	IdleTimeActive	BOOL	FALSE	FALSE	idle time is running (T)
2.5	Spare_1_5	BOOL	FALSE	FALSE	
2.6	Spare_1_6	BOOL	FALSE	FALSE	
2.7	SystemInitialFlag	BOOL	FALSE	FALSE	indicator flag for initialisation
3.0	SpareByte2	BYTE	B#16#0	B#16#0	
4.0	StartOrgSendData	WORD	W#16#0	W#16#0	start data of FC Create
6.0	StartReceiveBuffer	WORD	W#16#0	W#16#0	start address of receive buffer
8.0	ReceiveBufferLength	INT	76	76	fix value = 76 byte
10.0	CurrentReceivedMessage	INT	0	0	start address of current data message
12.0	OffsetCurrentOrgBlockSeg	INT	0	0	offset to current org segment in a mes
14.0	ReceivedBlockLength	INT	0	0	max. 76 bytes when X communiction
16.0	RCV_DataExist	BOOL	FALSE	FALSE	receive data are existing (EV)

Figure 5-26 Current communication DB, DW10 CurrentReceivedMessage,
pointer to the start of the current received message in the receive mailbox

5.4.3 DB BasicData

This data block handles the central data storage; in other words, information that needs to be kept at a central location for all blocks. Among other things, the data block includes the subscriber records and the connection descriptions.

DB BasicData is automatically created with the necessary length, assigned the data specific to the subscribers and connections and then saved in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool 'Subscriber Administration' and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

DB BasicData replaces, among other things, the data blocks known from the TD1 package, ’stations record’, ’control center record’, ’interfaces record’ and the TD1 system bit memory address area from MB142 to MB199.

DB BasicData is available once on every CPU.

Note

In the SINAUT TD7 library, DB BasicData has the number DB127 and is also saved under this number in the CPUs when it is created for them. In principle, it would be possible to change the number but this requires a great deal of effort and may lead to errors when creating the rest of the SINAUT program. We therefore recommend: Leave DB127 free for DB BasicData if at all possible!

5.4.4 FB XCom

Auxiliary block for FC BasicTask, for processing a communication buffer of type DB XComData, in which a unconfigured connection (X connection) is handled using the SFCs X_SEND and X_RCV.

FB XCom also ensures that received messages are distributed immediately to the corresponding receive objects in the CPU. To do this, FB XCom calls FC Distribute as an auxiliary block.

In a normally configured SINAUT installation, FC XCom is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active. During compilation, there is a check to establish the CPU for which the program is being compiled. If an S7-300 CPU is involved, the FB-XCom is entered in the block directory of the CPU. With an S7-400 CPU, this is FB-BCom.

5.4.5 DB XComData

Instance data block for the communication block FB XCom. This communication DB contains a receive and a send buffer as well as central data required for the control and management of the X connection handled by this DB.

The data block is required in every CPU in which FB XCom is used. It can be inserted several times if the CPU maintains several X connections.

DB XComData is automatically created with the necessary length, assigned the data specific to the subscribers and connections and then saved in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

5.4.6 FB-PCom

Auxiliary block for FC BasicTask, for processing a communication buffer of type DB-PComData using SFCs WR_REC and RD_REC. Received messages are then distributed immediately to the relevant receive objects in the CPU. This is achieved by FB-PCom calling FC Distribute as an auxiliary block.
The FB-PCom block is used only with normal communication over the P bus. This relates to communication between a TIM module and a CPU module of the type series CPU 317 and CPU 319 and the CPU 315-2PN/DP.

5.4.7 DB PComData

Instance data block for the communication block FB PCom. This communication DB contains a receive and a send buffer as well as central data required for the control and management of the connection handled by this buffer.
The data block is required in every CPU in which FB PCom is used. It can be inserted several times if the CPU maintains several such connections.

5.4.8 FB BCom

Auxiliary block for FC BasicTask, for processing a communication buffer of type DB BComData, in which a configured connection (communication block connection) is handled using the SFBs BSEND and BRCV.

FB BCom also ensures that received messages are distributed immediately to the corresponding receive objects in the CPU. To do this, FB BCom calls FC Distribute as an auxiliary block.

In a normally configured SINAUT installation, FC BCom is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active. During compilation, there is a check to establish the CPU for which the program is being compiled. If an S7-400 CPU is involved, the FB BCom is entered in the block directory of the CPU. With an S7-300 CPU, this is FB XCom.

5.4.9 DB BComData

Instance data block for the communication block FB BCom. This communication DB contains a receive and a send buffer as well as central data required for the control and management of the communication block connection handled by this DB.

The data block is required in every CPU in which FB BCom is used. It can be inserted several times if the CPU maintains several communication block connections.

DB BComData is automatically created with the necessary length, assigned the data specific to the subscribers and connections and then saved in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

5.4.10 FC Create

Auxiliary block for putting together messages and entering them in one or more relevant send buffers. This block is required by the data point typicals for data and organizational messages and by FC BasicTask only for organizational messages.

This block is required in every CPU.

In a normally configured SINAUT installation, FC Create is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

5.4.11 FC Distribute

Auxiliary block for distributing messages from the receive buffer to the appropriate data point typicals or to the subscriber objects in the subscriber records.

This block is required in every CPU.

In a normally configured SINAUT installation, FC Distribute is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

5.4.12 FC Search

Auxiliary block for searching:

- For the initial address of a subscriber object within the subscriber records
- 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. This block is required in every CPU.

In a normally configured SINAUT installation, FC Search is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

5.4.13 FC Diagnose

Auxiliary block for entering SINAUT system messages in the diagnostic buffer of the CPU.

This block is required in every CPU.

In a normally configured SINAUT installation, FC Diagnose is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

5.5 Data point typicals

Introduction

Data point typicals consist of one or more data points of the same type, e.g. 4 bytes of binary information, or 4 analog values, or 1 byte commands, etc.

Data point typicals for a specific type or amount of information always come in two versions:

- A typical for acquiring and sending
  e.g. FB Bin04B_S, acquire and send 4 bytes of binary information (messages);
- A typical for receiving and outputting
  For example, FB Bin04B_R, for receiving and outputting 4 bytes of binary information (status signals).

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. At the send and receive ends this object number does not have to be identical.

Structure of typical names

Data point typicals have 8-character names based on 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">Column 1 ... 3
Data point type:</td><td colspan="2" rowspan="8">Column 4 ... 5
Amount of data in the format of column 6.</td><td rowspan="8">Data format:
X = bit
B = byte
W = word
D = double word
R = real</td><td rowspan="8">Not used
(filled by under-score)</td><td rowspan="8">S = send function

R = re-ceive function</td></tr>
<tr><td>Bin</td><td colspan="2">Binary information (status signal, alarm)</td></tr>
<tr><td>Ana</td><td colspan="2">Analog value</td></tr>
<tr><td>Cnt</td><td colspan="2">Counted value</td></tr>
<tr><td>Cmd</td><td colspan="2">Command</td></tr>
<tr><td>Set</td><td colspan="2">Setpoint (setpoint, parameter)</td></tr>
<tr><td>Par</td><td colspan="2">Parameter</td></tr>
<tr><td>Dat</td><td colspan="2">Data (any mixture of in-formation types)</td></tr>
</table>

The table above relates to the name structure for ST7 typicals.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

Overview of the available data point typicals

The following table provides an overview of the currently available ST7 data point typicals for data exchange between two SINAUT ST7 subscribers.

Table 5- 4 Overview of the available data point typicals

SINAUT protocol	Symbolic FB name	Explanation
Message typicals		
ST7	Bin04B_S	ST7 status message object, send 4 bytes of status/binary information
	Bin04B_R	ST7 status message object, receive 4 bytes of status/binary information
Analog value typicals		
ST7	Ana04W_S	ST7 analog value object, send 4 analog values (16-bit value in the INT format)
	Ana04W_R	ST7 analog value object, receive 4 analog values (16-bit value in the INT format)
Counted value typicals		
ST7	Cnt01D_S	ST7 counted value object, send 1 counted value (32-bit ST1 format)
	Cnt01D_R	ST7 counted value object, receive 1 counted value (32-bit ST1 format)
	Cnt04D_S	ST7 counted value object, send 4 counted values (32-bit ST1 format)
	Cnt04D_R	ST7 counted value object, receive 4 counted values (32-bit ST1 format)
Command typicals		
ST7	Cmd01B_S	ST7 command object, send 1 byte commands (1-out-of-8 ST1 format)
	Cmd01B_R	ST7 command object, receive 1 byte commands (1-out-of-8 ST1 format)
Setpoint/parameter typicals		
ST7	Set01W_S	ST7 setpoint object, receive 1 setpoint (16 bits) and receive current local setpoint
	Set01W_R	ST7 setpoint object, receive 1 setpoint (16 bits) and send current local setpoint
	Par12D_S	ST7 parameter object, send max. 12 double words with parameters and receive current local parameters.
	Par12D_R	ST7 parameter object, receive max. 12 double words with parameters and send current local parameters.
Other data typicals		
ST7	Dat12D_S	ST7 data object, send max. 12 double words with any information.
	Dat12D_R	ST7 data object, receive max. 12 double words with any information.
	Dat256D_S	ST7 data object, send max. 256 double words with any information.

SINAUT protocol	Symbolic FB name	Explanation
	Dat256D_R	ST7 data object, receive max. 256 double words with any information.

Notes on the SINAUT Time stamp

For many data point typicals you can use the TimeStamp parameter to instruct that the data object should be transferred with a time stamp. However for the data point typicals used to receive this data 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. This occurs in two data double words that always have the same name in all instance DBs (SINAUT object DB), namely:

Name of the double word	Contents
RecTimeStamp_1	Year, month, day and hour
RecTimeStamp_2	Minute, second, millisecond and time status

The date and time of day are coded in BCD format (exception: the half byte with the time status). The exact byte-for-byte content appears as follows:

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

The content of the half byte with the time status bit:

Bit place	Value	Meaning
2^0	0	Time is invalid
	1	Time is valid
2^1	0	Standard time
	1	Daylight saving time
2^2		Not used
2^3		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
```

Notes on *explanation of the parameters*

The detailed descriptions of the blocks in most of the following contain an *"Explanation of the parameters"* section. The following information is available there for each parameter:

Name:	**Name of the parameter**
Declaration:	SIMATIC parameter type INPUT, OUTPUT or IN_OUT
Data type:	SIMATIC data type The data types used: BOOL, BYTE, WORD, DWORD, INT, DINT, ANY, COUNTER and BLOCK_DB
Default	(only applicable for FB parameters) Default value for the parameter. This value is valid when the parameter is not specified when the FB is called.
Explanation:	Detailed description of the parameter and specification of the allowable value range.

Note

The data point typicals are described in detail in the following pages. Data point typicals that are identical except for the number of data points to be processed are included in a single description.

If there is a difference in the function or in the parameters, the block for which the described function or parameter is valid is shown in square brackets, for example, [Cnt01D_S].

5.5.1 Parameters you require often

Parameters you require often

Before coming to the others, the next section describes commonly recurring parameters used by several of the data point typicals.

PartnerNo

Name:	**PartnerNo**
Declaration:	INPUT
Data type	INT
Default:	0
Explanation	Subscriber no. of the partner

Range of values: 0 or 1 ... 32000

The subscriber number of the partner with which the FB communicates, i.e. to which the FB sends data or from which the FB receives data must be specified.

For a process typical, this is usually the subscriber no. of the master PLC or the ST7cc control center.

With an operator typical, this is normally the subscriber number of a station PLC.

Points to note when PartnerNo = 0 in various typical classes:

- **Sending process typicals**
  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S)

  When the parameter is set to 0, the data is transmitted to all subscribers for which a connection has been configured. In this case, the following parameter PartnerObjectNo is automatically sent with zero by the process typical.

  If the set PartnerNo was not found in the administration (in DB-BasicData), an entry to this effect is made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

  Note
  If the PartnerObjectNo is missing, there must be a list on the partner PLC from which the missing object number can be recognized (see FC ListGenerator300, FC ListGenerator400 (Page 364)).

  Note
  When using the block in the PLC of a node station, you should consider the consequences of PartnerNo = 0 If the PLC of the node station maintains both connections to higher-level subscribers as well as to lower-level stations, a message with PartnerNo = 0 is transferred to all subscribers both "up" and "down".

- **Receiving process typicals**
  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Dat12D_R, Dat256D_R)

  The parameter setting PartnerNo = 0 is not permitted!

  If the parameter setting is incorrect (< 1 or > 32000), an error message to this effect is entered in the diagnostic buffer (event ID B100). If the value range is correct, but the PartnerNo was not found in the administration (in DB-BasicData), an entry is also made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

  If the PLC receives a message for the object set here, the system checks whether the source subscriber number in the message is identical to the PartnerNo set here. If they are different, the received information is discarded. An error message to this effect is entered in the diagnostic buffer (event ID B130).

- **Sending process typicals**
  (Cmd01B_S, Set01W_S, Par12D_S)

  The parameter setting PartnerNo = 0 is not permitted!

  If the parameter setting is incorrect (< 1 or > 32000), an error message to this effect is entered in the diagnostic buffer (event ID B100). If the value range is correct, but the PartnerNo was not found in the administration (in DB-BasicData), an entry is also made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

- **Receiving process typicals**
  (Cmd01B_R, Set01W_R, Par12D_R)

  Set the parameter to 0 when the typical can receive data from more than one partner, for example, when there are several control centers wanting to send data to the typical configured here.

  If the PLC receives a message for the object set here, and PartnerNo is greater than 0, the system checks whether the source subscriber number in the message is identical to the PartnerNo set here. If they are different, the received information is discarded. An error message to this effect is entered in the diagnostic buffer (event ID B130).

  This check is not made if PartnerNo = 0. Regardless of the sender, each message addressed to the object is also passed on to the object.

  If the set PartnerNo is greater than 0 and this number was 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 FB is then no longer processed, however, until the parameter assignment error has been corrected.

  Caution
  If "PartnerNo = 0" is set, make sure that each partner sends the message with a complete destination address (target subscriber no. and target object no.).

  Note
  When using the block in the PLC of a node station, you should consider the consequences of PartnerNo = 0 If the PLC of the node station maintains both connections to higher-level subscribers as well as to lower-level stations, a message with PartnerNo = 0 is transferred to all subscribers both "up" and "down".

PartnerObjectNo

Name:	**PartnerObjectNo**
Declaration:	INPUT
Data type	INT
Default:	0
Explanation	Object number of the partner
	Range of values: 0 or 1 ... 32000
	The number of the object (= DB number) on the partner with which the FB communicates, i.e. to which the FB sends data or from the FB receives data, must be specified.
	Point to note when PartnerObjectNo = 0 in various typical classes:

- **Sending process typicals**
  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S)

  Setting the parameter to 0 is useful, if PartnerNo = 0 was set. If the PartnerObjectNo is missing, there must be a list on the partner PLC from which the missing object number can be recognized (see FC ListGenerator).

  If the subscriber specified by PartnerNo is an ST7cc control center, the PartnerObjectNo does not need to be specified in the FB because there are no DBs as destination objects in ST7cc as there are in a CPU. ST7cc decodes its messages solely based on the source address in the message.

- **Receiving process typicals**
  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Dat12D_R, Dat256D_R)

  The parameter setting PartnerObjectNo = 0 is not permitted!

  If the parameter setting is incorrect (< 1 or > 32000), an error message to this effect is entered in the diagnostic buffer (event ID B102). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

  If the PLC receives a message for the object set here, the system checks whether the source object number in the message is identical to the PartnerObjectNo set here. If they are different, the received information is discarded. An error message to this effect is entered in the diagnostic buffer (event ID B131).

- **Sending process typicals**
  (Cmd01B_S, Set01W_S, Par12D_S)

  The parameter setting PartnerObjectNo = 0 is not permitted!

  If the parameter setting is incorrect (< 1 or > 32000), an error message to this effect is entered in the diagnostic buffer (event ID B102). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

- **Receiving process typicals**
  (Cmd01B_R, Set01W_R, Par12D_R)

  The parameter setting PartnerObjectNo = 0 is necessary in the following situations:
  - The partner is not an S7-CPU; in other words, there is no object = DB number. This is, for example, the case when the partner is an ST7cc control center.
  - There is more than one partner (PartnerNo = 0) that wants to send data to this typical. The corresponding objects of these partners will then generally have different numbers; in other words, no unique number can be specified here.

  If the PLC receives a message for the object set here, and PartnerObjectNo is greater than 0, the system checks whether the source object number in the message is identical to the PartnerObjectNo set here. If they are different, the received information is discarded. An error message to this effect is entered in the diagnostic buffer (event ID B131).

  This check is not made if PartnerObjectNo = 0. Regardless of the sender object, each message addressed to the object is also passed on to the object.

Enabled

Name:	**Enabled**
Declaration:	INPUT
Data type	BOOL
Default:	TRUE
Explanation	Enable block processing

Range of values:

TRUE or FALSE	
No parameter specified: Default TRUE is valid	
Input	I 0.0 ... I n.7
Bit memory	M 0.0 ... M n.7
	L 0.0 ... L n.7
Data bit	DBm.DBX 0.0 ... n.7

Whether or not processing of the block is enabled must be specified.

If processing is enabled, all the functions of the FB execute.

The response is different if processing has not been enabled:

- **Processing of process typicals not enabled**
  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S, Dat256D_S)

  If processing is not enabled, the FB can only communicate at the organizational level; in other words, ORG messages can be sent and received. A query is, for example, answered, however the reply message contains the data valid at the time the function was disabled.

  Note
  The response described here does not apply to Cmd01B_R (see ST7 command typical FB Cmd01B_R (Page 318))!

- **Processing of operator typicals not enabled**
  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Set01W_S, Par12D_S, Dat12D_R, Dat256D_R)

  If processing is not enabled, the FB can only communicate at the organizational level; in other words, ORG messages can be sent and received. A request can, for example, still be sent and the answer received, the received information is, however, not output to the outputs.

  You will find the relevant outputs under "Enable" in the description of the individual data point typicals.

  Note
  The response described here does not apply to Cmd01B_S (see ST7 command typical FB Cmd01B_S (Page 316))!

ImageMemory

Name:	**ImageMemory**		
Declaration:	INPUT		
Data type	BOOL		
Default:	TRUE		
Explanation	Image memory principle for spontaneous data transmission		
	Range of values:	TRUE or FALSE	
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value TRUE is valid.	

You need to specify whether or not the message will be transmitted according to the image memory principle or the send buffer principle.

The image memory method reduces the memory required for storing messages on the TIM and produces as little message traffic on the WAN as possible. The default TRUE was selected because the image memory principle is the best choice in practice for most data transmissions.

In general, as the user you only need to change the default setting of the image memory parameter with a few objects, namely objects whose data changes must be stored on the TIM and sent to the partner singly, for example alarms with time stamp.

The parameter is used by the following process typicals:
Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S, Dat256D_S

Conditional

Name:	**Conditional**		
Declaration:	INPUT		
Data type	BOOL		
Default:	TRUE		
Explanation	Conditional spontaneous data transmission		
	Range of values:	TRUE or FALSE	
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value TRUE is valid.	

You will find information on the parameter assignment in the Unconditional parameter.

The parameter is used by the following process typicals:
Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S, Dat256D_S

Unconditional

Name:	**Unconditional**		
Declaration:	INPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Unconditional spontaneous data transmission.		
	Range of values:	TRUE or FALSE	
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7

No parameter specified: Default value TRUE is valid.

Note on the use of the Conditional and Unconditional parameter settings:
With the two parameters Conditional and Unconditional, you can decide whether a message is transmitted by the TIM immediately when data changes or at a later point in time.

1. If the transmission does not need to be made immediately, set the parameters as follows:
   Conditional = TRUE
   Unconditional = FALSE
2. If you require immediate transmission, the parameter combination should be:
   Conditional = FALSE
   Unconditional = TRUE

The decision for immediate or later transmission only relates to dial-up networks. On a dedicated line, the transmission is always immediate even if the combination of Conditional and Unconditional is set to "not immediately".
The default of the two parameters was chosen so that a message is not transmitted immediately (combination 1). On dedicated lines, you as the user do not need to make changes to the two parameter settings. Only in a dial-up network, do you need to decide which objects are so important that an immediate transmission is necessary if there is a change in the acquired data for the object. Only then do you need to change Conditional to FALSE and Unconditional to TRUE, for example for an object with alarms.

The parameter is used by the following process typicals:
Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S, Dat256D_S

Permanent

Name:	**Permanent**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Permanent data transmission.
	This parameter has no significance. The functionality of permanent data transmission is not supported by the TIM.

Note

The "Permanent" parameter is no longer implemented, it has been retained to ensure compatibility.

TimeStamp

Name:	**TimeStamp**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Time stamp Range of values: TRUE or FALSE Here, you specify whether the message is transferred with the time stamp. This is only possible if the time provided by the local TIM is available on the PLC. For more detailed information, refer to the description of FC TimeTask. If no parameter is specified, the default is FALSE; in other words, data is transmitted without a time stamp. The following data point typicals use the parameter: Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S, Set01W_R, Par12D_R **Note** With FB-Ana04W_S, remember that the parameter depends on the "Mean-ValueGeneration" parameter (seeST7 analog value typical FB Ana04W_S (Page 300)).

NewData

Name:	**NewData**		
Declaration:	OUTPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Receive new data.		
	Range of values:	Output	Q 0.0 ... Q n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified	Default value FALSE is valid

The NewData output is intended for user-specific further processing, for example to react in a specific way to receipt of new data.

The following data point typicals use the parameter:
Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Cmd01B_R, Set01W_S, Set01W_R, Par12D_S, Par12D_R, Dat12D_R

Whenever the FB has received new data and has output it to the outputs for the specific typical, the NewData output is set to TRUE for one OB1 cycle. You will find the relevant outputs for the specific typical under "NewData" 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 entry in the Local = 1 state.

If you do not require the parameter, simply leave it open.

5.5.2 ST7 binary information typical FB Bin04B_S

Function

Send 4 bytes of messages/binary information

Explanation of the parameters

Names:	**PartnerNo** **PartnerObjectNo** **Enabled** **ImageMemory** **Conditional** **Unconditional** **Permanent** **TimeStamp**

Refer to Parameters you require often (Page 287).

Name:	**InputByte_1 … _4**		
Declaration:	INPUT		
Data type	BYTE		
Default:	0 (B#16#0)		
Explanation	Input byte		
	Range of values:	Input byte	IB0 ... IBn PIB0 ... PIBn
		Memory bytes	MB0 ... MBn LB0 ... LBn
		Data bytes	DBm.DBB0 ... n
		No parameter specified: Default value 0 is valid.	

You can specify the bytes from where the binary information such as status messages, alarms etc. is taken by the FB to be transferred in the data messages. 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 as required.

If you do not require parameters, simply leave them open. The value 0 is transferred for these bytes in the message.

Name: **DisableMask**

Declaration: INPUT

Data type DWORD

Default: 0 (2#0)

Explanation Disable mask

Range of values:
as 32-bit binary number
2#0 ... 2#11111111_11111111_11111111_11111111
as 32-bit hex number
DW#16#0 ... DW#16#FFFF_FFFF
no parameter specified: Default value 0 (2#0) is valid

A 1 must be entered in the bit pattern that the position of each input to be disabled; 0 is entered for the other inputs. The assignment of the 32 inputs of the parameters InputByte_1 through InputByte_4 to the 32 bits in the bit pattern of the DisableMask parameter is shown in the following table.

A disabled input always has the value 0 in the message.

	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#	_				_				_				_				_				_				_				_			

Name: **InversionMask**

Declaration: INPUT

Data type DWORD

Default: 0 (2#0)

Explanation Inversion mask

Range of values:
as 32-bit binary number
2#0 ... 2#11111111_11111111_11111111_11111111

As 32-bit binary number

DW#16#0 ... DW#16#FFFF_FFFF

No parameter specified: Default value 0 (2#0) is valid.

A 1 must be entered in the bit pattern that the position of each input to be inverted; 0 is entered for the other inputs. The assignment of the 32 inputs of the parameters InputByte_1 through InputByte_4 to the 32 bits in the bit pattern of the InversionMask parameter is shown in the following table.

The inversion of input signals can, for example, be useful when using a mixture of sensors operating on the open and closed circuit principle.

	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#	_				_				_				_				_				_				_				_			

5.5.3 ST7 binary information typical FB Bin04B_R

Function

Receive 4 bytes of status/binary information

Explanation of the parameters

Names: **PartmerNo**
PartmerObjectNo
Enabled

Refer to Parameters you require often (Page 287).

Name:	**OutputByte_1 … _4**		
Declaration:	OUTPUT		
Data type	BYTE		
Default:	0 (B#16#0)		
Explanation	Output byte		
	Range of values:	Output byte	QB0 ... QBn PQB0 ... PQBn
		Memory bytes	MB0 ... MBn LB0 ... LBn
		Data bytes	DBm.DBB0 ... n

Where the binary information such as status messages, alarms etc is to be output can be selected byte by byte. Output bytes in the process output image, I/O bytes directly on digital output modules, data bytes of a data block and memory bytes can be mixed as required.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require parameters, simply leave them open.

Name: **NewData**

Refer to Parameters you require often (Page 287).

Explanation of the typical-specific outputs	Whenever the FB has received new data and has output it to the output bytes OutputByte_1 through OutputByte_4, the NewData output is set to TRUE for one OB1 cycle.

5.5.4 ST7 analog value typical FB Ana04W_S

Function

Send 4 analog values (16-bit value in the INT format).

FB Ana04W_S transfers the 4 analog values:

- As instantaneous values
  At the time of the transmission, the currently pending analog value is acquired and transferred to the partner.
  or
- 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

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 can be processed in the same way.

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional
Permanent

See Parameters you require often (Page 287).

Name: **TimeStamp**

Declaration, data type , default and range of values: See Parameters you require often (Page 287).

Explanation Time stamp

Here, you specify whether the message is transferred with the time stamp. The prerequisite is that the time provided by the local TIM is available on the PLC. For more detailed information, refer to the description of FC TimeTask.

The following applies to the time stamp in the message;

- MeanValueGeneration = FALSE (instantaneous values are transmitted in the message)
  The time stamp in the message is identical to the time of acquisition of the instantaneous values contained in the message.
- MeanValueGeneration = TRUE (the message 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 message. This is, however, identical to the time stamp of the previously transferred mean value message.

If no parameter is specified, the default is FALSE; in other words, data is transmitted without a time stamp.

Name: **ThresholdIntegration**

Declaration: INPUT

Data type BOOL

Default: FALSE

Explanation Threshold value processing according to the integration principle.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

With this parameter, you can specify whether the integration principle is used in threshold value processing.

If no parameter is specified, the default is FALSE; in other words, threshold values are processed without integration. This corresponds to the previous ST1 procedure. In this case, you can also expect less traffic on the telecontrol line and locally between CPU and TIM (over the MPI bus or party line).

Note

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the ThresholdIntegration parameter has no meaning.

Name: **ZeroLimitation**

Declaration: INPUT

Data type BOOL

Default:	TRUE
Explanation	Zero limitation.

Range of values:	TRUE or FALSE No parameter specified: Default value TRUE is valid.

This parameter allows you to specify whether negative values should be suppressed and replaced with the value 0.

If no parameter is specified, the default is TRUE. This means that the lowest value is limited to 0.

Name:	**TriggerInput**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Trigger input.

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid.	

If required, this parameter can be used to specify an input over which the user can trigger the transmission *) of the analog value message at any time (signal edge from 0 to 1).

Example:
Time-driven analog value transmission with time stamp for supplying an analog value archive in the control center. Note: To prevent these messages with time stamps from being overwritten when saving on the station TIM, the ImageMemory parameter must be set to FALSE.

If the block calculates mean values, the duration of the calculation period is defined 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 two message triggers therefore determines the duration of the mean value calculation period.

FC Trigger can be used for time-driven triggering of a transmission over TriggerInput (for more detailed information, refer to the description of this block).

If you do not require the parameter, simply leave it open. Message transmission should then be triggered based on the ThresholdValue and Threshold-Integration threshold parameters.

*) TriggerInput actually only triggers transmission indirectly. With a 0/1 edge at TriggerInput, the message 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. Transmission is immediate over a dedicated line/wireless link; with a dial-up connection, it is possible that the message is saved first on the TIM and sent at a later point in time (for example, because the message is marked as a "conditional spontaneous" message; see the Conditional parameter).

Name:	**MeanValueGeneration**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Mean value generation.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

With this parameter, you can specify whether 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 over 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 message 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 over 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 allowed range results in an immediate cancelation of the mean calculation. The value 7FFFH or 8000H is saved as an invalid mean value for the current mean value calculation period and sent when the next analog value message is triggered. The calculation of a new mean value is then started. If the analog value remains above or below the permitted range, this new value is again saved immediately as an invalid value and sent when the next message is triggered.

FaultSuppressionTime > 0
If the acquired analog value is above or below the permitted range, the bad values are excluded from the calculation of the mean value for a maximum duration as defined by the FaultSuppressionTime. If this period is exceeded, the value 7FFFH or 8000H is saved as an invalid mean value and sent when the next analog value message is triggered. This procedure is repeated in the new mean value calculation period; in other words, bad values are suppressed once again for the duration of the FaultSuppressionTime.
The FaultSuppressionTime period allows you to indirectly determine the percentage of invalid values for each mean value calculation period. For example, if the mean 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 or 33% of the analog values acquired are above or below the permitted range in the current mean value calculation period.

If no parameter is specified, the default is FALSE; in other words, instantaneous values are acquired and transmitted.

Name:	**AnalogInput_1 ... _4**
Declaration:	INPUT
Data type	WORD
Default:	0 (W#16#0)
Explanation	Analog input word.

Range of values:	I/O words	PIW0 ... PIWn
	Memory words	MW0 ... MWn
		LW0 ... LWn
	Data words	DBm.DBW0 ... n

For each analog value to be transmitted in the data message, 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 message.

Name: **SamplingPeriod**

Declaration: INPUT

Data type INT

Default: 500

Explanation Acquisition interval for analog inputs in ms.

Range of values:	0 ... 32767 [ms]
	No parameter specified: Default value 500 ms is valid.

The acquisition interval is required for the following parameters:

- For the processing of the threshold value according to the integration principle (threshold integration)
- For smoothing the analog input value (SmoothingFactor)
- For calculating the mean values (MeanValueGeneration)

The value must be selected high enough so that it is certain that a new value was acquired over 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).

If no parameter is specified, the default of 500 ms applies. This time is high enough to be applied even at the highest resolution and for analog modules with the maximum 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 internal accumulation counter (must not exceed max. value 2,147,483,647 of a double integer). When an overflow is detected, the invalid mean value of 8000H is transmitted for the current mean value calculation period.

Name: **ThresholdValue**

Declaration: INPUT

Data type INT

Default:	270	
Explanation	Threshold value.	
	Range of values:	0 or 1 ... 32767 No parameter specified: Default 270 is valid (corresponds to 1%).

The encoding range of the analog value must be taken into consideration when setting the threshold value. 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, the threshold value should be entered according to this.

If no parameter is specified, the default of 270 applies. 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 message will only be sent in the following situations:

1. When there is a trigger over the TriggerInput input, typically a time-driven or event-driven message trigger.
2. When there is a general request to the station or a single request for the message.
3. When the analog value moves into the overflow or underflow range (7FFFH or 8000H) (possibly after the suppression time set by FaultSuppressionTime has elapsed).

Note

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the ThresholdValue parameter has no meaning.

Name:	**SmoothingFactor**
Declaration:	INPUT
Data type	INT
Default:	1
Explanation	Smoothing factor.

Range of values:	
	1 (no smoothing)
	4 (weak smoothing)
	32 (medium smoothing)
	64 (strong smoothing)
	No parameter specified: Default 1 (no smoothing) is valid.

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 set (see ThresholdValue).

The smoothing factors are identical to the smoothing factors that are configured for some S7 analog input modules. The smoothing is handled in typical using the same formula as for input modules, described by the following:

$$y_n = \frac{x_n + (k-1)\,y_{n-1}}{k}$$

y_n = smoothed value in the current cycle n

y_n = acquired value in the current cycle n

k = smoothing factor

Note

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the SmoothingFactor parameter has no meaning.

Name:	**FaultSuppressionTime**
Declaration:	INPUT
Data type	INT
Default:	0
Explanation	Fault suppression time in seconds.

Range of values:	0 ... 32767 [s] No parameter specified: Default value 0 s is valid.

Transmission of an analog value located in the overflow or underflow range (7FFFH or 8000H) is suppressed for the time period specified here. The value 7FFFH or 8000H is only sent after this time has elapsed, if it is still pending. If the value returns to below 7FFFH or above 8000H 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 7FFFH or 8000H 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 several times the maximum range 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). When the user provides specific values, fault suppression is only possible if these also adopt the values 7FFFH or 8000H when the permitted ranges are exceeded. 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 temporary suppression of the values 7FFFH or 8000H (see description of the MeanValueGeneration) parameter.

When no parameter is specified, the default value of 0 seconds applies. An acquired value of 7FFFH or 8000H 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.

5.5.5 ST7 analog value typical FB Ana04W_R

Function

Receive 4 analog values (16-bit value in the INT format).

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled

See Parameters you require often (Page 287).

Name:	**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.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require parameters, simply leave them open.

Name: **NewData**

See Parameters you require often (Page 287).

Explanation of the typical-specific outputs	Whenever the FB has received new data and has output it to the outputs AnalogOutput_1 through AnalogOutput_4, the NewData output is set to TRUE for one OB1 cycle.

5.5.6 ST7 counted value typicals FB Cnt01D_S and FB Cnt04D_S

Function

[Cnt01D_S] send 1 counted value (32-bit ST1 format).

[Cnt04D_S] send 4 counted values (32-bit ST1 format).

Explanation of the parameters

Names:
PartnerNo
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional
Permanent
TimeStamp

Refer to Parameters you require often (Page 287).

Name: **GeneralTriggerCommand**

Declaration: INPUT

Data type	BOOL		
Default:	FALSE		
Explanation	General restore command:		
	Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.	

Here, you specify whether or not a counted value transmission should be triggered by a general restore command (the general restore command belongs to the organizational SINAUT system commands).

If an explicit destination subscriber number (PartnerNo > 0) is assigned in the typical, the general restore command is evaluated in the corresponding subscriber object in the central administration. If the destination subscriber number is missing (PartnerNo = 0, send 'to all'), the central system memory bit 'General restore command' is taken into account.

When the general restore command is detected, the currently accumulated counted value is transmitted regardless of other criteria that affect message transmission. The restore identifier US is inverted in this counted value.

Parameters GeneralTriggerCommand and TriggerInput can be used together. Transmission is then triggered by a signal edge change from 0 to 1 at the TriggerInput as well as when a general restore command is received.

If no parameter is specified, the default is FALSE; in other words, there is no restore using the general restore command.

Name:	**TriggerInput**		
Declaration:	INPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Trigger input		
	Range of values:	TRUE or FALSE	
		No parameter specified: Default value FALSE is valid	
		Input	I 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7

If required, this parameter can be used to specify an input over which the user can trigger (signal edge change from 0 to 1) a transmission at any time regardless of other criteria that affect message transmission. The currently accumulated counted value is transmitted. The restore identifier US is inverted in this counted value.

Note

FC Trigger is an easy way to trigger time-driven transmission of a counted value message. For more detailed information, refer to the description of the FC.

Parameters TriggerInput and GeneralTriggerCommand can be used together. Transmission is then triggered by a signal edge change from 0 to 1 at the TriggerInput as well as when an organizational restore command is received.

If no parameter is specified, the default is FALSE; in other words, there is no restore and transmission triggered over the TriggerInput input.

Name:	**[Cnt01D_S] Counter_1** **[Cnt04D_S] Counter_1 ... _4**
Declaration:	INPUT
Data type	COUNTER
Default:	-
Explanation	Number of the SIMATIC counter.

Range of values:	C0 as dummy parameter or C1 ... Cn (n depending on CPU type)

Here, you specify the SIMATIC counter in which the pulses were counted time-driven. This takes place in the background using FC PulseCounter that is called in a cyclic interrupt OB, for example OB35. Refer to the description of FC PulseCounter and 'Time-driven SINAUT program in a cyclic interrupt OB'.

The COUNTER data type cannot be assigned a default value. If you have used the C0 dummy parameter in the typical, the corresponding counted value is not processed.

Name:	**DifferenceValue**
Declaration:	INPUT
Data type	INT
Default:	0
Explanation	Difference value.

Range of values:	0 or 1 ... 31768 No parameter specified: Default value 0 is valid.

When a value between 1 and 31768 is specified, the counted value is sent as soon as the difference between the current and most recently transmitted counted value reaches or exceeds the configured value.

If no parameter is specified, default value 0 applies; in other words, a counted value is only sent when a signal edge change from 0 to 1 is detected at the TriggerInput input , or when (if GeneralTriggerCommand = TRUE) an organizational. restore command is received.

Note

The difference value must be selected by the user depending on the maximum pulse rate per second. The value should not be too low otherwise there is a constant transfer of the message to the TIM. On one hand, this would cause heavy load on the MPI bus/party line but also stretch the send queue on the CPU.

Note

[Cnt04D_S]

The DifferenceValue processing parameter exists only once in the typical. This parameter applies to all 4 counted values in common; in other words, 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.

5.5.7 ST7 counted value typicals FB Cnt01D_R and FB Cnt04D_R

Function

[Cnt01D_R] receive 1 counted value (32-bit ST1 format).

[Cnt04D_R] receive 4 counted values (32-bit ST1 format).

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled

Refer to Parameters you require often (Page 287).

Name:	**BCD_Format**
Declaration:	INPUT
Data type	BOOL
Default:	TRUE
Explanation	Counted value output in BCD format.

Range of values:	TRUE or FALSE No parameter specified: Default value TRUE is valid.

Here, you specify the format in which counted value received at the output / outputs CountedValueOutput_... is output.

If the parameter is left open or if you specify TRUE, the counted value is output with a maximum of seven places in BCD (= S7 format; the sign decade is always 0; in other words, +). The maximum counted value that can be represented is therefore restricted to 9,999,999.

If you do not require BCD format (BCD_Format = FALSE), the counted value is output as a 32-bit integer and this is also always positive. The maximum counted value that can be represented is then 2,147,483,647.

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

[Cnt04D_R]

The BCD_Format processing parameter exists only once in the typical. This parameter applies to all 4 counted values in common; in other words, it is not possible to set the parameter for the individual counted values. When using this parameter, each typical should therefore only output counted values that can have parameters set for an identical output format.

Name:	**CntValInvalid**
Declaration:	OUTPUT
Data type	BOOL
Default:	FALSE
Explanation	Counted value invalid.

Range of values:	Output	Q 0.0 ... Q n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

At the CntValInvalid output, the FB indicates whether the last received counted value was invalid (with Cnt04D_R, this counts as a group display for all four counted values). In principle, the output shows the current status of the validity bit A from the most recently received counted value in inverted form.

On the one hand, the output serves as an error indicator. On the other, the output is intended for user-specific further processing. For example, the user may wish to react to invalidity by correcting the counted value at CountedValueOutput_... by adding counter pulses that may have been lost.

If you do not require the parameter, simply leave it open.

Note

[Cnt04D_R]

Although all 4 counted values in the message have their own validity bit, only the validity bit A of the first counted value in the last received message is evaluated for the status at the CntValInvalid output. This status, however, applies to all 4 counted values, since all the counted values in the message always have the same validity status.

Note

When evaluating CntValInvalid, you should take into account the fact that it might only be possible for this bit to be set for one OB1 cycle.

Name:	**RestoreStatus**		
Declaration:	OUTPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Current status of the restore bit US in the received counted value.		
	Range of values:	Output	Q 0.0 ... Q n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid.	
	At the RestoreStatus output, the FB indicates the current status of the restore bit US from the last received counted value message.		
	The output is intended for user-specific further processing. For example, the user may only wish to access the information at CountedValueOutput_... 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

[Cnt04D_R]

Although all 4 counted values in the message have their own restore bit, only the restore bit US of the first counted value in the last received message is evaluated for the status at the RestoreStatus output. This status, however, applies to all 4 counted values since they are always restored together.

Name: **NewData**

Refer to Parameters you require often (Page 287).

Explanation of the typical-specific outputs: Whenever the FB has received new data and has output it to the output / outputs CountedValueOutput_..., the NewData output is set to TRUE for one OB1 cycle.

Name: **[Cnt01D_R] CountedValueOutput_1**

[Cnt04D_R] CountedValueOutput_1 ... _4

Declaration: IN_OUT

Data type DWORD

Default: 0

Explanation Counted value output.

Range of values:	(process image) output double words	QD0 ... QDn
	Memory double words	MD0 ... MDn
	Data double words	DBm.DBD0 ... n

The counted value output is a double word in which the counted value is stored in BCD format or as a 32-bit integer (depending on the BCD_Format parameter). In BCD format, the maximum counted value that can be represented is limited to 9,999,999, if the counted value is output as a 32-bit integer, the maximum counted value that can be represented is 2,147,483,647.

The counted value is always output as a positive number; in other words, 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 (declaration IN_OUT), the value can be reset to 0 or another value at the counted value output by the user at any time. 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 that the counted value output.

Note

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.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

5.5.8 ST7 command typical FB Cmd01B_S

Function

Send 1 byte commands (1-out-of-8 ST1 format).

Note

With FB Cmd01B_S, data can only be transmitted when FC Safe is included at the end of the cyclic SINAUT program. See also 'The cyclic OB1 program for a master station' in section Cyclic SINAUT program in OB1 (Page 269).

Explanation of the parameters

Names: **PartnerNo**

PartnerObjectNo

Refer to Parameters you require often (Page 287).

Name: **Enabled**

Declaration, data type , default and range of values: Refer to Parameters you require often (Page 287).

Explanation Whether or not processing of the block is enabled must be specified.

If processing is enabled, all the functions of the FB execute.

If processing is disabled the FB only checks to see if the disabled status has been canceled. The FB cannot communicate on the organizational level in this status because FB Cmd01B_S cannot send or receive organizational messages.

Name: **CommandInputByte_HW**

Declaration: INPUT

Data type BYTE

Default: 0 (B#16#0)

Explanation Command input byte for hardware input.

Range of values:	Input byte	IB0 ... IBn PIB0 ... PIBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n

This command input byte is specially designed for entering commands using hardware, i.e. over digital inputs. Input using memory or data bytes is also possible, but the user must then make sure that the command at the input byte is reset, which occurs at the hardware level when the command button is released.

When input is detected, the command is transmitted 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. 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 first read in when no hardware command has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command input blocks with hardware input. The FB enters the 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 over 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
Default:	0 (B#16#0)
Explanation	Command input byte for software input.

Range of values:	Memory bytes	MB0 ... MBn
	Data bytes	DBm.DBB0 ... n

This command input byte is specially designed for entering commands using software, i.e. by the user program or at an operator panel (OP). When input is detected, the command is reset at the input byte and transmitted if no error is detected during the 1-out-of-8 and 1-out-of-n check. 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 first read in when no software command has been acquired on the PLC for one OB1 cycle; in other words, not only for this block but also for all other command input blocks with software input. The FB enters the 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 more detailed information, refer to the description of FC Safe.

In principle it is possible to enter a new command to 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

The command inputs CommandInputByte_HW and CommandInputByte_SW can also be used at the same time; in other words, if you want to enter the same command over the hardware and software. If a command entry occurs at the same time over both input bytes, this is only accepted when coincidentally exactly the same command is entered over the hardware as well as the software input (the hardware entry is then processed). In all other cases the entry is rejected and an error message is entered in the diagnostics buffer (event ID B170). The error status is also indicated by the InputError output of FC Safe and appropriate error bits are set in the central data block "BasicData" where they can be queried by the software (see FC Safe).

Note

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.

5.5.9 ST7 command typical FB Cmd01B_R

Function

Receive 1 byte commands (1-out-of-8 ST1 format).

Explanation of the parameters

Names: **PartnerNo**

PartnerObjectNo

Refer to Parameters you require often (Page 287).

Name: **Enabled**

Declaration, data type , default and range of values: Refer to Parameters you require often (Page 287).

Explanation Whether or not processing of the block is enabled must be specified.

If processing is enabled, all the functions of the FB execute.

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 on the organizational level in this status because FB Cmd01B_R cannot send or receive organizational messages.

Note

If the Enabled input can be controlled by a switch, this local disable means that no more commands are output if they are still received. Since the block is, however, not capable of sending ORG messages, it cannot report this local disable back to the partner itself. This must be implemented by the user with a separate message, for example Bin04B_S.

Name: **MultipleOutput**

Declaration: INPUT

Data type BOOL

Default: FALSE

Explanation Simultaneous output of multiple commands permitted.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

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 is still being output (command output time has not yet elapsed or the user program has not yet reset this command).

FALSE (default):
Multiple command 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 command output is permitted. A newly received command is ORed into the current output byte. The command output time is retriggered. This applies to all pending commands.

Name:	**CommandOutputTime**
Declaration:	INPUT
Data type	INT
Default:	500
Explanation	500

Range of values:	Command up time for command outputs in ms. No parameter specified: Default value 500 [ms] is valid.

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 that pending commands are retriggered. All the command outputs are reset of the same time when the output time elapses.

Point to note with CommandOutputTime = 0

A set command output is not reset by the command typical. The user program is responsible for this.

If no parameter is specified, an output time of 500 ms is used as the default.

Name:	**NewData**

Refer to Parameters you require often (Page 287).

Explanation of the typical-specific outputs	Whenever the FB has received new data and has output it to the output byte CommandOuputByte, the NewData output is set to TRUE for one OB1 cycle.

Name:	**CommandOutputByte**
Declaration:	IN_OUT
Data type	BYTE
Default:	0 (B#16#0)
Explanation	Command output byte.

Range of values:	(process image) output bytes	QB0 ... QBn
	Memory bytes	MB0 ... MBn
	Data bytes	DBm.DBB0 ... n

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.

Note

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.

5.5.10 ST7 setpoint typical FB Set01W_S

Function

Send 1 setpoint (16 bits) and receive current local setpoint.

Note

With FB Set01W_S, data can only be transmitted when FC Safe is included at the end of the cyclic SINAUT program.

Explanation of the parameters

Names:	**PartnerNo**
	PartnerObjectNo
	Enabled

See Parameters you require often (Page 287).

Name:	**EnterInput**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Enter input (for 'hardware' setpoint).

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid	

A setpoint at the SetpointInput can be applied over this input triggered by a signal edge change.

A signal change at EnterInput is only taken into account when the parameter ContinuousEnterFunct = FALSE. If this condition is fulfilled, the setpoint entered at SetpointInput is applied and transmitted by a signal change from 0 to 1 even if the newly entered setpoint is identical to the previously sent setpoint.

This method of applying setpoints is suitable for input at appropriate hardware, for example a console or control panel but can also be used for entering setpoints at an operator panel (OP). In the latter case, it must be possible to trigger the input by a separate function key on the OP.

If you do not require the parameter, simply leave it open.

Name:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Apply setpoint continuously (for ’software’ setpoint).

Range of values: TRUE or FALSE
No parameter specified: Default value TRUE is valid.

With this parameter, you can decide whether the setpoint at SetpointInput should be continuously read in and changes checked. The change evaluation is made by comparing the current with the last setpoint that was sent.

This method of applying a setpoint is suitable for input by appropriate software but can also be used for entering setpoints at an operator panel (OP) if it 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.

Name:	**SetpointInput**
Declaration:	INPUT
Data type	WORD
Default:	0 (W#16#0)
Explanation	Setpoint input word.

Range of values:	Input words	IW0 ... IWn PIW0 ... PIWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

How a setpoint available at SetpointInput is processed depends on whether it is a hardware or software input. The user specifies the type of input with the ContinuousEnterFunct parameter:

- ContinuousEnterFunct = FALSE (= hardware input)
  The setpoint at SetpointInput is only read in as long as a 1 signal is detected at EnterInput. The setpoint that is read in is then transmitted 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). The next setpoint is first read in by the FB when a 0 signal is detected for at least one OB1 cycle at EnterInput.
  If a 1-out-of-n error is detected at the hardware input, the entered setpoint is no longer processed. A new setpoint is first read in when no hardware input has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command and setpoint input blocks with hardware input. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172). The error status is also indicated over the InputError output of FC Safe (see FC Safe, InputError parameter) and continues to be indicated as long as the error remains.
- ContinuousEnterFunct = TRUE (= software input)
  The setpoint at SetpointInput is read in continuously and checked for changes. The change evaluation is made by comparing the current with the last setpoint that was sent. The setpoint is sent immediately every time a change occurs unless the 1-out-of-n check detects an error. Without the setpoint having changed, a new transfer of the software setpoint will be triggered over the SendSoftSetpoint input (see below).
  While for hardware input an empty cycle must be detected before a new setpoint can be transferred by the block, for software input a new setpoint can be transferred in every OB1 cycle. This is possible only when there is no other software setpoint or software command in this cycle. Otherwise a 1-out-of- n error is detected.
  If a 1-out-of-n error is detected during the software input, the entered setpoint is no longer processed. A new setpoint is first read in when no software input has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command and setpoint blocks with software input. The FB enters the detected 1-out-of-n error in the diagnostic 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 more detailed information, refer to the description of FC Safe.

Name:	**ReturnedSetpoint**
Declaration:	OUTPUT
Data type	WORD

Default:	0 (W#16#0)		
Explanation	Output word for a returned setpoint.		
	Range of values:	Output words	QW0 ... QWn PQW0 ... PQWn
		Memory words	MW0 ... MWn LW0 ... LWn
		Data words	DBm.DBW0 ... n

The partner object receiving the setpoint reports back the currently valid local setpoint. This value is displayed at the ReturnedSetpoint output. If the partner object is set to ’local’ and if an input is made there, then the setpoint changed locally is displayed here at ReturnedSetpoint.

After startup of the local or partner PLC, or after restoring a connection, an automatic general request ensures that the current, local, valid setpoint is displayed at ReturnedSetpoint.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require the parameter, simply leave it open.

Name:	**LocalOperation**		
Declaration:	OUTPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Return message from the partner object: Object is set to local operation.		
	Range of values:	Output	Q 0.0 ... Q n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid.	

A setpoint can also be set locally at the partner object that receives the setpoint. The partner object then must be set to ’local’ at the Local input parameter (see FB Set01W_R below). The current status of the Local input parameter is reported by the partner object and displayed here at the LocalOperation output.

After startup of the local or partner PLC, or after restoring a connection, an automatic general request ensures that the current, local, valid status is displayed at LocalOperation.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require the parameter, simply leave it open.

Name:	**NewData**

See Parameters you require often (Page 287).

Explanation of the typical-specific outputs	Whenever the FB 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.

Name:	**SendSoftSetpoint**		
Declaration:	IN_OUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Trigger input for resending the last (software) setpoint.		
	Range of values:	Bit memory	M 0.0 ... M n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid.	
	See also SetpointInput parameter.		
	If you do not require the parameter, simply leave it open.		

Note

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.

5.5.11 ST7 setpoint typical FB Set01W_R

Function

Receive or enter 1 setpoint locally (16 bits) and send the current, locally valid setpoint.

Explanation of the parameters

Names:	**PartnerNo**
	PartnerObjectNo
	Enabled
	ImageMemory
	Conditional
	Unconditional
	Permanent

TimeStamp

Refer to Parameters you require often (Page 287).

Name:	**Local**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Enable block processing.

Range of values:	TRUE or FALSE	
	No parameter specified: Default value FALSE is valid	
	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

This input is used to enable local input of a setpoint over LocalSetpointInput. A setpoint sent for example by the master 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 Local = 1, the last values at ParameterOutput are held until new parameter values are entered over LocalParameterInput.
- When there is a switchback from Local = 1 to Local = 0, the last values at ParameterOutput are held until the block receives new parameter values from the remote partner.

Note

Please read the note on the ContinuousEnterFunct parameter.

Name:	**EnterInput**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Enter input for local setpoint input.

Range of values:	TRUE or FALSE
	No parameter specified: Default value FALSE is valid

Input	I 0.0 ... I n.7
Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
Data bit	DBm.DBX 0.0 ... n.7

A setpoint at the LocalSetpointInput can be applied over this input triggered by a signal edge change.

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 causes the setpoint at LocalSetpointInput to be applied and output at SetpointOutput.

This method of applying setpoints is suitable for input at appropriate hardware, for example a console or control panel but can also be used for entering setpoints at an operator panel (OP). In the latter case, it must be possible to trigger the input by a separate function key on the OP.

If you do not require the parameter, simply leave it open.

Name:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Continuous local setpoint acquisition.

Range of values: TRUE or FALSE
No parameter specified: Default value FALSE is valid.

With this parameter, you can decide whether the setpoint at LocalSetpointInput should be continuously read in and changes checked. The change evaluation is made by comparing the current with the last mirrored setpoint. The value is only read when the Local input parameter is set to TRUE.
If an array is detected, this is output immediately at SetpointOutput.

This method of acquiring a setpoint is suitable for input by appropriate software but can also be used for entering setpoints at an operator panel (OP) if it 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.

Note

When ContinuousEnterFunct = TRUE, the value available at LocalSetpointInput is entered immediately and passed to SetpointOutput when the 1 signal at the Local input is detected if the local input value differs from the last returned setpoint at this point in time!

Name:	**LocalSetpointInput**
Declaration:	INPUT
Data type	WORD
Default:	0 (W#16#0)

Explanation	Local setpoint input word.		
	Range of values:	Input words	IW0 ... IWn PIW0 ... PIWn
		Memory words	MW0 ... MWn LW0 ... LWn
		Data words	DBm.DBW0 ... n
		No parameter specified: Default value 0 is valid.	

A value at LocalSetpointInput is only adopted if the Local input parameter is set to TRUE. If this condition is met, how a pending setpoint is processed depends on whether it is a hardware or software input. The user specifies the type of input with the ContinuousEnterFunct parameter:

- ContinuousEnterFunct = FALSE (= hardware input)
  The setpoint at LocalSetpointInput is only read in when a signal change from 0 to 1 is detected at EnterInput. The setpoint entered locally is output over the output set with SetpointOutput and transferred to the partner for display. A further setpoint is then only read in by the FB when a 0 signal was detected at EnterInput for at least one OB1 cycle.
- ContinuousEnterFunct = TRUE (= software input)
  The setpoint at LocalSetpointInput is read in continuously and checked for changes. The change evaluation is implemented by comparing the current value with the last valid setpoint; in other words, the value stored as the returned setpoint. Every time there is a change, the setpoint is passed immediately to the output specified by SetpointOutput and sent to the partner for display.
  While for hardware input an empty cycle must be detected before a new setpoint can be read by the block, for software input a new setpoint can be entered in every OB1 cycle.

If you do not require the parameter, simply leave it open.

Name:	**SetpointOutput**		
Declaration:	OUTPUT		
Data type	WORD		
Default:	0 (W#16#0)		
Explanation	Setpoint output word.		
	Range of values:	Output words	QW0 ... QWn PQW0 ... PQWn
		Memory words	MW0 ... MWn LW0 ... LWn
		Data words	DBm.DBW0 ... n

The setpoint sent by the partner object or the setpoint entered locally at LocalSetpointInput is output to the output word specified here in SetpointOutput.

Name:	**NewData**

Refer to Parameters you require often (Page 287).

Explanation of the typical-specific outputs	Whenever the FB 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 new local input when Local = 1.

5.5.12 ST7 parameter typical FB Par12D_S

Function

Send 1 to 12 parameter values (each 1 double word) and receive back the current, locally valid parameter values.

Note

With FB Par12D_S, data can only be transmitted when FC Safe is included at the end of the cyclic SINAUT program.

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,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

The data area to be transferred is defined 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 data 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 ReturnedParameter. This output area (defined by an Any pointer) must also be within a data block and its length must match that defined for ParameterInput.

Separate data areas are normally specified for ParameterInput and ReturnedParameter. This makes it easy to recognize the most recently entered values and the current, locally valid values. 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 returned values are not needed, there is no need to specify a data area for ReturnedParameter.

Even when separate areas are specified for ParameterInput and ReturnedParameter, it is still possible to ensure that the ParameterInput area and the ReturnedParameter value always match. This 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 then must be set to ’local’ at the Local input parameter (see FB Par12D_R below). The current status of the Local input parameter is reported by the partner object and displayed here at the LocalOperation output. As long as the partner object is set to 'local', no parameters are accepted there from other locations.

Transmission of the data area defined by ParameterInput can be triggered in four ways:

- With the input parameter 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 over a digital input. The transmission of the entered values is then triggered by pressing this button. The entire data area specified by ParameterInput is always transmitted.
- With the input parameter ContinuousEnterFunct = TRUE
  You can use this parameter setting when the parameter is entered by software, for example at 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 transmission are transmitted (see note).
- Via the input parameter Release
  You can use this input parameter if the parameters are entered by software, e.g. 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 defined with ParameterInput that have changed since the last transmission (see note) are transmitted.
- Via the input parameter RetransmitAll
  You can use this input parameter if the parameters are entered by software, e.g. 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 entire data area defined by ParameterInput is transmitted. Changes are not checked.

Note

When the changed data area only is transmitted, 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 is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word.

Note

When only changed data is transmitted and the data area contains values in double word format, the user is responsible for ensuring that these 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 otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to processing problems on the partner that receives this value.

Explanation of the parameters

Names:	**PartnerNo** **PartnerObjectNo** **Enabled**

See Parameters you require often (Page 287).

Name:	**EnterInput**		
Declaration:	INPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Enter input.		
	Range of values:	TRUE or FALSE	
		No parameter specified: Default value FALSE is valid	
		Input	I 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7

The transmission of the parameter value at ParameterInput can be triggered over this input by a signal edge change.

A signal change at EnterInput is only taken into account when the parameter ContinuousEnterFunct = FALSE. If this condition is fulfilled, a transition from 0 to 1 causes the parameter values specified at ParameterInput to be entered and transmitted. Changes are not checked. The entire data area specified by ParameterInput is always transmitted.

This method of transmission triggering is suitable for input with appropriate hardware, for example at a console or control panel. For more detailed information and related parameters, refer to the section Function.

If you do not require the parameter, simply leave it open.

Data checks:

- The parameters that are read in are then transmitted 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). The input area is then only read in 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 no "hardware" input has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command, setpoint and parameter blocks with "hardware" input. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172). The error status is also indicated over the InputError output of FC Safe (see FC Safe, InputError parameter) and continues to be indicated as long as the error remains.

Name:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Continuous change checking.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

With this parameter, you can decide whether the parameter values at ParameterInput should be continuously read in and changes checked. The change evaluation is made by comparing the current with the last values that were sent. 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.

A new transmission of the parameter values can be triggered over 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) when the OP has no separate function key with which to trigger transmission. For more detailed information and related parameters, refer to the section Function.

If you do not require the parameter, simply leave it open. The default value FALSE then applies; in other words, the parameter values at ParameterInput are not read continuously and evaluated for changes.

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 sent from the block, for "software" input new parameter values can be transmitted in every OB1 cycle. This assumes that there is no other "software" entry at another block in this cycle. Otherwise a 1-out-of-n error is detected.
- When a 1-out-of-n error is detected at the "software" input, the entered parameters are no longer processed. New parameters are read in again only when no "software" input has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command, setpoint and parameter blocks with "software" input. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172).

Note

The changed data area that is transmitted 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 is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word.

Name:	**ApplyRemoteParamAuto**
Declaration:	IN
Data type	BOOL
Default:	FALSE
Explanation	Automatic synchronization of the input area with the returned area. Range of values: TRUE or FALSE No parameter specified: Default value FALSE is valid. If ApplyRemoteParamAuto = TRUE, the input area ParameterInput is automatically synchronized with the ReturnedParameter area. All the parameter values from the ReturnedParameter area are then copied to the ParameterInput area. The send buffer is also synchronized with the returned parameter values. 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. The default value FALSE then applies; in other words no automatic synchronization is performed.

Name:	**ParameterInput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Parameter input area.

Range of values:	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! No parameter specified: Default (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer defines 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 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,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

If the parameter setting is incorrect (null pointer, length greater than 12, data area not a data block), 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 parameter assignment error has been corrected.

How the parameters at ParameterInput are processed depends on whether they are "hardware" or "software" entries and how the transmission of this data area is triggered. For more information refer to the detailed description of in Function.

Note

When only changed data is transmitted and the data area contains values in double word format, the user is responsible for ensuring that these 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 otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to processing problems on the partner that receives this value.

Name:	**ReturnedParameter**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)

Explanation	Parameter output area.	
	Range of values:	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! No parameter specified: Default (null pointer) is valid.

The partner object receiving the parameter values reports back the currently valid local parameter values. These values are displayed at the ReturnedParameter output. If the partner object is set to 'local' and if an input is made there, then parameters changed locally are displayed here at 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 PLC, or after restoring a connection, an automatic general request ensures that the current, local, valid parameters are displayed at ReturnedParameter.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

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 parameter assignment error has been corrected.

Name:	**LocalOperation**
Declaration:	OUTPUT
Data type	BOOL
Default:	FALSE
Explanation	Return message from partner object: Object is set to local operation.

Range of values:	Output	Q 0.0 ... Q n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid.	

A parameter can also be set locally at the partner object that receives the parameter. The partner object then must be set to 'local' at the Local input parameter (see FB Par12D_R below). The current status of the Local input parameter is reported by the partner object and displayed here at the LocalOperation output. As long as the partner object is set to 'local', no parameters are accepted there from other locations.

After startup of the local or partner PLC, or after restoring a connection, an automatic general request ensures that the current, local, valid status is displayed at LocalOperation.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require the parameter, simply leave it open.

Name: **NewData**

See Parameters you require often (Page 287).

Explanation of the typical-specific outputs: Whenever the FB has received new data and has output it to the outputs ReturnedParameter or LocalOperation, the NewData output is set to TRUE for one OB1 cycle.

Name:	**Release**
Declaration:	IN_OUT
Data type	BOOL
Default:	FALSE
Explanation	Trigger input for sending the currently pending parameter values

Range of values:	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid.	

You can use this input parameter when the parameter is entered by software, for example at 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. They are only transmitted when the Release function key is activated because the change check only begins with a 1 signal at the Release input and the data double words that have changed since the last transmission are transmitted from the data area defined by ParameterInput.

If you always want to transmit the entire data area defined with ParameterInput and not only the changed parameter values, you should use the RetransmitAll 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 ContinuousEnterFunct = TRUE. Refer to the description there.

If you do not require the parameter, simply leave it open.

Note

The changed data area that is transmitted 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 is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word.

Note

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.

Name:	**RetransmitAll**
Declaration:	IN_OUT
Data type	BOOL
Default:	FALSE
Explanation	Trigger input for transmitting (or retransmitting) the entire data area defined by ParameterInput.

Range of values:	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid.	

You can use this input parameter when the parameter is entered by software, for example at an operator panel (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 entire data area defined by ParameterInput is transmitted. Changes are not checked.

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 transmission of all parameter values as long as the digital input has a 1 signal.

The RetransmitAll input can also be used as an option in addition to Release or ContinuousEnterFunct = TRUE when new parameter values were entered but could not be transmitted to the partner (for example because of a disrupted connection or because the partner object was previously set to 'local'). You can then trigger transmission of the entire data area defined by ParameterInput using the RetransmitAll input. All changes that were previously entered but are not yet available at the partner are consistently included.

The RetransmitAll input can also be used as an independent transmission trigger when you always want to send all entries and not just those that have changed. You should then use RetransmitAll instead of Release that only sends the changed parameter values.

Data checks:

The same safety checks are carried out as with ContinuousEnterFunct = TRUE. Refer to the description there.

If you do not require the parameter, simply leave it open.

Note

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.

Name:	**ApplyRemoteParamMan**
Declaration:	IN_OUT
Data type	BOOL
Default:	FALSE
Explanation	Trigger input for synchronization of the input area with the returned area.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

The input triggers a one-time synchronization of the ParameterInput input area with the ReturnedParameter area. All the parameter values from the ReturnedParameter area are then copied to the ParameterInput area. The send buffer is also synchronized with the returned parameter values.

The ApplyRemoteParamMan 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. The result would be a constant synchronization as long as the digital input has a 1 signal.

If you do not require the parameter, simply leave it open.

Note

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.

5.5.13 ST7 parameter typical FB Par12D_R

Function

Receive 1 to 12 parameter values (each 1 double word) or enter locally and send back the current, locally valid parameter values.

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,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

The data area in which the received parameter values are output is defined with 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 defined 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 at the ParameterOutput parameter.

The block only sends the changed data area. However, the complete parameter set is returned in response to a general or single request.

Bumpless switchover between the Local and Remote operating modes is guaranteed.

Explanation of the parameters

Names:	**PartnerNo** **PartnerObjectNo** **Enabled** **ImageMemory** **Conditional** **Unconditional** **TimeStamp**

See Parameters you require often (Page 287).

Name:	**Local**		
Declaration:	INPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Local parameter input released.		
	Range of values:	TRUE or FALSE	
		Input	I 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid	

This input is used to enable local input of a parameter over the data area addressed with LocalParameterInput. Parameters sent for example by the master are 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 Local = 1, the last values at ParameterOutput are held until new parameter values are entered over LocalParameterInput.
- When there is a switchback from Local = 1 to Local = 0, the last values at ParameterOutput are held until the block receives new parameter values from the remote partner.

Special case:
You can also enter the parameter values during local input directly in the output area defined 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 the parameter entry cannot be prevented by the Local input. Regardless of the Local status, the values entered in the output area are sent immediately to the partner by the function block.
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 remote partner:

- If Local = 0, the parameters sent by the remote partner are accepted and output to the ParameterOutput data area.
- If Local = 1, any parameters sent by the remote partner are rejected.

In this special situation, the Release input and ContinuousEnterFunct have no function.

A status change of the Local parameter is always sent by the TIM 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 FB Par12D_S, parameters ApplyRemoteParamMan and ApplyRemoteParamAuto).

Note

Please read the note on the ContinuousEnterFunct parameter.

Name:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Continuous local parameter acquisition.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

With this parameter, you can decide whether the parameter values in the LocalParameterInput input area should be continuously read in and changes checked. The change check is implemented by comparing the current parameter values at ParameterOutput. Changes in the input area are copied immediately to the output area and transmitted 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 ContinuousEnterFunct = TRUE parameter setting only takes effect when the following conditions are met:

- An input area is defined by the LocalParameterInput parameter and this is not identical to the output area defined by ParameterOutput.
  and
- There is a 1 signal at the Local input (= TRUE).

This method of local parameter acquisition is suitable when the parameter values are entered in the LocalParameterInput area by software, but can also be used for entering the parameters from an operator panel (OP) when the OP has no separate function key with which to trigger acceptance. For more detailed information and related parameters, refer to the section Function.

If you do not require the parameter, simply leave it open.

Note

The changed data area that is transmitted 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 is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word.

Note

When ContinuousEnterFunct = TRUE, the values available at LocalparameterInput are entered immediately and passed to ParameterOutput when the 1 signal at the Local input is detected assuming the local input values differ from the current parameter value output at this point in time.

Name:	**LocalParameterInput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Local parameter input area.

Range of values:	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! No parameter specified: Default (null pointer) is valid.

The ANY pointer defines the data area in which the parameter values to be acquired locally 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 with the length set for ParameterOutput.

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,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

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 ParameterOutput), 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 parameter assignment error has been corrected.

Note

Only changed data is transmitted to the partner. If the data area contains values in double word format, the user is responsible for ensuring that these 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 otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to problems in processing on the partner that receives this value (applies, for example to ST7cc, but not to an S7 CPU).

Name:	**ParameterOutput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Parameter output area.

Range of values:	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! No parameter specified: Default (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer defines 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.

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,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

FB Par12D_R stores the received data without further processing in the data area defined by ParameterOutput. The user program is responsible for evaluating and processing received data.

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, namely, the area in which the changes were detected at the acquisition end.

If the parameter setting is incorrect (null pointer, length greater than 12, data area not a data block), 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 parameter assignment error has been corrected.

Note

When only the changed data area is received, 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 is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word.

Name:	**NewData**

See Parameters you require often (Page 287).

Explanation of the typical-specific outputs	Whenever the FB 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 when Local = 1.

Name:	**Release**
Declaration:	IN_OUT
Data type	BOOL
Default:	FALSE
Explanation	Input for the acceptance of local parameter entry. No parameter specified: Default value FALSE is valid

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

The acceptance of the parameter value at the LocalParameterInput parameter input can be triggered over 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 defined by the LocalParameterInput parameter and this is not identical to the output area defined by ParameterOutput.
  and
- There is a 1 signal at the Local input (= TRUE).

You can use this Release input parameter when the parameter is entered by software, for example at 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 check is implemented by comparing the current parameter values at ParameterOutput. Changes in the input area are then copied immediately to the output area and transmitted 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. The triggering of the acquisition over Release is triggered by a signal edge change; in other words, it occurs only once.

If you do not require the parameter, simply leave it open.

Note

The changed data area that is transmitted 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 is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word.

Note

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.

5.5.14 ST7 data typical FB 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 format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

Sending the data area can be triggered in two ways:

- By a change check. Transmission is triggered as soon as a bit changes. (Parameter SendOnChange = TRUE)
- The user program decides when a transmission will take place (signal edge change from 0 to 1 at TriggerInput).
  This could also be a time-driven transmission. This case, you could use FC Trigger.

You can also specify whether the transmission includes all data or only the data double words that have changed.

Note

When only changed data is transmitted and the data area contains values in double word format, the user is responsible for ensuring that these 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 otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to problems in processing on the partner that receives this value (applies, for example to ST7cc, but not to an S7 CPU).

Explanation of the parameters

Names:	**PartnerNo**
	PartnerObjectNo
	Enabled
	ImageMemory
	Conditional
	Unconditional
	TimeStamp

See Parameters you require often (Page 287).

Name:	**SendOnChange**	
Declaration:	INPUT	
Data type	BOOL	
Default:	FALSE	
Explanation	Send on change.	
	Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.
	Here, you specify whether the FB checks for changes within the acquired data area (to determine whether at least one bit has changed). If a change is detected, a transmission of the data area is triggered automatically. Whether the entire area is transmitted or only the changed part can be specified with the SendAll parameter (refer to the explanations on the relevant parameter). If no parameter is specified, the default is FALSE; in other words, there is no change-driven data transmission. Transmission must then be triggered by the user at the TriggerInput input parameter.	

Name:	**TriggerInput**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE

Explanation	Trigger input.		
	Range of values:	Input	I 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid.	

If required, this parameter can be used to specify an input over which the user can trigger the transmission *) of the data message at any time (signal edge from 0 to 1).

Example:
Time-driven analog value transmission with time stamp for supplying an analog value archive in the control center. Note: To prevent these messages with time stamps from being overwritten when saving on the station TIM, the ImageMemory parameter must be set to FALSE.

FC Trigger can be used for time-driven triggering of a transmission over TriggerInput (for more detailed information, refer to the description of this block).

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.

*) TriggerInput actually only triggers transmission indirectly. With a 0/1 edge at TriggerInput, the message is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. Transmission is immediate over a dedicated line or wireless link; with a dial-up connection, it is possible that the message is saved first on the TIM and sent at a later point in time (for example, because the message is marked as a "conditional spontaneous" message; see the Conditional parameter).

Note

You can also select a combination of SendOnChange plus TriggerInput. This means that a transmission is triggered both when a change is detected and at every signal edge change from 0 to 1 at the TriggerInput.

Note

If you use neither SendOnChange nor TriggerInput to trigger data transmission, the data will only be transmitted when there is a single request for this data object or within the framework of a general request.

Name:	**SendAll**
Declaration:	INPUT
Data type	BOOL
Default:	TRUE

Explanation	Send all data with every transmission.

Range of values: TRUE or FALSE

Here, you specify whether the FB will transfer or data of the area defined with DataInput or only changed data. The transmission 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 a transmission is triggered over TriggerInput, and no data has changed at this point in time, the entire area is transmitted (in this exceptional situations, corresponds to SendAll = TRUE).

If no parameter is specified, the default TRUE applies; in other words, the entire area is always transmitted.

Note

When only the changed data area is transmitted (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.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word.

Note

If there is a single request for this data object or within the framework of a general request, all data words of the area defined by DataInput are always transmitted.

Name:	**DataInput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Data input area.

Range of values:	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! No parameter specified: Default (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer describes 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.

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,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

If the parameter setting is incorrect (null pointer, length greater than 12, data area not a data block), 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 parameter assignment error has been corrected.

Note

When only changed data is transmitted and the data area contains values in double word format, the user is responsible for ensuring that these 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 otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to problems in processing on the partner that receives this value (applies, for example to ST7cc, but not to an S7 CPU).

5.5.15 ST7 data typical FB 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 format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

FB Dat12D_R stores the received data without further processing in the data area defined by DataOutput. The user program is responsible for evaluating and processing received data correctly.

Explanation of the parameters

Names:	**PartnerNo**
	PartnerObjectNo
	Enabled

See Parameters you require often (Page 287).

Name:	**DataOutput**	
Declaration:	INPUT	
Data type	ANY	
Default:	P#P 0.0 VOID 0 (null pointer)	
Explanation	Data output area.	
	Range of values:	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! No parameter specified: Default (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer describes the data area in which the received data will be stored. This data area must be within a data block and its length can vary between 1 and 12 data double words.

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,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

FB Dat12D_R stores the received data without further processing in the data area defined by DataOutput. The user program is responsible for evaluating and processing received data correctly.

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, namely, the area in which the changes were detected at the acquisition end.

How to read out the time stamp received with the data is described in the section "Notes on the SINAUT time stamp".

If the parameter setting is incorrect (null pointer, length greater than 12, data area not a data block), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 4). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

Note

When only the changed data area is received, 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 is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word.

Name: **NewData**

See Parameters you require often (Page 287).

Explanation of the typical-specific outputs: Whenever the FB has received new data and has stored it in the data area defined by DataOutput, the NewData output is set to TRUE for one OB1 cycle.

5.5.16 ST7 data typical FB-Dat256D_S

Function

Send 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
- Byte
- 2 words
- Each combination (e.g. 2 bytes + 1 word, etc.)

Sending the data area can be triggered in two ways:

- By the change control: Transmission is triggered as soon as a bit changes (parameter SendOnChange = TRUE).
- The user program specifies when a transmission takes place (edge change from 0 to 1 at TriggerInput).

You can decide whether all the data should be sent (SendAll) or only the part of the net data that contains changes.

With S7-300 CPUs with X communication, the maximum message length is 76 bytes. For this reason, 1024 bytes of net data are transferred using a serial transfer process consisting of a sequence of at least 22 data messages (segments). Each message apart from the last contains a segment of 48 bytes of net data of the input data area.

To ensure data consistency in the SendAll mode or during a general or single request, the data is sent 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

If TriggerInput is triggered when SendAllBusy = TRUE, this leads to the DataLoss error message (status in the message header) if the transfer is triggered again.
The edge change from 0 to 1 via TriggerInput occurs only when SendAllBusy = FALSE is set.

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. An entry is also made in the diagnostics buffer with the event ID B13BTD7_Diagnostics.

Note

If the status of the partner changes from "available" to "unavailable", the transfer of all data is stopped immediately and all object data is deleted from the TIM buffer. This can lead to data loss.
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.

Parameter

Names	**PartnerNo**
	PartnerObjectNo
	Enabled
	ImageMemory
	Conditional
	Unconditional
	TimeStamp

Refer to section Parameters you require often (Page 287) for an explanation of these parameters.

Explanation of the other parameters

Name	**SendOnChange**
Declaration	INPUT
Data type	BOOL
Default	FALSE
Explanation	Send on change.
	Range of values: TRUE or FALSE
	Without a parameter setting, the default value FALSE is valid.
	Here, you specify whether the FB checks for changes within the acquired data area (to determine whether at least one bit has changed). If a change is detected, a transmission of the data area is triggered automatically. Whether the entire area is transmitted or only the changed part can be specified with the SendAll parameter (refer to the explanations on the relevant parameter).
	If no parameter is specified, the default is FALSE; in other words, there is no change-driven data transmission. Transmission must then be triggered at the TriggerInput input parameter.

Name	**TriggerInput**		
Declaration	INPUT		
Data type	BOOL		
Default	FALSE		
Explanation	Trigger input.		
	Range of values:	Input	I 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7

Without a parameter setting, the default value FALSE is valid.

If required, this parameter can be used to specify an input via which you can trigger the transmission *) of the data message at any time (signal edge from 0 to 1).

Example:
Time-driven analog value transmission with time stamp for supplying an analog value archive in the control center.

TriggerInput actually only triggers transmission indirectly. With a 0/1 edge at TriggerInput, the message is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. Transmission is immediate over a dedicated line or wireless link; with a dial-up connection, it is possible that the message is saved first on the TIM and sent at a later point in time (for example, because the message is marked as a "conditional spontaneous" message; see the Conditional parameter).

FC Trigger can be used for time-driven triggering of a transmission over TriggerInput.

Select suitable trigger points so that the messages on the TIM are 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.

Note

You can also select a combination of SendOnChange plus TriggerInput. This means that a transmission is triggered both when a change is detected and at every signal edge change from 0 to 1 at the TriggerInput.

If you use neither SendOnChange nor TriggerInput to trigger data transmission, the data will only be transmitted when there is a single request for this data object or within the framework of a general request.

Note

Do not transfer any analog values for which the SendOnChange parameter = TRUE is set without prior process data preprocessing. You will find more detailed information on this with the analog value typical FB-Ana04W_S, ThresholdValue.

Name	**SendAll**
Declaration	INPUT
Data type	BOOL
Default	TRUE
Explanation	Send all data with every transmission.

Range of values:
- TRUE (always all data)
- FALSE (only changed data)

Without a parameter setting, the default value TRUE is valid.

Here, you specify whether the FB will transfer or data of the area defined with DataInput or only changed data. The transmission 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 = FALSE is set and the transfer is triggered by TriggerInput but no data change has occurred up to now, the entire data area is transferred. This corresponds to SendAll = TRUE.

Note

If there is a single request for this data object or within the framework of a general request, all data words of the area defined by DataInput are always transmitted.

Name	**DataInput**
Declaration	INPUT
Data type	ANY
Default	P#P 0.0 VOID 0 (null pointer)
Explanation	Data input area

Range of values:	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!

Without a parameter setting, the default value (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer describes 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.

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
- 2 bytes plus 1 word etc.

If the parameter setting is incorrect (null pointer, length greater than 256, data area not a data block), 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 parameter assignment error has been corrected.

Note

If a data segment to be transferred consists of a maximum of 48 bytes, data consistency during the transfer is always assured.

Note

When only changed data is transmitted and the data area contains values in double word format, you are responsible for ensuring that these 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 otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to problems in processing on the partner that receives this value (applies, for example to ST7cc, but not to an S7 CPU).

Name	**SendAllBusy**
Declaration	OUTPUT
Data type	BOOL

Default	FALSE
Explanation	The block is currently being processed in SendAll mode. Range of values: TRUE or FALSE This output indicates that the function 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 from 0 to 1). The edge change from 0 to 1 has no effect whatsoever with an external TriggerInput as long as SendAllBusy indicates TRUE. The edge change from 0 to 1 of TriggerInput only takes effect when SendAllBusy = FALSE.

Name	**SendAllError**
Declaration	INOUT
Data type	BOOL
Default	FALSE
Explanation	An error occurred during processing of SendAll. Range of values: TRUE or FALSE SendAllError is set to TRUE when the connection is interrupted during processing of SendAll. An entry is also made in the diagnostics buffer. SendAllError remains set to TRUE until it is reset by the user or by the next CPU restart.

5.5.17 ST7 data typical FB-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.

FB Dat12D_R stores the received data without further processing in the data area defined by DataOutput. The user program is responsible for evaluating and processing received data correctly.

With S7-300 CPUs with X communication, the maximum message length is 76 bytes. For this reason, 1024 bytes of net data are transferred using a serial transfer process consisting of a sequence of at least 22 data messages (segments). Each message 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 (messages), 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 (message) is not part of a received sequence (SendAll = FALSE), the status remains set to TRUE for only one CPU cycle.

To ensure data consistency in the "SendAll" mode 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. This happens in particular 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 (message) is received during an active sequence.

Parameter

Names: **PartnerNo**
PartnerObjectNo
Enabled

Refer to section Parameters you require often (Page 287) for an explanation of these parameters.

Explanation of the other parameters

<table>
<tr><td>Name</td><td colspan="3">SingleRequest</td></tr>
<tr><td>Declaration</td><td colspan="3">INPUT</td></tr>
<tr><td>Data type</td><td colspan="3">BOOL</td></tr>
<tr><td>Default</td><td colspan="3">TRUE</td></tr>
<tr><td>Explanation</td><td colspan="3">A single request is sent to the partner object.</td></tr>
<tr><td></td><td>Range of values:</td><td>Input:
Bit memory:

Data bit:</td><td>I 0.0 … I n.7
M 0.0 … M n.7
L 0.0 … L n.7
DBm.DBX 0.0 … n.7</td></tr>
<tr><td></td><td colspan="3">Without a parameter setting, the default value TRUE is valid.</td></tr>
<tr><td></td><td colspan="3">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 defined 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.</td></tr>
</table>

Note

Consistency of the data output area across segments is only assured if the receive sequence was completed successfully.

The time taken for the response to the single request is evaluated by the MonitoringTime parameter.

Name	**MonitoringTime**
Declaration	INPUT
Data type	INT
Default	0

Explanation Maximum time for a complete response to a single request

Range of values: 0 (= no limit) or 1 ... 32000 (in seconds)

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). An entry is also made in the diagnostics buffer (event ID B13CTD7_Diagnostics).

Each time a single request starts, MonitoringTime is reactivated.

Name **DataOutput**

Declaration INPUT

Data type ANY

Default P#P 0.0 VOID 0 (null pointer)

Explanation Data output area.

Range of values: 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!

Without a parameter setting, the default value (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer describes the data area in which the received data will be stored. This data area must be within a data block and its length can vary between 1 and 256 data double words.

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
- 2 bytes plus 1 word etc.

FB Dat256D_R stores the received data without further processing in the data area defined by DataOutput. The user program is responsible for evaluating and processing received data correctly.

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, namely, the area in which the changes were detected at the acquisition end.

How to read out the time stamp received with the data is described in the section "Notes on the SINAUT time stamp".

If the parameter setting is incorrect (null pointer, length greater than 256, data area not a data block), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 4). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

Name	**NewData**
Declaration	OUTPUT
Data type	DWORD
Default	FALSE
Explanation	New data was received.

Range of values:	Output (DWORD):	QD0 … QDn PQD0 … PQDn
	Bit memory (DWORD):	MD0 … MDn LD0 … LDn
	Data (DWORD):	DBm.DBB0 … n

Whenever the FB receives new data segments, the indication of status bits 1 to 22 appears in NewData according to the received data segments 1 to 22.

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 (messages), 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 (message) 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.

Name	**DataStatus**
Declaration	OUTPUT
Data type	BYTE
Default	FALSE
Explanation	Status of the receipt of a new data segment

Range of values:	Output (DWORD):	QB0 … QBn PQB0 … PQBn
	Bit memory (DWORD):	MB0 … MBn LB0 … LBn
	Data (DWORD):	DBm.DBB0 … n

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 data 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 the DataStatus parameter indicates an error.

  The status bit DataValid is set to FALSE.

  The SequenceState status shows the error according to the following table.

  An entry is made in the diagnostics buffer.

The following table shows the bit assignment of DataStatus:

Table 5- 5 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 completed without error.
		1	1. Segment of a sequence received
		2	2. or higher segment of a sequence received
		3	Errors: • Transmission sequence aborted • Monitoring time exceeded • Other error receiving

5.6 Blocks for optional expansion

5.6.1 FC ListGenerator300, FC ListGenerator400

Function

The FC ListGenerator is required in a CPU that receives messages 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 these typicals. Due to the missing destination information, the send message is automatically transmitted to all destinations for which a connection is configured. Such messages are therefore received without a destination address at the various destinations.

Note

Messages without a destination address that are sent to the partner over a TIM have the destination subscriber number added by the sending TIM, and sometimes several addresses if there are several destinations along the way. 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 message therefore contains the destination subscriber number but the destination object number is 0.

If the destination object number is not contained in the received message, FC Distribute, which is responsible for distributing the received messages, references an object reference list. Using the source address (source subscriber no. + source object no.) contained in every message, 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 must be included in the cyclic SINAUT program (in OB1) following FC BasicTask.

Note

There is an FC ListGenerator version for S7-300 (symbolic name, ListGenerator300) and for S7-400 (symbolic name ListGenerator400).

When creating the list, FC ListGenerator uses the addresses set in the parameters for the typicals that receive data. The parameters PartnerNo and PartnerObjectNo are mandatory for these typicals. These parameters are identical with the source address in the corresponding received message. 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 makes a request to all typicals that can receive data to enter their configured addresses 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.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

How it works

FC ListGenerator creates the list(s) after startup in three consecutive OB1 cycles:

1. 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.
2. 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.
3. 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 a list has not yet been created, it first searches for a free DB number; the first free DB number below the number for DB BasicData is used.

If a list already exists, FC ListGenerator checks to see if its present length is adequate 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 S7-300 and S7-400:

- For S7-300, a new DB is generated. The old DB remains in memory because S7- 300 has no delete function for data blocks. This DB must be deleted by the user with the programming device. If there is not enough memory on the CPU to be able to generate a new DB, the existing DB must be deleted by the user before restarting.
- In an S7-400, the existing DB is deleted and a new DB is generated with the same number and the new length.

5.6.2 FC TimeTask

Function

FC TimeTask is responsible for continuous date and time management on a CPU.

The FC has no parameters. It must be included in the cyclic SINAUT program (in OB1) following FC BasicTask.

FC TimeTask requires that the CPU is synchronized by a local TIM. This synchronization must be enabled in the ’Time Services’ tab in the Properties dialog for the corresponding TIM. See figure below.

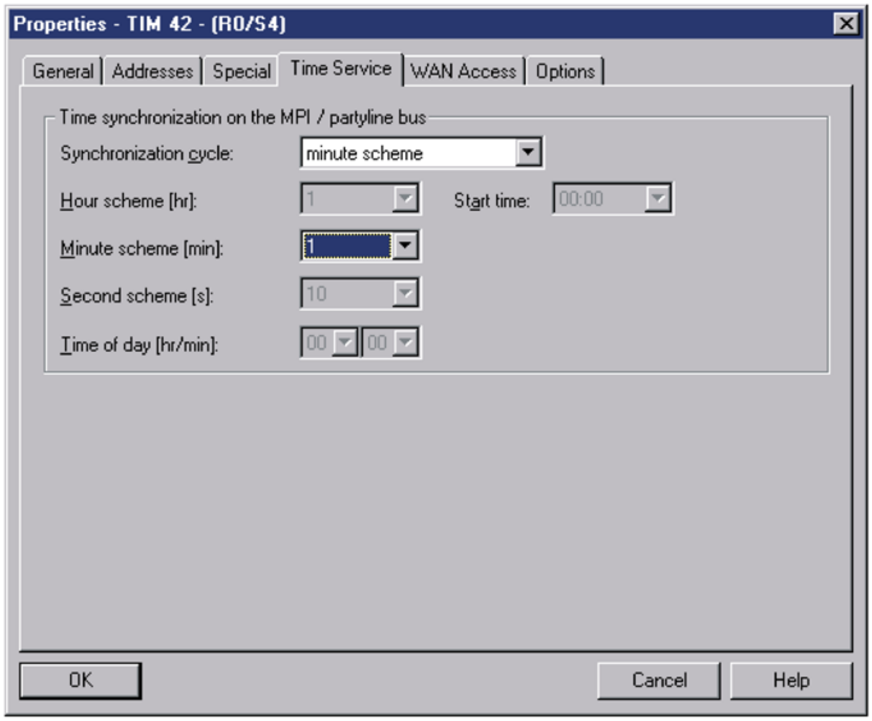
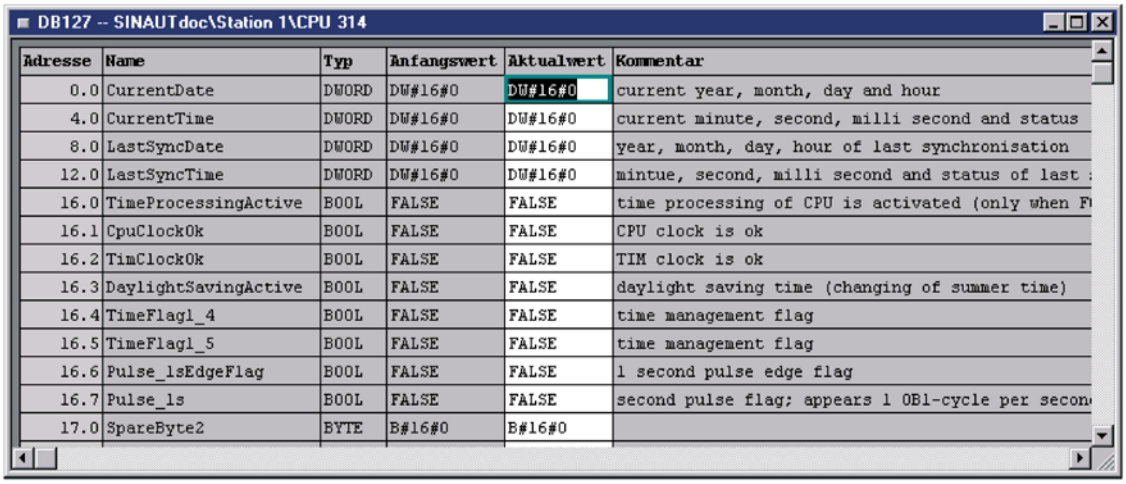


Figure 5-27 TIM parameter assignment tabs - setting the synchronization on the MPI / party line

For more detailed information on setting the synchronization, refer to the chapter "Configuration software for SINAUT ST7" in this manual, section 'Parameter assignment for TIM modules'.

After the CPU has started up, the TIM supplies the current date/time for the first time in an organizational message. Following this, the synchronization continues at the time interval specified by the configuration of the TIM (a one minute time scheme is recommended for synchronization on MPI/party line).

FC TimeTask sets the CPU clock to the synchronized time received from the TIM and reads this clock in every OB1 cycle. As long as it continues to advance and remains plausible, the read time is entered in the first two double words of DB BasicData and is marked as valid or invalid and as daylight-saving or standard time. All blocks take the current time from there as they need it. For example, data point typicals do this to time stamp their messages, as does FC Trigger to check whether a point in time configured for the FC has been reached or a specified time period has elapsed. This time is, of course, also available to the user program.



Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	CurrentDate	DWORD	DW#16#0	DW#16#0	current year, month, day and hour
4.0	CurrentTime	DWORD	DW#16#0	DW#16#0	current minute, second, milli second and status
8.0	LastSyncDate	DWORD	DW#16#0	DW#16#0	year, month, day, hour of last synchronisation
12.0	LastSyncTime	DWORD	DW#16#0	DW#16#0	mintue, second, milli second and status of last :
16.0	TimeProcessingActive	BOOL	FALSE	FALSE	time processing of CPU is activated (only when F
16.1	CpuClockOk	BOOL	FALSE	FALSE	CPU clock is ok
16.2	TimClockOk	BOOL	FALSE	FALSE	TIM clock is ok
16.3	DaylightSavingActive	BOOL	FALSE	FALSE	daylight saving time (changing of summer time)
16.4	TimeFlag1_4	BOOL	FALSE	FALSE	time management flag
16.5	TimeFlag1_5	BOOL	FALSE	FALSE	time management flag
16.6	Pulse_1sEdgeFlag	BOOL	FALSE	FALSE	1 second pulse edge flag
16.7	Pulse_1s	BOOL	FALSE	FALSE	second pulse flag; appears 1 OB1-cycle per secon
17.0	SpareByte2	BYTE	B#16#0	B#16#0	

Figure 5-28 DB BasicData, CurrentDate and CurrentTime

Table 5- 6 The exact assignment of the data words with data, time of day 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	Millisecond * 100	Millisecond * 10
	Data byte 7	Millisecond * 1	Time status

Table 5- 7 Assignment of the 4 time status bits:

2^0	0 = Date/time invalid 1 = Date/time valid
2^1	0 = Standard time 1 = Daylight saving time
2^2	(not used)
2^3	(not used)

In addition to using the time status, data bit 16.1 *CpuClockOk* also indicates whether the date/time is valid. Once the time of day is valid on the CPU, this bit is set to 1 by FC TimeTask. In the user program this bit can be directly queried using its symbolic name *"BasicData".CpuClockOk*.

5.6.3 FC Trigger

Function

The function block sets an output (memory bit, data bit or digital output) at a time set by the user or at a specified time interval. The block resets this output after an OB1 cycle.

The FC can be called (more than once) at any point in the cyclic SINAUT program (in OB1).

If a single program section or software function needs to be triggered using FC Trigger, it is recommended that the call executes directly before the block that for example should be run through conditionally due to a memory bit set by FC Trigger or that has to perform a certain function controlled by the memory bit (e.g. triggering transfer of a count value every 2 seconds).

if several (software) functions need to be activated at the same time, this can be implemented with FC Trigger. All query the memory bit set by FC Trigger. This works, however, only problem-free when the triggered blocks do not reset this memory bit. Remedy if triggered blocks reset the memory bit:

- FC Trigger is included often enough, in each case with the same time but a different output memory bit or
- after FC Trigger the user duplicates the set output memory bits by setting a suitable 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 SINAUT program and this is synchronized with the date and time at regular intervals by a local TIM. FC Trigger compares the time set for it with the current time only when the SINAUT time is marked as O.K. (if the time is valid bit 0 is set in data byte 7, the time status byte, in the DB BasicData).

The accuracy with which FC Trigger operates depends on the one hand on the accuracy if the SINAUT time and on the other on the OB1 cycle time. If the OB1 cycle time is lower than 1 s (this is normal) the output is set exactly at the set second value (with the inaccuracy of the OB1 cycle time of less than 1 s). If the OB1 cycle time is higher than 1 s, the FC operates with a tolerance window of 4 s, in other words if the FC is processed too late but still within 4 s of the set time or time interval, the output is still set.

The parameter flag to be set for the FC is set at the same time as the output and reset 5 s after the set time/time interval. No dummy parameter can be used for the edge memory bit (!) and it must not be reset by the user program.

After the following explanation of the parameters you will find several examples that illustrate how a certain time or time interval can be set with FC Trigger.

Explanation of the parameters

Name: **IntervalMode**
Declaration: INPUT

Data type BOOL
Explanation In the FC a time interval is set
Range of values: TRUE or FALSE
FALSE = No, i.e. a point in time is set.
TRUE = Yes, i.e. a time interval is set
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 below.

Name: **Second_Day**
Declaration: INPUT
Data type WORD
Explanation Specifies the values for seconds and day.
Further explanation: refer to the parameter Month_Year below.

Name: **Month_Year**
Declaration: INPUT
Data type WORD
Explanation Specifies the values for month and year.
Range of values: 00 ... 99, or FF
Each parameter is divided into two, 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
Explanation Trigger output.

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

The output is set for the duration of one OB1 cycle when the time or time interval set for Hour ... Year is reached.

Name:	**Flag**
Declaration:	IN_OUT
Data type	BOOL
Explanation	Edge memory bit for the TriggerOutput output.

Range of values:	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

No dummy memory bit may be specified! The edge memory bit must also not be reset by the user program.

Note

The parameter 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.

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 when the check is made as to whether the point in time has been reached. This allows time intervals to be set in certain areas (although IntervalMode = 0). See 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:

1. IntervalMode : FALSE The output TriggerOutput is set once
   Hour_Minute : W#16#0645 at 06:45:12 on 04.02.91.
   Second_Day : W#16#1204
   Month_Year : W#16#0291
2. IntervalMode : FALSE The output TriggerOutput is set every
   Hour_Minute : W#16#0600 day at 6:00:00.

Second_Day : W#16#00FF
Month_Year : W#16#FFFF

3. IntervalMode : FALSE The output TriggerOutput is set every
   Hour_Minute : W#16#0600 1. of a month at 06:00:00
   Second_Day : W#16#0001.
   Month_Year : W#16#FFFF
4. IntervalMode : FALSE The output TriggerOutput is set every
   Hour_Minute : W#16#0600 year on October 1st
   Second_Day : W#16#0001 at 06:00:00.
   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:

1. IntervalMode : TRUE Every 6 hours the output
   Hour_Minute : W#16#06FF TriggerOutput is set at:
   Second_Day : W#16#FFFF 00:00:00, 06:00:00, 12:00:00 and
   Month_Year : W#16#FFFF 18:00:00.
2. IntervalMode : TRUE Every 30 minutes the output
   Hour_Minute : W#16#FF30 TriggerOutput is set at:
   Second_Day : W#16#FFFF 00:00:00, 00:30:00, 01:00:00,
   Month_Year : W#16#FFFF 01:30:00, 02:00:00, 02:30:00 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 partly dependent on the IntervalMode parameter. If the parameter setting is incorrect , only during startup an error message to this effect is entered in the diagnostics buffer (event ID B113). The CPU does not change to STOP. Afterwards the FC only checks the parameters (without error message) until the parameter assignment 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). Depending on the IntervalMode parameter causes of the diagnostics message can be:
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 parameter settings are permitted:

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:

1. The permitted ranges of values for the parameters Hours, Minutes and Seconds were not kept to. Apart from FF, the following parameter settings 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.
2. 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.

5.6.4 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 messages (Cnt01D_S, Cnt04D_S and ZTZ01, ZTZ02, ZTZ03).

The acquisition of the count pulses is time-controlled To do this the FC PulseCounter must be included in a cyclic interrupt, 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 'Time-controlled SINAUT program in a cyclic interrupt OB.

Explanation of the parameters

Name:	**InByte**		
Declaration:	INPUT		
Data type	BYTE		
Explanation	Input byte for count pulses.		
	Range of values:	Input bytes	PEB0 ... PEBn
		Memory bytes	MB0 ... MBn LB0 ... LBn
		Data bytes	DBm.DBB0 ... n
	The parameters for inputs for count pulse acquisition can can be set in bytes.		

Note

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.

Name:	**EnableMask**
Declaration:	INPUT
Data type	BYTE
Explanation	Enable mask for the counting inputs

Range of values: B#16#00 to B#16#FF

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 byte blocked for acquisition
1 = Input bit enabled for acquisition

Note

The input of the mask can only be made 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: EnableInMask : 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
Explanation	Pulse counter

Range of values: Z0 or Z1 ... Zn (n depends on the CPU)

For every 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 count value function blocks Cnt01D_S, Cnt04D_S and ZTZ01, ZTZ02, ZTZ03. 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 EnableInMask : = B#16#83

CntIn_0 : = C10
CntIn_1 : = C11
CntIn_2 : = C0
CntIn_3 : = C0
CntIn_4 : = C0
CntIn_5 : = C0
CntIn_6 : = C0
CntIn_7 : = C12

5.6.5 FC Safe

Function

The block ensures reliable input of commands and setpoints. If an input is pending, the FC checks whether there is only one input pending in the current OB1 cycle and then enables the block reading in.

In the CPU in which commands and/or setpoints are acquired, the FC Safe needs to be called in the cyclic SINAUT program (in OB1) once on completion of all command and setpoint FBs. For more information on the program structure refer to the section 'The cyclic OB1 program for a master station'.

The FC has separate monitoring for commands and setpoints that are entered via hardware (input modules) or software (Operator Panel (OP) etc.). These two types of input can be used at the same time. The FC enables the hardware or software 'rail' separately, possibly at the same time. The principle however applies that per 'rail' only one command or setpoint input may be detected.

For the hardware input, there is a further condition: the input must be applied constantly for a certain time. This delay time can be set in Fc Safe with the parameter InputDelayTime. Only when the currently entered command or setpoint is applied unchanged for the set delay time and during this time no other command or setpoint input when the time has elapsed the enable is given. The actual putting together of the command or setpoint message is handled by the block that read in the command or setpoint.

For the hardware input the FC Safe provides two display bits:

InputOK :	Has 1 signal as soon as the current input was enabled. The display bit goes off when the input is reset, in other words when the command key is released or with setpoint input as soon as the key at the EnterInput input is released.
InputError :	Has 1 signal as soon as an input error is recognized within the hardware 'rail'. Either two or more command and/or setpoint inputs were recognized at the same time or at one of the inputs a 1 signal was detected over a longer period of time, i.e. the input is 'hanging'. This longer monitoring time can be set with the MaxInputTime parameter in FC Safe.

Via the output GlobalCmdOutputError FC-Safe shows a command output error detected in a station. A command output error occurs at the receive end only when the content of the two command bytes in the received message was not identical or when more than one bit was set in the command byte. If such an error is detected, this is sent by the station with an organizational message to the subscriber that sent the command message. FC Safe on this subscriber then indicates the problem at the output GlobalCmdOutputError. This is a group display. When an error is detected, the output remains set to the 1 signal until the user requests the group signal to be reset via the ResetError input.

Explanation of the parameters

Name:	**InputDelayTime**
Declaration:	INPUT
Data type	INT
Explanation	Delay time in ms for commands and setpoints that are input via hardware. Range of values: 0 or 1 ... 32000 [ms] A delay time of at least 1000 ms is recommended. 0 can be specified if the parameter is not required. For more detailed information on this parameter, refer to the section Function.

Name:	**MaxInputTime**
Declaration:	INPUT
Data type	INT
Explanation	Monitoring time in ms for commands and setpoints that are input via hardware. Range of values: 0 or 1 ... 32000 [s] A monitoring time of at least 30 s is recommended. 0 can be specified if the parameter is not required. For more detailed information on this parameter, refer to the section Function.

Name:	**ResetError**
Declaration:	INPUT
Data type	BOOL
Explanation	Input for resetting the output GlobalCmdOutputError.

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

If the parameter is not required, a memory bit or data bit is specified that always has a 0 signal.

For more detailed information on this parameter, refer to the section Function.

Name:	**InputOK**
Declaration:	OUTPUT
Data type	BOOL
Explanation	Hardware command or setpoint input was correct.

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

If the parameter is not required, a memory bit or data bit is specified that can be used as a scratchpad memory bit.

For more detailed information on this parameter, refer to the section Function.

Name:	**InputError**
Declaration:	OUTPUT
Data type	BOOL
Explanation	There is an error in the area of the hardware command or setpoint input.

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

If the parameter is not required, a memory bit or data bit is specified that can be used as a scratchpad memory bit..

For more detailed information on this parameter, refer to the section Function.

Name:	**GlobalCmdOutputError**
Declaration:	OUTPUT
Data type	BOOL
Explanation	Group message: in a station a command output error was detected.

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

If the parameter is not required, a memory bit or data bit is specified that can be used as a scratchpad memory bit..

For more detailed information on this parameter, refer to the section Function.

5.6.6 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 SINAUT program (in OB1).

If you want to monitor the status of more than 8 subscribers, an appropriate number of PartnerStatus FCs must be included in the SINAUT program.

A SINAUT subscriber (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 (or 0): Subscriber is disrupted (or the corresponding input parameter is not being used, i.e. configured as 0, or subscriber is unknown).

TRUE (or 1): Subscriber OK

Explanation of the parameters

Name:	**Partner1 ... Partner8**
Declaration:	INPUT
Data type	INT
Explanation	SINAUT subscriber number of the subscriber to be monitored Range of values: 0 or 1 ... 32000 0 = Dummy value for unrequired parameter 1 ... 32000 = Number of the subscriber to be monitored If a set subscriber number not found in the administration (in DB-BasicData), an entry to this effect is made in the diagnostic buffer only during startup (event ID B101). The CPU does not change to STOP. The status of a correctly configured subscriber is indicated in the PartnerStatus output byte; unknown subscribers are not processed until the parameter error has been corrected. Their status bits are set to 0.

Name:	**PartnerStatus**
Declaration:	OUTPUT
Data type	BYTE
Explanation	Output byte for indicating the status of a subscriber to be monitored.

Range of values:	Output bytes	QB0 ... QBn PQB0 ... PQBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bit bytes	DBm.DBB0 ... n

The assignment of the status bits in the PartnerStatus output byte in relation to the parameters **Partner1 ... Partner8**:

	.7	.6	.5	.4	.3	.2	.1	.0
Partner X	8	7	6	5	4	3	2	1

Status:
0 = subscriber **Partner X** disrupted or parameters not set or unknown
1 = subscribe **Partner X** OK.

5.6.7 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 in the cyclic program (OB1) after calling the FC BasicTask, refer to the section "Cyclic SINAUT program in OB1".

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 SINAUT program.

Explanation of the parameters

Name:	**Partner**		
Declaration:	INPUT		
Data type	INT		
Explanation	SINAUT subscriber no. of the partner		
	Range of values:	1 ... 32000 (subscriber number)	

Name:	**PathStatus**		
Declaration:	OUTPUT		
Data type	BYTE		
Explanation	Output byte for displaying the path status of the connection to the partner		
	Range of values:	Output byte	QB0 ... QBn PQB0 ... PQBn
		Memory byte	MB0 ... MBn LB0 ... LBn
		Data byte	DBm.DBB0 ... n

Table 5- 8 Explanation of the status bits in the PathStatus output byte

Bit 6 + 7	Bit 4 + 5	Bit 2 + 3	Bit 0 + 1
Configured interface		**Path status**	
No. for substitute path	No. for main path	Substitute path (2nd path)	Main path (1st path)

Configured interface

The TIM interfaces "Ethernet 1" (IE1), "Ethernet 2" (IE2), WAN1 and WAN2 are numbered through from 0 .. 3 (decimal):

- 0 = "Ethernet 1" (IE1)
- 1 = "Ethernet 2" (IE2)
- 2 = WAN1
- 3 = WAN2

Table 5- 9 Status table: Coding of bits 4 + 5 or bits 6 + 7

Status of bit 5 (7)	Status of bit 4 (6)	Meaning
0	0	Coding for interface IE1 (decimal): No. 0)
0	1	Coding for interface IE2 (decimal: no. 1)
1	0	Coding for interface WAN1 (decimal: no. 2)
1	1	Coding for interface WAN2 (decimal: no. 3)

Path status

- 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 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

Example of coding options

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 example for 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

5.6.8 FC PartnerMonitor

Function

FC PartnerMonitor displays important status information about a SINAUT subscriber (see PartnerStatus parameter). The FC can also be used to trigger a general request to the subscriber and to establish and disconnect a permanent connection to the subscriber.

The FC can be called at any point in the cyclic SINAUT program (in OB1).

If you want to monitor and control more than one subscriber, include an appropriate number of PartnerMonitor FCs in the SINAUT 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 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, i.e. when the local TIM is a master TIM.

Explanation of the parameters

Name:	**PartnerNo**
Declaration:	INPUT
Data type	INT
Explanation	SINAUT subscriber number of the subscriber to be monitored and controlled. Range of values: 1 ... 32000 [ms] If the set PartnerNo is not found in the administration (in DB-BasicData), an entry to this effect is made in the diagnostic buffer only during startup (event ID B101). The CPU does not change to STOP. The status of a correctly configured subscriber is indicated in the PartnerStatus output word and the control inputs are processed. Unknown subscribers are not processed until the parameter assignment error has been corrected. The PartnerStatus output word remains set to 0 during this time.

Name:	**MaxConnectTime**
Declaration:	INPUT
Data type	INT

Explanation Maximum connection duration for a permanent connection.

Range of values: 0 (= no limit) or 1 ... 480 [minutes]

If the time specified here is greater than 0, it 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

Explanation Output word to indicate the status of the subscriber to be monitored.

Range of values:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data bit words	DBm.DBW0 ... n

If you do not require the parameter, simply leave it open.

The meaning of the status bits in the PartnerStatus output word:

	Bit no.															
PartnerStatus	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0

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 disrupted 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 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.

Note

Bit 11 cannot be displayed in conjunction with TimeTask version ≤ 1.6.

Note

The following parameters, GeneralRequest, PermanentCall_On and PermanentCall_Off are in/out parameters (declaration IN_OUT). 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
Explanation	Input for triggering a general request to the subscriber specified with PartnerNo.

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

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), the user is responsible for resetting the signal at the input. This must be done before ending the currently running general request otherwise another general request is triggered immediately.

Name:	**PermanentCall_On**
Declaration:	IN_OUT
Data type	BOOL
Explanation	Input for triggering a permanent connection to the subscriber specified with PartnerNo.

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

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), the user is responsible for resetting the signal at the input. This should be done at the latest before terminating 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 word (see PartnerStatus parameter).

Note

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
Explanation	Input for triggering termination of an existing permanent connection to the subscriber specified with PartnerNo.

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

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), the user is responsible for resetting the signal at the input. This should be done at the latest before establishing the 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 word (see PartnerStatus parameter).

Note

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.

Note

A permanent connection cannot be terminated by a station. This control input cannot therefore be used when FC PartnerMonitor is used in a station.

5.6.9 FC ST7ObjectTest

Function

Calling FC-ST7ObjectTest in the programming error OB121 prevents a CPU stop, if the CPU receives a message 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 a different block, the user can specify what should happen then in the StopInOtherCases parameter: STOP or continue running.

For more information on the programming error OB121 and background information relating to the use of FC ST7ObjectTest: refer to the section 'SINAUT test program in the programming error OB21'.

Explanation of the parameters

Name:	**StopInOtherCases**	
Declaration:	INPUT	
Data type	BOOL	
Explanation	The CPU should go to STOP in other error sitations.	
	Range of values:	TRUE or FALSE
	With this parameter it is possible to specify what should happen in other error situations: stop or continue running if OB121 was called because another data block or an FB or FC is missing.	

5.6.10 FB SMS_Control

Function

The SMS typical FB-SMS_Control is used to send event-driven SMS messages (SMS: Short Message Service) to a mobile phone specified in the parameter assignment).

The configured SMS texts and the phone number are located in a DB SMS_Data that must exist on the CPU. This DB is generated using the SINAUT ST7 configuration tool (see chapter *Configuration software for SINAUT ST7*, section *Parameters for individual subscribers*). The texts are assigned 1:1 in an array of digital inputs, memory bits or data of a data block.

The SMS typical works internally with a job list. This contains all SMS jobs that are detected due to a change in the array. The SMS jobs remain in the job list until they are completed (with or without error). Afterwards they are deleted from the job list.

An SMS job counts as being completed when

- an SMS message could be transferred successfully to the SMS control center (SMS-C) via a fixed line network access and this in turn successfully transferred this to the mobile phone,
- an SMS without mandatory acknowledgment could be transferred to the SMS-C via a direct GSM access,
- an SMS with mandatory acknowledgment could be transferred to the SMS-C via a direct GSM access and was acknowledged by the mobile phone.

An SMS job is completed with error when

- an internal monitoring time has expired without the SMS message being supplied to the SMS-C,
- The period of validity of the SMS job has expired in the SMS-C and CPU without the SMS message being transferred by the SMS-C to the mobile phone or if acknowledgment is mandatory the SMS message was not acknowledged within this time.

If an SMS message is completed with an error, a diagnostic message is entered in the diagnostics buffer of the CPU.

The jobs remain in the job list of the SMS typical until they are completed correctly or with an error, or the SMS typical is disabled (Enable = FALSE).

If the SMS typical is disabled while there are still SMS jobs in the job list, every SMS job that completes is noted in the diagnostics buffer of the CPU. A delete job for all still active SMS jobs is transferred to the TIM responsible.

Note

If an SMS triggering signal occurs several times, the SMS message is only sent again when the corresponding SMS job was completed first.

Operation with two mobile phone numbers

The SMS typical provides the option of working with a main and a substitute mobile phone. If access to the SMS-C is via an M20-GSM module, the two mobile phones can also belong to different networks.

After the set monitoring time has expired, all SMS messages already sent previously but not yet successfully completed are sent again in chronological order to the substitute mobile phone and the monitoring time is restarted.

If the SMS typical cannot complete the SMS jobs successfully and within the time allowed via the substitute mobile phone, it switches back to the main mobile phone to repeat all incomplete SMS messages, etc.

Each SMS message that is repeated is noted in the diagnostics buffer at the start of the repetitions. As the counterpart of this, for each repeated SMS that completes successfully, a success message is entered in the diagnostics buffer.

The switchover between the main and substitute mobile phone is not made for each individual SMS message separately, but always affects all SMS messages of a DB SMS_Data being processed.

Whenever no more SMS messages are pending in the SMS typical, internally there is an automatic switchover back to the main mobile phone number so that when the next event occurs, the corresponding SMS message is once again sent to the main mobile phone.

Note

There is no switchover to a disrupted SMS-C. The SMS typical then behaves as if only 1 mobile phone number was set in the parameters. There is only ever one of the two mobile phone numbers active, the other remains passive.

Meaning of the period of validity

The period of validity for SMS jobs can be set with the SINAUT ST7 configuration tool (see Subscriber administration, *SMS configuration* tab, *SMS DB configuration* dialog, parameter *Valid period*).

The time for the period of validity is started as soon as the SMS could be delivered to the SMS-C. Within the period of validity, the SMS-C keeps trying to deliver the SMS transferred to it to the mobile phone. The repetition intervals can, however, be very irregular.

After the period of validity has expired, the relevant SMS is deleted in the SMS job list of the SMS typical, and a diagnostics message is entered in the diagnostics buffer of the CPU. Normally the SMS is also deleted in the SMS-C at this time. How the period of validity in the SMS-C is kept to differs considerably from provider to provider.

Note

If the SMS-C can only be reached by a fixed line network access, the period of validity in the SMS-C is always approximately 48 hours. It is therefore independent of the "Valid period" parameter. Here, the set period of validity is only valid for the CPU and can be set to a maximum of 11 hours and 55 minutes.

Repetition mechanisms

If the SMS job has been transferred from the CPU to the TIM, the repeated transfer is the responsibility of the TIM if the SMS-C cannot be reached.

If the TIM was able to deliver the SMS message to the SMS-C, the SMS-C automatically takes over the repetition if the mobile phone is temporarily unreachable.

If the SMS message has been transferred to the SMS-C but this cannot deliver it to the mobile phone, after the period of validity has expired the typical switches over to the substitute mobile phone if this is configured.

The same repetition mechanism is used with an SMS message with mandatory acknowledgement if this SMS could be transferred to the mobile phone but there is no acknowledgement within the current period of validity. Once again the SMS typical switches over to the substitute mobile phone if this exists.

If an SMS message can also not be delivered to the substitute mobile phone or there is no acknowledgement with messages requiring acknowledgement, there is a constant switchover between the main and substitute mobile phone. This occurs until a delivery is possible or an acknowledgement is received or until the SMS typical is disabled (ENABLE = FALSE).

Other functions

Apart from sending and managing SMS messages, the SMS typical processes a further two globally valid functions. Global functions because their execution is valid for all SMS jobs in conjunction with the DB SMS_Data that is processed by the SMS typical. The two global functions are as follows:

- Status query of all SMS jobs of the DB SMS_Data processed by the typical.
  The typical transfers the status query to the TIM that is responsible for transfer of the SMS message. This returns the current status of every SMS job still stored on it to the typical (status messages 2, 3 or 8). The TIM completes the sequence of status messages

with status message 11 that says that there is no further status information available. This status query function is only run through when the CPU starts up and only if the SMS typical is enabled at this time (ENABLE = TRUE).

- Deletion of all SMS jobs of the SMS_Data DB processed by the typical.
  The deletion takes place at several points:
  - On the CPU: All jobs stored in the SMS_Data DB.
  - On the TIM: All jobs stored on the TIM for the SMS_Data DB.
  - In the SMS-C: All jobs stored in the SMS-C for the SMS_Data DB (only when the SMS-C can be reached via a fixed line network access).
- The delete function is executed automatically by the typical in the following situations:
  - When the instance DB belonging to the typical is initialized.
  - When the SMS_Data DB to be processed by the typical is initialized.
  - When a status change is recognized at the ENABLE input of the typical.

The status query and the delete function are monitored over time. The duration of the monitoring time is decided by the SupervisionTime (default 900 seconds) of the corresponding subscriber object in the BasicData DB.

If the monitoring time expires before the status query or the delete function could be ended correctly, a message is entered in the diagnostics buffer of the CPU and the group message "Monitoring time expired" is set in the Status output parameter of the SMS typical.

As long as the status query or the delete function are running, the processing of the SMS messages is postponed and it is not possible to trigger a new SMS message.

The current status of the status query or of the delete function is displayed in the Status output word of the SMS typical.

Explanation of the parameters

Name:	**MessageObjectDB**
Declaration:	INPUT
Data type	BLOCK_DB
Default	-
Explanation	The SMS_Data DB to be processed by the typical needs to be specified. Range of values: DB1 ... DBn (n depends on the CPU type) The SMS_Data DB contains all data and SMS texts for processing a group of message or alarm bits. The user can configure this data block simply using the SINAUT ST7 configuration tool and fill it with the required data and texts.

Name:	**Enabled**
Declaration:	INPUT
Data type	BOOL
Default	FALSE
Explanation	Enable block processing.

Range of values:	TRUE or FALSE	
	No parameter specified: Default TRUE is valid	
	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

Via this input, you switch SMS processing on or off.

If you use the block, enable block processing by initially setting this parameter manually to TRUE. Only after this is the send function active for SMS.

If processing is disabled after an enable, initially all not yet delivered, unacknowledged SMS jobs are deleted. Only after this does the typical change to the actual disabled status.

Example of an application:
Via this input a time-controlled enable and disable by the user program can be implemented if the SMS messages are only sent evenings and nights and/or at the weekend to the standby personnel..

Name:	**Status**
Declaration:	OUTPUT
Data type	WORD
Default	0
Explanation	Based on the status bits in this word, the SMS typical can be monitored.

Range of values:	Output words	QW0 ... Qwn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

If you do not require the parameter, simply leave it open.

The meaning of the status bits in the Status output word:

	Bit No.															
Status	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0

Bit.0	To mobile phone numbers exist
Bit.1	At least one SMS message being processed
Bit.2	At least one SMS message not yet acknowledged
Bit.3	At least one SMS message will be repeated
Bit.4	DB-specific deletion job running
Bit.5	DB-specific status query job running
Bit.6	Substitute mobile phone number is activated

Bit.7	Supervision time exceeded. Delete job or status query ended with error (more details in the diagnostics buffer)
Bit.8	SMS processing activated
Bit.9	SMS processing temporarily halted (for more details see Operating Mode, bit 12...15)
Bit.10	Not used
Bit.11	Not used

Bit	.15	.14	.13	.12	Operating Mode: Status of the current processing
	0	0	0	0	0 = Turned off
	0	0	0	1	1 = Enabled, SMS processing ready / running
	0	0	1	0	2 = Blocked: Global status query running
	0	0	1	1	3 = Blocked: Global deletion of an SMS-C running
	0	1	0	0	4 = Blocked, initialization of the instance DB plus SMS_Data DB running
	0	1	0	1	5 = Blocked: Global deletion of all SMS-Cs running
	..	..	..	..	6-11 = Not used
	1	1	0	0	12 = Blocked, global SMS-C switchover running
	1	1	0	1	13 = Blocked, current SMS-C disrupted or unreachable
	1	1	1	0	14 = Blocked, keyword at the end of the SMS_Data DB is wrong
	1	1	1	1	15 = Not used

No SMS messages can be processed when:

- DB-specific deletion is running
- A DB-specific status query is running, e.g. when the CPU restarts
- The instance DB or the SMS_Data DB is being initialized
- There must be a reaction to changes at the Enable input
- A switchover between the main and substitute mobile phone is running or a mobile phone number change with renewed SMS transfer
- Access to the SMS-C or the SMS-C itself is disrupted
- The keyword in the SMS_Data DB cannot be found

This temporarily blocks the typical which sets bit 9.

Name:	**ReloadMobilPhoneNo**
Declaration:	IN_OUT
Data type	BOOL
Explanation	If the phone number of one or both mobile phones is to be changed online, the new phone numbers can be activated via this input.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid

Input	I 0.0 ... I n.7
Memory bit	M 0.0 ... M n.7
Data bit	DBm.DBX 0.0 ... n.7

After the user has changed the phone number data of the mobile phones in the
SMS_Data DB, the input can be set and the new phone number data activated online.

Otherwise the typical adopts the phone numbers stored in the SMS_Data DB whenever the typical detects a signal change at the ENABLE input from FALSE to TRUE.

If you do not require the parameter, simply leave it open.

Note

The parameter 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.

5.6.11 DB SMS_Data

Function

The SMS_Data DB contains all data and SMS texts that the SMS typical FB SMS_Control needs to send SMS messages and to manage them. The user can configure this data block simply using the SINAUT ST7 configuration tool and fill it with the required data and texts.

Based on the information stored in the SMS_Data DB, the user can read out the current status separately for each SMS message and can determine when the message was last transferred as a coming or going message (assuming time stamps are used) and with messages with mandatory acknowledgement when was last acknowledged.

Structure of the SMS_Data DB

The DB consists of a trunk with globally required information and one or more sections, known as SMS objects, that contain the data required for each SMS message. The data of the SMS object is defined in UDT125 ShortMessageObject that exists in the TD7 library.

The global information in the trunk of the DB includes, among other things

- The SINAUT subscriber number of the main SMS control center (main SMS-C) and possibly a substitute SMS control center.
- The phone number of the main mobile phone and, if it exists, the phone number of the substitute mobile phone.
- The start address of the gapless array of digital inputs, memory bits or data bits). If a bit in this array changes, the corresponding SMS message is sent.

Structure of an SMS object

The structure of an SMS object is specified in UDT125 ShortMessageObject For every bit in the specified bit array a ShortMessageObject is included in the SMS_Data DB.

The following among other things are stored here:

- The text of the SMS message to be sent,
- When the SMS message was last sent with a coming or going message,
- The current, typical-internal processing status of the SMS message,
- The current processing status of the SMS message on the TIM / SMS-C including the time stamp belonging to the last status message and the ID number assigned by the TIM (TIM ID) that allows the reference to the diagnostics buffer entries (on the TIM and CPU),
- Various entries for organization and processing required by the typical.

Each SMS object consists of a maximum of 82 data words (with max. 122 character text). The structure is as follows:

Table 5- 12 Structure of an SMS object in the SMS_Data DB

Number of data words	Short name		Meaning
Max. 62	SMS text		SMS text string, max. 122 characters long
1	Year	Month	Acquisition time stamp for the last sent coming SMS message
1	Day	Hours	
1	Minute	Second	
1	Year	Month	Acquisition time stamp for the last sent going SMS message
1	Day	Hours	
1	Minute	Second	
1	CurrentMessageStatus		Typical-internal SMS processing status (see description of CurrentMessageStatus)
1	Status of the coming SMS message:	Status of the going SMS message:	Last SMS status that the CPU received from the TIM or SMS-C (see description of the status coming SMS message)
1	TIM-ID for coming SMS messages		ID number assigned by the TIM for coming SMS messages
1	TIM-ID for going SMS messages		ID number assigned by the TIM for going SMS messages
1	Year	Month	Time stamp of the status message last transferred from the TIM to the CPU
1	Day	Hours	
1	Minute	Second	
1	LastControlMessage		Last control instruction from the CPU to the TIM (without time stamp)
1	RemainingValidPeriodP		Remaining period of validity of the coming SMS message
1	RemainingValidPeriodN		Remaining period of validity of the going SMS message
1	SupervisionTime		Remaining monitoring time by which a coming or going SMS message must have been transferred to the SMS-C

Number of data words	Short name	Meaning
1	PreviousMessageAddress	Previous address (for chained job list)
1	NextMessageAddress	Following address (for chained job list)
1	MessageNumber	Message number of the SMS message for identification and diagnostics

CurrentMessageStatus

The meaning of the status bits in CurrentMessageStatus:

	Bit No.															
Status	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0

Bit	.1	.0	Status of the SMS after coming signal edge:
	0	0	0 = Job completed without error
	0	1	1 = TIM confirms: (coming) SMS message received
	1	0	2 = SMS-C confirms: (coming) SMS message received
	1	1	3 = Job completed with error

Bit	.3	.2	Status of the SMS message after going signal edge (only relevant when an SMS message is to be sent with a coming and going message):
	0	0	0 = Job completed without error
	0	1	1 = TIM confirms: (going) SMS message received
	1	0	2 = SMS-C confirms: (going) SMS message received
	1	1	3 = Job completed with error

Bit.4	Edge memory bit On an edge change, the SMS is entered in the job list
Bit.5	Coming SMS message: The job is being repeated
Bit.6	Going SMS message: The job is being repeated
Bit.7	No signal edge processing as long as the SMS job is running (a coming or going SMS message is not sent again, if the previous identical message is not yet completed).
Bit.8	SMS is disabled
Bit.9	SMS requires acknowledgment
Bit.10	SMS is waiting for acknowledgment
Bit.11	SMS is preceded by +/- for coming or going
Bit.12	Not used
Bit.13	Not used
Bit.14	Reserved
Bit.15	Reserved

Status coming / going SMS message

The SMS processing status valid on the TIM is reported by the TIM to the CPU. The SMS typical enters the return messages´ belonging to a coming SMS message in the byte "Status coming SMS" and those belonging to a going SMS message in the byte "Status going SMS".

The status 1 is an exception here. It is not reported by the TIM to the CPU, but is formed by the SMS typical itself and entered in the status byte for coming or going SMS messages.

You will find the following classifications in the explanations that follow:

Status
status information: is transferred by the TIM to the CPU if there is a status change or after a status query.

Pulse
"Pulse" information: is transferred to the the CPU once due to an event on the TIM, it cannot be queried.

Fixed network
Status information valid with fixed network access to the SMS-C via an analog phone network or ISDN network.

Mobile network
Status information valid with a mobile network access to the SMS-C via an M20 module.

The following status entries are possible:

1. Job monitoring time expired, SMS message could not transferred to the to the SMS-C in time.
   Status, Fixed network, Mobile network
2. SMS message not yet transferred to SMS-C.
   Status, Fixed network
   Comes as the answer to a status query, cannot be queried at a mobile network access.
3. SMS message stored on SMS-C but not yet delivered to a mobile phone.
   Status, Fixed network
   This status cannot be queried at a mobile network access.
4. SMS message successfully delivered to a mobile phone.
   Counts as a finished without error message.
5. SMS message cannot be delivered to the mobile phone.
   Pulse, Fixed network
   Counts as a finished with error message.
6. SMS message successfully delivered to SMS-C.
   Pulse / Status, Mobile network
   Pulse for messages without mandatory acknowledgment. Then counts as a finished without error message.
   Status for messages with mandatory acknowledgment. The TIM waits for an acknowledgment from the mobile phone or a delete job. Does not count as a finished message.
7. SMS message was acknowledged by the mobile phone
   Counts as finished without error message.
8. SMS message could not be transferred to the SMS-C.
   Status, Mobile network

The TIM handles the repetition of the transfer until the SMS message can be delivered or the SMS is deleted.
Possible causes are:

- "ERROR" message from the M20 module
- Monitoring time expired on the TIM (the M20 module has not responded)
- Disruption of the GSM network

9. SMS message successfully deleted.
   Pulse, Fixed network, Mobile network
   Response of the TIM to a global delete job transferred by the CPU to the TIM. For every SMS job that was deleted on the TIM, the TIM sends a separate status message 9 to the CPU.
   With fixed network access, the SMS message is deleted on the SMS-C and in the record keeping of the TIM.
   With mobile network access it is not possible to delete the message on the SMS-C. It is only deleted in the record keeping of TIM.
10. No entries to be deleted or all entries deleted.
    Pulse, Fixed network, Mobile network
    Response of the TIM to a global delete job transferred by the CPU to the TIM. In this case all entries for an
    SMS_Data DB have been deleted or there were not entries present for this SMS_Data DB.
11. No further entry present or no entry at all present.
    Pulse, Fixed network, Mobile network
    Final response of the TIM to a global status query transferred by the CPU to the TIM.
    Previously the TIM had sent the current status of all still active entries to the CPU.
12. The SMS message could not be deleted on the SMS-C. Only the entry in the TIM record-keeping is deleted.
    Pulse, Fixed network
    Response of the TIM to a specific delete job for a single SMS message. Delete job was transferred by the CPU to the TIM.
    In this case, a message on the SMS-C could not be deleted. The SMS message is probably not yet on the SMS-C or the SMS message has already been delivered.
    Therefore only the corresponding entry in the TIM record-keeping is deleted.

5.7 Test blocks

5.7.1 FC TestCopy

Function

With the aid of FC TestCopy the message traffic between SINAUT ST7 subscribers can be noted in extracts or the entire traffic can be noted. With such masks to be set in the DB TestCopy individual message types can be filtered out and then copied from the send or receive buffer for further evaluation in the DB TestCopyData.

Send and receive messages are all stored in the same DB TestCopyData. This makes it simple to track the chronological order of the copied send and receive messages.

The functions SendCopy (= copy send messages) or RecvCopy (= copy receive messages) can be activated individually or both at the same time, however only a common mode is possible for both communication directions. The modes are set in data byte DBB0 of the DB TestCopyData and the following assignment applies:

- DBB0, bit 0...3: Mode for RecvCopy function (mode 0, 1, 2 or 3)
- DBB0, bit 4...7: Mode for SendCopy function (mode 0, 1, 2 or 3)

As long as a mode > 0 is set in the less significant half byte, this always applies to both communication directions. Only if BA = 0 in bit 0...3 (no RecvCopy function required), the value in DBX 4...7 applies (only SendCopy function). Exception: To delete the DB TestCopyData, DBB0 must have FF written to it, 0F is not enough!

Examples:

DBB0 = 00h; no TestCopy function activated

DBB0 = 03h; only RecvCopy function, mode 3, no SendCopy fct. required

DBB0 = 30h; only SendCopy function, mode 3, no RecvCopy fct. required

DBB0 = 33h; RecvCopy function and SendCopy function, mode 3, required

DBB0 = 23h; RecvCopy function and SendCopy function required, mode = 3

DBB0 = FFh; delete content of DB TestCopyData

Requirements

To use the TestCopy function, the user program must meet the following conditions:

- FC TestCopy function must be present on the CPU.
- DB TestCopyData must be present on the CPU and have an adequate length. To achieve this copy DB99 TestCopyData from the TD7 library to your CPU. If necessary change the length of the buffer area by increasing or decreasing the area TestCopyBuffer in the declaration header of the DB, that is provided as an array with a length of [0..240] WORD.
- In the communication DB (of the type DB XComData, DB PComData or DB BComData), whose send or receive messages you want to copy, the following entries must be made (it is best to use the ready-made variable table VAT_TestCopy from the TD7 library):
  - The number of the DB TestCopyData must be entered in DW32 (symbolic name TestCopyDBNo).
  - The number of the FC TestCopy must be entered in DW34 (symbolic name TestCopyFCNo).

Linking FC TestCopy into the user program

The test function is processed cyclically by FB-XCom, FB-PCom or by FB-BCom, if the number of the FC TestCopy was entered in DBW34 in the data block to be monitored DB XComData, DB PComData or DB BComData.

Mode and filter settings of FC TestCopy

The operator works with Fc TestCopy using a VAT (variable table comparable with Control Variable in the S5 world). This VAT id ready made and present in the TD7 library as VAT_TestCopy.

The following settings are possible using the VAT_TestCopyData in the DB TestCopy:

Name	Perm. values	Meaning
OperationMode	00 h	Function blocked.
	11 h	Message entry always as of the start of the DB TestCopyData.
	22 h	Write to DB TestCopyData endlessly as circulating buffer.
	33 h	Fill DB TestCopyData once, then set mode 0.
	FF h	Delete the entire DB TestCopyData and preset new.
????_TgrmType	FF h	Copy all message types (TA = 0, 1, 2, 3) to DB TestCopyData.
	00 h	Copy only ORG messages (TA = 0).
	11 h	Copy only queried ORG messages (TA = 1).
	22 h	Copy only ORG data messages (TA = 2).
	33 h	Copy only queried data messages (TA = 3).
	01 h	Copy messages with TA = 0 or TA = 1.
	23 h	Copy messages with TA = 2 or TA = 3.
	Any combin.	Copy any 0, 1, 2, 3 combinations.
????_DestSubscr	All perm. subs	Filter for the dest. subscriber no. in the message.

Name	Perm. values	Meaning
	-1	Copy all messages regardless of the dest. subscriber no.
????_DestObject	all perm. obj	Filter for the dest. object no. in the message.
	-1	Copy all messages regardless of the dest. object no.
????_SourceSubscr	all perm. subs	Filter for the source subscriber no. in the message.
	-1	Copy all messages regardless of the source subscriber no.
????_SourceObject	all perm. obj	Filter for the source object no. in the message.
	-1	Copy all messages regardless of the source object no.
????_StartIndex	all perm. ind.	Filter for the start index no. in the message.
	-1	Copy all messages regardless of the start index no.

As the output value FC TestCopy enters a count value in DBW12 of the DB TestCopyData for the receive messages received since setting mode 1, 2 or 3 that match the filter criteria and in DBW26 it enters the number send messages.

In DBB28 the user receives a return value that indicates the errors that occurred when processing the FC. Up to now the following are defined.

RetVal = 0: no error

RetVal = 1: the specified DB TestCopyData is too short;

RetVal = 10d: the mode entered in DBB0 is not defined;

Notes on operator input

When changing from one mode to the next, the content of DB TestCopyData is not deleted, only internal pointers and message 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 message buffer area with 0. The makes the copied message blocks easier to read.

If send and receive messages 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 is used:

Bit	.7	.6	.5	.4	.3	.2	.1	.0
	Mode for SendCopy				Mode for RecvCopy			

For mode 0, 1, 2 and 3:
If the buffer is to be deleted, FFh must always be entered; separate deletion of the receive and send messages is not possible.

5.7.2 DB TestCopyData

Structure of DB TestCopyData

The following table shows the structure of the DB TestCopyData:

Address		Name	Format	Explanation
DBB	0	OperationMode	BYTE	Mode
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.
DBW	12	Recv_TgramCounter	INT	Number of copied received messages
DBB	14	SpareDBB14	BYTE	Reserve
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.
DBW	22	Send_SourceObject	INT	Send filter: Source object no.
DBW	24	Send_StartIndex	INT	Send filter: Start index no.
DBW	26	Send_TgramCounter	INT	Number of copied sent messages
DBB	28	RetVal	BYTE	Error information: 0 = No error 1 = DB TestCopyData too short 10 = Unknown operating mode
DBB	29	SpareDBB29	BYTE	Reserve
DBB	30	SpareDBB30	BYTE	Reserve
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
DBB	40	TestCopyBuffer[0]	BYTE	Copy area, byte 0
DBB	41	TestCopyBuffer[1]	BYTE	Copy area, byte 1

Address		Name	Format	Explanation
DBB	42	TestCopyBuffer[2]	BYTE	Copy area, byte 2
DBB	43	TestCopyBuffer[3]	BYTE	Copy area, byte 3
:	:	:		:
DBB	xxxx	TestCopyBuffer[xxxx]	BYTE	Copy area, byte xxxx

The individual areas of DB TestCopyData

The DB for the TestCopy function is divided into several different areas:

- Area 1: DBB 0 ... DBB28
  User interface for setting the TestCopy operating mode and displaying any errors.
  This user interface in turn is divided into the following areas:
  - Area 1.1: DBB 1 ... DBB 13:
    Filter settings for RecvCopy function and number of counted received messages.
  - Area 1.2: DBB 15 ... DBB 27:
    Filter settings for SendCopy function and number of counted sent messages.
- Area 2: DBB 29 ... DBB 39:
  Internal management pointers
- Area 3: DBB 40 ... DBB xxxx:
  Buffer area for storing messages that match the filter criteria; the buffer area must be configured as an array [0...xxxx] of WORD.

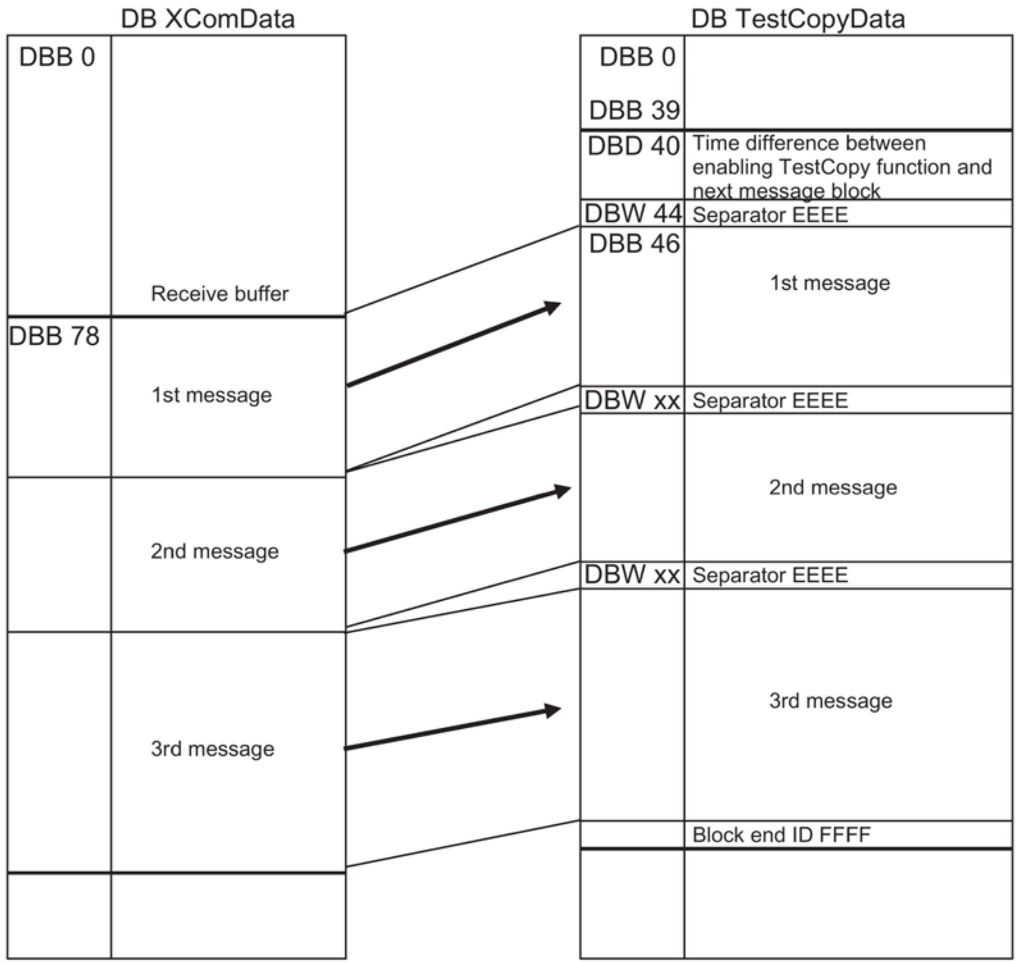
Structure of a copied message block

A message block can contain several messages. The messages are stored in 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 message block.
4. Separation sign AAAA, or EEEE:
5. Storage of the last message from the message block.
6. Block end ID FFFF.

Example

All received messages will be stored in DB TestCopyData. Communication is implemented using X blocks, i.e. a max. of 76 bytes per receive block. The receive buffer of the XComData DB is the source for FC TestCopy. The current receive block contains 3 messages.



Length calculation

FC TestCopy uses the following parameters for determining the minimum length for the DB TestCopyData:

Length of communication buffer	= LenComBuffer	= 76 or 202 bytes
Minimum message length	= LenMinTgrm	= 14 bytes
Offset management area	= Offset	= 40 bytes
Length of the time difference	= LenDt	= 4 bytes
Length of the block separators	= LenSeparator	= 2 bytes

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:

a) For X communication:

Len_{Min_Xcom} $=Len_{ComBuffer} + Offset + Len_{dt} + (Len_{ComBuffer} / Len_{MinTgrm} + 1) * Len_{Separator}$

$= 76 + 40 + 4 + (76 / 14 + 1) * 2$

$= 120 + 12 = 132$ bytes minimum

a) For B communication:

Len_{Min_Bcom} $=Len_{ComBuffer} + Offset + Len_{dt} + (Len_{ComBuffer} / Len_{MinTgrm} + 1) * Len_{Separator}$

$= 202 + 40 + 4 + (202 / 14 + 1) * 2$

$= 236 + 40 = 276$ bytes minimum

If FC TestCopy determines that the DB TestCopyData is not long enough, an error message to this effect is entered in data byte DBB28.

5.8 SFC / SFB system blocks used

Introduction

In the operating system there are system functions SFCs and system function blocks SFBs that are used by the TD7 blocks as auxiliary blocks. Since the TD7 package should be capable of running on all CPU types, only system blocks are used that are available on all CPU types. Here the 300 CPUs mainly decide which system blocks are used. There are only deviations from this rule in exceptional cases (currently only the FC ListGenerator exists in a separate version for S7-300 and S7-400 and FB BCom only exists for the S7-400 because PBC connections are only possible there).

Because the system blocks are part of the operating system they do not occupy any user memory.

Below there is an overview of the system blocks used that exist on all CPU types.

SFC 0 SET_CLK

Block for setting the date and time on the CPU.

SFC 1 READ_CLK

Block for reading the date and time on the CPU.

SFC 20 BLKMOV

Block for copying interrelated data areas.

SFC 22 CREATE_DB

Block for online creation of a data block.

SFC 46 STP

Block for changing the CPU to 'Stop'.

SFC 52 WR_USRMSG

Block for entering a user message in the diagnostics buffer.

SFC 64 TIME_TCK

Block for reading the system time of the CPU.

SFC 65 X_SEND

Block for sending data via an unconfigured connection.

SFC 66 X_RCV

Block for receiving data via an unconfigured connection.

Other system blocks only available in 400 CPUs:

SFC 23 DEL_DB

Block for online deletion of a data block.

SFC 24 TEST_DB

Block with which, among other things, it is possible to determine whether a certain data block exists on the CPU.

SFC 25 COMPRESS

Block for online compression of the user memory, e.g. after deleting data blocks.

SFB 12 BSEND

Block for block-oriented sending of data via a configured connection.

SFB 13 BRCV

Block for block-oriented reception of data via a configured connection.

SINAUT Diagnostics and Service tool 6

6.1 Overview of the functions and operation of the SINAUT Diagnostics and Service tool

Introduction

The SINAUT Diagnostics and Service tool provides the user with functions for checking connections, interfaces and communication as well as the firmware and software components of the network subscribers of a SINAUT installation.
The most important functions are as follows:

- Reading the diagnostic data from a TIM or CPU module
- Reading the diagnostic buffer
- Checking and setting the module time
- Reading the module parameter assignment
- Activating a message trace
- Firmware update of TIM modules
- Downloading a new parameter assignment to the TIM modules

Note

Diagnostics functions that are also available in the SIMATIC Manager are described here with the emphasis on diagnostics of TIM modules.

6.1.1 Starting the program and types of access

Opening the SINAUT diagnostics and service tool

The SINAUT diagnostics and service tool is opened in the Windows start menu *SIMATIC / SINAUT ST7 / Diagnostics and Service*.

You can access the module-specific diagnostic information alternatively over:

- *Accessible nodes*
- the *SINAUT subscriber list* of the STEP 7 project

Note

Regardless of whether you access this information using *Accessible nodes* or the *SINAUT subscriber list*, unless you activate the PG routing function, you can only access subscribers of the subnet of the local MPI bus to which the PG is connected.

Access using *Accessible Nodes*

To access the diagnostic data using *Accessible Nodes*, follow the steps outlined below:

1. Click on the *Accessible Nodes* button
   or click on the *Project / Accessible Nodes* menu.
   The *Accessible Nodes* dialog opens.
2. In the *Selectable Nodes* dialog, select the required subscriber from the list of MPI addresses by clicking on it with the mouse.

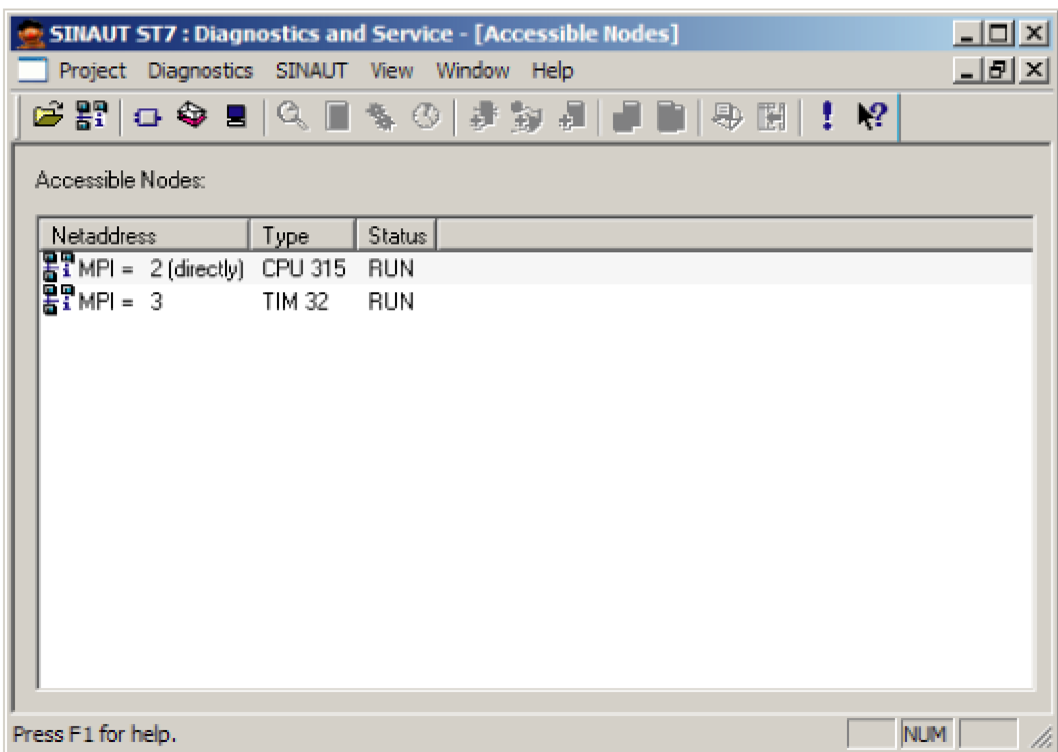


Figure 6-1 The *Accessible Nodes* window of a sample system

Access using the *SINAUT subscriber list* of a STEP 7 project

To open the STEP 7 project in the diagnostics and service tool and to access the diagnostic data using the *SINAUT subscriber list*, follow the steps outlined below:

1. Click on the *Open Project* button in the toolbar or select the *Project / Open* menu. The *Open* dialog is displayed.
2. Select the STEP 7 project in the *User Projects* tab of the *Open* dialog and click on the *OK* button. The project window with the *SINAUT subscriber list* of the relevant project opens.
3. If the required project is not displayed in the *Open* dialog, click the *Browse* button. In the Browse dialog that opens, you can search for other projects and include them in the project list.
   As an alternative, you can open a current project with the *Project / Recently Used* menu.
4. Select the subscriber you require for the subsequent diagnostic functions in the *SINAUT subscriber list* by clicking on it with the mouse.

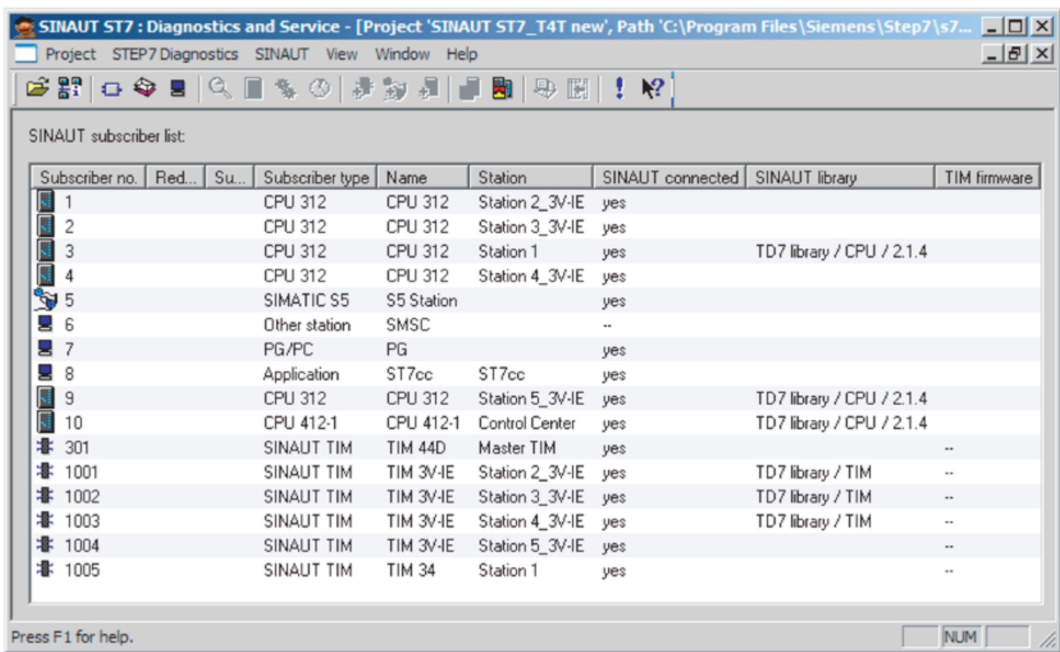


Figure 6-2 The *SINAUT subscriber list* of a sample project

Note

Attempting to access a remote subscriber using the *SINAUT subscriber list* of a STEP 7 project can lead to "misunderstandings" if the subscriber is not connected to the local MPI bus and the PG routing function is not activated. With functions involving access to the module, the remote subscriber is displayed in the *Path* field of the diagnostics dialog, however the diagnostic data is that of the locally connected subscriber.

The SINAUT subscriber list displays the following entries for each subscriber:

- **Subscriber no.**: The subscriber number of the SINAUT subscriber that is unique throughout the project
- **Red. Subscriber no.**: The *Redundant subscriber number* parameter is used only when there is a redundant partner for the subscriber in question. The number specifies the common subscriber number under which the redundant system can be addressed by other subscribers.
- **Subscriber no. of red. Partner**: The *Subscriber number of the redundant partner* parameter is used only when there is a redundant partner for this subscriber. The parameter specifies which of the subscribers belong to a redundant relationship.
- **Subscriber type**: The *subscriber type* specifies the class of subscriber involved. The subscriber type cannot be set by the user.
- **Name**: The module, application or PC/PG name. This can be changed in the configuration. As default, this is the name of the module type or the application as specified in the configuration.
- **Station**: Name of the station specified by the user in the configuration using NetPro.
- **SINAUT connected**: Specifies whether a SINAUT connection was configured for the subscriber.
- **TD7 library version**: With CPU modules and modules of the type TIM 3V-IE, the name of the SINAUT system library for the TD7 software blocks is displayed.
- **TIM firmware version**: With TIM modules, the version of the TIM firmware is displayed.

PG Routing

If you connect a programming device (PG) or a PC to access the diagnostic data, you only have access to the local MPI network. The diagnostic data of remote subscribers in other network sections is not accessible.

To access subscribers in other network sections, you can use PG Routing. If you access data in a subordinate subnet after activating PG Routing, remember that you can only access subordinate subnets and not subnets higher in the network hierarchy. The requirements, functions and activation of PG routing are described in a separate section.

The PG Routing function is possible only when using the *SINAUT subscriber list*; PG Routing is not possible when using *Accessible Nodes*.

6.1.2 Access to SINAUT subscribers and working with the diagnostics dialogs

Activating diagnostic functions

The diagnostic functions are activated as follows:

1. Select a SINAUT subscriber by clicking on it with the mouse in *Accessible Nodes* or in the *SINAUT subscriber list* of a STEP 7 project.
2. Start the required diagnostic functions with one of the following alternatives:
   – Clicking on the corresponding button in the toolbar
   – Selecting the function in the *Project*, *STEP 7 Diagnostics* or *SINAUT* menus
   – Pressing the relevant function key
   – Right-clicking on the subscriber in *Access of Nodes* or in the *SINAUT subscriber list*. After clicking on the subscriber, select the required function with the right mouse button in the displayed context menu.
3. The dialog belonging to the selected diagnostic function is displayed.

Working with the dialogs

The graphic user interface of the SINAUT Diagnostics and Service tool is designed based on Windows technology. To use diagnostic functions, you must generally first select a particular subscriber or a component from a list in the Windows and dialogs and the function will then be executed and the diagnostic data displayed for this subscriber or component. The function is then activated from a menu or by selecting a button and a dialog for the specific diagnostic function then opens.

When *selecting* a menu, a subscriber, or object is described, this involves clicking on the object once within the left mouse button.

Buttons found in many of the diagnostics dialogs are explained here and not in each subsection. These include the buttons:

- *Print:*
  Starts a printout of the currently open dialog.
- *Update:*
  Updates the content of the dialog with the current diagnostic data of the selected subscriber.
- *Save:*
  Saves the content of the open dialog in a file. You can select any directory and file name in the *Save* dialog.

- *Load:*
  Loads the diagnostic data relevant to the current dialog content from a previously saved file into the open dialog. The loaded diagnostic data is displayed in the dialog.

Note

When *loading* data from a file, the current project data in the dialog is overwritten by the data from the file. To display the data of the connected subscriber again, the dialog must be closed and reopened, in some cases, the display can be updated with the data of the connected subscriber again using the *Update* button.

- *Close:*
  Closes the current dialog. You return to the *Accessible Nodes* or *SINAUT subscriber list*.
- *Help:*
  Opens the online help function for the currently selected diagnostic function.
- *OK:*
  Confirms the entries made and closes the dialog.
- *Cancel:*
  Discards the entries made and closes the dialog.

6.1.3 Functions of the Diagnostics and Service tool

Overview of the diagnostic and service functions

The diagnostic functions of the SINAUT Diagnostics and Service tool can be grouped together as follows:

- STEP 7 diagnostics
- SINAUT diagnostics (TIM status information and TD7 software diagnostics)
- Message protocol diagnostics
- Service functions

The following table shows the diagnostic and service functions and all the menus in which the functions of the SINAUT Diagnostics and Service tool can be called.

The two right-hand columns in the table indicate that the scope of information when using the *SINAUT subscriber list* of a STEP 7 project is greater than when using *accessible nodes*.

Table 6- 1 Overview of the diagnostic and service functions of the SINAUT Diagnostics and Service tool

Function group, diagnostic function (remarks)	**Subscriber type relevant for diagnostics**	**Called in menu**	**Access over STEP 7 project**	**Access using Accessible Nodes**
STEP 7 diagnostics		**STEP 7 diagnostics**		
CPU messages	CPU, TIM	"	X	X

Function group, diagnostic function (remarks)	**Subscriber type relevant for diagnostics**	**Called in menu**	**Access over STEP 7 project**	**Access using Accessible Nodes**
Module information (including messages in diagnostic buffer)	CPU, TIM	"	X	X
Operating mode	CPU, TIM	"	X	X
Setting the time	CPU, TIM	"	X	X
SINAUT				
SINAUT diagnostics		**SINAUT**		
TIM Diagnostics	TIM	"	X	X
TIM subscriber diagnostics	TIM	"	X	X
TIM diagnostic messages	TIM	"	X	
TIM Message Monitor	TIM	"	X	
TD7 software diagnostics		**SINAUT**		
TD7 CPU diagnostics (TD7 messages in diagnostics buffer)	CPU	"	X	X
TD7 block structure (configured data)	CPU	"	X	
TD7 block structure for all CPUs (configured data)	CPU	"	X	
TD7 CPU program comparison (configured data)	CPU	"	X	
TD7 communication configuration check (configured data)	CPU	"	X	
TD7onTIM diagnostics	TIM	"	X	
SDB Viewer	CPU, TIM	"	X	X
Service functions		**SINAUT**		
Download SDB	TIM	"	X	
Firmware update	TIM	"	X	
Repair	TIM	"	X	
Message protocol diagnostics		**Project**		
Testcopy DB	CPU	"	X	X
TIM message protocol	TIM	"	X	X
ST7cc / ST7sc message protocol	TIM	"	X	X

The diagnostic information is displayed only for SIMATIC CPU modules and SINAUT TIM modules.

6.2 STEP 7 diagnostics

Introduction

STEP 7 diagnostics covers the standard diagnostics functions of SIMATIC STEP 7 V5.x. In addition to the information relating to pure SIMATIC systems, STEP 7 diagnostics in the SINAUT diagnostics and service tool not only provides information about the CPU modules but also about the TIM modules of the project. The functions in the SINAUT Diagnostics and Service tool that are implemented on the TIM module:

- CPU messages
- Module information
- Operating mode
- Setting the time

6.2.1 CPU messages

Description of the functions

The *CPU messages* function is used to archive diagnostic messages entered by a CPU or TIM module in its diagnostic buffer. Without archiving, messages in the ring buffer of the CPU or TIM would be successively overwritten once the buffer is full.

The *CPU messages* function registers the PG used for diagnostics with one or more modules. The modules then transfer all newly generated diagnostic messages to the registered PG. The diagnostic messages of one or more modules are archive in a common list on the PG. The archive is designed as a ring buffer. The oldest messages are overwritten by newly arriving messages once the archive is full.

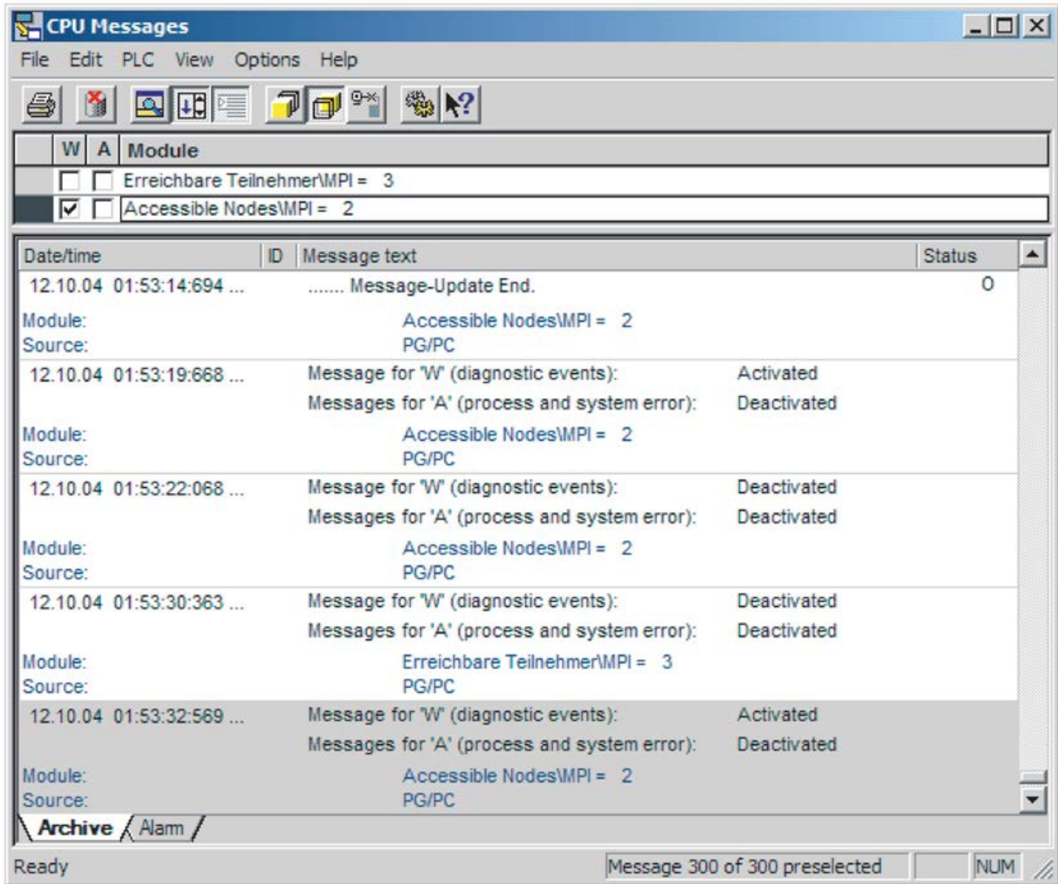


Figure 6-3 *CPU Messages* dialog

The messages for diagnostic events are entered at the bottom of the dialog in the *Archive* tab of the message list.

From the menu of the dialog or using the buttons of the toolbar, various user-specific settings can be made for message output such as emptying the archive, processing messages, the view of the message window, the settings for the archive size and saving the PG connections to the registered modules for the next time the *CP messages* function is called.

Operator activities

1. Select a subscriber by clicking on it in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the *CPU Messages* dialog by selecting the *STEP 7 Diagnostics / CPU Messages* menu.
3. To register the PG/PC for the *CPU Messages* function, select the module in the *W column* of the module list at the top of the dialog. After the registration, the connection option (check box) of the module is selected in the *W column*. All the generated diagnostic messages of the module are then displayed in chronological order in the *Archive* tab of the message list at the bottom of the dialog.
   If no connection can be established to the subscriber, a symbol is displayed in the first column of the module list indicating that the connection is interrupted.

4. Click on the relevant field for the module in the *W column* of the module list again to deactivate archiving of the diagnostic messages.
5. Select the menu or the button of the dialog to change the settings.
6. Close the *CPU Messages* dialog by clicking on the *close dialog* button (x) in the title bar or double-clicking on the dialog name in the title bar of the dialog.

Closing the dialog deactivates the *CPU Messages* function.

6.2.2 Module information

Description of the functions

The *module information* function reads diagnostic data from the module of the connected station. The diagnostic data is displayed for the specific module in a series of tabs:

- *General* tab
  List of hardware and firmware components with their versions and information on the status of the CPU module
- *Diagnostic Buffer* tab
  List of diagnostic messages
- *Memory* tab
  Information on the utilization of the load and work memory.
- *Time System* tab
  Information on the data, time, time system and time synchronization as well as on the operating hours counter of CPU modules
- *Performance Data* tab
  Lists of the organization blocks, system blocks and address ranges
- *Communication* tab
  Information on transmission speeds, connection resources and cycle load caused by a communication

Further tabs are displayed for CPU modules:

- *Cycle Time* tab
  Set and measured cycle times of CPU modules
- *Stacks* tab
  Information on the content of the blocks stack (B stack), interrupt stack (I stack) and local data stack (L stack) of CPU modules

Operator activities

1. Select a local subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *STEP 7 Diagnostics / Module Information* menu.
3. Select the individual tabs with the mouse.

General tab

In the *General* tab, the operating status of the local CPU module and the operating status of the connected module are displayed if this output of the diagnostics data was selected. In the *Status* text box, information about the status of the connected module is displayed from the perspective of the local CPU module. The following possibilities for the status are distinguished:

- Status *OK*: Module exists, access possible
- Status *Error*: Disruption, access to module not possible (parameter assignment or access error)

Information about the module designation and system ID is also displayed and in the *Version* output box a list of the hardware and firmware components of the module with order number or designation and version is displayed.

Below this follows information about the rack, address and slot of the CPU module.

Diagnostic Buffer tab

The *Diagnostic Buffer* tab displays the content of the diagnostic buffer of the module with information on the message number, time of day, date and event. The entries are sorted in descending chronological order; in other words, the latest message is at the top.

For the TIM, the last 50 entries of the diagnostic buffer are displayed, for a CPU normally the last 10 diagnostic messages.

For TIMs all diagnostic messages are displayed in plain language.

For CPUs, the system diagnostic messages are displayed as plain language and the TD7 diagnostic messages (in other words the messages created by the SINAUT user program) are displayed in hexadecimal format.

The station number (STA no.) listed with some messages in the *Details on Event* text box is the WAN network address of the relevant SINAUT network.

Note

If you have selected a CPU and want to see the plain text equivalent of TD7 diagnostic messages displayed in hexadecimal format in the *Diagnostic Buffer* tab, select the *TD7 CPU Diagnostics* function for the same CPU.

You will then see the same TD7 diagnostic messages

- in the *Module Information / Diagnostic Buffer* dialog in hexadecimal format and
- in the *TD7 CPU Diagnostics* dialog as plain text.

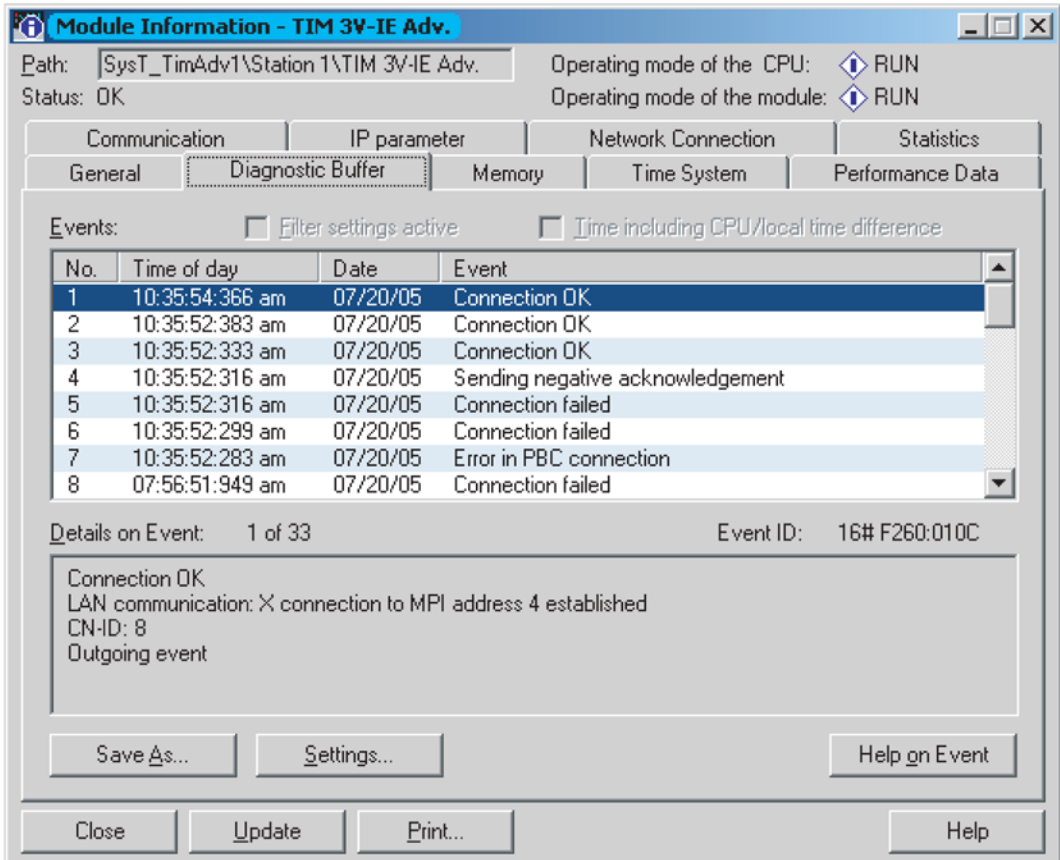


Figure 6-4 *Module Information*, *Diagnostic Buffer* tab

To change the settings and select the event types of the message display in the *Diagnostic Buffer* tab, follow the steps outlined below:

1. Select the *Settings* button to open the *Setting for Display Diagnostic Buffer* dialog. The default number of entries can be changed neither for CPUs or TIMs.
2. In the *Display Events* box, select or deselect the event types for message output. The selection is displayed or hidden.
3. In the lower part of the dialog, select the following options if necessary:
   - *Output event information in hexadecimal format*
   - *Update display during operating mode transition*
   - *Save settings for this dialog box*
4. Confirm your settings by clicking on the *OK* button or to discard the settings, click on *Cancel*. You then return to the *Diagnostic Buffer* tab.

Memory tab

The *Memory* tab displays information on the utilization of the free and assigned load memory and work memory.
The work memory utilization of a TIM of approximately 90% is normal and adequate for the TIM to function.

Time System tab

The *Time System* tab provides information on the time system of the module in three boxes:

- The current state and time of the module, its resolution and the existence of a real-time clock
- Time-of-day synchronization (CPU only)
- Run-time meter (CPU only)

Performance Data tab

The *Performance Data* tab does not contain any diagnostic information relevant to TIM modules. For CPU modules, information is displayed on organization blocks (OB), system blocks (SFC, SFB) and address ranges.

Communication tab

The *Communication* tab displays the following information:

- Maximum and unused connection resources for
  - PG communication
  - OP communication
  - S7 basic communication
- Configured cycle load due to communication. For a TIM, this is 100%.

Information on a communication relates only to the CPU.

IP Parameter tab

The *IP Parameters* tab displays the most important IP parameters of an Ethernet TIM:

- IP address: Configured IP address of the module.
- Subnet mask: Configured subnet mask of the module.

- Default router: If a default router was specified during configuration, the IP address of the default router is displayed here.
- IP settings: Indicates where the module obtained the IP parameters from.

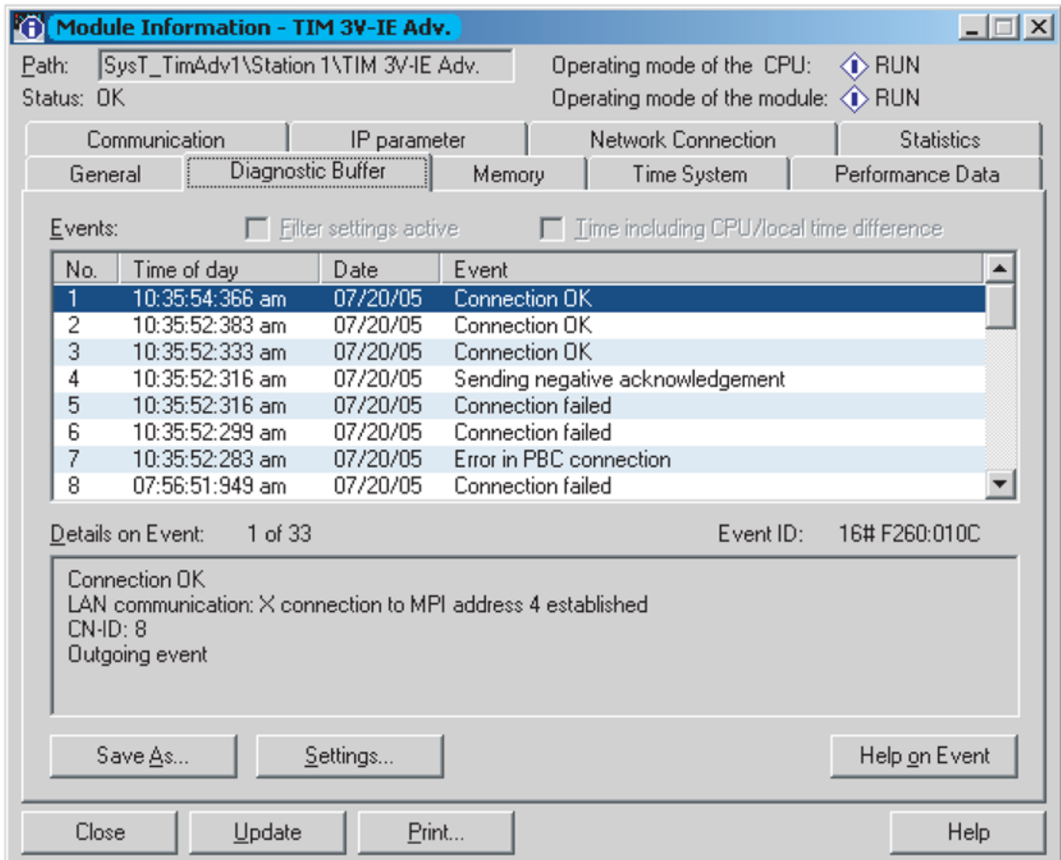


Figure 6-5 *Module Information* dialog, *IP Parameter* tab

Note

With a TIM 4R-IE, only information on the first Ethernet port P1 is displayed. For an overview of the status and parameters of both Ethernet ports of the module, refer to SINAUT Diagnostics, *IP Parameters* tab.

Network Connection tab

The *Network Connection* tab for an Ethernet TIM displays the MAC address of the module and information on the status and settings of the Ethernet port:

- Link Status: Indicates whether or not a physical connection to Ethernet exists.
- Settings: Shows the setting for detecting network settings, here: "Automatic" (Autosensing)
- Mode: Indicates the transmission speed and duplexity on Ethernet.

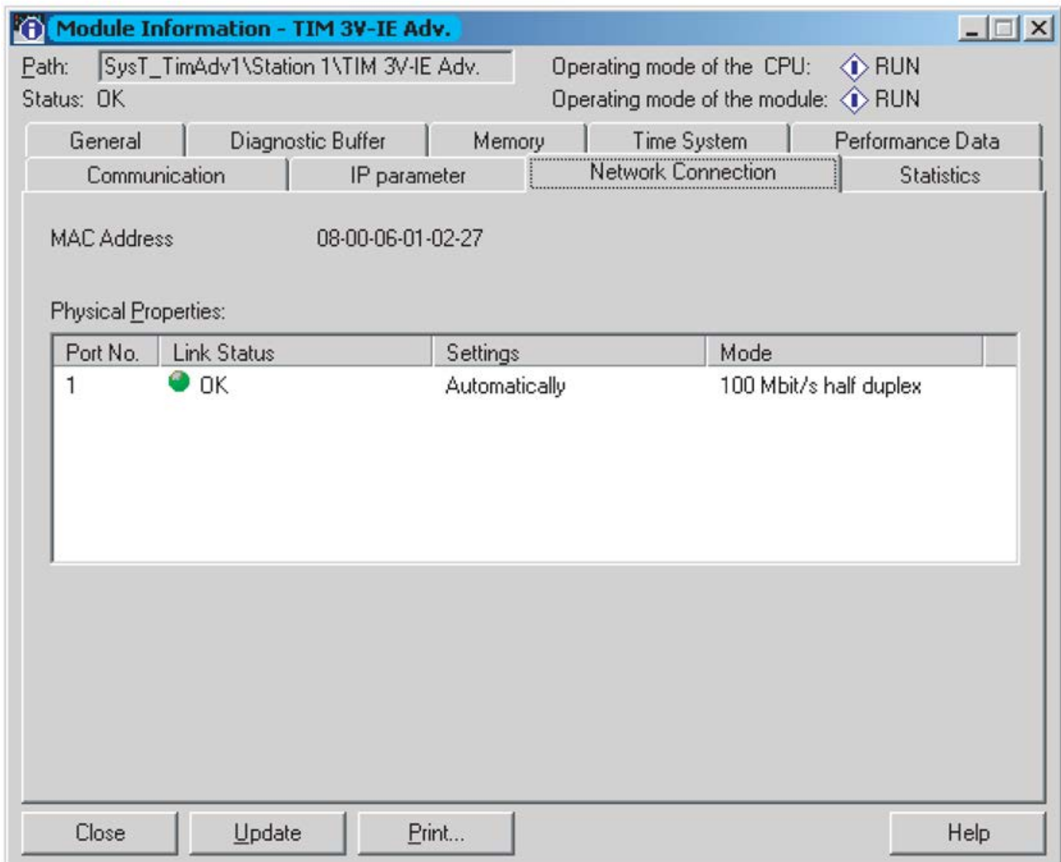


Figure 6-6 *Module Information* dialog, *Network Connection* tab

---

Note

With a TIM 4R-IE, only information on the first Ethernet port P1 is displayed. For an overview of the status and parameters of both Ethernet ports of the module, refer to SINAUT Diagnostics, *IP Parameters* tab.

Statistics tab

This tab is available only for Ethernet TIMs.

The *Statistics* tab contains transmission statistics for the Ethernet ports. The number of transferred data packets with and without errors since the last reset or restart of the module is displayed for the send and receive directions. This time time is displayed as module time in the tab. The statistical values can be reset to zero with the *Reset* button.

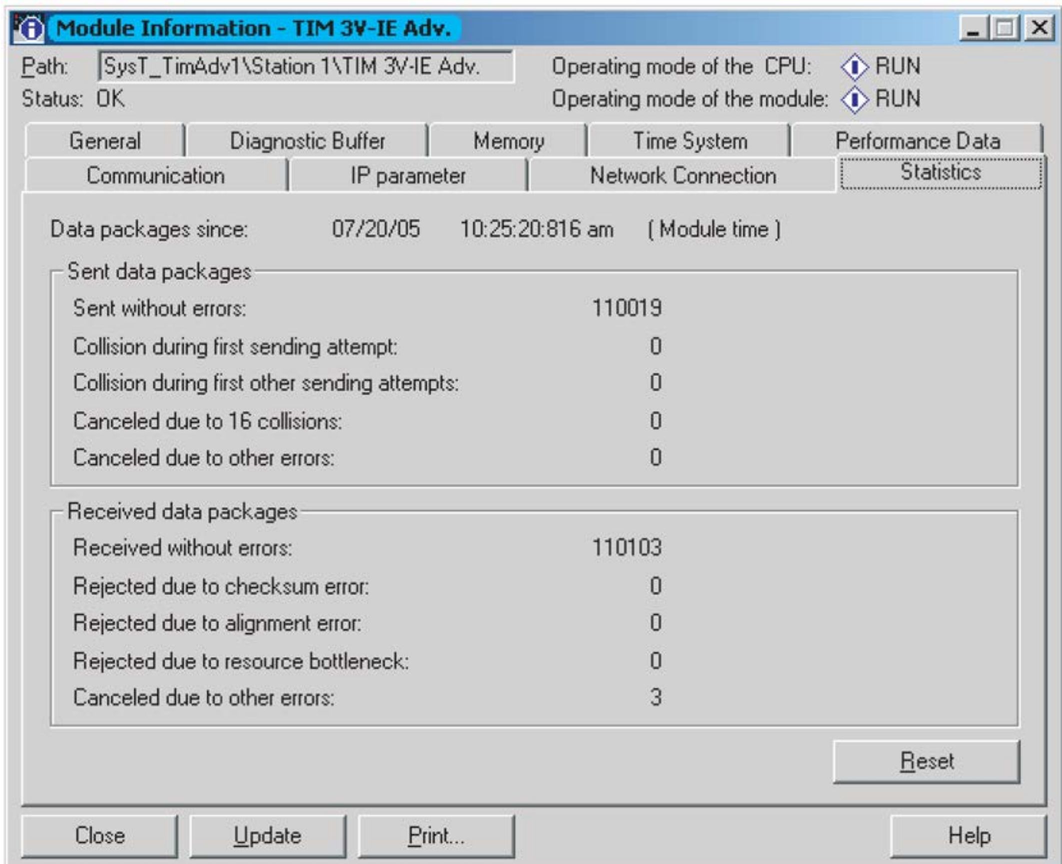


Figure 6-7 *Module Information* dialog, *Statistics* tab

6.2.3 Operating mode

Description of the functions

With the *Operating mode* function, you can change the operating mode of TIM and CPU modules. Apart from the operating mode, the current keyswitch setting and the last operating mode are displayed for CPU modules.

With TIM and CPU modules, the operating mode can be changed from *Run* to *Stop* or from *Stop* to *Run*. Changing the operating mode from *Stop* to *Run* triggers a restart on the TIM module.

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *STEP 7 Diagnostics / Operating Mode* menu.
3. Click on the *Stop* button to stop the module.
4. Click on the *Warm Restart* button to restart the module.
   A TIM goes through a warm restart after approximately 10 seconds.

6.2.4 Setting the time

Description of the functions

The *Set Time of Day* function is used to display and set the date and time of a module. It is possible to set the module time to the PG/PC time or to set an edited time.

CPU modules have a hardware clock.
TIM modules have a software clock in the operating system of the module.

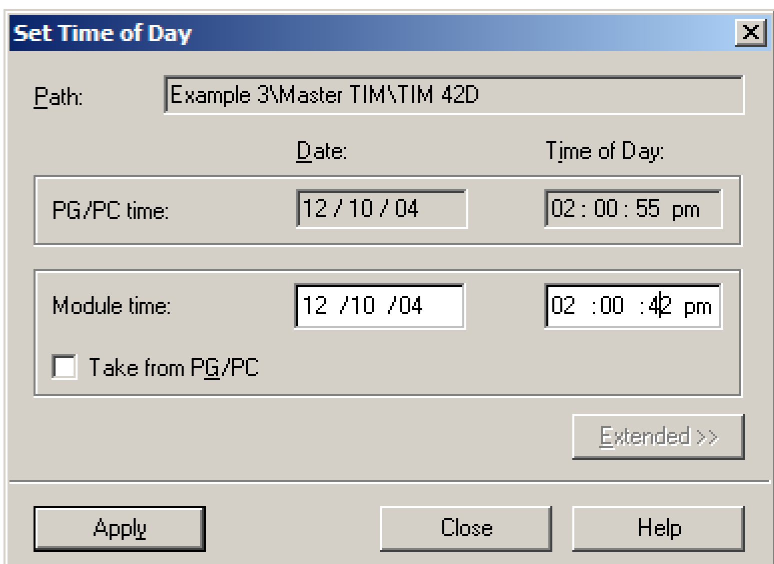


Figure 6-8 *Set Time of Day* dialog

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *STEP 7 Diagnostics / Set Time of Day* menu.
3. To set the module time manually, click in the date or time display with the mouse, change the data and/or time using the keyboard and confirm by clicking the *Apply* button

or or select the *Apply from PG/PC* option in the *Module time* field and confirm by clicking the *Apply* button.

6.3 SINAUT diagnostics

6.3.1 TIM Diagnostics

Description of the functions

The *TIM Diagnostics* function provides various diagnostic data of a TIM module. This is displayed in the following tabs:

- *Memory* tab:
  Information on memory and disk configuration
- *Message buffer* tab:
  Information on the buffer areas of an Ethernet TIM for messages
- *Communication* tab:
  Displays the installed communication drivers on the various interfaces of the TIM
- *Time synchronization* tab:
  Status of the time-of-day synchronization on the interfaces of the TIM
- *Time* tab:
  Information on the system clock of the TIM
- *Filesystem*:
  Displays all the files in the flash file system or (if installed) on the RAM disk of the TIM
- *IP Parameters* tab (TIM 4R-IE only):
  Displays the current IP parameters and settings of the Ethernet ports
- *Statistics* tab (TIM 4R-IE only):
  Displays the transmission statistics for the Ethernet ports

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *SINAUT / TIM Diagnostics* menu.
3. Select the individual tabs with the mouse.
4. To display the interface-specific diagnostic data in the *Communication* and *Time Synchronization* tabs, select the name of an interface. The information on the relevant interface is displayed in the fields in the lower part of the two tabs.
   – In the *Communication* tab: Select an interface in the *Communication drivers* list box.
   – In the *Time Synchronization* tab: Select an interface in the *Communication interfaces for time synchronization* list box

Memory tab

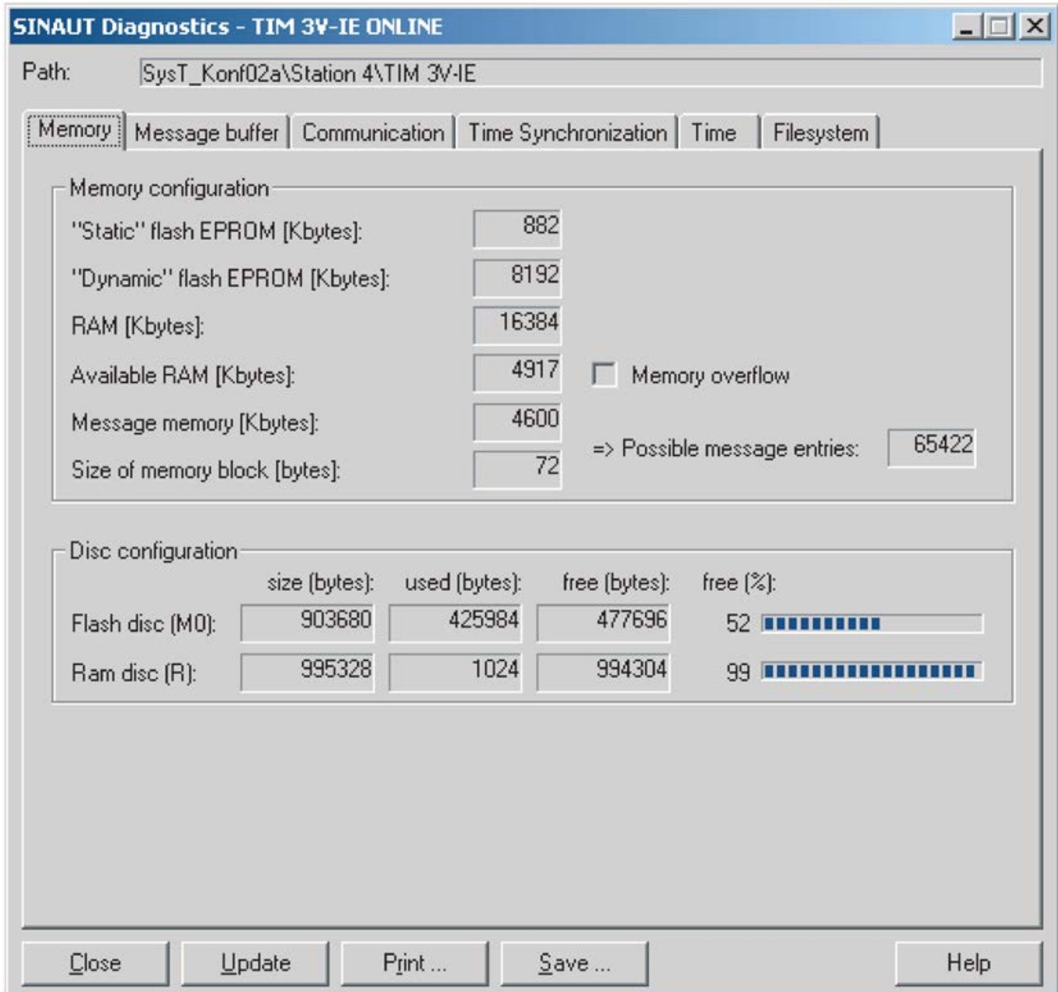


Figure 6-9 *SINAUT Diagnostics* dialog *Memory* tab

The *Memory* tab displays current diagnostic data on the memory configuration on the TIM:

- In the *Memory configuration* box:
  - Static flash EPROM
  - Dynamic flash EPROM
  - RAM
  - Available RAM: Size of the free storage space available to the drivers on the TIM for dynamic data.
  - Memory overflow: If the free RAM is no longer adequate, a checkmark appears in the *Memory overflow* check box.
  - Message memory: Size of the memory for data messages that can be stored
  - Size of a memory block that is reserved for a data message.
  - Number of possible message entries calculated based on the size of the message memory and the size of a memory block
- In the *Disc configuration* box:
  - Storage space, used and free storage capacity of the flash or RAM disk.

Message buffer tab

The *Message buffer* tab is available only for TD7onTIM-compliant TIM modules (for example TIM 3V-IE). The tab displays the current diagnostic data on the size and utilization of the buffer areas for messages on the selected TIM module.

With TD7onTIM-compliant TIM modules, the message buffer is divided into various buffer areas:

- Buffer areas for data messages sorted according to destination subscribers (destination subscriber buffers)
- Buffer areas for messages in TIM - TIM communication (TIM buffers)
- Buffer area for local communication with the CPU or an ST7cc/sc

The *Message buffer* diagnostics function analyzes the buffer areas of the destination subscribers in which the data messages are stored. These are of particular interest to the user.

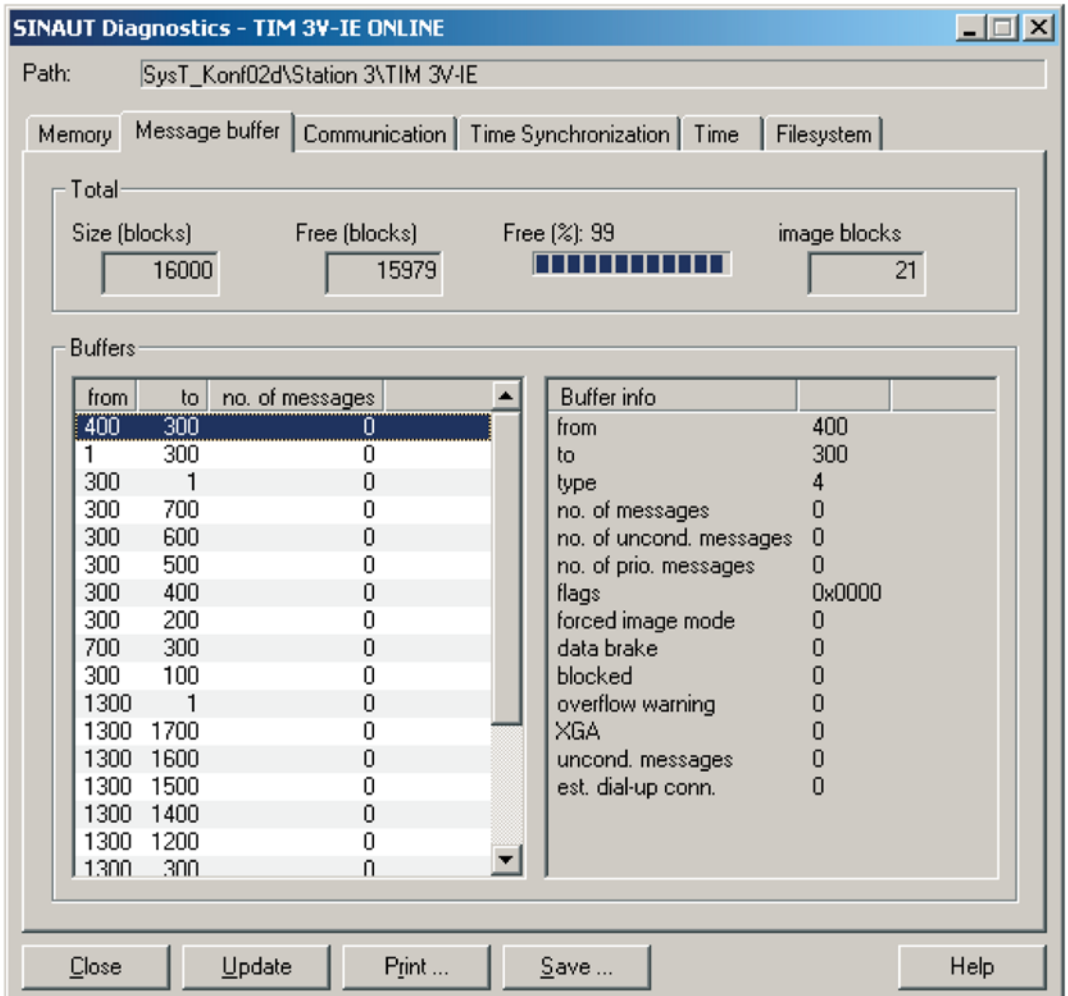


Figure 6-10 TIM Diagnostics - *Message buffer* tab

The *Total* output box provides the following information:

- *Size (blocks)*:
  Total size of the message buffer. The value indicates the total number of message memory blocks. This is calculated from the total size of the message memory set for the TIM and the byte size of a memory block. The parameter assignment is made in the network configuration in *NetPro* in the *Properties* dialog, *Options* tab, *Global message memory* field.
- *Free (blocks)*:
  Free area of the message buffer. The value indicates the number of free memory blocks.
- *Free (%)*:
  Free area of the message buffer as a percentage
- *image blocks*:
  Number of blocks occupied by the TIM in the message memory for data messages transmitted using the image memory principle.

The *Buffers* list box shows the message buffers for various communication partners with the following information:

- *from*:
  Subscriber number of the source subscriber
- *to*:
  Subscriber number of the destination subscriber
- *no. of messages*:
  Total number of stored messages for the relevant source and destination subscriber

If a message buffer is selected on the left with the mouse in the *Buffers* box, the following detailed information is displayed in the *Buffer info* list:

- *from*:
  Subscriber number of the source subscriber
- *to*:
  Subscriber number of the destination subscriber
- Type:
  - = 2: Buffer for organizational messages, hand-shake messages or messages transmitted from one TIM to another TIM.
  - = 4: Buffer for messages to a remote subscriber (CPU or ST7cc).
  - = 8: Buffer for messages to a local subscriber (CPU or ST7cc).
- *no. of messages*:
  Total number of stored messages for the source and destination subscriber named at the top
- *no. of uncond. messages*:
  Number of stored messages to be sent unconditionally and spontaneously (only relevant in dial-up networks)
- *no of prio. messages*:
  Number of stored messages to be sent with high priority.
- *Status:*
  The status is a hexadecimal value that codes the buffer information following it into binary.
- *Forced image mode*:
  1 = forced image mode
  To avoid overflow of the message memory, all data messages are treated as image messages; in other words, even send buffer messages are treated as image messages and overwritten.
- *data brake*:
  The sending of messages to the remote partner is currently disabled either because the remote partner is unavailable or there is a lack of memory on the remote partner.
- *blocked:*
  Reserved for future functions. Nothing is currently displayed.
- *overflow warning:*
  Reserved for future functions. Nothing is currently displayed.

- *XGA:*
  Reserved for future functions. Nothing is currently displayed.
- *uncond. messages:*
  Reserved for future functions. Nothing is currently displayed.
- *est. dial-up conn.*
  Reserved for future functions. Nothing is currently displayed.

Communication tab

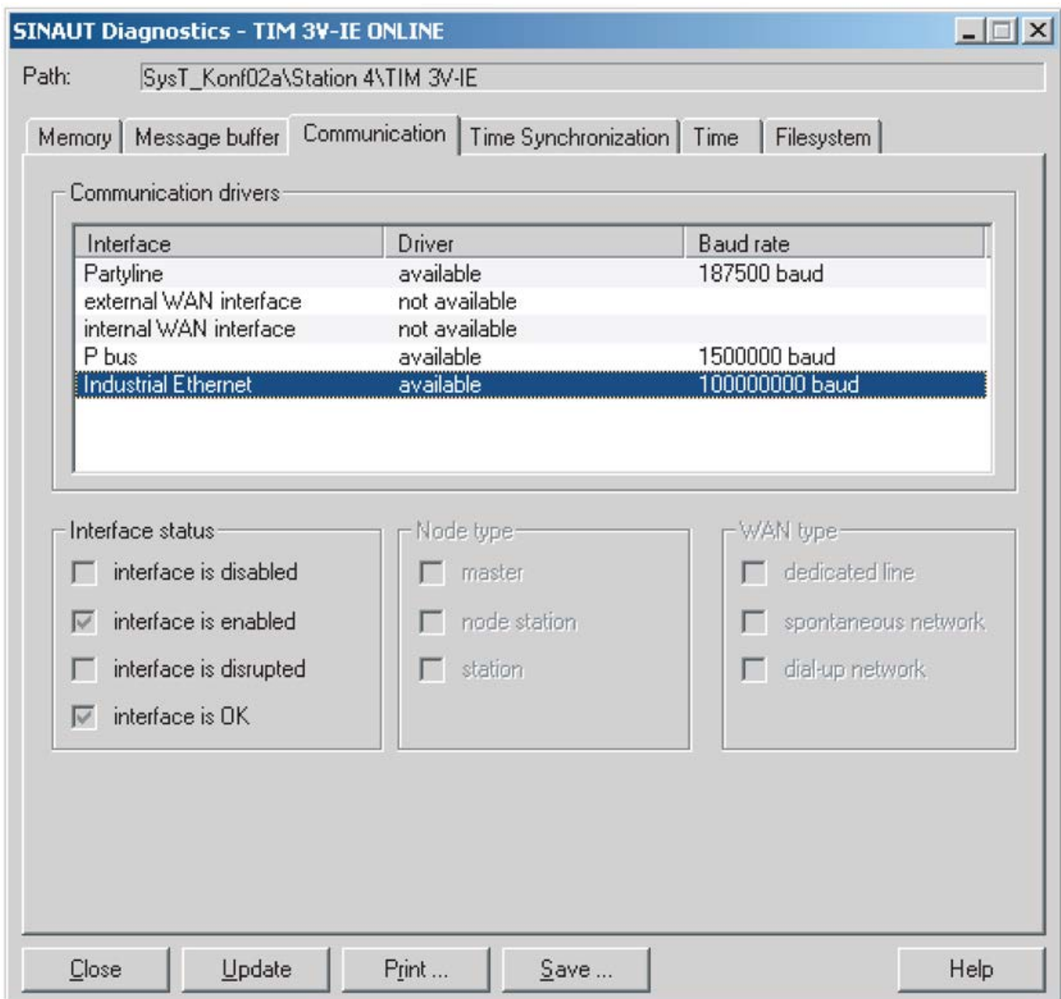


Figure 6-11 *SINAUT Diagnostics* dialog *Communication* tab

The *Communication* tab displays information on the status of communication of the TIM with information on interfaces, drivers (available/not available) and baud rate. The data is displayed in the lower part of the dialog when you select one of the communication interfaces.

Time Synchronization tab

The *Time Synchronization* tab displays information on the time synchronization on the various interfaces of the TIM with information on the interface, synchronization and status of time synchronization. The information is displayed in the lower part of the dialog when you select one of the communication interfaces.

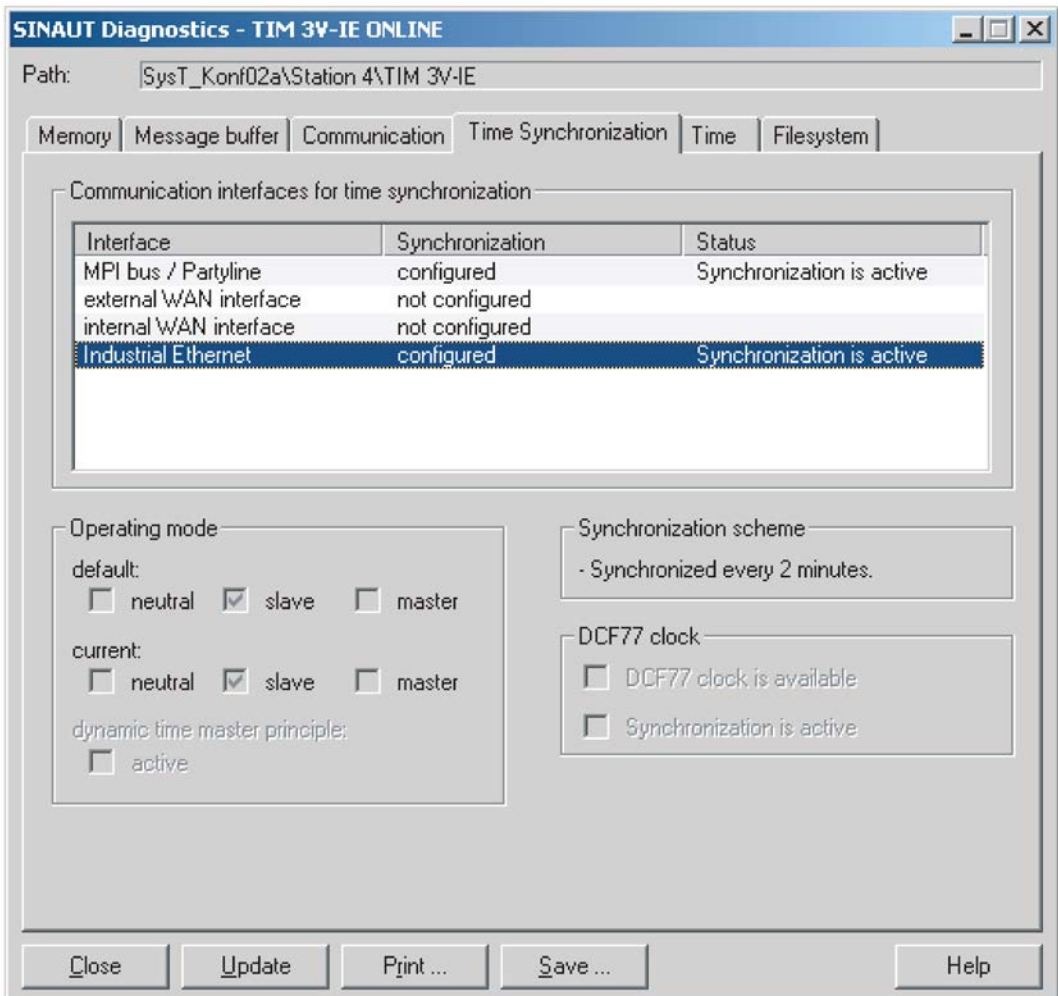


Figure 6-12 *SINAUT Diagnostics* dialog, *Time Synchronization* tab

Time tab

The *Time* tab displays the data and current module time of the TIM on the left in the *Current time* area. On the right in the *Clock status* area, information on the validity of the time, daylight saving/standard time and the changeover from daylight saving to standard time is displayed.

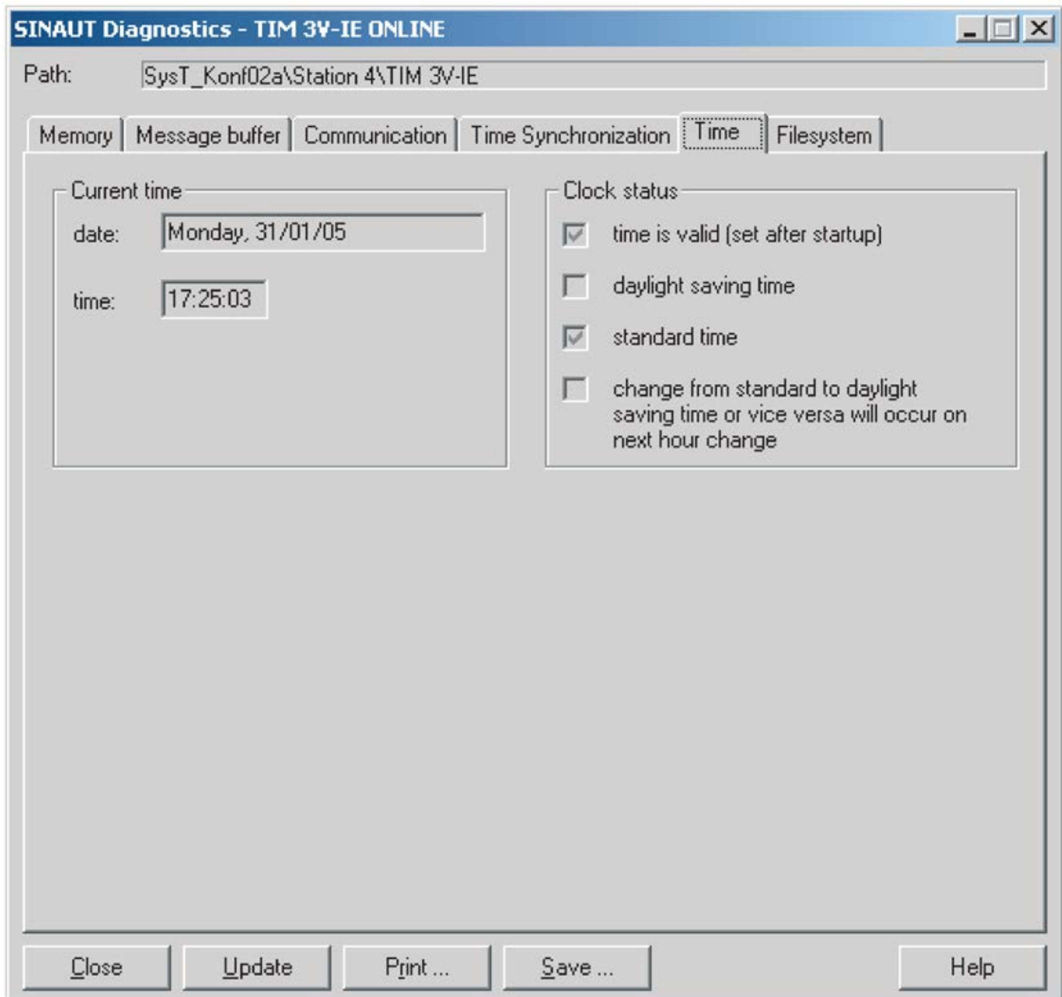


Figure 6-13 *SINAUT Diagnostics* dialog, Time tab

Filesystem tab

The *Filesystem* town displays all the system data blocks and files of the individual firmware components installed on the flash file system.

If a RAM disk is configured on the TIM, this is also displayed with the files it contains.

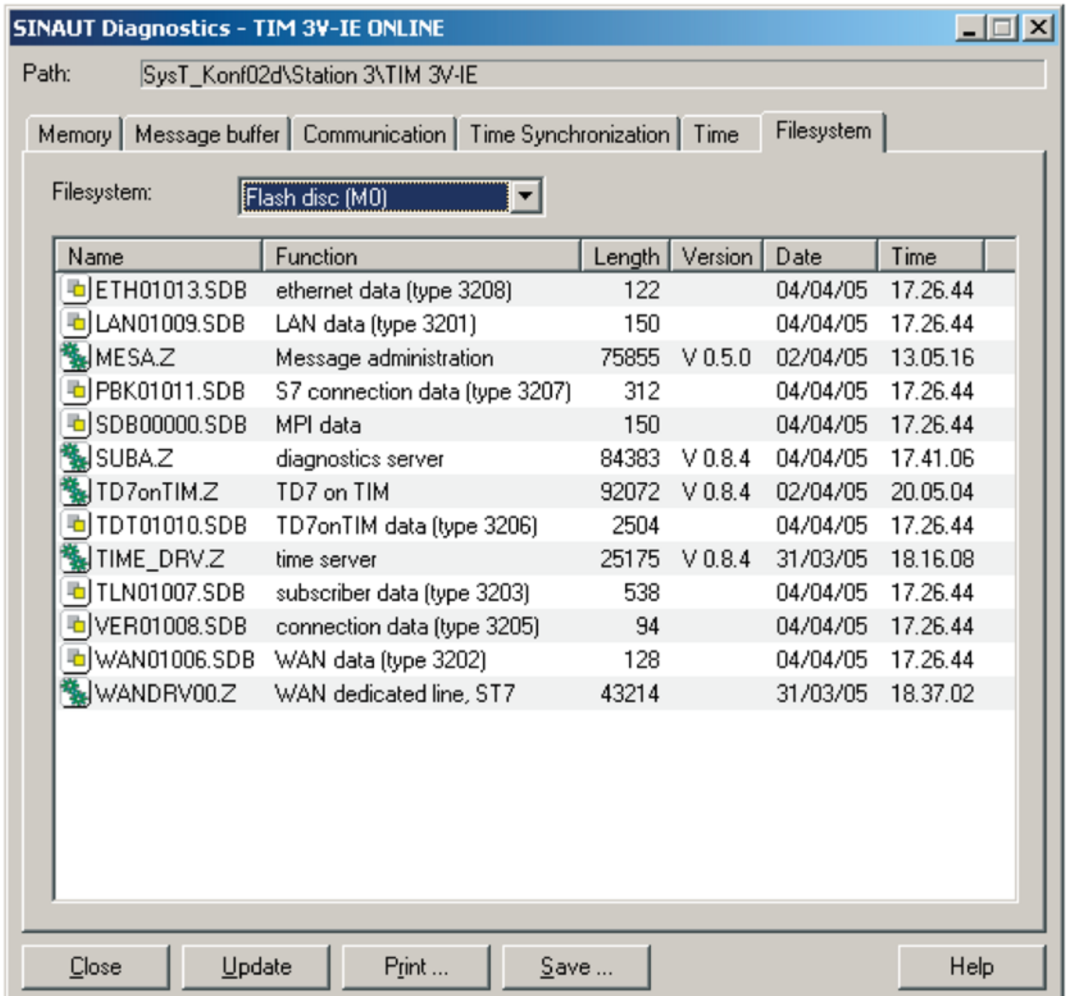


Figure 6-14 *SINAUT Diagnostics* dialog, Filesystem tab

IP Parameter tab

This tab is available only for the TIM 4R-IE.

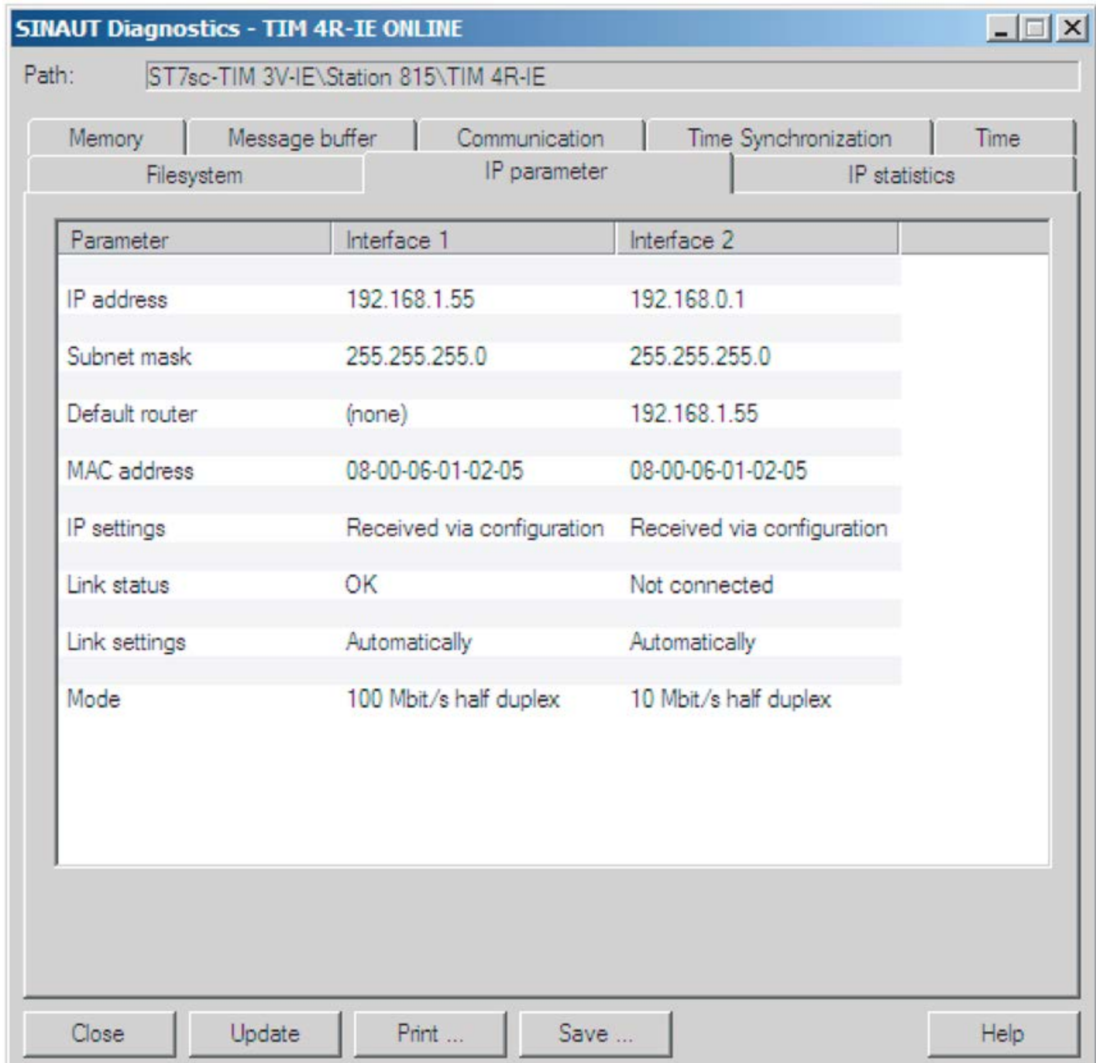


Figure 6-15 *SINAUT Diagnostics* dialog, *IP Parameters* tab

The *IP Parameters* tab displays the current IP parameters and settings of the Ethernet ports:

- IP Address: Configured IP address of the module.
- Subnet Mask: Configured subnet mask of the module.
- Default router: If a default router was specified during configuration, the IP address of the default router is displayed here.
- MAC address: MAC address of the module
- IP settings: Indicates where the module obtained the IP parameters from.
- Link Status: Indicates whether or not a physical connection to Ethernet exists.

- Link settings: Shows the setting for detecting network settings, here: "Automatic" (Autosensing)
- Mode: Indicates the transmission speed and duplexity on Ethernet.

Statistics tab

This tab is available only for the TIM 4R-IE.

The *Statistics* tab contains transmission statistics for the Ethernet ports. The number of transferred data packets with and without errors since the last reset or restart of the module is displayed for the send and receive directions.

MSC Communication tab

This tab exists only for a TIM 4R-IE as MSC master station.

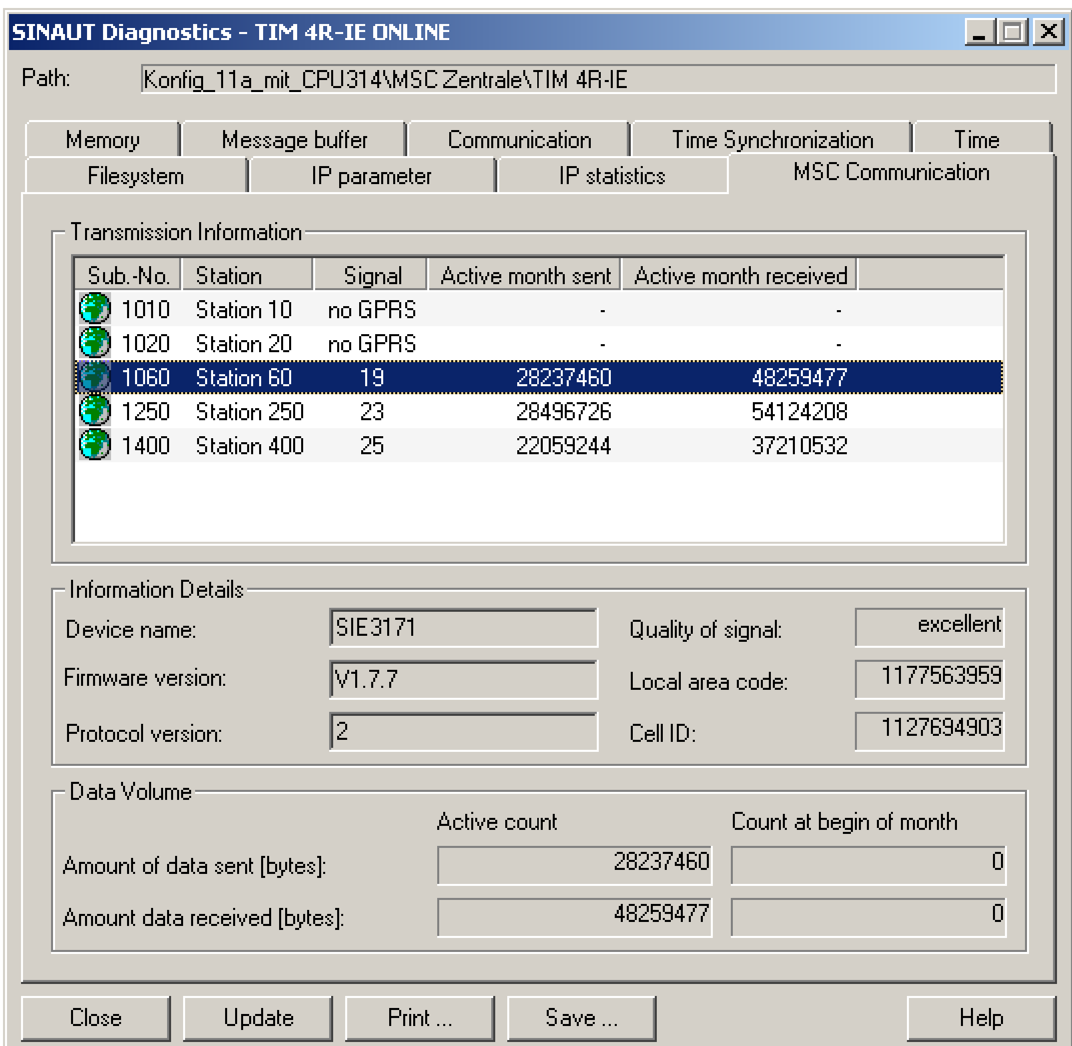


Figure 6-16 *SINAUT Diagnostics* dialog, *MSC Communication* tab

The "Transmission Information" list shows all MSC stations along with the following detailed information stored on the MSC master when the dialog is opened.

- Globe icon

  The globe icon in this dialog has a the following meaning depending on its color (from the perspective of the master station):

  – Black and white ( )

    The station can be reached online but the displayed data is not up to date. The reason for this may be that the response to a query was not received, for example due to a timeout. *)

  – Green ( )

    All the required properties of the network provider are configured. The network provider can be reached online.

  – Blue and red ( )

    The network provider cannot be reached.

  *) When the "MSC Communication" tab is opened, the icons of all stations are gray. The change to green or blue and red takes place only after the data has been updated (see below).

- Subscriber number
- Station
- Signal quality

  see below → "Quality of the received signal"

- Sent current month / Received current month

  Sent/received volume of data since the last change of month

  Difference between "current count - count previous month" (see below)

  – When the TIM of a DSL station is restarted, the counter is set to 0.

  – When the MD720 of a GPRS station is restarted, the counter is set to 0.

The fields below this show more details of the MSC station selected in the list:

- Device name

  Device name of the TIM

- Firmware version

  TIM firmware version

- Protocol version

  MSC protocol version

The following information is only supplied by GPRS stations with MD720:

- Signal quality

  Relative receive quality
  Range: 0...99

0 = error-free signal
99 = loss of synchronization

- Local area code

  Location of area in the mobile wireless network

- Cell ID

  ID number of the active GSM cell

- Data sent [bytes] / Data received [bytes]

  Volumes of data in bytes at the following times:

  – Current count

    Current counter reading: Volume of data since the first station TIM startup to the present time

  – Count in previous month

    Volume of data at the last change of month (1st day of the month)

Buttons in the *MSC Communication* tab

- Update

  The "Update" button updates the data of the stations as follows:

  – No station selected: The data of all stations is updated.

  – One station selected: The data of the selected station is updated.

  The requested data can take up to 30 seconds.

- Print...

  Opens a dialog from which you can print the displayed information.

- Save...

  Opens a dialog from which you can save the displayed information in a text file.

6.3.2 TIM subscriber diagnostics

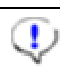
Description of the functions

The *TIM Subscriber Diagnostics* function displays the diagnostic data of the known SINAUT subscribers of the connected TIM module. The following detailed information is available:

- Selection list of the known subscribers (on the left):
  The selection list of the known subscribers is used to select individual subscribers known to the connected TIM module allowing the information to be displayed in the tabs on the right. The known subscribers are listed with their subscriber number and subscriber type, if accessed over a STEP 7 project the name and station is also displayed.
- *Status* tab:
  Information on the availability of the partners or connection disruptions and information on data communication and the operation of the send buffer
- *Partner* tab:
  Displays the known partners:
  - With CPU modules: Display in the known partners tab of the CPU selected in the list of known subscribers on the left of the dialog
- *Dialing extern* (optional):
  Dial-up service and command of the connected TIM module
- *Polling intern* (optional):
  Data of the station poll of the partners of the connected TIM module connected over a dedicated line with the option of disabling or enabling a connected partner

The colored symbols in the selection list of known subscribers indicate the availability of the individual subscribers and have the following meaning:

Table 6- 2 Symbols indicating subscriber availability in subscriber diagnostics

Symbol	Status
	Subscriber is available, all connections OK
	Subscriber is available, at least one connection is disrupted
	Subscriber is not available

Operator activities

1. Select a TIM module in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *SINAUT / TIM Subscriber Diagnostics* menu.
3. Select a subscriber in the selection list of the known subscribers on the left in the dialog.
4. To display the information on *Status*, *Partner*, *Dialing extern* or *Polling intern*, click on the relevant tab.

Status tab

The *Status* displays the following information on the subscriber selected on the left in the list of known subscribers from the perspective of the connected TIM module:

- In the *General* field:
  - Availability of the known subscriber. Entries indicating problems are highlighted.
  - Any connection disruptions
  - Information on gateways to the known subscriber (subscriber local / remote)
- In the *Connection* field:
  - Interface of the connection
  - Type of connection
  - Connection enable
  - Information on polling
  - Status of data communication
- In the *Special* field:
  - Information on the operation of the send buffer of the known subscriber

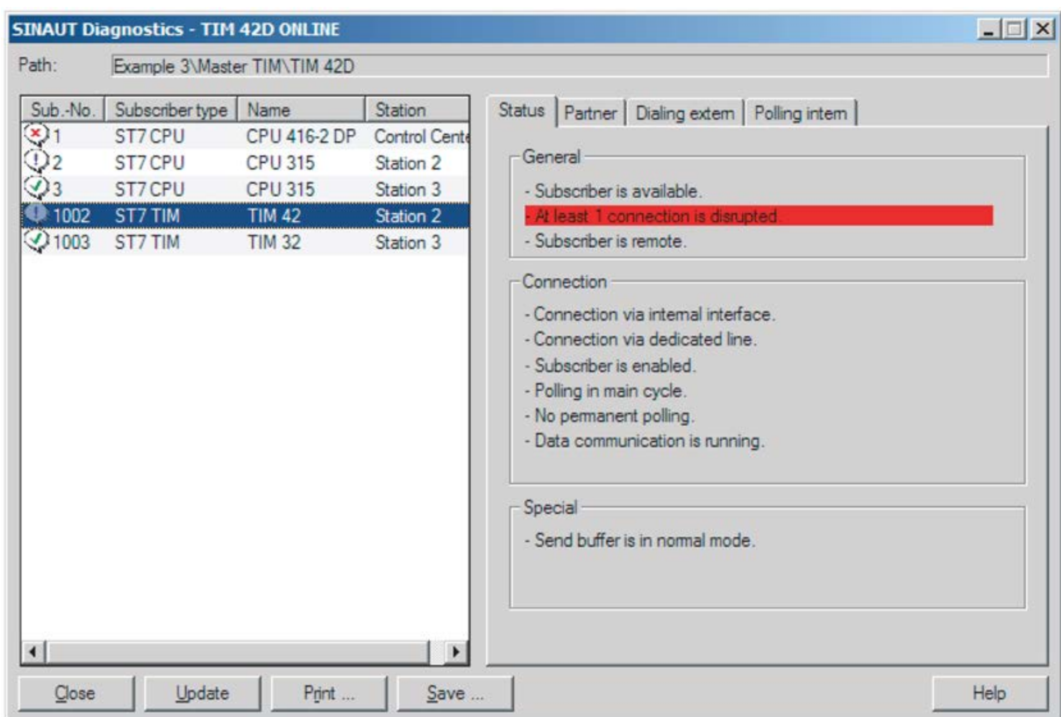


Figure 6-17 *TIM Subscriber Diagnostics* dialog, *Status* tab

Partner tab

The *Partner* tab displays the following known partners with their subscriber number, name and station in the *List of partners* box:

- With CPU modules: Display of the communication partners of the CPU selected in the list on the left of the dialog
- With TIM modules: Display of the communication partners of the connected TIM module selected in the subscriber list prior to opening the dialog.
  If different subscribers are selected on the left in the dialog, the same subscribers are always displayed in the *List of partners*.

With TIM modules, partners are only displayed if the TIM is installed in the master station and is connected to partners over a dedicated line.

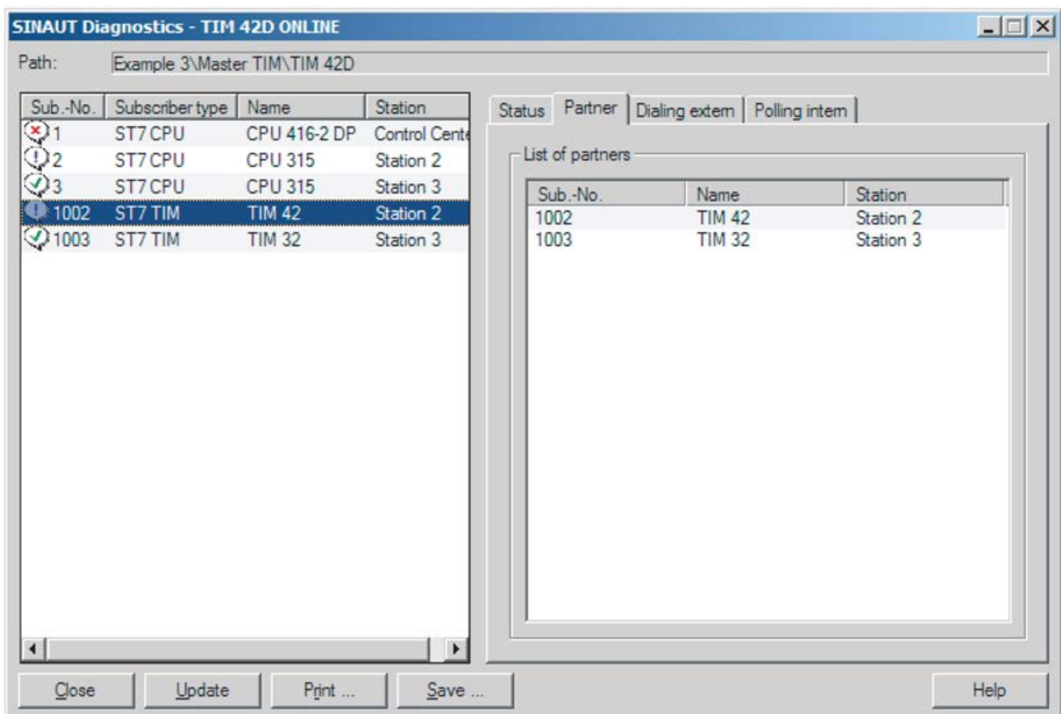


Figure 6-18 *TIM subscriber diagnostics* dialog, *Partner* tab

Dialing extern tab

If a TIM with master functionality is selected *before* you open the dialog, the *Dialing extern* tab displays the special services and the telephone number list (dial command) of the dial-up network driver of the connected TIM modules.

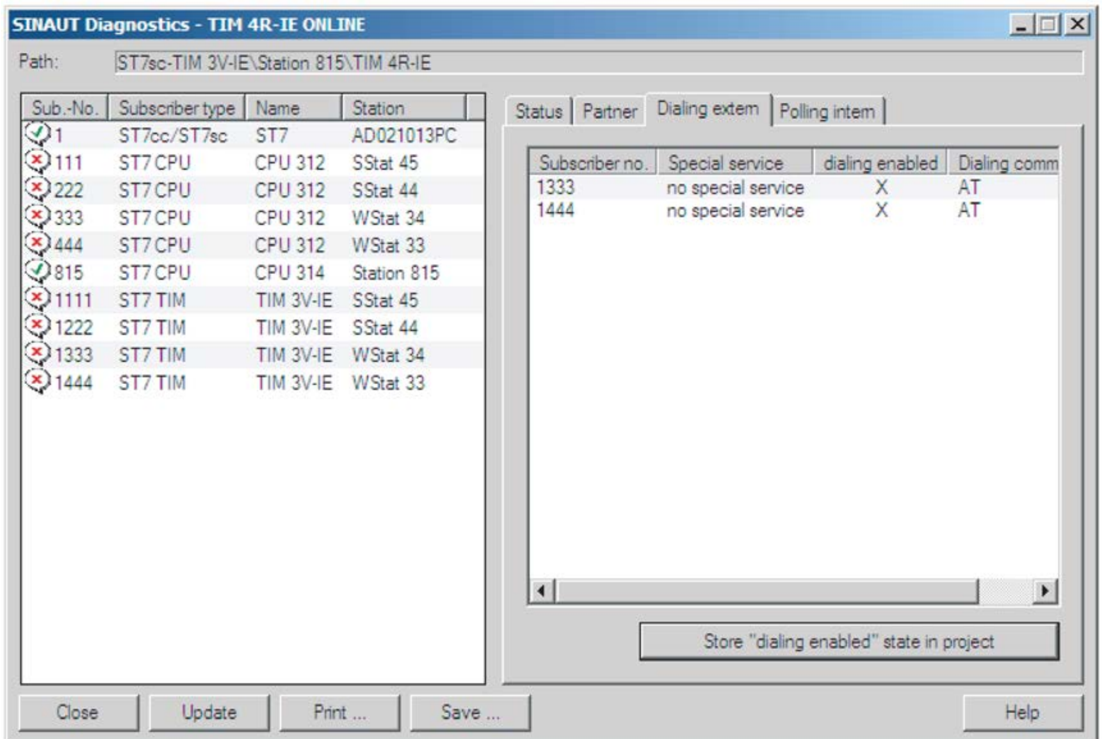


Figure 6-19 *TIM Subscriber Diagnostics* dialog, *Dialing extern* tab

The parameters *Subscriber number*, *Special service*, *Call enabled* and *Dial command* are displayed for the listed TIM modules.

An "X" in the *Call enabled* column means that the connection is enabled.

The following functions are available only for the TIM 4R-IE:

- By double-clicking on a subscriber in the dialing list, the *Disable / Enable Subscribers* dialog opens in which the configured and current enable states of the selected subscriber are displayed from the perspective of the master TIM. The current enable status can be changed.
- The settings for enabling subscribers are stored permanently.
- Using the button below the list, you store the current enable status in the STEP 7 project of the connected PG/PC.

Polling intern tab

If a TIM with master functionality is selected *before* opening the dialog, the *Polling intern* tab displays the data of the station for for the known subscriber connected to the connected TIM over a dedicated line.

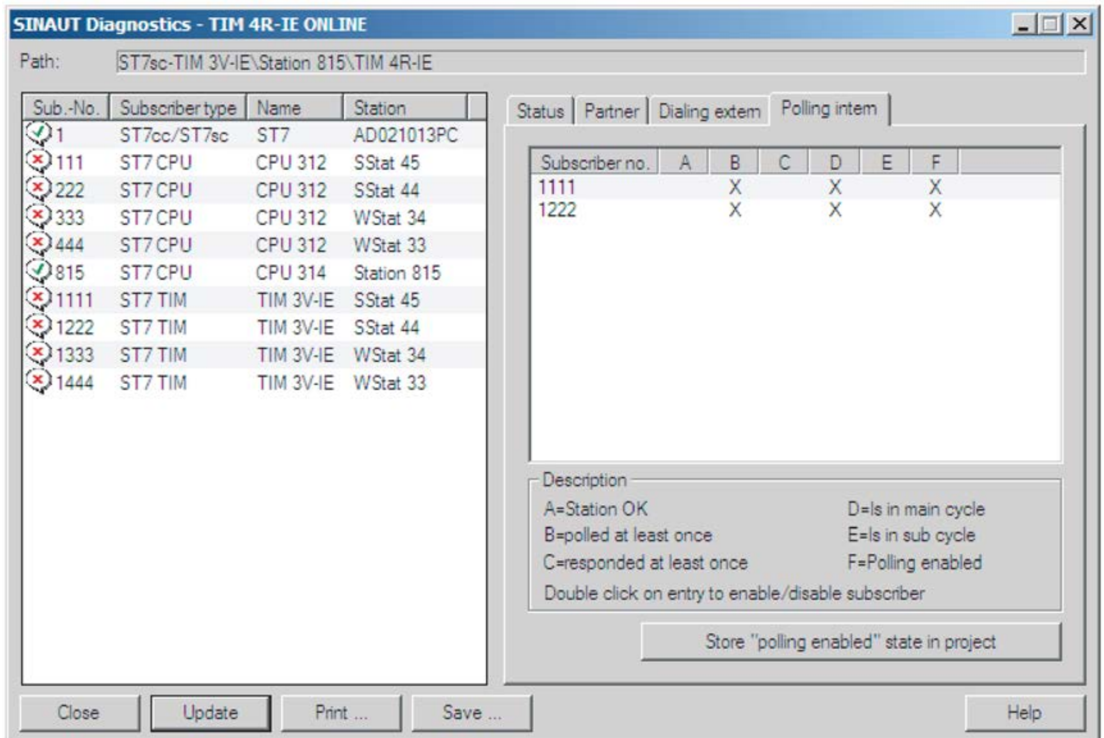


Figure 6-20 *TIM Subscriber Diagnostics* dialog, *Polling intern* tab

By double-clicking on a subscriber in the polling list, the *Disable / Enable Subscribers* dialog opens in which the configured and current enable states of the selected subscriber are displayed from the perspective of the master TIM. The current enable status can be changed.
If you change the status to *Disable* in this dialog, the configured status is adopted again next time the master TIM is reset.

The following functions are available only for the TIM 4R-IE:

- The settings for enabling subscribers are stored permanently.
- Using the button below the list, you store the current enable status in the STEP 7 project of the connected PG/PC.

6.3.3 TD7onTIM diagnostics

Description of the functions

Note

The *TD7onTIM Diagnostics* function is available only for TD7onTIM-compliant TIM modules on which parameters were set for the *TD7onTIM* software (for example TIM 3V-IE).

TD7onTIM Diagnostics provides information on the status of the data transmission of the *TD7onTIM* software package of the TIM module selected in the SINAUT subscriber list.

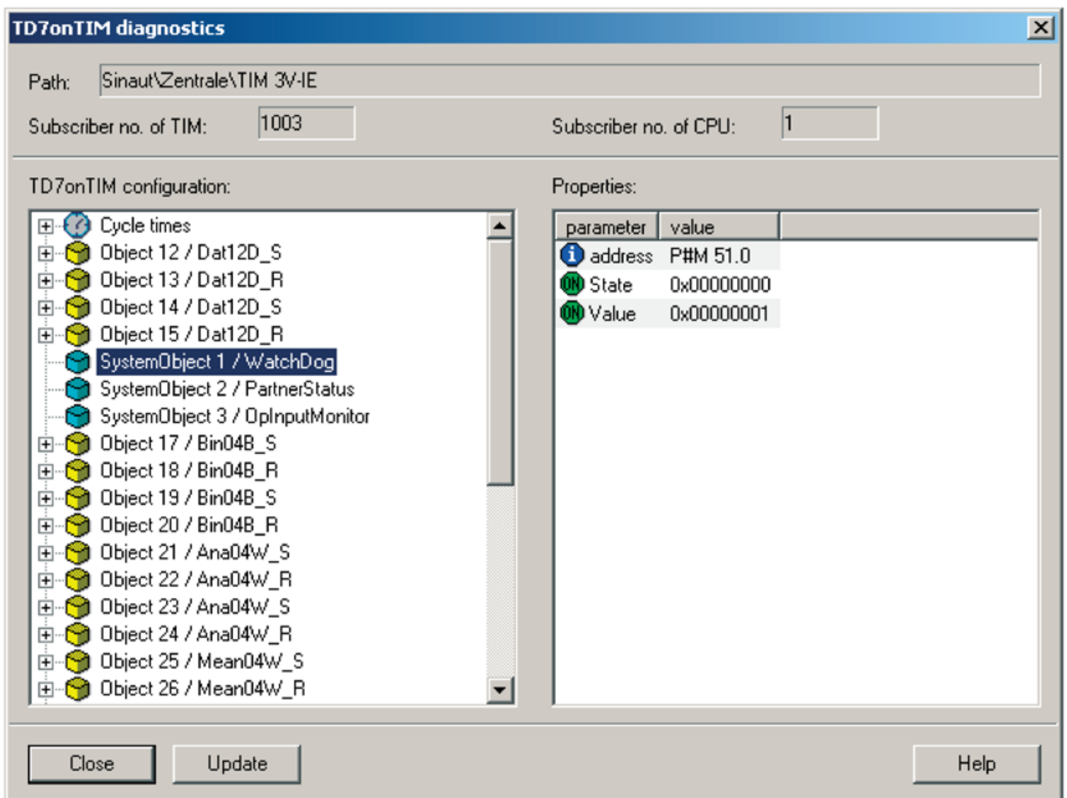


Figure 6-21 *TD7onTIM Diagnostics* dialog with system object parameters based on the example of *WatchDog*

In keeping with the parameter assignment of *TD7onTIM*, the diagnostic functions are displayed for the following objects:

- Status of system objects
- Status of data objects
- Status of the input and output channels

The dialog displays the following information for the station of the selected TIM module:

- The *path* of the TIM in the project
- The *subscriber no. of the TIM*
- The *subscriber no. of the CPU*

The lower part of the dialog displays the SINAUT objects and channels with their parameters:

- *TD7onTIM Configuration*:
  This box lists the following directories of the selected TIM as they are successively expanded:
  – System objects (blue symbols)
  – Data objects (yellow symbols)
  – The scan cycle time
  – Send and receive channels
- *Properties*:
  This area displays the following properties of an object selected in the directory tree:
  – *Parameter name*
  – *Value*
  – *Comment*

The *Parameter name* column lists the individual parameters with colored symbols indicating the following status:
- Blue symbols: Configured data
- Red symbols: Online data

The cycle time is displayed in the *TD7onTIM Configuration* box below the last data object. This is the current time of a sampling cycle in which TD7onTIM samples the work memory of its local CPU. If you click on the cycle time, the corresponding value is displayed in the *Properties* area.

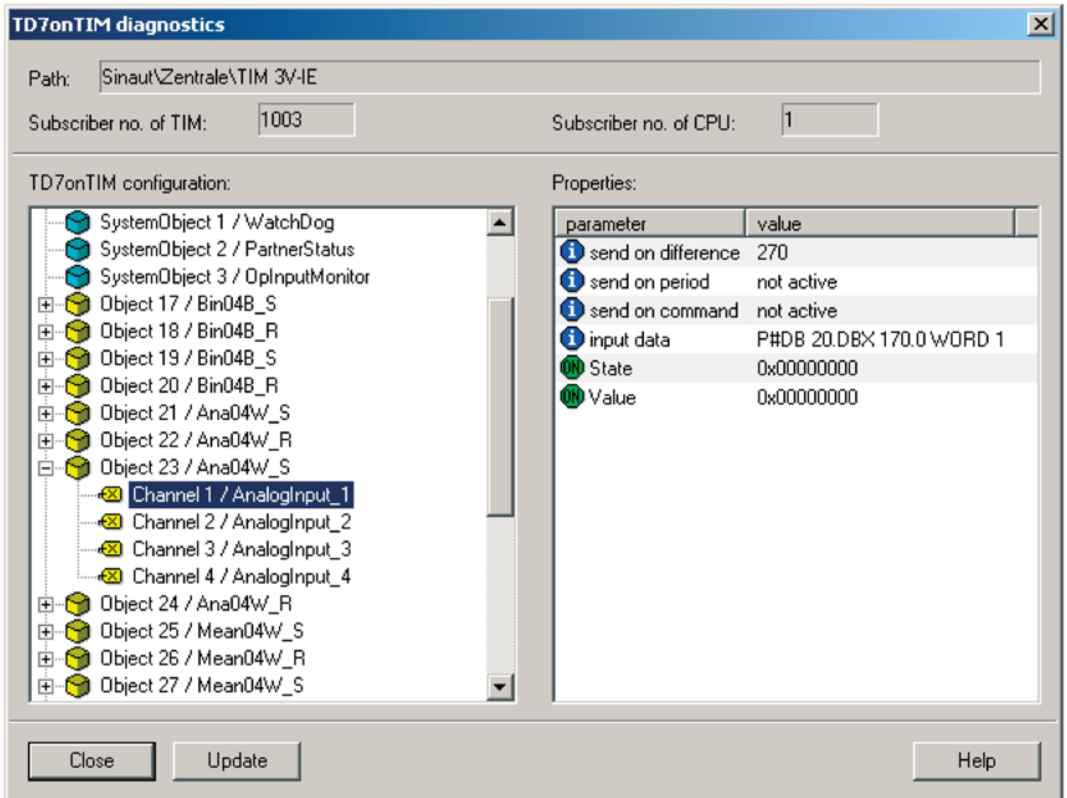


Figure 6-22 *TD7onTIM Diagnostics* dialog channel parameters based on the example of *Send analog value*

Operator activities

1. Select a TD7onTIM-compliant module in the *SINAUT subscriber list* of the open project.
2. Open the dialog by selecting the *SINAUT / TD7onTIM Diagnostics* menu. The dialog opens.
3. Expand the directory tree in the *TD7onTIM Configuration* area.
4. Select a SINAUT object or a send or receive channel in the opened directory tree. The relevant parameters are displayed in the *Properties* area.

6.3.4 TIM diagnostic messages

Functional description

With the *TIM Diagnostic Messages* function, **Extended diagnostic messages** are activated or deactivated and selected levels for various components of the TIM firmware. The extended diagnostic messages contain detailed information on subfunctions of individual firmware components and are entered in the diagnostic buffer of the TIM.

The extended diagnostic messages are displayed in hexadecimal format.

The display of extended diagnostic messages can be selected for various firmware components and some functions (diagnostic areas). The diagnostic areas of the TIM 3 / TIM 4 and the Ethernet TIMs are different:

Table 6- 3 Diagnostic areas of the TIM 3 / TIM 4

Firmware section of the TIM	Diagnostic area	Meaning / subfunction
Diagnostic server		
	1	Reception of organizational messages
	2	Management of organizational messages
	3	System status list query
	4	Data synchronization with driver and device redundancy
	5	Not used
Routing server		
	1	not assigned
	2	not assigned
	3	not assigned
	4	not assigned
	5	not assigned
	6	not assigned
Installation program		
	1	not assigned
	2	not assigned
Clock driver		
	1	WAN/LAN synchronization cycle
	2	not assigned
	3	not assigned
	4	not assigned
External WAN driver		
	1	Interrupt level (receive direction) error messages
	2	Interrupt level (receive direction) level 1
	3	Interrupt level (receive direction) level 2
	4	Control level
	5	Task management in WAN driver
	6	not assigned
	7	not assigned
Internal WAN driver		
	1	Interrupt level (receive direction) error messages
	2	Interrupt level (receive direction) level 1
	3	Interrupt level (receive direction) level 2
	4	Control level
	5	Task management in WAN driver
	6	not assigned
	7	not assigned

Table 6- 4 Diagnostic areas of the Ethernet TIMs

Firmware section of the TIM	Diagnostic area	Meaning / subfunction
Start manager		
	1 - 8	Reserved
P bus driver		
	1 - 8	Reserved
Clock driver		
	1 - 8	Reserved
LAN communication		
	1 - 32	Reserved
LAN communication		
	1 - 32	Reserved
Subscriber administration		
	1 - 32	Reserved
Message buffer		
	1 - 32	Reserved
TD7onTIM		
	1 - 32	Reserved
WAN driver 1		
	1 - 32	Reserved

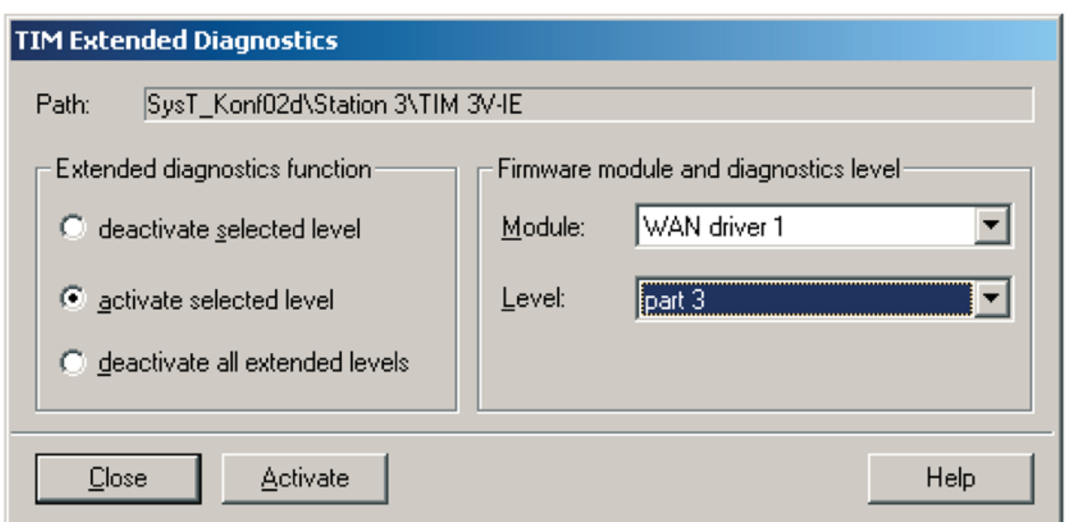


Figure 6-23 *TIM Extended Diagnostics* dialog

Operator input

1. Select a subscriber in the *SINAUT subscriber list* of the open project.
2. First open the *Module Information / Diagnostic Buffer tab* by selecting the *STEP 7 Diagnostics/ Module Information / Diagnostic Buffer tab* and click on the*Settings* button.
3. Make sure that the *Update display during operating mode transition* option is deselected (no check mark) at the bottom of the *Settings for Display Diagnostic Buffer* dialog and confirm with *OK*. You can leave the *Module Information* dialog open.
4. Change to the SINAUT diagnostics and service tool and open the *TIM Extended Diagnostics* dialog by selecting the *SINAUT / TIM Diagnostics* menu.
5. Select the *required function* in the field on the left of the dialog.
6. Then select the following in the *Firmware module and diagnostics level* area
   - The required firmware component in the *Module* list box and
   - The required level (area) in the *Level* list box.
7. Confirm your entries by clicking on the *Activate* button. A dialog *Loading* opens briefly and indicates that the activation information for extended diagnostics is being sent to the module by displaying a progress bar. Once the information has been sent successfully, the *Loading* and *TIM Extended Diagnostics* dialogs are closed.
   Any diagnostic messages are activated on the selected module and displayed in the active diagnostic buffer.
   If multiple extended diagnostic messages are activated, you must confirm the activation of the message output for each individual firmware component and level with *Activate*.
8. Change back to the *Module Information / Diagnostic Buffer tab* that is still open and click on *Update*, if necessary, several times. Extended diagnostic messages are displayed in hexadecimal code.
   If necessary, save the diagnostic messages as a text file as described for the *Module Information* function.
9. To disable the output of extended diagnostic messages for an individual level, select the *deactivate selected level* option in the *TIM Extended Diagnostics* dialog and close the dialog with the *Activate* button.
10. To disable the output of all extended diagnostic messages for all firmware components and all levels of the selected subscriber, select the *deactivate all extended levels* option in the *TIM Extended Diagnostics* dialog and close the dialog with the *Activate* button.

Note

If even one level of the extended diagnostic messages is activated, the size of the TIM diagnostic buffer is increased from 50 to 200 entries. Due to the increased memory requirements, extended diagnostic messages should not be activated permanently.

After recording and saving the extended diagnostic messages, deactivate the *Extended Diagnostics* function again. The simplest method is to a deactivate the function with the *deactivate all extended levels* option. When you deactivate extended diagnostics, the size of the TIM diagnostic buffer is set back from 200 to the basic setting of 50 entries.

6.3.5 TIM Message Monitor

Description of the functions

The *TIM Message Monitor* function is used to specify the settings for message monitoring of a selected TIM and starts the monitoring.

In TIM message monitoring, the messages received and sent by the TIM are recorded. As soon as the monitoring function is activated, copies of every message are stored in a buffer created specifically for this function. The messages are read out of the buffer of the TIM and saved in a monitoring file.

To start to message monitoring, you must set the following:

- the output file in which the recorded messages are saved and
- the stop action for the monitoring, either
  - manual for
    - continuous sampling (reading out the buffer at 5 second intervals) or
    - read data once after stop of monitor,
  - fill buffer only once or
  - time-limited reading by specifying the elapsed time.

Reading the data once after stopping monitoring is set if the messages are to be monitored following any intervention, for example turning off the TIM.

If the message buffer of the TIM is read once without any time limitation, the entire saved SINAUT data traffic of the relevant TIM is read out. With the fixed buffer size of the TIM, this involves 400 messages.

Monitoring is always started manually in the *TIM Message Monitor* dialog. When monitoring is started, the *TIM Message Monitor* progress bar is displayed and the group error LED of the TIM flashes while monitoring is active.

If the *manual* stop option is selected, monitoring is also stopped in the *TIM Message Monitor* progress bar.

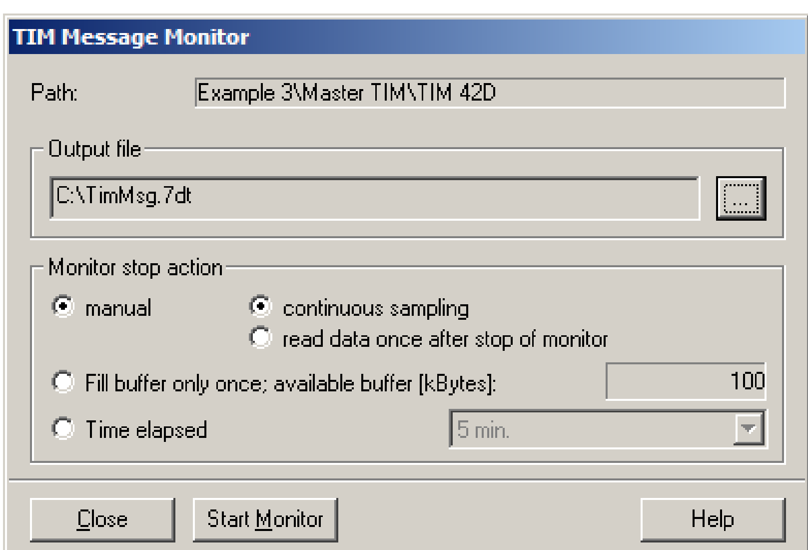


Figure 6-24 *TIM Message Monitor* dialog

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project.
2. Open the *TIM Message Monitor* dialog by selecting the *SINAUT / TIM Message Monitor* menu.
3. In the *Output file* box, enter the name of a file of the type **.7dt*, in which your recorded TIM messages will be saved or browse for the directory of the 7dt file in the file tree using the square button.
4. In the *TIM Message Monitor* dialog, in the *Monitor stop action* area, select one of the three options for starting message monitoring. If you select manual, you must decide whether the recorded messages are read by continuous sampling or after stopping monitoring.
5. Start monitoring with the *Start Monitor* button. The *TIM Message Monitor* progress dialog opens and displays information on the acquisition mode and the progress of the message recording over time indicating the elapsed and remaining time. The amount of data read and data remaining is also displayed.
6. Click on the *End Monitor* button in the *TIM Message Monitor* dialog to stop monitoring manually.
7. In the next dialog, decide whether you want to read the monitored messages immediately (the *TIM Message protocol* window opens) or whether you want to read the monitored messages at a later point in time (you return to the SINAUT subscriber list).

Note

You can open and evaluate the TIM message protocol later using the *TIM message protocol* function of the Diagnostics and Service tool.

6.3.6 TD7 CPU Diagnostics

Description of the functions

With the *TD7 CPU Diagnostic* function, all diagnostic messages in the diagnostic buffer of a previously selected CPU are displayed. As default this is 10 messages for the CPU. The display in this dialog is for:

- SINAUT TD7 generated diagnostic messages as plain text
- All other STEP 7 diagnostic messages in hexadecimal code.

The message list at the top in the dialog shows the diagnostic messages with information on the message number, time of day date and event. The entries are sorted in descending chronological order; in other words, the latest message is at the top.

In the lower part of the dialogue under *Details on the event*, the event ID and additional information for the message selected above in the message list are displayed.

Note

If in addition to the TD7 generated messages, you also want to see the STEP 7 diagnostic messages as plain text, open the *STEP 7 Diagnostics Module Information - Diagnostic buffer* tab and place this next to the already open TD7 CPU Diagnostics dialog.
You can then see the STEP 7 diagnostic messages displayed in the *TD7 CPU Diagnostics* dialog in hexadecimal format as plain text in the *STEP 7 Diagnostics / Module Information dialog*.

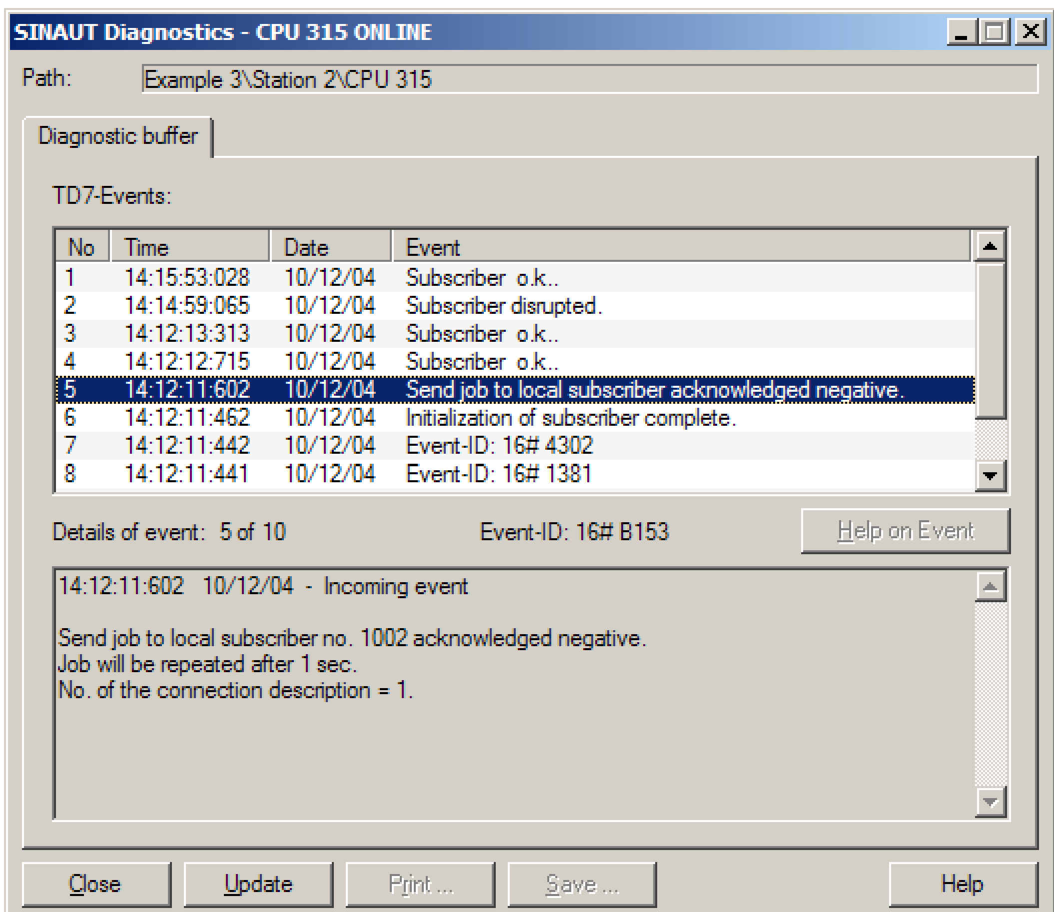


Figure 6-25 *SINAUT Diagnostics* dialog of the function *TD7 CPU Diagnostics*

Operator activities

1. Select a CPU module in the *SINAUT subscriber list* of the open project.
2. Open the dialog by selecting the *SINAUT / TD7 CPU Diagnostics* menu.
3. Select a message in the list, the additional information on this message is displayed below the message list in the *Details of event* text box.
4. Click on *Update* to update the display in the dialog with the most recent messages from the diagnostic buffer.

6.3.7 TD7 Block Structure

Description of the functions

The TD7 block structure function provides information on the software blocks of a CPU. The function starts the block structure analysis of a previously selected CPU. It does not access the module but rather the project data.

The results are displayed in four tabs.

- *Statistics* tab:
  Shows number, name, type und station of the selected subscriber as well as communications partners and number of objects of the communications relations
- *Block structure*:
  Shows information on the path and all SINAUT blocks and the data of the calls of all SINAUT blocks in a tree structure for the subscribers of the entire project
- *Block list*:
  Represents the data of all calls of SINAUT blocks in the form of a list
- *Plausibility* tab:
  Shows the results of plausibility checks for the SINAUT block calls for *BasicTask*, *Startup, Safe* and *ListGenerator* and on the unique assignment of DB numbers, partner numbers and partner object numbers

In all the tabs, the block structure of the subscriber can be stored in the form of an XML file using the *Save* function.

The block structure of another version, another subscriber or project that was previously saved as an XML file can be displayed for visualization in all tabs of the open dialog using the *Loading* function The previous view is then overwritten.

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project.
2. Open the dialog by selecting the *SINAUT / TD7 Block Structure* menu.
3. Select the relevant tab to display the information.

Statistics tab

The left-hand side of the *Subscriber information* box on the *Statistics* tab lists the subscriber number, the name, type and station name of the CPU selected earlier in the subscriber list.

If you use the diagnostics function "TD7 block structure for all CPUs", you can select the required CPU in the "Subscriber number" drop-down list.

The *Communication links* box on the right displays the communications partners of the selected CPU and the number of communications objects per communications partner (target subscriber).

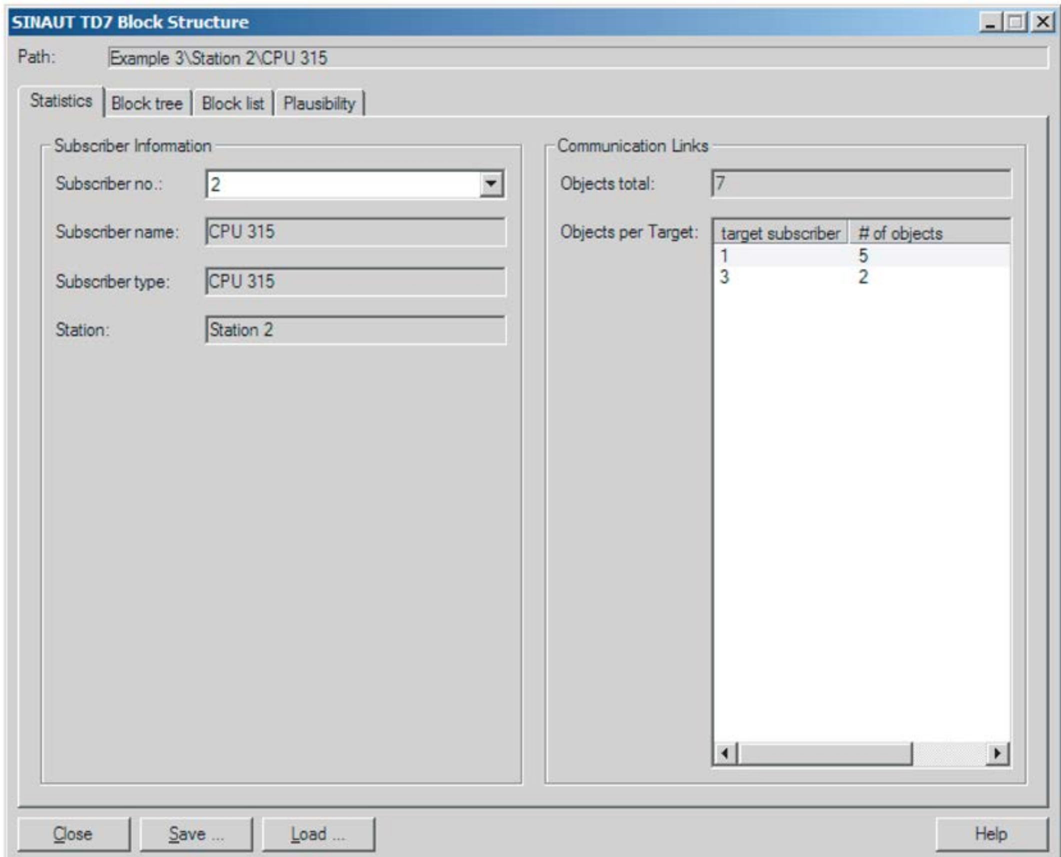


Figure 6-26 *SINAUT TD7 Block Structure* dialog, *Statistics* tab

Block structure tab

The *Block structure* tab shows the information found relating to the project partners and the previously selected subscriber in a tree structure. The following information is listed:

- Project information (*__Info*)
  with general information on the project path and the logical path
- *Subscriber* number / *station*name
  with all system blocks and user objects of the selected subscriber :
  - All SINAUT data blocks (*__Blocks*) in the program directory of the CPU
  - Information (__Info) on the module name, station name, type the name and subnumber of the subscriber
  - The system blocks, these can be: BASICTASK, PARTNERMONITOR, SAFE, PARTNERSTATUS, PULSECOUNTER, STARTUP, TIMETASK, STARTUP
  - The user objects (analog value, binary value, command, counter, setpoint objects etc.) With
    - information about their call data and a
    - list of the most important parameters of the individual user objects

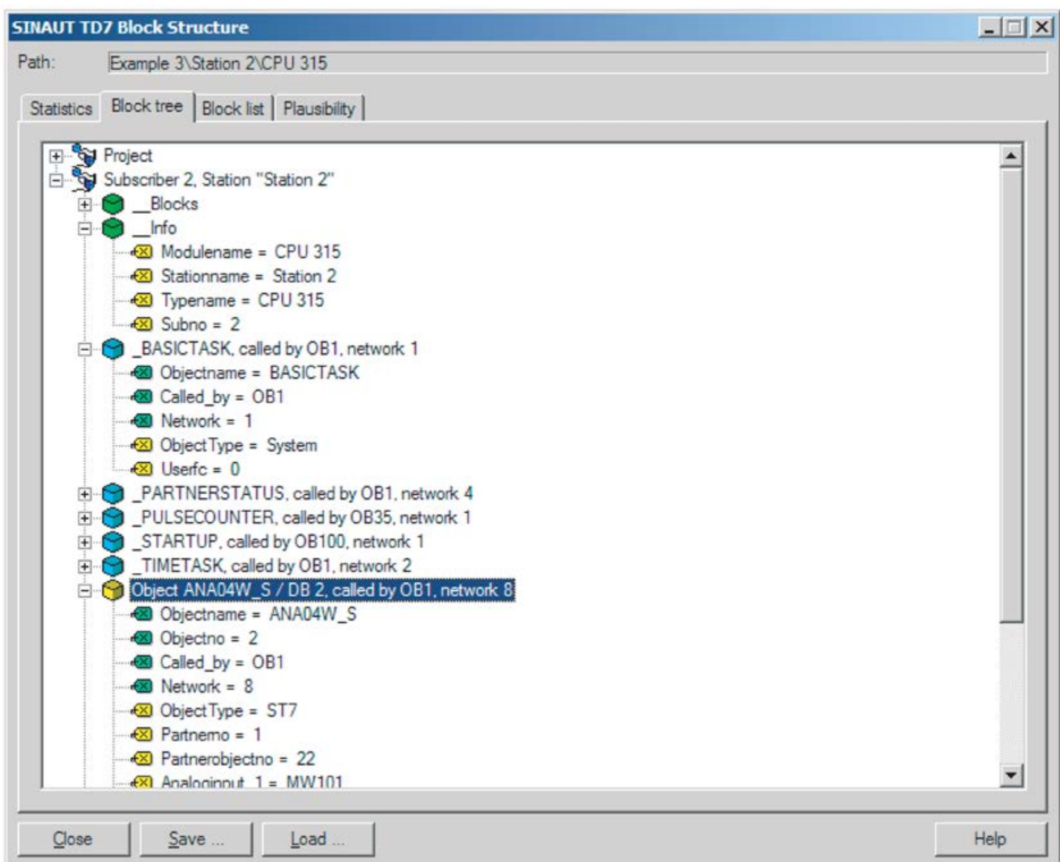


Figure 6-27 *SINAUT TD7 Block Structure* dialog, *Block structure* tab

Block list tab:

The *Block list* tab shows the data of all SINAUT block calls in the form of a list. As default, the blocks are displayed in the following five columns:

- **Subscriber**: This contains the block symbol and the subscriber number of the CPU. The system blocks have a blue symbol, the user blocks a yellow symbol.
- **Objectname**: Name of the system block or the user object
- **Objectno.**: Object number of the SINAUT object, the number of the instance DB used in the CPU
- **Partnerno.**: Subscriber number of the SINAUT destination or source subscriber as part of the SINAUT addressing (subscriber number, object number)
- **Partnerobjectno.**: Destination or source object number as part of the SINAUT addressing

The ascending sorting of the blocks within the block list can be changed by clicking on the column headers of the individual columns.

Further parameters are added to the view of the block list as follows:

1. Press the right mouse button within the and click *Add columns* in the shortcut menu that appears.
2. In the *Add columns* dialog that opens, click the required parameters and confirm with the *OK* button.
   The other block parameters are not included permanently in the list view, they are hidden again the next time the dialog is called.

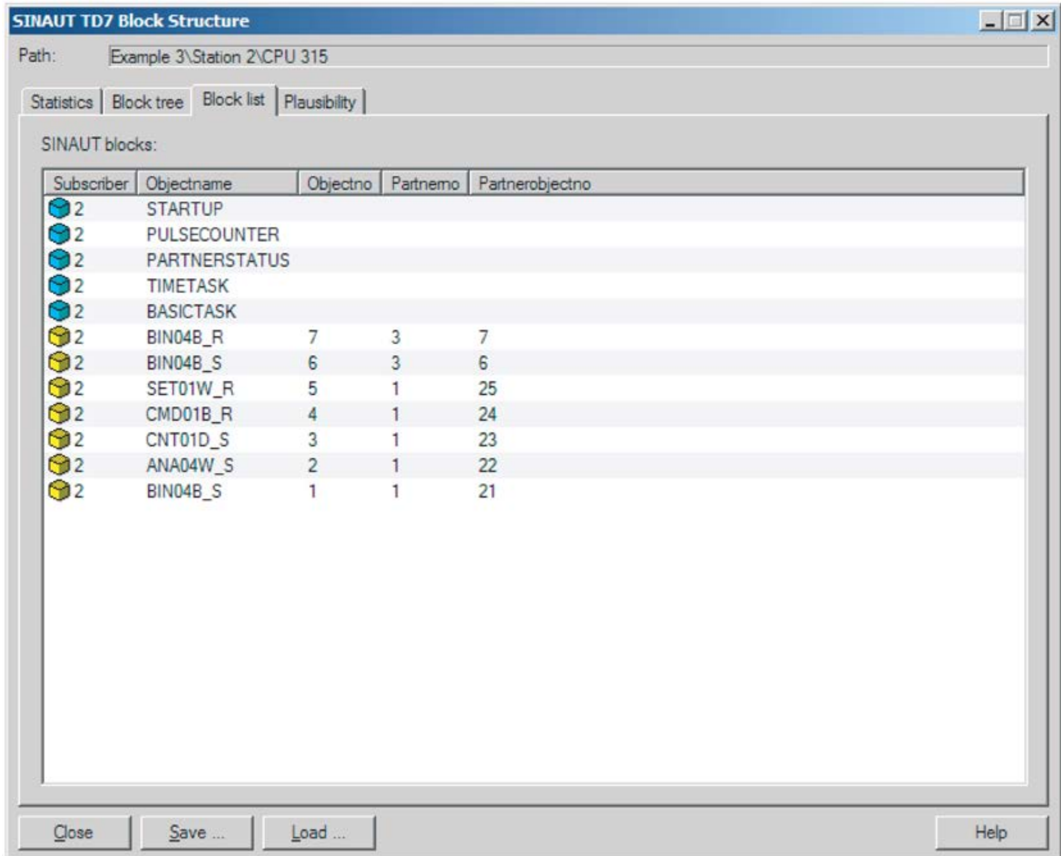


Figure 6-28 *SINAUT TD7 Block Structure* dialog, *Block list* tab

---

Note

The data of the block list is required when configuring a control center SINAUTcc/sc and can be stored for this purpose in an XML file.

Plausibility tab

In the *Plausibility* tab, a series of CPU-specific plausibility checks are made for the selected subscriber and displayed as the *Check result*. The following is checked:

- whether the SINAUT block *Startup* was called correctly,
- whether the SINAUT block *Safe* needs to be called and whether it was called,
- whether the object numbers (DB numbers) of the SINAUT blocks were assigned uniquely,
- whether the partner numbers and partner object numbers werw assigned uniquely,
- whether the SINAUT block *BasicTask* was called correctly,
- whether the SINAUT block *ListGenerator* needs to be called and whether it was called,

If errors occur in the plausibility checks listed above, a message to this effect is displayed in the *Check result* text output.

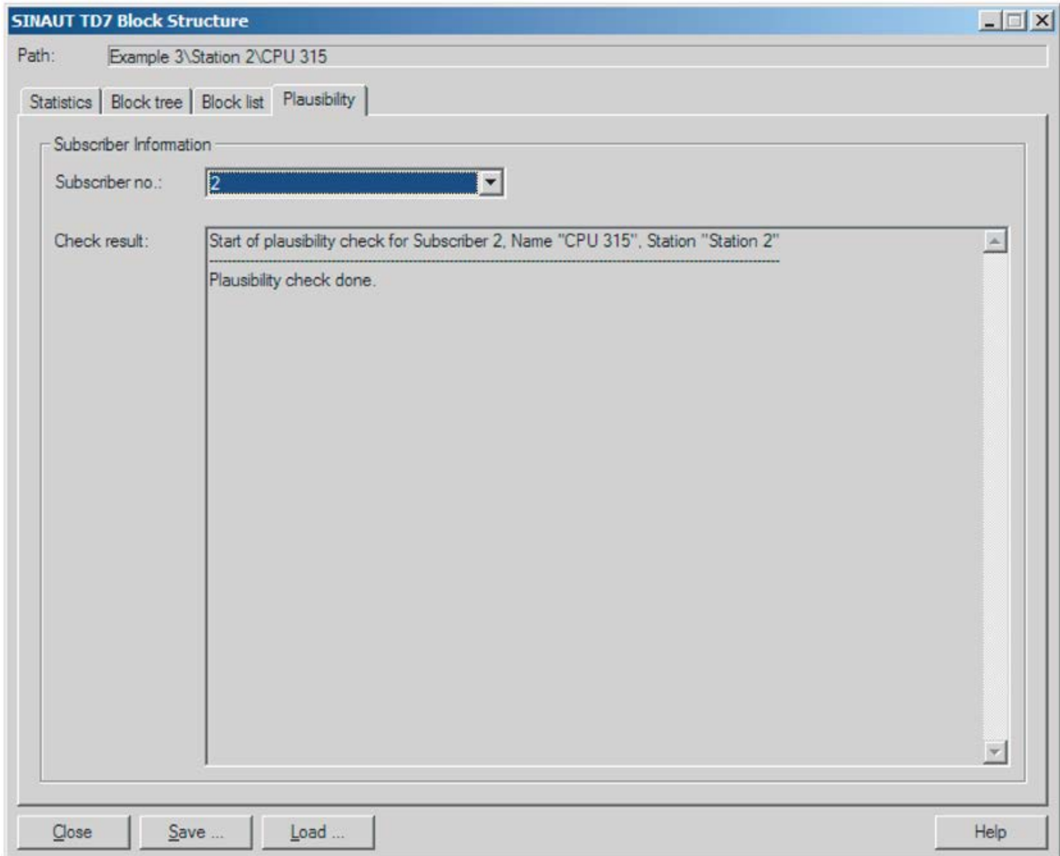


Figure 6-29 *SINAUT TD7 Block Structure* dialog, *Plausibility* tab

6.3.8 TD7 Block Structure for all CPUs

Description of the functions

The *Block Structure crore CPUs* function starts the CPU block structure analysis for the entire project. This function does not access the modules but rather the project data.

A CPU in the current project can also be compared with the CPU of different project by opening the *Block structure for all CPUs* dialog a second time in the SINAUT diagnostics and service tool. The data of the other CPU whose block structure data was created earlier as an XML file is loaded using the "Load" button in the dialog.

The results of the block structure analysis are displayed in four tabs. The meaning is analogous to that of diagnostics of the block structure of a single CPU (refer to the link on the relevant tab).

- *Statistics* tab:

  Refer to TD7 Block Structure (Page 451)

- *Block structure* tab:

  Refer to TD7 Block Structure (Page 451)

- *Block list* tab:

  Refer to TD7 Block Structure (Page 451)
- *Plausibility* tab:

  Refer to TD7 Block Structure (Page 451)

In all the tabs, the block structure of the project can be stored in the form of an XML file using the *Save* function.

Note

Since the blocks of all CPU modules must be decompiled into STL source files for the *TD7 Block Structure for all CPUs* function, this function can take a considerable time in extensive projects.

Operator activities

1. Open the dialog by clicking on the *SINAUT / TD7 Block Structure for all CPUs* menu.
2. Select the relevant tab to display the information.

6.3.9 TD7 CPU Program Comparison

Description of the functions

The *TD7 CPU program comparison* function displays the results of the comparison of the program of two CPU modules for all TD7 objects. This function does not access the data of the modules but rather the project data.

The *CPU program compare result* dialog displays the names and stations of the compared CPU modules in the *Compared stations / CPUs* area. On the left-hand side of the *Results* area below this, the components and subcomponents (TD7 objects) of the two CPU modules are listed with a symbol indicating the status of the comparison.

Table 6- 5 Significance of the comparison symbols for components in a CPU program comparison

Symbol	Status
	The objects found are different.
	The objects found are identical.
	The object serves only as information.

When you select a component in the list on the left, the parameter names of the components and a comparison symbol are displayed for CPU 1 and CPU 2 in the right-hand part of the *Results* area. If the name of a parameter is identical on both CPU modules, the names displayed only under CPU 1. The comparison symbols have the following meaning:

Table 6- 6 Significance of the comparison symbols for parameters in a CPU program comparison

Symbol	Meaning
1	Subcomponents exist only on CPU 1.
2	Subcomponents exist only on CPU 2.
≠	Components on CPU 1 and 2 are different.
=	Components are identical on both CPUs.

The data of the program comparison can be saved in an XML file. In the same way, data of an earlier program comparison that was saved in an XML file can be displayed again in the dialog using the *Load* button. The previous display is then overwritten.

If you selected the wrong file when loading, an error message is displayed and the dialog display is deleted.

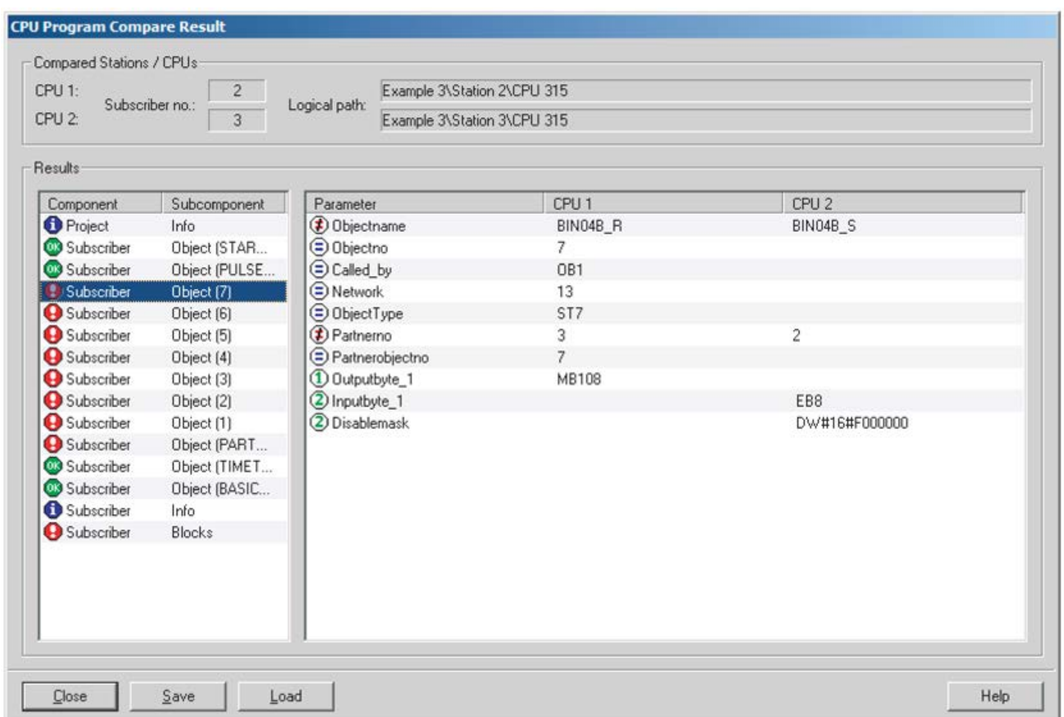


Figure 6-30 *CPU Program Compare Result* dialog

Operator activities

1. Select a CPU in the *SINAUT subscriber list* of the open project. This is identified as CPU 1 in the following dialog.
2. Start the *TD7 CPU program comparison* function by selecting the *SINAUT / TD7 CPU Program Comparison* menu.

   The "Compare TD7 block structures" dialog opens.
3. Select a CPU of the project has CPU 2 for comparison in the *CPU 2* box, below the default option *Use CPU*.
4. As an alternative, select *Use file* in the *CPU 2* box and open a previously saved XML file with the block data of any other CPU module by clicking on the *...* button.
5. Click on the *Compare* button. The *CPU Program Compare Result* dialog opens

---

Note

If you use the XML file with the block structure data of a CPU other than *CPU 2* in the *Compare TD7 Block Structures* dialog, the XML file must only contain the data of one CPU. Otherwise, the dialog with the compare results will not open.

6.3.10 TD7 Check of the Communication Configuration

Description of the functions

The *TD7 communication configuration check* function is used to compare the TD7 communications objects of two CPU modules that communicate via SINAUT connections.

In the CPU program - comparison results dialog, the name of the station of the compared CPU modules is displayed in the Selected stations / CPUs box.
In the *Results* box below this, the objects of the two CPU modules are listed in two columns on the left-hand side with a symbol indicating the comparison status.

Table 6- 7 Comparison symbol for components of the TD7 check of the communication configuration

Symbol	Status
	The objects found are different.
	The objects found are identical.
	The object serves only as information.

When you select a pair of objects in the list on the left, the names of the relevant object parameters and a comparison symbol are displayed for CPU 1 and CPU 2 in the right-hand part of the *Results* area. The comparison symbols have the following meaning:

Table 6- 8 Comparison symbol for parameters of the TD7 check of the communication configuration

Symbol	Meaning
1	Subcomponents exist only on CPU 1.
2	Subcomponents exist only on CPU 2.
≠	Components on CPU 1 and 2 are different.
=	Components are identical on both CPUs.

Since parameters of the two CPU modules with the same functions are listed side by side in the right-hand part of the *Results* area, the objects of the two CPU modules must be crossed over to compare their communication plausibility. For example, the *object* number X of CPU 1 is the *partner object* number X of CPU 2 and vice versa.

The data resulting from comparing the communication configuration of two CPU modules can be saved in an XML file. In the same way, data of an earlier comparison of the communication configuration that was saved in an XML file can be displayed again in the dialog using the *Load* button. The previous display is then overwritten.

If you selected the wrong file when loading, an error message is displayed and the dialog display is deleted.

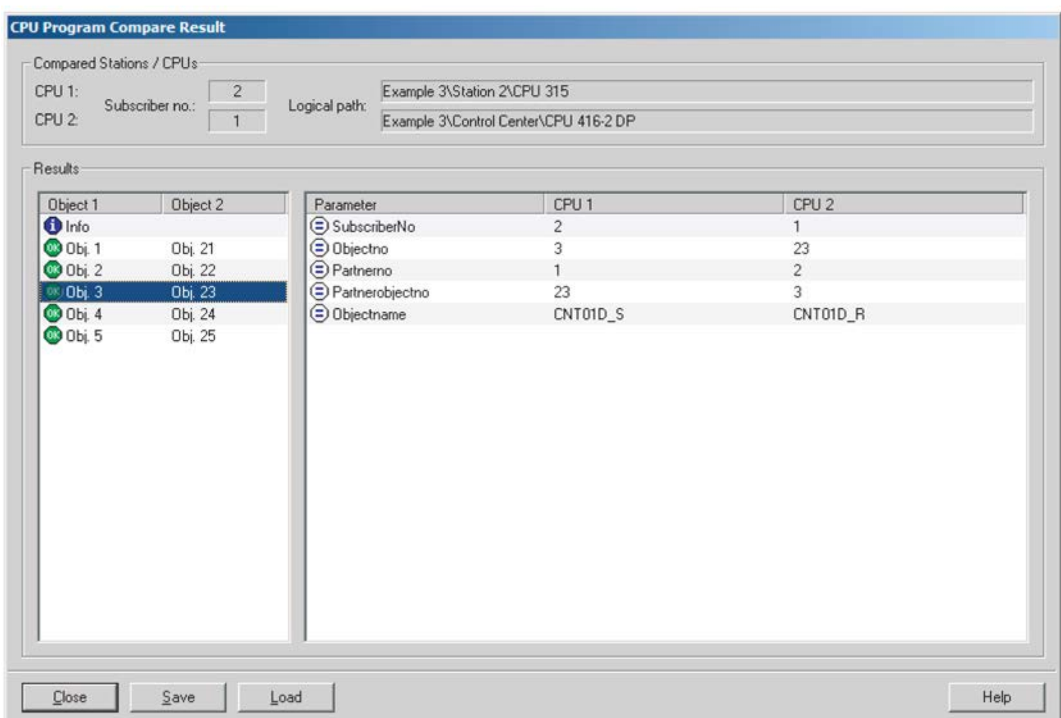


Figure 6-31 *CPU Program Compare Result* dialog

Operator activities

1. Select a CPU in the *SINAUT subscriber list* of the open project. This is identified as CPU 1 in the following dialog.
2. Start the *TD7 communication configuration check* function by selecting the *SINAUT > TD7 communication configuration check* menu.

   The "Compare TD7 block structures" dialog opens.
3. Open the *Compare TD7 block structures* dialog by selecting the *SINAUT > TD7 communication configuration check* menu.
4. Select a CPU of the project has CPU 2 for comparison in the *CPU 2* box, below the default option *Use CPU*.
5. As an alternative, select *Use file* in the *CPU 2* box and open a previously saved XML file with the block data of any other CPU module by clicking on the *...* button.
6. Click on the *Compare* button. The *CPU Program Compare Result* dialog opens

---

Note

If you use the XML file with the block structure data of a CPU other than *CPU 2* in the *Compare TD7 Block Structures* dialog, the XML file must only contain the data of one CPU. Otherwise, the dialog with the compare results will not open.

6.3.11 SDB Viewer

Description of the functions

The *SDB Viewer* function lists the content of the system data blocks (SDBs) of a previously selected CPU or TIM module.

For TIM modules, the following SDB classes can be selected for display:

- SDB0
- WAN data
- Subscriber data
- Connection data
- LAN connections
- TD7onTIM data (only TIM modules on which parameters have been set for TD7onTIM)
- Ethernet data (only TIM modules that are connected to Ethernet)
- Routing data
- Connection data (PBC)

Apart from SDB0, the representation of the SDB data for the TIM modules is in plain text (as default), you can, however, change to hexadecimal display. The CPU DBs are always displayed in hexadecimal and the corresponding button cannot be deactivated here.

When accessing the subscriber using the *SINAUT subscriber list*, you have the option of displaying the system data blocks from the module (*online*) or from the project (*offline*). There may be differences between online and offline access.

The content of all system data blocks can be saved as a text file.

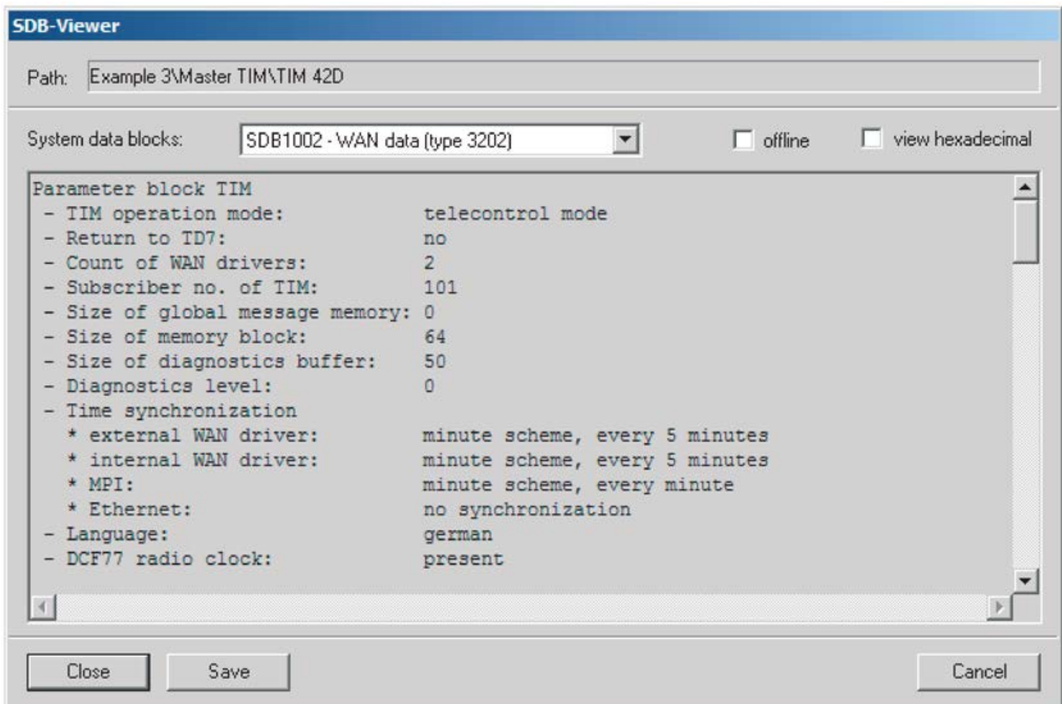


Figure 6-32 *SDB display* dialog. In the example, *SDB1000 - WAN data* is selected.

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the *SDB Viewer* dialog by selecting the *SINAUT / SDB Viewer* menu.
3. To access data online on the connected module (instead of the project data), deselect the *offline* option on the right to the dialog. The data display is updated immediately with the current module data.
4. Select the required system block class (SDB0, WAN data etc.) in the *System data blocks* list box.
5. Click on the *Save* button to save the content of all system data blocks as a text file.

SDB Viewer - SDB0

SDB0 is displayed in hexadecimal format and contains information on the following points:

- Communication parameters for the MPI bus
- Parameters for the time response of the MPI bus

- MPI address of the module
- Rack configuration with rack and slot addresses for S7-300 stations

WAN data SDB

The WAN data SDB contains information on the parameter assignment of the TIM module and the WAN drivers as illustrated below based on an example.

Table 6- 9 Example of the information of a WAN data SDB (type 3202)

Parameter block	**Parameter**
Parameter block TIM	- TIM operation mode: telecontrol mode - Return to TD7: no - Count of WAN drivers: 2 - Subscriber no. of TIM: 1002 - Size of global message memory: 0 - Size of memory block: 64 - Size of diagnostics buffer: 50 - Diagnostics level: 0 - Time synchronization * external WAN driver: minute scheme, every 5 minutes * internal WAN driver: minute scheme, every 5 minutes * MPI: minute scheme, every minute * TIM bus: no synchronization - Language: german - DCF77 radio clock: not present - Size of ram disk: 0 - Minimum heap reserve: 85 - TIM bus present: 0 - No of master: 1
Parameter block WAN driver 1	Base parameters: - Interface: external - TIM type: station - Net type: dial-up network - Operating mode: spontaneous - Message format: FT1.2 - Acknowledgment: short acknowl. - Retry factor: 7 - WAN protocol: ST7 - Master/node station/station no.: 2 - Max. message length: 240 - Number of local CPUs: 1 - General request priority: 0 - Number of spontaneous messages: 0 - Baud rate: 38,400 - Call answer delay: 0 - Country mode: Germany - Dialing mode: AT mode - Dialing format: 8 data bits, no parity, 1 stop bit - Extra transmission time: 0 - Customer identification: 0 - Paramet. for cond. spont. mess.: standard conditions - Limit for locked messages: 0 - Transfer mode: send single messages

Parameter block	Parameter
	- Cancel delay time: 0 - Operating mode: Interrupt (Block) - # of subscribers: 2
	Telephone number list: - Block 1 own telephone number * Telephone number 1 Telephone number: 2
	- Block 2 remote telephone numbers Redialing attempts: 3 Cancel parameter: 0 * Telephone number 1 driver type master No special service Station address: 1 Telephone number: ATDP3
	AT string: ATS45=3\N0F0&W
Parameter block WAN driver 2	Base parameters: - Interface: internal - TIM type: station - Net type: dedicated line - Operating mode: polling - Message format: FT1.2 - Acknowledgment: short acknowl. - Retry factor: 3 - WAN protocol: ST7 - Master/node station/station no.: 2 - Max. message length: 240 - Number of local CPUs: 1 - General request priority: 0 - Number of spontaneous messages: 20 - Number of permanent messages: 0 - Station address of cyclic partner: 1 - Baud rate: 19,200 - Polling time: 0 - RTS/CTS delay time: 0 - Send delay time: 0 - Extra transmission time: 0 - Limit for locked messages: 0 - Operating mode: Interrupt (Block) - # of subscribers: 2

Subscriber data SDB

The subscriber data SDB contains information on the settings for data of all subscribers of a project. The content of a subscriber SDB is shown below based on an example.

Table 6- 10 Example of the information of a subscriber data SDB (type 3203)

Subscriber	Parameter
# of subscribers:	5

Subscriber	Parameter
Subscriber 1	Subscriberno.: 1 Subscriber info: 0x0 ST7-CPU additional info: 0x0 send NO status object to subscriber count of connection blocks: 2 device | subnet ID | CFB/MPI | state | connection | STA int. WAN | 008a00000001 | 0 | remote | WAN connection.,ST7 | 0x01 ext. WAN | 008a00000001 | 0 | remote | WAN connection.,ST7 | 0x01 count of partner blocks: 1 subscriber no.: 2
Subscriber 2	Subscriberno.: 1003 Subscriber info: 0x1 ST7 TIM etc.

Connection data SDB

The connection data SDB contains information on the parameter settings of the local X connections of a TIM module to their CPU. This is illustrated below based on an example.

Table 6- 11 Example of the information of a connection data SDB (type 3205)

```
count of X com. blocks:      1
block                 connection type         loc. device ID      target MPI
1                     static                  0                   2
```

LAN connection SDB

The LAN connection SDB contains summarized information on the parameter assignment of all LAN connections on a TIM module. This is illustrated below based on an example.

Table 6- 12 Example of the information of a LAN connection SDB (type 3201)

```
local subscriberno.:           1002
local MPI address:              3
count of LAN blocks:           2
                               block   connection time     CFB/MPI  size of queue
                               1       X com not conf.     2        64
                               2       PBC                 4        64
count of TIM blocks:           1
                               Sno.    MPI   additional info
                               1001    8     0x0
DCF77 NOT built in
CPU slave
external WAN driver NOT active
internal WAN driver NOT active
count of local partners:       2
                               MPI     Rack  Slot
                               2       0     2
                               3       0     7
```

TD7onTIM data SDB

The TD7onTIM data SDB contains information on the parameter assignment of the SINAUT objects, their send and receive channels and the source or destination subscribers. An excerpt of this information is shown below based on an example.

Table 6- 13 Example of the information of a TD7onTIM data SDB (type 3206)

```
Parametertype T4T SDB Main Header = 61185
Length of block:                                        12
Count of target subscribers:                            3
Count of source subscribers:                            1
Count of objects:                                       7
Count of system objects:                                3
----------------------------------------------------
----------------------
Parametertype T4T SDB Target Subscriber = 61186
Length of block:                                        12
Type of target subscriber:                              4
Subscriberno.:                                          8
Supervision time:                                       900
Timestamp:                                              1
----------------------------------------------------
----------------------
Parametertype T4T SDB Target Subscriber = 61186
Length of block:                                        12
Type of target subscriber:                              0
Subscriberno.:                                          2
Supervision time:                                       900
Timestamp:                                              1
----------------------------------------------------
----------------------
Parametertype T4T SDB Source Subscriber = 61187
Length of block:                                        30
Subscriberno. of TIM:                                   1003
MPI address of TIM:                                     8
Subscriberno. of CPU:                                   4
MPI address of CPU:                                     7
Ext. Time Stamp:                                        0
Max Spon Out:                                           3
Max Main In:                                            2
Max Sub In:                                             5000
Scan Delay:                                             100
Max Input Time:                                         0
Input Delay Time:                                       0
Max Connect Time:                                       0
Address Check:                                          1
----------------------------------------------------
----------------------
Parametertype T4T SDB System Object = 61188
Length of block: 12
Type of system object: 32,512
DBNo | memLoc | byteAdr | bitAdr|
 7 | DB | 1 | 0 |
```

```
----------------------------------------------------
---------------------
Parametertype T4T SDB System Object = 61188
Length of block: 28
Type of system object: 32,513
DBNo | memLoc | byteAdr | bitAdr|
 7 | DB | 2 | 0 |
Partner | Subscriberno.
 1 | 8
 2 | 2
 3 | 0
 4 | 0
 5 | 0
 6 | 0
 7 | 0
 8 | 0
----------------------------------------------------
---------------------
Parametertype T4T SDB Partner = 61190
Length of block: 6
Partnerno.: 1
----------------------------------------------------
---------------------
Parametertype T4T SDB Partner = 61190
Length of block: 6
Partnerno.: 8
----------------------------------------------------
---------------------
Parametertype T4T SDB Channel = 61191
Length of block: 48
Channel active: 1
Channel type = Message Send = 60929
Send On Difference: 1
Send On Period Active: 0
Send On Period: 10
Send On Command Active: 0
Send On Command:
DBNo | memLoc | byteAdr | bitAdr|
 0 |UNKNOWN| 0 | 0 |
Alarm Mask: 0
Send Buffer Principle Mask: 0
Disable Mask: 0
Input Data:
dataType | repeatF | dbNo | memLoc | byteAdr | bit-
Adr|
 BYTE | 1 | 0 | M | 11 | 0 |
----------------------------------------------------
---------------------
```

Ethernet data SDB

The Ethernet data SDB contains information on the IP address of the selected TIM module, the subnet mask and any configured router. If, as in the case here, the address of the router is the same as the IP address of the TIM, no router is set. The IP addresses are set in network configuration.

Table 6- 14 Example of the information of an Ethernet data SDB (type 3100)

```
IP address       : 140. 80. 0. 3
subnet mask      : 255. 255. 0. 0
default router   : 140. 80. 0. 3
```

Routing data SDB

The routing data SDB contains information on the individual subnets of a project. This is illustrated below based on an example.

Table 6- 15 Example of the information of a routing data SDB (type 3002)

Subnet type	**ID, address**
local subnet	- local subnet ID : 47 11 00 00 00 0c - local device ID : 03
local subnet	- local subnet ID : 47 11 00 00 00 14 - local device ID : 02
remote subnet	- remote subnet ID : 00 8a 00 00 00 01 - next station addr : 01 - local device ID : 03
etc.	

Connection data (PBC) SDB

The connection data (PBC) SDB contains information on the PBC connections of a TIM module. This is illustrated below based on an example.

Table 6- 16 Example of the information of a connection data (PBC) SDB (type 700)

```
STEP 7 connection   - Connection setup: fixed configured, static
                    - Connection type: active connection setup
                    - Operating Mode: send no operating mode messages
                    - Connection ID: 1
                    - local device ID : 01
                    - local TSAP ID: 11 04
                    - remote Station addr.: 02
                    - remote TSAP ID: 11 04
```

Consistent data SDB

The consistent data SDB is used to check consistency of the SDBs. Based on these SDBs, the TIM can check the consistency of the SDBs generated for it.

Table 6- 17 Example of the information of a consistency SDB (type 3118)

SDB no.	Type	Time stamp
1000	2	07/14/05 17:11:11
1001	3	07/14/05 17:11:11
1002	5	07/14/05 17:11:11
1003	1	07/14/05 17:11:11
1004	6	07/14/05 17:11:11
1005	7	07/14/05 17:11:11
1006	8	07/14/05 17:11:11
1008	1024	07/14/05 17:11:11

6.4 Service functions

6.4.1 Download SDB

Description of the functions

The *Download SDB* function downloads the system data blocks of a TIM module from the program directory of the SIMATIC Manager to the module. This is the same procedure as the *Download to Module* function in the SIMATIC Manager.

To activate newly downloaded SDBs on a TIM module, the TIM must be restarted.

Note

When a TIM module is restarted after downloading new SDBs, the connections between the TIM and other SINAUT partners (SINAUT connections) is terminated. This leads to error messages on the partners of the TIM module. In the case of a node TIM, the connections to lower-level stations are also reported as being disrupted.

With a node TIM, during the restart data messages stored on the TIM may also be lost. This can be relevant particularly in dial-up networks.

When downloading SDBs to TIM modules, you should therefore note the following points:

- Before you transfer the SDBs, you should give the TIM the opportunity of transferring any messages stored on it.
- After restarting the TIM, the SINAUT connections are established again automatically, the connection between the PG and the TIM must, however, be activated by the user on the PG if it is required.

Operator activities

1. Select a TIM module in the *SINAUT subscriber list* of the open project.
2. Start the function by selecting the *SINAUT / Download SDB* menu. The *Open* dialog is displayed.
   Follow the instructions in the subsequent dialogs. When necessary, you can cancel the procedure in these dialogs.
3. After downloading SDBs, the *Open* message dialog asks you when you want to start the module again. To activate the downloaded SDBs, you must restart the module.
4. Click on *Yes* to restart module. A message is displayed indicating that the SDBs were successfully loaded.
5. Confirm this message by clicking on *OK*.

6.4.2 Firmware update

Description of the functions

The *Firmware Update* function allows you to load a new firmware version on a TIM module. The function is supported on TIM modules that have the RMOS for TIM-ST7 operating system as of version 2.04.

Note

You can read out the version of the operating system of a TIM using the *Module Information* function / *General* tab.

To use this function, the firmware must have been installed on the computer using the setup. If the firmware is not installed on the PG or is incomplete, a message is displayed.

By clicking the *Update details* button in the *Firmware Update* dialog, you open the *Update details* dialog that displays the firmware version installed on the TIM module and located on your computer.

After the download, the module is automatically reset to activate the new firmware.

The parameter assignment of the module is not affected by the firmware update.

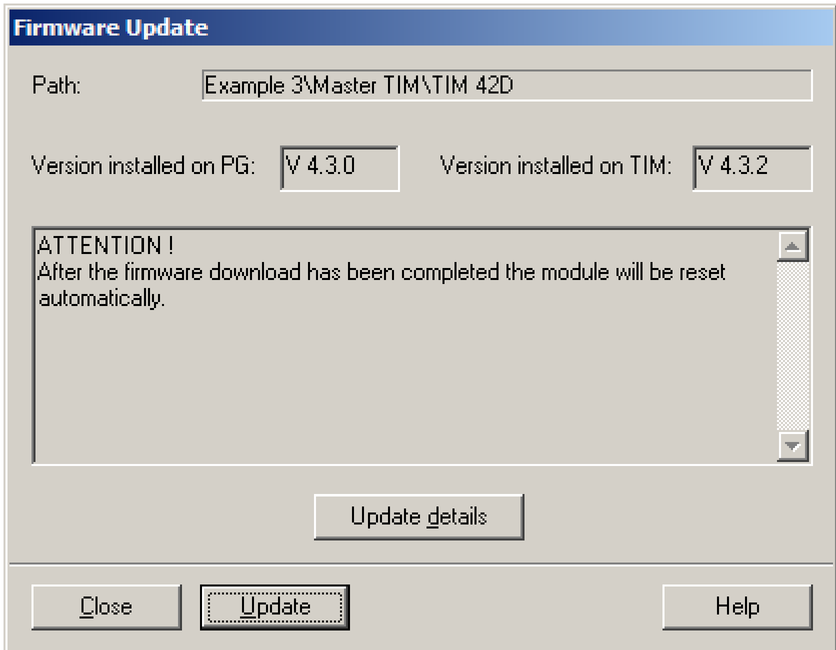


Figure 6-33 *Firmware Update* dialog

Note

Downloading the firmware to the module can take several minutes.

Operator activities

Note

Make sure that you select the correct TIM module in your project. The station and module name must match and the configured module must be of the same type as the module to which you are downloading.
If this is not the case, a dialog will inform you of this at regular intervals during the update. The display of this message interrupts the update until the dialog is acknowledged by clicking on *OK*.

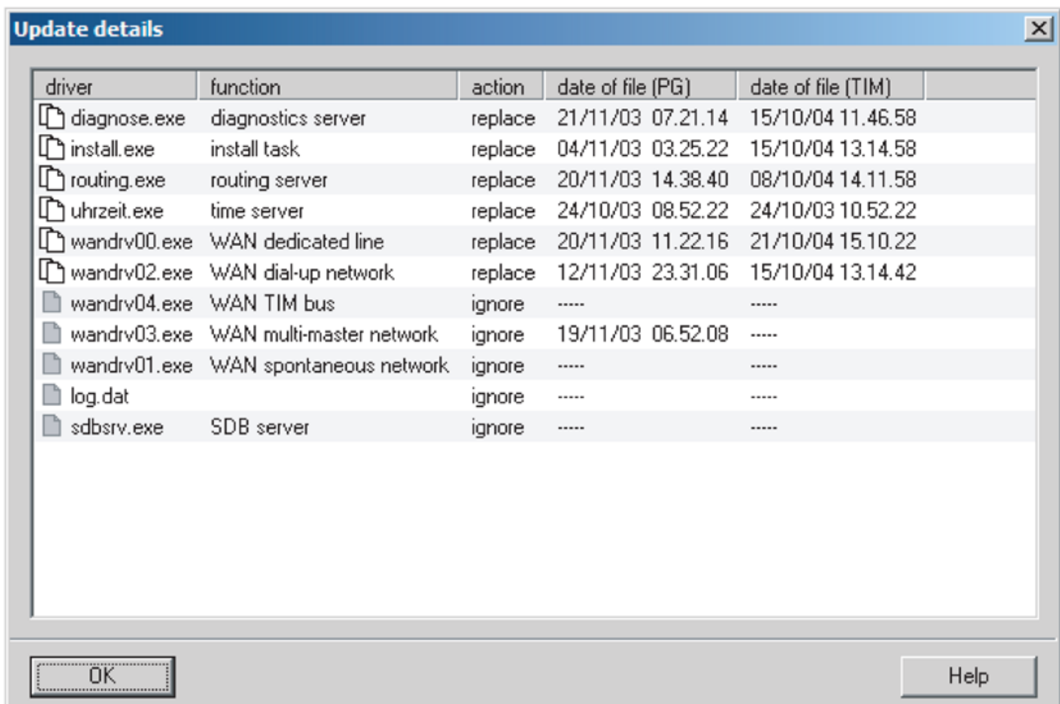
1. Select the relevant TIM in the *SINAUT subscriber list* of the open project.
2. Start the function by selecting the *SINAUT / Firmware Update* menu. The *Firmware Update* dialog opens.
3. Click on the *Update details* button if you require a detailed information on the firmware update. The *Update details* dialog opens.
4. Click on the *Update* button in the *Firmware Update* dialog to start the firmware update. The following dialog *Loading* informs you of the current progress of the update.
5. On completion of the firmware update, a dialog appears with a message to this effect. Confirm the message with *OK*.

Update details dialog

The *Update details* dialog that can be opened from the *Firmware update* dialog shows detailed information about the process of the firmware update. For every firmware component, the function and the version are specified on the TIM module and on the PG and it is indicated whether the relevant component will be copied, replaced, ignored or deleted during the firmware update.

The various actions mean:

- **copy**: The file will be copied from the PG to the TIM module.
- **replace**: The file on the TIM module will be replaced by the file on the PG.
- **ignore**: The file will not be taken into account by the firmware update..
- **delete**: There is no newer version for the file. The existing file is no longer required and is deleted during the firmware update.



driver	function	action	date of file (PG)	date of file (TIM)
diagnose.exe	diagnostics server	replace	21/11/03 07.21.14	15/10/04 11.46.58
install.exe	install task	replace	04/11/03 03.25.22	15/10/04 13.14.58
routing.exe	routing server	replace	20/11/03 14.38.40	08/10/04 14.11.58
uhrzeit.exe	time server	replace	24/10/03 08.52.22	24/10/03 10.52.22
wandrv00.exe	WAN dedicated line	replace	20/11/03 11.22.16	21/10/04 15.10.22
wandrv02.exe	WAN dial-up network	replace	12/11/03 23.31.06	15/10/04 13.14.42
wandrv04.exe	WAN TIM bus	ignore	-----	-----
wandrv03.exe	WAN multi-master network	ignore	19/11/03 06.52.08	-----
wandrv01.exe	WAN spontaneous network	ignore	-----	-----
log.dat		ignore	-----	-----
sdbsrv.exe	SDB server	ignore	-----	-----

Figure 6-34 *Update details* dialog of the *firmware update* function

6.4.3 Repair

Description of the functions

The *Repair* function allows you to restore TIM modules with a defective flash disk. The *Firmware Update* function is available for loading firmware on a functioning TIM module.

The *Repair* function runs a completely new installation of the firmware on a TIM module.

Note

The *Repair* function should not be used without consulting the hotline.

The *Repair* function is supported on TIM modules that have the RMOS for TIM-ST7 operating system as of version 2.04.

Note

You can read out the version of the operating system of a TIM using the *Module Information* function / *General* tab.

To use this function, the firmware must have been installed on the computer using the setup. If the firmware is not installed on the PG or is incomplete, a message is displayed.

By clicking the *Installation details* button, you open the *Update details* dialog that displays the firmware version installed on the TIM module and the version on your computer.

After the download, the module is automatically reset to activate the new firmware.

Steps in repairing

Repairing involves the following steps:

- The flash disk of the TIM module is formatted. After formatting, the TIM module runs a reset.
- This is followed by a default startup.
- Once the wait time for the default startup has elapsed, the firmware version installed on the computer is downloaded to the TIM.
- Following this, the system data blocks are transferred to the TIM module.
- The module is then reset and resumes operation with its full functionality.

Note

While the firmware is being reinstalled, the TIM module has MPI address 3.

Make sure that this address is free on the MPI bus to which the TIM module is connected.

Operator activities

Note

Make sure that you select the correct TIM module in your project. The station and module name must match and the configured module must be of the same type as the module to which you are downloading.
If this is not the case, a dialog will inform you of this at regular intervals during the update. The display of this message interrupts the update until the dialog is acknowledged by clicking on *OK*.

1. Make sure that MPI address 3 is either free or is occupied by the module on which you want to install.
2. Select the TIM you want to repair in the *SINAUT subscriber list* of the open project.
3. Start the function by selecting the *SINAUT / Repair* menu and then *Complete reinstallation* in the context menu. The *Firmware Install* dialog opens. The version installed on the PG is displayed. If the firmware on the PG is incomplete, a message to this effect is also displayed.
4. Click on the *Update details* button if you require a detailed information on the firmware update. The *Installation details* dialog opens.
5. Click on the *Start Installation* button in the *Firmware Install* dialog to start the repair. The following dialog *Download* informs you of the current progress of the procedure.
6. On completion of the repair, a dialog opens with a message to this effect. Confirm the message with *OK*.

Installation details dialog

The *Installation details* dialog that can be opened from the *Firmware Install* dialog displays detailed information on the repair. For each firmware component, the function and version on the TIM module and on the PG are shown and you can also see whether the component will be copied, replaced, ignored or deleted during the repair.

The various actions have the following significance:

- **copy**: The file is copied from the PG to the TIM module.
- **replace**: The file on the TIM module is replaced by the file on the PG.
- **ignore**: The file is not affected by the repair.
- **delete**: There is no newer version for the file. The existing file is no longer required and is deleted during the repair.

6.5 Message protocol diagnostics

Introduction

To read out messages, SINAUT ST7 provides you with the option of recording messages transferred in the CPU, TIM and ST7cc/ST7sc PC components; in other words, to archive them in protocols.

The following protocol types are distinguished:

- Testcopy DB:
  This is used to record messages on a CPU module.
- ST7cc/ST7sc protocol:
  This is used to record messages in SINAUT ST7cc or SINAUT ST7sc.
- TIM message protocol:
  This is used to record messages received and sent by a TIM module.

The message protocols are displayed in a message list in the form of a table.

Note

The functions of *message protocol diagnostics* are used only to analyze message protocols that have already been saved.
Message recording is activated

- for the Testcopy DB by setting up DB99 in the SIMATIC Manager
- for the SINAUT ST7cc/sc protocols by setting up the Textcopy block
- for the TIM message logging in the *TIM Message Monitor* function of the SINAUT diagnostics and service tool

---

6.5.1 Testcopy DB

Description of the functions

The recording of the message traffic on the CPU is made possible by the TestCopyData data block. With the aid of the FC Testcopy function, you can filter out certain message types that can be copied from the send or receive buffer of the CPU for further evaluation in a TestCopyData data block. The default for the TestCopyData data block is DB99.

When setting the the data block, in the control field of the TestCopyData DB, you set filters for certain message types, subscribers and objects and make further settings. All send and receive messages are stored in the TestCopyData DB in chronological order. For a detailed description of the functions and setting up the TestCopyData DB, refer to the chapter *SINAUT TD7 software package* of the SINAUT ST7 system manual.

With the aid of the *Testcopy DB* function of the SINAUT Diagnostics and Service tool, a TestCopyData data block is opened and the recorded messages displayed as a message list for further analysis.

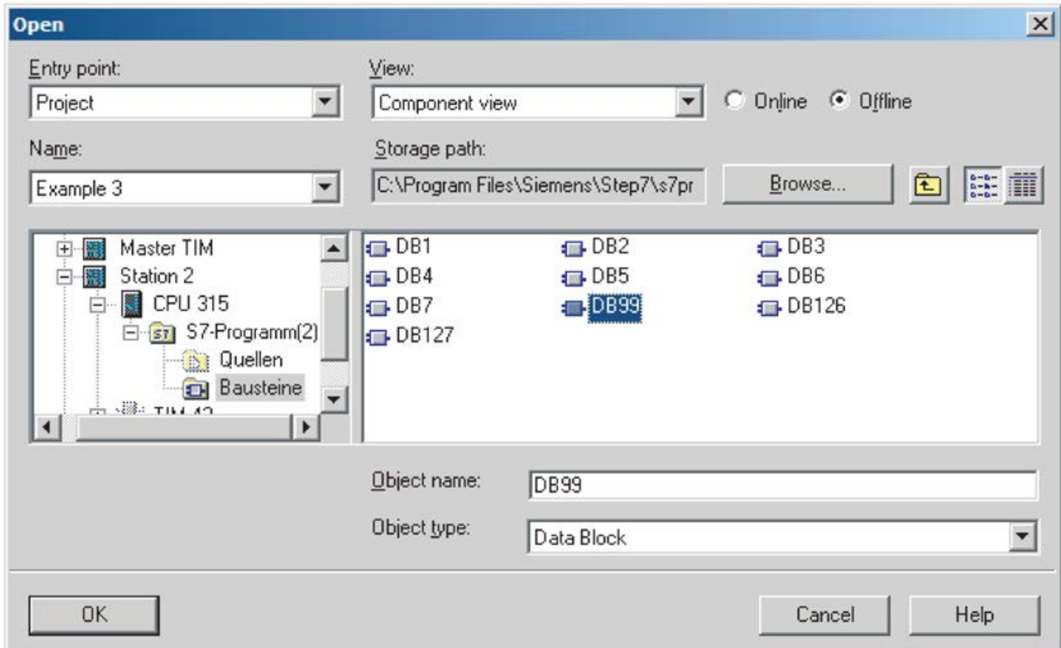


Figure 6-35 *Open* dialog of the *Testcopy DB* function

Operator input

1. Open the *Open* dialog by selecting the *Project / Testcopy-DB* menu.
2. In the *Open* dialog, select the *Online* option at the top right if you want to access the DB TestCopyData of the CPU module directly. This is only possible if there is a functioning connection between the PG and the relevant CPU module.
3. In the selection list *Entry point* specify a *Project* or *Accessible Nodes* as the type of project.
4. Select the required project in the selection list *Name* or click the *Browse* button if the project is not shown in the *Name* selection list.
   The *Browse* dialog opens with the file tree of the PG/PC. In the input box *Find in Directory*, enter the directory path and confirm the button with the *Start Search* button.
   Select the project you are looking for on the right in the *User project*s tab and click the *OK* button. You return to the *Open* dialog.
5. Open the list of project stations on the left in the folder list by double-clicking on the project name and further select:
   - the required station
   - its CP module
   - the S7 program and
   - the block folder.
   In the object list on the right, the individual objects of the selected block folder are displayed. In the basic setting, only data blocks are displayed because below in the selection list *Object type*, the object type *Data block* is the default.

6. On the right in the object list, select the DB TestCopyData (DB99), it is entered in the input box *Object name* below.
7. Then confirm with *OK*, the message protocol *Testcopy-DB* opens. If the Testcopy DB does not contain any messages, a message to this effect is displayed.

Working with the open message protocol *Testcopy DB* corresponds to working with the TIM message protocol.

6.5.2 ST7cc / ST7sc protocol

Description of the functions

The recording of the message traffic between SINAUT subscribers and an ST7cc or ST7sc control center is possible in ST7cc or ST7sc using the trace. The function of the trace is explained in the *SINAUT ST7cc Control Center Manual*. This contains precise information on starting the trace functions, displaying messages in the output window, activating the trace output files and other functionalities.

Operator input

1. Open the ST7cc or ST7sc protocol by selecting the *Project / ST7cc/ST7sc protocol* menu. The *Open* dialog is displayed.
2. In the file tree, select the directory and the *7DS* file of the required TIMST7cc or ST7sc message protocol and confirm with the *Open*
 button. The message protocol opens in a separate window.

Working with the open ST7cc or ST7sc message protocol corresponds to working with the TIM message protocol.

6.5.3 TIM message protocol

Description of the functions

The recording of the TIM messages is started with the *TIM Message Monitor* of the SINAUT Diagnostics and Service tool. The messages received and sent by the routing server of the TIM are recorded.

The *TIM message protocol* function is used only to open a TIM message protocol for subsequent evaluation.

Operator activities

1. Open the TIM message protocol by selecting the *Project / TIM Message Protocol* menu. The *Open* dialog is displayed.
2. In the file tree, select the directory and the *7DT* file of the required TIM message protocol and confirm with the *Open* button. The TIM message protocol is opened in a separate window.

Note

You activate and deactivate recording of the TIM message protocol and specify the name and storage location of the message protocol file using the *TIM Message Monitor* function of the Diagnostics and Service tool.

6.5.4 Diagnostics of the TIM message protocol

The functions and operator input of the TIM message protocol are largely - with a few differences in the format and diagnostics data -- identical to those of the other message protocol types *Testcopy DB* and *ST7cc/ST7sc protocol*. The description can therefore also be applied to the other message protocol types.

Structure of the TIM message protocol

The upper part of the *TIM message protocol* dialog summarizes the following information:
- Total number of messages,
- Number of messages shown,
- Source and path of the protocol file

Below this there is the **list of TIM messages** that contains nine columns in the basic setting with the following information:
- A symbol for incoming and outgoing messages
- Message number
- Source: Number of the open message protocols in ascending order
- msec: Logging time for the DB TestCopyData and ST7cc/ST7sc protocols
- Block: Number of the message block with TIM message protocols
- Subscriber number of the message source and destination
- ST1 message number (only for SINAUT ST 1 messages)
- Object number of the message source and destination
- Index no.: Address parameters for net data in data messages
- Org. Information

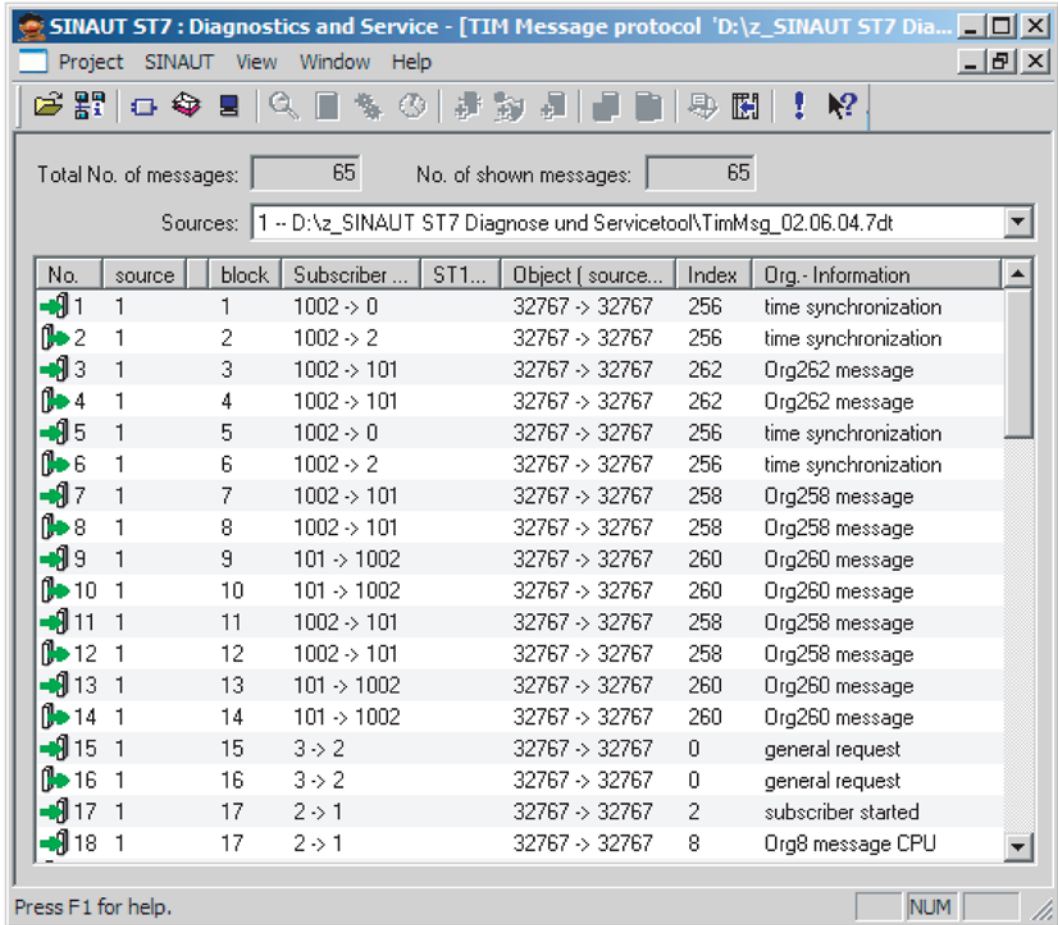


Figure 6-36 The *TIM Message Protocol* dialog

Functions of the TIM message protocol

Further functions are available with the right mouse button in a shortcut menu. These can be grouped as follows:

- Presentation of the messages
- Details (of the message content)
- Statistics
- Filter functions
- Exporting protocol files

Working with the message list

After opening the *TIM message protocol* with the *Project / TIM Message Protocol* menu, the following options are available in the open dialog:

1. If you click on the header of any column, the message list will be sorted according to this criterion instead of the consecutive number.
2. Right-click (cursor within the protocol window) to activate further functions. A context menu opens with other functions.
3. Select the required function with the left mouse button in the context menu. Each function opens a dialog.

Presentation of the messages

Add new columns

With this function, additional freely selectable columns with additional message information can be shown in the TIM message list.

After calling the function *Add new columns* the dialog *Add column* opens in which you can select the properties by mouse click. The selected message properties are shown in the message list as additional columns.

Delete additional columns

When you click on this function you delete all previously inserted extra columns. No further dialog is displayed.

Details

The *Details* function provides you with detailed information on the content of the individual messages.

To open the *Details* dialog, you must first select a message. As an alternative to using the right mouse button, the *Details* dialog can also be opened by double-clicking on a message in the list.

In the upper part of the *Details* dialog, you can see the path of the protocol file and five tabs containing further information.

To page to other messages within the *Details* dialog, click on the *<<* or *>>* button. In each tab, the dialog view switches to the previous or next message.

- The *Message Header* tab displays a table containing three columns with the following data from the message header of the selected message:
  – Variable name or short name of the message
  – Value of the individual variables
  – Variable name

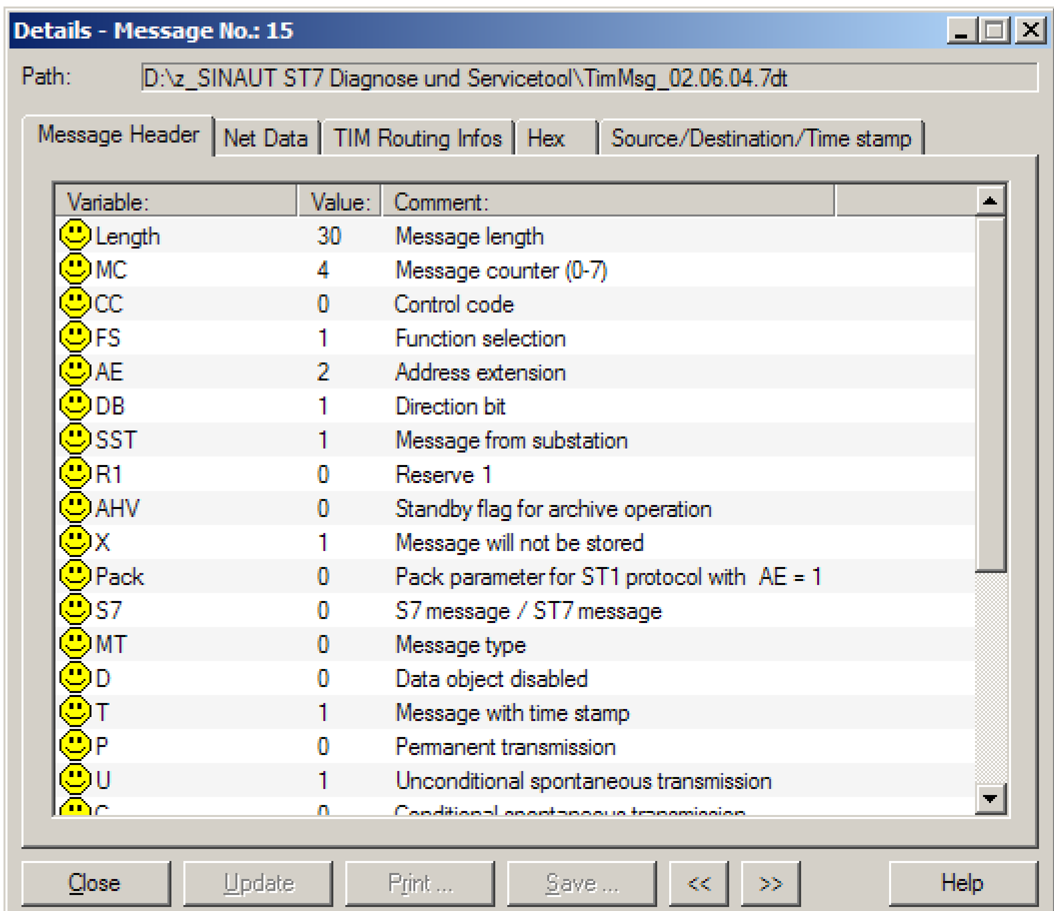


Figure 6-37 *Details* dialog, *Message Header* tab

- The *Net Data* tab shows the net data of the message.
  With the message type 0 and 1 (organizational messages), the data is displayed as plain text.
  with message type 2 and 3 (data messages), the values are displayed. The mode of the display can be changed with the *Display mode* list box using the mouse. The options are:
  - ST7 analog value
  - ST1 analog value
  ST7/ST1 counted value
  - ST7/ST1 message

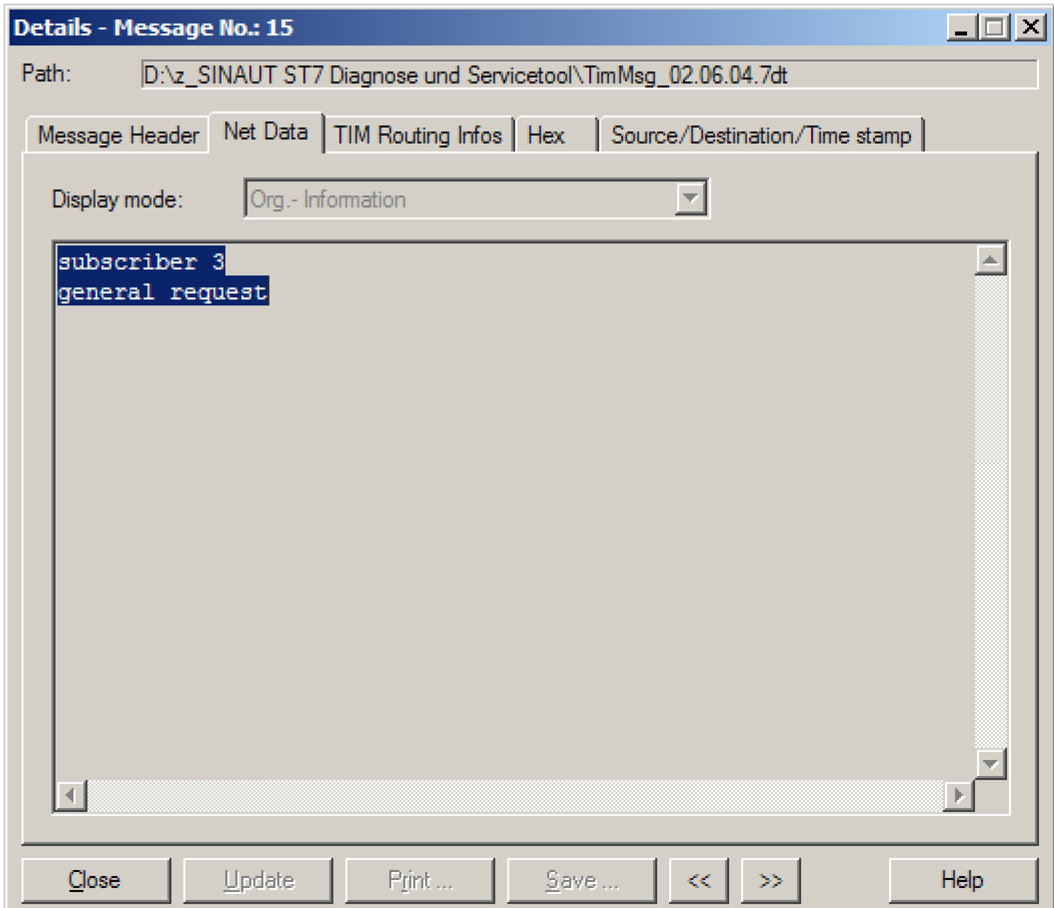


Figure 6-38 *Details* dialog, *Net Data* tab

- The *TIM Routing Infos* tab shows the following:
  - At the top in the *Message* area:
    - the internal task ID
    - the complete message length [bytes]
  - Below in the *Address infos* area:
    - the device ID as a number and in plain text (for example MPI bus)
    - the CN ID
    - the station address

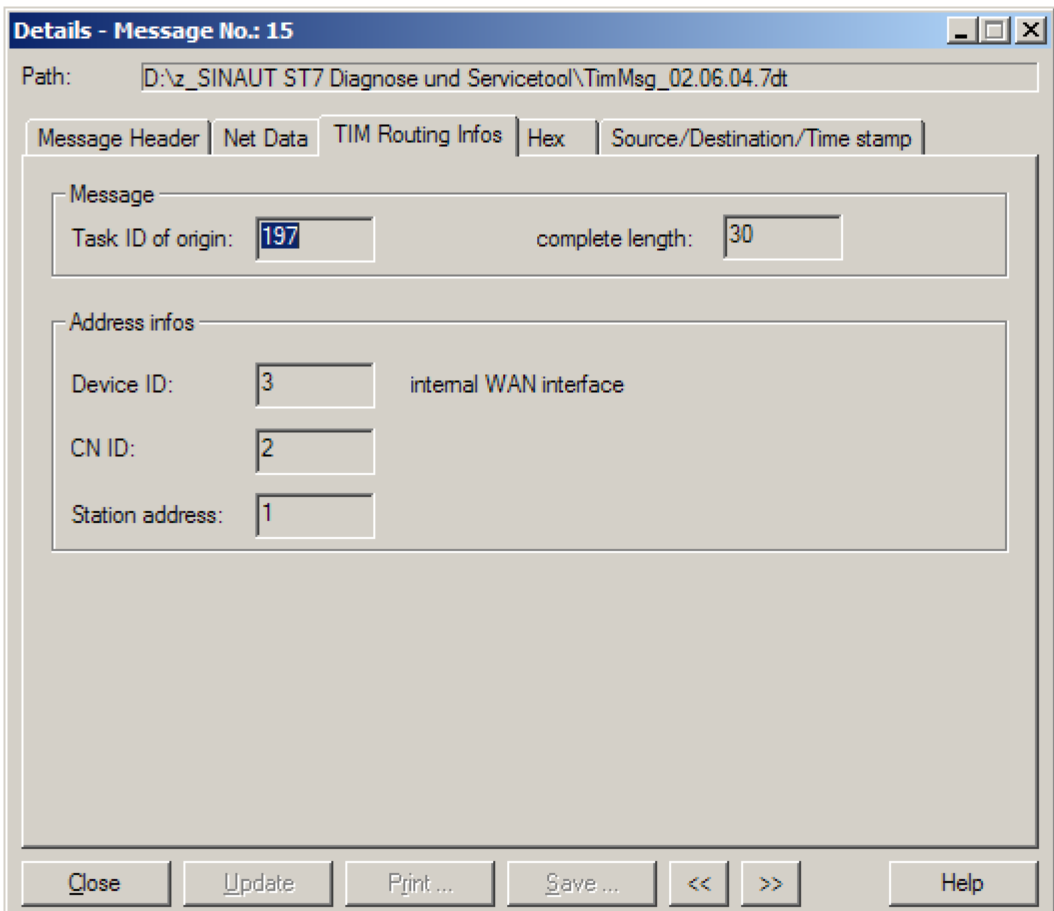


Figure 6-39 *Details* dialog, *TIM Routing Infos* tab

- The *Hex* tab shows the following in hexadecimal format:
  - In the *Complete buffer* area, the content of the entire message
  - In the *net data* box, only the net data of the message

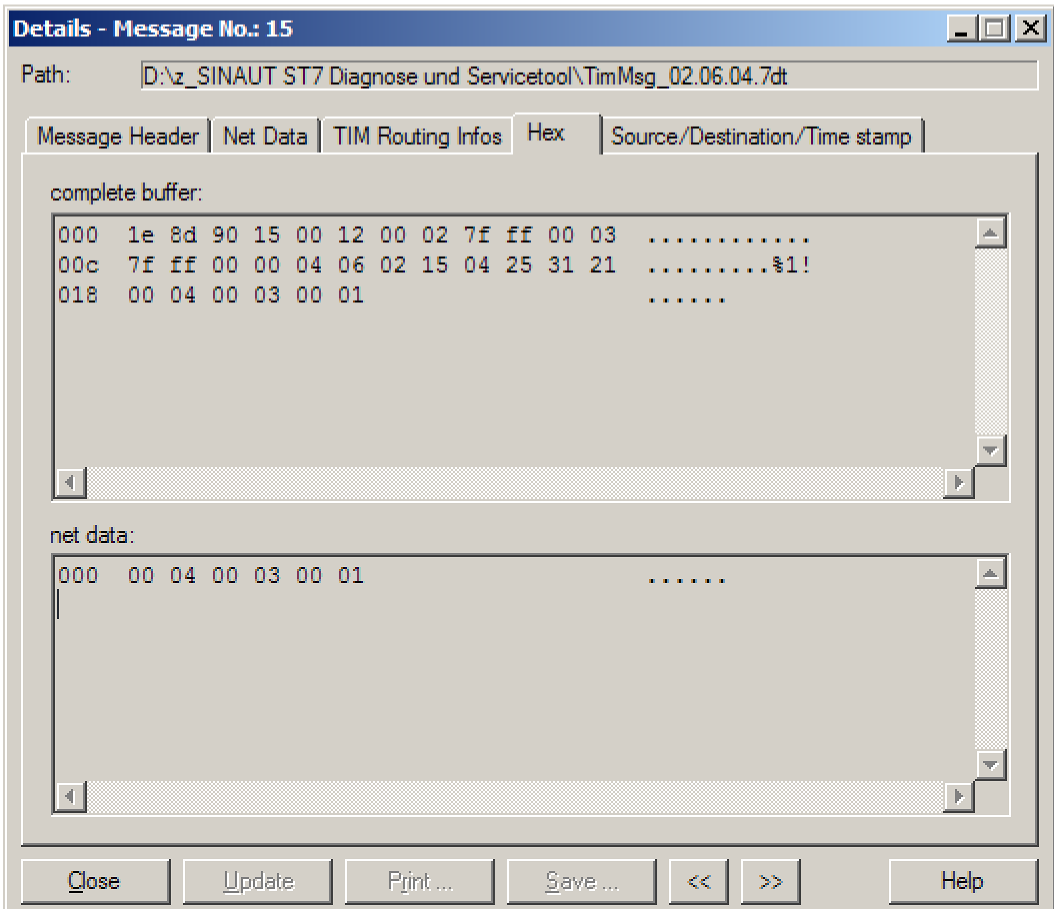


Figure 6-40 *Details* dialog, *Hex* tab

- The *Source/Destination/Time stamp* tab provides information in its three boxes on the source, and destination and time stamp of the message. The fields provide the following information:
  - *Source*: Information on the subscriber number, name, type name, station name, object number and index number
  - *Destination*: Information on the subscriber number, name, type name, station name and object number.
  - *Time stamp*: Information on the date, time, status, status info.

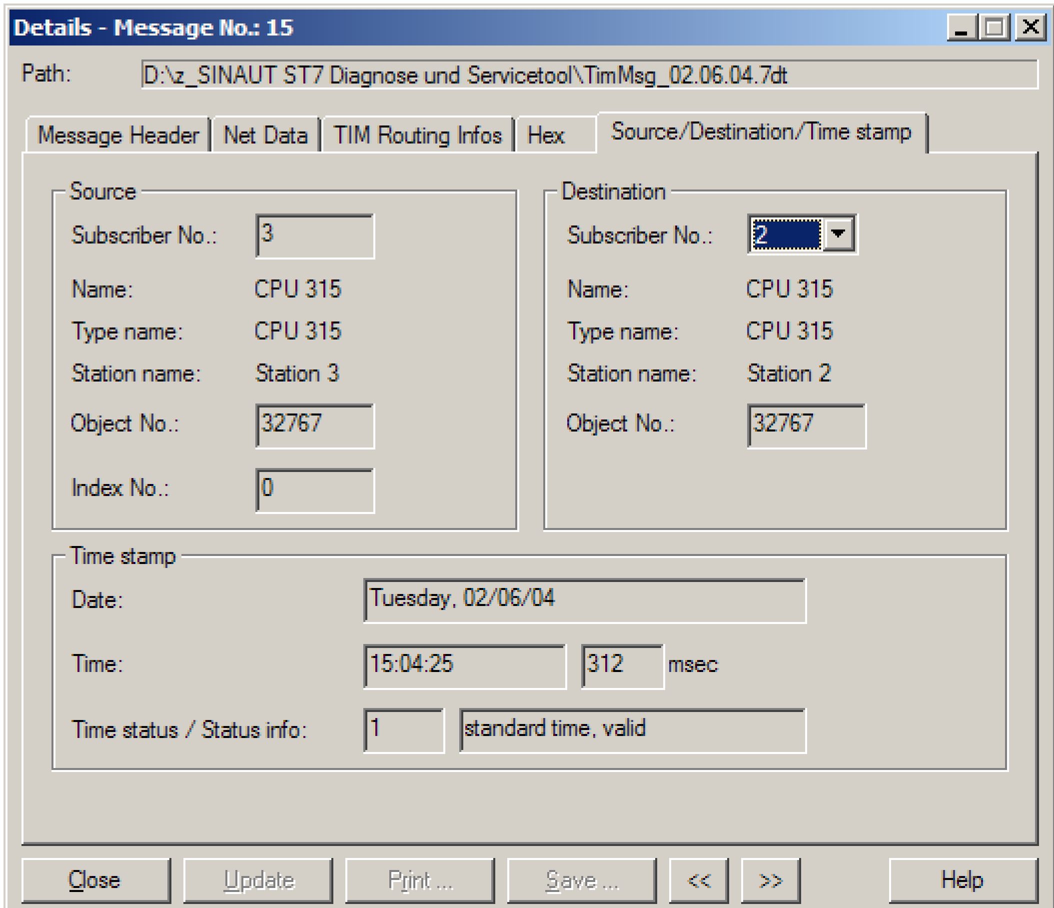


Figure 6-41 *Details* dialog, *Source/Destination/Time stamp* tab

Statistics

The *Statistics* function provides a statistical evaluation of the entire message protocol in terms of numbers, types, and throughput of messages of the subscribers involved sorted according to

- All messages, data messages and organizational messages and according to
- Requested and spontaneous messages.

With the aid of the statistics, you can, for example recognize particular concentrations of certain message types with individual subscribers allowing you to decide whether normal or acceptable message traffic is possible in the particular installation.

The *Statistics* dialog displays the statistical data of the TIM message protocol in three tabs. In each tab, the period of message logging is displayed at the top. The three tabs of the dialog list the messages as tables sorted according to the following:

- The *Counters* tab provides information on the total number of messages and the number of different sent and received message types.
- The *Message flow* tab provides information on the amount of message traffic per minute. It shows the total number of messages and the number of different sent and received message types per minute.
- The *Subscriber* tab displays a table with the number of different message types per subscriber. With the list box at the top right, you can sort the messages according to subscriber number or message type.

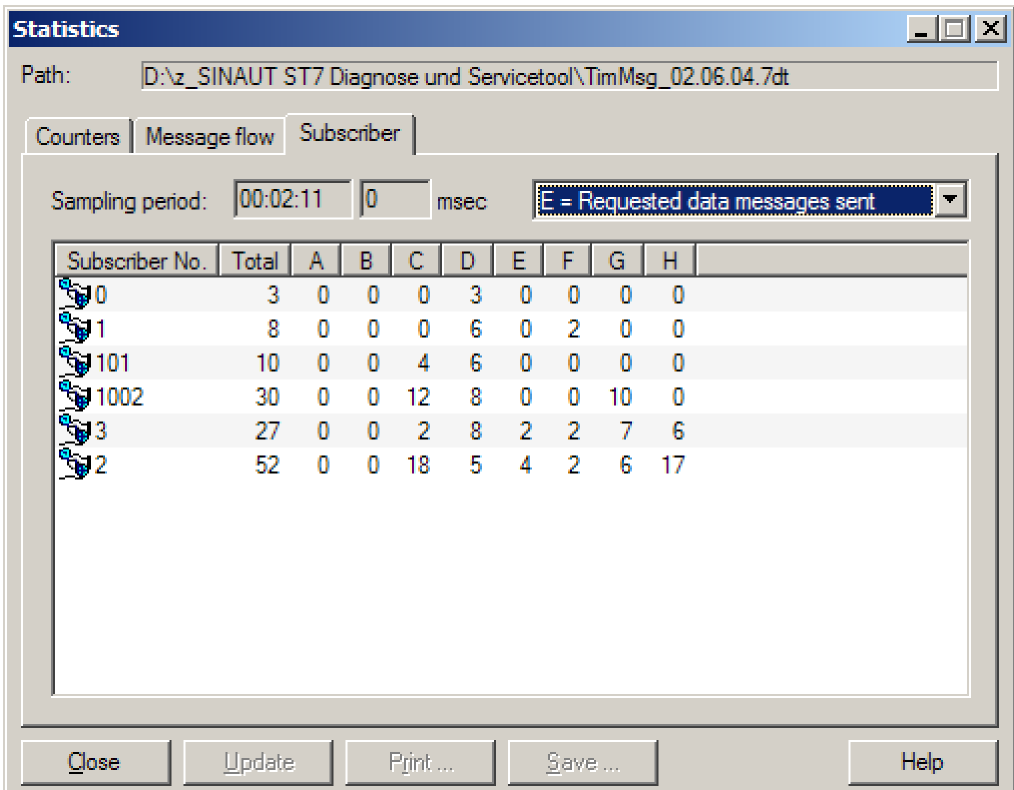


Figure 6-42 *Statistics* dialog, *Subscriber* tab

The *Statistics* function is only available for individual message protocols (sources) and is not supported, for example, a Testcopy DB or ST7cc protocol was inserted in a TIM message protocol.

Filter functions

Delete list:

When you select the function *Delete list* with the mouse, the displayed TIM messages are *reversibly deleted* from the list view. The messages can be inserted in the list view again with the two functions *Selection* and *Show all messages*.

Selection

The *Selection* function is used to select certain message types that should be displayed again after deleting the messages from the list view. For example, messages can be selected according to individual subscribers, directions, message header entries etc. With this function the number of messages can be reduced so that the list view is clearer.

Show all messages

With the function *Show all messages*, all messages of the TIM message protocol are inserted in the list view again that were previously completely deleted from the list view with the *Delete list* function or selectively with the *Selection* function.

Exporting protocol files

Save current list

The *Save current list* function is used to save the currently open list with all data of the current list view in a CSV file. The CSV file can then be read in MS EXCEL.

When the function is activated an input box opens initially in which you can enter a comment on the protocol to be saved as free text. This comment is later included in the upper lines of the CSV file to be saved. Following this, the directory and file name of the CSV file are specified.

Export complete list

The *Export complete list* function is used to save the complete currently open list with *all* available data in an MS EXCEL-compatible CSV file.

When you activate the *Export complete list function*, initially an input box opens in which you can, when necessary, enter a comment on the protocol to be saved as free text. This comment is later included in the upper lines of the CSV file to be saved. Following this, the directory and file name of the CSV file are specified.

Note

Open the CSV file that was created with the *Save current list* or *Export complete list* functions in the MS EXCEL program by selecting the menu *File / Open* so that the data from the individual columns of the list view of the TIM message protocol are shown in separate columns in EXCEL.
If the CSV file is opened from the Explorer by double-clicking, the data will be shown in a single column although with delimiters

---

6.6 Messages in the diagnostic buffer of the TIM

Introduction

In much the same way as on and S7 CPU, a diagnostic buffer is also maintained on the TIM. The TIM stores its specific diagnostic messages in this buffer. The diagnostic messages of the TIM module are read out in the same way as those of a CPU.

Note

If there is no text file with the diagnostic texts of the TIM events on the PG with which the diagnostic buffer is read out, the events are displayed in hexadecimal format.

6.6.1 Diagnostic messages of the TIM

Classification of TIM messages

The TIM uses a reserved area within the event class F, namely Fx60, known as the event ID. All TIM diagnostics messages start with Fx60 in the hexadecimal representation, where x a digit is the placeholder for an identifier that allows a global classification of the message:

Table 6- 18 Global classification of the TIM diagnostics messages

ID x	Resulting event ID	Classification
2	F260	Event message, exiting state
3	F360	Event message, entering state
4	F460	Event message, internal error, exiting state
5	F560	Event message, internal error, entering state
8	F860	Event message, external error, exiting state
9	F960	Event message, external error, entering state

The event ID Fx60 is followed by the actual message, the detailed event. This occupies the numeric range from 0000h to 0FFFh. Depending on the message, there may be additional information under *Additional info 1/2/3* or *Additional info 4/5*.

The diagnostics messages of the TIM in hexadecimal and plain text format

The following table lists all the TIM diagnostics messages in ascending order of the detailed event in hexadecimal format.

To complete the picture, the corresponding event ID is also listed. The event ID is used only to classify the message and has no relevance for the order.

Table 6- 19 Diagnostics messages of the TIM

Event ID (hex)	Detailed event (hex)	Meaning
General messages		
F560	0001	Entering state: Heap memory overflow.
F460		Exiting state: Heap memory overflow eliminated.
Init task messages		
F360	0060	ST7 installation started.
F560	0061	Error creating the message queue of the INIT task.
F360	0062	Archive was created.
F560	0063	Error installing the DNA interrupt service routine.
F560	0064	Error installing the interrupt service routine for the external interface.
F560	0065	Error installing the interrupt service routine for the internal interface.
F560	0066	Error installing the TIMER interrupt service routine.
F560	0067	Structure of the interface administration could not be entered in the catalog.
F560	0068	Dongle flag not found in catalog.
F560	0069	EXE loader not found in catalog.
F360	006A	Internal / external WAN interface: Driver not released.
F560	006B	Internal / external WAN interface: Error sending the load job for a driver.
F560	006C	Internal / external WAN interface: Error loading the basic task of a driver.
F560	006D	Internal / external WAN interface: Error creating the basic task of a driver.
F560	006E	Internal / external WAN interface: Error receiving the task ID of a driver.
F560	006F	Internal / external WAN interface: Error starting the basic task of a driver.
F560	0070	WAN SDB could not be opened.
F560	0071	WAN SDB could not be found.
F560	0072	Error sending the load job for the clock driver.
F560	0073	Error loading the basic task of the clock driver.
F560	0074	Error creating the basic task of the clock driver.
F560	0075	Error receiving the task ID of the clock driver.
F560	0076	Error starting the basic task of the clock driver.
F560	0077	WAN SDB does not start with the *TIM* parameter block.
F560	0078	Not enough memory available.
F560	0079	Error creating the global message memory.
F560	007A	Error sending the load job for the routing program.
F560	007B	Error loading the basic task of the routing program.
F560	007C	Error creating the basic task of the routing program.
F560	007D	Error receiving the task ID of the routing program.
F560	007E	Error creating the main task of the routing program.
F560	007F	Routing tables were not created within the specified time.

Event ID (hex)	Detailed event (hex)	Meaning
F560	0080	WAN driver was not installed within the specified time.
F560	0081	Unknown parameter block in WAN SDB.
F360	0082	Startup of module completed.
F560	0083	Flag group unknown.
F360	0084	RAM drive was created.
F560	0085	Error creating RAM drive.
F560	0086	Error in memory analysis in HEAP.
F560	0087	Error installing the clock driver. No message received at end of installation.
F360	0088	Message of the message and HEAP memory initialized on the TIM.
F360	0089	Routing SDB could not be found.
F560	008A	Routing SDB could not be opened.
F560	008B	Routing SDB starts with incorrect subnet block ID.
F560	008C	No memory available for routing function.
F560	008D	Incorrect block ID detected in routing SDB.
F560	008E	Error sending a job to the LAN task.
F560	008F	Installation of the routing function was aborted.
F560	0090	ST7 installation started.
F360	0091	Wrong TIM firmware loaded.
F560	0092	Installation error in the Start manager
F560	0100	Module startup: Installation error
LAN messages		
F560	0101	Installation of the AMPLUS-L emulation aborted.
F560	0103	LAN communication: Error sending a message.
F560	0104	LAN communication: Error receiving a message or ID unknown.
F360	0105	LAN communication: Unknown job.
F360	0106	LAN communication: Connection could not be entered in routing table.
F360	0107	LAN communication: Connection could not be deleted in routing table.
F560	0108	LAN communication: Error in a connection SDB.
F360	0109	LAN communication: No resources available for connection.
F560	010A	LAN communication: PBC connection could not be established. Reference number unknown.
F560	010B	LAN communication: PBC connection could not be established. Reference number unknown.
F360	010C	Entering state: LAN communication: Connection down.
F260		Exiting state: LAN communication: Connection OK.
F560	010D	Entering state: LAN communication: Threat of send queue overflow for a connection.
F460		Exiting state: LAN communication: Threatening send queue overflow for a connection eliminated.
F560	010E	Entering state: LAN communication: Send queue overflow for a connection.
F460		Exiting state: LAN communication: Send queue overflow for a connection eliminated.
F560	010F	Entering state: Disruption of MPI/party line interface (SPC/2) detected.
F460		Exiting state: Disruption of MPI/party line interface (SPC/2) eliminated.
F560	0110	LAN communication: Error reading the LAN SDB.

Event ID (hex)	Detailed event (hex)	Meaning
F560	0111	LAN communication: Error occurred during PBC send.
F560	0112	LAN communication: Error in SDB0 – bad MPI parameter.
F560	0113	LAN communication: A negative acknowledgment was sent. Byte 6 of the message codes the error class (ERRCLS) and byte 7 codes the error code (ERRCOD). For the meaning of ERRCLS and ERRCOD, see Meaning of ERRCLS and ERRCOD (Page 501).
F560	0114	No Ethernet SDB.
F560	0115	Ethernet SDB could not be opened.
F560	0116	Ethernet SDB with bad block ID received.
F560	0117	Error creating a socket.
F560	0118	Error linking a socket.
F560	0119	Invalid socket.
F560	011A	Error listening on a socket.
F560	011B	Ethernet port: RFC1006 has received a packet whose length exceeds the maximum.
F560	011C	Bad RFC1006 PDU header.
F560	011D	Undefined PDU received.
F560	011E	Bad TCP/IP packet.
F560	011F	Error setting a socket.
F560	0120	Error in ACCEPT socket.
F560	0121	TCP/IP connection termination by partner.
F560	0122	TCP/IP reception error.
F560	0123	TCP/IP send error.
F560	0124	TCP/IP connection number invalid.
F560	0125	Error receiving a CR-PDU.
F560	0126	Illegal access over TCP/IP.
F560	0127	Invalid PDU length.
F560	0128	KEEPALIVE expired.
F560	0129	Entering state: Connection information: Ethernet port problem.
F460		Exiting state: Connection information: Ethernet port ok.
F560	012A	Error in socket CONNECT job.
F560	012B	Error receiving a CC-PDU.
F560	012C	Unknown error code.
F560	012D	Maximum number of S7 connections exceeded.
WAN messages		
F560	0300	Internal / external WAN interface: Installation of the WAN driver was aborted.
F360	0301	Internal / external WAN interface: WAN driver is installed.
F360	0302	Internal / external WAN interface: Connection to a subscriber established (incoming call; subscriber number identified).
F560	0303	Bad organizational message from routing task.
F360	0304	Entering state: Internal / external WAN interface: Send buffer changed over to image.
F260		Exiting state: Internal / external WAN interface: Send buffer changed back from image.

Event ID (hex)	Detailed event (hex)	Meaning
F360	0305	Entering state: Internal / external WAN interface: Send buffer overflow occurred.
F260		Exiting state: Internal / external WAN interface: Send buffer overflow eliminated.
F960	0306	Bad message: max. number of destination subscribers exceeded.
F960	0307	Bad message: Unknown source subscriber number.
F960	0308	Bad message: S7 PDU not with AE ID = 2.
F960	0309	Entering state: Internal / external WAN interface: CTS disturbance occurred on modem.
F860		Exiting state: Internal / external WAN interface: CTS disturbance eliminated on modem.
F560	030A	Entering state: Internal / external WAN interface: USART error occurred.
F460		Exiting state: Internal / external WAN interface: USART error eliminated.
F360	030B	Entering state: Subscriber failed.
F260		Exiting state: Subscriber OK.
F360	030C	Entering state: Internal / external WAN interface: WAN driver disabled.
F260		Exiting state: Internal / external WAN interface: WAN driver enabled.
F360	030D	Entering state: Internal / external WAN interface: Subscriber call disabled.
F260		Exiting state: Internal / external WAN interface: Subscriber call enabled.
F360	030E	Entering state: Internal / external WAN interface: Subscriber call in subcycle.
F260		Exiting state: Internal / external WAN interface: Subscriber call in main cycle.
F360	030F	Entering state: Internal / external WAN interface: Permanent call to a subscriber enabled.
F260		Exiting state: Internal / external WAN interface: Permanent call to a subscriber ended.
F360	0310	Entering state: Internal / external WAN interface: Lack of resources on a subscriber.
F260		Exiting state: Internal / external WAN interface: Lack of resources on a subscriber eliminated.
F360	0311	Entering state: Internal / external WAN interface: Alternative path changeover on.
F260		Exiting state: Internal / external WAN interface: Alternative path changeover off.
F560	0312	Internal / external WAN interface: No message memory available for new image element.
F560	0313	Internal / external WAN interface: Image element too large for image memory.
F560	0314	Internal / external WAN interface: Set number of subscribers exceeded in image.
F560	0316	Internal / external WAN interface: 'List of Active Stations' (LAS) not available.
F560	0317	Internal / external WAN interface: Error receiving a message.
F560	0318	Internal / external WAN interface: Error enabling message memory.
F560	0319	Entering state: Internal / external WAN interface: Communication with AMPLUS-L task disrupted.
F460		Exiting state: Internal / external WAN interface: Communication with AMPLUS-L task OK.
F560	031A	Entering state: Internal / external WAN interface: Communication with clock driver disrupted.
F460		Exiting state: Internal / external WAN interface: Communication with clock driver OK.
F560	031B	Entering state: Internal / external WAN interface: Communication with routing task disrupted.
F460		Exiting state: Internal / external WAN interface: Communication with routing task OK.
F560	031C	Internal / external WAN interface: Modem command invalid.
F560	031D	Internal / external WAN interface: Invalid dialing string or bad call number transferred to modem when calling a subscriber.
F960	031E	Internal / external WAN interface: Incorrect handshake PDU received from a subscriber.
F360	031F	Internal / external WAN interface: Own telephone connection occupied.

Event ID (hex)	Detailed event (hex)	Meaning
F960	0320	Internal / external WAN interface: Modem not replying.
F960	0321	Internal / external WAN interface: Access to called subscriber not permitted.
F960	0322	Internal / external WAN interface: No answer tone received from modem of called subscriber.
F960	0323	Internal / external WAN interface: Called subscriber is not operational.
F960	0324	Internal / external WAN interface: Modem of called subscriber has no power.
F960	0325	Internal / external WAN interface: Telephone line is disrupted.
F360	0326	Internal / external WAN interface: Supervision time exceeded. Repetition starting.
F360	0327	Internal / external WAN interface: All attempts to dial a subscriber were executed. No connection was established.
F360	0328	Entering state: Internal / external WAN interface: Telephone number list with telephone number(s) of a subscriber deactivated.
F260		Exiting state: Internal / external WAN interface: Telephone number list with telephone number(s) of a subscriber activated.
F560	0329	Internal / external WAN interface: Telephone number list with telephone number(s) of a subscriber is invalid or disrupted.
F360	032A	Internal / external WAN interface: Telephone number of a subscriber temporarily disabled.
F360	032B	Internal / external WAN interface: Telephone number of a subscriber was changed.
F560	032C	Internal / external WAN interface: STA number not found in telephone number list.
F360	032D	Entering state: Internal / external WAN interface: Permanent connection established to a subscriber.
F260		Exiting state: Internal / external WAN interface: Permanent connection to a subscriber was terminated.
F360	032E	Entering state: Internal / external WAN interface: Permanent connection to a subscriber was registered.
F260		Exiting state: Internal / external WAN interface: Permanent connection to a subscriber was deregistered.
F560	032F	Internal / external WAN interface: Permanent connection to a subscriber was aborted.
F360	0330	Entering state: Internal / external WAN interface: Incoming call disabled.
F260		Exiting state: Internal / external WAN interface: Incoming call enabled.
F360	0331	Internal / external WAN interface: Establishing connection to a subscriber.
F360	0332	Internal / external WAN interface: Connection to a subscriber established (outgoing call).
F360	0333	Internal / external WAN interface: Connection to a subscriber established (incoming call; subscriber number not yet identified).
F360	0334	Internal / external WAN interface: Connection to a subscriber was terminated.
F560	0335	Internal / external WAN interface: Connection to a subscriber was aborted.
F360	0336	Internal / external WAN interface: Connection to a subscriber is already terminated.
F360	0337	Internal / external WAN interface: Supervision time exceeded. No repetition.
F360	0338	Internal / external WAN interface: Send buffer was deleted.
F360	0339	Internal / external WAN interface: Image memory and send buffer were deleted.
F360	033A	Internal / external WAN interface: No telephone number in modem memory.
F960	033B	Internal / external WAN interface: PDU received with unknown STA number.
F360	033C	Entering state: Internal / external WAN interface: Driver redundancy - memory management switched over.

Event ID (hex)	Detailed event (hex)	Meaning
F260		Exiting state: Internal / external WAN interface: Driver redundancy - memory management switched back.
F560	033D	Internal / external WAN interface: Incorrect service request to pager (SMS).
F360	033E	Internal / external WAN interface: No entries in the send queue.
F560	033F	Internal / external WAN interface: Unknown message type.
F560	0340	Internal / external WAN interface: Communication with WAN driver disrupted.
F360	0341	Internal / external WAN interface: Incoming call (RING).
F960	0342	Internal / external WAN interface: No connection with incoming call.
F960	0343	Internal / external WAN interface: Call or connection abort.
F960	0344	Internal / external WAN interface: No carrier frequency detected on partner.
F360	0345	Entering state: Internal / external WAN interface: Further dialing attempts were made in the background to a disturbed subscriber.
F260		Exiting state: Internal / external WAN interface: Subscriber is available again. Dialing attempts in the background will be stopped.
F960	0346	Internal / external WAN interface: Incorrect PIN number transferred to GSM module.
F960	0348	Internal / external WAN interface: Error occurred in GSM module.
F960	0349	Internal / external WAN interface: GSM module not responding or not available.
F960	034A	Internal / external WAN interface: SMS server of the TIM has received an unknown message.
F960	034C	Internal / external WAN interface: Short message (SMS) acknowledgment received from an unknown mobile subscriber.
F360	034E	Internal / external WAN interface: Incoming call detected. Incoming calls are disabled.
F360	034F	Internal / external WAN interface: Incoming call detected. DTR signal was activated.
F360	0350	Internal / external WAN interface: SMS server of the TIM was installed and started.
F360	0351	Internal / external WAN interface: SMS status, global status request/deletion.
F360	0352	Internal / external WAN interface: SMS status, single status request/deletion.
F360	0353	Internal / external WAN interface: Spontaneous SMS status message.
F960	0354	Internal / external WAN interface: Short message (SMS) acknowledgment incorrect. Format or ID no. unknown.
F560	0355	Internal / external WAN interface: No send buffer could be made available for sending a short message (SMS).
F960	0356	Internal / external WAN interface: The GSM module expects the PUC number.
F960	0357	Internal / external WAN interface: Telephone number of the SMS recipient could not be found.
F360	0358	Internal / external WAN interface: The GSM signal strength is xx dBm.
F360	0359	Subscribers cannot be blocked.
F360	035A	Internal / external WAN interface: GSM module detects wrong service ID.
F360	035B	Internal / external WAN interface: GSM module ready to receive.
F360	035C	Internal / external WAN interface: The switchover to the image method for blocked messages was forced.
F360	035D	Internal / external WAN interface: Threat of forced switchover to image method.
F360	035E	Internal / external WAN interface: Threat of forced switchover to image method.
F360	035F	Internal / external WAN interface: Permanent connection already active.
F360	0360	Internal / external WAN interface: Maximum number of messages exceeded.

Event ID (hex)	Detailed event (hex)	Meaning
F360	0361	Entering state: Internal / external WAN interface: The data brake for the connection to a sub-scriber was enabled.
F260		Exiting state: Internal / external WAN interface: The data brake for the connection to a sub-scriber was disabled.
F960	0362	Internal / external WAN interface: After transferring the PIN to the GSM module, no network contact could be established.
Messages from the routine task		
F560	0500	Installation of the routing program aborted.
F560	0501	Internal / external WAN interface: Receive task of the WAN driver unknown.
F560	0502	Receive task of the clock driver unknown.
F560	0503	Read time function unknown.
F560	0504	Receive task of the LAN task unknown.
F560	0505	Partner table unknown.
F560	0506	Error occurred receiving a message.
F560	0507	Unknown PDU received.
F560	0508	PDU with bad address received. Destination subscriber number not found.
F560	0509	No WAN driver available.
F560	050B	Error enabling heap memory.
F560	050C	Installation of the routing program ended. All routing tables available.
F560	050D	Error occurred sending a message.
F560	050E	No resources for creating the destination address table.
F560	050F	Destination address table not created.
F560	0510	No resources for PDU copy.
F560	0511	No resources for copy of partner table.
F360	0512	Entering state: Internal / external WAN interface: Redundancy function activated.
F260		Exiting state: Internal / external WAN interface: Redundancy function deactivated.
F960	0513	Message with incorrect block length received or block length is zero.
F560	0514	Max. number of messages exceeded.
F560	0515	Error in time-of-day synchronization over LAN.
F560	0516	MesA - Error sending a message.
F560	0517	The partner table/substitute table of a subscriber does not exist.
F560	0518	Error releasing memory.
F560	0519	Error in the MesA memory management
F560	051A	Entering state: MesA - Start of indication of a message memory overflow.
F460		Exiting state: MesA - End of indication of a message memory overflow.
F560	051B	Entering state: MesA - Start of indication of a message memory overflow.
F460		Exiting state: MesA - End of indication of a message memory overflow.
F560	051C	Message memory overflow occurred.
F560	051D	Entering state: Start of indication of a message buffer overflow.
F460		Exiting state: End of indication of a message buffer overflow.
F560	051E	Entering state: Overflow of message buffer active.

Event ID (hex)	Detailed event (hex)	Meaning
F460		Exiting state: Overflow of message buffer deactivated.
F560	051F	Buffer with messages was deleted.
F560	0520	MesA - Error in the dynamic assignment table.
F560	0521	MesA - WAN/LAN router: Error sending messages.
F560	0522	MesA could not send the acknowledgment for an org. 262 PDU to SubA.
F560	0523	MesA - WAN/LAN router; error sending over WAN/LAN driver.
F560	0524	Installation error occurred in the MesA.
F560	0530	MesA - System error occurred.
F560	0531	MesA - Error in the request for an Org4/14 message.
F560	0532	MesA - Unknown control command received.
F560	0533	MesA - Error in MesA system status list query.
F560	0534	MesA: Bad PDU detected.
Messages from clock driver		
F560	0620	Installation of clock driver aborted.
F360	0621	for error code = 0: Installation of the clock driver completed for error code = 1: Installation of clock driver aborted.
F560	0622	Entering state: Time synchronization (master) disturbed.
F460		Exiting state: Time synchronization (master) OK.
F960	0623	Entering state: Time synchronization (slave) disturbed.
F860		Exiting state: Time synchronization (slave) OK.
F560	0624	Error occurred setting the RMOS clock.
F560	0625	Error occurred reading the RMOS clock.
F360	0626	Illegal setting of the RMOS clock by *Set clock* PG service. TIM has onboard DCF77 clock.
F360	0627	Unknown message received.
F360	0628	Time synchronization PDU with incorrect ORG number received.
F360	0629	Bad synchronization PDU received from LAN.
F560	062A	Entering state: Time synchronization by DCF77 clock disturbed.
F460		Exiting state: Time synchronization by DCF77 clock OK.
F960	062B	Entering state: DCF77 clock radio signal disturbed.
F860		Exiting state: DCF77 clock radio signal OK.
F360	062C	TIM time OK. RMOS clock set for the first time.
F360	062D	Entering state: Changeover to daylight saving time.
F260		Exiting state: Changeover to standard time.
F360	062E	Entering state: Notification hour for daylight saving/standard time changeover active.
F260		Exiting state: Notification hour for daylight saving/standard time changeover completed.
F360	0630	Change in synchronization mode.
F360	0631	Daylight saving/standard time changeover performed manually.
F360	0632	Error occurred in manual daylight saving/standard time changeover.
F560	0633	Error occurred starting the synchronization task.
F560	0634	Error occurred starting the control task for synchronization.
F960	0635	Two time masters detected in one network.

Event ID (hex)	Detailed event (hex)	Meaning
F960	0636	Incorrect synchronization mode on MPI. Master mode expected.
F960	0637	No further module exists on MPI bus/party line.
F560	0638	External / internal WAN interface: A subscriber could not be synchronized following restart.
F360	0639	Error detected in time-of-day synchronization.
F360	063A	Bad time-of-day message received from DCF77 module.
F360	063B	Bad time-of-day message received from DCF77 module.
F360	063C	Time jump occurred.
F360	063D	Bad time-of-day message received from DCF77 module.
F560	063E	Error in synchronization request.
Messages from diagnostics server		
F560	0660	Error occurred installing the diagnostics server.
F360	0661	Installation of diagnostics server completed.
F560	0662	Error occurred sending a message.
F560	0663	Entering state: Error occurred receiving a message.
F460		Exiting state: Error occurred receiving a message.
F560	0664	Entering state: Heap memory overflow.
F460		Exiting state: Heap memory overflow eliminated.
F960	0665	Entering state: Unknown PDU received.
F860		Exiting state: Unknown PDU received.
F560	0666	Incorrect firmware version installed on TIM.
F360	0667	Entering state: All [n] LAN connections are disrupted.
F260		Exiting state: [x] of [n] LAN connections are OK.
F360	0668	Message buffer of TIM records was deleted.
F360	0669	Entering state: ST7 Message Monitor on
F260		Exiting state: ST7 Message Monitor off
F360	066A	Entering state: Extended diagnostics on
F260		Exiting state: Extended diagnostics off
F360	066B	Extended diagnostics - modification
F360	066C	Error occurred in system status list query.
F560	066D	Wrong firmware.
Messages from P bus server		
F560	06B0	Installation of P bus server aborted.
F360	06B1	Unknown message received from task.
F560	06B2	Error occurred receiving a message.
F360	06B3	Entering state: Power outage on P bus.
F260		Exiting state: Power supply on P bus OK.
F360	06B4	Entering state: I/O disabled by CPU.
F260		Exiting state: I/O enabled by CPU.
F560	06B5	Module on P bus not capable of communication.
F960	06B6	Parity error in P bus communication.

Event ID (hex)	Detailed event (hex)	Meaning
F960	06B7	Bit shift error in P bus communication.
F360	06B8	Diagnostics interrupt cannot be sent. Module is not enabled on P bus.
F360	06B9	Diagnostics interrupt cannot be sent. Diagnostics interrupt is not enabled on P bus.
F360	06BA	SDB0.SDB cannot be opened. File overwritten without comparison.
F360	06BB	SDB0.SDB cannot be created.
F360	06BC	SDB0.SDB cannot be written.
F560	06BD	BUS3-ASIC could not be initialized.
F360	06BE	SDB5.SDB cannot be opened. File overwritten without comparison.
F360	06BF	SDB5.SDB cannot be created.
F360	06C0	SDB5.SDB cannot be written.
Messages from the SDB handler		
F560	06E0	Installation of the SDB handler aborted.
F560	06E1	SDB could not be copied.
F560	06E2	SDB could not be deleted.
F560	06E3	SDB could not be loaded.
F560	06E4	SDB information of hierarchy 1 not available.
F560	06E5	SDB information of hierarchy 2 not available.
F560	06E6	SDB information of hierarchy 3 not available.
F560	06E7	Error chaining SDBs.
F360	06E8	SDB handler: Unknown job.
F360	06E9	Control instruction unknown.
F560	06EB	Error occurred during firmware update.
TD7onTIM messages		
F560	0700	Entering state: TD7 installation started.
F460		Exiting state: TD7 installation ready.
F560	0701	Semaphores not created.
F560	0702	Semaphores not created.
F560	0703	SecIntervall task was not started.
F560	0704	TD7_ObjectAdmin task was not started.
F560	0705	The path for SDB files could not be opened.
F560	0706	The TD7 SDB could not be found.
F560	0707	The TD7 SDB could not be opened.
F560	0708	Header with incorrect length of block ID.
F560	0709	TD7-SDB: No TD7 parameters found.
F560	070A	TD7-SDB: Block not found.
F560	070B	TD7-SDB: T4T_MAINHDR has incorrect length.
F560	070C	TD7-SDB: T4T_SUBDATA has incorrect length.
F560	070E	TD7-SDB: Unknown format in a destination subscriber block.
F560	0711	TD7-SDB: Number of partner blocks incorrect.
F560	0712	TD7-SDB: Unknown format in a partner block.

Event ID (hex)	Detailed event (hex)	Meaning
F560	0713	TD7-SDB: Unknown channel type.
F560	0715	TD7-Run: Not enough memory for the channel list.
F560	0716	TD7-SDB: Channel block not found.
F560	0717	TD7-Run: Not enough memory for a channel object.
F560	0718	TD7-SDB: The number of data entries is incorrect.
F560	071B	TD7-SDB: The number of object entries is incorrect.
F560	071C	TD7-Par: Subscriber object for a partner not found.
F560	071D	TD7-Par: Unknown partner.
F560	0721	TD7-Par: Invalid scan cycle ID.
F560	0722	TD7-Run: Object not in fast cycle.
F560	0725	TD7-SDB: No objects found in header ID.
F560	0726	TD7-SDB: Unknown format in an object.
F560	0729	TD7-Par: Not enough memory for scan cycle job list.
F560	072A	Basic channel memory assignment error.
F560	072D	TD7-Par: Wrong channel data type.
F560	072E	Unknown channel type.
F560	0730	TD7-Par: Invalid scan cycle ID.
F560	0731	Memory assignment error creating the object list for the current subscriber.
F560	0732	TD7-SDB: Number of objects does not match the number of objects in the header.
F560	0733	A read job to the CPU was not responded to after 1 ms.
F560	0736	TD7-Com: CPU communication error in object X, channel Y.
F560	0737	TD7-Com: CPU access error for object X, channel Y.
F560	073B	Memory assignment error creating the message buffer for a scan cycle.
F560	073D	A write job to the CPU was not responded to after 1 ms.
F560	073E	TD7-Com: Negative acknowledgment from LAN communication for job from scan cycle.
F560	0742	TD7-Run: Error reading an input trigger.
F560	0744	TD7-Par: Object without channels.
F560	074A	TD7-Par: Invalid address with trigger signal.
F560	074B	TD7-Par: Invalid address for net data in object X, channel Y.
F560	074C	An invalid address was reported for object X in channel Y.
F560	0752	TD7-SDB: Number of subscriber blocks does not match main header entry.
F560	0755	Unknown channel type.
F560	0759	Not enough memory to create the TD7onTIM send job list.
F560	0760	Initialization of TD7onTIM for source subscriber complete.
F560	0761	The general request of a subscriber is incomplete.
F560	0762	Timeout in the general request to a subscriber.
F560	0763	The general request of object X of a destination subscriber is incomplete.
F560	0766	Message with unknown source subscriber.
F560	0767	Unknown start index in received organizational message.
F560	0768	TD7-Run: Received organizational message not accepted due to invalid length.

Event ID (hex)	Detailed event (hex)	Meaning
F560	076C	Invalid start index in received data message.
F560	076D	TD7-Run: Received data too large for destination object.
F560	076E	TD7-Run: Start index of received data message does not match the receive channel of the destination object.
F560	0773	TD7-Run: Object without partner or channels.
F560	0774	TD7-Run: No destination object found.

Diagnostics messages and activation of the group error LED

If an error occurs during startup, the read group LED (SF) of the TIM lights up and a message to this effect is entered in the diagnostics buffer of the TIM.

If the TIM is installed as a CP in an S7-300 rack, a diagnostics interrupt is sent to the CPU.

The following table contains a summary of all error messages that caused the group error LED (SF) to light up.

Table 6- 20 Classification of the messages and activation of the group error LED

Error class	Detailed event (hex) in the diagnostics message
Internal error	0061 0063, 0064, 0065, 0066, 0067 0069 0078, 0079, 0080 0083 06B0
External error	0320
No parameter assignment	0070 0071 0620
Bad parameter assignment	0077 0081 0110 0112 0300 031C 0500 0620
RAM error	0085 0086

6.6.2 Meaning of ERRCLS and ERRCOD

Meaning of ERRCLS and ERRCOD in messages 0113 and 080C

Table 6- 21 Meaning of ERRCLS and ERRCOD (hex)

ERRCLS	ERRCOD	Meaning
FF	01	The LAN connection via which the frame is to be sent is not established. The message is deleted.
FF	02	The queue of the LAN connection via which the frame is to be sent has overflowed. The message is deleted.
FF	03	Wrong LAN connection: The connection index is higher than the maximum index limit. The message is deleted.
FF	04	Wrong LAN connection: • The send channel was not found. or • The addressed connection does not exist. The message is deleted.
FF	05	No memory could be reserved for the send job. The message is deleted.
FF	06	No memory could be reserved for the send job. The message is deleted.
FF	07	No send job exists. The message is deleted.
FF	08	The send job is already being used. The message is deleted.
FF	09	During the LAN connection establishment, there were still messages in the queue. All messages in the queue will be deleted.
FF	20	Only with TD7onTIM: The message is generated when a read or write job is transferred by TD7onTIM to the LAN COM and no acknowledgement is received from the CPU within 5 seconds. The job is returned with a negative acknowledgement to TD7onTIM by the LAN COM.
FF	22	Only with TD7onTIM: The message is generated when a read or write job is transferred by TD7onTIM to the LAN COM and there is no connection (CR connection) to the local CPU. The job is returned with a negative acknowledgement to TD7onTIM by the LAN COM.
81	04	Context is not supported: • Errors in the PDU structure or • Service unknown.
83	01	No memory can be reserved to forward the received message. The message is acknowledged negatively to the sender with result code 0x8301.
84	02	The TIM has sent a message and has received a negative acknowledgement from the partner with result code 0x8402 (remote BRcv block is in the incorrect status). The message is repeated.
84	04	A message was received via the LAN but could not be forwarded because there was no receive job. The message is acknowledged negatively to the sender with result code 0x8404.
84	05	The TIM has sent a message and has received a negative acknowledgement from the partner with result code 0x8405 (remote BRcv block is DISABLED). The message is repeated.

ERRCLS	ERRCOD	Meaning
85	00	Illegal PDU size: • Received PDU is larger than the negotiated maximum PDU size. or • The response does not fit in a PDU of the negotiated maximum PDU size.
87	02	A message with an unknown RID was received. The message is discarded and acknowledged negatively with event code 0x8702.
E0	FF	Error coding

6.7 Messages in the diagnostic buffer of the CPU

Introduction

The SINAUT TD7onCPU software package generates a series of operating and error messages that are entered in the diagnostic buffer of the CPU. These can be displayed in *SINAUT / TD7 CPU Diagnostics*. If a PG is connected and if this is registered for CP messages, these diagnostic messages are displayed on the PG immediately when they arise.

The messages generated by TD7 are, however, only displayed in plain text when they are called up in *SINAUT / TD7 CPU Diagnostics*. If you call up the messages generated by TD7 in *STEP 7 Diagnostics / Module Information, Diagnostic Buffer tab*, they are displayed in hexadecimal format.

Note

If there is no text file with the diagnostic texts of the TD7 events on the PG with which the diagnostic buffer is read out, the events are displayed in hexadecimal format.
Based on the following list, you can identify the significance.

6.7.1 SINAUT diagnostic messages of TD7onCPU

Classification of TD7 messages

The TD7onCPU software package uses the numeric range of event class B for its diagnostic messages. This event class is reserved for user-defined events in STEP 7. It is, however, used in SINAUT ST7 for SINAUT diagnostic messages.

Note

In SINAUT ST7, the user can only use event class A for user-defined diagnostic messages. Just as in event class B, this is also reserved for user-defined messages in STEP 7.

The SINAUT diagnostic messages of TD7onCPU use the range from Bx00h to BxFFh. To distinguish the individual messages, only in the last two numbers are significant, in other words the range from 00h to FFh.

x is a placeholder for an ID that allows the global classification of messages. For the TD7 diagnostic messages, the IDs 0 and 1 are used for events exiting state (0) and entering state (1).

Table 6- 22 Global classification of the SINAUT diagnostic messages of TD7onCPU

ID x	Resulting event IDs	Classification
0	B000h ... B0FFh	Exiting state messages
1	B100h ... B1FFh	Entering state messages

Most messages are entering state, only a few are existing state.

The TD7 diagnostic messages in hexadecimal and plain text format

The following table shows or SINAUT TD7 diagnostic messages according to their event IDs numbers in ascending order in hexadecimal format.

Most diagnostic messages have additional information added to them. If the diagnostic messages are displayed as text, this additional information is included in the texts suitably formatted.

In the explanations below, this additional information is specified within the texts as [Info1] or [Info2/3]. If the texts are not shown, the additional information is shown as 'Additional info 1 / 2 / 3' in the hexadecimal representation on the PG.

Table 6- 23 SINAUT diagnostic messages of TD7onCPU

No.	Event ID (hex)	Meaning
Parameter assignment errors, configuration errors		
0	B100	Parameter assignment error for object no. [Info1]: PartnerNo [Info2/3] not permitted.
1	B101	Parameter assignment error for object no. [Info1]: PartnerNo [Info2/3] unknown.
2	B102	Parameter assignment error for object no. [Info1]: PartnerObjectNo [Info2/3] not permitted.
3	B103	Parameter assignment error for object no. [Info1]: ST1 message no. [Info2/3] not permitted.
4	B104	Parameter assignment error for object no. [Info1]: ST1 object no. [Info2/3] not permitted.
5	B105	Parameter assignment error for object no. [Info1]: ST1 index no. [Info2/3] not permitted.
6	B106	Parameter assignment error for object no. [Info1]: ST1 PACK value [Info2/3] not permitted.
7	B107	Errors generating the object reference list: DB[Info1] cannot be created. Cause: [Info2/3].
8	B108	Parameter assignment errors in the object reference list: Reference to subscriber no. [Info1] and object no. [Info2/3] exists more than once.
9	B109	Object reference list 1 missing. Searching for reference to subscriber no. [Info1] and object no. [Info2/3].
10	B10A	Object reference list 2 missing. Searching for reference to subscriber no. [Info1] and object no. [Info2/3].

No.	Event ID (hex)	Meaning
11	B10B	Parameter assignment errors in object reference list 1: Reference to subscriber no. [Info1] and object no. [Info2/3] missing.
12	B10C	Parameter assignment errors in object reference list 2: Reference to subscriber no. [Info1] and object no. [Info2/3] missing.
13	B10D	Error in object DB no. [Info1]: SINAUT ST7 ID A5h missing.
14	B10E	Configuration error: No subscriber contained in DB BasicData (DB[Info1]) .
15	B10F	Unknown connection type for connection no. [Info1] to local subscriber no. [Info2/3].
16	B110	Unknown subscriber type (= [Info1]) for subscriber no. [Info2/3].
17	B111	Data received from an unknown local communication partner (MPI no. = [Info2/3]).
18	B112	In the send buffer of communication DB no. [Info1], an org. message with an unknown destination subscriber (= [Info2/3]) was deleted.
19	B113	Parameter assignment error for FC [Info1] in parameter [Info2/3].
20	B114	Parameter assignment error object no. [Info1] in parameter [Info2/3].
Reception errors		
32	B120	Content of receive buffer in communication DB no. [Info1] not plausible: Length of a message greater than the maximum receive length of [Info2/3] bytes.
33	B121	Content of receive buffer in communication DB no. [Info1] not plausible: Length of a message = 0.
34	B122	Content of receive buffer in communication DB no. [Info1] not plausible: Length of all messages greater than length of the receive data block.
35	B123	Received message in communication DB no. [Info1] incorrect: too many destination subscribers (= [Info2/3]).
36	B124	Received message in communication DB no. [Info1] incorrect: Subscriber no. too long (L = [In-fo2/3]).
37	B125	Org. Received message in communication DB no. [Info1] incorrect: Number of destination sub-scribers (= [Info2/3]) must be 1.
38	B126	Org. Received message in communication DB no. [Info1] incorrect: Address extension not permit-ted (AE = [Info2/3]).
39	B127	Received message in communication DB no. [Info1] incorrect: Address extension not permitted (AE = [Info2/3]).
40	B128	Org. Received message in communication DB no. [Info1] incorrect: Destination object no. (= [In-fo2/3]) not permitted (only 1 ... 32000 or 32767).
41	B129	Org. Received message (ST1) in communication DB no. [Info1] incorrect: Source object no. (= [Info2/3]) not permitted (only 0 or 32767).
42	B12A	Org. Received message for object no. [Info2/3] incorrect: Start index is >< 0. Current communica-tion DB = DB[Info1].
43	B12B	Org. Received message for object no. [Info2/3] incorrect: Number of net received bytes is > 2. Cur-rent communication DB = DB[Info1].
44	B12C	Received message for object no. [Info2/3] incorrect: Too many net received bytes. Current commu-nication DB = DB[Info1].
45	B12D	Org. Received message for subscriber object no. [Info2/3] incorrect: Subscriber unknown. Current communication DB = DB[Info1].
46	B12E	Org. received message for subscriber object no. [Info2/3] incorrect: Too many net received bytes. Current communication DB = DB[Info1].
47	B12F	Received message for object no. [Info1] incorrect: Source subscriber no. [Info2/3] unknown.
48	B130	Received message for object no. [Info1] rejected: Source subscriber no. [Info2/3] incorrect.

No.	Event ID (hex)	Meaning
49	B131	Received message for object no. [Info1] rejected: Source object no. [Info2/3] incorrect.
50	B132	Short Message Service (SMS): Status return message for object no. [Info1] from SMS center subscriber no. [Info2/3] cannot be interpreted.
51	B133	Short Message Service (SMS): Status return message for object no. [Info1] contains unknown status no. [Info2/3]
52	B134	Received message in communication DB no. [Info1] incorrect: Index no. is negative
53	B135	Org. Received message in communication DB no. [Info1] incorrect: The block length [Info2] of the org. section is not plausible.
54	B136	The receive error was detected by the TIM with subsbcriber number [Info1] when reading the data record. TIM error code = [Info2].
55	B137	A receive error occurred when the TIM with subsbcriber number [Info1] was reading the data record. An unexpected status = [Info2] was read in from the TIM I/O.
56	B138	A receive error occurred when the TIM with the subsbcriber number [Info1] was reading the data record. The length [Info2] of the indicated receive block is not permitted.
57	B139	A receive error occurred when the TIM with the subsbcriber number [Info1] was reading the data record. The actual data record length differs by [Info2] bytes from the indicated data record length.
58	B13A	A receive error occurred when the TIM with the subsbcriber number [Info1] was reading the data record. Error code (SFC RD_REC) = [Info2].
Send errors		
64	B140	Content of send buffer in communication DB no. [Info1] not plausible. Length of a send message too short: [Info2/3] bytes.
65	B141	Content of send buffer in communication DB no. [Info1] not plausible. Length of a send message too long: [Info2/3] bytes.
66	B142	Content of send buffer in communication DB no. [Info1] not plausible. A send message longer than the send buffer length: [Info2/3] bytes.
67	B143	Content of send buffer in communication DB no. [Info1] not plausible. Length of a send message odd: [Info2/3] bytes.
68	B144	Content of send buffer in communication DB no. [Info1] not plausible. Sum of all send message lengths incorrect: [Info2/3] bytes.
69	B145	Data message from object no. [Info1] to destination subscriber no. [Info2/3] could not be entered in the communication DB.
70	B146	Organisational message from object no. [Info1] to destination subscriber no. [Info2/3] could not be entered in the communication DB.
71	B147	Short Message Service (SMS): Monitoring time elapsed for SMS message no. [Info1] sent over SMS center subscriber no. [Info2/3]. No repetition.
72	B148	Entering state: Short Message Service (SMS): Monitoring time elapsed for SMS message no. [Info1] sent over SMS center subscriber no. [Info2/3]. Being repeated with backup mobile phone no.
	B048	Exiting state: Short Message Service (SMS): SMS message no. [Info1] sent over SMS center subscriber no. [Info2/3] could be delivered after repetition.
73	B149	Short Message Service (SMS): SMS message no. [Info1] was deleted because object no. [Info2/3] is disabled.
74	B14A	Short Message Service (SMS): Monitoring time elapsed for send job from object no. [Info1] sent to SMS center subscriber no. [Info2/3].
75	B14B	The send error was detected when writing the data record to the TIM with subsbcriber number [Info1]. Error code (SFC WR_REC) = [Info2].

No.	Event ID (hex)	Meaning
76	B14C	The send error was detected when writing the data record by the TIM with subsbcriber number [Info1]. TIM error code = [Info2].
Connection displays		
80	B150	Entering state: Connection to local subscriber no. [Info1] disrupted.
	B050	Exiting state: Connection to local subscriber no. [Info1] OK.
81	B151	Entering state: Subscriber no. [Info1] disrupted.
	B051	Exiting state: Subscriber no. [Info1] OK.
82	B152	Monitoring time for send job on local subscriber no. [Info1] elapsed. [Info2] = 0: The job could not be completed in time. [Info2] = 1: The job could not be started in time. [Info2] > 1: No. of the communication DB.
83	B153	Send job on local subscriber no. [Info1] failed. The job will be repeated. Communication DB no. = [Info2].
84	B154	Monitoring time for receive job from local subscriber no. [Info1] elapsed. [Info2] = 0: The job could not be completed in time. [Info2] = 1: The job could not be started in time.
85	B155	Receive job of local subscriber no. [Info1] failed. The job will be repeated. Communication DB no. = [Info2].
Request displays		
96	B160	General request to subscriber no. [Info1] incomplete.
97	B161	Single request to subscriber no. [Info1] incomplete.
98	B162	Error in general request to object no. [Info1]. Source subscriber no. = [Info2/3].
99	B163	Error in single request to object no. [Info1]. Source subscriber no. = [Info2/3].
Command/setpoint displays		
112	B170	Command input error for object no. [Info1]: Simultaneous input over hardware and software input.
113	B171	Command input error for object no. [Info1]: 1-out-of-8 error.
114	B172	Command/setpoint input error for object no. [Info1]: 1-out-of-n error. Additonal information [Info2/3] (0 = Comm. or Setp.; 1 = Comm.; 2 = Setp.)
115	B173	Command output error for object no. [Info1]: Command and control byte not identical.
116	B174	Command output error for object no. [Info1]: 1-out-of-8 error.
117	B175	Command output error for object no. [Info1]: 1-out-of-n error.
Time displays		
128	B180	Daylight saving/standard time changeover at change of day not permitted.
129	B181	Date/time error when setting the CPU clock. Incorrect date: [Info1].
130	B182	Date/time error when setting the CPU clock. Incorrect time: [Info2/3].
131	B183	Time synchronization disrupted on CPU.
Processing errors		
144	B190	Data loss in object no. [Info1]. Last received data was not processed by typical.
145	B191	Received data loss in subscriber object no. [Info1]. Last received data was not processed by BasicTask.
Operating displays		
160	B1A0	Initialization of subscriber no. [Info1] completed.

7 SINAUT PG Routing

7.1 What is PG Routing?

7.1.1 Introduction

In the SIMATIC world, the term routing is defined as follows:

Routing is finding the path for information beyond network boundaries.

In the SIMATIC world at the present time, it is not possible to configure a connection over which data can be transmitted automatically between the two endpoints involved if this connection goes beyond the boundaries of a network.

What is possible, however, is PG routing. Using the PG routing, it is possible to access a programmable module or a module with diagnostic capability beyond network boundaries from a programming device (PG) or computer (PC).

SIMATIC PG routing and SINAUT PG routing

PG routing allows any type of diagnostics with diagnostics-compliant modules. Test, commissioning, and service functions can be executed, such as opening blocks online, monitoring, editing and overwriting or changing the operating mode mode of modules.

SIMATIC PG routing
SIMATIC PG routing is possible only over network types such as MPI, PROFIBUS, and Ethernet. SIMATIC PG routing was released with STEP 7 V5.0.

SINAUT PG-Routing
Expanding SIMATIC PG routing, SINAUT PG routing also works over SINAUT networks; in other words, in WANs.

In terms of functionality, SIMATIC PG routing and SINAUT PG routing are largely identical with the only difference being that SINAUT PG routing also functions beyond the boundaries of SINAUT networks. This gives the user a convenient option of remote programming and remote diagnostics over the company telecontrol network.

Note

In the remainder of this chapter, the terms *PG routing* and *remote* are used with the following meaning:

- PG routing
  PG routing is PG routing over network types such as MPI, PROFIBUS, and Ethernet as well as SINAUT networks.
- Remote
  A remote CPU or remote TIM is a "remote" module that can be accessed from a PG over SINAUT telecontrol networks.

7.1.2 Examples of configuration for PG routing

The following figures illustrate the basic principle of PG routing based on configurations.

Basic configuration of PG routing

In the basic configuration, a PG in the master station is configured and connected to the MPI bus. PG routing extends from the master TIM over a SINAUT network (WAN) to a station TIM over which the CPU connected to it can be reached.

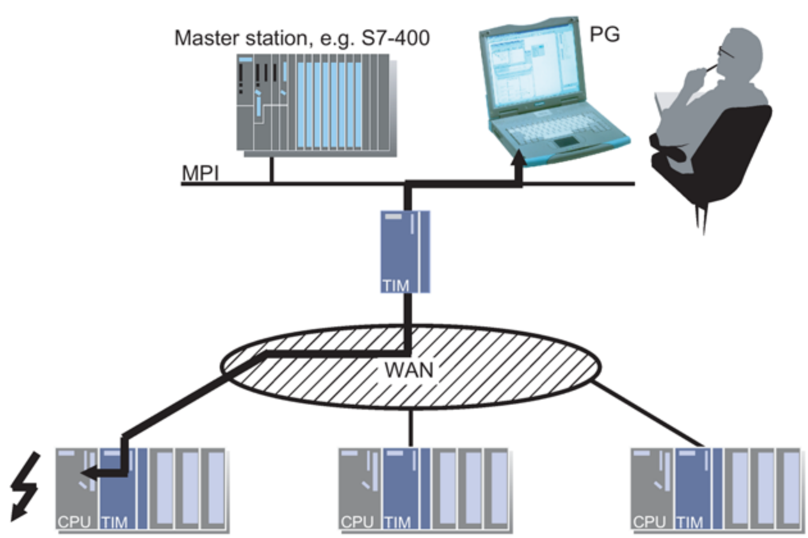


Figure 7-1 Basic configuration of PG routing

PG routing from a SINAUT ST7cc control center with SIMATIC STEP 7

In this case, PG routing is not from a PG in the master station but from the SINAUT ST7cc central control system that is connected to the MPI bus of the master station TIM. In principle the routing path outlined here is the same as in the described basic configuration with a central configured PG.

A SINAUT system configuration equipped with SINAUT ST7cc as the central control system already has the full range of functions of PG routing if SIMATIC STEP 7 is also installed in addition to the SINAUT ST7cc software. PG routing is generally activated here. This means that no further preparatory measures are necessary for PG routing to be able to perform parameter assignment, diagnostics and service functions for remote subscribers via SINAUT networks.

For this configuration, PG routing is from a PC with STEP 7 software via the master station TIM and a SINAUT network to a SINAUT station where both the station TIM and the CPU can be accessed.

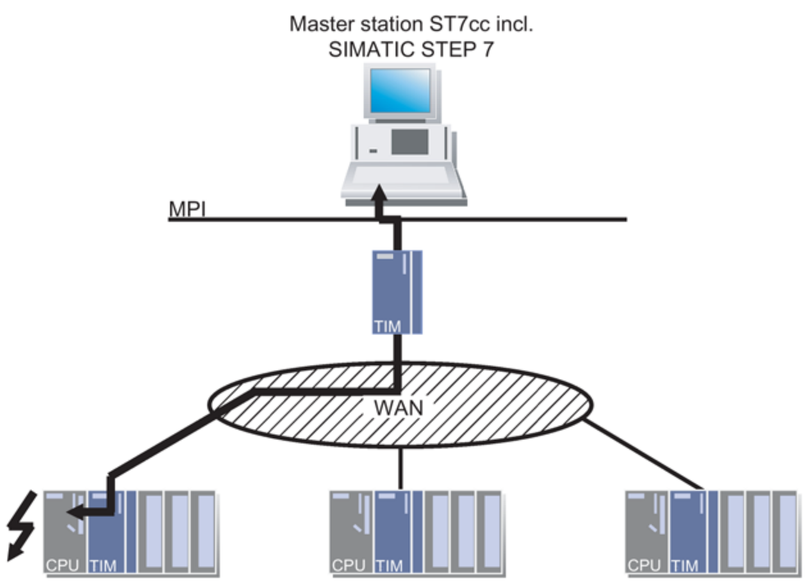


Figure 7-2 PG routing from a SINAUT ST7cc control center with SIMATIC STEP 7

PG routing indirectly via an outlying PC / laptop with remote access

The configuration shown below is initially the same as with the central control system SINAUT ST7cc. PG routing is now used in this case via a telephone/ISDN network with an outlying PC or laptop. With the aid of remote access software a connection is established to the PC in the master station, so that this can be operated via the telephone/ISDN connection by the outlying PC / laptop. From there, PG routing is via the master station TIM and a SINAUT network to a SINAUT station where both the station TIM and the CPU can be accessed.

As remote access software various products are available on the market such as Symantec pcAnywhere.

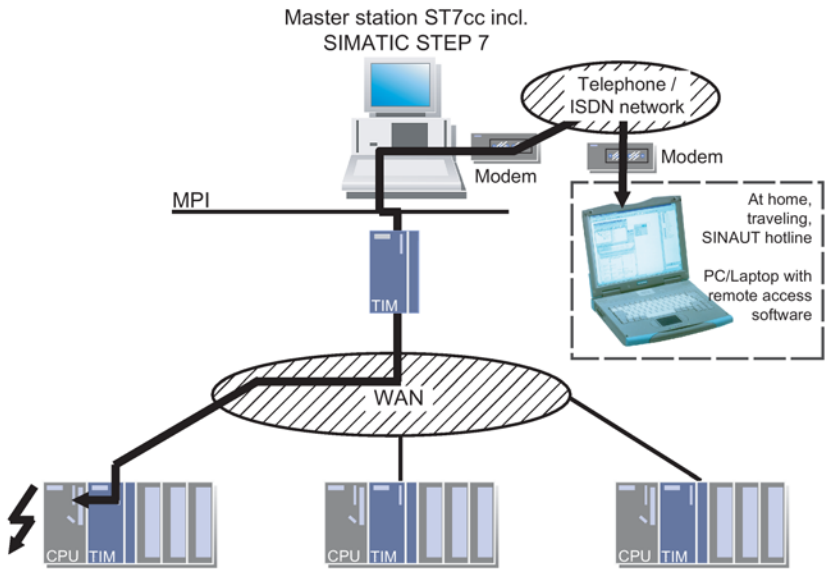


Figure 7-3 PG routing indirectly via an outlying PC or laptop with remote access

PG routing indirectly via an outlying PG/PC with SIMATIC STEP 7

The basic configuration with a configured PG in the master station can be expanded by a teleservice adapter. The PG is configured on the MPI bus in the master station and connected. The teleservice adapter allows the relocation of the PG via a telephone /ISDN network. This extends the MPI bus in the master station via a modem connection.

PG routing is from the PG with the SIMATIC STEP 7 software package via the modem connection, the master station TIM and a SINAUT network to a SINAUT station where both the station TIM and the CPU can be accessed.

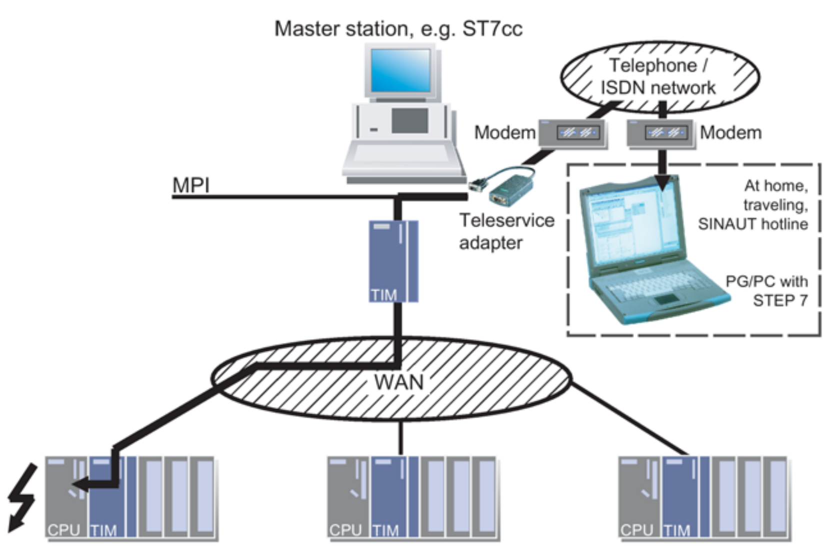


Figure 7-4 PG routing indirectly via an outlying PG/PC with SIMATIC STEP 7

7.1.3 Range of functions of PG routing

Range of functions of PG routing

SINAUT ST7 allows the configuration of simple and more complex networks. For PG routing this means that the access to remote TIM and CPU modules is always along the hierarchical arrangement of a communications network "from top to bottom", in other words from the master station to the node station or station.

On the communications path therefore three levels are significant:

Table 7- 1 Communications path of PG routing

Starting point	Intermediate point	Destination
PG/PC on the MPI bus of the master station TIM	TIM module of a node station (possibly several at different levels)	Station TIM or station CPU

The following table explains from which starting point via which communications paths to which destination PG routing operates.

Table 7- 2 Overview of the communications paths capable of routing

Starting point	via	Intermediate point	via	Destination
Master station	-	-	D	Station
Master station	-	-	DT	Station
Master station	-	-	DMT	Station
Master station	-	-	DN	Station
Master station	-	-	SPN	Station
Master station	-	-	D	Node station
Master station	-	-	DT	Node station
Master station	-	-	DMT	Node station
Master station	-	-	DN	Node station
Master station	-	-	SPN	Node station
Node station	-	-	D	Station
Node station	-	-	DT	Station
Node station	-	-	DMT	Station
Node station	-	-	DN	Station
Node station	-	-	SPN	Station
Master station	D	Node station	D	Station
Master station	D	Node station	DN	Station
Master station	DN	Node station	D	Station
Master station	DMT	Node station	D	Station

Table 7- 3 Abbreviations in the table of the communications paths capable of routing

	Network type / mode
D	Dedicated line / polling
DT	Dedicated line / polling with time slots
DMT	Dedicated line / multi-master polling with time slots
DN	Dial-up network / spontaneous mode
SPN	Spontaneous network / spontaneous mode

7.1.4 Properties and restrictions of PG routing

When using PG routing with the SINAUT Diagnostics and Service tool or the SIMATIC Manager, certain special features and restrictions to the functions must be kept in mind.

Functions of the SINAUT Diagnostics and Service tool with PG routing

- *TIM message monitor* function

  Activating the *TIM message monitor* on a remote TIM using PG routing is not possible.

- *Firmware update* function

  When using the *firmware update* function, remember that large amounts of data are transferred. With remote modules, long processing times of several minutes can occur.

  Note

  No firmware update via dial-up connections

  The loading of new firmware using PG routing is not possible via dial-up connections.

- *Repair* function

  The *repair* function must not be used over PG routing.
  If you use the *repair* function, the flash disk of the TIM is formatted and the software completely deleted on the TIM. Following this, the module is no longer accessible over the SINAUT network. Reloading the TIM software is then only possible locally over the MPI bus.

Functions of the SIMATIC Manager with PG routing

- Display *Accessible nodes* function
  The display *Accessible nodes* function is available only for subscribers connected to the local MPI bus. This restriction applies to SIMATIC PG routing and therefore also to SINAUT PG routing.

- *Hardware diagnostics* function
  The hardware diagnostics function is available only for subscribers connected to the local MPI bus. This restriction applies to SIMATIC PG routing and therefore also to SINAUT PG routing.

- *Download* function
  The *download* of an entire station (CPU 300 plus TIM) leads to a connection abort. Since

the TIM system data are downloaded first followed by a restart on the TIM, the second step of the CPU data download is interrupted.
It is possible to repeat the CPU download. We, nevertheless, recommend that you download the block folder of the CPU and the TIM module separately.

- *Upload to PG* function
  The *upload to PG* function is available only for subscribers connected to the local MPI bus. This restriction applies to SIMATIC PG routing and therefore also to SINAUT PG routing.

Restrictions for PG routing with SINAUT ST1 components

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

7.2 System requirements for PG routing

Introduction

To use the PG routing function in the SINAUT telecontrol network with SINAUT ST7, the following requirements must be met or preparations made.

- **STEP 7** with at least Version 5.1 and service pack 3 is required on the PG/PC.
- The **SINAUT software package** version V3.0 or higher must be installed on your PG/PC.
- The **RMOS operating system** of the TIM modules of the type TIM 3 and TIM 4 in your telecontrol network should be at least version 2.14. TIM modules of version 2.04 can be used with certain restrictions.
- The **TIM firmware** for the TIM 3 and TIM 4 must have a certain minimum version.
  With dedicated lines, firmware version V 3.14 is required.
  In dial-up networks, a minimum firmware version of V 3.46 is required.
- For SINAUT dedicated line or dial-up networks, a **maximum message length** of 240 must be set.
- On slow connections with only 1200 bauds transmission speed, the **Retry factor** must be set higher than 0.
- For PG routing, **new SDBs** must be transferred to the TIM modules. These are compiled with the SINAUT software package as of V3.0.

Note

PG routing is released for the Ethernet TIMs regardless of the firmware version.

7.2.1 STEP 7

STEP 7 at least version 5.1,service pack 3 must be installed on your PG/PC. Only then is your PG/PC capable of routing and it is guaranteed that the routing SDBs are compiled with the correct content.

7.2.2 The SINAUT software package

Only with the SINAUT software package as of version V 3.0 can the so-called routing SDB (SDB type 3002) be generated for the TIM modules. In addition to this the LAN SDB (SDB type 3201) has extra parameters required for PG routing. PG routing can only be used via TIM modules on which these routing and LAN SDBs are loaded.

Note

The routing SDB (SDB type 3002) is only generated for TIM modules that actually require this information.
For a TIM inserted in an SIMATIC S7-300 station as a CP, no routing SDB is generated if there is no local network such as MPI bus or PROFIBUS DP in this station. The absence of the routing SDB in the SDB directory is in this case not a generation error. Although the routing SDB is missing, this TIM and the connected S7-300 CPU can be reached via PG routing.

Table 7- 4 Overview of the SDB types

SDB type	Content
0	Standard SIMATIC SDB
700	PBC connection parameters
3002	Routing SDB
3201	LAN data
3202	WAN data
3100	Ethernet data
3203	SINAUT subscriber data
3205	Parameter for X connection
3206	TD7onTIM data

Apart from SDB type 0, the SDBs of the type 3201, 3202, 3203 and 3205 are always generated for the TIM, the SDB types 700, 3002, 3100 and 3206 on the other hand only when necessary.

7.2.3 RMOS operating system of the TIM 4

Whether a TIM of the type TIM 4 is suitable for PG routing, depends, among other things, on the version of the RMOS operating system of the TIM. In this respect, there are versions ideally suited for PG routing and versions that mean certain restrictions:

- TIM modules with an RMOS version older than 2.04 are not routing-compliant.
- TIM modules with an RMOS version as of 2.04 are routing-compliant but involve restrictions regarding the number of remote subnets. No more than 10 remote subnets may be configured in the SINAUT project.
- TIM modules with an RMOS version as of 2.08 are routing-compliant but involve restrictions regarding the number of remote subnets. No more than 20 remote subnets may be configured in the SINAUT project.
- TIM modules with an RMOS version as of 2.14 are fully routing-compliant.

The operating system of the TIM module can be upgraded to the current version. A special cable is required for this. This can be borrowed from the hotline. Along with the cable, the current version of the operating system is supplied on diskette with instructions on upgrading using a PG/PC.

7.2.4 TIM firmware for TIM 4

Not only the version of the operating system but also the version of the TIM firmware for the TIM 4 decides whether a TIM is suitable for PG routing. In this respect, there are versions ideally suited for PG routing and earlier versions that mean certain restrictions.

For PG routing, a TIM firmware version of V 3.46 is generally recommended. It depends on the driver activated on the TIM (dedicated line or dial-up network driver) whether an older version will be adequate. PG routing over the dedicated line driver is possible as of firmware version V 3.14. If the dial-up network driver is activated on a TIM module, firmware version V 3.46 or higher must be installed.

7.2.5 Settings for SINAUT networks

Retry factor and maximum message length

PG routing can only be used when the *maximum message length* network parameter for the SINAUT networks is set to a maximum of 240. Since this value is normally entered as default in the *Properties* dialog for SINAUT networks, this is normally already set.

If a transmission speed of 1200 bauds is set in a SINAUT network, make sure that the retry factor is higher than 0. We recommend that you use the default values available for the retry factor in the configuration dialogs:

- For dedicated lines: 3
- For dial-up networks: 7

7.2.6 Recompiling system blocks

If you want to enable PG routing in an existing SINAUT ST7 system, the project must be recompiled in the subscriber management of the SINAUT configuration tool (as of V 3.0). It is adequate to simply recompile the new SDBs there. The programs for CPU modules do not change for PG routing.

After compiling the SDBs, the entire SINAUT system configuration must be downloaded and activated on the TIM modules.

7.2.7 Loading newly generated SDBs on a running TIM

New SDBs during operation

If new SDBs are generated for an existing SINAUT ST7 system, these then need to be loaded on the relevant TIM modules of the system.

Without PG routing, the SDBs are loaded on each TIM directly on site via the MPI bus and the MPI interface of the CPU. With an Ethernet TIM, the Ethernet interface of the TIM can be used.

A further option for loading the SDBs from the master station on the remote TIM modules is described in the manual in the section "Central loading of the SDBs with PG routing".

Note

Loading new system data blocks on a TIM module is performed using SINAUT diagnostics and service tools with the "Load SDB" function For further information, refer to the relevant section of the manual.

When downloading system data blocks in hardware configuration, make sure that no connection SDBs (SDB7xx) are downloaded there.

To activate newly downloaded SDBs on a TIM module, the TIM must be restarted. The consequences of the restart are, however, as follows:

- The connection of the PG to the TIM is lost.
- The connections of the TIM to other SINAUT partners (SINAUT connections) are lost. This leads to error messages on the partners of the TIM module.
- In the case of a node TIM, the connections to lower-level stations are also reported as being disrupted.
- With a node TIM, during the restart data messages stored on the TIM may also be lost. Above all this is relevant in station TIMs in the dial-up network.

When downloading SDBs to TIM modules, you should therefore note the following points:

- Before you transfer the SDBs, you should give the TIM the opportunity of transferring any messages stored on it.
- After restarting the TIM, the SINAUT connections are established again automatically. The connection between the PG and the TIM must, however, be activated by the user on the PG if it is required.

7.2.8 Central SDB download using PG routing

The SDBs can be downloaded as previously for every TIM directly on site via the MPI bus. It is, however, conceivable to do this from the master station, in other words making use of PG routing although the system has not yet been activated for this. This procedure can be considered if it is already known that the TIM modules in the stations are suitable or prepared for PG routing in terms of the operating system and TIM firmware.

For the following example of a configuration, the central downloading and activation of the SDBs is performed in four consecutive steps:

Step 1:

First download the newly generated SDBs to the TIM module in the master station, in the example shown TIM 4 a, and activate the SDBs by subsequently restarting the TIM module. TIM 4 a is now capable of routing.

Repeat this procedure for all TIM modules in the master station, if there are more TIM modules present.

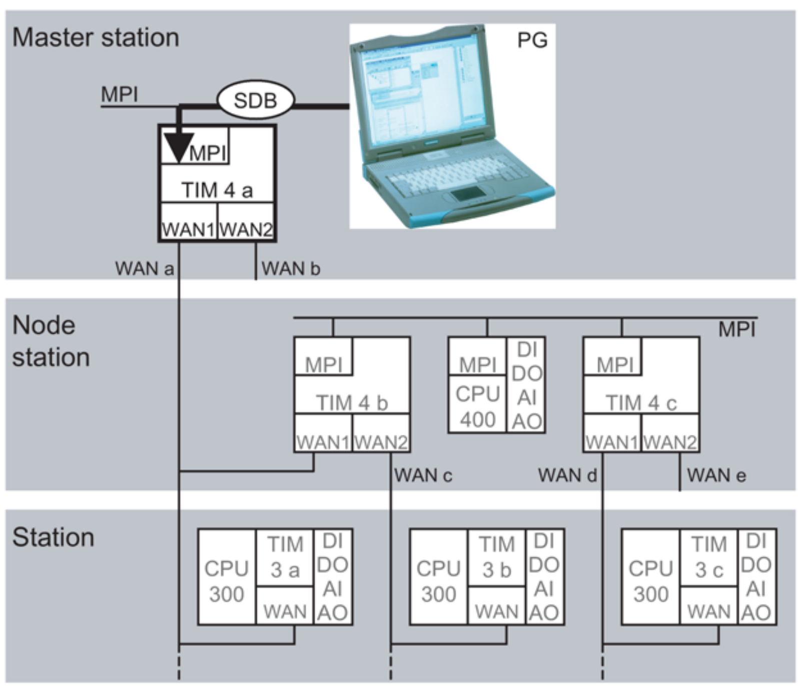


Figure 7-5 Step 1 of the central downloading of SDBs in the configuration example

Via the master station TIM 4 a that is now capable of routing, the PG has access to all station or node TIM modules that are connected via WAN a and WAN b directly to this master station TIM. In the example the PG can reach the two TIM modules TIM 4 b and TIM 3 a, however the local CPU modules there not yet, nor the TIM 4 c of the node station.

Because the two TIM modules TIM 4 b and TIM 3 a can now be reached, the PG can run the diagnostics and service functions for these two TIM modules. From the PG, it is also possible to transfer SDBs to these two TIM modules.

Step 2:

Download the newly generated SDBs to the TIM modules TIM 4 b and TIM 3 a and activate the SDBs by restarting the two TIM modules. TIM 4 b and TIM 3 a are now capable of routing.

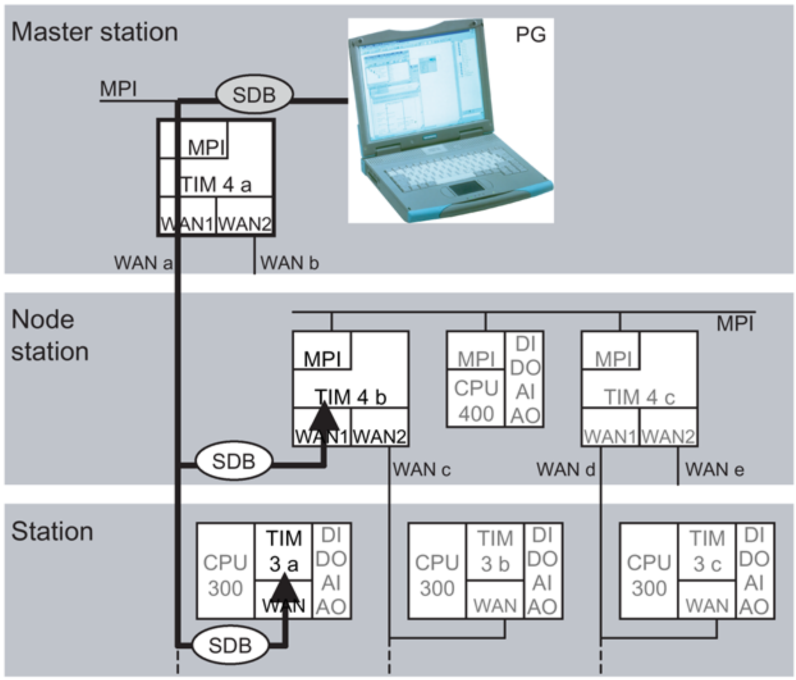


Figure 7-6 Step 2 of the central downloading of SDBs in the configuration example

Via the two modules TIM 4 b and TIM 3 a that are capable of routing, the PG now has access to all locally connected CPU and TIM modules. All CPU and TIM modules connected to a TIM with routing capability via the backplane bus (with S7-300), via MPI or via WAN are local.

In the example, the PG can now reach the following CPU and TIM modules:

- In the stations the CPU connected to TIM 3 a via the backplane bus.
- In the node station the CPU and TIM 4 c via MPI, the station TIM 3 b connected to TIM 4 b via WAN c, but the local CPU there not yet.

As of now the reachable CPU modules can be remotely programmed. For the two newly reachable TIM modules TIM 4 c and TIM 3 b, the PG can once again run the diagnostics and service functions. The PG can therefore also transfer SDBs to these TIM modules.

Step 3:

Download the newly generated SDBs to the TIM modules TIM 4 c and TIM 3 b and activate the SDBs by restarting the two TIM modules. TIM 4 c and TIM 3 b are now also capable of routing.

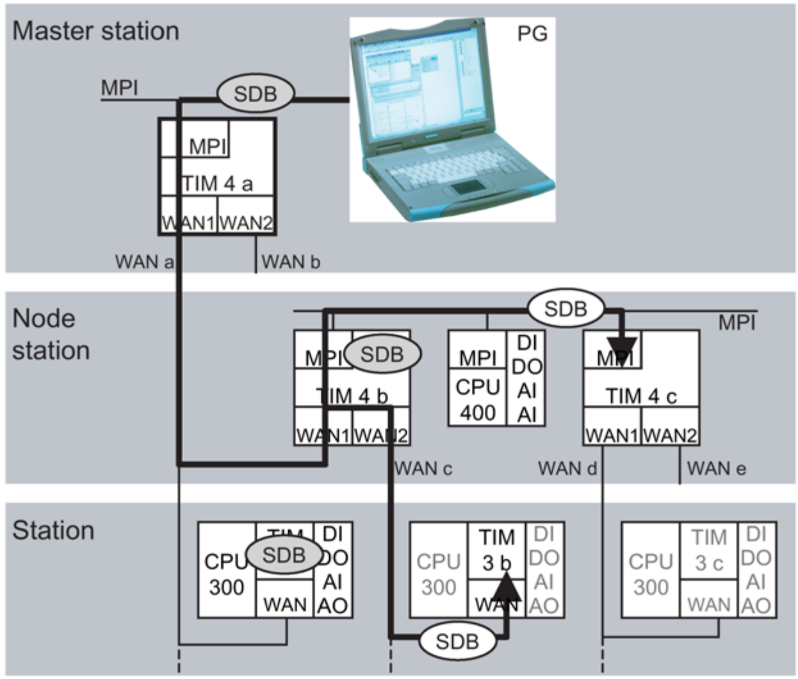


Figure 7-7 Step 3 of the central downloading of SDBs in the configuration example

Via the two TIM modules TIM 4 c and TIM 3 b that are now capable of routing, the PG has access to all locally connected CPU and TIM modules there. In the example, the PG can now reach the following CPU or TIM modules:

- In the stations the CPU connected to TIM 3 b via the backplane bus.
- The station TIM 3 c connected to TIM 4 c in the node station via WAN d, the local CPU in the station connected to TIM 3 c not yet.

As of now the reachable CPU module can be remotely programmed.

For the two newly reachable TIM module, the PG can once again run the diagnostics and service functions. The PG can therefore also transfer SDBs to this TIM module.

Step 4:

Download the newly generated SDBs to the TIM module TIM 3 c and activate the SDBs by restarting the TIM module.

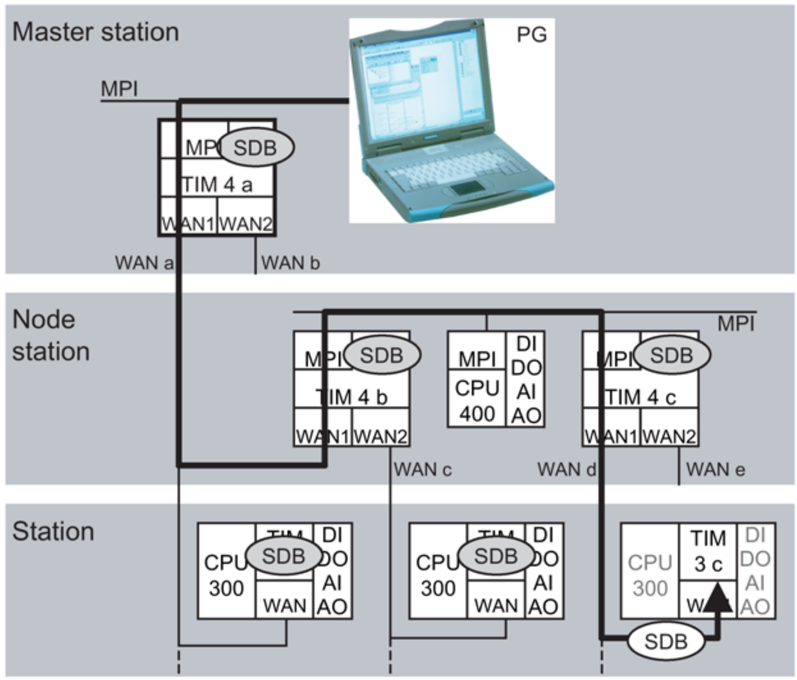


Figure 7-8 Step 4 of the central downloading of SDBs in the configuration example

TIM 3 c is now also capable of routing. As of now the CPU module connected to TIM 3 b can be remotely programmed.

7.3 Application of PG routing

Before you perform PG routing over the SINAUT telecontrol network with your PG of PC, you must first adapt the properties of the PG/PC interface and set the assignment of the PG/PC in the SINAUT network.

7.3.1 Properties of the PG/PC interface

Adapting the PG/PC interface

1. Open the *Control Panel* window by clicking on the *Start / Settings / Control Panel* menu.
2. Select the *Set PG/PC interface* icon.
3. In the *Set PG/PC Interface* dialog, set the MPI interface in the *Interface parameter assignment used* box.
4. Then click the *Properties* button.
5. Confirm the warning dialog with *Yes*. The *Properties* dialog opens.

6. In the *MPI* tab of the *Properties* dialog, select option *100 s* in the *Timeout* list.
7. Close the dialogs *Properties* and *Set PG/PC interface* with the *OK* button and close the *Control Panel* window.

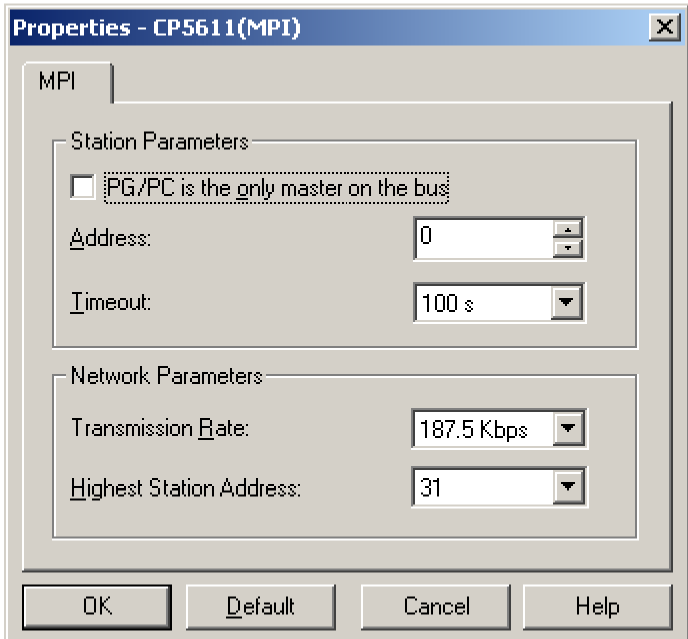


Figure 7-9 *Properties* dialog of the interface in the Control Panel

This completes the adaptation of the PG/PC interface.

7.3.2 PG/PC assignment in the SINAUT network

Before you can use PG routing with a PG/PC over SINAUT networks, this must be configured and assigned within a SINAUT project.

Assigning the PG/PC

The PG/PC is assigned in the network using the SIMATIC network configuration tool NetPro.

1. Open the project in which you want to use PG routing in the SIMATIC STEP 7 *NetPro* network configuration tool.
2. Open the NetPro catalog if it is not yet open.
3. Drag a PG/PC to the network window from the NetPro catalog directory *Stations* and place it at a suitable position.
4. Right-click on the PG/PC you have just installed. A context menu opens.
5. In the context menu, click on the *Assign PG/PC* option. The *Properties* dialog opens.

6. Select the MPI network to which the configured PG/PC is connected in the *Configured interface* box of the *Properties* dialog.

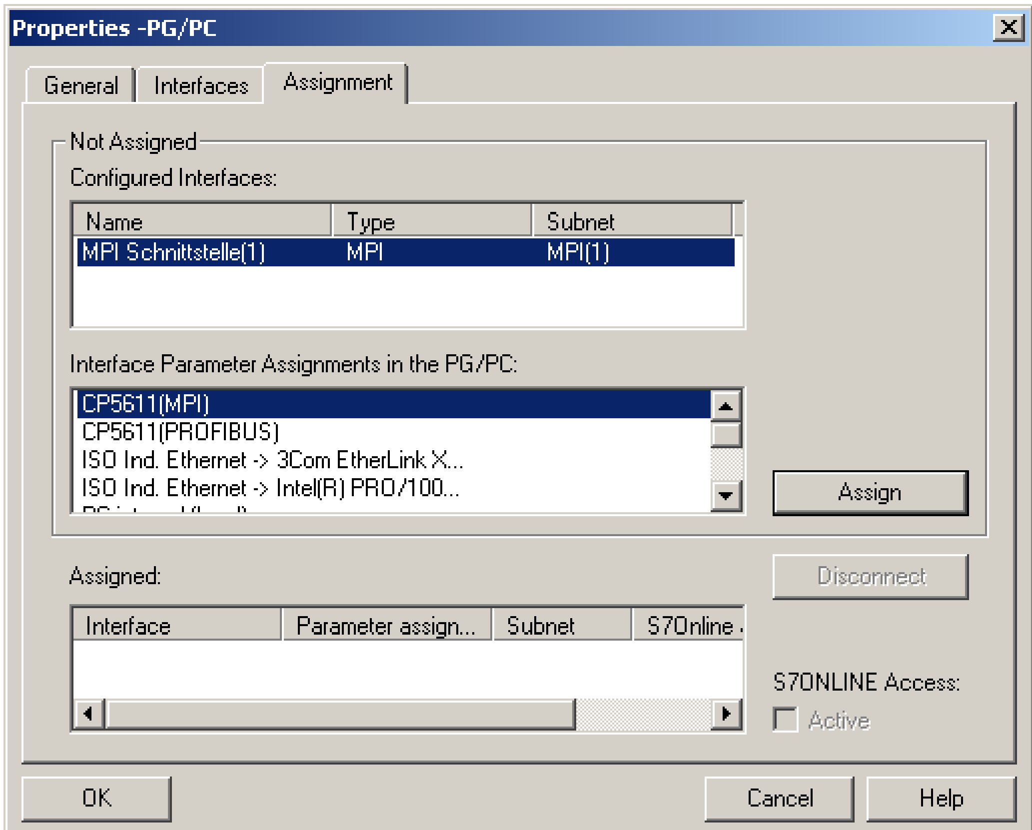


Figure 7-10 *Properties - PG/PG* dialog / *Assignment* tab in NetPro

7. Select the MPI interface you want to use in the *Interface Parameter Assignments in the PG/PC*.
8. Click on *Assign*. The assigned MPI interface is displayed in the *Assigned* box. The interface is now enabled for PG routing access.
9. Close the dialog with the *OK* button. The successful assignment of the PG/PC is indicated by an MPI connection on a yellow background and a yellow arrow pointing upwards in the PG/PC icon in *NetPro* and in the *SIMATIC Manager*.

10. Save your project in *NetPro*.
11. Connect your PG/PC to the MPI bus to which you assigned your PG/PC in *NetPro* over a PG cable.

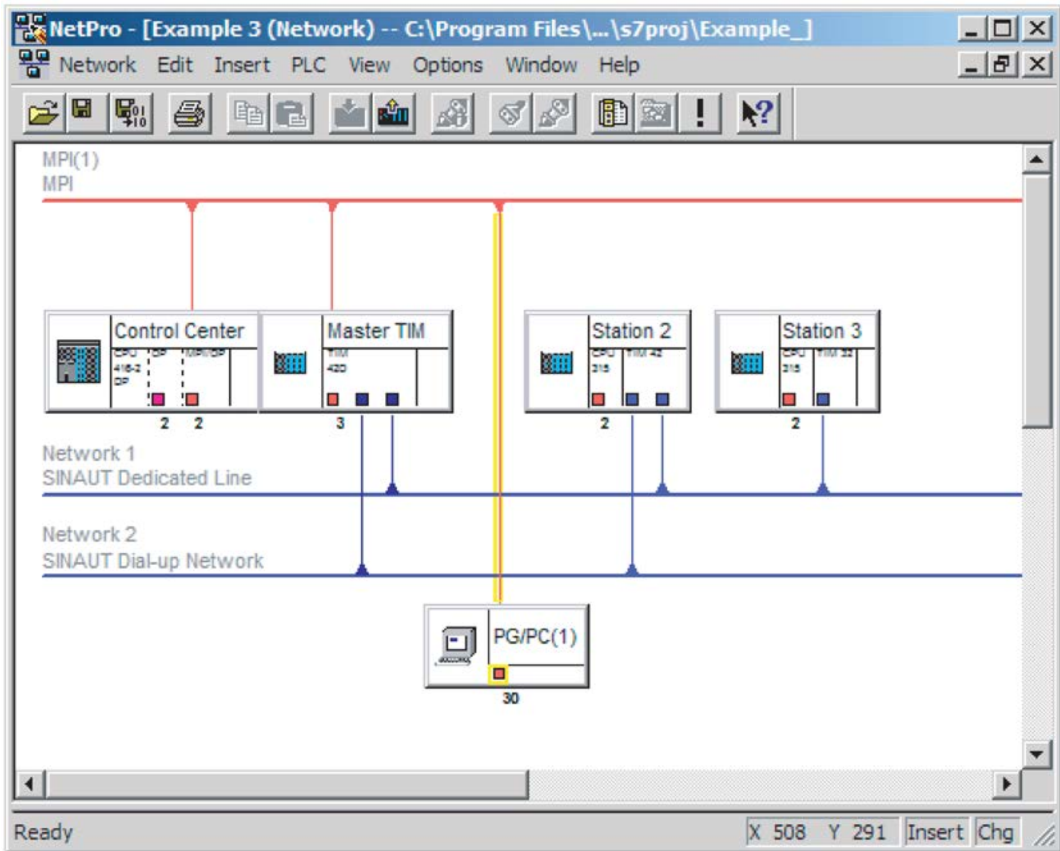


Figure 7-11 Project view in *NetPro* with assigned PG/PC

Note

As long as you leave your PG/PC connected to the point in the network as you assigned it in *NetPro*, you do not need to cancel the assignment. Not even if you want to turn off the PG/PC. Each time you turn on the PG/PC and open the project, you can use PG routing again immediately. You do not need to make settings or assignments again.

If, on the other hand, your PG/PC is not always at the same location or if you change to different projects on your PG/PC, we strongly recommend that you always cancel the assignment before your PG/PC is turned off or before you change to a different project. This ensures that the PG/PC can be assigned again when used at a different location or in a different project.

7.3.3 PG routing in dial-up networks

There are PG functions that maintain constant message traffic with the CPU or the TIM after they have started because these functions want to update themselves constantly. Such a function in a dial-up network would lead to a permanent connection with the relevant station until you deselect the function.

Other PG functions execute only once. They do not need constant updating. In a dial-up network, in this case, the dial-up connection is established briefly and then terminated again.

If a continuous connection to a dial-up station is required with PG routing, it is advisable to start a PG function at the beginning that requires constant updating regardless of the actual PG function that has just been activated. This function should then be maintained while PG routing is being used thus forcing a permanent connection. At the same time, other PG functions can be performed.

The PG function most suitable for maintaining a permanent connection is the *Operating mode* function. This requires little time to execute and the load on the other data traffic caused by the constant updating is only minimal. Other suitable functions include *Module Information* and *Set Time of Day*.

PG routing places load on the transfer of process data; in other words, this data is transferred to the control center while the PG connection is established. This will will be slower than when no PG routing is activated.

On the other hand, the parallel transmission of data messages also slows down the reactions to PG functions that have started. This is particularly the case when the station TIM is currently being used for PG routing and this TIM has a lot of stored messages, PG routing will be very slow at the beginning. It is therefore advisable to give the station TIM the opportunity to transfer its messages before intensive PG functions are started.

7.3.4 Canceling the PG/PC attachment in the SINAUT network

Canceling the PG/PC assignment

If a PG was assigned in a project and then needs to be used at a different location, for example locally connected directly to a CPU or TIM, the assignment must first be canceled. Follow the steps outlined below:

1. Right-click on the PG/PC that is still assigned in the project you have opened in *NetPro*.
2. In the context menu that opens, select *Cancel PG/PC Assignment*.
3. Acknowledge the warning dialog with *OK*.
   The PG/PC assignment is now canceled, this is indicated as follows in the network image of *NetPro*:
   – The connecting line from the PG/PC to the MPI bus is no longer on a yellow background.
   – The yellow arrow in the PG/PC icon disappears
4. Save your project in *NetPro*.

Glossary

Alarm Logging

In Runtime mode, alarm logging controls the acquisition and archiving of events and provides display and operator input options. Using the message blocks, message class and message type structure elements, the configuration engineer can class the events according to their significance and allow the operator a fast evaluation of the status of the system.

Analog value

An analog value is an analog process variable such as pressure, temperature etc. It is acquired over an analog input as a current or voltage value and converted by this module to a binary-coded value. In total, the converted value occupies 1 word; in other words, 16 bits including sign bit.

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).

Automation program

The automation program is the program section on the CPU that monitors and controls the technological process.

COM port

→ *RS-232*

Command (DNP3)

Context: SINAUT DNP3

A command it is a setpoint or a switching command from the DNP3 master station. It is sent to a station.

Command (ST7)

Context: SINAUT ST7

A command is binary information that is transferred once as a 1 when there is a signal change from 0 to 1. The trailing edge from 1 to 0 does not trigger transmission again. At the receiving end, a command is either output as a pulse (selectable duration) or it is reset by the local user program after it is has been executed. 8 commands are put together to form a byte.

When inputting and outputting commands, reliability and safety are important. At the input end, for example, there is a check to determine whether only one command is pending at the time of acquisition (1-out-of-n check). Only then is the transmission triggered. If several commands are pending at the same time, an error is detected. There is no transmission. To transmit a command byte, a total of 1 word is used: One "original" command byte and a copy. At the receiving end, a command is only output when the "original" command byte and the copy have the identical content and when only 1 command was received (once again a 1-out-of-n check).

Conditional spontaneous message

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 messages reach or exceed a size of 202 bytes.
- When an important message (priority "conditional spontaneous" or "alarm") needs to be transmitted immediately.
- When the collected messages have not yet reached a volume of 202 bytes, but the TCP/IP keepalive interval expires.

Configuration

During configuration, communication- and connection-specific system settings are made for each device.

Counted value

A counted value (for example amount of flow) is acquired via a digital input as a pulse train and totaled to produce a binary-coded value. A counted value is 2 words: 28 bits for the binary-coded value 4 display bits.

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 in GSM for wireless transmission of data at 9 600 bps full duplex. Connections can be established to other GSM devices, to analog modems or to ISDN modems in the fixed network. The connection establishment can be started at both ends. Only dial-up connections are supported.

CTS

Clear to send

Signal in the data flow control

Data manager

The accrued data is managed by the data manager in WinCC. The data manager works with data created in the WinCC project and data stored in the project database. It handles the entire management of WinCC tags while WinCC is in runtime mode. All WinCC applications request the data from the data manager in the form of WinCC tags. These applications include Graphics Runtime, Alarm Logging Runtime and Tag Logging Runtime.

To allow WinCC to communicate with the widest variety of data sources (programmable controllers, ST7cc servers etc.), various communications drivers are used.

A communications driver is a C++ DLL that communicates with the data manager over a specified interface known as the channel API. The WinCC tags are supplied with process values over the communications driver.

Data messages

The actual transmission of data takes the form of data messages. These contain a fixed amount of a specific information type.
There are status messages, analog value messages, counted value messages, command messages, setpoint messages, parameter messages and data messages for the various information types.

DCF77 radio clock

DCF77 is a time signal transmitted on the normal frequency 77.5 kHz as encoded time information. Reception of the time signal is restricted to Western Europe.

Some TIM variants (TIM 4VD, TIM 42D, TIM 43D, TIM 44D, TIM 4RD) are equipped with a DCF77 receiver module that can receive the DCF77 time signal either over an indoor or outdoor antenna. The ST7 time management currently requires the existence of a DCF77 time signal to allow time synchronization of the stations and the control center throughout a network.

For applications that cannot receive the DCF77 time signal, SINAUT provides a GPS receiver module with a GPS outdoor antenna that converts the GPS Time signal for the DCF77 receiver module.

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.
Compare "inter-station communication"

Context: Telecontrol / SINAUT

DNP

Distributed Network Protocol

DNP3 protocol

The DNP3 protocol is a telecontrol protocol that allows the process data of sensors, field devices and RTUs to be connected to operator control and monitoring systems. The protocol was first specified in 1990. In 1993, this specification was released for general use under a public domain license. The protocol is maintained and has been further developed by the DNP user group founded in October 1994. The DNP user group is made up of users and manufacturers of DNP-compliant devices.

Apart from the continuous further development of the DNP specification, the DNP user group provides a test procedure for checking the conformity of DNP products. The certification of the DNP device using this procedure ensures the compatibility of devices from different vendors. The test procedure divides the DNP-compliant devices into different conformity levels according to their range of functions (simple sensors, field devices, complex controllers).

The SIMATIC NET modules TIM 3V-IE DNP3 and TIM 4R-IE DNP3 were certified with this test procedure according to DNP level 2 "outstation".

DNP3 TIM

→ *TIM*

DNS

Domain Name System, a distributed database that manages the name space in the Internet

DSL

Digital Subscriber Line

DSL is used mainly for transmission of telephone and Internet data. By using a higher bandwidth, data can be transmitted at transmission speeds in the two to three figure Mbps range.

Ethernet / Industrial Ethernet

Industrial Ethernet is a powerful communications network complying with the international standard IEEE802.3 (Ethernet) that was optimized to meet the requirements of industrial application. Ethernet is designed with a linear or star topology. The transmission media are shielded coaxial cables, twisted pair, or fiber-optic cables. SIMATIC NET Industrial Ethernet uses both the Ethernet and Fast Ethernet standards.

Ethernet TIM

→ *TIM*

Firewall

A firewall is a network component via which a secure network can be linked with an unsecure network. The task of a firewall is to control data exchange between the networks.

General request

With a general request (GR), subscribers in a SINAUT network can request a current process image from their communications partners. This happens 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 at any time by the user program or from the control center.

TD7onTIM does not support the general request.

Global script

The term Global script means all the C functions and actions that can be used throughout the project or over several projects. C actions are used in process control during runtime.

GPRS

General Packet Radio Service

GPRS is an expansion of GSM mobile wireless that adds packet-oriented data transmission. Network connections are established over GPRS either in the Internet or in private networks. The data is transmitted using the Internet protocols TCP/IP or UDP/IP.

GPS

Global Positioning System

GPS is a worldwide US satellite navigation system for highly accurate location, navigation and time distribution. It operates with 24 orbiting satellites (21 operational and 3 spare satellites) on 6 satellite orbits at a height of approximately 20,000 kilometers. Each satellite contains an atomic clock whose time is transmitted continuously along with the orbit data. The GPS receiver receives data from a maximum of six satellites and calculates its position based on these values. Once the position has been calculated, the transmission time of the data from the individual satellites can be calculated. The GPS world time (UTC) is calculated in the system based on these values.

Graphics Designer

The Graphics Designer is a vector-oriented drawing program for creating process pictures. Complex process pictures can be created using a wide variety of graphic objects from an object and style palette.

ST7cc Config uses the ODK interface of the Graphics Designer to create picture typicals for the SINAUT subscribers and to make these available to the WinCC configuration engineer in a sample picture for use in further process pictures.

GSM

Global System for Mobile Communications

Worldwide standard for wireless transmission of voice, data, fax and text messages (SMS). GSM-based wireless networks can be found in many countries throughout the world. A distinction is made according to frequencies: PCS 850 MHz, GSM 900 MHz, DCS 1800 MHz and PCS 1900 MHz.

Image memory / send buffer

A TIM has a send buffer and an image memory for buffering send messages.

Before it is forwarded to the communications partner, each message is entered in the send buffer. Once it has been sent, the message is cleared from the send buffer. A message is transmitted either according to the send buffer principle or the image memory principle. The user specifies the transmission principle during configuration of the data messages.

The storage capacity of the send buffer depends on the particular TIM type and ranges from 10 000 to 56 000 data messages.

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. The image memory therefore contains all data messages with their up-to-date content from the process.

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.

See also Image memory / send buffer

Inter-station communication

With inter-station communication, S7 stations communicate with each other by having an intermediary forward the frames to the relevant destination node.
Compare "direct communication"

Context: Telecontrol / SINAUT

IP

Internet Protocol

IP address

An IP address consists of a numeric code made up of four numbers from 0 to 255 (for example 192.168.0.55). It is the numeric address of a particular computer in the network / Internet.

ISDN

Integrated Services Digital Network

ISDN integrates various services in one transmission network. ISDN networks integrate telephone, telefax, teletext, Datex-J, video phones and data transfer. This makes a wide variety of digital services available to the user: Language, texts, graphics and other data.

LAN

Local Area Network

Local data traffic is possible in the SINAUT ST7 system via the LAN types "MPI" and "Industrial Ethernet".

In SINAUT, the LAN is also used as a "WAN", for example when communication via the Internet is required or when GPRS stations transfer their data onto the Internet to a master station via a GPRS network.

Local buffer

If the ST7cc server cannot forward its data to WinCC, all messages (ST7 data messages and organizational messages) are stored in the local buffer. Once WinCC becomes available again, the buffered messages are processed. This mechanism achieves two aims:

- Form the perspective of the stations, the master station can be reached when WinCC is not available.
- General requests resulting from temporary deactivation of WinCC are avoided.

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

Copper dedicated lines are highly susceptible to electromagnetic interference. The coupling of extraneous voltages can be inductive or capacitive, for example due to the effects of lightning. Direct conductive coupling is also possible due to bad insulation.
The LTOP overvoltage protection modules limit extraneous voltage and overvoltage to a non-critical level. The floating transformer also provides electrical isolation preventing coupling of voltages into other cable sections. An LTOP protects persons and investment and is therefore an indispensable safety element in private dedicated line networks.

Main and subcycle

The sequence of the polling cycle can be structured on the master TIM by assigning individual polling stations to a main cycle or a subcycle. The subcycle is always activated at the end of the main cycle; in other words, once all stations from the main cycle have been polled, a certain, selectable number of stations are polled in the subcycle. Following this, all the stations in the main cycle are polled again.

Master TIM

WAN interface of a TIM set to the "master" function. Use in control centers and node stations.

MPI

Multi Point Interface

MPI is the programming device interface of SIMATIC S7. Devices with an MPI interface (for example a TIM), can also communicate with each other (MPI bus).

MSC protocol

The MSC protocol is a proprietary protocol and is used in SINAUT ST7 for the data exchange between TCP/IP-compliant devices in Ethernet, landline or mobile wireless networks. The MSC protocol provides an authentication mechanism and simple encryption of data. To transfer messages in IP-based WANs, the MSC protocol is linked into the protocol of SINAUT ST7.

Multimaster polling with time slots

Context: SINAUT ST7

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 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. It is possible to have several node stations in a SINAUT network; in other words the data traffic between stations and master station can involve two or more node stations.

Node station TIM

WAN interface of a TIM in a node station set to the "node station" function.

If the TIM used in a node station has two WAN interfaces, the "station", "node station" and "master station" functions can be used in any combination on one TIM.

OP

Operator panel, HMI device

OPC

OLE for Process Control

OPC includes a series of specifications for data exchange in automation engineering between controllers, alarm transmitters etc. and control systems.

Organizational message

Organizational messages are used to execute organizational system functions.

These include, for example:

- General requests
- Time 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

Parameter assignment

A device is assigned the parameters it requires for starting up when it is supplied with data.

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 programmable controllers

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).

PLC

Programmable logic controller

Polling

→ *Polling mode*

Polling mode

Polling is a method of SINAUT data transmission in which a master TIM controls the data exchange in the telecontrol network.

Using a polling message, the master TIM instructs the connected stations (station TIMs) one after the other to transmit their stored data messages to the master TIM. If a polled station has no stored data messages, 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 message or, if block transfer was configured, several messages in a block. In the reply, the station indicates whether it still has other stored data messages. In this is the case, the station is then polled again immediately until all stored messages have been transmitted or until a selected maximum number of messages have been sent. The polling cycle then moves on to the next station.

Polling with time slots

Context: SINAUT ST7

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.

PROFIBUS

PROFIBUS is the open, internationally standardized (EN50170) bus system for process and field communication with field devices and for data communication within an automation cell. The uses of PROFIBUS range from production and process automation to building automation.

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.

Remote buffer

The remote buffer is set up only for redundant ST7cc. The ST7cc server recognizes whether or not the redundant mode is required based on the existence of the redundancy license.

The remote buffer is organized as a ring buffer and records all incoming messages so that it can be used as a data source for the redundant partner during a restart. If the partner of a redundant ST7cc system starts up again, it can recognize the time for which messages are missing and can request these from the redundancy partner. The remote buffer is necessary to ensure data consistency when using a redundant ST7cc system.

Requested message

Polled messages are data messages 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.

RJ-12

This describes a 6-pin connecting cable with a standardized modular (Western) connector.

RJ-45

This describes a 8-pin connecting cable with a standardized modular (Western) connector.

RS-232

RS-232 is a standard for serial (i.e. bit-by bit) data transmission with +12 V and -12 V signals. RS-232 is a Recommended Standard of the Electronic Industries Association. For the RS-232 interface, 9-pin and 25-pin connections with D-sub connectors are normal. These are sub-miniature connectors with a D-shaped face.

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

Send buffer

→ *Image memory / send buffer*

Send buffer principle

If a send message is transmitted using the send buffer principle, each time the message is transferred to the TIM, it is entered completely in the send buffer. 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.

See also Image memory / send buffer

Setpoint

A setpoint is a selected digital or analog value that is transmitted once after the value has been set. The entered value is recalculated when necessary. A setpoint is always transmitted as 1 word. At the receiving end, the setpoint can either the output directly to the process as an analog signal (for example to an external controller) or the value is made available to the local program for further processing (setpoint for internal controller, limit value, threshold value etc.).

Setpoint and command input are interlocked for safety reasons; in other words, a setpoint input cannot be made at the same time as a command input. In this case, the acquisition program recognizes an error. Neither the setpoint nor the command are transmitted.

SIM card

The SIM card is an identification card for a subscriber of a GSM mobile wireless network. (SIM: Subscriber Identity Module)

SIMATIC S7

Siemens automation system

Simple Internet communication

In SINAUT ST7, simple Internet communication means data exchange between TCP/IP-compliant devices in Ethernet, landline or mobile wireless networks using the MSC protocol.

SINAUT

SIemens Network AUTomation

Station control system or telecontrol system based on SIMATIC S7

SINAUT message

Message used by the SINAUT ST7 or SINAUT DNP3 protocol. SINAUT messages are divided into organizational messages and data messages.

ST7 messages 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.

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 protocol

This protocol is used in the SINAUT ST7 system for data transmission via classic WAN networks and IP-based networks.

SINAUT TD7 Library

The SINAUT TD7 software in the stations allows change-driven 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. When necessary, data messages can be given a time stamp.

The following variants of the TD7 software exist:

- **TD7onCPU**

  The SINAUT TD7 library consists of software blocks for the CPU. It can be run both on an S7-300 or an S7-400 CPU. There are only a few blocks intended specifically for the S7-300 or S7-400.

  If the DNP3 protocol is used, TD7onCPU is not supported.

- **TD7onTIM**

  This TD7 variant runs on the TIM. It is available for the Ethernet and DNP3 TIMs. It can be used in a TIM with S7-300 and in a standalone TIM 4R-IE with S7-400.

  TD7onTIM can only be used as an alternative to TD7onCPU. Simultaneous use in a station is not possible.

SMS

Short Message Service

The short message service in the GSM standard is used to transfer short text messages to mobile wireless 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

SNAP

Subnetwork Access Protocol

Spontaneous message

In SINAUT networks, messages are always transmitted spontaneously; in other words, messages are created and transmitted only when changes to process values occur or event-driven. These messages are known as spontaneous messages.

See also "Conditional spontaneous message" and "Unconditional spontaneous message"

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.

Squelch

Squelch

Squelch reduces the interfering noise in communications during breaks in transmission. This increases the signal-to-noise ratio.

ST7 message

→ *SINAUT message*

ST7 protocol

This protocol is used in the SINAUT ST7 system for transmitting process data over WANs.

ST7cc

SINAUT ST7cc is the ideal control center system based on SIMATIC WinCC for SINAUT ST7.
It is specially designed for event-driven and time-stamped data transmission in the SINAUT system. It avoids the possible loss of data that can occur with cyclic polling in WinCC. It also ensures the use of the correct event time supplied by the SINAUT stations for all WinCC messages and archive entries. The integrated process image contains all process data as well as the status of all SINAUT subscribers in the network and makes this data available directly to WinCC for fast transfer to the process image.

ST7cc Config

ST7cc Config allows the parameter values required for message processing to be specified. When the WinCC data framework is generated (in other words, message management), the individual WinCC messages are generated and imported into WinCC.

ST7cc tag

An ST7cc tag is a data section from the data area of a SINAUT object that is managed and processed as a separate information unit in the ST7cc server. The tags are processed, however, in ST7cc and in WinCC. When the tags are defined, different processing can be assigned to them depending on their type. A tag can contain both a process value as well as status information from system components. System components are the SINAUT subscribers.

ST7cc tag management

ST7cc tag management covers all ST7cc tags. The content of the ST7cc tags represents the current process image. WinCC writes and reads the ST7cc tags.

ST7sc

The SINAUT system allows the networking of SIMATIC stations with a control center over a WAN. This control center can also be a SIMATIC station or a PC-based control center, for example, WinCC with the SINAUT ST7cc add-on.

SINAUT ST7sc allows the manufacturers of control systems to connect to SINAUT without needing to integrate a SINAUT interface. Communication is over OPC: As an OPC server, SINAUT ST7sc forms the interface between the SINAUT system and a control system connected as an OPC client.

The OPC interface is also suitable for data exchange with other applications, for example, the Microsoft Office application Excel.

Station

In the SINAUT world, the term station includes the entire hardware components required for acquisition, processing and communication with other stations or a master station. A station can, for example, consist of a modem, a TIM and a programmable controller (in turn consisting of a CPU and I/O modules). A SINAUT station can also include several programmable controllers, or, in the case of a node station, several TIMs.

Station TIM

WAN interface of a TIM set to the "station" function. Use in stations.

Status message

A status message is a process status (for example pump on, valve open) or alarm (for example limit value exceeded). This is binary information with the values 0 or 1. 8 messages are put together to form a byte.

STEP 7

The basic STEP 7 software is the standard tool for the SIMATIC S7, SIMATIC C7 and SIMATIC WinCC automation systems.

Subnet mask

The subnet mask specifies which parts of an IP address are assigned to the network number. The bits in the IP address whose corresponding bits in the subnet mask have the value 1 are assigned to the network number.

TCO (TIM Connect)

The TCO component monitors the local TIMs connected over MPI or Ethernet, maps their most important status displays on ST7sc tags, forwards received messages for message decoding or transfers the messages to be sent to the relevant TIM for WAN communication.

TCP

→ *TCP/IP, UDP*

TCP/IP, UDP

Transmission Control Protocol / Internet Protocol, User Datagram Protocol

Network protocols for connecting two subscribers in the Internet. IP is the basic protocol.

- TCP is used to make the connection reliable and makes sure, for example, that data packets are passed on to the application in the correct order. In addition to the IP addresses, UDP and TCP also provide port numbers between 1 and 65535 with which the various services can be distinguished.
- UDP sends individual packets. These can arrive at the recipient in a different order from the order in which they were sent - they can even be lost.

On a Windows PC, the WINSOCK.DLL (or WSOCK32.DLL) is responsible for handling the two protocols.

TD7 software

→ *SINAUT TD7 Library*

TD7onTIM

→ *SINAUT TD7 Library*

TIM

Telecontrol Interface Module

The TIM transmission processor is a communications module that handles all data transmission functions provided by the SINAUT system independently. Depending on the type, the TIM has one or two WAN interfaces, an MPI interface or one or two Ethernet interfaces. Depending on the requirements, a variety of transmission equipment can be connected. The module is supplied in an S7-300 housing. Depending on the protocol being used, the TIM is available in the following basic variants:

- **SINAUT ST7**
  - TIM 3

    The TIM 3 series includes the older modules TIM 3V, TIM 32, TIM 33 and TIM 34, see release 05/2007 of the system manual SINAUT ST7:
    https://support.industry.siemens.com/cs/ww/de/ps/15931/man
    IDs 24621696 and 24619519
  - TIM 4

    TIM 4R / TIM 4RD with MPI connector both for installation as a CP in an S7-300 and as a stand-alone device for connection over MPI to one or more S7-400 and S7-300 PLCs. The TIM 4R / TIM 4RD has two serial WAN connectors for external modems for connection to identical or different WAN networks, for example dedicated line or telephone network.
    The TIM 4 series also includes the older modules TIM 42/42D, TIM 43/43D, TIM 44/44D and TIM 4V/4VD, see edition 05/2007 of the SINAUT ST7 system manual:
    https://support.industry.siemens.com/cs/ww/de/ps/15931/man
    IDs: 24621696 and 24619519
  - Ethernet TIM

    TIM 3V-IE variants with two WAN connectors via a serial interface and an RJ-45 Ethernet interface. Only for installation as a CP in S7-300.
    The TIM 3V-IE can handle SINAUT communication via the serial RS-232 interface or via the Ethernet interface.
    The TIM 3V-IE Advanced can use the RS-232/RS-485 interface at the same time as the Ethernet interface.

    TIM 4R-IE with four WAN connectors: Two RS-232/RS-485 and two RJ-45 Ethernet. Installation as a CP in an S7-300 or as standalone device in a separate TIM rack. The TIM 4R-IE can handle SINAUT communication simultaneously via all interfaces.
- **SINAUT DNP3**
  - The TIM 3V-IE DNP3 with two WAN connectors via a serial interface and an RJ-45 Ethernet interface. Only for installation as a CP in an S7-300. The TIM 3V-IE DNP3 can use the RS-232/RS-485 interface at the same time as the Ethernet interface.
  - TIM 4R-IE DNP3 with four WAN connectors: Two RS-232/RS-485 and two RJ-45 Ethernet. Installation as a CP in an S7-300 or as standalone device in a separate TIM rack. The TIM 4R-IE DNP3 can handle SINAUT communication simultaneously via all interfaces.

Topology

The topology describes the network structure. It specifies how a network (transmission medium and connectable devices or computers) is interconnected. Possible structures are linear (bus), star, ring, redundant ring and tree structure.

Transmission Control Protocol

Protocol for connection-oriented data transmission in networks; it belongs to the family of Internet protocols.

UART

Universal Asynchronous Receiver Transmitter

Hardware interface for sending / receiving data

UDP

→ *TCP/IP, UDP*

Unconditional spontaneous message

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.

See also "Conditional spontaneous message" and "Spontaneous message"

VPN

Virtual Private Network

Technology for secure transportation of confidential data in public IP networks, for example 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**

  SINAUT communication is possible via the following networks:

  – Private or leased dedicated lines

  – Analog telephone network

  – Digital ISDN network

  – Private wireless networks (directional or omnidirectional wireless and private mobile wireless with time slots)

  – Mobile wireless network

  A classic WAN is connected via suitable transmission device (modem) to a serial interface of the TIM.

- **WAN, IP-based**

  SINAUT communication is possible via the following IP-based networks:

  – Over wireless by using special wireless systems optimized for Ethernet, for example, the components of a SCALANCE W IWLAN system

  – Via fiber-optic cable, for example in conjunction with SCALANCE X switches with optical ports. This allows distances of up to 70 km to be covered.

  – Over public networks and Internet with DSL or GPRS

  – Over wideband systems such as OTN, PCM30 etc.

  An IP-based WAN is normally connected to an RJ-45 interface of an Ethernet TIM via an Ethernet-compliant module.

  In SINAUT ST7, the serial interface of a station TIM 3V-IE can be connected to a GSM network via the GSM modem MD720-3 to allow use of GPRS. After the necessary activation of the MSC protocol in STEP 7, the serial WAN interface behaves like an Ethernet interface.

WinCC

WinCC is a cross-branch and technologically neutral system for the solution of visualization and process control tasks in production and process engineering. It provides function modules for graphics display (Graphics Designer), for signaling (Alarm Logging), for archiving (Tag Logging) and logging.

WinCC API

As a completely open and expandable system, WinCC makes a comprehensive API (Application Program Interface) available. This involves an interface over which the user programs such as ST7cc Server and ST7cc Config can access WinCC. A comprehensive description is available with the WinCC ODK (Open Developers Kit).

WinCC buffer

Message and archive processing can be assigned to the ST7cc tags. If this is the case, individual messages or archive data are generated in ST7cc, that are transferred over the ODK interface to Alarm Logging or Tag Logging for further processing.

The results of processing an ST7cc tag can, however, accrue faster than they can be accepted by WinCC. The WinCC buffer takes the WinCC jobs from the ST7cc processing and therefore separates the asynchronous procedures of job creation and job processing.

WinCC channel DLL

To allow WinCC to communicate with the widest variety of data sources (programmable controllers, ST7cc servers etc.), various communications drivers are used.

A communications driver is a C++ DLL that communicates with the data manager over an interface known as the channel API that is specified by the data manager. The WinCC tags are supplied with process values over the communications driver.

WinCC tag

WinCC tags are the central elements that allow process values to be accessed. Within a WinCC project, they have a unique name and a data type. A WinCC tag is assigned to a logical connection that specifies which channel supplies the process values of the tags and over which connection.

For the WinCC tags whose data sources are the ST7cc tags, the channel DLL is the connection over which the ST7cc server supplies the process values.

The WinCC tags required for ST7cc can be generated with ST7cc Config.

WinCC Tag Logging

Tag Logging is used to receive data from running processes and to prepare it for display and archiving. The data formats and the acquisition times and archiving times can be set as required.

WinCC Tag Logging is computer time-oriented and not intended for the arrival of data with a delay offset as is the case with SINAUT ST7. This means that the ST7cc server must make certain archiving preparations for WinCC. The ST7cc server transfers the values to be archived to Tag Logging over the ODK interface. This ensures the chronological arrangement of the archive values even if process data is delivered by the ST7 stations, for example, with an offset of an hour.

WinCC Tag Management

Each element (in other words, every tag) used in WinCC is collected in Tag Management and managed there.

WinCC tag management

WinCC tag management covers all WinCC tags.

Index

N

O

P

R

S

T

W