









S7-Compact-PLC Manual

-  S7-Panel-PLC
-  S7-Compact-PLC
-  S7-Panel-HMI
-  Periphery
-  Software
-  Energy Management

Index of contents

General instructions.....	7
About INSEVIS.....	8
Product family S7-Compact-PLC.....	9
Communication overview.....	9
Technical data of INSEVIS-S7-CPU's.....	10
CPU-V.....	10
CPU-T.....	11
Decentral touch panel units for the Compact-PLCs.....	12
Periphery in the Compact-PLCs.....	13
Communication to PLC.....	13
Software in the Compact-PLCs.....	14
Accessories for S7-Compact-PLCs.....	15
Wiring of the S7-Compact-PLCs.....	16
Wiring of CPU-V in devices with/without periphery slots.....	16
Wiring of CPU-T in devices with/without periphery slots.....	17
General.....	18
Memory organization.....	18
Communication.....	19
Ethernet.....	19
Set up IP-address in the Simatic® -Manager.....	20
Set up the CPU and the Ethernet connection in the Simatic®-Manager.....	22
IP Configuration of PLC with SFB 129 "IP_CFG".....	23
Modbus TCP (Server).....	23
S7 active connection to Partner-PLCs (RFC1006).....	24
Read data from a Remote PLC with SFB 120 "GET".....	24
Write data to a Remote PLC with SFB 121 "PUT".....	26
Read in IP configuration with SFB 131 "IP_GET" (CPU-T).....	28
Set IP-configuration with SFB 132 "IP_SET" (CPU-T).....	29
TCP Send/Receive Connection (RFC 793).....	30
Send data via TCP with SFB 122 "TSEND".....	30
Receiving data via TCP with SFB 123 "TREC".....	32
Terminate the TCP connection with SFB 124 "TDISCON".....	34
Configure the partner parameter by TCP with SFB 125 "TCONFIG".....	35
Query the Status of TCP connection with SFB 126 "TSTATUS".....	36
UDP Send/Receive Connection (RFC 768).....	37
Send data via UDP with SFB 127 "USEND".....	38
Receiving data via UDP with SFB 128 "URECV".....	39
Profinet IO Controller.....	42
Configuration.....	42
Status-LEDs.....	44
Profibus DP.....	45
Profibus DP-Master.....	45
Configure the DP-Master.....	45
Profibus DP-Slave.....	46
Configure the DP-Slave.....	46
DP Master configuration for INSEVIS Profibus DP-slave.....	48
Structure of DP-Slave diagnostic data.....	49
DP-Slave address assign with SFC 75 „SET_ADDR“.....	52
DP-Slave address and baud rate assign with SFC 76 „SET_DPS“.....	53
UART.....	54
Sending data with SFB 60 "SEND_PTP", free ASCII.....	54
Receive data with SFB 61 "RCV_PTP", free ASCII.....	55
Sending data with SFB 60 "SEND_PTP", Modbus RTU (Client).....	56
Receive data with SFB 61 "RCV_PTP", Modbus RTU (Client).....	57
Modbus RTU (Server).....	58
Receive data via RS232 with defined End Code.....	59
Change the UART-configuration by SFB 130 "CFG_PTP".....	59
CAN.....	60
CANopen® Configuration with the ConfigStage.....	60
Send CAN telegrams with SFB105.....	62
Receive CAN telegrams with SFB106.....	62
Receive event OB106.....	62
SDO transfer SFB107.....	63
FB107 SDO upload.....	63
FB108 SDO download.....	63
FB109 SDO upload.....	63
FB110 SDO download.....	64

FB111 SDO upload	64
FB112 SDO download	64
FB113 SDO transfer string	64
Alternative function of SFB107.....	64
CAN status request SFB114.....	65
CAN error event OB186.....	65
Automatic return.....	65
Configuration with the Software „ConfigStage“	67
Basic settings.....	67
Assign the IP-address.....	68
Change target device.....	68
Addressing of the onboard periphery.....	68
Standard addressing ind the INSEVIS- PLCs.....	68
Different addressing in PLCs and addressing of decentral periphery.....	69
CPU settings.....	69
WebServer.....	71
Communication settings.....	72
RS232 and RS485.....	72
Ethernet.....	72
Profinet.....	74
CAN configuration.....	74
Decentral INSEVIS periphery.....	74
Configure decentral external periphery manually.....	75
Create library elements of your CAN-slaves.....	76
Configure decentral external peripherie by EDS-file.....	77
Decentral external periphery / motion controller pre defined.....	78
Samples for external CANopen devices.....	79
System functions.....	81
Organisation blocks (OB).....	81
Organisation blocks (OB) - compatible to STEP@7 from Siemens.....	81
Organisation blocks (OB) - in addition to STEP@7 from Siemens.....	81
System function blocks (SFB).....	81
System function blocks (SFB) - compatible to STEP@7 from Siemens.....	81
System function blocks (SFB) - what are not available at INSEVIS-CPU's yet or available on request.....	81
System function blocks (SFB) - in addition to STEP@7 from Siemens.....	82
System functions (SFC).....	82
System functions (SFC) - compatible to STEP@7 from Siemens.....	82
System functions (SFC) - what are not available at INSEVIS-CPU's yet or only available on request	83
System functions (SFC) - in addition to STEP@7 from Siemens.....	83
System functions (SFC) - in addition to STEP@7 from Siemens.....	84
Import of INSEVIS-blocks into the S7-Project.....	84
Installation of the INSEVIS-S7-library in the Simatic@-manager.....	84
Installation of the INSEVIS-S7-library in the TIA-Portal.....	85
Overview about SFCs for data backup and restore.....	86
Save data blocks to ROM/ into flash memory.....	86
Life-backup for a change/ spare device.....	86
Manual program-backup for manufacturers.....	86
S7-Program-restore for both kinds of backup.....	86
Archive/ dearchive (write back) data.....	87
Read in data from PC-edited CSV-archive file.....	87
Archiving DBs with SFC 207 "ARCHIVE"	87
Backup & Restore with the SFC 208 "BACKUP".....	91
Read data from archive file (SD) with SFC 209 "READFILE".....	93
Copy data blocks from work memory to load memory with SFC 210 "COPY_DBL".....	95
Copy data blocks from load memory to work memory with SFC 211 "COPY_DBW".....	96
Read data from CSV-formatted archive file (SD)with SFC 213 „Read_CSV“.....	97
Overview about SFCs for a File Management System.....	99
Create folders on Micro-SD-Card with SFC 223 „CREATE“	100
Copy files on Micro-SD-Card with SFC 222 „COPY“.....	101
Rename files / folders on Micro-SD-Card with SFC 220 „RENAME“	102
Inform about files / folders on Micro-SD-Card with SFC 224 „INFO“.....	103
Inform about file size and name on Micro-SD-Card with SFC 225 „FILEINFO“.....	104
Delete files / folders on Micro-SD-Card with SFC 221 „DELETE“	106
Adjust LCD brightness with SFC 214 "LCD_BRIG"	107
Copy RAM to ROM with SFC 254 "RAM2ROM"	108
Update operation system in the PLC.....	109
General reset by hardware.....	110

Remote access with the software „RemoteStage“	111
General.....	111
Selection of the remote-device.....	111
Insert PIN code.....	112
Using and setting up the remote screen	112
Storing the remote (on the PC) archived data.....	112
Uploading, converting and storing archive data as csv-files.....	113
Display and export event and alarm archives.....	113
Recipe handling.....	115
Data (DB) archive upload, display, storing and download.....	116
Command line of RemoteStage (Viewing mode).....	117
Automatic connecting to remote panel after starting PC.....	117
Automatic full screen after starting PC.....	117
Set up TCP port number for S7-communication.....	117
Command line of RemoteStage (Archive mode).....	118
Read in (upload) message- and trend archives from remote device.....	118
Read in (upload) recipe archive from remote device.....	119
Write back (download) recipe archive into remote device.....	120
Read in (upload) data (DB-) archive from remote device.....	121
Write back (download) data (DB-) archive into remote device.....	121
Return codes.....	122
Working with the software „ServiceStage“	123
Function overview.....	123
General information.....	123
Operating mode.....	123
Date and time.....	123
Memory.....	124
Diagnostic buffer.....	124
Block Update.....	124
Creating backup files (Online Backup).....	125
Know-how-protection.....	125

Changes to older versions of the manual

Rev. 02 / 2012:

new: SFB130 (Modbus), SFC75 (Profibus DP-Slave), SFC254 (RAM to ROM)
 changed: SFB125/126 (TCP)-description, SFB127/128 (UDP)-description,
 Profibus-description, SFB/SFC overview

Rev. 03 / 2012:

changed: SFB60/61 (UART) parameter description, alarm handling, ConfigStage completely reworked

Rev. 04 / 2012:

changed: description of Profibus-signals made compatible to Siemens-manuals,
 explanation of cycling time added (ConfigStage-CPU settings)

Rev. 01 / 2013:

new: Assigning an IP-address with ConfigStage 1.0.14.5 and PLC-firmware (operating system/OS) 2.0.35
 changed: description of Ethernet UDP-Receive (SFB128), Profibus-Slave Subnets

Rev. 02 / 2013:

new: ServiceStage added

Rev. 3 / 2013:

new: Memory organization added, chapter CAN: more information

Rev. 04 / 2013:

new: New description for ConfigStage: PDO-/ SDO-Mapping, address overview list (from ConfigStage 1.0.14.7)

Rev. 05 / 2013:

new: New PLC-function: SFC206: read/ write recipe record into PLC with S7, SFC207: archiving DBs
 (with PLC-OS 2.0.41 and HMI-OS 1.0.11)

Rev. 06 / 2013:

new: Service Stage 1.0.1.1: Know-how-protection available, ConfigStage 1.14.9 with EDS-Import,
 SFC208 Backup&Restore (with OS-firmware 2.1.0 and S7-Lib 2_1_0)
 changed: description of CAN SFB114

Rev. 07 / 2013:

changed: description of embedding S7-lib into S7-project moved to system functions

Rev. 08 / 2013:

changed: description of SFB60/61 UART improved, ConfigStage 1.0.14.10 „Change device“ function added

Rev. 01 / 2014:

new: from PLC-firmware 2.1.5 SFC209 READFILE, SFC2010 LOAD_DBL, SFC2011 LOAD_DBW new
 changed: Backup&Restore- resume in chapter System functions added

Rev. 02 / 2014:

changed: S7-program lines as sample inserted for all SFCs and Ethernet-SFBs, Hints for VIDEOS added

Rev. 03 / 2014:

changed: Improved description of SFB 120 Get and SFB 121 Put in one connection,
 Improved description of SFB 208 (Saving password file)

Rev. 01 / 2015:

changed: Description of TSAPs improved, are changeable now, with firmware 2.2.1 and ConfigStage 1.0.14.15

Rev. 02 / 2015:

changed: System functions: better overview OBs, SFBs, SFCs, explanation how to create a WLD-file
 UART: Description of RS232 receive data with defined end code

Rev. 03 / 2015:

new: CPU-T-inserted, adaption (extensions) for CPU-T functionality inserted,
 changed: New ServiceStage functionality Firmware update by ETH, description: how to make a WLD,
 (with PLC-Firmware 2.3.1, ConfigStage 1.0.14.21, VisuStage 2.0.4.7, ServiceStage 1.0.1.5, RemoteStage 1.0.4.13)

Rev. 01 / 2016:

new: Profinet handling added, UART-, CAN-, Ethernet chapter debugged,
 changed: System functions: Description of SFC208 increased, image inserted for manual backup

Rev. 02 / 2016:

new: SFC 214 LCD_BRIG added in system functions
 changed: description for manual backup changed

Rev. 03 / 2016:

new: new up- and downloads in command line mode of RemoteStage 1.0.4.17, online-backup in Service Stage 1.0.1.7
 from PLC firmware 2.3.9 (all CPUs) or HMI firmware 1.2.7 CPU-V/-P and 1.2.6 CPU-T

Changes to older versions of the manual

Rev. 04 / 2016:

changed: RemoteStage: Description at download recipes and data archives into remote-device improved
ServiceStage: Description at Online Backup expanded

Rev. 05 / 2016:

changed: Description of Hardware-reset improved, ServiceStage: Description at Online Backup expanded once more
Design changed

Rev. 01 / 2017:

new: SFC215 to change user level by S7-program, SFC76 to set address AND baud rate of Profibus-DP-slave
ConfigStage 1.0.14.32: Webserver for all CPU-T-PLCs, RemoteStage 1.0.4.23,
firmwares: Compact-/Panel-PLCs with CPU-V/-P 2.4.5, Compact-/Panel-PLCs with CPU-T 2.4.8

Rev. 02 / 2017:

new: Ethernet; info for communication assignments for S7-1200/1500 – settings in TIA-Portal explained
changed: CAN-chapter: table with length and baud rate added

Rev. 01 / 2018:

changed: description of RemoteStage batch file handling improved

Rev. 02 / 2018:

new: SFC 220...224 for file management system on Micro-SD-Card added (system functions)

Rev. 01 / 2019:

new: SFB 131 IP_GET and 132 IP_SET for 2 separated Ethernet ports in CPU-T-PLCs
changed: new screenshots from 2019-Stage software inserted

Rev. 02 / 2019:

new: Fixed and variable local port at Ethernet-TCP, ConfigStage from 1.0.14.39, from firmware: CPU-V/P 2.4.9 / CPU-T 2.6.2

Rev. 03 / 2019:

new: SFC 225 "FINEINFO" for PLCs with CPU-T, from firmware: CPU-T 2.6.6

Hint for better understanding by application videos

In the English YouTube-channel INSEVIS En we supply different playlists with handling videos for single details referring to functions, described in this manual. This will help you to get familiar with INSEVIS much faster – PLEASE use it beside this manual!

Suchen

YouTube DE

INSEVIS

for independent minds

INSEVIS En

ÜBERSICHT VIDEOS PLAYLISTS KANÄLE DISKUSSION KANALINFO

Kostenlose Remotevisualisierung erstellen [INSEVIS Video-Tu...]

453 Aufrufe • vor 10 Monaten

WEBSITE: <http://www.insevis.de/>

Willkommen bei INSEVIS
INSEVIS-Kunden haben völlig neue Möglichkeiten für einen nachhaltigen Marktvorsprung ihrer eigenen Produkte. Und das alles mit den vorhandenen bekannten S7-Programmierungstools. Dabei bieten INSEVIS-Produkte die Chance auf technischen MEHR INFOS

ENGLISH

INSEVIS En

ABONNIEREN

BELIEBTE KANÄLE

Getting started ALLE WIEDERGEBEN

Free Visu

For all (!) S7-CPU's 2:48

Free Visualisation for all S7 CPUs [INSEVIS Video-]

INSEVIS En

First Steps

#1 2:38

S7-project (S7-Classical)

INSEVIS En

First Steps

#2 1:48

Importing a S7-library into the TIA-Portal [INSEVIS]

INSEVIS En

First Steps

#3 3:36

Library import S7 classic [INSEVIS Video-Tutorials]

INSEVIS En

First Steps

#4 1:23

Changing the IP-address in the TIA-Portal [INSEVIS]

INSEVIS En

General instructions

Safety instructions

This manual contains instructions to avoid material damage and must be carefully attended for your own safety. These instructions are identified with a warning triangle with a note of exclamation inside and a signal word (*Signalwort*) below.

Danger Death, heavy bodily harm or material damage will appear, if appropriated precautions are not taken over.

Warning Death, heavy bodily harm or material damage will appear, if appropriated precautions are not taken over.

Caution Bodily harm or material damage will appear, if appropriated precautions are not taken over.

Attention means, that a unwished results or states can occur, if the appropriated instruction is not noticed.

Important means the commitment to a special behavior or operation for the safe treatment of the controller / machine.

Qualified personnel

All devices described in this manual may only be used, built up and operated together with this documentation. Installation, initiation and operation of these devices might only be done by instructed personnel with certified skills, who can prove their ability to install and initiate electrical and mechanical devices, systems and current circuits in a generally accepted and admitted standard.

Operation according to regulations

This device might be only used for this operation written in this manual and only in combination with other certified external devices. For a correct operation a proper transportation, storage, initiation and maintenance is necessary.

All valid safety instructions and regulations for the prevent of industrial accidents are to be attended carefully. The power supply must be connected to a central ground potential in a starlike wiring.



Maintenance

Modifications / repairs of an INSEVIS device might be done only by special educated and trained personnel of INSEVIS in an ESD-safe area. Every unauthorized opening might cause damages and will terminate all warranty claims.



Data security

Each customer is responsible by himself for protecting his IT-environment against illegal external attacks. INSEVIS shall not be held liable for any direct, indirect or consequential damages respect to any claims arising from the possible illegal external access to their PLCs or HMIs by Ethernet. If you are not sure, how to protect your environment ask for help at professional legal IT-companies.

Copyright

This and all other documentation and software, supplied or hosted on INSEVIS web sites to download are copyrighted. Any duplicating of these data in any way without express approval by INSEVIS GmbH is not permitted.

All property and copy rights of theses documentation and software and every copy of it are reserved to INSEVIS GmbH.

Trade Marks

INSEVIS refers that all trade marks of particular companies used in own documentation as e.g.

- STEP®, SIMATIC® and other as reserved trade mark of Siemens AG.

- CANopen® and other as reserved trade mark of CAN in Automation eG

- WINDOWS® and other as reserved trade mark of MICROSOFT AG

and more reserved trade marks are property of the particular owners and are subjected to common protection of trade marks.

Disclaimer

All technical details in this documentation were created by INSEVIS with highest diligence. Anyhow mistakes could not be excluded, so no responsibility is taken by INSEVIS for the complete correctness of this information. This documentation will reviewed regulary and necessary corrections will be done in next version.

With publication of this manual all other versions are no longer valid.

Essential knowledge and experiences

To understand this documentation basic knowledge and experiences of the automation technology in general and the programming with STEP®7 are essential.

About INSEVIS

S7-system components for industrial automation technology

The range of INSEVIS- product families enables an integrated solution and easy to handle for small and medium automation application with latest technology, very high quality level and with additional interfaces like CANopen® and Modbus, to be configured easily.

The easy integration of INSEVIS-products into the S7-world meanwhile is famous and exemplary. Complex communication settings will be assigned easily and intuitively, so that these properties expand the common S7-world by far. A large and multilingual visualization in a modern design is done by a few clicks and the work flow is known by every WinCCflex user. It can be simulated on the visualization PC and is accessible remote.

The S7-CPU- V and -P are the base of the successfully INSEVIS product families with Profibus DP Master/Slave. With the new S7-CPU-T Panel-PLCs and Compact-PLCs are available with Profinet IO Controller.

Step®7-Programability

INSEVIS-S7-CPU- s are programmable by STEP 7® - AWL, KOP, FUP, S7-SCL, S7-Graph from Siemens and in general command-compatible to Siemens-CPU S7-315-2PN DP. Some special INSEVIS-blocks expand the functionality and allow outstanding solutions. The S7-programming will be done by good known tools SIMATIC®-Manager or by TIA-Portal® from Siemens always.

Independence

INSEVIS-products does not base on Windows or Linux, they have an own firmware. Thereby the hard- and software can be exactly designed for a perfect co-ordination with this firmware and a low power consumption. Booting times of less than 4 seconds and completely no software licenses and a current drain of <100mA @ 24V are the result of these facts.

Get your software rid of licenses

INSEVIS stands for a clear and honest license policy, what gives the customer sustainable cost benefits. Because of the ownership of BIOS, firmware and PC-software for visualization, configuration and remote access INSEVIS can offer its products completely without licenses.

Made in Germany

Development, PCB-design and -production, test and mounting of all INSEVIS-products - all this is made in Germany. So every product is a proof for the combination of German engineering and economy and is available with a certification of German origin.



INSEVIS operates a yearly certified quality management system ref. to DIN EN ISO 9001.

All suppliers of INSEVIS obligate to this quality management and contribute to the high quality level of INSEVIS-products.

Already during planning these families one goal was indicated as most important: to design highest quality and ergonomics into all products.

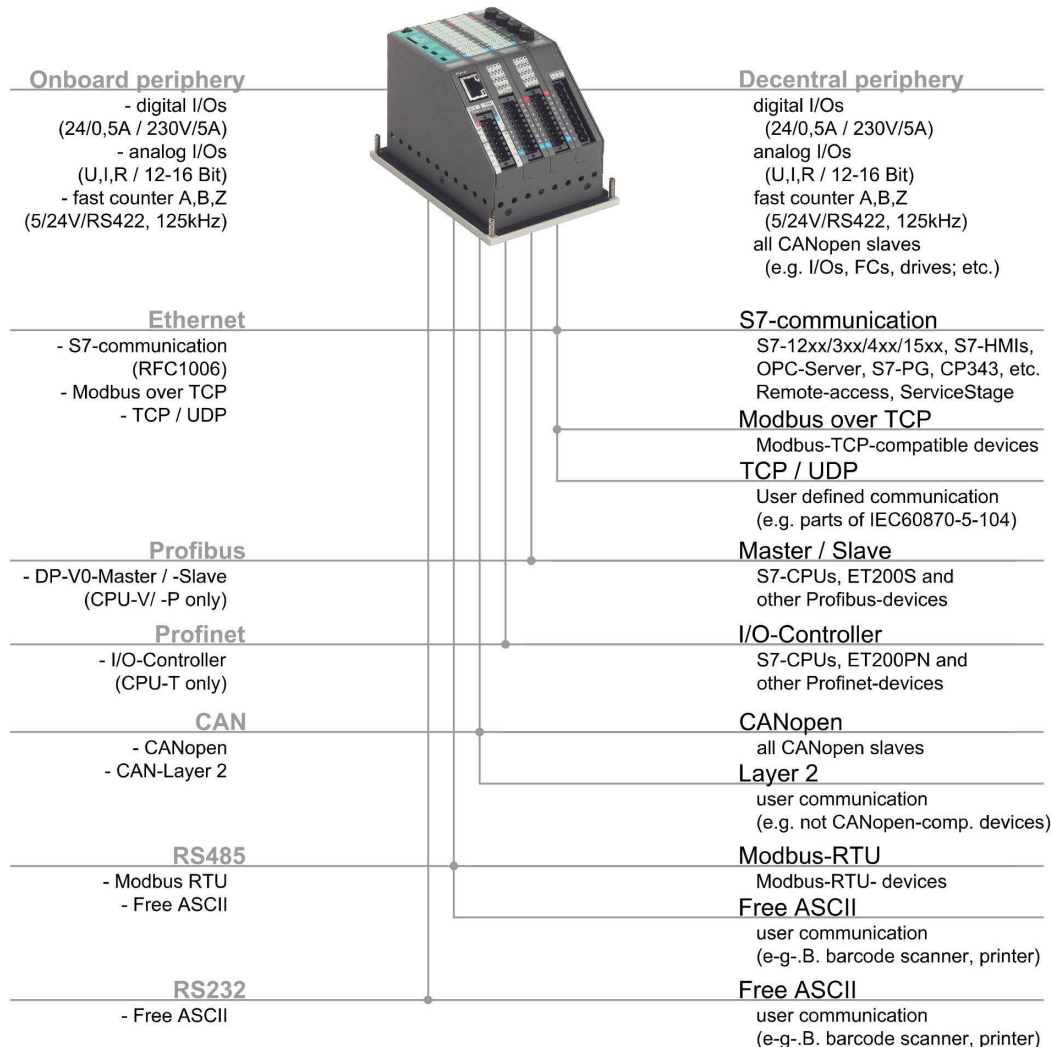
These products were put into comprehensive validation tests before they were produced in selected and certified production lines.

INSEVIS - Made in Germany

Product family S7-Compact-PLC

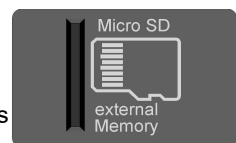
Communication overview

INSEVIS-S7-Compact-PLCs contain versatile possibilities for a connection to diverse peripherals or communication with other devices. Lots of protocols are implemented already, others can be realized with the cost free INSEVIS-S7-SFBs and SFCs by the S7-programmer itself.



External memory card

Each controller has an own slot for an optional Micro-SD-card in the standard FAT32-format. This card is needed for archiving or updating only, not to run the S7-program. The S7-program data are kept in the flash memory (Flash) and the S7-process data are kept in the akku-buffered RAM (SRAM) and not on the Micro-SD-card. Use this card only for archiving of message data, of data from DBs, of trend data as well as of data of the recipe management and for backup/restore. While updating the PLC-firmware by this card, the S7-data will kept untouched - as they were before.



Most important properties at a glance

S7-Programming

Use existing Siemens-S7-programming tools; either SIMATIC®-Manager or TIA-Portal® in the programming languages KOP, FUP, AWL, SCL. And use existing FB's like for PID in analog operations...

Backup & Restore

Save all data easily; user program, process data, visualization and archives - protected by password as a binary file for using in an equal equipped device, what will proceed with all data from the old PLC

Data archiving

Save and archive process data to the Micro-SD-card and read it back to the PLC after updating S7-program. Completely without programming device - by using INSEVIS-SFCs and SFBs.

Gateway functionality

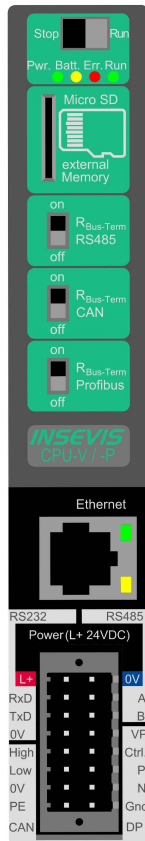
Ethernet with TCP, UDP, RFC1006 or Modbus TCP, Profinet IO Controller, CANopen® or Layer2, free ASCII on RS232 and RS485 and Modbus RTU. INSEVIS-S7-PLCs are communication talents.

Technical data of INSEVIS-S7-CPU's

CPU-V

Devices with **CPU-V** fit properly to small and medium sized applications in the low cost-areas of Compact-PLCs with lots of communication interfaces. Profibus is optional available.

CPU-V



Property	Technical data
OB, FC, FB, DB Local data Number of inputs and outputs Process image Number of Merkerbytes Number of Taktmerker Number of timer, counter Depth of nesting	each 1.024 32kByte (2kByte per block) in each case 2.048 Byte (16.384 Bit) addressable in each case 2.048 Byte (default set is 128 Byte) 2.048 (remanence adjustable, default set is 0..15) 8 (1 Merkerbyte) in each case 256 (each remanence adjustable, default set is 0) up to 16 code blocks
Real-time clock elapsed hour counter	yes (accumulator-backed hardware clock) 1 (32Bit, resolution 1h)
Program language Program system	STEP 7® - AWL, KOP, FUP, S7-SCL, S7-Graph from Siemens SIMATIC® Manager from Siemens or products compatible to it
Operating system Program unit to reference	compatible to S7-300® from Siemens CPU 315-2DP/PN (6ES7 315-2EH14-0AB0 firmware V3.1 Siemens)
Communication	
Serial interfaces (protocols)	COM1: RS 232 (free ASCII) shielded wiring COM2: RS 485 (free ASCII, Modbus-RTU)
Ethernet (protocols)	Ethernet: 10/100 MBit with CP343 functionality (RFC1006, TCP, UDP, Modbus-TCP)
CAN (protocols)	CAN-telegrams (Layer 2), compatible to CANopen® master/ slave 10 kBaud ... 1 MBaud
optional interfaces (protocols)	Profibus DP V0 master/ slave 9,6kBaud ... 12 MBaud
Periphery access	
Decentral periphery	- INSEVIS- periphery (with automatic configuration via „ConfigStage“) - diverse external periphery families (Modbus RTU/TCP, CAN) - all CANopen® slaves according to DS401 - all Profibus DP-V0-slaves

Memory	CPU-V
Working memory, thereof buffered by akku	512kB 256 kByte remanent
Load memory	2MB flash memory
Memory for visualization	4MB flash memory
external memory	Micro SD, up to 8 GB

Most important properties at a glance

S7-Programming

Use existing Siemens-S7-programming tools; either SIMATIC®-Manager or TIA-Portal® in the programming languages KOP, FUP, AWL, SCL.
Or use existing FB's like for PID in analog operations...

Individualization

Keep your own logo as 3D-Doming on the front foil or as bitmap fix included in your OEM-firmware, or as inserting stripe with order-no at the rear side? Everything is possible.

System boot time 4 seconds

No Windows-firmware means to boot up in less than 4 seconds and primarily: no licenses. And also no run-time limitations for tags. Therewith todays devices still may be updateable in more than 10 years...

Gateway functionality

Ethernet with TCP, UDP, RFC1006 or Modbus TCP, Profibus-DP V0 Master/Slave, CANopen® or Layer2, free ASCII on RS232 and RS485 and Modbus RTU
INSEVIS-S7-PLC - a communication talent

Technical data of INSEVIS-S7-CPU's

CPU-T

Devices with **CPU-T** have more memory, a higher speed and can drive larger panels with more visualization objects. They fit perfectly for medium sized automation solutions. 2 separated Ethernet ports for separated networks or together as an Ethernet switch are onboard. Profinet IO Controller is available as an option. Panel-PLC and Panel-HMIs of the **Generation II** are equipped with this CPU only.

CPU-T



Property	Technical data
OB, FC, FB, DB	each 2.048
Local data	32kByte (2kByte per block)
Number of inputs and outputs	in each case 4.096 Byte (32.769 Bit) addressable
Process image	in each case 4.096 Byte (default set is 128 Byte)
Number of Merkerbytes	4.096 (remanence adjustable, default set is 0..15)
Number of Taktmerker	8 (1 Merkerbyte)
Number of timer, counter	in each case 512 (each remanence adjustable, default set is 0)
Depth of nesting	up to 16 code blocks
Real-time clock	yes (accumulator-backed hardware clock)
elapsed hour counter	1 (32Bit, resolution 1h)
Program language	STEP 7® - AWL, KOP, FUP, S7-SCL, S7-Graph from Siemens
Program system	SIMATIC® Manager from Siemens or products compatible to it
Operating system	compatible to S7-300® from Siemens
Program unit to reference	CPU 315-2DP/PN (6ES7 315-2EH14-0AB0 firmware V3.1 Siemens)
Communication	
Serial interfaces (protocols)	COM1: RS 232 (free ASCII) shielded wiring COM2: RS 485 (free ASCII, Modbus-RTU)
Ethernet (protocols)	2x Ethernet: (switch or separated ports): 10/100 MBit with parts of CP343 functionality (RFC1006, TCP, UDP, Modbus-TCP)
CAN (protocols)	CAN-telegrams (Layer 2), compatible to CANopen® master/ slave 10 kBaud ... 1 MBaud
optional interfaces (protocols)	Profinet IO Controller
Periphery access	
Decentral periphery	- INSEVIS- periphery (with automatic configuration via „ConfigStage“) - diverse external periphery families (Modbus RTU/TCP, CAN) - all CANopen® slaves according to DS401 - all Profibus DP-V0-slaves

Memory	CPU-V
Working memory (thereof buffered by akku)	1MB 512 kByte remanent
Load memory	8MB flash memory
Memory for visualization	48MB flash memory
external memory	Micro SD, up to 8 GB

Most important properties at a glance

S7-Programming

Use existing Siemens-S7-programming tools; either SIMATIC®-Manager or TIA-Portal® in the programming languages KOP, FUP, AWL, SCL.
Or use existing FB's like for PID in analog operations...

2 Ethernet ports

Exclusive at the CPU-T:
Use both Ethernet interfaces as separated ports with own IP-address-ranges to drive the PLC as a gateway between office network and machine network.

Know-how-protection

Save your work from illegal copying, save you know-how to sell it more than once.
Set really heavy protections by free ServiceStage (Siemens-password functions are still available.)

Gateway functionality

Ethernet with TCP, UDP, RFC1006 or Modbus TCP, Profinet IO Controller, CANopen® or Layer2, free ASCII on RS232 and RS485 and Modbus RTU
INSEVIS-S7-PLC - a communication talent.

Decentral touch panel units for the Compact-PLCs

General

By using TFT- displays a high brilliance of color fastness and brightness and a wide viewing angle are possible. LED-backlight does assure typical half-live periods of 50.000 operating hours and increases the range of operating temperature from -20°C to +60°C (without condensation). All Panel-PLCs are multilingual and can be used in vertical and horizontal way. The complete front (including the touch area) is protected by a laminated IP65 foil.

Customized logos can be implemented on front- or rear side easily. Operating systems with custom logos are available as well. So INSEVIS products can be adapted to customers design easily and without initial costs. Inox-fronts are available too.

With CPU -V and -P

The S7-Panel-PLC-family with CPU-V/-P consist of 4 different display diagonals:

- 3,5" with QVGA-resolution (320x240pixel, 4:3-format)
→ for front panel use with a height of 96mm,
- 5,7" with QVGA- resolution (320x240pixel, 4:3-format) and
→ for front panel use (also from the side),
- 7" with WVGA- resolution (800x480pixel, 16:9-format)
→ for front panel use,
- 10,2" with WVGA- resolution (800x480pixel, 16:9-format)
→ for front panel use



With CPU -T

The S7-Panel-HMI family with the CPU-T consist of 4 different display diagonals:

- 4,3" with a resolution of 480x272pixel, 16:9-format
→ for front panel use with a height of 96mm,
- 7" with WVGA-resolution of 800x480pixel, 16:9-format
→ for front panel use
- 10,1" with a resolution of 1024x600pixel, 16:9-format
→ for front panel use,
- 15,6" with a resolution of 1366x768pixel, 16:9-format
→ for front panel use,



Periphery in the Compact-PLCs

Communication to PLC

While onboard periphery has its access to INSEVIS S7-CPU by the integrated rear bus, the head stations of the decentral periphery communicate with the CPUS by a protocol, compatible to CANopen®. Because CAN is not so common in the S7-world, INSEVIS maps its decentral periphery only by inserting the CAN-node. All others is made automatically and no INSEVIS-customer needs to know anything about CAN.

Onboard-Peripherie

digital I/Os
(24V/0,5A / 230V(5A)
analog I/Os
(U,I,R / 12-16 bit)
counter
(5V/24V/RS422, 125kHz)



1...127x Dezentrale Peripherie DP3xxC

digital I/Os
(24V/0,5A / 230V(5A)
analog I/Os
(U,I,R / 12-16 bit)
counter
(5V/24V/RS422, 125kHz)
all CANopen slave
(e.g. I/Os, drives, FCs, etc.)



Periphery modules

Decentral head stations DP303C/DP307C/DP311C

with 3 / 7 / 11 periphery slots

Periphery module DI16

(16 digital inputs 24V)

Periphery module DIO16

(16 digital in- or outputs 24V / 0,5A)

Periphery module DO4-R

(4 relay outputs 230V / 3A)

Periphery module MIO84

(8 digital- and 4 analog in- or outputs)

Periphery module AI4O4

(4 analog in- and 4 analog outputs)

Periphery module AI8O2

(8 analogin- and 2 analog outputs)

Periphery module RTD8O2

(8 analog in- and 2 analog outputs)

Function module DIO8-Z

(2 Encoder inputs A,B,Z and 2 digital in- or outputs)

Hint:

More product information to all modules are available at INSEVIS-web sites in the Product / Periphery - area and contain more information for wiring and block schemes

Data valid for all periphery and function modules:

Property	Technical data
Operating temperature range	-20°C ... +60°C
Storing temperature range	(without condensation) -30°C ... +80°C
Dimension (W x H x D)	20 x 108 x 70 mm
Weight	ca. 150 g
Wire length	
unshielded (max.)	30 m
shielded (max.)	100 m
Connection technology	Cage clamp technology for cross section up to max. 1,5mm²

Delivery scope

- Rear foil with
 - signal overview stripe (changeable)
 - inserting stripe for identification V
- Technical data

Accessories

- Connector 20-pin (lift arms/ bolt flanges)
with max. 1,5mm²,
- Inserting stripes V with customer logo,

Most important properties at a glance

High packing factor

By a module width of < 20mm lots of I/Os fit in a Compact PLC or decentral head station. A slim mounting depth of < 95mm and an angled connection layer towards the cable channel pre-assembled cable harness

Easy configurable head station

Assign decentral head stations by 2 node-IDs only. Once directly at the head stations turn switches and once in the cost free configuration software. That's all.

High resolution

All analog INSEVIS-I/Os resolution is minimum 12Bit. If you allow a little more integration time to the inputs of AI4O4, you can increase this resolution up to 16Bit. Of course without more costs. As always at INSEVIS.

Intelligent configuration

Selecting the functionality bitwise as digital input or output. Assign the wiring of analog inputs as 2-, 3- or 4- wire configuration. Choose between current or voltage an analog I/Os.

Software in the Compact-PLCs

Configuration



With the free configuration tool „ConfigStage“ you can config the **additional functions** of the INSEVIS-CPU's and download it into the PLC. The onboard- or decentral INSEVIS-periphery will be added easily by drag'n drop to the periphery slots. Parameters and address areas will be assigned in a box right below.

Also you may assign **S7-CPU-parameters** like in your programming tools from Siemens (like startup, diagnostic, cycle and clock, retentive memory, etc.).

With the „ConfigStage“-software can be assigned these interfaces :

- RS232 with free ASCII,
- RS485 with free ASCII and Modbus RTU,
- Ethernet-connections (active S7-connection-RFC1006, TCP, UDP, Modbus-TCP, INSEVIS-Panel-HMI),
- CAN (CANopen® by pre-defined parameters or by imported and mapped EDS-files),

Remote access



With the free-of-charge software „RemoteStage“ data archives can be read in from the Micro-SD-card in the Compact-PLC. This program is a portable solution but with lots of functions. It communicates via TCP/IP with the Compact-PLC. These archive data can be converted into csv-format and stored anywhere. With an SFC213 READ_CSV modified csv-files can be read back to the Compact-PLC, what offerst lots of solutions.

This procedure is also possible as command line in any batch process. Multiple RemoteStages can be operated in one PC to get a kind of master display with multiple remote screens.

Service tool



The „ServiceStage“ is made to have easy service access to the PLCs. It allows to identify a S7-CPU by its IP-address and to analyze it by diagnostic buffer, to update the user-programs and to set the protection levels.




This software is free of licenses, offers lots of service functions easily to be found even if you are not working every day with this software tool. Installed in a minute, useful service functions, absolutely easy to understand and to use. These functions can be solved with the „ServiceStage“

- Device specific data like (ser.-no., firmware version, IP-address (editable), MAC-address, etc.),
- Update firmware (CPU-T only),
- Changing of the operation mode RUN ↔ STOP,
- Set and synchronize date and time,
- Memory diagnostic and comprimation,
- Read out, show and store CPU-diagnostic buffer,
- Download of S7-program, visualization- and configurations binary,
- Assign the know-how-protection levels

Accessories for S7-Compact-PLCs

There are available pin marked removable connectors either with lift arms or with bolt flanges to connect INSEVIS-devices. This allows a explicit positioning of each pin to the signals and makes the wiring easier. The wire-contact is done by maintenance free cage-clamps for max. 1,5mm² cross sections without wire end sleeves.

A mounting set with grounding terminal is part of every delivery. If periphery modules are ordered, they will be mounted for free at the INSEVIS production together with the referring rear foil, standard inserting and signal stripes.

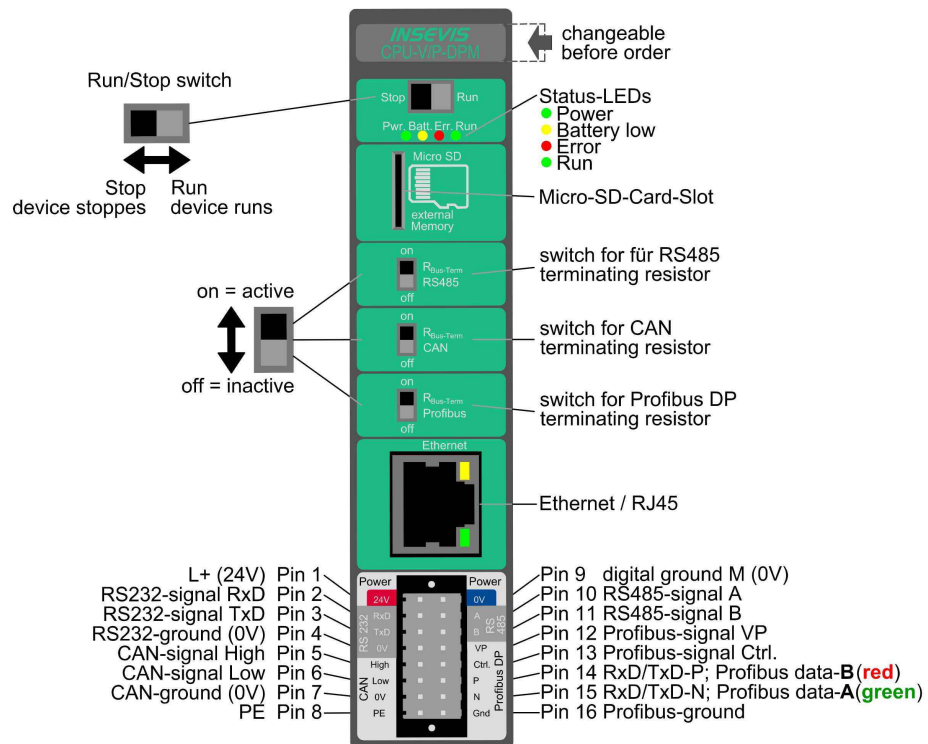
Figure of accessories	accessories	Article-no.	MOQ (pcs)
Connectors   E-CON16* E-CONS16 (pin marked connectors for max. 1,5mm ² cross sections)	For CPU V: Connector 2x8pin, lift arms* (discontinued in 2018, not recommended for new projects, use E-CONS16 instead) Connector 2x8pin, bolt flanges Adapter for Profibus SUB-D 9	E-CON16-00* E-CONS16-00 E-AD-DP12-00	 1 1 1
  E-CON10* E-CONS10 (pin marked connectors for max. 1,5mm ² cross sections)	For CPU T: Connector 2x5pin, lift arms (discontinued in 2018, not recommended for new projects, use E-CONS10 instead) Connector 2x5pin, bolt flanges	E-CON10-00* E-CONS10-00	 1 1
External memory  Hint: S7-program runs without these cards, these parts are used for archiving / recipes / updating only	Micro SD-card 1GB (external memory) Micro SD-card 2GB (external memory) Micro SD-card 4GB (external memory) Micro SD-card 8GB (external memory)	E-MSD1-00 E-MSD2-00 E-MSD4-00 E-MSD8-00	 1 1 1 1
Customized labeling Hint: ¹⁾ Single fix costs only, no run-time costs per single PLC	3D-doming labels with customer logo (front) OEM-firmware with integrated customer logo Inserting stripes H with customer logo (rear) Inserting stripes V with customer logo (rear)	E-LAB3D-00 SW-BS-OEM ¹⁾ E-LABH-00 E-LABV-00	 100 1 100 100
Software Hint: required, otherwise only old versions do run Software available for free download at INSEVIS web sites	ConfigStage RemoteStage ServiceStage	- - -	 free download free download free download
Spare parts Hint: A mounting set with grounding terminal is part of every delivery	Additional grounding terminal set (with 10pcs of it)	E-MNT-00	 10

Stainless steel front plates, attached parts and customized designs on request.

Wiring of the S7-Compact-PLCs

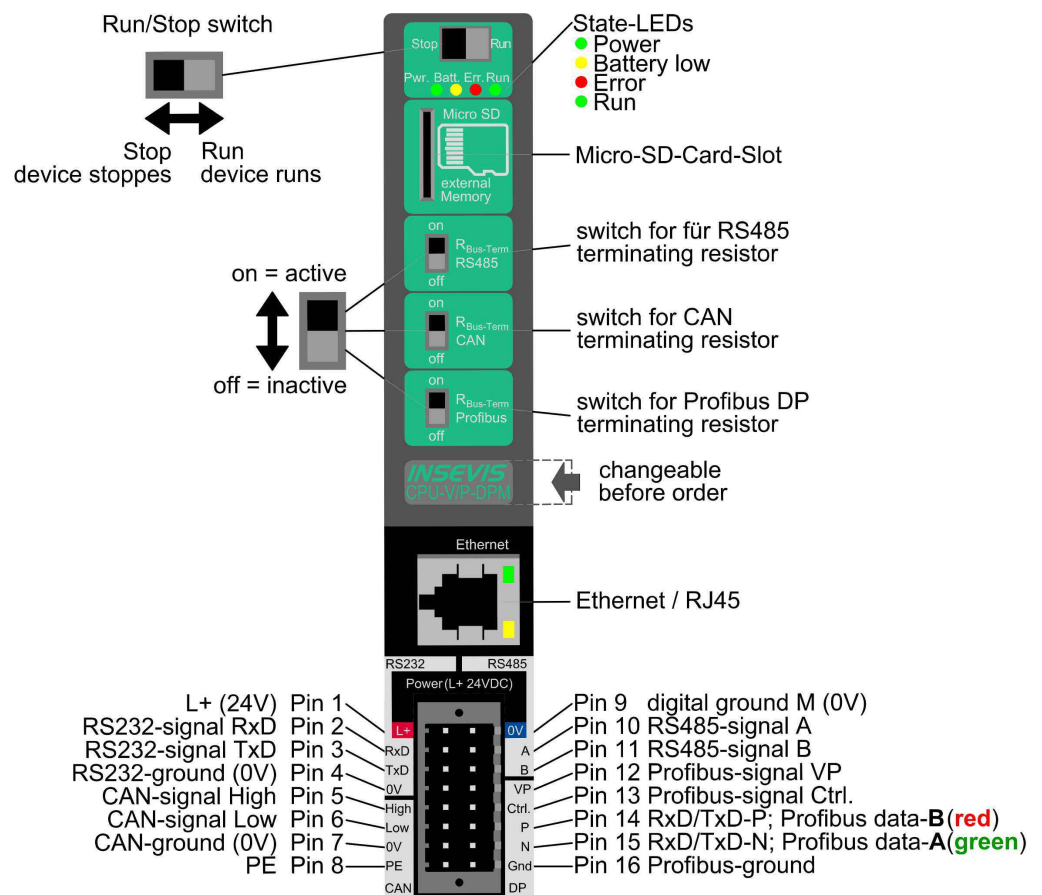
Wiring of CPU-V in devices with/without peripheral slots

Wiring in basic devices (without peripheral slots)
CC300V



RS232 Wiring shielded (ref. to RS232 spec.)

Wiring in devices with peripheral slots
CC303V
CC307V
CC311V

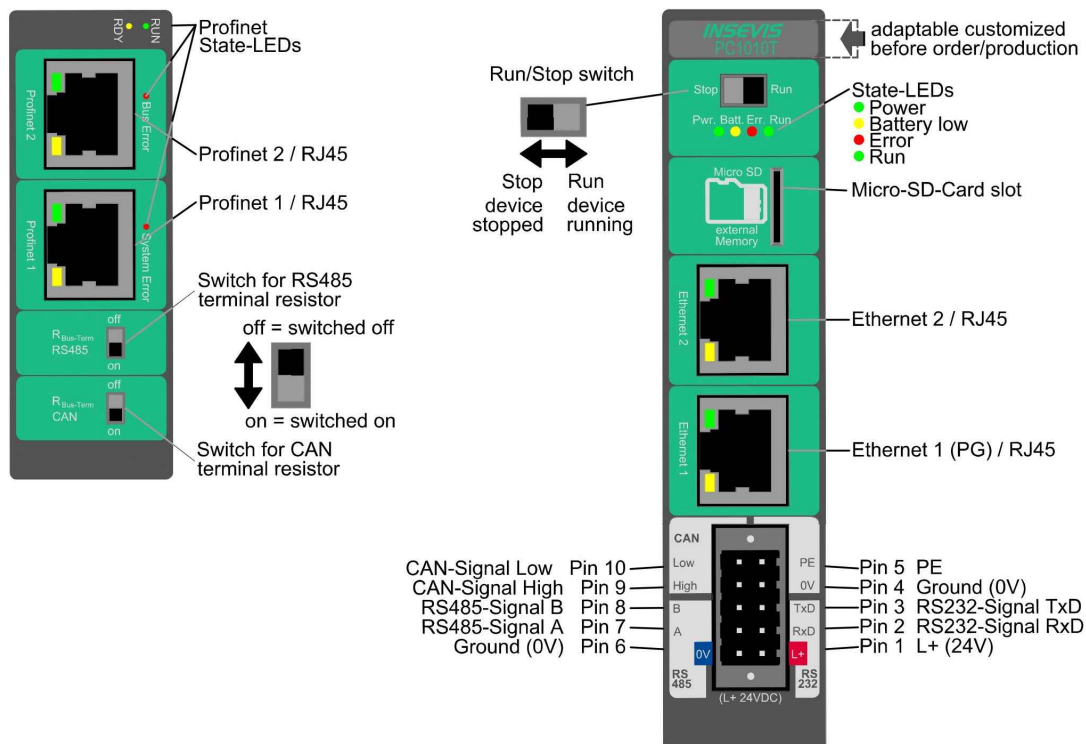


RS232 Wiring shielded, (ref. to RS232 spec.)

Wiring of the S7-Compact-PLCs

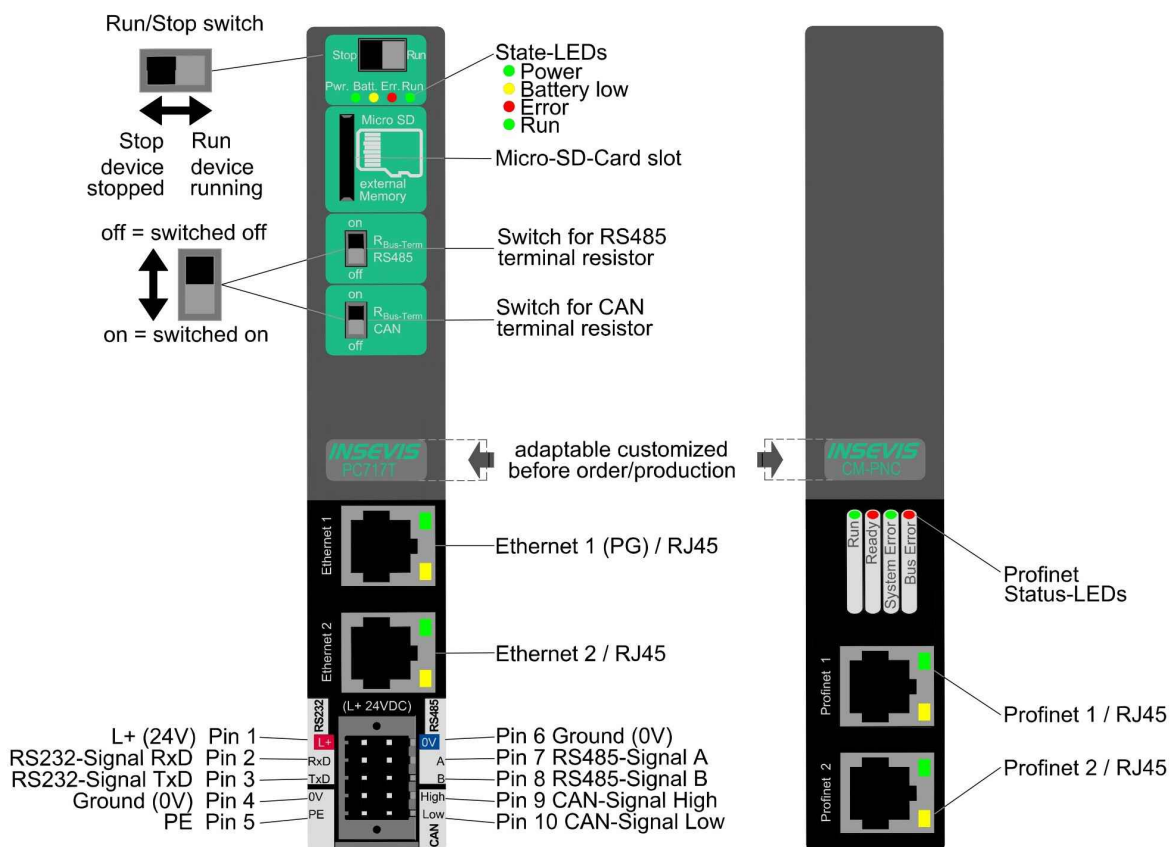
Wiring of CPU-T in devices with/without periphery slots

Wiring in basic devices (without periphery slots)
CC300T



RS232 Wiring shielded (ref. to RS232 spec.)

Wiring in devices with periphery slots
CC303T
CC307T
CC311T



RS232 Wiring shielded (ref. to RS232 spec.)

General

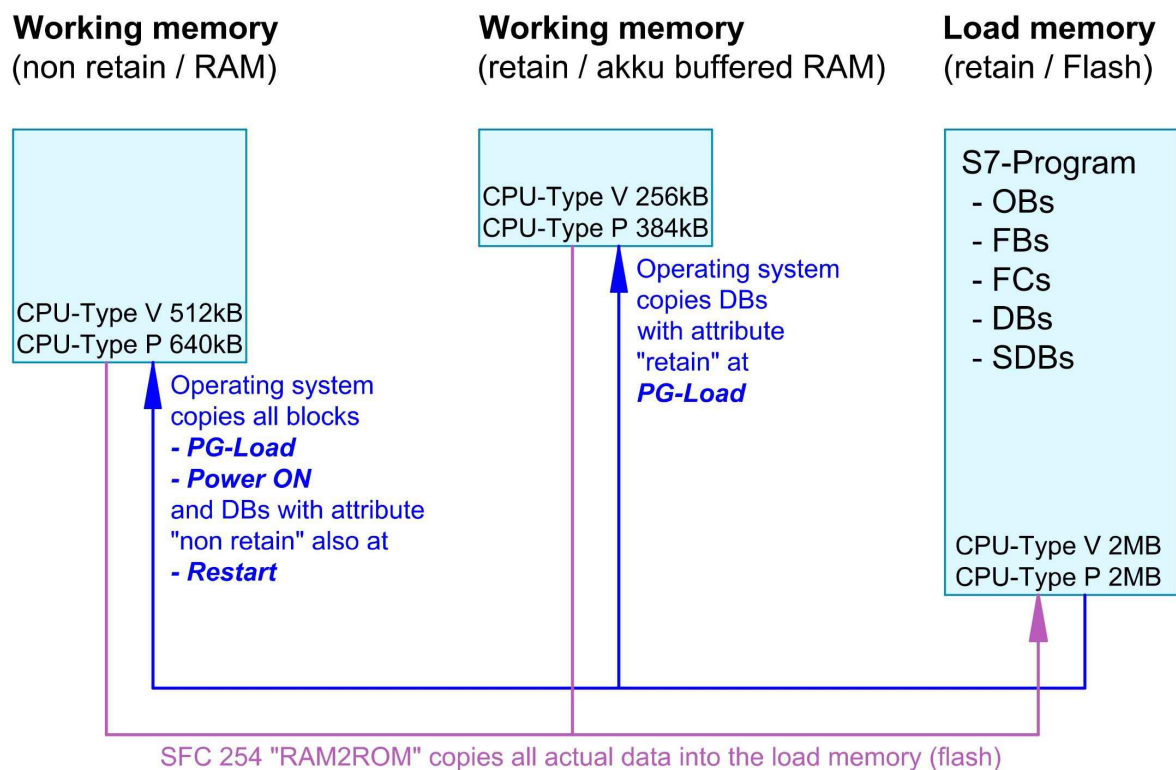
Memory organization

The memory organization matches in general to the Siemens S7-CPU's. The S7-program is kept in the load memory (flash) and will be copied from the INSEVIS- operating system into the working memory (RAM)

A battery (accumulator) keeps the data in the battery buffered RAM approx. 3 months. Before a longer Power OFF it is recommended to copy the data from the working memory (RAM) into the load memory (flash=ROM) by the SFC 254 „RAM2ROM“.

If the accumulator is empty and the PLC comes in power ON, „Battery error“ will be entered in the diagnostic buffer and the yellow battery-LED on the rear side of the PLC is on.

The operating system copies all blocks from the load- into the working memory and flag memory, timer or counter which are configured as remanent will be reset. Then the CPU changes into RUN-state.



Communication

Ethernet

The connection to the Simatic®-Manager is done by Ethernet TCP/IP. Thereby the PLC will be identified by an own IP-address.



ATTENTION:

The INSEVIS- (Panel- or Compact-) PLCs will be delivered with default **IP-address 192.168.80.50 (CPU-V-P) or 192.168.80.55 (CPU-T)**

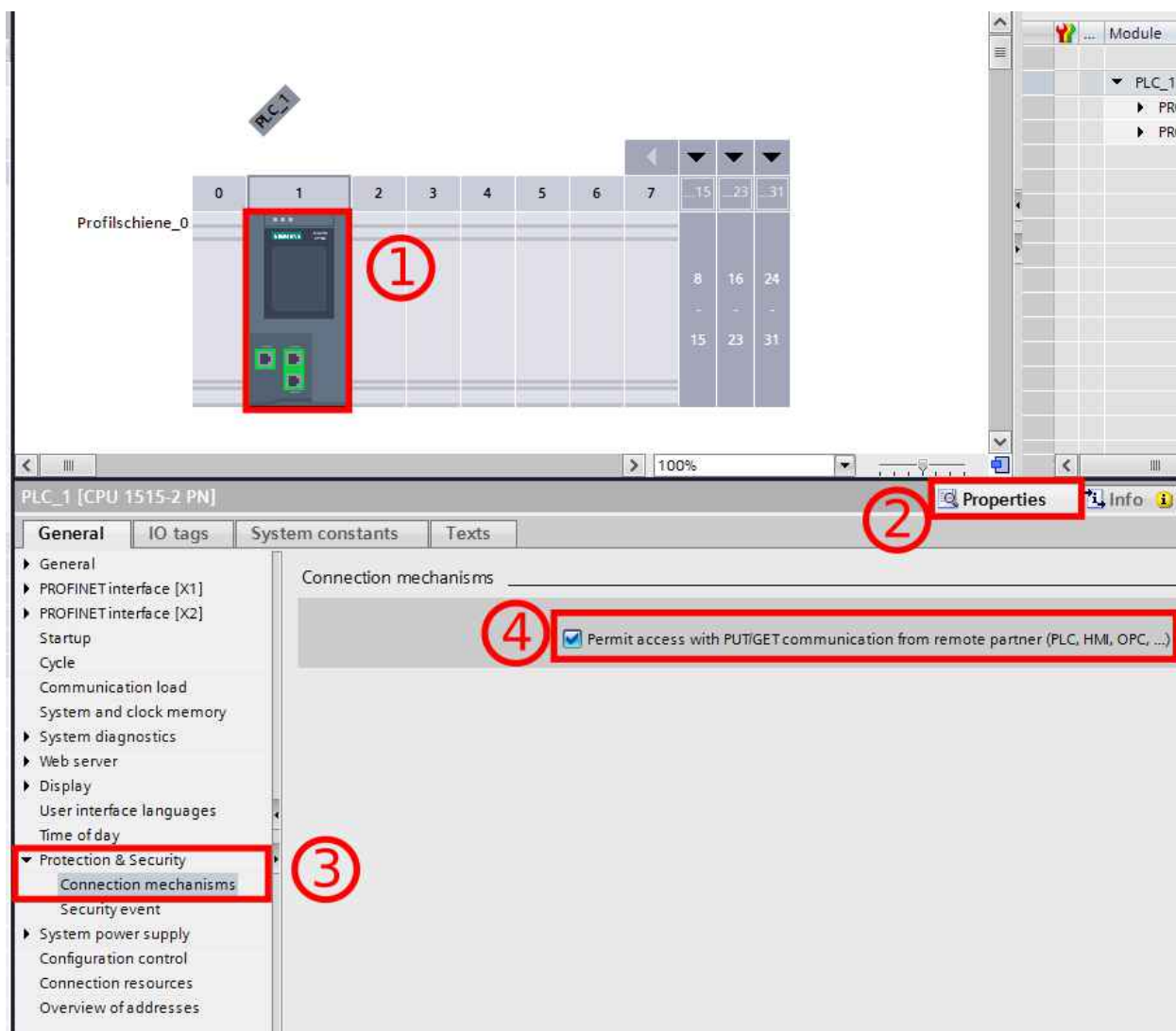
Change this IP-address of your PLC, before you set up your communication. Panel-PLCs will be delivered with an pre-installed S7-program and visualization to change the IP-address on the panel directly at run time.



ATTENTION:

To enable the communication with CPUs from the series 1200 and 1500 you have to activate the PUT/GET communication. To achieve this execute the following steps in your TIA® project:

1. Select your PLC.
2. Choose the tab "Properties"
3. Select "Protection & Security" → "Connection mechanisms"
4. Activate the option:
"Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...)"



Communication - Ethernet

Set up IP-address in the Simatic®-Manager

Essential equipment

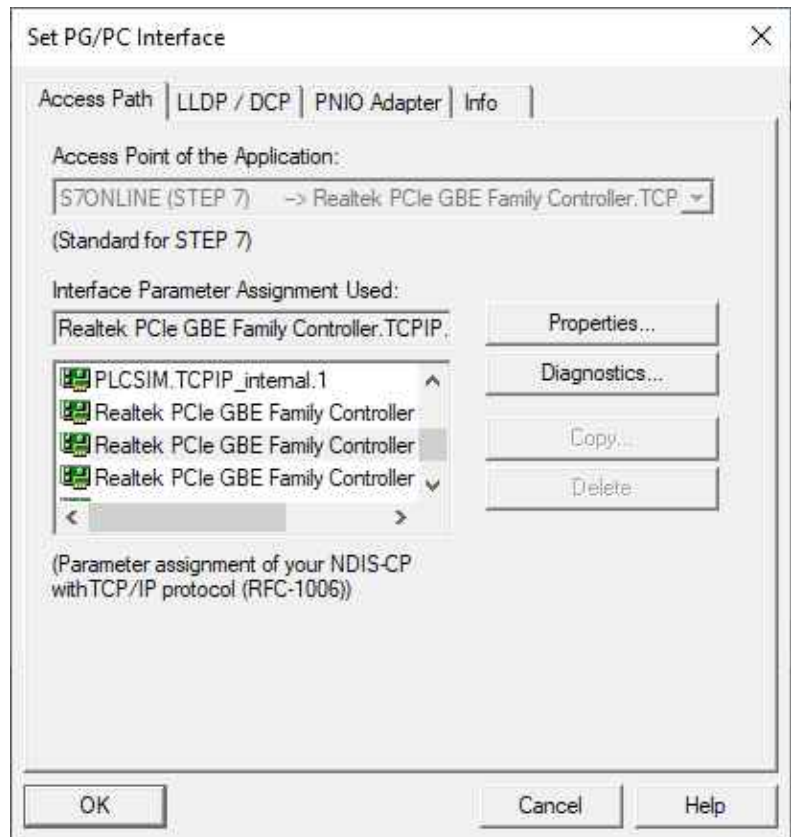
PC with Simatic®-Manager from
Version 5.4 SP5 (PG)
(better use V5.5 or V5.6)
INSEVIS-PLC (AG)
Power supply 24V
Patch cable or crossover
patch cable

Procedure

connect PG with AG
Power on AG
Start Simatic® Manager

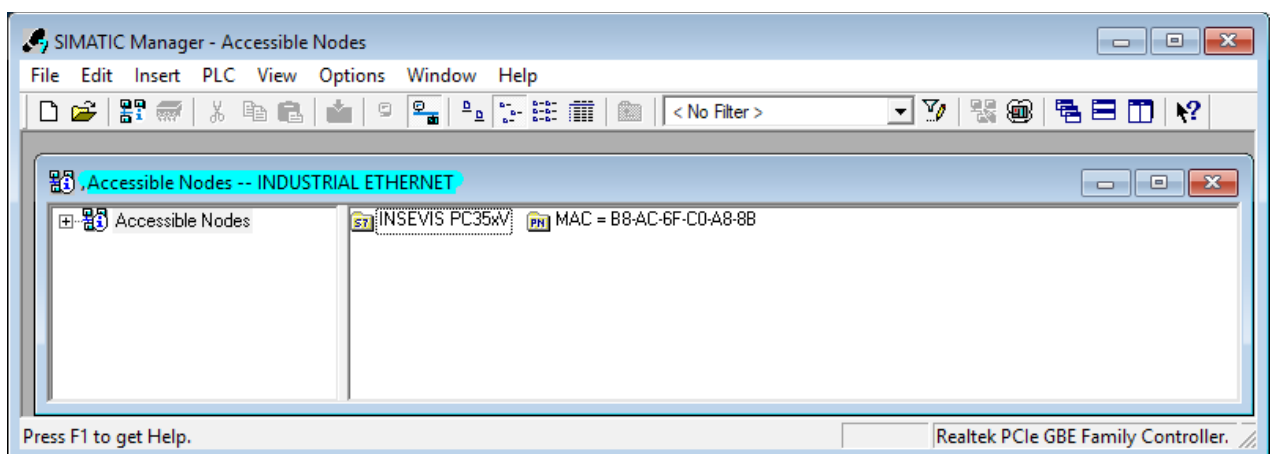
Set up PG/PC-interface

→ Extra
→ Set PG/PC Interface
→ select „Access Path“
→ select **TCP/IP**
→ acknowledge with „OK“



Address the PLC

→ PLC
→ Display Accessible Nodes
(appear as INSEVIS S7-PLC) → e.g. as **INSEVIS S7-PC35xV** select it / mark it
→ PLC
→ Edit Ethernet Node



Communication - Ethernet

Change IP-address

- enter new box
- „Set IP-configuration“
- select „Use IP parameters“
- enter IP address
- enter Subnet mask
- push the button: „Assign IP Configuration“
- acknowledge with „OK“

refresh IP-address

Before using the new IP-address, it must be searched & found again:

- PLC
- Display Accessible Nodes
- OR:
- refresh with „F5“

Control IP-address (without fig.)

- PLC
- Display Accessible Nodes
- right mouse button
- Object properties

Hint:

Other Ethernet-settings like net mask and router-address will be assigned only in the INSEVIS-configuration software ConfigStage at the Ethernet-settings of the CPU.
Beside the PLC's IP-address there will be nothing accepted from Siemens-HW-Config-settings.



VIDEO-tutorial available

There is available a link to the corresponding YouTube®-video at the INSEVIS-download website for use under Simatic®-Manager and TIA®-Portal.

Communication - Ethernet

Set up the CPU and the Ethernet connection in the Simatic®-Manager

Create the INSEVIS-CPU in the Simatic®-Manager

Part:
CPU315-2PN/DP

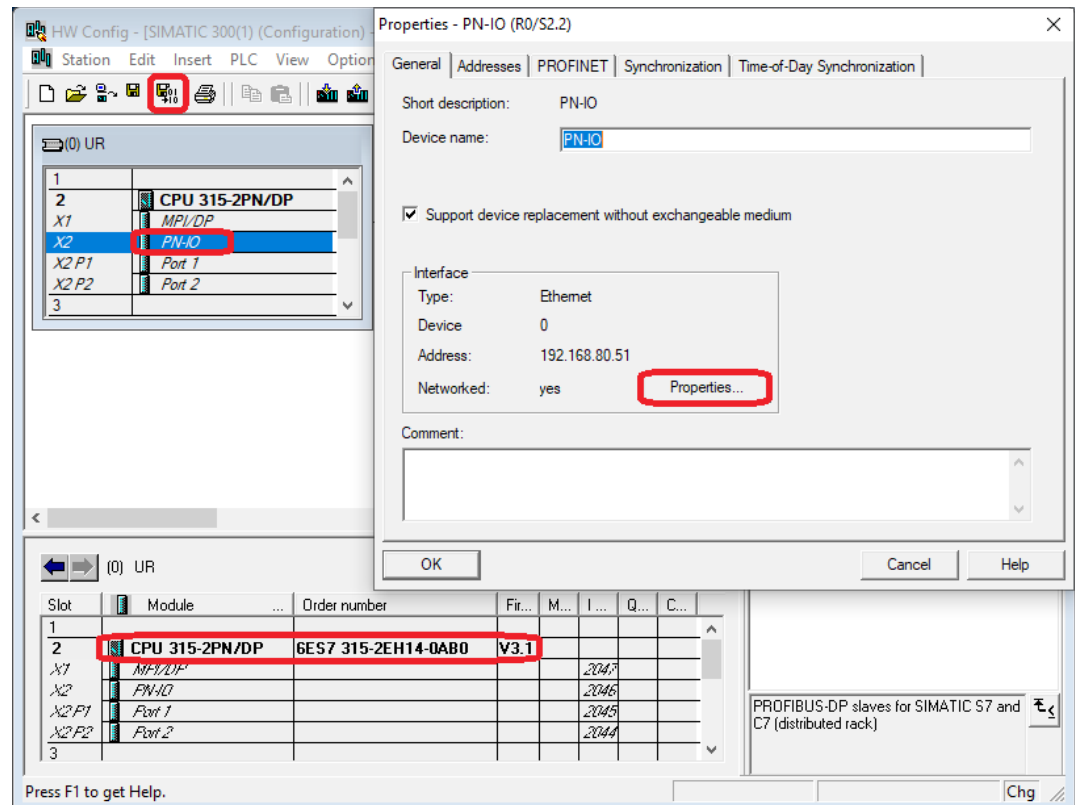
Order no.:
6ES7315-2EH14-0AB0

Firmware:
V 3.1

(this CPU is available with V5.4 SP5 of the Simatic®-Manager only. If you don't find it, please update it to V5.5 or V5.6).

Selection of the Ethernet communication in the Simatic®-Manager:

- Double click on **PN-IO**
- press „property“-button
- set up IP address
- acknowledge with „OK“
- compile & save



VIDEO-tutorial available

There is available a link to the corresponding YouTube®-video at the INSEVIS-download website for use under Simatic®-Manager and TIA®-Portal.

- Trouble shooting:**
- 1) Firewall prevents online connection
→ Reconfigure (all) your firewall(s) to allow this connection.
 - 2) IP-address of your PC and of the PLC must be in a common net (in this sample: 192.168.80.xxx)
→ Check the IP-address of your PC with the cmd-command "ipconfig" and verify the net with those of your PLC.

IP Configuration of PLC with SFB 129 "IP_CFG"

INSEVIS PLC provides the SFB 129 "IP_CFG" to get/set the IP configuration parameter of the PLC. The previously valid configuration data is overwritten.



Hint: This function block is available for PLCs with CPU-V/P and for devices with CPU-T with Ethernet configured as switch. For CPU-T-devices with Ethernet configured as 2 separated ports use SFB 131 "IP_GET" and SFB 132 "IP_SET".

Parameter	Declaration	Data type	Description
SET	INPUT	BOOL	TRUE=Set the IP Configuration, FALSE=Get the IP Configuration
DONE	OUTPUT	BOOL	TRUE=Successfully configured, FALSE=Configuration failed.
ERROR	OUTPUT	BOOL	TRUE=Configuration failed.
STATUS	OUTPUT	WORD	Status code
IP_ADDR	IN_OUT	DWORD	IP Address e.g 192.168.80.50 => DW#16#C0A85032
NETMASK	IN_OUT	DWORD	Netmask. e.g 255.255.255.0 => DW#16#FFFFFF00
ROUTER	IN_OUT	DWORD	Router IP address. DW#16#0 - No Router

DONE, ERROR and STATUS parameters meaning

DONE	ERROR	STATUS W#16#..	Description
1	0	0	Successfully configured.
0	1	8000	Parameter IP_ADDR invalid.

Modbus TCP (Server)

To use the protocol Modbus TCP as server (slave) it is necessary to config it all in the ConfigStage:

- Modbus-TCP-Server enable in menu box
 - Define mapping on (existing!) S7-operators
- The defined S7 operator areas are mapped to Modbus registers or bits starting at **Modbus address 0**.
The number of supported register or bits follows the length of the defined area.

following commands will be supported

- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs
- 03 (0x03) Read Holding Registers
- 04 (0x04) Read Input Registers
- 05 (0x05) Write Single Coil
- 06 (0x06) Write Single Register
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple Registers

In S7 nothing must be programmed. All the data will be processed from the operating system in background synchronous to the control point.

Communication - Ethernet

S7 active connection to Partner-PLCs (RFC1006)

Connections must be configured in ConfigStage and downloaded to local PLC.

Maximal 16 connections can be configured. During the configuration, connection parameters such as connection type, partner address, port number etc., are assigned.

The configured ID number of connection must be given to communication SFB's ID parameter.

INSEVIS PLCs provide the following system function blocks (SFB) for exchanging data through the user program with other Ethernet-capable communication partners.

- SFB 120 "GET" Read data from a Remote PLC
- SFB 121 "PUT" Write data to a Remote PLC

Read data from a Remote PLC with SFB 120 "GET"

You can read data from a remote PLC (e.g INSEVIS, SIEMENS, VIPA etc. RFC1006-capable PLC) by using SFB 120 "GET". This program controls reading of variables and does not need additional communication functions in the user program of the communication partner.

The parameters ID, ADDR, RD are transferred and data exchange starts on rising edge at REQ.

The new job can only be accepted again after the last job is completed and on rising edge at REQ.

Job completion is signaled by NDR or ERROR parameter.

Ensure that areas defined with parameters ADDR and RD match in terms of number, length and data type.



Hint for Get and Put in one connection

Each connection has a job buffer. If a job (SFB 120 „Get“ or SFB 121 „Put“) is in process, any other job is not accepted.

If you call the SFB 120 „Get“ and SFB 121 „Put“ in one connection, the SFB 121 „Put“ will not be accepted, rejected with an error code W#16#8003 and terminated.

Here you have to wait the finishing of the first job, before you start the second job, or you work with 2 connections, if the partner device is supporting 2 connections.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	Request the data exchange at a rising edge
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
NDR	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code
ADDR	IN_OUT	ANY	ANY- pointer to the area on the partner PLC that is to be read
RD	IN_OUT	ANY	ANY- pointer to the area on the local PLC in which the read data are entered

Communication - Ethernet

NDR, ERROR and STATUS parameters meaning

NDR	ERROR	STATUS W#16#..	Description
1	0	0	Job has successfully completed.
0	0	7000	Job has not been accepted, because: <ul style="list-style-type: none"> this is the first call with REQ=0 or because another job (Get) is still active
0	0	7001	First call with REQ=1. Job has been accepted and is in process
0	0	7002	Interim call (REQ irrelevant). Job is in process
0	0	7003	Job assignment is processing in a priority class with lower priority (Nesting call)
0	1	8001	Parameter ID is invalid
0	1	8002	Connection with ID is not configured or Invalid Connection type
0	1	8003	Job cannot be assigned since previous job is not yet completed
0	1	8004	Parameter RD is invalid
0	1	8005	Could not connect to Remote PLC
0	1	8006	Job cannot be assigned, due to connection state.
0	1	8007	Job failed due to Communication problem (cable disconnected, connection is rejected by partner)
0	1	8008	Job failed due to Negative response from partner
0	1	8009	Job failed due to access error on partner area
0	1	800A	Job failed due to access error on local area
0	1	800B	Parameter ADDR is invalid

S7-program sample for using the SFB 120

Purpose: Read 20 bytes data starting MB500 from Partner S7-CPU and write into MB200 in local S7-CPU

e.g

```

CALL  SFB120, DB120
REQ   :=M600.0           // TRUE = Read data from partner S7-CPU
ID    :=1                // Connection ID number, which is configured in ConfigStage
NDR   :=M600.1           // TRUE = New data available
ERROR :=M600.2           // TRUE = Error occurred
STATUS:=MW602            // Result / return code
ADDR  :=P#M 500.0 BYTE 20 // 20 bytes read from partner S7-CPU
RD    :=P#M 200.0 BYTE 20 // write into flag memory starting 200 in local CPU

O      M      600.1       // if successful read from
O      M      600.2       // or error occurred, then
R      M      600.0       // reset the request signal

```


Communication - Ethernet

Write data to a Remote PLC with SFB 121 "PUT"

You can write data to a remote PLC (e.g. INSEVIS, SIEMENS, VIPA etc. RFC1006-capable PLC) by using SFB 121 "PUT". This program controls writing of variables and does not need additional communication functions in the user program of the communication partner.

The parameters ID, ADDR, SD are transferred and data exchange starts on rising edge at REQ.

The new job can only be accepted again after the last job is completed and on rising edge at REQ.

Job completion is signaled by DONE or ERROR parameter.

Ensure that areas defined with parameters ADDR and SD match in terms of number, length and data type.



Hint for Get and Put in one connection

Each connection has a job buffer. If a job (SFB 120 „Get“ or SFB 121 „Put“) is in process, any other job is not accepted.

If you call the SFB 120 „Get“ and SFB 121 „Put“ in one connection, the SFB 121 „Put“ will not be accepted, rejected with an error code W#16#8003 and terminated.

Here you have to wait the finishing of the first job, before you start the second job, or you work with 2 connections, if the partner device is supporting 2 connections.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	Request the data exchange at a rising edge
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
DONE	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code
ADDR	IN_OUT	ANY	ANY- pointer to the area on the partner PLC in which the data is to be written
SD	IN_OUT	ANY	ANY- pointer to the area on the local PLC which contain the data to be sent

Communication - Ethernet

DONE, ERROR and STATUS parameters meaning

DONE	ERROR	STATUS W#16#..	Description
1	0	0	Job has successfully completed.
0	0	7000	Job has not been accepted, because: <ul style="list-style-type: none"> this is the first call with REQ=0 or because another job (Put) is still active
0	0	7001	First call with REQ=1. Job has assigned and is in process
0	0	7002	Interim call (REQ irrelevant). Job is in process
0	0	7003	Job assignment is processing in a priority class with lower priority (Nesting call)
0	1	8001	Parameter ID is invalid
0	1	8002	Connection with ID is not configured or Invalid Connection type
0	1	8003	Job cannot be assigned since previous job is not yet completed
0	1	8004	Parameter SD is invalid
0	1	8005	Could not connect to Remote PLC
0	1	8006	Job cannot be assigned, due to connection state.
0	1	8007	Job failed due to Communication problem (cable disconnected, connection is rejected by partner)
0	1	8008	Job failed due to Negative response from partner
0	1	8009	Job failed due to access error on partner area
0	1	800A	Job failed due to access error on local area
0	1	800B	Parameter ADDR is invalid

S7-program sample for using the SFB 121

Purpose: Write 20 bytes data starting MB200 in local CPU to Partner CPU starting MB500.

```

CALL SFB121, DB121
REQ  :=M600.0           // TRUE = Write data to partner S7-CPU
ID   :=1                // Connection ID number, which is configured in ConfigStage
DONE :=M600.1           // TRUE = Successful written
ERROR:=M600.2           // TRUE = error occurred
STATUS:=MW602           // Result / return code
ADDR :=P#M 500.0 BYTE 20 // 20 bytes write to partner S7-CPU
SD   :=P#M 200.0 BYTE 20 // 20 bytes data starting from MB200 in local CPU

O    M    600.1          // if successful written to
O    M    600.2          // or error occurred, then
R    M    600.0          // reset the request signal

```

Communication - Ethernet

Read in IP configuration with SFB 131 "IP_GET" (CPU-T)

SFB131 "IP_GET" is used to get Ethernet port settings and mode. This function block is available on PLC with CPU-T.


Hint:

This function is for CPU-T-devices, for CPU-V/P-devices use SFB 129 "IP_CFG".

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE: Get the Ethernet settings
PORT	INPUT	INT	Port number. 1=Ethernet 1; 2=Ethernet 2
DONE	OUTPUT	BOOL	TRUE: Successfully done
ERROR	OUTPUT	BOOL	TRUE: Failed
STATUS	OUTPUT	WORD	Status code
IP_ADDR	OUTPUT	DWORD	IP address of Ethernet port e.g 192.168.80.50 => DW#16#C0A85032
NETMASK	OUTPUT	DWORD	Netmask of Ethernet port e.g 255.255.255.0 => DW#16#FFFFFF00
ROUTER	OUTPUT	DWORD	IP address of Router. DW#16#0 - No Router
MAC	OUTPUT	ARRAY[1..6] OF BYTE	MAC address of Ethernet port
PHYMODE	OUTPUT	BYTE	Ethernet port mode Bit 7: 1 = Mode is valid, 0 = Mode is not valid Bit 3: 1 = Auto Negotiation is enabled, 0 = Manual mode Bit 2: Link status 1 = Link, 0 = No link Bit 1: Speed 1 = 100M, 0 = 10M Bit 0: Duplex mode 1 = Full, 0 = Half

Status code meaning

STATUS	DONE	ERROR	Description
W#16#0000	1	0	Successfully got the settings
W#16#7000	0	0	REQ is FALSE
W#16#8000	0	1	Invalid port number. Port must be 1 or 2.

S7-User program: Get the IP settings and mode of Ethernet port 1

```

CALL SFB 131 , DB131
REQ    :=M1502.5
PORT   :=1
DONE   :=M1002.0
ERROR  :=M1002.1
STATUS :=MW1000
IP_ADDR:=MD2030
NETMASK:=MD2034
ROUTER :=MD2038
MAC     :=#MacAddr
PHYMODE:=MB1800

CLR
=      M    1502.5
U      M    1002.0
BEB                                         // if successful done
L      MW 1000                               // end the function
// error evaluation                         // error occurred
// ...
BEA                                         // end of function

```

Communication - Ethernet

Set IP-configuration with SFB 132 "IP_SET" (CPU-T)

SFB132 "IP_SET" is used to set IP settings to Ethernet port. This function block is available on PLC with CPU-T.



Hint:

This function is for CPU-T-devices, for CPU-V/P-devices use SFB 129 "IP_CFG".

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE: Request to set the IP settings
PORT	INPUT	INT	Port number. 1=Ethernet 1; 2=Ethernet 2
IP_ADDR	OUTPUT	DWORD	IP address of Ethernet port e.g 192.168.80.50 => DW#16#C0A85032
NETMASK	OUTPUT	DWORD	Netmask of Ethernet port e.g 255.255.255.0 => DW#16#FFFFFF00
ROUTER	OUTPUT	DWORD	IP address of Router. DW#16#0 - No Router
DONE	OUTPUT	BOOL	TRUE: Successfully done
ERROR	OUTPUT	BOOL	TRUE: Failed
STATUS	OUTPUT	WORD	Status code

Status code meaning

STATUS	DONE	ERROR	Description
W#16#0000	1	0	Successfully set the settings
W#16#7000	0	0	REQ is FALSE
W#16#8000	0	1	Invalid port number. Port must be 1 or 2.

S7-User program: Set the IP Settings of Ethernet port 1

```

CALL SFB 132 , DB132
REQ      :=M1502.6
PORT     :=1
IP_ADDR:=MD2030
NETMASK:=MD2034
ROUTER  :=MD2038
DONE     :=M1002.0
ERROR    :=M1002.1
STATUS  :=MW1000

CLR
=      M   1502.6
U      M   1002.0
BEB
L      MW  1000
// error evaluation
// ...
BEA
// if successful done
// end the function
// error occurred
// end of function

```

Communication - Ethernet

TCP Send/Receive Connection (RFC 793)

INSEVIS PLC provides the following system function blocks (SFB) for exchanging data through the user program with other Ethernet-capable communications partner.

- SFB 122 "TSEND" Send data via TCP Send/Receive connection
- SFB 123 "TREC" Receiving data via TCP Send/Receive connection
- SFB 124 "TDISCON" Terminate the TCP Send/Receive connection
- SFB 125 "TCONFIG" Configure the Partner parameter of TCP Send/Receive connection
- SFB 126 "TSTATUS" Query the status of TCP Send/Receive connection

The connection must be configured in ConfigStage and downloaded to PLC.

In configuration, connection establishment type (Active or Passive) must assigned.

Connection with active establishment, initiates the connection establishment, whereas with Passive, waits the partner requests.

In active establishment, partner parameters should be assigned but in Runtime the partner parameter can be changed by using SFB 125 "TCONFIG".



ATTENTION:

Connection with passive establishment: the local port number must be assigned and unique.

Send data via TCP with SFB 122 "TSEND"

You can send data to a remote Partner by using SFB 122 "TSEND" via TCP (RFC 793). It works asynchronously, in other words, its processing extends over several SFB calls.

The parameters ID, LEN, DATA are transferred and data transfer starts on rising edge at REQ.
The new job can only be accepted again after the last job is completed and on rising edge at REQ.
Job completion is signaled by DONE or ERROR parameter.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	Request the data exchange at a rising edge
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
LEN	INPUT	INT	Number of bytes to send
DONE	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
BUSY	OUTPUT	BOOL	TRUE=Job is in Process
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code
DATA	IN_OUT	ANY	ANY- pointer to the send area. Only Input image (I), Output image (Q), Bit Memory (M), Data Block (DB) area allowed.

Communication - Ethernet

DONE, BUSY, ERROR and STATUS parameters meaning

DONE	BUSY	ERROR	STATUS W#16#..	Description
1	0	0	0	Job has successfully completed.
0	0	0	7000	First call with REQ=0. Job has not assigned
0	1	0	7001	First call with REQ=1. Job has assigned and is in process
0	1	0	7002	Interim call (REQ irrelevant). Job is in process
0	0	0	7003	Job assignment is processing in a priority class with lower priority (Nesting call)
0	0	1	8001	Parameter ID is invalid
0	0	1	8002	Connection with ID is not configured or Invalid Connection type
0	0	1	8003	Parameter DATA is invalid. Only I, Q, M, DB area allowed
0	0	1	8004	Parameter DATA is invalid. e.g DB not loaded.
0	0	1	8005	Parameter LEN is 0 or larger than specified in Parameter DATA
0	0	1	8006	Could not connect to Partner
0	0	1	8007	Job failed due to Communication problem (cable disconnected, connection is rejected by partner)

S7-program sample for using the SFB 122

Purpose: Send 20 bytes data, starting MB200, to a partner.

```

CALL SFB122, DB122
REQ  :=M600.0           // TRUE = Send data to partner
ID   :=1                // Connection ID number, which is configured in ConfigStage
LEN  :=20                // 20 bytes data to send
DONE :=M600.1           // TRUE = Successfully sent
BUSY :=M600.2           // TRUE = In process
ERROR:=M600.3           // TRUE = Error occurred
STATUS:=MW602           // Result / return code
DATA :=P#M 200.0 BYTE 20 // 20 byte data starting from MB200 in local CPU

O    M    600.1          // if successful sent to
O    M    600.3          // or error occurred, then
R    M    600.0          // reset the request signal

```


Communication - Ethernet

Receiving data via TCP with SFB 123 "TREC"V"

The SFB 123 "TREC"V" works asynchronously, which means the job processing extends over several SFB calls.

The parameters ID, LEN, DATA are evaluated and starts the receiving on EN_R = TRUE.

The new job can only be accepted again after the last job is completed and EN_R = TRUE.

Job completion is signaled by NDR or ERROR parameter.

If the length of the data to be received (LEN parameter) is larger than the actually received data, then SFB will copy the received data into the receiver area (DATA parameter) and NDR become TRUE only after the length specified by the LEN parameter has been reached.

If the length of the data to be received (LEN parameter) is less than actually received data, SFB will copy as many bytes into the receiver area (DATA parameter) as specified in the LEN parameter and sets TRUE to NDR and writes to LEN_R the length of data which has written. With each additional call, you will receive rest of data.

If the LEN parameter is 0, then SFB will copy the received data into receiver area (DATA parameter) and immediately sets NDR as TRUE.

Parameter	Declaration	Data type	Description
EN_R	INPUT	BOOL	Enable the receiving data
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
LEN	INPUT	INT	Expected receiving data size in bytes
NDR	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
BUSY	OUTPUT	BOOL	TRUE=Job is in process
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code
LEN_R	OUTPUT	INT	Number of bytes actually received
DATA	IN_OUT	ANY	ANY- pointer to the receiving area. Only Input image (I), Output image (Q), Bit Memory (M), Data Block (DB) area allowed.

DONE, BUSY, ERROR and STATUS parameters meaning

DONE	BUSY	ERROR	STATUS W#16#..	Description
1	0	0	0	Job has successfully completed.
0	0	0	7000	First call with EN_R=0. Job has not assigned
0	1	0	7001	First call with EN_R=1. Job has assigned and is in process
0	1	0	7002	Interim call (EN_R irrelevant). Job is in process
0	0	0	7003	Job assignment is processing in a priority class with lower priority (Nesting call)
0	0	1	8001	Parameter ID is invalid
0	0	1	8002	Connection with ID is not configured or Invalid Connection type
0	0	1	8003	Parameter DATA is invalid. Only I, Q, M, DB area allowed
0	0	1	8004	Parameter DATA is invalid. e.g., DB not loaded.
0	0	1	8005	Parameter LEN is larger than specified in Parameter DATA
0	0	1	8006	Could not connect to Partner
0	0	1	8007	Job failed due to Communication problem (cable disconnected, connection is rejected by partner)

Communication - Ethernet

S7-program sample for using the SFB 123

Purpose: Receive data from a partner and write into buffer starting MB200 and 200 bytes large.

```
CALL SFB123, DB123
EN_R  :=M600.0          // TRUE = Enable to receive data
ID    :=1               // Connection ID number, which is
                        // configured in ConfigStage
LEN   :=0               // 0 = When data received, then
                        // immediately set NDR and copy to DATA
NDR   :=M600.1          // TRUE = New data available
BUSY  :=M600.2          // TRUE = In process
ERROR :=M600.3          // TRUE = Error occurred
STATUS:=MW602           // Result / return code
LEN_R :=MW604           // Number of bytes copied to DATA
DATA  :=P#M 200.0 BYTE 200 // Receive buffer area

O      M      600.1      // if successful received
O      M      600.3      // or error occurred, then
R      M      600.0      // reset the request signal
```

Communication - Ethernet

Terminate the TCP connection with SFB 124 "TDISCON"

The SFB 124 "TDISCON" terminates a communications of connection (TCP, S7 Client Connection). It works asynchronously, which means the job processing extends over several SFB calls.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	Request to start the terminating the connection on TRUE
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
DONE	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
BUSY	OUTPUT	BOOL	TRUE=Job is in process
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code

DONE, BUSY, ERROR and STATUS parameters meaning

DONE	BUSY	ERROR	STATUS W#16#..	Description
1	0	0	0	Job has successfully completed.
0	0	0	7000	First call with REQ=0. Job has not assigned
0	1	0	7001	First call with REQ=1. Job has assigned and is in process
0	1	0	7002	Interim call (REQ irrelevant). Job is in process
0	0	1	8001	Parameter ID is invalid
0	0	1	8002	Connection with ID is not configured or Invalid Connection type

S7-program sample for using the SFB 124

Purpose: Disconnect connection

```

CALL  SFB124, DB124
REQ   :=M600.0           // TRUE = Disconnect the connection
ID    :=1                // Connection ID number, which is configured in ConfigStage
DONE  :=M600.1           // TRUE = Successful
BUSY  :=M600.2           // TRUE = In process
ERROR :=M600.3           // TRUE = Error occurred
STATUS:=MW602            // Result / return code

O      M      600.1       // if successful
O      M      600.3       // or error occurred, then
R      M      600.0       // reset the request signal

```

Communication - Ethernet

Configure the partner parameter by TCP with SFB 125 "TCONFIG"

With SFB 125 "TCONFIG", you can specify the Partner parameter (Port number and IP Address, in case of S7 Client connection, the TSAP number).

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	Request to start the terminating the connection on TRUE
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
PORT	INPUT	WORD	Partner Port/TSAP number e.g: port 2000 _{dec} , enter as W#16#7D0
IPADDR	INPUT	DWORD	Partner IP Address e.g: Address 192.168.80.50, enter as DW#16#C0A85032
TCONN	INPUT	WORD	Reserved, must be 0
TRESP	INPUT	WORD	Reserved, must be 0
DONE	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
BUSY	OUTPUT	BOOL	TRUE=Job is in process
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code

DONE, BUSY, ERROR and STATUS parameters meaning

DONE	BUSY	ERROR	STATUS W#16#..	Description
1	0	0	0	Job has successfully completed.
0	0	0	7000	First call with REQ=0. Job has not assigned
0	1	0	7001	First call with REQ=1. Job has assigned and is in process
0	1	0	7002	Interim call (REQ irrelevant). Job is in process
0	0	1	8001	Parameter ID is invalid
0	0	1	8002	Connection with ID is not configured or Invalid Connection type
0	0	1	8003	Parameter IPADDR is invalid (e.g., 0 by Active establishment)

S7-program sample for using the SFB 125

Purpose: Change Partner configuration to IP address 192.168.80.52 and port number 2000.

```

CALL SFB125, DB125
REQ  :=M600.0           // TRUE = Change partner configuration
ID   :=1                // Connection ID number, which is configured in ConfigStage
PORT :=W#16#7D0         // Partner Port number e.g 2000 (dez)
IPADDR:=DW#16#C0A85034 // Partner IP address e.g 192.168.80.52
TCONN :=W#16#0          // Reserved, must be 0
TRESP :=W#16#0          // Reserved, must be 0
DONE  :=M600.1          // TRUE = Successful
BUSY  :=M600.2          // TRUE = In process
ERROR :=M600.3          // TRUE = Error occurred
STATUS:=MW602           // Result / return code

O     M     600.1        // if successful
O     M     600.3        // or error occurred, then
R     M     600.0        // reset the request signal

```

Communication - Ethernet

Query the Status of TCP connection with SFB 126 "TSTATUS"

With SFB 126 "TSTATUS", you can query the current status of the connection (TCP, S7 Client connection).

Parameter	Declaration	Data type	Description
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
DONE	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code
PORT	OUTPUT	WORD	Partner Port/TSAP number e.g: port 2000 _{dec} , output as W#16#7D0
IPADDR	OUTPUT	DWORD	Partner IP Address e.g: Address 192.168.80.50, output as DW#16#C0A85032

DONE, ERROR and STATUS parameters meaning

DONE	ERROR	STATUS W#16#..	Description
1	0	0XYZ	<p>Job has successfully completed.</p> <p>Bit 0-3: Send job status Bit 0: 1=Connection accepts Send job Bit 1: 1=Send job is in Process Bit 2: 1=Send job has completed successfully Bit 3: 1=Send job has failed</p> <p>Bit 4-7: Receive job status Bit 4: 1=Connection accepts Receive job Bit 5: 1=Receive job is in Process Bit 6: 1=Receive job has completed successfully Bit 7: 1=Receive job has failed</p> <p>Bit 8-12: Connection status 1 Active establishment is in process 2 Connection terminating 3 Connection established/Connected with Partner</p> <p>Parameter PORT, IPADDR contains partner port number and IP address.</p>
0	0	7000	First call with REQ=0. Job has not assigned
0	0	7001	First call with REQ=1. Job has assigned and is in process
0	0	7002	Interim call (REQ irrelevant). Job is in process
0	1	8001	Parameter ID is invalid
0	1	8002	Connection with ID is not configured or Invalid Connection type
0	1	8003	Connection is not yet established

Communication - Ethernet

S7-program sample for using the SFB 126

Purpose: Query connection status

```

UN    M    600.0    // TRUE = Request to query connection status
BEB                                     // otherwise end the function

CALL #TSTATUS
ID    :=1           // Connection ID number, which is configured in ConfigStage
DONE  :=M600.1      // TRUE = Successful
ERROR :=M600.3      // TRUE = Error occurred
STATUS:=MW602       // Result / return code
PORT  :=MW604       // Partner port number
IPADDR:=MD606       // Partner IP address

O     M     600.1    // if successful
O     M     600.3    // or error occurred, then
R     M     600.0    // reset the request signal

```

UDP Send/Receive Connection (RFC 768)

INSEVIS PLC provides the following system function blocks (SFB) for exchanging data through the user program with other Ethernet-capable communications partner via UDP (RFC 768) protocol.

- SFB 127 "USEND" Send data via UDP Send/Receive connection
- SFB 128 "URECV" Receiving data via UDP Send/Receive connection

The connection must be configured in ConfigStage and downloaded to PLC.



ATTENTION:

In configuration of passive connections the local port number must assigned and unique and unequal to 0.

Communication - Ethernet

Send data via UDP with SFB 127 "USEND"

You can send data to a remote Partner by using SFB 127 "USEND" via UDP (RFC 768) protocol. It works asynchronously, in other words, its processing extends over several SFB calls. It can send until 1472 bytes.

The parameters ID, LEN, R_PORT, R_ADDR, DATA are transferred and data transfer starts on rising edge at REQ. The new job can only be accepted again after the last job is completed and on rising edge at REQ. Job completion is signaled by DONE or ERROR parameter.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	Request the data exchange at a rising edge
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
LEN	INPUT	INT	Number of bytes to send
R_PORT	INPUT	WORD	Partner Port/TSAP number e.g: port 2000 _{dec} , enter as W#16#7D0
R_ADDR	INPUT	DWORD	Partner IP Address e.g: Address 192.168.80.50, enter as DW#16#C0A85032
DONE	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
BUSY	OUTPUT	BOOL	TRUE=Job is in Process
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code
DATA	IN_OUT	ANY	ANY- pointer to the send area. Only Input image (I), Output image (Q), Bit Memory (M), Data Block (DB) area allowed.

DONE, BUSY, ERROR and STATUS parameters meaning

DONE	BUSY	ERROR	STATUS W#16#..	Description
1	0	0	0	Job has successfully completed.
0	0	0	7000	First call with REQ=0. Job has not assigned
0	1	0	7001	First call with REQ=1. Job has assigned and is in process
0	1	0	7002	Interim call (REQ irrelevant). Job is in process
0	0	0	7003	Job assignment is processing in a priority class with lower priority (Nesting call)
0	0	1	8001	Parameter ID is invalid
0	0	1	8002	Connection with ID is not configured or Invalid Connection type
0	0	1	8003	Parameter DATA is invalid. Only I, Q, M, DB area allowed
0	0	1	8004	Parameter DATA is invalid. e.g DB not loaded.
0	0	1	8005	Parameter LEN is 0 or larger than specified in Parameter DATA or larger than 1472
0	0	1	8006	Parameter R_PORT or R_IPADDR is invalid.
0	0	1	8007	Job failed due to Communication problem (cable disconnected)

S7-program sample for using the SFB 127

Purpose: Send 20 bytes starting MB200 via UDP to Partner with IP address 192.168.80.52 and UDP port number 2000.

```
CALL SFB127, DB127
REQ  :=M600.0           // TRUE = Send data to partner
ID   :=1                // Connection ID number, which is configured in ConfigStage
LEN  :=20                // 20 bytes data to send
R_PORT:=W#16#7D0         // Partner UDP port number e.g 2000 (dez)
R_ADDR:=DW#16#C0A85034   // Partner IP address e.g 192.168.80.52
DONE  :=M600.1           // TRUE = Successful
BUSY  :=M600.2           // TRUE = In process
ERROR :=M600.3           // TRUE = Error occurred
STATUS:=MW602            // Result / return code
DATA  :=P#M 200.0 BYTE 20 // 20 byte data starting from MB200 in local CPU

O      M      600.1       // if successful sent to
O      M      600.3       // or error occurred, then
R      M      600.0       // reset the request signal
```

Receiving data via UDP with SFB 128 "URECV"

You can receive data from a remote Partner by using SFB 128 "URECV" via UDP (RFC 768) protocol. It works asynchronously, which means the job processing extends over several SFB calls.

You can specify the remote partner by assigning specific address and port number of the remote partner in R_ADDR and/or R_PORT parameter with EN_R = TRUE. E.g, if you want to receive data only from the partner with IP address 192.168.80.60 and any port number, then you can set

```
L      DW#16#C0A8503C    // partner IP address
T      MD 54
L      0                  // port number: 0=Any port
T      MW 52
SET
=      M0.0               // start receive
```

...

```
CALL SFB128, DB128
EN_R   := M0.0
ID     := 1
LEN    := 0
NDR    := M0.1
ERROR  := M0.2
STATUS := MW2
LEN_R  := MW4
R_PORT := MW52
R_ADDR := MD54
DATA   := P#DB100.DBX0.0 BYTE 100
```

If you assign DW#16#0 to R_ADDR parameter, it receives UDP packets from any remote partner whereas assigned 0 to R_PORT, from any port number.

Please note, that R_PORT and R_ADDR parameters are input and output parameters. After receiving data, it overwrites the parameters with actual remote address which was sent, and sets NDR parameter to TRUE.

The parameters ID, LEN, R_PORT, R_ADDR, DATA are evaluated and starts the receiving on EN_R = TRUE.

The new job can only be accepted again after the last job is completed and EN_R = TRUE.

Job completion is signaled by NDR or ERROR parameter.

Communication - Ethernet

If the length of the data to be received (LEN parameter) is larger than the actually received data, then SFB will copy the received data into the receiver area (DATA parameter) and NDR become TRUE only after the length specified by the LEN parameter has been reached.

If the length of the data to be received (LEN parameter) is less than actually received data, SFB will copy as many bytes into the receiver area (DATA parameter) as specified in the LEN parameter and sets TRUE to NDR and writes to LEN_R the length of data which has written. With each additional call, you will receive rest of data.

If the LEN parameter is 0, then SFB will copy the received data into receiver area (DATA parameter) and immediately sets NDR as TRUE.

You can specify the partner by assigning R_PORT and/or R_ADDR unequal to 0 and by call the SFB with EN_R = TRUE. If you assign 0 to R_PORT and/or R_ADDR, it receives data from partner with any Port number and/or any IP address.

Parameter	Declaration	Data type	Description
EN_R	INPUT	BOOL	Enable the receiving data
ID	INPUT	INT	Connection ID number, which configured in ConfigStage
LEN	INPUT	INT	Expected receiving data size in bytes
NDR	OUTPUT	BOOL	TRUE=Job completed successfully, FALSE=job is in process or failed.
BUSY	OUTPUT	BOOL	TRUE=Job is in process
ERROR	OUTPUT	BOOL	TRUE=Job failed
STATUS	OUTPUT	WORD	Job status code
LEN_R	OUTPUT	INT	Number of bytes actually received
R_PORT	IN_OUT	WORD	Partner Port/TSAP number e.g: port 2000 _{dec} , enter or receive as W#16#7D0
R_ADDR	IN_OUT	DWORD	Partner IP Address e.g: Address 192.168.80.50, enter or receive as DW#16#C0A85032
DATA	IN_OUT	ANY	ANY- pointer to the receiving area. Only Input image (I), Output image (Q), Bit Memory (M), Data Block (DB) area allowed.

DONE, BUSY, ERROR and STATUS parameters meaning

DONE	BUSY	ERROR	STATUS W#16#..	Description
1	0	0	0	Job has successfully completed.
0	0	0	7000	First call with EN_R=0. Job has not assigned
0	1	0	7001	First call with EN_R=1. Job has assigned and is in process
0	1	0	7002	Interim call (EN_R irrelevant). Job is in process
0	0	0	7003	Job assignment is processing in a priority class with lower priority (Nesting call)
0	0	1	8001	Parameter ID is invalid
0	0	1	8002	Connection with ID is not configured or Invalid Connection type
0	0	1	8003	Parameter DATA is invalid. Only I, Q, M, DB area allowed
0	0	1	8004	Parameter DATA is invalid. e.g., DB not loaded.
0	0	1	8005	Parameter LEN is larger than specified in Parameter DATA or 1472 bytes

Communication - Ethernet

S7-program sample for using the SFB 128

Purpose: Receive receive via UDP from any host (unspecified partner) and write into data block DB200 with 2000 bytes large.

```

L      0                                // Receive data
T      MW      606                      // from any Partner UDP port
L      0                                // Receive from
T      MD      608                      // from any Partner

CALL   SFB128, DB128

EN_R   :=M600.0                        // TRUE = Enable to receive data
ID     :=1                             // Connection ID number,
                                         // which is configured in ConfigStage

LEN     :=0                             // 0 = When data received,
                                         // then immediately set NDR and copy to DATA

NDR     :=M600.1                        // TRUE = New data available
BUSY    :=M600.2                        // TRUE = In process
ERROR   :=M600.3                        // TRUE = Error occurred
STATUS:=MW602                          // Result / return code
LEN_R   :=MW604                          // Number of bytes copied to DATA
R_PORT:=MW606                           // Partner UDP port number
R_ADDR:=MD608                           // Partner IP address
DATA    :=P#DB200.DBX 0.0 BYTE 2000    // Receive buffer area

O      M      600.1                     // if successful received to
O      M      600.3                     // or error occurred, then
R      M      600.0                     // reset the request signal

UN     M      600.1                     // if data not received
BEB                                         // end the function
L      MD      608                       // Data received from Partner with IP address
L      MW      606                       // and UDP port number

```

Communication - Profinet IO Controller

Profinet IO Controller

INSEVIS- PLCs with CPU-T can be equipped with Profinet IO Controller optionally.



Hint: The CPU configuration can be made either in the Simatic® Manager or in the ConfigStage. The download of the Simatic® Manager overwrites all SDBs completely. Please do config Profinet and CPU **FIRST** with the Siemens-tools and download it into the PLC. Than configure all the other configurations (Ethernet, RS232/485, Modbus, CAN and (de)-central periphery and alternatively the CPU) with the ConfigStage and download it **LATER**. This download does overwrite only the configured SDBs and keeps the Profinet configuration.

Connecting

When Profinet-option is included, there are 2 additional RJ45-interfaces (Profinet 1 and 2) visible. These plugs can be used as switch with common IP-adress.

Configuration

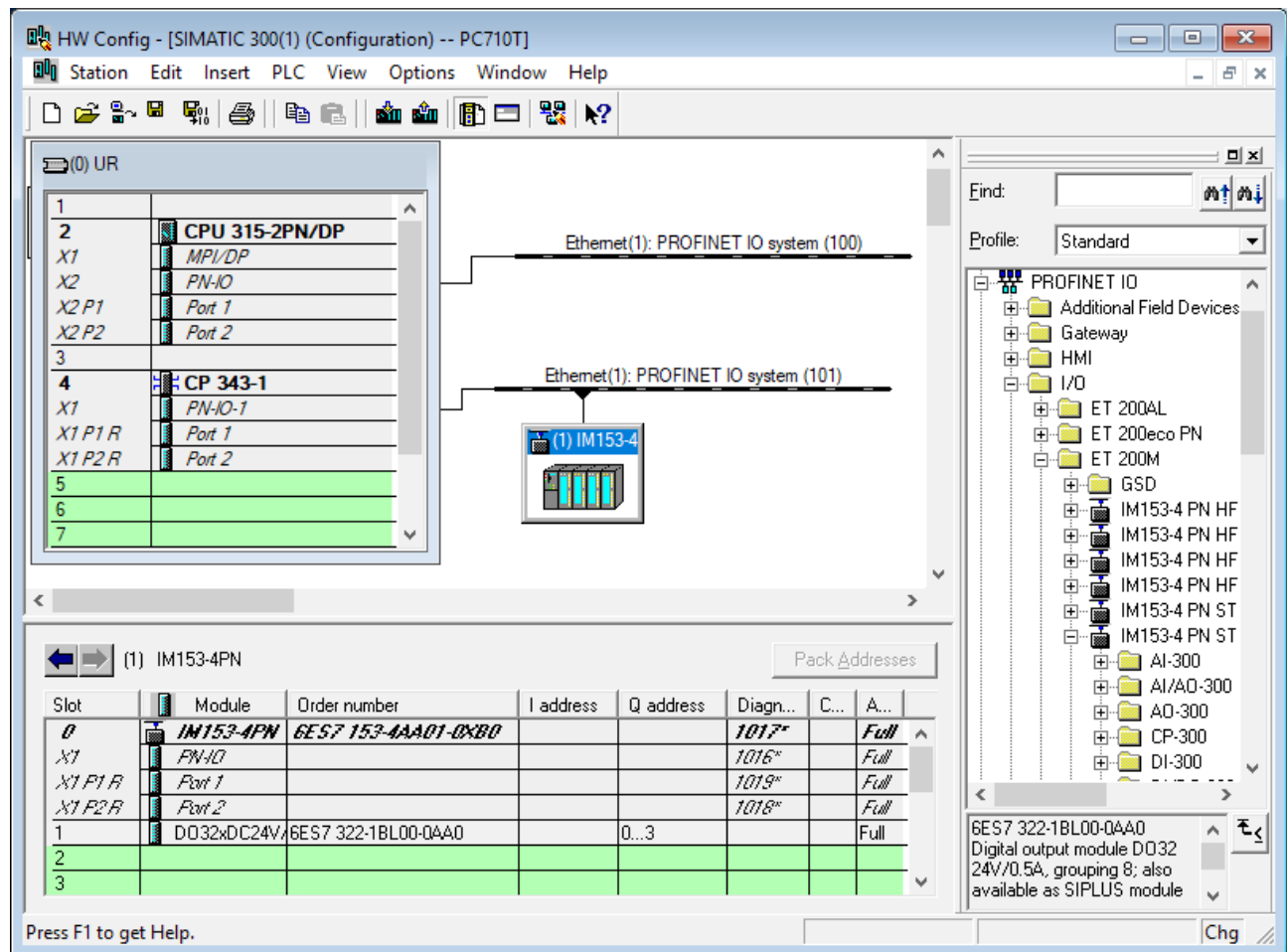
Profinet is available as an additional option to the CPU-T. To realize this option while configuring in the SimaticManager / TIA-Portal, it is necessary to offer the Profinet as an option as well and this is made with an CP 343-1 (Siemens-order-no. 6GK7 343-1EX30-0XE0 with version V2.4).

The configuration of the Profinet in INSEVIS-projects is simple:

- Select CPU 315-2 PN/DP (Siemens-order-no. 6ES7 315-2EH14-0AB0) and place it on slot 2
- Select CP 343-1 (Siemens-order-no. 6GK7 343-1EX30-0XE0 with Version V2.4) and place it on slot 4.

Communication - Profinet IO Controller

Then you can select your Profinet-devices and add it by drag'n drop as you are used to it.



Despite of INSEVIS-Profinet-interface is configured like an additional CP343-1, the handling for S7-programming is like the onboard-Profinet on the S7-315-2 PN/DP.

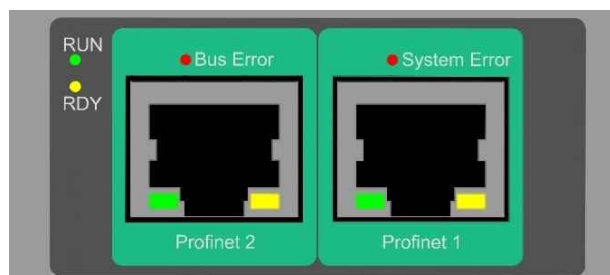
This means in detail, that:

- It is not necessary to configure a process data exchange between CPU and CP
- The alarm-OBs will be called by CPU by it self

Communication - Profinet IO Controller

Status-LEDs

Below the description of the Status-LEDs:



LED	Color	LED-Status	Description
RUN	green	on	Profinet system running
		off	Profinet system down (e.g. no Profinet configuration loaded)
RDY	yellow	blinking	Profinet system error: Boot Error
		on	Profinet system error: Boot Loader Waiting
		off	Profinet system down (Boot Error)
RUN + RDY	green + yellow	off	Power fail / damaged (if BOTH LEDs are off)
Bus Error	red	blinking	Profinet configuration error (e.g. not all IO-devices connected)
		on	Profinet connection error No Link (z. B. no cable inserted)
		off	No bus errors
System Error	red	blinking	Profinet system error (e.g. invalid configuration, watchdog)
		on	Profinet diagnostic alarm reported (from min. one IO-device)
		off	No bus errors
Bus Error + System Error	red + red	on	Non or invalid Profinet-Master license in the device

Communication - Profibus DP-Master

Profibus DP

INSEVIS- PLCs with CPU-V and -P can be equipped with Profibus DP optionally. Therewith only the version DP-V0 is supported.



Hint: The CPU configuration can be made either in the Simatic® Manager or in the ConfigStage. The download of the Simatic® Manager overwrites all SDBs completely. Please do config Profibus and CPU **FIRST** and download it into the PLC. Than configure all the other configurations (Ethernet, RS232/485, Modbus, CAN and (de-)central periphery and alternatively the CPU) with the ConfigStage and download it **LATER**. This download does overwrite only the configured SDBs and keeps the Profibus configuration.

Connecting

INSEVIS	description, other references
VP	5V-supply for repeater (only connect on demand)
Ctrl	Control signal for repeater (only connect on demand)
P	Signal RxD/TxD-P; PROFIBUS data line B; Siemens color code: red
N	Signal RxD/TxD-N; PROFIBUS data line A; Siemens color code: green
GND	potential separated Profibus-ground

Profibus DP-Master

INSEVIS PLC offers the DPV0 Master as an optional module to be configured with the Siemens-Simatic Manager.

Configure the DP-Master

The Hardware configurator of Simatic is used to configure the DP Master as same as configuring the integrated DP interface of CPU315-2PN/DP as DP Master.

Specifics

The only deviations to the S7-315-2PN/DP- target PLC while using of Profibus are mentioned below:

- The Profibus-DP- master starts parameterization and configuration the Profibus-DP- slaves slaves first at NewStart after configuration download or PowerOn.
- Profibus Master diagnose address is fix 2047 (W#16#7FF)
- The specific deviations at OB 86 and SFC 11 are described in Chapter „System functions“.



VIDEO-tutorial available

There is available a link to the corresponding YouTube®-video at the INSEVIS-download website for use under Simatic®-Manager.

Communication - Profibus DP-Slave

Profibus DP-Slave

INSEVIS PLC offers the DP V0 Slave as an optional module to be configured with the Siemens-Simatic Manager.

Feature:

- max. 244 bytes of cyclic input data
- max. 244 bytes of cyclic output data
- max. 24 configurable slots
- GSD (device database file) "HIL0C0F.GSD" for DP-Master configuration
- DP V0

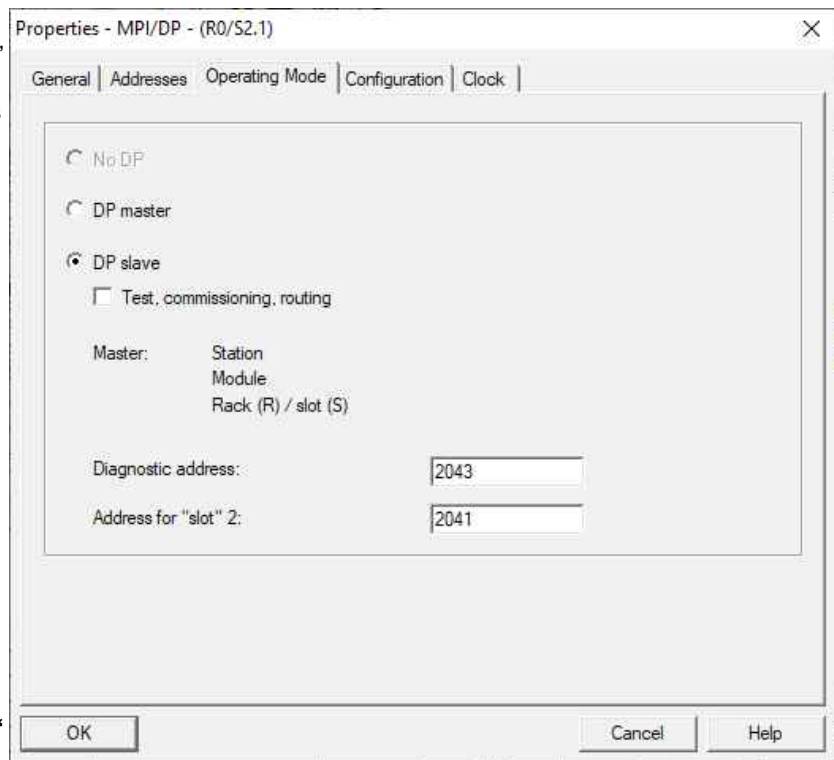
Differences compare to CPU315-2PN/DP:

- No active service (Testing, Routing)
- No direct data exchange (DX) service

Configure the DP-Slave

The Simatic hardware configurator is used to configure the DP-slave as same as configuring the integrated DP interface of CPU315-2PN/DP as DP-slave.

- Set integrated MPI/DP interface type as **PROFIBUS** under Properties->"General" tab
- Set DP-Slave-address under „Properties Profibus interface“ tab (opens automatically, otherwise press „Properties“)
 - Tab „Parameter“
 - Set the DP-Slave-Address (Hint Address 2 is often reserved for PG!) and
 - Set up the subnet as „not networked“ to prevent address conflicts
- confirm with the „OK“ button
- Set the interface as "DP-slave" under Properties
 - "Operating Mode" tab



Note:

- **Do not select "Test, commissioning, routing"!**
- **Do only use „Subnet / not networked“ to prevent address conflicts!**
- Via the "Diagnostic address" the DP-slave diagnoses the failure or return of the DP Master. In case of DP master failure / return, (or connector unplugged/ plugged), operating system calls OB86 as station failure (event W#16#39C4) / return (event W#16#38C4). If OB86 is not loaded, PLC goes into STOP mode.
- In case of DP-slave optional module is not placed but DP interface configured via hardware configurator, operating system does not make any entries in diagnostic buffer.
- In case of DP-slave module is placed, but DP interface configured as master via hardware configurator, operating system makes the following entry in diagnostic buffer:

Event:	16# BF02
OB:	16# 00
PK:	16# FF
DatID 1/2:	16# 00 01
Additional info 1 / 2 / 3:	16# 0000 0000 0000
- In case of CPU of DP-Slave mode changes (RUN ↔ STOP), the slave sends diagnose telegram to DP-Master (structure is described later on)



VIDEO-tutorial available

There is available a link to the corresponding YouTube®-video at the INSEVIS-download website for use under Simatic®-Manager.

Communication - Profibus DP-Slave

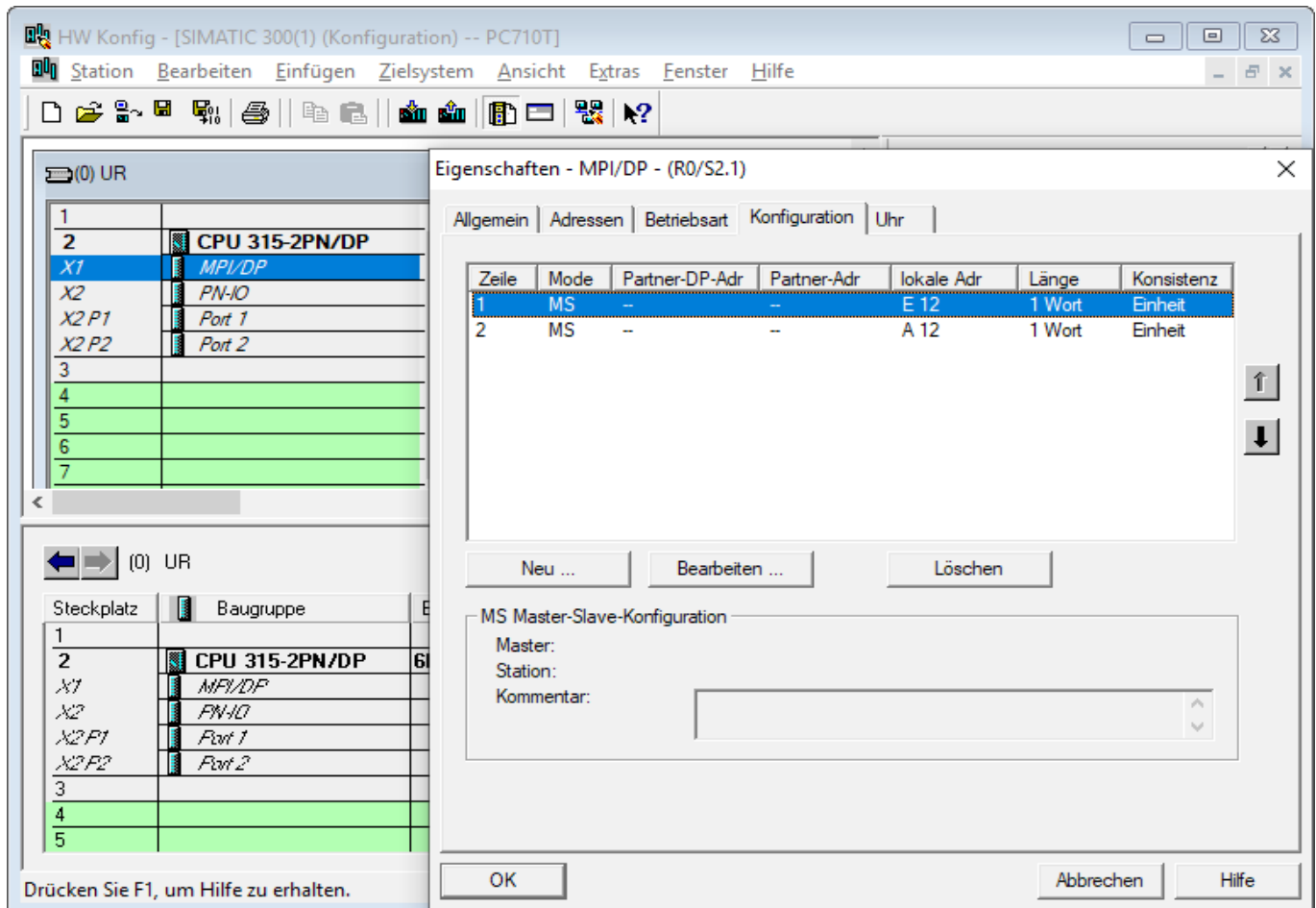
Configure the input / output slots of DP-slave

The DP Master does not access the connected inputs/ outputs of the INSEVIS PLC with integrated DP-Slave, but accesses a transitional area (slots) in the input/ output address space of the INSEVIS PLC - CPU. The user program for the INSEVIS PLC must take care of data exchange between the address area and the inputs/outputs.

Note:

- The configured input/output areas for data exchange between master and slaves must not be "occupied" by I/O modules.

The input / output virtual slots under Properties->"Configuration" tab in Hardware configurator.



Note:

- Maximum 21 virtual slots (rows) can be configured
- "Direct data exchange" (DX) mode is not supported, only Master-Slave configuration (MS) is supported
- In case of DP Master failure (or cable disconnected), and periphery direct access (e.g. L PEW 2) to the configured slots causes the periphery access error OB122. If OB122 is not programmed, PLC goes to STOP.

Communication - Profibus DP-Slave

DP Master configuration for INSEVIS Profibus DP-slave

Please note that, in DP Master configuration to configure the INSEVIS PLC with DP-slave interface, use device database (GSD) file "HIL0C0F.GSD". **Do not use "CONFIGURED STATIONS → CPU 31X" in Hardware configurator.**

- Drag "COMX DP/DPS" modul from "Hardware Catalog" → PROFIBUS DP → "Additional Field Devices" → "General"
- Drop in DP Master system

HW Config - [SIMATIC 300(1) (Configuration) -- PC710T]

Station Edit Insert PLC View Options Window Help

PROFIBUS(1): DP-Mastersystem (1)

(3) COMX DP/DPS

Slot	DP ID ...	Order Number / Designation	I Address	Q Address	Comment
1	0	blank space			
2	0	blank space			
3	0	blank space			
4	160	1 Byte Out		256	
5	224	1 Word Out		257...258	
6	208	1 Word In	256...257		
7	144	1 Byte In	258		
8					
9					
10					

Press F1 to get Help.

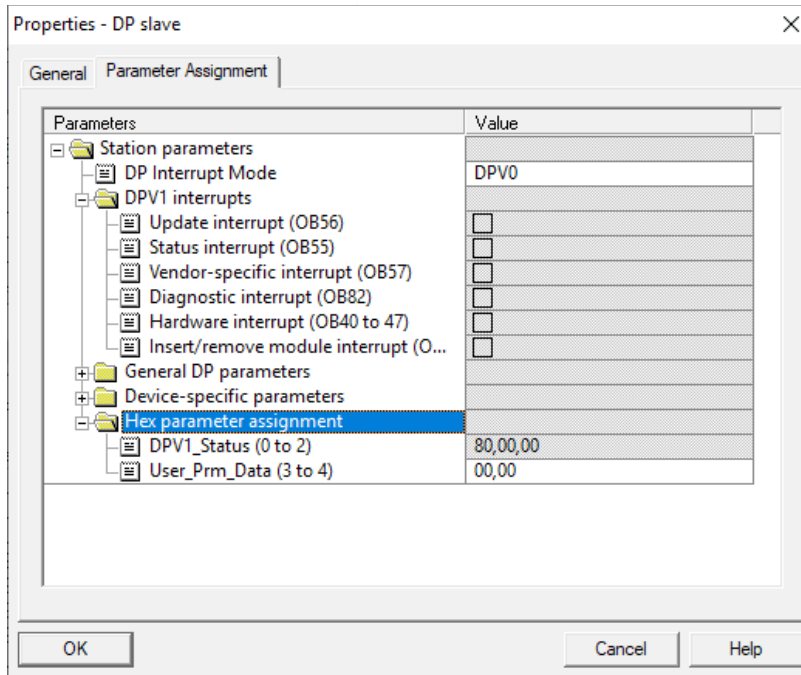
Field devices defined via device database files as DP slaves

Note:

- First 3 slots must be occupied blank space modul
- From 4th slot, the virtual input/outputs can be configured by **drag'n drop** (Alternatively the Universal module can be moved to a free slot and be parameterized completely, what is more complicated)
- Maximum 21 slots can be configured (exclusive first 3 blank space)

Communication - Profibus DP-Slave

Set DP-Slave parameter in properties → "Parameter Assignment" tab



Note:

- Uncheck all DPV1 interrupts
- Set "DP Interrupt mode" as DPV0
- Set "User_Prm_Data (3 to 4)" as "00, 00"

Structure of DP-Slave diagnostic data

DP-Slave diagnostic data consists of Standard diagnostic data (first 6 byte), Module diagnostic (Identification related diagnostics) and Modul status (Device related diagnostics).

Byte 0	Station status byte 1
Byte 1	Station status byte 2
Byte 2	Station status byte 3
Byte 3	Master Profibus address
Byte 4	Manufacturer ID (High byte) 0C
Byte 5	Manufacturer ID (Low byte) 0F
Byte 6	Module diagnostic (Identification related diagnostics)
...	
Byte x - 1	(length depends on the number of the configured slots)
Byte x	Module status (Device related diagnostics)
...	
Byte y	(length depends on the number of the configured slots)

Communication - Profibus DP-Slave

Station status byte 1

Bit	Description	Remedy
0	1: Station does not exists	Check the Slave Address Check the bus connector
1	1: Not ready for data exchange	
2	1: Configuration data sent by DP-Master inconsistent	Check the configuration
3	Diagnostic interrupt 0: DP-Slave CPU in RUN 1: DP-Slave CPU is in STOP	Analyse Slave CPU's Diagnose buffer
4	1: Function is not supported	Check the configuration
5	Always 0	
6	1: DP-Slave type inconsistent	Check parameter assignment
7	1: DP-Slave configured by other DP-Master	

Station status byte 2

Bit	Description
0	1: DP-Slave requires new parameters and configuration
1	1: A diagnostic message was received. The DP-Slave cannot resume operation until the error has been cleared (static diagnostic message).
2	Always 1
3	1: The watchdog monitor is enabled on DP-Slave.
4	1: DP-Slave has received control command "FREEZE".
5	1: DP-Slave has received control command "SYNC".
6	Always 0
7	1: DP-Slave is disabled, it has been excluded from cyclic processing.

Station status byte 3

Bit	Description
0..6	Always 0
7	1: The incoming diagnostic messages (sent by DP-slave) exceed the memory capacity of the DP-master. The DP master cannot write all diagnostic messages sent by the DP-Slave to its diagnostic buffer

Master PROFIBUS address

Bit	Description
0..7	DP Master address, which has configured the DP-Slave
	FF: DP-Slave was not configured by a DP-Master

Manufacturer ID

Byte 4	Byte 5
0C_H	0F_H

Communication - Profibus DP-Slave

Module diagnostics (Identification related diagnostics)

Byte 6	Bit 6..7 = 01 : Code for module diagnostics Bit 0..5 = Length of module diagnostics inclusive byte 6. (dependent on the number of configured slots up to 4 byte)
Byte 7	Bit 0 = Always 0 Bit 1 = 1 : Slave CPU in STOP 0 : Slave CPU in RUN Bit 2 = Always 0 Bit 3 = Entry for slot 4 Bit 4 = Entry for slot 5 Bit 5 = Entry for slot 6 Bit 6 = Entry for slot 7 Bit 7 = Entry for slot 8
Byte 8	Bit 0 .. 7 = Entry for slot 9 to 16
Byte 9	Bit 0 .. 7 = Entry for slot 17 to 24

Module status (Device related diagnostics)

The module status indicates the status of configured slots.

Byte x	Bit 6..7 = 00 Code for module status Bit 0..5 = Length of module status inclusive byte x. (dependent on the number of configured slots up to 10 byte)
Byte x + 1	82_H : Code for module status
Byte x + 2	Always 0
Byte x + 3	Always 0
Byte x + 4	Bit 0..1 = Always 0 Bit 2..3 = CPU status, 00 : Ok, 01 : Fault Bit 4..5 = Always 0 Bit 6..7 = Slot 4 status, 00 : Ok, 01 : Fault
Byte x + 5	Bit 0..1 = Slot 5 status ... Bit 6..7 = Slot 8 status
...	...
Byte y	Bit 0..1 = Slot 21 status Bit 2..3 = Slot 22 status Bit 4..5 = Slot 23 status Bit 6..7 = Slot 24 status

Communication - Profibus DP-Slave

DP-Slave address assign with SFC 75 „SET_ADDR“

INSEVIS PLC provides the SFC 75 „SET_ADDR“ to change the DP-Slave address at runtime.

General Hints:

- It works asynchronously, in other words, its processing extends over several SFC calls
- It does not update the newly assigned address into system configuration (SDB).
- By STOP and next NEWSTART, the new address is preserved
- By next POWERON, DP- Slave assigned with the originally configured DP- address.

Hints for DP- Master:

- DP-Slave with old address diagnosed as station failure
- DP-Slave with new address diagnosed as station return

Hints for DP- Slave CPU (INSEVIS PLC):

- DP- Master diagnosed as failure und return with diagnose address (configured in DP- Slave). DP- Slave CPU calls OB86. If OB86 is not programmed, CPU goes into STOP.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE=Start the address assignment
LADDR	INPUT	WORD	DP-Slave interface address, fixed W#16#7FF
ADDR	INPUT	BYTE	New DP-Address
RET_VAL	OUTPUT	INT	Error code 0000 _H No error 7000 _H First call with REQ=FALSE, No assignment 7001 _H First call with REQ=TRUE, Assignment is in process 7002 _H Intermediate call, Parameters are evaluated 8090 _H LADDR parameter invalid, must be fixed W#16#7FF 8091 _H ADDR parameter invalid 809b _H Interface is not configured as DP-Slave 80C3 _H SFC called in lower priority level
BUSY	OUTPUT	BOOL	TRUE=Job is in process

Communication - Profibus DP-Slave

DP-Slave address and baud rate assign with SFC 76 „SET_DPS“

INSEVIS PLC provides the SFC 76 „SET_DPS“ to change the DP-Slave address and baud rate at runtime.

- It works asynchronously, in other words, its processing extends over several SFC calls
- It does not update the newly assigned address into system configuration (SDB).
- By STOP and next NEWSTART, the new address and baud rate is preserved.
- By next POWERON, DP-Slave assigned with the originally configured DP-Address and baud rate.

In DP-Master:

- DP-Slave with old address diagnosed as Station failure
- DP-Slave with new address diagnosed as Station return

In DP-Slave CPU:

- DP-Master diagnosed as Failure and Return with Diagnose address (configured in DP-Slave). DP-Slave CPU calls OB86. If OB86 is not programmed, CPU goes to STOP.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE=Start the address assignment
LADDR	INPUT	WORD	DP-Slave interface address, fixed W#16#7FF
ADDR	INPUT	BYTE	New DP-Address
BAUD	INPUT	BYTE	New DP baud rate B#16#00 9.6 kbit/s B#16#01 19.2 kbit/s B#16#02 93.75 kbit/s B#16#03 187.5 kbit/s B#16#04 500 kbit/s B#16#06 1.5 Mbit/s B#16#07 3 Mbit/s B#16#08 6 Mbit/s B#16#09 12 Mbit/s B#16#0a 31.25 kbit/s B#16#0b 45.45 kbit/s B#16#0f Auto
RET_VAL	OUTPUT	INT	Error code 0000 _H No error 7000 _H First call with REQ=FALSE, No assignment 7001 _H First call with REQ=TRUE, Assignment is in process 7002 _H Intermediate call, Parameters are evaluated 8090 _H LADDR parameter invalid, must be fixed W#16#7FF 8091 _H ADDR parameter invalid 8092 _H BAUD parameter invalid 809b _H Interface is not configured as DP-Slave 80C3 _H SFC called in lower priority level
BUSY	OUTPUT	BOOL	TRUE=Job is in process

UART

The INSEVIS UART interface allows a free programmable serial communication with STEP®7 as well as a comfortable communication in the background of standard protocols, to be configured with the Software tool „ConfigStage“. Common parameters (Baudrate, data format) have to be assigned in the „ConfigStage“ anyway. The operating system handles send and receive and buffers data. The following system functions allow access to these data buffers. Send and receive runs independent from S7, e.g. in stop-state too.



VIDEO-tutorial available

There are available samples how to assign a serial interface by ConfigStage and a link to the corresponding YouTube®-video at the INSEVIS-download website.

Sending data with SFB 60 "SEND_PTP", free ASCII

Data from a data block will be transferred with SFB 60.

The activation of the sending procedure starts after calling the data block and a positive edge on the control input REQ.

The range of data to send will be assigned by DB-no. and start address in DATA. Type- and length information of the ANY-pointer will be ignored.

LEN determines the length of the data block to send at the initiating call and returns always the current transmitted length. To enable the SFB to process the job it has to be called with R = FALSE.

At a positive edge on control input R the current transfer is aborted and the SFB is reset back into its basic state. A aborted job is terminated with an error message (STATUS- output).

With UART the interface will be selected.

The return value signalizes, if the data are entered complete into the send buffer or if the call needs to be repeated.

Either DONE will be set to TRUE, if the task was finished without any error, or ERROR will be set to TRUE, if the task was finished with any error. If the task has passed with DONE=TRUE it means, that data were sent to the communication partner. It is not sure, that these data were received from the communication partner as well.

In case of any error or warning the CPU shows in STATUS the corresponding event-no. DONE or ERROR/STATUS will be given at RESET of the SFB (R=TRUE) as well.

Parameter	Declaration	Data type	Offset	Description
REQ	INPUT	BOOL	0.0	"Request": Activates data transfer on a rising edge
R	INPUT	BOOL	0.1	"Reset": Job is aborted. Transmission is locked.
LADDR	INPUT	WORD	2	UART-index 0 = RS232, 1 = RS485
DONE	OUTPUT	BOOL	4.0	FALSE: inactive or sending is still executed. TRUE: task is executed faultless The parameter is only set for one call.
ERROR	OUTPUT	BOOL	4.1	Status parameter The parameter is only set for one call.
STATUS	OUTPUT	WORD	6	0303hex = ANY- pointer to data area not valid 051Dhex = reset by input R
SD_1	INOUT	ANY	8	ANY- pointer in data area of send data
LEN	INOUT	INT	18	Length in bytes

The length of data to send is only limited by the parametrized data area. Consistent data will be transferred with variable length (max. 256 byte). The first consistent block has a length of 256 byte.

Communication - UART

Receive data with SFB 61 "RCV_PTP", free ASCII

Received data will be transferred with SFB 61 into a data block.

The data block is ready to receive after a call with value TRUE on input EN_R.

A running transfer can be canceled with signal state FALSE on parameter EN_R. A canceled task will be finished with an error message (STATUS- output).

The area of data to receive will be assigned by DB-no. and start address in RD_1. Type- and length information of the ANY-pointers will be ignored.

LEN determines the length of the data block to receive at the initiating call and returns always the current transmitted length. An initiating call is the next call following a call with R = TRUE or if LEN bytes are reached with returning NDR = TRUE.

Therewith the SFB can execute the task, it has to be called with R = FALSE.

At a positive edge on control input R the current transfer is aborted and the SFB is reset back into its basic state. An aborted job is terminated with an error message (STATUS- output).

The parameter UART selects the interface.

If the task was finished without any error NDR will be set to TRUE, if the task was finished with any error ERROR will be set to TRUE.

In case of any error or warning the CPU shows in STATUS the corresponding event-no. NDR or ERROR/STATUS will be given at RESET of the SFB (R=TRUE) as well.

Mode of operation:

The operating systems receive buffer is a ring buffer and has a size of 2 kByte and will be used with its whole length for user data. The SFB61 fetches data and copies this data linear into the specified S7 data area.

Using parameter LEN and NDR assumes to receive a known number of bytes (LEN). Receiving the last byte (byte[LEN-1]) NDR will be set TRUE and the next call starts again to write data into RD_1 from start position RD_1[0].

The Parameter EN_R is static and enables/disables the SFB, it does not disturb the receive and does not clear any buffer.

The Parameter R resets the internal state of SFB only and does not clear any buffer as well.

i.e.: Expecting disturbing data in the buffer, the SFB should be called to receive these data until LEN =0 will be returned.

Example: To receive data telegrams with unknown length set LEN to the maximum length. The SFB61 returns in LEN meanwhile the number of current received bytes. Now the user program can interpret the valid bytes in RD_1. Detecting the end of data, the task must be stopped by a call with R = TRUE.

Alternatively all bytes can be read separate using LEN = 1.

Hint: LEN must not be greater then the data area RD_1

Parameter	Declaration	Data type	Offset	Description
EN_R	INPUT	BOOL	0.0	"Enable receive": enables receiving
R	INPUT	BOOL	0.1	"Reset": Job is aborted.
LADDR	INPUT	WORD	2	UART-Index 0 = RS232, 1 = RS485
NDR	OUTPUT	BOOL	4.0	FALSE: Job has not been started or is still being executed TRUE: Job was completed successfully, data accepted The parameter is only set for one call.
ERROR	OUTPUT	BOOL	4.1	Job completed with error. The parameter is only set for one call.
STATUS	OUTPUT	WORD	6	0303hex = ANY- pointer to data area not valid 051Dhex = reset by input R
RD_1	INOUT	ANY	8	ANY- pointer in data area of receive data
LEN	INOUT	INT	18	Length in bytes

Communication - UART

Sending data with SFB 60 "SEND_PTP", Modbus RTU (Client)

To use the protocol Modbus RTU as Client (Master) it is necessary to config it all in the ConfigStage:

- protocol RS485 → to "RTU"
- no Modbus-RTU-Server enabled

A Modbus-RTU PDU from a data block will be transferred with SFB 60.

The activation of the sending procedure starts after calling the data block and a TRUE on the control input REQ.

The range of data to send will be assigned by DB-no. and start address in SD_1. Type- and length information of the ANY-pointer will be ignored.

In LEN the length of the data block to send will be assigned and the length of the actual transferred data will be given back. (This should be always the same lengths, because of Modbus-telegrams need to be transferred as a single, not interrupted block!) The length value refers to the Modbus user data (UID + RTU) without checksum. The operating system writes the checksum 2 bytes longer than the assigned length is. That's why the data area must be 2 bytes longer than the assigned sending length is.

To enable the SFB to process the job it has to be called with R = FALSE.

The return value signalizes, if the data are entered complete into the send buffer or if the call needs to be repeated.

Either DONE will be set to TRUE, if the task was finished without any error, or ERROR will be set to TRUE, if the task was finished with any error. If the task has passed with DONE=TRUE it means, that data were sent to the communication partner. It is not sure, that these data were received from the communication partner as well.

In case of any error or warning the CPU shows in STATUS the corresponding event-no. DONE or ERROR/STATUS will be given at RESET of the SFB (R=TRUE) as well.

Parameter	Declaration	Data type	Offset	Description
REQ	INPUT	BOOL	0.0	"Request": Activates data transfer while TRUE
R	INPUT	BOOL	0.1	"Reset": sending is disabled (Not necessary because of compatibility reasons)
LADDR	INPUT	WORD	2	only 1 = RS485
DONE	OUTPUT	BOOL	4.0	FALSE: inactive or sending is still executed. (not conform to Modbus timing rules) TRUE: task is executed faultless The parameter is only set for one call.
ERROR	OUTPUT	BOOL	4.1	„Error“ failure state in STATUS The parameter is only set for one call.
STATUS	OUTPUT	WORD	6	0303hex = ANY- pointer to data area not valid 051Dhex = reset by input R
SD_1	INOUT	ANY	8	ANY- pointer in data area of send data - UID (1 byte target address) - PDU: Modbus-command (1 Byte) ... Modbus-address, number ... (specific for each commando) -CRC (reserving 2 byte space in data area)
LEN	INOUT	INT	18	Length of user data (without checksum) in bytes

Communication - UART

Receive data with SFB 61 "RCV_PTP", Modbus RTU (Client)

To use the protocol Modbus RTU as Client (Master) it is necessary to config it all in the ConfigStage (see SFB60)
Received data will be transferred with SFB 61 into a data block.

The activation of the sending procedure starts after calling the data block and a TRUE on the control input REQ.
The range of data to send will be assigned by DB-no. and start address in RD_1. Type- and length information of the ANY-pointer will be ignored.

In LEN the length of the data block to receive will be assigned and the length of the actual transferred data will be given back.
The length value refers to the Modbus user data (UID + RTU) without checksum. The operating system writes the checksum 2 bytes longer than the assigned length is. That's why the data area must be 2 bytes longer than the assigned sending length is.

To enable the SFB to process the job it has to be called with R = FALSE.

Either NDR will be set to TRUE, if the task was finished without any error, or ERROR will be set to TRUE, if the task was finished with any error.

In case of any error or warning the CPU shows in STATUS the corresponding event-no. NDR or ERROR/STATUS will be given at RESET of the SFB (R=TRUE) as well.

Parameter	Declaration	Data type	Offset	Description
EN_R	INPUT	BOOL	0.0	"Enable receive": release of receive
R	INPUT	BOOL	0.1	"Reset": sending is disabled (Not necessary because of compatibility reasons)
LADDR	INPUT	WORD	2	UART-index only 1 = RS485
NDR	OUTPUT	BOOL	4.0	TRUE: task is executed faultless, Data assumed FALSE = still receiving
ERROR	OUTPUT	BOOL	4.1	„Error“ failure state in STATUS The parameter is only set for one call.
STATUS	OUTPUT	WORD	6	0303hex = ANY- pointer to data area not valid 051Dhex = reset by input R 0000hex = ok 0001hex = blocked by an other job 0002hex = nothing received 0003hex = received < 2 bytes 0004hex = CRC Error, data ignored 0005hex = Parity error 0006hex = receiving in progress
RD_1	INOUT	ANY	8	ANY- pointer in data area of receive data - UID (1 byte target address) - PDU: Modbus-command (1 Byte) ... Modbus-address, number ... (specific for each commando) -CRC (reserving 2 byte space in data area)
LEN	INOUT	INT	18	Length of user data (without checksum) in bytes

It is not allowed to use SFC60/61 similar in multiple processing layers (OB1, OB35).

Modbus RTU (Server)

To use the protocol Modbus RTU as server (slave) it is necessary to config it all in the ConfigStage:

- RS485 baud rate max. 19200
- Data format 8E1 (=Modbus-recommendation)
- Protocol RS485 to "RTU"
- Modbus-RTU-Server enable
- Define mapping on (existing!) S7-operators

following commands will be supported

- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs
- 03 (0x03) Read Holding Registers
- 04 (0x04) Read Input Registers
- 05 (0x05) Write Single Coil
- 06 (0x06) Write Single Register
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple Registers

In S7 nothing must be programmed. All the data will be processed from the operating system in background synchronous to the control point.



VIDEO-tutorial available

There are available samples how to assign a Modbus RTU interface by ConfigStage and a link to the corresponding YouTube®-video at the INSEVIS-download website.

Communication - UART

Receive data via RS232 with defined End Code

To use the protocol "Receive with defined End Code" it is necessary to config it all in the Config Stage

Mode "On Expiry of Character Delay Time"

SFB61 must be called with the maximum length of a expected receive message (in parameter LEN).

NDR will be returned true, if the maximum length is reached or no data are received for longer than defined Delay Time since last receive by SFB61.

Due to the protocol is completely handled in runtime of SFB61 it is necessary that 2 consecutive messages are separated by a time gap longer then twice SFB61-cycle time. Elsewise these messages are merged.

Mode "On Receipt of End Character"

SFB61 must be called with the maximum length of a expected receive message (in parameter LEN).

NDR will be returned true, if the maximum length is reached or the specified end character(s) are detected. LEN and received message are returned inclusive end character(s).

Due to the protocol is completely handled in runtime of SFB61 it is necessary that no characters are sent after the end key character(s). Elsewise characters received between end key character(s) and SFB61-call get lost.

Change the UART-configuration by SFB 130 "CFG_PTP"

INSEVIS-PLCs offer a SFB 130 "CFG_PTP", to query ore to write the parameters of the UART-interface of the PLC. Therewith the valid configuration,will be overwritten to the next new start (warm start).

Parameter	Declaration	Data type	Offset	Description
REQ	INPUT	BOOL	0.0	TRUE = Overwriting the UART-configuration, FALSE = Query the UART-configuration,
UART	INPUT	WORD	2	0 = RS232, 1 = RS485
DONE	OUTPUT	BOOL	4.0	TRUE = configured successfully FALSE = configuration failed
ERROR	OUTPUT	BOOL	4.1	TRUE = configuration failed
STATUS	OUTPUT	WORD	6	Statuscode: 10hex = Index UART not valid 20hex = Baud rate BAUD not valid 30hex = Data format DATA_FMRT not valid 40hex = Modbus-Server Unit-Identifier UID not valid
BAUD	INOUT	DINT	8	Baud rate in bit/s, e.g. "9600" Value must be > 0
DATA_FMRT	INOUT	BYTE	12	CPU-V/P: 0 = 8 data bit, no parity 1 = 8 data bit, even parity 2 = 8 data bit, odd parity CPU-T (above OS 2.5.1): bit .3 - .0 0 = no parity 1 = even parity 2 = odd parity 3 = mark parity (1) 4 = space parity (0) bit .5 - .4 0 = 1 stop bit 1 = 1,5 stop bits (5 data bits only) 2 = 2 stop bits (except 5 data bits) bit .7 - .6 0 = 8 data bits 1 = 7 data bits 2 = 6 data bits 3 = 5 data bits
UID	INOUT	BYTE	13	0 = configuration ASCII or Modbus-RTU client 1..247 = Modbus-RTU Server Unit-Identifier (RS485 only)

It is not allowed to use SFB130 at the same time in multiple process layers (OB1, OB35).

CAN

All INSEVIS-PLCs contain a CAN-Master.

The INSEVIS CAN-interface allows

- a **comfortable configuration -conform to CANopen-** with the cost free software tool „ConfigStage“
 - a) of decentral INSEVIS-periphery (completely without CAN-knowledge) and
 - b) of decentral external periphery (easily with automatic importable EDS-file).
- as well as a **free programming of the CAN-communication with STEP7.**

As global parameter a **baudrate** is to be determined.
For physical reasons the maximum baudrate of a CAN bus depends on the maximum line length. In case of inserted repeater or isolators an additional reduction of baudrate may be required.

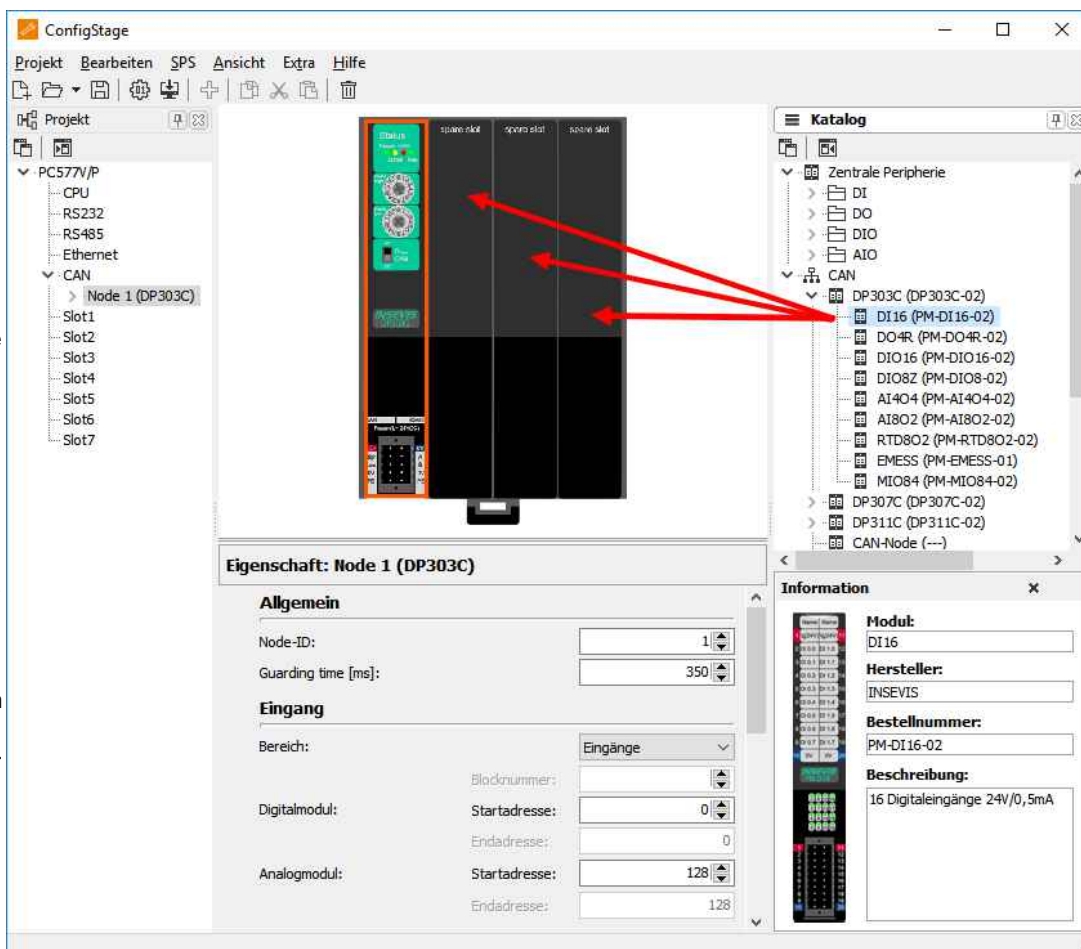
Baudrate (kbit)	maximum busline length (m)
1000	20 ... 50
500	100
250	250
125	500
50	1000

CANopen® Configuration with the ConfigStage

The settings are easy to do in the configuration software ConfigStage and
→ explained in the chapter „ConfigStage“ below the issue „CAN-settings“ in detail.

- a) **With INSEVIS-periphery as decentral periphery there is no CANopen® knowledge necessary.**
Only these little assignments are needed:

- Type in the node-ID (what are to be set up on hardware hexadecimal turn-switches on INSEVIS CAN-head station)
- Assign the guarding time you want. (Within this time the PLC requests the state of the CAN-node and forwards error messages by the SFB114 (CAN-state request) to the OB186.
- The start address of the decentral I/O addressing can not be assigned to the single decentral periphery modules, only once to the head station. Than these addresses can be used like INSEVIS-onboard periphery.



Communication - CAN

b) To link external slaves conform to CANopen® there is an easy configuration as well by:

RxPDO Configuration

PDO Number: 1 Description:

COB ID
☐ Constant
☐ "Node ID" + Offset (hex)
☒ According to CiA DS301
☐ PDO Disable-Flag ☐ RTR-not-allowed-flag

Communication parameter
☒ Download
 Inhibit time [1/10 ms]: 0
 Event time [ms]: 0
 Transmission type: 254

S7 Mapping
 Area: Data block
 Block number: 10
 Byte offset: 0

Mapping
☒ Download
 Data length (bytes): 8

	Data type	Index (hex)	Sub index (hex)	Description
1:	Word	6040	00	Controlword
2:	Word	6060	00	Modes of Operation
3:	DWord	607A	00	Target Position
4:				
5:				
6:				
7:				
8:				

CAN Objects

Description	Index	Sub index	Data type	Low...	Default ...	High limit
Controlword	6040	00	UINT16			
Modes of Operation	6060	00	INT16			
Target Position	607A	00	INT32			
Profile Velocity in pp-mode	6081	00	UINT32			
Profile Acceleration	6083	00	UINT32			
Profile Deceleration	6084	00	UINT32			
Interpolation Data Record....	60C1	01	INT32			
Interpolation Data Record....	60C1	02	INT32			
Interpolation Data Record....	60C1	03	INT32			
Interpolation Data Record....	60C1	04	INT32			
Target Velocity	60FF	00	INT32			
Digital_Output.DigitalOutput...	6300	01	UINT16			
Digital_Output.DigitalOutput...	6300	02	UINT16			
AnalogOutput0_DemandValue	2019	00	INT16			
AnalogOutput1_DemandValue	201A	00	INT16			
Controlword2	201B	00	UINT16			
Statusword2	201C	00	UINT16			
DeviceControl	2028	00	UINT16			
POSITION_position_Y4	202A	00	INT32			
POSITION_speed_Y4	202B	00	INT32			
POSITION_speed_Y2	202C	00	INT16			
POSITION_accel_U16	202D	00	UINT16			
SPEED_speed_Y4	202E	00	INT32			
SPEED_speed_Y2	202F	00	INT16			
PG2POSITION_position	2030	00	INT32			
PG2POSITION_speed	2031	00	INT32			
PG2POSITION_accel	2032	00	UINT32			
PG2POSITION_position_Y4	2036	00	INT32			
PG2POSITION_speed_Y4	2037	00	INT32			
PG2POSITION_speed_Y2	2038	00	INT16			
PG2POSITION_accel_U16	2039	00	UINT16			
PG2SPEED_speed	203A	00	INT32			
PG2SPEED_speed_Y4	203C	00	INT32			
PG2SPEED_speed_Y2	203D	00	INT16			

- Import of the vendor specific EDS-file
- and
- configuration of the CAN-objects by the ConfigStage software.

Communication - CAN

Send CAN telegrams with SFB105

Any CAN telegrams can be sent with SFB 105. Although the CAN telegrams will be entered into the send buffer and the further cycle does not need to be considered.

The return value signals the buffer overflow, what normally only occurs at physical damages or heavy program errors.

Parameter	Declaration	Type of data	Offset	Description
COB-ID	INPUT	DWORD	0	CAN telegram identifier (11 or 29 bit, right justified)
RTR	INPUT	BOOL	4.0	„Remote Transmit Request“ Requests the sending of the CAN telegram of the proper identifier. Useful without data only, it means DLC = 0.
IDE	INPUT	BOOL	4.1	Identifier format 0: 11 bit 1: 29 bit
DLC	INPUT	BYTE	5	„Data Length Code“ Number of bytes to send in data field (0...8)
DATA	INPUT	ARRAY[1..8] of BYTE	6	User data
Status	OUTPUT	WORD	14	Return value with state bits: Bit 0: Buffer overflow e.g. - due to „flood“ of transmit buffer by user program - caused by wiring error no telegrams can be sent Bit 1: Parameter error COB-ID Bit 2: Parameter error DLC

Receive CAN telegrams with SFB106

All received CAN telegrams, what could not be dedicated to any service, will be kept in a receive buffer and can be read with SFB106.

The return value signals the buffer content and buffer overflow.

Parameter	Declaration	Type of data	Offset	Description
COB-ID	OUTPUT	DWORD	0	CAN telegram identifier (11 or 29 bit, right justified)
RTR	OUTPUT	BOOL	4.0	„Remote Transmit Request“ Requests the sending of the CAN telegram of the proper identifier. Useful without data only, it means DLC = 0.
IDE	OUTPUT	BOOL	4.1	Identifier format 0: 11 bit 1: 29 bit
DLC	OUTPUT	BYTE	5	„Data Length Code“ Number of bytes to send in data field (0...8)
DATA	OUTPUT	ARRAY[1..8] of BYTE	6	User data
Status	OUTPUT	WORD	14	Return value .0 Buffer empty: no data received .1 Buffer overflow: received too much telegrams since last call

Receive event OB106

Every entry in the user receive buffer will be signaled by a OB106 call.

Thereby can be realized an event controlled communication. Also buffer overflows at cycle time exceedings and high data volumes can be prevented.

Communication - CAN

SDO transfer SFB107

With the SFC107 it is possible to have a comfortable access to variables of CAN-nodes compatible to CANopen®. The FB107-113 in the sample are useful as interface to ease parameter handling.

On the beginning of a SDO transfer it is necessary to set the request bit. It will be ignored while running communication.

Because of the SDO communication will be receipted always, the interface needs to be called multiple times, until the return bits „Done“ ore „error“ are set. On segmented transfer will be communicated up to 7 bytes an every call.

Parameter	Declaration	Type of data	Offset	Description
REQ	INPUT	BOOL	0.0	Request
ULD	INPUT	BOOL	0.1	Upload 0: SDO download, write CANopen variable 1: SDO upload, read CANopen variable
SEG	INPUT	BOOL	0.2	Segmented SDO transfer for objects > 4 byte 0: expedited SDO (1..4 byte) 1: segmented SDO transfer
NODE	INPUT	BYTE	1	Node ID
OBJ	INPUT	WORD	2	CANopen-object index
SUB	INPUT	BYTE	4	CANopen-object subindex
CNT	INPUT	BYTE	5	Number of bytes of variable [1,2,4]
TIMEOUT	INPUT	WORD	6	Timeout in ms [1..32767], 0 = endless
DONE	OUTPUT	BOOL	8.0	Finished without failures
ERROR	OUTPUT	BOOL	8.1	Finished with failures
STATE	OUTPUT	BYTE	9	Return value 0: OK 1: Parameter error 2: no SDO channel available 3: overflow of send buffer 4: received SDO error response 5: Error object index / subindex 6: Unexpected answer 7: Faulty length at seg. SDO 8: Faulty length at exp. SDO 9: Timeout
DATA	INOUT	ARRAY[1..6] of BYTE Area pointer	10	Expedited SDO: User data 1..4 byte Segmented SDO: array pointer for user data
LEN	INOUT	WORD	16	Number of communicated bytes (relevant only for segmented SDO)
INTERN	STATIC	ARRAY[1..7] of BYTE	18..24	internal variables

Next follow the samples of user FBs,:

FB107 SDO upload variable byte Request Node: node-ID Object Subindex Data byte Return values: Done, Error, Error code	FB108 SDO download variable byte Request Node: node-ID Object Subindex Data byte Return values: Done, Error, Error code	FB109 SDO upload variable word Request Node: node-ID Object Subindex Data word Return values: Done, Error, Error code
---	---	---

Communication - CAN

FB110 SDO download variable word Request Node: node-ID Object Subindex Data word Return values: Done, Error, Error code	FB111 SDO upload variable double word Request Node: node-ID Object Subindex Data double word Return values: Done, Error, Error code	FB112 SDO download variable double word Request Node: node-ID Object Subindex Data double word Return values: Done, Error, Error code
FB113 SDO transfer string Request Upload-Control Flag (0=download, 1=upload) Node: node-ID Object Subindex S7-Area pointer in data Length of data field / of data to be sent / of received data in byte Return values: Done, Error, Error code		

Alternative function of SFB107

The configuration software „ConfigStage“ assigns for each CAN-node a list of SDO-downloads to configure the CAN-node, what will be processed while PLC is booting.

SDO-object index 0 is ref. to CANopen not allowed as a variable. If SFB107 is called with SDO-object index 0, the complete internal structure of the respective CAN-node will be written.

Parameter	Declaration	Data type	Offset	Description
REQ	INPUT	BOOL	0.0	Request
ULD	INPUT	BOOL	0.1	Upload 0: SDO download, writing CANopen-variable
SEG	INPUT	BOOL	0.2	Segmented SDO transfer for objects > 4 Byte 0: expedited SDO (1..4 Byte)
NODE	INPUT	BYTE	1	Node ID
OBJ	INPUT	WORD	2	CANopen-object-index (must be 0 here)
SUB	INPUT	BYTE	4	CANopen-object-sub-index (must be 0 here)
CNT	INPUT	BYTE	5	Not used
TIMEOUT	INPUT	WORD	6	Timeout in ms [1..32767], 0 = infinity Timeout valid for each single transfer
DONE	OUTPUT	BOOL	8.0	Complete liste processed
ERROR	OUTPUT	BOOL	8.1	Single-transfer finished with error; If the SFB107 is still called, the processing of the list goes on
STATE	OUTPUT	BYTE	9	Return value 0: OK 1: Parameter error 2: no SDO chanal available 3: Send buffer overflow 4: SDO-error received response 5: Error object-index/ -subindex 6: Unexpected answer 9: Timeout
DATA	INOUT	ARRAY[1..6] of BYTE	10	Internal used, Watching of actual user data possible, Return of SDO-error code when abort
LEN	INOUT	WORD	16	Index of the actual processed list entry (via all nodes) Done: Total number of processed entries (all nodes)
INTERN	STATIC	ARRAY[1..7] of BYTE	18..24	Internal variables

Communication - CAN

CAN status request SFB114

SFB114 delivers the instance data block a status word and a list with actual status information from nodes compatible to CANopen.

Parameter	Declaration	Type of data	Offset	Description
TXOFL	OUTPUT	BOOL	0.0	CAN transmit buffer overflow
RXOFL	OUTPUT	BOOL	0.1	CAN receive buffer overflow
L2BUFOFL	OUTPUT	BOOL	0.2	CAN unser receive buffer overflow
NMT_ERROR	OUTPUT	BOOL	0.3	CANopen® NMT errorr (Node-Guarding, Heartbeat)
OP_ERROR	OUTPUT	BOOL	0.4	CANopen® node not OPERATIONAL
HW_ERROR	OUTPUT	BOOL	0.5	CAN-hardware error
SDO_ERROR	OUTPUT	BOOL	0.6	CANopen® initializing of node by SDO faulty
NODESTATE	OUTPUT	ARRAY [1..128] of BYTE	2	Status value of CANopen node list 00 hex BOOTUP = at last received bootup telegram from node 01 hex UNKNOWN = configured but no response from node 02 hex MISSED = no more answer from node 03 hex UNAVAILABLE = no Guarding because of communication errors 04 hex STOPPED 05 hex OPERATIONAL 7F hex PREOPERATIONAL 80 hex node not configured - unused

CAN error event OB186

The OB106 signalizes a device failure of configured CAN-devices.

Parameter	Declaration	Type of data	Offset	Description
EVENT	OUTPUT	BYTE	0	39 hex error comes 38 hex error goes = node is working again
	OUTPUT	BYTE	1	C4 hex
	OUTPUT	BYTE	2	
	OUTPUT	BYTE	3	
	OUTPUT	DWORD	4	
	OUTPUT	BYTE	8	Node ID

Automatic return

At CANopen the nodes will be configured in a preperation (PREOPERATIONAL) and transfer user data autonomous at run-time (OPERATIONAL). Although there is a controlling, if all nodes still do communicate (Node-Guarding or Heartbeat).

If the communication to a CAN-node temporary fails during the operation, the node is during return in PREOPERATIONAL mode. The error handling and a new integration of the node is due to the application program (to the S7-programmer).

Failed nodes will be recognized by SFB114.

The node state values of SFB114 are results of Node-Guarding or Heartbeat.

Communication - CAN

If a node is in PREOPERATIONAL mode, it is ready to take part in communication again. But it is probable, that the node has lost its configuration (e.g. by to less voltage).

For the new configuration the SFB107 (with OBJ = 0) can be used, the command „goto OPEATIONAL“ will be sent by SFB105. Wait until a new nodestate is available afterwards.

```
// call SFB114 (via local instance)
CALL #CAN_State
TXOFL :=
RXOFL :=
L2BUFOFL :=
NMT_ERROR:=
OP_ERROR :=
HW_ERROR :=
SDO_ERROR:=
NODESTATE:=

...
// test all critical nodes for state == PREOPERATIONAL or BOOTUP
U(
  L #CAN_State.NODESTATE[2] // Node 2
  L B#16#7F // 7F = PREOPERATIONAL
<>I
)
U(
  L #CAN_State.NODESTATE[2] // Node 2
  L B#16#00 // 00 = BOOTUP
<>I
)
SPB end // exit or test next node

// call SFB107 with OBJ = 0
CALL #SFB107i
REQ :=TRUE
ULD :=FALSE
SEG :=FALSE
Node :=B#16#2// current Node
OBJ :=W#16#0
SUB :=B#16#0
CNT :=
TIMEOUT:=W#16#64 // 100ms
DONE :=
ERROR :=
STATE :=
DATA :=
LEN :=
...
U #SFB107i.ERROR
... // Error handling

U #SFB107i.DONE
... // finished: goto „send NMT command“
SPA end

// send NMT command „goto OPERATIONAL“
L B#16#1 // CMD goto operational
T #Data[1]
L B#16#2 // Node 2
T #Data[2]

CALL #CAN_Send
COBID :=DW#16#0
RTR :=FALSE
IDE :=FALSE
DLC :=B#16#2 // 2 Bytes
DATA :=#Data
STATUS:=

// wait > Guarding/Heartbeat-Time before use return value of SFB114 again
```

Configuration with the Software „ConfigStage“

It is very easy to configure and parametrize all INSEVIS- PLCs and internal or external CAN-Periphery with the free software tool „ConfigStage“. First you select your device from the list of offered devices. After having selected your device there opens up a window with some areas.

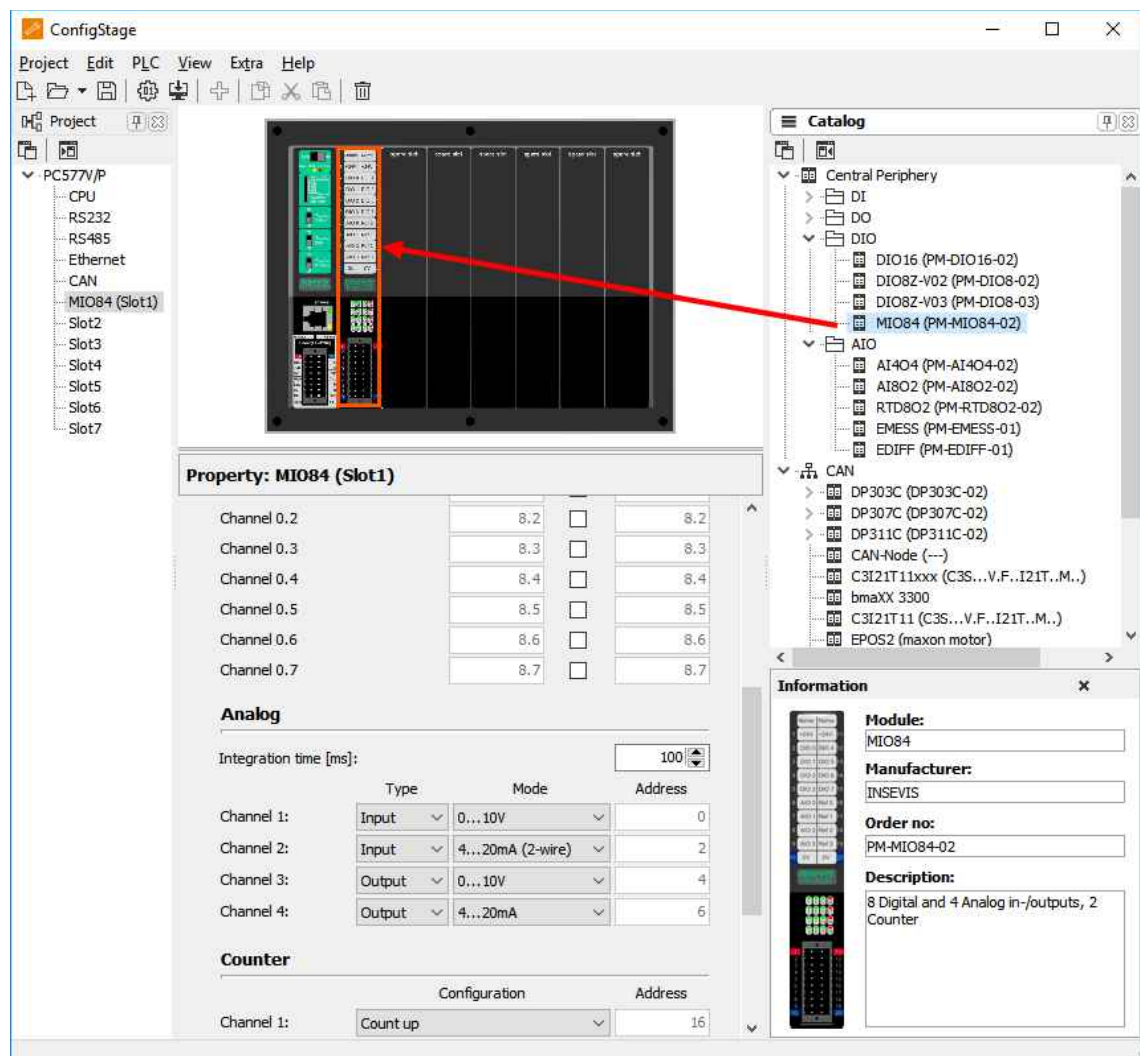


VIDEO-tutorials available

There are available different video tutorial in the ConfigStage play list at INSEVIS Simatic®-channel INSEVIS EN. Sample programs are ready to download from INSEVIS web site.

Basic settings

- Middle PLC you selected (from rear side) without any periphery
- Left self constructing project tree with the configurable functions of the CPU, free slots and the periphery already configured
- Right catalog area with all periphery modules and decentral CAN-peripherals available
 - periphery modules will be moved by drag`n drop to the preferred slot
 - CAN-head stations will be also moved by drag`n drop to the green area in the CPUs image or to the CAN-pins in the connectors image
- Bottom Configuration area, where the properties will be configured
- Bottom right Information area of the selected module



Hint: The Profibus- and CPU-configuration (without INSEVIS-specific settings for Ethernet, serial and CAN) can be made in the Simatic® Manager. The download of the Simatic® Manager overwrites all SDBs completely.

Please do config Profibus and CPU **FIRST** and download it into the PLC. Than configure all the other configurations (CPU with Ethernet, RS232/485, Modbus, CAN and (de-)central periphery and) with the ConfigStage and download it **LATER**. This download does overwrite only the configured SDBs and keeps the Profibus- and CPU-settings of the Simatic®- Manager.

Configuration with the Software „ConfigStage“

Assign the IP-address

ConfigStage can change the IP-address of the target device.

Either

- Click on „Ethernet“ in the project tree
- Enter the IP-address, netmask and router-address (if checked)

IP Protocol

IP Address: 192.168.80.50

Netmask: 255.255.255.0

☒ Router address: 192.168.80.1

or:

- enter the download-dialog box (F12).
(There will be used the service “Accessible devices” like in other INSEVIS-Stages)

In this sample a IP-address 192.168.80.50 was assigned in the configuration, see below.

This new configuration must be downloaded to the old IP-address 192.168.80.67 once before it is active. This target-IP-address is to be

- entered manually
- or
- selected from a list of accessible devices (press „Refresh“)

For the download the PLC will be switched to STOP mode and restarts later.

ConfigStage

Target device IP address
Enter target device IP address or
select the device from accessible device list.

Target device IP address: 192.168.80.50

Accessible devices

Interface: Intel(R) Ethernet Connection I217-LM

Name	MAC address	IP Address	Netmask	Router address
INSEVIS PC57xV	00-50-C2-DF-3E-3D	192.168.80.67	255.255.255.0	192.168.80.67

Refresh

☒ Ok
☐ Cancel

Change target device

If an existing configuration should be transferred to another INSEVIS-S7-PLC, it can be done at the „Project“-menu. It helps saving time and prevents errors.

Addressing of the onboard periphery

Standard addressing ind the INSEVIS- PLCs

Without using software tool „ConfigStage“ the following address area will be set up in a standard way:

digital module: 4 byte inputs, 4 byte outputs
analog module: 16 byte inputs, 16 byte outputs

start address \ slot	slot 1	slot 2	slot 3	slot n
digital inputs	byte 0	byte 4	byte 8	byte (n-1)x4
digitale outputs	byte 0	byte 4	byte 8	byte (n-1)x4
analog inputs	byte 128	byte 144	byte 160	byte (n-1)x16 +128
analog outputs	byte 128	byte 144	byte 160	byte (n-1)x16 +128



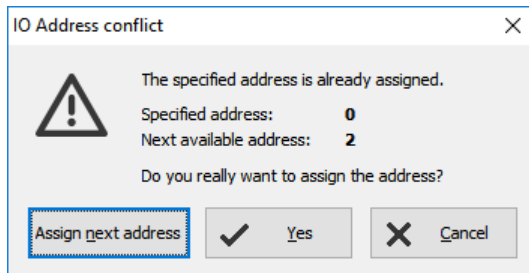
If a functional module FM DIO8-Z is placed, all following digital address areas will be moved for 8 byte more.

Configuration with the Software „ConfigStage“

Different addressing in PLCs and addressing of decentral periphery

If decentral periphery is used and/or other should be used other address areas in the PLCs, it will be configured with the software tool „ConfigStage“ and the set up date will be stored in the system data blocks (SDBs).

The used periphery address area can be displayed in an own window.
The pull down menu „View“ → „Address overview“ creates this window with all periphery addresses used.



When addresses will be assigned there is an plausibility check already and only if you skip the warning an address conflict is possible. This conflict is shown in the address overview in red.

Address overview

Input and Output address overview

Filter: ☒ Input address ☒ Output address ☐ Only conflicted address

Type	Address	Module	Node	Slot
Input	0 .. 1	DI16		1
Input	1 .. 2	DI16		2
Input	8 .. 9	DIO16		3
Output	8 .. 9	DIO16		3
Input	12 .. 23	DIO8Z-V03		4
Output	12 .. 23	DIO8Z-V03		4
Output	24 .. 24	DO4R		7
Input	24 .. 25	DP303C.DIO16	1	1
Output	25 .. 26	DP303C.DIO16	1	1

CPU settings

The CPU settings are compatible to the Siemens-CPU S7-315-2PN DP in the Simatic®-Manager. Either via project tree or by clicking directly on the image all different CPU properties can be set up easily in the ConfigStage.

General

Insert plant- and location designation

Startup

Box checked: CPU shall start, if the present configuration mismatches to actual configuration
Timeout: multiply this value with 100 milliseconds, the maximal value is limited to 1000 seconds

Diagnostic

Box checked: Reports the cause for STOP-condition to the connected devices (PG, host system,...) to be entered in diagnostic buffer too.

Know-how protection

The write or read/write protection.
Password protection from Simatic®- Manager is not affected of it and to activate there.

Communication

INSEVIS-CPU's contain as well as the Siemens CPU 315-2PN/DP system internal **16 passive S7-connections**. 1

Additionally there can be parameterized **up to 16 active S7-connections** by the ConfigStage and numbered by an own ID-number.

More at **Information about TSAPs**

General

Plant designation
Extruder line 1

Location designation
Area 1

Startup

☒ Startup if present configuration does not match actual configuration
Timeout for transferring of parameter to modules [100ms]: 100

Diagnostic

☒ Report cause of STOP
Number of messages in diagnostic buffer: 10

Protection

☒ No protection
☐ Write-Protection
☐ Write-/Read Protection

Password:
Reenter password:

Communication

Reserved connection resources for
PG Communication: 1
OP Communication: 1
Maximum number of connection resources: 16

Configuration with the Software „ConfigStage“

Cycle

Cycle monitoring time:

(Insert it in ms, max. 6000ms = 6 seconds):

Cycle load from communication

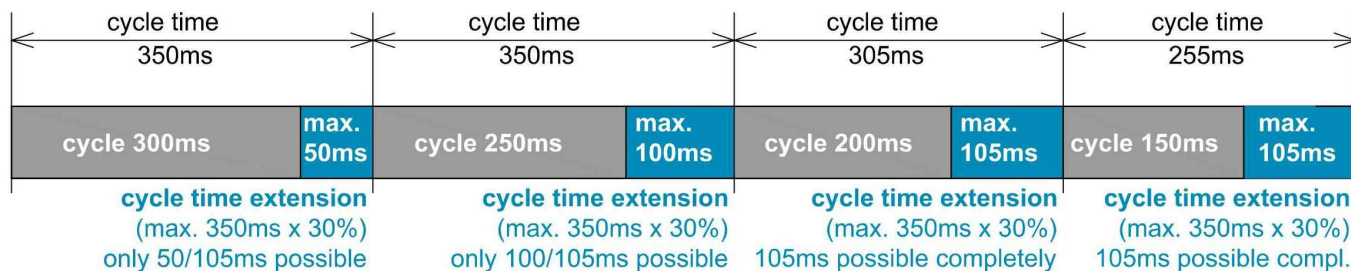
(cycle time extension, min. 10% , max. 50%)

for communication (Ethernet, field bus, PG), visualization, file system of Micro-SD®-card)

Cycle	
Scan cycle monitoring time [ms]:	320
Scan cycle load from communication [%]:	20
Process input image area size:	128
Process output image area size:	128

The declaration of the cycle time in the PG is carried out from control point to control point. This contains communication and visualization. Independent from the S7-program an unsteady cycle time can occur.

For that case it is possible to declare a limit value in relation to the cycle monitoring time, what is used for filling out the „free time“ with time for communication (or visualization, etc).



In this sample the cycle time can be extended max. by 30% of 350ms (= 105ms). If this time is not needed for communication (visualization, etc), it is available for other tasks.

Clock

check memory byte and insert its number from value 0

Retentive memory

Merkerbytes:

total 2.048 (thereof 0..15 preset)

Timer and counter:

total 256 (no preset)

Time-Of-Day interrupt

Box checked: Time-Of-Day interrupt activated

Insert interval areas, starting date and starting time

Cyclic interrupt

CPUs -V/P support the OB35 only Value in milliseconds (ms), maximal value is 1 minute (60,000 ms)

CPU -T supports the OB32...OB34 too Value in milliseconds (ms), maximal value is 1 minute (60,000 ms)

Clock	
<input type="checkbox"/> Clock memory byte	0
Retentive memory	
Number of memory bytes starting MB0:	16
Number of S7 timers starting with T0:	0
Number of S7 counters starting with C0:	8
Time-Of-Day interrupt	
<input type="checkbox"/> OB10 (Priority 2)	Execution: None
	Start date: 01.01.1994
	Start time: 00:00:00
Cyclic interrupt	
OB35 (Priority 12)	Execution [ms]: 100

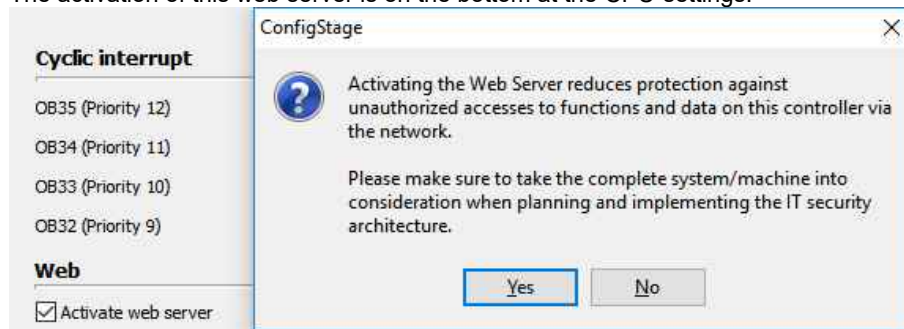
Configuration with the Software „ConfigStage“

WebServer

The PLCs with CPU-T inside contain a static web server, what provides

- general data (serial no., etc.),
- actual entries of diagnostic buffer,
- addresses of I/Os and,
- an fiel to require the actual value of a variable.

The activation of this web server is on the bottom at the CPU-settings.



By connecting to the internet it is possible that unauthorized get access to the PLC and they could manipulate it and cause harms.
The programmer and final user are responsible for installung and running suitable security actions to provide this unauthorized access.
INSEVIS is not responsible and does not take over responsibility for damages caused by unauthorized access.

Configuration with the Software „ConfigStage“

Communication settings

RS232 and RS485

The assign of RS232 and RS485 is self-explanatory. If you select at the RS485 the protocol „Modbus-RTU“, you will be asked to assign the node-ID as well as to map your S7-operands to input-bits and -words and to output-bits and -words.

If „Modbus Server is deactivated, Modbus RTU-telegrams will be received and send by SFB60/61

Ethernet

If you want to use the Ethernet-interface, there are to assign up to 16 connections, to select the communication protocol and to parameterize it. Each Ethernet-connection gets a connection-ID to be assigned in the S7-program

CPU-VI-P: There is ONE Ethernet-interface available (RJ45) (right)

CPU-T: There are TWO Ethernet-interfaces available (RJ45), what can be used either
 - as switch with common IP-address (left image) or
 - as separated Ports with differen IP- (and MAC-) addresses and net masks. (right image).

Property: Ethernet

General

Operation mode
☒ 2-port ethernet Switch
☐ 2 separate ethernet ports

Ethernet 1
☒ Activate this port for use
 Transmission medium / duplex: Automatic
☐ Disable auto negotiation

Ethernet 2
☒ Activate this port for use
 Transmission medium / duplex: Automatic
☐ Disable auto negotiation

IP Protocol

IP Address: 192.168.80.50
 Netmask: 255.255.255.0
 Router:

Protection

Permit access with S7 Communication from remote partner (PG, PLC, HMI, OPC, ...) via
☒ Ethernet 1
☒ Ethernet 2

Connections

ID	Type	Active	Local	Partner	Partner IP
1	S7 connection	Yes	10.02	10.02	0.0.0.0

Both ports can be used

Property: Ethernet

IP Protocol

IP Address: 192.168.80.50
 Netmask: 255.255.255.0
☐ Router address: 192.168.80.50

Connections

ID	Type	Active	Local	Partner	Partner IP
1	S7 connection	Yes	10.02	10.02	0.0.0.0

Property: Ethernet

General

Operation mode
☐ 2-port ethernet Switch
☒ 2 separate ethernet ports

Ethernet 1
☒ Activate this port for use
 Transmission medium / duplex: Automatic
☐ Disable auto negotiation

Ethernet 2
☒ Activate this port for use
 Transmission medium / duplex: Automatic
☐ Disable auto negotiation

IP Protocol

Ethernet 1 Ethernet 2
 IP Address: 192.168.80.50 192.168.70.50
 Netmask: 255.255.255.0 255.255.255.0
 Router:

Protection

Permit access with S7 Communication from remote partner (PG, PLC, HMI, OPC, ...) via
☒ Ethernet 1
☒ Ethernet 2

Connections

ID	Type	Active	Local	Partner	Partner IP
1	S7 connection	Yes	10.02	10.02	0.0.0.0

Port 2 can not more communicate by S7 (protection)

Configuration with the Software „ConfigStage“



Hint:

To prevent unauthorized access by the „classic“ S7-communication, the ConfigStage allows to switch off this communication at one port. This prevents an unauthorized manipulation of PLC-data.

Every connection gets a connection-ID to assign it to the S7-program. Referring to the connection type the selected connection will be parameterized in separate boxes with these parameters:

parameters at S7-connection (Active)

- Local TSAP,
- Partner-TSAP,
- Partner-IP-address

parameters at INSEVIS-Panel-HMI

- Local TSAP
(only necessary, if PLC shall change language/pages in HMI)

parameters at TCP Send/Receive

- Local port (fix or variable),
- Partner-port,
- Partner-IP-address

parameters at UDP Send/Receive

- Local port

parameters at Modbus-TCP (Server)

Assign of S7-operand areas for Modbus-

- Input bits
- Input words
- Output bits
- Output words

(As sample the „mapping“ at Modbus-TCP-Server)

Modbus-TCP-Client will be programmed by S7, a sample is available at the INSEVIS web sites)

Property: Ethernet

IP Protocol

IP Address:

Netmask:

☐ Router address:

Connections

ID	Type	Active	Local	Partner	Partner IP
1	S7 connection	Yes	10.02	10.02	192.168.80.10
2	INSEVIS Panel-HMI		FF.02		
3	TCP Send/Receive	No	2000	0	0.0.0.0
4	UDP Send/Receive		2000		
5	ModBus TCP Server		502		

Properties - ModBus TCP Server connection

Discrete Inputs (Bits)

Area:

Block number:

Byte offset:

Length in bytes:

Coils (Output bits)

Area:

Block number:

Byte offset:

Length in bytes:

Input Registers

Area:

Block number:

Byte offset:

Length in bytes:

Holding (Output) Registers

Area:

Block number:

Byte offset:

Length in bytes:

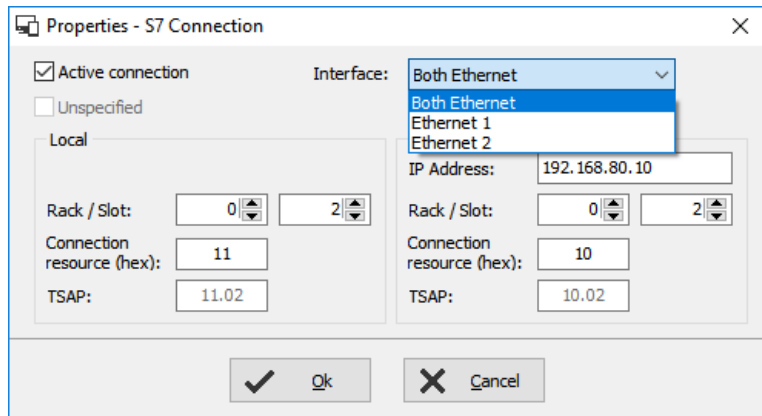
Configuration with the Software „ConfigStage“

Communication / Information about TSAPs

INSEVIS-CPU's contain as well as the Siemens CPU 315-2PN/DP system internal **16 passive S7-connections**. Their local TSAPs are defined by Siemens-definition xx.yy as follows:

- for PG= 01.02,
- for OP= 02.02,
- for WinCC etc.= 03.01
- xx=01 (for PG),
- xx=02 (for OP) and
- xx=03 (for WINCC etc.)
- yy for all 02 (= Slot2 for CPU)

CPU-T: Assigning properties to one or both of the single Ethernet interfaces (see image right)



Additionally there can be parameterized **up to 16 active or passive S7-connections** by the ConfigStage and numbered by an own ID-number. Here the Siemens-definition is not valid but it is **necessary**

1. to keep the TSAPs unique and
2. to not even use the TSAPs from the system internal passive connections.

Profinet

Profinet settings will be assigned by Simatic®-Manager or TIA®-Portal (like as Profibus settings)

CAN configuration

Decentral INSEVIS periphery

There is **no need to have CAN-knowledge** to include decentral INSEVIS- periphery to the INSEVIS-S7-CPU's.

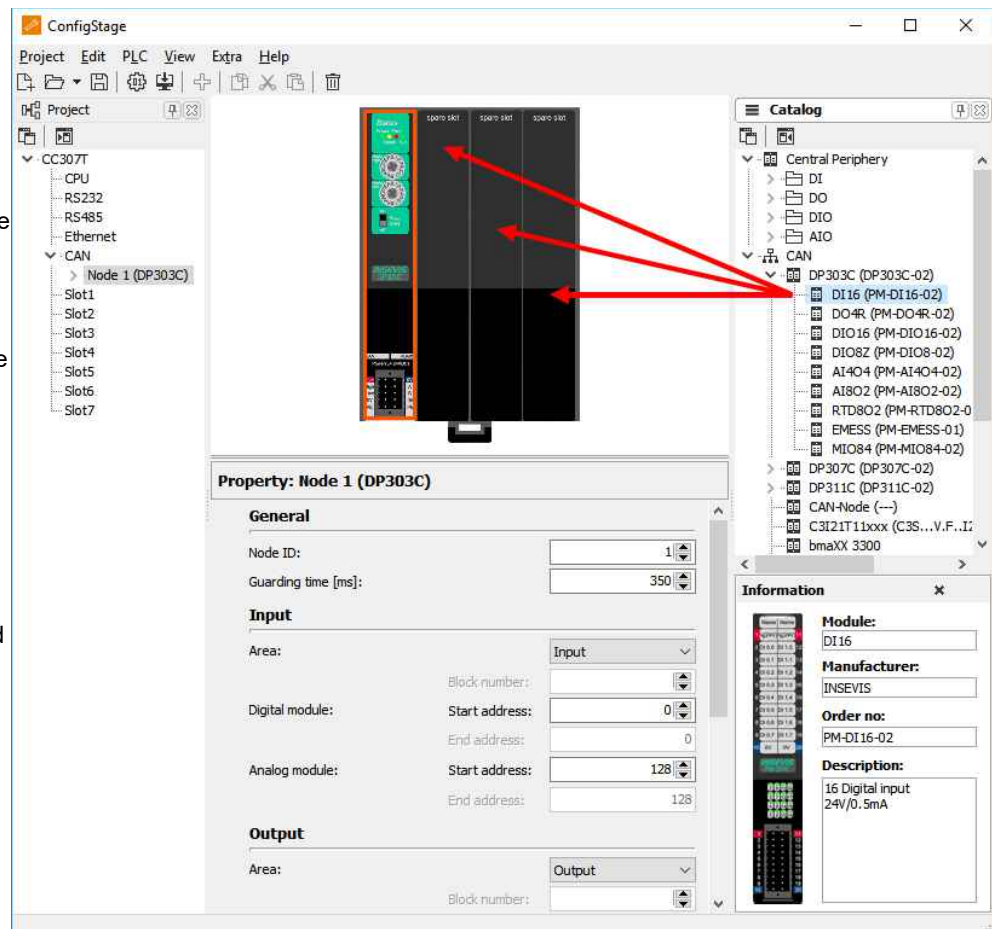
After having placed your INSEVIS head station on the CPU, this module appears in the project tree and in the display.

Now you can insert general start addresses for the head station for in-/outputs (no more possible at the single modules).

The periphery modules will be added per drag'n drop from a special sub area of the catalog tree below the CAN-title.

You type in the node-ID-number you have set up before at the INSEVIS- head stations hardware with the hexadecimal turn-switches.

Also you insert the guarding time you want. Than all i/os will be used in Step®7 like onboard I/Os of an INSEVIS-PLC.



Configuration with the Software „ConfigStage“

Configure decentral external periphery manually

Instead of an INSEVIS- head station you drag'n drop a neutral CAN-node onto your CPU. Then all common settings of for the **node-ID** and the **guarding parameters** will be done.

If the field **"NMT Control"** is activated, the node will be started and stopped together with the PLC. Therefore the NMT-messages "goto OPERATIONAL" with change to RUN (after OB100, before first OB1) and "goto PREOPERATIONAL" with change to STOP will be sent to the node.

Is the node not ready while PLC starts, the "goto OPERATIONAL"-message will be ignored. The must be considered in the user program with a state request. Eventually the NMT-commands need to be programmed in the S7-program.

With the check box **„NMT-Download“** will be assigned, if the communication parameters „Guarding-Time“ / „Lifetime“ / Heartbeat-Time“ should be downloaded to the node during the start up or not. This is useful only, if these parameters are not implemented in the node itself or assigned by other tools

For the process data are available each **32 Tx- and Rx-PDOs**. If more as **4 PDOs** of it will be used, PDO-identifiers of other node-IDs will be used for it.

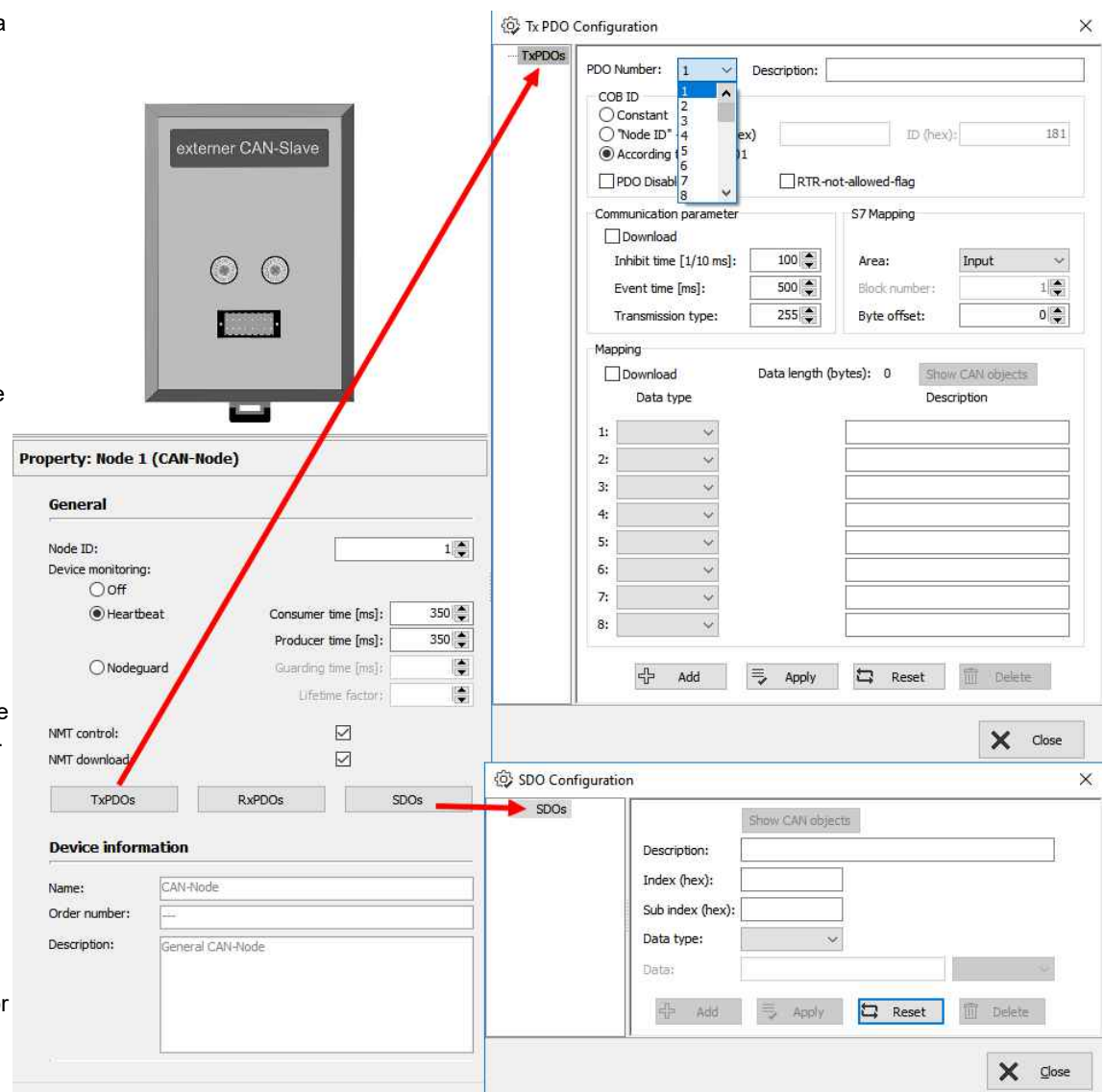
These node-IDs should not be assigned again. (Only CiA-conform TxPDO-identifier are valid.)

Received Tx-PDOs will be buffered and transferred synchronous to the control point to the process image,

RxPDOs will be send event- and time-controlled in the control point always.

An 8-byte-data field on a selectable S7-operand is to assign to every PDO.

The Tx-PDO communication parameters define the sending behavior of the node, the Rx-PDO define the sending behavior of the master.



Configuration with the Software „ConfigStage“

The inhibit-time prevents a too excessive sending, when data are changing permanently; the event-time forces the sending, even if there is no data change.

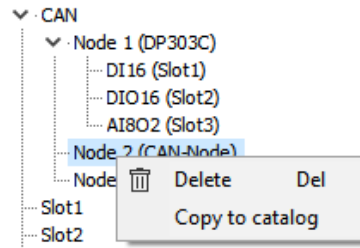
Entering of single CAN-objects (Index, Subindex) in the mapping field is optional, as long the download of the mapping is not activated. Always assign the data type to define the telegrams length and to allow a real byte swapping.
With activated download of complete mapping-parameter the PLC configures the content of the PDS during start up.

To assign more configurations in the start up, **SDOs** can be defined (e.g. for operational mode or metering ranges, etc.).

Create library elements of your CAN-slaves

Optional: If this „mapping“ is made, the new configured CAN-node (CAN-slave) can be stored locally in the ConfigStage to use this part as a library element with pre-defined values.

Therefore click with the right mouse key on the just configured CAN-node.



Then a new mask opens. Insert all information (and image) and acknowledge with „OK“.

So you will get an own CAN-slave, pre-defined for further use and with your internal information.

ConfigStage

Create a new CAN-Node
Set the CAN-Node descriptions

Library file name:

Device information

Name:

Manufacturer:

Order no:

Description:

Module image:
This image is shown in module viewer window when the module is selected.

Catalog image:
This image is shown in Catalog viewer window when the module is selected.

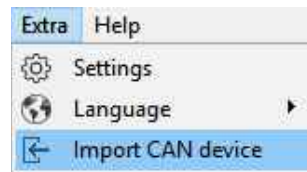


The nomenclature **TX** (Transmit) and **RX** (Receive) are at CANopen®-PDOs always from the CAN-nodes view (slave). It means, with a TX-PDO a node sends its input data to the PLC and with a RX-PDO it receives its output data.

Configuration with the Software „ConfigStage“

Configure decentral external peripherie by EDS-file

You need a EDS-file of the external CAN-slave you want to configure. Import it like shown here:



Then a new window opens like described above (Create library elements of your CAN-slave). After having assigned texts, order information and images there is an „empty“ library element, what must be configured like described in following items.

PDO configuration

By „Show CAN objects“ an object browser opens up with all CAN-objects available. These were filtered from the EDS-file automatically. Move single or multiple CAN-objects by Drag'n Drop into your configuration.

Tx PDO Configuration

PDO Number: 1 Description:

COB ID
☐ Constant
☐ "Node ID" + Offset (hex) ID (hex): 181
☒ According to CiA DS301

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag

Communication parameter
☐ Download
 Inhibit time [1/10 ms]: 100
 Event time [ms]: 500
 Transmission type: 255

S7 Mapping
 Area: Input
 Block number:
 Byte offset: 0

Mapping
☐ Download
 Data length (bytes): 8 **Show CAN objects**

Data type
 1: Byte
 2: Byte
 3: Byte
 4: Byte
 5: Byte
 6: Byte
 7: Byte
 8: Byte

Description
 ReadState8_InputLines.Digital Input By
 ReadState8_InputLines.Digital Input By
 ReadState8_InputLines.Digital Input By
 ReadState8_InputLines.Digital Input By
 ReadState8_InputLines.Digital Input By
 ReadState8_InputLines.Digital Input By
 ReadState8_InputLines.Digital Input By
 ReadState8_InputLines.Digital Input By

CAN Objects

Search CAN object ☐ Show values in Hexadecimal format

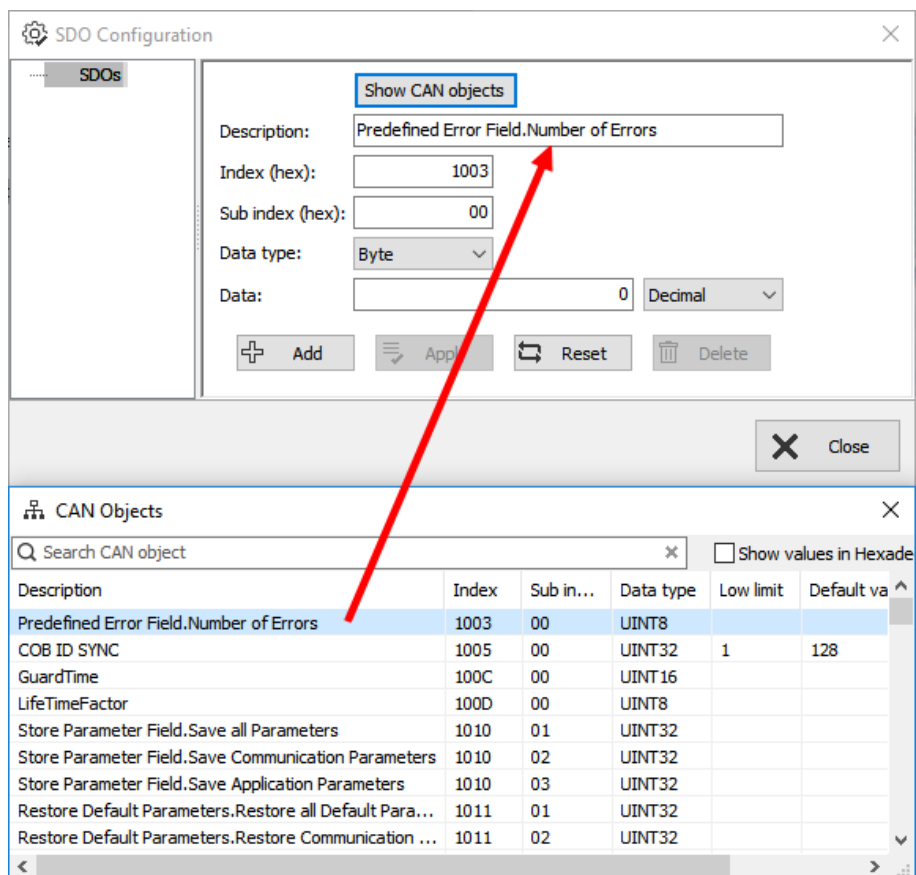
Description	Index	Sub index	Data type
ReadState8_InputLines.Digital Input Byte 1	6000	01	UINT8
ReadState8_InputLines.Digital Input Byte 2	6000	02	UINT8
ReadState8_InputLines.Digital Input Byte 3	6000	03	UINT8
ReadState8_InputLines.Digital Input Byte 4	6000	04	UINT8
ReadState8_InputLines.Digital Input Byte 5	6000	05	UINT8
ReadState8_InputLines.Digital Input Byte 6	6000	06	UINT8
ReadState8_InputLines.Digital Input Byte 7	6000	07	UINT8
ReadState8_InputLines.Digital Input Byte 8	6000	08	UINT8
ReadState8_InputLines.Digital Input Byte 9	6000	09	UINT8
ReadState8_InputLines.Digital Input Byte 10	6000	0A	UINT8
ReadState8_InputLines.Digital Input Byte 11	6000	0B	UINT8
ReadState8_InputLines.Digital Input Byte 12	6000	0C	UINT8
ReadState8_InputLines.Digital Input Byte 13	6000	0D	UINT8
ReadState8_InputLines.Digital Input Byte 14	6000	0E	UINT8
ReadState8_InputLines.Digital Input Byte 15	6000	0F	UINT8
ReadState8_InputLines.Digital Input Byte 16	6000	10	UINT8
ReadState8_InputLines.Digital Input Byte 17	6000	11	UINT8
ReadState8_InputLines.Digital Input Byte 18	6000	12	UINT8
ReadState8_InputLines.Digital Input Byte 19	6000	13	UINT8
ReadState8_InputLines.Digital Input Byte 20	6000	14	UINT8
ReadState8_InputLines.Digital Input Byte 21	6000	15	UINT8
ReadState8_InputLines.Digital Input Byte 22	6000	16	UINT8
ReadState8_InputLines.Digital Input Byte 23	6000	17	UINT8
ReadState8_InputLines.Digital Input Byte 24	6000	18	UINT8
ReadState8_InputLines.Digital Input Byte 25	6000	19	UINT8
ReadState8_InputLines.Digital Input Byte 26	6000	1A	UINT8
ReadState8_InputLines.Digital Input Byte 27	6000	1B	UINT8
ReadState8_InputLines.Digital Input Byte 28	6000	1C	UINT8
ReadState8_InputLines.Digital Input Byte 29	6000	1D	UINT8
ReadState8_InputLines.Digital Input Byte 30	6000	1E	UINT8
ReadState8_InputLines.Digital Input Byte 31	6000	1F	UINT8
ReadState8_InputLines.Digital Input Byte 32	6000	20	UINT8
ReadState8_InputLines.Digital Input Byte 33	6000	21	UINT8
ReadState8_InputLines.Digital Input Byte 34	6000	22	UINT8
ReadState8_InputLines.Digital Input Byte 35	6000	23	UINT8

Configuration with the Software „ConfigStage“

SDO- configuration

By „Show CAN objects“ an object browser opens up with all CAN-objects available. These were filtered from the EDS-file automatically.

Move single or multiple CAN-objects by Drag'n Drop into your configuration. In the „Data“ field assign your parameter value.



Decentral external periphery / motion controller pre defined

Pre-defined CAN-devices, especially motion controller, allow the S7-user an immediately access to these devices. Because this configuration data is open to change, it could be adapted to customers needs or extended easily.

INSEVIS offers on its website free of charge various DataBlocks to drive complex motion controllers as e.g. Parker C3, Maxon EPOS2. With these DBs it is possible to control different complex systems with one S7-program

The special advantage is the architecture with different small DBs für each function (jog, acc, dec, home,...). This allows to swap these motion engines with only a few efforts to adapt the S7-program. Normally you can re-write your program, when you replace one drive-specific large DB of one motion vendor to another large DB of another motion vendor.

The complete description with demonstration program and data blocks is available at our website

Configuration with the Software „ConfigStage“

Samples for external CANopen devices

Sample for decentral CANopen periphery ref. to DS401

If a device with digital and analog In- and outputs is used referring to CANopen-Profile DS401, the in/outs are

- digital inputs in TxPDO1, e.g. 32 bit

The screenshot shows the 'Tx PDO Configuration' window. The 'PDO-Nummer' is set to 1 and the 'Beschreibung' is 'Digital Inputs'. Under 'Setting', 'PDO Disable-Flag', 'RTR-not-allowed-flag', and '29-bit-enable-flag' are unchecked. 'COB ID' is set to 'gemäß CIA DS301'. Under 'Kommunikationsparameter', 'Download' is checked, 'Inhibit time [1/10 ms]' is 100, 'Event time [ms]' is 500, and 'Übertragungstyp' is 255. Under 'S7 Mapping', 'Bereich' is 'Eingänge', 'Blocknummer' is 1, and 'Byte-Offset' is 0. The 'Mapping' section shows a table with 8 rows, each with a 'Datentyp' of 'Byte' and a 'Beschreibung' of 'Digital Input Byte 0' through 'Digital Input Byte 7'. Buttons at the bottom include 'Hinzufügen', 'Übernehmen', 'Reset', 'Löschen', and 'Schließen'.

- digital outputs in RxPDO1, e.g. 16 bit

The screenshot shows the 'Rx PDO Configuration' window. The 'PDO-Nummer' is set to 1 and the 'Beschreibung' is 'Digital Outputs'. Under 'Setting', 'PDO Disable-Flag', 'RTR-not-allowed-flag', and '29-bit-enable-flag' are unchecked. 'COB ID' is set to 'gemäß CIA DS301'. Under 'Kommunikationsparameter', 'Download' is checked, 'Inhibit time [1/10 ms]' is 100, 'Event time [ms]' is 500, and 'Übertragungstyp' is 255. Under 'S7 Mapping', 'Bereich' is 'Ausgänge', 'Blocknummer' is 1, and 'Byte-Offset' is 0. The 'Mapping' section shows a table with 8 rows, each with a 'Datentyp' of 'Byte' and a 'Beschreibung' of 'Digital Output Byte 0' through 'Digital Output Byte 7'. Buttons at the bottom include 'Hinzufügen', 'Übernehmen', 'Reset', 'Löschen', and 'Schließen'.

- analog inputs in TxPDO2 to 4

The screenshot shows the 'Tx PDO Configuration' window. The 'PDO-Nummer' is set to 2 and the 'Beschreibung' is 'Analog Inputs'. Under 'Setting', 'PDO Disable-Flag', 'RTR-not-allowed-flag', and '29-bit-enable-flag' are unchecked. 'COB ID' is set to 'gemäß CIA DS301'. Under 'Kommunikationsparameter', 'Download' is checked, 'Inhibit time [1/10 ms]' is 100, 'Event time [ms]' is 500, and 'Übertragungstyp' is 255. Under 'S7 Mapping', 'Bereich' is 'Eingänge', 'Blocknummer' is 1, and 'Byte-Offset' is 4. The 'Mapping' section shows a table with 8 rows, each with a 'Datentyp' of 'Word' and a 'Beschreibung' of 'AnalogueInput Word 0' through 'AnalogueInput Word 7'. Buttons at the bottom include 'Hinzufügen', 'Übernehmen', 'Reset', 'Löschen', and 'Schließen'.

- analog outputs in RxPDO2 to 4

The screenshot shows the 'Rx PDO Configuration' window. The 'PDO-Nummer' is set to 2 and the 'Beschreibung' is 'Analog Outputs'. Under 'Setting', 'PDO Disable-Flag', 'RTR-not-allowed-flag', and '29-bit-enable-flag' are unchecked. 'COB ID' is set to 'gemäß CIA DS301'. Under 'Kommunikationsparameter', 'Download' is checked, 'Inhibit time [1/10 ms]' is 100, 'Event time [ms]' is 500, and 'Übertragungstyp' is 255. Under 'S7 Mapping', 'Bereich' is 'Ausgänge', 'Blocknummer' is 1, and 'Byte-Offset' is 2. The 'Mapping' section shows a table with 8 rows, each with a 'Datentyp' of 'Word' and a 'Beschreibung' of 'Analog Output Word 0' through 'Analog Output Word 7'. Buttons at the bottom include 'Hinzufügen', 'Übernehmen', 'Reset', 'Löschen', and 'Schließen'.

(Images in German language but self explaining)

Therewith the in- and outputs can be used on the configured S7-address (in this sample from EB0, AB0) as usually.



The PDO-number 1 is reserved for digital I/Os only.
The analog I/Os can be mapped starting from PDO-number 2

Configuration with the Software „ConfigStage“

To transfer parameters (e.g. metering ranges for analog modules), SDOs need to be configured specific referring to the vendor.

SDO Configuration

..... SDOs

Show CAN objects

Description: Module Parameter data Slot 1.Parameter Byte 1

Index (hex): 3001

Sub index (hex): 01

Data type: Byte

Data: 255 Decimal

+ Add Apply Reset Delete

Close

Sample configuration CANopen-drive controller ref. to DS402

If a CANopen-drive controller ref. to profile DS402 is used, at least the controller-internal state machine must be managed by

- a 16-bit control word and

Rx PDO Configuration

PDO-Number: 1 Beschreibung:

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant ☐ "Node ID" + Offset (hex) ID (hex): 202 ☒ gemäß CIA DS301

Kommunikationsparameter

☒ Download

Inhibit time [1/10 ms]: 100 Bereich: Datenblock

Event time [ms]: 500 Blocknummer: 101

Übertragungstyp: 255 Byte-Offset: 0

S7 Mapping

Mapping

☒ Download Datenlänge (Bytes): 6 Zeige CAN-Objekte

Datentyp	Index (hex)	Sub index (hex)	Beschreibung
Word	6040	00	Controlword
Word	6060	00	Modes of Operation
Word	607A	00	Target Position

Hinzufügen Übernehmen Reset Löschen

Schließen

- 16-bit status word

Tx PDO Configuration

PDO-Number: 1 Beschreibung:

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant ☐ "Node ID" + Offset (hex) ID (hex): 182 ☒ gemäß CIA DS301

Kommunikationsparameter

☒ Download

Inhibit time [1/10 ms]: 100 Bereich: Datenblock

Event time [ms]: 500 Blocknummer: 100

Übertragungstyp: 255 Byte-Offset: 0

S7 Mapping

Mapping

☒ Download Datenlänge (Bytes): 8 Zeige CAN-Objekte

Datentyp	Index (hex)	Sub index (hex)	Beschreibung
Word	6041	00	Statusword
Word	6061	00	Modes of Operation Display
Word	6100	01	DigitalInput
Word	603F	00	Error code

Hinzufügen Übernehmen Reset Löschen

Schließen

(Images in German language but self explaining)

Assign these both words to S7-operands by the PDO-mapping and then the controller can be driven by the S7-program.

System functions

INSEVIS offers OBs, SFBs and SFCs, who are compatible to STEP®7 from Siemens. But INSEVIS offers more: There are lots of OBs, SFBs and SFCs with special functions available in a free S7-library at the website.

Compared to Siemens-CPU S7-315-2PN DP with Firmware 3.2.7 (6ES7 315-2EH14-0AB0) at INSEVIS-CPU type V and type P these OB, SFB and SFC are available.

Organisation blocks (OB)

Organisation blocks (OB) - compatible to STEP®7 from Siemens

OB	Function
OB 1	Cycle-OB
OB 10	Time alert
OB 32, 33, 34	Cyclic interrupt (CPU-T only)
OB 35	Cyclic interrupt
OB 80	Time error
OB 82	Diagnostic Alert
OB 85	Program sequence error
OB 86	Device failure (at INSEVIS failure of external periphery) <i>Local data „OB86_MDL_ADDR“ (LW6) has a fix value 2047 (W#16#7FF)</i>
OB 100	Restart
OB 121	Programming error
OB 122	Periphery access error

Organisation blocks (OB) - in addition to STEP®7 from Siemens

OB	Function <i>(more information at single OB-chapters)</i>
OB 106	CAN- receive event (see „Communication / CAN“)
OB 186	CAN- error event

System function blocks (SFB)

System function blocks (SFB) - compatible to STEP®7 from Siemens

SFB	Function
SFB 0 CTU	Forward counter
SFB 1 CTD	Backward counter
SFB 2 CTUD	Forward- and backward counter
SFB 3 TP	Pulse generation
SFB 4 TON	Switch on delay
SFB 5 TOF	Switch off delay
SFB 60	Send data with ASCII- driver (from CPU S7-314, order-no. 6ES7314-1AG14-0AB0)
SFB 61	Receive data with ASCII- drive (from CPU S7-314, order-no. 6ES7314-1AG14-0AB0)

System function blocks (SFB) - what are not available at INSEVIS-CPU yet or available on request

SFB	Function
SFB 32 DRUM	Realization of a step switching systems
SFB 52 / 53 / 54 / 81	These functions are not available because on different CPU-hardware
SFB 73 / 74	Profibus DP → internal Siemens block, not necessary for INSEVIS
SFB 75 SALRM	These functions are out of the INSEVIS-Profibus functionality DP V0
SFB 104 IP_CONF	Configure IP-addresses → use SFB 129 IP_CFG instead

System functions

System function blocks (SFB) - in addition to STEP®7 from Siemens

SFB	Function (more information at single SFB-chapters)
SFB 105	CAN-sending
SFB 106	CAN-receiving
SFB 107	CAN-SDO-transfer
SFB 114	CAN-status request
SFB 120 GET	Read data from a Remote PLC
SFB 121 PUT	Write data to a Remote PLC
SFB 122 TSEND	Send data over TCP
SFB 123 TREC	Receive Data over TCP
SFB 124 TDISCON	terminate TCP-connection
SFB 125 TCONFIG	Configure partner parameter at TCP
SFB 126 TSTATUS	State query at TCP
SFB 127 USEND	Send data over UDP
SFB 128 URECV	Receive data over UDP
SFB 129 IP_CFG	IP-configuration of the PLC
SFB 130 CFG_PTP	Change UART-configuration
SFB 131 IP_GET	Read IP-configuration (CPU-T only)
SFB 132 IP_SET	Set IP-configuration (CPU-T only)

System functions (SFC)

System functions (SFC) - compatible to STEP®7 from Siemens

SFC	Function
SFC 0 SET_CLK	Set system clock
SFC 1 READ_CLK	Read system clock
SFC 6 RD_SINFO	Read OB start information
SFC 11 DPSYC_FR	Synchronize Groups of DP slaves <i>Parameter „LADDR“ must be 2047 (W#16#7FF)</i>
SFC 12 D_ACT_DP	Deactivation and activation of DP slaves
SFC 13 DPNRM_DG	Read diagnostic data of a DP slave (Slave Diagnostics)
SFC 14 DPRD_DAT	Read consistent data of a standard DP Slave
SFC 15 DPWR_DAT	Write consistent data to a DP standard Slave
SFC 20 BLKMOV	Copy Variables
SFC 21 FILL	Initialize a memory area
SFC 24 TEST_DB	Testing Datablock
SFC 28 SET_TINT	Set time-of-day interrupt
SFC 29 CAN_TINT	Cancel time-of-day interrupt
SFC 30 ACT_TINT	Activate time-of-day interrupt
SFC 31 QRY_TINT	Query time-of-day interrupt
SFC 43 RE_TRIGR	Retrigger cycle time monitoring
SFC 46 STP	Change the CPU to STOP
SFC 51 RDSYSST	Reading a System Status List <i>(only supported the SSL_ID W#16#0092, W#16#0292, W#16#0692 for Profibus DP master system)</i>
SFC 64 TIME_TCK	Read the system time
SFC 101 RTM	Handling runtime meters

System functions

System functions (SFC) - what are not available at INSEVIS-CPU's yet or only available on request

SFC	Function
SFC 2 / 3 / 4	Use SFC101 (32bit counter for operating hours instead)
SFC 5 / 50 / 70 / 71	These functions are not available because on different CPU-hardware
SFC 7 / 55 / 56 / 57 / 102	These functions are out of the INSEVIS-Profibus functionality DP V0
SFC 17 / 18 / 19 / 107 / 108	Use alarm / event management in the VisuStage instead
SFC 22 CREAT_DB	Create a data block
SFC 23 DEL_DB	Delete a data block
SFC 32 / 33 / 34 / 47	INSEVIS does not support time delay interrupts
SFC 36	Mask synchronuos error events
SFC 37	Demask synchronuos error events
SFC 38	Read event state register
SFC 39	at CPU-T only / not available for CPU-V/P Block processing of new alarm- and asynchronous error events
SFC 40	at CPU-T only / not available for CPU-V/P Release processing of new alarm- and asynchronous error events
SFC 41	at CPU-T only / not available for CPU-V/P Retard processing of new alarm- and asynchronous error events with higher priority
SFC 42	at CPU-T only / not available for CPU-V/P Release processing of new alarm- and asynchronous error events with higher priority
SFC 44	Carry over default value into AKKU 1
SFC 58	Write record
SFC 59	Read record
SFC 65 / 66 / 67 / 68 / 69 / 72 / 73 / 74	INSEVIS does not support MPI- communication
SFC 81 UBLKMOV	Copy variable in-interruptable
SFC 82 CREA_DBL	Create a data block in load memory, not necessary for INSEVIS
SFC 83 READ_DBL	Read from data block in load memory, not necessary for INSEVIS
SFC 84 WRIT_DBL	Write into data block in load memory, not necessary for INSEVIS
SFC 99 WWW	internal Siemens block, not necessary for INSEVIS

System functions (SFC) - in addition to STEP®7 from Siemens

SFC	Function <i>(more information at single SFC-chapters)</i>
SFC 75 SET_ADDR	Assign a Profibus DP slave address
SFC 76 SET_DPS	Assign a Profibus DP slave address and baud rate
SFC 200 SCR_GET	Query the current the screen ID number
SFC 201 SCR_SET	Changing the screen
SFC 202 TRENDSRT	Starting the trend sampling
SFC 203 TRENDSTP	Stopping the trend sampling
SFC 204 TRENDCNT	Continue the trend sampling
SFC 205 TRENDSTA	Query the trend status
SFC 206 RECIPE	Handling with recipe records
SFC 207 ARCHIVE	Archiving data to / create file on Micro-SD-Card
SFC 208 BACKUP	Backup
SFC 209 READFILE	Read data from archive file on Micro-SD-Card
SFC 210 COPY_DBL	Copy selected DBs from RAM to ROM

System functions

System functions (SFC) - in addition to STEP®7 from Siemens

SFC	Function (more information at single SFC-chapters)
SFC 211 COPY_DBW	Copy selected DBs from ROM to RAM
SFC 213 READ_CSV	Read data from CSV-formatted file on Micro-SD-Card
SFC 214 LCD_BRIG	Change LCD-backlight brightness
SFC 215 LOGIN	Change user level at runtime
SFC 220 RENAME	Rename files/folders on Micro-SD-Card
SFC 221 DELETE	Delete files/folders on Micro-SD-Card
SFC 222 COPY	Copy files on Micro-SD-Card
SFC 223 CREATE	Create folders on Micro-SD-Card
SFC 224 INFO	Inform about files/folders on Micro-SD-Card
SFC 225 FILEINFO	Inform about file names and sizes on Micro-SD-Card
SFC 254 RAM2ROM	Copy RAM to ROM

Import of INSEVIS-blocks into the S7-Project

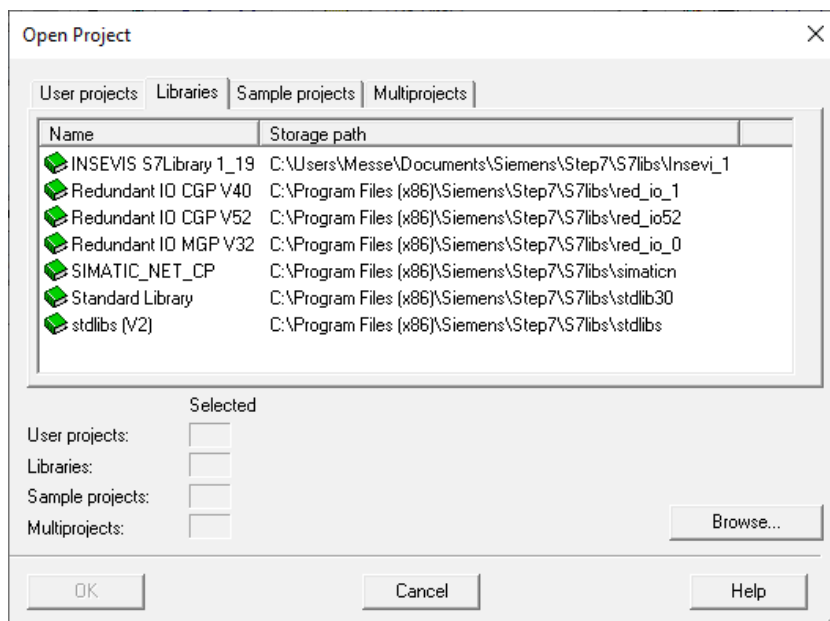


ATTENTION:

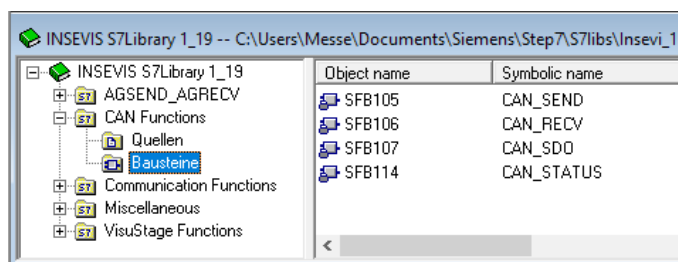
All additional INSEVIS- blocks are available as **S7-Library** for free download at the INSEVIS-web sites in the download area. It will be offered always the newest library with the most functions, what needs the referring firmware in the PLC. (e.g. Insevis_S7-library_from_2_1_5.zip - works from firmware 2.1.5 and higher).

Installation of the INSEVIS-S7-library in the Simatic®-manager

1. File → dearchive → choose the INSEVIS S7-library.zip file
2. Store → (choose your folder „S7-Libs“) → press „OK“
3. Open the library project
→ go to „Libraries“
→ choose „INSEVIS S7-Library“
→ press „OK“



4. Copy the selected SFB's by drag'n drop into your project

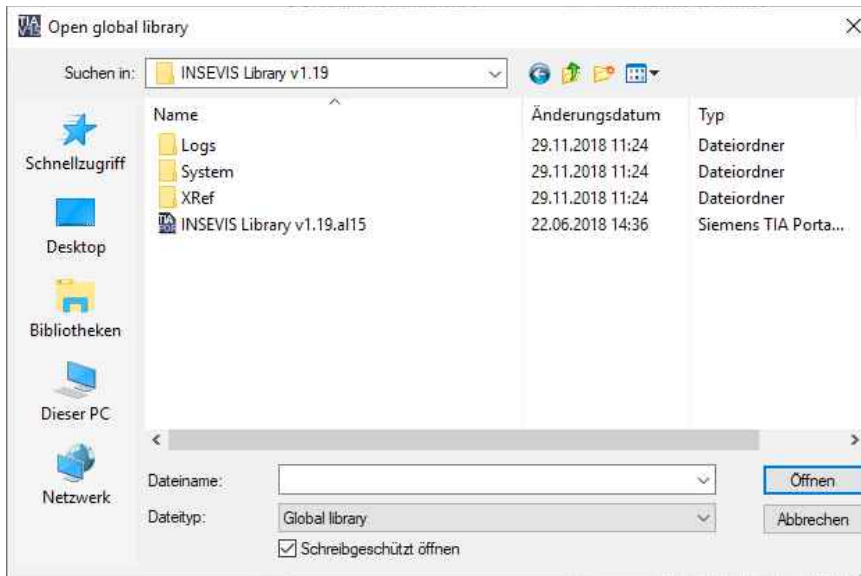


VIDEO-tutorial available

There is available a link to the corresponding YouTube®-video at the INSEVIS-download website for use under Simatic®-Manager.

System functions

Installation of the INSEVIS-S7-library in the TIA-Portal

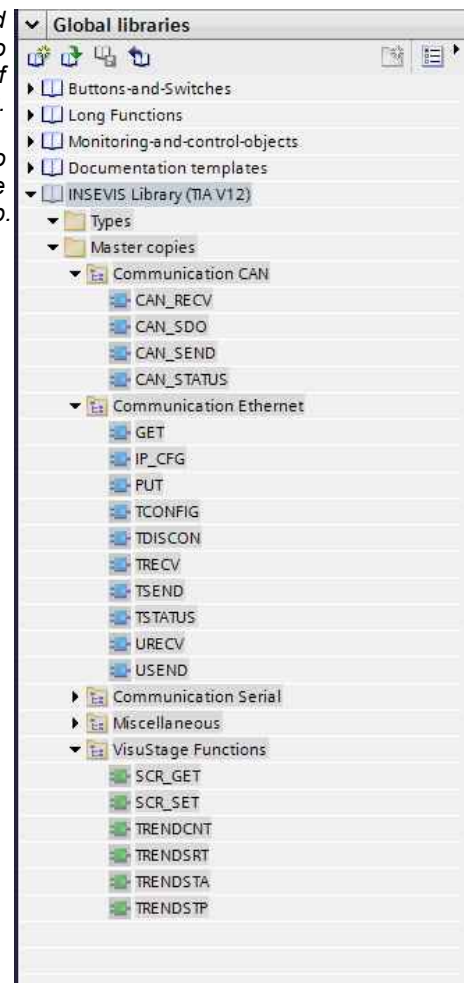


1. Download latest version of INSEVIS library for TIA-portal
2. store it and extract it in any folders
3. Open library **"INSEVIS Library v1.19a115"** from the referring folder in TIA-Portal as "global library".→

→ For more details to „open a library“ use TIA-Portal online help.

4. Drag'n drop the desired library elements into the program blocks of your project.

→ For more details to „open a library“ use TIA-Portal online help.



VIDEO-tutorial available

There is available a link to the corresponding YouTube®-video at the INSEVIS-download website for use under TIA-Portal.

System functions

Overview about SFCs for data backup and restore

Here you find a selection of possibilities to save or backup and restore your data manually or by System-SFCs.

Please note: The S7-Program itself is always kept in the load memory (Flash) at INSEVIS.

Save data blocks to ROM/ into flash memory

With the **SFC 254** RAM2ROM all DBs will be copied from battery buffered work memory (RAM) to the load memory (ROM), so they are stored permanently.

SFC 210 COPY_DBL copies only selected DBs from battery buffered work memory (RAM) to the load memory (ROM).

SFC 211 COPY_DBW work in opposite direction and copies selected DBs from load memory (ROM) to battery buffered work memory (RAM).

→ sample1: putting serial devices in operation

- After program download (see below) the device-specific process data will be assigned manually and are kept in battery buffered work memory (RAM)
- before shipping this device these data will be saved to load memory (ROM) and are protected against low battery voltage

→ sample 2: save process data at operation by end user

- If process data are modified by end user, he can save it by access limited button or key switch with RAM2ROM function.

Life-backup for a change/ spare device

With the **SFC 208** BACKUP all data (S7-Program, visualization, process data) will be combined to a binary (encryptable) and written to the Micro-SD-card. For that process the CPU will be switched into STOP mode and return, when it is ready.

With this file the end user can create a spare device with similar data (need a similar hardware) without any PC-programs. To secure data a password file can be activated.

→ sample: Create a spare machine

- A customer must have a 1:1 spare machine to go on producing while maintaining the main machine. It is necessary to keep process data of last production lot of the main machine.
- On every change of production lots in the main machine a backup process is started
- Inserting the Micro-SD-Card from the main machine to the spare machine, it will go on with their last data.

Manual program-backup for manufacturers

There is a chance to create a backup manually and without any SFCs for programmers with access to the source code:

- Create a folder „**backups**“ on the Micro-SD-card on the PC,
- Create a S7-Memory-Card-file with the name „**s7programs.wld**“ with menu: File→ "Memory Card File" → New
- Copy S7-Programs, data blocks **and system data** (contain ConfigStage-configuration) into S7-Memory-Card-file by Drag'n Drop, store and close this file and copy it into the directory „backups“ on the Micro-SD-card.
- Create a visualization binary by compiling it in the VisuStage, rename it in „**visudata.vsb**“ and copy this file into the directory „backups“ on the Micro-SD-card too.
- From ServiceStage 1.0.1.7 this all can be done by the new function "Create online backup" automatically

→ Sample: Easy creating multiple devices with the same program data

- A programmer sends this Micro-SD-card with all the program data to his customer
- After hardware reset every PLC loads the same program data from this card - complete without PC-programs

S7-Program-restore for both kinds of backup

Regardless how a backup was created, it is restored from the Micro-SD-card always in the same way:

- Power OFF, insert Micro-SD-card with the backup into the PLC
- Power ON, do a hardware-reset and wait, until RUN- and BAT-LEDs do no more blink together
(If the backup was created by SFC208, add the password-file „*backup.psw*“ to the Micro-SD-card data)

System functions

Archive/ dearchive (write back) data

Free definable data can be copied (archived) to a file (**binary or csv-format**) by **SFC 207 ARCHIVE**.

Binary data can be copied back (dearchived) by the **SFC 209 READFILE**. Also a read in the RemoteStage-PC by Ethernet from Micro-SD-card is possible.

→ sample: **Save process data while S7-Update**

- A customer created lots of process data by his S7-program.
- Now he should get an S7 update/upgrade with new functions.
- All process data are lost normally if the new S7-program is downloaded to the PLC.
- With the **SFC 207 ARCHIVE** and **SFC 209 READFILE** these data can be „evacuated“ to the Micro-SD-card during the S7-update and „brought back“ after it.
- So the customer can go on with a new S7-program and old process data.

Read in data from PC-edited CSV-archive file

If you created a csv-file by **SFC 207 ARCHIVE** to the Micro-SD-card, it can be modified on the PC with an editor, stored and downloaded into the PLC by **SFC 213 READ_CSV**.

→sample: **Customized configuration by an operator csv-file**

- Your S7-program is always the same, but the pre-settings are different for each of your customer and must be assigned before it can be shipped.
- A csv-file (saved as data archive) can be used as project configuration file and can be edited on the PC by the team, who assigns customer settings.
- With the **SFC 213 READ_CSV** you can read in the csv-file from the Micro-SD-card back to the PLC and your serial program is assigned with customized values.

Archiving DBs with SFC 207 "ARCHIVE"

With this SFC207 „ARCHIVE“ S7-(Process or data block) data can be archived in SD card as binary or text (CSV) format.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE= Assign and start job
FILENAME	INPUT	STRING	File name (must be in 8.3 format)
APPEND	INPUT	BOOL	TRUE= Append to existing file FALSE= Overwrite existing file
CSV	INPUT	BOOL	TRUE= Convert data into text and write as text record (CSV) format. FALSE= Write data as binary to file
CSV_STRUCT	INPUT	ANY	CSV record structure. This parameter is ignored if CSV is FALSE.
DATA	INPUT	ANY	Pointer to S7-Process data
RET_VAL	OUTPUT	WORD	Status code of job
BUSY	OUTPUT	BOOL	TRUE = Job running, FALSE = Job done
DONE	OUTPUT	BOOL	TRUE= Job successfully done
ERROR	OUTPUT	BOOL	TRUE= Job failed

Notes (CSV formatted text file; CSV=TRUE):

- Field delimiter character is “;” semicolon, ASCII code 59)
- Decimal point is “.” (point, ASCII code 46)
- Record separated by carriage return line feed character (CRLF ASCII code 13, 10)
- Please take care about SIMATIC S7-(Process or Data block) data offset alignments in data block. (Data in data block always starts from even offset expect BYTE, CHAR and BOOL data type)
- CSV_STRUCT points to record structure of text CSV file.
Each field of record in CSV file consists 2 Byte in CSV_STRUCT.
 - First byte indicates data type where as
 - second byte indicates how to convert data to text.

System functions



ATTENTION:

The structure of a data archive is assigned by S7-program only.
The RemoteStage can read in, display an store it, but does not know the structure and that's why it can not edit or modify theses data archives. This can be done by customer with his PC tools only.

Meaning of CSV_STRUCT

First byte	Second byte	Format
1	Data is BOOL	
	0	As integer 0 or 1
	1	As text: FALSE or TRUE
2	Data is BYTE	
	0	As decimal integer 0 .. 255
	1	As hexadecimal B#16#0 .. B#16#FF
	2	As hexadecimal 0x00 .. 0xFF
3	Data is CHAR	
	0	As character if value is between B#16#20 .. B#16#80. e.g A
	1	As hexadecimal B#16#0 .. B#16#FF
	2	As hexadecimal 0x00 .. 0xFF
	3	As decimal integer -128 ... 127
4	Data is WORD	
	0	As decimal integer 0 ... 65535
	1	As hexadecimal W#16#0 .. W#16#FFFF
	2	As hexadecimal 0x0000 .. 0xFFFF
5	Data is INT	
	0	As decimal integer -32768 ... 32767
	1	As hexadecimal W#16#0 .. W#16#FFFF
	2	As hexadecimal 0x0000 .. 0xFFFF
6	Data is DWORD	
	0	As decimal integer 0 ... 4294967295
	1	As hexadecimal DW#16#00000000 .. DW#16#FFFFFFFF
	2	As hexadecimal 0x00000000 .. 0xFFFFFFFF
7	Data is DINT	
	0	As decimal integer -2147483648 ... 2147483647
	1	As hexadecimal DW#16#00000000 .. DW#16#FFFFFFFF
	2	As hexadecimal 0x00000000 .. 0xFFFFFFFF
8	Data is REAL	
	0..4	Number of digits after decimal point (e.g 12.456)
	Otherwise	2 digits after decimal point
		Note: Displays NaN (Not A Number) if data is not REAL format

System functions

First byte	Second byte	Format
9	Data is DATE	
	0	D#YYYY-MM-DD (e.g D#2013-07-24)
	1	As decimal integer, 0..65535 = number of days since 1990.01.01
	2	As hexadecimal W#16#0 .. W#16#FFFF
	3	As hexadecimal 0x0000 .. 0xFFFF
10	Data is TIME_OF_DAY (TOD)	
	0	TOD#HH:MM:SD.MS (e.g TOD#12:45:23.102) if hour value is larger than 23 then, as decimal integer 0 ... 4294967295
	1	As decimal integer 0 ... 4294967295
	2	As hexadecimal DW#16#00000000 .. DW#16#FFFFFFFF
	3	As hexadecimal 0x00000000 .. 0xFFFFFFFF
11	Data is TIME	
	0	T#dhmsMS (T#24d20h31m23s647ms) d=day h=hour m=minute s=second MS=millisecond if one of time unit is 0, then its not displayed. e.g T#20d2h10s130ms
	1	As decimal integer -2147483648 ... 2147483647
	2	As hexadecimal DW#16#00000000 .. DW#16#FFFFFFFF
	3	As hexadecimal 0x00000000 .. 0xFFFFFFFF
12	Data is S5TIME	
	0	S5T#hmsMS (S5T#2h46m30s) h=hour m=minute s=second MS=millisecond
	1	As decimal integer 0 ... 65535
	2	As hexadecimal W#16#0 .. W#16#FFFF
	3	As hexadecimal 0x0000 .. 0xFFFF
13	Data is STRING	
	<ignored>	Only actual length characters are copied
14	Data is DT	
	0	DT#YYYY-MM-DD-HH:MI:SS:MS YYYY – year 1990 – 2089 MM – month DD – day HH – hour MI – minute SS – second MS – millisecond e.g. DT#2013-07-24-13:05:12.340
	1	8 byte stream e.g 1307241305123404

System functions

Status code meaning

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Job is successfully done
W#16#7000	1	0	0	Job is process
W#16#7001	0	0	0	Job resource is temporary locked
W#16#7002	0	0	0	REQ Parameter is FALSE
W#16#7003	0	0	1	Job is not accepted, due to previous job is still in process
W#16#8000	0	0	1	File system is not ready (e.g SD card not inserted)
W#16#8001	0	0	1	Could not write to file
W#16#8002	0	0	1	Invalid data type ID number in CSV_STRUCT
W#16#8003	0	0	1	CSV_STRUCT – size is must be even number
W#16#822A	0	0	1	Invalid format in FILENAME parameter
W#16#8xyy	0	0	1	General error code (S7 compatible)

S7-program sample for using the SFC207

Purpose: Write data (DB207) to archive file in SD card. File name is in DB207.

```

CALL  SFC  207
REQ    :=M1501.3           // TRUE = Request to archive data
FILENAME :=DB207.ARCHIVE_NAME // Archive file name
APPEND  :=TRUE             // Appends to the end of file
CSV     :=FALSE            // as binary data
CSV_STRUCT:=DB207.ARCHIVE_CSV_FORMAT // CSV field format
DATA    :=DB207.ARCHIVE_DATA // Data to archive
RET_VAL :=MW1000           // Return / result code
BUSY    :=M1701.3         // TRUE = Request is in process
DONE    :=M1002.0         // TRUE = Successfully archived
ERROR   :=M1002.1         // TRUE = Error occurred

U      M    1701.3         // IF Archive data is in process; then
BEB
R      M    1501.3         // end this function
                        // reset request signal

O      M    1002.0         // If successful done,
ON     M    1002.1         // or no error, then
BEB
                        // end the function

L      MW    1000         // error occurred
                        // error evaluation
                        // ...

BEA
                        // end of function

```


System functions

Backup & Restore with the SFC 208 "BACKUP"

With the SFC208 „BACKUP“ you can backup the PLC data including user S7-program, process-data, runtime remanent data, visualization binary and visualization runtime remanent data to SD card and restore it from this Micro-SD card into a PLC with the same hardware configuration.

This procedure can only be processed in STOP condition. A password prevents an unauthorized backup. All protection levels will be kept in the backed up data as well.

Procedure

- Implement backup function by SFC208 into S7-program and start it (e.g. by a button with special access rights or a key switch).
→ Backup-file *backup.bin* will be created on Micro-SD-card in the folder „Backups“.
- Pull off Micro-SD-card and insert it into a PLC with the same hardware configuration.
- Optional:*
 - Store a password defined in S7-program as *backup.psw* in the same folder like backup file on the Micro-SD-card (see picture on next page).
- Proceed the [general reset by hardware](#) during PLC is booting.
All data will be transferred from Micro-SD-card into the PLC and written as restore into the device.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	Request: TRUE assign and start job
LADDR	INPUT	WORD	Logical base address = 0 (for all Panel-PLCs) Logical base address = 1...16 (connection-ID-No. to the Panel-HMI)
ATTRIBUTE	INPUT	BYTE	Attributes Bit0: 1= Backup WITH actual data of Retain-DBs 0= Backup WITHOUT actual data of Retain-DBs Bit1: 1= Backup WITH remanent S7-memory 0= Backup WITHOUT remanent S7-memory (remanent merker, S7-Timer, counter, operating hours) Bit2: 1= Backup WITH VisuStage RunTime memory 0= Backup WITHOUT VisuStage RunTime memory (Message, trend actual buffer, user PINs) Bit3: 1= Newstart PLC after backup end 0= NO newstart, PLC stays in STOP mode Bit7: 1= Decrypt backup file WITH password 0= Decrypt backup file WITHOUT password
PASSWORD	INPUT	ANY	Pointer to Keyword for encryption. Data type must be BYTE or CHAR. Keyword must be min. 5 and max. 16 characters long.
RETVAL	OUTPUT	WORD	Status code of job W#16#... 7000 = Backup is requested, and in process 7001 = HMI- connection is in invalid state (e.g not connected) 7002 = Job is not accepted, due to previous job is in process 7003 = Connection resource is temporary locked 7004 = REQ parameter is FALSE (Backup is not accepted) 8001 = Invalid connection ID number 8003 = SD card not inserted 842A = PASSWORD parameter is not BYTE or CHAR type 842B = PASSWORD parameter length is invalid 8xyy = General error code (S7 compatible)
ERROR	OUTPUT	BOOL	TRUE= Job is failed, error code in RETVAL



ATTENTION:

S7 programmer must take care about process stability and security **before** calling SFC208. Calling SFC208 stops the PLC immediately and starts backup procedure. During this procedure PLC stays in STOP mode and will only return to RUN, when assigned in ATTRIBUTE BIT3 = 1

System functions

Note:

- The call SFC208 "BACKUP" causes the device (HMI, or PLC) to change to STOP mode and immediately writes data to Micro SD card.
- Backup process takes normally up to 5 seconds.
- The STOP event is indicated by diagnose buffer entry with EVENT ID W#16#497C "STOP triggered by integrated technology".
- Backup result is indicated by diagnose buffer entry with

EVENT ID	16#BF00
OB	16#00
PK	16#FF
DatID1/2	Result
	16#0002 = Successful
	16#0001 = Backup failed
- Backup result is indicated by OB100_STRT_INFO parameter of OB100 by next NEWSTART.
OB100_STRT_INFO (bit 16..23)

16#30	= Backup successful done
16#31	= Backup failed
- If Bit 3 of ATTRIBUTE parameter is set (TRUE), then PLC is changes again to NEWSTART (RUN) after Backup process end. Otherwise PLC left in STOP mode.

S7-program sample for using the SFC208

Purpose: Create a S7- User program and remanent data backup in SD card with encrypted.

```

U      M      1501.5      // backup request is set
FP      M      1601.5      // only start at rising edge of
                                // REQ_BACKUP_PROGRAM

SPBN      n000

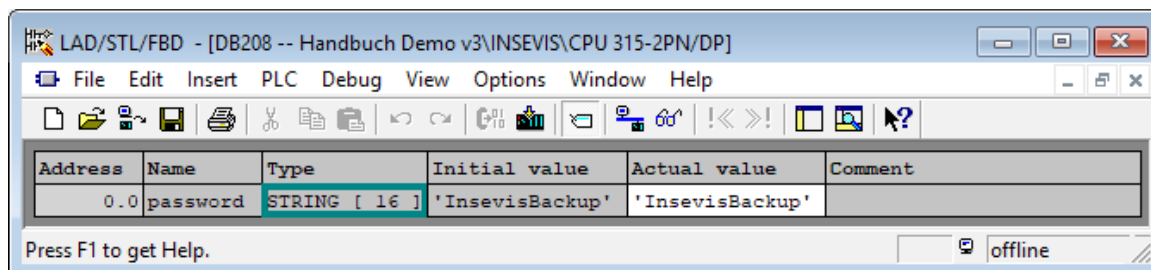
CALL      SFC      208
REQ      :=TRUE      // TRUE = Start the backup
LADDR      :=MW2000      // VisuStage connection ID number.
                                // In Panel-PLC (e.g PC700P) must be W#16#0.

ATTRIBUTE:=B#16#8F      // Encode with password, backup all data
PASSWORD :=P#DB208.DBX2.0 BYTE 13      // password
RET_VAL  :=MW1000      // Result / return code
ERROR    :=M1002.1      // TRUE = Error occurred

UN      M      1002.1      // if no error, then
BEB      // end the function
R      M      1501.5      // otherwise, reset the request signal

L      MW      1000      // error occurred
                                // error evaluation
                                // ...

BEA      // end of function
  
```



ATTENTION:

To BACKUP successfully, store the password defined in S7-program as backup.psw in the same folder on the Micro-SD-card like the backup file itself.

System functions

Read data from archive file (SD) with SFC 209 "READFILE"

This function is used to copy data from archive file (SD) to S7 data area (e.g Data block).

- Note:**
- This SFC is asynchronous, it means its processing extends over several SFC calls.
 - This function should not interrupted (e.g., Power OFF)

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	REQ = TRUE: Request to read from the file
FILENAME	INPUT	STRING	File name, Length must be smaller than 60 characters
OFFSET	INPUT	DWORD	Starting offset to read from file beginning
LEN	INPUT	WORD	Data size to read in bytes, LEN = 0: Read available data starting OFFSET
RET_VAL	OUTPUT	WORD	Status code
BUSY	OUTPUT	BOOL	TRUE : File reading is in process
DONE	OUTPUT	BOOL	TRUE : File reading is ended successful
ERROR	OUTPUT	BOOL	TRUE : File reading is ended with error
DSTBLK	OUTPUT	ANY	Destination block area to write data
LEN_R	OUTPUT	WORD	Number of actually written data to DSTBLK in bytes

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Successful read
W#16#0080	0	1	0	Read done, Available data is smaller than requested.
W#16#0081	0	1	0	Destination area (DSTBLK) is larger than requested data (LEN), only requested data copied.
W#16#8081	0	1	0	Requested (LEN) data is larger than destination area (DSTBLK), only data copied which can fit to destination area.
W#16#7000	1	0	0	Reading is in in process
W#16#7001	0	0	0	Internal resource temporary occupied
W#16#7002	0	0	0	REQ = FALSE, No reading started
W#16#7003	0	0	1	Internal resource occupied by other request, (multiple request)
W#16#8000	0	0	1	SD card is not inserted
W#16#8001	0	0	1	File is not found, or could not read file
W#16#8002	0	0	1	File is empty
W#16#8003	0	0	1	OFFSET parameter is larger than file size
W#16#822A	0	0	1	FILENAME parameter is invalid.
W#16#8xyz	0	0	1	General error code (S7 compatible)

System functions

S7-program sample for using the SFC209

Purpose: Read data from file in SD card into S7-data area

```
CALL SFC 209
REQ      :=M1501.6           // TRUE = Request to read file
FILENAME:=DB209.ARCHIVE_NAME // File name
OFFSET   :=DW#16#0          // From beginning of file
LEN       :=W#16#0           // Read all data which fits to DST_BLK
RET_VAL   :=MW1000           // Return / result code
BUSY      :=M1701.6          // TRUE = Request is in process
DONE      :=M1002.0          // TRUE = Successfully read
ERROR     :=M1002.1          // TRUE = Error occurred
DSTBLK    :=DB209.ARCHIVE_DATA // Destination data area which data be written
LEN_R     :=MW1004           // Number of bytes to written into DSTBLK

U      M   1701.6           // Archive data is in process
BEB
R      M   1501.6           // reset request signal

O      M   1002.0           // If successful done,
ON     M   1002.1           // or no error, then
BEB                                         // end the function

L      MW   1000           // error occurred
                                         // error evaluation
                                         // ...

BEA                                         // end this function
```



The FAT16/32 Filesystem on the Micro-SD-Karte does not allow some signs in the file- or directory name

These are: \ / : * ? " < > |

These signs will be ignored.

System functions

Copy data blocks from work memory to load memory with SFC 210 "COPY_DBL"

This function is used to copy data block(s) from work memory to load memory.

- Note:**
- This SFC is asynchronous, it means its processing extends over several SFC calls.
 - This function should not be interrupted (e.g., Power OFF, or PG memory related functions, such as Clear all, Block delete, ...).
 - Data blocks which are marked as "Non-Retain", are also copied (actual data) to load memory.
 - If parameter LOW_NR = HIGH_NR, then only one data block with LOW_NR will be copied.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	REQ = TRUE : Request to copy the data blocks from work to load memory
LOW_NR	INPUT	WORD	Lowest data block number. Must be larger than 0.
HIGH_NR	INPUT	WORD	Highest data block number. Must be larger than 0.
RET_VAL	OUTPUT	WORD	Status code
BUSY	OUTPUT	BOOL	TRUE : Copying is in process

RET_VAL	BUSY	Description
W#16#0000	0	Successfully copied
W#16#7000	0	First call with REQ=0, Copying is not started
W#16#7001	1	First call with REQ=1, Copying is started
W#16#7002	1	Intermediate call (REQ irrelevant): Copying is in process
W#16#8001	0	Other memory manipulation function is active. (e.g PG Block download, delete etc..) Copying is not started.
W#16#8002	0	Invalid data block number (HIGH_NR is smaller than LOW_NR)
W#16#822A	0	LOW_NR is invalid.
W#16#832A	0	HIGH_NR is invalid.
W#16#8xyz	0	General error code (S7 compatible)

S7-program sample for using the SFC210

Purpose: Copy data block DB209 to load memory. Actual values are overwritten in load memory.

UN	M	1502.0	// If request is not set, then
BEB			// end the function
CALL	SFC	210	
REQ	:	=M1502.0	// TRUE = Request to copy data block(s) to load memory
LOW_NR	:	=W#16#D1	// Data block number 209 (dec)
HIGH_NR	:	=W#16#D1	// only one data block
RET_VAL	:	=MW1000	// Return / result code
BUSY	:	=M1702.0	// TRUE = Request is in process
U	M	1702.0	// if Copy request is in process, then
BEB			// end the function
R	M	1502.0	// reset the request signal
L	MW	1000	// return / result code
L	0		// if no error, then
==I			
BEB			// end the function
L	MW	1000	// error occurred
			// error evaluation
			// ...
BEA			// end the function

System functions

Copy data blocks from load memory to work memory with SFC 211 "COPY_DBW"

This function is used to copy data block(s) from load memory to work memory.

- Note:**
- This SFC is asynchronous, it means its processing extends over several SFC calls.
 - This function should not interrupted (e.g., Power OFF, or PG memory related functions, such as Clear all, Block delete, ...).
 - If parameter LOW_NR = HIGH_NR, than only one data block with LOW_NR will be copied.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	REQ = TRUE: Request to copy the data blocks from load to work memory
LOW_NR	INPUT	WORD	Lowest data block number. Must be larger than 0.
HIGH_NR	INPUT	WORD	Highest data block number. Must be larger than 0.
RET_VAL	OUTPUT	WORD	Status code
BUSY	OUTPUT	BOOL	TRUE : Copying is in process

RET_VAL	BUSY	Description
W#16#0000	0	Successfully copied
W#16#7000	0	First call with REQ=0, Copying is not started
W#16#7001	1	First call with REQ=1, Copying is started
W#16#7002	1	Intermediate call (REQ irrelevant): Copying is in process
W#16#8001	0	Other memory manipulation function is active. (e.g PG Block download, delete etc..) Copying is not started.
W#16#8002	0	Invalid data block number (HIGH_NR is smaller than LOW_NR)
W#16#822A	0	LOW_NR is invalid.
W#16#832A	0	HIGH_NR is invalid.
W#16#8xyz	0	General error code (S7 compatible)

S7-program sample for using the SFC211

Purpose: Copy data block (DB209) to work memory. Actual values which are last saved with SFB210 or SFC25 or downloaded from PG, are overwritten in work memory.

```

UN    M    1502.1      // If request is not set, then
BEB                                     // end the function

CALL  SFC  211
REQ   :=M1502.1        // TRUE = Request to copy data block(s) to work memory
LOW_NR :=W#16#D1        // Data block number 209 (dez)
HIGH_NR:=W#16#D1        // only one data block
RET_VAL:=MW1000         // Return / result code
BUSY   :=M1702.1        // TRUE = Request is in process

U      M    1702.1      // if Copy request is in process, then
BEB                                     // end the function
R      M    1502.1      // reset request signal

L      MW    1000        // return / result code
L      0                // no error, then
==I
BEB                                     // end the function

L      MW    1000        // error occurred
// error evaluation
// ...

BEA                                     // end the function

```

System functions

Read data from CSV-formatted archive file (SD) with SFC 213 „Read_CSV“

This function is used to copy data from CSV formatted archive file (SD) to S7 data area (e.g. Data block). This SFC is asynchronous, it means its processing extends over several SFC calls. Only one job is queued at the time for processing.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	REQ = TRUE: Request to read from the file
FILENAME	INPUT	STRING	File name, Length must be smaller than 60 characters
CSV_STRUCT	INPUT	ANY	CSV record structure. See SFC 207 "ARCHIVE" documentation
RET_VAL	OUTPUT	WORD	Status code
BUSY	OUTPUT	BOOL	TRUE : File reading is in process
DONE	OUTPUT	BOOL	TRUE : File reading is ended successful
ERROR	OUTPUT	BOOL	TRUE : File reading is ended with error
LINE	OUTPUT	INT	By ERROR=TRUE: Line number which causes converting error
FIELD	OUTPUT	INT	By ERROR=TRUE: Field number which causes converting error
DATA	INPUT	ANY	Pointer to S7-Process area

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Successful read
W#16#7000	1	0	0	Reading is in process
W#16#7001	0	0	0	Internal resource temporarily occupied
W#16#7002	0	0	0	REQ = FALSE, No reading started
W#16#7003	0	0	1	Internal resource occupied other Request (e.g. SFC207 "ARCHIVE" ...)
W#16#8000	0	0	1	SD card is not inserted
W#16#8001	0	0	1	File is not exist in SD
W#16#8002	0	0	1	Invalid data type in CSV_STRUCT parameter
W#16#8003	0	0	1	Invalid length in CSV_STRUCT parameter
W#16#8004	0	0	1	S7-Process data area should start from even offset (DATA parameter)
W#16#8005	0	0	1	File is empty
W#16#8006	0	0	1	Error occurred on reading the file
W#16#8007	0	0	1	Invalid line / record separator. It should be CRLF.
W#16#8008	0	0	1	Invalid character. It should be only ASCII characters.
W#16#8009	0	0	1	Could not convert the field data. (e.g. syntax error ...)
W#16#800A	0	0	1	Too big file. Not enough system resource to read file.
W#16#822A	0	0	1	FILENAME parameter is invalid.
W#16#8xyz	0	0	1	General error code (S7 compatible)



The FAT16/32 Filesystem on their Micro-SD-Karte does not allow some signs in the file- or directory name

These are: \ / : * ? " < > |

These signs will be ignored.

System functions

Note:

- Field delimiter character is “;” (semicolon, ASCII code 59).
- Decimal point is “.” (point, ASCII code 46).
- Record separated by carriage return line feed character (CRLF, ASCII code 13, 10)
- Process data (DATA parameter) offset should start from even address
- One record data size in bytes should be even.
- In case of empty field (;), corresponding process data area is not written (no data).
- If a field in a record is smaller than in CSV_STRUCT “field count” assigned, converting is continued with the next record (like a line break) and data is written to next record data area.
- If there are more fields in a record than in CSV_STRUCT “field count” assigned, the surplus data fields are ignored
- If the process data area size (DATA) is smaller than the converted archive data size, converted data will be filled until reaching the assigned data size and the rest will be ignored.
- If the process data area size (DATA) is larger than converted archive data size, all converted data will be written by field by field and the rest of the data area size is not overwritten.

S7-Program sample to use the SFC213

Purpose: Read in and convert data from csv-file into DB213

```
CALL SFC 213
  REQ      :=M1501.7          // TRUE = Request to read CSV file
  FILENAME :=DB207.ARCHIVE_NAME_CSV // CSV archive name
  CSV_STRUCT :=DB207.ARCHIVE_CSV_FORMAT // CSV field format
  RET_VAL   :=MW1000          // Return / result code
  BUSY      :=M1701.7          // TRUE = Request is in process
  DONE      :=M1002.0          // TRUE = Successfully read
  ERROR     :=M1002.1          // TRUE = Error occurred
  LINE      :=MW1006          // In error case, Line number which
                                // causes converting error
  FIELD     :=MW1008          // In error case, Field number which
                                // causes converting error
  DATA     :=DB213.ARCHIVE_DATA // Destination data area which data
                                // be written
U  M      1701.7              // Data read is in process
BEB
R  M      1501.7              // reset request signal

O  M      1002.0              // If successful done,
ON M      1002.1              // or no error, then
BEB              // end the function

L      MW 1000                // error occurred
// error evaluation
// ...
BEA              // end of function
```

System functions

Overview about SFCs for a File Management System

INSEVIS offers SFCs to handle files and directories on the Micro-SD-Karte just like a file management system: creating, copying, renaming, informing about and deleting files and directories there.



The directories and files are kept always in the folder UserData\ and all (hierarchic) levels below.

Creating of directories and files

Directory: SFC 223 „CREATE“ creates a directory
File: SFC 207 „ARCHIVE“ creates a file

Reading of directories and files

Directory: can not be read in
File: SFC 209 „READFILE“ writes back file content into PLC
 SFC 213 „READ_CSV“ writes back csv-file content into PLC

Copying of directories and files

Directory: can not be copied
File: SFC 222 „COPY“ copies a file

Renaming of directories and files

Directory: SFC 220 „RENAME“ renames a directory
File: SFC 220 „RENAME“ renames a file too

Informing about directories and files

Directory: SFC 224 „INFO“ shows all entries into an directory
File: SFC 224 „INFO“ informs about the file size

Informing about directories and files

Directory: SFC 224 „INFO“ shows all entries into an directory
File: SFC 224 „INFO“ informs about the file size

Informing about files

Directory: SFC 225 „FILEINFO“ shows all entries into an directory
File: SFC 225 „FILEINFO“ informs about the file name and size

Deleting of directories and files

Directory: SFC 221 „DELETE“ deletes an directory
File: SFC 221 „DELETE“ deletes a file too

System functions

Create folders on Micro-SD-Card with SFC 223 „CREATE“

SFC222 "CREATE" is used to create a folder on Micro-SD card referred to the folder UserData\ (it contains all (hierarchic) level below). This SFC is asynchronous, it means its processing extends over several SFC calls.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE: Assign and start job
DIRNAME	INPUT	STRING	Folder name to be created
RET_VAL	OUTPUT	WORD	Status/Error code
BUSY	OUTPUT	BOOL	TRUE: Job is in processing
DONE	OUTPUT	BOOL	TRUE: Job successfully done
ERROR	OUTPUT	BOOL	TRUE: Job failed

Status code meaning

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Job is successfully done
W#16#7000	1	0	0	Job is process
W#16#7001	0	0	0	Job resource is temporary locked

Status code meaning

RET_VAL	BUSY	DONE	ERROR	Description
W#16#7002	0	0	0	REQ Parameter is FALSE
W#16#7003	0	0	1	Job is not accepted, due to previous job is still in process
W#16#8000	0	0	1	File system is not ready (e.g SD card not inserted)
W#16#8001	0	0	1	Name is invalid (DIRNAME parameter)
W#16#8002	0	0	1	Name is too long (DIRNAME parameter)
W#16#8003	0	0	1	Entry with a name already exists (DIRNAME parameter)
W#16#8004	0	0	1	Could not create folder
W#16#822A	0	0	1	Invalid format in DIRNAME parameter
W#16#8xyy	0	0	1	General error code (S7 compatible)

S7-User program example to create a folder

```

CALL SFC 223
REQ      :=M1800.0
DIRNAME:=DB223.DIRNAME
RET_VAL:=MW1000
BUSY     :=M1800.1
DONE     :=M1800.2
ERROR    :=M1800.3

O      M   1800.2           // Job done
O      M   1800.3           // or failed
R      M   1800.0           // reset request signal

UN     M   1800.3
BEB                               // No error occurred

L      MW 1000               // error occurred
// error evaluation
// ...
BEA                               // end of function

```

System functions

Copy files on Micro-SD-Card with SFC 222 „COPY“

The SFC222 "COPY" is used to copy the file on Micro-SD card referred to the folder UserData\ (it contains all (hierarchic) level below). This SFC is asynchronous, it means its processing extends over several SFC calls.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE: Assign and start job
OVERWR	INPUT	BOOL	TRUE: Copy the file even DSTNAME exists
SRCNAME	INPUT	STRING	Name of the file to be copied
DSTNAME	INPUT	STRING	Destination name (path)
RET_VAL	OUTPUT	WORD	Status/Error code
BUSY	OUTPUT	BOOL	TRUE: Job is in processing
DONE	OUTPUT	BOOL	TRUE: Job successfully done
ERROR	OUTPUT	BOOL	TRUE: Job failed

Status code meaning

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Job is successfully done
W#16#7000	1	0	0	Job is process
W#16#7001	0	0	0	Job resource is temporary locked
W#16#7002	0	0	0	REQ Parameter is FALSE
W#16#7003	0	0	1	Job is not accepted, due to previous job is still in process
W#16#8000	0	0	1	File system is not ready (e.g SD card not inserted)
W#16#8001	0	0	1	File with destination name already exists (DSTNAME parameter)
W#16#8002	0	0	1	File with source name is not found (SRCNAME parameter)
W#16#8003	0	0	1	Name is too long (SRCNAME, DSTNAME parameter)
W#16#8004	0	0	1	Name is invalid (SRCNAME, DSTNAME parameter)
W#16#8005	0	0	1	Not a file (SRCNAME parameter)
W#16#8006	0	0	1	Could not copy the file
W#16#832A	0	0	1	Invalid format in SRCNAME parameter
W#16#842A	0	0	1	Invalid format in DSTNAME parameter
W#16#8xyy	0	0	1	General error code (S7 compatible)

S7-User program example to copy a file

```

CALL SFC 222
REQ      :=M1800.0
OVERWR   :=TRUE                // overwrite the file if exists
SRCNAME  :=DB222.SRCNAME
DSTNAME  :=DB222.DSTNAME
RET_VAL  :=MW1000
BUSY     :=M1800.1
DONE     :=M1800.2
ERROR    :=M1800.3

O        M    1800.2            // Job done
O        M    1800.3            // or failed
R        M    1800.0            // reset request signal

UN       M    1800.3
BEB                                // No error occurred

L        MW 1000                // error occurred
// error evaluation
// ...
BEA                                // end of function

```

System functions

Rename files / folders on Micro-SD-Card with SFC 220 „RENAME“

SFC220 "RENAME" used to rename the file / folder on Micro-SD card referred to the folder UserData\ (it contains all (hierarchic) level below). This SFC is asynchronous, it means its processing extends over several SFC calls.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE: Assign and start job
OLDNAME	INPUT	STRING	Name of the file / folder to be renamed
NEWNAME	INPUT	STRING	A new name
RET_VAL	OUTPUT	WORD	Status/Error code
BUSY	OUTPUT	BOOL	TRUE: Job is in processing
DONE	OUTPUT	BOOL	TRUE: Job successfully done
ERROR	OUTPUT	BOOL	TRUE: Job failed

Status code meaning

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Job is successfully done
W#16#7000	1	0	0	Job is process
W#16#7001	0	0	0	Job resource is temporary locked
W#16#7002	0	0	0	REQ Parameter is FALSE
W#16#7003	0	0	1	Job is not accepted, due to previous job is still in process
W#16#8000	0	0	1	File system is not ready (e.g SD card not inserted)
W#16#8001	0	0	1	File / folder with a new name already exists (NEWNAME parameter)
W#16#8002	0	0	1	File / folder with a old name is not found (OLDNAME parameter)
W#16#8003	0	0	1	Name is too long (OLDNAME, NEWNAME parameter)
W#16#8004	0	0	1	Name is invalid (OLDNAME, NEWNAME parameter)
W#16#8005	0	0	1	Could not rename a file / folder
W#16#822A	0	0	1	Invalid format in OLDNAME parameter
W#16#832A	0	0	1	Invalid format in NEWNAME parameter
W#16#8xyy	0	0	1	General error code (S7 compatible)

S7-User program example to rename a file

```

CALL SFC 220
REQ      :=M1800.0
OLDNAME:=DB220.OLDNAME
NEWNAME:=DB220.NEWNAME
RET_VAL:=MW1000           // Status code
BUSY     :=M1800.1
DONE     :=M1800.2
ERROR    :=M1800.3

O      M      1800.2           // Job done
O      M      1800.3           // or failed
R      M      1800.0           // reset request signal

UN     M      1800.3
BEB                               // No error occurred

L      MW 1000                 // error occurred
// error evaluation
// ...

BEA                               // end of function

```

System functions

Inform about files / folders on Micro-SD-Card with SFC 224 „INFO“

SFC224 "INFO" is used to query the file size or number entries in the folder on Micro-SD card referred to the folder UserData\ (it contains all (hierarchic) level below). This SFC is asynchronous, it means its processing extends over several SFC calls.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE: Assign and start job
DIRECTORY	INPUT	BOOL	TRUE: Query folder
ENTRYNAME	INPUT	STRING	Name of the file / folder
RET_VAL	OUTPUT	WORD	Status/Error code
BUSY	OUTPUT	BOOL	TRUE: Job is in processing
DONE	OUTPUT	BOOL	TRUE: Job successfully done
ERROR	OUTPUT	BOOL	TRUE: Job failed
SIZE	OUTPUT	DWORD	File size / Number of entries in the folder

Status code meaning

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Job is successfully done
W#16#7000	1	0	0	Job is process
W#16#7001	0	0	0	Job resource is temporary locked
W#16#7002	0	0	0	REQ Parameter is FALSE
W#16#7003	0	0	1	Job is not accepted, due to previous job is still in process
W#16#8000	0	0	1	File system is not ready (e.g SD card not inserted)
W#16#8001	0	0	1	Name is invalid (ENTRYNAME parameter)
W#16#8002	0	0	1	Name is too long (ENTRYNAME parameter)
W#16#8003	0	0	1	File / folder is not found
W#16#8004	0	0	1	Could not read information of entry
W#16#832A	0	0	1	Invalid format in ENTRYNAME parameter
W#16#8xyy	0	0	1	General error code (S7 compatible)

S7-User program get the file size

```

CALL SFC 224
  REQ      :=M1800.0
  DIRECTORY:=FALSE           // Get file size
  ENTRYNAME:=DB224.FILENAME
  RET_VAL  :=MW1000
  BUSY     :=M1800.1
  DONE     :=M1800.2
  ERROR    :=M1800.3
  SIZE     :=MD1004         // file size in bytes

O      M      1800.2         // Job done
O      M      1800.3         // or failed
R      M      1800.0         // reset request signal

UN     M      1800.3
BEB                                         // No error occurred

L      MW 1000                     // error occurred
// error evaluation
// ...
BEA                                         // end of function

```

System functions

Inform about file size and name on Micro-SD-Card with SFC 225 „FILEINFO“

SFC225 "FILEINFO" is used to query the file name and file size by zero based index in the folder on Micro-SD card referred to the folder UserData\. This SFC is asynchronous, it means its processing extends over several SFC calls.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE: Assign parameter and start job
DIRECTORY	INPUT	STRING	(Sub-) Directory name to search the file
INDEX	INPUT	WORD	Zero based file index in the DIRECTORY
RET_VAL	OUTPUT	WORD	Status/Error code
BUSY	OUTPUT	BOOL	TRUE: Job is in processing
DONE	OUTPUT	BOOL	TRUE: Job successfully done
ERROR	OUTPUT	BOOL	TRUE: Job failed
FILE_NAME	OUTPUT	STRING	Name of the file
FILE_SIZE	OUTPUT	DWORD	File size

Status code meaning

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Job is successfully done
W#16#7000	1	0	0	Job is process
W#16#7001	0	0	0	Job resource is temporary locked
W#16#7002	0	0	0	REQ Parameter is FALSE
W#16#7003	0	0	1	Job is not accepted, due to previous job is still in process
W#16#8000	0	0	1	File system is not ready (e.g Micro-SD card not inserted)
W#16#8001	0	0	1	Could not open directory (e.g directory does not exist)
W#16#8002	0	0	1	INDEX parameter is greater than or equal to number of available files
W#16#8003	0	0	1	Could not read directory
W#16#812A	0	0	1	Invalid format in DIRECTORY parameter
W#16#882A	0	0	1	Invalid format in FILE_NAME parameter
W#16#8xyy	0	0	1	General error code (S7 compatible)

S7-User program get the file info

```

UN    M    1900.0
BEB                                     // Job is not requested

CALL  SFC   225
REQ    :=M1900.0
DIRECTORY:=DB225.DIRECTORY           // Directory to be searched in
INDEX   :=MW1902                     // File index in directory (zero based)
RET_VAL :=MW1000
BUSY    :=M1900.1
DONE    :=M1900.2
ERROR   :=M1900.3
FILE_NAME:=DB225.FILENAME           // file name
FILE_SIZE:=MD1904                   // file size in bytes

O      M    1900.2                   // Job done
O      M    1900.3                   // or failed
R      M    1900.0                   // reset request signal

UN    M    1900.3
BEB                                     // No error occurred

L      MW    1000                     // error occurred
// error evaluation
// ...

```


System functions

Delete files / folders on Micro-SD-Card with SFC 221 „DELETE“

SFC221 "DELETE" used to delete the file / folder on Micro-SD card referred to the folder UserData\ (it contains all (hierarchic) level below). This SFC is asynchronous, it means its processing extends over several SFC calls.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE: Assign and start job
FILENAME	INPUT	STRING	Name of the file / folder to be deleted
RET_VAL	OUTPUT	WORD	Status/Error code
BUSY	OUTPUT	BOOL	TRUE: Job is in processing
DONE	OUTPUT	BOOL	TRUE: Job successfully done
ERROR	OUTPUT	BOOL	TRUE: Job failed

Status code meaning

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Job is successfully done
W#16#7000	1	0	0	Job is process
W#16#7001	0	0	0	Job resource is temporary locked
W#16#7002	0	0	0	REQ Parameter is FALSE
W#16#7003	0	0	1	Job is not accepted, due to previous job is still in process
W#16#8000	0	0	1	File system is not ready (e.g SD card not inserted)
W#16#8001	0	0	1	File / folder is currently used
W#16#8002	0	0	1	File / folder is not found (FILENAME parameter)
W#16#8003	0	0	1	Name is too long (FILENAME parameter)
W#16#8004	0	0	1	Folder is not empty
W#16#8005	0	0	1	Could not delete a file / folder
W#16#822A	0	0	1	Invalid format in FILENAME parameter
W#16#8xyy	0	0	1	General error code (S7 compatible)

S7-User program example to delete a file

```

CALL SFC 221
REQ      :=M1800.0
FILENAME:=DB221.FILENAME
RET_VAL  :=MW1000
BUSY     :=M1800.1
DONE     :=M1800.2
ERROR    :=M1800.3

O   M   1800.2           // Job done
O   M   1800.3           // or failed
R   M   1800.0           // reset request signal

UN   M   1800.3
BEB                                     // No error occurred

L   MW 1000                                     // error occurred
// error evaluation
// ...

BEA                                     // end of function

```

System functions

Adjust LCD brightness with SFC 214 "LCD_BRIG"

This function is used to change LCD brightness of Panel-PLC.

This SFC is asynchronous, it means its processing extends over several SFC calls.

Only one job is queued at the time for processing.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	REQ = TRUE: Request to set/get LCD brightness
SET	INPUT	BOOL	TRUE= Set brightness, FALSE= Get brightness
LADDR	INPUT	WORD	Logical base address 0 Panel-PLC (intern LCD) 1...16 Panel-HMI connection-ID number
BRIGHTNESS	INPUT	INT	LCD brightness value in percent 0..100%
RET_VAL	OUTPUT	WORD	Status code
DONE	OUTPUT	BOOL	TRUE=Job is done
BUSY	OUTPUT	BOOL	TRUE=Job is in process
ERROR	OUTPUT	BOOL	TRUE=Job failed, error code in RETVAL
BACKLIGHT	OUTPUT	BOOL	TRUE= LCD backlight is ON, FALSE= LCD back light is OFF (by key with VisuStage -system function)
BRIGHTNESS_CUR	OUTPUT	INT	Current LCD brightness value

RET_VAL	BUSY	DONE	ERROR	Description
W#16#0000	0	1	0	Successful read
W#16#7000	1	0	0	Job is in process
W#16#7001	0	0	1	Connection is in invalid state (e.g not connected)
W#16#7002	0	0	1	Job not accepted, due to previous job is in still in process
W#16#7003	0	0	1	Connection resource occupied (temporarily locked)
W#16#7004	0	0	0	REQ parameter is FALSE, Job is not started
W#16#8001	0	0	1	Invalid connection ID number in LADDR parameter
W#16#8002	0	0	1	Visualization is not running (on Panel-HMI)
W#16#8004	0	0	1	Invalid LCD brightness value in BRIGHTNESS parameter.
W#16#8xyz	0	0	1	General error codes (compatible to STEP®7)

S7-Program sample to use the SFC214

Purpose: Change backlight brightness value by MW2024

```
CALL SFC 214
REQ      :=M1502.3
SET      :=TRUE           // TRUE= Change the LCD brightness
LADDR    :=MW2000         // VisuStage connection ID number.
                        // In Panel-PLC (e.g PC700P) must be W#16#0.
BRIGHTNESS :=MW2024       // LCD Brightness value to change
                        // (in present, 0..100)
RET_VAL   :=MW1000        // Result / return code
DONE      :=M1002.0       // TRUE = Successful created
BUSY      :=M1002.1       // TRUE = Request is in process
ERROR     :=M1002.2       // TRUE = Error occurred
BACKLIGHT :=M2026.0       // LCD Back light status
BRIGHTNESS_CUR :=MW2028   // Current LCD brightness value in present
```

System functions

```

O      M      1002.0          // successful done
O      M      1002.2          // or failed
R      M      1502.3          // reset request signal

UN     M      1002.2          // if not failed, then
BEB                                         // end the function

L      MW     1000 // error occurred
// error evaluation
// ...
BEA                                         // end of function

```

Copy RAM to ROM with SFC 254 "RAM2ROM"

INSEVIS PLC provides the SFC 254 "RAM2ROM" to copy the data blocks from work (battery buffered) memory to load memory (in the flash memory).

- Note:**
- This SFC is asynchronous, it means its processing extends over several SFC calls.
 - This function should not interrupted (e.g., Power OFF, or PG memory related functions, such as Clear all, Block delete, ...).
 - Data blocks which marked as "Non-Retain", also copied (actual data) to load memory.

Sample long time storage of the PLC with actual process data:

Use SFC254 before switching off and when battery is empty and the RAM-content (process data) will be lost, all will be red from flash after re-starting the PLC. Only time/date data has to be entered new.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE = Start the processing
RET_VAL	OUTPUT	WORD	Status code W#16#0000 Copying is in process, if BUSY=TRUE Copying is done, if BUSY=FALSE W#16#7000 No copying in process, if REQ = FALSE W#16#8001 Copy is not allowed, other memory manipulating operation is active.
BUSY	OUTPUT	BOOL	TRUE = Copying is in process

S7-program sample for using the SFC254

Purpose: Copy all data blocks to load memory. Actual values are overwritten in load memory.

UN	M	1502.2	// If request is not set, then
BEB			// end the function
CALL	SFC	254	
REQ	:	=M1502.2	// TRUE = Request to start RAM 2 ROM
RET_VAL	:	=MW1000	// Return / result code
BUSY	:	=M1702.2	// TRUE = Request is in process
U	M	1702.2	// If Copy request is in process, then
BEB			// end the function
R	M	1502.2	// reset the request signal
L	MW	1000	// return / result code
L		0	// no error, then
==I			
BEB			// end the function
L	MW	1000	// error occurred
			// error evaluation
			// ...
BEA			// end the function

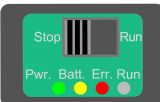
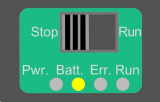

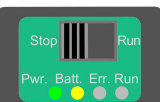




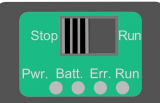
System functions

Update operation system in the PLC

It is possible to update the PLC's operation system, **without having any influence on the S7-user program** kept in the PLC.

The operation is described below; you need a Micro-SD-Card with **operation system binary data in the ROOT folder** (e.g. PC35Vx.bin, PC57Px.bin, etc), your PLC, and its power supply 24V

→ Devices with CPU-T can get its new firmware via Ethernet by ServiceStage software too.

Procedure of OS-update		Display in LCD (if LCD is available)
Preparation - switch Run/Stop into STOP - switch device power OFF - plug in the micro-SD-card - switch device power ON	1)  Stop Power off SD-Card in Power on	
Internal check <i>(depends on existing software version)</i> LED Batt. (yellow) blinks while internal CPU check runs	2)  <i>internes Prüfen</i> <i>internal check</i> <i>abwechselnd</i> <i>alternatively</i>	 Checking ...
Loading process in operation <i>(Do not switch off the device, no interrupt in power supply!!)</i> „Power“-LED shines permanently LEDs „Batt.“ (yellow) and „Run“ (green) blink <u>alternatively</u>	3)  <i>Ladevorgang...</i> <i>Loading process</i> <i>(ca. 1min)</i> <i>abwechselnd</i> <i>alternatively</i>	a)  Updating ... b)  Checking ...
Loading process finished „Power“-LED shines permanently LEDs „Batt.“ (yellow) and „Run“ (green) blink <u>together</u>	4)  <i>Laden beendet</i> <i>Loading finished</i> <i>gleichzeitig</i> <i>synchronously</i>	 Update done
initialize the new OS - switch device power OFF - pull out the micro-SD-card - switch device power ON	5)  Power off SD-Card out Power on Run	

General reset by hardware

General resetting is possible in 2 ways:

- by software (with the SIMATIC®-Manager/TIA from Siemens or by ServiceStage (belongs to basic knowledge and is not explained again)
- by hardware (Run/Stop- switch) directly on the PLC) explained hereby

After a hardware reset only IP-address and time remain in the battery buffered memory (like at Siemens).

Procedure for general reset by hardware:

Run/Stop-switch into Stop
switch device power OFF
switch device power ON

During the first 3 seconds*, when the LEDs „Power“, „Batt.“, „Error“ glow,

- 1 Run/Stop-switch into Run
red „Error“-LED now is off
green „Run“-LED now glows

Within the next second

- 2 Run/Stop-switch into Stop
red „Error“-LED now glows
green „Run“-LED now is off

Within the next second

- 3 Run/Stop-switch in Run
red „Error“-LED now is off
green „Run“-LED now glows

Within the next second

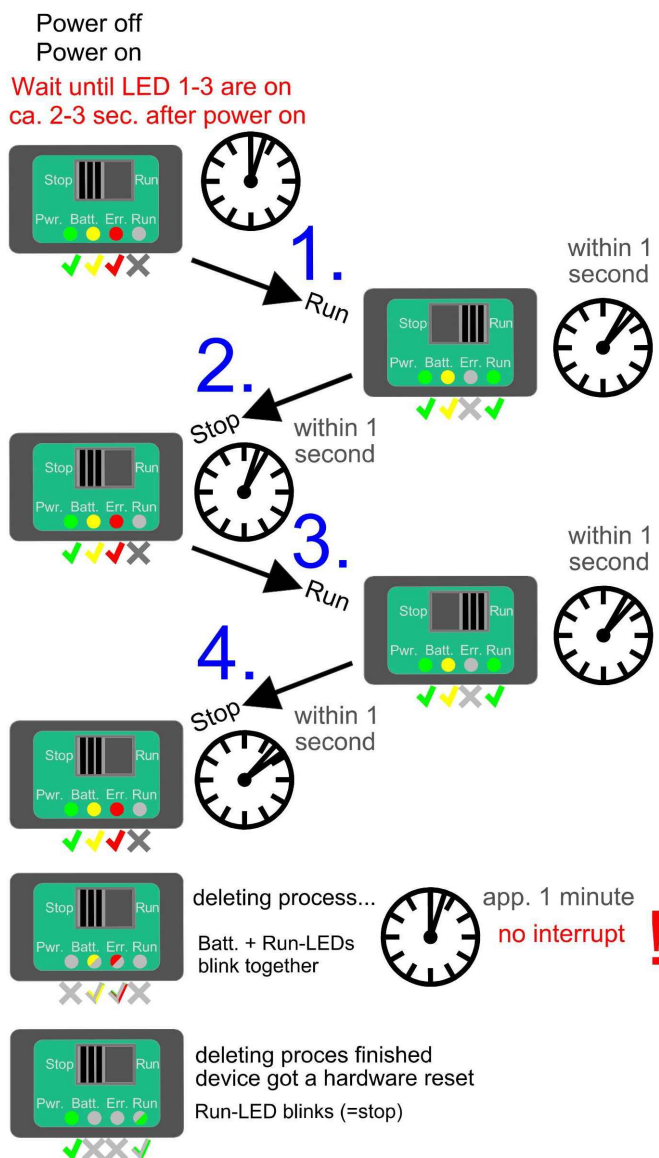
- 4 Run/Stop-switch into Stop
red „Error“-LED now glows
green „Run“-LED now is off

Deleting in operation

DO NOT power off device
LEDs „Batt.“ (yellow) and
„Err.“ (red) blink together

Deleting finished

„Run“-LED blinks
(device is in Stop mode)



Hint for RESTORE after Hardware-Reset:

After this operation the PLC tries to read data from the micro-SD-card. RUN- and BATT-LEDs are blinking together during this time.

If there is no or an empty micro-SD-card plugged in, the user data and the process data will be written over with „0“ (general reset).

During the first 3 seconds* is a value for the CPU-V/-P. The larger CPU-T needs longer for booting process, than the red LEDs start after the progress bar has finished (if you have a display, ~ 6-8 seconds), please be patient – you will catch this moment surely.

Remote access with the software „RemoteStage“

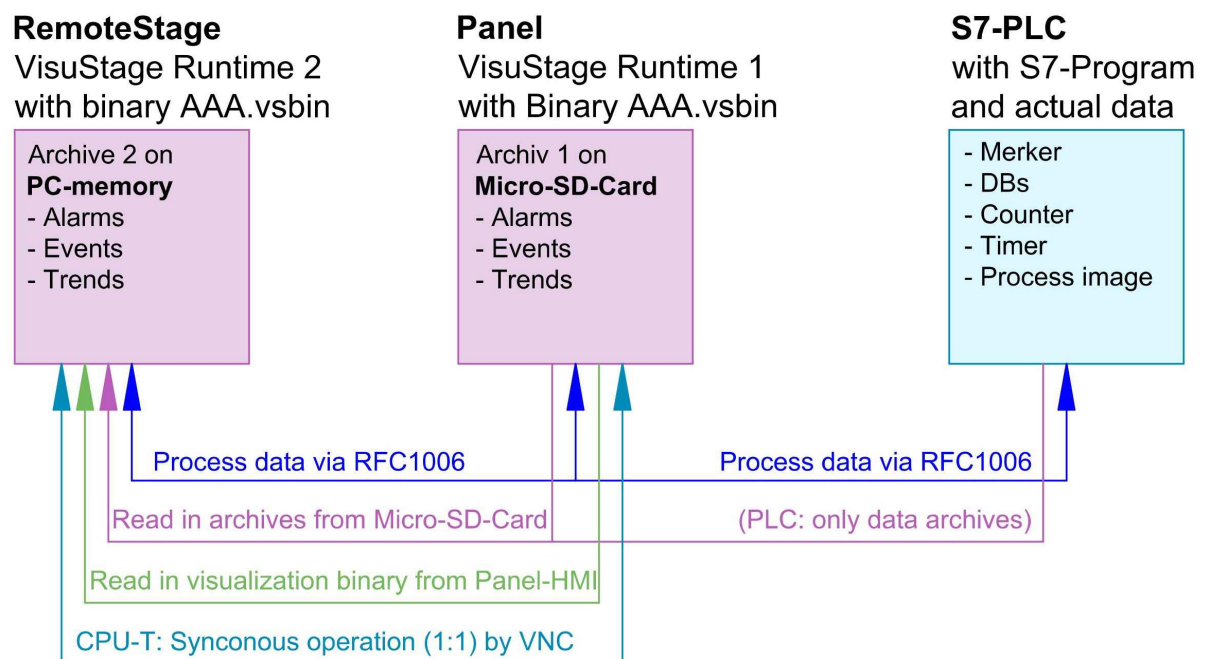
General

The RemoteStage is a multiple instanciable command line software tool, what is not to install. It can be kept in a folder on the remote-PC or on a mobile data carrier as **remotestage.exe** (and if you use German program language with the file **remotestage.deu** as well). As command line tool it can proceed diverse functions in a batch file.

The software RemoteStage uses the binary files (binaries / *.vsbin) of visualization projects, made with „VisuStage“. These binaries are not backreadable, what offers an effective protection of the users know-how and allows to forward these files towards the final user or operator. It is possible to download the visualization binary directly by Ethernet from the Panel-PLC / Panel-HMI to reduce the organization efforts. A PIN-code protects the visualization against unauthorized uploads.

The RemoteStage works **like an additional Panel-HMI** on the remote-PC, what gets the process (actual) data by a TCP/IP-connection straight from the PLC. System data like time and buzzer are related to even this PC of course. This allows a free remote visualization and account to external S7-PLCs by S7-Ethernet (Put/Get).

In a 2nd program instance (Archive mode) the RemoteStage reads in (direct form Micro-SD-card or by Ethernet) from the Micro-SD-card, displays the values, converts into csv-format and saves it to the remote PC.



If the RemoteStage should be installed as an batch call on the end user's PC this is all you need to write:

```
[path of RemoteStage]\remotestage.exe /r=[IP-address of the remote-PLC in the same sub net]
/V=[path of visualization binary]\complete file name]
/start
```

Selection of the remote-device

With opening of the visualization binary („visualization“.vsbin) the IP-address of the remote PLC will be required (Fig.). (By using the function „accessible devices“ - loupe- you can select one of these directly or type in the IP-address manually.)

This address will be linked with this binary.
So it is not necessary to re-type it again.

The dialog box titled "Remote address" contains the following fields and controls:

- A text input field for the remote PLC address with a search icon.
- A section for "Remote TCP port" with a checkbox "Use different TCP port number as default".
- A "TCP port number" field with a value of 102 and a spinner control.
- A note: "Note: Default TCP port number is 102."
- Buttons for "Ok" (with a checkmark icon) and "Cancel" (with an X icon).

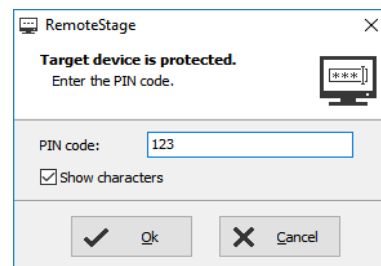
Remote access with the software „RemoteStage“

Insert PIN code

In the VisuStage-project it is possible to allow an upload of the compiled binary to the remote PC. This upload can be protected by a PIN code, what is to be inserted in a new window (right)

This PIN must be inserted at every new upload (also when you use this upload for backup function in ServiceStage). At INSEVIS sample projects this PIN is 123.

After successful upload a black window appears, what changes to the first menu in the VisuStage project after clicking the “connect” button. (Not 1:1 of the original panel, this will be done by VNC)



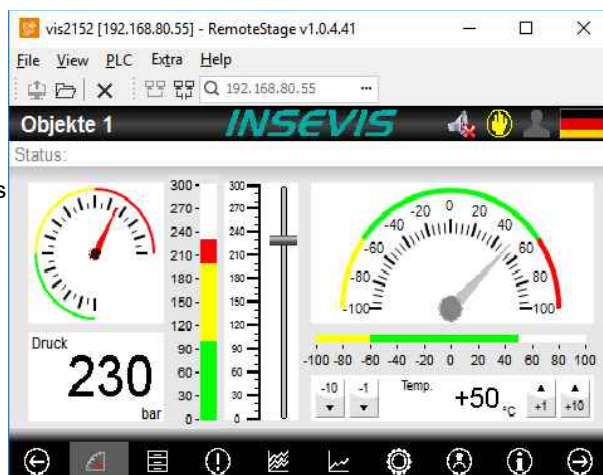
Using and setting up the remote screen

The PopUp-menus are self explaining.

The actual remote state is shown on the bottom of the screen.

The switching between online/offline can be done by the green button beside the IP-address line or by the menu „PLC“.

Hint: Referring the system data (like buzzer, archives and time) always the PC-system data will be used, not these of the remote device.



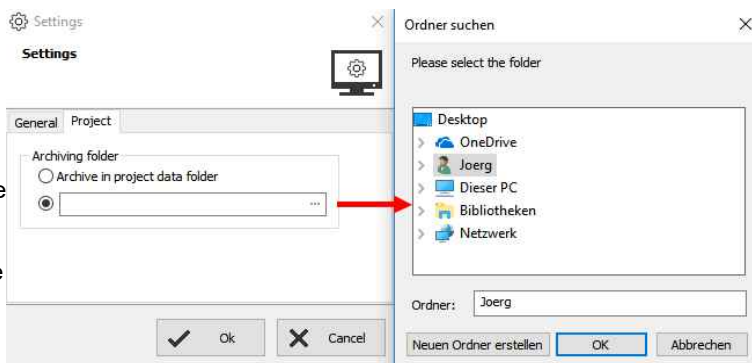
Storing the remote (on the PC) archived data

The archive data, logged during the remote session on the PC, will be stored on the working place in the remote-PC **only**. (not depending from the archive in the PLC in the Micro-SD-card during the same time).

Note: there are 2 archives of each item. If you want to archive a remote-trend, you need to start this trend in the remote visualization.

At the menu „Settings“ will be set up, if these date will be stored

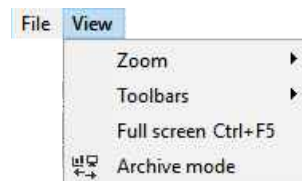
- in the project folder (where the visu binary is kept) or
- at another path.



Hint: If you upload the visualization binary from the remote device (this function will be activated in the VisuStage-project), this file will be kept in the local TEMP-directory of your PC as long you are remote online. There will be stored the archive files as binary too as long the remote session is active. To save and convert these archives into CSV-format please go to archive mode and select “File Open” and select the TEMP-directory.

Most of the archive data are kept on the Micro-SD-card as binaries and can be converted to csv-format by RemoteStage “Archive mode” (a 2nd instance of RemoteStage) only.

Open this 2nd instance in the “View”.menu (see right).



ATTENTION: Use only the binary file what fits to the archives (where these archives were projected)

Remote access with the software „RemoteStage“

Uploading, converting and storing archive data as csv-files

These data can be archived by these products on the Micro-SD-card:

Messages (alarms, events)	by all Panel-PLCs and Panel-HMIs → by activating the archive function „archiving“ in the VisuStage-project
Trends (with all channels)	by all Panel-PLCs and Panel-HMIs → by activating the archive function „archiving“ in the VisuStage-project
Recipes (with all records and elements)	by all Panel-PLCs and Panel-HMIs → will always be stored / archived on the Micro-SD-card
Archiv -(data blocks) (as bin or csv)	by all Panel-PLCs and Panel-HMIs → will be configured by SFCs in the S7-program

In the newly appeared PC-window you may find different groups of icons with different functions:



Reads archive by inserting Micro-SD-card in PC slot from left → right:

- Data(DB-)archive
- Alarm archive
- Event archive
- Trend archive

Read in archives by Ethernet directly into PC from left → right:

- All archives
- Alarm archive
- Event archive
- Recipe archive
- Trend archive
- Data(DB-)archive

Exports archive data to PC into the PC-network from left → right:

- Alarm archive
- Event archive
- Recipe archive
- Trend archive
- Data(DB-)archive

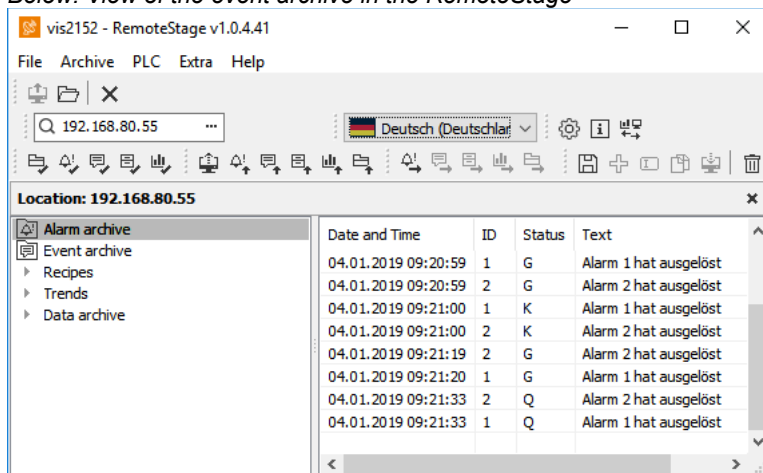
Editing and downloading of modified archives from left → right:

- Save archive
- Add new archive
- Rename archive
- Copy archive
- Download archive

Display and export event and alarm archives

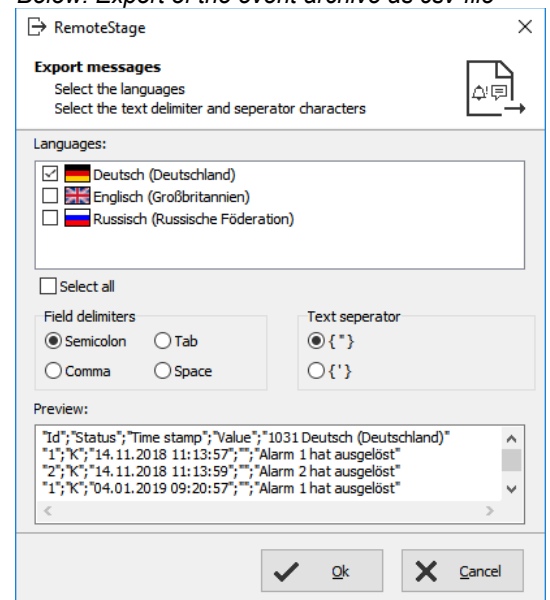
Archive data will be read in from Micro-SD-card, can be displayed and e.g. manually saved as csv-file.

Below: View of the event archive in the RemoteStage



(Displays all archived alarm messages of the Panel-PLC with the IP-address 192.168.80.55)

Below: Export of the event archive as csv-file



ATTENTION:

Do not confuse the uploaded (from Micro-SD-card) archive file with the other archive file created during the remote-session in the remote-PC itself. Pay attention to format the csv-file according to your own needs.

Remote access with the software „RemoteStage“

Display and export trend archives

Archive data will be read in from Micro-SD-card, can be displayed and e.g manually saved as csv-file.

Figure right displays all archived trend with name „Druck“ of the Panel-PLC with the IP-address 192.168.80.55.

The time will be displayed in the language format you assigned on your PC (here in German = DE)



ATTENTION:

Do not confuse the uploaded (from Micro-SD-card) archive file with the other archive file created during the remote-session in the remote-PC itself. Pay attention to format the csv-file according to your own needs.

Remote access with the software „RemoteStage“

Recipe handling

Recipe archive data from Micro-SD-card can be

- read in / uploaded (→ You have to select the whole root-folder “recipes” of the Micro-SD-card !),
- displayed and maybe edited in RemoteStage directly what is self explaining and

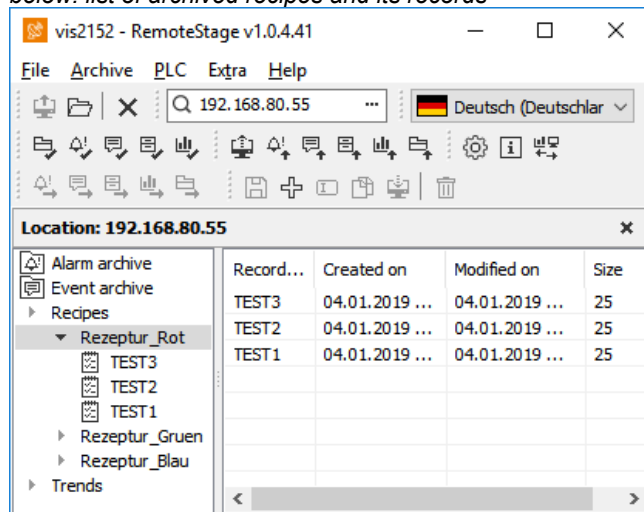
for FURTHER EDITING stored as csv-file at the PC

- converted into csv and stored, edited and written back / downloaded into the remote PLC or

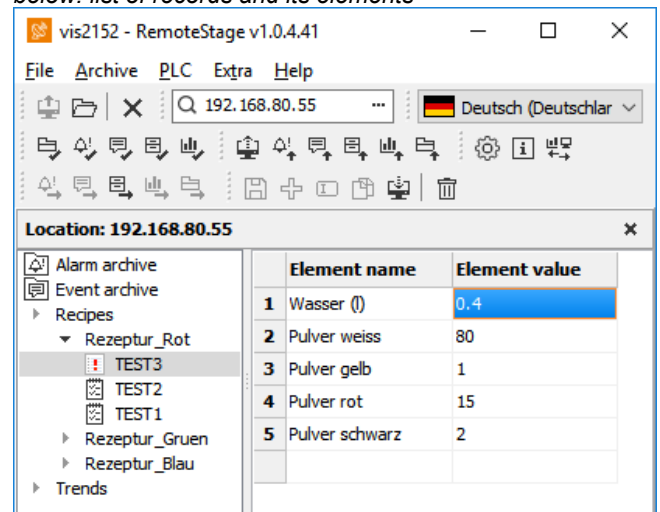
for COPYING THE WHOLE RECIPE SYSTEM saved as binary at the PC

- stores the whole recipe tree uneditable as binary, to be opened again and downloaded to another remote device (You have to select the whole root-folder “recipes” in the PC!)

below: list of archived recipes and its records



below: list of records and its elements



The record „TEST3“ was modified in the RemoteStage and this is shown by an „!“ . After downloading into the remote device this sign disappears.

The storing of all records (as binary files in an own folder into the network drive for modifying) is done, when “recipes” are marked in the tree view (left).

When recipes are edited in a network drive you can see their path in the address line.

When recipes will be uploaded from the remote device you see its IP-address.

The creating of a record can be done when in the tree view (left) a single recipe is marked.



ATTENTION:

The structure of the recipes MUST be kept the same.
Records of a recipe can be added or deleted within this recipe.

Remote access with the software „RemoteStage“

Data (DB) archive upload, display, storing and download

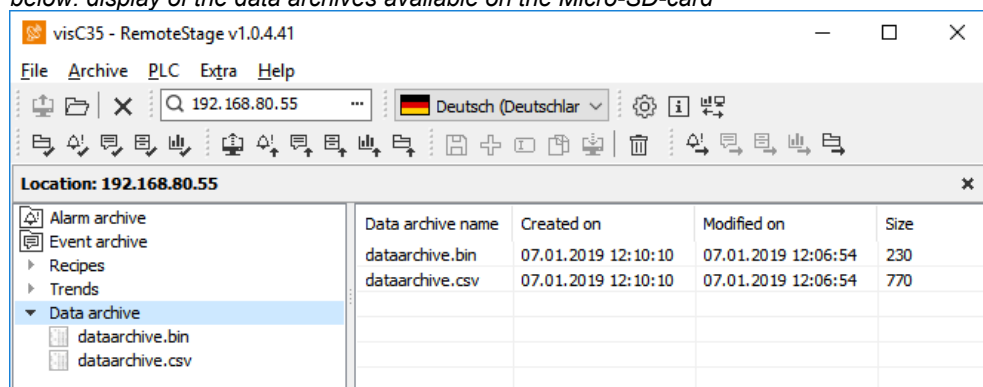
Any data can be archived with the **SFC 207 ARCHIVE** onto the Micro-SD-card (binary or csv-data) and read back with the **SFC 209 READFILE** (see system functions).

Data archive display and storing

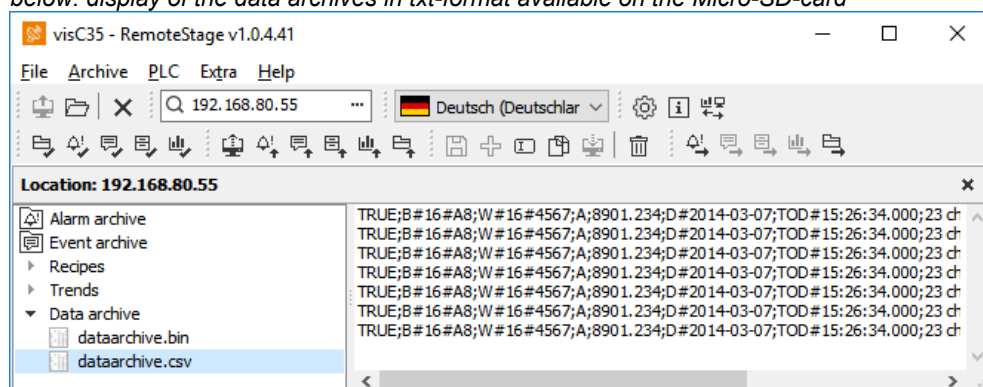
These data (DB) archived over S7-program to Micro-SD-card, can be read in and displayed and stored to the PC in the assigned format.

The format of the csv-file will be assigned by the S7-program (SFC207) and is not configurable by the RemoteStage. This can be done later on with any table calculation program by customer (because only he knows the format).

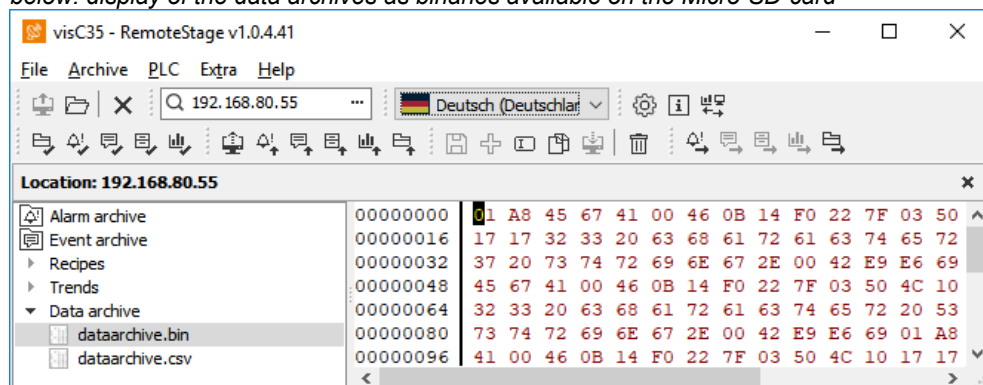
below: display of the data archives available on the Micro-SD-card



below: display of the data archives in txt-format available on the Micro-SD-card



below: display of the data archives as binaries available on the Micro-SD-card



Remote access with the software „RemoteStage“

Command line of RemoteStage (Viewing mode)

RemoteStage can be inserted in „planned tasks“ of the remote PC. A combination of multiple parameters is possible.

Here are available these special functions.

Automatic connecting to remote panel after starting PC

With his function a binary of the visualization is called and linked with the predefined IP-address of the remote panel (if this is available online only)

Command line

remotestage.exe /start /r=<remote> /v=<project binary>

Hint: If the binary name contains space-signs it is necessary to write it in “ like „C:\Test 1 2 3.vsbin“)

Automatic full screen after starting PC

With his function a binary of the visualization is called and the RemoteStage changes into full screen at the remote PC after automatic starting (Screen stays dark, because no IP-address was assigned here...)

Command line

remotestage.exe /start /full /r=<remote> /v=<project binary>

Hint: Use STRG+F5 to switch between full-/ part screen

Set up TCP port number for S7-communication

If there are multiple PLCs to connect by a web enabled router/ gateway with port forwarding service

In this case the RemoteStage starts a communication with a remote device with the IP address 192.168.80.50 via the TCP port with the number 4500.

Command line

remotestage.exe /v=d:\visu\process.vsbin /r=192.168.80.50 /p=4500

Hint:

The default port number for TCP is port 102 at the PLCs.

Remote access with the software „RemoteStage“

Command line of RemoteStage (Archive mode)

RemoteStage can be used in batch process from command line with command line parameters, to save archives somewhere with a self defined name.

In the PC-OS Windows® batch files can be done **time controlled** in the task manager

Start menu → All Programs → Accessories → System programs → Planned tasks → add a task

In the wizard dialog can be assigned date and time for that job.

Read in (upload) message- and trend archives from remote device

The binaries of the messages and trends (assigned in the VisuStage before and created at runtime) will be read in, converted and stored in a path on the PC.

Command line

remotestage.exe /mode=<mode> /r=<remote> /o=<output file> /v=<project binary> /t=<trend number>

Parameters

- mode:** Archive file read in
alarm_archive - read in alarm archive from remote device (alarm.bin)
event_archive - read in event archive from remote device (event.bin)
trend_archive - read in trend archive from remote device (trend_<trend number>.bin)
 (trend ID is assigned automatically by VisuStage at resources / grayed out you can not change it)
- remote:** Remote device IP address
- output file:** File name of archive file to be stored in PC.
 This is optional parameter. If this parameter is not assigned, default archive name is used (e.g alarm.bin) and stored in same folder where remotestage.exe located.
 If file name includes space character, it should be quoted with " (quotation mark).
 Archive converted to CSV format if file name has .csv or .txt extension, otherwise it writes as binary format.
- project binary:** Visualization project binary file name
- trend number:** Trend ID-number which is automatically assigned in VisuStage.
 It is only used at "trend_archive" mode.

Samples

For example to read in alarm archive from INSEVIS PLC with IP address 192.168.80.50 and write to „d:\archive\alarm.csv“ file with CSV format you need this command

Command line

**C:\>remotestage.exe /mode=alarm_archive /r=192.168.80.50
 /o=d:\archive\alarm.csv /v=d:\visu\myvisu.vsb**

For example to read in the sample-trend with ID6163 (from sample visualization of PC1560T) from Panel-PLC with IP-address 192.168.80.50 and to save it as csv-file to D:\archive\trend6163.csv you need this command

Command line

**C:\>remotestage.exe /mode=trend_archive /r=192.168.80.50
 /o=D:\archive\trend6163.csv /t=6163 /v=D:\PC156xT_2017_03.vsb**

Remote access with the software „RemoteStage“

Read in (upload) recipe archive from remote device

The binaries of the (in VisuStage predefined) recipes and the of the records (created in runtime by the operator or by SFC206 from the S7-program) will be read in, converted and stored in a path on the PC.

Command line

```
remotestage.exe /mode=recipe /r=<remote> /v=<project binary> /recipe=<recipe name>
/record=<record name> /o=<output file> /format=csv
```

Parameters

mode	Recipe = read in the recipe data from remote-device
remote	Remote device IP address
project binary	Visualization binary name
recipe name	Recipe name assigned in visualization project (NOT: View name)
record name	<p>Name of the record for saving of SPECIAL (single) records.</p> <p>If this parameter is not assigned, RemoteStage reads in ALL records, what belong to the recipe <i><recipe name></i> and stores it in a folder assigned by the parameter <i>"/o="</i> before.</p> <p>→ To configure the format of the output file as csv, the parameter <i>"/format=csv"</i> must be set, otherwise all data will be stored as a binary.</p> <p>In the csv-files this format is predefined: ; (Semicolon) as field limiter and " (quotation mark) as text separator.</p> <p>If this parameter is assigned, the format of the output file MUST BE predefined by the extension of the record name.</p> <p>→ If the file has an extension .csv or .txt, than it is a csv-format, otherwise the output file format is binary.</p>
output file:	<p>Path and file name of the recipe record for the storing in the PC (optional parameter)</p> <p>If the parameter <i>"/o="</i> is not assigned, a folder <i>"Recipes\<recipe name>"</i> will be created in the project folder (where the visualization binary is kept) automatically and the recipe record stored as <i><record name>.rec</i>.</p> <p>If file name includes space character, it should be quoted with " (quotation mark).</p>

Samples

To read in e.g. a single recipe record „Orange“ from the recipe „Rezeptur_Rot“ from the INSEVIS Panel-PLC with the IP-address 192.168.80.50 and to store it as csv-file to „D:\archive\“ this command line is required:

Command line

```
C:\>remotestage.exe /mode=recipe /r=192.168.80.50 /v=d:\visu\myvisu.vsbm
/recipe=Rezeptur_Rot /record=Orange /o=D:\archive\Orange.csv
```

To read in e.g. ALL recipe records from the recipe „Rezeptur_Rot“ from the INSEVIS Panel-PLC with the IP-address 192.168.80.50 and to store it as csv-file to „D:\archive\“ this command line is required:

Command line

```
C:\>remotestage.exe /mode=recipe /r=192.168.80.50 /v=d:\visu\myvisu.vsbm
/recipe=Rezeptur_Rot /o=D:\archive /format=csv
```

Remote access with the software „RemoteStage“

Write back (download) recipe archive into remote device

With this function the uploaded (and maybe modified) recipe data will be written back (downloaded) into the remote device. (from rev. 1.0.4.17 and PLC firmware 2.3.9 (all CPUs) or HMI firmware 1.2.7 CPU-V/-P and 1.2.6 CPU-T)

Attention: The structure of the receipes in the remote device may not be modified between upload and download. (No adding, deleting elements or changing the row of order)

Command line

```
remotestage.exe /mode=recipe_download /v=<project binary>
/recipe=<recipe name> /record=<record file(s)>
```

Parameters

project binary	Binary file of visualization project
recipe name	In VisuStage projected recipe name (ATTENTION: not view name!)
record file(s)	Recipe data to download into remote device

Samples

Following command line downloads ONE recipe record „Rot_123.rec“ of the recipe „Recipe_Rot“ from the folder D:\recipes into the remote device with the IP address 192.168.80.50

Command line

```
remotestage.exe /mode=recipe_download /v=d:\visu\process.vsbm
/recipe=Recipe_Rot /record=d:\recipes\Rot_123.rec /r=192.168.80.50
```

Following command line downloads ALL recipe record of the recipe „Recipe_Rot“ from the folder D:\recipes into the remote device with the IP address 192.168.80.50

Command line

```
remotestage.exe /mode=recipe_download /v=d:\visu\process.vsbm
/recipe=Recipe_Rot /record=d:\recipes\*.rec /r=192.168.80.50
```

Remote access with the software „RemoteStage“

Read in (upload) data (DB-) archive from remote device

This function copies the data(DB-) archives created by S7-program with SFC207 to the requested target path into the PC. A modification of the csv-format is not possible.

Command line

```
remotestage.exe /mode=data_archive /r=<remote> /archive=<archive file> /o=<output file>
```

Parameters

mode	data_archive = read in the date (DB-) archive from remote-device
remote	Remote device IP address
archive file:	File name of the data (DB-) archive If there is no archive file assigned to this parameter, the RemoteStage stores all data (DB-) archives in a folder assigned by the parameter "/o=" before.
output file:	Path and file name of the data (DB-) archive for the storing in the PC (optional parameter) If nothing is assigned for the parameter "/o=", a folder "UserData" is created in the program folder (where the remotestage.exe is kept. If file name includes space character, it should be quoted with " (quotation mark).

To read in e.g. a data (DB-) archive „Test-DB“ from the INSEVIS Panel-PLC with the IP-address 192.168.80.50 and to store it as csv-file to „D:\archive\Testdaten.csv“ this command line is required:

Command line

```
C:\>remotestage.exe /mode=data_archive /r=192.168.80.50  
/archive=Test-DB /o=D:\archive\Testdaten.csv
```

Write back (download) data (DB-) archive into remote device

With this function the uploaded (and maybe modified) data (DB-) archives will be written back (downloaded) into the remote device. (from rev. 1.0.4.17 and PLC firmware 2.3.9 (all CPUs))

Attention: The programmer itself is responsible to keep the structure of the data (DB-) archives. If they will be modified, all referring reports must be adapted too!

Command line

```
remotestage.exe /mode=data_download /archive:=<archive file(s)>
```

Parameters

archive file(s) S7 User data archive file(s) to download to remote PLC

Samples

Following command line downloads ONE archive file "default_cfg.csv" from the folder D:\configs into the remote device with the IP address 192.168.80.50

```
remotestage.exe /mode=data_download /archive=d:\configs\default_cfg.csv /r=192.168.80.50
```

Following command line downloads ALL archive files from the folder D:\configs into the remote device with the IP address 192.168.80.50

```
remotestage.exe /mode=data_download /archive=d:\configs\*. * /r=192.168.80.50
```

Remote access with the software „RemoteStage“

Return codes

Return codes of remotestage.exe indicate the result of operation.

Return code	Meaning
0	Successfully read out and saved.
1	Invalid parameter or mandatory parameters are missing
2	Project binary file is invalid or not found
3	Remote IP address is invalid
4	Could not connect to remote device
5	SD card not inserted on remote device
6	Archive file not found on remote device
7	Remote device does not support archive file read service
8	Invalid archive data
9	Not enough system resource
11	Not enough system resource on remote device
14	Could not write to in give <output file>
15	Connection is broken

Sample of RemoteStage usage in batch process:

```
@remotestage /mode=alarm_archive /r=192.168.80.50 /o=d:\archive\alarm.csv /v=d:\visu\myvisu.vsbm
@echo.
@goto res%ERRORLEVEL%

:res4
@echo Could not connect to remote device
@goto :EOF

:res0
@echo Successful read out and saved
@goto :EOF
```

Working with the software „ServiceStage“

With the free-of-cost software „ServiceStage“ it is possible to recognize INSEVIS-CPUs online to check their state, backup/update programs, set a know-how protection, etc.

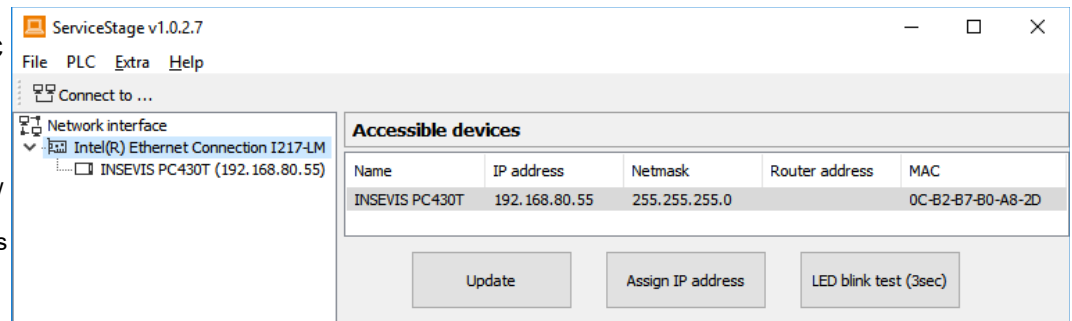


VIDEO-tutorials available

More information are provided in different video tutorials in the ServiceStage play list at INSEVIS YouTube®-channel “INSEVIS EN”.

With this program a member of the service-team can do all the main functions for diagnostics and update by his PC/Laptop without other expensive tools.

Therefore you choose the network adapter of your PC (This one, who makes the network access to the CPUs).
Select one of the CPUs displayed in the left window and press „Update“.
The „LED blink test“ helps to identify the selected CPU. (yellow battery-LED blinks for 3 seconds).



In this view you can assign a new IP-address of the connected device (Assign IP address).

Function overview

General information

Displays the system data to inform only (like serial number, firmware version, MAC-address).

Shows editable data like

- IP-address,
- Net mask and
- Router-IP-address

as well as a blink test button to identify your CPU again.

Only for CPU-T:

Firmwareupdate possible by Ethernet

Operating mode

Here you can change the operating mode of you CPU between RUN and STOP (not for Panel-HMI). This change must be confirmed in an extra window.

Date and time

For manual adjustment of system date and time.

By checking the box „Take from PC“ these data will be taken over from your PC after pressing „Apply“.

General information

Device name:

PC430T

Station name:

PC430T

Plant designation:

Location designation:

Serial number:

201162

Firmware version:

V2.6.1

PN/DP version:

IP address:

192.168.80.55

Netmask:

255.255.255.0

MAC address:

0C-B2-B7-B0-A8-2D

Firmware update

Assign IP address

LED blink test (3sec)

Operating mode

Current mode:

RUN

STOP

Mode switch position:

RUN

RUN

Date and Time

03.01.2019

17:12:59

☒ Take from PC

Apply

Working with the software „ServiceStage“

Memory

Shows the workload of the load- and work- memory (not for Panels-HMI).

The function „**Compress memory**“ should be used manually after multiple download of S7-programs to clear the memory.

The function „**Copy RAM to ROM**“ copies the battery buffered load memory content (like process data) to the onboard flash, to prevent a loss of data.

The function „**Clear Memory**“ deletes all user and process data in the CPU, only the firmware (operating system) remains.

Diagnostic buffer

It displays the last 100 entries of the diagnostic buffer listed by date and time (not at Panels-HMI).

By checking the function „**Information in hexadecimal format**“ it is shown in an hexadecimal system.

By pressing „**Update**“ new events will be displayed, what happened meanwhile.

With „**Save as**“ the displayed entries will be saved into a text file (*.txt).

Block Update

This function makes it possible to download

- the S7-user program as MemoryCard-file ***.wld** * (not for Panel-HMIs)
 - the INSEVIS-binary for visualization file as ***.vsbin** and
 - the INSEVIS-configuration file as ***.csbin** (not for Panel-HMIs)
- separately via the network into the CPU.

If the Hardware configuration block of the target device contains another IP-address than it is assigned to the configuration blocks of the blocks you do download into the device (WLD / VSBIN / CSBIN), it can happen, that you loose the connection after transfer. To avoid this, the ServiceStage from Rev. 1.0.1.9 asks, if the target-IP-configuration shall be replaced or not.

Memory

Load memory: Used in bytes: 6.768

Total size in bytes: 8.388.608

Work memory: Used in bytes: 4.268

Total size in bytes: 1.048.576

Diagnostic buffer

No.	Time	Date	Event
1	16:59:22.229	03.01.2019	Mode transition from STARTUP to RUN
2	16:59:22.169	03.01.2019	Request for automatic warm restart
3	16:59:22.166	03.01.2019	Mode transition from STOP to STARTUP
4	16:59:22.062	03.01.2019	Power on
5	14:01:46.031	03.01.2019	Power failure
6	13:03:01.228	03.01.2019	Mode transition from STARTUP to RUN
7	13:03:01.168	03.01.2019	Request for automatic warm restart
8	13:03:01.165	03.01.2019	Mode transition from STOP to STARTUP

Event ID: 16# **4302** 1 of 10

Mode transition from STARTUP to RUN
Previous operating mode: STARTUP
Requested operating mode: RUN

Hexadecimal: 43 02 FF 68 C7 00 00 00 08 10 77 10 19 01 03 16 59 22 22 95

☐ Information in hexadecimal format

Block

Download blocks:

ServiceStage v1.0.1.9

Do you want to keep current TCP/IP configuration of the remote device?



* The **WLD-file** is a binary file with S7-system data (SDBs) and S7-program blocks (OBs, FBs, FCs, DBs). After creating a WLD file in SimaticManager/TIA-Portal (at the menu: „File → Memory Card File → New“) you can copy DIFFERENT DBs with or without their system data into the window of the new S7-Memory-Card-File by drag'n drop.

So it is possible to download WHOLE S7-projects OR SINGLE DBs with/without their system data.



ATTENTION: When inserting system data by SimaticManagers/ TIA-Portal,

- they shall be uploaded from a PLC with included ConfigStage-configuration (Menu: „PLC → Upload Station to PG“), because they are „ConfigStage-configured“ already or
- the system data of the Simatic-Manager/TIA-Portal overwrite the ConfigStage-system configuration. In that case the *.csbin-file of the project must be downloaded again after the WLD-file was downloaded.

Working with the software „ServiceStage“

Creating backup files (Online Backup)

This function creates

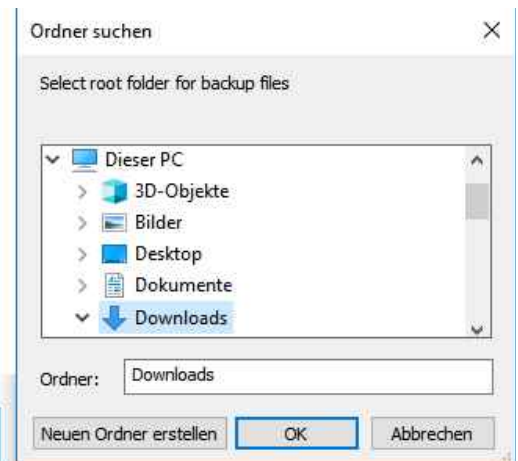
- an WLD-file „S7programs.wld“ with all the S7-system data and all OB, DB, FB, SFB and SFC and
- a visualization binary file „Visudata.vsbin“ and copies this files into a folder „backups“.

With these function S7-programs and visualizations can be backed up from connected devices and by Block update these devices can be updated either by ServiceStage or by a Micro-SD-card only – complete without any PC.

(more at „System functions“, chapter „Data backup and restore“)

Online backup:

Create backup files



ATTENTION:

- To make a visualization backup, it must be allowed to RemoteStage BEFORE, when doing the visualization in VisuStage (check the upload-function in VisuStage-project-header), if necessary a PIN-input is required. Additional a password for communication must be typed in, when assigned in SimaticManager/ TIA/ ConfigStage.
- If “Read Protection” is assigned to the device an “Online Backup” is not possible any more.



ATTENTION: When using Block Upload for backup / restore

- With firmware 2.3.6 a Restore can only be done by WLD- and VSBIN-file. The CSBIN-file will be ignored. So the WLD-file MUST CONTAIN these system data, who where downloaded with CSBIN before.
- If the WLD-file does not contain system data, program stops after restore because of missing data.

Know-how-protection

There are different levels of to protection of the S7-program against unauthorized reading or writing.

Therefore the PLC will be switched to STOP-mode.

KNOW-HOW Protection		
Protection level	Read access	Write access
<input type="checkbox"/> No protection	✓	✓
<input checked="" type="checkbox"/> Read protection	✗	✓
<input type="checkbox"/> Write protection	✓	✗
<input type="checkbox"/> Read write protection	✗	✗
Set protection level		

No protection: The S7-program blocks (DBs) can be read and overwritten from everyone. The protection level can be increased by assigning another level in the „ServiceStage“. There will be asked for an authentication by password, if it was configured so in the Simatic-Manager or TiA-Portal. So only authorized personnel can activate higher protection levels.

Read protection: The S7-program blocks (DBs) can still be manipulated from everyone, but not read any more. Only the Panel-HMIs can still read out the process data (DBs) to visualize it. This protection level can be reset by the PG-function „Clear All“ or by ServiceStage. Than the S7-program is deleted in the PLC and the unprotected original program may be downloaded into the PLC again.

Write protection: The S7-program blocks (DBs) can be read from every one, but no more manipulated. This protection level can only be reset by general reset by hardware directly on the device. Than the S7-program is deleted in the PLC and the unprotected original program may be downloaded into the PLC again.

Read/Write protection: The S7-program blocks (DBs) can no more be read or manipulated. Only the Panel-HMIs can still read out the process data to visualize it. This protection level can only be reset by general reset by hardware directly on the device. Than the S7-program is deleted in the PLC and the unprotected original program may be downloaded into the PLC again.

INSEVIS - Gesellschaft für industrielle
Systemelektronik und Visualisierung mbH

Am Weichselgarten 7
D - 91058 Erlangen

Fon: +49(0)9131-691-440
Fax: +49(0)9131-691-444
Web: www.insevis.de
E-Mail: info@insevis.de

Zertifiziert nach DIN EN ISO 9001:2008