

SIMATIC NET

Industrial Remote Communication - TeleControl SINAUT ST7 - Volume 3

System Manual

Preface

Functions and
requirements

1

Configuration

2

Block library TeleControl
ST7

3

Commissioning,
diagnostics and
maintenance

4

STEP 7 V5 configuration of
the proxy

A

OUC program blocks (CP)

B

SINEMA Remote Connect
(CP)

C

References

D

Volume 3: Configuration under STEP 7 Professional

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 Aktiengesellschaft. 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.

Preface

Validity of this manual

This configuration manual is valid for the following SIMATIC NET communications modules that support the ST7 telecontrol protocol:

- TIM 1531 IRC - V2.4
- CP 1243-8 IRC - V3.5
- CP 1542SP-1 IRC - V2.3
- TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE V2.8
Observe the notes below on product discontinuation.

You will find the required version of the configuration software in section Software requirements (Page 38).

You will find the device versions and the required configuration software products in the section Communications modules (Page 17).

New in this edition

- New firmware versions of the following communications modules (see also relevant operating instructions)
 - TIM 1531 IRC - V2.4
 - CP 1243-8 IRC - V3.5
 - CP 1542SP-1 IRC - V2.3

Expansion of functions of these three devices, including:

- Expansion of functions for time synchronization
- Expansion of telecontrol functions:
 - Option to transfer the time stamp of the CPU with the data point (TIM 1531 IRC, CP 1243-8 IRC)
 - Registration of tags on the CPU (Data point configuration > Read cycle)
 - Manual request of stations (TIM 1531 IRC / CP 1542SP1-IRC)
 - Expansion of parameters during data point export
- Configuration with STEP 7 Basic / Professional V19 Update 1

- Block library Telecontrol ST7 (TD7onCPU)
Block library V3.0:
 - New data point typicals without 1-out-of-n check (FC Safe not required) for the CPU 1500:
Cmd01B_FS / Cmd01B_FR / Set01W_FS / Par12D_FS
 - Accelerated general request (XGR) also via the blocks of the CPU 1500
 - ListGenerator1500 and ListGenerator400 with receipt of 10922 receive objects without destination object number

Block library V3.0 SP4 together with TIM 1531 IRC V2.4:

- Blocks for S7-300
- Editorial revision

Also observe the publications for new device firmware versions on the Internet pages of Siemens Industry Online Support:

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21764/dl>)

Replaced manual edition

Edition 12/2019

You will find the manual on the following Internet page of Siemens Industry Online Support:

Link: (<https://support.industry.siemens.com/cs/ww/en/view/109756253>)

Abbreviations/device names

The following terms and abbreviations are used in this manual:

- **Module / device / CP / TIM**
Names for the respective communications module
- **STEP 7**
This short form will be used below for the STEP 7 Basic / Professional configuration tool.
- **WBM**
"WBM" is the acronym for the "Web Based Management", the pages of the TIM 1531 IRC Web server for configuration and diagnostics data.
- **GR**
"GR" is the abbreviation for a "General request", with which the current process state of a station can be queried.
See also Glossary (Page 431) for information on this.
- **XGR**
"XGR" is the abbreviation for an "Accelerated general request".

Structure of the documentation

The documentation for the SIMATIC NET telecontrol communications modules consists of the following manuals in each case:

- Operating instructions or product manual
- Configuration manuals
(one configuration manual for each telecontrol protocol)
Only the configuration manual for the SINAUT ST7 protocol is listed below.

You can find the Internet links for the manuals in the References (Page 425).

In detail, the documentation of the device groups that use the protocol ST7 consists of the following documents:

Telecontrol CP

- **Operating instructions**
Valid for the respective CP
 - Application, functions, requirements (CPUs, software etc.)
 - Hardware description
 - Installation, wiring, commissioning, operation
 - Configuration
Description of only the configuration of the telecontrol-independent functions.
If you use telecontrol functions, read the respective configuration manual.
 - Diagnostics, maintenance
 - Technical specifications, approvals, accessories
- **SINAUT ST7 system manual volume 3**
 - Configuration and diagnostics in STEP 7 Professional (TIA Portal)
Exception (CP 1200):
The description of the configuration of cloud access of the CP 1243-8 ("CloudConnect" parameter group) is not contained in this manual, but in the operating instructions for the CP.

TIM 1531 IRC

- **Manual**
 - Application and functions
 - Requirements (CPUs, configuration software, etc.)
 - Hardware description
 - Installation, wiring, commissioning, operation
 - Diagnostics, maintenance
 - Technical specifications, approvals, accessories
- **SINAUT ST7 system manual volume 3**
 - Configuration and diagnostics in STEP 7 Professional (TIA Portal)

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

- **SINAUT ST7 system manual**

- Volume 1 - System and hardware**

- Hardware description, installation, wiring, commissioning, operation, technical specifications, approvals

- Volume 2 - Configuration under STEP 7 V5**

- Configuration and diagnostics in STEP 7 V5

- Volume 3 - Configuration under STEP 7 Professional**

- Configuration and diagnostics in STEP 7 Professional (TIA Portal)

Current manual edition and more information on the Internet

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

/1/ SINAUT ST7 system manual (Page 425)

You can find a range of application examples on the following page:

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/15915/ae>)

Required experience

Knowledge in the following areas is required for configuration and diagnostics of the devices:

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

Cross references

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

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

Product discontinuation

Note

Discontinuation of modules (product discontinuation)

Note the following products to be discontinued in October 2023.

- TS MODULE MODEM (6ES7972-0MM00-0XA0)
- TS MODULE RS232 (6ES7972-0MS00-0XA0)

For more information, see the product notifications about this on the Internet:

Link: (<https://support.industry.siemens.com/cs/us/en/pv/6ES7972-0MM00-0XA0/pi?dl=de>)

Link: (<https://support.industry.siemens.com/cs/us/en/pv/6ES7972-0MS00-0XA0/pi?dl=de>)

Product phase-out of SIMATIC S7-300

The "Product phase-out" milestone has been announced for the SIMATIC S7-300 system and the ET 200M I/O system with effect from October 2023.

Type discontinuation has been announced with effect from October 2025.

Read the product notifications about this on the Internet:

Link: (<https://support.industry.siemens.com/cs/ww/en/view/109809890>)

Discontinued products

Note

Discontinuation of modules (product discontinuation)

Note the following products to be discontinued.

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

- The following products have the status "Type discontinuation" as of 15 April 2015:

- TIM 4R (6NH7800-4AA90)
- TIM 4RD (6NH7800-4AD90)

Read the product notifications about this on the Internet:

Link: (<https://support.industry.siemens.com/cs/ww/en/view/109475169>)

- The following products have the status "Type discontinuation" as of 1 October 2016:

- Modem MD2
- Modem MD3
- LTOP1 / LTOP2 and accessories

Read the product notifications about this on the Internet:

Link: (<https://support.industry.siemens.com/cs/ww/en/view/109740147>)

- EGPRS router MD741-1

Read the product notifications about this on the Internet:

Link: (<https://support.industry.siemens.com/cs/ww/en/view/62607452>)

Documentation 09/2016 for TIM 4R / TIM 4RD and accessories

This edition of the system manual no longer contains information on the TIM 4R and TIM 4RD modules.

For information on these products, refer to the 09/2016 edition of the system manual, which is available on the Siemens Industry Online Support website:

- Volume 1
Link: (<https://support.industry.siemens.com/cs/ww/en/view/109745599>)
- Volume 2
Link: (<https://support.industry.siemens.com/cs/ww/en/view/109748055>)

Documentation 05/2007 for SINAUT ST1 and older ST7 modules

This edition of the system manual no longer contains 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 Edition 05/2007 of this manual, which you can find on the website of Siemens Industry Online Support:

- Volume 1
Link: (<https://support.industry.siemens.com/cs/ww/en/view/24621696>)
- Volume 2
Link: (<https://support.industry.siemens.com/cs/ww/en/view/63112659>)

License conditions

Note

Open source software

The products contain open source software. Read the license conditions for open source software carefully before using the products.

The operating instructions of the relevant product provide information on finding the license conditions.

Cybersecurity information

Siemens provides products and solutions with industrial cybersecurity 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 cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

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

For additional information on industrial cybersecurity measures that may be implemented, please visit
<https://www.siemens.com/cybersecurity-industry>.

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

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under
<https://new.siemens.com/cert>.

SIMATIC NET glossary

The SIMATIC NET glossary describes terms that may be used in this document.

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:

- On the accompanying DVD
- In the Siemens Industry Online Support at the following address:
Link: (<https://support.industry.siemens.com/cs/ww/en/view/50305045>)

A SINAUT-specific glossary can be found in the appendix of this manual.

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>)

Table of contents

	Preface	3
1	Functions and requirements.....	17
1.1	Communications modules.....	17
1.2	Communication via the SINAUT ST7 protocol.....	18
1.3	Mobile wireless and Internet communication.....	21
1.4	PG routing	23
1.5	Performance data and configuration limits	23
1.5.1	TIM 1531 IRC	23
1.5.2	CP 1542SP-1 IRC	29
1.5.3	CP 1243-8 IRC.....	30
1.5.4	TIM 3V-IE / TIM 4R-IE	31
1.6	TD7 software: Use and variants	32
1.7	Overview - Connection options of the modules.....	35
1.8	General request and accelerated general request.....	37
1.9	Software requirements	38
1.10	Hardware requirements	39
1.10.1	CP 1542SP-1 IRC	39
1.10.2	TIM 1531 IRC	41
1.11	Expansion of SINAUT projects.....	41
1.11.1	Modules for new SINAUT projects and those to be expanded	41
1.11.2	Requirements for the expansion	43
2	Configuration.....	45
2.1	Security recommendations.....	45
2.2	Configuration - Overview	48
2.3	Configuring and copying modules	52
2.4	Compatibility mode of the CP 1243-8 IRC V3.0	52
2.5	MPI address	53
2.6	Communication types	53
2.7	Basic settings	56
2.7.1	Configuration.....	56
2.7.2	Retentive saving of events.....	57
2.7.3	IP routing.....	57
2.8	Options	58
2.9	Networking, configuration of networks and network nodes	60
2.9.1	Networking of the interfaces.....	60

2.9.2	WAN settings of the interfaces	63
2.10	Ethernet interface	65
2.10.1	Ethernet interface > address parameters	65
2.10.2	Advanced options	66
2.10.2.1	MSC protocol settings	66
2.10.2.2	TCP connection monitoring	69
2.10.2.3	Transmission settings	70
2.10.2.4	MTU size	70
2.10.3	Web server access	70
2.10.3.1	CP	70
2.10.3.2	TIM 1531 IRC	71
2.11	Serial interface of the TIM	71
2.11.1	WAN parameters	71
2.11.2	Advanced options - MSC settings	71
2.11.3	Advanced options - Dedicated line	72
2.11.4	Advanced options - Dial-up network	74
2.12	Serial interface of the CP 1243-8 IRC	78
2.12.1	Configuration of the serial interface	78
2.12.2	Serial connection for TS module	79
2.12.3	Configuring a TS module	80
2.12.4	WAN settings	81
2.12.5	WAN parameters (networking the CP)	81
2.12.6	Advanced options	83
2.13	Configuring WAN networks	83
2.14	Web server (TIM 1531 IRC)	86
2.15	Web diagnostics of the TIM 1531 IRC	87
2.16	Partner stations	88
2.16.1	Importing configuration data	88
2.17	Communication with the CPU	94
2.17.1	CP diagnostics	95
2.17.2	Partner status and Path status	96
2.17.3	Manual request	101
2.17.4	Local operator input (TIM 3V-IE / TIM 4R-IE)	103
2.18	Time-of-day synchronization	105
2.19	Time stamp	113
2.20	SMSC	114
2.21	I/O addresses (TIM 3V-IE / TIM 4R-IE)	114
2.22	MSC authentication	114
2.23	E-mail configuration	115
2.24	Subscriber numbers	117
2.25	Secure communication between CPU and module	118
2.26	SNMP	120
2.27	Security / Certificates	121

2.27.1	Security user	121
2.27.2	MSC authentication	123
2.27.3	SMSC	124
2.27.4	Log settings - Filtering of the system events	125
2.27.5	VPN	125
2.27.5.1	CP as passive subscriber of VPN connections	125
2.27.5.2	VPN (Virtual Private Network)	125
2.27.5.3	Creating a VPN tunnel for S7 communication between stations	127
2.27.5.4	VPN communication with SOFTNET Security Client (engineering station)	128
2.27.5.5	Establishment of VPN tunnel communication between the CP and SCALANCE M	129
2.27.6	SNMP	129
2.27.7	Certificate manager	131
2.27.8	Handling the TLS certificates	132
2.28	Protection (TIM 1531 IRC)	137
2.28.1	Protection	138
2.28.2	Configuring access protection	139
2.29	Creating telecontrol connections for the ST7 communication	140
2.30	Data points	147
2.30.1	Data point configuration	147
2.30.2	Datapoint types	155
2.30.3	ST7 data points	158
2.30.4	"General" tab	160
2.30.5	Process image, type of transmission, event classes	161
2.30.6	Read cycle	163
2.30.7	"Local setpoint input" tab	164
2.30.8	"Trigger" tab	167
2.30.9	Threshold value trigger	170
2.30.10	Analog value preprocessing	172
2.30.11	Command options	179
2.30.12	Masks	182
2.30.13	Reset	183
2.30.14	Partner stations	183
2.31	Messages	183
2.32	Character set for user names, passwords and messages	189
3	Block library TeleControl ST7	191
3.1	Version information and validity	191
3.2	Creating a library	191
3.3	Structure of the library	193
3.4	Using, calling and generating the blocks	194
3.5	Note on the "BasicTask_*" FCs	196
3.6	Structure of user program for TD7onCPU	197
3.6.1	The cyclic SINAUT program	197
3.6.2	Cyclic interrupt OB	201
3.6.3	Programming error OB	201
3.7	Types	202
3.7.1	PLC data types (UDT)	202

3.7.2	System data types (SDT).....	203
3.8	Master copies	203
3.8.1	Data point typicals	204
3.8.1.1	Types and overview of the data point typicals	204
3.8.1.2	Reoccurring parameters	207
3.8.1.3	Time stamp.....	215
3.8.1.4	Analog value typical Ana04R_S	216
3.8.1.5	Analog value typical Ana04R_R	220
3.8.1.6	Analog value typical Ana04W_S	221
3.8.1.7	Analog value typical Ana04W_R	229
3.8.1.8	Binary value typical Bin04B_S.....	229
3.8.1.9	Binary value typical Bin04B_R.....	232
3.8.1.10	Binary value typical Bin08X_S.....	233
3.8.1.11	Binary value typical Bin08X_R	234
3.8.1.12	Command typical Cmd01B_S	235
3.8.1.13	Command typical Cmd01B_FS.....	237
3.8.1.14	Command typical Cmd01B_R	240
3.8.1.15	Command typical Cmd01B_FR.....	243
3.8.1.16	Command typical Cmd08X_S	244
3.8.1.17	Command typical Cmd08X_R	245
3.8.1.18	Counted value typicals Cnt01D_S / Cnt04D_S	247
3.8.1.19	Counted value typicals Cnt01D_R / Cnt04D_R	250
3.8.1.20	Data typical Dat12D_S.....	253
3.8.1.21	Data typical Dat12D_R	257
3.8.1.22	Data typical Dat12x1D_S.....	258
3.8.1.23	Data typical Dat12x1D_R.....	264
3.8.1.24	Data typical Dat256D_S.....	267
3.8.1.25	Data typical Dat256D_R	273
3.8.1.26	Parameter typical Par12D_S.....	279
3.8.1.27	Parameter typical Par12D_FS.....	288
3.8.1.28	Parameter typical Par12D_R	296
3.8.1.29	FC Safe	302
3.8.1.30	Setpoint typical Set01W_S.....	306
3.8.1.31	Setpoint typical Set01W_FS.....	310
3.8.1.32	Setpoint typical Set01W_R.....	315
3.8.2	Optional blocks.....	319
3.8.2.1	ListGenerator1500/300/400 FC	319
3.8.2.2	FC PartnerMonitor	321
3.8.2.3	FC PartnerStatus	326
3.8.2.4	FC PathStatus.....	328
3.8.2.5	FC PulseCounter.....	331
3.8.2.6	FC ST7ObjectTest	333
3.8.2.7	FC TestCopy	334
3.8.2.8	DB TestCopyData.....	335
3.8.2.9	Operation of TestCopyMonitor	338
3.8.2.10	FC TimeTask.....	341
3.8.2.11	FC Trigger	342
3.8.3	System blocks.....	347
3.8.3.1	BasicTask_* FC.....	347
3.8.3.2	DB BasicData.....	348
3.8.3.3	FC Create.....	349
3.8.3.4	Diagnose / Diagnostics FC	349

3.8.3.5	FC Distribute	349
3.8.3.6	FC Search	350
3.8.3.7	FC Startup.....	350
3.8.4	Communication blocks BCom (CPU1500)	350
3.8.4.1	FB BCom	350
3.8.4.2	BConnect FB	350
3.8.4.3	DB BConnectData.....	351
3.8.5	Communications block FB BCom (CPU300/CPU400).....	351
3.8.5.1	BCom FB.....	351
3.8.5.2	DB BComData	352
3.8.6	Communication blocks PCom (CPU300)	353
3.8.6.1	FB PCom	353
3.8.6.2	DB PComData	353
3.8.7	Communication blocks XCom (CPU300)	353
3.8.7.1	FB XCom.....	353
3.8.7.2	DB XComData	354
4	Commissioning, diagnostics and maintenance	355
4.1	Commissioning ST7: General requests	355
4.2	Diagnostics	356
4.2.1	Diagnostics options.....	356
4.2.2	Online security diagnostics via port 8448 of the CP.....	359
4.2.3	Web server S7-1200: Connection establishment	359
4.2.4	Telegram protocol diagnostics	360
4.2.4.1	Message protocol: Structure and functions	360
4.2.4.2	Details	362
4.2.5	SINAUT special diagnostics.....	363
4.2.5.1	TIM diagnostics.....	363
4.2.5.2	TIM subscriber diagnostics	371
4.2.5.3	TD7onTIM diagnostics	376
4.2.6	Messages in the diagnostic buffer of the TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE.....	378
4.2.6.1	Introduction.....	378
4.2.6.2	Diagnostic messages of the TIM	379
4.2.6.3	Meaning of ERRCLS and ERRCOD	391
4.2.7	SNMP	392
4.2.8	Processing status of the messages (SMS / e-mail).....	393
4.3	Maintenance.....	395
4.3.1	Loading firmware - CP 1200	395
4.3.2	Update firmware (TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE).....	397
4.3.3	Update firmware (TIM 1531 IRC).....	398
4.3.4	Reset to factory settings (TIM 1531 IRC)	400
4.3.5	Module replacement (TIM 1531 IRC)	402
A	STEP 7 V5 configuration of the proxy	403
A.1	Configuration in STEP 7 V5	403
A.2	Proxies in HW Config	404
A.3	Special features of the PROXY TIM 1531 IRC.....	405
A.4	Special features of the PROXY CP1243-8 IRC	405
A.5	SINAUT configuration.....	409

A.6	Exporting configuration data	410
B	OUC program blocks (CP)	413
B.1	Program blocks for OUC	413
B.2	Changing the IP address during runtime	416
C	SINEMA Remote Connect (CP)	419
C.1	Validity and requirements	419
C.2	Connection to SINEMA RC	419
C.3	Telecontrol via SINEMA RC	421
C.4	Security > VPN > SINEMA Remote Connect	421
D	References	425
D.1	/1/ SINAUT ST7 system manual	425
D.2	/2/ TeleControl configuration manuals	426
D.3	/3/ TIM 1531 IRC	426
D.4	/4/ CP 1243-8 IRC	426
D.5	/5/ CP 1542SP-1 IRC	426
D.6	/6/ RTU3000C	426
D.7	/7/ SINAUT ST7cc	427
D.8	/8/ SINAUT ST7sc	427
D.9	/9/ SCALANCE M870	427
D.10	/10/ SCALANCE S	427
D.11	/11/ S7-1200 system manual	427
D.12	/12/ ET 200SP system manual	428
D.13	/13/ ET 200SP Manual Collection	428
D.14	/14/ S7-300 automation system	428
D.15	/15/ MODEM MD720	428
D.16	/16/ TS Adapter	428
D.17	/17/ MODEM MD2	429
D.18	/18/ MODEM MD3	429
D.19	/19/ MODEM MD4	429
D.20	/20/ SNMP	429
	Glossary	431
	Index	445

Functions and requirements

1.1 Communications modules

Communications modules for the telecontrol protocol ST7

The following SIMATIC NET communications modules can be used for the telecontrol protocol SINAUT ST7.

You will find the TIM 4R-IE with article number 6NH7800-4BA00 twice in the STEP 7 catalog, as built-in unit (S7-300 rack) and as stand-alone device.

Meaning of symbols in the table:

- X = Supported
- - = Not supported

Table 1-1 Communications modules for ST7

Module Article number	Number of inter- faces *		Station type			STEP 7 prod- uct	Required firmware
	IE	RS	Master	Node station	Station		
TIM 1531 IRC 6GK7543-1MX00-0XE0	3	1	X	X	X	STEP 7 Pro- fessional	V2.4
TIM 4R-IE 6NH7800-4BA00	2	2	X	X	X	STEP 7 Pro- fessional	V2.8
TIM 3V-IE Advanced 6NH7800-3CA00	1	1	X	X	X	STEP 7 Pro- fessional	V2.8
TIM 3V-IE 6NH7800-3BA00	1	1	X	X	X	STEP 7 Pro- fessional	V2.8
CP 1243-8 IRC 6GK7243-8RX30-0XE0	1	1 **	-	-	X	STEP 7 Basic	V3.5
CP 1542SP-1 IRC 6GK7542-6VX00-0XE0	1	-	-	-	X	STEP 7 Pro- fessional	V2.3

* IE = Ethernet interfaces, RS = serial interfaces

** Serial interface of the CP 1243-8 IRC via the connector of a TS module

You can find notes on RTU3000C in the section Communication types (Page 53).

Notes on the table

Notes on the columns:

- **Station type "Node station"**

A node station is located in the plant hierarchy between the master station and other lower-level stations. The module requires at least two interfaces.

In the configuration, the "network node type" of the interface connected to the master station is configured as "Node station". See section WAN settings of the interfaces (Page 63) for more on this.

- **Firmware**

Unless indicated otherwise, the described functions apply to all devices named here. Individual functions are not supported by all devices.

The required firmware versions of the modules relate to the complete configuration described in this manual. You can find the required STEP 7 version for this in the section Software requirements (Page 38).

Modules with lower firmware versions can be configured in older STEP 7 versions with a deviating scope of functions.

1.2 Communication via the SINAUT ST7 protocol

The ST7 protocol

Network types

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

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

- Classic WAN networks
- IP-based WAN networks

Subscriber types

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

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

Connections

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

Communication via single or redundant telecontrol connections

The ST7 communication between two subscribers runs via configurable paths. To do this, telecontrol connections are created in the "Network data" editor in the "Telecontrol" tab, see section Creating telecontrol connections for the ST7 communication (Page 140).

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

To check the reachability of a communications partner of the module, you can write the connection status to "PLC tags for partner status / path status", see section Communication with the CPU (Page 94).

For redundant telecontrol connections, the behavior is as follows:

- **Main path**
If the connection between the two subscribers is undisturbed, the main path is always used. The status of the main path is rechecked each time there is communication and saved in the PLC tag.
- **Substitute path**
If the connection via the main path is disturbed, communication is switched to the substitute path.
The status of the substitute path, on the other hand, is only registered when communication is established.
If you configure a dialup network connection as the substitute path via which generally a connection is seldom established, you can test the reachability via this path for example by synchronizing the time of day once daily via this path. To do this a connection must be established once daily via the dialup network. The reachability is then updated once daily in the PLC tag.

Addressing communications partners and network nodes

Since the communication between two ST7 subscribers can run via different paths and subnets, precise addressing of the individual subscribers in the ST7 network is necessary. Addressing must be unique within a subnet and within the STEP 7 project.

If you want to use subscriber numbers/station addresses twice in different subnets, you must create two STEP 7 projects.

1.2 Communication via the SINAUT ST7 protocol

The following two parameters are used for addressing in a SINAUT ST7 network:

- **Subscriber number**

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

- **TIM**

Communication module for the ST7 protocol

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

Communication module for the ST7 protocol

- **CPU**

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

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

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

The TIM 1531 IRC can be assigned a CPU of the following SIMATIC families: S7-300, S7-400, S7-1500, ET 200SP

When you assign the TIM 1531 IRC to an S7-1500R/H CPU, the system IP address of the S7-1500R/H CPU is used for communication of the TIM with the CPU.

- **Application of the master station PC**

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

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

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

- **WAN address**

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

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

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

1.3 Mobile wireless and Internet communication

Alternative configurations

The following options are available for communication of the TIM via the Internet and, if applicable, the mobile network:

- **Highly secure communication**
Highly secure communication over mobile wireless network and Internet with SCALANCE M87x
- **Secure communication**
Secure communication via GPRS + Internet or only via Internet/DSL with the MSCsec protocol
- **Simple communication with basic protection mechanisms**
Simple communication via GPRS + Internet or only via Internet/DSL with the MSC protocol

Highly secure communication with SCALANCE M

Communication over a mobile wireless network in combination with the Internet is made possible with a SCALANCE M87x router.

Simple or secure Internet communication with the MSC or MSCsec protocol

The protocols MSC and MSCsec

- **MSC**
The MSC protocol is used for "simple Internet communication with basic protection mechanisms".
The MSC protocol supports authentication of the communications partners and simple encryption of data. A user name and a password are included in the encryption.
An MSC tunnel is established between the MSC station and MSC master station.
- **MSCsec**
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, the shared automatically generated key is renewed between the communications partners in a configurable key exchange interval.

Network node types and usable devices

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

- MSC master station
The following can be used as the MSC master station:
 - TIM 4R-IE / TIM 1531 IRCEthernet interface 1 of the TIM is connected to the Internet via a DSL router.
- MSC node station
 - TIM 3V-IE Advanced / TIM 4R-IE / TIM 1531 IRC
 - CP 1243-8 IRC
- MSC station
 - TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE / TIM 1531 IRC
 - CP 1243-8 IRC, CP 1542SP-1 IRC

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.
- **GSM network and Internet**
 - TIM
Only the MODEM MD720 can be used to connect the MSC station/node station to the GSM network and downstream to the Internet. The MD720 operates in terminal mode and as default is controlled by the application of the connected TIM.
With the TIM 4R-IE, the switchable WAN 1 interface is configured as the Ethernet interface during interface configuration.
 - CP 1243-8 IRC
The MODEM MD720 can also be connected to an S7-1200 via the CP.

Configuring wireless communication with the MSC protocol

1. Network the interfaces.
For a TIM 3V-IE Advanced, use the switchable WAN interface X1, because only this interface uses the MSC protocol when setting the WAN type "IP-based".
2. Set the appropriate network node type for each interface (master station/node station/station).
3. You configure the connection to the wireless network in the MSC protocol settings (Page 66) parameter group.
4. Create the telecontrol connection from the master station to the station.

1.4 PG routing

PG routing between telecontrol modules

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

Requirement for CPs

The "Online functions" options are enabled in the "Communication types" parameter group.

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

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

* As of STEP 7 V18, it may be possible that CPU data cannot be loaded via CP routing when using the MSC protocol. In this case use a direct connection to the CPU to load data to the CPU.

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

1.5 Performance data and configuration limits

1.5.1 TIM 1531 IRC

Telecontrol communication of the TIM 1531 IRC

Network types

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

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

1.5 Performance data and configuration limits

- Mobile wireless networks
 - GSM / GPRS (2G) with MODEM MD720 (only ST7)
 - UMTS (3G) / HSPA+ with SCALANCE M router
 - LTE (4G) with SCALANCE M router
- IP-based wireless networks

You will find an overview of the transmission paths and network types, as well as the interfaces that can be used in each case, in the relevant configuration manual.

Telecontrol communication

For telecontrol communication via telecontrol networks, the TIM uses one of the above-named protocols on the application layer (OSI layer 7).

The following functions and services are supported:

- **Communication with the control center**
The TIM communicates via LAN or WAN with an application in the master station.
You will find the supported master station types in the manual /3/ TIM 1531 IRC (Page 426).
- **Direct communication**
In dial-up networks, mobile wireless networks and Ethernet networks, there is direct communication between the subscribers.
- **Inter-station communication**
In dedicated line networks and with communication via the Internet with a mobile wireless network (GSM/MSC), the TIM supports inter-station communication between S7 stations via the master station.
With inter-station communication, the TIM establishes a connection to the master station. The master station forwards the frames to the destination station.
- **Messages: SMS / e-mail**
When configurable events occur, the TIM can send SMS messages to cell telephones and e-mails to PCs with an Internet connection.
 - When the MSC protocol is used, SMS messages can be sent if the TIM is connected to a mobile wireless network via the serial interface and a GSM or GPRS module (MODEM MD720).
 - If the TIM is connected, e-mails can be sent via the Ethernet interface.
You configure the e-mail protocol to be used and the server access in STEP 7 in the parameter group "E-mail configuration".

You configure the messages in the message editor. The use of program blocks is not necessary for this.

Protocols with security functions

- **DNP3 / IEC**
Secure TLS connections can be established under these protocols.
- **MSC**
For secure telecontrol communication, the transmission protocol "MSC" (OSI layer 3) is available for the ST7 telecontrol protocol. MSC can be used for communication between two TIM modules (not between the TIM and a master station application).
MSC is IP-based and can be used in the following networks:
 - Ethernet
 - Internet (DSL)
 - Mobile wireless network (GSM) and InternetThe following variants of the protocol are available:
 - MSC
Simple Internet communication via the Internet (DSL)
 - MSCsec
Secure Internet communication when security requirements are higher.
- **IPsec / VPN (via router SCALANCE M)**
VPN stands for highly secure communication via mobile wireless and the Internet (DSL) using a SCALANCE M mobile wireless router.
Alternatively, TLS connections can be used under DNP3 and IEC.

For a description of the protocols, refer to the configuration manual.

S7 communication

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

- **PG communication**
Communication with an engineering station
- **PG routing**
S7 communication across subnetwork boundaries
- **PUT/GET**
To exchange data with a local S7-300/400 CPU the TIM supports communication using the program blocks PUT/GET as client and server.
- **READ/WRITE**
To exchange data with a local S7-1500 CPU the TIM supports communication using the program blocks READ/WRITE as client and server.

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

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

Web server connections (WBM)

- **HTTP/HTTPS**

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

The Web pages display diagnostics information and configuration data.

You will find information on the content and operation in the manual /3/ TIM 1531 IRC (Page 426).

Telecontrol connections

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

Note that redundant connection paths of a connection between two partners require two connection resources on each partner.

The maximum number of connections depends on the protocol used.

Telecontrol connections - ST7

Max. number of connections: 128

When the ST7 protocol is used, distribution over the 4 interfaces depends on the network node type of the interface, the WAN type and the network type.

Refer to the following tables for the connection configuration limits of the TIM with use in the plant hierarchy as master station, as node station and as station.

Table 1-2 ST7 connections: TIM 1531 IRC as master station

Max. number overall	Network node type	WAN type	Network type	Max. number	Interface
128 *	Master station	IP-based	MSC	127	Ethernet X1
			<ul style="list-style-type: none">• Neutral (S7)• Mobile wireless via VPN	128	Ethernet X1 / X2 / X3
		Classic WAN	<ul style="list-style-type: none">• Dedicated line• Dialup network	32	Serial X4

Table 1-3 ST7 connections: TIM 1531 IRC as node station

Max. number overall	Network node type	WAN type	Network type	Max. number	Interface
128 *	Master station (Interface with connection to the station)	IP-based	MSC	127	Ethernet X1
			<ul style="list-style-type: none"> Neutral (S7) Mobile wireless via VPN 	128 *	Ethernet X1 / X2 / X3
		Classic WAN	<ul style="list-style-type: none"> Dedicated line Dialup network 	32	Serial X4
	Node station (Interface with connection to the master station)	IP-based	MSC	1	Ethernet X1
			<ul style="list-style-type: none"> Neutral (S7) Mobile wireless via VPN 	128 *	Ethernet X1 / X2 / X3
		Classic WAN	<ul style="list-style-type: none"> Dedicated line Dialup network 	16	Serial X4

Table 1-4 ST7 connections: TIM 1531 IRC as station

Max. number overall	Network node type	WAN type	Network type	Max. number	Interface
128	Station	IP-based	MSC	1	Ethernet X1
			<ul style="list-style-type: none"> Neutral (S7) Mobile wireless via VPN 	128 *	Ethernet X1 / X2 / X3
		Classic WAN	<ul style="list-style-type: none"> Dedicated line Dialup network 	16	Serial X4

*** TIM 1531 IRC with S7-300-CPU and TD7onCPU**

If you use TD7onCPU in the TIM 1531 IRC and assign an S7-300 CPU to the TIM, the S7-300 CPU supports up to 16 S7 connections.

Telecontrol connections - DNP3

Max. number of connections: 128

Distribution over 4 interfaces can take place in any way (max. 128 per interface).

Telecontrol connections - IEC 60870-5-101/104

Max. number of connections: 128

Distribution over 4 interfaces can take place in any way (max. 128 per interface).

Further connection resources

- **E-mail / SMS**
At runtime, a connection can be established to send an e-mail or SMS message.
- **S7 connections**
 - When using ST7 communication:
Max. 132 connection resources in total incl. telecontrol connections (see above) and incl. max. 4 PG/OP connections (see below)
 - Without using ST7 communication:
Max. 4 connection resources only for PG/OP connections (see below)
- **PG/OP connections**
4 connection resources for connections to the engineering station or HMI devices (included in the configuration limits of the S7 connections, see above)
- **PG routing**
Max. 4 connections at the same time
- **Online functions**
See PG/OP connections
- **HTTP/HTTPS**
Max. 2 connections per Ethernet interface
- **Connectable control center applications**
Number of ST7cc / ST7sc SINAUT control centers connectable via Ethernet: Max. 4

Number of data points for the data point configuration

The maximum number of configurable data points is 3000.

Frame memory

Send buffer

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

Due to the different frame structure in the various telecontrol protocols, the send buffer has the following maximum size depending on the protocol:

- ST7: 100000 frames
- DNP3: 250000 events
- IEC: 250000 events

The size of the frame memory is divided equally among all configured communications partners.

You will find details of how the send buffer works (storing and sending events) as well as the options for transferring data in the description of configuring the data points, see /3/ TIM 1531 IRC (Page 426).

SD card

You can set saving of frames on an optional SD card in the "Basic settings", see /3/ TIM 1531 IRC (Page 426).

The maximum number of frames that can be buffered on an SD card is the same as that in the send buffer (see above).

Messages: E-mail / SMS

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

- Number of characters per SMS message
Maximum number of characters that can be transferred per SMS message: 160 ASCII characters including any value sent at the same time
- Number of characters per e-mail
Maximum number of characters that can be transferred per e-mail: 256 ASCII characters including any value sent at the same time

1.5.2 CP 1542SP-1 IRC**Information in the operating instructions**

You will find the following information on configuration limits in the operating instructions of the CP:

- Number of CPs per station (2) and slot rules
- Connection resources
- Number of data points
- Frame memory (send buffer)
For details and the data transmission options, see section Process image, type of transmission, event classes (Page 161).
- E-mail (10)
- IPsec tunnel (VPN) (4)
- Firewall rules

For information on the operating instructions, see /5/ CP 1542SP-1 IRC (Page 426).

See also

Communication with the CPU (Page 94)

1.5.3 CP 1243-8 IRC

Information in the operating instructions

You will find the following information on configuration limits in the operating instructions of the CP 1243-8:

- Number of CMs/CPs per station
- Connection resources
- Number of data points (500)
- Frame memory (send buffer) (64000)
For information on setting the size in STEP 7, refer to the section Communication with the CPU (Page 94).
For details and the data transmission options, see section Process image, type of transmission, event classes (Page 161).
- E-mail (10)
- IPsec tunnel (VPN) (8)
- Firewall rules

For information on the operating instructions, see /4/ CP 1243-8 IRC (Page 426).

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

Connection resources

- **Telecontrol connections**

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

Note that redundant connection paths of a connection between two partners require two connection resources on each partner.

The maximum number of ST7 telecontrol connections is:

- TIM 3V-IE
12
+ max. 1 MSC connection
- TIM 3V-IE Advanced
24
+ max. 1 MSC connection
- TIM 4R-IE
64
+ max. 1 MSC connection
+ max. 128 MSC connection as master station
Total S7 + MSC connections as station: 65
Total S7 + MSC connections as master station: 128

- **S7 connections**

Max. number of configurable S7 connections:

- TIM 3V-IE
8
Of which:
2 PG connections
8 OP connections
- TIM 3V-IE Advanced
24
Of which:
4 PG connections
20 OP connections
- TIM 4R-IE
64
Of which:
2 PG connections
62 OP connections

- **Connectable control center applications**

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

- **E-mail / SMS**

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

- **Online functions**

See PG/OP connections

Size of the frame memory (send buffer)

Max. number of frames that can be saved:

- TIM 3V-IE
16000
- TIM 3V-IE Advanced
32000
- TIM 4R-IE
56000

1.6 TD7 software: Use and variants

Tasks 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:

- **TD7onTIM**

TD7onTIM is part of the TIM firmware; it is configured in the TIM and runs in productive mode in the TIM. You perform the configuration in STEP 7 Professional.

Advantages of TD7onTIM are:

- Easy to configure
- Independence of the TD7 software from the CPU program
- Relief of the CPU when the CPU memory is low on reserves
TD7onTIM needs no or very little work memory on the CPU.

- **TD7onCPU**

TD7onCPU is implemented via program blocks for the CPU. The block library is available under the following name:

Telecontrol ST7

Configuration tool: STEP 7 Professional (TIA Portal)

The following blocks are located under the "Master copies":

- CPU300/400
Program blocks for the S7-300/400 CPU
- CPU1500
Program blocks for the S7-1500 CPU

Advantages of TD7onCPU are:

- High speed if data changes
Changes are detected with every CPU cycle.
- Larger selection of data typicals for the master station and the station

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.

In addition to process data transmission via WAN, the TD7 blocks are also suitable for local communication between CPUs, if they are connected to each other via MPI (S7-300). Here, as well, local connections and CPUs are constantly monitored and there is an automatic update after startup or elimination of disturbances.

Specification of the TD7 version used

Only one of the two TD7 versions can be used in a station. You define the version via the configuration:

- **TD7onTIM**

Configure the TD7 software for the TIM of this station through the parameter groups of the TIM. Do not use program blocks of the TD7 block library for this station.

- **TD7onCPU**

Use the program blocks of the "Telecontrol ST7" global library. Do not configure TD7onTIM for this station.

The TD7onCPU libraries

TD7onCPU is available in two block libraries for the following SIMATIC families:

- S7-1500
The blocks can run on an S7-1500.
- S7-300 / 400
The blocks can run on an S7-300 and on an S7-400 CPU. Exceptions will be pointed out explicitly.
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 update after startup or elimination of disturbances.

For a description of TD7onCPU, refer to the section Block library TeleControl ST7 (Page 191).

Use of the TD7 software

The TD7 versions can be used in the following configurations.

		Master station: S7 type, TIM type, TD7 version									
		CPU 300				CPU 400 / CPU 400H single CPU 1500			CPU 400H redundant *	ST7cc / ST7sc / ST7 SC **	PCS7 Tele-Control
		3V-IE Advanced 4R-IE in the rack		TIM 1531 IRC		3V-IE Adv.	4R-IE stand-alone TIM 1531 IRC		4R-IE stand-alone TIM 1531 IRC		
		TD7on CPU	TD7on TIM	TD7on CPU	TD7on TIM	TD7on CPU	TD7on CPU	TD7on TIM	TD7on TIM	-	-
Station											
S7 type, TIM type	TD7 variant										
CPU 300											
3V-IE / 3V-IE Adv., 4R-IE in the rack, TIM 1531 IRC	TD7on CPU	+	+	+	+	+	+	+	+	+	+
	TD7on TIM	+	+	+	+	+	+	+	+	+	+
CPU 400, CPU 1500											
4R-IE Stand-alone, TIM 1531 IRC	TD7on CPU	+	+	+	+	+	+	+	+	+	+
	TD7on TIM	+	+	+	+	+	+	+	+	+	+
CPU 400H single											
4R-IE Stand-alone	TD7on CPU	+	+	+	+	+	+	+	+	+	+
	TD7on TIM	+	+	+	+	+	+	+	+	+	+

CPU 400H red. *											
4R-IE Stand- alone *	TD7on CPU	-	-	-	-	-	-	-	-	-	-
	TD7on TIM	+	+	+	+	+	+	+	+	+	+
CPU 1200, CPU 1500SP											
CP 1200, CP 1542 SP-1 IRC	TD7on CPU	-	-	-	-	-	-	-	-	-	-
	TD7on TIM	+	+	+	+	+	+	+	+	+	+

* Only single TIM 4R-IE, see note below.

** ST7sc / ST7 ScadaConnect

Note

400H systems

Only a single TIM 4R-IE can be used in all configurations with S7-400H systems. Communication via two redundant TIM 4R-IE is not supported.

1.7 Overview - Connection options of the modules

Transfer options

The interfaces of the ST7-capable communications modules support the following network types and protocols:

Ethernet interface

- **IP-based**

Communication via LAN (copper / FO cable), Internet and IP-based wireless networks

- Neutral (via S7 connection)

- MSC / MSCsec

For information on the protocol variants, refer to the section Mobile wireless and Internet communication (Page 21).

- Mobile wireless with VPN

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

Serial interface

- **Classic WAN**
 - Dedicated line (incl. analog wireless network)
 - Dialup network (analog, ISDN, mobile wireless)
- **IP-based**
Max. 1 interface of the TIM modules:
 - MSC / MSCsec via mobile wireless

Connection combinations:

The following table provides an overview of the various connection options and the devices required for them (modems, routers, switches). You can find information on the required accessories in Volume 1 of the system manual.

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

Network type / transmission path	Connection TIM		Standard, protocol, service	Connection to partner	Partner type *
	Serial interface	Ethernet interface			
Ethernet	-	SCALANCE M / SCALANCE S	VPN	SCALANCE M / SCALANCE S	<ul style="list-style-type: none"> • PC • TIM / CP
	-	SCALANCE X / W		SCALANCE X / W	<ul style="list-style-type: none"> • PC • TIM / CP
Mobile wireless + Internet	-	DSL router	MSC / MSCsec	DSL router	TIM / CP
	-	SCALANCE M	MSCsec	SCALANCE M	TIM / CP
	-	SCALANCE M	VPN	SCALANCE M	TIM / CP
IP wireless network	-	IP wireless modem	IP	IP wireless modem	TIM / CP
Dedicated line	Dedicated line modem	-	RS-232 / RS-485	Dedicated line modem	TIM / CP
	Analog wireless modem			Analog wireless modem	
Analog dial-up network	Dial-up network modem	-	V.32bis/V.34bis	Dial-up network modem	TIM / CP
ISDN dial-up network	ISDN modem	-	<ul style="list-style-type: none"> • ISDN • ISDN + GSM/CSD 	<ul style="list-style-type: none"> • ISDN modem • MODEM MD720 	TIM / CP
Mobile network	MODEM MD720	-	GSM/CSD	<ul style="list-style-type: none"> • DSL router • MODEM MD720 	TIM / CP
			GSM + MSC/MSCsec	<ul style="list-style-type: none"> • DSL router 	

* PC: Control station computer with ST7-compatible application, e.g. SINAUT ST7cc/ST7sc.

1.8 General request and accelerated general request

General request (GR)

If a station connected to ST7cc returns following a disruption, ST7cc sends a general request (GR) to the station to update the current process status of the station. If ST7cc itself fails and starts up again, it sends a general request to all connected stations to bring its process data up to date.

Apart from these GRs triggered automatically by ST7cc due to disturbances, the operator can also send a GR to one station when necessary. The operator can trigger sending of the GR in the faceplate.

In response to a GR, the requested station enters all its data messages along with the current process values in the send buffer of the station TIM. The data messages are preceded by a message indicating "GR start" and the requested messages are followed by a message indicating "GR end". If there are still messages stored in the buffer of the station TIM that are waiting for transmission, the requested messages and the "GR end" message are entered after the buffered messages. Only the "GR start" message is entered at the start of the send buffer.

The messages are then transmitted in the order in which they occur, in other words, first the "GR start" message shown in the picture typical of the queried station by the GR LED flashing yellow. Any stored messages are then transmitted followed by the requested messages with the current process image and finally the "GR end" message. With the "GR end" message, the GR LED is permanently lit green in the picture typical. If the operator triggered a GR manually, and the GR LED is lit green, the operator knows that all archives and the process image of the queried station are up to date.

The duration of GR processing depends on the speed of the transfer network and the number of messages saved in the station. Longer connection breaks and thus longer transfer times can often be expected in stations connected over dial-up networks.

Stations connected over dial-up networks generally have longer connection pauses because the connections are established as seldom as possible to save costs. When there is a GR, it must always be assumed that there are buffered messages to be dealt with and that some time will be needed before there is a response to the general request.

The advantage of the GR process based on the first-in/first-out principle is that the data is transferred in the correct chronological order. This allows normal process control systems to process the data without any problems; in other words, they are capable of supplying their process image and their archives correctly with data.

Accelerated general request (XGR)

If you require the current process image more quickly, you can trigger an accelerated general request (XGR). In an accelerated GR, the messages with the requested process image are entered at the start of the send buffer of the TIM; in other words, before any other messages still buffered in the TIM.

An accelerated GR is supported by:

- SINAUT ST7cc
- Ethernet TIM

1.9 Software requirements

- TD7onTIM
- TD7onCPU in S7-1500
An accelerated GR is only supported by S7-1500 stations with TD7onCPU. S7-300 and S7-400 stations with TD7onCPU answer an accelerated GR like a standard general request.

ST7cc can be set so that the accelerated GR is used automatically instead of the standard GR following disruptions. This makes the current process image available more quickly following a disruption.

Although triggering of a GR is possible for stations connected over dedicated line or wireless, it does not generally achieve a lot because the ST7cc control center is normally kept permanently up to date with these network types.

An accelerated GR may make sense in the case of stations connected via dial-up network, for example, because the master station is not always supplied with the current data of the station in this case. One example application would be a situation whereby you inform yourself of the current status of the station with an accelerated GR prior to a switching action or setpoint change. This would allow you to enter the command or setpoint faster and inform yourself of the current response of the process in the station by means of another accelerated GR.

1.9 Software requirements

Configuration and online functions

The following version of STEP 7 is required for configuring the full range of functions described in this manual:

- STEP 7 Basic / Professional V19 Update 1

For the required STEP 7 product, see section Communications modules (Page 17).

Expansion of SINAUT systems (optional)

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

- STEP 7 V5.6
together with
- SINAUT engineering software V5.5 + SP3

1.10 Hardware requirements

1.10.1 CP 1542SP-1 IRC

BusAdapter

To connect to the Ethernet network, the CP requires a BusAdapter. A BusAdapter does not ship with the CP.

You can find the BusAdapters supported by the CP in the appendix /5/ CP 1542SP-1 IRC (Page 426).

CPUs of the ET 200SP

A CPU V2.9 is required to use the full range of functions of the current firmware of the CP.

The CP supports operation in stations that contain one of the following CPUs:

- CPU 1510SP-1 PN
Article number: 6ES7510-1DJ01-0AB0
Article number: 6ES7510-1DK03-0AB0
- CPU 1510SP F-1 PN
Article number: 6ES7510-1SJ01-0AB0
Article number: 6ES7510-1SK03-0AB0
- CPU 1512SP-1 PN
Article number: 6ES7512-1DK01-0AB0
Article number: 6ES7512-1DM03-0AB0
- CPU 1512SP F-1 PN
Article number: 6ES7512-1SK01-0AB0
Article number: 6ES7512-1SM03-0AB0
- CPU 1514SP-2 PN
Article number: 6ES7514-2DN03-0AB0
- CPU 1514SP F-2 PN
Article number: 6ES7514-2SN03-0AB0
- CPU 1514SP T-2 PN
Article number: 6ES7514-2VN03-0AB0
- CPU 1514SP TF-2 PN
Article number: 6ES7514-2WN03-0AB0

1.10 Hardware requirements

The respective SIPLUS CPUs are also supported.

Note

Keep the firmware version of the CPUs and communications modules up to date

To avoid possible problems when using newer CPUs together with older communications modules, you should keep the firmware of both your communications modules and your CPUs up to date.

Other components of the ET 200SP

Further parts and modules that are also required to set up the ET 200SP station, such as rails, I/O modules or cabling are not listed here. See also /13/ ET 200SP Manual Collection (Page 428) for information on this.

Components of the communications partner

Components required by the communications partners of the CP 1542SP-1 IRC are not listed here. You will find references to other products (e.g. TCSB) in the list of references in the appendix of the manual.

Compatible CPUs

The CPU types that can be used together with the respective TIM module can be found in the relevant manual, e.g. SINAUT system manual Volume 1, section "Installation guidelines and compatible CPUs".

1.10.2 TIM 1531 IRC

Compatible CPUs

The following can be used as the local CPU of the TIM 1531 IRC:

- S7-1500
All CPUs that can be configured in STEP 7 as of firmware version V2.1 from the following series:
 - Standard CPUs (CPU 15xx)
 - Compact CPUs (CPU 15xxC)

For TLS communication as of TIM firmware V2.2: All CPUs that can be configured in STEP 7 as of firmware version V2.9
Redundant CPUs as of firmware version V2.6:

 - H-CPU (CPU 1517H-x PN)
 - R-CPU (CPU 151xR-x PN)
- ET 200SP
As of TIM firmware V2.1: All CPUs that can be configured in STEP 7 as of firmware version V2.5
- S7-300
All CPUs that can be configured in STEP 7 with a PROFINET interface
- S7-400
All CPUs that can be configured in STEP 7

1.11 Expansion of SINAUT projects

1.11.1 Modules for new SINAUT projects and those to be expanded

New SINAUT projects in the TIA Portal

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

Table 1-5 Configuration of modules for ST7 projects in STEP 7 Basic / Professional as of V15

Module (firmware version)	STEP 7 catalog module	STEP 7 product
TIM 3V-IE (V2.8)	TIM 3V-IE	STEP 7 Professional
TIM 3V-IE Advanced (V2.7)	TIM 3V-IE Advanced	STEP 7 Professional
TIM 4R-IE (V2.7)	TIM 4R-IE / TIM 4R-IE Stand-alone	STEP 7 Professional
TIM 1531 IRC (Vx) *	TIM 1531 IRC	STEP 7 Professional

* STEP 7 V15: Firmware V1.0; STEP 7 V15.1: Firmware V2.0; STEP 7 V15.1: Firmware V2.1

Expansion of existing SINAUT ST7 projects in the TIA Portal

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

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

- TIM 1531 IRC
STEP 7 Professional V15 or higher

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

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

1. Configuration of a placeholder (proxy) for an S7-1200/1500 module in the STEP 7 V5 project
The proxy receives the SINAUT-specific communication, connection and address parameters.
2. Export the configuration data (SDBs) of the proxy from STEP 7 V5 as a text file.
3. Import the configuration data of the proxy into a compatible module in STEP 7 Basic / Professional.
The new module adopts the SINAUT-specific communication, connection and address parameters from STEP 7 V5.
4. Complete the configuration of the new module in STEP 7 Basic / Professional.

This procedure is supported by the following modules:

Table 1-6 Module migration from STEP 7 V5 to STEP 7 Basic / Professional (TIA Portal)

Module for STEP 7 V5 project expansion			Module in STEP 7 Basic / Prof. V17	
TIM (function) for expansion	Proxy to be used in the catalog		Compatible modules	Required STEP 7 version
TIM 3V-IE Advanced	PROXY CP1243-8 IRC	⇒	CP 1243-8 IRC	STEP 7 Basic
			CP 1542SP-1 IRC	STEP 7 Professional
TIM 4R-IE	PROXY TIM 1531 IRC	⇒	TIM 1531 IRC	STEP 7 Professional
TIM 4R-IE Stand-alone	PROXY TIM 1531 IRC	⇒	TIM 1531 IRC	STEP 7 Professional

Note

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

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

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

You can find details on configuration in Requirements for the expansion (Page 43).

1.11.2 Requirements for the expansion

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

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

Requirements: Software versions

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

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

Requirements: Firmware versions/module update

- **Firmware of the communications modules**
The following firmware versions are required:
 - **TIM 1531 IRC - V2.4**
A TIM 1531 IRC V1.x can be exchanged for a TIM 1531 IRC V2.3, but not the other way round.
The mode of the "Telecontrol configuration" ("Basic settings") is set to "Configure". The existing data is adopted.
 - **CP 1542SP-1 IRC - V2.3**
Adoption of the data from STEP 7 V5 without the option of making changes
 - **CP 1243-8 IRC - V3.5**
Adoption of the data from STEP 7 V5 without the option of making changes
A physical V2.1 CP can be exchanged for a V3.5 CP but not the other way round. The CP used must have hardware product version 2. The existing data is adopted.

Requirements: TIM 1531 IRC

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

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

- The CPU 1500 must have at least the firmware version V2.5.
- TIM and CPU must be networked in STEP 7 Professional before the import.
- The network addresses in the STEP 7 Professional project must correspond to those in the STEP 7 V5 project.
- When using TD7onCPU, the CPU must not yet be assigned to the TIM in STEP 7 Professional.

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

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

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

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

Configuration

2.1 Security recommendations

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

General

- You should make regular checks to make sure that the device meets these recommendations and other internal security guidelines if applicable.
- Evaluate your plant as a whole in terms of security. Use a cell protection concept with suitable products.
- Do not connect the device directly to the Internet. Operate the device within a protected network area.
- Check regularly for new features on the Siemens Internet pages.
 - Here you can find information on Industrial Security:
Link: (<http://www.siemens.com/industrialsecurity>)
 - You can find a selection of documentation on the topic of network security here:
Link: (<https://support.industry.siemens.com/cs/ww/en/view/92651441>)
- Keep the firmware up to date. Check regularly for security updates of the firmware and use them.
Information regarding product news and new firmware versions is available at the following address:
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21764/dl>)

Physical access

Restrict physical access to the device to qualified personnel.

Network attachment

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

Security functions of the product

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

- Protection levels
Configure a protection level of the CPU.
You will find information on this in the information system of STEP 7.
- Security function of the communication
 - Using the security functions of the telecontrol protocols.
 - Use the secure protocol versions, for example NTP (secure).

Passwords

- Define rules for the use of devices and assignment of passwords.
- Regularly update the passwords to increase security.
- Only use passwords with a high password strength. Avoid weak passwords for example "password1", "123456789" or similar.
- Make sure that all passwords are protected and inaccessible to unauthorized personnel. See also the preceding section for information on this.
- Do not use one password for different users and systems.

Protocols

Secure and non-secure protocols

- Only activate protocols that you require to use the system.
- Use secure protocols when access to the device is not prevented by physical protection measures.
 - The NTP protocol provides a secure alternative with NTP (secure).
 - The HTTP protocol provides a secure alternative with HTTPS when accessing the Web server.
- Deactivate DHCP at interfaces to public networks such as the Internet, for example, to prevent IP spoofing.

Table: Meaning of the column titles and entries

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

- **Protocol / function**
Protocols that the device supports.
- **Port number (protocol)**
Port number assigned to the protocol.

- **Default of the port**
 - Open
The port is open at the start of the configuration.
 - Closed
The port is closed at the start of the configuration.
- **Port status**
 - Open
The port is always open and cannot be closed.
 - Open after configuration
The port is open if it has been configured.
 - Open (login, when configured)
As default the port is open. After configuring the port, the communications partner needs to log in.
 - Closed after configuration
The port is closed because the TIM is always client for this service.
- **Authentication**
Specifies whether or not the protocol authenticates the communications partner during access.

Protocol / function	Port number (protocol)	Default of the port	Port status	Authentication
S7 and online connections	102 (TCP)	Open	Open *	No
Online security diagnostics (CP)	8448 (TCP)	Closed **	Open after configuration	No
HTTP	80 (TCP)	Closed	Open after configuration	No
HTTPS	443 (TCP)	Closed	Open after configuration	Yes
SNMP	161 (UDP)	Open	Open after configuration	Yes (with SNMPv3)

* Some service providers consider the opening of port 102 a security vulnerability. To avoid opening port 102 during online diagnostics, see section Online security diagnostics via port 8448 of the CP (Page 359).

** With the TIM 1531 IRC, Port 8448 is open in the default setting.

Ports of communications partners and routers

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

These can be:

- MSC / 26382 (TCP) - configurable with a central interface: 1024 .. 65535
- NTP / 123 (UDP)
- DNS / 53 (UDP)
- DHCP / 67, 68 (UDP)

- SMTP / 25 (TCP) - Open in CP if configured (only outgoing)
- SMTPS / 587 (TCP) - Open in CP if configured (only outgoing)
- SINEMA RC Autoconfiguration / 443 (TCP) - can be set
- SINEMA RC and OpenVPN / 1194 (UDP) - can be set in SINEMA RC
- IPSec / 500 (TCP) / 4500 (UDP)
- Syslog / 514 (UDP)

2.2 Configuration - Overview

Configuration in STEP 7

You configure the modules and networks in STEP 7 Professional in the TIA Portal. You can find the required version in the section Software requirements (Page 38).

The individual configuration steps are divided into the following groups in accordance with the recommended sequence in the following overview:

1. Create a project
2. Create stations
3. Structure of the stations with TIM and assignment of the CPU
4. Configure the station
5. Create networks
6. Create the telecontrol connections
7. Configure the TD7 software, alternatively:
 - TD7onTIM (via data points)
 - TD7onCPU (via program blocks)
8. Configuration of the messages (SMS/e-mail)

Creating a project with security functions

1. Open STEP 7 in the TIA Portal.
2. Create a project with security functions activated.

The configuration of a TIM 1531 IRC is only possible with activated security functions. Also for access to the Web server of the TIM, the users must be created in "Global security settings".

Note**Activated security functions**

If you have not yet activated any security functions or have logged on in "Global security settings", you first need to log on as a security user. The following roles must be assigned to the user:

- Engineering administrator
 - NET Administrator
-

Create stations

To create a station, drag-and-drop a CPU or TIM from the catalog into the network view of the project. The station is created automatically.

To complete the station, drag additional modules from the catalog onto this station. You can find the modules at the following location in the catalog:

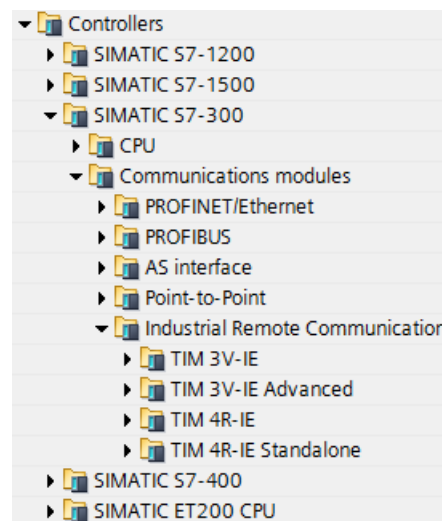


Figure 2-1 Detail of the STEP 7 catalog with S7-300 module folders

1. Create the required SIMATIC stations by inserting a CPU.
You can find a list of compatible CPUs in Volume 1 of the system manual.
2. Insert the required TIM modules into the stations.
3. Drag a "TIM 4R-IE Standalone" from the catalog into the network view to couple a TIM with an S7-400 CPU.
4. To complete the device configuration, switch to the device view of the individual stations.
The catalog now shows directories with device groups suitable for this type of station.
5. Complete the individual stations with input/output modules and other required modules.
The appropriate slots in the device view of the station are framed in blue when selecting a device in the catalog.
See the STEP 7 Information System (Rules for Modules) for help with slot rules.

Pay particular attention to the copy functions for larger projects, see section Configuring and copying modules (Page 52).

Structure of the stations with TIM and assignment of the CPU

Each TIM must have one CPU assigned to it for a consistent project structure with SINAUT ST7 components.

In STEP 7, you configure a station with a TIM and associated CPU just as you build the physical rack:

- TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE in the S7-300 rack
The CPU is inserted in the same rack as the TIM with these stations. The local CPU is automatically assigned to the TIM modules via the rack.
- TIM 4R-IE Stand-alone
A "TIM 4R-IE Stand-alone" is dragged into the network view as a single device. The station of the TIM is created automatically. No additional devices are configured in the TIM station. The CPU with which the stand-alone TIM exchanges data is created in a separate rack. The TIM and CPU are connected by an Ethernet network (see below). Only then is the CPU manually assigned to the TIM ("Subscriber numbers" parameter group of the TIM). Read the note on TD7onCPU in the section "Subscriber numbers".

Configure the station

Configure the station and the TIM, see section Configuring and copying modules (Page 52) and the following sections.

Create networks

Create the required networks for networking the interfaces, see section Networking, configuration of networks and network nodes (Page 60).

Create the telecontrol connections

Create telecontrol connections for the ST7 communication.

You will find the description in the section Creating telecontrol connections for the ST7 communication (Page 140).

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

Configuration of TD7onTIM (data points) or TD7onCPU (blocks)

You can configure the required TD7 software using one of the following two versions:

TD7onTIM

This TD7 version is implemented via data points.

1. Create PLC tags or data blocks for the data to be transferred in the CPU.
The input and output data of the station is not directly addressed by the TIM, but instead via PLC tags or DBs.
2. Open the data point editor, see section Data point configuration (Page 147).
3. Create the required data points.

TD7onCPU

This TD7 version is implemented via program blocks. In this case, you cannot create data points.

1. You must import the blocks for a global library in STEP 7.
Follow the steps described in section Block library TeleControl ST7 (Page 191).
2. Generate TD7onCPU for each CPU involved and use the required blocks.

Configuration of the messages (SMS/e-mail)

1. Open the message editor, see section Data point configuration (Page 147).
2. Create the required messages, see section Messages (Page 183).
3. Save the project.

Commissioning the stations

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

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

2.3 Configuring and copying modules

Copying functions for larger projects

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

- Copying devices and stations
If you use several similar communications modules or CPUs in a project, it is advisable to configure the first device first and then to copy it including the configuration data. This can include the following objects:
 - The user program of CPUs
 - PLC tags and data blocks of CPUs
 - Data points of communications modules

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

Recommendation:

Network the interfaces only after copying the devices.

- Copying PLC tags
If you use several similar items of input and output data, you can copy PLC tags in the tag table.
After copying you only need to adapt the address parameters.
- Copying data points
In much the same way as with PLC tags, you can also copy data points.
For a detailed description of the functions, refer to the section Data point configuration (Page 147).

2.4 Compatibility mode of the CP 1243-8 IRC V3.0

If you have a system that contains the CP 1243-8 IRC with firmware version V2.1, you can configure ST7 connections for the CP via the "Network data" editor as of STEP 7 V15.1. The "Compatibility mode V2.1" option is available to you for this purpose.

The option is only available for the following combinations:

- Real CP (in the plant) with firmware version V2.1
- Configured CP (STEP 7) with firmware version V3.0
Create the CP in STEP 7 with firmware version V3.0 and then enable the option "Compatibility mode V2.1".

General

- **Compatibility mode V2.1**

If this option is enabled, you can configure V3-compatible ST7 connections for the CP in the "Network data" editor.

The configuration data can be loaded into a real CP V2.1.

Restrictions for activation:

- No protocol switchover
- No time of day from local station and NTP
- No parameter group "Communication with the CPU > CP diagnostics"
- No online function "Save service data"

2.5 MPI address

MPI address

The parameter group is only displayed for TIM modules in stations that contain an S7-300 CPU with a party line. For information on the party line, see Glossary.

- **Address**

By default, one CPU has MPI address 2 and the first TIM in the station has MPI address 3.

The address of the network node can be changed.

2.6 Communication types

"Communication types"

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

- **Enable telecontrol communication**

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

- **Protocol type**

- ST7
- DNP3
- IEC 60870-5
- TeleControl Basic

To use the ST7 protocol with the CP 1243-8 IRC and CP 1542SP-1 IRC, the Security functions have to be enabled in addition.

Note that if you change the telecontrol protocol later, all protocol-specific data will be deleted. This includes the data point and partner information.

- **Enable online functions**

Enables access to the CPU for the online functions via the CP (diagnostics, loading project data etc.). If the function is enabled, the engineering station can access the CPU via the CP. If the option is disabled, you have no access to the CPU via the CP with the online functions. Online diagnostics of the CPU with a direct connection to the interface of the CPU remains possible, however.

- **Enable S7 communication to the CPU**

Enables the functions of S7 communication between the engineering station and the station CPU, as well as S7 routing.

If you configure S7 connections to the relevant station, and these run via the communications module, you need to enable this option for the communications module. S7 routing is supported by the following modules:

- CP 1243-8
As of CP firmware V2.1 with CPU $\geq$ V4.2
- CP 1542SP-1 IRC
As of CP firmware V1.0 with CPU $\geq$ V2.0
- TIM 1531 IRC
- TIM 3V-IE / TIM 4R-IE

Please note:

Disabling this function means no security measure. To protect the station, use suitable security functions such as firewall, VPN or password protection of the CPU.

Open User Communication does not need to be enabled since you then need to create the relevant program blocks. Unintended access to the CP is therefore not possible.

- **Activate telecontrol communication via SINEMA Remote Connect**

Configurable with:

- CP 1243-1
- CP 1243-7 LTE
- CP 1243-8 IRC
- CP 1542SP-1 IRC

For additional details, see appendix SINEMA Remote Connect (CP) (Page 419).

RTU3000C**Configuration**

With the RTU3000C, you must configure the following parameters in the STEP 7 project:

- **Communication types**

- Protocol type
The telecontrol communication type configured in the RTU
- Temporary connection
Enable this option if the RTU is connected via mobile wireless and is only in the network temporarily.

- **Ethernet interface**

- WAN settings
The settings are permanently preset depending on the selection of the telecontrol protocol.

Ethernet address

- Dynamic IP address
If a dynamic IP address is assigned by the mobile wireless network provider, this option must be enabled.
- Fixed IP address from the mobile wireless network provider
Enter the fixed IP address of the RTU assigned by the mobile wireless network provider here.
If a mobile router is connected in between, enter the IP address of the router here.

- **Subscriber numbers**

- Subscriber number (RTU)
Only for ST7: Here, you assign the subscriber number of the RTU as it was configured in the WBM of the RTU.
- Subscriber number of the CPU (RTU)
Here, you assign the subscriber number of the CPU as it was configured in the WBM of the RTU.

- **Security**

The security settings configured in the RTU are permanently enabled in STEP 7 in the default setting.

- **Telecontrol connection**

Then create the telecontrol connection between the RTU and its connection partner in the "Network data > TeleControl" editor.

Compiling and loading

The STEP 7 configuration data cannot be loaded into the RTU3000C. The RTU3000C therefore cannot be compiled in STEP 7.

2.7 Basic settings

2.7.1 Configuration

Configuration

- **Telecontrol configuration**

Configurable with:

- TIM 1531 IRC
- CP 1243-8 IRC
- CP 1542SP-1 IRC

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

- **Configure**
The subscriber data and SINAUT connections are configured in the parameter groups of the module and in the connections editor of STEP 7 Professional.
- **Import**
The configuration data of the proxy module from STEP 7 V5 is imported into the module in STEP 7 Professional.
You can perform the import yourself in the parameter group Partner stations (Page 88). To expand STEP 7 V5 projects via proxy modules, see section Expansion of SINAUT projects (Page 41).

Note

Deletion of configuration data

Define one of the two options before you continue configuring the module.

If you change the parameter later, all configuration data up to that point will be discarded.

Max. number of registered tags

Only with TIM 1531 IRC

Maximum number of PLC tags that can be registered by the module in the CPU for automatic change control. Value changes of PLC tags with automatic change control are signaled automatically by the CPU to the module.

The PLC tags are registered in the order in which they were created.

Range of values: 1...32767

Default: 500

Note: The number of PLC tags per CPU is limited:

- CPU 1511/1512/1513: Max. 2000
- CPU 1515: Max. 4000
- CPU 1516: Max. 8000

- CPU 1518: Max. 20000
- CPU 1519: Max. 40000

2.7.2 Retentive saving of events

Retentive saving of events

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

For the general reaction, the following parameters are available:

- **Activate retentive saving**
When there are connection disruptions activates the retentive saving of events on the SD card.
- **Number of events before saving**
Saving events on the SD card starts when the number of events in the send buffer configured here is reached after a connection failure.
- **Interruption time before saving**
Saving events on the SD card starts when the time of the connection interruption configured here is reached.

Whether or not the values of the individual data points are saved is specified in the configuration of the individual data point in the "Trigger" tab, see section "Trigger" tab (Page 167).

For the maximum number of savable messages see section Performance data and configuration limits (Page 23).

2.7.3 IP routing

IP routing

These parameters are available with the TIM 1531 IRC.

Note

The function is not intended for large data volumes.

Restrict routing via the TIM to approx. 1 Mbit/s so that productive operation of the TIM is not impaired.

IP routing

- Allow IP routing
Enables IP routing via the interfaces of the module.
- Routing method
Defines the paths for IP routing:
 - Local: IP routing only between the Ethernet interfaces of the module
 - Via subnets: IP routing via a maximum of 10 configurable routers that can be accessed via the module's interfaces.

Router addresses

- Ethernet interface
Ethernet interface of the module via which IP routing is to be configured. The IP addresses of the interfaces used for IP routing need to be permanently configured.
An interface can be selected several times for different routes.

Note**Consistency of the address parameters**

STEP 7 does not check the consistency between manually configured addresses and the parameters of the module's Ethernet interfaces.

Ensure consistency with the address parameters of the respective interface.

- Address type
Selection of the IP version of the address parameters configured below (IPv4 / IPv6)
When using IPv6 addresses, you must enable IPv6 for the respective interface.
- Network address
Network address of the routing destination (IP address * subnet mask)
- Subnet mask / Prefix
Subnet mask (IPv4) or prefix (IPv6) of the routing destination
- Router address

2.8 Options

Configurable with: TIM 3V-IE / TIM 4R-IE

Frame memory

- **Frame memory size**
Size of the memory in which the frames to be sent are stored for all configured WAN drivers. The size of the frame memory is divided equally among all configured communications partners. The size of the frame memory can be found in Volume 1 of the System Manual (Performance Data).
Here, you set the size of the frame memory for events (send buffer). If the value 0 is entered, the entire free memory following startup is used.
You will find details of how the send buffer works (storing and sending events) as well as the options for transferring data in the section Process image, type of transmission, event classes (Page 161).
- **Size of memory block**
Size of the blocks into which the global frame memory is segmented. Every send frame occupies at least this memory in the memory. The default size is 128 bytes.

RAM

- **Frame memory size**
Size of the RAM memory. This area in the main memory of the TIM that is created temporarily for test purposes is deleted after the TIM restarts. The value is preset at 100 kB.

Replacing a module without a programming device

- **Save configuration data on the CPU**
When the option is enabled, the system data blocks (SDBs) of the TIM are stored on their local CPU. The TIM receives its SDBs during startup from the CPU.
If the TIM fails, the defective TIM can be replaced by a TIM of the same type without downloading the SDBs to the TIM again.

Basic data point settings of the TIM

Frame receipt

- **Check of the source address**
When this option is enabled, the address of the communication partner is checked for received data. All data that not originating from configured partners are discarded.
If the TIM is to receive and process data from multiple partners, the option must be disabled.

CPU sampling cycle

This is where the number of configured data points assigned to the sampling cycles of the CPU is displayed. The assignment is made in the data point configuration of the "General" tab.

You also specify the number of data points that are read from the CPU per sampling cycle.

- **Number of configured data points in the normal cycle**
Number of data points that were assigned to the normal read cycle.
- **Number of configured data points in the fast cycle**
Number of data points that were assigned to the fast read cycle.
- **Read max.**
Maximum number of data points read from the CPU per sampling cycle.

2.9 Networking, configuration of networks and network nodes

2.9.1 Networking of the interfaces

Interfaces of the TIM

The arrangement of the module interfaces in the STEP 7 device symbol (network view) corresponds largely to the structure of the respective device.

The interfaces of TIM modules are arranged as follows:

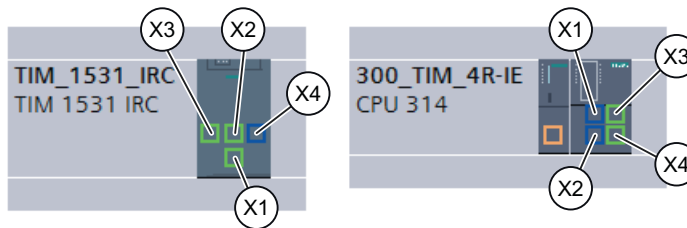


Figure 2-2 Device symbols of the TIM 1531 IRC and TIM 4R-IE with interface numbers

Networking interfaces

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

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

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

Interface networking with the TIM

You will find the transmission options of the TIM depending on the interface in the section Overview - Connection options of the modules (Page 35).

Recommendation networking:

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

1. Network the WAN networks in the network view of STEP 7.
In the graphic network view, you have an overview of the subnets of the entire system in the project.
2. First configure the interface parameters described in the section WAN settings of the interfaces (Page 63):
 - WAN type
 - Network type
 - Network node type
 - Modem type
3. Select the relevant interface to create a new WAN network. Alternatively:
In the parameter group "Network interface with" of the interface:
 - Using the "Add new subnet" button

On the interface in the device symbol of the TIM

 - Using the shortcut menu "Create subnet"
 - Graphically by dragging (holding the mouse pointer pressed) to the interface symbol of the communications partner

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

Network representation of a classic WAN

A classic WAN network is displayed in blue.

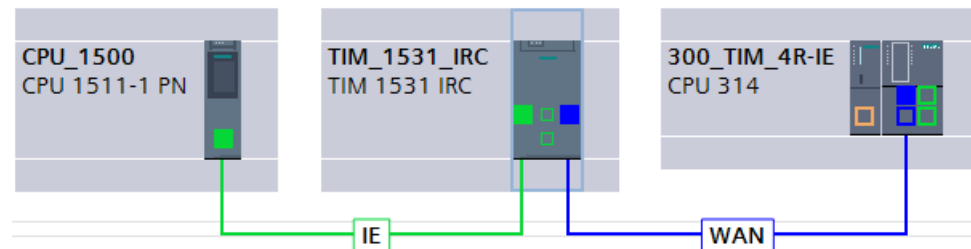


Figure 2-3 TIM modules, serial interfaces networked via classic WAN.

MSC connections

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

Note

MSC via TIM interfaces:

- **Ethernet interface X1**
MSC is only supported by interface X1 on TIM modules with multiple Ethernet interfaces.
- **Serial interface**
For the serial interface you need to set the WAN type to "IP-based" and the Network type to "MSC".

MSC connections are represented as follows:

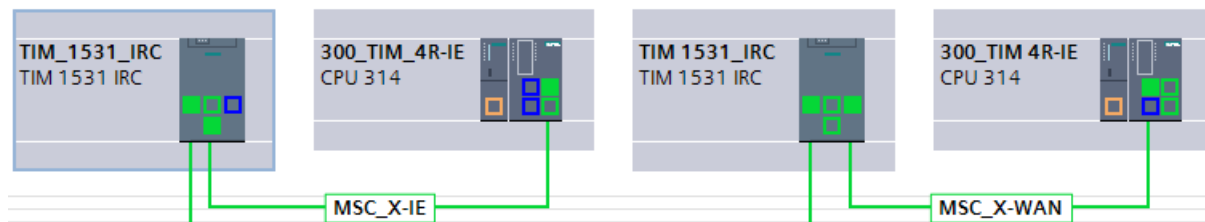


Figure 2-4 MSC connection via Ethernet interfaces (left) and via serial interfaces (right)

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

"Network node type" parameter for node stations

In the following figure, the center TIM is a node station.

The "Network node type" parameter is configured as follows for the four marked interfaces:

1. **Master station (top):**
Interface in the direction of the node station: "Master station"
2. **Node station (center):**
Interface in the direction of the master station: "Node station"
3. **Node station (center):**
Interface in the direction of the station: "Master station"
4. **Station (bottom):**
Interface in the direction of the node station: "Station"

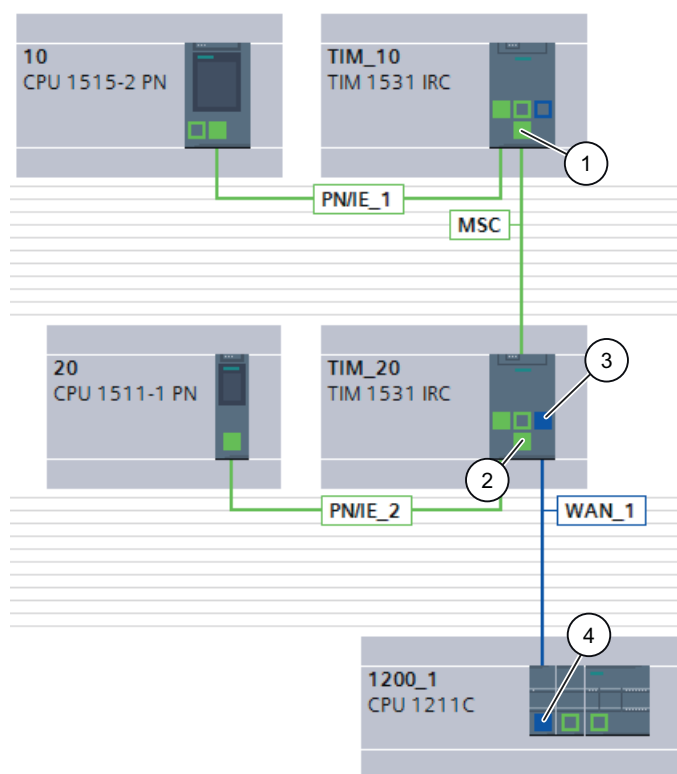


Figure 2-5 Network with master station (top), node station (center) and station (bottom)

2.9.2 WAN settings of the interfaces

WAN settings

In this parameter group of the interfaces you will find parameters that decide the properties of the connected WAN network: First, configure the respective interface of the TIM.

The subsequently connected WAN network adopts the most important settings. For information on configuring the WAN networks, refer to the section Configuring WAN networks (Page 83).

Parameters of interfaces:

- **WAN type**

Selection of the WAN type of the interface:

- **IP-based**

Default setting of the Ethernet interface

- **Classic WAN**

Default setting of the serial interface

- **Serial interface with WAN type "IP-based"**

The serial interface of the TIM and of the CP 1243-8 IRC as of V3 can also be configured "IP-based". In this case, the MSC/MSCsec protocol is used via mobile wireless.

A MODEM MD720 must be connected to the serial interface.

The CP 1243-8 IRC requires the connection of a TS Module RS-232 in addition for this application.

The communications partners of the IP-based serial interface is the Ethernet interface of a TIM 1531 IRC with the settings "Network type" = "MSC" and "Network node type" = "Master station".

Some of the other settings of the IP-based configured serial interface are preset:

- "Network type" = "MSC"

- "Network node type" for the CP: "Station"

- "Network node type" for the TIM: "Station" or "Node station"

- "Modem type" = "Mobile wireless modem MD720"

- **Network type**

For WAN type "IP-based":

- Neutral

S7 connections (e.g. TIM 1531 IRC ↔ CPU) and ST7 connections

- Mobile wireless with VPN

- MSC

ST7 connections with the secure MSC or MSCsec protocol

For WAN type "Classic WAN":

- Dedicated line

- Dialup network

- **Network node type**

Decides the Network node type of the interface:

- Master station
- Node station

For TIM modules that act as node stations, set the network node type as follows:

- Interface in the direction of the master station: "Node station"
- Interface in the direction of the lower-level network: "Master station"

You will find an illustration in the section Networking of the interfaces (Page 60).

- Station

- **Modem type**

The modem type for connection to the serial interface must be configured for the network type "Dialup network" and, in the case of the classic TIM modules, also for the network type "Dedicated line".

The entries have the following meanings:

- MD2
Dedicated line modem (network type "Dedicated line")
- MD3
Modem for analog dialup networks (network type "Dialup network")
- MD4
ISDN modem (network type "Dialup network")
- Mobile wireless modem MD720
Mobile wireless modem MODEM MD720 for mobile wireless transmission (network type "Dialup network" or in the case of an IP-based configured serial interface "MSC")
- Third-party modem
Any compatible modem for the network types "Dedicated line" or "Dialup network" (analog / ISDN / mobile wireless)

2.10 Ethernet interface

2.10.1 Ethernet interface > address parameters

Ethernet interface > Ethernet addresses > IP protocol ...

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

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

Ethernet interface > Port [Xn P1]

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

For information on configuring the WAN settings, refer to the section WAN settings of the interfaces (Page 63).

2.10.2 Advanced options

2.10.2.1 MSC protocol settings

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

For information on each of the following parameter groups "TCP connection monitoring" and "Port" see section Ethernet interface > address parameters (Page 65).

MSC protocol settings - Ethernet: MSC master station

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

Access to MSC master station

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

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

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

- **MSC port of the router**

The port number is preset. If necessary, you can change the port number. Note that the port number of both partners must be identical. The port number cannot be configured for the station.

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

- **Host name**

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

- **IP address DNS server**

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

MSC monitoring settings

- **MSC monitoring time**

Monitoring time (minutes) for the MSC connection

- Behavior of the master station:

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

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

- Behavior of the station:

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

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

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

- **Collect data volume**

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

MSCsec protocol

- **Enable security protocol MSCsec**

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

The secure "MSCsec" variant is permanently preset in interfaces with network node type "Master station". However, the interface also processes data with the simple variant MSC.

- **Key exchange interval**

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

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

Coordinate the values of the master station and stations with one another.

Note**Compiling the MSCsec stations**

Configure all stations and the master station that use MSCsec and create the telecontrol connections.

Compile the stations first and the master station last.

MSC protocol settings - Ethernet: MSC station/node station

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

MSCsec protocol

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

- **Enable security protocol MSCsec**
If the option is enabled, the secure version of the MSC protocol "MSCsec" is used.
- **Key exchange interval**
Time (h) after which the secret key is renewed between the MSC station and master station.
Default setting: 0. Permitted range: 0...255
Coordinate the values of the master station and stations with one another.

MSC protocol settings - serial interface: MSC station

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

GPRS access

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

- **Activate PIN**
Activates the PIN of the installed SIM card. If the contract with your network provider does not include a PIN, then disable this function.
- **PIN / Repeat PIN**
Entry of the PIN of the SIM card inserted in the modem. Not required for mobile wireless contracts without a PIN.
- **Extra transmission time**
Time offset (s) with slow networks to prevent premature stopping of connection establishment.
Default setting: 10. Permitted range: 0...65535. If you enter 0 (zero), the function is disabled.
The following values are usual:
 - Mobile wireless networks: 10
 - Wireless or satellite transmission: 0 .. 1
- **Max. allowed disruption time**
Tolerance time (s) for a detected connection disruption.
If there is still a disruption on the connection when the set time has elapsed, the disruption is signaled to all accessible connection partners of the disrupted station.
Default setting: 0. Permitted range: 0...255. If you enter 0 (zero), the function is disabled.

MSCsec protocol

- **Enable security protocol MSCsec**
If the option is enabled, the secure version of the MSC protocol "MSCsec" is used.
Can only be configured if the setting of the Network node type of the interface is "Station" or "Node station".
- **Key exchange interval**
Time (h) after which the secret key is renewed between the MSC station and master station.
Default setting: 0. Permitted range: 0...255
Coordinate the values of the master station and stations with one another.

APN settings

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

- **APN - country**
Country of the network provider that provides the APN.
Select the country and one of the preset network providers or select the entry "User-defined" and configure the APN manually.
- **APN - provider**
name o the network provider that provides the APN. Select the name from the drop-down list.
- **APN**
Access Point Name
Name of the access point. You obtain this information from your network provider.
- **APN user name**
Entry of the user name for Internet access via the mobile wireless network. You obtain this information from your network provider.
- **APN password**
Entry of the password for Internet access via the mobile wireless network. You obtain this information from your network provider.

2.10.2.2 TCP connection monitoring

Ethernet interface > Advanced options > TCP connection monitoring

- **TCP connection monitoring time**
Function: If no data traffic takes place within the TCP connection monitoring time, the module sends a keepalive frame to the communications partner.
Default setting: 180 s
If you enter 0 (zero), monitoring is disabled.
The permissible range depends on the module type.
Default setting: 180 s
- **TCP keepalive timeout**
After sending a keepalive frame, the module expects a response from the communications partner within the keepalive timeout. If the module does not receive a response after three keepalive frames, it closes the connection.
Default setting: 10 s
The permissible range depends on the module type.

2.10.2.3 Transmission settings

Transmission settings

- **Send conditional spontaneous frames as blocks**
Indicates the form of transfer of data frames with the transmission mode "Conditional spontaneous".
 - Option enabled
Several data items are collected and sent only when a frame is filled to the maximum.
 - Option disabled
The data is transmitted spontaneously as a single frame.The transmission mode "Conditional spontaneous" is set differently for the TD7 versions:
 - TD7onTIM
In the "Trigger" tab of the data point (data point editor)
 - TD7onCPU
Via the "Unconditional" parameter of the data point typicals (TD7 blocks)
- **Max. allowed disruption time**
Tolerance time (s) for a detected connection disruption.
If there is still a disruption on the connection when the set time has elapsed, the disruption is signaled to all accessible connection partners of the disrupted station.
Default setting: 0. Permitted range: 0...255. If you enter 0 (zero), the function is disabled.

2.10.2.4 MTU size

MTU size

Option for interface-specific reduction of the maximum packet size on OSI Layer 3 (Maximum Transmission Unit - MTU)

If the frame size (e.g. segments) is limited in your plant, you can reduce the size of the packages sent by the communications module with this option.

Range of values: 1000 .. 1500 Byte

Default setting: 1500

2.10.3 Web server access

2.10.3.1 CP

Access to the Web server of the CPU

The Web server is located in the CPU. Via the CP, you have access to the Web server of the CPU.

From a PC you can access the Web server of the station if the PC is connected to the system network via LAN.

You will find information on the Web server of the S7-1200 in the manual /11/ S7-1200 system manual (Page 427).

You will find information on the Web server of the ET 200SP in the manual /12/ ET 200SP system manual (Page 428).

2.10.3.2 TIM 1531 IRC

Access to the Web server

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

As default access is disabled. Refer to the explanations in section Security recommendations (Page 45).

You enable the Web server and make further settings in the parameter group "Web server", see section Web server (TIM 1531 IRC) (Page 86). There you can also enable or disable access.

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

2.11 Serial interface of the TIM

For information on configuring the WAN settings, refer to the section WAN settings of the interfaces (Page 63).

2.11.1 WAN parameters

WAN address

This is where you set the WAN address of the serial interface.

For the meaning, see section Communication via the SINAUT ST7 protocol (Page 18).

2.11.2 Advanced options - MSC settings

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

For a description of the parameters, see section MSC protocol settings (Page 66).

2.11.3 Advanced options - Dedicated line

Settings dedicated line

Settings serial interface

- **Interface standard**

Possible standards of the serial interface: RS-232 / RS-485

Select alternatively:

- RS-232

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

- RS-485

When a modem with an RS-485 interface is connected

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

- **RS-485 termination**

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

- **Operating mode**

Configurable with: TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE

The operating mode of the serial interface specifies the mode of the data transfer.

A maximum of one serial interface of a TIM module may be operated in DMA mode.

Range of values:

- Interrupt (block)

This operating mode applies to the transmit and receive mode. The default mode

Interrupt (block) is suitable for all connections. 4 characters are transferred per block.

Following this, there is an interrupt. The received characters are checked only after a complete frame has arrived.

- DMA

This operating mode applies to the transmit and receive mode. The DMA mode should be used for connections with a high baud rate or a lot of frame traffic, however not for GSM networks.

- Interrupt (individual characters)

This operating mode is used only in the receive direction. In the transmit direction, the block mode continues to be used. The interrupt mode is suitable for extremely bad lines.

An interrupt is triggered per transmitted character and each character is analyzed immediately after it is received allowing extremely good diagnostics of transmission errors. This mode is more reliable than the block mode but is slower.

Number of spontaneous frames

- **Max. number of spontaneous frames**
Only for interfaces with the Network node type "Station" / "Node station".
The parameter decides after how many frames the master station has the opportunity of transferring its pending frames to the station. As maximum, the configured number of spontaneous frames that are pending at the time of the first call frame are transferred.
Default setting: 200. Permitted range: 0...255
If 0 (zero) is set, all spontaneous frames are sent at the call.
- **Limit for locked data frames**
Configurable with: TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE
Maximum percentage of locked data frames in the send buffer. If the value is exceeded, the image procedure is used for new incoming data to prevent the send buffer from overflowing.
Data frames that cannot be sent due to communications problems are locked.

Connection establishment

- **Extra transmission time**
Time offset (s) with slow networks to prevent premature stopping of connection establishment.
Default setting: 10. Permitted range: 0...65535. With the setting 0 (zero), the function is disabled.
Usually in dedicated line networks the value is 0:

Time options

- **RTS/CTS delay time**
The RTS/CTS delay (ms) can be required when a modem is connected via an RS-485 interface.
Default setting: 0. Permitted range: 0...65535
 - Value 0
After setting the RTS signal, (turn transmitter on) transmission only starts when the CTS signal (Clear To Send) was set by the modem.
 - Value > 0
Transmission is not delayed until the CTS signal of the modem.
After the RTS signal has been set, transmission is delayed for the configured time and then started immediately.
- **Send delay time**
The send delay time (ms) is used only when the ready to send (CTS signal) comes from the modem (RTS/CTS delay time = 0).
As soon as the CTS signal comes from the modem, the send delay time is started. Sending starts after the time has elapsed.
Default setting: 0. Permitted range: 0...65535

- **Polling monitoring time**

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

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

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

- Parameters configured:
After the configured time elapses, the TIM sends a message to its local CPU.
- Parameter not configured at a transmission speed of 9600 bps:
After 4 seconds without frame traffic, the TIM outputs a message.
- Parameter not configured at a transmission speed of 1200 bps:
After 32 seconds without frame traffic, the TIM outputs a message.

- **Max. allowed disruption time**

Tolerance time (s) for a detected connection disruption.

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

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

Polling parameters

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

- **Ratio "Polling / Spontaneous"**

Max. number of spontaneous frames that can be sent from a master station between two calls.

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

- **Stations per polling sub-cycle**

Configurable with: TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE

Number of stations that can be called in the sub-cycle per polling cycle of the master station.

Form of transfer

- **Form of transfer**

Specifies the form in which data frames are sent:

- As single frames
- Several frames as a block

2.11.4 Advanced options - Dial-up network

Settings dialup network

Only the configurable parameters are explained.

Settings serial interface

- **Interface standard**

Standard of the serial interface: RS232 / RS485 - can be switched over
Select one of the following values:

- RS232

When a modem is connected to the interface of the TIM

- RS485

Connection of the internal terminating resistor of the TIM

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

- **RS-485 termination**

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

- **Operating mode**

Configurable with: TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE

The operating mode of the serial interface specifies the mode of the data transfer.

A maximum of one serial interface of a TIM module may be operated in DMA mode.

Range of values:

- Interrupt (block)

This operating mode applies to the transmit and receive mode. The default mode

Interrupt (block) is suitable for all connections. 4 characters are transferred per block.

Following this, there is an interrupt. The received characters are checked only after a complete frame has arrived.

- DMA

This operating mode applies to the transmit and receive mode. The DMA mode should be used for connections with a high baud rate or a lot of frame traffic, however not for GSM networks.

- Interrupt (individual characters)

This operating mode is used only in the receive direction. In the transmit direction, the block mode continues to be used. The interrupt mode is suitable for extremely bad lines.

An interrupt is triggered per transmitted character and each character is analyzed immediately after it is received allowing extremely good diagnostics of transmission errors.

This mode is more reliable than the block mode but is slower.

Number of spontaneous frames

Configurable with: TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE / CP 1243-8 IRC

- **Max. number of spontaneous frames**

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

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

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

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

Connection

Configurable with: TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE / CP 1243-8 IRC

- **Extra transmission time**

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

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

The following values are usual:

- Mobile wireless networks: 10
- Wireless or satellite transmission: 0 .. 1

Call parameters

- **Data format dial phase**

Configurable with: TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE / CP 1243-8 IRC

Defines the frame format.

Select the appropriate combination of number of data bits, parity and stop bits.

- **Dialing command**

Dialing command for the local modem

Possible values:

- D (AT command)
- DP (AT command, pulse dialing)
- DT (AT command, tone dialing)

When possible use the dialing command "D".

- **Dialing prefix**

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

Permitted range: Max. 12 digits

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

- **Own phone number**

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

Permitted values:

- Digits 0 ... 9
- Plus character (+) as placeholder for the trunk prefix (usually 00 or 09) before the international area code

Example: +1230123456789

AT initialization

- **User-defined**

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

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

- MD3 : ATS45=3\N0F0&W
- MD4 : ATS45=83\$P1\N0&W
- MD720 : For information on the initialization string, refer to the manual /15/ MODEM MD720 (Page 428).

- **AT string**

Input box for the AT initialization string

Transmission settings for conditionally spontaneous frames

Configurable with: TIM 3V-IE / TIM 4R-IE / CP 1243-8 IRC

- **Transmission criterion**

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

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

Range of values:

- Standard conditions
No connection will be established due to the existence of conditionally spontaneous frames .
Only in the following cases will a connection be established to send conditional spontaneous frames:
 - Threatened overflow of the send buffer
 - Connection establishment by the communications partner
- Degree of filling
The TIM only establishes a connection when the configured fill level of the send buffer for conditional spontaneous frames is exceeded.
In the input box, enter the fill level (%) of the send buffer at which when exceeded the TIM establishes a connection.
- Time
The TIM sends conditional spontaneous frames at a configured time of day, configurable with "Hours" / "Minutes".
- Time scheme
The TIM sends the conditional spontaneous frames cyclically at a configurable interval, configurable with "Hours" / "Minutes".

Time options

- **Call acceptance delay**

Wait time (s) before accepting an incoming call

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

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

- **Dial test interval**

The test interval (min) is started when no connection could be established by the module after 3 attempts. When the test interval elapses, the module attempts to establish a connection again.

If the connection establishment fails again, the test interval is restarted.

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

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

- **Minimum connection duration**

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

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

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

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

- **Abort delay time**

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

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

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

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

Event transmission

Configurable with: TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE / CP 1243-8 IRC

- **Form of transfer**

Specifies the form in which data frames are sent:

- As single frames
- Several frames as a block

Mobile wireless settings

- **PIN**

PIN of the SIM card of the modem MD720

2.12 Serial interface of the CP 1243-8 IRC

2.12.1 Configuration of the serial interface

Serial interface with TS Module

The serial interface can only be used for communication using the ST7 protocol.

The physical connector for the serial interface is located on the left-hand side of the CP, see Serial connection for TS module (Page 79).

For connection to the network you require a suitable TS module.

The following TS modules are supported:

- **TS Module Modem**
Suitable for connection to an analog dial-up network
- **TS Module RS-232**
Suitable for connection to dedicated line, ISDN dial-up network or mobile wireless network

For details, see also the operating instructions /4/ CP 1243-8 IRC (Page 426), appendix TS modules.

Configuration of the serial interface

You configure depending on the application or the firmware version of the CP.

- **CP as replacement for TIM 3V-IE Advanced**
Use a CP with firmware V2.1.
 - You have exported the configuration data of the substitute for the CP (PROXY CP1243-8 IRC) from STEP 7 V5 as a text file with the SDB content.
 - You import this text file in STEP 7 Basic in the parameter group Partner stations (Page 88).
- **CP for new systems**
Use a CP with firmware V3.
You configure the serial interface only in STEP 7 Basic, refer to the section below.

SMS

If the CP is intended to use the sending of SMS messages, you need to activate the function:

- CP with firmware V2.1
Enable or disable the sending of SMS messages at this point and assign the number of the SMSC (Short Message Service Center).
- CP with firmware V3
You enable SMS and configure the SMSC in the parameter group SMSC (Page 114).

2.12.2 Serial connection for TS module

Connector for TS module

The connector for a TS module is located on the left-hand side of the CP under the cover, see /4/ CP 1243-8 IRC (Page 426).

2.12.3 Configuring a TS module

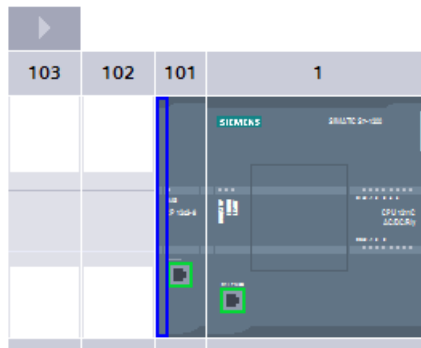
Configuration of the serial interface is only possible and necessary for the CP with a firmware version as of V3.

To be able to configure the serial interface, you must first insert the suitable TS module in STEP 7. Without a configured TS module, the serial interface is not displayed and the CP is only displayed with the green symbol for the Ethernet interface.

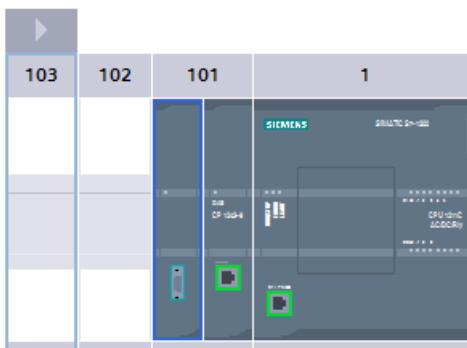
Inserting a TS module for the serial interface

To insert a TS module in STEP 7, follow the steps outlined below:

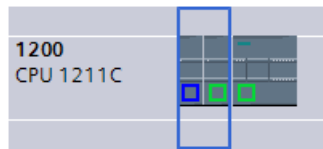
1. Open the device view of the CP.
When the network view is open: Open the device view by double-clicking on the CP or by clicking on the work area "Device view".
2. Select the CP in the device view.
The CP is displayed with a blue frame.
3. In the catalog open the following directory:
"Communications modules > Industrial Remote Communication > TS modules"
For information on selecting the suitable TS module, refer to the operating instructions /4/ CP 1243-8 IRC (Page 426), appendix "TS modules".
4. Expand the directory of the required TS module and select the TS module.
A vertical blue line appears on the symbol of the CP.



5. Drag the TS module precisely onto the blue line of the CP.
The TS module is inserted beside the CP on the same slot.



In the network view the CP is now displayed with the blue symbol for the serial WAN interface.



You can configure the serial WAN interface in the device or in the network view.

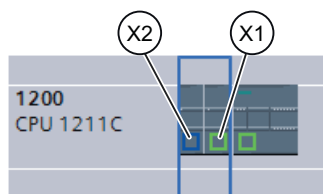
2.12.4 WAN settings

For descriptions of the parameters, refer to the section WAN settings of the interfaces (Page 63).

2.12.5 WAN parameters (networking the CP)

Interfaces of the CP

The following figure shows both interfaces of the CP in the STEP 7 device icon (network view):



X1 Ethernet interface

X2 Serial interface

Figure 2-6 Device icon of the CP with interface numbers

Networking interfaces

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

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

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

Networking the serial interface

You will find the supported transmission options of the CP in the section Overview - Connection options of the modules (Page 35).

Recommendation networking:

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

1. Network the WAN networks in the network view of STEP 7.
In the graphic network view, you have an overview of the subnets of the entire system in the project.
2. First configure the interface parameters described in the section WAN settings of the interfaces (Page 63):
 - WAN type
 - Network type
 - Network node type
 - Modem type
3. Select the interface to create a new WAN network. Alternatively:
In the parameter group "Network interface with" of the interface:

- Using the "Add new subnet" button

On the interface in the device icon of the CP

- Using the shortcut menu "Create subnet"
- Graphically by dragging (holding the mouse pointer pressed) to the interface symbol of the communications partner

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

Network representation of a classic WAN

A classic WAN network is displayed in blue.

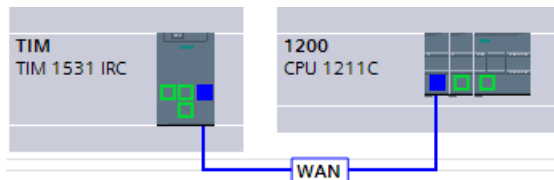


Figure 2-7 Serial interface of a TIM and a CP 1243-8 IRC connected via classic WAN.

Networking the Ethernet interface

You network the Ethernet interface in the same way as the serial interface.

The following example shows two modules with one MSC connection via Ethernet.

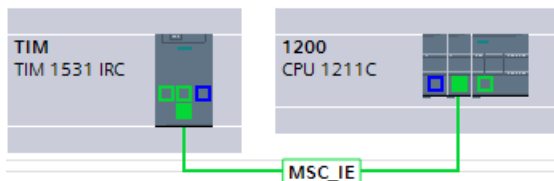


Figure 2-8 MSC connection between a TIM and a CP

Address parameters

- **WAN address**
For every networked interface in a classic ST7 network a WAN address is assigned. This is unique throughout the network.
Since telecontrol connections can run via several network nodes and node stations, unique addressing of the interfaces involved is necessary.
The WAN address is configured in the parameter group "WAN parameters" of the serial interface.
- **Subscriber numbers**
You configure the subscriber numbers in a separate parameter group, see section Subscriber numbers (Page 117).

2.12.6 Advanced options

See Advanced options - Dedicated line (Page 72)

See Advanced options - Dial-up network (Page 74)

2.13 Configuring WAN networks

Parameters of classic WAN networks

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

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

General

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

Network settings

Network configuration

- **WAN type**

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

For classic WAN:

- Dedicated line
- Dialup network

For IP-based WAN:

- Neutral
- Mobile wireless with VPN
- MSC

Access method

- **Access method**

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

- Dedicated line: Polling / Polling with time slots
- Dial-up network: ST7 protocol (spontaneous)

Frame parameters

- **Frame format**

- FT1.2
- FT2

- **Acknowledgment type**

- Short acknowledgment (1 byte)
- Long acknowledgment (5 bytes)

- **Repetition factor**

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

- **Max. frame length**

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

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

Network settings

- **Dependence on direction**

Direction dependency of the network

- Duplex
- Half duplex

- **Transmission speed**

Speed at which the CP and modem communicate.

Dial-up network options

- **Cancel parameter**
Number of dialing attempts with unavailable partner until the attempts to establish the connection are finally ended. (0...127)
 - 1...127: Final abort when attempted calls failed n times (no connection or no data transfer).
 - 0: Final abort when a connection was established 127 times without data transfer.
- **Redial**
Number of attempted calls until a disruption is reported.
- **Network password**
Formerly "Customer identification"
The optional network password (formerly "Customer identification") makes sure in dial-up networks that connections are only established to configured partners.
Range of values: 0...65535
Default setting: 0 (no password)

Time-of-day synchronization

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

You specify the synchronization cycle if the parameter is enabled.

Note**Transfer of the setting by stations**

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

Station list

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

2.14 Web server (TIM 1531 IRC)

The Web server of the TIM

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

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

For a description of the content, refer to the manual for the TIM /3/ TIM 1531 IRC (Page 426).

Access rights via "Global security settings"

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

The following preset roles are relevant for Web server access:

- NET Standard
- NET Diagnose

The required rights for diagnostics, access to the Web server and reading and writing data are thus enabled.

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

Access to the Web server and starting Web diagnostics

To be able to connect to the Web server of the TIM, access to the Web server must be enabled for every Ethernet interface, see also section Web server access (Page 70). As default access is disabled.

For information on starting Web diagnostics, see section Web diagnostics of the TIM 1531 IRC (Page 87).

"Web server" parameter group

General

- **Enable Web server on this module**
Enables data processing in the Web server of the TIM and allows access to this data.
- **Allow access only using HTTPS**
Allows access to the Web server only with the secure protocol HTTPS.

Note

"Allow access only using HTTPS" (security function enabled)

Note the following when the "Allow access only using HTTPS" option is enabled in the "Web server" parameter group:

- The data is transferred encrypted.

Requirements

- The roles specified above with the corresponding rights must be assigned to the user.
 - If the firewall is activated, the HTTP/HTTPS protocols must be allowed.
-

Automatic update

- **Enable automatic update**
Enables automatic updating of the displayed values.
If the option is disabled, only the values at the time of connecting to the Web server are displayed.
- **Update interval**
Select the interval at which you require an update of the displayed values.
Default setting: 30. Permitted range: 5...999

Overview of the interfaces

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

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

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

2.15 Web diagnostics of the TIM 1531 IRC

Requirements

- The Web server of the module is enabled in the configuration, "Web server" parameter group, and the interface is selected.
- The interface is enabled for access to the Web server in the configuration, "Ethernet interface > Access to Web server" parameter group.

Starting Web diagnostics

1. Establish a physical connection between the engineering station and the SIMATIC station.
2. Set the PC interface in such a way that the module can be reached.
Further help is available in the "Set PG/PC Interface" dialog box.
3. In the STEP 7 project, click the "Web diagnostics" button under the "Web diagnostics" parameter group to establish the connection to the Web browser of the module.

The content is supplied by the integrated Web server of the module. For information on operation and on the contents, see the manual of the TIM /3/ TIM 1531 IRC (Page 426).

2.16 Partner stations

2.16.1 Importing configuration data

Importing configuration data from STEP 7 V5

The parameters described below are only available for ST7 modules in which the configuration method "Import" was selected in the "Basic settings" parameter group, see section Configuration (Page 56). You will find other requirements below.

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

1. In your STEP 7 Professional project, select the module into which you want to import the SDBs from the STEP 7 V5 project.
2. Select the "Partner stations" parameter group.
3. Click the "Import partner configuration" button.

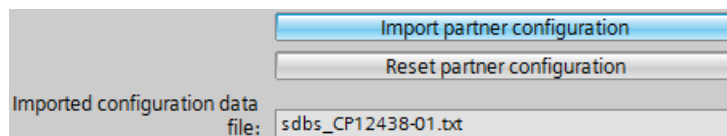


Figure 2-9 Importing the configuration data

The dialog for selecting the file with the configuration data opens

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

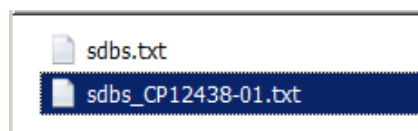


Figure 2-10 Selection of the file "sdbb_xxx.txt" with the configuration data from STEP 7 V5.

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

Import of the proxy configuration data

- **Import partner configuration**
With this button you import the configuration file (SDBs) from the STEP 7 V5 project.
The configuration file must be located in the file system of the engineering station.
- **Reset partner configuration**
With this button, you delete all the configuration data of the module that you previously imported from a text file with the "Import partner configuration" button. With this you reset the partner configuration data to the status of an unconfigured module.
Exception:
If the module configured in STEP 7 used the TD7 variant "TD7onCPU", the created TD7 program blocks are retained after the reset.
- **Imported configuration file**
Shows the name of the text file (SDB) with the configuration data you last loaded.
- **Last import**
Shows the time at which you imported the configuration file.

Please note:

- If you import a second time, the data of the first import will be overwritten.
If you import a second time, always check the configuration, the connections and data points of the module.
- If you change from "Import" to "Configure" following an import, all imported ST7 connections including the corresponding time data will be deleted. Configured data points lose their partners.

Requirements

The import options are only available for communications modules that were previously configured as proxy in STEP 7 V5.

The parameters described below are only available for ST7 modules in which the configuration method "Import" was selected in the "Basic settings" parameter group, see section Configuration (Page 56).

For the supported modules, see section Expansion of SINAUT projects (Page 41).

Imported configuration data

Imported data

When importing the configuration data of a proxy module from STEP 7 V5 into the compatible module in STEP 7 Professional, the ST7-specific subscriber data and the SINAUT connections are imported.

The import file generated in STEP 7 V5 contains the data of the following SDBs:

- **SDB 3202: WAN data**
Data of a serial interface cannot be adopted in a CP 1542SP-1 IRC, see also section Expansion of SINAUT projects (Page 41).
- **SDB 3203: Subscriber data**

- SDB 3207: S7 connection data
- SDB 3208: Ethernet data

Data point information / SINAUT objects

The transfer of STEP 7 V5 configuration data differs depending on the TD7 version used in STEP 7 V5:

- TD7onTIM
No configuration data is converted to data points.
- TD7onCPU
When importing to a TIM 1531 IRC from the information about the blocks assigned in STEP 7 V5 the TD7onCPU blocks of the CPU 1500 are created.
If you revoke the import ("Reset partner configuration", see above) the TD7onCPU blocks are retained.

Data of time-of-day synchronization

Data of time-of-day synchronization is adopted from STEP 7 V5.

With the following modules, the time-of-day data of a serial interface can then be changed in STEP 7 Professional.

- CP 1243-8 IRC as of firmware version V3
- TIM 1531 IRC

Partner stations

After the import, the table shows the following data of all communications partners of the module configured in STEP 7 V5:

- Subscriber number
Subscriber number of the partner
- Subscriber ID
Identifier for the subscriber type:
 - 0 = CPU
 - 1 = TIM
 - 4 = WinCC
 - 7 = SMSC
- Subscriber additional identifier
You can find detailed information in the section of the same name below.
- Number of connections
Number of connections to the respective partner
- Number of partners
Number of communications partners configured for the respective partner.

In the following tables, you can view additional data on each of the partners listed in the "Partner stations" table.

List of partners and connections

- **Select partner**

In the drop-down list, select one of the partners listed in the "Partner stations" table.

"Connections" table

The table displays the following information for the partner selected in the drop-down list:

- **Partner subscriber number**

Subscriber numbers of all communications partners configured for this partner.

- **Interface ID**

Type identifier of the interface used for connection to the respective partner:

- 0 = Backplane bus (connection partner = CPU)
- 1 = Ethernet interface 1
- 2 = Ethernet interface 2
- 3 = Ethernet interface 3
- 4 = Serial interface

- **Length of interface ID**

Length of S7 subnet ID: 6 bytes

- **Length of subnet ID**

Subnet ID (hexadecimal) of the network used for the connection to the partner:

- Byte 0..2: Number of the subnet
- Byte 3..5: Number of the STEP 7 V5 project

The displayed value may be empty, depending on the imported SDB data.

- **CFB reference**

Internally assigned index of a configured S7 connection for the transport tunnel to a local subscriber

- Connection type
Connection identifier (decimal)
 - 128: PBC connection (programmed block communication)
 - 136: WAN connection

Byte assignment:

Bit 0..6

- For Ethernet interfaces:
Bit 0..3:
0 = S7 connection
1 / 2 / 3 = Local connection
5 = MSC connection
Bit 5:
0 = Permanent connection
1 = Temporary connection
Bit 6:
1 = GPRS
- For serial interfaces:
Bit 0..3:
8 = ST7 protocol

Bit 7

- 0 = Local path
- 1 = Remote path
- Length of WAN address
Length of WAN address: 1...6 bytes
- WAN address
 - For connections via Ethernet interfaces:
IP address of the partner
 - For connections via serial interfaces:
WAN address of the partner (0...255)

"Partner list" table

Additional data on the partner selected in the "Select partner" drop-down list is displayed in the table.

- **Partner subscriber number**
Own subscriber number of the partner selected above (partner of the communications module)
- **Connection partner**
Subscriber number of the target partner with which the selected partner communicates.

Subscriber additional identifier

The "Subscriber additional identifier" (32 bits) is displayed in the "Partner stations" table. Depending on the subscriber type of the partner, the subscriber additional identifier contains the following information:

- **CPU (TD7)**

Status of bit 0:

- 0: CPU does not send operating status to partner.
- 1: CPU sends operating status to partner.

The remaining bits are not used for the CPU.

- **TIM (ST7)**

Bit no.	Meaning
Bit 0	DCF77 clock <ul style="list-style-type: none"> • 0: Not available • 1: Available
Bit 1	Connection establishment from communications module to CPU <ul style="list-style-type: none"> • 0: Not permitted • 1: Permitted
Bit 2	Serial interface 1 (WAN driver) <ul style="list-style-type: none"> • 0: Interface not available • 1: Interface available
Bit 3	Serial interface 2 (WAN driver) <ul style="list-style-type: none"> • 0: Interface not available • 1: Interface available
Bit 4...15	Not used
Bit 16...19	Serial interface 1: Network type <ul style="list-style-type: none"> • 0: Dedicated line • 1: Spontaneous network (mobile network) • 2: Dialup network • 3...4: Not used • 5: MSC network • 6...15: Not used
Bit 20...21	Serial interface 1: Network node type <ul style="list-style-type: none"> • 0: Master station • 1: Node station • 2: Station
Bit 22	Serial interface 1: Frame format <ul style="list-style-type: none"> • 0: FT1.2 • 1: FT2
Bit 23	Not used

Bit no.	Meaning
Bit 24...27	Serial interface 2: Network type <ul style="list-style-type: none"> • 0: Dedicated line • 1: Spontaneous network (mobile network) • 2: Dialup network • 3...4: Not used • 5: MSC network • 6...15: Not used
Bit 28...29	Serial interface 2: Network node type <ul style="list-style-type: none"> • 0: Master station • 1: Node station • 2: Station
Bit 30	Serial interface 2: Frame format <ul style="list-style-type: none"> • 0: FT1.2 • 1: FT2
Bit 31	Not used

2.17 Communication with the CPU

Communication with the CPU

Configurable with: TIM 1531 IRC / CP 1243-8 IRC / CP 1542SP-1 IRC

Using the first three parameters, you specify the CPU access of the communications module in the CPU sampling cycle. You will find the structure of the CPU sampling cycle in the section Read cycle (Page 163).

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

- **Cycle idle time**
Wait time between two sampling cycles of the CPU memory area
- **Max. number of write jobs**
Maximum number of write jobs to the CPU memory area within a CPU sampling cycle
- **Max. number of read jobs**
Maximum number of low-priority read jobs from the CPU memory area within a CPU sampling cycle
- **Frame memory size**
Here, you set the size of the frame memory for events (send buffer).
The size of the frame memory is divided equally among all communications partners. For information on the size of the frame memory, see "Performance data and configuration limits".
You will find details of how the send buffer works (storing and sending events) as well as the options for transferring data in the section Process image, type of transmission, event classes (Page 161).

Watchdog bit

- **TIM monitoring / CP monitoring**

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

The module transfers the bit to the CPU every 5 seconds and resets it in the next CPU sampling cycle. The PLC tag of the watchdog bit must be evaluated by the user program.

CP time

Configurable with: CP 1243-8 IRC / CP 1542SP-1 IRC

- **CP time to CPU**

The function allows the CPU to read the time of day of the CP. Using this approach, the CP can synchronize the CPU time.

Procedure:

- The CPU sets the input "Time trigger variable" (BOOL) to 1 with the user program.
- The CP then writes its time to the "CP time variable" (DTL) and resets the "Time trigger variable" value to 0.
- The user program reads the "CP time variable" to set the CPU time.

Recommendation:

Set the "Time trigger variable" no more frequently than once per second to avoid placing an unnecessary communication load on the backplane bus.

2.17.1 CP diagnostics

CP diagnostics

In the parameter group "CP diagnostics", you have the option of providing the CPU with advanced diagnostics data of the CP using PLC tags.

You can display the states of the PLC tags via the Web server of the CPU. You can also use the value of diagnostic tags to send the states of the CP to communication partners via messages.

- **Enable advanced CP diagnostics**

Enable the option to use advanced CP diagnostics.

If the option is enabled, at least the "Diagnostics trigger tag" must be configured.

The following PLC tags for the individual items of diagnostics data can be enabled selectively.

- **Diagnostics trigger tag**

If the PLC tag (BOOL) from the user program of the CPU is set to 1, the CP updates the values of the following PLC tags for the advanced diagnostics.

After writing the current values to the following PLC tags, the CP sets the "Diagnostics trigger tag" to 0, signaling to the CPU that the updated values can be read from the PLC tags.

Note**Fast setting of the diagnostics trigger tag**

Trigger should not be set more often than once per second.

- **Frame memory overflow**
PLC tag (data type Byte) for the send buffer overflow pre-warning. Bit 0 is set to 1 when 80% of the fill level of the send buffer is reached.
- **Frame memory occupation**
PLC tag (data type DWord) for the occupation of the send buffer. The number of saved frames is specified.
- **Current IP address**
PLC tag (data type String) for the current IP address of the interface of the CP.
- **VPN IPsec status**
The PLC tag (BOOL) indicates whether a VPN IPsec tunnel is established:
 - 0 = No tunnel established
 - 1 = Tunnel established
- **Connection to SINEMA Remote Connect**
The PLC tag (BOOL) indicates whether there is a connection to the SINEMA RC server:
 - 0 = No connection established
 - 1 = Connection established

2.17.2 Partner status and Path status

You will find the parameters described below in the following parameter group of modules with the type of communication "ST7":

"Communication with the CPU" > "Partner status"

The following statuses are monitored and written to PLC tags:

- **Partner status**
Availability of a communication partner
- **Path status**
Status of the connections to a communication partner

Partners / Partner name

This is where you define the communication partners of the selected module whose reachability status and path status of the telecontrol connections should be monitored. Partners can be CPUs or a control center application (for example, ST7cc).

All partners with which a telecontrol connection has been configured can be selected in the table.

PLC tags with the various TIM types

The PLC tags in which the statuses are written differ depending on the TIM type:

- **TIM 3V-IE, TIM 4R-IE**
 - **Partner status**
The Partner status is written to a PLC tag of the type Bool (DB, bit memory, output).
For the coding see below.
 - **Path status**
The Path status is written to a PLC tag of the type Byte (DB, bit memory, output).
For the assignment of the PLC tags, see below.
- **TIM 1531 IRC / CP 1243-1 / CP 1243-7 LTE / CP 1243-8 IRC / CP 1542SP-1 IRC**
 - **Partner status / path status**
The two statuses are stored in a common PLC tag of the type Word or UInt (DB, bit memory, output).
The two statuses each occupy one byte of the PLC tag. The coding is written separately for byte 0 (partner status) and byte 1 (path status).

Partner status

- **TIM 3V-IE, TIM 4R-IE**
The bit for partner status is coded as follows (range of values):
 - 0: Partner not reachable
 - 1: Partner reachable
- **TIM 1531 IRC / CP 1243-8 IRC / CP 1542SP-1 IRC**
Byte 0 of the common PLC tag for partner status / path status codes information on the availability of the communications partner, on the existing connections and connection paths and on the status of the send buffer of the TIM.

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

* Partners that support temporary connections are set to 'reachable' if the partner itself terminates the connection.

** For the behavior of the frame memory, refer to the section "Send buffer" in the section Process image, type of transmission, event classes (Page 161).

Path status

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

2.17 Communication with the CPU

A maximum of 2 paths (main and substitute path) to a partner can be configured, see section Creating telecontrol connections for the ST7 communication (Page 140).

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

The PLC tag shows the following:

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

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

- **Byte assignment**

The byte is assigned as follows:

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

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**

In the following table the interfaces of the TIM modules are numbered through from 0 .. 3 and have the following meaning:

- **"0"**
TIM 4R-IE: Ethernet interface 1 (X3)
TIM 1531 IRC: Ethernet interface 1 (X1)
CP 1243-8 IRC / CP 1542SP-1 IRC: Ethernet interface (X1)
- **"1"**
TIM 4R-IE: Ethernet interface 2 (X4)
TIM 1531 IRC: Ethernet interface 2 (X2)
CP 1243-8 IRC: WAN interface (X2)
- **"2"**
TIM 4R-IE: WAN interface 1 (X1)
TIM 1531 IRC: Ethernet interface 3 (X3)
- **"3"**
TIM 4R-IE: WAN interface 2 (X2)
TIM 1531 IRC: WAN interface 1 (X4)

For the CP 1243-8 IRC, the interface numbers "2" and "3" are not relevant.

Status bit 5 / bit 7	Status bit 4 / bit 6	Meaning
0	0	Coding for interface "0"
0	1	Coding for interface "1"
1	0	Coding for interface "2"
1	1	Coding for interface "3"

- **Path status**

- Main path = 1st path, coding by bits 0 and 1
- Substitute path = 2nd path, coding by bits 2 and 3

Status bit 1 / bit 3	Status bit 0 / bit 2	Meaning of bit 1 / bit 3	Meaning of bit 0 / bit 2
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 for the path status

The following table shows an example of the coding options of the path status for the TIM 1531 IRC.

The abbreviations for the interfaces have the following meaning:

- IE1 = Ethernet interface 1 (X1)
- IE2 = Ethernet interface 2 (X2)
- IE3 = Ethernet interface 3 (X3)
- WAN = serial interface (X4)

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 2-1 Coding example for the path status.

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

2.17.3 Manual request

Manual request

Option to manually trigger a single request of a communications partner or a general request

Validity

The function is supported by:

- TIM 1531 IRC V2.4 together with an S7-300 / 400 / 1500 CPU
- CP 1542SP-1 IRC V2.3

Functions

In addition to the requests of the master station during startup, the option offers the possibility of a manually initiated request using a tag of the CPU.

This option supports the following request types:

- General request
Request of all stations with transfer of the process image and all buffered frames
- Accelerated general request
Request of all stations with transfer of the process image only

The following applies generally:

Requests can only be made on a station-specific basis. Requesting of multiple stations must be carried out sequentially.

The request is initiated by the tag "Read request". The request type and the addresses are also specified in this tag. The tag contains a user data type, which you must create beforehand.

The result of the request is written to the tag (Byte) "Read result", which can be evaluated by the CPU or the user.

Tags

The following tags are needed for the request function:

- **Partner request**

A special data type, which you must create as a user data type (UDT), is needed for initiating the request and specifying the request type and the addresses. Proceed as follows:

- Navigate to the "PLC data types" entry of the assigned CPU in the project tree.
- Click the "Add new data type" entry.
- Create the required elements of the UDT (see next section).

UDT_Request							
	Name	Data type	Default value	Accessible ...	Writable ...	Visible ...	Setpoint
	StartFlag	Byte	16#0				
	RequestType	Byte	16#0				
	SubscriberNumber	Word	16#0				

Figure 2-11 Structure of the UDT

- Create a data block in which you insert the previously created UDT.
The UDT can be selected in the drop-down list of the data types under the name you assigned.

For the tag "Partner request", reference the tag of the UDT created for it in the data block you create.

- **Request result**

For the result of the request, create a tag of type Byte and reference it as the tag "Request result".

Structure and coding of the "Partner request" UDT

Structure and contents of the UDT for the ST7 protocol:

Data type	Parameter	Range of values and meaning
Byte	Start flag	The TIM starts the request at a 0 → 1 change. After execution of the request, the TIM resets the value to 0.
Byte	Request type	<ul style="list-style-type: none"> • 1: General request • 2: Accelerated general request
Word	Station address	1 ... 65535

Coding of the "Request result" byte

The results of the request are written to the tag "Request result". The values (hexadecimal) have the following meaning:

Value (hex)	Meaning
No result:	
0x00	#define TCH_DATA_REQUEST_RESULT_NO_RESULT
"Result OK":	

Value (hex)	Meaning
0x01	#define TCH_DATA_REQUEST_RESULT_OK
"Partner not OK":	
0x88	#define TCH_DATA_REQUEST_RESULT_STATION_NOT_OK
Error:	
0x80	#define TCH_DATA_REQUEST_RESULT_ERROR
0x81	#define TCH_DATA_REQUEST_RESULT_STATION_UNDEF
0x82	#define TCH_DATA_REQUEST_RESULT_TYPE_UNDEF
0x83	#define TCH_DATA_REQUEST_RESULT_GROUP_UNDEF
0x84	#define TCH_DATA_REQUEST_RESULT_VARIATION_UNDEF
0x85	#define TCH_DATA_REQUEST_RESULT_TIMEOUT
0x86	#define TCH_DATA_REQUEST_RESULT_ALREADY_ACTIVE
0x87	#define TCH_DATA_REQUEST_RESULT_REFUSED
0x89	#define TCH_DATA_REQUEST_RESULT_REFUSED_BY_PARTNER
0x90	#define TCH_DATA_REQUEST_RESULT_REQUEST_INCOMPLETE

Control of requests

You control the requests in the user program of the CPU.

You set the start flag, which initiates a request, and the respective contents of the parameters "Request type" and "Station address" via the CPU.

Note

Recommendation

Set the start flag after you have set all the other parameters.

For the request of various stations, the relevant contents in each case must be written to the UDT by the user program.

2.17.4 Local operator input (TIM 3V-IE / TIM 4R-IE)

Local operator input

In this parameter group, you can make settings for monitoring of operator input (buttons, etc.) to protect against incorrect entries.

A minimum and maximum duration is specified for monitored operator inputs. The monitoring status is saved in the "PLC tag for operator input monitoring" (byte).

The PLC tag can be used for entries via the following data point types:

- Command output (Cmd01B_S)
- Setpoint output (Set01W_S)
- Parameter entries (Par12D_S)

The Operator input monitoring is recommended particularly when commands are entered via digital inputs, for example using buttons connected to them. This also applies to transferring setpoint and parameter inputs by triggering of a digital input, e.g. by means of a button.

Parameters:

- **Operator input monitoring**
Enables monitoring of operator input. A minimum and maximum duration is specified for monitored operator inputs. The monitoring status is stored in a PLC tag.
- **PLC tag for operator input monitoring**
PLC tag for the monitoring of operator input. Data type byte (DB, memory bit, output).
- **Max. input time**
Maximum duration for manual operator input. Entries that take longer than this value are not evaluated as input.
- **Min. input time**
Minimum duration for manual operator input. Entries that are shorter than this value are not evaluated as input.

Monitoring incorrect input (input time)

Using Operator input monitoring reduces the risk of incorrect inputs when the entries are made via digital inputs. The following times and the code bits are relevant only for operator entries via digital inputs.

- **Min. input time**
For these entries, a minimum input time can be specified, in other words, the button must be pressed for the minimum time. The accidental activation of a button does not then lead to an unwanted transfer. When the minimum input time has elapsed and the button can be released, the "PLC tag for operator input monitoring" shows "entry OK" in bit 0 (see below). A time of at least 1 second is recommended.
If you do not require the parameter, configure 0 (zero).
- **Max. input time**
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 written as "input error" to bit 1 of the PLC tag.
Further hardware entries are not processed as long as the "input error" bit is set.
If you do not require the parameter, configure 0 (zero).

Monitoring for 1 out of n errors

Generally for all operator input via digital inputs. bit memory or data blocks the PLC tag returns the error status "1 out of n error" in bit 2. The error status is set when more than 1 bit is set at the same time.

The error is output in the following situations:

- 1-out-of-n error with "Cmd01B_S"
More than 1 bit was set in the input byte of the command data point type "Cmd01B_S". To increase reliability of command input, only 1 bit may ever be set with this object. If two or more bits are set at the same time, the command input is rejected.
- 1-out-of-n error with all command, setpoint and parameter entries in the fast read cycle
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 read cycle. All command, setpoint and parameter objects in the fast cycle are then subjected to a 1-out-of-n check. 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 if no entry was acquired in at least one fast cycle beforehand.

Note

Resetting the command bit

If commands are entered over a memory or data byte, or a setpoint or parameter entry is enabled by a memory or data bit (trigger signal), the set command bit or trigger signal is automatically reset to zero.

If, however, a 1-out-of-n error is detected, the command bits are not automatically reset. They must then be reset by the user program.

Assignment of the status byte of "PLC tag for operator input monitoring"

Table 2-2 Bit assignment of the status byte

Bit:	.7 *	.6 *	.5 *	.4 *	.3 *	.2	.1	.0
Status:	-	-	-	-	-	1-out-of-n error	Input error	Input OK
For value:	0	0	0	0	0	1	1	1
* Unused bits are written with 0.								

2.18 Time-of-day synchronization

Time-of-day synchronization and security

When security functions are enabled, you need to regularly synchronize the time of day of the communications module.

If you enable the security functions in security CPs, you will find the time parameters under "Security".

Basics of time-of-day synchronization

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

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

When an external time source is used, the connected S7 station can obtain the current time of day both via the CPU and via a communications module (TIM, CP).

Note

Recommendations

- **Time-of-day synchronization only by 1 module of the station**

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

If the CPU takes the time from a communications module, disable time-of-day synchronization of the CPU.

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

- **Longer synchronization cycles with unstable networks**

If a network often has connection problems, you can increase the synchronization cycle. In this way, you avoid the frames being marked as "invalid" and discarded after the synchronization time expected by the time client has elapsed.

Time-of-day concept

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

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

Methods for receiving the time of day

Time-of-day synchronization of the CPU

As the synchronization method, only NTP can be selected for the CPU.

For the time-of-day synchronization of the CPU via a CP, see below.

Time synchronization of the communications modules

The following synchronization methods are available with the telecontrol communications modules (S7-1200 / ET 200SP / TIM 1531 IRC):

- **No time source**
The module is not synchronized.
- **From WAN**
The module adopts the time of day from a communications partner.
ST7 uses the local time of the master station without daylight-saving time designation for the time synchronization of the communications modules.
- **From NTP server / NTP (secure)**
The module obtains the time from one or more configured NTP servers.
 - NTP
NTP with support of RFC5905
NTP servers can be configured manually in the table.
 - NTP (secure)
NTP with support of RFC5905 and RFC5906 with shared key
NTP servers of the type NTP (secure) are configured in the global security settings and can be selected for the CP in the "Security > Time synchronization" parameter group.
- **From local station**
The module takes the time from the CPU.
The CPU 1200 as of V4.2 synchronizes all CMs/CPs of the station with a synchronization cycle of 10 seconds.
Parameters of the CPU:
If the option "CPU synchronizes the modules of the device" is enabled, you can initiate synchronization of all telecontrol CPs of the station with firmware $\geq$ V2.1.77 with the CPU time in a synchronization cycle of 10 seconds.
- **Setting the time of day manually using the WBM (TIM 1531 IRC)**
If you have configured a time source for the TIM, you can also set the time via the WBM; see manual /3/ TIM 1531 IRC (Page 426).

CP time during startup

For the CP 1200, you can find the following option at the bottom in the "Receive time" parameter group:

- Use CPU time before first synchronization by master/master station
The option can be enabled when the "From WAN" time source is configured.
 - When the option is enabled, the CP takes the time from the CPU during startup, if a time has not yet been received from the partner over WAN.
 - If the option is disabled, the CP has no valid time during startup.

For CPs with firmware version $\leq$ V2.0, the CP reads the CPU time once during start-up, even if the option "No time source" is selected as Time source. The subsequent time synchronization then depends on the configured synchronization method.

Saving the time and type of time

In STEP 7, the time is saved as "DTL" data type by the telecontrol communications modules.

Communications modules with the ST7 protocol use the local time of the master station/the master with daylight-saving time designation.

"Time" parameter group: UTC and time zone settings

Validity:

The functions of this parameter group are valid for:

- CP 1243-8 as of firmware version V3.5
- CP 1542SP-1 IRC as of firmware version V2.3
- TIM 1531 IRC as of firmware version V2.4

Parameters:

- Take time zone values with daylight saving time from PLC
When the option is enabled, the communications module takes the time from the CPU.
When the option is disabled, the time zone of the communications module can be configured independent of the CPU.
TIM 1531 IRC: The option is activated if the CPU has been assigned to the TIM (parameter group "Subscriber numbers").
- Time zone
Here you set the local time zone that the communications module will use.
- Use UTC
 - Option enabled:
The module uses UTC in communication with its connection partner. With synchronization via WAN, the communications module expects the time from the partner as UTC.
 - Option disabled:
The communications module uses the local time of the station.

If the communications module receives the time from WAN and synchronizes the CPU, it transmits the time as UTC to the CPU, regardless of whether this option is enabled.
Online diagnostics entries of the station and WBM entries of the TIM 1531 IRC are shown as local time of the PC.

If the communications module does not take the time from the CPU, you can configure the changeover between daylight saving time and standard time:

- Activate daylight saving time
Enables the time changeover (daylight saving/standard time)
- Difference between standard and daylight saving time
Time difference in minutes between daylight saving time and standard time
- Start of daylight saving time / standard time
Here you configure when each time changeover starts:
 - Week of the month
Example:
"First" means "First Sunday in March", if "Sunday" is configured as weekday and "March" as month.
 - Weekday
 - "in" = Month
 - "at" = time of changeover

Forwarding the time from the CP to the CPU

Note

Forwarding the time to the CPU

Depending on the firmware version of the modules involved, the time-of-day of the CP can be forwarded to the CPU in different ways.

- Optional forwarding of the CP time to the CPU using a PLC tag:
 - S7-1200
 - ET 200SP
 - Forwarding of the CP time via the backplane bus to the CPU from the firmware versions mentioned below:
 - S7-1200
 - ET 200SP
-

The forwarding of the CP time to the CPU depends on the firmware version of the CP and the CPU. Note the following behavior.

- **Forwarding of the CP time via PLC tag**

With the following firmware versions, this is the only option for forwarding the clock time from the CP to the CPU:

- S7-1200: CP firmware $\leq$ V2.1.5
- ET 200SP: CP firmware $\geq$ V2.0

With this firmware version, the CP can make the time of day available to the CPU as an option via a PLC tag. When this PLC tag is read cyclically by the CPU, the CPU adopts the CP time. Configure the PLC tag in the parameter group "Communication with the CPU" of the CP.

- **Forwarding of the CP time to the CPU via the backplane bus**

The CP time is automatically forwarded to the CPU, when a synchronization method is configured at the CP, and CP and CPU have one of the following firmware versions:

- S7-1200: CP firmware $\geq$ V3.0 and CPU firmware $\geq$ V4.2

If the option "CPU synchronizes the modules of the device" is enabled for the CPU in "PROFINET interface > Time synchronization", all smart modules of the station are synchronized with the CPU time.

- ET 200SP: CP firmware $\geq$ V2.1 and CPU firmware $\geq$ V2.0

As of firmware version V2.1 of the CP, only one module in the station can be time server. This module distributes the time of day within the station. If you want to have the time of the station synchronized via the CPU, disable the time-of-day synchronization for the CP. Since the CPU automatically adopts the CP time, you no longer require the forwarding option using the PLC tag for these firmware versions.

CPs do not support forwarding the time to connected subnets.

Forwarding time of day by the TIM

The TIM can forward its time of day as follows:

- **To connected networks**

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

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

- **On the assigned CPU**

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

Configuration on the Ethernet interface of the TIM

Time server

1. In the "Receive time" parameter group, configure the TIM that will be the time server and configure the time source using one of the following options:
 - From NTP server
 - From WAN
(take the time from a network)
2. Configure the interface of the TIM via which time frames will be forwarded in the parameter group "WAN settings" as network node type "Master station".

3. In the parameter group "Send time" for the interface from step 2, enable the option "Send time to WAN".
The time frames are forwarded to the connected network.
4. If necessary, enable the "To local station" option in the "Send time" parameter group if the assigned CPU should also be synchronized.
When using TD7onCPU, observe the following:
In this case, also enable forwarding of the time on the interface of the TIM that is networked with the CPU.

Time clients

1. Configure the interfaces of the other TIM modules that will be time slaves in the parameter group "WAN settings" as network node type "Node station" or "Station".
The function is supported for the Ethernet interface and the serial interface.
2. Network the interfaces of the TIM modules involved with each other and with the interface of the time server.
3. For the stations, set the parameters of time-of-day synchronization in the parameter group "Receive time".
4. If necessary, enable the "To local station" option in the "Send time" parameter group if the assigned CPU should also be synchronized.
When using TD7onCPU, observe the following:
In this case, also enable forwarding of the time on the interface of the TIM that is networked with the CPU.

Configuration with the TIM and with WAN networks

Parameter groups for time-of-day synchronization

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

- **TIM**
 - **Receive time**
Here you specify via which of the connected networks the TIM will receive the time of day.
 - Master station TIM: Usually via the control center
 - Node station: Definition of the interface (network node type "Node station") via which the TIM receives the time from the master station TIM.
 - Station: Definition of the interface via which the TIM receives the time from the master station TIM or a node station.
 - **Send time**
 - Master station TIM (time server): Definition of the networks to which the TIM forwards the time.
 - Node station: Definition of the interface (network node type "Master station") via which the TIM forwards the time to lower-level networks.
- **Classic WAN network**
For classic networks, the "Time-of-day synchronization" is enabled in the network's parameter group of the same name. You also specify the synchronization cycle.
The settings for synchronization are then adopted by all connected TIM modules; see below.
The send direction of the time-of-day frames is derived automatically from the network node type of the connected interfaces:
Master station ⇒ Node station ⇒ Station

Configuring time-of-day synchronization via classic WAN networks

TIM modules (time server and clients)

1. In the "Receive time" parameter group, configure the TIM that will be the time server and configure the time source using one of the following options:
 - From NTP server
 - From WAN
(take the time from a network/master station)
2. Configure the interface of the master TIM to which the lower-level stations or node stations are connected with the network node type "Master station".
3. Configure the interfaces of the other TIM modules (time clients) with the network node type "Node station" or "Station".
4. If necessary, enable the "To local station" option in the "Send time" parameter group of the stations if the assigned CPU should also be synchronized.

When using TD7onCPU, observe the following:

In this case, also enable forwarding of the time on the interface of the TIM that is networked with the CPU.

WAN network

1. Network the serial interfaces of the involved TIM modules with the respective WAN.
2. In the parameter group "Time-of-day synchronization" of the WAN, enable the option "Activate time synchronization for WAN".
3. There configure the required synchronization cycle.
4. Enable the time source "From WAN" at the serial interfaces of the stations (time clients). The settings configured for the WAN network are adopted in the following parameter groups of the connected TIM modules:
 - For the time server (master station): "Send time" parameter group (serial interface)
 - For the time clients (node stations / station): "Receive time" parameter group (serial interface)

The send direction of the time-of-day frames is derived automatically from the network node type of the connected interfaces:

Master station ⇒ Node station ⇒ Station

Optional: Multiple time servers

If multiple time servers are connected in the network, you can define a specific subscriber as time server:

1. Configure the telecontrol connections via the interfaces of the two devices.
2. After the telecontrol connections have been created, you can select the partner configured via the connection at the time client.
Selection at parameter "Receive time > Obtain time from partner" of the serial interface

Alternatively:

In the "No or any connection partner" setting, the communications module accepts the time from all connected time servers.

2.19 Time stamp

Time stamp of the ST7 frames

The values of the data points are transferred by the communications module with a time stamp, see also section Creating telecontrol connections for the ST7 communication (Page 140), subsection "Connections to redundant control centers and other properties".

Format of the time stamp

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

The structure of 8 bytes is assigned as follows:

Table 2-3 Assignment of the structure of the time stamp

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

Table 2-4 The assignment of the half byte "time status" (low nibble of byte 7)

Bit No.	Value	Meaning
0	0	Time is invalid
	1	Time is valid
1	0	Standard time
	1	Daylight saving time
2		<i>Not used</i>
3		<i>Not used</i>

2.20 SMSC

Enabling SMSC

If you want to use SMS, enable the SMSC (Short Message Service Center) for the following modules:

- CP 1243-8 IRC
- TIM 3V-IE / TIM 4R-IE

With the TIM 1531 IRC, configuration is not necessary.

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

2.21 I/O addresses (TIM 3V-IE / TIM 4R-IE)

I/O addresses

This is where you specify the address ranges which the TIM occupies in the input/output area from the point of view of the CPU.

These addresses are only relevant for you if you configure TD7onTIM and if the CPU is to be supplied with the date and time by the TIM. In this case, the TIM supplies the time data to the inputs specified here.

The start address and length of the address ranges are assigned by the system. Alternatively, you can change the inputs or outputs.

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.

2.22 MSC authentication

MSC authentication on the different interfaces

When the secure MSC protocol is used (MSC / MSCsec), you need to configure the authentication data for the interfaces of the network type "MSC" (Station, Node station).

Parameter groups

- **MSC authentication - Ethernet interface**
Relevant when the Ethernet interface of the communications module is configured with the following settings:
 - Network type: MSC
 - Network node type: Station / Node station
- **MSC authentication - MD720**
Relevant when the serial interface of the module is configured with the following settings:
 - WAN type: IP-based
The network type is already preset with this setting.
 - Network type: MSC
 - Network node type: Station / Node station
The network node type "Node station" can only be selected with the TIM 1531 IRC.
The network node type "Station" is permanently preset for the CP 1243-8 IRC.
 - Modem type
"Mobile wireless modem MD720" is permanently preset.

With these settings, the serial interface of the station/node station can communicate with the Ethernet interface of the master station via the "MSC" or "MSCsec" protocol.
With the CP 1243-8 IRC, the "TS Module RS-232" must be connected to the serial interface for these settings.

Parameter

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

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

When the station/node station is compiled, the access data of the station/node station is saved for the master station connected in the STEP 7 project.

2.23 E-mail configuration

Configuring e-mails

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

In the message editor ("Messages" entry), you configure the individual e-mails, see section Messages (Page 183).

Requirements for e-mail

Note the following requirements in the CP configuration for the transfer of e-mails:

- The security functions are enabled.
- The time of the CP is synchronized.

For the configuration, you require the data of the SMTP server and the user account:

- Server address, port number, user name, password, e-mail address of the sender (CP)
- With encrypted transfer: Server certificate

E-mail configuration

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

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

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

- Port no. 587
By using STARTTLS, the module sends encrypted e-mails to the SMTP server of your e-mail service provider.
Recommendation: If your e-mail provider offers both options (STARTTLS / SSL/TLS), you should use STARTTLS with port 587.
- Port no. 465
By using SSL/TLS (SMTPS), the module sends encrypted e-mails to the SMTP server of your e-mail service provider.

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

On configuration of the passwords, see Character set for user names, passwords and messages (Page 189).

Importing the certificate with encrypted transfer

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

Use the certificate by taking the following steps:

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

For the procedure, refer to the section Certificate manager (Page 131).

2.24 Subscriber numbers

Subscriber numbers

For the meaning of the subscriber numbers, see section Communication via the SINAUT ST7 protocol (Page 18).

- **Assigned CPU**

With the following communications modules, which are located in the same rack as the CPU, the local CPU is automatically assigned to the module:

- S7-1200 telecontrol CPs
- TIM 3V-IE...
- TIM 4R-IE

For TIME modules that use TD7onTIM and are not in the same rack as a CPU, you must assign the TIM a CPU with which it is networked via the drop-down list. This affects:

- TIM 4R-IE Standalone
- TIM 1531 IRC

The TIM 1531 IRC can be assigned a CPU of the following SIMATIC families:
S7-300, S7-400, S7-1500/1500 R/H, ET 200SP

The drop-down list only shows the stations that are networked with the TIM.

Note on TD7onCPU:

If TD7onCPU is used for SINAUT communication, no CPU can be assigned to the TIM.

- **Subscriber number of the assigned CPU**

Assign the subscriber number of the assigned CPU.

CPUs are assigned numbers starting with 1 by default.

- **Subscriber number of the communications module**

Assign the subscriber number of the module.

The communications modules are assigned numbers starting with 1000 by default.

The address must be unique within a subnet and within the STEP 7 project.

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

Note:

You configure the WAN address of a serial interface in the parameter group "WAN parameters", see section Serial interface of the TIM (Page 71).

Other subscriber numbers

- **ST7cc / ST7sc / ST7 ScadaConnect**
For the ST7 application of the master station PC, you configure the subscriber number in the telecontrol connection.
- **RTU3000C**
 - **Subscriber number (RTU)**
Here, you assign the subscriber number of the RTU as it was configured in the WBM of the RTU.
 - **Subscriber number of the CPU (RTU)**
Here, you assign the subscriber number of the CPU as it was configured in the WBM of the RTU.

For information on configuring the RTU, see Communication types (Page 53).

2.25 Secure communication between CPU and module

Secure communication with certificates

You need to assign the certificate of the CPU to a communications module in the following cases.

Required for:

- Communications module in separate rack
- CPU 1500 / CPU 1500 SP as of version V2.9

Procedure

Requirements:

- The CPU is assigned to the module under "Subscriber numbers".
- You are logged in with the role "NET Administrator".
- The password for protection of confidential PLC configuration data has been created in the CPU. The communication certificate with ID "1" and common name of subject "PLC name/Communication-1" is thus created in the certificate manager of the CPU.
- Additionally for ST7:
 - You use TD7onTIM (Telecontrol configuration = "Configure").
When using TD7onCPU, the procedure described below is not necessary.

The communication certificate must be assigned to the module that is assigned to the CPU.
There are various procedures for this:

Using the global certificate manager for the CPU

The local communication certificate of the CPU is hereby recreated.

1. Select the CPU: "Protection & Security > Certificate manager".
2. Select the "Use global security settings for certificate manager" check box.

3. After the following dialog is confirmed, the communication certificate is removed from the certificate manager of the CPU.
4. To create a new communication certificate, select the "Protection & Security > Connection mechanisms" menu in the general settings.
5. Create a new communication certificate under "PLC communication certificate". To do so, click on "..." on the right.
6. Click "Add" in the subsequent window.
7. Confirm the settings in the "Create certificate" window with "OK".
Result: The certificate is available in the certificate manager of the CPU and in the global certificate manager.

To assign the certificate to the module:

1. In the project navigation, open "Security settings" > "Security features" > "Certificate manager".
2. Select the "Device certificates" tab.
3. Right-click on the communication certificate and select "Assign to".
4. Click the "Assigned" check box for the module to which the certificate should be assigned.
5. For "Usage", switch from "Device certificate" to "Trusted certificate".

Assignment can also take place via the local certificate manager of the module.

Using the local certificate manager of the CPU

The local communication certificate of the CPU is used directly, but needs to be assigned to the communications module via export and import.

1. Select the CPU and choose the "Protection & Security > Certificate manager" menu in the general settings.
2. Right-click on the communication certificate and select "Export".
3. Save the certificate.

To assign the certificate to the module:

1. In the project navigation, open "Security settings" > "Security features" > "Certificate manager".
2. Select the "Device certificates" tab.
3. Import the communication certificate.
4. Right-click on the communication certificate and select "Assign to".
5. Click the "Assigned" check box for the module to which the certificate should be assigned.
6. For "Usage", switch from "Device certificate" to "Trusted certificate".

Assignment can also take place via the local certificate manager of the module.

2.26 SNMP

SNMP

The scope of performance of the modules is given in the relevant manual.

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

SNMP

- **"Enable SNMP"**
If the option is enabled, communication via SNMP is released on the device. As default, SNMPv1 is enabled.
If the option is disabled, queries from SNMP clients are not replied to either via SNMPv1 or via SNMPv3.
- **"Use SNMPv1"**
Enables the use of SNMPv1 for the device. For information on the configuration of the required community strings see below (SNMPv1).
- **"Use SNMPv3"**
Enables the use of SNMPv3 for the device. For information on the configuration of the required algorithms see below (SNMPv3).

SNMPv1

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

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

- **"Reading community string"**
The string is required for read access.
Leave the preset string "public" or configure a string.
- **"Allow write access"**
If the option is enabled write access to the device is released and the corresponding community string can be edited.
- **"Writing community string"**
The string is required for write access and can also be used for read access.
Leave the preset string "private" or configure a string.

Note

Security of the access

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

SNMPv3

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

- **"Authentication algorithm"**
Select the authentication method to be used from the drop-down list.
- **"Encryption algorithm"**
Select the encryption method to be used from the drop-down list.

User management

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

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

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

2.27 Security / Certificates

2.27.1 Security user

Creating a security user

You need the relevant configuration rights to be able to configure security functions. For this purpose, you need to create at least one security user with the corresponding rights.

Navigate to the global security settings > "User and roles" > "Users" tab.

1. Create a user and configure the parameters.
2. Assign this user the role "NET Standard" or "NET Administrator" in the area below "Assigned roles".

After logging on, this user can make the necessary settings in the STEP 7 project.

In the future, continue to log on as this user when working on security parameters.

Security functions of the S7-1500 V3.0 together with TIM 1531 IRC V2.4

To be able to use TLS connections of the TIM 1531 IRC together with the S7-1500-CPU V3.0 you need to enable "Legacy access control" for the CPU (parameter group "Protection & Security > Access control").

Security user for S7-1500 (general)

This section does not apply to the TIM 1531 IRC V2.4.

To be able to use the security functions of the S7-1500 V3.0 in general, you need to create a security user to whom a role with full access for the S7-1500 is assigned in the project.

1. Create a new role with full access for the S7-1500.

Roles				
	Name	Description	Runtime time	
	Engineering administrator	System-defined role "Engineering ...	30	M
	Engineering standard	System-defined role "Engineering ...	30	M
	HMI Administrator	System-defined role "HMI Adminis...	30	M
	HMI Operator	System-defined role "HMI Operator"	30	M
	HMI Monitor	System-defined role "HMI Monitor"	30	M
	HMI Monitor Client	System-defined role "HMI Monitor ...	30	M
	HMI Online Configuration Engineer	System-defined role "HMI Online C...	30	M
	NET Administrator	System-defined role "NET Adminis...	30	M
	NET Standard	System-defined role "NET Standard"	30	M
	NET Diagnose	System-defined role "NET Diagnos...	30	M
	NewRole	User-defined role	30	Min
	<Add new role>			

Engineering rights

Runtime rights

User-specific runtime rights

Function rights categories

Function rights

Runtime rights

S7-1500 V3.1

PLC_st7-3_jeCM

Name

Group

☐ User authentication o...

OPC UA

☒ Full access

Access level

2. Create a new user to whom you assign the new role.

Users				
		User name	Password	Runtime t
	☐	Anonymous		
	☒	admin	*****	300
	☒	NewUser	*****	300
		<Add new user>		

Assigned user groups		Assigned roles		Assigned rights	
Assigned roles					
	Assigned to	Name	Description		
	☐	Engineering administrator	System-defined role "Engineering ...		
	☐	Engineering standard	System-defined role "Engineering ...		
	☐	HMI Administrator	System-defined role "HMI Adminis...		
	☐	HMI Operator	System-defined role "HMI Operator"		
	☐	HMI Monitor	System-defined role "HMI Monitor"		
	☐	HMI Monitor Client	System-defined role "HMI Monitor ...		
	☐	HMI Online Configuration Engineer	System-defined role "HMI Online C...		
	☐	NET Administrator	System-defined role "NET Adminis...		
	☐	NET Standard	System-defined role "NET Standard"		
	☐	NET Diagnose	System-defined role "NET Diagnos...		
	☒	NewRole	User-defined role		

2.27.2 MSC authentication

MSC authentication on the different interfaces

When the secure MSC protocol is used (MSC / MSCsec), you need to configure the authentication data for the interfaces of the network type "MSC" (Station, Node station).

Parameter groups

- **MSC authentication - Ethernet interface**
Relevant when the Ethernet interface of the communications module is configured with the following settings:
 - Network type: MSC
 - Network node type: Station / Node station
- **MSC authentication - MD720**
Relevant when the serial interface of the module is configured with the following settings:
 - WAN type: IP-based
The network type is already preset with this setting.
 - Network type: MSC
 - Network node type: Station / Node station
The network node type "Node station" can only be selected with the TIM 1531 IRC.
The network node type "Station" is permanently preset for the CP 1243-8 IRC.
 - Modem type
"Mobile wireless modem MD720" is permanently preset.

With these settings, the serial interface of the station/node station can communicate with the Ethernet interface of the master station via the "MSC" or "MSCsec" protocol.
With the CP 1243-8 IRC, the "TS Module RS-232" must be connected to the serial interface for these settings.

Parameter

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

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

When the station/node station is compiled, the access data of the station/node station is saved for the master station connected in the STEP 7 project.

2.27.3 SMSC

Enabling SMSC

If you want to use SMS, enable the SMSC (Short Message Service Center) for the following modules:

- CP 1243-8 IRC
- TIM 3V-IE / TIM 4R-IE

With the TIM 1531 IRC, configuration is not necessary.

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

2.27.4 Log settings - Filtering of the system events

Communications problems if the value for system events is set too high

If the value for filtering the system events is set too high, you may not be able to achieve the maximum performance for the communication. The high number of output error messages can delay or prevent the processing of the communications connections.

In "Security > Log settings > Configure system events", set the "Level:" parameter to the value "3 (Error)" to ensure the reliable establishment of the communications connections.

Use of SYSLOG only with 1 VPN connection

If you want to use SYSLOG with level 7 (debug) via VPN connections, this is only possible with a single configured VPN connection.

2.27.5 VPN

2.27.5.1 CP as passive subscriber of VPN connections

Setting permission for VPN connection establishment with passive subscribers

If the CP is connected to another VPN subscriber via a gateway, you need to set the permission for VPN connection establishment to "Responder".

This is the case in the following typical configuration:

VPN subscriber (active) ⇔ gateway (dyn. IP address) ⇔ Internet ⇔ gateway (fixed IP address) ⇔ CP (passive)

Configure the permission for VPN connection establishment for the CP as a passive subscriber as follows:

1. In STEP 7, go to the devices and network view.
2. Select the CP.
3. Open the parameter group "VPN" in the local security settings.
4. For each VPN connection with the CP as a passive VPN subscriber, change the default setting "Initiator/Responder" to the setting "Responder".

2.27.5.2 VPN (Virtual Private Network)

VPN - IPsec

Virtual Private Network (VPN) is a technology for secure transportation of confidential data in public IP networks, for example the Internet. With VPN, a secure connection (IPsec tunnel) is set up and operated between two secure IT systems or networks via a non-secure network.

The IPsec tunnel forwards all data even from protocols of higher layers (HTTP, FTP, etc.).

The data traffic between two network components is transported unrestricted through another network. This allows entire networks to be connected together via a neighboring or intermediate network.

Properties

- VPN forms a logical subnet that is embedded in a neighboring (assigned) network. VPN uses the usual addressing mechanisms of the assigned network, however in terms of the data, it transports its own frames and therefore operates independent of the rest of this network.
- VPN allows communication of the VPN partners with the assigned network.
- VPN is based on tunnel technology and can be individually configured.
- Communication between the VPN partners is protected from eavesdropping or manipulation by using passwords, public keys or a digital certificate (authentication).

Areas of application

- Local area networks can be connected together securely via the Internet ("site-to-site" connection).
- Secure access to a company network ("end-to-site" connection)
- Secure access to a server ("end-to-end" connection)
- Communication between two servers without being accessible to third parties (end-to-end or host-to-host connection)
- Ensuring information security in networked automation systems
- Securing the computer systems including the associated data communication within an automation network or secure remote access via the Internet
- Secure remote access from a PC/programming device to automation devices or networks protected by security modules via public networks.

Cell protection concept

With Industrial Ethernet Security, individual devices or network segments of an Ethernet network can be protected:

- Access to individual devices and network segments protected by security modules is allowed.
- Secure connections via non-secure network structures becomes possible.

Due to the combination of different security measures such as firewall, NAT/NAPT routers and VPN via IPsec tunnels, security modules protect against the following:

- Data espionage
- Data manipulation
- Unwanted access

2.27.5.3 Creating a VPN tunnel for S7 communication between stations

Requirements

To allow a VPN tunnel to be created for S7 communication between two S7 stations or between an S7 station and an engineering station with a security CP, the following requirements must be met:

- The two stations have been configured.
- The CPs in both stations must support the security functions.
- The Ethernet interfaces of the two stations are located in the same subnet.

Note

Communication also possible via an IP router

Communication between the two stations is also possible via an IP router. To use this communications path, however, you need to make further settings.

Procedure

To create a VPN tunnel, you need to work through the following steps:

1. Creating a security user
If the security user has already been created: Log on as this user.
2. Enable the "Activate security features" option
3. Creating the VPN group and assigning security modules
4. Configure the properties of the VPN group
5. Configure local VPN properties of the two CPs

You will find a detailed description of the individual steps in the following paragraphs of this section.

Select "Activate security features"

After logging on, you need to select the "Activate security features" check box in the configuration of both CPs.

You now have the security functions available for both CPs.

Creating the VPN group and assigning security modules

1. In the global security settings, select the entry "Firewall" > "VPN groups" > "Add new VPN group".
2. Double-click on the entry "Add new VPN group", to create a VPN group.
Result: A new VPN group is displayed below the selected entry.

3. In the global security settings, double-click on the entry "VPN groups" > "Assign module to a VPN group".
4. Assign the security modules between which VPN tunnels will be established to the VPN group.

Note

Current date and current time on the CP for VPN connections

Normally, to establish a VPN connection and the associated recognition of the certificates to be exchanged, the current date and the current time are required on both stations.

The establishment of a VPN connection to an engineering station that is also the telecontrol server at the same time (TCSB installed), runs as follows along with the time of day synchronization of the CP:

On the engineering station (with TCSB), you want the CP to establish a VPN connection. The VPN connection is established even if the CP does not yet have the current time. Otherwise the certificates used are evaluated as valid and the secure communication will work.

Following connection establishment, the CP synchronizes its time of day with the PC because the telecontrol server is the time master if telecontrol communication is enabled.

Configure the properties of the VPN group

1. Double-click on the newly created VPN group.
Result: The properties of the VPN group are displayed under "Authentication".
2. Enter a name for the VPN group. Configure the settings of the VPN group in the properties. These properties define the default settings of the VPN group that you can change at any time.

Note

Specifying the VPN properties of the CPs

You specify the VPN properties of the CPs in the "Security" > "Firewall" > "VPN" parameter group of the relevant module.

Result

You have created a VPN tunnel. The firewalls of the CPs are activated automatically: The "Activate firewall" check box is selected as default when you create a VPN group. You cannot deselect the check box.

Download the configuration to all modules that belong to the VPN group.

2.27.5.4 VPN communication with SOFTNET Security Client (engineering station)

Set up VPN tunnel communication between the SOFTNET Security Client and the CP as described in section Creating a VPN tunnel for S7 communication between stations (Page 127).

VPN tunnel communication works only if the internal node is disabled

Under certain circumstances the establishment of VPN tunnel communication between SOFTNET Security Client and the CP fails.

SOFTNET Security Client also attempts to establish VPN tunnel communication to a lower-level internal node. This communication establishment to a non-existing node prevents the required communication being established to the CP.

To establish successful VPN tunnel communication to the CP, you need to disable the internal node.

Use the procedure for disabling the node as explained below only if the described problem occurs.

Disable the node in the SOFTNET Security Client tunnel overview:

1. Remove the checkmark in the "Enable active learning" check box.
The lower-level node initially disappears from the tunnel list.
2. In the tunnel list, select the required connection to the CP.
3. With the right mouse button, select "Enable all members" in the shortcut menu.
The lower-level node appears again temporarily in the tunnel list.
4. Select the lower-level node in the tunnel list.
5. With the right mouse button, select "Delete entry" in the shortcut menu.

Result: The lower-level node is now fully disabled. VPN tunnel communication can be established.

2.27.5.5 Establishment of VPN tunnel communication between the CP and SCALANCE M

Create a VPN tunnel between the CP and a SCALANCE M router as described for the stations.

VPN tunnel communication will only be established if you have selected the check box "Perfect Forward Secrecy" in the global security settings of the created VPN group ("VPN groups > Authentication").

If the check box is not selected, the CP rejects establishment of the tunnel.

2.27.6 SNMP

SNMP

The scope of performance of the modules is given in the relevant manual.

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

SNMP

- **"Enable SNMP"**
If the option is enabled, communication via SNMP is released on the device. As default, SNMPv1 is enabled.
If the option is disabled, queries from SNMP clients are not replied to either via SNMPv1 or via SNMPv3.
- **"Use SNMPv1"**
Enables the use of SNMPv1 for the device. For information on the configuration of the required community strings see below (SNMPv1).
- **"Use SNMPv3"**
Enables the use of SNMPv3 for the device. For information on the configuration of the required algorithms see below (SNMPv3).

SNMPv1

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

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

- **"Reading community string"**
The string is required for read access.
Leave the preset string "public" or configure a string.
- **"Allow write access"**
If the option is enabled write access to the device is released and the corresponding community string can be edited.
- **"Writing community string"**
The string is required for write access and can also be used for read access.
Leave the preset string "private" or configure a string.

Note

Security of the access

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

SNMPv3

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

- **"Authentication algorithm"**
Select the authentication method to be used from the drop-down list.
- **"Encryption algorithm"**
Select the encryption method to be used from the drop-down list.

User management

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

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

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

2.27.7 Certificate manager

Certificate manager of SIMATIC NET devices

SIMATIC NET communications modules generally use the global certificate manager. You can find this in the project navigation under "Security settings > Security features".

All certificates that are created and displayed by the configuration with the SIMATIC NET communications module can also be found in the global certificate manager.

Certificates for authentication

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

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

Examples of services and functions that require certificates:

- Communication of the TIM 1531 IRC with the higher-level CPU
The connection of the TIM to the CPU uses secure communication via TLS for all telecontrol protocols supported by the TIM.
For communication via TLS, you need to use a newly created CPU certificate and specify it to the TIM in the "Subscriber numbers" parameter group.
For information on the procedure, refer to the section Handling the TLS certificates (Page 132).
- Secure transfer of e-mails
An SSL certificate is created for the module to transfer data via SSL/TLS. It is visible in STEP 7 in the local certificate manager and under "Global security settings > Certificate manager > Device certificates" with the "WebServer" service.

If a STEP 7 communications module is to communicate securely with non-Siemens partners, the relevant certificates of the communications partners must be exchanged. To do this, follow these steps:

1. Import the third-party certificates of the external communications partner
Global security settings of the project > "Trusted certificates and root certification authorities"
2. Assigning a certificate to the communications module

Assigning certificates

If you create certificates for modules in STEP 7 retroactively or import third-party certificates into STEP 7, you need to assign them to the respective module, either via:

- Local certificate manager of the communications module > "Certificates of the partner devices"
- Global security settings of the project > "Trusted certificates and root certification authorities"

For information on the procedure, refer to the section Handling the TLS certificates (Page 132).

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

2.27.8 Handling the TLS certificates

Communication between CPU and telecontrol module

CP: Communication via backplane bus

Since the CPU and CP are in the same rack, communication between them runs via the backplane bus. The CPU is automatically assigned to the telecontrol CP.

TIM 1531 IRC: Communication with CPUs via Ethernet

The TIM 1531 IRC is not inserted in the rack of the CPU. The TIM can be connected with one or more CPUs via Ethernet. The CPU used for telecontrol communication must be assigned to the TIM.

TIM 1531 IRC: TLS communication with the CPU

The connection of the TIM to the CPU uses secure communication via TLS for all telecontrol protocols supported by the TIM.

For communication via TLS, you need to use a newly created CPU certificate and specify it to the TIM in the "Subscriber numbers" parameter group.

Order of steps

When you create a new certificate for the CPU and then assign the CPU to the TIM, the certificate of the CPU is entered automatically in the subscriber configuration of the TIM.

If you first assign the CPU to the TIM and then create the CPU certificate, you need to look up the ID of the CPU certificate and enter it manually in the subscriber configuration of the TIM or delete the CPU from the "Assigned CPU" field and then assign it again.

Validity and requirements

Validity

The description below applies to the following modules:

- TIM 1531 IRC V2.3 as of firmware version V2.3
Together with CPU 1500 as of firmware version V2.9
- CP 1542SP-1 IRC as of firmware version V2.2
Together with ET 200SP CPU as of firmware version V2.9
- CP 1543-1 as of firmware version V3.4
Together with CPU 1200 as of firmware version V4.5

Requirements

The following requirements must be met to use the certificates as described below:

- As STEP 7 project user, you have at least the rights of the "NET Administrator" role.
For more on this, see "Security settings > Users and roles > Assigned roles".
If you use the security functions of the S7-1500 V3.0, you also need a created user with the required rights; see Security user (Page 121).
- The devices have the required minimum firmware version.
- The configuration data of the CPU is protected.
For more on this, see "Protection & Security > Protection of confidential PLC configuration data"
- The desired telecontrol protocol is enabled for the TIM under "Communication types".
- CPU and TIM 1531 IRC are networked.
- The certificate of the CPU has been created and is in the global certificate manager; see section Certificate manager (Page 131).

Configuring TLS for telecontrol connections

You can configure secure communication via TLS for telecontrol connections of communications modules:

Network data editor > "TeleControl" tab)

A prerequisite for CPs is activation of the security functions.

TLS connections with project partners

1. First create the telecontrol connections.
2. Select the main connection and the "Secure Communication (TLS)" parameter group in the connection properties.
3. Activate the "Enable secure communication" option.

When all certificates of the connection partners are present, the "own certificate ID" and the "partner certificate ID" are automatically applied to the entire connection, including sub-connections, for Siemens devices.

TLS connections with third-party devices

If you want to use secure communication via TLS for telecontrol connections with third-party devices, you need to perform some additional steps.

1. First, set the secure listener port for TLS or check the setting (Basic settings > Secure listener port). The port number is needed for the connection partner.
2. Create a certificate for the third-party device in STEP 7.
Alternatively, import the certificate into STEP 7, if there is one for the third-party device. See below for information on importing and creating certificates (certificates for TLS connections with third-party devices).
3. Assign the third-party certificate to the STEP 7 connection partner as "Trusted certificate". See below for the procedure (certificates for TLS connections with third-party devices).
4. Export the device certificate of the communications module, the device certificate of the third-party device (if this was created in STEP 7) and the associated CA certificate from STEP 7 and import it into the third-party device or its configuration tool.
5. Create the telecontrol connection and activate the "Enable secure communication" option.
6. Transfer the IDs to the connection parameters.
 - Enter the ID of the imported or created third-party certificate (from the global certificate manager) in the "Partner certificate ID" field.
 - Enter the ID of the STEP 7 connection partner certificate in the "Own certificate ID" field.

Certificates for TLS connections with third-party devices**Importing a third-party certificate or creating and assigning it**

Alternatively, you can make the third-party certificate available to the STEP 7 project:

- **Import**
You can save and import the certificate of the third-party device, if there is one, in the file system of the engineering station and in STEP 7.
To import, open the "Trusted certificates and root certification authorities" tab in the global certificate manager, click in a free row and open the "Import" shortcut menu.
You must then assign the imported certificate to the communications module (see below).
- **Create**
Alternatively, you can create a certificate for the third-party device in STEP 7 and import it into the third-party device.
To do this, follow these steps:
 1. Open the global certificate manager in the project navigation:
"Security settings > Security features > Certificate manager > tab Device certificates"
 2. Click on the "Create" shortcut menu in a free row.
The "Create certificate" dialog opens.

3. Select the following options for the certificate of the third-party device:

- Usage: TLS Client / Server
- Certificate authority (CA): Self signed
- Common name of subject: Enter the name of the third-party device.

Adapt the other parameters to the functionality of the third-party device.

Recommended settings:

- Encryption method: RSA
- Key length: 2048
- Hash algorithm: sha256

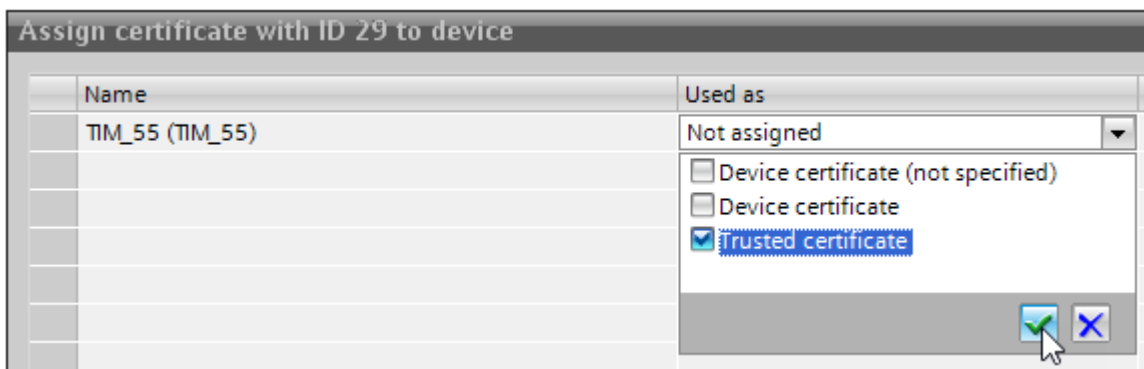
4. Click "OK" to close the dialog.

The certificate appears in the table.

You will need the certificate ID for the assignment and the telecontrol connection.

• Assign

1. In the global certificate manager, select the imported or newly created certificate and open the "Assign" shortcut menu.
2. In the list, select the module the third-party device will communicate with via the telecontrol connection.
3. In the "Used as" row in the ("Not assigned") cell, select the "Trusted certificate" option and click on the green checkmark.



4. Close the dialog with OK.

You will need the certificate ID for the telecontrol connection.

Creating a telecontrol connection and entering a certificate ID

1. Create a telecontrol connection as described above.
2. In the connection properties, select the "Secure Communication (TLS)" parameter group and activate the "Enable secure communication" option.

3. Enter the following values manually:
 - Partner certificate ID: ID of the imported or manually created certificate for the third-party device
 - Own certificate ID: ID of the certificate of the communications module
4. Check the port numbers and third-party device parameters.
 - The Secure Listener port is configured in the "Basic settings" parameter group of the respective module.
 - Configure the network node type in the connection properties ("Third-party device parameters" parameter group).
If the third-party device is the master, select "Master station".

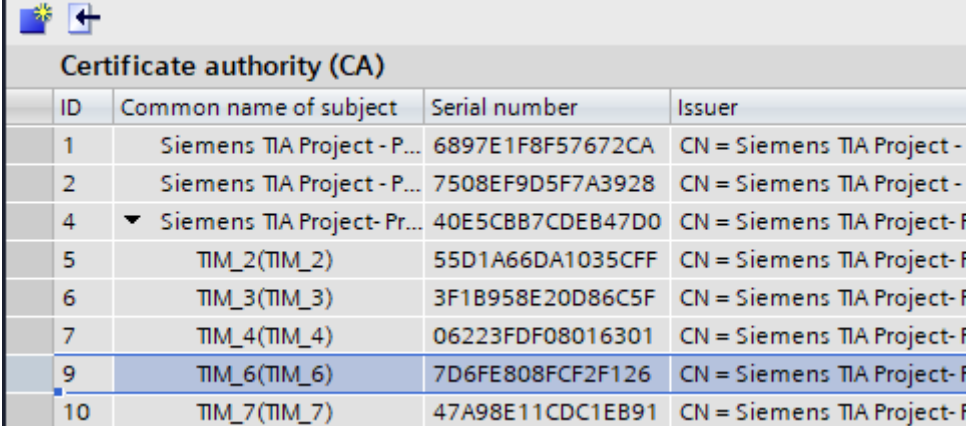
Exporting a STEP 7 certificate for a third-party device and importing it in a third-party device

For the TLS connection with the third-party device, you need to export a device certificate of the communications module created in STEP 7 and the associated CA certificate.

1. Device certificate for the third-party device created in STEP 7
 - In the global certificate manager, open the "Device certificates" tab.
 - Select the device certificate for the third-party device and click the "Export certificate" shortcut menu.
 - Save the certificate in the file system of the engineering station.
You can change the default file format "*.der".
You can find a description of the functions of the certificate file formats in the "Exporting certificates" section in the help system.
2. Device certificate of the communications module
 - In the "Device certificates" tab, select the device certificate of the communications module as partner certificate for the third-party device.
 - Make the "Issuer" column fully visible.
You need the name of the issuer to export the signing CA certificate.
 - Export the selected device certificate of the module using the "Export certificate" shortcut menu.

3. Signing CA certificate

- Change to the "Certificate authority (CA)" tab.
- Select the CA certificate that signed the device certificate of the communications module. When you expand the CA certificate, all derived device certificates, including that of the communications module, are displayed below it.



Certificate authority (CA)			
ID	Common name of subject	Serial number	Issuer
1	Siemens TIA Project - P...	6897E1F8F57672CA	CN = Siemens TIA Project -
2	Siemens TIA Project - P...	7508EF9D5F7A3928	CN = Siemens TIA Project -
4	▼ Siemens TIA Project-Pr...	40E5CBB7CDEB47D0	CN = Siemens TIA Project-f
5	TIM_2(TIM_2)	55D1A66DA1035CFF	CN = Siemens TIA Project-f
6	TIM_3(TIM_3)	3F1B958E20D86C5F	CN = Siemens TIA Project-f
7	TIM_4(TIM_4)	06223FDF08016301	CN = Siemens TIA Project-f
9	TIM_6(TIM_6)	7D6FE808FCF2F126	CN = Siemens TIA Project-f
10	TIM_7(TIM_7)	47A98E11CDC1EB91	CN = Siemens TIA Project-f

Figure 2-12 Global certificate manager: The CA certificate has the ID 4 here, the device certificate has the ID 9.

- Export the CA certificate via the "Export certificate" shortcut menu.
4. Import all saved certificates as trusted certificates into the third-party device or its configuration tool.
The certificates are required for secure communication during runtime.

Change certificate: Subject Alternative Name

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

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

2.28 Protection (TIM 1531 IRC)

For information on handling certificates for use of TLS for the telecontrol connections, see section Security / Certificates (Page 121).

2.28.1 Protection

Protection functions

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

NOTICE**Configuring an access level does not replace the know-how protection**

Configuring access levels prevents unauthorized changes to the module by restricting the download rights.

This does not, however, provide write or read protection for blocks on a memory card. Use the know-how protection to protect the code of blocks on the memory card.

The table of access levels

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

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

You can configure the following access levels:

- **Full access (no protection)**
The configuration and the blocks can be read and modified by anybody.
- **Read access**
With this access level, without entering the password, only read access to the hardware configuration and the blocks is possible; in other words, you cannot download the blocks or hardware configuration to the TIM without entering the password. Without the password, writing test functions and firmware updates are also not possible.
- **No access (complete protection)**
If the module is completely protected, neither read nor write access to the hardware configuration and blocks is possible.

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

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

Behavior of a password-protected module during operation

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

Before an online function is executed, a check is made to establish whether or not it is permitted. If there is password protection, you will be prompted to enter the password.

Example:

The module was configured for read access and you want to use the "Modify tags" function. Since this is write access, the configured password must be entered before the function can be executed.

The functions protected by the password can only be executed by one PG/PC at any one time. Another PG/PC cannot log on.

The access rights to the protected data apply for the duration of the online connection or until the access rights are canceled again with "Online > Delete access rights".

Each access level allows unrestricted access to certain functions even without entering a password, for example identification using the "Accessible devices" function.

2.28.2 Configuring access protection

Configuration

You can enter several passwords setting up different access rights for different user groups.

The passwords are entered in the table so that precisely one access level is assigned to each password.

The "Access level" column shows how the password takes effect.

Example:

You select the access level "No access (complete protection)" for the module and enter your own password for each of the access levels higher up the table.

For users that do not know any of the passwords, the module is completely protected.

For users who know one of the set passwords the effect depends on the table row in which the password is located:

- The effect of the password in row 1 "Full access (no protection)" is as if the CP was unprotected. Users that know this password have unrestricted access to the module.
- The effect of the password in row 2 "Read access" is as if the module was write-protected. Despite knowing the password, users that know this password only have read access to the module.
- The effect of the password in row 3 "No access (complete protection)" is as if the CP was write and read protected. Users that know this password only have read access to the module.

Procedure

Follow the steps below to set the parameters for the access levels of the module:

1. Open the module properties in the Inspector window.
2. Open the "Protection" entry in the navigation panel.
A table with the possible access levels is displayed in the Inspector window.
3. Select the required access level in the first column of the table. The green check mark in the columns on the right of the particular access level indicate which operations are still possible without entering the password.

4. If you have selected an access level other than "Full access":
 - Assign a password for full access in the "Password" column in the first row (full access).
 - Repeat the selected password in the "Confirm password" column to protect against incorrect entries.
 - Make sure that the password is adequately secure; in other words, that it does not include a pattern that can be machine read!
 - The entry of the password in the first row "Full access (no protection)" is obligatory and allows a user who knows the password unrestricted access to the module, regardless of the selected access level.
5. As necessary, assign other passwords to the required access levels if the selected access level permits this.
6. Download the hardware configuration so that the access level takes effect.

Result

The hardware configuration and the blocks are protected from unauthorized online access according to the set access level. If an operation cannot be executed without a password due to the set access level, a dialog appears prompting entry of a password.

2.29 Creating telecontrol connections for the ST7 communication

Single and redundant connections

For ST7 communication of a communications module, you need to create at least one telecontrol connection in the "Network data" editor.

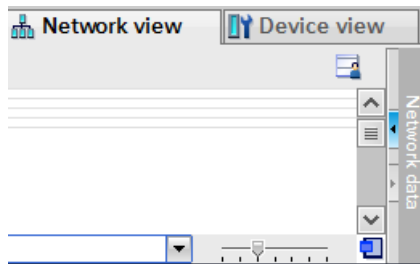
If two subscribers can be reached via different networks, to establish path redundancy for modules with multiple ST7 capable interfaces you create a maximum of two connections between the two subscribers. You specify redundant connections in the "Connection path" table (see below).

Not all communications modules have serial interfaces or only one Ethernet port with ST7 capability. These modules cannot use redundant paths.

Opening the editor "Network data" > "Telecontrol" tab

To open the editor, follow the steps below:

1. Open the network view of the project.
On the right you will find the collapsed "Network data" editor.



2. Open the "Network data" editor using the arrow symbol.
The editor is displayed with several tabs, on the left the "Network overview" tab.
3. Expand the editor until the "Telecontrol" tab appears.
In this tab you configure the telecontrol connections.

Creating telecontrol connections for the ST7 communication of the module

For communication between two ST7 subscribers, you require a telecontrol connection. To create the connection, follow the steps below:

1. In the "Starting point" box, click on a free row in the "Telecontrol" tab.
The drop-down list shows the configurable starting points that can use the ST7 protocol:
 - S7 stations: Assigned CPU
 - PC stations: Application (e.g. ST7cc / ST7sc)

For connections between an application on the master station PC and an S7 station, always select the application as the starting point because it can only be selected in the list of starting points.

2. Select the starting point of the connection by double-clicking.
The connection will then be included in the list of telecontrol connections.
The name of the connection is preset, but you can change it.
3. Now in the "Endpoint" box select the connection partner from the drop-down list.

Boxes with a missing or bad configuration are shown on a red background, in the example the representation "TC_Connection_3".

Network overview		Connections		I/O communication		VPN		TeleControl							
ST7		DNP3		IEC 60870											
Telecontrol connections															
Connection		Starting point		Start subscriber		Endpoint		End subscriber							
 Conn_1		 1500		2		 300		1							
 Conn_100		 PC.ST7cc		3		 1500		2							
<Add new>															
  															
Configured connection paths															
Priority		Name		Module name		Subscriber number		Networks							
1		Conn_1-1		1500 > TIM_1 > TIM 4R-IE_1 > 300		2 > 1002 > 1001 > 1		PN/IE_1 > PN/IE_2 > Backplane							
2		Conn_1-2		1500 > TIM_1 > TIM 4R-IE_1 > 300		2 > 1002 > 1001 > 1		PN/IE_1 > WAN_1 > Backplane							
Connection path															
Position		Module name		Subscriber number		Station name		Interface		Interface type		Mode		Network	
		1500		2		S71500/ET200..		PROFINET-Schnittstelle_1		Ethernet		Neutral		PN/IE_1	
		TIM_1		1002		TIM_1		Ethernet-Schnittstelle_3		Ethernet		Neutral		PN/IE_1	
		TIM_1		1002		TIM_1		Ethernet-Schnittstelle_1		MSC		Master		PN/IE_2	
		TIM 4R-IE_1		1001		S7300/ET200M		Ethernet-Schnittstelle_1		MSC		Station		PN/IE_2	
		TIM 4R-IE_1		1001		S7300/ET200M						None		Backplane	
		300		1		S7300/ET200M						None		Backplane	

Figure 2-13 "Network data" editor, "Telecontrol" tab

Specification of the connection sequence

Then specify the connection sequence of the created telecontrol connection.

- Above the second table "Configured connection paths" click the paths symbol . The dialog "Add connection paths" opens and the possible connection path are searched for.
- Select one of the possible connection paths in the top table. The connection path(s) are shown with details in the second table.
 - If several connections are displayed in the upper table, you can select one or two redundant connections.
 - If no connection is displayed in the upper table, there is a configuration error in the corresponding stations or networks. In this case, close the dialog with the "Close" button and complete the configuration.
- If you want to choose the selected connection path as a "configured connection" click on "Add". "Information" in the lower part of the dialog shows whether the connection was added or it is already configured.
- Close the dialog with the "Close" button if the configured connections correspond to the demands of the project. The connection details of every configured connection are displayed in the lower table "Connection path".

The "Configured connection paths" and "Connection path" tables

"Configured connection paths" table

This table shows the configured connections (connection paths) of a telecontrol connection.

Projektierte Verbindungen

Priority	Name	Module name	Subscriber number	Networks
1	Route_1	ST7cc_1 > TIM_11 > PLC_11	5 > 1001 > 1	PN > PN/IE_1
2	Route_2	ST7cc_1 > TIM_12 > TIM_11 > PLC_11	5 > 1002 > 1001 > 1	PN > > WAN_1 > PN/IE_1

<

Figure 2-14 Redundant connection with incorrectly configured connection path (Route_2)

Connections shown on a red background are invalid.

In the example the PC station was connected via the PC CP to the MSC network between TIM_11 and TIM_12 which is not allowed.

"Connection path" table

When checking the configured connections, the "Connection path" table supports you.

For every configured connection, the detailed connection path is shown here.

The "Position" column shows a station symbol with a symbol for the connection point (starting point, node, endpoint). Based on the color of the symbols for the connection endpoint you can recognize connections that are not permitted.

- Blue symbols for the connection point: Valid connections
- Red symbols for the connection point: Invalid connections

Symbol	Meaning
	Starting point
	Node - input
	Node - output
	Endpoint
	Examples of invalid connection points

Bad connections

In the example the connection path "Route_2" was originally created from the application "ST7cc_1" (PC_1) via the MSC network to TIM_12. This continued via the dedicated line network from TIM_12 to TIM_11.

A connection path from the application via an MSC network or via modules with ST7 capability is not permitted.

The connection of the PC station to the MSC network was deleted in the meantime, so that the connection points "ST7cc_1" and "TIM_12" are invalid and their symbols are red.

Redundant telecontrol connections

When different networks between two subscribers have been created in the network configuration the different connection paths are shown in the "Add connection paths" table.

If you only want to use one connection, select only this connection in the table and click "Add".

If you only want to establish a redundant connection, select two connection paths in the table and click "Add". You specify the main and substitute paths between the subscribers after closing the dialog using the "Configured connections" table.

Main and substitute path of a redundant connection

When you have created two connection paths for a telecontrol connection, the first of the two is the main path and the second the substitute path. This is displayed in the "Priority" column of the "Configured connections" table.

You can change the priority of the two connection paths later, i.e. the specification of the main and substitute path. The arrow keys serve this purpose: 

Delete invalid or redundant connections

If you have non-permitted or unwanted redundant connections you need to delete one connection. Follow the steps outlined below:

1. To delete a connection in the table "Configured connections", select the connection you do not need.
2. Delete the connection using the shortcut menu "Delete".

Rules for connection configuration

Note the following rules for connection configuration:

- An MSC connection can only run via the first Ethernet interface [X1] of a TIM or via the serial interface of a module with the setting "IP-based".
- Connections between a PC application and a module can only have the network type "Neutral" (S7 connections).
- For connections between a redundant control center (ST7cc / ST7sc) and a module, create two PC stations with one application each and network both with the communications module.

In the "Network data" editor, create a telecontrol connection between one of the two PC stations and the communications module.

Then, in the properties dialog of the telecontrol connection, assign the redundant control center application to this connection (see below).

- Two connections to one partner (endpoint) via the same interface of a module (start point) are not permitted.
- A connection via an inconsistent network is invalid.
Examples:
 - Subscriber with incompatible modems
 - Incompatible settings of two modems in a connection
 - Incompatible settings between modem and network parameters
 - Connections via nodes that are not configured as node stations.
 - A connection via a network in which two MD3 modems communicate with each other and that are configured with 1200 bps / half duplex / AT mode is invalid.

Properties dialog of the telecontrol connections

If you select a connection in the table "Telecontrol connections" at the top in the "Network data" editor, three parameter groups are displayed in the "Properties" tab of the Inspector window. Here you can check and, if necessary, correct the connection configuration and configure the following properties:

- **General**

Here, you will find the basic properties of a telecontrol connection.

You can change the subscriber number of the connection starting point and endpoint. This affects the configuration data of the subscriber.

- **Redundant ST7cc/sc**

If you use a redundant control center in your project (ST7cc / ST7sc), you must first create two PC stations with application and network these with the communications module via Ethernet.

Then you create a telecontrol connection from the application of a PC station (starting point) and the communications module (end point).

Afterwards, assign the second application in the redundancy group to this telecontrol connection.

- Add redundant ST7cc/cc partner

Enable the option if you want to use the selected telecontrol connection for a redundant master station application.

- Redundant application

Select the second application from the "Redundant application" drop-down list.

- **Destination subscriber properties**

The parameter group relates to the destination subscriber that was configured as the endpoint of the connection.

- **General request supervision time**

This is the maximum time that may be required by a destination subscriber (communications module capable of ST7) to respond fully to a general request (GR). If the GR response has not arrived completely at the requesting partner when the supervision time has expired, a message is entered in the diagnostics buffer of the module.

The general request supervision time is 900 seconds.

- **Frames to destination subscriber with time stamp**

Regardless of the configuration all frames are sent to the destination subscriber with a time stamp.

2.30 Data points

2.30.1 Data point configuration

Data point-related communication with the CPU

No program blocks need to be programmed for telecontrol modules with data point configuration to transfer user data between the station and communications partner.

The data areas in the memory of the CPU intended for communication with the communications partner are configured data point-related on the module. Each data point is linked to a PLC tag or the tag of a data block.

Requirement: Created PLC tags and/or data blocks (DBs)

PLC tags or DBs must first be created correspondingly on the CPU to allow configuration of the data points.



WARNING

Writing values to outputs

- PLC tags
When referencing to PLC tags, note that the values are written immediately to the outputs of the CPU without first being processed by the user program with write access. Writing values has a direct influence on the process.
- DB variables
When referencing to DB variables, written values are only used when processed by the user program.

The PLC tags for data point configuration can be created in the standard tag table or in a user-defined tag table. All PLC tags intended to be used for data point configuration must have the attribute "Visible in HMI".

Address areas of the PLC tags are input, output or bit memory areas on the CPU.

Note

Number of PLC tags

Note the maximum possible number of PLC tags that can be used for data point configuration, see Configuration limits in Volume 1.

The formats and S7 data types of the PLC tags that are compatible with the protocol-specific data point types of the module can be found in the section Datapoint types (Page 155).

Access to the memory areas of the CPU

The values of the PLC tags or DBs referenced by the data points are read and transferred to the communications partner by the module.

Data received from the communications partner is written by the module to the CPU via the PLC tags or DBs.

Configuring the data points and messages in STEP 7

You configure the data points in STEP 7 in the data point and message editor. You can open both editors alternatively as follows:

- Selecting the communications module
Shortcut menu "Open the data point and messages editor"
- Via the project navigation:
Project > directory of the relevant station > Local modules > TIM
By double-clicking on the entry, the data point or message editor opens.

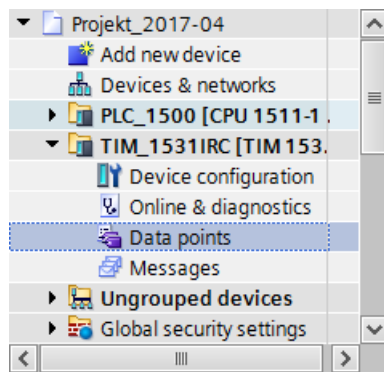


Figure 2-15 Configuring data points and messages

Using the two entries to the right above the table, you can switch over between the data point and message editor.

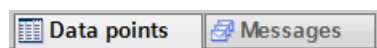


Figure 2-16 Switching over between the two editors

Creating objects

With the data point or message editor open, create a new object (data point / message) by double clicking "<Add object>" in the first table row with the grayed out entry.

A preset name is written in the cell. You can change the name to suit your purposes but it must be unique within the module.

		Name	PLC tag
1		DataPoint	"Tag_1-BI"
2		DataPoint_1	"Tag_2-BQ"
3		DataPoint_2	"Tag_1-BI"

Figure 2-17 Data point table

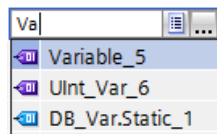
You configure the remaining properties of every object using the drop-down lists of the other table columns and using the parameter boxes shown at the bottom of the screen.

Assigning data points to their data source

After creating it, you assign a new data point to its data source. Depending on the data type of the data point a PLC tag can serve as the data source.

For the assignment you have the following options:

- Click on the table symbol  in the cell of the "PLC tag" column.
All configured PLC tags and the tags of the created data blocks are displayed. Select the required data source with the mouse or keyboard.
- Click the symbol .
A selection list of the configured PLC Tags and the blocks is displayed. From the relevant table, select the required data source.
- In the name box of the PLC tag, enter part of the name of the required data source.
All configured PLC tags and tags of the data blocks whose names contain the letters you have entered are displayed.



Select the required data source.

Note

Assignment of parameter values to PLC tags

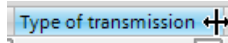
The mechanisms described here also apply when you need to assign the value of a parameter to a PLC tag. The input boxes for the PLC tag (e.g.: PLC tag for partner status) support the functions described here for selecting the PLC tag.

Arranging and copying objects

As with many other programs in the data point or message editor you can also arrange the columns, sort the table according to your requirements and copy and insert objects.

- Arrange columns
If you click on a column header with the left mouse button pressed, you can move the column.
- Sorting objects
If you click briefly with the left mouse button on a column header, you can sort the objects of the table in ascending or descending order according to the entries in this column. The sorting is indicated by an arrow in the column header.
After sorting in descending order of a column the sorting can be turned off by clicking on the column header again.

- Adapting the column width
You can reach this function with the following actions:
 - Using the shortcut menu that opens when you click on a column header with the right mouse key.
"Optimize width", "Optimize width of all columns"
 - If you move the cursor close to the limit of a column header, the following symbol appears:



When it does, click immediately on the column header. The column width adapts itself to the broadest entry in this column.

- Showing / hiding columns
You call this function using the shortcut menu that opens when you click on a column header with the right mouse key.
- Copying, pasting, cutting and deleting objects
If you click in a parameter box of an object in the table with the right mouse key, you can use the functions named with the shortcut menu (copy, paste, cut, delete).
You can paste cut or copied objects within the table or in the first free row below the table.

Enable/disable data point validation

When new data points are created or during an import, the data point index, object number and object channel are checked for consistency. With inconsistent numbers, the affected fields are highlighted in red in the data point table.

If you want to temporarily disable the check, you can switch off this function with the following button above the data point table:



These parameters are checked for consistency during compilation, at the latest.

Exporting and importing data points

To simplify the engineering of larger plants, you can export the data points of a configured module and import them into other modules in the project. This is an advantage particularly in projects with many identical or similar stations or data point modules.

You can also access the export/import function:

- Via the shortcut menu of the module in the network or device view

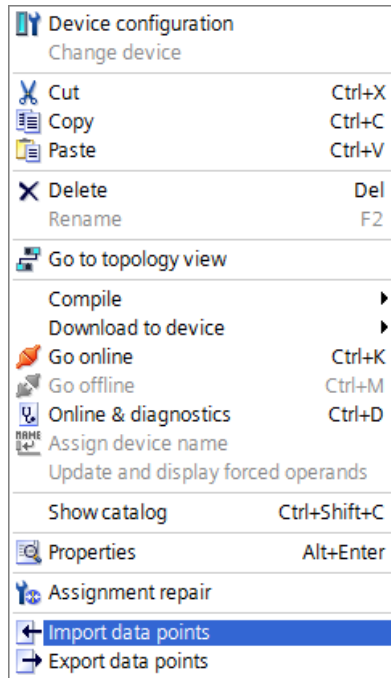


Figure 2-18 Shortcut menu of the module

- Using the two arrow symbols above the data point list

When it is exported the data point information of a module is written to a CSV file.

It is possible to import data points of an older project into a newer project as of STEP 7 V18. If missing parameters were not added in the CSV file, the missing information is filled with default values during the import. In such cases, check the data points carefully.

Recommendation:

Migrate your STEP 7 project to the latest STEP 7 version, if available.

Export

When you call the export function, the export dialog opens. Here, you select the module or modules of the project whose data point information needs to be exported. When necessary, you can export the data points of all modules of the project at one time.

In the export dialog, you can select the storage location in the file directory. When you export the data of a module you can also change the preset file name.

When you export from several modules, the files are formed with preset names made up of the station name and module name.

The file itself contains the following information in addition to the data point information:

- Module name
- Module type

- CPU name
- CPU type

Editing the export files

You can edit the data point information in an exported CSV file. This allows you to use this file as a configuration template for many other stations.

If you have a project with many stations of the same type, you can copy the CSV file with the data points of a fully configured module for other as yet unconfigured stations and adapt individual parameters to the particular station. This saves you having to configure the data points for every module in STEP 7. Instead, you simply import the copied and adapted CSV file to the other modules of the same type. When you import this file into another module, the changed parameter values of the CSV file are adopted in the data point configuration of this module.

The lines of the CSV file have the following content:

- Line 1: ,Name,Type,TIA-Version;<Designation of STEP 7 version>
This line must not be changed.
- Line 2: PLC,<CPU name>, <CPU type>,
Meaning: PLC (designation of the station class), CPU name, CPU type
Only the elements <CPU name> and <CPU type> may be changed.
The CPU type must correspond exactly to the name of the CPU in the catalog.
- Line 3: Module,<module name>,<module type>,
Meaning: Module (Designation of the module class), module type, module name
Only the elements <module name> and <module type> may be changed.
Be careful when changing the module names if you want to import data points into several modules (see below).
The module type must correspond exactly to the name of the module in the catalog.
- Line 4: Parameter names (English) of the data points
This line must not be changed.
See below for the meaning (under "Parameters").
- Lines 5..n: Values of the parameters according to line 4 of the individual data points
You can change the parameter values for the import into a new communications module.

Parameter

Name and meaning of the parameters of the exported CSV file:

- Name: Data point name
- PLC tag: CPU tag
- Datapoint type: Data point type
- Value monitoring: Value monitoring
- Object number: Object number
- Index: Object channel
- Partner Object Number: Partner object number
- Type of transmission: Type of transmission

- Priority in the scan cycle: Read cycle
- Threshold value trigger: Threshold value trigger
- Threshold value: Threshold
- Time trigger: Time trigger
- Time Trigger Cycle: Trigger
- Event trigger: Event trigger
- Trigger tag: Trigger tag
- Transmission mode: Transmission mode
- Meanvalue generation: Mean value generation
- Unipolar transfer: Unipolar transfer
- Fault suppression time: Fault suppression time
- Smoothing factor: Smoothing factor
- Set limit value 'high': Set limit value 'high'
- Limit value 'high': Limit value 'high'
- Set limit value 'low': Set limit value 'low'
- Limit value 'low': Limit value 'low'
- Control Code: Control Code (Command options tab)
- Pulse Limit: Max. number of pulses
- Command output time: Command output time
- Mirroring of local value: Mirroring of local value
- PLC tag for local value: PLC tag for local value
- Partners of datapoint: Partners enabled at the data point with which a telecontrol connection is configured.
The partner number is specified for third-party devices.
Multiple partners are separated by commas and placed in quotation marks.
Example: "PLC_1,Partner_1,PLC_4"
- Enable archiving: Enable archiving ("Trigger" tab)
Validity: TIM 1531 IRC
- Integration interval: Integration interval (Analog value preprocessing tab)
- Status trigger: Status trigger
- Timestamp trigger: Time stamp trigger
- S7 Repetition Factor: (is only exported for informational purposes, but the value is not imported again)
- S7 Data Type Value: (is only exported for informational purposes, but the value is not imported again)

Importing into a module

Before importing the data points make sure that the PLC tags required for the data points have been created.

Note that when you import a CSV file all the data points existing on the module will be deleted and replaced by the imported data points.

Select a module and select the import function from the shortcut menu of the module. The import dialog opens in which you select the required CSV file in the file directory.

If the information on the assignment of the individual data points to the relevant PLC tags matches the assignment in the original module, the data points will be assigned to the corresponding PLC tags.

When you import data points into a module, but some required PLC tags have not yet been created in the CPU, the corresponding data point information cannot be assigned. In this case, you can subsequently create missing PLC tags and then assign them the imported data point information. The "Assignment repair" function is available for this (see below).

If the names of the PLC tags in the module into which the import is made have different names than in the module that exported, the corresponding data points cannot be assigned to your PLC tags.

Importing into several modules

You can import the data points from several modules into the modules of a different project. To do this in the import dialog select all the required CSV files with the control key.

Before importing the data points, make sure that the respective stations have been created with CPUs of the same name, modules of the same name and PLC tags of the same name.

When you import the corresponding stations of the project are searched for based on the module names in the CSV files. If a target station does not exist in the project or the module has a different name, the import of the particular CSV file will be ignored.

Restrictions for the import of data points

In the following situations the import of data points will be aborted:

- An attribute required by the module is missing in the CSV file to be imported.
Example: If a data point to be imported uses a time trigger, the import will be aborted if no time-of-day synchronization was configured for the module.
- The telecontrol protocol used by the module differs from that of the original module.

Only when importing into several modules:

- The import is aborted when a module or CPU name is different from the data in the CSV file.

During the import, a log file indicating any errors is generated.

Compatible modules

Data points can be imported and exported between compatible modules.

Modules with the same telecontrol protocol are compatible with each other:

- ST7
 - CP 1243-8 IRC, CP 1542SP-1 IRC
 - TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE / TIM 1531 IRC

Assignment repair

If you have named the PLC tags in a station into which you want to import differently from the station from which the CSV file was exported, the assignment between data point and PLC tag is lost when you import.

You then have the option to either rename the existing PLC tags appropriately or add missing PLC tags. You can then repair the assignment between unassigned data points and PLC tags. This function is available either via the shortcut menu of the module (see above) or with the following icon to the upper left in the data point editor: 

If a PLC tag with a matching name is found for a data point by the repair function, the assignment is restored. However the data type of the tag is not checked.

After the assignment repair make sure that you check whether the newly assigned PLC tags are correct.

Recommendation:

Before the data point import, copy the tag table or the data block from the source CPU from which the data points are exported into the target CPU into whose communications module the exported data points are to be imported.

2.30.2 Datapoint types

During the configuration of the user data to be transferred by the module, each data point is assigned a protocol-specific data point type. The data point types supported by the module along with the compatible S7 data types are listed below. They are grouped according to format (memory requirements).

Note

Effect of the change of arrays for data points

If an array is modified later, the data point must be recreated.

Data point types of the "ST7" protocol

The direction of the data transmission can be seen in the "Object" column:

- Sending objects have the suffix "_S".
- Receiving objects have the suffix "_R".

Table 2-5 Supported data point types and compatible S7 data types

Format (memory requirements)	Data point type	S7 data type	Operand area	Data points (channels) per object	Object
Bit	Digital input	Bool	I, Q, M, DB	1 .. 8	Bin08X_S
	Digital output	Bool	I, Q, M, DB	1 .. 8	Bin08X_R
	Command input	Bool	I, Q, M, DB	1 .. 8	Cmd08X_S
	Command output	Bool	I, Q, M, DB	1 .. 8	Cmd08X_R
Byte	Digital input	Byte, USInt ³⁾	I, Q, M, DB	4	Bin04B_S
	Digital output	Byte, USInt ³⁾	Q, M, DB	4	Bin04B_R
	Command input	Byte, USInt ³⁾	I, Q, M, DB	1	Cmd01B_S
	Command output	Byte, USInt ³⁾	I, Q, M, DB	1	Cmd01B_R
Integer (16 bits) (‘Int’ with sign)	Analog input	Int	I, Q, M, DB	4	Ana04W_S
	Mean value input	Int	I, Q, M, DB	4	Mean04W_S
	Analog output	Int	Q, M, DB	4	Ana04W_R
	Mean value output	Int	Q, M, DB	4	Mean04W_R
	Setpoint output ¹⁾	Int, UInt, Word	Q, M, DB	1	Set01W_R
	Setpoint input ¹⁾	Int, UInt, Word	Q, M, DB	1	Set01W_S
Counter (16 bits)	Counter input	UInt, Word ³⁾	I, Q, M, DB	1	Cnt01D_S
	Counter input	UInt, Word ³⁾	I, Q, M, DB	4	Cnt04D_S
	Counter output	UDInt, DWord ³⁾	I, Q, M, DB	1	Cnt01D_R
	Counter output	UDInt, DWord ³⁾	I, Q, M, DB	4	Cnt04D_R
Floating-point number (32 bits)	Analog input	Real ³⁾	M, DB	4	Ana04R_S
	Analog output	Real ³⁾	M, DB	4	Ana04R_R
Data block (4 .. 48 bytes)	Data input	ARRAY [0...11] of DInt / UDInt / DWord / Real ^{2) 3)}	DB	12	Dat12D_S
	Data output		DB	12	Dat12D_R
	Parameter input ¹⁾		DB	12	Par12D_S
	Parameter output ¹⁾		DB	12	Par12D_R
Data block (1 .. 12 bytes)	Data input	Byte, USInt, Int, Word; UInt; DWord, UDInt; Real, DInt; DWord ³⁾	I, Q, M, DB	1 .. 12	Dat12x1D_S
	Data output		I, Q, M, DB	1 .. 12	Dat12x1D_R
	Parameter input ¹⁾		I, Q, M, DB	1 .. 12	Par12x1D_S
	Parameter output ¹⁾		I, Q, M, DB	1 .. 12	Par12x1D_R

¹⁾ See below, section "1: Mirroring back"

²⁾ See below, section "2: Data block via array"

³⁾ USInt, UInt, Real and UDInt are not supported by S7-300/400.

1: Mirroring

With some data point types, you can activate mirroring of the current value of the station to the master.

In these objects, the local value can be monitored for change and the change transferred to the master station. A change to a local value can be caused by manual operator input on site, for example.

To allow the value resulting from local events or interventions to be transferred to the master station, the data point in question requires a channel for mirroring back.

You configure the mirroring back function at the two corresponding objects in the station and in the master station.

Configuration using the example of the object pair of a CP:

- **Station**
Enable the "Value monitoring" option in the "General" tab for the receiving object (Par12..._R).
The object then becomes a receiving and sending object.
If the receiving object has received data from the partner and forwarded it to the CPU, it monitors the value of the CPU in the following cycles. If the object detects a value change (e.g. due to manual operator input), it sends the value back to the partner.
- **Master station**
Enable the "Mirroring of local value" option in the "Command options" tab for the sending object (Par12..._S).
The object then becomes a sending and receiving object.
You write the value mirrored back by the station to the "PLC tag for local value" in the sending master station object.

Remember that you need to interconnect the mirrored value in the master station with a separate variable of the data point.

- **CP / TIM 1531 IRC**
Mirroring back of the local values to the partner can be configured for the following data point types:
 - Parameter output (Par12D_R / Par12x1D_R)
For the data point in the station, enable the option "General" > "Value monitoring".
 - Parameter input (Par12D_S / Par12x1D_S)
For the data point in the master station, enable the option "Command options" > "Value mirroring".
Configure the "PLC tag for local value" to which the value mirrored by the station is to be written.
- **TIM 3V-IE / TIM 4R-IE**
Mirroring of the current local data to the partner can be configured for the following data point types:
 - Setpoint input/output (Set01W_R / Set01W_S)
 - Parameter input/output (Par12D_R / Par12D_S,)

When the "Setpoint input mode" option is enabled in the "Local setpoint input" tab, the respective mirroring back function is enabled.
For information on the configuration, see section "Local setpoint input" tab (Page 164).

2: Data block via array

With the ARRAY data type, blocks of data from contiguous memory areas up to a size of 4 .. 48 bytes can be transferred.

Compatible components of ARRAY are DInt, UDIInt, DWord or Real. The components within an array must be of the same type.

Format of the time stamp

For information on the format of the time stamp, refer to the section Time stamp (Page 113).

2.30.3 ST7 data points

Objects of TD7 software

The telecontrol communication of modules that use the SINAUT ST7 telecontrol protocol is handled by the TD7 software.

The data to be transferred from the SINAUT objects is alternatively configured or parameterized:

- Via data points (TD7onTIM)
- Via the program blocks of the "Telecontrol ST7" library (TD7onCPU)

For information on the two variants of the TD7 software, refer to the section TD7 software: Use and variants (Page 32).

Data objects

The sending and receiving of process data is configured with the aid of standardized data objects. According to the two transmission directions, these are divided into:

- **Send objects (.._S)**
Data objects for acquiring and sending data
Their names have the ending "_S" (send).
- **Receive objects (.._R)**
The data objects for receiving and outputting data
Their names have the ending "_R" (receive).

The names of the data objects are identical in TD7onTIM and TD7onCPU. In terms of functionality, they are compatible with each other.

Object numbers

Each data object has an object number. When new data points are created, the numbers are assigned in ascending order by default. The numbers can be changed.

Rules:

- The object numbers of a module must be unique.
- The lowest object number is 1.

Object channels

Each data object has one or more object channels, see section Datapoint types (Page 155).

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 command 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.

Rules:

- The channel numbers of an object must begin with 1 and be assigned consecutively without a gap.
- All channels of an object must have the same data point type.
- All channels of an object must have the same partner.

2.30.4 "General" tab

General

You will also find the most important parameters in the first tab of the data point editor in the data point table.

- **Data source**
 - For the assignment of the PLC tabs, see section Data point configuration (Page 147).
 - For the data point type, see Datapoint types (Page 155).
- **Send parameters**
See following sections
 - **Type of transmission**
See section Process image, type of transmission, event classes (Page 161).
 - **Read cycle**
See section Read cycle (Page 163).
 - **High priority**
If the option is enabled, the transfer has higher priority (important process data). The values of the data point are transferred before those of lower priority data points. In dial-up networks the high priority leads to connection establishment only when the value of the data point is transferred with the transfer mode "Spontaneous (direct transfer)" (see "Trigger" tab).
 - **Volatile**
Validity: TIM 3V-IE / TIM 4R-IE
Command/setpoint/parameter outputs
If the option is enabled, the following applies for this data point: If the target subscriber cannot be reached, frames already saved in the send buffer are deleted and no new frames are entered in the send buffer for this subscriber.
This prevents outdated commands from being sent to the target subscriber in the event of a recurring connection.
 - **Value monitoring**
Validity: TIM 1531 IRC
Parameter output (Par12D_R / Par12x1D_R)
If this option is enabled, the value of the variable is monitored for changes on the CPU. This is usually the station CPU.
At the assigned sending object (Par12..._S) of the partner (usually the master station), you need to enable the "Mirroring of local value" parameter under "Command options" and write the mirrored value to a separate variable.
On a value change in the station, e.g. due to manual operation on location, the local value is mirrored back to the data point of the master station and written to the variable of the master station.
- **Object**
 - **Object number**
The object number of the data point must be unique. The maximum permitted number of data points per SINAUT object must not be exceeded.
Refer to the column "Number of data points per SINAUT object" in the section Datapoint types (Page 155).

- **Object channel**
Channel number of the SINAUT object
The channel numbers of an object must begin with 1 and be assigned in ascending order without a gap.
- **Partner object number**
Object number of the data point on the partner.
Per object number, you need to specify a unique object number of the communications partner. This results in unique pairs of "Object number" and "Partner object number".
See also section Partner stations (Page 183).

2.30.5 Process image, type of transmission, event classes

Storage of values

As a rule the values of all data points are stored in the image memory of the module. Values in the image memory are transferred only after being called by master station TIM.

Events are also stored in the send buffer and can be transferred unsolicited.

The image memory, the process image of the module

The image memory is the process image of the TIM. All the current values of the configured data points are stored in the image memory. New values of a data point overwrite the last stored value in the image memory.

The values are sent only after a query by the communications partner - see below "Transfer after call" in the "Types of transmission" section - or along with a message from the send buffer that needs to be transferred immediately.

The send buffer

The send buffer of the TIM is the memory for the individual values of data points that are configured as an event. You will find the size of the send buffer in the manual of the relevant module.

The capacity of the send buffer is divided up equally for all enabled partners.

If the connection to a communications partner is interrupted, the individual values of the events are retained in the buffer. When the connection returns, the buffered values are sent. The frame memory operates chronologically; in other words, the oldest frames are sent first (FIFO principle).

If a frame was transferred to the communications partner, the transferred value is deleted from the send buffer.

If frames cannot be transferred for a longer period of time and the send buffer is threatening to overflow, the "Forced image mode" applies.

- If the send buffer reaches a fill level of 90%, the module changes to the forced image mode. New values from data points configured as an event are no longer added to the send buffer but rather they overwrite older existing values in the image memory.
- When the connection to the communications partner returns, the TIM changes back to the send buffer mode if the fill level of the send buffer has fallen below 50%.

Saving the data point values

As a rule, the values of data points are stored in the image memory of the module and transferred only when queried by the communications partner.

Events are also stored in the send buffer and can be transferred unsolicited.

Data points are configured as a static value or as an event using the "Type of transmission" parameter (see below):

- **Static value (no event)**
Static values are entered in the image memory (process image).
Static values correspond to the following type of transmission "Transfer after call (class 0)".
- **Event**
The values of data points configured as an event (triggered type of transmission) are also entered in the image memory of the module. The values are also entered in the send buffer.

Types of transmission and event classes

The following types of transmission are possible:

- **Transfer after call (class 0)**
The current value of the data point is entered in the image memory. New values of a data point overwrite the last stored value in the image memory.
After being called by the communications partner, the current value at this time is transferred.
- **Triggered**
Data points are configured as an event using a triggered type of transmission. The values of these data points are entered in the image memory and also in the send buffer.
The values of an event are saved as soon as the configured trigger conditions are met.
The following event classes are available:
 - **Every value triggered**
Each value change is entered in the send buffer in chronological order.
 - **Current value triggered**
Only the last, current value is entered in the send buffer. It overwrites the value stored there previously.

For information on the different trigger types, see section "Trigger" tab (Page 167).

2.30.6 Read cycle

You can control the cyclic reading and writing of PLC or DB tags of the CPU by the communications module by limiting the number of tags to be read and written. You make this setting in the "Communication with the CPU" parameter group.

You can prioritize the cyclic reading of values of input data points from their assigned CPU tags. Less important input data points do not need to be read in every CPU scan cycle. On the other hand, you can prioritize the updating of important input data points. You make this setting for each individual data point in the data point configuration in the "General" tab with the "Read cycle" parameter.

Options of the "Read cycle" parameter

- Normal cycle
In the normal cycle, a proportion of the values of the tags is read in each scan cycle.
- Fast cycle
The fast cycle is suitable for data to be acquired quickly, e.g. alarms. The value is read in each cycle.

For S7-1200/1500/ET 200SP, additionally:

- Spontaneous
The data point is registered for monitoring on the CPU and is no longer requested cyclically by the communications module.
As soon as the value in the CPU changes, it is actively transferred by the CPU to the communications module.
This ensures timely transfer of all changed values while limiting the load on the backplane bus.
- No monitoring
The data point is neither monitored nor requested. The value requested from a master station is always equal to zero.
The option can be useful for outputs that will not be monitored and do not return mirrored-back values.

Structure of the CPU scan cycle

The cycle with which the CP scans the memory areas of the CPU is made up of the following phases:

- **Read jobs**
 - **High-priority read jobs (Fast cycle)**

For all data points with the assignment "Fast cycle", the CPU tags are read in each scan cycle.
The number of high priority read jobs is not limited.
As a rule, it is sufficient to assign only data to be acquired quickly, such as alarms and contact changeover messages, to the fast cycle.
 - **Low-priority read jobs - proportion (Normal cycle)**

For data points with the assignment "Normal cycle", a proportion of the values of their tags is read in each scan cycle.
At most, as many tags are read in a scan cycle as is configured for "Max. number of read jobs" ("Communication with the CPU"). The values of the other low-priority tags that could not be read in a cycle are read in one of the subsequent cycles.
- **Write jobs**

The values of a configurable number of spontaneous write jobs are written to the CPU in each cycle. You specify the number of values written per cycle in the "Communication with the CPU" parameter group with the "Max. number of write jobs" parameter. The tags that could not be written in a cycle (tags in excess of the configured maximum number) are written in one of the subsequent cycles.
- **Cycle idle time**

This waiting time between two scan cycles is used to reserve adequate time for other processes that access the CPU via the backplane bus.

Since a fixed time cannot be configured for the cycle and since a fixed number of objects has not been assigned to the individual phases, the duration of the scan cycle is variable and can change dynamically.

2.30.7 "Local setpoint input" tab

Setpoints and their mirroring

Using the options described below, you can acquire the locally specified values from setpoint outputs and mirror back the current values to the partner upon change. A change to a local value can be caused by, for example, manual operator input on site.

The functions are available for the following modules:

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

You can configure the functions for the following data point types:

- Setpoint input (Set01W_S)
- Setpoint output (Set01W_R)
- Parameter input (Par12D_S)
- Parameter output (Par12D_R)

The corresponding input and output data points must be created for both communication partners for the functions. For the mirror function, move the local value in the controller to the relevant PLC tag of the data point.

The parameters for local setpoint input mode and mirroring are optional. If you do not require this, leave it disabled.

Parameter

The list shows the functions of the data point types as well as the tabs and parameter groups in which the functions are configured.

- **Setpoint/parameter input**
"General" > "Data source" tab:
 - Send setpoint to partner
The setpoint is stored in the PLC tag."Local setpoint input" > "Receive" tab:
 - Receipt of the 'Local input' setpoint input mode sent by the partner object
The input mode is transferred to the PLC tag "Setpoint input mode" (Bool).
 - Reception of the return value from the partner
The value mirrored by the partner is transferred to the PLC tag "Mirrored value".
- **Setpoint/parameter output**
"General" > "Data source" tab:
 - Setpoint received from partner
The setpoint is transferred to the PLC tag."Local setpoint input" > "Send" tab:
 - Send the 'Local input' local setpoint input mode to the partner object
The information as to whether the locally assigned setpoint or the setpoint received from the partner is valid is read from the PLC tag "Setpoint input mode".
 - 0: The received setpoint is valid.
 - 1: The locally entered setpoint is valid.
 - Send the locally entered setpoint to the partner
The setpoint is read from the PLC tag "Local setpoint".

Local setpoint input - Reception

The parameter group is active for sending data types (*_S).

The following parameters are available for local setpoints of the partner:

- **Setpoint input mode**

Via the parameter, the feedback is received from the partner data point about its setpoint input mode.

The information is stored in the PLC tag (Bool).

- 0 = Data point of the partner in 'Setpoint receipt' mode.
- 1 = Data point of the partner in 'local' setpoint input mode.

The parameter is used only for signaling. This parameter and the corresponding parameter of the partner data point do not cause an interlock for the partner with the local setpoint specification. Interlocking must be implemented by the user program.

After startup of the CPU or partner CPU, or after the return of the connection, a general request ensures that the current, valid status of the partner is displayed at the parameter.

- **Return value**

The partner data point that receives the setpoint feeds back the currently valid setpoint if the parameter "Setpoint status" is enabled there. This mirrored back setpoint is displayed here. If the partner object is set to 'local' and if new input is made there, the setpoint changed there is displayed here if the Local setpoint parameter is enabled for the partner object.

After startup of the CPU or partner CPU, or after the return of the connection, a general request ensures that the current, valid setpoint of the partner is displayed at the parameter.

Local setpoint input - Sending

The parameter group is enabled for receiving data types (*_R).

The following parameters are available for local setpoints of the own station:

- **Setpoint input mode**

This input can be used to signal whether the locally specified setpoint or the setpoint sent by the partner is valid.

The information is stored in the PLC tag (Bool).

- 0 = The setpoint of the partner is valid.
- 1 = The locally entered setpoint is valid.

A setpoint sent by the partner (e.g. the master station) can also be accepted by the object if the parameter is set (True).

The parameter is used only for information. An interlock with the remote setpoint assignment must be implemented in the user program.

- **Local setpoint**

If this parameter is enabled, the locally entered setpoint is read from the PLC tag and mirrored back to the send data point of the partner.

2.30.8 "Trigger" tab

Trigger

Data points are configured as a static value or as an event using the "Type of transmission" parameter:

Saving the value of a data point configured as an event

Saving the value of a data point configured as an event in the send buffer (message memory) can be triggered by various trigger types:

- **Threshold value trigger**
The value of the data point is saved when this reaches a certain threshold. The threshold is calculated as the difference compared with the last stored value, refer to the section Threshold value trigger (Page 170).
- **Time trigger**
The value of the data point is saved at configurable intervals or at a specific time of day.
- **Event trigger (Trigger tag)**
The value of the data point is saved when a configurable trigger signal is fired. For the trigger signal, the edge change (0 → 1) of a trigger tag is evaluated that is set by the user program. When necessary, a separate trigger tag can be configured for each data point.
Resetting the trigger tag in the bit memory area / DB:
If the memory area of a trigger tag is in the bit memory or in a data block, the module resets the trigger tag itself to 0 (zero) as soon as the value of the data point has been transferred. This can take up to 500 milliseconds.

Note

Fast setting of triggers

Triggers must not be set faster than a minimum interval of 500 milliseconds. This also applies to hardware triggers (input area).

Note

Hardware trigger

You need to reset hardware triggers via the user program

Transferring the value of a data point configured as an event

You specify whether the value of a data point is transferred to the communications partner immediately after the trigger occurs or after a delay in the "Transmission mode" parameter; see below.

Time stamp trigger

Validity: TIM 1531 IRC V2.4, CP 1200 V3.5

The trigger initiates sending of the data point value when the time stamp is changed in the CPU. The value of the data point is transferred together with the time stamp of the CPU.

The time stamp of the CPU is to be interpreted as local time or UTC, depending on the time settings. The time stamp of the CPU is forwarded to the connected partners by the communications module.

Valid for the following data point types:

- Bin04B_S / Bin08X_S
- Ana04W_S / Ana04R_S
- Mean04W_S
- Cnt01D_S / Cnt04D_S
- Dat12D_S / Dat12x1D_S

With multi-channel objects, only the time stamp of the first object is transferred.

Requirement for the data points:

- The data point is triggered (event).
- A separate UDT is created for each data point that is to use these functions.

To use these two triggers, you must create a data point whose tag references the UDT specified below.

Structure of the UDT for the "Value" and "Time stamp trigger" tags

The values of "Value" and "Time stamp trigger" are written together to one user data type (UDT).

The UDT has the following structure:

- "Value" tag
The data type must correspond to the data point type whose value will be transferred.
The supported data point types are listed above.
- "Time" tag
Data type: DTL

Creating the UDT

You must first create a user data type (UDT) for the two tags. Proceed as follows:

1. Navigate to the "PLC data types" entry of the assigned CPU in the project tree.
2. Click the "Add new data type" entry.
3. Optional: Change the name of the UDT
(select new user data type > "Properties" shortcut menu > "General" tab)
4. Then create the needed tags in the UDT.
The data type of the "Value" tag must correspond to the data type of the data point to be transferred in each case.

ValueTime						
	Name	Data type	Default value	Accessible..	Writable...	Visible...
	Value	Real	0.0			
	Time	DTL	DTL#1970-01-...			

Figure 2-19 Structure of the UDT for a measured value with data type "Real"

5. Open a data block or create a new one.
6. Insert the previously created UDT into the data block.
The UDT can be selected under the name that was assigned automatically or that you changed.

Note**Transfer data of the UDT together**

Make sure to transfer the data of the UDT together in one OB1 cycle to guarantee consistency of the data.

Transmission mode

The transmission mode of a frame is set in the "Trigger" tab of the data point. With the option, you specify whether messages of events are sent immediately or following a delay:

- Spontaneous (unsolicited - direct transfer)
The value is transferred immediately.
- Conditional spontaneous (buffered transfer)
The value is transferred only when one of the following conditions is fulfilled:
 - The communications partner queries the station.
 - The value of another event with the transmission mode "Spontaneous" is transferred.
 - The fill level of the send buffer has reached 90% of its maximum capacity.

Enable archiving

Validity: TIM 1531 IRC

With the TIM 1531 IRC you can save the values of triggered data points on the SD card of the CPU.

This option enables retentive storage of the values of events (send buffer) on the SD card in case of connection faults.

Requirement for enabling the function:

- Enabling of the parameter "Enable retentive saving" ("Basic settings" parameter group)

Archiving is possible for the following data point types:

- Bin04B_S / Bin08X_S
- Ana04W_S / Ana04R_S
- Mean04W_S
- Cnt01D_S / Cnt04D_S
- Dat12D_S / Dat12x1D_S
- Cmd01B_S / Cmd08X_S
- Set01W_S
- Par12D_S / Par12x1D_S

2.30.9 Threshold value trigger

Note**Threshold value trigger: Calculation only after "Analog value preprocessing"**

Note that the analog value preprocessing is performed before the check for a configured threshold value and before calculating the threshold value.

Smoothing factors and any configured integration interval are taken into account in the calculation.

This affects the value that is configured for the threshold value trigger.

Note**Threshold value trigger with configured mean value generation**

With enabled mean value generation, the absolute method for the calculation of the threshold value deviation is used for analog values with the threshold value trigger.

For the time sequence of the analog value preprocessing refer to the section Analog value preprocessing (Page 172).

Threshold value trigger

Function

If the process value deviates by the amount of the threshold value, the process value is saved.

Two methods are used to calculate the threshold value deviation:

- **Absolute method**

With binary and counter values as well as with analog values with configured mean value generation, the absolute method is used to calculate the threshold value deviation.

- **Integrative method**

With analog values without configured mean value generation, the integrating method is used to calculate the threshold value deviation.

In the integrating threshold value calculation, it is not the absolute value of the deviation of the process value from the last stored value that is evaluated, but rather the integrated deviation.

The application of the integrative method to analog values can be switched off via the "Integration interval" (Analog value preprocessing).

Absolute method

There is a check for each binary value to determine whether the current (possibly smoothed) value is outside the threshold value band. The current threshold value band results from the last saved value and the amount of the configured threshold value:

- Upper limit of the threshold value band: Last saved value + threshold value
- Lower limit of the threshold value band: Last saved value - threshold value

As soon as the process value reaches the upper or lower limit of the threshold value band, the value is saved. The newly saved value serves as the basis for calculating the new threshold value band.

Integrative method

The integration threshold value calculation works with a cyclic comparison of the integrated current value with the last stored value. The calculation cycle in which the two values are compared is configurable; see "Integration interval" in section Analog value preprocessing (Page 172).

(Note: The calculation cycle must not be confused with the scan cycle of the CPU memory areas).

The deviations of the current process value are totaled in each calculation cycle. The trigger is set only when the totaled value reaches the configured value of the threshold value trigger and a new process value is entered in the send buffer.

The method is explained based on the following example in which a threshold value of 2.0 is configured.

Table 2-6 Example of the integration calculation of a threshold value configured with 2.0

Time [s] (calculation cycle)	Process value stored in the send buffer	Current process value	Absolute devia- tion from the stor- ed value	Integrated devia- tion
0	20.0	20.0	0	0
0.5		20.3	+0.3	0.3
1.0		19.8	-0.2	0.1
1.5		20.2	+0.2	0.3
2.0		20.5	+0.5	0.8
2.5		20.3	+0.3	1.1
3.0		20.4	+0.4	1.5
3.5	20.5	20.5	+0.5	2.0
4.0		20.4	-0.1	-0.1
4.5		20.1	-0.4	-0.5
5.0		19.9	-0.6	-1.1
5.5		20.1	-0.4	-1.5
6.0	19.9	19.9	-0.6	-2.1

With the change in the process value shown in the example, the threshold value trigger configured with 2.0 is initiated twice:

- At the time 3.5 s: The value of the integrated deviation is at 2.0. The new process value stored in the send buffer is 20.5.
- At the time 6.0 s: The value of the integrated deviation is at 2.1. The new process value stored in the send buffer is 19.9.

In this example, if a deviation of the process value of approximately 0.5 should fire the trigger, then with the behavior of the process value shown here a threshold value of approximately 1.5 ... 2.5 would need to be configured.

2.30.10 Analog value preprocessing

For analog value data points, some or all of the functions described below can be configured.

Requirements and restrictions

You will find the requirements for the configuration of the preprocessing options and restrictions in the section relating to the particular function.

Note

Restrictions due to configured triggers

The analog value preprocessing options "Error suppression time", "Limit value calculation" and "smoothing" are not performed if no threshold value trigger is configured for the relevant data point.. In these cases, the read process value of the data point is entered in the image memory and transferred transparently before the preprocessing cycle of the threshold value calculation (500 ms) elapses.

Sequence of the analog value preprocessing options

The values of analog inputs configured as an event are processed according to the following scheme:

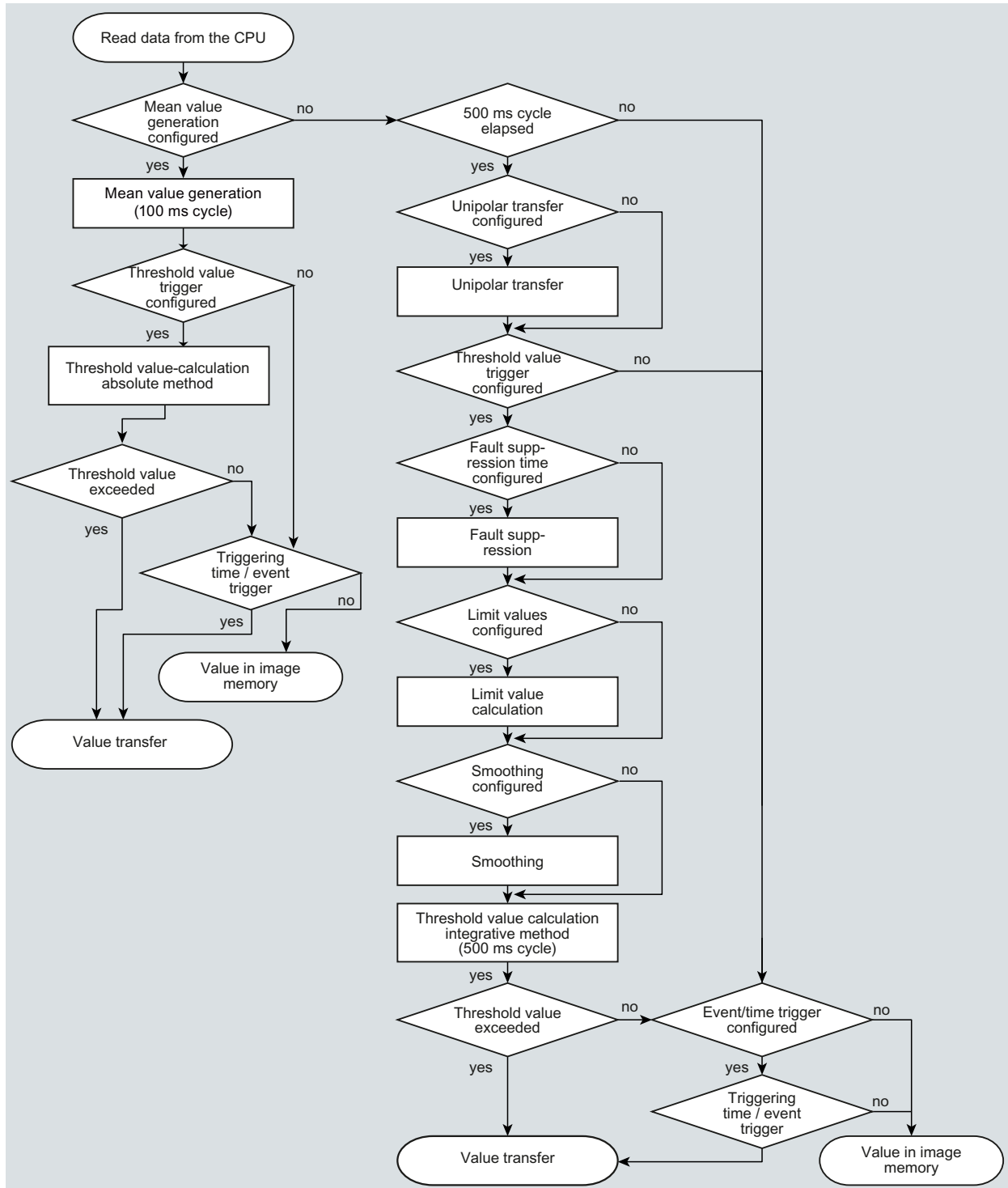


Figure 2-20 Sequence of the analog value preprocessing

The 500 millisecond cycle is started by the integrative threshold value calculation. In this cycle, the values are saved even when the following preprocessing options are enabled:

- Unipolar transfer
- Fault suppression time
- Limit value calculation
- Smoothing

Mean value generation

With this parameter, acquired analog values are transferred as mean values.

For the following protocols, mean value generation is only supported for integers of type "Int":

- TeleControl Basic
- DNP3
- IEC 60870-5

Note

Restricted preprocessing options if mean value generation is configured

If you configure mean value generation for an analog value event, the following preprocessing options are not available:

- Unipolar transfer
 - Fault suppression time
 - Smoothing
 - Threshold value calculation only with the absolute method
-

Function

If mean value generation is active, it makes sense to configure a time trigger..

The current values of an analog data point are read in a 100 millisecond cycle and totaled. The number of read values per time unit depends on the read cycle of the CPU and the CPU sampling cycle of the CP.

The mean value is calculated from the accumulated values as soon as the transfer is triggered by a trigger. Following this, the accumulation starts again so that the next mean value can be calculated.

The mean value can also be calculated if the transmission of the analog value message is triggered by a request from the communications partner. The duration of the mean value calculation period is then the time from the last transmission (for example triggered by the trigger) to the time of the request. Once again, the accumulation restarts so that the next mean value can be calculated.

Input modules: Overflow range / underflow range

As soon as a value is acquired in the overflow or underflow range, mean value generation is stopped. The value 32767 / 7FFF_h or -32768 / 8000_h is saved as an invalid mean value for the current mean value calculation period and sent with the next message.

The calculation of a new mean value is then started. If the analog value remains in the overflow or underflow range, one of the two values named is again saved as an invalid mean value and sent when the next message is triggered.

Note**Fault suppression time > 0 configured**

If you have configured an error suppression time and then enable mean value generation, the value of the error suppression time is grayed out but no longer used. If mean value generation is enabled, the error suppression time is set to 0 (zero) internally.

Unipolar transfer**Restrictions**

Unipolar transfer cannot be configured at the same time as mean value generation. Enabling unipolar transfer has no effect when mean value generation is activated.

Function

With unipolar transfer, negative values are corrected to zero. This can be desirable if values from the underrange should not be transferred as real measured values.

Exception: With process data from input modules, the value -32768 / 8000_h for wire break of a live zero input is transferred.

With a software input, on the other hand, all values lower than zero are corrected to zero.

Fault suppression time**Requirements for the function**

Configuration of the threshold trigger for this data point

Restrictions

The fault suppression time cannot be configured at the same time as mean value generation. A configured value has no effect when mean value generation is activated.

Function

A typical use case for this parameter is the suppression of peak current values when starting up powerful motors that would otherwise be signaled to the control center as a disruption.

The transmission of an analog value in the overflow (7FFF_h) or underflow range (8000_h) is suppressed for the specified time. The value 7FFF_h or 8000_h is only sent after the fault suppression time has elapsed, if it is still pending.

If the value returns to the measuring range before the fault suppression time elapses, the current value is transferred.

Input modules

The suppression is adjusted to analog values that are acquired directly by the S7 analog input modules as raw values. These modules return the specified values for the overflow or underflow range for all input ranges (also for live zero inputs).

An analog value in the overflow range (32767 / 7FFF_h) or underflow range (-32768 / 8000_h) is not transferred for the duration of the fault suppression time. This also applies to live zero inputs. The value in the overflow/underflow range is only sent after the fault suppression time has elapsed, if it is still pending.

Recommendation for finished values that were preprocessed by the CPU:

If the CPU makes preprocessed finished values available in bit memory or in a data block, suppression is only possible or useful if these finished values also adopt the values listed above 32767 / 7FFF_h or -32768 / 8000_h in the overflow or underflow range. If this is not the case, the parameter should not be configured for preprocessed values.

For finished values preprocess in the CPU, the limits for the overflow and underflow can be freely assigned.

Integration interval

The integration interval is used for threshold value processing of analog values according to the integration principle. The threshold value is used for the threshold value trigger; see also section Threshold value trigger (Page 170).

The entered value determines the time interval in which analog values are integrated.

With the setting 0 (zero), the threshold value is calculated without integration. This means lower data volume. In this case, the absolute method is used.

Smoothing factor**Requirements for the function**

Configuration of the threshold trigger for this data point

Restrictions

The smoothing factor cannot be configured at the same time as mean value generation. A configured value has no effect when mean value generation is activated.

Function

Analog values that fluctuate quickly can be evened out using the smoothing function.

The smoothing factors are calculated according to the following formula as with S7 analog input modules.

$$y_n = \frac{x_n + (k - 1) y_{n-1}}{k}$$

where

y_n = smoothed value in the current cycle n

y_{n-1} = smoothed value in the previous cycle n-1

x_n = value acquired in the current cycle n
k = smoothing factor

The following values can be configured for the module as the smoothing factor.

- 1 = No smoothing
- 4 = Weak smoothing
- 32 = Medium smoothing
- 64 = Strong smoothing

Set limit value 'low' / Set limit value 'high'

Requirements for the function

- Configuration of the threshold trigger for this data point
- Supported variable types of the CPU
The analog value data point must alternatively be linked to one of the following variables:
 - PLC tag in the bit memory address area
 - DB variable (variable in data block)

Limit value configuration is not possible for PLC tags that access hardware modules (input/output operand area).

The configuration of limit values is pointless for measured values that have already been preprocessed on the CPU.

Function

In these two input boxes, you can set a limit value in the direction of the start of the measuring range or in the direction of the end of the measuring range.

You can also evaluate the limit values, for example as the start or end of the measuring range.

Recommendation for quickly fluctuating analog values:

If the analog value fluctuates quickly, it may be useful to smooth the analog value first if limit values are configured.

Status identifier "OVER_RANGE" / "OV overflow"

With protocols that support status identifiers, if the high or low limit value is violated, the status identifier of the data point is set for measuring range violation.

Configuration of the limit value

Depending on the data type, a limit value is configured as an integer decimal number or as a floating-point number.

Table 2-7 Value ranges of the limit values

Data type	Range of values
Int	-32768 ... 32767
DInt	-2147483648 ... 2147483647
Real/LReal	-7.9000E+028..7.9000E+028

Note

Limit value 0 (zero)

- The entry of the value 0 is interpreted as a deactivated limit value for most modules.
Exceptions:
- The value 0 can also be used as a limit value for the following modules as of the indicated firmware version:
 - CP 1243-7 LTE V3.3
 - CP 1542-1 IRC V2.2
 - TIM 1531 IRC V2.2

The following table specifies the ranges of a 32-bit number with regard to the ranges of the raw value of an analog input or output module.

Range	Value of the 16-bit PLC variable *		Module output [mA]			Measuring range [%]
	Decimal	Hexadecimal	0 .. 20 (unipolar)	-20 .. +20 (bipolar)	4 .. 20 (life zero)	
Overflow	32767	7FFF	> 23.515	> 23.515	> 22.810	> 117.593
Overrange	32511	7EFF	23.515	23.515	22.810	117.593
	... 27649	... 6C01	... 20.001	... 20.001	... 20.001	... 100.004
Nominal range (unipolar / life zero)	27648	6C00	20		20	100
	...	...	...		...	...
	0	0000	0		4	0
Nominal range (bipolar)	27648 ...	6C00 ...		20 ...		100 ...
	0	0000		0		0
	... -27648	... 9400		... -20		... -100
Underrange (unipolar / life zero)	-1	FFFF	-0.001		3.999	-0.004
	...	...	...		...	...
	-4864	ED00	-3.518		1.185	-17.59
Underrange (bipolar)	-27649	93FF		-20.001		-100.004
	...	...		...		...
	-32512	8100		-23.516		-117.593
Undershoot / wire break	-32768	8000	< -3.518		< 1.185	< -17.593

* The value ranges (underflow / overflow) in PLC variables with different data types are as follows:

- Int
 - -32768
 - 32767
- DInt
 - -2147483648
 - 2147483647
- Real/LReal
 - -7.9000E+028
 - 7.9000E+028

Note**Evaluation of the value even when the option is disabled**

If you enable one or both options and configure a value and then disable the option later, the grayed out value is nevertheless evaluated.

To disable the two options, delete the previously configured values limit values from the input boxes and then disable the relevant option.

2.30.11 Command options

TIM 3V-IE / TIM 4R-IE

Parameters of the objects "Cmd01B_R/S" and "Cmd08X_R/S":

- Command output time

Output time for the command outputs of the object. Once the time has expired, the command output is reset automatically.

If the output time = zero, a set command output is not reset automatically. In this case, you must set the command output to zero via the user program of the CPU.

Range of values: 0...60 (tenths of a second)

The entered value is multiplied by the factor 0.1. Example: The input "5" means 0.5 seconds.

TIM 1531 IRC / CP**Output options**

For data points in stations that output commands from a master station, the following output options can alternatively be activated:

- **LATCH_ON/OFF**

The function permanently latches a command output to the value 0 or 1.

Note:

The latched value is only canceled by a new command. Alternatively, the command can be reset by the user program.

- **PULSE_ON**

The function evaluates the "Command output time" for a single pulse (see below).

The output options can be configured for the following data point types:

- Command output (Cmd01B_R)

Name: **Control Code**

Range of values:

- PULSE_ON
- LATCH_ON/OFF

Explanation: Output option of the command output. See above for the meaning.

Parameters for "PULSE_ON"

Name: **Max. number of pulses**

Range of values: Fixed assignment with "1"

Default: 1

Explanation: Monitors the number of pulses sent by the master station. If the number of pulses received from the master station exceeds the value configured here, the command is rejected.

Name: **Pulse control**

Range of values: NUL

Explanation: The data point transmits the pulse to the corresponding data point of the station.

Name: **Command output time**

Range of values: 0...65535 (seconds)

Default: 5
Explanation: Output time for the command outputs of the object. Once the time has expired, the command output is reset automatically.
If the output time = zero, a set command output is not reset automatically. In this case, you must set the command output to zero via the user program of the CPU.

Value mirroring

Parameters of the objects "Par12D_S" and "Par12x1D_S"

- **Mirroring of local value**
If the option is activated on the sending master station object, the value mirrored back by the partner object (Par12..._R) of the station is received. You write the value mirrored back by the station to the "PLC tag for local value" in the sending master station object.
With an assigned partner object (Par12..._R) of the station, the "Value monitoring" parameter must be enabled under "General".

Value mirroring

This function is used to mirror the local value to the communications partner of the data point.

The option can be used to mirror back the local value to the master station in the event of a change, for example due to manual operation on location.

The local value is written to the "PLC tag for local value" and mirrored back to the communications partner of the data point.

At the corresponding data point of the partner (e.g. Par12x1D_R), the "Value monitoring" option must be enabled in the "General" tab. You write the value mirrored back by the station to the "PLC tag for local value" in the sending master station object.

Valid for the following master station and station data points:

- Setpoint input (Set01W_S)
- Parameter input (Par12D_S / Par12x1D_S)

Name: **Mirroring of local value**
Range of values: 0 / 1
Explanation: Enables mirroring of the local value to the communications partner.

Name: **PLC tag for local value**
Data type: Same data type as the respective data point
Explanation: The local value must be mapped to this tag.

2.30.12 Masks

Masks at input data points

Parameter valid for TIM 3V-IE/4R-IE:

- All three masks can be configured for the "Digital input" (Bin04B_S) data point type.
- Only the disable mask can be configured for the "Command input" (Cmd01B_S) data point type.

The masks offer three options for transmitting binary value frames. You can specify bit-by-bit whether certain bits do not trigger frame transmission or which bits trigger a different transmission than that specified in the "Type of transmission" parameter.

The bits are masked in hexadecimal format. All masks have the default 00_H.

- Alarm mask
Masked bits in the byte cause an unconditional spontaneous frame transmission upon change. Changes from 0 to 1 and from 1 to 0 are evaluated.
The alarm mask only makes sense if the object is transmitted via a dial-up network and "Conditionally spontaneous" transmission mode has been configured under "Trigger".
- Send buffer principle mask
In a change occurs, masked bits in the byte cause transmission according to the transmit buffer principle. Changes from 0 to 1 and from 1 to 0 are evaluated.
- Disable mask
Masked bits in the byte are ignored with change control. This causes changes in the masked bits to not trigger a transfer. A masked bit always has the value 0 in the message.

Input of the masks

Masking the bits within the byte:

- Bits with 1 are masked.
- Bits with 0 are not masked.

Mask the bits as in the following example, in which the hexadecimal value A3 is entered. In the example, bits no. 0, 1, 5 and 7 are masked.

Byte assignment								
Bit	.7	.6	.5	.4	.3	.2	.1	.0
Masking	1	0	1	0	0	0	1	1
Entry	A				3			

2.30.13 Reset

Reset

Parameter valid for TIM 3V-IE/4R-IE

- Reset PLC tag
Resets the counter output via the edge change (0→1) reset a PLC tag (Bool) to zero.
When the closest counter value is received, the difference to the previous value is added up to zero.

2.30.14 Partner stations

Activating the partners of the data point

All partners with which a telecontrol connection has been configured are shown in the table.

Enable the partner or partners with which the selected data point is to exchange data using the check box: ☒

2.31 Messages

Configuring messages

The TIM can send messages to a communications partner on important events. No program blocks are required for the following messages.

The following are configurable:

- E-mails
The recipient can be a PC with an Internet connection or an S7 station.
- SMS (only mobile wireless CPs)
The recipient can be a mobile phone or an S7 station.

You perform the configuration in STEP 7 in the editor for the data point and message configuration. You can find this using the project tree:

Project > Directory of the relevant module

For the view in STEP 7, refer to the section Data point configuration (Page 147).

Requirements and necessary information

Remember the following requirements in the configuration for the transfer of messages:

- Enabling telecontrol communication ("Communication types") parameter group
- SMS: Configuration of the SMSC in the "SMSC" parameter group

Configuring e-mails

Required information:

- Access data of the SMTP server: Address, port number, user name, password
- When using STARTTLS or SSL/TLS: Certificate of the e-mail service provider
- E-mail addresses of the recipients

You perform the configuration in the following parameter groups:

- Enabling security functions (CPs)
To use e-mails, you need to enable the security functions of the CP, parameter group "Security".
- Configuration of the service / protocol:
"E-mail configuration", see section E-mail configuration (Page 115).
- When using STARTTLS or SSL/TLS:
 - Import of the certificate of the e-mail service provider:
"Global security settings"
 - Using the imported certificate for the communications module:
"Certificate manager"

Configuring SMS messages

Required information:

- Number of the SMSC

You perform the configuration in the following parameter groups:

- Enabling the SMS function:
"Communication types" > "Enable SMS"
- Configuration of the SMSC
 - CP: "Mobile wireless communication settings", see above.
 - TIM 1531 IRC: With the TIM 1531 IRC, configuration is not necessary.
 - TIM 3V-IE / TIM 4R-IE: "SMSC" parameter group
- Configuring the SMS
Message editor, see above.

"Message parameter"

CP, TIM 1531 IRC

Here you configure the phone number (SMS) or the recipient, the subject (e-mail) and the text of the message.

TIM 3V-IE / TIM 4R-IE

Configure the following parameters with the TIM 3V-IE / TIM 4R-IE:

- The PLC tag in which the telephone number of the recipient is stored.
- Text of the message
- Number of the configured SMSC
- Priority in the sampling cycle
 - Important process data may be preferred for sending the data stored in the transmit buffer.
 - High priority
 - If this option is selected, the SMS is given higher priority and is preferred over the other saved data.
 - 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 disabled for this object.
 - Low priority
 - If this option is selected, the SMS is sent with normal priority.
- SMS object number
 - Number of the SINAUT ST7 object from the data point configuration of the TIM

"Trigger"

In the "Trigger" parameter group you configure triggering for sending the message and other parameters.

CP, TIM 1531 IRC

- **E-mail trigger / SMS trigger**

Specifies the event for which the sending of the message is triggered.

Depending on the module used and the enabled services, you can select the following triggers:

- **Use PLC tag**

For the trigger signal to send the e-mail, the edge change (0 → 1) of the trigger bit "PLC tag for trigger" that is set by the user program is evaluated. When necessary, a separate trigger bit can be configured for each message. For information on the trigger bit, see below.

Resetting the trigger bit:

If the memory area of the trigger bit is in the memory area or in a data block, the trigger bit is reset to zero when the message is sent.

In all other cases, you need to reset the trigger bit with the user program.

Note

Fast setting of the diagnostics trigger tag

Trigger should not be set more often than once per second.

Frequent sending of SMS

Depending on the system environment, sending an SMS can take up to 2 minutes. To guarantee secure transmission of SMS in mobile wireless modules, a minimum interval of 10 seconds is recommended for triggering SMS.

- **CPU changes to STOP**

- **CPU changes to RUN**

- **Connection to a partner interrupted**

Triggers the sending of the message when the telecontrol connection to a partner is interrupted.

- **Connection to a partner established**

Triggers the sending of the message when the connection returns.

- **Connection establishment to partner failed**

Triggers the sending of the message when the connection to the telecontrol server could not be established.

- **Weak mobile wireless network**

(SMS only)

If the mobile wireless connection for telecontrol communication is too weak, an SMS message is triggered and sent to the configured recipient.

- **VPN connection established**

(CP 1243-8 IRC, CP 1542SP-1 IRC)

Triggers the sending of the message when the VPN connection is established or returns.

- **VPN connection terminated**

(CP 1243-8 IRC, CP 1542SP-1 IRC)

Triggers the sending of the message when the VPN connection is interrupted.

- **SINEMA RC connection established**
(CP 1243-8 IRC, CP 1542SP-1 IRC)
Triggers the sending of the message when the VPN or OpenVPN connection is established or returns.
- **SINEMA RC connection terminated**
(CP 1243-8 IRC, CP 1542SP-1 IRC)
Triggers the sending of the message when the VPN or OpenVPN connection is interrupted.
- **PLC tag for trigger**
PLC tag for the trigger "Use PLC tag"
- **Enable identifier for processing status**
If the option is enabled, every attempt to send returns a status with information about the processing status of the sent message.
The status is written to the "PLC tag for processing status". If there are problems delivering messages, you can determine the status via the Web server of the CPU by displaying the value of the PLC tag there.
For the significance of the status output in hexadecimal, refer to the section Processing status of the messages (SMS / e-mail) (Page 393).
- **PLC tag for processing status**
PLC tag of the type DWORD for the processing status
- **Include value**
When the option is enabled, the module sends a value from the memory area of the CPU for the placeholder in the message. To do this, enter "\$\$" or "\$DATA\$" as a placeholder for the value to be sent in the message text, depending on the module type. See below.
Select a variable whose value will be integrated in the message. The value is entered in the message text instead of the placeholder.
The placeholder for the value of a variable supports the following data types:
 - TIM 1531 IRC, CP 1200, CP 1542SP-1 IRC
 - Bool, Byte, Char, USInt, Int, UInt, Word, DWord, UInt, DInt, Real, String
 - Arrays of these data typesInput of the placeholder in the message text for these modules: \$\$
 - TIM 3V-IE, TIM 3V-IE Advanced, TIM 4R-IE
 - Only "Array[2..16] of Char" from tag tableInput of the placeholder in the message text for these modules: \$DATA\$

Note that the number of characters increases with the value. For the maximum number of characters, refer to the configuration limits of the module.
- **PLC tag for value**
PLC tag in which the value to be sent is written.

TIM 3V-IE / TIM 4R-IE

- **SMS trigger**

Specifies the event for which the sending of the message is triggered:

- CPU changes to STOP
- Changing the CPU operating state
- Use PLC tag
For the trigger signal to send the message, the edge change (0 → 1) of the trigger bit "PLC tag for trigger" that is set by the user program is evaluated. When necessary, a separate trigger bit can be configured for each e-mail. For information on the trigger bit, see below.
- Event with time trigger
Triggers the sending of the message when the time condition configured below is met.
- Include value change
Triggers the sending of the message if the value of the PLC tag configured under "Include value" changes.

- **PLC tag for trigger**

PLC tag for the message trigger "Use PLC tag"

Resetting the trigger bit:

If the memory area of the trigger bit is in the memory area or in a data block, the trigger bit is reset to zero when the message is sent.

In all other cases, you need to reset the trigger bit with the user program.

Note

Fast setting of the diagnostics trigger tag

Trigger should not be set more often than once per second.

- **Enable identifier for processing status**

If the option is enabled, every attempt to send returns a status with information about the processing status of the sent message.

The status is written to the "PLC tag for processing status". If there are problems delivering messages, you can determine the status via the Web server (diagnostics status) or read it out from the PLC tag.

For the significance of the status output in hexadecimal, refer to the section Processing status of the messages (SMS / e-mail) (Page 393).

- **PLC tag for processing status**

PLC tag of the type DWORD for the processing status

- **Include value**

See above, section "CP, TIM 1531 IRC".

- **PLC tag for value**

PLC tag in which the value to be sent is written.

2.32 Character set for user names, passwords and messages

Character set for user names, passwords and message texts

The following permitted characters apply to:

- E-mail server:
 - User name and password
- Messages in the message editor:
 - Message texts

Entered as ASCII character sets (hexadecimal value and character name):

- 0x20
Space
- 0x21 ... 0x5F
!"#\$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN OPQRST
UVWXYZ[\]^_
- 0x61 ... 0x7E
abcdefghijklmnopqrstuvwxyz{|}~
- 0x7C, 0x7E
|~

In addition for message texts:

- Manual line break (↵)
In message texts, you can insert a line break using <Shift>+<Enter>.

Block library TeleControl ST7

3.1 Version information and validity

Version

The following description is valid for the "Telecontrol ST7" block library:

- Version: V3.0 Service Pack 4

Validity

The block library can be used in CPUs of the following series:

- SIMATIC S7-1500
- SIMATIC S7-300
An Ethernet interface of the TIM 1531 IRC must be connected directly to the Ethernet interface of the CPU 300. The connection cannot run via a CP.

New in this version

Troubleshooting in S7-300 communication

Requirements

Requirements for use of the block library version specified above are:

- TIM 1531 IRC as of firmware version V2.4
- STEP 7 Professional (TIA Portal)
 - V17 Update 7
 - V18 Update 3
 - V19 or higher

3.2 Creating a library

Delivery form of the block library

The program blocks of the TD7onCPU library are used as global library for STEP 7 in the TIA Portal.

The library is archived and compressed on the Internet at the following address:

Link: (<https://support.industry.siemens.com/cs/ww/en/view/109755374>)

The file name has the following structure: Telecontrol_ST7_<Version>.exe

3.2 Creating a library

Creating the library

To use the library in STEP 7, follow these steps:

1. Create a suitable directory in the STEP 7 installation directory of your engineering station, for example, one named "Telecontrol ST7".
2. Save the library in this directory.
3. Run the "*.exe" file.
The self-extracting library is stored as a zip archive in a selectable directory.
4. Unzip the file.
The library is now available as an archived global library under the name "Telecontrol ST7.zal<version>" together with the readme files.
5. Open the STEP 7 project.
6. In the "Options" menu, select the command "Global libraries > Retrieve library".
The "Retrieve archived global library" dialog opens.
7. Select the folder and the library archive.
Recommendation: Leave the default "Open read-only" option activated to download the global library read-only.
8. Click Open.
The "Select target directory" window opens.
9. Select the target directory (your project) to which the archived global library is to be unpacked (select folder).
The library is unpacked.
If you are working with a newer STEP 7 version, you will be asked whether you want to upgrade the library. Confirm this using the "Upgrade" button.

The library is opened after unpacking.

You can find the library in the STEP 7 project in the library view (task card) in the "Global libraries" pane under the name "Telecontrol ST7_<Version>".

3.3 Structure of the library

Structure of the global library

The opened library contains the following directories:

- **Types**
 - PLC data types (UDT)
Here you will find the PLC data types (UDTs) used by the libraries.
 - System data types (SDT)
Here you will find the system data types used by the libraries.
- **Master copies**
You can find the following program blocks here:
 - **CPU300/400**
Program blocks for the S7-300/400 CPU
 - **CPU1500**
Program blocks for the S7-1500 CPU

Master copies

The two libraries under "Master copies" for S7-300/400 and S7-1500 are divided into the following folders:

- Data point typicals
Data point typicals (blocks) for transferring data
- Optional blocks
Optional blocks that you can use when necessary.
- System blocks
Blocks that are used for basic communication.
Depending on the SIMATIC product series used, you can find the following auxiliary blocks for communication between TIM and CPU in separate folders:
 - BCom
Blocks for processing the communication mailbox of the S7-400 and S7-1500
Blocks for processing the communication mailbox of the S7-300 (from TD7onCPU V3.0 SP4 with TIM 1531 IRC V2.4)
 - PCom
Blocks for processing the communication mailbox of an S7-300 with P bus (CPU without partyline)
 - XCom
Blocks for processing the communication mailbox of an S7-300 with X connections

For information on the "party line", see Glossary (Page 431).

3.4 Using, calling and generating the blocks

Generating TD7onCPU blocks

Once you have opened the TD7 library in the pane "Global libraries" under the name "Telecontrol ST7", you can generate the blocks for basic communication for one or more CPUs.

Requirements

The following requirements must be met before generating the blocks:

- The communications modules must be networked with each other.
- Valid telecontrol connections must be created between the communication partners involved.
- A TIM 1531 IRC must be networked with a CPU via Ethernet.
- A TIM 1531 IRC cannot be assigned to a CPU.
The "Assigned CPU" field in the "Subscriber numbers" parameter group must be empty.
- No data point must be created in the data point editor of a TIM.
- No message must be created in the message editor of a TIM.
- The "Telecontrol ST7" library must be open (visible) in the "Global libraries" task card.

Different requirements for modules with imported configuration data

Note the other requirements for communications modules whose SINAUT configuration was imported from a STEP 7 V5 project. With these modules, the option "Telecontrol configuration" is set to "Import" in the parameter group "Basic settings > Configuration".

These modules have different requirements for generating the blocks:

- No ST7 telecontrol connections need to be created between the communication partners involved.
The relevant information is stored in the imported configuration data.
- The CPU of the communications module must have the same IP address as in the STEP 7 V5 project. The CPU is assigned to the communications module via its IP address.

Procedure for generating

Generate the blocks for basic communication as follows:

1. Select the desired TIM.
2. Select the shortcut menu command (right mouse button) "Generate TD7onCPU blocks".
3. In the dialog that follows, select one or more CPUs and click "Generate..."

The TD7onCPU blocks for the SINAUT basic communication are created and compiled in the selected CPUs under the program blocks in a new directory "TD7onCPU".

Using blocks

- **System blocks**

The required blocks for basic communication are created automatically during the generation in the CPU.

You do not need to use or call these blocks manually in the CPU.

The system blocks handle central tasks such as startup, monitoring of the connections, reachability of the connection partners, entering data in the send mailbox or taking data from the receive mailbox of the communication DBs, general request, time keeping, handling communication etc.

Communication DBs:

- S7-300/400

For each connection between the TIM and the CPU, a separate communication DB is created automatically: It contains the send and receive mailbox and all the data required for controlling and monitoring this connection.

- S7-1500

For the CPU 1500, there is only one communication DB "BConnectData". A connection instance is stored in this DB in the "BConnection" array for each local connection with the TIM.

- **Data point typicals**

You need to use data point typicals on the relevant CPU according to the communications tasks.

Drag the individual blocks with the mouse from the global library to the user program of the CPU. The corresponding instance data block is created automatically when using a data point typical.

The number of the instance data block is also the source object number, which you need to specify at the corresponding partner object of the communication partner under "PartnerObjectNo". For this reason, each typical needs to use its own instance data block. You can also assign the instance DB number manually.

- **PLC data types (UDT)**

- Dat256D_S / Dat256D_R

Copy also the UDT "TransmitBlock" for these typicals from the library (Types > PLC data types (UDT)) into the "PLC data types" directory of the CPU. The UDT is not automatically referenced by the typical.

- **Optional blocks**

You should also use the optional blocks on the CPU depending on the communications tasks of the station.

Drag the individual blocks with the mouse from the global library to the user program of the CPU.

Call in the user program

- **System blocks**

Only the two following FCs need to be called in the user program:

- StartUp

Call the FC in the startup OB. The block has no parameters.

To be able to call the FC StartUp, you must first create the startup OB (OB100).

- BasicTask

Call the FC in the cyclic OB or user program.

The DB BasicData is created automatically. The DB contains all the centrally required data including the records of all communication partners and the connections to be managed. For information on the various "BasicTask" FCs, see section Note on the "BasicTask_*" FCs (Page 196).

- **Data point typicals**

Depending on the communication tasks, call up all required data point typicals in the user program.

- **Optional blocks**

Depending on the communication tasks, call up all required optional blocks in the user program.

See also section The cyclic SINAUT program (Page 197) for information on calling the blocks.

Note

Error display for library conformance

In the "Compilation" parameter group of blocks, an error can be displayed under the entry "Library conformance" (call of single instances). This is caused by the call of global data blocks, for example, DB BasicData.

Since the "Telecontrol ST7" library has no "know-how protection", you can ignore these error notifications.

3.5 Note on the "BasicTask_*" FCs

Versions of the "BasicTask" FC

There are different versions of the "BasicTask" FC in the TD7 block library "Telecontrol ST7" for STEP 7 Professional.

Depending on the SIMATIC product series and the backplane bus, the FCs have a different suffix at the end of the block name:

- **BasicTask_B**

For S7-1500 CPU

- **BasicTask_B**

For S7-400 CPU

(prepared for S7-300)

- **BasicTask_X**
For S7-300 PU with partyline
- **BasicTask_P**
For S7-300 CPU without partyline

For information on "Partyline", see Glossary.

For the block description, see section BasicTask_* FC (Page 347).

"BasicTask" label for three FCs

References to the respective FC are made in many places in the following sections. Since the description of the different blocks usually applies to several SIMATIC product series, the three FCs are referred to hereafter as "BasicTask". The FC version for the relevant CPU type is meant in each context.

3.6 Structure of user program for TD7onCPU

3.6.1 The cyclic SINAUT program

Introduction

The structure of the TD7 blocks in the user program (OB1) is explained below.

The TD7 blocks must be proceeded in every OB1 cycle. Keep to the call sequence of the blocks unless instructed otherwise.

Other user-specific parts of the program can be linked in before or after the TD7 blocks in OB1 or, if practical, within the TD7 blocks.

You can structure the SINAUT-specific part in OB1 by calling it in lower-level FCs.

All data point typicals are FBs. An instance DB must be specified when an FB is called. The number of this instance DB is identical to the object number of the data point object.

The structure of the TD7 blocks in OB1

Cyclic OB1	
BasicTask	The respective FC BasicTask must always be called at the start. It handles basic SINAUT tasks that are always required.
Optional SINAUT basic functions	After FC BasicTask, call the following optional blocks (FCs) if you need them: <ul style="list-style-type: none"> • ListGenerator Creation of address lists for received frames with incomplete destination address. • TimeTask Provides the time of day.
	If necessary, call additional optional FCs in the subsequent program execution, for example: <ul style="list-style-type: none"> • Trigger Scheduled start of user programs and data frames • PartnerStatus Display the status of partners • PartnerMonitor Extended subscriber-specific display and control options. • PulseCounter Count pulse acquisition • PathStatus Display the status of the connection path to a partner
	Note: The following two blocks may not be called explicitly, since they are activated via an internal interface. <ul style="list-style-type: none"> • TestCopy • TestcopyDB
Data point typicals	In the subsequent program execution, call the data point typicals for sending and receiving data. The sequence of the individual typicals is unimportant. The number of typicals to call and the required types depend on the amount and type of data to be sent and received. Note: The instance DBs of these FBs are referenced using the partner object number of the communication partner.

Example of a TD7 program for a station

Cyclic OB1	
BasicTask	<p>The respective FC BasicTask must always be called at the start of the cyclic program. Generally, data is ready for further processing after it has been received.</p> <p>The FC has only one parameter, "UserFC". Normally 0 can be specified. If you require user-specific processing for received data, specify here the number of an FC containing the program for this processing.</p> <p>You can also copy received data chronologically to an archive memory via this interface.</p>
TimeTask	<p>As an option, you can call the FC TimeTask immediately after FC BasicTask. The FC has no parameters. FC TimeTask must be included if you need the time of day. This allows data frames to be time-stamped.</p> <p>However, you can also use the time of day to start program components at a specific point in time or to schedule the transmission of data frames. FC Trigger, described below, is then required.</p> <p>For this FC Time Task to be used, the CPU must be provided with the time from a local TIM module. You specify this in the configuration.</p>
Trigger	<p>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.</p> <p>The FC can be inserted several times if several times or various time intervals are required. Using the FC requires that FC TimeTask must be called first in the OB1 program (see above) and the CPU time must have already been set once.</p>
PartnerStatus	FC PartnerStatus can be included as an option. The FC shows the reachability for a maximum of 8 communications partners.
ListGenerator	<ul style="list-style-type: none"> ListGenerator300 for S7-300 CPU ListGenerator400 for S7-400 CPU ListGenerator1500 for S7-1500 CPU <p>The FC can be installed as an option. The FC is required if the station receives frames containing no destination address or an incomplete destination address. This happens if the configuration of the destination address has been completely or partially omitted in a partner for data point typicals. ("PartnerNo" and "PartnerObjectNo" were not specified, so transmission to all known destination subscribers takes place.)</p>
PulseCounter PathStatus PartnerMonitor	When necessary

Cyclic OB1	
Data point typical	<p>Following the FCs for SINAUT basic tasks, data point typicals for sending and receiving data are called. The sequence of the individual typicals is unimportant. The number of typicals to call and the required types depend on the amount and type of data to be sent and received.</p> <p>For one station applies:</p> <ul style="list-style-type: none"> The following are typically sent: <ul style="list-style-type: none"> Binary information, such as status messages and alarms Analog values Counted values Additional data if necessary The following are typically received: <ul style="list-style-type: none"> Commands Setpoints, parameters
Bin04B_S	One or more FBs for the acquisition and transfer of binary information, such as messages, alarms, etc.
Ana04W_S	One or more FBs for the acquisition and transfer of analog values
Cnt01D_S / Cnt04D_S	<p>One or more FBs for the acquisition and transfer of counted values</p> <p>A requirement for the use of the FBs is that FC PulseCounter is included in a cyclic interrupt OB, e.g. OB35. This FC is responsible for the time-driven acquisition of counted pulses in the background.</p>
Cmd01B_R	One or more FBs for receipt and output of commands.
Set01W_R / Par12D_R	One or more FBs for receipt and output of setpoints or parameters
Dat12D_S / Dat256D_S	<p>One or more FBs for the acquisition and transmission of 1 to 12 or up to 256 data double words with any informational content</p> <p>There is no data-specific processing and change control for these typicals. The user program is responsible for this. As an option change control can be activated that triggers a transfer with each bit change.</p>

TD7 programs for master stations and node stations

In principle, the TD7 programs of master stations and node stations look the same as for a station. The corresponding receive typicals are used in a central station for the send typicals of a station, the corresponding send typicals are used in a central station for the receive typicals of the station.

In a node station, both send typicals and the corresponding receive typicals are used according to the transmission direction.

In the master station it is practical to structure the OB1 program according to stations, in other words, all data typicals belonging to the same station are called in one FC. The best overview is provided when the number of the FC is identical to the subscriber number of the station.

3.6.2 Cyclic interrupt OB

Introduction

Only include time-controlled TD7 blocks in a CPU if fast count pulses must be detected in this CPU which could not be reliably detected within an OB1 cycle due to an excessively long cycle time.

The count pulses are acquired via any digital input module. To acquire the pulses reliably, the digital inputs use must be queried for change at a fixed time interval. The time interval is based on the duration of the shortest count pulse. The minimum count pulse duration may be 50 ms. The same applies to the duration of the pause. This results in a maximum count frequency of 10 Hz.

The time interval in which the count pulse acquisition is performed must be approximately half the count pulse duration, i.e. at 50 ms at an interval of approximately 25 ms.

For this time-controlled count pulse acquisition, OB35 needs to be configured for an S7-300 CPU, one of the available cyclic interrupt OBs OB30 to OB38 for an S7-400 CPU, and a cyclic interrupt OB with a number from 30..38 for an S7-1500 CPU.

All cyclic interrupt OBs have a preset time interval, for OB35 for example 100 ms. This can be changed in increments of 1 ms. This makes it possible to set a cyclic interrupt OB, for example, to 25 ms.

You change the time interval for a cyclic interrupt OB in the Properties dialog (S7-400/1500), or in the Properties dialog of the CPU with S7-300.

In highly time-critical applications, count pulses can also be recorded in a process alarm OB of an alarm module, e.g. in OB40.

Table 3-1 Calling FC PulseCounter in the alarm OB

OB	
PulseCounter	<p>One or more FC PulseCounters can be integrated in an alarm OB for the acquisition of count pulses if several count pulses can occur during an OB1 cycle and are to be acquired.</p> <p>FC PulseCounter processes up to 8 pulse inputs of any digital input. The acquired count pulses are added together in programmable SIMATIC counters. These access the function blocks that put together the count value frames (FB-Cnt0x_S).</p>

3.6.3 Programming error OB

Validity

Only for S7-300/400

Function

When a block that does not exist is called in an S7-300/400 CPU, the CPU normally changes to STOP. The diagnostics buffer indicates which FB, FC or DB was missing. You can then reload the missing block and restart the CPU.

3.7 Types

If, however, you do not want the CPU to change to STOP if there is a missing block or only changes to STOP when certain block types or block numbers are missing, you can specify the reaction you require in OB121.

Even if you have loaded OB121 as an empty block on the CPU, this is enough to have the CPU continue running if a block is missing. If you want to decide more selectively when the CPU should continue running or change to STOP, include OB121 in the user program.

In conjunction with SINAUT ST7 it is possible that a CPU changes to STOP when it receives data from another CPU that it does not (or not yet) know. This is, for example, the case when you add a data point typical to a station and give it a complete destination address (destination subscriber no. plus destination object no.). The specified destination object no. can lead to a stop on the destination subscriber in the following situation:

- As soon as you have installed a new data point typical in a station, the data is transferred to the destination.
- If there is no corresponding receive typical installed on the destination CPU, the destination object no. (= instance DB of the received typical) does not exist.

The result is that the CPU changes to STOP as soon as the data is received.

To avoid this it is advisable to call FC ST7ObjectTest in OB121.

Table 3-2 Call of FC ST7ObjectTest in the programming error OB

OB121	
ST7ObjectTest	Calling FC ST7ObjectTest in OB121 prevents a CPU STOP, if the CPU receives data with an unknown destination object no. Other calls can be included at any point in OB121 regardless of the FC ST7ObjectTest call. FC ST7ObjectTest has a parameter "StopInOtherCases". Here you can specify what happens in other situations (STOP or continue running) if OB121 was called because another data block or an FB or FC is missing.

3.7 Types

3.7.1 PLC data types (UDT)

UDTs used

The following PLC data type (UDT) is used by the block library:

- TransmitBlock
UDT for the data transfer between the CPU and TIM
When using the "Dat256D_S / Dat256D_R" typical, copy the UDT into the "PLC data types" directory of the respective CPU. The UDT is not automatically referenced by the typical.
The UDT does not require any parameter assignment.

3.7.2 System data types (SDT)

SDTs used

The following system data types (SDTs) are used by the DB BasicData and stored automatically in the directory "PLC data types" of the CPU when generating/compiling the TD7 blocks:

- **ConnectionDescription**
SDT is part of the "BasicData" DB and has an "Array[1..1]" of "ConnectionDescription" data type. It contains the following details for the connection description:
 - Number of the configuration DB
 - Connection type
 - CFB number in the case of an S7 connection
 - Subscriber number of the local TIM.
- **SubscriberObject**
SDT is part of the "BasicData" DB and has an "Array[1..1]" of "SubscriberObject" data type. It contains the following information about the communication partner:
 - Subscriber number of the partner
 - Array index of the connection description
 - Subscriber type
 - Information whether the partner is local or remote.

3.8 Master copies

Validity

The following program blocks are located in the master copies of the global library:

- **CPU300 / CPU400**
Program blocks for the S7-300 and S7-400 CPU
- **CPU1500**
Program blocks for the S7-1500 CPU

The blocks are largely identical. Unless otherwise stated, the description applies equally to the blocks of the S7-300/400 and S7-1500.

If individual blocks cannot be used for all SIMATIC product series mentioned or if functions for the SIMATIC product series differ, this is expressly indicated.

3.8.1 Data point typicals

3.8.1.1 Types and overview of the data point typicals

Types of the data point typicals

Data point typicals process one or more data points of the same information type, e.g. 4 bytes of binary information, 4 analog values or 1 byte commands, etc.

When a data point typical (FB) is called the corresponding instance DB needs to be specified in which the data will be written or from which the data to be transferred will be read.

Transfer direction of the typicals

A data point typical for a specific type or amount of information always come in two versions:

- A typical for acquiring and sending
- A typical for receiving and outputting

When using data point typicals a distinction is therefore made according to the transfer direction:

- **Sending typicals**

Sending typicals process data and send it to the remote partner.

They have the ending "_S" Examples: Bin04B_S, Ana04W_S

- **Receiving typicals**

Receiving typicals receive the data from their remote partner.

They have the ending "_R" Examples: Bin04B_R, Ana04W_R

When transferring data on the two communicating partners there is always a corresponding pair of typicals involved.

Examples:

- Monitoring direction
 - The station sends binary data with the typical Bin04B_S.
 - The central station receives the data with the typical Bin04B_R.
- Control direction
 - The central station sends a command with the typical Cmd01B_S.
 - The station receives the command with the typical Cmd01B_R.

Structure of typical names

Note

No SMS blocks

Unlike the block library for STEP 7 V5, there are no blocks for sending SMS in the global library "Telecontrol ST7".

You configure the sending of e-mails and/or SMS in STEP 7 Professional in the TIA Portal via the data point and message editor.

The names of the data point typicals are assigned according to the following scheme:

Table 3-3 Structure of typical names

Data point type	Number of values	Data format	Under-score	Function
Bin = Binary information	Maximum number of values sent per data frame	X = bit		S Send
Ana = Analog value		B = byte		R Receive
Cnt = Count value		W = word		FS Sending without 1-out-of-n check
Cmd = Command		D = double word		FR Receiving without 1-out-of-n check
Set = Setpoint, parameter		R = real		
Par = Parameter				
Dat = Data (any mixture of information types)				

Overview of the data point typicals

The following table provides an overview of the data point typicals.

Table 3-4 Overview of the data point typicals

Data format	Symbolic block name	Function
Binary value typicals		
Byte	Bin04B_S	Send 4 byte binary values
Byte	Bin04B_R	Receive 4 byte binary values
Bit	Bin08X_S	Send 8 binary values for specific bits
Bit	Bin08X_R	Receive 8 binary values for specific bits
Analog value typicals		
Int	Ana04W_S	Send 4 analog values
Int	Ana04W_R	Receive 4 analog values
Real	Ana04R_S	Send 4 analog values as floating point number (CPU 1500 only)
Real	Ana04R_R	Receive 4 analog values as floating point number (CPU 1500 only)
Counted value typicals		
UInt, Word *	Cnt01D_S	Send 1 counted value

3.8 Master copies

Data format	Symbolic block name	Function
UInt, Word *	Cnt01D_R	Receive 1 counted value
UInt, Word *	Cnt04D_S	Send 4 counted values
UInt, Word *	Cnt04D_R	Receive 4 counted values
Command typicals		
Byte, UInt *	Cmd01B_S	Send 1 byte commands (1-out-of-8)
Byte, UInt *	Cmd01B_FS	Send 1 byte commands without (1-out-of-n) control
Byte, UInt *	Cmd01B_R	Receive 1 byte commands (1-out-of-8)
Byte, UInt *	Cmd01B_FR	1 byte commands received without 1-out-of-8 control
Bit	Cmd08X_S	Send 8 commands for specific bits (only CPU 1500)
Bit	Cmd08X_R	Receive 8 commands for specific bits (only CPU 1500)
Setpoint and parameter typicals		
Word	Set01W_S	Send 1 setpoint and receive current local setpoint. Transmission is subject to 1-out-of-n control (via FC Safe).
Word	Set01W_FS	Send 1 setpoint (without 1-out-of-n control) and receive current local setpoint.
Word	Set01W_R	Receive 1 setpoint and send current local setpoint
ARRAY [1...12] of DInt / UInt / DWord / Real *	Par12D_S	Send max. 12 double words with parameters and receive current local parameters Transmission is subject to 1-out-of-n control (via FC Safe).
ARRAY [1...12] of DInt / UInt / DWord / Real *	Par12D_FS	Send max. 12 double words (without 1-out-of-n control) with parameters and receive current local parameters
ARRAY [1...12] of DInt / UInt / DWord / Real *	Par12D_R	Receive max. 12 double words with parameters and send current local parameters
Typicals for variable data types and amounts		
ARRAY [1...12] of DInt / UInt / DWord / Real *	Dat12D_S	Send maximum of 12 double words with any data
ARRAY [1...12] of DInt / UInt / DWord / Real *	Dat12D_R	Receive maximum of 12 double words with any data
Div. data types **	Dat12x1D_S	Send max. 12 channels with data (only CPU 1500)
Div. data types **	Dat12x1D_R	Receive max. 12 channels with data (only CPU 1500)
DWord	Dat256D_S	Send maximum of 256 double words with any data
DWord	Dat256D_R	Receive maximum of 256 double words with any data

* UInt, UInt and UInt are not supported by S7-300/400.

** For the supported data types, refer to the block description.

Parameters not required

Parameters not required in data point typicals can be left open.

3.8.1.2 Reoccurring parameters

Description of the typical parameters

In the following description data point typicals that are identical except for the number of data points to be processed are described together.

In the tables of the data point typicals you will find the following information about the individual parameters:

- Parameter
Name of the parameter
- Declaration
Parameter type
 - INPUT
Input parameter
 - OUTPUT
Output parameter
 - IN_OUT
In-out parameter
- Data type
Data type supported for this parameter
- Range of values
- Default
Preset value of the parameter
If you do not configure individual parameters of a data point typical, the default value is used.
- Explanation
Description of the function and the typical-specific properties of the parameter

Parameters used by many typicals

The following parameters are used by many of the typicals of the TD7onCPU library. These parameters are described once here and the description is not repeated in the following sections for the individual data point typicals.

Depending on the use of the typicals, some parameters are configured differently. Note the typical use described below.

PartnerNo

Declaration: INPUT
 Data type: INT
 Range of values: 0 / 1 ... 32000
 Default: 0

Explanation: Subscriber number of the partner

The subscriber number of the partner with which the block communicates must be specified..

- In a station typical, this is normally the subscriber number of the central station or the application of the control station (e.g. ST7cc).
- For a typical that is used in a central station, this is normally the subscriber number of a station.

Effects of the value 0 (zero) on various typical classes

- **Sending typicals (1)**

(Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S)

With the value 0 the data is sent to all subscribers to which an ST7 connection is configured. In this case, the parameter "PartnerObjectNo" is automatically sent with the value zero by the process typical.

If PartnerNo is not found in the administration (DB BasicData), an entry to this effect is made in the diagnostics buffer (event ID B101). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Notes

- If the "PartnerObjectNo" is missing, there must be a list on the partner CPU from which the missing object number can be recognized (see ListGenerator1500/300/400 FC (Page 319)).
- Use of the block in a node station
If the CPU of the node station maintains both connections to higher-level subscribers as well as to lower-level stations, a data frame with PartnerNo = 0 is transferred to all subscribers in the direction master station and in the direction stations.

- **Sending typicals (2)**

(Cmd01B_S, Set01W_S, Par12D_S)

The value 0 is not permitted!

If "PartnerNo" is < 1 or > 32000, an error message is entered in the diagnostic buffer (event ID B100).

If the configured value is permitted and correct but "PartnerNo" is not found in the administration (DB BasicData), an entry is also made in the diagnostics buffer (event ID B101). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

- **Receiving typicals (1)**

(Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Dat12D_R, Dat256D_R)

The value 0 is not permitted!

If PartnerNo is < 1 or > 32000, an error message is entered in the diagnostic buffer (event ID B100).

If the configured value is permitted and correct but "PartnerNo" is not found in the administration (DB BasicData), an entry is also made in the diagnostics buffer (event ID B101). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

If the CPU receives a data frame for this typical, a check is made to establish whether the source subscriber number in the data frame is identical to the "PartnerNo" configured here. If the two subscriber numbers are different, the received information is discarded and an error message is entered in the diagnostic buffer (Event ID B130).

- **Receiving typicals (2)**

(Cmd01B_R, Set01W_R, Par12D_R)

Set the value 0 if the typical is to receive data from more than one partner, e.g. if it is to receive data from several control centers.

If the CPU receives data for this typical and "PartnerNo" is > 0, a check is made to establish whether the source subscriber number in the data frame is identical to the "PartnerNo" configured here. If the two are different, the received information is discarded and an error message is entered in the diagnostic buffer (Event ID B130).

This check is not made if "PartnerNo" is = 0. Regardless of the sender, each data frame addressed to the typical is passed on to the typical.

If "PartnerNo" is > 0 and this number is not found in the administration (in DB-BasicData), an entry is made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Notes

- If "PartnerNo = 0" is set, make sure that each partner sends the data with a complete destination address (target subscriber no. and target object no.).
- Use of the block in a node station
If the CPU of the node station maintains both connections to higher-level subscribers as well as to lower-level stations, a data frame with PartnerNo = 0 is transferred to all subscribers in the direction master station and in the direction stations.

PartnerObjectNo

Declaration: INPUT

Data type: INT

Range of values: 0 / 1 ... 32000

Default: 0

Explanation: Object number of the partner

The number of the object (= DB number) on the partner with which the block communicates.

Effects of the value 0 (zero) on various typical classes

- **Sending typicals (1)**

(Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S)

Setting the parameter to 0 makes sense if PartnerNo = 0 was set for the preceding parameter. If the "PartnerObjectNo" is missing, there must be a list on the partner CPU from which the missing object number can be recognized (see FC-ListGenerator).

If the partner is an ST7cc control center specifying the "PartnerObjectNo" for this block can be omitted because in ST7cc there are no DBs as target objects. ST7cc decodes its data solely based on the source address in the data frame.

- **Sending typicals (2)**

(Cmd01B_S, Set01W_S, Par12D_S)

The value 0 is not permitted!

If the parameter assignment is incorrect (< 1 or > 32000), an error message is entered in the diagnostic buffer (event ID B102). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

- **Receiving typicals (1)**

(Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Dat12D_R, Dat256D_R)

The value 0 is not permitted!

If the parameter assignment is incorrect (< 1 or > 32000), an error message is entered in the diagnostic buffer (event ID B102). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

If the CPU receives data for the object configured here, there is a check to establish whether the source object number in the data frame is identical to the "PartnerObjectNo" configured here. If they are different, the received information is discarded. An error message is entered in the diagnostic buffer (event ID B131).

- **Receiving typicals (2)**

(Cmd01B_R, Set01W_R, Par12D_R)

0 must be set in the following situations:

- The partner is not an S7 CPU, i.e. there is no DB number as object. This is, for example, the case when the partner is an ST7cc control center.
- There is more than one partner (PartnerNo = 0) from which the typical is to receive data. The corresponding objects of these partners will then generally have different numbers; in other words, no unique number can be specified here.

If the CPU receives data for the object configured here and "PartnerObjectNo" is > 0, there is a check to establish whether the source object number in the data frame is identical to the "PartnerObjectNo" configured here. If they are different, the received information is discarded. An error message is entered in the diagnostic buffer (event ID B131).

This check is not made if "PartnerObjectNo" is = 0. Regardless of the sender object, each data frame addressed to the object is also passed on to the receiving object.

Enabled

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: TRUE

Address Input

I 0.0 ... I n.7

range: Memory

M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit

DBm.DBX 0.0 ... n.7

Explanation: Enable block processing

- If processing is enabled, all the functions of the block execute.
- The response is different if processing has not been enabled:

Processing not enabled

- **Sending typicals**

(Bin04B_S, Ana04W_S, Cnt01D_S, Cnt04D_S, Dat12D_S, Dat256D_S)

Also applies to the mirror-image value of: Set01W_R, Par12D_R

If processing is not enabled, the block can only communicate at the organizational level; in other words, Org frames can be sent and received.

Please note: A general query is answered, but the response frame contains the data valid at the time of the lock.

Note

The response described here does not apply to Cmd01B_R; see Command typical Cmd01B_R (Page 240).

- **Receiving typicals**

(Bin04B_R, Ana04W_R, Cnt01D_R, Cnt04D_R, Dat12D_R, Dat256D_R)

Also applies to the mirror-image value of: Set01W_S, Par12D_S

If processing is not enabled, the block can only communicate at the organizational level; in other words, Org frames can be sent and received.

Please note: A request can still be sent and the answer received, the received information is, however, not output to the outputs. You will find the relevant outputs in the description of the appropriate data point typicals.

Note

The response described here does not apply to Cmd01B_S; see Command typical Cmd01B_S (Page 235).

ImageMemory

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: TRUE

Address Input

I 0.0 ... I n.7

range: Memory

M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit

DBm.DBX 0.0 ... n.7

3.8 Master copies

Using in typicals Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S

Explanation: Image memory principle for spontaneous data transmission

- TRUE
The data is transferred according to the image memory principle.
The image memory principle reduces the memory required for storing data frames and produces as little data traffic on the WAN as possible. The default TRUE is the correct choice in most cases.
- FALSE
The data is transferred according to the send buffer principle.
The send buffer principle is only required with data points whose individual data changes will be saved and transferred to the partner, for example alarms with time stamps.

Conditional

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: TRUE

Address range: Memory M 0.0 ... M n.7
L 0.0 ... L n.7
Data bit DBm.DBX 0.0 ... n.7

Using in typicals Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S

Explanation: Conditional spontaneous data transfer

You will find information on this below with the parameter "Unconditional".

Unconditional

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address range: Memory M 0.0 ... M n.7
L 0.0 ... L n.7
Data bit DBm.DBX 0.0 ... n.7

Using in typicals Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S, Dat256D_S

Explanation: Unconditional spontaneous data transfer

With the two parameters "Conditional" and "Unconditional" you decide whether a data frame is transferred immediately by the module if the value changes (unconditional spontaneous) or at a later point in time (conditional spontaneous).

The two parameters are configured as follows:

- **Conditional spontaneous transfer (not necessarily immediately)**
 - Conditional = TRUE
 - Unconditional = FALSE
- **Unconditional spontaneous transfer (immediately)**
 - Conditional = FALSE
 - Unconditional = TRUE

The default of the two parameters was chosen so that a data frame is not transmitted immediately.

The decision whether a data frame is transferred immediately or later only relates to dial-up networks.

In a dial-up network you must decide from case to case whether a change to the value of the data point requires immediate transfer and therefore the immediate establishment of a connection. This can, for example, be required for data points with alarms.

On a dedicated line, the transmission is always immediate even if the combination of "Conditional" and "Unconditional" is set to "not immediately". On dedicated lines, you do not need to make changes to the two parameter settings.

Permanent

Note

Parameter irrelevant

The parameter (INPUT, Bool) only appears in blocks for S7-300/400.

It is no longer supported functionally with S7-300/400, the value is always = FALSE.

TimeStamp

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

3.8 Master copies

Using in typi- Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Dat256D_S, Set01W_R,
cals Par12D_R

Explanation: Time stamp

- TRUE
The data frame will be transferred with a time stamp.
The prerequisite is that the time provided by the local TIM is available in the CPU. For more detailed information, refer to the description of FC TimeTask.

Note

With Ana04W_S, remember the dependency of the time stamp on the parameter "MeanValueGeneration"; see Analog value typical Ana04W_S (Page 221).

- FALSE
The data frame is transferred without a time stamp.

For information on the format of the time stamp, refer to the section Time stamp (Page 215).

Note

Re-initialization

This parameter requires re-initialization of the CPU.

Recommendation:

Do not change the parameter in runtime or after the restart.

NewData

Declaration: OUTPUT

Data type: BOOL

DWORD with 1 status per segment for Dat256D_R and Dat12x1D_R

Range of val- TRUE / FALSE
ues:

Default: FALSE

Address Output

Q 0.0 ... Q n.7

range: Memory

M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit

DBm.DBX 0.0 ... n.7

Using in typicals Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Cmd01B_R, Set01W_S, Set01W_R, Par12D_S, Par12D_R, Dat12D_R, Dat12x1D_R, Dat256D_R

Explanation: Receive new data

The NewData output is intended for user-specific further processing, for example to react in a specific way to receipt of new data.

Whenever the block has received new data and has output it to the outputs for the specific typicals, the parameter is set to TRUE for one OB1 cycle.

You will find the specific outputs in the description of the individual data point typicals.

With the data point typicals Set01W_R and Par12D_R, "NewData" is also set to TRUE for one OB1 cycle if there is a new local value entered in the Local = 1 state.

If you do not require the parameter, simply leave it open.

3.8.1.3 Time stamp

Format of the SINAUT time stamp

For many data point typicals you can use the TimeStamp parameter to instruct that the data of the object should be transferred with a time stamp.

However for the receiving data point typicals there is no output parameter with which to output the received time stamp. The time stamp is only saved in the instance DB which you have specified when calling the respective receive typical. To further process the time stamp, the data must be read out of the DB by the user program.

The time stamp is saved in two data double words that have the same name in all object DBs:

Name of the double word	Contents
RecTimeStamp_1	Year, month, day and hour
RecTimeStamp_2	Minute, second, millisecond and time status

With the exception of the half byte with the time status, the date and time are coded in BCD format.

Table 3-5 Assignment of the structure of the time stamp

Name of the double word	Byte no.	Contents	
		High nibble	Low nibble
RecTimeStamp_1	0	Year * 10	Year * 1
	1	Month * 10	Month * 1
	2	Day * 10	Day * 1
	3	Hour * 10	Hour * 1
RecTimeStamp_2	0	Minute * 10	Minute * 1
	1	Second * 10	Second * 1
	2	Millisecond * 100	Millisecond * 10
	3	Millisecond * 1	Time status

3.8 Master copies

Table 3-6 The bit assignment of the half byte "time status" (low nibble of byte 3 of RecTimeStamp_2)

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

The time double words occupy different addresses depending on the typical. Look in the instance DB or in the declaration header of the FB to find the absolute address of both double words.

It is more convenient to give the instance DBs symbolic names. You can then use the symbolic addresses to read out the information. In this case, you do not need to worry about the actual absolute addresses. These are used automatically by STEP 7. The following example clarifies this procedure.

Example

Symbolic name of instance DB: ObjectDB27

The STEP 7 program for reading the date and time of day and for saving in DB20 beginning with data byte 100 may appear as follows programmed in STL:

```

L "ObjectDB27".RecTimeStamp_1
T DB 20.DBD 100
L "ObjectDB27".RecTimeStamp_2
T DB 20.DBD 104

```

3.8.1.4 Analog value typical Ana04R_S

Validity

S7-1500

Function

Send 4 analog values as 32-bit floating-point number

Ana04R_S transfers the 4 analog values as instantaneous values. At the time of the transfer, the currently pending analog value is acquired and transferred to the partner.

Note

Common processing of the four analog values

The processing parameters such as threshold, smoothing factor etc. exist only once in a typical. These parameters apply to all 4 analog values in common; in other words, it is not possible to set the parameters for the individual analog values. For this reason, each typical should only acquire analog values that should be processed in the same way.

Parameters

General parameters:

- PartnerNo**
- PartnerObjectNo**
- Enabled**
- ImageMemory**
- Conditional**
- Unconditional**
- TimeStamp**

For a description, see section Reoccurring parameters (Page 207).

Name: **ThresholdIntegration**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Threshold processing according to the integration principle

With this parameter, you can specify whether the integration principle is used in threshold value processing.

With the default FALSE the threshold value is calculated without integration. In this case, you can expect less data traffic on the telecontrol line and locally between CPU and TIM.

Name: **ZeroLimitation**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

3.8 Master copies

Default: TRUE
 Explanation: Zero limitation
 If this parameter is activated negative values are suppressed and replaced with the value 0.

Name: **TriggerInput**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address	Input	I 0.0 ... I n.7
range:	Memory	M 0.0 ... M n.7
		L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Trigger input

With the edge change 0 → 1 of the "TriggerInput" input the transfer of the analog value frame can be triggered at a required time.

Example:

Time-driven analog value transfer with time stamp for supplying an analog value archive in the control center.

Make sure that you set the "ImageMemory" parameter to FALSE to prevent these data frames with time stamps from being overwritten when saving on the station TIM.

The FC Trigger block can be used for time-driven triggering of a transmission over "TriggerInput".

If you do not require the parameter, simply leave it open. The transfer should then be triggered based on the "ThresholdValue" and "ThresholdIntegration" threshold parameters.

"TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. With dedicated lines or wireless networks the transfer is immediate. With a dial-up connection, it is possible that the data frame is saved first on the TIM and sent at a later point in time. The reason can, for example, be that the data frame is marked as "Conditional spontaneous", see parameter "Conditional".

Name: **AnalogInput_1 ... _4**

Declaration: INPUT

Data type: REAL

Range of values: See address range

Default: 0.0

Address	Bit memory address	MDO ... MDn
range:	Data address	LD0 ... LDn
		DBm.DBDO ... DBDn

Explanation: Analog input word

For each analog value to be transmitted in the data frame, you can specify from where the FB will take the analog information. Any combination of a data block and the bit memory address area can be used.

If you do not require parameters, simply leave them open. The value 0 is transferred for these analog inputs in the data frame.

Name: **SamplingPeriod**

Declaration: INPUT

Data type: INT

Range of values: 0 ... 32767 [ms]

Default: 500

Explanation: Acquisition interval for analog inputs in milliseconds

The acquisition interval is required for the following parameters:

- Threshold formation according to the integration principle (ThresholdIntegration)
- Smoothing the analog input value (SmoothingFactor)

The value must be selected high enough so that it is certain that a new value was acquired within the encryption time of the analog input. The interval to be specified has to be at least as long as the encoding time of the analog input module being used at the selected resolution (8 ... 15 bits).

The value must also be selected generously so that analog values are acquired even with the highest resolution and with analog modules with the highest number of inputs.

Specifying a "SamplingPeriod" that is too short may lead to an overflow of the internal accumulation counter.

Name: **ThresholdValue**

Declaration: INPUT

Data type: REAL

Range of values: +1.175495e-38 ... +3.402823e+38

3.8 Master copies

Default: 1.0

Explanation: Threshold value

Without configuration, the default value 1.0 is used.

Point to note with "ThresholdValue" = 0: Changes are not checked based on the threshold value. The analog value frame will only be sent in the following situations:

- When there is a trigger via the "TriggerInput" input, typically a time-driven or event-driven trigger.
- When there is a general request to the station or a single request for the data frame.

Name: **SmoothingFactor**

Declaration: INPUT

Data type: REAL

Range of values: 1.0 (no smoothing)
4.0 (weak smoothing)
32.0 (medium smoothing)
64.0 (strong smoothing)

Default: 1.0

Explanation: Smoothing factor

Using the smoothing factor, quickly fluctuating analog values can be smoothed to a greater or lesser extent depending on the factor. This may allow a narrower threshold band to be selected (ThresholdValue).

The smoothing factors are identical to the smoothing factors that are configured for some S7 analog input modules. The smoothing is calculated using the same formula as on the input module:

$$y_n = \frac{x_n + (k - 1) y_{n-1}}{k}$$

y_n = smoothed value in the current cycle n

y_{n-1} = smoothed value in previous cycle n-1

x_n = acquired value in the current cycle n

k = smoothing factor

3.8.1.5 Analog value typical Ana04R_R

Validity

S7-1500

Function

Receive 4 analog values as 32-bit floating-point number

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled

For a description, see section Reoccurring parameters (Page 207).

Name: **AnalogOutput_1 ... _4**

Declaration: OUTPUT

Data type REAL

Range of values: See address range

Default: 0.0

Address Bit memory address

MD0 ... MDn

range: Data address

LD0 ... LDn

DBm.DBDO ... DBDn

Explanation: You can select where the individual analog values received by the FB are output. Addresses of a data block and in the bit memory address area can be mixed as required.

If you do not require parameters, simply leave them open.

Name: **NewData**

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Typical-specific response: Whenever the FB has received new data and has output it to the outputs "AnalogOutput_1" to "AnalogOutput_4", the "NewData" output is set to TRUE for one OB1 cycle.

3.8.1.6 Analog value typical Ana04W_S

Function

Send 4 analog values as 16 bit values

Ana04W_S transfers the 4 analog values alternatively:

- As instantaneous values
At the time of the transfer, the currently pending analog value is acquired and transferred to the partner.
- As mean values
The pending analog value is accumulated at selectable intervals. At the time of the transmission, a mean value is formed from the total value and transferred to the partner.

Note**Common processing of the four analog values**

The processing parameters such as threshold, smoothing factor etc. exist only once in a typical. These parameters apply to all 4 analog values in common; in other words, it is not possible to set the parameters for the individual analog values. For this reason, each typical should only acquire analog values that should be processed in the same way.

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional

For a description, see section Reoccurring parameters (Page 207).

Name: **TimeStamp**

For the declaration, data type, range of values, default and function, refer to the section Re-occurring parameters (Page 207).

Typical-specific functions With the setting TRUE the time stamp depends on the setting of the parameter "MeanValueGeneration":

- **MeanValueGeneration = FALSE**
Instantaneous values are transferred in the data frame.
The time stamp in the data frame is identical to the time of acquisition of the instantaneous values contained in the data frame.
- **MeanValueGeneration = TRUE**
The data frame contains mean values.
The time stamp is identical to the time at which the mean value calculation period was completed.
The start of the mean value calculation period is not included in the data frame. This is, however, identical to the time stamp of the previously transferred mean value frame.

Name: **ThresholdIntegration**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Threshold processing according to the integration principle

With this parameter, you can specify whether the integration principle is used in threshold value processing.

With the default FALSE the threshold value is calculated without integration. In this case, you can expect less data traffic on the telecontrol line and locally between CPU and TIM.

When MeanValueGeneration = TRUE, (analog values are sent as mean values), the "ThresholdIntegration" parameter has no meaning.

Name: **ZeroLimitation**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: TRUE

Explanation: Zero limitation

If this parameter is activated negative values are suppressed and replaced with the value 0.

Name: **TriggerInput**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address	Input	I 0.0 ... I n.7
range:	Memory	M 0.0 ... M n.7
		L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Trigger input

With the edge change 0 → 1 of the "TriggerInput" input the transfer of the analog value frame can be triggered at a required time.

Example:

Time-driven analog value transfer with time stamp for supplying an analog value archive in the control center.

Make sure that you set the "ImageMemory" parameter to FALSE to prevent these data frames with time stamps from being overwritten when saving on the station TIM.

If the block calculates mean values, the duration of the calculation period is decided by the "TriggerInput" input. The current period is ended and a new period begun each time a transmission is triggered by this input. The interval between triggering a data frame twice determines the duration of the mean value calculation period.

The FC Trigger block can be used for time-driven triggering of a transmission over "TriggerInput".

If you do not require the parameter, simply leave it open. The transfer should then be triggered based on the "ThresholdValue" and "ThresholdIntegration" threshold parameters.

"TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values/mean values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. With dedicated lines or wireless networks the transfer is immediate. With a dial-up connection, it is possible that the data frame is saved first on the TIM and sent at a later point in time. The reason can, for example, be that the data frame is marked as "Conditional spontaneous", see parameter "Conditional".

Name: **MeanValueGeneration**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Mean value generation

If the parameter is enabled the analog values to be acquired are transferred as mean values.

If you select mean value generation, the currently pending analog value is acquired cyclically and accumulated. The acquisition cycle depends on the "SamplingPeriod" parameter (for example 500 ms, see also the description of this parameter). The mean value is calculated from the accumulated values as soon as a transmission is triggered via the "TriggerInput" input. Following this, the accumulation starts again so that the next mean value can be calculated.

The mean value can also be calculated if the transmission of the analog value frame is triggered by a general or single request. The duration of the mean value calculation period is then the time from the last transmission (for example triggered via TriggerInput) to the time of the general or single request. Once again, the accumulation restarts so that the next mean value can be calculated.

If the acquired analog value is above or below the permitted range (7FFFH bzw. 8000H), this value can either be taken into account immediately in the calculation of the mean value or it can be suppressed for a specific period for the calculation of the mean value. The required response can be decided with the "FaultSuppressionTime" parameter:

- **FaultSuppressionTime = 0**

Acquisition of a value above or below the over- or underrange results in immediate cancellation of the mean value calculation. The value 7FFF_H or 8000_H is saved as an invalid mean value for the current mean value calculation period and sent when the next analog value frame is triggered. The calculation of a new mean value is then started. If the analog value remains in the overshoot or undershoot range, this new value is again saved immediately as an invalid mean value and sent when the next frame is triggered.

- **FaultSuppressionTime > 0**

If the acquired analog value is in the overshoot or undershoot range, the bad values are excluded from the calculation of the mean value for a maximum duration as defined by the FaultSuppressionTime. If this time is exceeded, the value 7FFF_H or 8000_H is saved as an invalid mean value and sent when the next analog value frame is triggered. The procedure is identical in each new mean value calculation period averaging period; in other words, bad values are again suppressed for the duration of the "FaultSuppressionTime".

The duration of the "FaultSuppressionTime" also indirectly decides the proportion of invalid values per mean value calculation period. For example, if the mean value is calculated every 15 minutes and "FaultSuppressionTime" is set to 5 minutes, the mean value is only sent as invalid when more than 1/3 of the analog values acquired are above or below the overshoot or undershoot range in the current mean value calculation period.

Name: **AnalogInput_1 ... _4**

Declaration: INPUT

Data type: WORD

Range of values: See address range

Default: 0 (W#16#0)

Address	I/O words	PIW0 ... PIWn
range:	Memory words	MW0 ... MWn
	Data words	LW0 ... LWn
		DBm.DBW0 ... n

Explanation: Analog input word

For each analog value to be transmitted in the data frame, you can specify from where the FB will take the analog information. I/O words from analog input modules, data words from a data block and memory words can be mixed as required.

If you do not require parameters, simply leave them open. The value 0 is transferred for these analog inputs in the data frame.

Name: **SamplingPeriod**

Declaration: INPUT

Data type: INT

Range of values: 0 ... 32767 [ms]

Default: 500

Explanation: Acquisition interval for analog inputs in milliseconds

The acquisition interval is required for the following parameters:

- Threshold formation according to the integration principle (Threshold Integration)
- Smoothing of the analog input value (Smoothing Factor)
- Mean value generation

The value must be selected high enough so that it is certain that a new value was acquired within the encryption time of the analog input. The interval to be specified has to be at least as long as the encoding time of the analog input module being used at the selected resolution (8 ... 15 bits).

The value must also be selected generously so that analog values are acquired even with the highest resolution and with analog modules with the highest number of inputs.

If mean values are calculated, SamplingPeriod should not be less than 500 ms. If mean values are calculated over very long periods, the time must be increased as follows:

- Mean value calculation period 12 h: SamplingPeriod = 1000 [ms]
- Mean value calculation period 24 h: SamplingPeriod = 2000 [ms]

Specifying a "SamplingPeriod" that is too short may lead to an overflow of the internal accumulation counter. The maximum value of 2 147 483 647 of a double integer must not be exceeded. When an overflow is detected, the invalid mean value of 8000_H is transferred for the current mean value calculation period.

Name: **ThresholdValue**

Declaration: INPUT

Data type: INT

Range of values: 0 / 1 ... 32767

Default: 270

Explanation: Threshold value

When specifying the threshold value, take the encryption range of the analog values into account. Raw values from S7 analog inputs are always encoded in the range from 0 ... 27648 (= 0 ... 100 %) or + 27648 (= + 100%). Depending on the resolution of the analog input, the value jumps from 128 (with 8-bit resolution) or 1 (with 15-bit resolution). If the acquired analog values have a different encoding range, specify a threshold value oriented toward this.

If the parameter is not configured, the default value of 270 is used. This corresponds to approximately 1% of the normal S7 analog raw value range.

Point to note with "ThresholdValue" = 0:

Changes are not checked based on the threshold value. The analog value frame will only be sent in the following situations:

- When there is a trigger via the "TriggerInput" input, typically a time-driven or event-driven trigger.
- When there is a general request to the station or a single request for the data frame.
- When the analog value moves into the overshoot or undershoot range (7FFF_H or 8000_H) (possibly after the suppression time set for "FaultSuppressionTime" has elapsed).

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the "ThresholdValue" parameter has no meaning.

Parameter: SmoothingFactor

Declaration: INPUT

Data type: INT

Range of values: 1 (no smoothing)
4 (weak smoothing)
32 (medium smoothing)
64 (strong smoothing)

Default: 1

Explanation: Smoothing factor

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the "ThresholdValue" parameter has no meaning.

Using the smoothing factor, quickly fluctuating analog values can be smoothed to a greater or lesser extent depending on the factor. This may allow a narrower threshold band to be selected (ThresholdValue).

The smoothing factors are identical to the smoothing factors that are configured for some S7 analog input modules. The smoothing is calculated using the same formula as on the input module:

$$y_n = \frac{x_n + (k - 1) y_{n-1}}{k}$$

y_n = smoothed value in the current cycle n

y_{n-1} = smoothed value in previous cycle $n-1$

x_n = acquired value in the current cycle n

k = smoothing factor

Name: **FaultSuppressionTime**

Declaration: INPUT

Data type: INT

Range of values: 0 ... 32767

Default: 0

Explanation: Fault suppression time in seconds.

Transmission of an analog value located in the overshoot or undershoot range (7FFF_H or 8000_H) is suppressed for the time period specified here. The value 7FFF_H or 8000_H is only sent after this time has elapsed, if it is still pending. If the value returns to below 7FFF_H or above 8000_H again before this time elapses, it is immediately sent again as normal. The suppression time is started again for the full duration the next time 7FFF_H or 8000_H is acquired.

This is typically used for temporary suppression of current values that may occur when powerful pumps and motors are started. The analog input may exceed the maximum range several times under some circumstances. Suppression prevents these values from being signaled as faults in the control center system.

The suppression is adjusted to analog values that are acquired by the S7 analog input modules as raw values. These modules return the specified values for the overflow or underflow range for all input ranges (also for life-zero inputs). With provided ready values, fault suppression is only possible if these also adopt the values 7FFF_H or 8000_H when there is overshoot or undershoot. If this is not the case, the parameter does not need to have a value entered.

The parameter can also be used in combination with the mean value calculation for temporary suppression of the values 7FFF_H or 8000_H (see parameter MeanValueGeneration).

If no parameter is specified, the default of 0 seconds applies. An acquired value of 7FFF_H or 8000_H is then sent immediately when it is first detected or, with mean value calculation, as an invalid mean value for the current mean value calculation period.

3.8.1.7 Analog value typical Ana04W_R

Function

Receive 4 analog values as 16-bit values

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled
 For a description, see section Reoccurring parameters (Page 207).

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

Data words

LW0 ... LWn

DBm.DBW0 ... n

You can select where the individual analog values received by the FB are output. I/O words from analog output modules, data words from a data block and memory words can be mixed as required.

If you do not require parameters, simply leave them open.

Name: **NewData**

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Typical-specific response: Whenever the FB has received new data and has output it to the outputs "AnalogOutput_1" to "AnalogOutput_4", the "NewData" output is set to TRUE for one OB1 cycle.

3.8.1.8 Binary value typical Bin04B_S

Function

Send 4 bytes of binary information

Parameters

General parameters:

- PartnerNo**
- PartnerObjectNo**
- Enabled**
- ImageMemory**
- Conditional**
- Unconditional**
- TimeStamp**

For a description, see section Reoccurring parameters (Page 207).

Name: **InputByte_1 ... _4**

Declaration: INPUT

Data type: BYTE

Range of values: See address range

Default: 0 (B#16#0)

Address	Input bytes
---------	-------------

range: PIB0 ... PIBn

Memory bytes	MB0 ... MBn
--------------	-------------

LBO ... LBN

Data bytes DBm.DBB0 ... n

Explanation: Input byte

Specify the memory area (1 to 4 bytes) of the binary information to be transferred by the FB.

Input bytes from the process input image, I/O bytes directly from digital input modules, data bytes from a data block and memory bytes can be mixed.

If you do not require parameters, simply leave them open.

For unconfigured bytes, the value 0 (zero) is transferred.

Name: **DisableMask**

Declaration: INPUT

Data type: DWORD

Range of val- 0 ... 2147483647

- As 32 bit binary number
2#0 ... 2#11111111_11111111_11111111_11111111
- As 32 bit hexadecimal number
DW#16#0 ... DW#16#FFFF FFFF

Default: 0 (2#0)

Explanation: Disable mask

- For every input to be blocked enter a 1 at the relevant position in the bit pattern.
- For the other inputs enter a 0.

A disabled input always has the value 0 (zero) during the transfer.

For the assignment of the 32 inputs from "InputByte_1" to "InputByte_4" to the 32 bits of the blocking mask, see the following table.

	InputByte_1								InputByte_2								InputByte_3								InputByte_4							
Bit																																
2#	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		
DW#16#																																

Name: **InversionMask**

Declaration: INPUT

Data type: DWORD

Range of values: 0 ... 2147483647

- As 32 bit binary number
2#0 ... 2#11111111_11111111_11111111_11111111
- As 32 bit hexadecimal number
DW#16#0 ... DW#16#FFFF_FFFF

Default: 0 (2#0)

Explanation: Inversion mask

The inversion of input signals can, for example, be useful when using a mixture of sensors operating on the open and closed circuit principle.

- For every input to be inverted enter a 1 at the relevant position in the bit pattern.
- For the other inputs enter a 0.

For the assignment of the 32 inputs from "InputByte_1" to "InputByte_4" to the 32 bits of the inversion mask, see the following table.

	InputByte_1								InputByte_2								InputByte_3								InputByte_4							
Bit	$\dot{}$ 7	$\dot{}$ 6	$\dot{}$ 5	$\dot{}$ 4	$\dot{}$ 3	$\dot{}$ 2	$\dot{}$ 1	$\dot{}$ 0	$\dot{}$ 7	$\dot{}$ 6	$\dot{}$ 5	$\dot{}$ 4	$\dot{}$ 3	$\dot{}$ 2	$\dot{}$ 1	$\dot{}$ 0	$\dot{}$ 7	$\dot{}$ 6	$\dot{}$ 5	$\dot{}$ 4	$\dot{}$ 3	$\dot{}$ 2	$\dot{}$ 1	$\dot{}$ 0	$\dot{}$ 7	$\dot{}$ 6	$\dot{}$ 5	$\dot{}$ 4	$\dot{}$ 3	$\dot{}$ 2	$\dot{}$ 1	$\dot{}$ 0
2#	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DW#16#	—								—								—								—							

3.8.1.9 Binary value typical Bin04B_R

Function

Receive 4 bytes of binary information

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled
 For a description, see section Reoccurring parameters (Page 207).

Name: **OutputByte_1 ... _4**

Declaration: OUTPUT

Data type: BYTE

Range of values: See address range

Default: 0 (B#16#0)

Address range:	Output bytes	QB0 ... QBn PQB0 ... PQBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n

Explanation: Output byte

Specify the memory area (1 to 4 bytes) of the binary information to be output in the binary information.

Output bytes from the process image output, I/O bytes directly to digital output modules, data bytes from a data block and memory bytes can be mixed.

If you do not require parameters, simply leave them open.

For information on reading out the time stamp received with the data using the user program, see the section Time stamp (Page 215).

Name: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the output bytes "OutputByte_1" to "OutputByte_4"

3.8.1.10 Binary value typical Bin08X_S

Function

Send 8 binary values for specific bits

Parameters

General parameters:

- PartnerNo**
- PartnerObjectNo**
- Enabled**
- ImageMemory**
- Conditional**
- Unconditional**
- TimeStamp**

For a description, see section Reoccurring parameters (Page 207).

Name: **Input_1 ... _8**

Declaration: INPUT

Data type: BOOL

Range of values: See address range

Default: 0 (B#16#0)

Address	Input	I0.0 ... In.7
range:		PI0.0 ... Pln.7
	Bit memories	M0.0 ... Mn.7
		L0.0 ... Ln.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Input bit

The parameter exists eight times, once per channel.

Specify the memory area (1 to 8 bits) of the binary information to be transferred by the FB.

Input bits from the process image input, I/O bits directly from digital input modules, data bits from a data block and memory bits can be mixed.

If you do not require parameters, simply leave them open.

Name: **InversionMask**

Declaration: INPUT

Data type: BYTE

Range of values: 0 ... 255

- As 8-bit dual number
2#0 ... 2#11111111
- As 8-bit hexadecimal number
DW#16#0 ... DW#16#FF

Default: 0 (2#0)

Explanation: Inversion mask

The inversion of input signals can, for example, be useful when using a mixture of sensors operating on the open and closed circuit principle.

- For every input to be inverted enter a 1 at the relevant position in the bit pattern.
- For the other inputs enter a 0.

A disabled input always has the value 0 (zero) during the transfer.

One of the 8 bits of the inversion mask is assigned to each of the 8 inputs from "Input_1" to "Input_8".

3.8.1.11 Binary value typical Bin08X_R

Function

Receive 8 binary values for specific bits

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled

For a description, see section Reoccurring parameters (Page 207).

Name: **OutputBit_1 ... _8**

Declaration: OUTPUT

Data type: BOOL

Range of values: See address range

Default: 0 (B#16#0)

Address range:	Output	Q0.0 ... Qn.7 PQ0.0 ... PQBn
	Bit memories	M0.0 ... Mn.7 L0.0 ... Ln.7
	Data bits	DBm.DBX 0.0 ... n.7
	Explanation:	Output bit
		Specify the memory area (1 to 8 bits) of the binary information to which the binary information will be output.
		The parameter exists eight times, once per channel.
		Output bits from the process image output, I/O bits directly to digital output modules, data bits from a data block and memory bits can be mixed.
		If you do not require parameters, simply leave them open.
		For information on reading out the time stamp received with the data using the user program, see the section Time stamp (Page 215).
Name:	NewData	
Explanation:	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).	
	The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the output bytes "OutputBit_1" to "OutputBit_4".	

3.8.1.12 Command typical Cmd01B_S

Function

Send 1 byte commands with 1-out-of-8 check

The 1-out-of-8 check is performed by the data point typical.

The 1-out-of-n check is performed by FC Safe.

Note

FC Safe required

With Cmd01B_S, data can only be transmitted when FC Safe is linked in cyclic program.

Parameters

Parameter:	PartnerNo
	PartnerObjectNo
	For a description, see section Reoccurring parameters (Page 207).

Parameter: **Enabled**

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Typical-specific response: Enable block processing

If processing is disabled the FB only checks to see if the disabled status has been canceled. The block cannot communicate at the organizational level in this status because it cannot send any Org. frames.

Parameter: **CommandInputByte_HW**

Declaration: INPUT

Data type: BYTE

Range of values: See address range

Default: 0 (B#16#0)

Address range:	Input bytes	IB0 ... IBn PIB0 ... PIBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n

Explanation: Command input byte for hardware input.

This command input byte is specially designed for entering commands using hardware, i.e. via digital inputs. Input using memory or data bytes is also possible, but you must then make sure that the command pending at the input byte is reset, which occurs during hardware input when the command button is released.

When input is detected, if no error is detected during the 1-out-of-8 and 1-out-of-n check, and if the central enable memory bit is set, the command is transferred. This is automatically set by FC Safe following a selected time delay set there (see FC Safe, "InputDelayTime" parameter).

If a 1-out-of-8 or 1-out-of-n error is detected, the entered command is no longer processed. A new command is only read in again if previously for the time of one OB1 cycle no hardware command was detected in the CPU at this or another command input block with a hardware input.

The FB enters a detected 1-out-of-8- or 1-out-of-n error in the diagnostic buffer (event ID B171 or B172). The error status is also indicated via the "InputError" output of FC Safe (see FC Safe, "InputError" parameter) and continues to be indicated as long as the error remains.

Name: **CommandInputByte_SW**

Declaration: IN_OUT

Data type: BYTE

Range of values: See address range

Default: 0 (B#16#0)

Address range:	Memory bytes Data bytes	MB0 ... MBn DBm.DBB0 ... n
Explanation:	<p>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.</p> <p>Command input byte for software input</p> <p>This command input byte is specially designed for entering commands using software, i.e. by the user program or an operator panel (OP). When an input is detected and if no error is detected during the 1-out-of-8 and 1-out-of-n check, the command id reset at the input byte and transferred. The central enable memory bit is ignored here because it is only intended for command input over hardware (see "CommandInputByte_HW").</p> <p>If a 1-out-of-8 or 1-out-of-n error is detected, the entered command is no longer processed. A new command is only read in again if previously for the time of one OB1 cycle no software command was detected in the CPU at this or another command input block with a software input.</p> <p>The FB enters a detected 1- out-of-8- or 1-out-of-n error in the diagnostic buffer (event ID B171 or B172). Appropriate error bits are also set in the central data block BasicData where they can be queried by the software. For further details, refer to the description of FC Safe.</p> <p>In principle it is possible to create a new command at "CommandInputByte_SW" in every OB1 cycle. However, only one command per OB1 cycle is allowed and this applies to all command input blocks with software input (1-out-of-n check). An 'empty cycle' between two consecutive software commands is therefore not necessary.</p>	

Note**CommandInputByte_HW / CommandInputByte_SW**

If individual commands need to be entered over the hardware and over the software, the two command inputs "CommandInputByte_HW" and "CommandInputByte_SW" can also be used at the same time.

If a command is entered at the same time over both input bytes, this is only accepted when the same command is entered over the hardware as well as the software input. The hardware input is then processed further.

In all other cases the input is rejected and an error message is entered in the diagnostic buffer (Event ID B170). The error status is also indicated via the InputError output of FC Safe. Appropriate error bits are set in the central data block BasicData where they can be queried by the software (see FC Safe).

3.8.1.13 Command typical Cmd01B_FS**Validity**

S7-1500 only

Function

Send 1 byte commands without 1-out-of-8 test This means that it is not necessary to call the FC Safe in the cyclic program.

There is less transmission security due to the lack of a 1-out-of-8 check.

Parameters

Parameter: **PartnerNo**

PartnerObjectNo

For a description, see section Reoccurring parameters (Page 207).

Parameter: **Enabled**

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Typical-specific response: Enable block processing

If processing is disabled the FB only checks to see if the disabled status has been canceled. The block cannot communicate at the organizational level in this status because it cannot send any org. frames.

Parameter: **HWmode**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: The parameter specifies whether the block works in hardware mode (TRUE) or software mode (FALSE). Depending on the selected mode, the CommandInputByte is available for hardware input or for software input.

In hardware mode, sending of commands is edge-controlled (edge transition from 0 to 1), the commands are not automatically reset.

If a command is recognized, it is only included in the frame to be sent and sent when it has been continuously pending for the configured delay time (InputDelayTime).

A possible delayed second command, for which the input delay time has not yet been reached, is ignored, but is only sent when its input delay time has also expired.

At the same time is entered in the send buffer, the „InputOK" is set for 1 cycle. The same command can only be sent again if it has been reset to "0" beforehand. The „InputDelayTime", „MaxInputTime" and „InputError" parameters apply to all 8 commands of the command byte together and are only relevant in hardware mode.

Name: **InputDelayTime**

Declaration: INPUT

Data type: INT

Range of values:

- 0 or open
When the parameter is not required
- 1 ... 32000
Value range for delay time, only relevant in hardware mode.

Explanation: Delay time in ms for commands entered via hardware.
A delay time of at least 1000 ms is recommended.
The parameter applies to all 8 commands of the command byte. The parameter is irrelevant if the parameter HWmode = FALSE is set.

Name: **MaxInputTime**

Declaration: INPUT

Data type: INT

Range of values:

- 0
When the parameter is not required
- 1 ... 32000
Value range for monitoring time, only relevant in HW mode.

Explanation: Monitoring time in s for commands entered via hardware.
A monitoring time of at least 30 s is recommended.
The parameter applies to all 8 commands of the command byte. The parameter is irrelevant if the parameter HWmode = FALSE is set.

Name: **InputOK**

Declaration: OUTPUT

Data type: BOOL

Range of values:

Output	Q 0.0 ... Q n.7
Memory	M 0.0 ... M n.7
	L 0.0 ... L n.7
Data bit	DBm.DBX 0.0 ... n.7

Explanation: The "InputOK" display is also set for 1 cycle with the entry in the send buffer. The display is independent of the "HWmode" parameter.

Name: **InputError**

Declaration: OUTPUT

Data type: BOOL

3.8 Master copies

Range of values:	Output	Q 0.0 ... Q n.7
	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	Display of input errors for hardware commands	
	The parameter applies to all 8 commands of the command byte.	
	The output is set to 1 if at least one command is pending longer than the value configured under "MaxInputTime".	
	The output is only reset to 0 if all 8 possible commands are free of errors. If a timeout occurs, you have to reset the respective command manually.	

Name:	CommandInputByte	
Declaration:	IN_OUT	
Data type	BYTE	
Range of values:	See address range	
Address range:	Memory bytes	MB0 ... MBn
	Data bytes	DBm.DBBO ... n
	Address range for hardware inputs:	
	Input bytes	IB0 ... IBn
	This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.	
Explanation	Command input byte for hardware input	
	When the HWmode parameter has the value TRUE, this command input byte is specially designed for entering commands using hardware, i.e. via digital inputs. The command is only entered in the send buffer if the partner can be reached. You have to reset the command pending at the input byte yourself.	
	Command input byte for software input	
	When the HWmode parameter has the value FALSE, this command input byte is specially designed for entering commands using software, i.e. by the user program or an operator panel (OP). The command is only entered in the send buffer if the partner can be reached. Commands that have been entered are automatically reset.	

3.8.1.14 Command typical Cmd01B_R

Function

Receive 1 byte commands (1-out-of-8 format)

The 1-out-of-n check is performed by FC Safe.

Note

FC Safe required

With Cmd01B_R, data can only be received if the FC Safe is integrated at the end of the cyclic program.

Parameters

General parameters:	PartnerNo PartnerObjectNo For a description, see section Reoccurring parameters (Page 207).
Name:	Enabled For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).
Typical-specific response:	Enable block processing If processing is disabled the FB only checks to see if the disabled status has been canceled. Any commands that are still received are not output. The FB cannot communicate at the organizational level in this status because Cmd01B_R cannot send or receive Org. frames. If the "Enabled" input is operable by a switch, this local disable means received commands are no longer output. Since the block does not send Org. frames, it cannot report this local disable back to the partner itself. This must be done with another typical, e.g. Bin04B_S.
Name:	MultipleOutput
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE

Explanation: Simultaneous output of multiple commands permitted

With this parameter, you can specify whether or not several (consecutively received) commands can be output simultaneously; in other words, you specify how the block reacts when a new command is received and the previously received command still needs to be output.

Requirement: The command output time has not yet elapsed and the user program has not yet reset this command.

- FALSE
Multiple output is not permitted. The newly received command overwrites the output byte. Any command still pending is therefore reset to 0 unless the new command is identical to the old one.
- TRUE
Multiple output is permitted. A newly received command is written to the current output byte. The command output time is restarted and applies to all pending commands.

Name: **CommandOutputTime**

Declaration: INPUT

Data type: INT

Range of values: 0 ... 500

Default: 500

Explanation: Command output time for command outputs in milliseconds

The specified time applies to all command outputs.

If more than one output can be set at the same time (MultipleOutput = TRUE), the output time is restarted with each newly received command. This means retriggering for already pending commands. All the command outputs are reset at the same time only when the output time elapses.

With the value 0 a set command output is not reset by the command typical. You need to do this via the user program.

Name: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the output byte "CommandOutputByte".

Name: **CommandOutputByte**

Declaration: IN_OUT

Data type: BYTE

Range of values: See address range

Default: 0 (B#16#0)

Address range:	(process image) output bytes	QB0 ... QBn
	Memory bytes	MB0 ... MBn
	Data bytes	DBm.DBB0 ... n
Since the parameter is an IN_OUT parameter, direct I/O output of the command byte to PQB0 ... PQBn is not permitted. It is also difficult to specify local bit memory with this parameter type and this should not be used.		
Explanation: Command output byte		
To allow the command outputs to be reset both by the command typical itself as well as by the user program (when output time = 0), the parameter was declared as an IN_OUT parameter.		

3.8.1.15 Command typical Cmd01B_FR

Validity

S7-1500 only

Function

1 byte commands received without 1-out-of-8 check. There is less transmission security due to the lack of a 1-out-of-8 check.

Parameters

Parameter:	PartnerNo PartnerObjectNo For a description, see section Reoccurring parameters (Page 207).
Parameter:	Enabled For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).
Typical-specific response:	Enable block processing If processing is disabled the FB only checks to see if the disabled status has been canceled. Any commands that are still received are not output. If the "Enabled" input is operated via a switch, this local disable has the effect that received commands are no longer output. Since the block is, however, not capable of sending org. frames, it cannot report this local block back to the partner itself. This must be done with another typical, e.g. Bin04B_S.
Parameter:	CommandOutputTime
Declaration:	INPUT
Data type:	INT

3.8 Master copies

Range of values: 0 ... 500

Default: 500

Explanation: Command output time for command outputs in milliseconds

The specified time applies to all command outputs.

The output time is restarted with each newly received command. This means retriggering for already pending commands. All the command outputs are reset at the same time only when the output time elapses.

With the value 0 a set command output is not reset by the command typical. You need to do this via the user program.

Parameter: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the output byte "CommandOutputByte".

Parameter: **CommandOutputByte**

Declaration: IN_OUT

Data type: BYTE

Range of values: See address range

Default: 0 (B#16#0)

Address (process image) output bytes QB0 ... QBn

Memory bytes MB0 ... MBn

Data bytes DBm.DBBO ... n

Since the parameter is an IN_OUT parameter, direct I/O output of the command byte to PQB0 ... PQBn is not permitted. It is also difficult to specify local bit memory with this parameter type and this should not be used.

Explanation: Command output byte

To allow the command outputs to be reset both by the command typical itself as well as by the user program (when output time = 0), the parameter was declared as an IN_OUT parameter.

3.8.1.16 Command typical Cmd08X_S

Validity

S7-1500

Function

Sending 8-byte commands over 8 channels for specific bits

No 1-out-of-n test is performed because the FC Safe is not required by the typical.

Parameters

General pa- **PartnerNo**

rameters: **PartnerObjectNo**

For a description, see section Reoccurring parameters (Page 207).

Name: **Enabled**

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Typical-spe- Enable block processing

cific response: If processing is disabled the FB only checks to see if the processing lock has been canceled. The block cannot communicate at the organizational level in this status because it cannot send any Org. frames.

Name: **CommandInput01..08**

Declaration: IN_OUT

Data type: Bool

Range of val- See address range

ues: Default: 0

Address Input I 0.0 ... I n.7

range: Memory M 0.0 ... M n.7

Data bit DBm.DBX 0.0 ... n.7

Since the parameter is an in/out parameter (IN_OUT declaration), a direct I/O output of command bits to PQ0 ... PQn is not permitted. In addition, the specification of local memories should not be used.

Explanation: Command input bit

The parameter exists eight times, once per channel.

When the input of a command is detected, the command is immediately transferred. Each command is transferred to bit 0 of a byte and transferred in a copy of the byte.

When a command is detected, the entire command byte and its copy are always transferred.

The command inputs are not reset automatically and are not monitored for a permanently pending value 1. Make sure you reset the pending command.

3.8.1.17 Command typical Cmd08X_R

Validity

S7-1500

Function

Receiving 8-byte commands over 8 channels for specific bits

For 1-out-of-8 check see below.

Parameters

General parameters: **PartnerNo**
PartnerObjectNo

For a description, see section Reoccurring parameters (Page 207).

Name: **Enabled**

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Typical-specific response: Enable block processing

If processing is disabled the FB only checks to see if the processing lock has been canceled. Any commands that are still received are not output. The FB cannot communicate at the organizational level in this status because it cannot send or receive org. frames.

If the "Enabled" input is operable by a switch, this local disable means received commands are no longer output. Since the block does not send Org. frames, it cannot report this local disable back to the partner itself. This must be done with another typical, e.g. Bin04B_S.

Name: **CommandOutputTime**

Declaration: INPUT

Data type: INT

Range of values: 0 ... 500

Default: 500

Explanation: Command output time for command outputs in milliseconds

The specified time applies to all command outputs.

The output time is started separately for each command channel. Only if the same command is received again within the output, is its output time restarted. The output times of the other commands are not affected.

With the value 0 a set command output is not reset by the command typical. You need to do this via the user program.

Name: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the respective "CommandOutputXX" output.

Name: **CommandOutput01..08**

Declaration: IN_OUT

Data type: Bool

Range of values: See address range

Default: 0 (B#16#0)

Address range: Output bits Q 0.0 ... Q n.7

Memory M 0.0 ... M n.7

Data bit DBm.DBX 0.0 ... n.7

Since the parameter is an in/out parameter (IN_OUT declaration), a direct I/O output of command bits to PQ0 ... PQn is not permitted. In addition, the specification of local memories should not be used.

Explanation: Command output bit

The parameter exists eight times, once per channel.

To allow the command outputs to be reset both by the command typical itself and by the user program (when output time = 0), the parameter is declared as an in/out parameter (IN_OUT).

An internal 1-out-of-8 check is performed in each byte.

- If more than 1 bit is received in one byte, the following error is output:
0xB174 - 1-out-of-8 error
- If the copy of the command byte is not identical, the following error is output:
0xB173 - Command and control byte not identical

The command is deleted in both cases.

3.8.1.18 Counted value typicals Cnt01D_S / Cnt04D_S

Function

- Cnt01D_S: Send 1 counted value (32 bits).
- Cnt04D_S: Send 4 counted values (32 bits).
Note that the parameter "DifferenceValue" for forming the difference value can only be activated at the same time for all four counted values.

Parameters

General parameters:

- PartnerNo**
- PartnerObjectNo**
- Enabled**
- ImageMemory**
- Conditional**
- Unconditional**
- TimeStamp**

For a description, see section Reoccurring parameters (Page 207).

Name: **GeneralTriggerCommand**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Restore collective command

The restore collective command as a central system memory bit belongs to the organizational SINAUT system commands.

Set the parameter to TRUE if the counted value transfer is to be triggered by a restore collective command.

- When the destination subscriber no. (PartnerNo) = 0 (transfer to all), the restore collective command is taken into account.
When the restore collective command is detected, the currently accumulated counted value is transferred regardless of other triggers for transfer. The restore bit is inverted in this counted value.
- If in the typical an explicit destination subscriber no. (PartnerNo > 0) is configured, the restore collective command is evaluated in the corresponding subscriber object in the central administration.

You can use the parameters "GeneralTriggerCommand" and "TriggerInput" at the same time. In this case the transfer is triggered both by an edge change 0 → 1 at "TriggerInput" and when a restore collective command is received.

Name: **TriggerInput**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address range:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Trigger input

With the edge change 0 → 1 of the "TriggerInput" input the triggered transfer can be triggered at a required time regardless of other criteria for a transfer.. The currently accumulated counted value is transmitted. The restore bit (see above) is inverted in this counted value.

Example:
Time-driven transmission with time stamp for supplying an archive in the control center.

You can use the parameters "GeneralTriggerCommand" and "TriggerInput" at the same time. In this case the transfer is triggered both by an edge change 0 → 1 at "TriggerInput" and when a restore collective command is received.

With the setting FALSE no restoring and no transfer via the "TriggerInput" input is triggered.

Name: **Counter_1 (Cnt01D_S)**
Counter_1 ... _4 (Cnt04D_S)

Declaration: INPUT

Data type: COUNTER

Range of values: 0 ... 32767

- Z0 as placeholder
or
- Z1 ... Zn
n depends on the CPU type.

Default: -

Explanation: Number of the SIMATIC counter

Here, you specify the SIMATIC counter in which the pulses were counted time-driven. This counting takes place in the background using FC PulseCounter that is called in a cyclic interrupt OB (for example in OB35). See also section FC PulseCounter (Page 331) and Cyclic interrupt OB (Page 201).

The COUNTER data type cannot be preassigned a value.

If you configure the Z0 placeholder, the corresponding counted value is not processed.

Name: **DifferenceValue**

Declaration: INPUT

Data type: INT

Range of values: 0 ... 31767

Default: 0

Explanation: Difference value

- When a value between 1 and 31767 is configured, the counted value is transferred as soon as the difference between the current and most recently transferred counted value reaches the value specified here.
- If the default value 0 is configured, a counted value is transferred only in the following situations:
 - On an edge change 0 → 1 at the "TriggerInput" input
 - On receipt of a restore command when "GeneralTriggerCommand" = TRUE.

Select the difference value dependent on the maximum pulse rate per second.

Do not select a value that is too low so that the counted value is not constantly transferred to the TIM. This would put load on the communications path to the CPU and the send queue of the CPU.

Note on Cnt04D_S

In the typical this processing parameter for forming the difference value only exists once. It applies to all 4 counted values together. It is not possible to set the parameter for the individual counted values. When using this parameter, each typical should therefore only acquire counted values that can be processed identically.

3.8.1.19 Counted value typicals Cnt01D_R / Cnt04D_R

Function

- Cnt01D_S: Receive 1 counted value (32 bits)
- Cnt04D_S: Receive 4 counted values (32 bits)

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled

For a description, see section Reoccurring parameters (Page 207).

Name: **BCD_Format**
Declaration: INPUT
Data type: BOOL
Range of values: TRUE / FALSE
Default: TRUE

Explanation: Counted value output in BCD format

- If the parameter is activated the received counted value is output as a positive BCD value at the "CountedValueOutput_n" output.
- If the parameter is deactivated the counted value is output as a positive 32-bit integer value.

For the different value ranges of the two formats, see the parameter "Counted-ValueOutput_n".

If the maximum counted value that can be represented is exceeded, the counted value starts again at 0 and counting continues in the positive numeric range.

Note on Cnt04D_R

The parameter exists only one in the typical and it applies to all 4 counted values together. It is not possible to make an individual setting per counted value. When using this parameter, per typical only counted values with an identical output format should be output.

Name: **CntValInvalid**

Declaration: OUTPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address Output Q 0.0 ... Q n.7

range: Memory M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit DBm.DBX 0.0 ... n.7

Explanation: Counted value invalid

When evaluating "CntValInvalid" remember that the bit might only be set for one OB1 cycle.

The CntValInvalid output indicates whether the last received counted value was invalid. . With "Cnt04D_R", this counts as a group display for all 4 counted values, see the note below.

The output shows the validity status of the most recently received counted value in inverted form.

The output serves the following purposes:

- Error display
- Signal for user-specific further processing
You can, for example, react to the lack of up-to-dateness, by correcting the counted value output at "CountedValueOutput_n" with possibly lost counting pulses.

If you do not require the parameter, simply leave it open.

Note on Cnt04D_R

Although all 4 counted values have their own status bit in the data frame, for the status at the output "CntValInvalid" only the bit of the first counted value in the previously received data frame is evaluated.

This status, however, applies to all 4 counted values.

(All counted values in the data frame always have the same status.)

Name: **RestoreStatus**

Declaration: OUTPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address	Output	Q 0.0 ... Q n.7
range:	Memory	M 0.0 ... M n.7
		L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Status of the restore bit in the received counted value.

The "RestoreStatus" output indicates the current status of the restore bit from the last received counted value frame.

You can use the output for user-specific further processing.

Example:

You can only access the information at "CountedValueOutput_n" when a change has been detected at the "RestoreStatus" output; in other words, when the counted value has been received due to a restore, such as a local time-driven restore.

If you do not require the parameter, simply leave it open.

Note on Cnt04D_R

Although all 4 counted values have their own restore bit in the data frame, for the status at the "RestoreStatus" output only the restore bit of the first counted value in the previously received data frame is evaluated.

This status, however, applies to all 4 counted values.

(All counted values in the data frame always have the same status.)

Name: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

The output "NewData" is always set to TRUE for one OB1 cycle when the FB has received new data and output it to the output / outputs "CountedValueOutput_1" to "CountedValueOutput_4"

Name: **CountedValueOutput_1 (Cnt01D_S)**
CountedValueOutput_1 ... _4 (Cnt04D_S)

Declaration: IN_OUT

Data type: DWord, UInt

Range of values:	• Integer:	0 ... 2 147 483 647
	• BCD:	0 ... 9 999 999

Default: 0

Address range:	Output (DWORD)	QD0 ... QDn
	Memory (DWORD)	MD0 ... MDn
	Data (DWORD)	DBm.DBB0 ... n
<p>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.</p> <p>It is also difficult to specify local bit memory with this parameter type and this should not be used.</p>		
Explanation:	<p>Counted value output</p> <p>The counted value typical always adds the newly formed difference value (difference between the new and last received counted value) to the value currently output at the counted value output.</p> <p>The counted value output is a double word in which the counted value is stored in BCD format or as a 32-bit integer value (depending on the "BCD_Format" parameter, see above).</p> <p>The counted value id always output as a positive number. If the maximum counted value that can be represented is exceeded, the counted value starts again at 0 and counting continues in the positive numeric range.</p> <p>Since the parameter is an in-out parameter (IN_OUT), the value can be reset to 0 or another value at the counted value output by the user program at any time.</p>	

3.8.1.20 Data typical Dat12D_S

Function

Send maximum of 12 double words with any data content.

The content of each double word may be a value in double word (DWORD, DINT, REAL) format, it can also be a mixture of other data types which together form a double word, for example:

- 4 bytes
- 2 words
- 2 bytes + 1 word

Sending the data area can be triggered in two ways:

- By a change check
The data is transferred as soon as a bit changes ("SendOnChange" = TRUE).
- By the user program
The transfer can be triggered by an edge change 0 → 1 at the "TriggerInput" input
For time-driven transfer FC Trigger can be used.

With "SendAll" you can also specify whether the transfer always includes all data or only the data double words that have changed.

Note**Remember double word boundaries**

When changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 12 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Note**DB with standard access**

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional
TimeStamp

For a description, see section Reoccurring parameters (Page 207).

Name: **SendOnChange**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Send on change

With the setting TRUE, the block runs a change check within the acquired data area "DataInput". The block checks whether at least one bit has changed. If a change is detected, a transfer of the data area is started automatically. You specify whether the entire area is transferred or only the changed part with the "SendAll" parameter.

If the setting is FALSE, you need to trigger the transfer via the input parameter "TriggerInput".

Name: **TriggerInput**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address	Input	I 0.0 ... I n.7
range:	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Trigger input

With the edge change 0 → 1 of the "TriggerInput" input, the transfer of the data frame can be triggered at a required time.

Example:

Time-driven analog value transfer with time stamp for supplying an analog value archive in the control center.

Make sure that you set the "ImageMemory" parameter to FALSE to prevent this data with time stamps from being overwritten when saving on the station TIM.

The FC Trigger block can be used for time-driven triggering of a transmission over "TriggerInput".

"TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. With dedicated lines or wireless networks the transfer is immediate. With a dial-up connection, it is possible that the data frame is saved first on the TIM and sent at a later point in time. The reason can, for example, be that the data frame is marked as "Conditional spontaneous", see parameter "Conditional".

If you do not require the parameter, simply leave it open. You should, however, then set the "SendOnChange" parameter to TRUE so that the data is transmitted automatically at every change.

For the triggering you can also select a combination of "SendOnChange" plus "TriggerInput". This means that a transfer is triggered both when a change is detected and at every edge change from 0 to 1 at the "TriggerInput" input.

If you use neither "SendOnChange" nor "TriggerInput" to trigger data transfer, the data will only be transferred when there is a single request for this data object or within the framework of a general request.

Name: **SendAll**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: TRUE

Explanation: Send all data with every transfer

With the parameter, you specify whether the block will always transfer all data of the area specified with "DataInput" or only changed data. The transfer can be triggered by the activated change check (SendOnChange = TRUE) or by "TriggerInput".

- SendAll = TRUE
Always send all data
- SendAll = FALSE
Send only changed data

Exception:

If "SendAll" is set to = FALSE, the transfer is triggered by "TriggerInput" and if no data has changed at this time, the complete area will be transferred. For this exception this corresponds to "SendAll" = TRUE.

When only the changed data area is transferred ("SendAll" = FALSE), this area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

If there is a single request for this data object or within the framework of a general request, all data words of the area specified by "DataInput" are always transferred.

Name: **DataInput**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

Address range: P#DBxx.DBX yy.0 DWORD zz

- xx: Data block number 1...32767
- yy: Byte number
- zz: Number of double words 1...12 starting at byte number yy

Example:

P#DB20.DBX100.0 DWORD 4

Remember the periods and spaces when entering the pointer!

Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation: Data input area

The ANY pointer addresses the data area in which the data to be acquired is located. This data area must be within a data block and its length can vary between 1 and 12 data double words.

For information on the content and formats, refer to the section "Function" above.

If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

3.8.1.21 Data typical Dat12D_R

Function

Receive maximum of 12 double words with any data content.

The content of each double word may be a value in double word (DWORD, DINT, REAL) format, it can also be a mixture of other data types which together form a double word, for example:

- 4 bytes
- 2 words
- 2 bytes + 1 word

Dat12D_R stores the received data without further processing in the data area specified by "DataOutput". You need to evaluate and process the received data with the user program.

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled

For a description, see section Reoccurring parameters (Page 207).

Name: **DataOutput**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
 (null pointer)

Address range: P#DBxx.DBX yy.0 DWORD zz

- xx: Data block number 1...32767
- yy: Byte number
- zz: Number of double words 1...12 starting at byte number yy

Example:

P#DB20.DBX100.0 DWORD 4

Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation: Data output area

The ANY pointer addresses the data area in which the received data is saved. This data area must be within a data block and its length can vary between 1 and 12 double words.

For information on the content and formats, refer to the section "Function" above.

Dat12D_R stores the received data without further processing in the data area specified by "DataOutput". You need to evaluate and process the received data with the user program.

When only changed data is sent by the partner object Dat12D_S, it is possible that only part of the data output area is newly written. This is the area in which the changes were detected at the acquisition end.

If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Name: **NewData**

Explanation: Receive new data

For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Whenever the block has received new data values from the partner object and has output them to the output field "DataOutput", the "NewData" output is set to TRUE for one OB1 cycle.

3.8.1.22 Data typical Dat12x1D_S

Validity

S7-1500

Function

Send data a maximum of 12 times with any data content via 12 channels for specific double words.

You define the number of channels of the object used (1..12) in the "ChnCnt" parameter.

One data type is permitted per channel. You specify the data type for the content of a channel in the "DataInputXX" parameter using the respective ANY pointer.

The number of transferred data per channel must not exceed the length of 1 double word (32 bits). If you specify the number of data for one channel with 3 CHAR, for example, the fourth byte is not read and will not be evaluated for the transfer time (SendOnChange).

Associated instance data block

Note

DB with standard access

The typical uses ANY pointers in the "DataInputXX" parameters. Disable the "Optimized block access" attribute in the properties of the DB.

Triggering transfer and transferred data areas

As a maximum, the frame is sent completely with all channels specified in "ChnCnt". Depending on the parameter assignment, not all data of all channels are transferred.

Sending the data can be triggered in the following ways:

- **Time-driven**

The transfer can be triggered via the user program.

With an edge change 0 → 1 at the "TriggerInput", all data specified under "DataInputXX" is always transferred.

For time-driven transfer FC Trigger can be used.

- **On change**

The transfer is triggered via the change control ("SendOnChange" = TRUE).

- On change in only one channel:

Only the data of the changed channel is transferred.

- On change in multiple channels:

The transferred data comprises a contiguous area from the first changed channel to the last changed channel.

Unchanged channels that may be before or after the transferred area are not transferred.

- **Requests**

If there is a single request for this data object or within the framework of a general request, all data of the typical is always transferred.

To trigger the data transfer, you can also select a combination of "SendOnChange" plus "TriggerInput". This means that a transfer is triggered both when a change is detected and at every edge change from 0 → 1 at the "TriggerInput" input.

If you use neither "SendOnChange" nor "TriggerInput" to trigger data transfer, the data is only transferred when there is a single request for this data object or within the framework of a general request.

Two parameter sets for different prioritization of the channels

The typical provides the possibility to handle channels differently with respect to saving and the transfer time by means of two parameter sets. Both parameter sets contain the following parameters:

Parameter set 1

- ImageMemory01
- Conditional01
- SendOnChange01

Parameter set 2

- ImageMemory02
- Conditional02
- SendOnChange02

Function of the parameters:

- **ImageMemory**
Determines whether the data is transferred according to the image memory principle or the send buffer principle.
- **Conditional**
Determines whether the data is transferred "Conditional spontaneous" or "Unconditional spontaneous".
- **SendOnChange**
Determines whether the data is transferred on change.

You can prioritize the channels as follows, for example, using the different parameter assignment of the respective parameters:

- Channels for parameter set 1
Important events to be saved individually and transferred immediately.
- Channels for parameter set 2
Operating data whose values may be overwritten and which does not need to be transferred immediately.

Each channel is assigned to one of the two parameter sets using the "ChnSets" parameter. You will find a parameter assignment example below.

Parameters

General parameters:

PartnerNo

PartnerObjectNo

Enabled

ImageMemory01..02 *

Conditional01..02 *

TimeStamp

For a description, see section Reoccurring parameters (Page 207).

* Parameter one time each for parameter set 1 and 2

Name: **SendOnChange01..02 ***

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Parameter as- * Parameter once for parameter set 1 and 2
signment:

Explanation: Send on change

The parameter exists two times, once per channel.

With the setting TRUE, the block runs a change check within the acquired data area "DataInput". The block checks whether at least one bit has changed. If a change is detected, a transfer of the data area is started automatically. See above for information on the transferred area.

If the setting is FALSE, you need to initiate the transfer via the input parameter "TriggerInput".

Name: **TriggerInput**

Declaration: INPUT

Data type: BOOL

Range of val- TRUE / FALSE
ues:

Default: FALSE

Address Input

I 0.0 ... I n.7

range: Memory

M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit

DBm.DBX 0.0 ... n.7

Explanation: Trigger input

With the edge change 0 → 1 of this input, the transfer of the data frame can be triggered at a selected time.

Example: Time-driven analog value transfer with time stamp for supplying an archive in the control center.

Make sure that you set the "ImageMemory" parameter to FALSE to prevent this data with time stamps from being overwritten when saving on the station TIM. The FC Trigger block can be used for time-driven triggering of transmission over "TriggerInput".

"TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values and transferred to the local TIM.

With dedicated lines or wireless networks, the TIM transfers the data immediately.

With a dial-up connection, the transfer depends on the setting of the parameter "Conditional":

- "Conditional" = TRUE (conditional spontaneous)
The data frame is first saved on the TIM and transferred on the next connection establishment.
- "Conditional" = FALSE (unconditional spontaneous)
The data frame is transferred immediately.

Name: **ChnCnt**

Declaration: INPUT

Data type: Int

Range of values: 1..12

Explanation: Number of transfer channels

Of the max. 12 channels of the object, between 1 and 12 channels can be defined.

Name: **ChnSets**

Declaration: INPUT

Data type: WORD

Format: Bit sequence of 16 bits

Default: 0000000000000000

Explanation: Assignment of transfer channels to parameter set 1 or 2

Assign each channel via its bit to one of the two parameter sets, beginning with bit 1 for channel 1. Bit 0 always needs to be occupied with 0 (null) (reserved).

Coding of the individual bits:

- **0 = Assignment of a channel to parameter set 1**
- **1 = Assignment of a channel to parameter set 2**

With the two parameter sets, you can transfer the channels differently.

Example:

- Parameter set 1
Important events, possible parameter assignment:
 - ImageMemory01 = FALSE (Send buffer principle)
 - Conditional01 = FALSE (Unconditional spontaneous)
 - SendOnChange01 = TRUE (Transfer to TIM on change)
- Parameter set 2
Operating data, possible parameter assignment:
 - ImageMemory02 = TRUE (Image memory principle)
 - Conditional02 = TRUE (Conditional spontaneous)
 - SendOnChange02 = TRUE (Transfer to TIM on change)

Note:

"ImageMemory" and "Conditional" apply to the entire frame.

- If sending of the frame is triggered via "TriggerInput", then the entire frame is sent as unconditional spontaneous send buffer frame as soon as "ImageMemory" and "Conditional" are set to FALSE for a channel.
- If sending of the frame is triggered via "SendOnChange" to multiple channels, then the entire frame is sent as unconditional spontaneous send buffer frame as soon as "ImageMemory" and "Conditional" are set to FALSE for a channel.
- If a change is only detected at one channel ("SendOnChange" = TRUE), then the frame is sent in the mode that is specified by the "ImageMemory" and "Conditional" parameters of this channel.

Table 3-7 Assignment of the bits of the "ChnSets" parameter to the channels

Bit	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
Channel	-	-	-	12	11	10	9	8	7	6	5	4	3	2	1	- *
* Bit 0 is reserved.																

Table 3-8 Parameter assignment example for "ChnSets": Assignment of seven channels to parameter set 1 or 2

Bit	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
Assignment	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0 *
* Bit 0 is reserved. Seven channels are assigned in the example. Channels 8..12 are not used. Coding of the individual bits: <ul style="list-style-type: none"> • 0 = Assignment to parameter set 1 In the example, the channels 2, 3 and 6 are assigned to parameter set 1. • 1 = Assignment to parameter set 2 In the example, the channels 1, 4, 5 and 7 are assigned to parameter set 2. 																

Name:	DataInput01..12
	The parameter exists twelve times, once per channel.
Declaration:	INPUT
Data type:	ANY
Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)
Address range:	P#DBxx.DBXyy.0 TYPE zz <ul style="list-style-type: none"> • xx: Data block number • yy: Byte number • TYPE: Data type Permitted data types are: Bool, Byte, Char, SInt, USInt, Int, UInt, Word, DInt, DWord, Real, UDInt • zz: Length as number (specified data type) as of byte no. yy Max. length: 1 DWord
	Example: P#DB20.DBX100.0 BYTE 3 Area with 3 bytes in DB20 as of DBB100
Explanation:	Data input area <p>The ANY pointer addresses the data area in which the data to be acquired is located. The occupied null pointer is not permitted. Specify a pointer with a real address.</p> <p>The data area must be in a data block and can have a maximum length of 4 bytes (max. 1 double word).</p> <p>If not all 4 bytes are transferred for a channel or if they are not to be evaluated for triggering of the frame (SendOnChange), make sure you observe the correct parameter assignment.</p> <p>For information on the content and formats, refer to the section "Function" above.</p> <p>If the parameter assignment is incorrect (null pointer, length > 4, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.</p>

3.8.1.23 Data typical Dat12x1D_R

Validity

S7-1500

Function

Receive data a maximum of 12 times with any data content via 12 channels for specific double words

You define the number of channels of the object used (1..12) in the "ChnCnt" parameter.

One data type is permitted per channel. You specify the data type for the content of a channel in the "DWxx_DataOut" parameter using the respective ANY pointer.

Dat12x1D_R saves the received data without further processing for specific channels in the respective data area specified at "DWxx_DataOut". You need to evaluate and process the received data with the user program.

Note

DB with standard access

The typical uses an ANY pointer in the "DWx_DataOut" parameter. Therefore, disable the "Optimized block access" attribute in the properties of the DB.

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled

For a description, see section Reoccurring parameters (Page 207).

Name: **ChnCnt**

Declaration: INPUT

Data type: Int

Range of values: 1..12

Explanation: Number of transfer channels

Of the max. 12 channels of the object, between 1 and 12 channels can be defined.

Name: **DW1_DataOut .. DW12_DataOut**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

Address range: P#DBxx.DBXyy.0 TYPE zz

- xx: Data block number
- yy: Byte number
- TYPE: Data type
Permitted data types are: Bool, Byte, Char, SInt, USInt, Int, UInt, Word, DInt, DWord, Real, UDInt
- zz: Length as number (specified data type) as of byte no. yy
Max. length: 1 DWord

Example:

P#DB20.DBX100.0 BYTE 3

Area with 3 bytes in DB20 as of DDB100

Explanation: Data output area

The parameter is available once for each channel (max. 12).

The ANY pointer addresses the data area in which the received data is saved. The occupied null pointer is not permitted. Specify a pointer with a real address.

The data area must be in a data block and can have a maximum length of 4 bytes (max. 1 double word).

If not all 4 bytes are to be output for a channel, make sure you observe the correct parameter assignment.

For information on the content and formats, refer to the section "Function" above.

Dat121xD_R stores the received data without further processing in the data area specified at "DataOutput". You need to evaluate and process the received data with the user program.

When only changed data is sent by the partner object Dat121xD_S, it is possible that only part of the data output area is re-written. This is the area in which the changes were detected at the acquisition end. You can identify changed areas by "NewData".

If the parameter assignment is incorrect (null pointer, length > 4, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Name: **NewData**
 Declaration: OUTPUT
 Data type: Array [0..11] of BOOL
 Format: Bit sequence of 16 bits
 Range of values: See address range
 Default: 0000000000000000

Address range:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n
	Explanation:	Receive new data

Whenever the block receives new data segments, the display of status bits 1 to 12 according to the received data segments 1 to 12 appears in "NewData".
If at least one data segment of the received data contains changes, bit 0 of "NewData" is set to TRUE for one OB1 cycle.
When receiving a sequence of several data segments (data frames), the status bits 1 to 12 in the "NewData" parameter are set to TRUE one after the other and remain set to TRUE until the last segment has been received.
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.

3.8.1.24 Data typical Dat256D_S

Validity

Only CPU 1500

Function

Send a maximum of 256 double words with any data content

The content of each double word can be a value in double word format (DINT, REAL etc.). A combination of other formats is permitted that together result in a double word again, for example

- 32 Bool
- 4 bytes
- 2 words
- Any combination such as 2 bytes plus 1 word etc.

Note

Remember double word boundaries

When changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 256 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Sending the data area can be triggered in two ways:

- By a change check
The data is transferred as soon as a bit changes ("SendOnChange" = TRUE).
- By the user program
The transfer can be triggered by an edge change 0 → 1 at the "TriggerInput" input
For time-driven transfer FC Trigger can be used.

With "SendAll" you can also specify whether the transfer always includes all data or only the data double words that have changed.

With S7-300 CPUs with X communication, the maximum length of a data frame is 76 bytes. 1024 bytes of net data are transferred using a serial transfer process consisting of a sequence of at least 22 data frames (segments). Each data frame apart from the last contains a segment of 48 bytes of net data of the input data area.

To ensure data consistency when the "SendAll" parameter is activated or during a general or single request, the data is transferred in consecutive segments. During the transfer process, the status is indicated by "SendAllBusy". On the recipient, the status is indicated at the "DataStatus" output.

Note

TriggerInput - SendAllBusy

If "TriggerInput" is triggered when "SendAllBusy" = TRUE, this leads to the "DataLoss" error message (status in the frame header) if the transfer is triggered again.

Only when "SendAllBusy" = FALSE is set is the edge change 0 → 1 triggered at "TriggerInput".

If the transfer is interrupted, "SendAllError" is indicated. An entry is also made in the diagnostics buffer with the event ID B14DTD7_Diagnostics.

If the transfer is incomplete, the data status at the recipient is also "invalid". This is indicated on the recipient in the DataStatus parameter. Apart from this an entry with the event ID B13BTD7_Diagnostics is written to the diagnostics buffer.

Note

Availability of the partner

If the status of the partner changes from "available" to "unavailable", the transfer of all data is stopped immediately. All object data is deleted from the TIM buffer. This can lead to loss of data.

As soon as the partner is available again, the automatic general request ensures that the data of the partner is up-to-date again for the next transfer.

Note

Dat256D_S and Dat256D_R require the UDT "TransmitBlock".

When using the typical, copy the UDT from the global library into the "PLC data types" directory of the CPU. The UDT is automatically referenced by the typical from the block directory of the CPU, but not from the global library.

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameters

General parameters:

- PartnerNo**
- PartnerObjectNo**
- Enabled**
- Unconditional**
- TimeStamp**

For a description, see section Reoccurring parameters (Page 207).

Name: **SendOnChange**
 Declaration: INPUT
 Data type: BOOL
 Range of values: TRUE / FALSE

3.8 Master copies

Default: FALSE

Explanation: Send on change

With the setting TRUE, the block runs a change check within the acquired data area "DataInput". The block checks whether at least one bit has changed. If a change is detected, a transfer of the data area is started automatically. You specify whether the entire area is transferred or only the changed part with the "SendAll" parameter.

If the setting is FALSE, you need to trigger the transfer via the input parameter "TriggerInput".

Name: **TriggerInput**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address	Input	I 0.0 ... I n.7
range:	Memory	M 0.0 ... M n.7
		L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Trigger input

With the edge change 0 → 1 of the "TriggerInput" input, the transfer of the data frame can be triggered at a required time.

Example:

Time-driven analog value transfer with time stamp for supplying an analog value archive in the control center.

The FC Trigger block can be used for time-driven triggering of a transmission over "TriggerInput".

"TriggerInput" actually only triggers transmission indirectly. With a 0 → 1 edge change at "TriggerInput", the data frame is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. With dedicated lines or wireless networks the transfer is immediate. With a dial-up connection, it is possible that the data frame is saved first on the TIM and sent at a later point in time. The reason can, for example, be that the data frame is marked as "Conditional spontaneous", see parameter "Conditional".

Select suitable trigger points so that the data on the TIM is not overwritten by buffer overflow (intervals too long).

If you do not require the parameter, simply leave it open. You should, however, then set the "SendOnChange" parameter to TRUE so that the data is transmitted automatically at every change.

For the triggering you can also select a combination of "SendOnChange" plus "TriggerInput". This means that a transfer is triggered both when a change is detected and at every edge change from 0 to 1 at the "TriggerInput" input.

If you use neither "SendOnChange" nor "TriggerInput" to trigger data transfer, the data will only be transferred when there is a single request for this data object or within the framework of a general request.

Do not transfer any analog values for which the "SendOnChange" parameter = TRUE is set without first preprocessing the process data. You will find more detailed information on this with the analog value typical Ana04W_S, parameter "ThresholdValue".

Name: **SendAll**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: TRUE

Explanation: Send all data with every transfer

With the parameter, you specify whether the block will always transfer all data of the area specified with "DataInput" or only changed data. The transfer can be triggered by the activated change check (SendOnChange = TRUE) or by "TriggerInput".

- SendAll = TRUE
Always send all data
- SendAll = FALSE
Send only changed data

Exception:

If "SendAll" is set to = FALSE, the transfer is triggered by "TriggerInput" and if no data has changed at this time, the complete area will be transferred. For this exception this corresponds to "SendAll" = TRUE.

If there is a single request for this data object or within the framework of a general request, all data words of the area specified by "DataInput" are always transferred.

Name: **DataInput**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

Address range: P#DBxx.DBX yy.0 DWORD zz

- xx: Data block number 1...32767
- yy: Byte number
- zz: Number of double words 1...256 starting at byte number yy

Example:

P#DB20.DBX100.0 DWORD 200

Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation: Data input area

The ANY pointer addresses the data area in which the data to be acquired is located. This data area must be within a data block and its length can vary between 1 and 256 data double words. For information on the possible double word formats, refer to the section "Function" above.

If the parameter assignment is incorrect (null pointer, length > 256, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Data consistency;

If a data segment to be transferred consists of a maximum of 48 bytes, data consistency during the transfer is assured.

Name: **SendAllBusy**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Block being processed with "SendAll" = TRUE

This output indicates that the block is currently transferring the data specified by "DataInput". The procedure is activated either by a remote single or general request or by a local internal or external trigger.

If "SendAll" is set to TRUE, the transfer of all data is triggered either by internal change control (SendOnChange = TRUE) or by the external "TriggerInput" (edge change 0 → 1).

The edge change 0 → 1 has no effect whatsoever with an external "TriggerInput" as long as "SendAllBusy" indicates TRUE. The edge change 0 → 1 of "TriggerInput" only takes effect when "SendAllBusy" = FALSE.

Name: **SendAllError**
 Declaration: INPUT
 Data type: BOOL
 Range of values: TRUE / FALSE
 Default: FALSE
 Explanation: Error when processing "SendAll"
 "SendAllError" is set to TRUE when the connection is interrupted during processing of "SendAll". Apart from this an entry is written to the diagnostics buffer.
 "SendAllError" remains set to TRUE until it is reset by the user program or by the next CPU restart.

3.8.1.25 Data typical Dat256D_R

Validity

Only CPU 1500

Function

Receive maximum of 256 double words with any data content.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 32 Bool
- 4 bytes
- 2 words
- Any combination such as 2 bytes plus 1 word etc.

Note

Remember double word boundaries

When changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 256 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Dat256D_R stores the received data without further processing in the data area defined by "DataOutput". Evaluate the received data with the user program.

3.8 Master copies

With S7-300 CPUs with X communication, the maximum length of a data frame is 76 bytes (net 48 bytes). 1024 bytes of net data are transferred using a serial transfer process consisting of a sequence of at least 22 data frames (segments). Each data frame apart from the last contains a segment of 48 bytes of net data of the output data area.

Each time a detected data segment is received, this is indicated by a corresponding status (bit 1 to 22) of the "NewData" output parameter.

If a change was detected in the data segment, the status bit 0 is also set to TRUE in "NewData" for one CPU cycle. This makes it possible to recognize which segment of the output data area has changed.

Note

When receiving a sequence of several data segments (data frames), the status bits 1 to 22 in the "NewData" parameter are set to TRUE one after the other and remain set to TRUE until the last segment has been received.

If a data segment (data frame) is not part of a received sequence (SendAll = FALSE), the status remains set to TRUE for only one CPU cycle.

To ensure data consistency when "SendAll" = TRUE or during a general or single request, the data area is updated in consecutive individual segments.

During receipt, the status is indicated by the "DataStatus" output byte ("SequenceState" status). If the receive sequence was completed successfully, the data output area is up-to-date and the output data is consistent. This is indicated by "DataStatus" ("DataValid" = TRUE status).

Note

The consistency of the data segments or the limit segments cannot be guaranteed if the "SendAll" parameter was set to FALSE on the sender.

Receipt of a sequence can be disrupted by the following causes:

- The receive sequence was interrupted when communication to the partner fails during an active sequence (event ID B13BTD7_Diagnostics).
- The monitoring time was exceeded. Not all segments could be received within the time set for the "MonitoringTime" parameter (event ID B13CTD7_Diagnostics).
- Other receiving errors occur (event ID B13DTD7_Diagnostics), for example:
 - A new receive sequence is registered during an active, error-free sequence.
 - A spontaneous segment (data frame) is received during an active sequence.

Note

Dat256D_S and Dat256D_R require the UDT "TransmitBlock".

When using the typical, copy the UDT from the global library into the "PLC data types" directory of the CPU. The UDT is automatically referenced by the typical from the block directory of the CPU, but not from the global library.

Note**DB with standard access**

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameters

General parameters: **PartnerNo**
PartnerObjectNo
Enabled

For a description, see section Reoccurring parameters (Page 207).

Name: **SingleRequest**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: TRUE

Address Input

range: Memory

I 0.0 ... I n.7

M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit

DBm.DBX 0.0 ... n.7

Explanation: A single request is sent to the partner object.

If the partner is available, you can send a single request to the partner object. If a reply is returned, the information is forwarded to the data area specified in "DataOutput".

In terms of transfer sequences, there are priorities:

- **Lowest priority: TriggerInput**
An active transfer triggered, for example, by "TriggerInput" on the sender can be interrupted by a single request or a general request.
- **Medium priority: Single request**
An active transfer triggered, for example, by "TriggerInput" on the sender can be interrupted by a single request or a general request.
The interrupted or restarted request leads to a restart of the active sequences without an error message.
- **Highest priority: General request**
A general request can interrupt itself or a single request.
The interrupted or restarted request leads to a restart of the active sequences without an error message.

The request interrupted or restarted by a single or general request leads to a restart of the active sequences without an error message. If a sequence was completed successfully, the "DataValid" status of the "DataStatus" output byte remains set to TRUE.

The time taken for the response to the single request is evaluated by the "MonitoringTime" parameter.

Note:

Consistency of the data output area across segments is only assured if the receive sequence was completed successfully.

Name: **MonitoringTime**

Declaration: INPUT

Data type: INT

Range of values: 0 (no limit) / 1 ... 32000 (seconds)

Default: 0

Explanation: Maximum time for a complete response to a single request

Each time a single request starts (see SingleRequest parameter), the time specified here is started in "SingleRequest".

If a value higher than 0 is entered and the time for the response sequence is exceeded, an error is indicated via the "DataStatus" output byte (status bits "SequenceState"). Apart from this an entry is written to the diagnostics buffer (event ID B13CTD7_Diagnostics).

Each time a single request starts, "MonitoringTime" is reactivated.

Name: **DataOutput**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

Address range:	<p>P#DBxx.DBX yy.0 DWORD zz</p> <ul style="list-style-type: none"> • xx: Data block number 1...32767 • yy: Byte number • zz: Number of double words 1...256 starting at byte number yy <p>Example:</p> <p>P#DB20.DBX100.0 DWORD 200</p> <p>Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.</p>
Explanation:	<p>Data output area</p> <p>The ANY pointer addresses the data area in which the received data is saved. This data area must be within a data block and its length can vary between 1 and 256 data double words. For information on the possible double word formats, refer to the section "Function" above.</p> <p>Dat256D_R stores the received data without further processing in the data area specified by "DataOutput". You need to evaluate and process the received data with the user program.</p> <p>When only changed data is sent by the partner object Dat256D_S, it is possible that only part of the data output area is newly written. This is the area in which the changes were detected at the acquisition end.</p> <p>If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.</p>
Name:	NewData
Declaration:	OUTPUT
Data type:	DWORD
Range of values:	See address range
Default:	0 (DW#16#0)

3.8 Master copies

Address range:	Output (DWORD)	QD0 ... QDn PQD0 ... PQDn
	Memory (DWORD)	MD0 ... MDn LD0 ... LDn
	Data (DWORD)	DBm.DBB0 ... n
	Explanation:	Receive new data

Whenever the block receives new data segments, the display of status bits 1 to 22 according to the received data segments 1 to 22 appears in "NewData".

If at least one data segment of the received data contains changes, bit 0 of "NewData" is set to TRUE for one OB1 cycle.

When receiving a sequence of several data segments (data frames), the status bits 1 to 22 in the "NewData" parameter are set to TRUE one after the other and remain set to TRUE until the last segment has been received.

If a data segment (data frame) is not part of a received sequence ("SendAll" = FALSE), the status remains set to TRUE for only one CPU cycle.

The output is intended for user-specific further processing, for example to react in a specific way to receipt of new data.

If you do not require the parameter, simply leave it open.

Name: **DataStatus**
Declaration: OUTPUT
Data type: BYTE
Range of values: See address range

Default: 0 (B#16#0)

Address range:	Output	QB0 ... QBn PQB0 ... PQBn
	Memory	MB0 ... MBn LB0 ... LBn
	Data	DBm.DBB0 ... n
	Explanation:	Currentness status of a received data segment

- During receipt of a sequence, the current status is indicated by the "DataStatus" output byte:
- If the receipt of the sequence was completed successfully, the "DataOutput" data output area is up-to-date. The status bit "DataValid" is set to TRUE. If "SendAll" is set to TRUE on the sender, the data is consistent.
 - If the receipt of a sequence is disrupted, the "DataOutput" data output area is not up-to-date and "DataStatus" indicates an error.
 - The status bit "DataValid" is set to FALSE and an entry is written to the diagnostics buffer.
 - The "SequenceState" status shows the error (see table).

Table 3-9 Bit assignment of "DataStatus"

Bit	Name	Value	Meaning
0	DataValid	FALSE	Data invalid
		TRUE	Data valid
1 ... 5	Reserved	- (FALSE)	Not used
7, 6	SequenceState	0	No data being received or receipt completed without error.
		1	First segment of a sequence received
		2	Second or higher segment of a sequence received
		3	Errors: <ul style="list-style-type: none"> • Transmission sequence aborted • Monitoring time exceeded • Other receiving error

3.8.1.26 Parameter typical Par12D_S

Function

Send 1 to 12 parameter values (each 1 double word) and receive the current, locally valid parameter values from the partner.

The block operates with 1-out-of-n test. The 1-out-of-n check is performed by FC Safe.

Note

FC Safe required

With Par12D_S, data can only be transmitted when FC Safe is linked in the end of cyclic program, see Section FC Safe (Page 302).

The content of each double word may be a value in double word (DWORD, DINT, REAL) format, it can also be a mixture of other data types which together form a double word, for example:

- 4 bytes
- 2 words
- 2 bytes + 1 word

See also the note "Note word boundaries" below.

The data area to be transferred is specified for the "ParameterInput" parameter in the form of an Any pointer. This data area must be within a data block and its length can vary between 1 and 12 double words. The data area sent to the partner or the parameter values entered locally at the partner are returned from there and output here at the "ReturnedParameter" parameter. This output area (Any pointer) must also be within a data block and its length must match that specified for "ParameterInput".

Separate data areas are normally specified for "ParameterInput" and "ReturnedParameter". This makes it easy to recognize what was most recently entered and what is locally valid. However, it is also possible to specify the same data area for both parameters. The two areas

then overlap 100% and therefore always match. In this case, you can no longer distinguish the difference between what has been entered most recently and what is locally valid.

Even when separate areas are specified for "ParameterInput" and "ReturnedParameter", it is still possible to ensure that the "ParameterInput" input area is always synchronized with the mirrored back values of "ReturnedParameter". This can be done manually from case to case with the "ApplyRemoteParamMan" input or automatically by setting the "ApplyRemoteParamAuto" parameter to TRUE.

A parameter can also be set locally at the partner object that receives the parameter. The partner object must then be set to 'local' at the "Local" input parameter (see block Par12D_R). The current status of the "Local" input parameter is reported by the partner object and indicated here at the "LocalOperation" output. As long as the partner object is set to 'local', no parameters are accepted there from other nodes.

The sending of the data area defined by "ParameterInput" can be triggered via the following parameters:

- **EnterInput**
You should use this input parameter when the data area defined at "ParameterInput" is entered over hardware (digital and analog input modules). "EnterInput" must then be connected to a button on a console or panel via a digital input. The transmission of the entered values is then triggered by pressing this button.
The entire data area specified by "ParameterInput" is transferred.
- **ContinuousEnterFunct**
Set the parameter to TRUE if you enter the parameters using software, for example via an operator panel (OP). There is a constant check for changes. When a change is detected in the data area defined with "ParameterInput", the data double words that have changed since the last transfer are transferred.
Only the changed data is transferred (see note "Changed data areas" below).
- **Release**
Use this input parameter if you enter the parameters using software, for example via an OP. The "Release" input should then be set using a function key on the OP. Changes are checked when a 1 signal is detected at the "Release" input. The data double words from the data area specified with "ParameterInput" that have changed since the last transfer are transferred.
Only the changed data is transferred (see note "Changed data areas" below).
- **RetransmitAll**
Use this input parameter if you enter the parameters using software, for example via an OP. The "RetransmitAll" input should then be set using a function key on the OP. When a 1 signal is detected at the "RetransmitAll" input, the data area configured in "ParameterInput" is transferred without checking for changes.
The entire data area specified by "ParameterInput" is transferred.

Note

Changed data areas

When only the changed data area is transferred, this area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read in is 10 double words long. In this case, changes were detected in the second, fifth and eighth double words. The transferred area is therefore from the 2nd to the 8th double word.

Remember word boundaries

When only changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 12 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameters

General parameters:	PartnerNo	
	PartnerObjectNo	
	Enabled	
	For a description, see section Reoccurring parameters (Page 207).	
Name:	EnterInput	
Declaration:	INPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7
		L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Enter input

The transfer of the parameter values at the "ParameterInput" input can be triggered via this input by a signal edge change.

A signal change at "EnterInput" is only taken into account when "ContinuousEnterFunct" = FALSE. If this condition is fulfilled, an edge change 0 → 1 causes the parameter values entered in "ParameterInput" to be adopted and transferred. There is no change check. The entire data area specified in "ParameterInput" is always transferred.

This method of triggering transfer is suitable for input via hardware, for example via a console or control panel. For further information and related parameters, refer to the section "Function" above.

If you do not require the parameter, simply leave it open.

Data checks:

- The parameters that are read in are then transferred 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 again by the FB when a 0 signal was detected at EnterInput for at least one OB1 cycle.
- When a 1-out-of-n error is detected at the hardware input, the entered parameters are no longer processed. New parameters are read in again only when previously for the length of an OB1 cycle, no hardware input via a command, setpoint or parameter block was detected. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172). As long as the error remains, the error status is indicated via the "InputError" output of FC Safe (see FC Safe, "InputError" parameter).

Name: **ContinuousEnterFunct**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Continuous change check

With this parameter, you can decide whether the parameter values at the "ParameterInput" input should be continuously read in and checked for changes. The change check is made by comparison with the last values that were transferred. Only changed values are sent. If more than one change is detected, the block sends the data area in which all changed parameter values are located.

The changed data area that is transferred consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

A new transfer of the parameter values can be triggered via the "RetransmitAll" input (see below) even when the parameter entries have not changed.

This method of transmission triggering is suitable when the parameter values are entered in the ParameterInput area by software, but can also be used for entering the parameters from an operator panel (OP).

For further information refer to the section "Function" above.

If you do not require the parameter, simply leave it open.

Data checks:

The parameters read in are only transmitted if no error is detected during the 1-out-of-n check.

- While for hardware input (see EnterInput) an empty cycle must be detected before new parameter values can be transferred from the block.
- For software input new parameter values can be transferred in every OB1 cycle. This assumes that there is no other software entry by another block pending in this cycle. Otherwise a 1-out-of-n error is detected.

When a 1-out-of-n error is detected during the software input, the entered parameters are no longer processed. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172).

New parameters are read in again only when previously for the length of an OB1 cycle in the CPU, no software input via a command, setpoint or parameter block was detected.

Name: **ApplyRemoteParamAuto**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Automatic synchronization of the input area with the returned area.

With the parameter all the parameter values from the "ReturnedParameter" mirror back area are then copied to the "ParameterInput" area.

In addition to this the mirrored back parameter values are written the send mailbox of the communications DB.

Automatic synchronization is then always performed when new data is received from the partner object (Par12D_R).

If you do not require the parameter, simply leave it open.

Name: **ParameterInput**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

Address range: P#DBxx.DBX yy.0 DWORD zz

- xx: Data block number 1...32767
- yy: Byte number
- zz: Number of double words 1...12 starting at byte number yy

Example:

P#DB20.DBX100.0 DWORD 4

Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation: Parameter input area.

The ANY pointer addresses the data area in which the parameter values to be acquired are located. This data area must be within a data block and its length can vary between 1 and 12 data double words.

For information on the content and formats, refer to the section "Function" above.

If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

How the parameters at ParameterInput are processed depends on whether they are hardware or software entries and how the transfer of this data area is triggered. You will find more information on this in the section "Function" above.

Name: **ReturnedParameter**

Declaration: OUTPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

Address range: P#DBxx.DBX yy.0 DWORD zz

- xx: Data block number 1...32767
- yy: Byte number
- zz: Number of double words 1...12 starting at byte number yy

Example:
P#DB20.DBX100.0 DWORD 4

Note that the preset value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation: Parameter output area

The partner object receiving the parameter values reports back the valid parameter values there. These values are displayed at the "ReturnedParameter" output. If the partner object is set to 'local' and a new input is made there, the parameters changed locally are indicated here by "ReturnedParameter".

The ANY pointer defines the data area in which the received parameter values are output. This data area must be within a data block and its length can vary between 1 and 12 data double words. The length must be identical with the length set for ParameterInput.

After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the current, local, valid parameters are indicated at "ReturnedParameter".

If the parameter setting is incorrect (data area not a data block, length greater than 12 or length different from the length set for ParameterInput), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The FB is then no longer processed, however, until the error has been corrected.

Name: **LocalOperation**

Declaration: OUTPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address range: Output Q 0.0 ... Q n.7

Memory M 0.0 ... M n.7
L 0.0 ... L n.7

Data bit DBm.DBX 0.0 ... n.7

Explanation: Return message from partner object: Object is set to local operation.

At the partner object that receives the parameters a local setpoint entry can also be made.. The partner object Par12D_R must then be set to 'local' at the "Local" input parameter. The current status of the "Local" input parameter is reported by the partner object and indicated here at the "LocalOperation" output.

After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the local currently valid status is displayed at "LocalOperation".

Name: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Whenever the block has received new data and has output it to the output "ReturnedSetpoint" or "LocalOperation", the "NewData" output is set to TRUE for one OB1 cycle.

Name: **Release**

Declaration: IN_OUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address Memory M 0.0 ... M n.7

range: Data bit DBm.DBX 0.0 ... n.7

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Explanation: Trigger input for sending the currently pending parameter values

You can use this input parameter when the parameter is entered by software, for example at an operator panel (OP). "Release" should then be set using a function key on the OP. You can then enter several parameters initially on the OP. The parameters are transferred only when the Release function key is activated.

A change check is performed only with signal 1 at the Release input. The data double words from the data area configured with "ParameterInput" that have changed since the last transfer are transferred.

If you always want to transfer the entire data area specified with "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, see above.

If you do not require the parameter, simply leave it open.

Name: **RetransmitAll**

Declaration: IN_OUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: TRUE

Address range:	Memory Data bit	M 0.0 ... M n.7 DBm.DBX 0.0 ... n.7
Explanation:	<p>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.</p> <p>Trigger input for retransferring the entire data area specified by "ParameterInput".</p> <p>You can use this input parameter when the parameter is entered by software, for example at an operator panel (OP). "RetransmitAll" should then be set using a function key on the OP. When a 1 signal is detected at the "RetransmitAll" input, the entire data area specified by "ParameterInput" is transferred. A change check is not performed</p> <p>The "RetransmitAll" input is reset automatically. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input. Since there is no change check, this would lead to continuous transfer of all parameter values as long as the input has a 1 signal.</p> <p>"RetransmitAll" can also be used as an option in addition to "Release" or "ContinuousEnterFunct" when new parameter values were entered but could not be transferred to the partner (disrupted connection or the partner object is set to 'local'). In this case you can then trigger repeated transfer of the entire data area specified by "ParameterInput" using "RetransmitAll". All changes that were previously entered but are not yet available at the partner are consistently included.</p> <p>"RetransmitAll" can also be used as an independent transfer trigger when you always want to send all entries and not just those that have changed. In this case you can use "RetransmitAll" instead of "Release" that only sends the changed parameter values.</p> <p>Data checks: The same safety checks are carried out as with "ContinuousEnterFunct", see above.</p> <p>If you do not require the parameter, simply leave it open.</p>	

Name:	ApplyRemoteParamMan
Declaration:	IN_OUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE

Address	This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.
Explanation:	<p>Trigger input for synchronization of the input area with the returned area.</p> <p>The input triggers a one-time synchronization of the ParameterInput input area with the "ReturnedParameter" mirror back area. All the parameter values from the "ReturnedParameter" mirror back area are then copied to the "ParameterInput" input area.</p> <p>The send mailbox of the communications DB is also synchronized with the returned parameter values.</p> <p>"ApplyRemoteParamMan" is reset automatically. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input. The result would be a constant synchronization as long as the parameter has a 1 signal.</p> <p>If you do not require the parameter, simply leave it open.</p>

3.8.1.27 Parameter typical Par12D_FS

Validity

- S7-1500

Function

Send 1 to 12 parameter values (each 1 double word) without a 1-out-of-n check and receive the current, locally valid parameter values from the partner. This means that it is not necessary to call the FC Safe in the cyclic program.

There is less transmission security due to the lacking 1-out-of-n check.

No parameters are sent to subscribers that cannot be reached.

Frames of this block can be received and evaluated with the Par12D_R block.

The content per double word can be a value in double word format (DWORD, DINT, REAL).

Note

Remember word boundaries

When only changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 12 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

The data area to be transferred is specified for the "ParameterInput" parameter in the form of an Any pointer. This data area must be within a data block and its length can vary between 1 and 12 double words. The data area sent to the partner or the parameter values entered locally at the partner are returned from there and output here at the "ReturnedParameter"

parameter. This output area (Any pointer) must also be within a data block and its length must match that specified for "ParameterInput".

Mirroring

Separate data areas are normally specified for "ParameterInput" and "ReturnedParameter". This makes it easy to recognize what was most recently entered and what is locally valid.

However, the same data area can also be specified for both parameters. Both areas then overlap 100 % and are then always synchronized. In this case, you can no longer distinguish the difference between what has been entered most recently and what is locally valid.

Even when separate areas are specified for "ParameterInput" and "ReturnedParameter", it is still possible to ensure that the "ParameterInput" input area is always synchronized with the mirrored back values of "ReturnedParameter". You can perform this synchronization automatically by setting the "ApplyRemoteParam" parameter to TRUE.

A parameter can also be set locally at the partner object that receives the parameter. The "Local" input parameter must then be set to TRUE at the partner object. The locally valid parameters can be created by the partner object at the "LocalParameterInput" input parameter; they are then output here to "ReturnedParameter". The local status can be reported by the partner object via the "Local" parameter and displayed here at the "LocalOperation" output. The partner object can be set so that it does not accept any parameters from Remote when it is working in local mode (see Par12D_R block).

Triggering sending

Sending the data area defined at "ParameterInput" can be triggered by the following parameters:

- ContinuousEnterFunct
Automatic transmission in case of a change in the area "ParameterInput" in the software mode of the block (HWmode = FALSE)
- Release
Single triggered transmission in hardware and software mode of the device

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameters

Parameter: **PartnerNo**
PartnerObjectNo
Enabled

For a description, see section Reoccurring parameters (Page 207).

Parameter: **HWmode**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: The parameter specifies whether the block works in hardware mode (TRUE) or software mode (FALSE).

- **Hardware mode**

Transmission can be triggered by a digital input module connected to the "Release" input.

The parameter value is applied in hardware mode when the signal at the release input changes from 0 to 1. Changes at the "ParameterInput" input while "InputDelayTime" is running are not sent.

In hardware mode, the complete data area defined with "ParameterInput" is sent.

In hardware mode, an empty cycle must be detected at the "Release" input before new parameter values can be sent by the block.

- **Software mode**

Sending can be triggered via the "Release" input or via the "ContinuousEnterFunct" function in software mode.

New parameter values can be sent in each OB1 cycle with a software input. The "ContinuousEnterFunct" function can be switched on and off as required in the cycle.

Note: Changed data areas

In software mode, the block transmits selectively, i.e. it only transmits the area of double words in which a change was detected.

Exception: If no change is detected when the "Release" input is actuated, it sends the complete parameter area.

The changed data area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:

The area to be read in is 10 double words long. In this case, changes were detected in the second, fifth and eighth double words. The transferred area is therefore from the 2nd to the 8th double word.

The "**HWmode**" parameter is applied during startup and cannot be changed in the cycle.

Parameter: **ContinuousEnterFunct**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Continuous change check

The parameter is not evaluated if HWmode equals TRUE.

Set the parameter to TRUE if you enter the parameters using software or, for example, via an operator panel (OP).

There is a constant check for changes. The change check is made by comparison with the last values that were transferred.

As soon as a change is detected in the data area defined with "ParameterInput", this results in transmission. The changed values are sent. If several changes are detected, the block sends the data area from the first to the last double word in which a change was detected.

A new transfer of the parameter values can be triggered via the "Release" input (see below) even when the parameter entries have not changed.

If you do not require the parameter, simply leave it open.

Parameter: **ApplyRemoteParam**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address	Input	I 0.0 ... I n.7
range:	Memory bit	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: If the parameter has the value "TRUE", an automatic synchronization of the input area with the returned area is performed.

With the parameter all the parameter values from the "ReturnedParameter" mirror back area are then copied to the "ParameterInput" area.

In addition to this the mirrored back parameter values are written the send mailbox of the communications DB.

Automatic synchronization is then always performed when new data is received from the partner object (Par12D_R). Synchronization can also be triggered once with a signal transition from 0 to 1, e.g. manually via a pushbutton.

If the parameter has the value "FALSE" or the parameter is not specified, no automatic synchronization of the input area with the returned area is performed.

Parameter: **ParameterInput**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

3.8 Master copies

Address range:	<p>P#DBxx.DBX yy.0 DWORD zz</p> <ul style="list-style-type: none">• xx: Data block number 1...32767• yy: Byte number• zz: Number of double words 1...12 starting at byte number yy <p>Example:</p> <p>P#DB20.DBX 100.0 DWORD 4</p> <p>Remember the periods and spaces when entering the pointer!</p> <p>Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.</p>
Explanation:	<p>Parameter input area.</p> <p>The ANY pointer addresses the data area in which the parameter values to be acquired are located. This data area must be within a data block and its length can vary between 1 and 12 data double words.</p> <p>For information on the content and formats, refer to the section "Function" above.</p> <p>If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 7). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.</p> <p>The way in which the values present at "ParameterInput" are subsequently processed depends on whether it is a hardware or a software input and how the transmission is triggered. You can find more information on this in the "Function" section above and under the "HWmode" parameter.</p>
Parameter:	ReturnedParameter
Declaration:	OUTPUT
Data type:	ANY
Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)

Address range: P#DBxx.DBX yy.0 DWORD zz

- xx: Data block number 1...32767
- yy: Byte number
- zz: Number of double words 1...12 starting at byte number yy

Example:
P#DB20.DBX 100.0 DWORD 4

Remember the periods and spaces when entering the pointer!

Note that the preset value (null pointer) is not permitted. A pointer with a real address must be specified.

Explanation: Parameter output area

The partner object receiving the parameter values reports back the valid parameter values there. These values are displayed at the "ReturnedParameter" output. If the partner object is set to 'local' and a new input is made there, the parameters changed locally are indicated here by "ReturnedParameter".

The ANY pointer defines the data area in which the received parameter values are output. This data area must be within a data block and its length can vary between 1 and 12 data double words. The length must be identical with the length set for ParameterInput.

After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the current, local, valid parameters are indicated at "ReturnedParameter".

If 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] = 8). The CPU does not change to STOP. The FB is then no longer processed, however, until the error has been corrected.

Name: **InputDelayTime**

Declaration: INPUT

Data type: INT

Range of values:

- 0 or open
When the parameter is not required
- 1 ... 32000
Value range for delay time (milliseconds) in hardware mode

Explanation: Delay time in milliseconds for the "Release" input when the block is operating in hardware mode.

A delay time of at least 1000 ms is recommended.

Changes to "ParameterInput" while "InputDelayTime" is running are not sent in the current cycle.

Name: **MaxInputTime**

Declaration: INPUT

Data type: INT

Range of values: • 0 or open
When the parameter is not required.

• 1 ... 32000
Value range for monitoring time (seconds) "HWmode" = TRUE.

Explanation: Monitoring time in seconds for the "Release" input if the block is operating in hardware mode.
A monitoring time of at least 30 s is recommended.
If you do not require the parameter, leave it open.

Parameters: **Release**

Declaration: IN_OUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address	Input	I 0.0 ... I n.7
range:	Memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Explanation: Trigger input for sending the currently pending parameter values
You can use the parameter in software or hardware mode of the block.

- **Hardware mode**

If you use the input via an OR, it should be operated via a function key of the OP. You can then enter several parameters initially on the OP. The parameters are transferred only when the Release function key is activated.

In hardware mode, the "Release" input is not automatically reset. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input.

The "Release" input is subject to the "InputDelayTime" and is monitored via "MaxInputTime". If the input is set longer than "MaxInputTime", the timeout is output at "InputError". The "InputError" output does not change back to "0" until the "Release" input is also = 0. (See also "InputError")

- **Software mode**

A change check is performed only with signal 1 at the Release input. The data from the "ParameterInput" area that has changed since the last transmission is sent. If no change is detected when activating the "Release" input, the block transmits the complete parameter area.

In software mode, the "Release" input is automatically reset. You should therefore only specify memory or data inputs as the input.

If you do not require the parameter, simply leave it open.

Parameter: **LocalOperation**

Declaration: OUTPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address Output Q 0.0 ... Q n.7

range: Memory bit M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit DBm.DBX 0.0 ... n.7

Explanation: Return message from partner object: Object is set to local operation.

A parameter can also be set locally at the partner object that receives the parameter. The partner object Par12D_R must then be set to 'local' at the "Local" input parameter. The current status of the "Local" input parameter is reported by the partner object and indicated here at the "LocalOperation" output.

After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the local currently valid status is displayed at "LocalOperation".

Parameter: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Whenever the block has received new data and has output it to the output "ReturnedParameter" or "LocalOperation", the "NewData" output is set to TRUE for one OB1 cycle.

Name: **InputOK**

Declaration: OUTPUT

Data type: BOOL

Range of values: Output Q 0.0 ... Q n.7

Memory M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit DBm.DBX 0.0 ... n.7

Explanation: The display "InputOK" is set for one cycle at the entry of the data frame into the send buffer. The display is independent of the "HWmode" parameter.

If you do not require the parameter, leave it open.

Name: **InputError**

Declaration: OUTPUT

Data type: BOOL

3.8 Master copies

Range of values:	Output	Q 0.0 ... Q n.7
	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	<p>Display of input errors at the "Release" input, if "HWmode" equals TRUE.</p> <p>The output has a 1 signal as soon as the "Release" command lasts longer than defined via MaxInputTime. The "InputError" output is only set to 0 again when the release input has a 0 signal. This can be used, for example, to detect that the input is defective.</p> <p>If you do not require the parameter, leave it open.</p>	

3.8.1.28 Parameter typical 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 to the partner.

The block can receive and output data of the "Par12D_S" and "Par12D_FS" typical.

The content of each double word may be a value in double word (DWORD, DINT, REAL) format, it can also be a mixture of other data types which together form a double word, for example:

- 4 bytes
- 2 words
- 2 bytes + 1 word

See also the note "Note word boundaries" below.

The data area to be transferred is specified for the "ParameterOutput" parameter in the form of an Any pointer. This data area must be within a data block and its length can vary between 1 and 12 double words.

You can also use the block to enter the parameter values locally. The input area for this is specified as an Any pointer with the "LocalParameterInput" parameter. It must be located within a data block and its length must be identical to the length configured in the "ParameterOutput" parameter.

The block only processes the changed data area. In response to a general or single request, on the other hand, the entire parameter set is transferred or mirrored back..

Bumpless switchover between the "Local" and "Remote" operating modes is guaranteed.

Note

Changed data areas

The changed data area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:

The area to be read in is 10 double words long. In this case, changes were detected in the second, fifth and eighth double words. The transferred area is therefore from the 2nd to the 8th double word.

Remember word boundaries

When only changed data is transferred and the data area contains values in double word format, make sure that the double word values are actually located in one of the maximum 12 double words of the data area to be acquired.

Distribution over two consecutive data double words could lead to the transfer of only one word of the double word value (high or low word) because a change has occurred in only that particular word. In this case, the missing word can lead to a data error on the receiving partner (applies to ST7cc, not for an S7 CPU).

Note

DB with standard access

The block has parameters of the "ANY" type. Therefore, leave the "Optimized block access" attribute in the properties of the DB disabled.

Parameters

General parameters:	PartnerNo
	PartnerObjectNo
	Enabled
	ImageMemory
	Conditional
	Unconditional

For a description, see section Reoccurring parameters (Page 207).

Name:	Local
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE

3.8 Master copies

Address range:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Local parameter input released

Release of the local parameter input of the data area specified by "LocalParameterInput".

A setpoint sent by the partner (master station) is not accepted by the object as long as "Local" = TRUE.

The current status of the "Local" input is transferred to the partner.

Bumpless switchover:

- When there is a switchover from "Local" = 0 to 1, the last values output at the "ParameterOutput" output are retained until new parameter values are entered via the local input area "LocalParameterInput".
- When there is a switchback from Local = 1 to 0, the last values at the "ParameterOutput" output are retained until the block receives new parameter values from the partner.

Read the note on the "ContinuousEnterFunct" parameter.

Special situation:

You can also enter the parameter values during local input directly in the output area specified by "ParameterOutput". Either you do not specify an input area for "LocalParameterInput" or you specify the same data area both for "LocalParameterInput" and "ParameterOutput".

This type of parameter input cannot be prevented by the "Local" input. Regardless of the status of the "Local" parameter, the values entered in the output area are sent immediately by the block to the partner.

Local parameter entries can therefore be made regardless of the status of the "Local" input. "Local" only influences the acceptance of parameters sent by the partner:

- Local = 0
The parameters sent by the partner are accepted and output to the "ParameterOutput" data area.
- Local = 1
The parameters sent by the partner are rejected.

In this special situation, the "Release" and "ContinuousEnterFunct" have no function.

A status change of the "Local" parameter is always transferred by the module according to the send buffer principle, even when the parameter "ImageMemory" = TRUE. This ensures that the optional synchronization of the input and output area on the partner is always performed correctly (see Par12D_S, parameters "ApplyRemoteParamMan" and "ApplyRemoteParamAuto").

Name: **ContinuousEnterFunct**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Continuous local parameter acquisition.

With this parameter, you can decide whether the values in the "LocalParameterInput" input area should be continuously read in and checked for changes. The change is checked by comparing the current values at the "ParameterOutput" output.

Changes in the input area are copied immediately to the output area and transferred to the partner object. Only changed values are sent. If there is more than one change, the block sends the data area in which all changed parameter values are located.

The "ContinuousEnterFunct" = TRUE setting only takes effect when the following conditions are met:

- An input area is defined by "LocalParameterInput" and this is not identical to the output area set for "ParameterOutput".
and
- there is a 1 signal at the "Local" input (= TRUE).

With the setting "ContinuousEnterFunct" = TRUE, when the signal 1 is detected at the "Local" input, the values pending at "LocalparameterInput" are adopted immediately and output at the "ParameterOutput" output. The condition is that the local input values differ from the currently output values at this time.

This method of acquisition of local values is suitable when the values are entered in the "LocalParameterInput" area by software.

If you do not require the parameter, simply leave it open.

Name: **LocalParameterInput**

Declaration: INPUT

Data type: ANY

Range of values: See address range

Default: P#P 0.0 VOID 0
(null pointer)

Address range:	<p>P#DBxx.DBX yy.0 DWORD zz</p> <ul style="list-style-type: none"> xx: Data block number 1...32767 yy: Byte number zz: Number of double words 1...12 starting at byte number yy <p>Example:</p> <p>P#DB20.DBX100.0 DWORD 4</p> <p>Note that the default value (null pointer) is not permitted. A pointer with a real address must be specified.</p>
Explanation:	<p>Local parameter input area</p> <p>The ANY pointer addresses the data area in which the parameter values to be acquired are located. This data area must be within a data block and its length can vary between 1 and 12 data double words. The length must be identical to the length specified for "ParameterOutput".</p> <p>For information on the content and formats, refer to the section "Function" above.</p> <p>If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.</p> <p>If you do not require the parameter, simply leave it open.</p>

Name:	ParameterOutput
Declaration:	INPUT
Data type:	ANY
Range of values:	See address range
Default:	P#P 0.0 VOID 0 (null pointer)
Address range:	<p>P#DBxx.DBX yy.0 DWORD zz</p> <ul style="list-style-type: none"> xx: Data block number 1...32767 yy: Byte number zz: Number of double words 1...12 starting at byte number yy <p>Example:</p> <p>P#DB20.DBX100.0 DWORD 4</p> <p>Remember the periods and spaces when entering the pointer!</p> <p>Note that the preset value (null pointer) is not permitted. A pointer with a real address must be specified.</p>

Explanation: Parameter output area

The ANY pointer addresses the data area in which the locally entered parameter values or those received from the partner are output. This data area must be within a data block and its length can vary between 1 and 12 double words.

For information on the content and formats, refer to the section "Function" above.

Par12D saves the received data without further processing in the data area specified by "ParameterOutput". You need to evaluate and process the received data with the user program.

When only changed data is sent by the partner object Par12D_S, it is possible that only part of the data output area is newly written. This is the area in which the changes were detected at the acquisition end.

If the parameter assignment is incorrect (null pointer, length > 12, data area not a DB), an error message is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The block is then no longer processed, however, until the error has been corrected.

Name: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Whenever the block has received new parameter values from the partner object and has output them to the output field "ParameterOutput", the "NewData" output is set to TRUE for one OB1 cycle. This also applies when there is new local input in the status "Local" = 1.

Name: **Release**

Declaration: IN_OUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address	Input	I 0.0 ... I n.7
range:	Memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Explanation: Input for the acceptance of local parameter input.

The acceptance of the parameter values at the "LocalParameterInput" parameter input can be triggered via this input by a signal edge change.

A change from 0 to 1 at the "Release" input is taken into account only when the following conditions are met:

- An input area is specified by the "LocalParameterInput" parameter and this is not identical to the output area specified by "ParameterOutput" and
- The "Local" input is set to TRUE.

You can use "Release" for parameter input via software, e.g. via an operator panel (OP). The "Release" input should then be set using a function key on the OP. You can then enter several parameters initially on the OP. The parameter values are read in and checked for changes only when the Release function key is activated.

The change is checked by comparing the current parameter values at the "ParameterOutput" output. Changes in the input area are then copied immediately to the output area and transferred to the partner.

Only changed values are sent. If there is more than one change, the block sends the data area in which all changed parameter values are located.

The "Release" input is reset automatically. Instead of a memory bit or data bit, a digital input can also be specified as the input. The automatic reset would not work with a digital input. This does not, however, have negative effects. Acquisition via "Release" is edge triggered, in other words only once.

If you do not require the parameter, simply leave it open.

3.8.1.29 FC Safe

Function

The FC Safe ensures reliable input of commands and setpoints by using the 1-out-of-n check.

The FC Safe checks the input for the following data point types:

- Cmd01B_S
- Set01W_S
- Par12D_S

If an input is pending, the FC checks whether there is only one entry pending in the current OB1 cycle and then enables the block reading in. As soon as two (or more) entries are pending in an OB1 cycle, both entries are ignored.

Note

Follow the call sequence when using FSblocks

If you want to use, in addition to the Cmd01B_S, Set01W_S or Par12D_S blocks, also the Cmd01B_FS, Set01W_FS or Par12D_FS blocks, you have to call these blocks after the FC Safe.

In every CPU in which commands and/or setpoints are acquired, the FC Safe needs to be called once in OB1 to complete the Cmd01B_S, Set01W_S or Par12D_S FBs.

The FC has separate monitoring for entries via different system memory areas. They are divided into hardware entries and software entries:

- Hardware entries
 - Input modules (I)
 - Peripheral inputs (PI)
- Software entries
 - Bit memory (M)
 - Data blocks (DB)
 - Local data (L)
 - Operator panels

The two types of entry "hardware entries" and "Software entries" are checked separately by the block. The FC enables the hardware and software entries separately. For each type of entry, only one command or setpoint entry may be detected.

If a single hardware entry and a single software entry are pending at the same time both entries are enabled.

For hardware entries, the following additional condition applies: The entry must be pending constantly for the duration of "InputDelayTime". Only when an entered command or setpoint is pending unchanged for this time and during this time no other command or setpoint entry is detected is the block processing enabled.

The actual putting together of the command or setpoint frame is handled by the block that read in the command or setpoint.

For the hardware entry the FC Safe provides the following two display bits:

- InputOK
Display of the enabling of hardware commands and setpoints
- InputError
Display of entry errors with hardware commands and setpoints

FC-Safe shows a command output error detected in a station via the following output:

- GlobalCmdOutputError
Group message: Command output error in a station

Parameters

Name: **InputDelayTime**
Declaration: INPUT
Data type: INT

Range of values: • 0
Value for unrequired parameter

• 1 ... 32000
Range of values for delay time

Explanation: Delay time in ms for commands and setpoints that are input via hardware.
A delay time of at least 1000 ms is recommended.

Name: **MaxInputTime**

Declaration: INPUT

Data type: INT

Range of values: • 0
Value for unrequired parameter

• 1 ... 32000
Range of values for monitoring time

Explanation: Monitoring time in ms for commands and setpoints that are input via hardware.
A monitoring time of at least 30 s is recommended.
If you do not require the parameter, specify 0 (zero).

Name: **ResetError**

Declaration: INPUT

Data type: BOOL

Range of values: Input I 0.0 ... I n.7

Memory M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit DBm.DBX 0.0 ... n.7

Explanation: Input for resetting the output GlobalCmdOutputError.
If you do not require the parameter, specify a memory or data bit that always has signal 0.

Name: **InputOK**

Declaration: • S7-300 • OUTPUT

• S7-1500 • IN_OUT

Data type: BOOL

Range of values: Output Q 0.0 ... Q n.7
Memory M 0.0 ... M n.7
L 0.0 ... L n.7
Data bit DBm.DBX 0.0 ... n.7

Explanation: Display of the enabling of hardware commands and setpoints
Has a 1 signal as soon as the current entry was enabled, i.e. when the hardware command or setpoint was entered correctly.
The display bit goes off when the entry is reset, in other words when the command key is released or the setpoint entry key "EnterInput" input is released.
If you do not require the parameter, specify a memory or data bit in the memory area of the local data.

Name: **InputError**

Declaration: • S7-300 • OUTPUT
• S7-1500 • IN_OUT

Data type: BOOL

Range of values: Output Q 0.0 ... Q n.7
Memory M 0.0 ... M n.7
L 0.0 ... L n.7
Data bit DBm.DBX 0.0 ... n.7

Explanation: Display of entry errors with hardware commands and setpoints
The output has a 1 signal when one of the following hardware entry errors is detected within the monitoring time "MaxInputTime":

- Two or more command and/or setpoint entries were detected at the same time.
- At one of the inputs a 1 signal was detected over a longer period of time (for example when the input is defective).

If you do not require the parameter, specify a memory or data bit in the memory area of the local data.

Name: **GlobalCmdOutputError**

Declaration: • S7-300 • OUTPUT
• S7-1500 • IN_OUT

Data type: BOOL

3.8 Master copies

Range of values:	Output	Q 0.0 ... Q n.7
	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
	Explanation:	Group message: In a station a command output error was detected. A command output error can occur at the receive end only in the following cases: <ul style="list-style-type: none">• The content of the two command bytes in the received frame is not identical.• More than one bit is set in the command byte (1-out-of-8 error). If such an error is detected, this is sent by the station with an organizational frame to the subscriber that sent the commands. FC Safe on the sending subscriber then indicates the disruption at the output GlobalCmdOutputError. When an error is detected, the output remains set to the 1 signal until you request the group signal to be reset via the "ResetError" input. If you do not require the parameter, specify a memory or data bit in the memory area of the local data.

3.8.1.30 Setpoint typical Set01W_S

Function

Send 1 setpoint as a 16 bit value and receive local setpoint from partner
The block operates with 1-out-of-n test. The 1-out-of-n check is performed by FC Safe.

Note

FC Safe required

With Set01W_S, data can only be transferred when the block FC Safe is linked in the end of the cyclic program, see section FC Safe (Page 302).

Parameters

General parameters:	PartnerNo
	PartnerObjectNo
	Enabled
	For a description, see section Reoccurring parameters (Page 207).
Name:	EnterInput
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE

Address range:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7
		L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Enter input for hardware setpoint

The adoption of a setpoint at the "SetpointInput" can be triggered by a signal edge change.

A signal change at "EnterInput" is only taken into account when "ContinuousEnterFunct" = FALSE. If this condition is fulfilled, an edge change 0 → 1 causes the setpoint entered in "SetpointInput" to be adopted and transferred. This also applies when the newly entered setpoint is identical to the last setpoint transferred.

This method of setpoint adoption is suitable for input via hardware, for example via a console or control panel.

It can also be used for entering setpoints at an operator panel (OP). In this the adoption must be triggered by a function key on the OP.

If you do not require the parameter, simply leave it open.

Name: **ContinuousEnterFunct**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Apply setpoint continuously for software setpoint

With this parameter, you specify whether the setpoint at "SetpointInput" should be continuously read in and checked for changes. Changes are checked by comparison with the last setpoint that was transferred.

This method of setpoint adoption is suitable for input via suitable software. It can, however, also be used for entering setpoints at an operator panel (OP) if the OP does not have a separate function key that can be used to trigger the input.

If you do not require the parameter, simply leave it open.

Name: **SetpointInput**

Declaration: INPUT

Data type: WORD

Range of values: See address range

Default: 0 (W#16#0)

Address range:	Input words	IW0 ... IWn
		PIW0 ... PIWn
	Memory words	MW0 ... MWn
		LW0 ... LWn
	Data words	DBm.DBW0 ... n

Explanation: Setpoint input

How a setpoint available at SetpointInput is processed depends on whether it is a hardware or software input. You specify the type of input with the "ContinuousEnterFunct" parameter:

- **ContinuousEnterFunct = FALSE (hardware input)**

A setpoint at "SetpointInput" is only read in as long as the signal is 1 at "EnterInput". The setpoint that is read in is then transferred if no error is detected during the 1-out-of-n check, and if the central enable memory bit is set. This is automatically set by FC Safe following a selected time delay set there (see FC Safe, "InputDelayTime" parameter).

A further setpoint is then only read in again by the FB when a 0 signal was detected at EnterInput for at least one OB1 cycle.

When a 1-out-of-n error is detected at the hardware input, the entered setpoint is no longer processed. A new setpoint is read in again only when previously for the length of an OB1 cycle in the CPU, no hardware input via a command, setpoint or parameter block was detected.

The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172). As long as the error remains, the error status is indicated via the "InputError" output of FC Safe (see FC Safe, "InputError" parameter).

- **ContinuousEnterFunct = TRUE (software input)**

A setpoint at SetpointInput is read in continuously and checked for changes. Changes are checked by comparison with the last setpoint that was transferred. The setpoint is sent immediately every time a change occurs unless the 1-out-of-n check detects an error.

- With hardware input (see EnterInput) an empty cycle must be detected before a new setpoint can be transferred by the block.

- With software input a new setpoint can be transferred in every OB1 cycle. This assumes that there is no other software entry by another block pending in this cycle. Otherwise a 1-out-of-n error is detected.

A new transfer of the software setpoint can be triggered via the "Send-SoftSetpoint" input even when the software input has not changed (see below).

When a 1-out-of-n error is detected at the software input, the entered setpoint is no longer processed.

A new setpoint is read in again only when previously for the length of an OB1 cycle in the CPU, no software input (command or setpoint) was detected. The block enters detected 1-out-of-n errors in the diagnostics buffer (event ID B172). Appropriate error bits are also set in the central data block BasicData where they can be queried by the software. For further details, refer to the description of FC Safe.

Name: **ReturnedSetpoint**

Declaration: OUTPUT

Data type: WORD

Range of values: See address range

Default: 0 (W#16#0)

Address range:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n
	Explanation:	Output for a returned setpoint The partner object receiving the setpoint reports back the currently valid setpoint there. This value is displayed at the "ReturnedSetpoint" output. If the partner object is set to 'local' and a new input is made there, the setpoint changed locally is indicated here by "ReturnedSetpoint". After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the locally valid setpoint is indicated by "ReturnedSetpoint". If you do not require the parameter, simply leave it open.

Name:	LocalOperation	
Declaration:	OUTPUT	
Data type:	BOOL	
Range of values:	TRUE / FALSE	
Default:	FALSE	
Address range:	Output	Q 0.0 ... Q n.7
	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
	Explanation:	Return message from the partner object: The object is set to local operation. A setpoint can also be set locally at the partner object that receives the setpoint. The partner object Set01W_R must then be set to 'local' at the "Local" input parameter. The current status of the "Local" input parameter is reported by the partner object and indicated here at the "LocalOperation" output. After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the local currently valid status is displayed at "LocalOperation". If you do not require the parameter, simply leave it open.

Name:	NewData
Explanation:	For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207). Whenever the block has received new data and has output it to the output "ReturnedSetpoint" or "LocalOperation", the "NewData" output is set to TRUE for one OB1 cycle.

Name: **SendSoftSetpoint**

Declaration: IN_OUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address Memory M 0.0 ... M n.7

range: Data bit DBm.DBX 0.0 ... n.7

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Explanation: Trigger input for resending the last software setpoint.

For further details, refer to the "SetpointInput" parameter.

If you do not require the parameter, simply leave it open.

3.8.1.31 Setpoint typical Set01W_FS

Function

Send 1 setpoint as a 16 bit value and receive local setpoint from partner

Frames of this block can be received and evaluated with the Set01W_R block.

Sending the setpoint can be triggered by the following parameters:

- ContinuousEnterFunct
Automatic transmission in case of a change in the area "ParameterInput" in the software mode of the block (HWmode = FALSE)
- Release
Single triggered transmission in hardware and software mode of the device

No setpoints are sent to stations which cannot be reached.

Parameters

Parameter: **PartnerNo**

PartnerObjectNo

Enabled

For a description, see section Reoccurring parameters (Page 207).

Parameter: **HWmode**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: The parameter specifies whether the block works in hardware mode (TRUE) or software mode (FALSE).

- **Hardware mode**

Transmission can be triggered by a digital input module connected to the "Release" input.

The setpoint is applied in hardware mode when the signal at the release input changes from 0 to 1.

Changes made to the setpoint input "SetpointInput" while the "InputDelay-Time" is running are not sent.

In hardware mode, an empty cycle must be detected before new parameter values can be sent by the block.

- **Software mode**

Sending can be triggered via the "Release" input or via the "ContinuousEnterFunct" function in software mode.

New parameter values can be sent in each OB1 cycle with a software input. The "ContinuousEnterFunct" function can be switched on and off as required in the cycle.

The "**HWmode**" parameter is applied during startup and cannot be changed in the cycle.

Parameter: **ContinuousEnterFunct**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Apply setpoint continuously for software setpoint

The parameter is not evaluated if HWmode is TRUE.

With this parameter, you specify whether the setpoint at "SetpointInput" should be continuously read in and checked for changes. Changes are checked by comparison with the last setpoint that was transferred. Sending occurs even if the value at "SetpointInput" has not changed since the last transmission.

This method of setpoint adoption is suitable for input via suitable software. It can, however, also be used for entering setpoints at an operator panel (OP) if the OP does not have a separate function key that can be used to trigger the input. If you do not require the parameter, simply leave it open.

Parameter: **SetpointInput**

Declaration: INPUT

Data type: WORD

Range of values: See address range

Default: 0 (W#16#0)

3.8 Master copies

Address range:	Input words	IWO ... IWn PIWO ... PIWn
	Memory words	MWO ... MWn LWO ... LWn
	Data words	DBm.DBWO ... n
	Explanation:	Setpoint input

The way in which the value at "SetpointInput" is subsequently processed depends on whether it is a hardware or a software input and how transmission is triggered. You can find more information on this in the "Function" section above and under the "HWmode" parameter.

Name:	InputDelayTime
Declaration:	INPUT
Data type:	INT
Range of values:	<ul style="list-style-type: none"> • 0 or open Value for unrequired parameter • 1 ... 32000 Value range for delay time (milliseconds) in hardware mode
Explanation:	<p>Delay time in milliseconds for the "Release" input when the block is operating in hardware mode.</p> <p>A delay time of at least 1000 ms is recommended.</p> <p>Changes to "SetpointInput" while "InputDelayTime" is running are not sent in the current cycle.</p>

Name:	MaxInputTime
Declaration:	INPUT
Data type:	INT
Range of values:	<ul style="list-style-type: none"> • 0 or open When the parameter is not required. • 1 ... 32000 Value range for monitoring time (seconds) "HWmode" = TRUE.
Explanation:	<p>Monitoring time in seconds for the "Release" input if the block is operating in hardware mode.</p> <p>A monitoring time of at least 30 s is recommended.</p> <p>If you do not require the parameter, leave it open.</p>

Parameters:	Release
Declaration:	IN_OUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE

Address range:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Explanation: Trigger input for sending the current setpoint

You can use the parameter in software or hardware mode of the block.

- **Hardware mode**
If you use the input via an OR, it should be operated via a function key of the OP. You can then first enter the setpoint at the OP. The value is only be sent when you press the release function key.
In hardware mode, the "Release" input is not automatically reset. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input.
The "Release" input is subject to the "InputDelayTime" and is monitored via "MaxInputTime". If the input is set longer than "MaxInputTime, the timeout is output at "InputError". The "InputError" output does not change back to "0" until the "Release" input is also = 0. (See also "InputError")
- **Software mode**
A change check is performed only with signal 1 at the Release input. Sending occurs even if the value at "SetpointInput" has not changed since the last transmission.
In software mode, the "Release" input is automatically reset. You should therefore only specify memory or data inputs as the input.

If you do not require the parameter, simply leave it open.

Parameter:	ReturnedSetpoint	
Declaration:	OUTPUT	
Data type:	WORD	
Range of values:	See address range	
Default:	0 (W#16#0)	
Address range:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n
Explanation:	Output for a returned setpoint	
	<p>The partner object receiving the setpoint reports back the currently valid setpoint there. This value is displayed at the "ReturnedSetpoint" output.</p> <p>If the partner object is set to 'local' and a new input is made there, the setpoint changed locally is indicated here by "ReturnedSetpoint".</p> <p>After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the locally valid setpoint is indicated by "ReturnedSetpoint".</p> <p>If you do not require the parameter, simply leave it open.</p>	

Parameter: **LocalOperation**

Declaration: OUTPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Address Output

Q 0.0 ... Q n.7

range: Memory bit

M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit

DBm.DBX 0.0 ... n.7

Explanation: Return message from the partner object: The object is set to local operation.

A setpoint can also be set locally at the partner object that receives the setpoint. The partner object Set01W_R must then be set to 'local' at the "Local" input parameter. The current status of the "Local" input parameter is reported by the partner object and indicated here at the "LocalOperation" output.

After startup of the local or partner CPU, or after restoring a connection, an automatic general request ensures that the local currently valid status is displayed at "LocalOperation".

If you do not require the parameter, simply leave it open.

Parameter: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Whenever the block has received new data and has output it to the output "ReturnedSetpoint" or "LocalOperation", the "NewData" output is set to TRUE for one OB1 cycle.

Name: **InputOK**

Declaration: OUTPUT

Data type: BOOL

Range of values: Output

Q 0.0 ... Q n.7

Memory

M 0.0 ... M n.7

L 0.0 ... L n.7

Data bit

DBm.DBX 0.0 ... n.7

Explanation: The display "InputOK" is set for one cycle at the entry of the data frame into the send buffer. The display is independent of the "HWmode" parameter.

If you do not require the parameter, leave it open.

Name: **InputError**

Declaration: OUTPUT

Data type: BOOL

Range of values:	Output	Q 0.0 ... Q n.7
	Memory	M 0.0 ... M n.7
		L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	<p>Display of input errors at the "Release" input, if "HWmode" equals TRUE.</p> <p>The output has a 1 signal as soon as the "Release" command lasts longer than defined via MaxInputTime. The "InputError" output is only set to 0 again when the release input has a 0 signal. This can be used, for example, to detect that the input is defective.</p> <p>If you do not require the parameter, leave it open.</p>	

3.8.1.32 Setpoint typical Set01W_R

Function

Receive 1 setpoint in the station as a 16 bit value or enter locally and return the local setpoint to the master station

The block can receive and output data of the "Set01W_S" and "Set01W_FS" typicals.

Parameters

General parameters:	PartnerNo
	PartnerObjectNo
	Enabled
	ImageMemory
	Conditional
	Unconditional
	TimeStamp
For a description, see section Reoccurring parameters (Page 207).	

Name:	Local
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Default:	FALSE

3.8 Master copies

Address range:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation:	<p>Release of the local setpoint input of the data area specified by "LocalSetpointInput".</p> <p>A setpoint sent by the partner (master station) is not accepted by the object as long as "Local" = TRUE.</p> <p>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).</p> <p>Bumpless switchover:</p> <ul style="list-style-type: none"> When there is a switchover from "Local" = 0 to 1, the last values output at the "SetpointOutput" output are retained until a new setpoint is entered via the local input area "LocalSetpointInput". When there is a switch back from "Local" = 1 to 0, the last value output at the "SetpointOutput" output is retained until the block receives a new setpoint from the partner. <p>Read the note on the "ContinuousEnterFunct" parameter.</p>	

Name: **EnterInput**
Declaration: INPUT
Data type: BOOL
Range of values: TRUE / FALSE

Default: FALSE

Address range:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation: Enter input for local setpoint input.

Edge triggered, the parameter triggers adoption of the setpoint at the setpoint input "LocalSetpointInput".

A signal change at "EnterInput" is only taken into account when the value TRUE is set at the "Local" input parameter and "ContinuousEnterFunct" = FALSE. If these conditions are fulfilled, a signal change from 0 to 1 at the causes the setpoint entered at "LocalSetpointInput" to be adopted and output at "SetpointOutput".

This method of setpoint adoption is suitable for hardware input, for example via a console, control panel or an operator panel (OP). With an OP, the adoption must be triggered by a function key.

If you do not require the parameter, simply leave it open.

Name: **ContinuousEnterFunct**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Default: FALSE

Explanation: Continuous local setpoint acquisition

When "Local" = TRUE and "ContinuousEnterFunct" = TRUE the setpoint at "LocalSetpointInput" is continuously read in and checked for changes.

The setpoint read in at "LocalSetpointInput" is output at the "SetpointOutput" output if at this time the local input value differs from the last returned setpoint.

If an array is detected at the setpoint input "LocalSetpointInput" this is output immediately at "SetpointOutput" (without a change check).

This method of setpoint adoption is suitable for software input.

It can, however, also be used for entering setpoints at an operator panel (OP) if the OP does not have a function key.

If you do not require the parameter, simply leave it open.

Name: **LocalSetpointInput**

Declaration: INPUT

Data type: WORD

Range of values: See address range

Default: 0 (W#16#0)

Address range: Input words

IWO ... IWn

PIWO ... PIWn

Memory words

MWO ... MWn

LWO ... LWn

Data words

DBm.DBWO ... n

Explanation: Local setpoint input

A value at "LocalSetpointInput" is only adopted if "Local" = TRUE. If this condition is met, how a pending setpoint is processed depends on whether it is a hardware or software input. The type of input is decided by the "ContinuousEnterFunct" parameter:

- ContinuousEnterFunct = FALSE
Hardware input
A setpoint at "LocalSetpointInput" is only read in when a signal change 0 → 1 is detected at "EnterInput". The setpoint entered locally is output via the "SetpointOutput" output and transferred to the partner.
A further setpoint is then only read in again by the block when a 0 signal was detected at "EnterInput" for at least one OB1 cycle.
- ContinuousEnterFunct = TRUE
Software input
A setpoint at "LocalSetpointInput" is read in continuously and checked for changes. The change check is achieved by comparison with the last valid setpoint; in other words, the value stored as the returned setpoint. With each change, the setpoint is output via the output set at "SetpointOutput" and transferred to the partner.
With software input a new setpoint can be entered in every OB1 cycle.
With hardware input an empty cycle must be detected before a new setpoint can be read in by the block.

If you do not require the parameter, simply leave it open.

Name: **SetpointOutput**

Declaration: OUTPUT

Data type: WORD

Range of values: See address range

Default: 0 (W#16#0)

Address range:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

Explanation: Setpoint output word.

The setpoint sent by the partner object or entered locally at "LocalSetpointInput" is output at the "SetpointOutput" output.

Name: **NewData**

Explanation: For the declaration, data type, range of values, default and function, refer to the section Reoccurring parameters (Page 207).

Whenever the block has received a new setpoint from the partner object and has output it to "SetpointOutput", the "NewData" output is set to TRUE for one OB1 cycle. This also applies when there is a new local setpoint input (Local = 1).

3.8.2 Optional blocks

3.8.2.1 ListGenerator1500/300/400 FC

The ListGenerator FC is available in three versions with the following symbolic names:

- For S7-300
ListGenerator300
- For S7-400
ListGenerator400
- For S7-1500
ListGenerator1500

Function

The FC ListGenerator is required in a CPU that receives data containing either an incomplete destination address or no destination address at all. The lack of the destination object number is the most important factor here because this points to the instance DB in which the received information should be stored.

Missing or incomplete destination addresses can occur when no or incomplete parameters are set for them in the station. This is permitted for typicals that send binary information, analog values or counted values. If these typicals send data to more than one destination, no destination address is set for them. Due to the missing destination information, the send frame is automatically transmitted to all destinations for which a connection was set up during configuration. This data is therefore received without a destination address at the various destinations.

Note

Supplement to destination subscriber no.

Data frames to be sent without a destination address have the destination subscriber number added by the sending TIM, and sometimes several if there are several destinations.

The TIM enters 0 in the address field for the destination object number, since the TIM does not have the relevant information. The only destination subscribers it knows are those to which it has a configured connection.

At the receiving end, the data frame therefore contains the destination subscriber number but the destination object number is 0.

If the destination object number is not contained in the received data frame, FC Distribute, which is responsible for distributing the received frames, references an object reference list.

Using the source address (source subscriber no. + source object no.) contained in every data frame, FC Distribute searches through the list for an entry that specifies the missing destination object number for the given source address; in other words, it searches for the number of the local instance DB.

This object reference list is created by FC ListGenerator. The FC has no parameters. It is linked into the cyclic user program (OB1) following the FC BasicTask.

3.8 Master copies

When creating the list, FC ListGenerator uses the addresses set in the parameters for the receiving typicals. Specification of "PartnerNo" and "PartnerObjectNo" is mandatory for these typicals. These parameters are identical to the source address in the corresponding receive frame. Since the typical also knows the number of its instance DB, it therefore knows all the addresses required for an entry in the reference list.

During startup, FC ListGenerator arranges that all receiving typicals enter the addresses from their parameter assignment with the number of the instance DB in the reference list. The object reference list therefore does not require special parameter settings, it is simply created from the existing parameters of the receive typicals and is therefore always consistent.

How it works

FC ListGenerator creates the lists after startup in three consecutive OB1 cycles:

- In the first cycle, it determines how many entries will be required in the first and, if applicable, in the second object reference list. The typicals involved only increment a counter during this run.
- In the second cycle, FC ListGenerator generates the data block for the first and, if applicable, the second object reference list with the required length and enters 0 in all the data words. During the same cycle, all typicals involved enter their addresses and the number of the corresponding instance DB in the list.
- In the third and final cycle, FC ListGenerator sorts all the entries in ascending order. Sorting speeds up the search in the list during actual operation.

When generating the data block, FC ListGenerator does the following:

If no list has yet been created, a search is made for a free DB number.

- S7-1500
The FC searches in descending order in the range from 60999 to 60000 (reserved area) for a free DB number.
- S7-300/400
Starting with the number of DB BasicData, the number of the next lowest free DB is taken.

If a list already exists, FC ListGenerator checks to see whether the existing DB is long enough for the currently required number of references. If the length is adequate, 0 is entered as the content and the addresses are written again and sorted.

The object reference list 1 can be used in full length (65532 bytes) for max. 10922 receive objects. The maximum number of objects without a target object number is 10922.

If the existing data block is too short, different procedures are used for the various SIMATIC product series:

- For S7-300
A new DB is generated. The old DB remains in memory because S7-300 has no delete function for data blocks.
With the 300 CPU, you have to delete the old DBs with the programming device.
Note:
Without generating a new DB, you would need to delete this DB with the programming device. If there is not enough memory on the CPU to be able to generate a new DB, you need to delete the existing DB before restarting.
- For S7-400
The existing DB is deleted, the memory is compressed and the DB is then generated with the same number in the new length.
With the 400 CPU, you may be able to manually compress the memory or reload the CPU.
- For S7-1500
The existing DB is deleted and generated in the new length with the same number.
With the 1500 CPU, the memory is automatically compressed, just like with the 300 CPU.

If the ListGenerator FC can no longer generate a DB, an error message is written to the diagnostic buffer of the CPU:

- 0xB107 "Error generating the object reference list"
 - DB[Info1] cannot be created.
 - Cause: [Info2].
In Info2, the return value of the SFC Create_DB function is output, refer to the description there.

3.8.2.2 FC PartnerMonitor

Function

FC PartnerMonitor has the following functions:

- It displays important status information about a SINAUT subscriber (see "PartnerStatus" parameter).
- The FC can also be used to trigger a general request of the subscriber, except to a control center (e.g. ST7cc/ST7sc).
- It can also be used to establish and terminate a permanent connection to the subscriber.

The FC can be called at any point in the cyclic user program (in OB1).

If you want to monitor and control more than one subscriber, include an appropriate number of FC PartnerMonitor in the user program.

A SINAUT subscriber (partner) can only be an ST7 CPU or an ST7cc to which a connection was configured. TIMs cannot be monitored or controlled by FC PartnerMonitor.

Note**FC PartnerMonitor in a station**

FC PartnerMonitor can also be used in a station. However, the control inputs for establishing and terminating a permanent connection can then no longer be used. This only works in the master station when the local TIM is a master station TIM.

Parameters

Name: **PartnerNo**

Declaration: INPUT

Data type: INT

Range of values: 1 ... 32000 [ms]

Explanation SINAUT subscriber no. of the subscriber to be monitored and controlled.

If the set "PartnerNo" is not found in the administration (DB BasicData), then (only during startup) an entry is written to the diagnostics buffer (event ID B101). The CPU does not change to STOP.

The status of a subscriber with correct parameter settings is indicated in the "PartnerStatus" output word and the control inputs are processed.

An unknown subscriber is not processed until the error is eliminated. The "PartnerStatus" output word remains set to 0 during this time.

Name: **MaxConnectTime**

Declaration: INPUT

Data type: INT

Range of values:

- 0 (no limitation)
- 1 ... 480 [minutes]

Explanation Maximum duration of a permanent connection

The time specified here (> 0) is activated at the start of a permanent connection (see "PermanentCall_On" parameter).

If the time elapses before the permanent connection is reset, it is automatically disconnected. The time is retriggered as long as the signal 1 is present at the "PermanentCall_On" input.

The time specified here applies to a permanent connection in a dial-up network as well as to a permanent connection (continuous polling) on a dedicated line.

Name: **PartnerStatus**

Declaration: OUTPUT

Data type: WORD

Range of values:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n
Explanation	Output word to indicate the status of the subscriber to be monitored. If you do not require the parameter, simply leave it open.	

The meaning of the status bits in the "PartnerStatus" output word:

Bit	.0	Status of the subscriber
	0	0 = Subscriber disrupted
	1	1 = Subscriber OK

Bit	.1	Status of the redundant connection
	0	0 = Redundant connection is disrupted
	1	1 = All connections OK.

Bit	.3	.2	Status of the general request (GR)
	0	0	0 = GR complete without error
	0	1	1 = GR started
	1	0	2 = GR start received
	1	1	3 = GR finished with error (GR incomplete or cannot be executed, e.g. due to fault at subscriber)

Bit	.6	.5	.4	Status of the dial-up connection
	0	0	0	0 = No connection
	0	0	1	1 = Outgoing call activated
	0	1	0	2 = Incoming call established
	0	1	1	3 = Outgoing call established
	1	0	0	4 = Permanent connection registered
	1	0	1	5 = Permanent connection established
	1	1	0	6 = Permanent connection disconnected

Bit	.7	Status of the dial-up connection
	0	0 = No dial-up connection check in background
	1	1 = Dial-up connection check in background is activated

Bit	.8	Status of continuous polling (on dedicated line)
	0	0 = No continuous polling
	1	1 = Continuous polling activated

Bit	.9	Status of the WAN connection resources
*)	0	0 = Sufficient resources on partner
	1	1 = Insufficient resources on partner

Bit	.10	Time status
	0	0 = Date/time not available / not OK on partner
	1	1 = Date/time OK on partner

Bit	.11	Time-of-day synchronization
	0	0 = The partner CPU received a plausible time during the last synchronization or no time of day has been received since startup.
	1	1 = The partner CPU has received an implausible time of day; the last valid time will continue to be used. (Display only with TimeMask > V1.6)

Note**In-out parameters**

The following parameters are in/out parameters (declaration IN_OUT):

- GeneralRequest
- PermanentCall_On
- PermanentCall_Off

It is difficult to specify local bit memory with this parameter type and this should not be used.

Name: **GeneralRequest**

Declaration: IN_OUT

Data type: BOOL

Range of values: Input I 0.0 ... I n.7
Memory M 0.0 ... M n.7
Data bit DBm.DBX 0.0 ... n.7

Explanation Input for triggering a general request to the subscriber specified with PartnerNo. A general request to the subscriber is triggered with a 1 signal at this input if no request is active for this subscriber at this time. The input is then automatically reset by the FC.

If an input of a digital input is specified (I 0.0 ... I n.7), you are responsible for resetting the signal at the input. Reset the signal before ending the currently running general request so that another general request is not triggered immediately.

Name: **PermanentCall_On**

Declaration: IN_OUT

Data type:	BOOL	
Range of values:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
Explanation	<p>Input for triggering a permanent connection to the subscriber specified with PartnerNo.</p> <p>A permanent connection to the subscriber is triggered with a 1 signal at this input if there is currently no permanent connection to this subscriber. The input is then automatically reset by the FC. If an input of a digital input is specified (I 0.0 ... I n.7), you are responsible for resetting the signal at the input at the latest before the termination of the existing permanent connection.</p> <p>A 1 signal at the input "PermanentCall_On" also activates the time specified with "MaxConnectTime" if it is greater than 0.</p> <p>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:</p> <ul style="list-style-type: none"> • For a dial-up connection: A dial-up connection is established by the master TIM to the appropriate subscriber and, regardless of the data traffic, maintained until the terminate command is sent. The current status of the permanent connection is indicated in the PartnerStatus output word with the bits 4 ... 6 (see PartnerStatus parameter). • For a dedicated line: In this case the master TIM operates in polling mode with the stations. A permanent connection is implemented in this case by 'continuous polling' of the subscriber. This is actually an intermittent poll to the subscriber; in other words, the other subscribers on the dedicated line network are still polled but the preferred subscriber is polled again after every poll to a 'normal' subscriber. The current status of the continuous polling is indicated by bit 8 in the "PartnerStatus" output parameter. <p>Special feature with stations: A permanent connection cannot be established from a station. This control input cannot therefore be used when FC PartnerMonitor is used in a station.</p>	

Name:	PermanentCall_Off	
Declaration:	IN_OUT	
Data type:	BOOL	
Range of values:	Input	I 0.0 ... I n.7
	Memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

Explanation Terminating an existing permanent connection

The input serves to trigger termination of an existing permanent connection to the subscriber specified with "PartnerNo".

A permanent connection to the subscriber is terminated with a 1 signal at this input if there is currently a permanent connection to this subscriber. The input is then automatically reset by the FC. If an input of a digital input is specified (I 0.0 ... I n.7), you are responsible for resetting the signal at the input via the user program. This should be done at the latest before establishing a permanent connection again.

Depending on whether the subscriber can be reached over a dial-up connection or a dedicated line, the command to terminate the permanent connection is processed as follows and indicated at the "PartnerStatus" output:

- **For a dial-up connection:**

The existing dial-up connection is terminated by the master TIM but only after any pending data has been sent.

The current status of the permanent connection is indicated in the PartnerStatus output word with the bits 4 ... 6 (see PartnerStatus parameter).

- **For a dedicated line:**

The master TIM deletes the registration for continuous polling of the corresponding subscriber. The polling cycle for all connected subscribers continues in normal mode.

The current status of the continuous polling is indicated by bit 8 in the "PartnerStatus" output parameter.

Continuous polling can also be canceled on a dedicated line by instructing the master TIM to start continuous polling of another subscriber. The existing job is then replaced by the new one.

Special feature with stations:

A permanent connection cannot be terminated by a station. This control input cannot therefore be used when FC PartnerMonitor is used in a station.

3.8.2.3 FC PartnerStatus

Function

The FC PartnerStatus can show the current status 'disrupted' or 'OK' for a maximum of 8 SINAUT subscribers.

The FC can be called at any point in the cyclic user program (in OB1).

If you want to monitor more than 8 subscribers, include the appropriate number of FCs PartnerStatus in the user program.

The partner can be an ST7 CPU or an ST7cc to which a connection was configured, or a local TIM.

One bit per subscriber is reserved in the "PartnerStatus" output byte to indicate the status of the respective subscriber:

- FALSE (0):
 - Subscriber disrupted
 - Corresponding input parameter not used (= 0)
 - Subscriber unknown
- TRUE (1): Subscriber OK

Parameters

Name: Partner1 ... Partner8

Declaration: INPUT

Data type: INT

Range of values:

- 0
Value for unrequired parameter
- 1 ... 32000
Number of the subscriber to be monitored

Explanation SINAUT subscriber number of the subscriber to be monitored

If a set subscriber number is not found in the administration (DB BasicData), then (only during startup) an entry is written to the diagnostics buffer (event ID B101). The CPU does not change to STOP.

The status of a subscriber with a correct parameter assignment is indicated in the "PartnerStatus" output byte.

An unknown subscriber is not processed until the error is eliminated. Their status bits are set to 0.

Name: PartnerStatus

Declaration: OUTPUT

Data type: BYTE

Range of values:

Output bytes	QB0 ... QBn
	PQB0 ... PQBn
Memory bytes	MB0 ... MBn
	LB0 ... LBn
Data bytes	DBm.DBB0 ... n

Explanation: Indication of the status of the subscriber to be monitored.

Assignment of the status bits in the "PartnerStatus" output byte depending the parameters Partner1 ... "Partner8":

Bit	.7	.6	.5	.4	.3	.2	.1	.0
Partner	8	7	6	5	4	3	2	1

Value:

- 0 = partner disrupted, not set in the parameters or unknown
- 1 = partner OK

3.8.2.4 FC PathStatus

Function

The block (FC) shows the status of the path to a partner from the perspective of the local TIM. A maximum of 2 paths (main and substitute path) to a partner can be configured. Both paths must begin or end on a local TIM.

The block shows the following:

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

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

In the output byte PathStatus the following bits are reserved:

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

The FC can be called at any point in the cyclic user program (OB1) after calling the FC BasicTask.

If the status of the path to more than one subscriber is to be shown, a corresponding number of FC calls need to be programmed in the user program program.

Parameters

Name: **Partner**
Declaration: INPUT
Data type: INT
Range of values: 1 ... 32000 (subscriber number)
Explanation: SINAUT subscriber no. of the partner

Name: **PathStatus**
Declaration: OUTPUT
Data type: BYTE

Range of values:	Output bytes	QB0 ... QBn PQB0 ... PQBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n
Explanation:	Display of the path status of the connection to the partner	

PathStatus - Coding

Table 3-10 Coding of the status bits in the "PathStatus" output byte

Bits 6 + 7	Bits 4 + 5	Bits 2 + 3	Bits 0 + 1
Configured interface		Path status	
No. for substitute path	No. for main path	Substitute path (2nd path)	Main path (1st path)

Configured interface (bits 4..7)

The TIM interfaces in the block are consecutively numbered decimally from 0 to 3, see "No." column in the table. The table shows the coding of the bits for the different TIM types.

Table 3-11 Coding of bits 4 + 5 (interface of the main path) or bits 6 + 7 (interface of the substitute path)

Status bit 5 (7)	Status bit 4 (6)	Interfaces			
		No.	TIM 3V-IE	TIM 4R-IE	TIM 1531 IRC
0	0	0	Ethernet "IE1" (X2)	Ethernet "IE1" (X3)	Ethernet "IE1" (X1)
0	1	1	-	Ethernet "IE2" (X4)	Ethernet "IE2" (X2)
1	0	2	Serial "WAN1" (X1)	Serial "WAN1" (X1)	Ethernet "IE3" (X3)
1	1	3	-	Serial "WAN2" (X2)	Serial "WAN" (X4)

Path status (bits 0..3)

- Main path = 1. Path (bits 0 + 1)
- Substitute path = 2nd path (bits 2 + 3)

Table 3-12 Status table: Coding of bits 0 + 1 or bits 2 + 3

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

Coding options - TIM 1531 IRC

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

Table 3-13 Coding options for the output byte "PathStatus"

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

Coding options - TIM 4R-IE

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

Table 3-14 Coding options for the output byte "PathStatus"

Configured interface		Path status	
No. for substitute path	No. for main path	Substitute path (2nd path)	Main path (1st path)
0 0	0 0 (Coding for IE1)	Irrelevant (not redundant)	Status IE1
0 0	0 1 (Coding for IE2)	Status IE1	Status IE2
0 0	1 0 (Coding for WAN1)	Status IE1	Status WAN1
0 0	1 1 (Coding for WAN2)	Status IE1	Status WAN2

0 1	0 0	Status IE2	Status IE1
0 1	0 1	Irrelevant (not redundant)	Status IE2
0 1	1 0	Status IE2	Status WAN1
0 1	1 1	Status IE2	Status WAN2
1 0	0 0	Status WAN1	Status IE1
1 0	0 1	Status WAN1	Status IE2
1 0	1 0	Irrelevant (not redundant)	Status WAN1
1 0	1 1	Status WAN1	Status WAN2
1 1	0 0	Status WAN2	Status IE1
1 1	0 1	Status WAN2	Status IE2
1 1	1 0	Status WAN2	Status WAN1
1 1	1 1	Irrelevant (not redundant)	Status WAN2

3.8.2.5 FC PulseCounter

Function

The FC PulseCounter is responsible for count pulse acquisition.

A maximum of 8 pulse strings are detected via digital inputs and fed to the function blocks with the aid of SIMATIC counters that put together the counted value frames (Cnt01D_S, Cnt04D_S).

The acquisition of the count pulses is time-controlled. To do this the FC PulseCounter must be included in a cyclic interrupt OB, e.g. OB35. The call interval of the cyclic interrupt OB must be matched to the pulse duration of the count pulses. You will find more information on count pulse acquisition with the cyclic interrupt OB in the section Cyclic interrupt OB (Page 201).

Parameters

Name: **InByte**
Declaration: INPUT
Data type: BYTE

3.8 Master copies

Range of values: Input bytes PEB0 ... PEBn
Memory bytes MB0 ... MBn
LB0 ... LBn
Data bytes DBm.DBBO ... n

If an input byte of a digital input is specified, this must be the address of the I/O byte (PIB) directly from the digital input modules. The current status of the count input can only be detected reliably by direct access.

When reading out from the process image of the inputs (PII) count pulses could remain undetected.

Explanation: Input byte for count pulses.
The parameters for inputs for count pulse acquisition can be set in bytes.

Name: **EnableMask**

Declaration: INPUT

Data type: BYTE

Range of values: B#16#00 ... B#16#FF

Explanation Enable mask for the counting inputs

With this parameter, it is possible to specify in the form of a bit mask at which inputs in the input byte count pulses are actually connected. The following applies to every bit in the bit mask:

- 0 = Input bit blocked for acquisition
- 1 = Input bit enabled for acquisition

The input of the mask is only permitted in hexadecimal format B#16#00 to B#16#FF.

Input as an 8-bit binary number from 2#0 to 2#1111 1111 is not possible with the data type BYTE.

The assignment of the bits in the mask to the inputs in the input byte "InByte":

InByte	.7	.6	.5	.4	.3	.2	.1	.0
EnableMask B#16#	0 ... F						0 ... F	

Example: EnableMask B#16#83
Enabled are: Inputs .7, .1 and .0
Blocked are: Inputs .6 to .2

Name: **CntIn_0 ... CntIn_7**

Declaration: INPUT

Data type: COUNTER

Range of values: C0 or C1 ... Cn
(n depends on the CPU)

Explanation	<p>Pulse counter</p> <p>For each of the enabled counting inputs a SIMATIC counter needs to be specified for the corresponding parameter CntIn_0 ... "CntIn_7". With each pulse acquired, the SIMATIC counter is incremented.</p> <p>The counters set here must be specified as input counters (parameter Counter_1 ... _4) in the actual counted value function blocks "Cnt01D_S" and "Cnt04D_S". These function blocks out read the assigned counter and then reset it.</p> <p>As the placeholder for parameters that are not required, it is recommended to specify the counter C0.</p> <p>Example of the parameter assignment of "CntIn_0" ... "CntIn_7" starting at "EnableMask" = B#16#83:</p> <pre> CntIn_0 := Z10 CntIn_1 := Z11 CntIn_2 := Z0 CntIn_3 := Z0 CntIn_4 := Z0 CntIn_5 := Z0 CntIn_6 := Z0 CntIn_7 := Z12 </pre>
-------------	---

3.8.2.6 FC ST7ObjectTest

Validity

S7-300/400

Function

Calling FC ST7ObjectTest in programming error OB121 prevents a CPU STOP, if the CPU receives data with an unknown destination object no.

FC ST7ObjectTest checks why OB121 was called, i.e. which block type is missing.

- If the missing block is a data block and this data block is an instance DB of a SINAUT object, the CPU does not change to STOP.
- If no SINAUT instance DB is missing but rather another block (DB, FB, FC), you can specify the reaction with the parameter "StopInOtherCases".
 - CPU changes to STOP.
 - CPU continues to run.

For more information on the programming error OB121 and background information relating to the use of FC ST7ObjectTest, see section Programming error OB (Page 201).

Parameters

Name:	StopInOtherCases
Declaration:	INPUT
Data type:	BOOL
Range of values:	TRUE / FALSE
Explanation:	<p>The CPU should change to STOP in other error situations. For details of the parameter see above, section "Function".</p> <ul style="list-style-type: none">• TRUE: CPU changes to STOP.• FALSE: CPU continues to run.

3.8.2.7 FC TestCopy

Function

Using FC TestCopy, extracts of the data traffic between ST7 subscribers can be recorded or the entire traffic can be recorded. With search masks to be set in the control field of DB TestCopyData individual frame types can be filtered out and then copied from the send or receive buffer for further evaluation in the DB TestCopyData. For details, see below.

Send and receive frames are all stored in the same data block DB TestCopyData. This makes it simple to track the chronological order of the copied send and receive frames.

Requirements

To use the TestCopy function, the user program must meet the following conditions:

- The FC TestCopy function must be available on the CPU.
- DB TestCopyData must be present on the CPU and have a sufficient length.
 - To achieve this, copy DB TestCopyData (DB99) from the TD7 library to your CPU.
 - If necessary, change the length of the buffer area in the DB by increasing or decreasing the size of the array "TestCopyBuffer" in DW40, which has the default length of [0..240] WORD.
- Make the following entries in the respective communication DB of the CPU 300/400 (BComData / XComData / PComData) whose send and/or receive frames you want to write:
 - In DW32 (TestCopyDBNo) of the communication DB, enter the number of the DB TestCopyData.
 - In DW34 (TestCopyFCNo) of the communication DB, enter the number of the FC TestCopy.

Proceed in the same way for the communication DB of CPU 1500. The tags in DB BConnectData have the same names.

Linking FC TestCopy into the user program

If the above-mentioned conditions are fulfilled, the test function is processed cyclically by the respective communication FB of the CPU.

FC TestCopy cannot be called in the user program.

Monitoring the written data

Use the ready-made watch table "TestCopyMonitor", which you can copy from the TD7 library into the "Watch and force tables" directory of the CPU.

If the settings are to be retained even after CPU startup, they can also be stored directly in the start values of the individual BConnection instances of the DB BConnectData.

3.8.2.8 DB TestCopyData

Structure of the TestCopyData DB

Areas of the TestCopyData DB

The DB for the TestCopy function is divided into the following areas (after Offset in the DB):

- **0 ... 27: User interface**
Interface for setting the TestCopy mode and functions. The area is divided into:
 - 1 ... 13
Filter settings for RecvCopy function and number of counted received frames
 - 15 ... 25
Filter settings for SendCopy function and number of counted sent frames
- **28: Error display**
- **31 ... 39: Internal management pointer**
- **40 ... (Default: 523): Buffer range**
Buffer area for storing frames that match the filter criteria.
The buffer area must be configured as an array [0...xxxx] of WORD.

The following table shows the structure of the DB TestCopyData:

Data type / (off-set)		Tag name	Format	Explanation
User interface				
DBB	0	OperationMode	BYTE	Mode
DBW	12	Recv_TgramCounter	INT	Number of copied receive frames
DBW	26	Send_TgramCounter	INT	Number of copied send frames
RecvCopy function				
DBB	1	Recv_TgrmType	BYTE	Receive filter: Message type (MT)
DBW	2	Recv_DestSubscr	INT	Receive filter: Destination subscriber no.
DBW	4	Recv_DestObject	INT	Receive filter: Destination object no.
DBW	6	Recv_SourceSubscriber	INT	Receive filter: Source subscriber no.
DBW	8	Recv_SourceObject	INT	Receive filter: Source object no.
DBW	10	Recv_StartIndex	INT	Receive filter: Start index no.
DBB	14	SpareDBB14	BYTE	Reserve
SendCopy function				
DBB	15	Send_TgrmType	BYTE	Send filter: Message type (MT)

3.8 Master copies

Data type / (off-set)		Tag name	Format	Explanation
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.
Error display				
DBB	28	FC_RetVal	BYTE	Error information: 0 = No error 1 = DB TestCopyData too short 10 = Unknown mode
DBB	29	SpareDBB29	BYTE	Reserve
DBB	30	SpareDBB30	BYTE	Reserve
Internal management pointer				
DBB	31	TestCopyStatus	BYTE	Status byte for TestCopy operation
DBB	32	TestCopyCmdByte	BYTE	Command byte for TestCopy operation
DBB	33	TestCopyDelCount	BYTE	Loop counter for TestCopy delete function
DBW	34	NextFreeCopyByte	INT	Address of the next free TestCopy-Buffer byte
DBD	36	StartTimeSFC64	DINT	SFC64 time at the start of the copy procedure
Buffer range				
DBB	40	TestCopyBuffer[0]	BYTE	Copy area, byte 0
DBB	41	TestCopyBuffer[1]	BYTE	Copy area, byte 1
DBB	42	TestCopyBuffer[2]	BYTE	Copy area, byte 2
DBB	43	TestCopyBuffer[3]	BYTE	Copy area, byte 3
DBB	n	TestCopyBuffer[n]	BYTE	Copy area, byte n

Structure of a copied message block

A frame block can contain several frames. The frames are saved in the DB TestCopyData according to the following rules:

1. The first entry indicates the time difference in milliseconds (7 decade BCD plus sign) since the last selection of an operating mode > 0.
2. This is followed by a separation sign AAAA for sent messages, EEEE for received messages.
3. Storage of the first message from the frame block.
4. Separation identifier AAAA, or EEEE:
5. Storage of the last frame from the message block.
6. Block end identifier FFFF.

Example

- All received frames will be stored in DB TestCopyData.
- Communication via X blocks, i.e. max. 76 bytes per receive block.
- The receive buffer of the DB XComData is the source for FC TestCopy.
- The current receive block contains 3 messages.

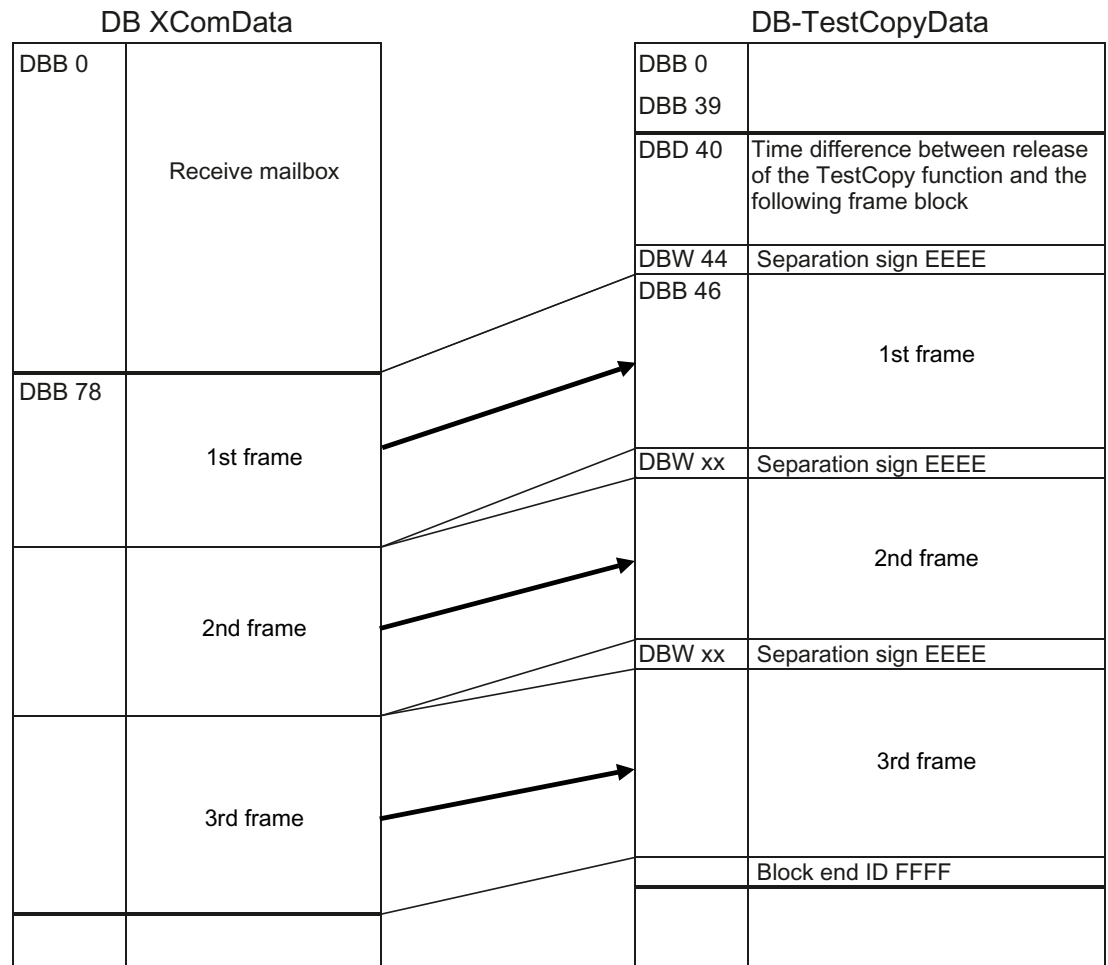


Figure 3-1 Example of filling the DB TestCopyData

Length calculation

FC TestCopy uses the following parameters for determining the minimum length for the DB TestCopyData:

Parameter	Parameter name	Length
Length of communication buffer	Len _{ComBuffer}	76 or 202 bytes *
Minimum frame length	Len _{MinTgrm}	14 bytes
Offset management area	Offset	40 bytes

3.8 Master copies

Parameter	Parameter name	Length
Length of the time difference	Len _{dt}	4 bytes
Length of the block separators	Len _{Trenner}	2 bytes

* 76 with X communication, 202 with B communication

The formula used for the actual calculation is the same for X communication and B communication. The results differ only due to different lengths for the communication buffer for X and B communication:

- Length with X communication**

Len _{Min_Xcom}	=Len _{ComBuffer} + Offset + Len _{dt} + (Len _{ComBuffer} / Len _{MinTgrm} + 1) * Len _{Trenner}
	= 76 + 40 + 4 + (76 / 14 + 1) * 2
	= 120 + 12 = 132 bytes minimum

- Length with B communication**

Len _{Min_Bcom}	=Len _{ComBuffer} + Offset + Len _{dt} + (Len _{ComBuffer} / Len _{MinTgrm} + 1) * Len _{Trenner}
	= 202 + 40 + 4 + (202 / 14 + 1) * 2
	= 236 + 40 = 276 bytes minimum

If FC TestCopy determines that the DB TestCopyData does not have the calculated minimum length an error message to this effect is entered in data byte DBB28.

3.8.2.9 Operation of TestCopyMonitor

Operating modes and transfer directions of DB TestCopyData

In byte 0 (OperationMode) of the DB TestCopyData, the mode of the FC is coded. The values 0 ... 3 identify the mode:

- Mode 0
Function blocked
- Mode 1
Frame entry as of the start of DB TestCopyData
- Mode 2
Write to DB TestCopyData endlessly as circulating buffer
- Mode 3
Fill DB TestCopyData once, then set mode 0.

You will find the detailed coding in the table below.

Along with the mode the coding of DBB0 also includes the transfer direction of the logged data. The following assignment applies for the coding:

- Bit 0...3
Modes (0, 1, 2, 3) for the direction "RecvCopy" (copy receive frames)
- Bit 4...7
Modes (0, 1, 2, 3) for the direction "SendCopy" (copy send frames)

Bit	.7	.6	.5	.4	.3	.2	.1	.0
Direction	SendCopy				RecvCopy			

The functions "SendCopy" and "RecvCopy" can be activated individually or at the same time, however only a common mode is possible for both communication directions.

When a mode > 0 is set in the less significant half byte, this always applies to both communication directions.

Only if mode 0 is coded in bit 0...3 (RecvCopy), does the value in bit 4...7 apply (SendCopy).

Exception:

To delete the DB TestCopyData, DBB0 must be written with FF, 0F is not enough.

Examples of coding of DBB0:

- 00_h
No TestCopy function enabled
- 03_h
Only RecvCopy function (OperationMode = 3), no SendCopy function.
- 30_h
Only SendCopy function (OperationMode = 3), no RecvCopy function.
- 33_h
RecvCopy function and SendCopy function (OperationMode = 3)
- FF_h
Delete content of DB TestCopyData

Filter settings via TestCopyMonitor

Run FC TestCopy via a watch table which is available in finished form as "TestCopyMonitor" in the TD7 library.

The following settings are possible using the watch table in the DB TestCopy:

Name (symbol)	Permitted values (hex.)	Meaning
OperationMode	00	Mode 0: Function blocked
	11	Mode 1: Frame entry always as of the start of DB TestCopyData
	22	Mode 2: Write to DB TestCopyData endlessly as circulating buffer
	02 20	<ul style="list-style-type: none"> • Write for SendCopy function • Write for RecvCopy function
	33	Mode 3: Fill DB TestCopyData once, then set mode 0.
	03 30	<ul style="list-style-type: none"> • Fill for SendCopy function • Fill for RecvCopy function
	FF	Delete the entire DB TestCopyData and reset to default

3.8 Master copies

As the output value, FC TestCopy returns a count value in DBW12 (Recv_TgramCounter) of the DB TestCopyData for the frames received since setting mode 1, 2 or 3 that match the filter criteria.

The number of send frames is entered in DBW26 (Sendcv_TgramCounter).

In DBB28, you receive a return value that indicates the errors that occurred during processing of the FC:

- FC_RetVal = 0
No error
- FC_RetVal = 1
The specified DB TestCopyData is too short.
- FC_RetVal = 10d
The mode entered in DBB0 is not defined.

In addition, the following special filters are available for the RecvCopy or SendCopy function:

Name (symbol) *	Permitted values (hex.)	Meaning
Xxxx_TgrmType	FF	Copy all frame types (TA) to DB TestCopyData
	00	Copy only spontaneous Org. frames (TA = 0)
	11	Copy only queried Org. frames (TA = 1).
	22	Copy only spontaneous data frames (TA = 2).
	33	Copy only queried data frames (TA = 3).
	01	Copy Org. frames with TA = 0 or 1
	23	Copy data frames with TA = 2 or 3
	Any combination	Copy any 0, 1, 2, 3 combinations.
Xxxx_DestSubscr	All permitted subscribers	Filter for the dest. subscriber no. in the frame
	-1	Copy all frames regardless of the dest. subscriber no.
Xxxx_DestObject	All permitted objects	Filter for the dest. object no. in the data frame.
	-1	Copy all frames regardless of the dest. object no.
Xxxx_SourceSubscr	All permitted subscribers	Filter for the source subscriber no. in the frame
	-1	Copy all frames regardless of the source subscriber no.
Xxxx_SourceObject	All permitted objects	Filter for the source object no. in the data frame
	-1	Copy all data frames regardless of the source object no.
Xxxx_StartIndex	All permitted indexes	Filter for the start index no. in the data frame
	-1	Copy all data frames regardless of the start index no.

* "Xxxx": Placeholder for "Recv" or "Send"

The desired filter tags can be easily added using the tag names of the TestCopy DB.

Notes on operator input

When changing from one mode to the next, the content of DB TestCopyData is not deleted. Only internal pointers and frame counters in the management area of the DB TestCopyData

are reset. When there is a mode change it is therefore recommended to use the delete function "FF" to preset the frame buffer area with 0. This makes the copied frame blocks easier to read.

If send and receive frames are to be copied, in the left half byte of the "OperationMode" parameter, the same mode must be entered as in the right half byte.

The following scheme applies to mode 0, 1, 2 and 3:

- If the buffer is deleted, FF_h must always be entered.
- Separate deletion of the receive and send frames is not possible.

3.8.2.10 FC TimeTask

Function

The FC TimeTask keeps a continuous date and time on a CPU. The FC has no parameters.

Link the FC into the cyclic user program (in OB1) following the FC BasicTask.

FC TimeTask can only be used when the CPU is synchronized by a local TIM. Activate the time-of-day synchronization for the relevant TIM modules.

After the CPU has started up, the TIM supplies the date and current time the first time with an Org. frame. After this, time-of-day synchronization is performed at the interval specified in the configuration of the TIM. For time-of-day synchronization on MPI/partline an interval of one minute is recommended. FC TimeTask sets the clock of the CPU with the time provided by the TIM.

The FC reads out the time in every OB1 cycle. The read time is entered in the first two double words of the DB BasicData and marked as valid or invalid and with an indication whether it is daylight saving or standard time.

From DB BasicData all blocks take the current time if they need it. For example the data point typically do this to time stamp their data or FC Trigger to check whether a time set for the FC has been reached or a preset interval has elapsed. This time of day is also available to the user program.

Table 3-15 Assignment of the data words with date, time and time status

CurrentDate	Data byte 0	Year * 10	Year * 1
	Data byte 1	Month * 10	Month * 1
	Data byte 2	Day * 10	Day * 1
	Data byte 3	Hour * 10	Hour * 1
CurrentTime	Data byte 4	Minute * 10	Minute * 1
	Data byte 5	Second * 10	Second * 1
	Data byte 6	Milliseconds * 100	Milliseconds * 10
	Data byte 7	Milliseconds * 1	Time status

Table 3-16 Assignment of the half byte "time status"

0	0 = date/time invalid 1 = date/time valid
1	0 = standard time 1 = daylight saving time
2	(not used)
3	(not used)

Apart from the time status, whether or not the date/time is valid can also be determined based on data bit 16.1 "CpuClockOk". As soon as the time on the CPU is valid, this bit is set to 1 by FC TimeTask. In the user program this bit can be queried directly under the symbolic name "BasicData.CpuClockOk".

3.8.2.11 FC Trigger

Function

The FC sets an output (memory bit, data bit or digital output) at a time that can be configured by the user or at a preset time interval.

The FC resets this output after one OB1 cycle.

The FC can be called at any point in the cyclic user program (OB1) also more than once.

If the running of a program section or a software function is to be triggered using FC Trigger we recommend that FC Trigger is called directly before execution of this function. Applications for triggering functions due to the memory bit set by FC Trigger are, for example, as follows:

- Running through a function
- Calling a block
- Triggering transfer of a counted value every 2 hours

If several functions need to be activated at the same time this can be implemented by one FC Trigger block if all functions query the same memory bit set by the FC. This works, however, only when the triggered functions do not reset this memory bit themselves.

Remedy if triggered blocks reset the memory bit:

- You call FC Trigger often, always with the same time but a different output memory bit.
- After calling FC Trigger you reproduce the set output memory bit in an appropriate number of further memory bits.

The FC accesses the SINAUT time of day in the first two data double words of the DB BasicData. These are constantly supplied if an FC TimeTask is included in the user program and this is synchronized at regular intervals by a local TIM. FC Trigger compares the time set for it with the current time only when the time in DB BasicData, data byte 7 (time status byte, bit 0 = 1) is marked as being valid.

The accuracy with which FC Trigger operates depends on the accuracy of the time and on the OB1 cycle time.

If the OB1 cycle time is lower than 1 second (this is normal) the output is set exactly at the programmed second value with the inaccuracy of the OB1 cycle time of less than 1 s.

If the OB1 cycle time is higher than 1 seconds, the FC works with a tolerance of 4 seconds. If the FC is processed too late but still within 4 seconds of the configured time, the output is still set.

The edge memory bit "Flag" configured for the FC is set at the same time as the output and reset 5 seconds after the configured time.

No placeholder parameter may be used for the edge memory bit and it must not be reset by the user program.

You can find examples of the parameter assignment for FC Trigger further below.

Parameters

Name: **IntervalMode**

Declaration: INPUT

Data type: BOOL

Range of values: TRUE / FALSE

Explanation Point in time / time interval

- FALSE = point in time
- TRUE = time interval

You will find examples of the parameter assignment for a time or time interval after this explanation of the parameters.

Name: **Hour_Minute**

Declaration: INPUT

Data type: WORD

Explanation Specifies the values for hours and minutes.

Further explanation: Refer to the parameter "Month_Year".

Name: **Second_Day**

Declaration: INPUT

Data type: WORD

Explanation Specifies the values for seconds and day.

Further explanation: Refer to the parameter "Month_Year".

Name: **Month_Year**

Declaration: INPUT

Data type: WORD

Range of values: • 00 ... 99
• FF

Explanation Specifies the values for month and year

Each parameter is in two parts. Each of the two values per parameter is specified with two digits as a BCD-coded value.

- The first two digits specify the value for hours, seconds or month.
- The two other digits specify the value for minutes, day or year.

For parts of the parameter that are not needed FF is entered.

Which parameters are permitted depends on the particular parameter and on the "IntervalMode" parameter. You will find further information following the explanation of the parameters.

Name: **TriggerOutput**

Declaration: OUTPUT

Data type: BOOL

Range of values: Output Q 0.0 ... I n.7
Memory M 0.0 ... M n.7
L 0.0 ... L n.7
Data bit DBm.DBX 0.0 ... n.7

Explanation Trigger output

The output is set for the duration of one OB1 cycle when the time or time interval set for "Hour" to "Year" is reached.

Name: **Flag**

Declaration: IN_OUT

Data type: BOOL

Range of values: Memory M 0.0 ... M n.7
Data bit DBm.DBX 0.0 ... n.7

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter and this should not be used.

Explanation Edge memory bit for the "TriggerOutput" output.

No placeholder memory bit may be specified!

The edge memory bit must not be reset by the user program.

Examples of the parameter assignment of a time or time interval

IntervalMode = FALSE (or 0)

The FC operates according the time principle. When the time set is reached, the output "TriggerOutput" is set for one OB1 cycle.

All time parameters can be used to specify the point in time. Parameters not required should be assigned FF. They are ignored during the check for reaching the time.

Even when "IntervalMode" = 0, time intervals can be set in certain ranges, see the following examples.

Permitted values for the time parameters:

Hours	00-23	Day	01-31
Minutes	00-59	Month	01-12
Seconds	00-59	Year	00-99

Examples:

- IntervalMode = FALSE
The output "TriggerOutput" is set once on 04.02.91 at 06:45:12:
 - Hour_Minute : W#16#0645
 - Second_Day : W#16#1204
 - Month_Year : W#16#0291
- IntervalMode = FALSE
The output "TriggerOutput" is set every day at 06:00:00:
 - Hour_Minute : W#16#0600
 - Second_Day : W#16#00FF
 - Month_Year : W#16#FFFF
- IntervalMode = FALSE
The output "TriggerOutput" is set on the 1st of every month at 06:00:00:
 - Hour_Minute : W#16#0600
 - Second_Day : W#16#0001
 - Month_Year : W#16#FFFF
- IntervalMode = FALSE
The output "TriggerOutput" is set every year on October 1, at 06:00:00:
 - Hour_Minute : W#16#0600
 - Second_Day : W#16#0001
 - Month_Year : W#16#10FF

IntervalMode = TRUE (or 1)

The FC operates according the time interval principle. When the time value set or a multiple of it is reached, the output "TriggerOutput" is set for one OB1 cycle.

Only the specifications for hours, minutes and seconds are relevant. The date parameters are ignored. A time interval can also only be set in hours or in minutes or in seconds. Time parameters not required should be assigned FF.

The following time intervals are permitted:

- Hours: 01, 02, 03, 04, 06, 08, 12, 24
- Minutes: 01, 02, 03, 04, 05, 06, 10, 12, 15, 20, 30, 60
- Seconds: 10, 12, 15, 20, 30, 60

Examples:

- IntervalMode : TRUE
The output "TriggerOutput" is set as follows:
 - Hour_Minute : W#16#06FF (every 6 hours)
 - Second_Day : W#16#FFFF (at 00:00:00, 06:00:00, 12:00:00 o'clock and ...)
 - Month_Year : W#16#FFFF (... at 18:00:00 o'clock)
- IntervalMode : TRUE
The output "TriggerOutput" is set as follows:
 - Hour_Minute : W#16#FF30
 - Second_Day : W#16#FFFF (at 00:00:00, 00:30:00, 01:00:00 o'clock and ...)
 - Month_Year : W#16#FFFF (... at 01:30:00, 02:00:00, 02:30:00 o'clock etc.)

Error message during startup

The FC checks the parameters Hour_Minute, Second_Day and Month_Year in every cycle to ensure that they keep to the permitted range of values. What is permitted is also dependent on the "IntervalMode" parameter.

If the parameter assignment is incorrect, an error message is entered in the diagnostics buffer (event ID B113) only during startup. The CPU does not change to STOP. Afterwards, the FC checks the parameters without outputting error messages until the error has been eliminated.

The diagnostics message provides a precise identification of the incorrect parameter (continuous number of the parameter, i.e. 2, 3 or 4). The causes of the diagnostics message can be depend on the parameter "IntervalMode".

IntervalMode = FALSE (or 0)

The permitted ranges of values for the parameters Hours, Minutes, Seconds, Day, Month and Year were not kept to. Apart from FF, the following can be configured:

Hours	00-23	Day	01-31
Minutes	00-59	Month	01-12
Seconds	00-59	Year	00-99

"IntervalMode" = TRUE (or 1)

In this case the error can have two different causes:

- The permitted ranges of values for the parameters Hours, Minutes and Seconds were not kept to. Apart from FF, the following can be configured:
 - Hours: 01, 02, 03, 04, 06, 08, 12, 24
 - Minutes: 01, 02, 03, 04, 05, 06, 10, 12, 15, 20, 30, 60
 - Seconds: 10, 12, 15, 20, 30, 60
- A time interval can only be set in hours or in minutes or in seconds. The two unused parameters must have FF written to them. Even if FF was entered for all three named parameters, an error exists.

3.8.3 System blocks

3.8.3.1 BasicTask_* FC

Block versions

Note

Only 1 FC type per CPU

You may only use one FC type in one CPU. A mixture of several communication types with different FCs "BasicTask_*" is not allowed.

The block is available in the following versions for the different CPU types:

- **BasicTask_B**
For S7-1500 CPU
The block for the S7-1500 CPU cannot be used for the S7-300 or S7-400 CPU.
- **BasicTask_B**
For S7-400 CPU
The block for the S7-400 CPU cannot be used for the S7-300 or S7-1500 CPU.
- **BasicTask_Bnn**
Prepared for S7-300 CPU (cannot be used in STEP 7 V16 yet)
The blocks for the S7-300 CPU cannot be used for the S7-400 or S7-1500 CPU.
As of version V3.0 + SP2 of the block library, 16 versions of the "BasicTask_B" FC are provided for an S7-300 CPU. These are provided for the respective configured number of local TIM 1531 IRC, via which the CPU communicates. According to the number (nn) of local TIM modules, the FC receives the suffix "nn", e.g. "BasicTask_B03".
When generating the TD7onCPU in the CPU a suitable "BasicTask_B" FC is generated. The block version, from which the number of local TIM 1531 IRC is determined, can be taken from the "Comment" block property of the "BasicTask_B" FC in the "TD7onCPU" directory of the CPU.
- **BasicTask_X**
For S7-300 PU with partyline
- **BasicTask_P**
For S7-300 CPU without partyline

For information on "Partyline", see Glossary.

The tasks of the various "BasicTask_*" FCs for handling basic communication tasks are the same in the respective CPU type. The only parameter of the FC is identical in each case.

When the TD7onCPU is generated, the correct FC is automatically created in the respective CPU.

Function

The block is required in every CPU. It handles the following tasks:

- Central tasks during startup
- The processing of all communication mailboxes
- Central organizational tasks such as starting, monitoring and answering general requests.

Call FC BasicTask as the first block in OB1.

Parameter

Name: **UserFC**

Declaration: INPUT

Data type: INT

Range of values:

- 1 ... 32000
Number of the FC
The maximum possible number depends on the CPU.
- 0
Substitute value if there is no FC present for the specified purpose.

Explanation: Number of a user FC for specific further processing of the received data.
If an FC is specified, this FC is called automatically by the user program with all received data. At the time of the call, the receive frame is still in the receive mailbox of the communication DB.
The program in the user FC can read the receive frame from the receive mailbox and process it further in any way required, e.g. write the received data to an intermediate buffer.
Using the user program, the required information on the communication DB can be read from the DB BasicData via the "CurrentComDB" variable in DW60. CurrentComDB contains the following information:

- S7-1500
Index of the current BConnection instance in the DB BConnectData
- S7-300/400
Number of the current communication DB

In the opened communication DB, the start of the current receive frame in the receive mailbox can be found via the "CurrentReceivedMessage" variable in DW10.

3.8.3.2 DB BasicData

This data block provides the central data management. It contains information that must be maintained centrally for all blocks. Among other things, the data block contains the subscriber records and the connection descriptions.

DB BasicData is automatically generated in the required length preset with the subscriber and connection specific data and then saved in the block directory of the CPU.

DB BasicData exists once on every CPU.

Note**Number of DB BasicData**

In the TD7 library, DB BasicData has the number DB127 and is also saved there under this number when the DB is generated for the various CPUs. In principle it would be possible to change the number, but this involves a lot of work and can cause errors in the further creation of the user program.

It is therefore recommended to keep the DB number 127 free for the DB BasicData.

3.8.3.3 FC Create

Auxiliary block for putting together the data to be sent and its entry in the relevant instances of the send mailbox(es) (SendBuffer array) in the respective communication DB. These are the following DBs:

- **S7-1500**
DB BConnectData > BConnection[n]
- **S7-400**
DB BComDataXX
- **S7-300**
 - DB BComDataXX
or
 - DB XComDataXX
or
 - DB PComDataXX

XX is a placeholder for the sequential number (01, 02, etc.).

FC Create is required by the data point typicals for data and organizational frames and by FC BasicTask for organizational frames only.

3.8.3.4 Diagnose / Diagnostics FC

Name of the block:

- For S7-300: FC Diagnose
- For S7-1500: FC Diagnosis

Auxiliary block for entering SINAUT system messages in the diagnostics buffer of the CPU.

3.8.3.5 FC Distribute

Auxiliary block for distribution of the data located in the receive mailbox to the data point typicals responsible or to the node objects in the subscriber records.

3.8.3.6 FC Search

Auxiliary block for the following search tasks:

- Search for the initial address of a subscriber object within the subscriber records
- Search for the local object no. (Instance DB) from one of the two object reference lists for a received message with an incomplete destination address

The auxiliary block is required by almost all blocks.

3.8.3.7 FC Startup

The block is required in every CPU. It must be linked into the startup program OB100.

The block has the task of setting the startup memory bit in the DB BasicData and to reset the corresponding edge memory bit if this is still set.

The block has no parameters.

3.8.4 Communication blocks BCom (CPU1500)

3.8.4.1 FB BCom

Validity

- S7-1500

The block for the S7-1500 CPU cannot be used for the S7-300/400 CPU and vice versa. The tasks of the two FBs are the same.

Function

Auxiliary block for FC BasicTask for processing a communication mailbox of the DB BComData type (S7-1500).

Via the DB, a configured PBK (programmed block communication) connection is handled using the SFBs "BSEND" and "BRCV".

FB BCom also ensures that received frames are distributed immediately to the receive objects responsible in the CPU. To do this FB BCom calls FC Distribute as an auxiliary block.

3.8.4.2 BConnect FB

Validity

S7-1500

Function

Auxiliary block for FC BasicTask for processing connections to the communication partners.

3.8.4.3 DB BConnectData**Validity**

S7-1500

Function

Instance data block for the communication block FB BCom. The instance DB represents the communication mailbox and contains the following, among other things:

- The receive mailbox (ReceiveBuffer)
- A send mailbox (SendBuffer)

It also contains central data required to control and manage the PBK connection that runs via this mailbox.

The data block is required in every CPU in which FB BCom is used. If the CPU has several PBK connections the DB is required more than once.

DB BComData is automatically generated in the required length preset with the connection specific data and then saved in the block directory of the CPU.

3.8.5 Communications block FB BCom (CPU300/CPU400)**3.8.5.1 BCom FB****Validity**

- S7-400
- S7-300

The blocks for the S7-300/400-CPU cannot be used for the S7-1500-CPU and vice versa. The tasks of the FBs are the same.

Function

Auxiliary block for FC BasicTask for processing a communication mailbox of the type DB BComData (S7-300/S7-400).

A configured PBK connection (programmed block communication) is handled via the DB using the "BSEND" and "BRCV" SFBs of an S7-400 CPU or the function blocks (FBs) of the same name of an S7-300 CPU.

FB BCom also ensures that received frames are distributed immediately to the receive objects responsible in the CPU. To do this FB BCom calls FC Distribute as an auxiliary block.

S7-300

Availability and function

The FB BCom V1.5 is in preparation (cannot be used in STEP 7 Professional V16).

The BCom FB can only be used as of block version V1.5 for an S7-300 CPU, . The block is included in the "Telecontrol ST7" library as of version V3.0 SP2. The block version can be found in the properties dialog of the block in the "TD7onCPU" container of the program blocks of the CPU.

The block is required for an S7-300 CPU especially for communication via a local TIM 1531 IRC.

Number of configurable S7 connections

S7-300 CPU supports up to 16 S7 connections for TD7onCPU. The configurable number of S7 connections depends on the respective CPU type and cannot be exceeded even by CPUs with more connection resources.

3.8.5.2 DB BComData

Validity

S7-300

S7-400

Function

Instance data block for the communication block FB BCom. The instance DB represents the communication mailbox and contains the following, among other things:

- The receive mailbox (ReceiveBuffer)
- A send mailbox (SendBuffer)

It also contains central data required to control and manage the PBK connection that runs via this mailbox.

The data block is required in every CPU in which FB BCom is used. If the CPU has several PBK connections the DB is required more than once.

DB BComData is automatically generated in the required length preset with the connection specific data and then saved in the block directory of the CPU.

3.8.6 Communication blocks PCom (CPU300)

3.8.6.1 FB PCom

Validity

S7-300 CPU without partyline

FB PCom is used only with communication via the P bus. This affects the communication between a TIM and a CPU with P bus.

Function

Auxiliary block for FC BasicTask_P for processing a communication mailbox of the type DB PComData using the SFCs WR_REC and RD_REC.

Received frames are also distributed immediately to the receive objects responsible in the CPU. To do this FB PCom calls FC Distribute as an auxiliary block.

3.8.6.2 DB PComData

Instance data block for the communication block FB PCom. The instance DB makes the communication mailbox available and contains the following:

- The receive mailbox (ReceiveBuffer)
- A send mailbox (SendBuffer)

It also contains central data required to control and manage the connection that runs via this mailbox.

The data block is required in every CPU in which FB PCom is used. If the CPU has several corresponding connections the DB is required more than once.

3.8.7 Communication blocks XCom (CPU300)

3.8.7.1 FB XCom

Validity

S7-300 CPU with partyline

Function

Auxiliary block for FC BasicTask_X for processing a communication mailbox of the type DB XComData.

3.8 Master copies

Via DB BComData a configured connection (PBK connection) is handled using the SFCs "X_SEND" and "X_RCV".

The FB XCom also ensures that received frames are distributed immediately to the receiving objects responsible in the CPU. To do this the FB XCom calls the FC Distribute as an auxiliary block.

3.8.7.2 DB XComData

Instance data block for the communication block FB XCom. The instance DB makes the communication mailbox available and contains the following:

- The receive mailbox (ReceiveBuffer)
- A send mailbox (SendBuffer)

It also contains central data required to control and manage the X connection that runs via this mailbox.

The data block is required in every CPU in which FB XCom is used. If the CPU has several X connections the DB is required more than once.

DB XComData is automatically generated in the required length preset with the connection specific data and then saved in the block directory of the CPU.

Commissioning, diagnostics and maintenance

4.1 Commissioning ST7: General requests

General request (GR)

On startup of a system, a general request (GR) is sent from most process control systems to all connected stations so that the master station is updated with the current process state of the stations. This means that conventional control center systems can update their process image and supply their archives with data.

The mechanisms of a GR depend on the protocol and the control system used.

General request with ST7

Standard general request

When a TIM is connected to the SINAUT ST7cc control center system, a GR can be triggered as follows:

- If a station connected to SINAUT ST7cc returns following a disruption, ST7cc sends a GR to the station to update itself with the current process state of the station.
- If ST7cc itself fails and starts up again, it sends a GR to all connected stations to bring its process data up to date.
- In addition to these automatically triggered GRs, you can also manually send a GR to a station in ST7cc.

In response to a standard GR, the queried station enters all its data frames along with the current process values in the send buffer of the station TIM. The data is transferred according to the first in/first out principle. The advantage of the standard GR is that the data is transferred in the correct chronological order.

Accelerated general request

If you require the current process image more quickly, you can trigger an accelerated GR. In an accelerated GR, the frames with the requested process image are entered at the start of the send buffer of the TIM; in other words, before any other frames still buffered in the TIM.

An accelerated GR is supported by:

- SINAUT ST7cc
- Ethernet TIM with TD7onTIM
(Stations with TD7onCPU answer an accelerated GR like a standard general request.)

It may make sense to trigger an accelerated GR in the case of stations connected via dialup network, for example, because the master station is not always supplied with the current data of the station in this case. One example application would be a situation whereby you inform yourself of the current status of the station with an accelerated GR prior to a switching action or setpoint change. This would allow you to enter the command or setpoint

4.2 Diagnostics

faster and inform yourself of the current response of the process in the station by means of another accelerated GR.

You can find additional information in the SINAUT ST7cc manual.

4.2 Diagnostics

4.2.1 Diagnostics options

The following diagnostic options are available with most modules. Some functions are limited to specific data types or protocols.

LEDs of the module

For information about the LED displays, see Volume 1 of the System Manual or the manuals for the modules.

STEP 7: The "Diagnostics" tab in the Inspector window

If your engineering station is connected to the TIM via Ethernet, here you will receive information about the selected module:

- Connection status of the engineering station with the TIM

Web server (WBM) of the TIM 1531 IRC

From a PC you can use HTTP/HTTPS to access the Web pages (WBM) of the TIM. The WBM returns a variety of information.

For information on access and the contents, see manual /3/ TIM 1531 IRC (Page 426).

Partner status and connection states in the WBM

You will see the configured partners and the status of the connections to the local and remote communications partners of the TIM on the page "Telecontrol" > "Partner information" of the WBM.

Partner and connection information to the CPU

The TIM can signal the status of the connection and the connection paths to the communications partner to its local CPU via a PLC tag. For information on the configuration, refer to section Communication with the CPU (Page 94).

STEP 7: Diagnostics functions in the "Online > Online and diagnostics" menu

Using the online functions, you can read various diagnostics information from the TIM from an engineering station on which the STEP 7 project is stored and perform maintenance functions.

You will find additional information on the diagnostics functions of STEP 7 in the STEP 7 information system.

Online access

This is where you establish the online connection to the module.

For the procedure, refer to the STEP 7 information system.

Diagnostics

Here, you can obtain the following static information on the selected module:

- **General**
General information on the module
- **Diagnostics status**
Information on the diagnostics status
- **Diagnostics buffer**
This is where you can find entries in the diagnostics buffer of the TIM 3V-IE / TIM 4R-IE.
- **Ethernet interface[X1/2/3]**
Address and statistical information
- **Industrial Remote Communication**
Here, you obtain WAN-specific information on the TIM module:
 - **Partner**
Here you will find address and configuration data of the partners, connection statistics and additional diagnostics information. Click on a subscriber to display additional information.
You will also find information on the partners in the WBM, see below.
 - **Data point list**
Information on the data points such as configuration data, value, connection status etc.
 - **Telegram protocol diagnostics**
With this function, you can enable the logging of frames of the module and evaluate it using the SINAUT engineering software.
To evaluate the file, you need to rename it to the format "*.7dt". You can decode and analyze the renamed file using the TIM frame monitor of the SINAUT diagnostics tool.
See section Telegram protocol diagnostics (Page 360) for more on this.
 - **Ethernet diagnostics**
With the Logging function, you can log the data traffic of the TIM using PCAP functionality for diagnostics purposes.
If an error occurs or if the TIM behaves in an unwanted manner, the communication behavior of the TIM can be recorded. The frame traffic of the TIM is recorded for a defined time or for a configurable number of frames.
The log files are stored as PCAP files on the connected PC and can be evaluated with the Wireshark program, for example.

4.2 Diagnostics

- **Security**
 - Status
Shows information on the general data of the module.
 - System log
Lists the entries of the system log.
 - Audit log
Lists the entries of the audit log.
 - Packet filter log
Lists the entries of the packet filter log.
 - Communication status
Shows the status of the endpoints and the VPN tunnel.
 - SINEMA RC - ...
Shows the status of the SINEMA RC configuration and communication.
- **Time**
Specification of the current time in the module and the time source

Functions

You can run the following functions here, depending on the module:

- **Set time**
- **Firmware update**
For a description, see section Update firmware (TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE) (Page 397).
- **Assign IP address**
- **Reset to factory settings**
For information on the functions, refer to the section Maintenance (Page 395).
- **Save service data**
The function is used for logging of internal module processes in situations in which you cannot eliminate unexpected or unwanted behavior of the module yourself.
The log file is created with the "Save service data" button. The data is saved in a file with the format "*.dmp" that can be evaluated by the Siemens hotline.
- **SINAUT special diagnostics**
Validity: TIM 3V-IE / TIM 4R-IE
For a description, see section SINAUT special diagnostics (Page 363).
- **Extended diagnostic functions**
Validity: TIM 3V-IE / TIM 4R-IE
This function is only intended for analyses by Siemens Customer Support.

4.2.2 Online security diagnostics via port 8448 of the CP

Security diagnostics via port 8448

Requirements:

- With an activated firewall, access must be enabled.

If you want to perform security diagnostics in STEP 7 Professional, follow the steps below:

1. Select the CP in STEP 7.
2. Open the "Online & Diagnostics" shortcut menu.
3. In the "Security" parameter group, click the "Connect online" button.

In this way, you perform the security diagnostics via port 8448.

Port 8448 of the TIM 1531 IRC

With the TIM 1531 IRC, Port 8448 is open in the default setting.

4.2.3 Web server S7-1200: Connection establishment

Establishing a connection to the Web server of the CPU

Follow the steps below to connect to the Web server of the CPU from a PC.

Requirements in the CPU configuration

1. Open the corresponding project on the engineering station.
2. Select the CPU of the station involved in STEP 7.
3. Select the "Web server" entry.
4. In the parameter group "General", select the "Enable Web server for this interface" option.
5. With a CPU version V4.0 or higher, create a user in the user administration with the required rights.

The procedure for establishing a connection to the Web server depends on whether you have enabled or disabled the "Allow access only using HTTPS" option in the "General" parameter group:

- **Connection establishment with HTTP**
Procedure if the "Allow access only using HTTPS" option is disabled
- **Connection establishment with HTTPS**
Procedure if the "Allow access only using HTTPS" option is enabled

These two variants are described in the following sections.

You will find the requirements for access to the Web server of the CPU (permitted Web browser) and the description of the procedure in the STEP 7 information system under the keyword "Information about the Web server".

4.2 Diagnostics

Connection establishment with HTTP

1. Connect the PC to the CPU via the Ethernet interface.
2. Enter the address of the CPU in the address box of your Web browser: `http://<IP address>`
3. Press the Enter key.
The start page of the Web server opens.
4. Click on the "Download certificate" entry at the top right of the window.
The "Certificate" dialog opens.
5. Download the certificate to your PC by clicking the "Install certificate ..." button.
The certificate is loaded on your PC.
You will find information on downloading a certificate in the help of your Web browser and in the STEP 7 information system under the key words "HTTPS" or "Access for HTTPS (S7-1200)".
6. When the connection has changed to the secure mode HTTPS ("`https://<IP address>/...`" in the address box of the Web server), you can continue as described in the next section.
When you terminate the connection to the Web server, the next time you can log in with the Web server without downloading the certificate using HTTP.

Connection establishment with HTTPS

1. Connect the PC to the CPU via the Ethernet interface.
2. Enter the address of the CPU in the address box of your Web browser: `https://<IP address>`
3. Press the Enter key.
The start page of the Web server opens.
4. Log in on the start page of the Web server as a user with the necessary rights.
Use the user data configured in the user administration of the Web server of the CPU.
5. After logging in, select the entry "Module status" in the navigation panel of the Web server.
6. Select the CP in the module list.
The CP-specific content is displayed.

4.2.4 Telegram protocol diagnostics

The telegram protocol diagnostics is available for the following modules:

- S7-1200 telecontrol CPs
- CP 1542SP-1 IRC
- TIM 1531 IRC

4.2.4.1 Message protocol: Structure and functions

The "Telegram protocol diagnostics" function is used to record transferred message frames.
See the "Operation" paragraph below for information on activating the function.

Structure of the "Message protocol" dialog

After activation, the columns show the following data:

- No.
 - Symbol for incoming or outgoing message frames
 - Message frame number, consecutive
- Block
Length of the message frame block in which the message frame was transferred.
- Header fields
Hexadecimal display of some header data
For the meaning, refer to section Details (Page 362).
- Subscriber (source / destination)
Subscriber numbers of the sending (source) and receiving (destination) subscriber
- Object (source / destination)
Object number of the data object in the message frame for source and destination subscriber
- Index
Address parameters for net data in data messages (channel number)
- Date, time and status
Time stamp of message frame and time status from time of transfer
For information on the display, see section Details (Page 362).
- Net Data
Hexadecimal display of the net data of the message frame
For the meaning, refer to section Details (Page 362).

Operation

You operate the function using the three buttons underneath the dialog described above.

- Activate telegram protocol diagnostics
Clicking the button starts logging of the transferred message frames.
400 message frames are recorded per recording cycle. A maximum of 10000 message frames can be stored in a circular buffer.
- Refresh
Updates the recording; a new recording cycle is started.
- Save
Saves the recorded message frames in a binary file. You define the storage location using the button.
- Show saved
Shows the recorded and saved frames.

4.2.4.2 Details

Detailed information

Header fields

The 5 fields output in hexadecimal format have the following meaning:

- 1st field: Message counter
0...7
- 2nd field: Control code
- 3rd field: Function selection
 - 0: Processing in TIM
 - 1: Processing in CPU
- 4th field: Address extension
 - 0: ST1 message without address extension
 - 1: ST1 message with address extension
 - 2: ST7 message
- 5th field: Direction bit
 - 0: Monitoring direction
 - 1: Control direction

Net Data

The column shows the net data of the message frame.

The values are displayed in hexadecimal format.

Date, time and status

The column shows the time stamp of the message frame in the following format:

Year/Month/Day_Hour:Minute:Second_Time status

Assignment of the time status:

- 2^0
 - 0: Time is invalid
 - 1: Time is valid
- 2^1
 - 0: Standard time
 - 1: Daylight saving time

- 2²
Not used
- 2³
 - 0: No meaning
 - 1: Notification time for daylight saving/standard time changeover
Only in time synchronization message frame

4.2.5 SINAUT special diagnostics

The SINAUT special diagnostics is available for the following modules:

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

4.2.5.1 TIM diagnostics

TIM Diagnostics - functions

Description of the functions

TIM Diagnostics 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" tab:
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
- "MSC Communication" tab
Information on communication with the MSC protocol

TIM Diagnostics - Memory tab

Memory tab

The Memory tab displays current diagnostic data of 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 sufficient, a check mark 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.

TIM Diagnostics - Message buffer tab

Message buffer tab

The Message buffer tab is available only for TD7onTIM-capable 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 of the CPU or an ST7cc/sc

The message buffer diagnostics function analyzes the buffer areas of the destination subscribers in which the data frames are stored. These are of particular interest to the user.

"Total" box

The Total box provides the following information:

- **Size (blocks):**
Total size of the frame 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 performed in the network configuration in NetPro in the Properties dialog, Options tab, Global frame memory field.
- **Free (blocks):**
Free area of the frame buffer. The value indicates the number of free memory blocks.
- **Free (%):**
Free area of the frame buffer as a percentage
- **Image blocks:**
Number of blocks occupied in the frame memory of the TIM for data frames transmitted using the image memory principle.

"Buffers" box

The Buffers box lists the frame 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 frames:**
Total number of stored frames for the relevant source and destination subscriber

If a frame 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 frames:**
Total number of stored frames for the source and destination subscriber named above
- **No. of uncond. frames:**
Number of stored frames to be sent unconditionally and spontaneously (only relevant in dial-up networks)
- **No of prio. frames:**
Number of stored frames to be sent with high priority.

4.2 Diagnostics

- **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 frame memory, all data frames are treated as image frames; in other words, even send buffer frames are treated as image frames and overwritten.
- **Data brake:**
The sending of frames 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. frames:**
Reserved for future functions. Nothing is currently displayed.
- **Est. dial-up conn.**
Reserved for future functions. Nothing is currently displayed.

TIM Diagnostics - Communication tab

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.

TIM Diagnostics - Time Synchronization tab

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.

TIM Diagnostics - Time tab

Time tab

The Time tab displays the date and current module time of the TIM on the left in the Current time field. Information on the validity of the time, daylight saving/standard time and the changeover from daylight saving to standard time is displayed on the right in the Time-of-day status field.

TIM Diagnostics - Filesystem tab

File system tab

The File system tab 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 will also be displayed with the files located there.

TIM diagnostics - IP Parameters tab

IP parameters tab

This tab is available only for the TIM 4R-IE.

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 direction dependency on Ethernet.

TIM diagnostics - IP statistics tab

Statistics tab

This tab is available only for the TIM 4R-IE.

The tab contains transmission statistics for the Ethernet interfaces. 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.

TIM diagnostics - MSC Communication tab

MSC communication tab

This tab exists only for a TIM 4R-IE as MSC master station.

SINAUT Diagnostics - TIM 4R-IE ONLINE

Path:

Memory | Message buffer | Communication | Time Synchronization | Time

Filesystem | IP parameter | IP statistics | MSC Communication

Transmission Information

Sub.-No.	Station	Signal	Active month sent	Active month received
1010	Station 10	no GPRS	-	-
1020	Station 20	no GPRS	-	-
1060	Station 60	19	28237460	48259477
1250	Station 250	23	28496726	54124208
1400	Station 400	25	22059244	37210532

Information Details

Device name: Quality of signal:

Firmware version: Local area code:

Protocol version: Cell ID:

Data Volume

	Active count	Count at begin of month
Amount of data sent [bytes]:	<input type="text" value="28237460"/>	<input type="text" value="0"/>
Amount data received [bytes]:	<input type="text" value="48259477"/>	<input type="text" value="0"/>

Close Update Print ... Save ... Help

Figure 4-1 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 station 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

4.2 Diagnostics

The following information is only supplied by GPRS stations with MD720:

- Signal quality
Relative receive quality (RSSI)
 - 0 = ≤ -113 dBm
 - 1 = -111 dBm
 - 2...30 = -109 ... -53 dBm
 - 31 = ≥ -51 dBm
 - 99 = Unknown or not measurable
- Local area code
Location of area in the mobile wireless network
- Cell ID
Identification 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 following stations:
 - 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 1/2 a minute.
- Print...
Opens a follow-on dialog from which you can print the displayed information.
- Save...
Opens a follow-on dialog from which you can save the displayed information in a text file.

4.2.5.2 TIM subscriber diagnostics

TIM Subscriber Diagnostics - functions

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 reachability of the partners, connections 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 list external tab (optional):
Dial-up service and command of the connected TIM module
- Polling list tab (optional):
Data of the station call 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 reachability of the individual subscribers and have the following meaning:

Table 4-1 Symbols indicating subscriber reachability in subscriber diagnostics

Symbol	Status
	Subscriber is reachable, all connections are OK
	Subscriber is reachable, at least one connection is disrupted
	Subscriber is not reachable

TIM Subscriber Diagnostics - Status tab

Status tab

The Status tab 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:
 - Reachability of the known subscriber. Entries indicating problems are highlighted in color.
 - 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 calling
 - Status of data communication
- In the Special field:
 - Information on the operation of the send buffer of the known subscriber

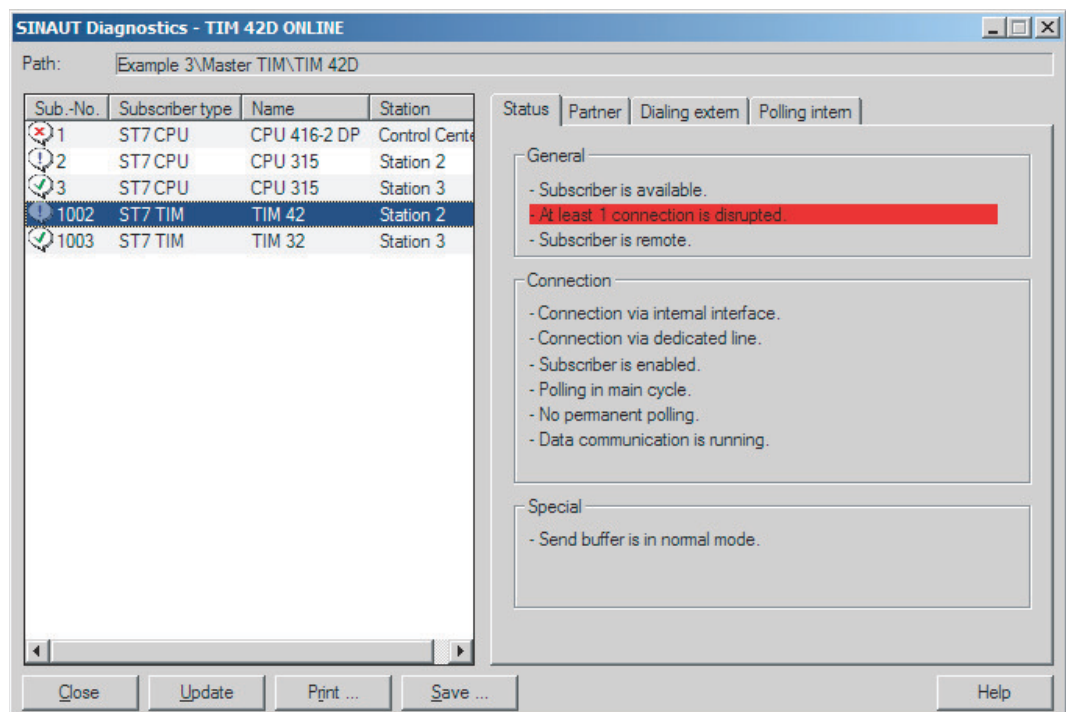


Figure 4-2 TIM subscriber diagnostics dialog, Status tab

TIM Subscriber Diagnostics - Partner tab

Partner tab

The Partner tab displays the following known partners with their subscriber number, name and station in the List of communication 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 communication 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.

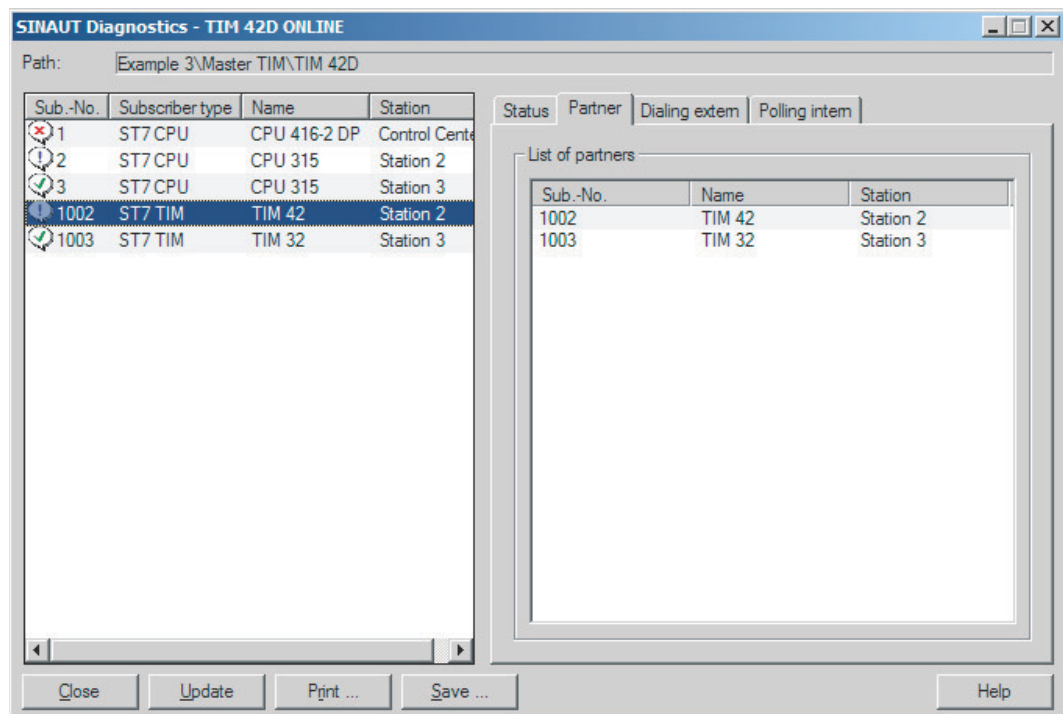


Figure 4-3 TIM subscriber diagnostics dialog, Partner tab

TIM Subscriber Diagnostics - Dialing extern tab

Dialing list external tab

If a TIM with a central function is selected before opening the dialog, then special services and the telephone number list (dial command) of the dial-up network driver of the connected TIM modules are displayed in the Dialing list external tab.

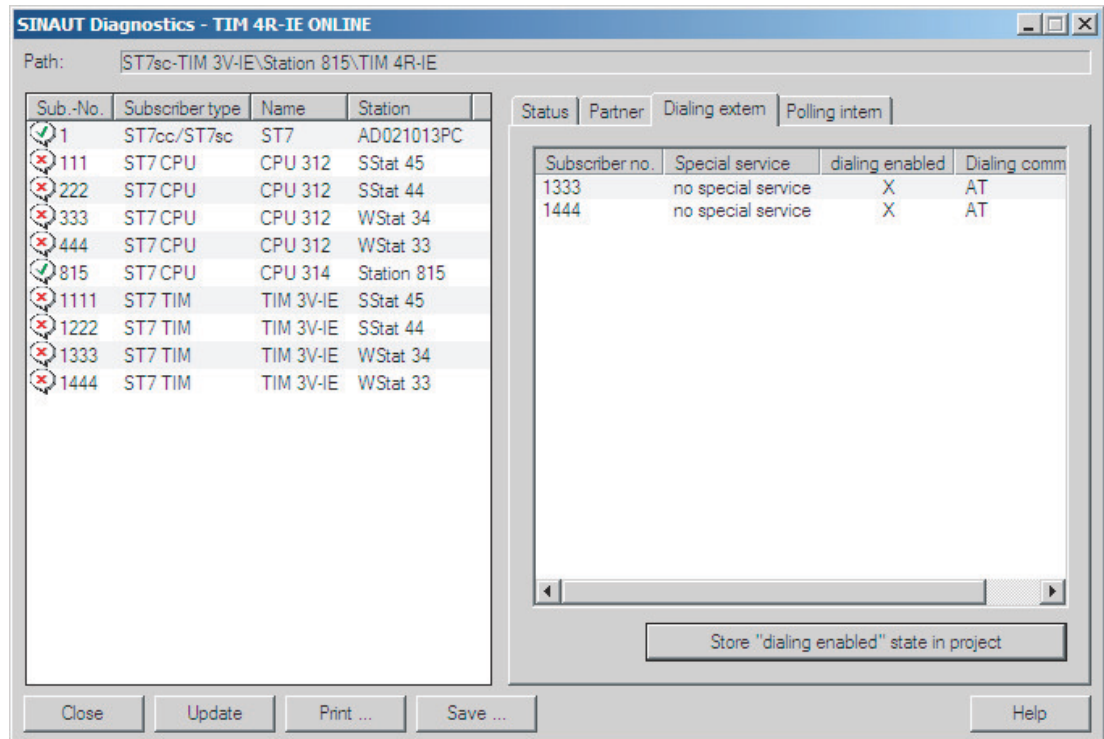


Figure 4-4 TIM subscriber diagnostics dialog, Dialing list external tab

The Subscriber no., Special service, Dialing enabled and Dialing command parameters are displayed for the listed TIM modules.

An "X" in the Dialing enabled column indicates 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 status of the selected subscriber is displayed from the perspective of the master TIM. The current enable status can be switched.
- The settings for enabling subscribers are stored permanently.
- Using the button below the list, you save the current enable state in the STEP 7 project of the connected PG/PC.

TIM Subscriber Diagnostics - Polling intern

Polling list internal tab

If a TIM with master function is selected before opening the dialog, the Polling list internal tab displays the data of the station for the known subscribers connected to the connected TIM over a dedicated line.

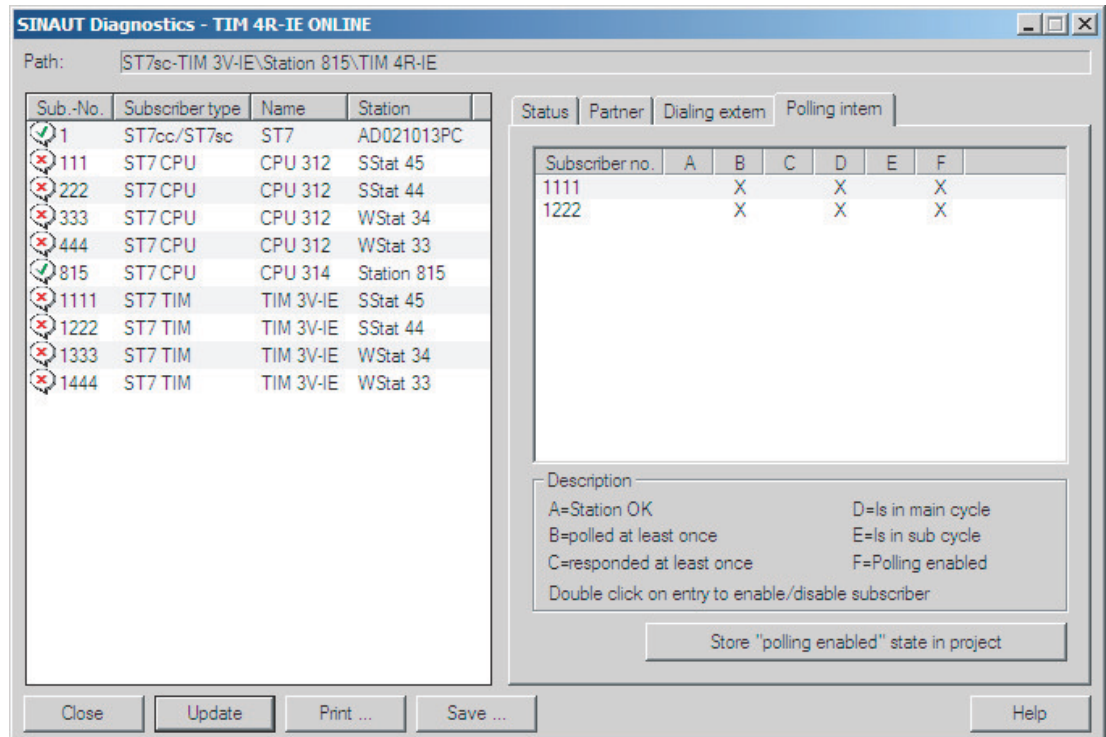


Figure 4-5 TIM subscriber diagnostics dialog, Polling list internal tab

By double-clicking on a subscriber in the polling list, the Disable/Enable subscribers dialog opens in which the configured and current enable status of the selected subscriber is displayed from the perspective of the master TIM. The current enable status can be changed. If you change the state to Disable in this dialog, the configured state 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 save the current enable state in the STEP 7 project of the connected PG/PC.

4.2.5.3 TD7onTIM diagnostics

TD7onTIM Diagnostics - functions

Description of the functions

Note

The TD7onTIM diagnostics function is available only for TD7onTIM-capable 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.

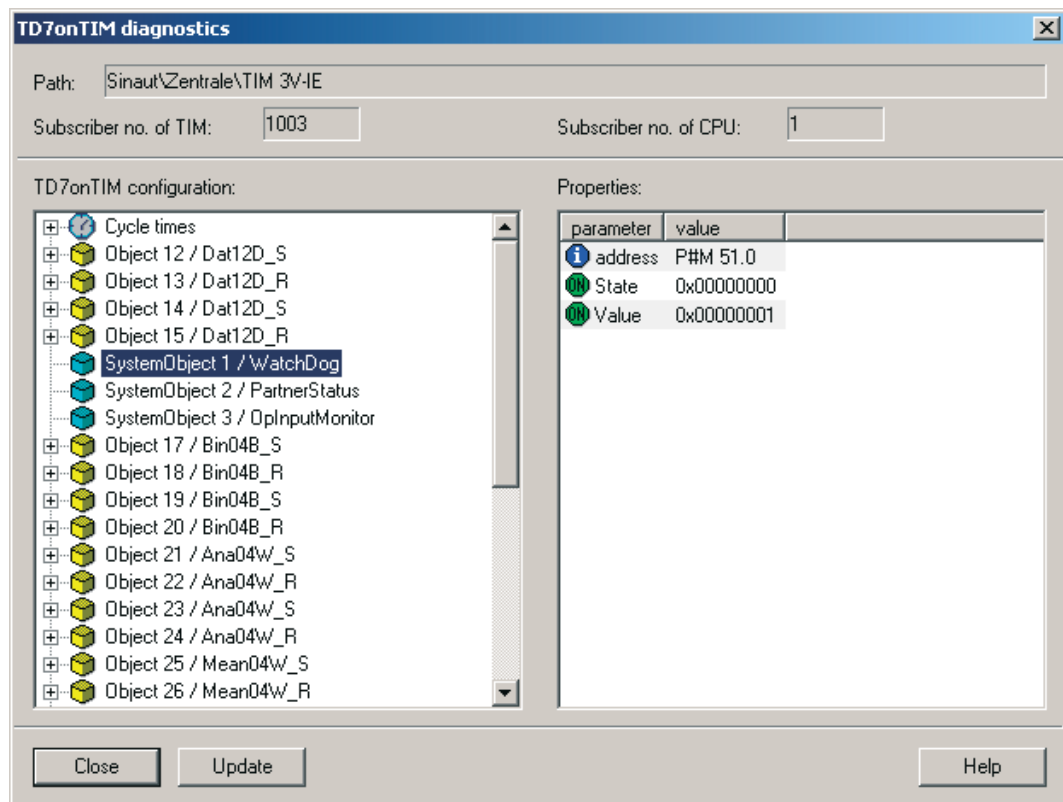


Figure 4-6 TD7onTIM diagnostics dialog with system object parameters based on the example of WatchDog

In keeping with the parameter assignment of the 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 cycle time
 - Send and receive channels
- Properties:
This box 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 box.

4.2 Diagnostics

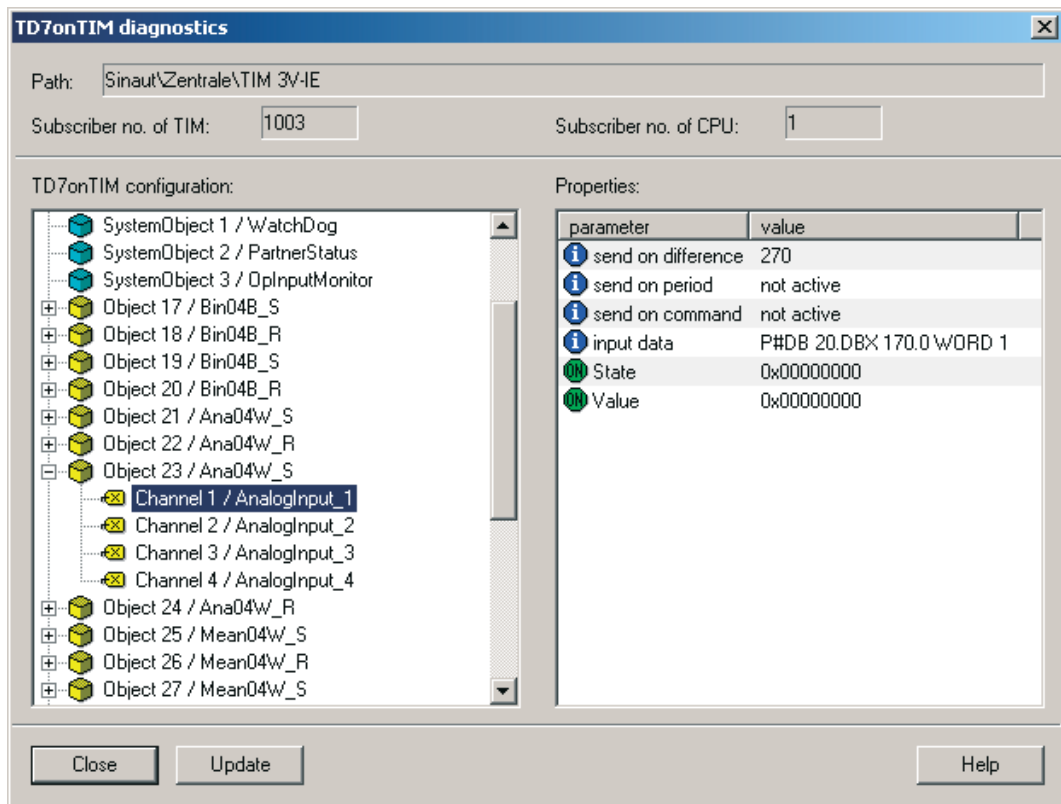


Figure 4-7 TD7onTIM diagnostics dialog with channel parameters based on the example of the Send analog value channel

4.2.6 Messages in the diagnostic buffer of the TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE

4.2.6.1 Introduction

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.

4.2.6.2 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 instead of x a digit is inserted that allows a global classification of the message:

Table 4-2 Global classification of the TIM diagnostic messages

Identifier x	Resulting event ID	Classification
2	F260	Operating message, exiting state
3	F360	Event message, entering state
4	F460	Operating message, internal error, exiting state
5	F560	Operating message, internal error, entering state
8	F860	Operating message, external error, exiting state
9	F960	Operating 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 detailed information under *Additional info 1/2/3* or *Additional info 4/5*.

The diagnostic messages of the TIM in hexadecimal and plain text format

The following table lists all the TIM diagnostic 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 4-3 Diagnostic 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.

4.2 Diagnostics

Event ID (hex)	Detailed event (hex)	Meaning
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 time driver.
F560	0073	Error loading the basic task of the time driver.
F560	0074	Error creating the basic task of the time driver.
F560	0075	Error receiving the task ID of the time driver.
F560	0076	Error starting the basic task of the time driver.
F560	0077	WAN SDB does not start with the <i>TIM</i> 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 starting the main task of the routing program.
F560	007F	Routing tables were not created within the specified time.
F560	0080	WAN driver was not installed within the specified time.
F560	0081	Unknown parameter block in WAN SDB.
F360	0082	Startup of the module completed.
F560	0083	Flag group unknown.
F360	0084	RAM drive was created.
F560	0085	Error creating the RAM drive.
F560	0086	Error in memory analysis in HEAP.
F560	0087	Error installing the time driver. No message received at end of installation.
F360	0088	Message of the message and HEAP memory set up 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.

Event ID (hex)	Detailed event (hex)	Meaning
F560	0092	Installation error in the Start manager
F560	0100	Module startup: Installation error occurred
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: X 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.
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 section Meaning of ERRCLS and ERRCOD (Page 391).
F560	0114	No Ethernet SDB exists.
F560	0115	Ethernet SDB could not be opened.
F560	0116	Ethernet SDB with wrong 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 interface: 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.

4.2 Diagnostics

Event ID (hex)	Detailed event (hex)	Meaning
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 via 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.
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 blocked.
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 subscriber call enabled.
F260		Exiting state: Internal / external WAN interface: Permanent subscriber call ended.

Event ID (hex)	Detailed event (hex)	Meaning
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 time driver disrupted.
F460		Exiting state: Internal / external WAN interface: Communication with time 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 wrong 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.
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: Time monitoring has expired. 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 blocked.
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.

4.2 Diagnostics

Event ID (hex)	Detailed event (hex)	Meaning
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 de-registered.
F560	032F	Internal / external WAN interface: Permanent connection to a subscriber was aborted.
F360	0330	Entering state: Internal / external WAN interface: Incoming call blocked.
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: Time monitoring has expired. 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.
F260		Exiting state: Internal / external WAN interface: Driver redundancy - memory management switched back.
F560	033D	Internal / external WAN interface: Wrong 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 reachable 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 reachable.
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 however blocked.
F360	034F	Internal / external WAN interface: Incoming call detected. DTR signal was activated.

Event ID (hex)	Detailed event (hex)	Meaning
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.
F360	0361	Entering state: Internal / external WAN interface: The data brake for the connection to a subscriber was enabled.
F260		Exiting state: Internal / external WAN interface: The data brake for the connection to a subscriber 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 routing 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 time 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.

4.2 Diagnostics

Event ID (hex)	Detailed event (hex)	Meaning
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 via 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.
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 time driver		
F560	0620	Installation of time driver aborted.
F360	0621	for error code = 0: Installation of the time driver completed for error code = 1: Installation of time driver aborted.
F560	0622	Entering state: Time synchronization (master) disrupted.
F460		Exiting state: Time synchronization (master) OK.
F960	0623	Entering state: Time synchronization (slave) disrupted.
F860		Exiting state: Time synchronization (slave) OK.
F560	0624	Error occurred setting the RMOS clock.
F560	0625	Error occurred reading the RMOS clock.

Event ID (hex)	Detailed event (hex)	Meaning
F360	0626	Illegal setting of the RMOS clock by <i>Set clock</i> 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 disrupted.
F460		Exiting state: Time synchronization by DCF77 clock OK.
F960	062B	Entering state: DCF77 clock radio signal disrupted.
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.
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 diagnostic server		
F560	0660	Error occurred installing the diagnostic server.
F360	0661	Installation of diagnostic 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.

4.2 Diagnostics

Event ID (hex)	Detailed event (hex)	Meaning
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 messages on
F260		Exiting state: Extended diagnostics messages off
F360	066B	Extended diagnostics messages - 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 blocked by CPU.
F260		Exiting state: I/O released by CPU.
F560	06B5	Module on P bus not capable of communication.
F960	06B6	Parity error in P bus communication.
F960	06B7	Bit shift error in P bus communication.
F360	06B8	Diagnostic interrupt cannot be sent. Module is not released on P bus.
F360	06B9	Diagnostic interrupt cannot be sent. Diagnostic interrupt is not released 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 to.
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 to.
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		

Event ID (hex)	Detailed event (hex)	Meaning
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 or 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.
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 for a header ID.
F560	0726	TD7-SDB: Unknown format for 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.

4.2 Diagnostics

Event ID (hex)	Detailed event (hex)	Meaning
F560	073E	TD7-Com: Negative acknowledgment of 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.
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.

Diagnostic messages and activation of the group error LED

If an error occurs during startup, the red group LED (SF) of the TIM lights up and a message to this effect is entered in the diagnostic buffer of the TIM.

If the TIM is installed as a CP in an S7-300 rack, a diagnostic 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 4-4 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

Error class	Detailed event (hex) in the diagnostics message
No parameter assignment	0070 0071 0620
Bad parameter assignment	0077 0081 0110 0112 0300 031C 0500 0620
RAM error	0085 0086

4.2.6.3 Meaning of ERRCLS and ERRCOD

Meaning of ERRCLS and ERRCOD in messages 0113 and 080C

Table 4-5 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: <ul style="list-style-type: none"> 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.

4.2 Diagnostics

ERRCLS	ERRCOD	Meaning
81	04	Context is not supported: <ul style="list-style-type: none"> 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.
85	00	Illegal PDU size: <ul style="list-style-type: none"> 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

4.2.7 SNMP

SNMP (Simple Network Management Protocol)

SNMP is a protocol for management and diagnostics of networks and nodes in the network. To transmit data, SNMP uses the connectionless UDP protocol.

The information on the properties of SNMP-compliant devices is stored in MIB files (MIB = Management Information Base).

Scope of performance of the module as SNMP agent

Not all functions described below are available with every module. Refer to the manual of the respective module for information on the functional scope.

The communications modules support data queries in the following SNMP versions:

- SNMPv1 (standard)
- SNMPv3 (Security)

They return the contents of MIB objects of the standard MIB II according to RFC1213.

- **MIB II**

The MIB supports the following groups of MIB objects:

- System
- Interfaces
The "Interfaces" MIB object provides status information about the module interfaces.
- IP
- ICMP
- TCP
- UDP
- SNMP

The following groups of the MIB II standard are not supported:

- Address Translation (AT)
- EGP
- Transmission

The modules do not support traps.

For more detailed information about the MIB files and SNMP, refer to the manual /20/ SNMP (Page 429).

Configuration

For information on the configuration, see section SNMP (Page 120).

4.2.8 Processing status of the messages (SMS / e-mail)

Processing status of messages

If the option "Enable identifier for processing status" option is set in the "Trigger" tab of the STEP 7 message configuration, the module outputs a status.

The processing status provides information about the processing state of the sent message. The status is written to a PLC tag of the type DWORD. Select this tag via the "PLC tag for processing status" box.

The processing status is returned by the module itself or the servers of the service after transfer of a message to be sent.

E-mails sent via program blocks of Open User Communication return a different status via the block (see block help).

The meaning of the statuses is as follows:

Processing status of the telecontrol messages

Table 4-6 SMS: Meaning of the status ID output in hexadecimal format

Status	Meaning
0000	Transfer completed free of errors
0001	Error in the transfer, possible causes: <ul style="list-style-type: none"> • SIM card invalid • No network • Wrong destination phone number (number not reachable)

Table 4-7 E-mail: Meaning of the status ID output in hexadecimal format

Status	Meaning
0000	Transfer completed free of errors
82xx	Other error message from the e-mail server Apart from the leading "8", the message corresponds to the three-digit error number of the SMTP protocol.
8401	No channel available. Possible cause: There is already an e-mail connection via the module. A second connection cannot be set up at the same time.
8403	No TCP/IP connection could be established to the SMTP server.
8405	The SMTP server has denied the login request.
8406	An internal SSL error or a problem with the structure of the certificate was detected by the SMTP client.
8407	Request to use SSL was denied.
8408	The client could not obtain a socket for creating a TCP/IP connection to the mail server.
8409	It is not possible to write via the connection. Possible cause: The communications partner reset the connection or the connection aborted.
8410	It is not possible to read via the connection. Possible cause: The communications partner terminated the connection or the connection was aborted.
8411	Sending the e-mail failed. Cause: There was not enough memory space for sending.
8412	The configured DNS server could not resolve specified domain name.
8413	Due to an internal error in the DNS subsystem, the domain name could not be resolved.
8414	An empty character string was specified as the domain name.
8415	An internal error occurred in the cURL module. Execution was aborted.
8416	An internal error occurred in the SMTP module. Execution was aborted.
8417	Requests to SMTP on a channel already being used or invalid channel ID. Execution was aborted.
8418	Sending the e-mail was aborted. Possible cause: Execution time exceeded.
8419	The channel was interrupted and cannot be used before the connection is terminated.
8420	Certificate chain from the server could not be verified with the root certificate of the module.
8421	Internal error occurred. Execution was stopped.
8450	Action not executed: Mailbox not available / unreachable. Try again later.

Status	Meaning
84xx	Other error message from the e-mail server Apart from the leading "8", the message corresponds to the three-digit error number of the SMTP protocol.
8500	Syntax error: Command unknown. This also includes the error of having a command chain that is too long. The cause may be that the e-mail server does not support the LOGIN authentication method. Try sending e-mails without authentication (no user name).
8501	Syntax error. Check the following configuration data: Alarm configuration > E-mail data (Content): <ul style="list-style-type: none"> • Recipient address ("To" or "Cc").
8502	Syntax error. Check the following configuration data: Alarm configuration > E-mail data (Content): <ul style="list-style-type: none"> • Email address (sender)
8535	SMTP authentication incomplete. Check the "User name" and "Password" parameters in the configuration.
8550	SMTP server cannot be reached. You have no access rights. Check the following configuration data: <ul style="list-style-type: none"> • Module configuration > E-mail configuration: <ul style="list-style-type: none"> – User name – Password – Email address (sender) • Alarm configuration > E-mail data (Content): <ul style="list-style-type: none"> – Recipient address ("To" or "Cc").
8554	Transfer failed
85xx	Other error message from the e-mail server Apart from the leading "8", the message corresponds to the three-digit error number of the SMTP protocol.

4.3 Maintenance

4.3.1 Loading firmware - CP 1200

New firmware versions of the CP

If a new firmware version is available for the module, you will find this on the Internet pages of Siemens Industry Online Support under the following address:

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21764/man>)

Note that firmware versions as of V3 cannot be loaded on CPs with hardware product version 1.

4.3 Maintenance

There are three different ways of loading a new firmware file on the CP:

- Saving the firmware file on the memory card of the CPU
You will find a description of the procedure for loading on the memory card of the CPU on the Internet page of the Siemens Industry Online Support.
- Loading the firmware with the online functions of STEP 7 via a WAN

Note

Effects on the retentive memory of the CPU

- If you use a SIMATIC memory card to install the firmware file, the retentive memory is retained.
 - If you use the online functions to install the firmware file, retentive memory is lost.
-

You can recognize that firmware is being loaded by the flashing LEDs of the CP, see /2/ TeleControl configuration manuals (Page 426).

Loading the firmware with the online functions of STEP 7 via a WAN

Requirements:

- The CP can be reached using its IP address.
- The engineering station and the CP are located in the same subnet.
- The new firmware file is stored on your engineering station.

Procedure:

1. Connect the engineering station to the network.
2. Open the relevant STEP 7 project on the engineering station.
3. Select the CP or the CPU of the station whose CP you want to update with new firmware.
4. Enable the online functions using the "Connect online" icon.
5. In the "Connect online" dialog, select the Ethernet interface "PN/IE" in the "Type of PG/PC interface" list box.
6. Select the slot of the CP or the CPU.
Both methods are possible.
7. Connect using the "Connect" button.

The "Connect online" wizard guides you through the remaining steps in installation.

You will find further information on the online functions in the STEP 7 information system.

4.3.2 Update firmware (TIM 3V-IE / TIM 3V-IE Advanced / TIM 4R-IE)

Validity

The information below applies to the following TIM modules for SIMATIC S7-300 / 400:

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

Sequence of the update

Different files are provided for the firmware update of these TIM modules:

- Firmware files
- Driver package

Note

Sequence of the update

Adhere to the correct update sequence for a firmware update:

1. **First update the driver package.**
2. **Then update the firmware files.**

If you update the firmware files first and inadvertently select the wrong file, the TIM module may no longer be accessible in certain circumstances.

Firmware update files of the TIM

The files for the firmware update are provided as compressed file archives. The archives contain files for the driver package and the firmware as well as readme files, which include information about the firmware version.

One archive is made available for each of the following TIM modules:

- TIM 3V archive for:
 - TIM 3V-IE
 - TIM 3V-IE Advanced
- TIM 4V archive for:
 - TIM 4R-IE
 - TIM 4R-IE Standalone

4.3 Maintenance

To save a file archive, follow these steps:

1. Copy the archive to the file system of your engineering station.
Specify the directory in the online dialog for the update process.

Recommendation:

Copy the archive for a firmware version of a module type (TIM 3V / TIM 4R) into a separate directory.

2. Unzip the archive after you have copied it into the file system of your engineering station.

4.3.3 Update firmware (TIM 1531 IRC)

New firmware versions of the TIM

If a new firmware version is available for the TIM, you will find this on the Internet pages of Siemens Industry Online Support:

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21764/cert>)

Firmware files have the file format *.upd.

Save the firmware file on your PC.

There are different ways of loading a new firmware file on the TIM:

- Loading the firmware via the WBM
- Loading the firmware with the online functions of STEP 7 via Ethernet / Internet
- Loading the firmware from an SD card

Note

SD card only for firmware file

For the firmware file you require a separate SD card. This must meet the requirements of the TIM, see /3/ TIM 1531 IRC (Page 426).

You cannot use the SD card with the configuration data.

Note

Duration of the firmware update

Downloading a new firmware file can take several minutes.

Always wait until the completion of the firmware update can be recognized from the LEDs (see below).

Loading the firmware via the WBM

Requirements:

The new firmware file is stored on your engineering station.

Procedure:

For a description of the procedure, see /3/ TIM 1531 IRC (Page 426).

Loading the firmware with the online functions of STEP 7 via Ethernet / Internet**Requirements:**

- The TIM can be reached using an IP address.
- The engineering station and the TIM are located in the same subnet.
- The new firmware file is stored on your engineering station.
- The engineering station is connected to the network.
- The relevant STEP 7 project is open on the engineering station.

Procedure:

1. Select the TIM that you want to update with new firmware.
2. Enable the online functions using the "Connect online" icon.
3. In the "Connect online" dialog, select the Ethernet interface in the "Type of PG/PC interface" list box.
4. Select the TIM.
5. Click on "Start search" to search for the module in the network and to specify the connection path.
When the module is found it is displayed in the table.
6. Connect using the "Connect" button.
The "Connect online" wizard guides you through the remaining steps in installation.
7. Select the TIM in the network view and select the "Online & Diagnostics" shortcut menu (right-click).
8. In the navigation panel of the Online & Diagnostics view, select the entry "Functions > Firmware update".
9. Using the "Browse" button (parameter group "Firmware loader") search for the new firmware file in the file system of the engineering station.
10. Start to download the firmware with the "Start update" button when the correct version of the signed firmware is displayed in the "Status" output box.

You will find further information on the online functions in the STEP 7 information system.

Loading the firmware via the SD card**Requirements:**

- You have copied the new firmware file from your PC to the SD card using a suitable card reader.
- You have saved a backup file of the firmware file currently being used.

Procedure:

1. Turn off the power supply to the TIM.
2. If you use an optional SD card for the configuration data, take the SD card out of the card slot of the TIM.
See also /3/ TIM 1531 IRC (Page 426) for information on this.
3. Insert the SD card with the firmware file in the card slot of the TIM.
4. Turn on the power supply to the TIM.
The TIM starts up with the new firmware from the SD card.
Wait until the CONNECT LED flashes.
5. Turn off the power supply to the TIM.
6. Remove the SD card with the firmware file from the card slot of the TIM.
If you use an optional SD card for the configuration data, plug this into the card slot of the TIM.
7. Turn on the power supply to the TIM again.
The TIM starts up with the new firmware and the configuration data on the SD card.
The LED pattern shows the end of the startup, see below.

LED patterns when updating the firmware

The table below describes the LED patterns when transferring a firmware file to the TIM.

Table 4-8 Meaning of the LED patterns

RUN (left)	ERROR (center)	CONNECT (right)	LED pattern / meaning
			The firmware file is transferred. <ul style="list-style-type: none">• "RUN" flashes alternately green and yellow.• "ERROR" flashes.• "CONNECT" flashes.
			Startup successfully completed, start of the firmware.

For the LED pattern during the restart, refer to the section /3/ TIM 1531 IRC (Page 426).

4.3.4 Reset to factory settings (TIM 1531 IRC)

Resetting to factory settings: Effect

Note

Configuration data is deleted

With the functions for resetting to factory settings described here, all configuration data on the TIM is deleted!

- **Deleted data**

The following data is deleted by resetting to factory settings:

- Configured IP addresses of the LAN interfaces X1 X2 and X3
- All other configuration data in the work memory of the TIM

- **Data not deleted**

The following data is not deleted by resetting to factory settings:

- MAC addresses of the LAN interfaces

Note

Configuration data when using an SD card

If you use an SD card and you want to reset the TIM to the factory settings, you must pull the SD card before resetting. If the SD card remains inserted, the TIM starts up again with the configuration data on the SD card.

When pulling the SD card, please note: /3/ TIM 1531 IRC (Page 426)

Executing the "Reset to factory settings" function

To reset to factory settings using the switch, follow the steps outlined below:

1. Put the switch to the position "MRES" and hold it in the "MRES" position for at least 5 seconds.
The TIM restarts.
2. During the restart, continue to hold the switch in the "MRES" position.
If the LED pattern of step 2 occurs (see below) the TIM is reset.

You can find information on the switch settings in /3/ TIM 1531 IRC (Page 426).

LED patterns when resetting

The table below describes the LED patterns when resetting the TIM.

Table 4-9 Meaning of the LED patterns

Step	LED pattern			Meaning
	RUN	ERROR	CONNECT	
1				After holding the switch in the "MRES" position for 5 seconds, chaser lights of the three LEDs appear (flashing alternately). The TIM restarts. Continue to hold switch.
2	  			As soon as the three LEDs flash three times synchronously the TIM is reset. It adopts the factory settings. You can release the button:

Startup after resetting to factory settings

After resetting return the switch of the TIM back to the "RUN" position.

The TIM restarts again. For the LED pattern, see /3/ TIM 1531 IRC (Page 426).

When resetting using the WBM, the TIM starts up again automatically.

The remaining behavior depends on the use of an optional SD card:

- **Startup without SD card**

If you do not use an SD card, the TIM starts up without configuration data with an error. You need to reload the configuration data.

The TIM can be reached via its Ethernet interfaces with default IP address (see above) set in the factory.

Generally, the default settings apply during initial commissioning; see /3/ TIM 1531 IRC (Page 426).

- **Startup with SD card**

If the SD card remains inserted, the TIM starts up again with the configuration data on the SD card.

4.3.5 Module replacement (TIM 1531 IRC)

Startup with configuration data from the SD card

Depending on the use of an optional SD card, the STEP 7 configuration data of the TIM are stored differently.

- **No use of an SD card**

The configuration data of the TIM is stored in the work memory of the TIM.

- **Use of an SD card**

If an SD card is used, the configuration data is stored on the SC card and read from there each time the TIM starts up.

If the TIM needs to be replaced, this allows simple replacement of the TIM without needing to download the configuration data again.

If you insert the SD card of the TIM being replaced in the new TIM, when it restarts it reads the configuration data from the SD card of the replaced TIM.

STEP 7 V5 configuration of the proxy

Note

Validity

The following sections on configuration in STEP 7 V5 apply only when you wish to expand an existing SINAUT ST7 system with the following communications modules.

Module (firmware version):

- TIM 3V-IE (V2.6)
- TIM 3V-IE Advanced (V2.6)
- TIM 4R-IE (V2.6)
- CP 1243-8 IRC (V3.x)
- TIM 1531 IRC
 - (V1.1) - STEP 7 Professional V15
 - (V2.x) - STEP 7 Professional as of V15.1
- CP 1542SP-1 IRC (V2.x)

You will find details on the modules in the section Modules for new SINAUT projects and those to be expanded (Page 41).

A.1 Configuration in STEP 7 V5

Configuration of a proxy in STEP 7 V5

1. Open the STEP 7 V5 project that you wish to expand with S7-1200/1500 modules.
2. Create the stations with CPU and the new networks, if necessary.
The CPU only serves as placeholder to receive the SINAUT subscriber number.
 - For a PROXY CP1243-8 IRC:
Create an S7-300 station with CPU.
 - For a PROXY TIM 1531 IRC:
Create an S7-300 rack for the proxy.
Create an S7-400 station with CPU.
The proxy is assigned to the S7-400 CPU like a TIM 4R-IE Stand-alone.
3. Insert the "proxy" in the station.
4. Perform the configuration in STEP 7.
Deviations from the normal configuration are described below.

5. Perform the configuration in the connection configuration of the SINAUT engineering software.
6. Export the configuration data of the proxy via the SINAUT diagnostics and service tool. For information on this, refer to section Expansion of SINAUT projects (Page 41). The data exported from STEP 7 V5 is adopted later in the STEP 7 Basic project for the S7-1200/1500 communications module.

A.2 Proxies in HW Config

PROXY modules in HW Config

You can find the proxy modules as representatives of S7-1200/1500 communication modules in the HW Config catalog under:

SIMATIC 300 > SINAUT ST7 > TIM IE

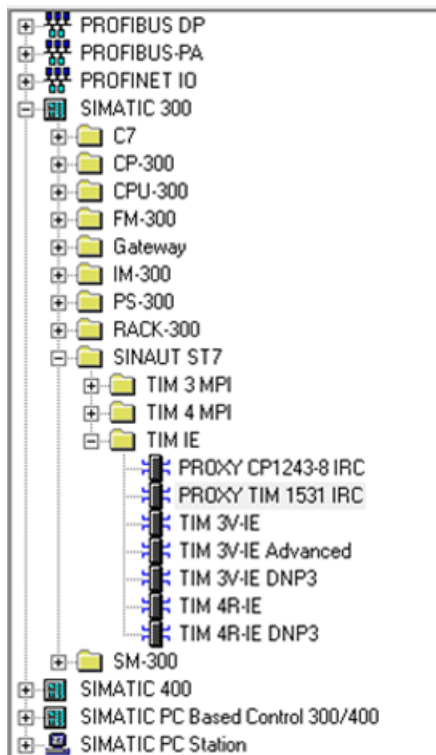


Figure A-1 PROXY modules in the catalog

The proxy modules can be used to import STEP 7 V5 configuration data into the following modules:

- **PROXY CP1243-8 IRC**
 - CP 1243-8 IRC
 - CP 1542SP-1 IRC
- **PROXY TIM 1531 IRC**
 - TIM 1531 IRC

A.3 Special features of the PROXY TIM 1531 IRC

Properties of the TIM or the "PROXY TIM 1531 IRC"

Compared to TIM 4R-IE, the TIM 1531 IRC and PROXY TIM 1531 IRC have the following differences:

- **Interfaces**

Similar to the TIM, the proxy has the following interfaces:

 - 3 Ethernet interfaces
Ethernet interface X1 can be configured for using the MSC protocol.
 - 1 serial WAN interface
The WAN interface can also be configured as an Ethernet interface for using the MSC protocol.

You perform the configuration in the "Interfaces" tab as for TIM 4R-IE. The "Modem type" and "Connection mode" parameters correspond to those of the TIM 4R-IE.

A.4 Special features of the PROXY CP1243-8 IRC

Properties of the CP or proxy "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. Both interfaces of the proxy can be configured.

Compared with the TIM 3V-IE Advanced, the proxy or CP has the following differences:

- **Station**

The interfaces of the proxy can only be configured as station. Use as a node station or master station is not possible. This is expressed in the STEP 7 properties dialog:
- **"Time" tab**

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

- **"Interfaces" tab**
 - Modem type
TS modules are added here due to the different connection of WAN transmission devices to the CP 1243-8 IRC.
 - Connection mode
When the CP 1243-8 IRC is used as successor module with imported configuration data, it can only be used as station. This changes the selection of the connection modes; see below.
- **AT strings for TS modules**
AT strings were added for the TS modules, see below.
- **"Special " tab**
The tab no longer exists.
- **"Options" tab**
The tab no longer exists.
- No loading of firmware
The loading of firmware files on a CP 1243-8 IRC is not supported.

"Interfaces" > "Connection mode" tab

Options of the "Connection mode" parameter for both interfaces:

- **Ethernet interface**
 - 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.
- **Serial WAN interface**
 - 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.

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.

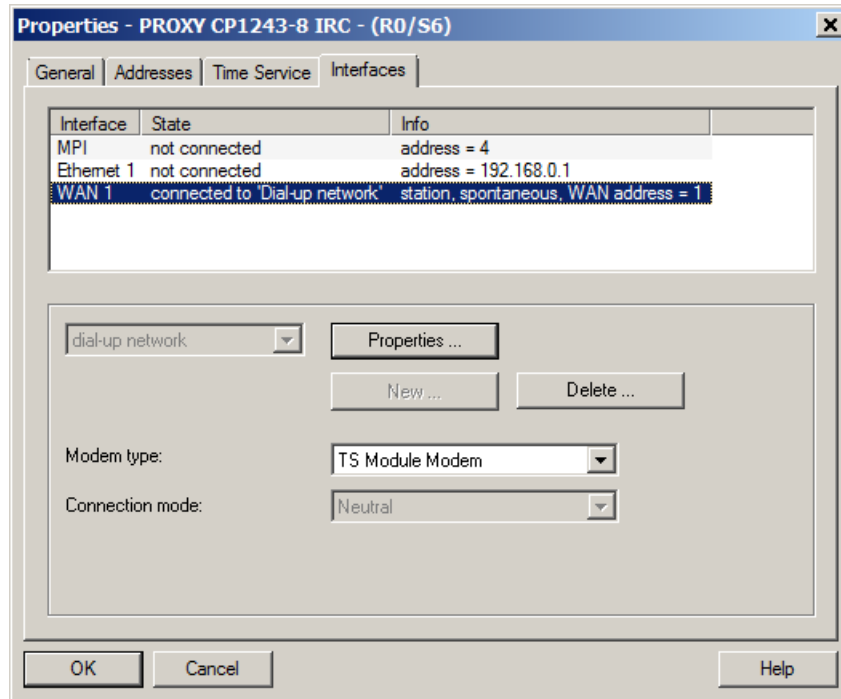


Figure A-2 Properties dialog, "Interfaces" tab. WAN interface connected to dialup network.

The following figure shows drop-down list for selecting the modem type.

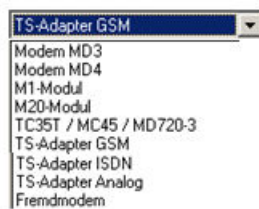


Figure A-3 Selectable modem types of a WAN interface with a connection to a dial-up network

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 RS232 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.

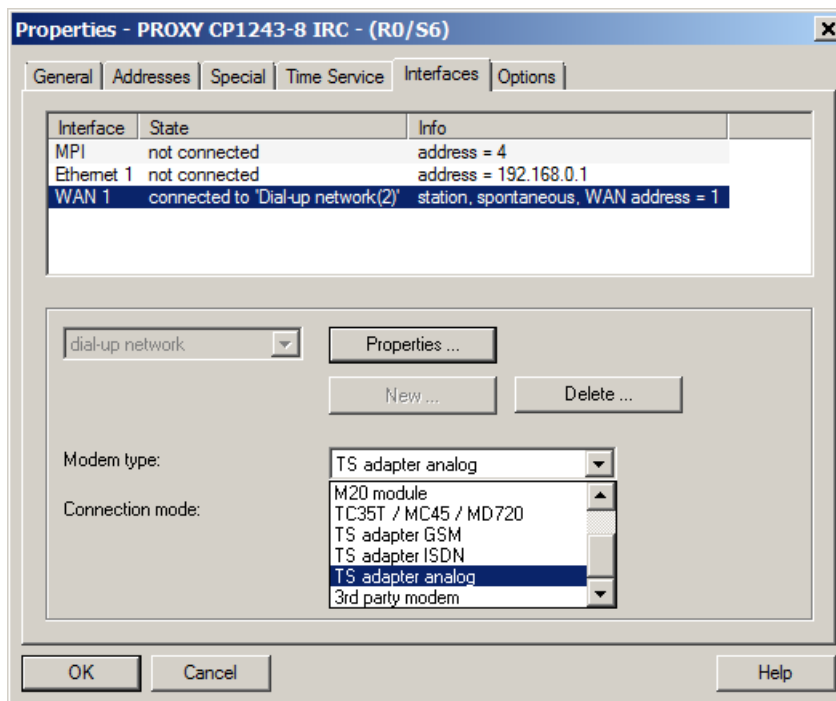


Figure A-4 Properties dialog of a dial-up network node, "AT initialization" tab

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

A.5 SINAUT configuration

Connection configuration with SINAUT Engineering Software

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

In the subscriber administration of the configuration tool, a proxy appears as follows:

Subscriber no.	Subscriber type	Module	Station	SINAUT connected	SINAUT library
☐ 1	CPU 314	CPU 314	Master	yes	
☐ 2	CPU 314	CPU 314	CP1243-8 Station	yes	
☐ 3	CPU 314	CPU 314	TIM3VIE-Adv Station	yes	
☐ 1001	SINAUT TIM	TIM 4R-IE	Master	yes	TD7 library / TIM
☐ 1002	SINAUT TIM	TIM 3V-IE Advanced	CP1243-8 Station	yes	TD7 library / TIM
☐ 1003	SINAUT TIM	TIM 3V-IE Advanced	TIM3VIE-Adv Station	yes	TD7 library / TIM
☒ 1004	SINAUT TIM	PROXY CP1243-8	CP1243-8 Station	yes	

Figure A-5 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:

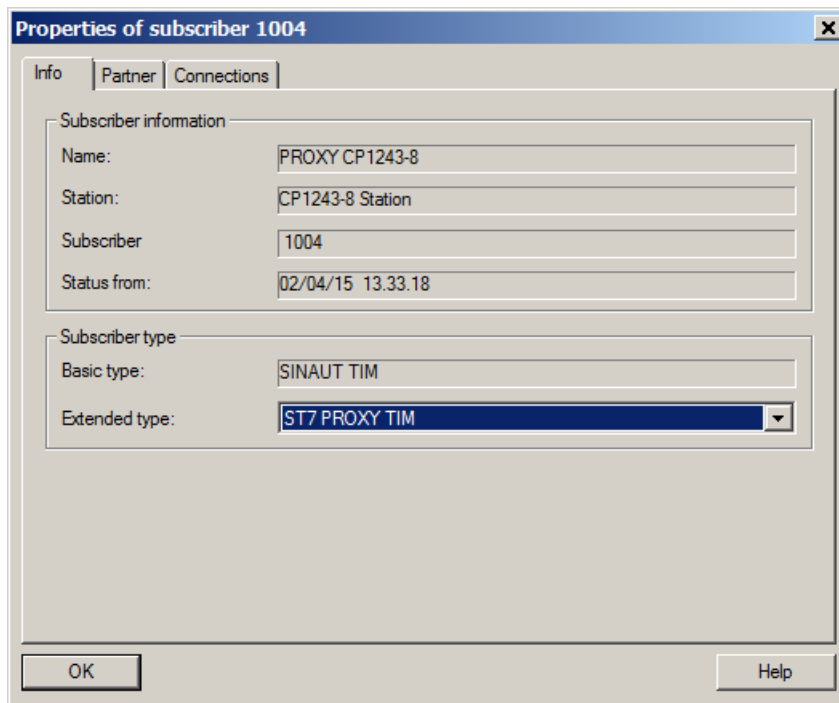


Figure A-6 Expanded type of the proxy in the Properties dialog of the subscriber
Note that neither TD7onCPU nor TD7onTIM is configured for the proxy.

A.6 Exporting configuration data

Exporting the configuration data using SDB text files

After completing the configuration of the proxy in STEP 7 V5 and in the SINAUT configuration tool, the specific configuration data for the telecontrol communication of the proxy is stored in system data blocks (SDBs) just as with TIM modules.

Follow the steps below to export the configuration data of the proxy:

1. Open the SINAUT diagnostics and service tool with the relevant project.
2. Select the proxy.

- Open the menu "SINAUT" > "SDB display".

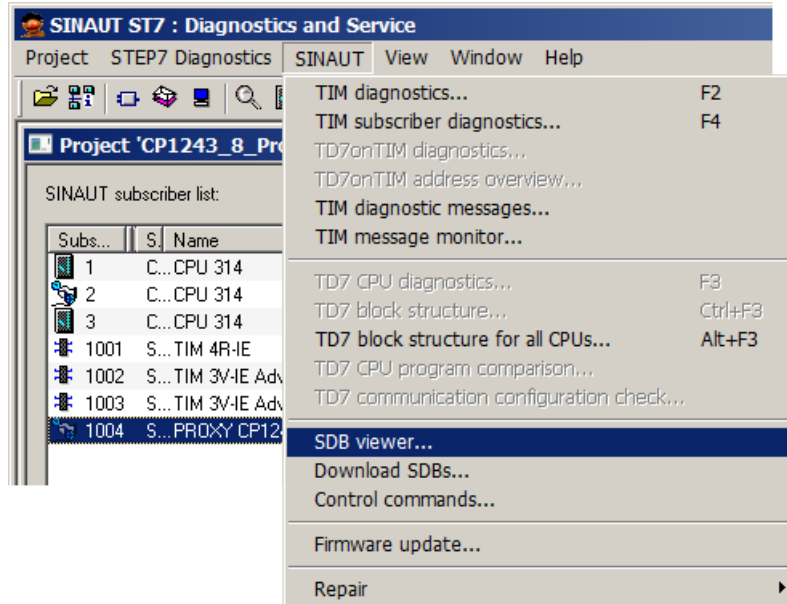


Figure A-7 Opening the "SDB display" dialog

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.

- Click the "Save" button.
The "Save as" dialog opens.

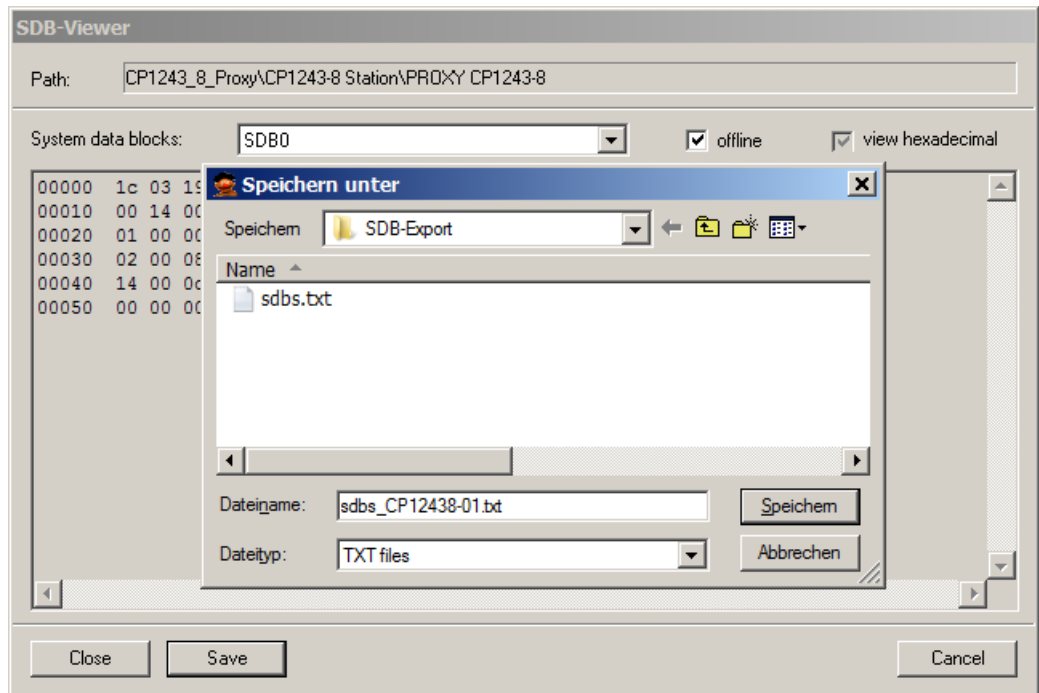


Figure A-8 Export of the configuration data of a proxy in the example as the file "sdb_CP12438-01.txt".

A.6 Exporting configuration data

5. In the file directory of the configuration PC/PG, select a suitable directory for the file of the configuration data.
6. Select a unique name for the file of the configuration data of this proxy.
Retain the default file type TXT file.
7. Click "Save".
A text file with the data of all SINAUT SDBs is saved in the file directory.

Note

TXT file for STEP 7 in the TIA Portal.

You then need the exported text file to import it into the respective successor module in STEP 7 Basic / Professional.

OUC program blocks (CP)

The functions described below are supported by the following modules:

- CP 1243-8 IRC
- CP 1542SP-1 IRC

B.1 Program blocks for OUC

Using the program blocks for Open User Communication (OUC)

You can use the instructions (program blocks) listed below for direct communication between S7 stations.

In contrast to telecontrol communication, Open User Communication does not need to be enabled in the configuration because the corresponding program blocks need to be created for this. You will find details on the program blocks in the information system of STEP 7.

Note**No different program block versions**

Note that you cannot use different versions of a program block in a station.

Requirements for Secure OUC

Requirements for the use of the secure transmission via Secure OUC:

- STEP 7: As of V16
- CPU firmware
 - CPU-1200: As of V4.4
 - CPU 151xSP: As of V2.0
- CP firmware
 - CP 1200: As of V3.2
 - CP 1542SP-1 IRC: As of V2.1

Supported program blocks for OUC

The following instructions in the specified minimum version are available for parameter assignment of the Open User Communication:

- **TSEND_C V3.0 / TRCV_C V3.0**

Compact blocks for:

- Connection establishment/termination and sending data
- Connection establishment/termination and receiving data

As an alternative, use:

- **TCON V4.0 / TDISCON V2.1**

Connection establishment / connection termination

- **TUSEND V4.0 / TURCV V4.0**

Sending and receiving data via UDP

- **TSEND V4.0 / TRCV V4.0**

Sending and receiving data via TCP or ISOonTCP

- **TMAIL_C V4.0**

Sending e-mails

To transfer encrypted e-mails with this block, the precise time of day is required on the module. Configure the time-of-day synchronization.

Note

Personal data unencrypted in TMAIL_C

Note that TMAIL_C uses personal data, such as user name and password, in its parameters. This data is stored unencrypted in a data block.

To change the configuration data of the module during runtime:

- **T_CONFIG V1.0**

Program-controlled configuration of the IP parameters

Refer to the information on T_CONFIG and on the SDTs "IF_CONF_..." in the section Changing the IP address during runtime (Page 416).

Note

No feedback from the CP

"T_CONFIG" does not support feedback from the CP to the CPU. Errors in the block call or in setting the address parameter are not reported. The block outputs "BUSY" or "DONE" regardless of whether the address parameter was set.

The address parameters can only be configured with temporary validity. In the respective "IF_CONF_..." SDT, the "Mode" = 2 parameter must be set.

- **TC_CONFIG**

Program-controlled change of configuration data of mobile wireless CPs

You can find the program blocks in STEP 7 in the "Instructions > Communication > Open User Communication" task card.

Connection descriptions in system data types (SDTs)

The blocks listed above use the CONNECT parameter for the relevant connection description. TMAIL_C uses the parameter MAIL_ADDR_PARAM.

The connection description is stored in a data block whose structure is specified by the system data type (SDT).

Creating an SDT for the data blocks

Create the SDT required for every connection description as a data block (global DB).

The SDT type is not created by selecting an entry from the "Data type" drop-down list in the declaration table of the block, but by entering the name manually in the "Data type" box, for example "TCON_IP_V4".

The corresponding SDT is then created with its parameters.

Usable SDTs

- **TCON_IP_V4**
For transferring frames via TCP or UDP
- **TCON_QDN**
For TCP or UDP communication via the fully qualified domain name (FQDN) (IPv4 / IPv6)
- **TCON_IP_RFC**
For transferring frames via ISO-on-TCP (direct communication between two S7 stations)
- **TADDR_Param**
For transferring frames via UDP
- **TMail_V4**
For transferring e-mails addressing the e-mail server using an IPv4 address
Recommendation for mobile wireless applications:
Set the parameter "WatchdogTime" from "MAIL_ADDR_PARAM" to a value higher than 3 minutes.
- **TMail_V6**
For transferring e-mails addressing the e-mail server using an IPv6 address
- **TMail_FQDN**
For transferring e-mails addressing the e-mail server using its name (FQDN)
- **IF_CONF**
For changing configuration data of mobile wireless CPs using the TC_CONFIG
- **TCON_IP_V4_SEC**
Only CP 1200
For the secure transfer of data via TCP
- **TCON_QDN_SEC**
Only CP 1200
For the secure transfer of data via the host name
- **TMail_V4_SEC**
For secure transfer of e-mails addressing the e-mail server using an IPv4 address

B.2 Changing the IP address during runtime

- **TMail_V6_SEC**
For secure transfer of e-mails addressing the e-mail server using an IPv6 address
- **TMail_QDN_SEC**
For secure transfer of e-mails addressing the e-mail server using the host name

Note on TMail_Vx_SEC / TMail_QDN_SEC:

With these SDTs, the mail server certificate is checked, but not the ID of the "TLSServerCertRef" (STEP 7 internal reference) certificate.

You will find the description of the SDTs with their parameters in the STEP 7 information system under the relevant name.

Connection establishment and termination

Connections are established using the program block TCON. Note that a separate program block TCON must be called for each connection.

A separate connection must be established for each communications partner even if identical blocks of data are being sent.

After a successful transfer of the data, a connection can be terminated. A connection is also terminated by calling TDISCON.

Note

Connection abort

If an existing connection is aborted by the communications partner or due to disturbances on the network, the connection must also be terminated by calling TDISCON. Make sure that you take this into account in your parameter assignment.

B.2 Changing the IP address during runtime

Changing the IP address during runtime

You can change the following address parameters of the CP at runtime controlled by the program:

- IP address
- Subnet mask
- Router address

Apart from the address parameters of the CP, with T_CONFIG the address parameters of DNS servers (IF_CONF_DNS) and NTP servers (IF_CONF_NTP) can also be changed program controlled.

Note**Changing the IP parameters with a dynamic IP address**

Note the effects of program-controlled changes to the IP parameters if the CP obtains a dynamic IP address from the connected router: In this case, the CP can no longer be reached by communications partners.

Requirements - Configuration

To be able to change the IP parameters program-controlled, the option "IP address is set directly at the device" must be enabled in the configuration of the IP address of the Ethernet interface of the CP.

Requirements - STEP 7 version

- STEP 7 $\geq$ V14

Requirements - Firmware versions

- CP 1243-1 / CP 1243-8 IRC
 - CP firmware $\geq$ V2.1.7x
 - CPU firmware $\geq$ V4.2
- CP 1542SP-1 IRC
 - CP firmware $\geq$ V1
 - CPU firmware $\geq$ V2.0 (CPU 151xSP)

Program blocks

Program-controlled changing of the IP parameters is supported by program blocks. The program blocks access address data stored in a suitable system data type (SDT).

The following program blocks and system data types can be used:

- **T_CONFIG**

Along with:

- IF_CONF_V4
- IF_CONF_NTP
- IF_CONF_V6
- IF_CONF_DNS

The address parameters can only be configured with temporary validity in the CP. In the respective "IF_CONF_..." SDT, the "Mode" = 2 parameter must be set.

Note

No feedback from the CP

"T_CONFIG" does not support feedback from the CP to the CPU. Errors in the block call or in setting the address parameter are not reported. The block outputs "BUSY" or "DONE" regardless of whether the address parameter was set.

You can find detailed information on parameter assignment of the blocks and SDTs in the STEP 7 information system.

SINEMA Remote Connect (CP)

C.1 Validity and requirements

Validity

Communication via SINEMA Remote Connect is supported by the following modules:

- CP 1243-1
 - As of firmware V3.1
- CP 1243-7 LTE
 - As of firmware V3.1
- CP 1243-8 IRC
 - As of firmware V3.1
 - Under ST7 as MSC station as of firmware V3.2
- CP 1542SP-1 IRC
 - As of firmware V2.0
 - Under ST7 as MSC station as of firmware V2.1

The functions are supported by the following software versions:

- SINEMA Remote Connect
 - As of software version V1.3

Under SINAUT ST7, the CPs specified above support SINEMA Remote Connect as MSC station.

C.2 Connection to SINEMA RC

Communication via SINEMA Remote Connect (SINEMA RC)

The "SINEMA RC Server" application provides end-to-end connection management of distributed networks via the Internet. This also includes secure remote access to lower-level stations. Communication between SINEMA RC Server and the remote devices takes place via a VPN tunnel with consideration of the stored access rights.

SINEMA RC uses OpenVPN for encryption of the data. The center of the communication is SINEMA RC Server via which communication runs between the subscribers and that manages the configuration of the communications system.

SCALANCE M routers, which you can use for the connection, also support OpenVPN and connection to SINEMA Remote Connect.

The CP can also handle telecontrol communication via the SINEMA RC server.

Parameter groups

You configure communication via SINEMA RC and telecontrol communication via SINEMA RC in two parameter groups:

- Communication via SINEMA RC:
 > "Security > VPN"
- Telecontrol communication via SINEMA RC:
 > "Communication types"

For information on the supported protocols and configuration, see section Telecontrol via SINEMA RC (Page 421).

Applications

The following application options of the CP result from the combination of the parameters for telecontrol communication and SINEMA RC:

- (1) No telecontrol and no SINEMA RC (CP for network separation only)
- (2) CP only for remote maintenance via SINEMA RC
- (3) CP for telecontrol communication only
- (4) CP uses telecontrol communication, but SINEMA RC only for remote maintenance.
- (5) CP uses SINEMA RC for telecontrol communication and remote maintenance.

The table provides an overview of the applications with the respective parameter settings.

- "On" means that the parameter is activated.
- "Off" means that the parameter is deactivated.

Table C-1 Use cases and parameters to be activated

Use case	Parameter settings (Parameters abbreviated) *		
	SRC	TC	TC-SRC
(1)	Off	Off	Off
(2)	On	Off	Off
(3)	Off	On	Off
(4)	On	On	Off
(5)	On	On	On

* Explanation of the parameter abbreviations:

SRC - Security > VPN (activated) > "VPN connection type":

"Automatic OpenVPN configuration via SINEMA Remote Connect Server"

TC - Communication types > Telecontrol communication enabled

TC-SRC - Communication types >

"Activate telecontrol communication via SINEMA Remote Connect"

C.3 Telecontrol via SINEMA RC

For information on possible applications of communication via SINEMA Remote Connect, see section Connection to SINEMA RC (Page 419).

Requirements

Perform the necessary configuration of SINEMA Remote Connect - Server (not in STEP 7) before configuring the CP in STEP 7. The CP and the communications partner of the CP must be configured in the SINEMA RC Server.

Configuration of the telecontrol communication via SINEMA Remote Connect

Follow the steps below when configuring the module for use of telecontrol communication via SINEMA RC:

1. In the "Communication types" parameter group activate telecontrol communication and select the protocol.
The option for communication via SINEMA RC is not yet visible.
2. Change to the "Security" parameter group and enable the security functions.
(In the "Communication types" parameter group the SINEMA RC option appears disabled and grayed out)
3. Open the "Security > VPN" parameter group and enable VPN.
4. For the parameter "VPN connection type" select the option "Automatic OpenVPN configuration via SINEMA Remote Connect Server" if this is not preset.
(In the "Communication types" parameter group the SINEMA RC option becomes usable.)
5. Change to the "Communication types" parameter group and enable the option "Telecontrol communication via SINEMA Remote Connect".
6. Create the remaining configuration of the SINEMA RC connection of the CP under "Security > VPN".
For information on the configuration, see section Security > VPN > SINEMA Remote Connect (Page 421).

C.4 Security > VPN > SINEMA Remote Connect

Remote maintenance with SINEMA Remote Connect (SINEMA RC)

The application "SINEMA Remote Connect" (SINEMA RC) is available for remote maintenance purposes.

SINEMA RC uses OpenVPN for encryption of the data. The center of the communication is SINEMA RC Server via which communication runs between the subscribers and that manages the configuration of the communications system.

Preparatory steps

Execute the following steps before start configuring the SINEMA RC connection of the module in STEP 7. They are the prerequisite for a consistent STEP 7 project.

- **Configuration of SINEMA Remote Connect Server**
Configure SINEMA RC Server as necessary (not in STEP 7). The communications module and its communications partners must be configured in the SINEMA RC Server.
- **Exporting the CA certificate (optional)**
If you want to use the server certificate as authentication method of the communications module during connection establishment, export the CA certificate from SINEMA RC Server. Then import the CA certificate from SINEMA RC Server to the engineering station. Alternatively, you can use the fingerprint of the server certificate as authentication method of the communications module. The fingerprint's duration of validity may be shorter than that of the certificate.
Please note that you need to repeat the import of a certificate in the event of a module replacement.

Configuration of SINEMA Remote Connect

Importing your own certificate

1. On the CP, navigate to the parameter group "Security > Certificate manager > Certificates of the partner devices".
2. Open the certificate selection dialog with a double-click on the first free table row.
3. Select the CA certificate of SINEMA RC Server.

Then navigate to the parameter group "Security > VPN".

VPN > General

1. Activate VPN
2. As "VPN connection type", select the option "Automatic OpenVPN configuration via SINEMA Remote Connect Server" if you wish to use communication via SINEMA Remote Connect.

SINEMA Remote Connect Server

Enter the address and port number of the server.

Server Verification

Here you select the authentication method of the communications module during connection establishment.

- **CA Certificate**
Under "CA certificate", select the CA certificate from SINEMA RC Server that was previously imported and assigned in the local certificate manager.
The module generally checks the CA certificate of the server and its validity period. The two options cannot be changed.
- **Fingerprint**
When you select this authentication method, you enter the fingerprint of the server certificate of SINEMA RC Server.

Authentication

- **Device ID**
Enter the device ID generated for the module in SINEMA RC.
- **Device password**
Enter the device password of the module configured in SINEMA RC.
Max. number of characters: 127

Optional settings

The connection establishment is configured in the "Security > VPN > Optional settings" parameter group with the parameter "Connection type".

- **Update interval**
With this parameter you set the interval at which the CP queries the configuration on the SINEMA RC Server.
Note that with the setting 0 (zero) changes to the configuration of the SINEMA RC Server may result in the CP no longer being capable of establishing a connection to the SINEMA RC Server.
- **"Connection type"**
The two options of the parameter have the following effect on the connection establishment:
 - **Auto**
The module establishes a connection to the SINEMA RCServer. The OpenVPN connection is retained until the connection parameters are changed by the SINEMA Remote Connect Server. If the connection is interrupted, the CP automatically re-establishes the connection.
If the connection parameters are changed by the SINEMA Remote Connect Server, the CP requests the new connection data after the update interval configured above has elapsed.
 - **PLC trigger**
The option is intended for sporadic communication of the module via the SINEMA RC Server.
You can use this option when you want to establish temporary connections between the module and a PC. The temporary connections are established via a PLC tag and can be used in servicing situations, for example.

Note**Connection abort**

With a STOP of the CPU, for example due to a firmware update or "Download to device", the OpenVPN connection is aborted.

These functions can only be used when the "Auto" option is enabled.

- **PLC tag for connection establishment**
If the option "PLC trigger" is selected, the module establishes a connection when the PLC tag (Bool) changes to the value 1. During operation the PLC tag can be set when necessary, for example using an HMI panel.
When the PLC tag is reset to 0, the connection is terminated again.

References

Where to find Siemens documentation

- Article numbers
You will find the article numbers for the Siemens products of relevance here in the following catalogs:
 - SIMATIC NET - Industrial Communication / Industrial Identification, catalog IK PI
 - SIMATIC - Products for Totally Integrated Automation and Micro Automation, catalog ST 70You can request the catalogs and additional information from your Siemens representative. You will also find the product information in the Siemens Industry Mall at the following address:
Link: (<https://mall.industry.siemens.com>)
- Manuals on the Internet
You will find SIMATIC NET manuals on the Internet pages of Siemens Industry Online Support:
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/15247/man>)
Go to the required product in the product tree and make the following settings:
Entry type "Manuals"
- Manuals on the data medium
You will find manuals of SIMATIC NET products on the data medium that ships with many of the SIMATIC NET products.

D.1 /1/ SINAUT ST7 system manual

SIMATIC NET
SINAUT ST7
System Manual
- Volume 1: System and hardware
- Volume 2: Configuration in STEP 7 V5
- Volume 3: Configuration in STEP 7 Professional
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21771/man>)

D.2 /2/ TeleControl configuration manuals

SIMATIC NET - TeleControl

Siemens AG

Configuration manuals of the protocols:

- TeleControl Basic

- SINAUT ST7

- DNP3

- IEC 60870-5

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21764/man>)

D.3 /3/ TIM 1531 IRC

SIMATIC NET

TIM 1531 IRC

Manual

Siemens AG

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

D.4 /4/ CP 1243-8 IRC

SIMATIC NET

CP 1243-8 IRC

Operating Instructions

Siemens AG

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21162/man>)

D.5 /5/ CP 1542SP-1 IRC

SIMATIC

CP 1542SP-1 IRC

Operating Instructions

Siemens AG

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/22143/man>)

D.6 /6/ RTU3000C

SIMATIC

RTU3000C

Operating instructions

Siemens AG

Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21767/man>)

D.7 /7/ SINAUT ST7cc

SIMATIC NET
SINAUT ST7cc
manual
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/15927/man>)

D.8 /8/ SINAUT ST7sc

SIMATIC NET
SINAUT ST7sc
manual
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/15929/man>)

D.9 /9/ SCALANCE M870

SIMATIC NET
Mobilfunkrouter SCALANCE M870
(M873 / M874 / M875 / M876)
Operating Instructions
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/15987/man>)

D.10 /10/ SCALANCE S

SIMATIC NET
Industrial Ethernet Security
SCALANCE S
Commissioning and Installation Manual
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/15327/man>)

D.11 /11/ S7-1200 system manual

SIMATIC
S7-1200 Automation System
system manual
Siemens AG
Link: (<http://support.automation.siemens.com/WW/view/en/34612486>)

D.12 /12/ ET 200SP system manual

SIMATIC
ET 200SP - Distributed I/O System
system manual
Siemens AG
Link: (<http://support.automation.siemens.com/WW/view/en/58649293>)

D.13 /13/ ET 200SP Manual Collection

SIMATIC
ET 200SP
Manual Collection
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/view/84133942>)

D.14 /14/ S7-300 automation system

SIMATIC S7
S7-300 Automation System
Siemens AG

- CPU 31xC and 31x Installation: Operating instructions
Link: (<https://support.industry.siemens.com/cs/ww/en/view/13008499>)
- Module Data: Reference manual
Link: (<https://support.industry.siemens.com/cs/ww/en/view/8859629>)

D.15 /15/ MODEM MD720

SIMATIC NET
MODEM MD720
Operating Instructions
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/21820/man>)

D.16 /16/ TS Adapter

SIMATIC
TS Adapter modular
Device Manual
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/16006>)

D.17 /17/ MODEM MD2

SIMATIC NET
MODEM MD2
Operating Instructions
Siemens AG
Link: (<http://support.automation.siemens.com/WW/view/en/17163799>)

D.18 /18/ MODEM MD3

SIMATIC NET
MODEM MD3
Operating Instructions
Siemens AG
Link: (<http://support.automation.siemens.com/WW/view/en/17164329>)

D.19 /19/ MODEM MD4

SIMATIC NET
MODEM MD4
Operating Instructions
Siemens AG
Link: (<http://support.automation.siemens.com/WW/view/en/17165032>)

D.20 /20/ SNMP

SIMATIC NET
Diagnostics and configuration with SNMP
Diagnostics manual
Siemens AG
Link: (<https://support.industry.siemens.com/cs/ww/en/ps/15392/man>)

Glossary

1-out-of-8 check

Mechanism for interlocking multiple simultaneous commands.

When entering commands, there is a check to determine if only one command is pending at the time of acquisition. Transmission of the command byte is only triggered if there is a single modified command bit in the command byte compared to the last cycle. If several bits within the command byte have been changed, errors are detected and the command byte is not sent.

The function is performed by the data point typical "Cmd01B_S" of the TD7onCPU block library. The "FC Safe" block is also required.

1-out-of-n check

Mechanism for interlocking multiple simultaneous commands.

When entering commands, there is a check to determine if only one command is pending at the time of acquisition. Transmission of the command to the communication partner is only triggered when a single command is pending. If several commands are pending at the same time, errors are detected and the command is not sent or not issued by the receiver.

The function is executed by the "FC Safe" block of the TD7onCPU block library. The function is supported by the data point typicals "Cmd01B_S", "Par12D_S" and "Set01W_S".

Accelerated general request

→ *General request*

APN

Access Point Name

DNS host name of the access point for an external network (in this case: access point in the GPRS network to the Internet).

Conditional spontaneous frame

→ *Spontaneous / conditional spontaneous / unconditional spontaneous frame*

CP

Communications processor

Module for expanded communications tasks that provides the CPU with additional interface types or communications options.

CPU

Central Processing Unit

Main processor of a SIMATIC controller

CSD

Circuit Switched Data

Service for transferring data in the GSM network. Possible are dial-in connections of GSM modems to GSM/ISDN/analog modems and other devices with modems. The transmission speed is 14400 bps full duplex for non-secure transmission and 9600 bps for secure transmission.

CTS

Clear to send

Signal in the data flow control

Data frame

Data unit transferred between communication partners. Meaning:

- Data unit transferred on the application layer (OSI layer 7)
- General term for a transferred data unit regardless of the relevant OSI layer.

Data frames contain the data of an ST7 object to be transmitted. Depending on the object type, a frame can contain either all data of an ST7 object or a contiguous subarea of the object data.

See also "Organizational frame"

Direct communication

With direct communication, the S7 stations communicate directly with each other without the frames needing to be forwarded by a master station or station.

See also "Inter-station communication"

DNP

Distributed Network Protocol

DSL

Digital Subscriber Line

Standards for transmission of telephone and Internet data with transmission speeds up to 1000 Mbps.

EGPRS

Enhanced GPRS

Packet-oriented service for IP-based data transmission in GSM networks. By using an additional modulation procedure (EDGE technology), a higher transmission speed is achieved compared with GPRS.

Engineering station

PC with the STEP 7 Professional project (TIA Portal)

Frame

→ *Data frame*

General request

- **General request (GR)**
With a general request, a central station requests the current process image from the connected subscribers.
Depending on the telecontrol protocol used, a GR can be started for a variety of reasons. With ST7, a GR is started automatically when a disrupted connection has been restored or when a failed partner reports a restart. Apart from the automatic general request, a GR can also be triggered by the user program or from the control center. TD7onTIM does not support general requests.
- **Accelerated general request (XGR)**
In an accelerated general request, the frames with the requested process image are entered at the start of the send buffer of the TIM; in other words, before any other frames still buffered in the TIM. This enables you to have the current process image available more quickly.

GPRS

General Packet Radio Service

Packet-oriented service for IP-based data transmission in GSM networks. The data is transmitted using the Internet protocols TCP/IP or UDP/IP.

GSM

Global System for Mobile Communication

Worldwide standard for mobile communication (2G)

Image memory

Memory area for the process image in a telecontrol module

Each data frame is saved exactly one time in the image memory. New values of a data point overwrite the existing value in the image memory.

See also send buffer

Image memory / send buffer principle

- **Image memory principle**

A fixed position is reserved in the image memory for each data frame transferred to the TIM for transmission. Each newly transferred frame always overwrites the old frame in the image memory.

If a send frame is entered using the image memory principle, only a reference to the location of the frame in the frame image memory is entered. If the TIM has not yet been able to transmit the frame when the same frame is transferred to it again, the frame is not entered in the send buffer a second time, but rather the image is simply updated.

At the time of transmission, the frame is sent with its up-to-date content from the image memory. Only then can the frame be entered in the send buffer again.

Transmission using the image memory principle achieves the following:

- There is less load on transmission link, fewer frames are transmitted.
- There is less load on the send buffer of the TIM; an image memory frame is entered a maximum of once in the send buffer.

- **Send buffer principle**

If a data frame is transmitted using the send buffer principle, it is entered completely in the send buffer each time it is transferred to the TIM. If such a frame cannot or should not be transmitted immediately, it may therefore exist more than once in the send buffer.

When it is sent, the frame is taken completely from the send buffer and transmitted.

Inter-station communication

Communication between two stations, which is mediated by a Telecontrol master station.

In dial-up networks, a direct connection between the two stations is established.

See also "Direct communication".

IRC

Industrial Remote Communication

SIMATIC NET product group for Telecontrol

ISDN

Integrated Services Digital Network

Standard for a digital transmission network for telephone, telefax, telex, teletext and datex-J/L/P services. The data of various services can be transferred simultaneously. Telephone connections normally operate at transmission speeds of 56 to 64 Kbps.

LAN

Local Area Network

Local network, usually "Industrial Ethernet".

Local CPU

CPU assigned to a TIM.

Local TIM

TIM that is connected to a CPU or a PC (ST7cc, ST7sc) via an IP-based network.

LTOP

Line Transformer with Overvoltage Protection

Overvoltage protection module of the SINAUT device program - discontinued

Main cycle / sub-cycle

The sequence of the polling cycle can be structured on the master TIM by assigning individual polling stations to the main cycle or the sub-cycle.

The subcycle is always activated at the end of the main cycle once all stations from the main cycle have been polled. A configurable number of stations is called in a subcycle.

Following this, all the stations in the main cycle are polled again. This is followed by a subcycle in which further stations that are assigned to the subcycle are called.

Master station

Station in the top hierarchy of a telecontrol network. It is connected to the control system and the substations or node stations.

The interfaces of a master module are set to the network node type "Master station".

MCC - Mobile Country Code

→ *PLMN*

Messages

Emails and SMS in the TeleControl context

See also Data frame.

MNC - Mobile Network Code

→ *PLMN*

MPI

Multi Point Interface

MPI is the programming device interface of the SIMATIC S7-300/400. Devices such as the TIM can communicate with each other via the MPI interface.

See also Partyline.

MSC

The MSC transmission protocol is a proprietary protocol on OSI layer 3 for the secure communication via Ethernet, landline or mobile wireless networks in SINAUT ST7. The MSC protocol provides an authentication mechanism and simple encryption of data.

The protocol is available in the MSC and MSCsec versions (with cyclic key exchange).

MSCsec

→ MSC

Multi-master polling with time slots

When stations need to communicate with more than one master station in dedicated line or wireless operation, the multimaster polling with time slots mode is used. Each of the connected master stations is assigned one or more defined time slots per minute for polling the stations. The master stations then have their turn to poll in every minute.

Node station

A node station is a station located between the master station and stations in the hierarchy of a telecontrol network. One or more subordinate stations are connected to a node station. The data traffic between these stations and the master station is handled via the node station. Direct data exchange between the node station and the subordinate stations is also possible. Multiple node station levels are possible in a SINAUT network.

Organizational frame

Organizational frames are used to execute organizational system functions, for example:

- General requests
- Time-of-day synchronization
- Counted value storage
- Coordinated connection establishment and termination in a dial-up network
- Message indicating station startup and station failure
- Requests for and transmission of subscriber records

Party line

- Party line CPUs are:
 - CPU 312/313/314/315 to CPU 315-2 DP
 - C7 devices

The communication bus of the smaller S7-300 CPUs is physically wired through to the MPI interface of the CPU.

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>

- Non-party line CPUs are:
 - CPU 315-2 PN/DP to CPU 319-3 PN/DP

With non-party line CPUs, the MPI interface and communications bus are separate.

Permanent call

A permanent call does not interrupt the normal polling cycle; it is always executed alternating with the standard poll from the normal polling cycle.

PG

Programming device

Allows access by the STEP 7 configuration software to the SIMATIC CPU.

PG routing

Using PG routing, it is possible to access programmable modules or modules with diagnostics capability beyond network boundaries from a programming device (PG) or computer (PC).

PLMN

Public Land Mobile Network

Worldwide unique identifier of mobile networks. The PLMN is made up of the three-digit Mobile Country Code (MCC) and the two-or three-digit Mobile Network Code (MNC) of the network provider.

Polled frames

Polled data frames are data frames of a station or node TIM with a special identifier indicating that they were sent in response to a general request from the master station.

Polling

→ *Polling mode*

Polling mode

The polling mode is a method of data transmission in which a central instance controls the data exchange with the communication partners.

Using a polling frame, the master TIM instructs the connected station TIMs one after the other to transmit their stored data frames to the master TIM. If a polled station has no stored data, it responds with an acknowledgment frame and the polling cycle then continues by polling the next station.

A station that has stored data sends a single data frame or, if block transfer was configured, several data frames in a block.

If the TIM has stored additional data, it indicates this in the response frame. In this case, the station is then immediately called up again until the stored data has been transferred.

Polling with time slots

The polling with time slots mode is used in a wireless network in which the use of the radio frequency assigned by the registration authorities must be shared with other users. Each user typically has 6 seconds per minute to exchange data with its stations. The frequency must then be released for other operators. During the allocated time slot, this pooling variant functions like a normal polling system.

Context: SINAUT ST7

Protocol

A protocol is a set of rules for controlled transfer of data. Protocols, for example, specify the data structure, the structure of data packets and the coding. Protocols can also specify a control mechanisms and hardware and software requirements.

RS-232

RS-232 is a standard for serial (bit-by bit) data transmission with +12 V and -12 V signals. RS-232 is a Recommended Standard of the Electronic Industries Association. 9-pin and 25-pin connections with D-sub connectors (subminiature connector with D-shaped surface area) are normal for the RS-232 interface.

RS-485

RS-485 is a standard for data transmission with 5 V differential signals. The RS-485 interface uses only one pair of wires and is operated in half duplex. The connection is multipoint-compliant; in other words, up to 32 subscribers can be connected.

RTS

Request to send

Signal in the data flow control

S₀ interface

Basic interface of ISDN for connecting end devices

Send buffer principle

→ *Image memory / send buffer principle*

SIM card

SIM - Subscriber Identity Module

The SIM card is an identification card for a subscriber of a mobile wireless service.

Simple Internet communication

In SINAUT ST7, simple Internet communication means data exchange between TCP/IP-compliant devices in Ethernet, landline or mobile networks using the MSC protocol.

SINAUT

Siemens Network Automation

Station control system or telecontrol system based on SIMATIC S7.

SINAUT ST7 works with the SINAUT ST7 telecontrol protocol.

SINAUT object

A SINAUT object contains the data of one or more process variables such as analog values, commands, calculated values, status information on motors, sliders etc. An ST7 object has type-specific processing functions and change checks assigned to it to minimize the communication traffic in the WAN. Type-specific processing functions include, for example, threshold checks or mean value calculation with the object type for analog values. The change check is designed so that a frame is generated only when the object data has changed compared with the last time it was transferred.

SINAUT ST7

Proprietary telecontrol protocol for SIMATIC NET telecontrol modules

SINAUT ST7cc

Control center system based on SIMATIC WinCC for SINAUT ST7.

SINAUT ST7sc

System for networking SIMATIC stations with a control station via WAN. The control center can also be a SIMATIC station or a PC-based control center, for example, WinCC with the SINAUT ST7cc add-on.

SINAUT TD7 Library

Software for control of ST7 communication of telecontrol modules. The TD7 software in the stations allows change-controlled transmission of process data between the individual CPUs and the control center, for example ST7cc. Failure of connections, CPUs, or the control center are displayed. Once a problem has been corrected or the CPUs or control center have started up, data is updated automatically. Data frames can be given a time stamp, if required.

The following variants of the TD7 software exist:

- **TD7onCPU**

Program blocks in the CPU user program

The SINAUT TD7 library consists of program blocks for the CPU. They are available in the following versions:

- Library for STEP 7 V5

The blocks are executable on S7-300- and S7-400-CPU (except CPU 400H). There are only a few blocks intended specifically for the S7-300 or S7-400.

- Libraries for STEP 7 Professional

There is a global library with two versions for STEP 7 projects in the TIA Portal:

- Blocks for S7-1500

- Blocks for S7-300 and S7-400

TD7onCPU is not supported when using the DNP3 and IEC 60870-5 protocols.

- **TD7onTIM**

Configurable part of the firmware of the communications module

TD7onTIM can be used as an alternative to TD7onCPU for an Ethernet TIM. TD7onTIM runs on the communications module and is configured as follows:

- STEP 7 V5: In the SINAUT engineering software

- STEP 7 Professional: Via the data points of the communications module

With CPs (S7-1200 / ET 200SP), TD7onTIM is the only variant that can be selected.

TD7onCPU and TD7onTIM cannot be used simultaneously in a station.

SMS

Short Message Service

The short message service in the GSM standard is used to transfer short text messages to mobile wireless subscribers.

When the short messages are transferred, they are first transferred to the SMS center (SMSC) using a store-and-forward technique. They are buffered there and then forwarded to the recipient. The sender can query the status of the message in the SMS center or can request acknowledgment of delivery.

SMSC

Short Message Service Center

When sending an SMS message, the message is first sent to the SMSC, buffered there and then forwarded to the recipient.

Spontaneous / conditional spontaneous / unconditional spontaneous frame

- **Spontaneous frame**

In SINAUT networks, data frames are always transmitted spontaneously; in other words, data are created and transmitted only when changes to process values occur or event-controlled. These frames are known as spontaneous frames.

- **Conditional spontaneous frame**

In the dial-up network, you can specify whether or not a change causes a "conditional spontaneous" or "unconditional spontaneous" transmission for each individual frame. Conditional spontaneous frames 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 frame needs to be transmitted or because the partner calls.

Even when using pay by volume transmission in a GPRS network, frame prioritization "conditional spontaneous" can also be used. Such a frame is not transmitted immediately, but is first buffered. In a GPRS network, the TIM stores "conditional spontaneous" frames in the following situations:

- When the collected frames reach or exceed a size of 202 bytes.
- An important frame ("unconditional spontaneous" or "spontaneous" priority) should be transmitted immediately.
- The collected frames have not yet reached a volume of 202 bytes, but the TCP/IP keep-alive interval expires.
- The fill level of the send buffer has reached 90% of its maximum capacity.

- **Unconditional spontaneous frame**

In the dial-up network, you can specify whether or not a change causes an "conditional spontaneous" or "unconditional spontaneous" transmission for each individual frame. Unconditional spontaneous frames cause the connection to be established immediately. Even with pay-by-volume/time transmission in a GPRS network, you can use the frame prioritization "unconditional spontaneous"; in other words, in contrast to a "conditional spontaneous" frame, a frame is transmitted immediately.

Spontaneous mode

Spontaneous mode is a method of ST7 data transmission in which subscribers can exchange data directly amongst themselves. 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 IP-based networks (for example GPRS), the data to be sent is assigned different priorities during configuration ("high" / "normal").

When data with high priority is ready for transmission, a connection is established immediately.

If the data has "normal" priority, it is first stored in the communications module. It is sent the next time a connection is established to the partner. This can be the case, for example, when data is to be transferred with high priority or when the partner establishes a connection.

ST7 protocol

→ *SINAUT ST7*

ST7cc

→ *SINAUT ST7cc*

ST7sc

→ *SINAUT ST7sc*

Station

- Hardware
SIMATIC controller with the required components for acquisition, processing and communication, consisting of: CPU, I/O modules, communications module, modem, etc.
- Network node type
Setting a WAN interface of the TIM. An interface of the "Station" network node type communicates at the lowest hierarchy level in a SINAUT network.

Subcycle

→ *Main cycle / sub-cycle*

TA

Frame type

- 0 = Spontaneous Org. frames
- 1 = Requested Org. frames
- 2 = Spontaneous data frames
- 3 = Requested data frames

TD7 software

→ *SINAUT TD7 Library*

TIM

Telecontrol Interface Module

Communications module that handles all data transmission functions provided by the SINAUT system independently.

Unconditional spontaneous frame

→ *Spontaneous / conditional spontaneous / unconditional spontaneous frame*

VPN

Virtual Private Network

Technology for secure transportation of confidential data in public IP networks, for example the Internet.

WAN

Wide Area Network

Data network with a large geographical span, such as the Internet. telephone or enterprise networks. We distinguish between the following WAN networks:

- **WAN, classic**
Includes dedicated lines (private or leased), private wireless networks, analog telephone network, digital ISDN network and mobile networks (without Internet).
A classic WAN is connected via suitable transmission device (modem) to a serial interface of the TIM.
- **WAN, IP-based**
Includes IP-based networks with telecontrol communication via wireless or fiber-optic cables, public networks and the Internet using services such as DSL, GPRS, UMTS or LTE or via broadband systems such as OTN and PCM30.
An IP-based WAN is normally connected to an RJ45 interface of a module via an Ethernet-capable module.

Index

1

1-out-of-n error, 104

A

Abbreviations, 4
Accelerated general request, 37
Assign CPU, 117
Assigned CPU, 117

C

Classic WAN, 61, 82
CloudConnect, 5
Conditional spontaneous, 169
Connection resources, 31
Connection status - diagnostics, 356
Connections - Number, 26
Cross references (PDF), 6

D

Data buffering, 28
Data points - Configuration, 147
Device replacement, 402
Direct communication, 24
Documentation - Structure, 5

E

E-mail, 24
 Configuration, 183

F

Forced image mode, 161, 366
Frame memory, 28

G

Gateway (VPN), 125
General request, 37, 355
General request supervision time, 146
GET, 25

Glossary, 9
GPRS, 21
GR, 37

H

H system, 34

I

Image memory, 161
Importing a certificate - e-mail, 116
Internet communication, 21
Inter-station communication, 24
IPsec, 125

M

Main path, 18
Manuals - others, 21
MIB, 392
Mirroring, 156
MSC master station, 21
MSC protocols, 21
MSC station, 21
MSC tunnel, 21

N

Node station, 18

O

Online functions, 54, 358
OUC (Open User Communication), 413

P

Partner status - diagnostics, 356
Passive VPN connection establishment, 125
Path redundancy, 18
Port 8448, 359

R

READ, 25

Redundancy
 H system, 34
Redundant telecontrol connection, 18
Redundant telecontrol connections, 144
Replacement situation, 402
Replacing a module without a programming device, 59
RS-485
 Configuration, 72, 75

S

SD card, 28, 57
SDBs, 410
Security diagnostics, 359
Security functions, 49
Send buffer, 28, 161
Service & Support, 9
SIMATIC NET glossary, 9
SMS, 24
 Configuration, 183
SMTPS, 116
SNMP, 392
SNMPv3, 120, 129
Spontaneous, 169
SSL/TLS, 116
ST7cc (connection), 146
ST7sc (connection), 146
STARTTLS, 116
Station address, 83
STEP 7 - version, 38
Subscriber number, 19
Substitute path, 18
SYSLOG, 125
System data blocks, 410

T

TD7onCPU, 33
TD7onTIM, 33
Time stamp, 113, 146
Training, 9
Transmission mode, 169
Trigger tag - resetting, 167
TS module
 configuring, 80

V

VPN, 125

W

WAN - creating a network, 60, 81
WAN address, 19, 83
Web server, 70
WRITE, 25