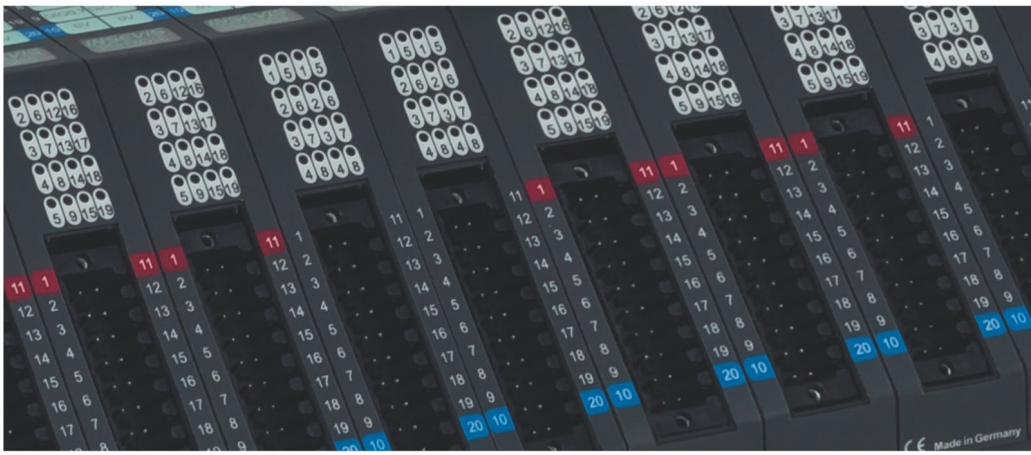
INSEVIS



S7-Panel-PLC Manual

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

Index of contents

General instructions ... 10
About INSEVIS ... 11
Product family S7-Panel-PLC ... 12
- Communication overview ... 12
- Technical data of INSEVIS-S7-CPUs ... 13
- CPU-V and CPU-P ... 13
- CPU-T ... 14
- Touchpanel units for the Panel-PLCs ... 15
- Periphery in the Panel-PLCs ... 16
- Communication to PLC ... 16
- Software for the Panel-PLCs ... 17
- Accessories for S7-Panel-PLCs ... 18
- Wiring of the S7-Panel-PLCs ... 19
  - Wiring of CPU-V/P in devices with/without periphery slots ... 19
  - Wiring of CPU-T in devices with/without periphery slots ... 20
- S7-Panel-PLC Cut out for switching cabinets ... 21

General ... 24
- Memory organization ... 24

Communication ... 25
- Ethernet ... 25
  - Set up IP-address in the Simatic®-Manager ... 26
  - Set up the CPU and the Ethernet connection in the Simatic®-Manager ... 28
  - IP Configuration of PLC with SFB 129 "IP_CFG" ... 29
  - Modbus TCP (Server) ... 29
  - S7 active connection to Partner-PLCs (RFC1006) ... 30
    - Read data from a Remote PLC with SFB 120 "GET" ... 30
    - Write data to a Remote PLC with SFB 121 "PUT" ... 32
    - Read in IP configuration with SFB 131 "IP_GET" (CPU-T) ... 34
    - Set IP-configuration with SFB 132 "IP_SET" (CPU-T) ... 35
  - TCP Send/Receive Connection (RFC 793) ... 36
    - Send data via TCP with SFB 122 "TSEND" ... 36
    - Receiving data via TCP with SFB 123 "TRECV" ... 38
    - Terminate the TCP connection with SFB 124 "TDISCON" ... 40
    - Configure the partner parameter by TCP with SFB 125 "TCONFIG" ... 41
    - Query the Status of TCP connection with SFB 126 "TSTATUS" ... 42
  - UDP Send/Receive Connection (RFC 768) ... 43
    - Send data via UDP with SFB 127 "USEND" ... 44
    - Receiving data via UDP with SFB 128 "URECV" ... 45
- Profinet IO Controller ... 48
  - Configuration with SimaticManager ... 48
  - Configuration with TIA-Portal ... 50
  - Status-LEDs ... 52
- Profibus DP ... 53
  - Profibus DP-Master ... 53
    - Configure the DP-Master ... 53
  - Profibus DP-Slave ... 54
    - Configure the DP-Slave ... 54
    - DP Master configuration for INSEVIS Profibus DP-slave ... 56
    - Structure of DP-Slave diagnostic data ... 57
    - DP-Slave address assign with SFC 75 „SET_ADDR“ ... 60
    - DP-Slave address and baud rate assign with SFC 76 „SET_DPS“ ... 61
- Profibus DP ... 62
  - Profibus DP-Master ... 62
    - Configure the DP-Master ... 62
  - Profibus DP-Slave ... 63
    - Configure the DP-Slave ... 63
    - DP Master configuration for INSEVIS Profibus DP-slave ... 65
    - Structure of DP-Slave diagnostic data ... 66
    - DP-Slave address assign with SFC 75 „SET_ADDR“ ... 69
    - DP-Slave address and baud rate assign with SFC 76 „SET_DPS“ ... 70
- UART ... 71
  - Sending data with SFB 60 "SEND_PTP", free ASCII ... 71
  - Receive data with SFB 61 "RCV_PTP", free ASCII ... 72
  - Sending data with SFB 60 "SEND_PTP", Modbus RTU (Client) ... 73
  - Receive data with SFB 61 "RCV_PTP", Modbus RTU (Client) ... 74
  - Modbus RTU (Server) ... 75
  - Receive data via RS232 with defined End Code ... 76
  - Change the UART-configuration by SFB 130 "CFG_PTP" ... 76
- CAN ... 77

CANopen® Configuration with the ConfigStage....77
Send CAN telegrams with SFB105....79
Receive CAN telegrams with SFB106....79
Receive event OB106....79
SDO transfer SFB107....80
FB107 SDO upload....80
FB108 SDO download....80
FB109 SDO upload....80
FB110 SDO download....81
FB111 SDO upload....81
FB112 SDO download....81
FB113 SDO transfer string....81
Alternative function of SFB107....81
CAN status request SFB114....82
CAN error event OB186....82
Automatic return....82
Configuration with the Software „ConfigStage“....84
Basic settings....84
Assign the IP-address....85
Change target device....85
Addressing of the onboard periphery....85
Standard addressing ind the INSEVIS- PLCs....85
Different addressing in PLCs and addressing of decentral periphery....86
CPU settings....86
WebServer....88
Time settings (time zone, daylight saving time, NTP- server)....88
Communication settings....89
RS232 and RS485....89
Ethernet....89
Profinet....91
CAN configuration....91
Decentral INSEVIS periphery....91
Configure decentral external periphery manually....92
Create library elements of your CAN-slaves....93
Configure decentral external peripherie by EDS-file....94
Decentral external periphery / motion controller pre defined....95
Samples for external CANopen devices....96
System functions....98
Organisation blocks (OB)....98
Organisation blocks (OB) - compatible to STEP®7 from Siemens....98
Organisation blocks (OB) - in addition to STEP®7 from Siemens....98
System function blocks (SFB)....98
System function blocks (SFB) - compatible to STEP®7 from Siemens....98
System function blocks (SFB) - what are not available at INSEVIS-CPUs yet or available on request....98
System function blocks (SFB) - in addition to STEP®7 from Siemens....99
System functions (SFC)....99
System functions (SFC) - compatible to STEP®7 from Siemens....99
System functions (SFC) - what are not available at INSEVIS-CPUs yet or only available on request....100
System functions (SFC) - in addition to STEP®7 from Siemens....100
System functions (SFC) - in addition to STEP®7 from Siemens....101
Import of INSEVIS-blocks into the S7-Project....101
Installation of the INSEVIS-S7-library in the Simatic®-manager....101
Installation of the INSEVIS-S7-library in the TIA-Portal....102
Overview about SFCs for data backup and restore....103
Save data blocks to ROM/ into flash memory....103
Life-backup for a change/ spare device → from device to Micro-SD-Card....103
Manual program backup for device manufacturers → from ServiceStage to PC....103
Creating a manual programme restore on an SD card....103
S7-Program-restore for both kinds of backup....103
Archive/ dearchive (write back) data....104
Read in data from PC-edited CSV-archive file....104
Archiving DBs with SFC 207 "ARCHIVE"....104
Backup & Restore with the SFC 208 "BACKUP“....109
Read data from archive file (SD) with SFC 209 "READFILE"....111
Copy data blocks from work memory to load memory with SFC 210 "COPY_DBL"....113
Copy data blocks from load memory to work memory with SFC 211 "COPY_DBW"....114
Read data from CSV-formatted archive file (SD)with SFC 213 „Read_CSV“....116
Overview about SFCs for a File Management System....118
Create folders on Micro-SD-Card with SFC 223 „CREATE“....119
Copy files on Micro-SD-Card with SFC 222 „COPY“....120
Rename files / folders on Micro-SD-Card with SFC 220 „RENAME“....121
Inform about files / folders on Micro-SD-Card with SFC 224 „INFO“....122

Inform about file size and name on Micro-SD-Card with SFC 225 „FILEINFO“ 123
Delete files / folders on Micro-SD-Card with SFC 221 „DELETE“ 124
Adjust LCD brightness with SFC 214 "LCD_BRIG" 125
Copy RAM to ROM with SFC 254 "RAM2ROM" 126
Update operation system in the PLC 127
General reset by hardware 128
Visualization with the software „VisuStage“ 129
Download and licensing 129
System functions 129
Communication between PLC and external Panel-HMI 129
General settings of the program shell 130
General functionality 131
Recommended procedures 131
Save your resources 131
Keep the overview 131
Transfer your work to other projects 131
General project formats and settings 132
File formats 132
General data 132
Version counter 132
Screen rotations 132
Data protection at upload 132
Conversion of third-party visualisations 133
Requirements 133
Procedure 133
Creation and management of resources 136
Resource Device Settings 136
Assign the target IP-address 136
VNC-Server (CPU-T devices only) 136
NTP-Server, time zones and daylight saving time (CPU-T devices only) 136
Configure a VNC-Viewer/-Client 137
Change screen orientation or resolution 138
Create and administrate resources 138
Resource languages 138
Resource variables 139
Synchronization of variables with TIA®-Portal/ Simatic®-Manager 143
Internal variables 144
Multiplex variables (variables list) 145
Indirect variables 145
Resources Texts and Text lists 147
Resources Images and Image lists 149
Multiple replacement of resources 151
Resource Messages (Alarms and Events) 152
Resource User administration 154
Change user level by SFC215 "LOGIN" 155
Resource Partner-PLC (for HMI only) 156
Partner-PLC Synchronization (for HMI only) 156
Resource Dialogs 157
Resource Screen templates 157
Resources Trends 158
Starting the trend sampling with SFC 202 "TRENDSRT" 159
Stopping the trend sampling with SFC 203 "TRENDSTP" 161
Continue the trend sampling with SFC 204 "TRENDCNT" 162
Query the trend status with SFC 205 "TRENDSTA" 163
Resource recipe administration and recipe viewer 165
Recipe handling with SFC 206 "RECIPE" 167
Ressource Menues 169
Basic functions 170
Query the current the screen ID number with SFC 200 "SCR_GET" 173
Changing the screen with SFC 201 "SCR_SET" 174
Creating a visualization screen 175
Static objects 175
Lines, rectangles, texts, images 175
Images 175
Dynamic objects 176
Date/Time fields 176
I/O fields 177
PIN input fields 178
Diverse kinds of buttons 179
Touch areas – invisible buttons 181
State areas 182
Radio boxes and Check boxes 183

Progress bars....184
Sliders....185
System data (IP-address field, languages, etc.)....186
Gauges....187
Viewers....189
Image and text lists....189
Trend viewer and trend archive viewer....190
Message viewing and archiving....192
Recipe viewer....194
Function graph....197
Working with library elements....199
Creating libraries and ~ elements....199
Using library elements....199
Working with image blocks....200
Simulation and download....201
Error treatment....201
Simulation and download....202
Download....204
Remote access with the software „RemoteStage“....205
General....205
Selection of the remote-device....205
Insert PIN code....206
Using and setting up the remote screen....206
Storing the remote (on the PC) archived data....206
Uploading, converting and storing archive data as csv-files....207
Display and export event and alarm archives....207
Recipe handling....209
Data (DB) archive upload, display, storing and download....210
Command line of RemoteStage (Viewing mode)....211
Automatic connecting to remote device after starting PC....211
Automatic reconnect to remote device after disconnect....211
Automatic full screen after starting PC....211
Set up TCP port number for S7-communication....211
Command line of RemoteStage (Archive mode)....212
Read in (upload) message- and trend archives from remote device....212
Read in (upload) recipe archive from remote device....213
Write back (download) recipe archive into remote device....214
Read in (upload) data (DB-) archive from remote device....215
Write back (download) data (DB-) archive into remote device....215
Return codes....216
Working with the software „ServiceStage“....217
Function overview....217
General information....217
Operating mode....217
Date and time....217
Memory....218
Diagnostic buffer....218
Block Update....218
Creating backup files (Online Backup)....219
Know-how-protection....219
Variable monitoring....220

Changes to older versions of the manual

Rev. 2 / 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. 3 / 2012:
new: chapter recipe administration and recipe viewer
changed: SFB60/61 (UART) parameter description

Rev. 4 / 2012:
new: chapter „Remote access with software „RemoteStage“

Rev. 5 / 2012:
new: chapter „Screens“ with backlight control, clean screen, etc

Rev. 6 / 2012:
changed: chapters „ConfigStage“; „RemoteStage“, „VisuStage“ completely rebuilt and referred to demo visu 2012_04

Rev. 7 / 2012:
changed: chapters „VisuStage“ new functions like color/text switching buttons, limited I/O-fields, etc., (from VisuStage 2.0.1.4 referred to demo visu 2012_04_1, from PLC-OS 2.0.31)

Rev. 8 / 2012:
changed: description of Profibus-signals made compatible to Siemens-manuals, explanation of cycling time added (ConfigStage-CPU settings), RemoteStage with „Read-In-by-Ethernet“-functionality, VisuStage: improvements at receipt administration and objects placing (from VisuStage 2.0.1.5 referred to demo visu 2012_04_2, from PLC-OS 2.0.33)

Rev. 1 / 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. 2 / 2013:
new: ServiceStage added

Rev. 3 / 2013:
new: Memory organization added, chapter CAN: more information, VisuStage V2.0.2.0 described, buzzer added, chapter VisuStage completely reworked (now referred to new demo visu 2013_01, from PLC-OS 2.0.37)

Rev. 4 / 2013:
new: VisuStage V2.0.2.2 and RemoteStage 1.0.3.3 new features (with OS 2.0.38), text buttons and watchdog added now referred to new demo visu 2013_02

Rev. 5 / 2013:
new: New description for VisuStage 2.0.2.3 / RemoteStage 1.0.3.4 (PLC-OS 2.0.39, HMI-OS 1.0.7) with color change at Texts, confirmation messages at recipes, sorting at message archive, adjustable font sizes and scales in progress bars and trend archives, new download mechanism, I/O-fields with integer-input with comma, now referred to new demo visu 2013_04

Rev. 6 / 2013:
new: New description for ConfigStage: PDO-/ SDO-Mapping, address overview list (from ConfigStage 1.0.14.7) description for VisuStage: Use of SFC201-5 from PLC with separated Panel-HMI (with PLS-OS 2.0.40 and HMI-OS 1.0.9)

Rev. 7 / 2013:
new: New remote-functions; recipe records: read out/ store, (DB-) archiving: read out/ store
Read in visualization binary over Ethernet directly from Panel (with RemoteStage V1.0.3.7)
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. 8 / 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. 9 / 2013:
new: VisuStage 2.0.3.1: library management, group/ungroup function, RemoteStage 1.0.4.0: PIN-code request at visualization-binary import valid from firmware 2.1.1
changed: description of embedding S7-lib into S7-project moved to system functions

Rev. 10 / 2013:
new: VisuStage 2.0.3.4 / RemoteStage 1.0.4.2: Rotation of screens possible, Filtering of variables, Additional display of trigger-and acknowledgment variables at alarm/event resources, from PLC-firmware 2.1.3 and HMI-firmware 1.1.3

Rev. 11 / 2013:
new: VisuStage 2.0.3.5: „Cross Reference“-function and display of list resources at image/text lists. ConfigStage 1.0.14.10 „Change device“ function added
changed: description of SFB60/61 UART improved

Rev. 01 / 2014:
new: VisuStage 2.0.3.6 and RemoteStage 1.0.4.3 : Grid/Snap, Screen template/Master foil, function „Screen change with PIN-input“, I/O-Fields: Input in calculator-style (with „comma“-key) inserted from PLC-firmware 2.1.4, HMI-firmware 1.1.4

Rev. 02 / 2014:
new: VisuStage 2.0.3.9 and RemoteStage 1.0.4.5 : new: Function graph, added: button design @ virtual keyboards from PLC-firmware 2.1.5 SFC209 READFILE, SFC2010 LOAD_DBL, SFC2011 LOAD_DBW new
changed: Structure in chapter VisuStage improved, Backup&Restore- resume in chapter System functions added

Rev. 03 / 2014:
changed: S7-program lines as sample inserted for all SFCs and Ethernet SFBs, Hints for VIDEOS added

Changes to older versions of the manual

Rev. 04 / 2014:
changed: Improved description of SFB 120 Get and SFB 121 Put in one connection,
Improved description of SFB 208 (Saving password file)
Rev. 05 / 2014:
changed: VisuStage: new Find-, Overview- and CleanUp functions with version 2.0.4.0
Rev. 06/ 2014:
new: VisuStage 2.0.4.2: new function: User Lever Selection at PIN-Input (with firmware 2.1.9 and Remote Stage 1.0.4.7)
changed: better description of how to assign a connection between Panel-HMI ↔ PLC
Rev. 01 / 2015:
new: VisuStage 2.0.4.4: Lean-Version after 30 days, new function: TSAP-configuration, select color of IP-field by variable,
insert hexadecimal values into I/O-field, ConfigStage 1.0.14.15: new function: TSAP-configuration (all with firmware 2.2.1)
Rev. 02 / 2015:
new: VisuStage 2.0.4.5: set colors / appearance to message texts / backgrounds of the message viewers
(with firmware Panel-PLC 2.2.3, Panel-.HMI 1.1.9) , remote visible by RemoteStage 1.0.4.10
Rev. 03 / 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. 04 / 2015:
new: CPU-T-inserted, adaptions (extensions) for CPU-T functionality inserted,
changed: new VisuStage-functionality at images, rectangles, texts, progress bars, 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, VisuStage: Import Variables from TIA (VS 2.0.5.0 / OS 2.3.5 (CPU-V/-P) 2.3.7 (CPU-T)
changed: System functions: Description of SFC208 increased, image inserted for manual backup,
UART-, CAN-, Ethernet chapter debugged,
Rev. 02 / 2016:
new: SFC 214 LCD_BRIG added in system functions
changed: description for manual backup changed, description added how to scale an I/O-field in VisuStage
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)
Rev. 04 / 2016:
new: VNC-Server in CPU-T-Panels (and Panel-PLCs), change language and screen by variable
Time synchronization with partner-device (for HMI only)
with VisuStage 2.0.5.2 and PLC-firmware 2.4.0 (CPU-T) or HMI-firmware 1.2.9 CPU-T
changed: RemoteStage: Description at download recipes and data archives into remote-device improved
ServiceStage: Description at Online Backup expanded
Rev. 05 / 2016:
new: VNC-Viewer in VisuStage described, from VS 2.0.5.2, firmware 2.4.0 (PLC) and 1.2.9 (HMI) -CPU-T only
changed: System functions, description of hardware reset expanded, ServiceStage: Description at Online Backup expanded once more
Changed to new design line
Rev. 06 / 2016:
new: Multiplex-variables in VisuStage from VS 2.0.5.3, Firmware 2.4.3 (only SPS-T), 1.3.1 (only HMI-T)
IP-request at download in ServiceStage from V 1.0.2.0, 2.4.3 (only CPU-T) / Firmware 2.4.0 (CPU-V/P), (1.3.1-HMI-all)
changed: Description o I/O-fields and date/time-fields separated and improved
Rev. 01 / 2017:
new: VisuStage 2.1.0.0: Gauges, Sliders, Key functions “Delete Arcives”, 1024 events for CPU-T-devices,
vertical alignment for multiline texts, access rights for recipe viewer, inserting objects by short cut and popup windows,
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: VisuStage 2.1.0.4: rotate CPU-T-Panel-HMIs and Panel-PLCs, requires firmware Panel-PLC 2.5.1
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. 03 / 2017:
new: VisuStage 2.1.0.7: for CPU-T-devices: new trend (archive) functions added and described,
PLC-firmware 2.5.4, Remote Stage 1.0.4.30 needed for remote functions
Rev. 01 / 2018:
new: VisuStage 2.1.0.13: to be simulated with S7-PLCSIM, new filter windows for resources, key function alternatively to navigation
bars at trend,- recipe, message viewer, Remote Stage 1.0.4.33 needed for remote functions
changed: more samples for batch file applications for RemoteStage, description of resource partner devices improved
Rev. 02 / 2018:
new: VisuStage 2.1.0.16: Assign of compilation-no´s (manual / automatical, Notification when operator changes the I/O-field value
changed: SFC206: new function No. 6 added (management of reciepe records on SD-card)
(only for CPU-T-devices and from firmware PLC-T 2.5.7, RemoteStage version must be 1.0.4.36)
Rev. 03 / 2018:
new: SFC 220...224 for file management system on Micro-SD-Card added (system functions)
Rev. 04 / 2018:
new: VisuStage 2.1.0.19: Synchronization of S7-variables with S7-/TIA-projects, find & replace of resources with (partly) the same
names, Button-function “variable driven multi menue changing” (for CPU-T-devices only, inactive buttons visible now, when no
access right, Firmware (Panel-)PLC-T: 2.5.9, Remote Stage 1.0.4.38 needed

Changes to older versions of the manual

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,
For T-CPU: indirectly variables for indirectly addressing, 50.000 alarm- and event archives from firmware: CPU-V/P 2.4.9 / CPU-T 2.6.2, VisuStage from 2.1.0.24, RemoteStage from 1.0.4.42

Rev. 03 / 2019:
new: SFC 225 “FINEINFO” for PLCs with CPU-T, from firmware: CPU-T 2.6.6

Rev. 04 / 2019:
changed: Profinet configuration be TIA-Portal explained

Rev. 01 / 2020:
new: for CPU-T-devices: VisuStage: import of variables from TIA-V16, new object ellipse, new way to set up text-/ image and multiplexvariables lists, 2 new system variables for screen and user, copying between 2 VisuStage instances with linked objects, VisuStage: 2.1.0.27, Firmware for PLC-T; 2.6.9, for HMI-T; 1.6.0
changed: VisuStage: description of first installation improved, RemoteStage: Description of DB-archiving improved

Rev. 02 / 2020:
changed: VisuStage: Description of TSAP at “PLC partner devices - for HMIs only“ improved,
RemoteStage: Description of command lines improved, not supported functions in view mode described

Rev. 03 / 2020:
new: SFB52, 53, 54 added, time settings, summer time, NTP- server added, ConfigStage 1.0.14.41, Firmware 2.7.2

Rev. 01 / 2021:
new: VisuStage 2.1.0.36, Firmware for Panel-PLC-T; 2.7.5, for HMI-T; 1.6.6 with indirect variables (for scaling), with integrated converter TIA-WinCC → VisuStage, needs RemoteStage 1.0.4.49
changed: ServiceStage 1.0.2.9 with displaying present protection level of the PLC,
Profinet state-LED description improved, ModbusTCP server description adapted to new features

Rev. 01 / 2022
changed CPU Typ P and lift arm connectors discontinued, AI8 Perephery Card added, Panel PLC Pictures updated, Corrections.

Rev. 01 / 2023:
new: VisuStage 2.2.0.1 with Image Blocks, RemoteStage 1.0.4.51 required

Rev. 02 / 2023:
New: VisuStage 2.2.1.0 New functions for image blocks (create, edit and duplicate from project), as well as plausibility check of the faceplates. New functions for lines and rectangles (round corners, other lines and arrow ends). General improvements and bug fixes.

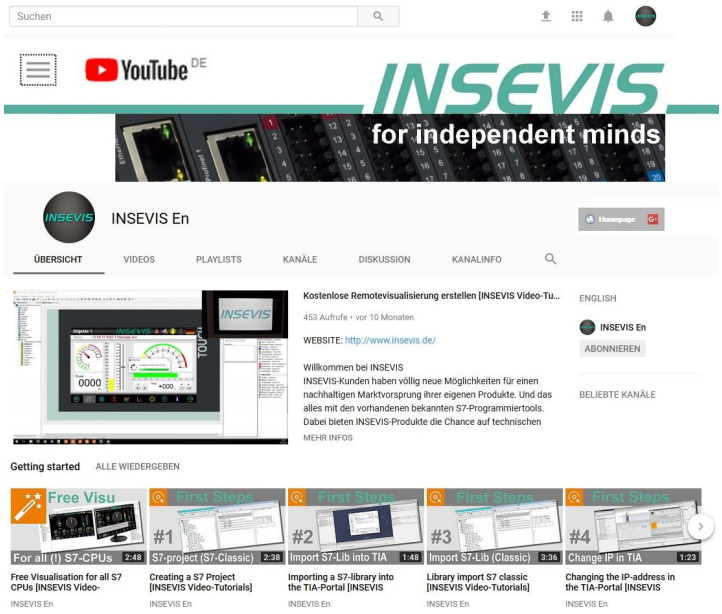
Rev. 01 / 2024
changed Additions to backup and restore functions, as well as the restore from SD card after an overall reset.

Rev 02/2024
new ServiceStage 1.0.4.0: Observation table for variables added, System functions SFC207 information added

Hint for better understanding by application videos

On our english **INSEVIS-En** Youtube channel we provide tutorial videos in various playlists sorted by topic, referring to single details and functions described in this manual

This will help you to get familiar with INSEVIS much faster – PLEASE use it beside this manual!



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 exclamination 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 starlikely 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®, TIA® and other as reserved trade mark of Siemens AG,
- CANopen® and other as reserved trade mark of CAN in Automation eG,
- WINDOWS® and DOS® 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-CPUs -V and -P are the base of the successfully INSEVIS product families with Profibus DP Master/Slave.
With the S7-CPU-T Panel-PLCs and Compact-PLCs are available with Profinet IO Controller.

Step®7-Programability

INSEVIS-S7-CPUs are programmable by STEP 7® - AWL, KOP, FUP, S7-SCL, S7-Graph from Siemens and in general command-compatible to Siemens-CPU S7-315-2PNDP. 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.

Certificato Certyfikat Certificate Certificado Certificat

Zertifikat

TAW Cert

Die Zertifizierungsstelle TAW Cert
Zertifizierungsgesellschaft mbH für QM-Systeme und Personal

bescheinigt hiermit, dass das Unternehmen

INSEVIS
Gesellschaft für industrielle Systemelektronik und Visualisierung mbH
Am Weichselgarten 7
91058 Erlangen

für den Geltungsbereich
Entwicklung, Service und Vertrieb von Systemelektronik und Software für industrielle Automatisierungslösungen

ein Qualitätsmanagementsystem eingeführt hat und anwendet.

Durch ein Audit am 31.07.2014 wurde der Nachweis erbracht, dass die Anforderungen der
DIN EN ISO 9001
(Ausgabe 2008)
erfüllt sind.

Erstzertifizierung: 14.09.2011

Dieses Zertifikat ist gültig bis 13.09.2017

Zertifikat-Registrier-Nr.: 19010511

TAW Cert

DAkkS

Altdorf, den 14.09.2014

Dr.-Ing. Dietmar Winner, Geschäftsführer TAW Cert GmbH

TAW Cert GmbH • Fritz-Bauer-Str. 13 • 90518 Altdorf

Dieses Zertifikat ist Eigentum der TAW Cert GmbH

INSEVIS operates a yearly certified quality management system ref. to DIN EN ISO 9001:2015.

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.

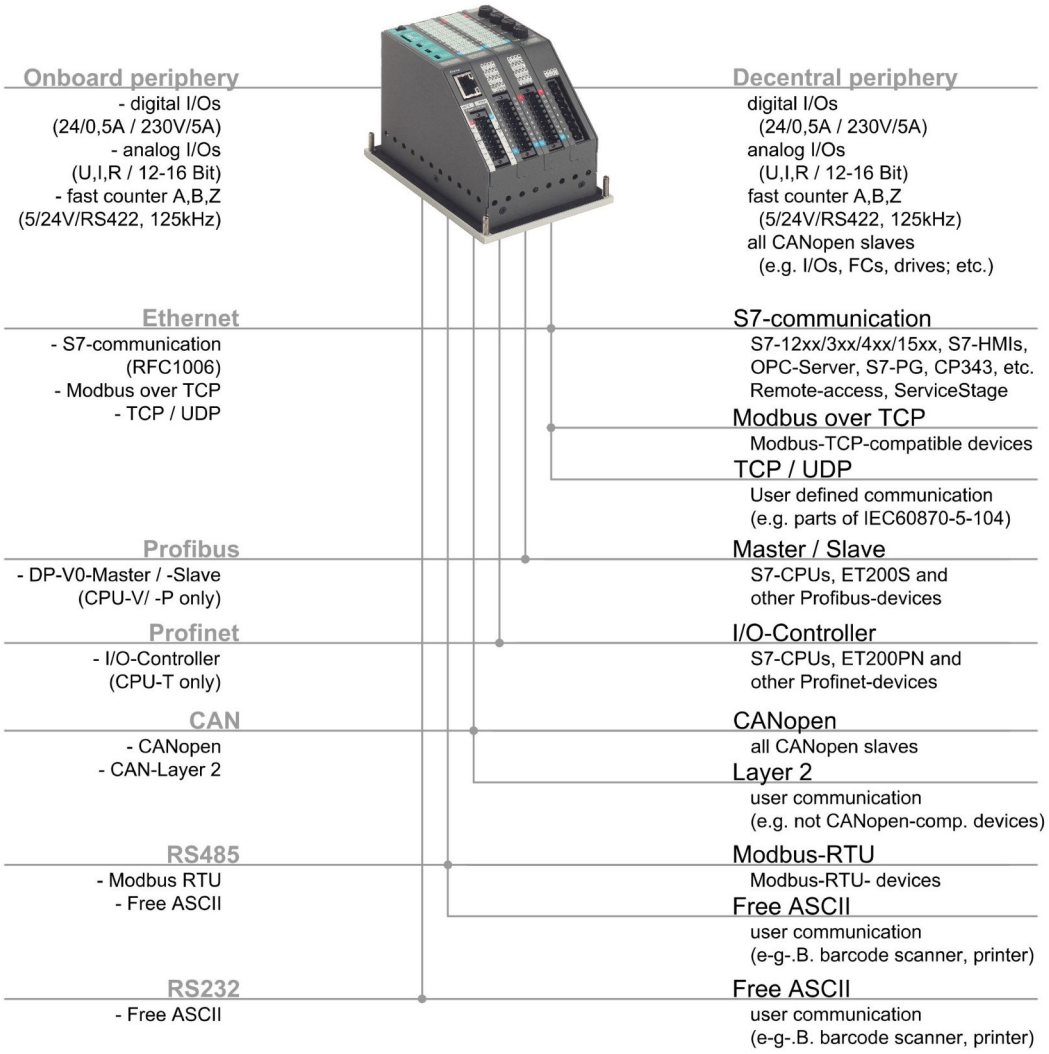
These products were put into comprehensive validation tests before they were produced in selected and certified production lines.

INSEVIS - Made in Germany

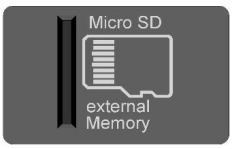
Product family S7-Panel-PLC

Communication overview

INSEVIS-S7-Panel-PLCs contain versatile possibilities for a connection to diverse peripheries 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 panel 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-CPUs

CPU-V and CPU-P

Devices with **CPU-V** and **CPU-P** fit properly to small and medium sized applications in the low cost-areas of Panel-PLCs with high graded visualization (Typ V best for 3,5 to 5,7“ and Typ P better for 7 to 10,2“) and with lots of communication interfaces. Profibus is optional available.

CPU-V and CPU-P

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) with shielded wires 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	CPU-P
Working memory, thereof buffered by akku	512kB 256 kByte remanent	640kB 384 kByte remanent
Load memory	2MB flash memory	2MB flash memory
Memory for visualization	4MB flash memory	24MB flash memory
external memory	Micro SD, up to 8 GB	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...

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...

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.

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-CPUs

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 seperated 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 Local data Number of inputs and outputs Process image Number of Merkerbytes Number of Taktmerker Number of timer, counter Depth of nesting	each 2.048 32kByte (2kByte per block) in each case 4.096 Byte (32.769 Bit) addressable in each case 4.096 Byte (default set is 128 Byte) 4.096 (remanence adjustable, default set is 0..15) 8 (1 Merkerbyte) in each case 512 (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) with shielded wires 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.

Touchpanel units for the Panel-PLCs

General

By using TFT- displays a high brilliance of colour fastness and brightness and a wide viewing angle are possible. LED-backlight does assure typical half-live periods of 50.000 operating houres 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 cut out matches with 7” - CPU-T),,
- (7” P-Series CPUs are discontinued and replaced by V / T Series CPUs.)
- 10,2“ with WVGA- resolution (800x480pixel, 16:9-format) → for front panel use cut out matches with 10,1” - CPU-T),
- (10” P-Series CPUs are discontinued and replaced by V / T Series CPUs.)

With CPU -T

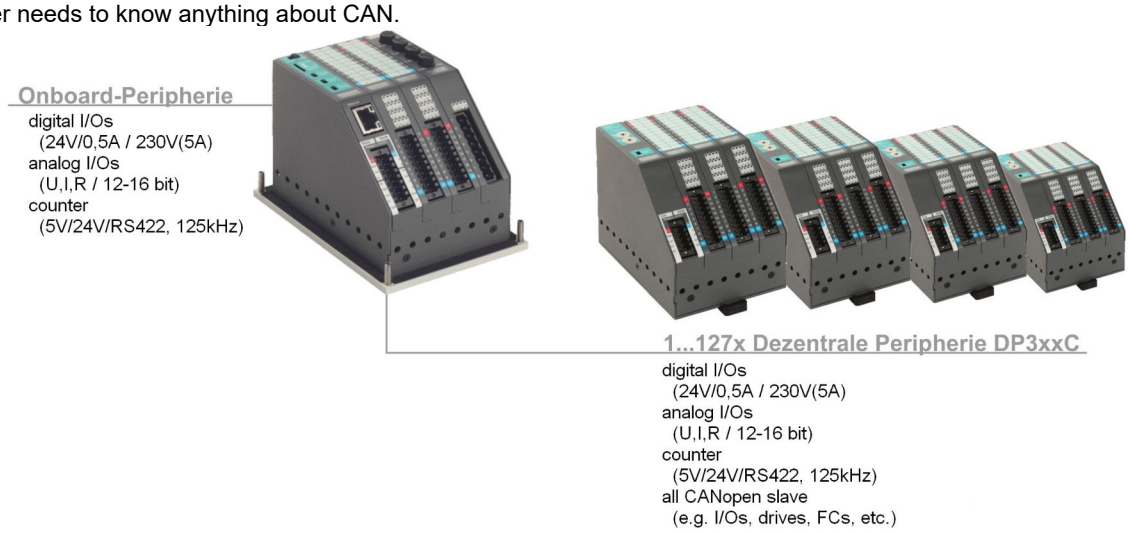
The S7-Panel-HMI family with the CPU-T consist of 4 different display diagonals:

- 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 (cut out matches with 7” - CPU-P),
-
- 10,1“ with a resolution of 1024x600pixel, 16:9-format → for front panel use, (cut out matches with 10,2” - CPU-P),
- 15,6“ with a resolution of 1366x768pixel, 16:9-format → for front panel use, (temp. range 0...50°C only)

Periphery in the Panel-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.



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 DO4R (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 analog in- and 2 analog outputs)
Periphery module AI8 (8 analog inputs)
Periphery mod. 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 Storing temperature range	-20°C ... +60°C (without condensation) -30°C ... +80°C
Dimension (W x H x D) Weight	20 x 108 x 70 mm ca. 150 g
Wire length unshielded (max.) shielded (max.)	 30 m 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 configerable 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 for the Panel-PLCs

The INSEVIS Software Tools are free, running on a Windows PC and can be executed in german or english language.

Configuration

With the configuration tool „ConfigStage“ you can config the additional functions of the INSEVIS-CPUs 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, retententive memory, etc.). A web server can be configured at CPU-T-PLCs.
A variety of Step7 Blocks for motion conrol functions, intelligent CAN-Drives and example configurations of external peripheries are listed on our website as free downloads.

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

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

Visualization

„VisuStage“ imports symbolic variables from SimaticManager or TIA-projects, it exports texts to be translated easily. Exacts error messages help you to debug your project until it can be simulated and downloaded.
For the use of the mulilinguistic version of “VisuStage” it's necessary to buy a company license.
The software is providing among other things the following features:

- Alarm- and event messaging system and -archive up to 1024 alarms (50000 storeable) on each,
- Trend data sampling, and archiviing for 4 trends with 16 channels each (655000 storeable on each),
- Recipe management system with up to 64 recipes with 256 elements in 256 records,
- User management system with 9 layers and password protection,
- Screen saver, backlight switch-off function, clean screen and buzzer,
- Library with 2D- and 3D-symbols, integrated simulation,
- 4x VNC-server for CPU-T-Panel-PLCs.

Remote access

With software „RemoteStage“ there can be made a remote visualization from the binary of a VisuStage-visualization created before. This program comes as an directly executable file and can be also used for creating batch files. The Software communicates via S7-Communications (Put/Get) with the S7-CPUs and reads the process data which can be remotly visualized. Multiple RemoteStages can be operated in one PC to get a kind of master display with multiple remote screens.

- 1024 alarm archives and 128 Eventchives (Panel-PLC and Panel-HMI),
- 4 trends with up to 16 channels (Panel-PLC and Panel-HMI),
- 256 recipe records with up to 256 elements (Panel-PLC and Panel-HMI),
- Data (DB) -archives (Panel-PLC and Compact-PLC)

Service tool

The „ServiceStage“ is installed in a minute and made to have easy service access to the PLCs. Designed to make service and maintenance work easier. Without using complex programming tools for this kind of work it's a real time saver i

- Reading and editing device specific data,updating firmware version (CPU-T only),
- Changing of the operation mode RUN ↔ STOP,
- Set and synchronize date and time,
- Memory diagnostic and compression,
- Read out, show and store CPU-diagnostic buffer,
- Download of S7-program, visualization- and configurations binary,
- create backups of visualization and S7-program,
- Assign the know-how-protection levels.

Accessories for S7-Panel-PLCs

There are available pin marked removable connectors 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.
The connectors E-CON XX with lift arms are discontinued and replaced by E-CONS XX connectors with bolt flanges.)

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-CONS16 (pin marked connectors for max. 1,5mm² cross sections)	**For CPU V:** Connector 2x8pin, bolt flanges From 2023 the Pin-Numbers are printed on the sides. Adapter for Profibus SUB-D 9	 E-CONS16-00 E-AD-DP12-00	 1 1
 E-CONS10 (pin marked connectors for max. 1,5mm² cross sections)	**For CPU T:** Connector 2x5pin, bolt flanges From 2023 the Pin-Numbers are printed on the sides.	 E-CONS10-00	 1
 E-CON09 (pin marked connector for max. 1,5mm² cross sections)	**Only for PC351V/P (periphery):** Connector 2x4pin	 E-CON09	 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:** [2] Company license, no run-time costs per single PC [3] Maintenance license due yearly, if new versions are required, otherwise only old versions do run Software available for free download at INSEVIS web sites	VisuStage ConfigStage RemoteStage ServiceStage VisuStage full version company license VisuStage full version maintenance license	- - - - SW-VS-2.0 [2] SW-VSW-2.0 [2] [3]	free download free download free download free download 1 1
Spare parts **Hint:** A mounting set with grounding terminal is part of every delivery	Additional mounting set with grounding terminal for 3,5“ and 4,3“-devices Additional mounting set with grounding terminal for 5,7“ and 7“-devices Additional mounting set with grounding terminal for 10,1“ and 10,2“-devices Additional mounting set with grounding terminal for 15,6““-devices	E-MNT35-00 E-MNT57-00 E-MNT100-00 E-MNT156-00	10 10 10 10

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

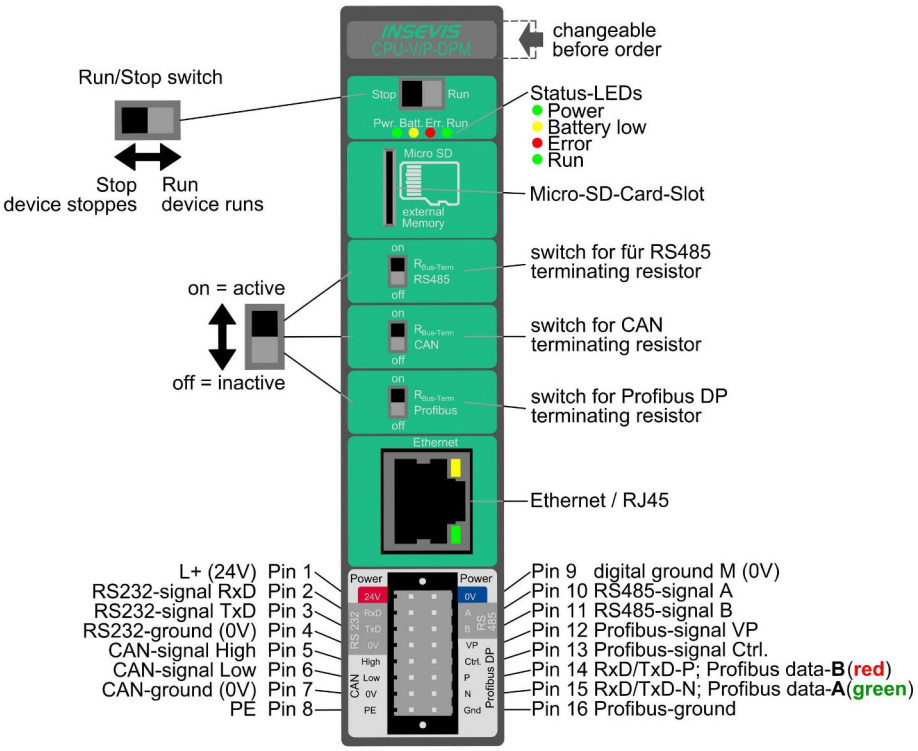
Wiring of the S7-Panel-PLCs

Wiring of CPU-V/P in devices with/without periphery slots

Wiring in basic devices (without periphery slots)

Pointer:
If the Battery is empty the Battery LED lights up yellow. When the Battery is recharged the yellow LED will be on till Power OFF/ON is switched.

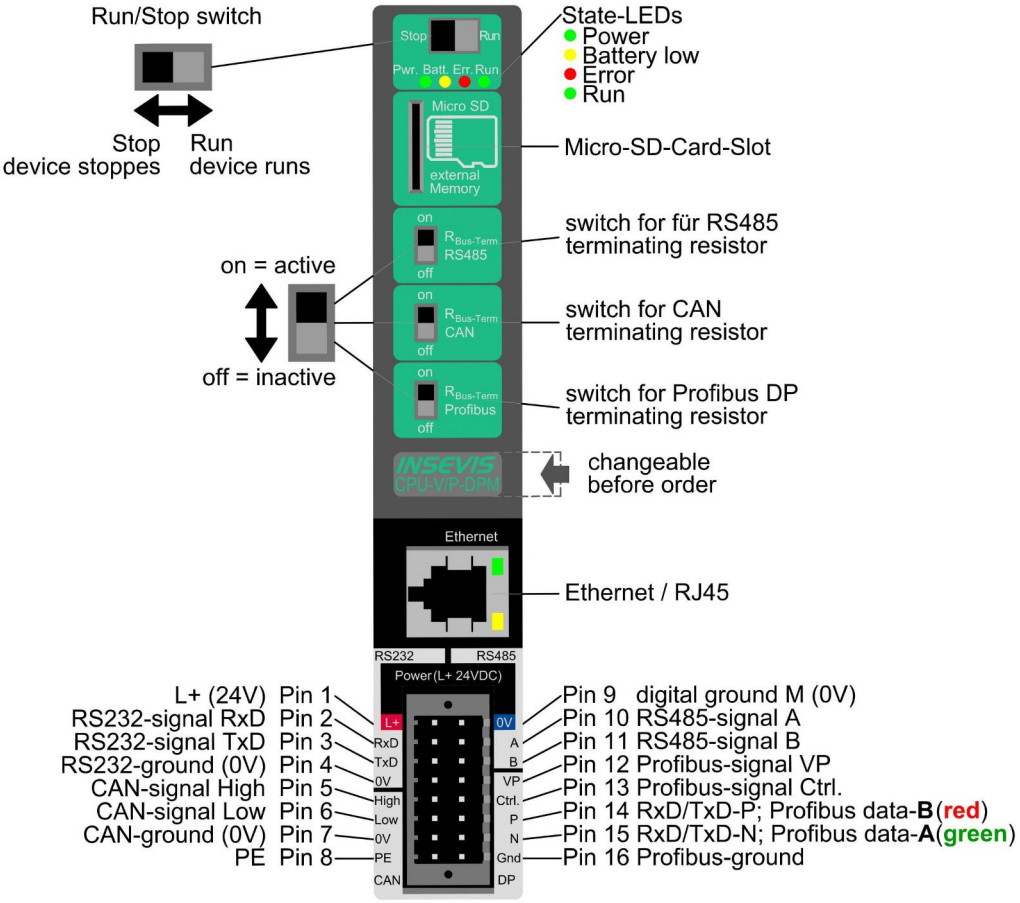
with shielded wiring like described in RS232-spec!



Wiring in devices with periphery slots

Pointer:
If the Battery is empty the Battery LED lights up yellow. When the Battery is recharged the yellow LED will be on till Power OFF/ON is switched.

with shielded wiring like described in RS232-spec!

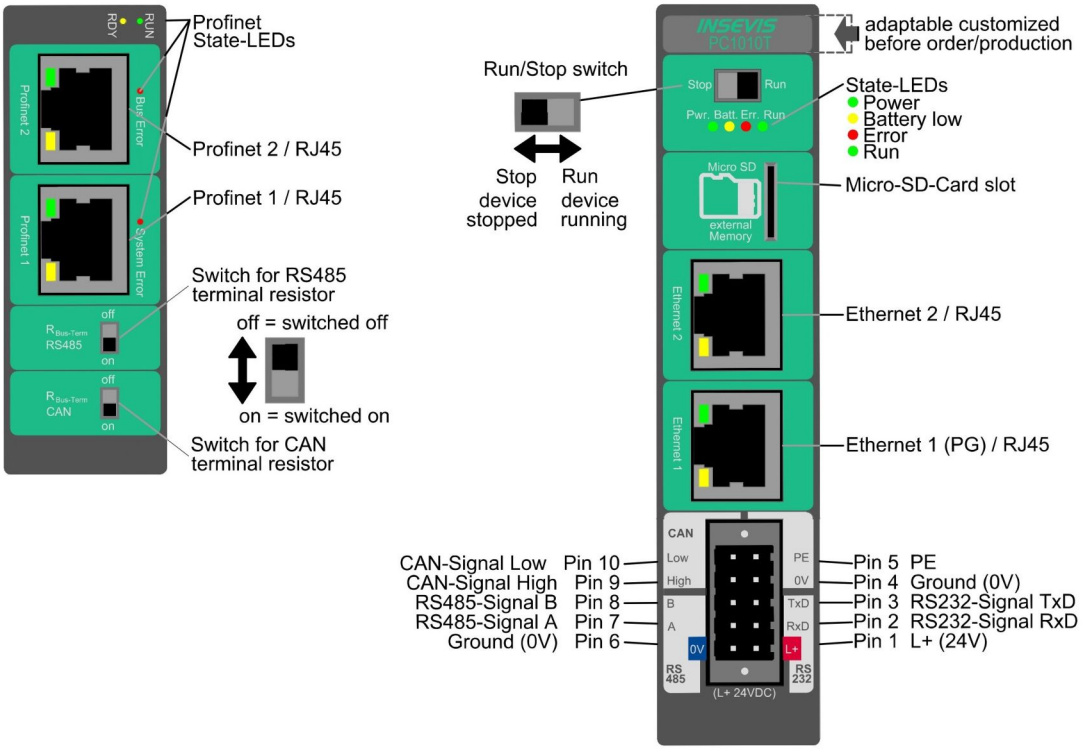


Wiring of the S7-Panel-PLCs

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

Wiring in basic devices (without periphery slots)

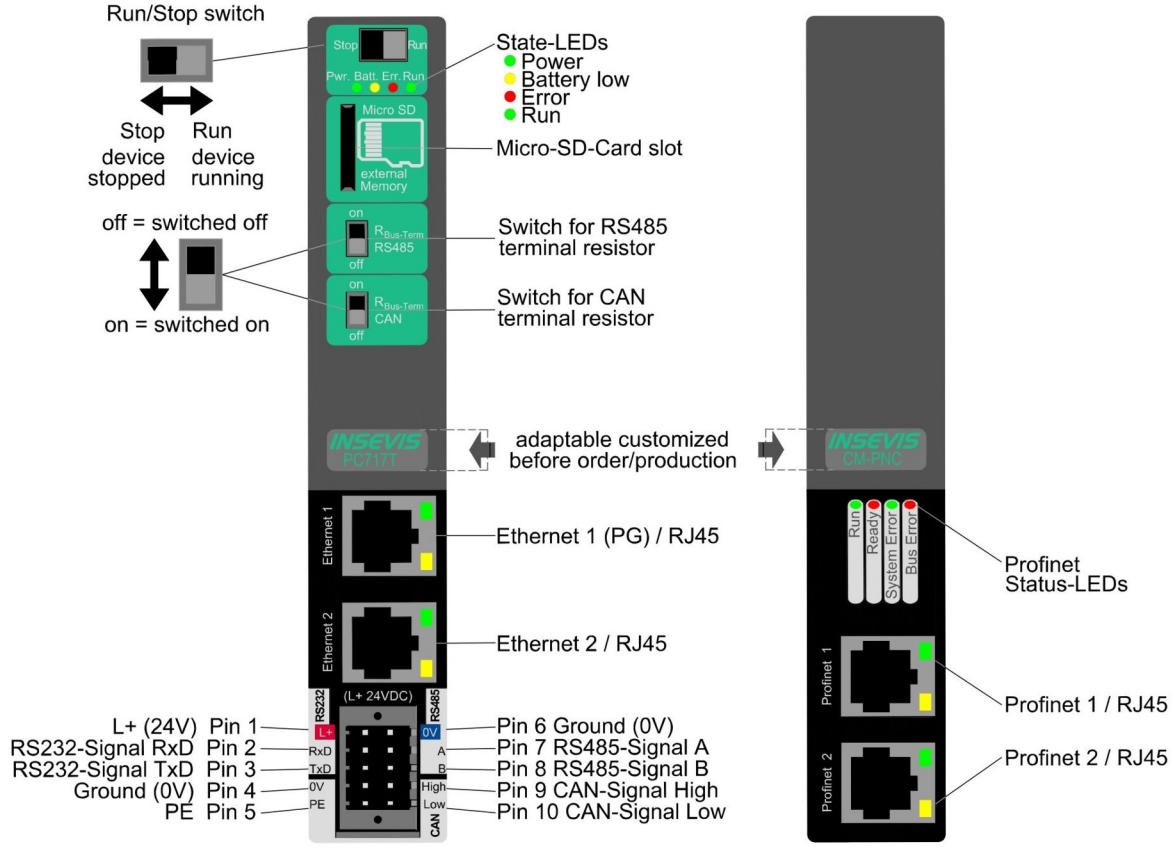
Pointer:
If the Battery is empty the Battery LED lights up yellow. When the Battery is recharged the yellow LED will be on till Power OFF/ON is switched.



with shielded wiring like described in RS232-spec!

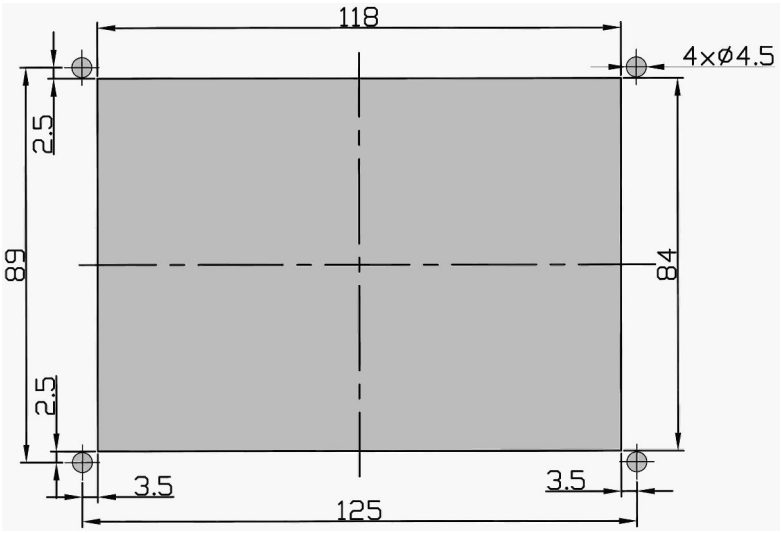
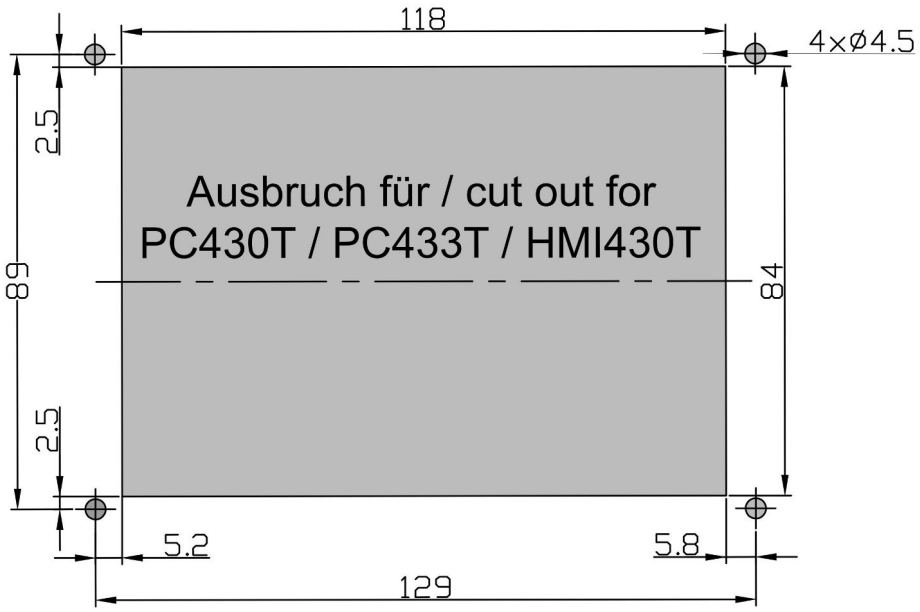
Wiring in devices with periphery slots

Pointer:
If the Battery is empty the Battery LED lights up yellow. When the Battery is recharged the yellow LED will be on till Power OFF/ON is switched.

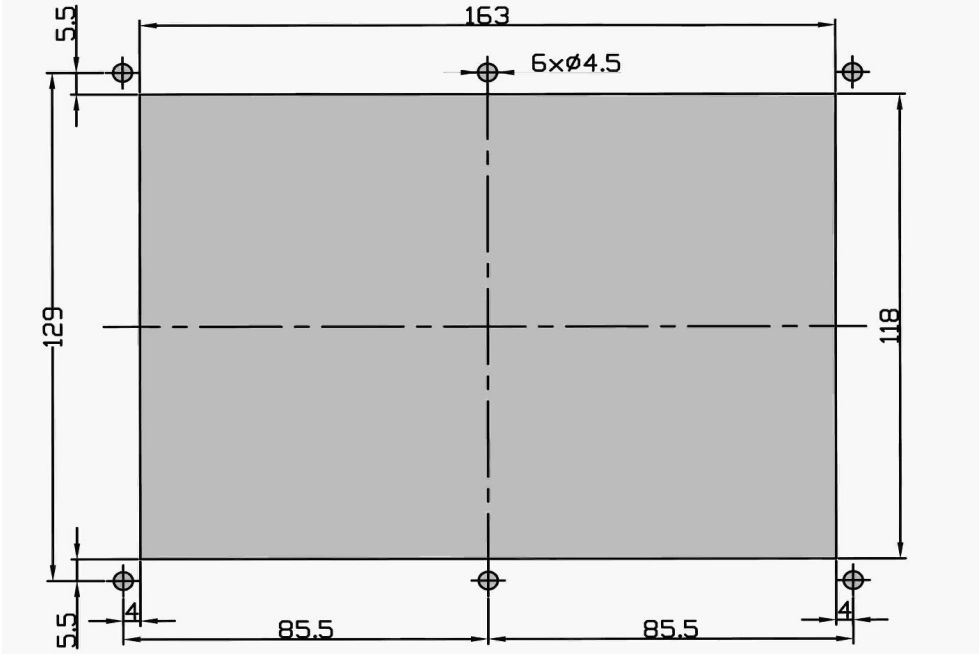
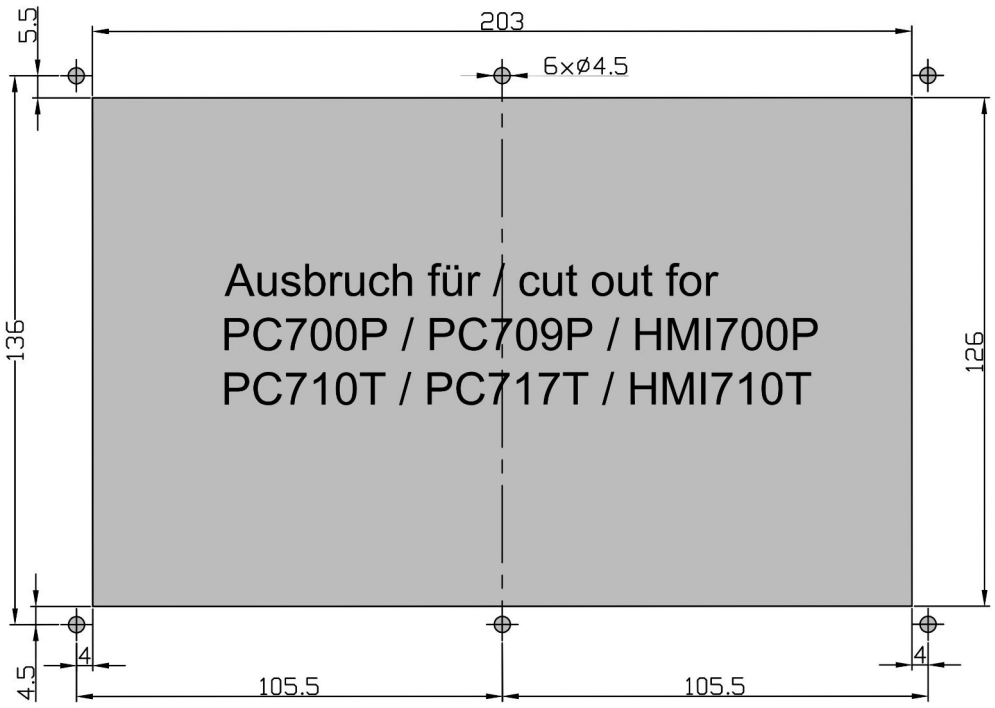


with shielded wiring like described in RS232-spec!

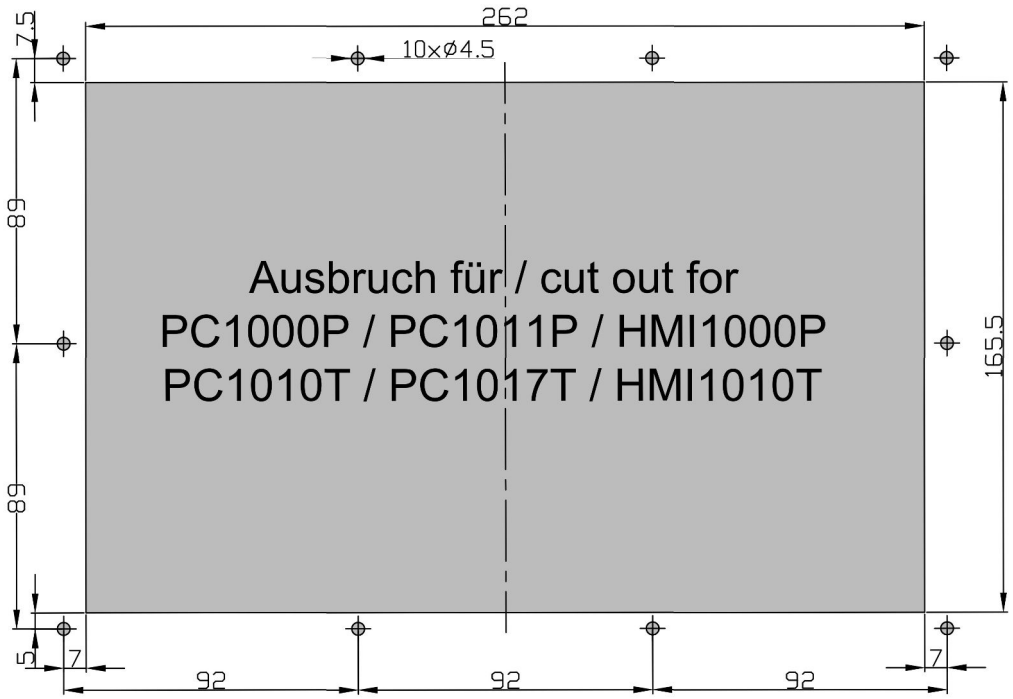
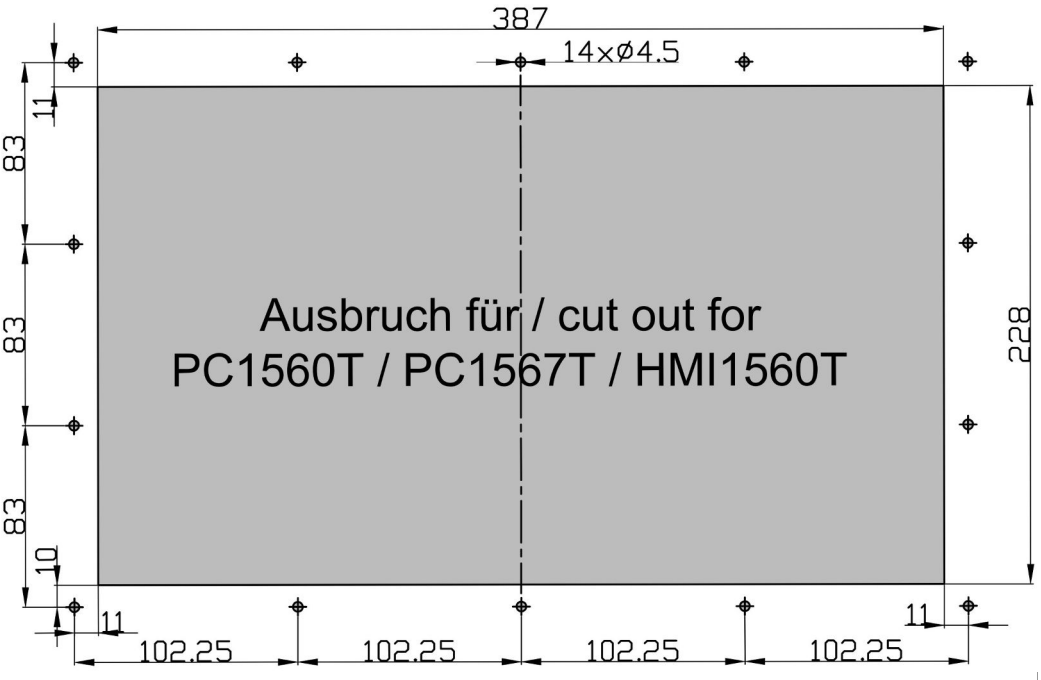
S7-Panel-PLC Cut out for switching cabinets

Device	Cut out for devic(es)
Dimensions Cut out W x H (mm) / 118 x 84 4 holes with D 4,5mm **Wiring outlet** at rear view and horizontal mounting **for PC350V/P, PC351V/P, HMI 350V/P** Wiring outlet - to the right Mounting depth ca. 49mm max. **for PC353V/P** Wiring outlet - 45° to the right Mounting depth ca. 89mm max.	 Fig.: Cut out for all 3,5”-devices
Dimensions Cut out W x H (mm) / 118 x 84 4 holes with D 4,5mm **Wiring outlet** at rear view and horizontal mounting **PC430T, HMI430T** Wiring outlet - VCC / 2xETH to the left - 2x RJ45 (PN) down Mounting depth ca. 44mm max. **PC433T** Wiring outlet - 45° to left Mounting depth ca. 89mm max	 Fig.: Cut out for all 4,3”-devices

S7-Panel-PLC Cut out for switching cabinets

Device	Cut out for devic(es)
Dimensions Cut out W x H (mm) / 163 x 118 6 holes with D 4,5mm **Wiring outlet** at rear view and horizontal mounting **for PC570V/P, HMI570V/P** Wiring outlet - in the center to the right Mounting depth ca. 49mm max. **for PC577V/P,** Wiring outlet PC5x7V - 45° down Mounting depth PC5x7V ca. 94mm max.	 Fig.: Cut out for all 5,7”-devices
Dimensions Cut out W x H (mm) / 203 x 126 6 holes with D 4,5mm **Wiring outlet** at rear view and horizontal mounting **for PC700P, HMI 700P,** Wiring outlet - RJ45: centered to the right - VCC-connector to the bottom **for PC710T, HMI710T** Wiring outlet - VCC / 2xETH to the right - 2x RJ45 (PN) up Mounting depth ca. 49mm max **for PC709P, PC717T** Wiring outlet - 45° down Mounting depth ca.95mm max.	 Fig.: Cut out for all 7”-devices

S7-Panel-PLC Cut out for switching cabinets

Device	Cut out for devic(es)
Dimensions Cut out (W x H / mm) 262 x 165,5 10 holes with D 4,5mm **Wiring outlet** at rear view and horizontal mounting **for PC1000P, HMI 1000P,** Wiring outlet - RJ45: centered to the right - VCC-connector to the bottom Mounting depth ca. 49mm max **for PC1010T, HMI 1010T,** Wiring outlet - VCC / 2xETH to the right - 2x RJ45 (PN) up Mounting depth ca. 47mm max **for PC1011P, PC1017T** Wiring outlet - 45° down Mounting depth PC1011P ca.95mm max. Mounting depth PC1017T ca.98mm max.	 Fig.: Cut out for all 10,1” and 10,2”-devices
Dimensions Cut out (W x H / mm) 203 x 126 10 holes with D 4,5mm **Wiring outlet** at rear view and horizontal mounting **for PC1560T, HMI 1560T,** Wiring outlet - VCC / 2xETH to the right - 2x RJ45 (PN) up Mounting depth ca. 55mm max **for PC1567T** Wiring outlet - 45° down Mounting depth ca.104mm max.	 Fig.: Cut out for all 15,6”-devices

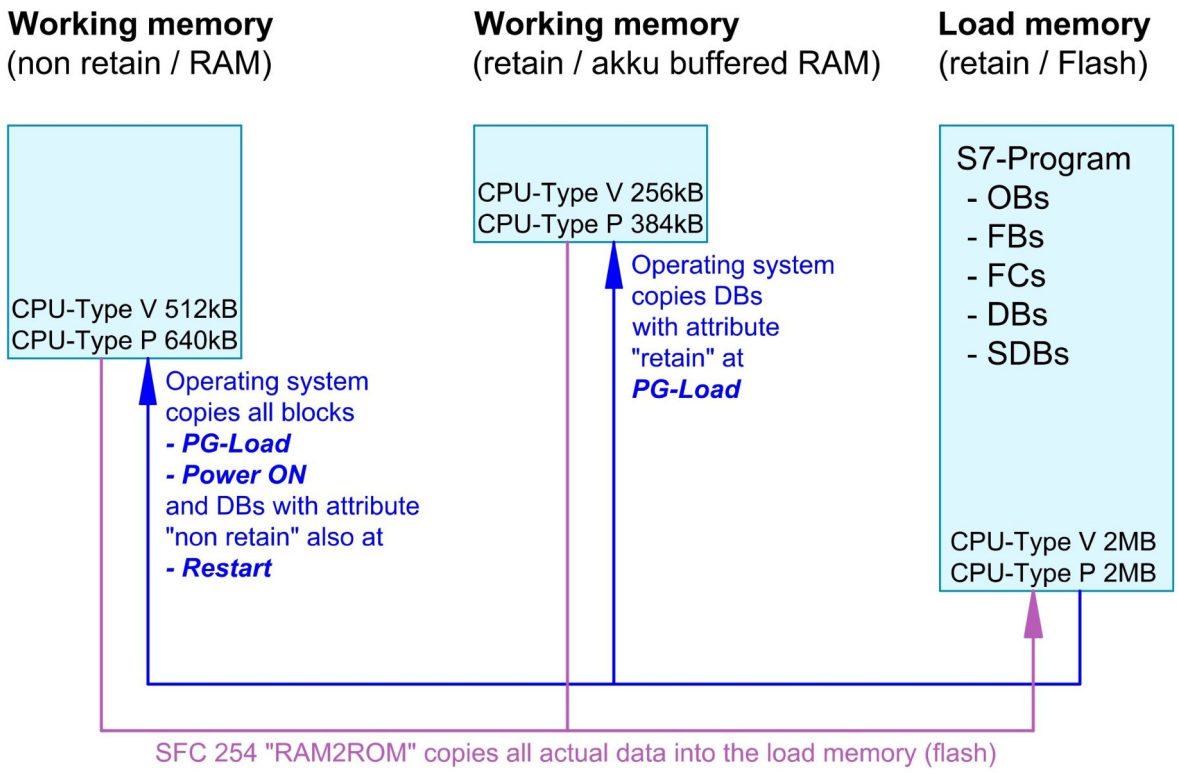
Memory organization

The memory organization matches in general to the Siemens S7-CPUs. 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 accu 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.



Ethernet

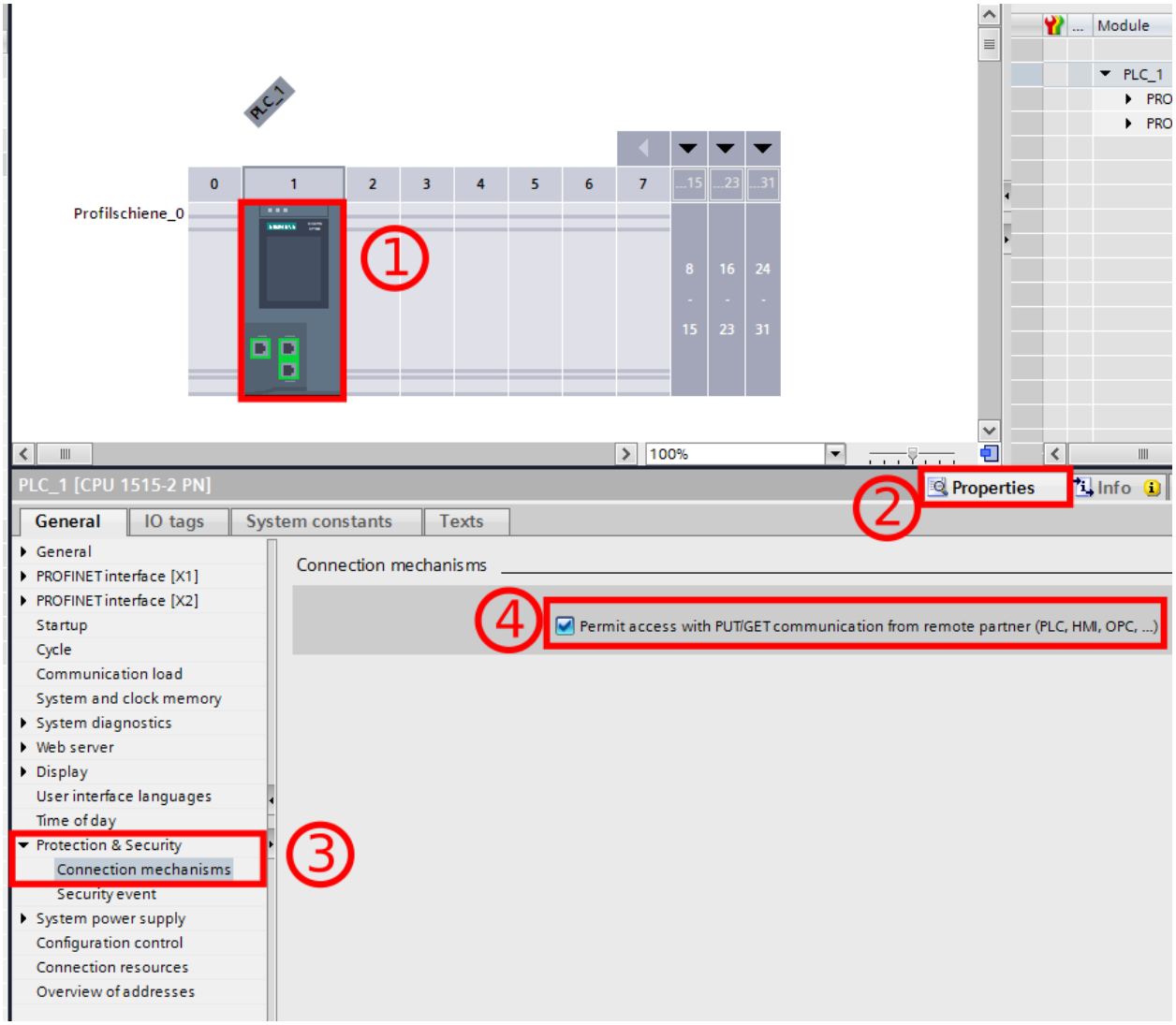
The connection to the Simatic®-Manager is done by Ethernet TCP/IP. Thereby the PLC will be identified by an own IP-address.

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

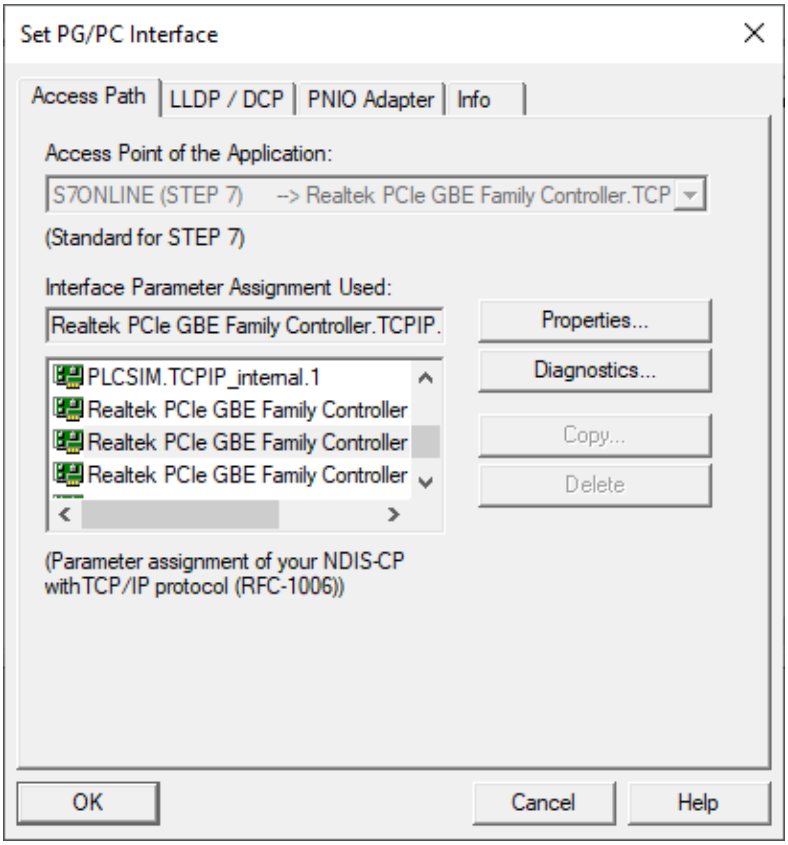
PC with Simatic®-Manager from
Version 5.4 SP5 (PG)
(better use V5.5 or V5.6)
INSEVIS-PLC (AG)
Power supply 24V
Patch cable or crossover
patch cable

Procedure

connect PG with AG
Power on AG
Start Simatic® Manager

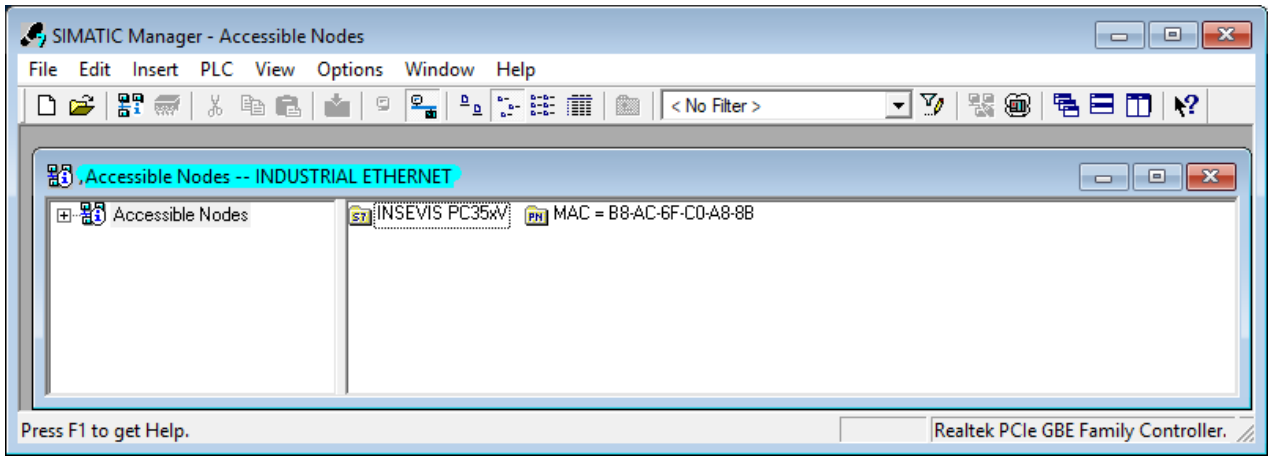
Set up PG/PC-interface

→ 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

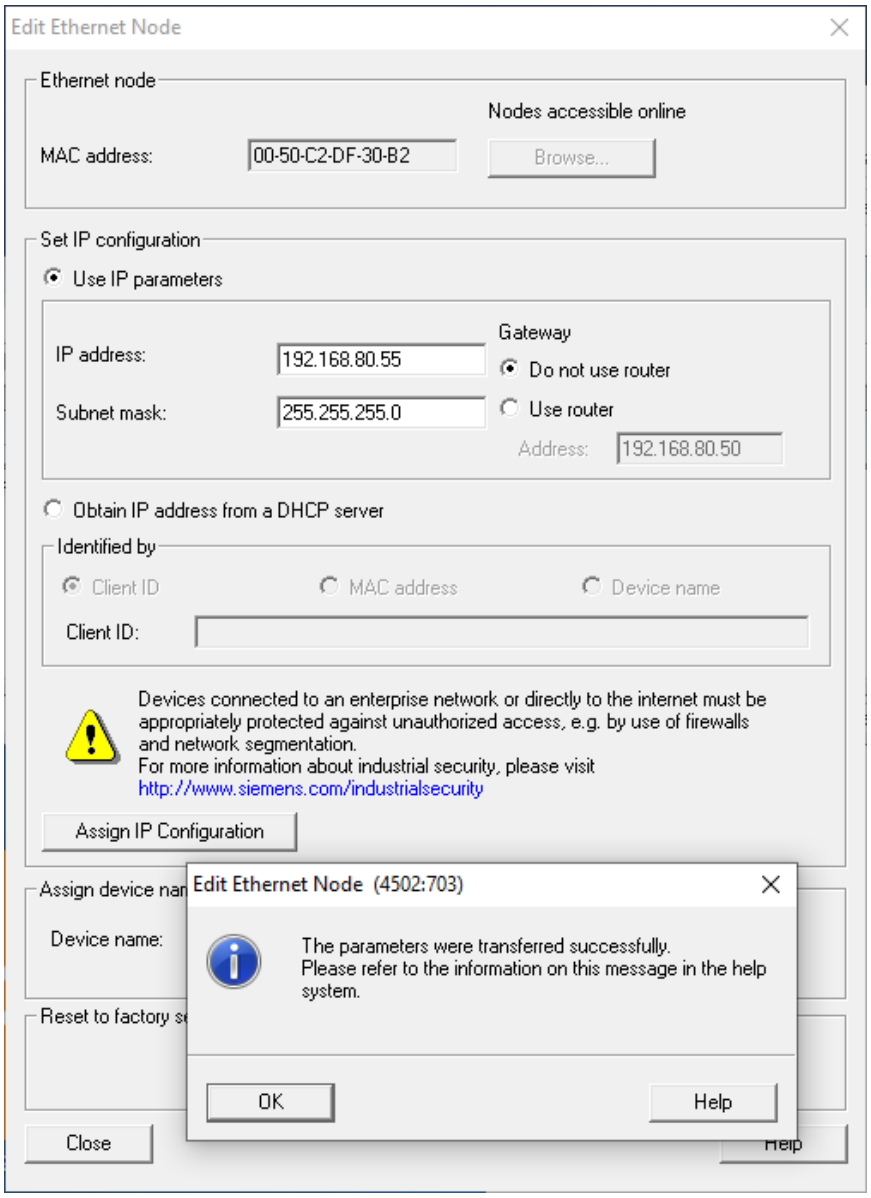


Communication - Ethernet

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 mus 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

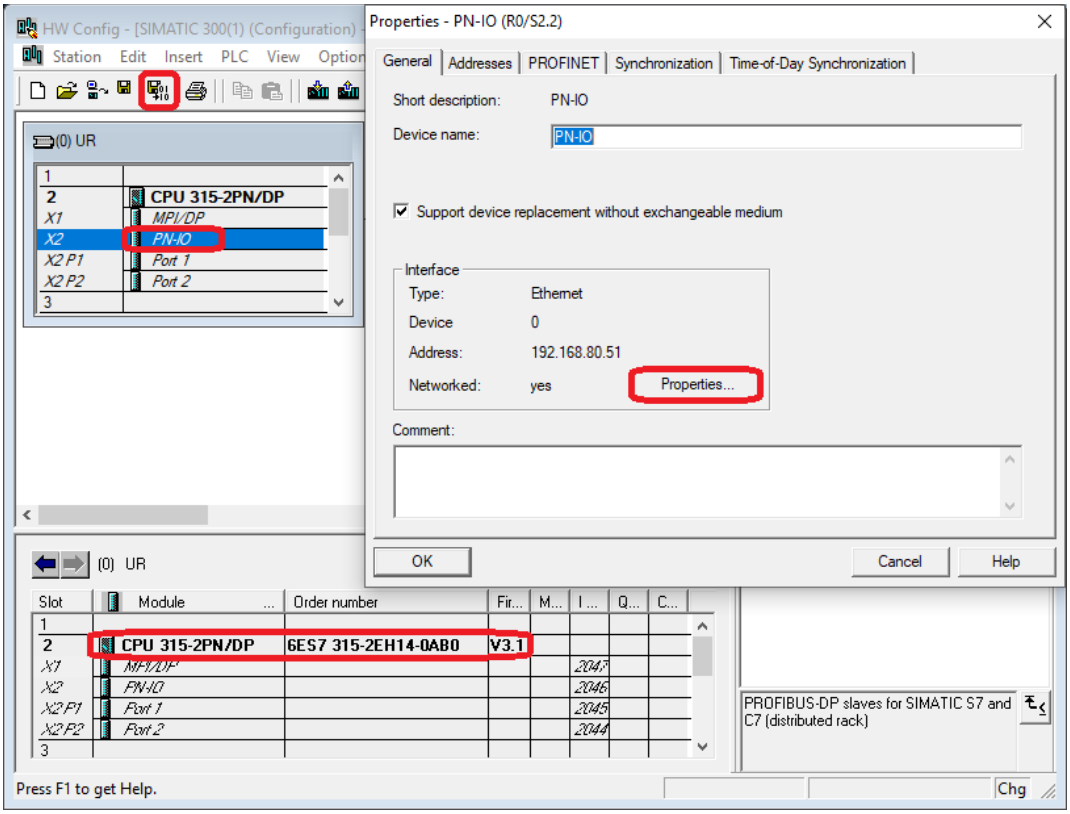
Create the INSEVIS-CPU in the Simatic® -Manager

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.

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

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: • 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
```

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

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: • 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
```

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                  // if successful done
      BEB                               // end the function
      L     MW  1000                    // error occurred
// error evaluation
// ...
      BEA                               // end of function
```

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                  // if successful done
      BEB                               // end the function
      L     MW  1000                    // error occurred
// error evaluation
// ...
      BEA                               // end of function
```

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 "TRECV" 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.

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
```

Receiving data via TCP with SFB 123 "TRECV"

The SFB 123 “TRECV” 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)

S7-program sample for using the SFB 123

Purpose: Receive data from apartner 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
```

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
```

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
```

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	Job has successfully completed. 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 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 Bit 8-12: Connection status 1 Active establishment is in process 2 Connection terminating 3 Connection established/Connected with Partner Parameter PORT, IPADDR contains partner port number and IP address.
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

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.

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.

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

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
```

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

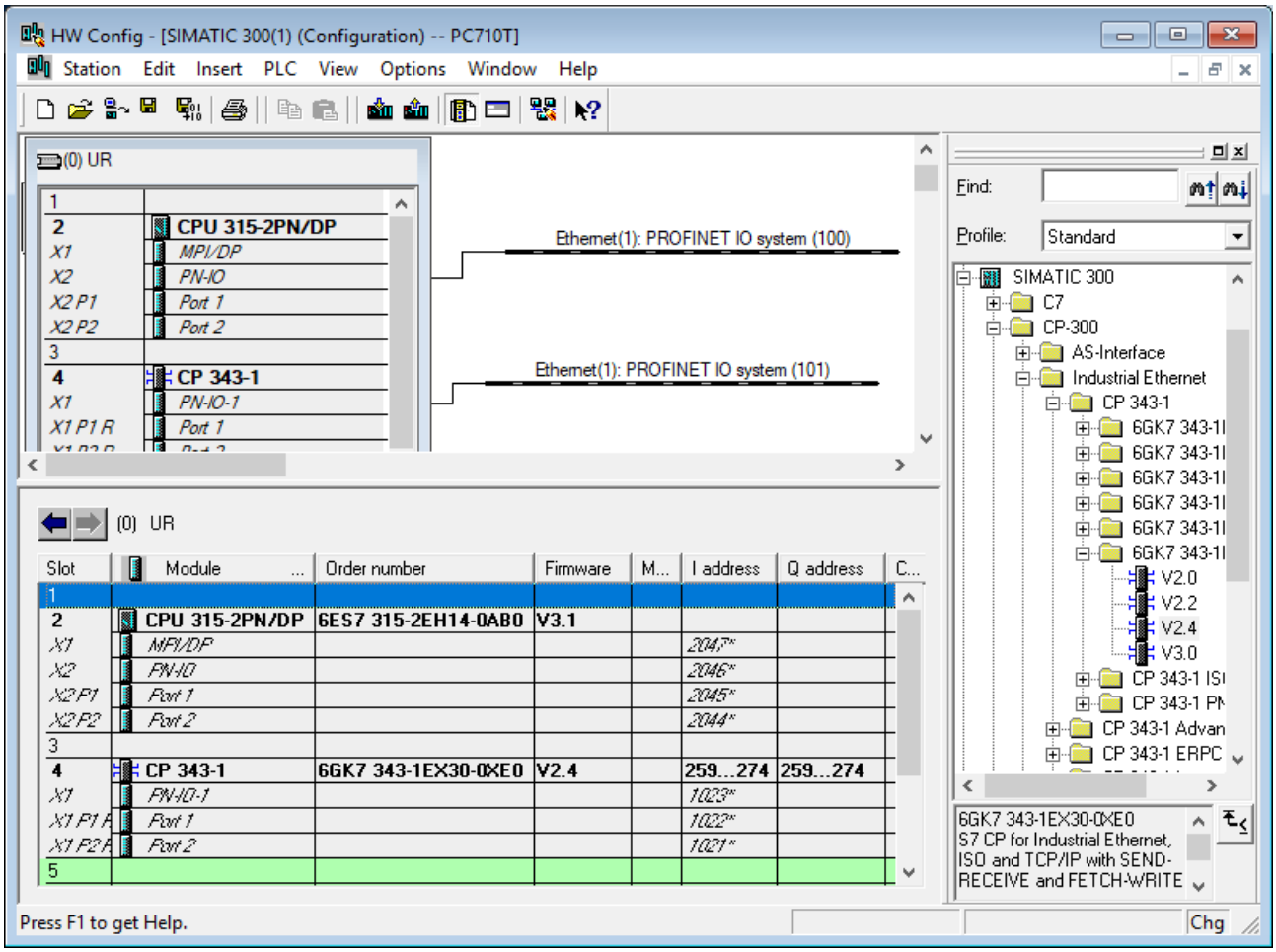
When Profinet-option is included, there are 2 additional RJ45-interfaces (Profinet 1 and 2) visible.
These plugs can be used as switch with common IP-adress.

Configuration with SimaticManager

Profinet is available as an additional option to the CPU-T. To realize this option while configuring in the SimaticManager 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).

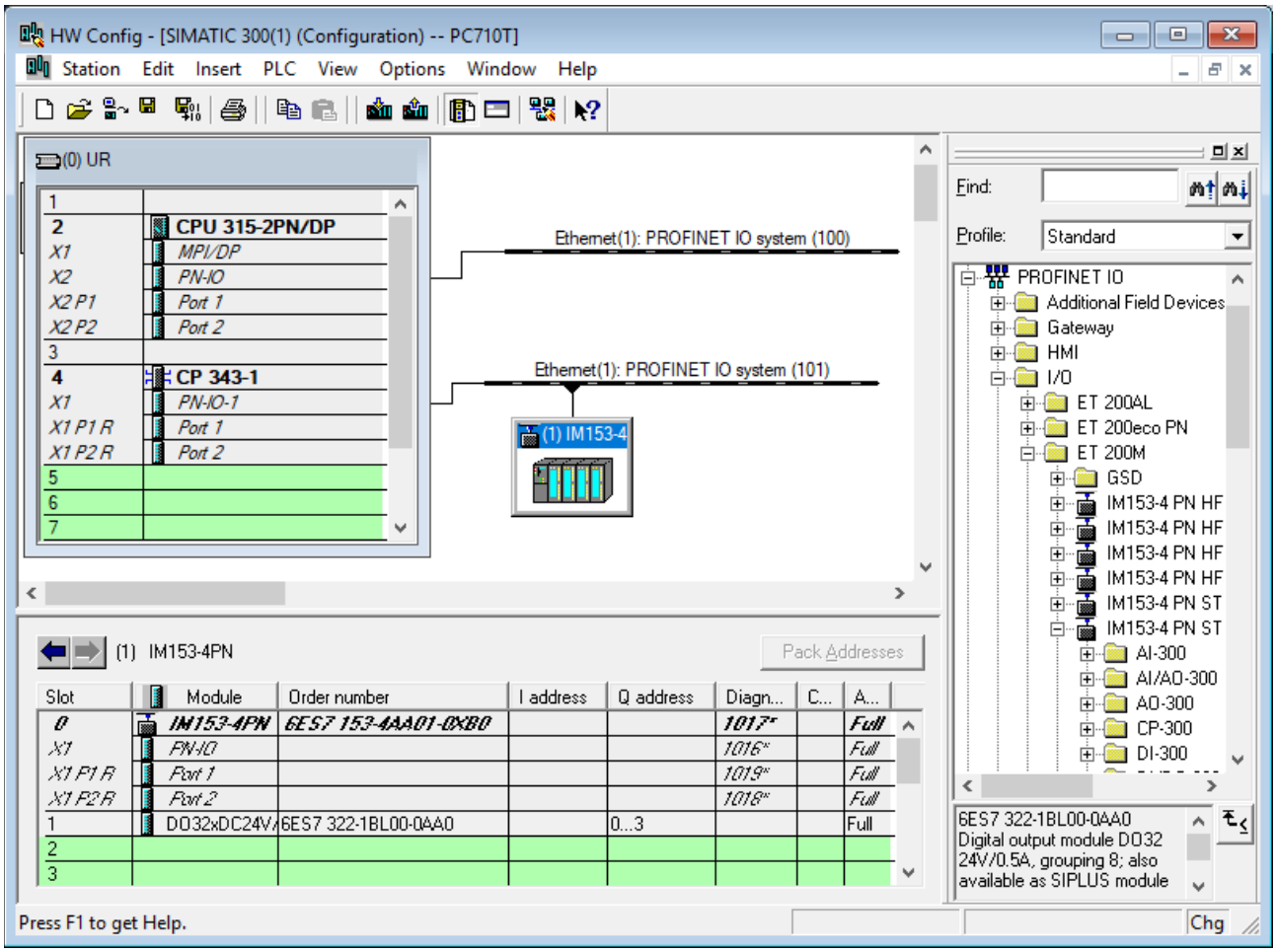
The configuration of the Profinet in INSEVIS-projects is simple:

- 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

Than you can select your Profinet-devices and add it by drag´n drop as you are used to 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:

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

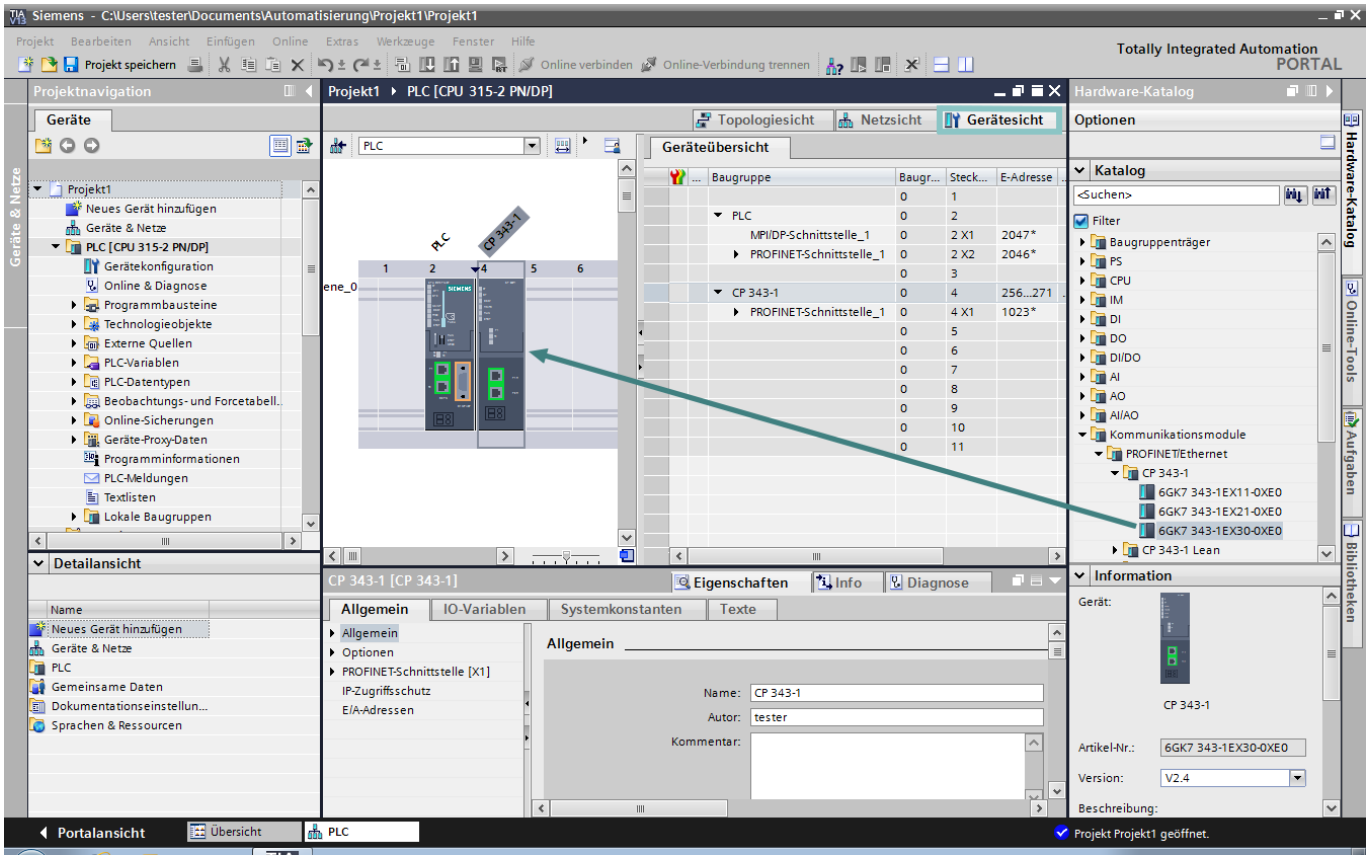
Configuration with TIA-Portal

To realize this option while configuring in the TIA-Portal, it is also 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:

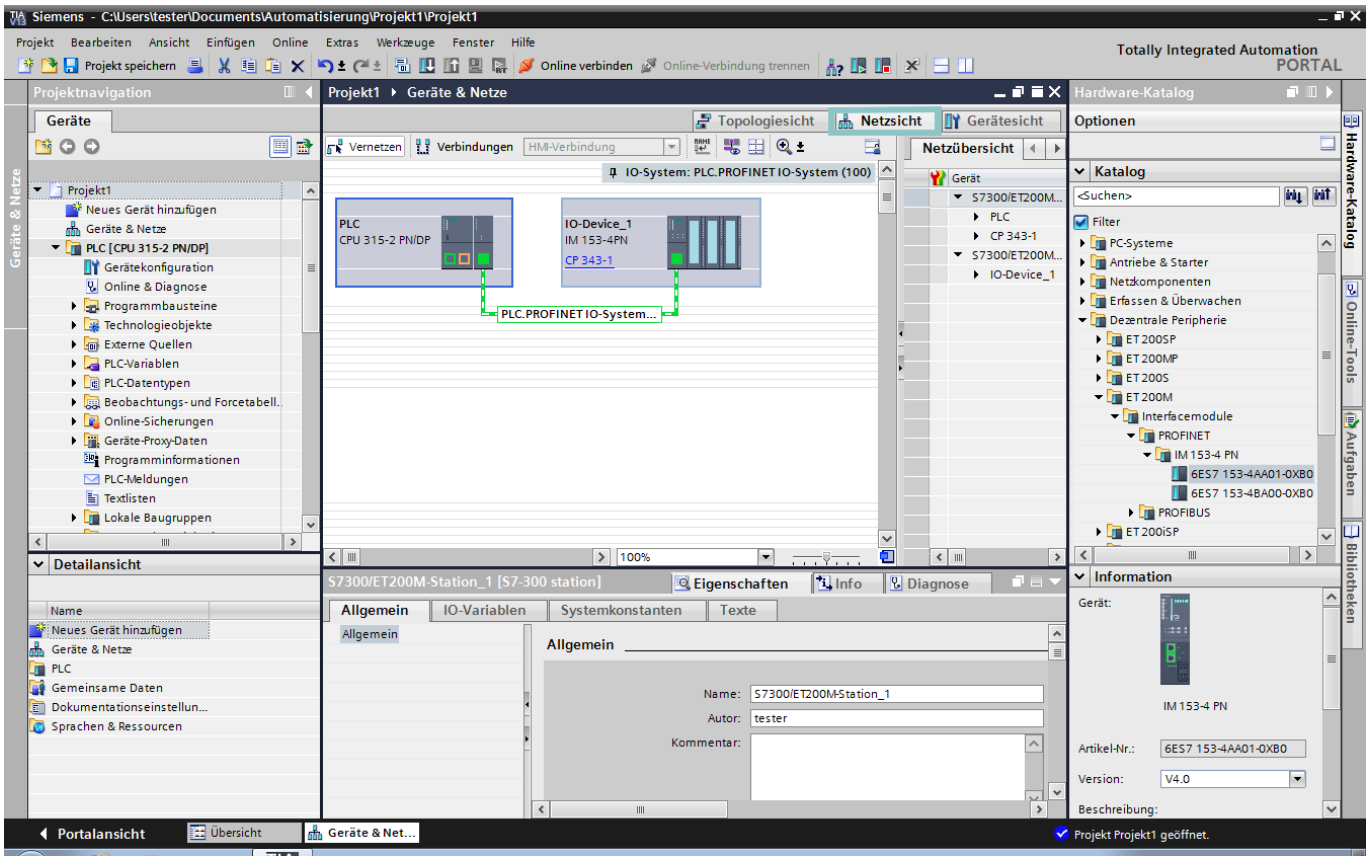
Select the **DEVICE VIEW**

- 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.

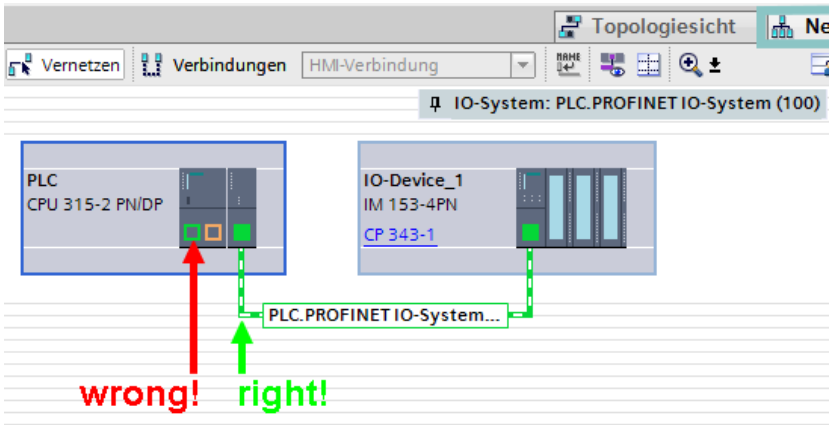


Communication - Profinet IO Controller

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



Set up connection as shown below:



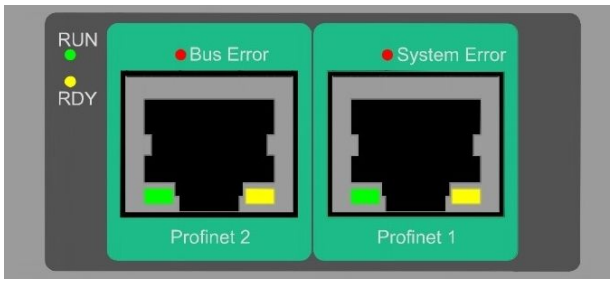
Despite of INSEVIS-Profinet-interface is configured like an additional CP343-1, the handling for S7-programming is like the onboard-Profinet on the S7-315-2 PN/DP.

This means in detail, that:

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

Status-LEDs

Below the description of the Status-LEDs:



LED	Color	LED-Status	Description
RUN	green	on	And RDY (yellow): off Profinet system running
		off	And RDY (yellow) off: Profinet system down (no power, defect) And RDY (yellow) blinking: Profinet system error: Boot Error And RDY (yellow) on: Profinet system error: Boot Loader Waiting
RDY	yellow	blinking	And RDY (green) off: Profinet system error: Boot Error
		on	And RDY (green) off: Profinet system error: Boot Loader Waiting
		off	And RDY (green) off: Profinet system down (no power, defect)
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 diagonstic 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.

Hint:

- at the INSEVIS Profibus- modules the „Hardware-Online-Diagnostic“ from SimaticManager is not available. That´s why the configured slaves will be displayed in gray.

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

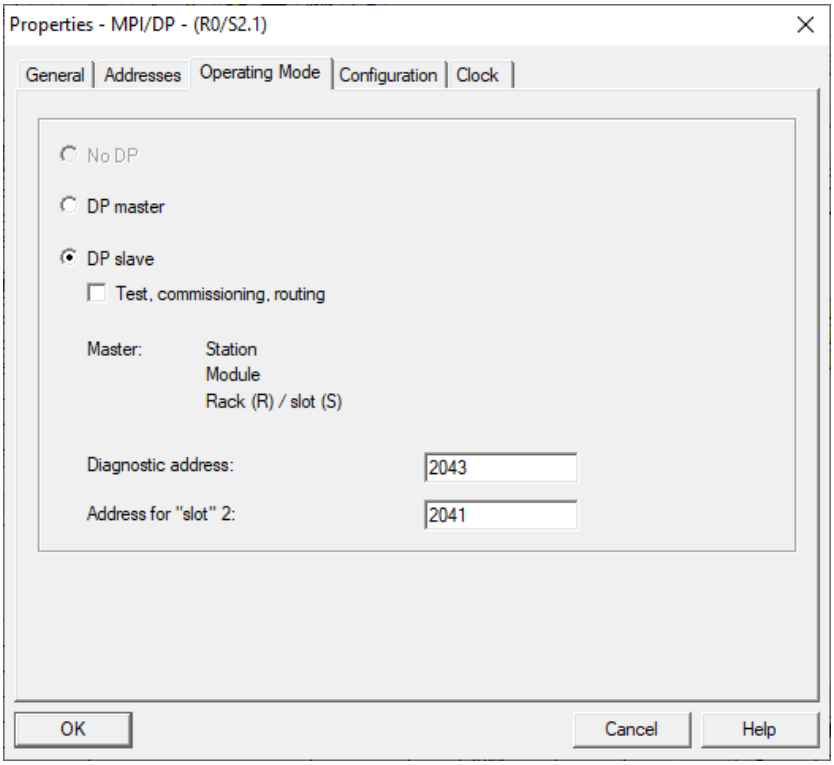
Differences compare to CPU315-2PN/DP:

- 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

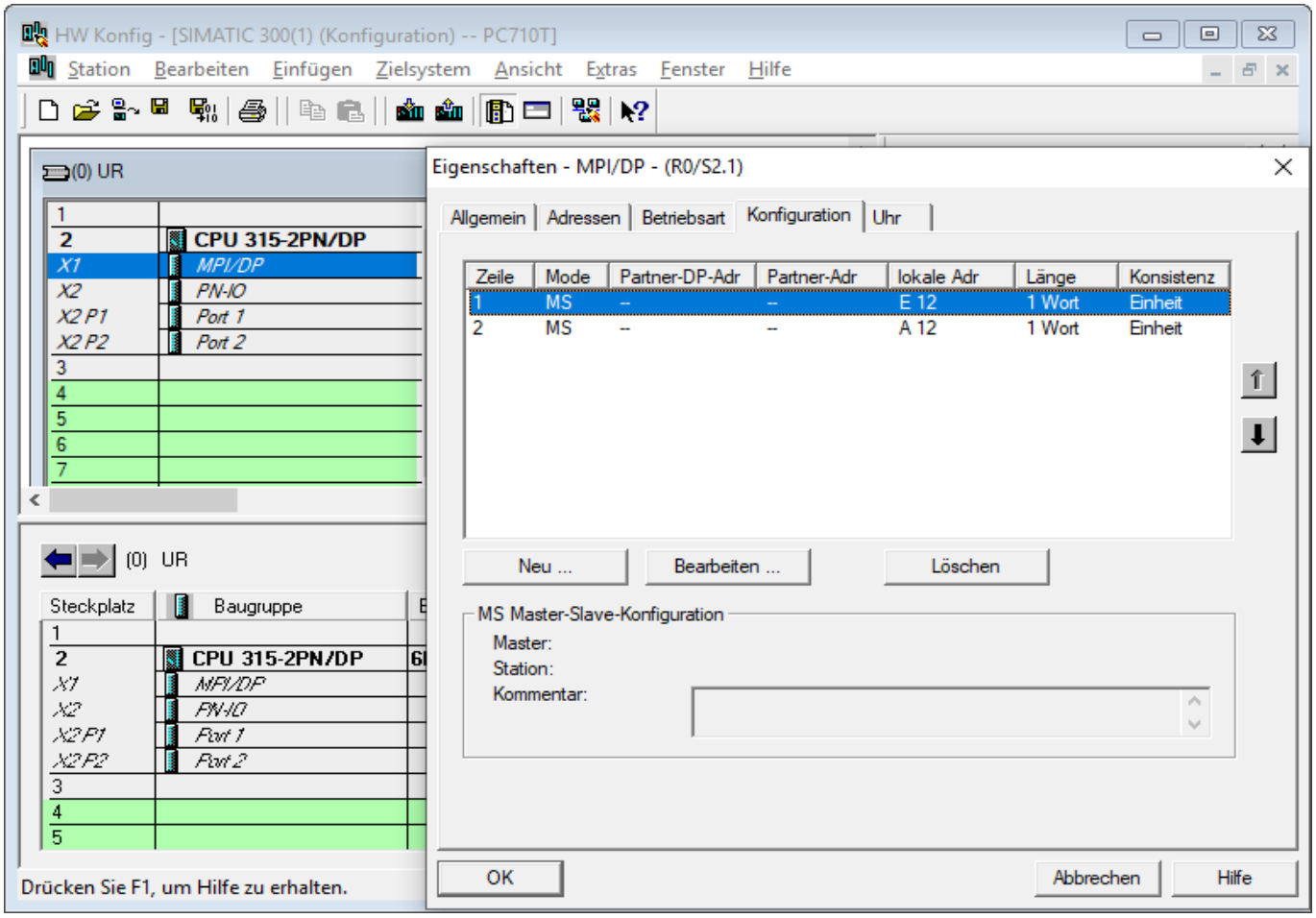
Configure the input / output slots of DP-slave

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:

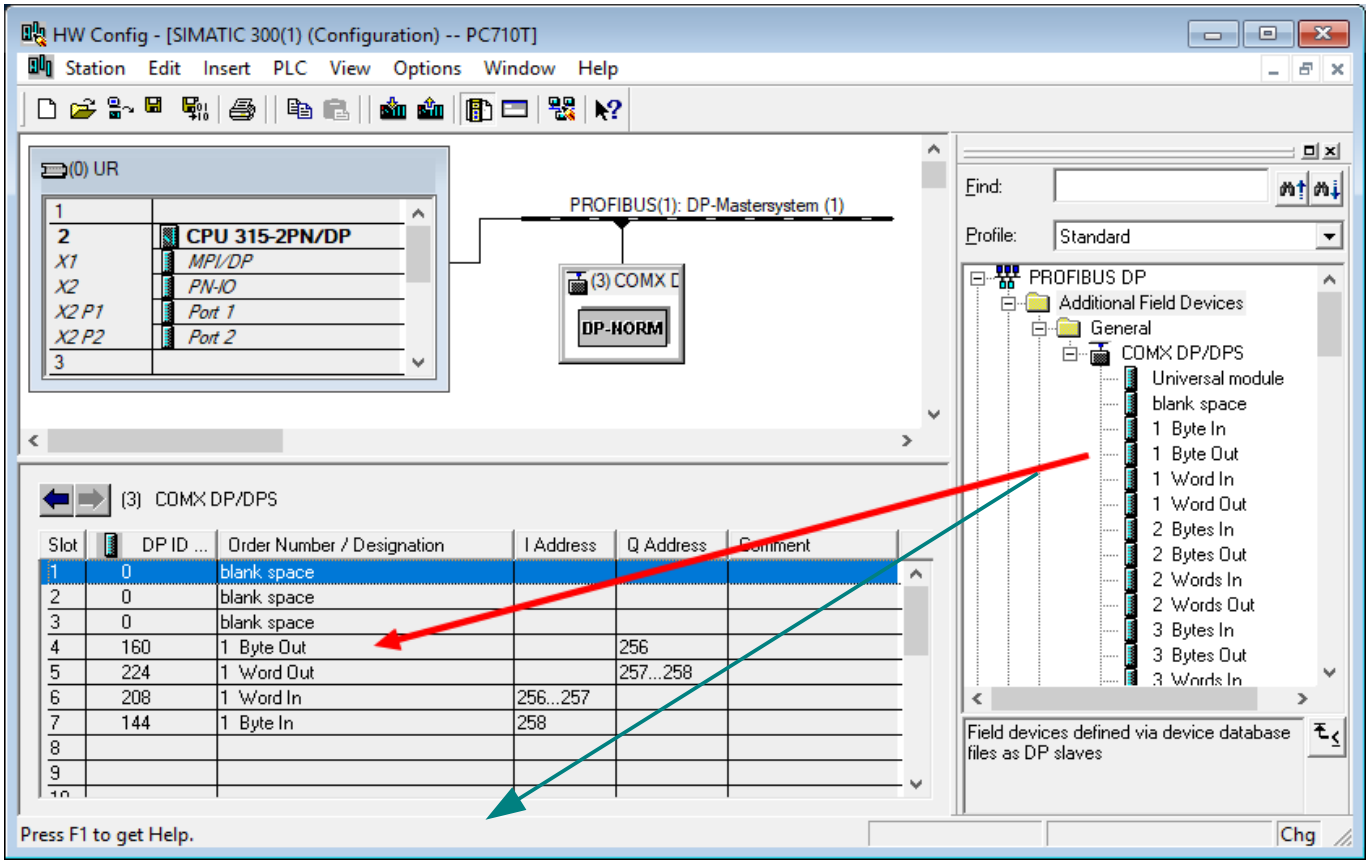
- Maximum 21 virtual slots (rows) can be configured
- "Direct data exchange" (DX) mode is not supported, only Master-Slave configuration (MS) is supported
- In case of DP Master failure (or cable disconnected), and periphery direct access (e.g L PEW 2) to the configured slots causes the periphery access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

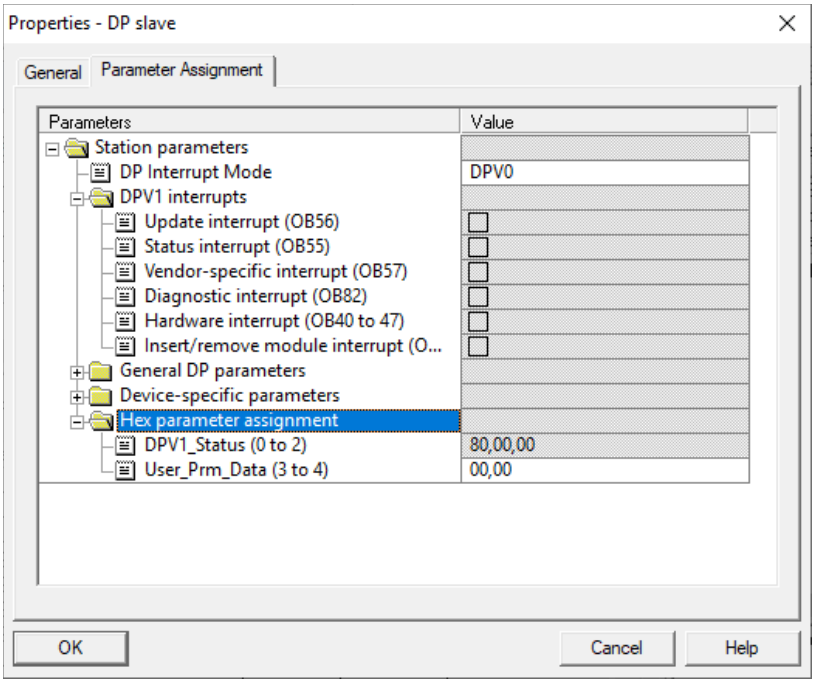
- Drag “COMX DP/DPS” modul from “Hardware Catalog”
  → PROFIBUS DP → “Additional Field Devices” → “General”
- Drop in DP Master system



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)

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 ... Byte x - 1	Module diagnostic (Identification related diagnostics) (length depends on the number of the configured slots)
Byte x … Byte y	Module status (Device related diagnostics) (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
$0C_H$	$0F_H$

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 80C3H 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 80C3H SFC called in lower priority level
BUSY	OUTPUT	BOOL	TRUE=Job is in process

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

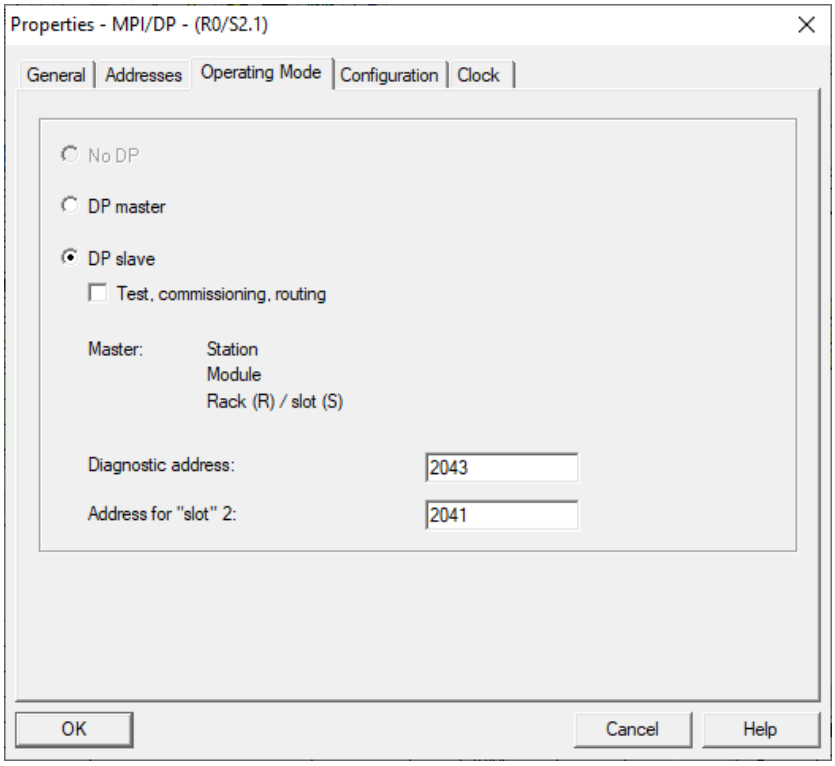
Differences compare to CPU315-2PN/DP:

- 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

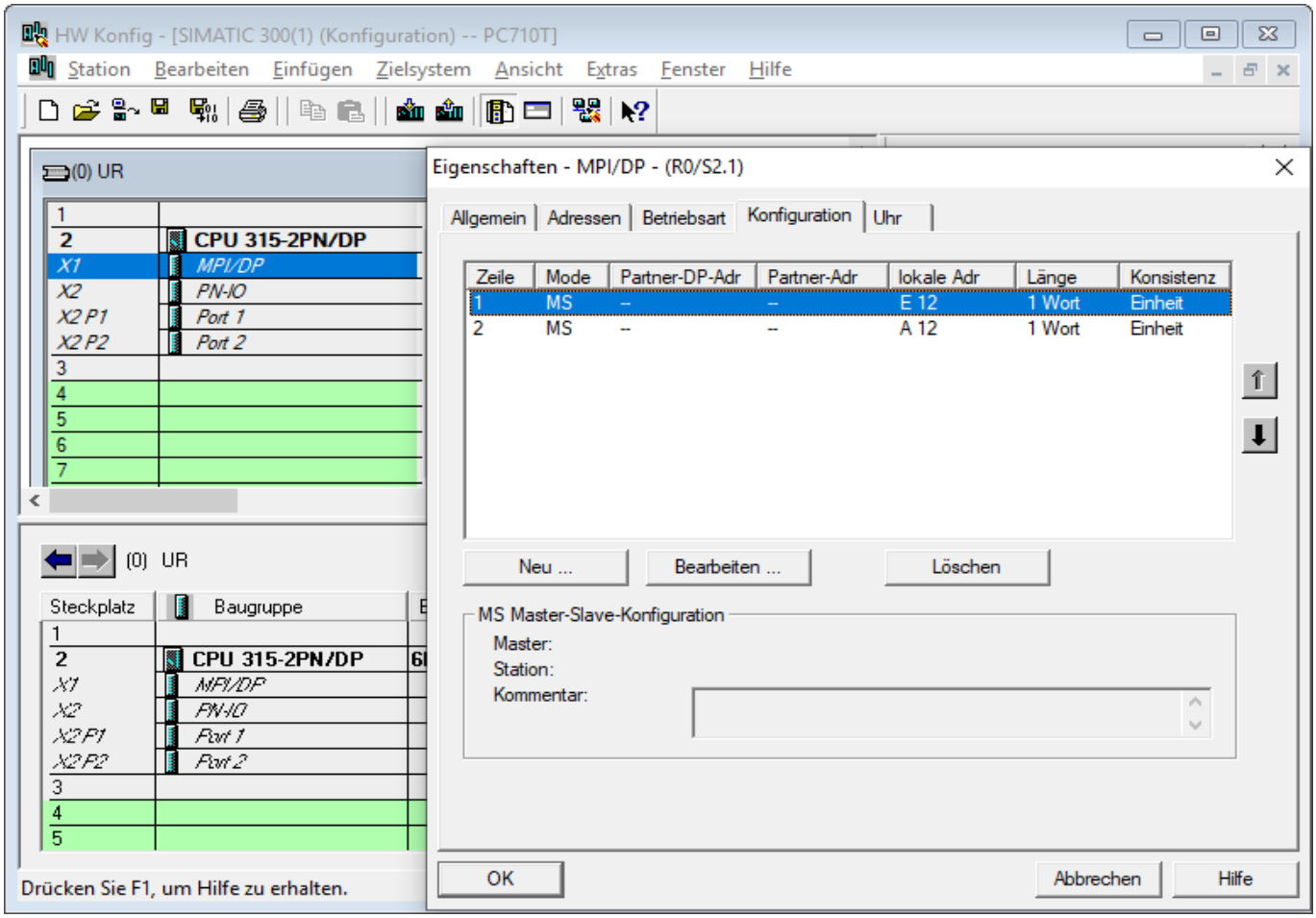
Configure the input / output slots of DP-slave

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:

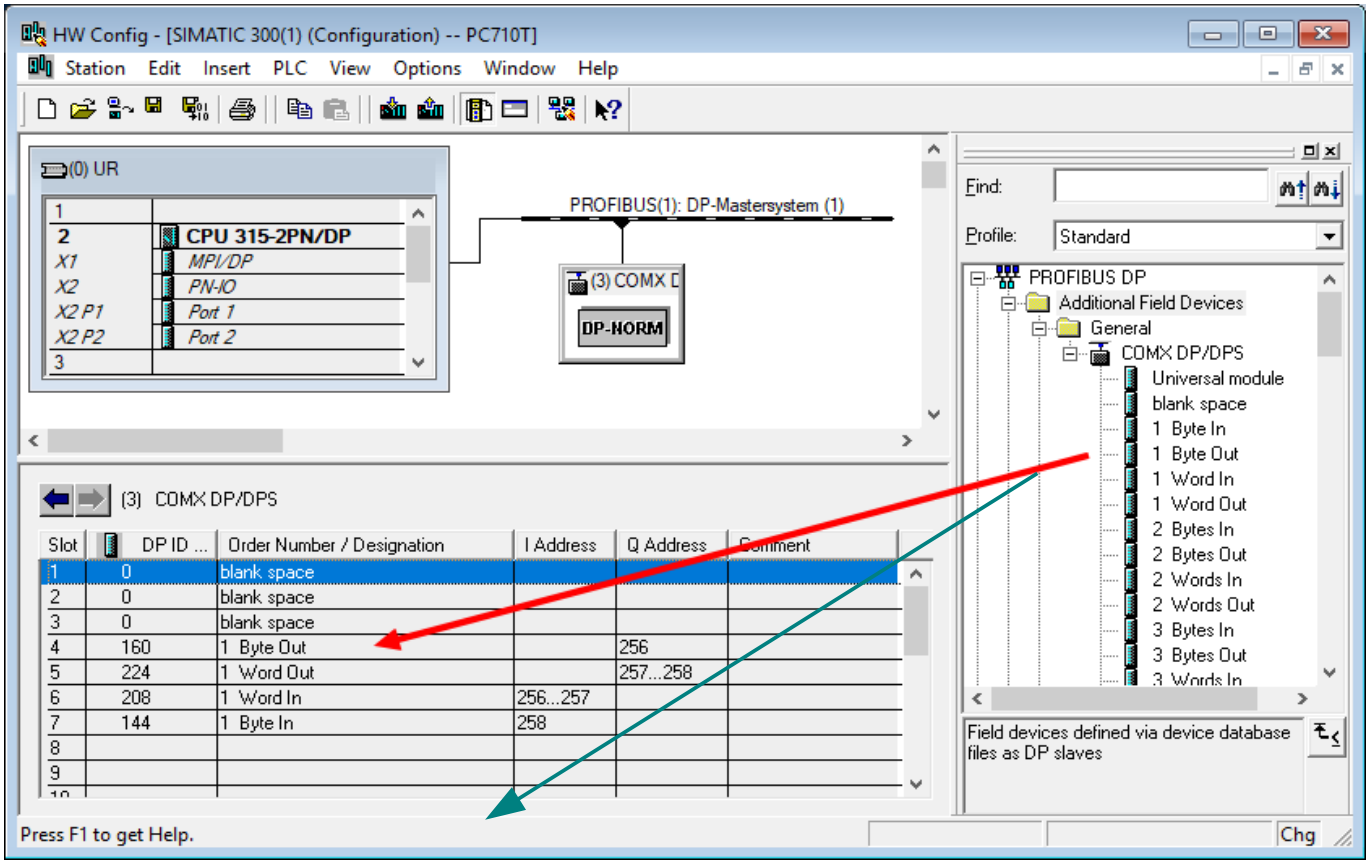
- Maximum 21 virtual slots (rows) can be configured
- "Direct data exchange" (DX) mode is not supported, only Master-Slave configuration (MS) is supported
- In case of DP Master failure (or cable disconnected), and periphery direct access (e.g L PEW 2) to the configured slots causes the periphery access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

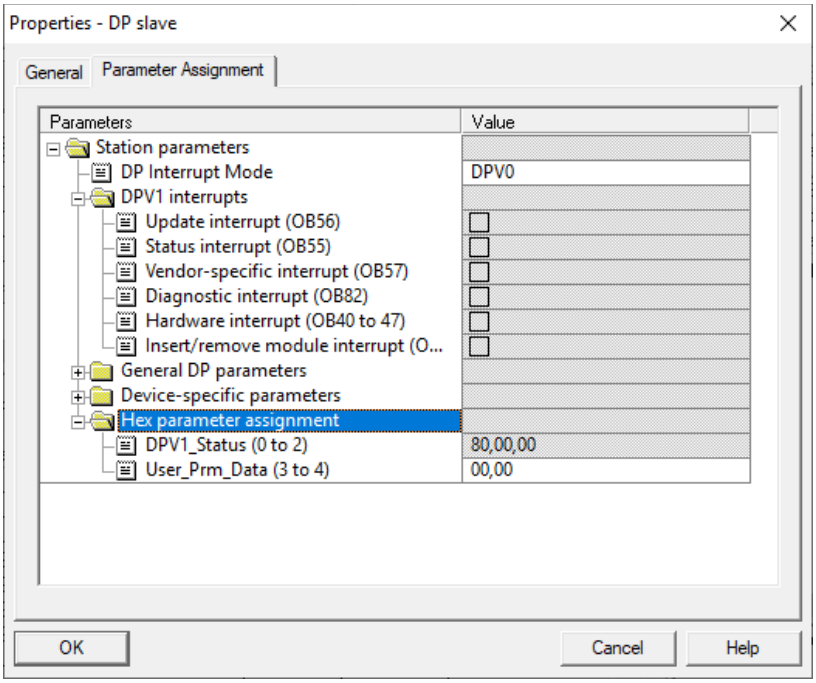
- Drag “COMX DP/DPS” modul from “Hardware Catalog”
  → PROFIBUS DP → “Additional Field Devices” → “General”
- Drop in DP Master system



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)

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 ... Byte x - 1	Module diagnostic (Identification related diagnostics) (length depends on the number of the configured slots)
Byte x … Byte y	Module status (Device related diagnostics) (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
$0C_H$	$0F_H$

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

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 80C3H 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 80C3H 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.

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 interprete 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

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

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.

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

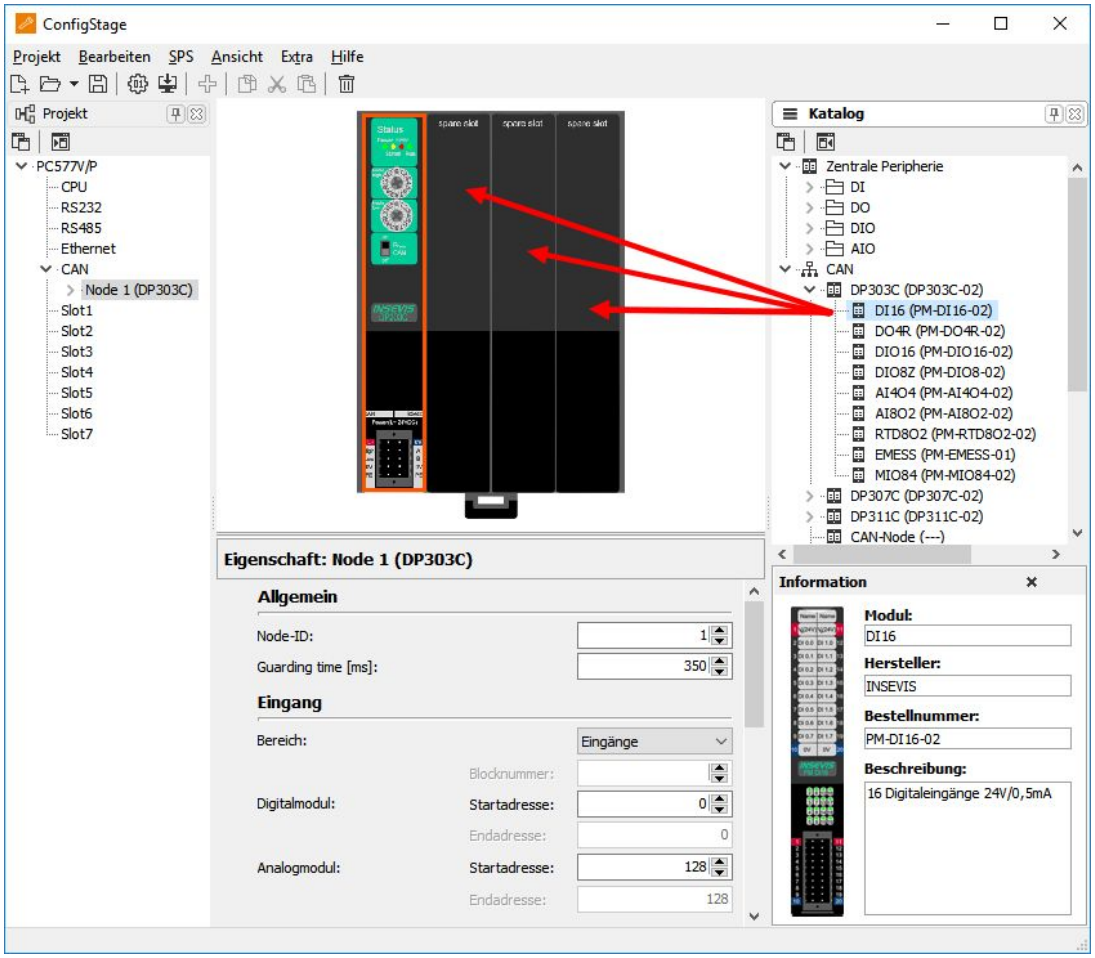
The settings are easy to do in the configuration software ConfigStage and
→ explained in the chapter „ConfigStage“ below the issue „CAN-settings“ in detail.

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

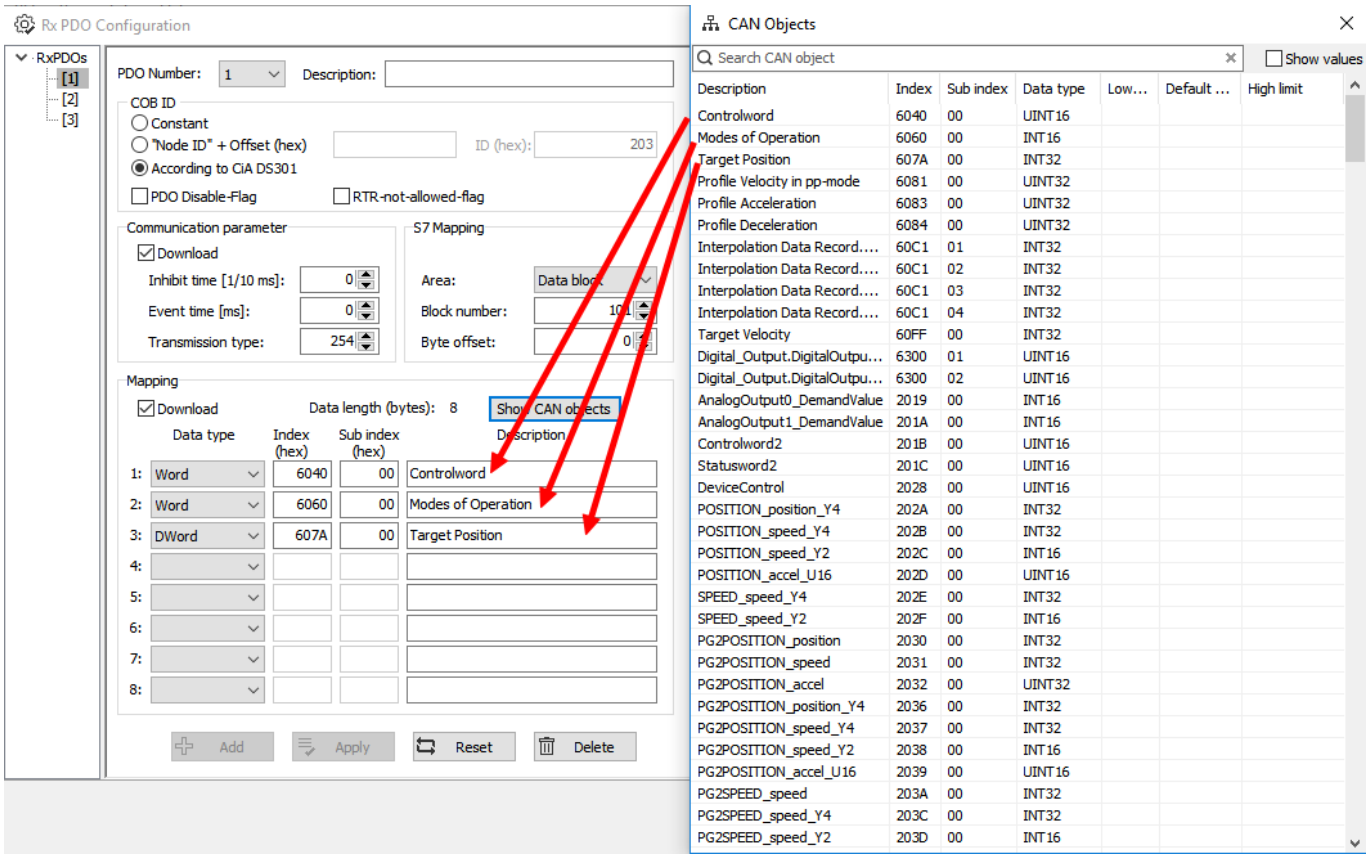
- Type in the node-ID (what are to be set up on hardware hexeadecimal turn-switches on INSEVIS CAN-head station)

- 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.



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



- 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 ind the send buffer and the further cycle does not need to considered.

The return value signalizes the buffer overflow, what normally only occures 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

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

The return value signalizes 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 signalized by a OB106 call.
Thereby can be realized an event controlled communication. Also buffer overflows at cylcle time exceedings and high data volumes can be prevented.

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

FB110 SDO download variable word	FB111 SDO upload variable double word	FB112 SDO download variable double word
Request Node: node-ID Object Subindex Data word Return values: Done, Error, Error code	Request Node: node-ID Object Subindex Data double word Return values: Done, Error, Error code	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“ assignes 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

CAN status request SFB114

SFB114 delivers the instance data block a status word and a list with actual status infoormationfrom 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
```

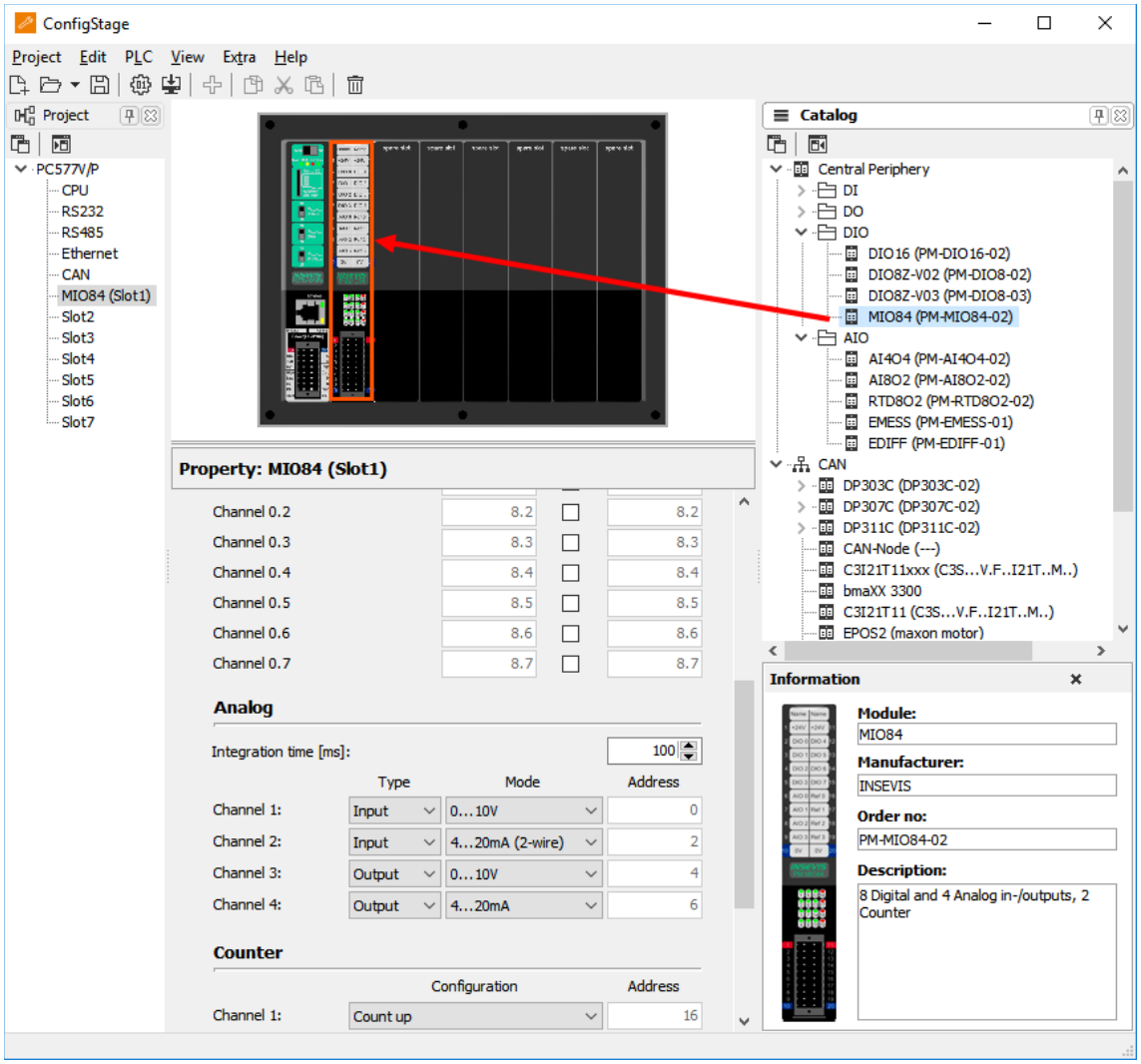
Configuration with the Software „ConfigStage“

This software can be used with operating system Windows 10 only. 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-peripheries 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 wil 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 oft the Simatic®- Manager.

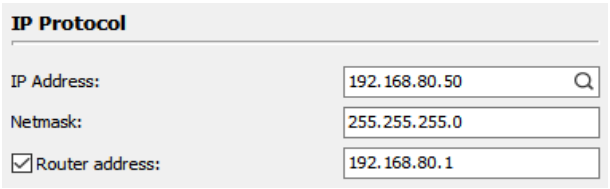
Configuration with the Software „ConfigStage“

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)



or:

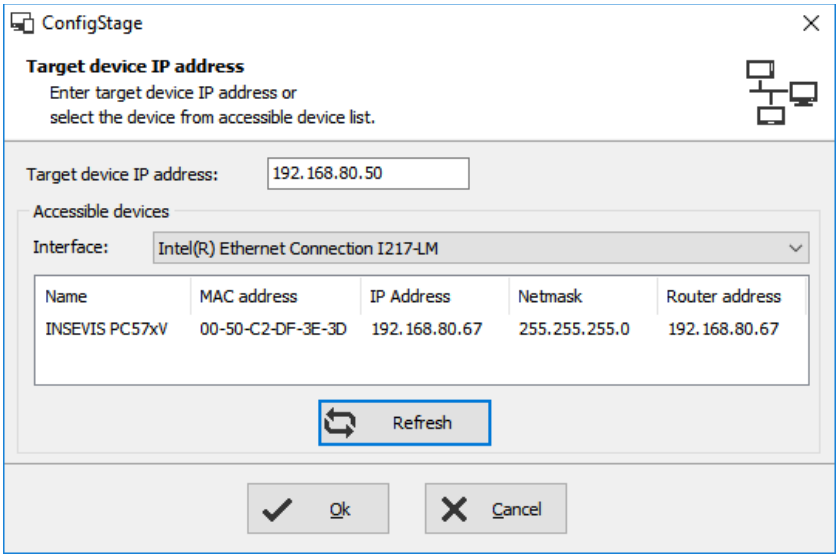
- enter the download-dialog box (F12). (There will be used the service “Accessible devices” like in other INSEVIS-Stages)

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-adress 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.



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

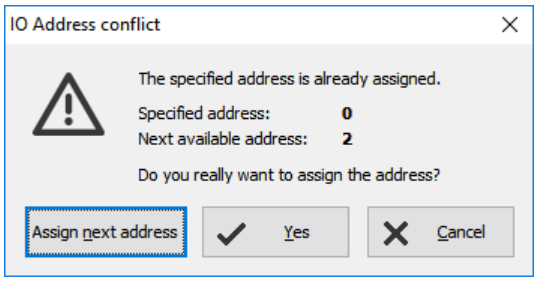
start address \ slot	slot 1	slot 2	slot 3	slot n
digital inputs	byte 0	byte 4	byte 8	byte (n-1)x4
digitale outputs	byte 0	byte 4	byte 8	byte (n-1)x4
analog inputs	byte 128	byte 144	byte 160	byte (n-1)x16 +128
analog outputs	byte 128	byte 144	byte 160	byte (n-1)x16 +128

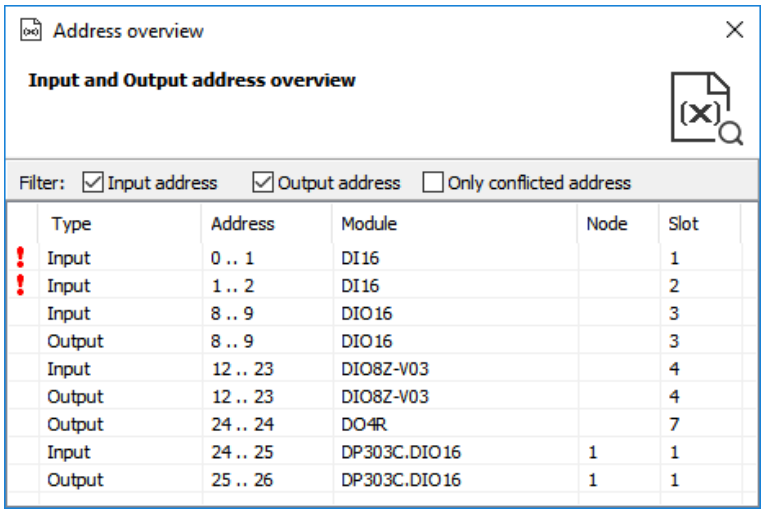
If a functional module FM DIO8-Z is placed, all following digital address areas will be moved for 8 byte more.

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





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

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 menue „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.

CPU settings

The CPU settings are compatible to the Siemens-CPU S7-315-2PNDP 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

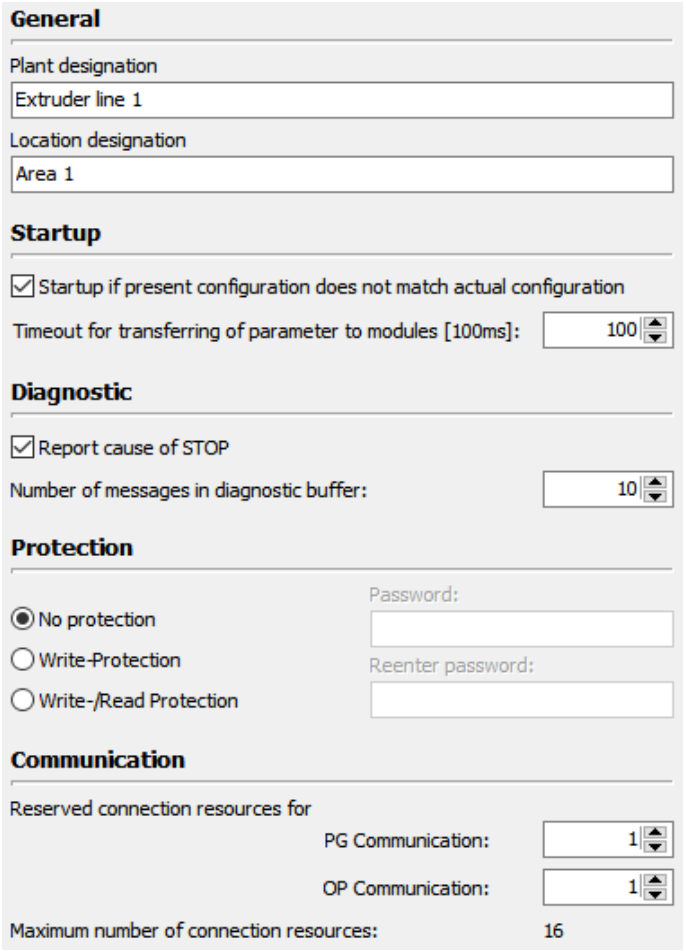
Diagnostic
Box checked: Reports the cause for STOP-condition to the connected devices (PG, host system,...) to be entered in diagnostic buffer too.

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-CPUs 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.

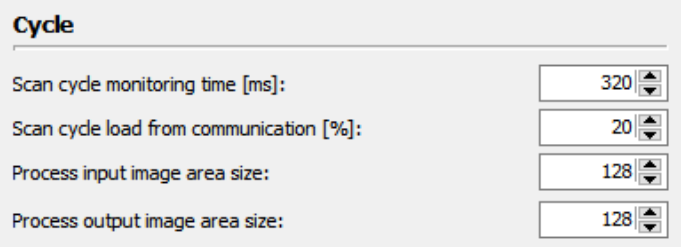
More at **Information about TSAPs**



Configuration with the Software „ConfigStage“

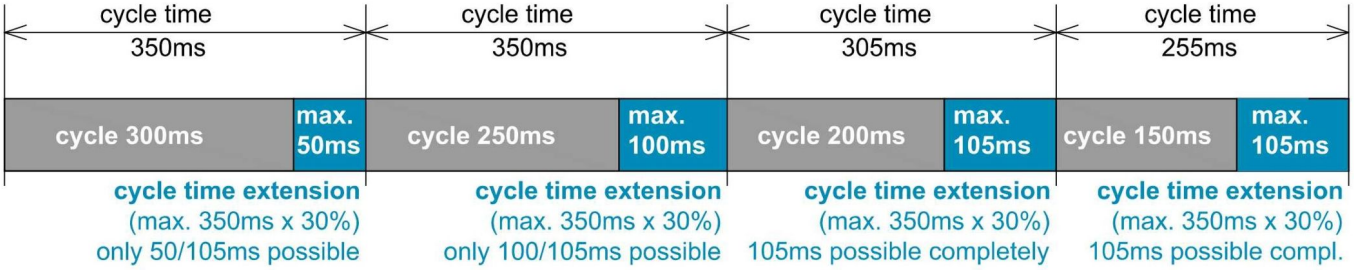
Cycle
Cycle monitoring time:
(Insert it in ms, max. 6000ms = 6 seconds):

Cycle load from communication
(cycle time extension, min. 10% , max. 50%)
for communication (Ethernet, field bus, PG), visualization, file system of Micro-SD®-card)



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).



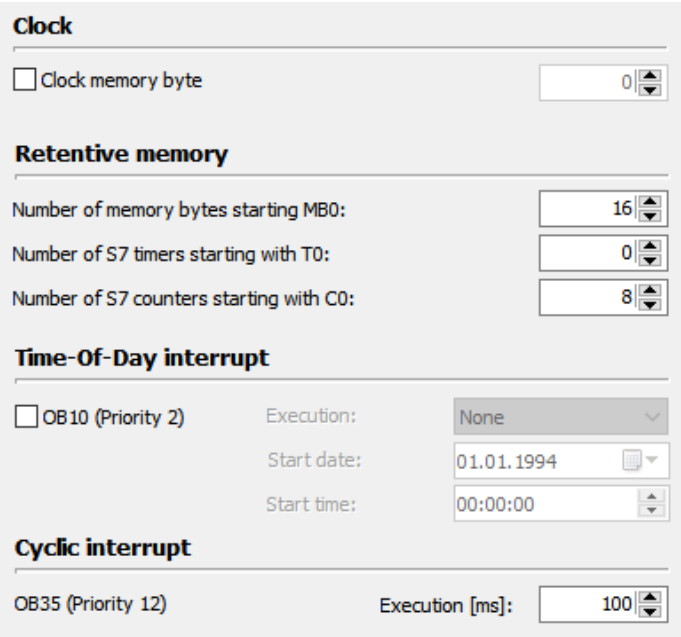
In this sample the cycle time can be extended max. by 30% of 350ms (= 105ms). If this time is not needed for communication (visualization, etc), it is available for other tasks.

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)

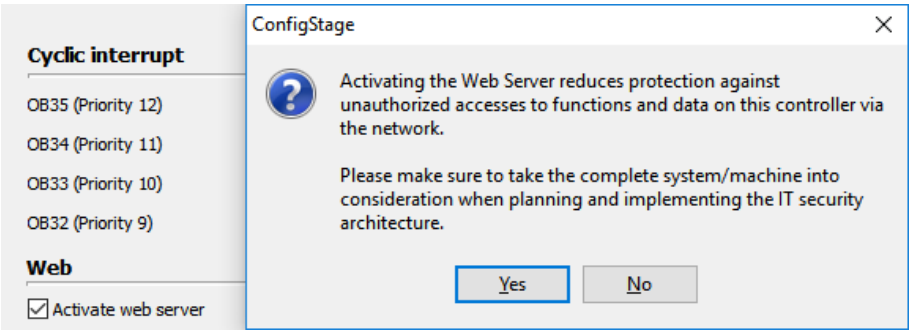


WebServer

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

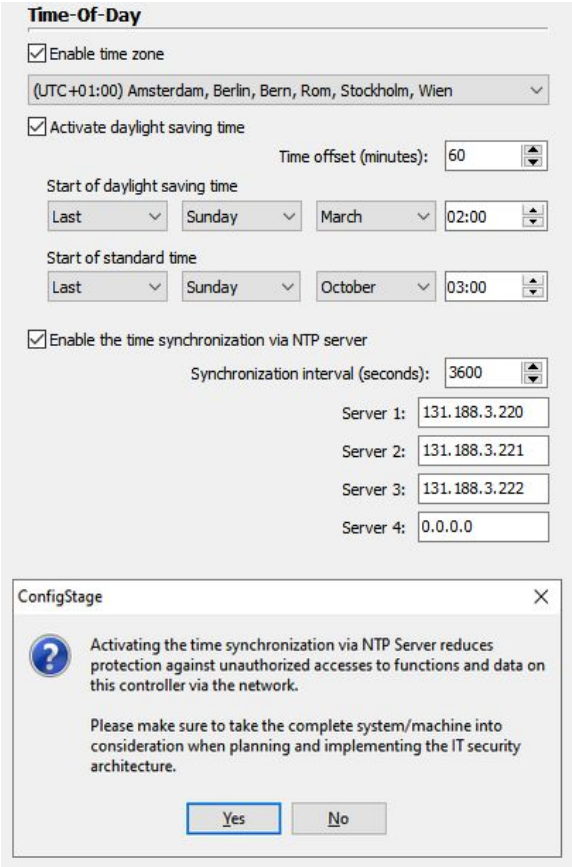
- general data (serial no., etc.),
- actual entries of diagnostic buffer,
- addresses of I/Os and,
- an fiel to require the actual value of a variable.

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 installing 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.

Time settings (time zone, daylight saving time, NTP- server)



For PLCs with CPU-T can be done these system time settings by ConfigStage 1.0.14.41

Activation of a time zone
(important for NTP- usage)

Activation of daylight saving time
The respective changeover dates are entered here. The PLC then changes over automatically.If the unit was switched off during the changeover and the battery has enough float voltage for the real time clock, the unit from firmware 2.7.2 adapts the time during booting process to the present time. If the unit has been switched off for a longer period of time so that the real time clock has become voltage-free, the time must be set manually.

Use of NTP servers
NTP servers send the Universal Coordinated Time (UTC), which is the basis for calculating your own time. In the figure on the left 3 time servers of the FAU University Erlangen-Nuremberg were addressed (131.188.3.220/1/ 2 correspond to the ntp0/1/2 of fau.de)

ATTENTION: In the ConfigStage under "Ethernet", configure the router according to the selected NTP servers (in this case 192.168.80.1)

ATTENTION: 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 installing 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.

Communication settings

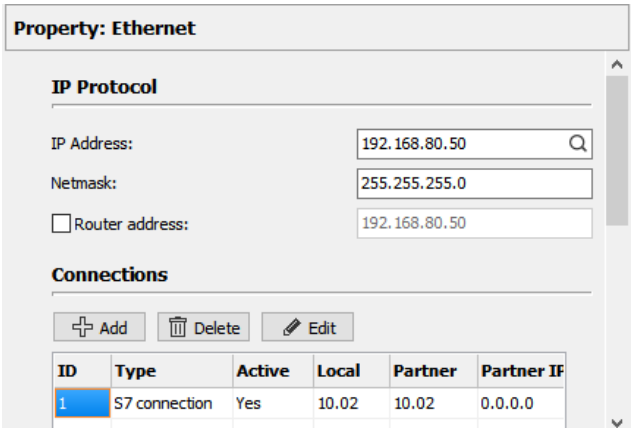
RS232 and RS485

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 sended by SFB60/61

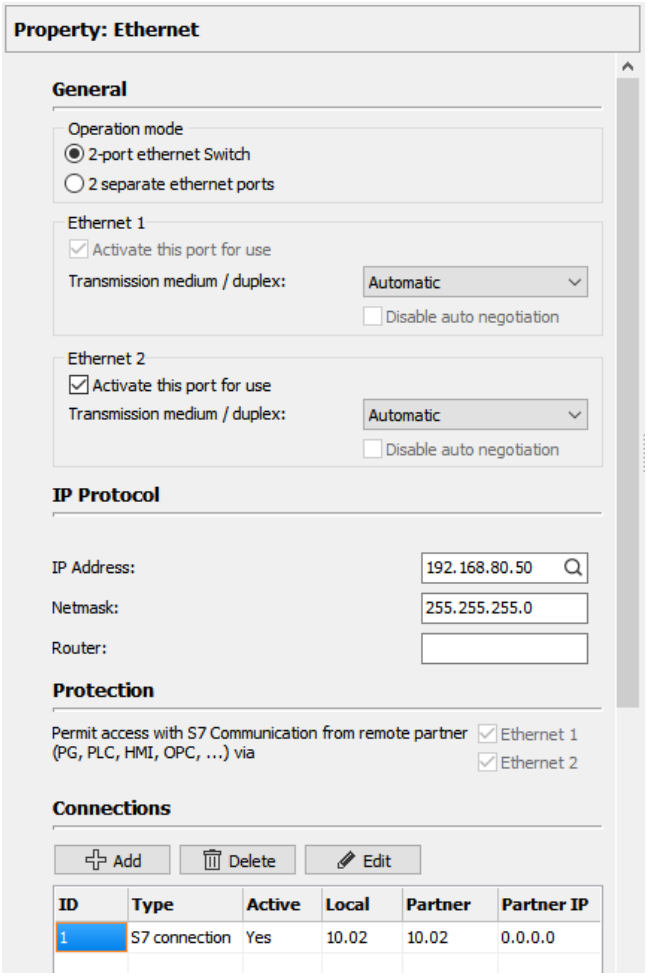
Ethernet

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

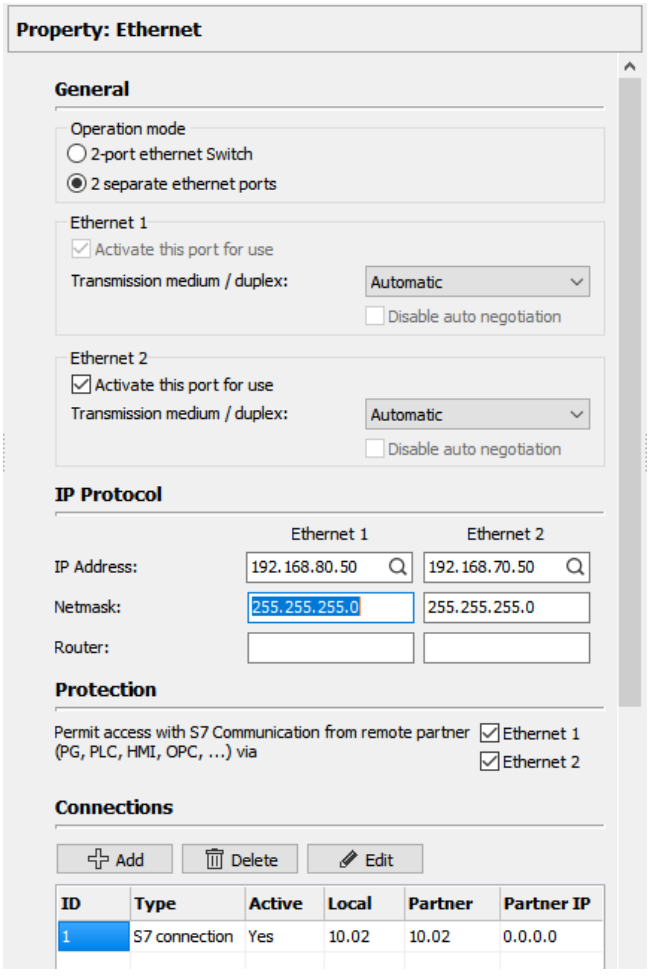
CPU-V/-P: There is ONE Ethernet-interface available (RJ45) (right)



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



Both ports can be used



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:

To set up the **Ethernet-interface** for communication, select here connections and protocol (connection type) and parameterize it *(see right figure)*.

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:

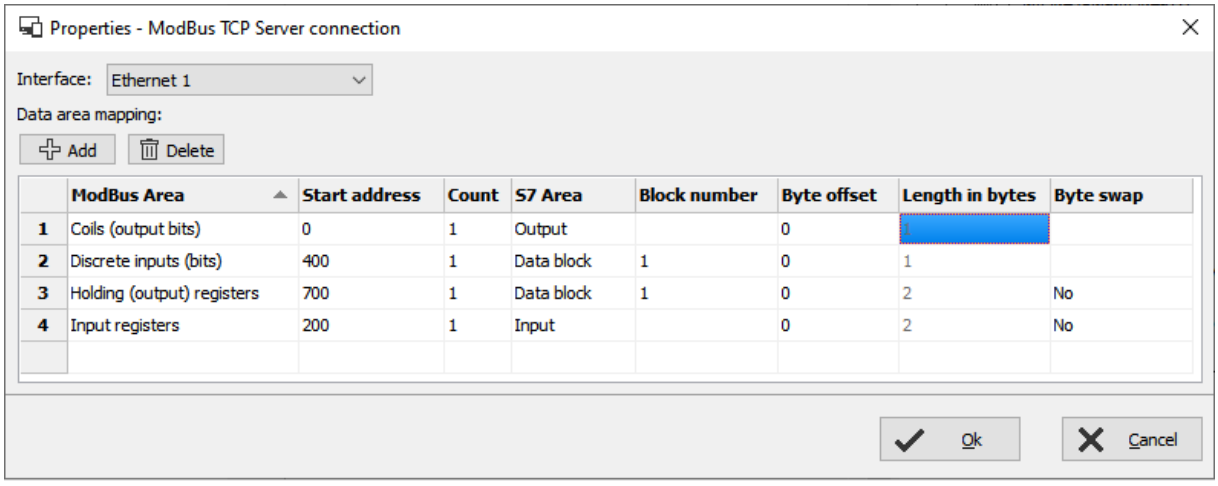
The assign of **RS232 and RS485** is self-explanatory.

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.

parameters at Modbus-TCP (Server) *see below*
Assign of S7-operand areas for
up to 16 Modbus register areas now *(see figure below)*

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

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



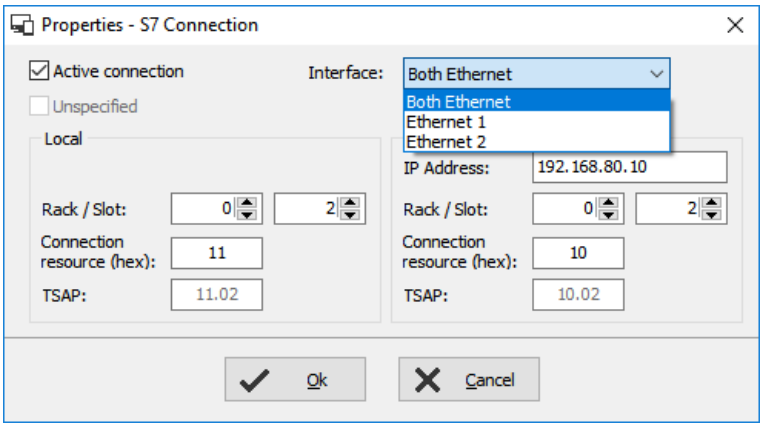
Configuration with the Software „ConfigStage“

Communication / Information about TSAPs

INSEVIS-CPUs 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-CPUs.

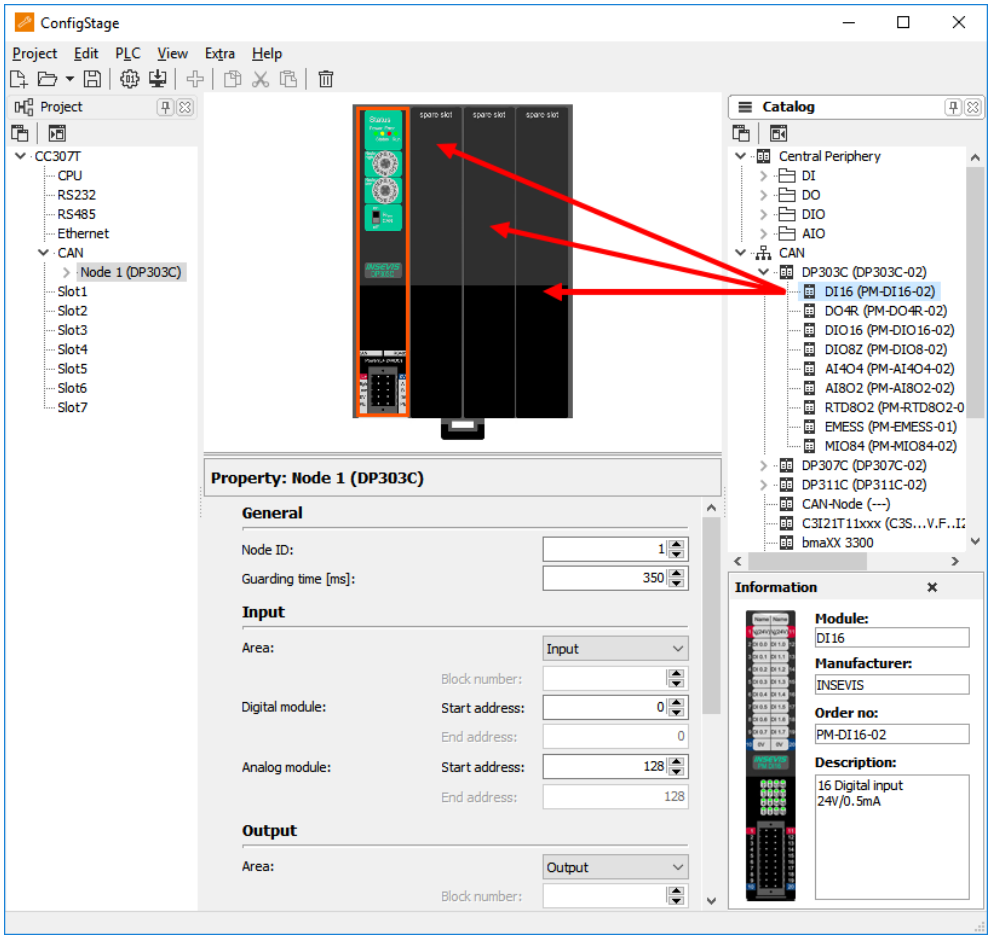
After having placed your INSEVIS head station on the CPU, this module appears in the project tree and in the display.

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

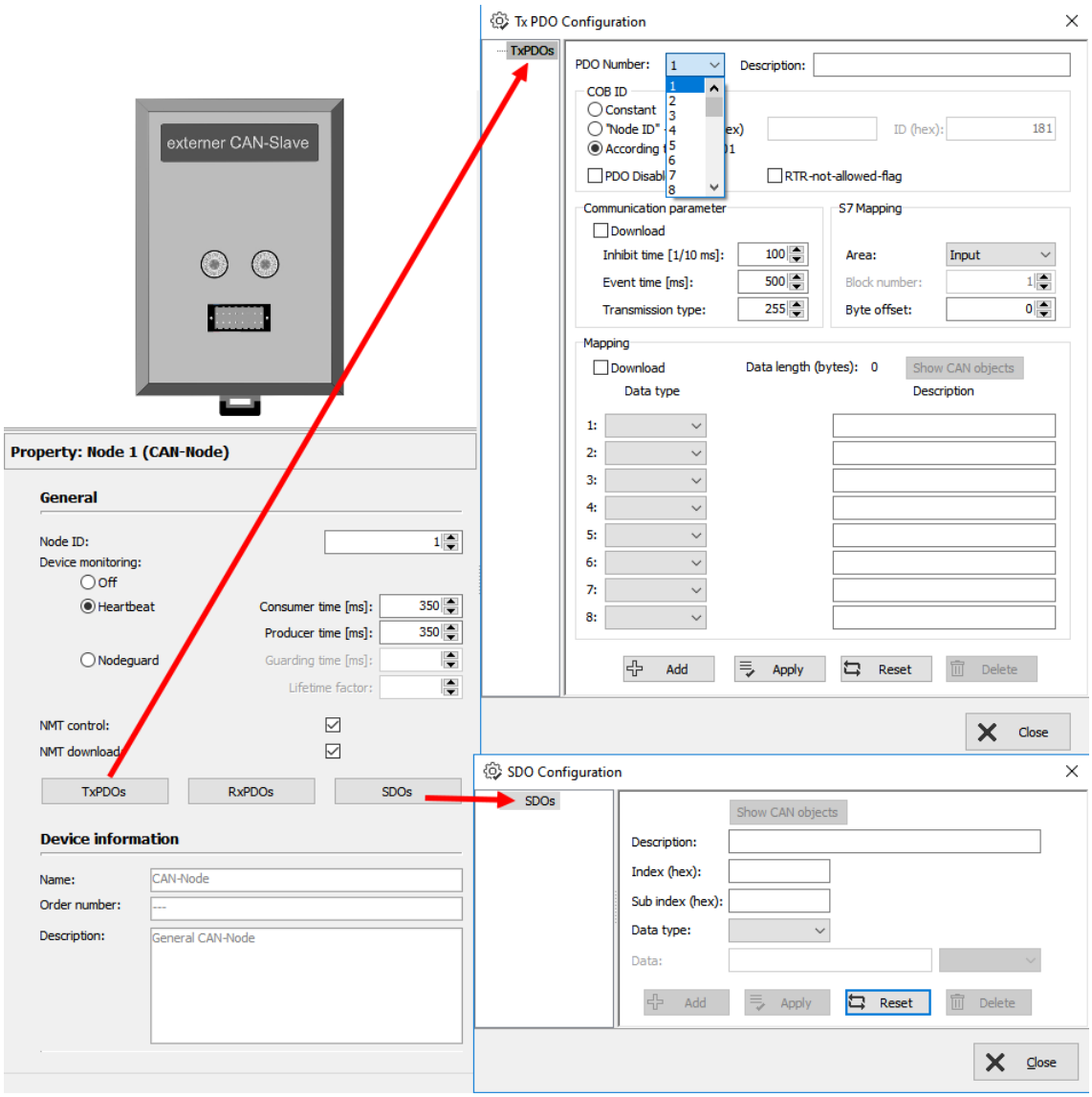
For the process data are available each **32 Tx- and Rx-PDOs**. If more as **4 PDOs** of it will be used, PDO-identifiers of other node-IDs will be used for it.

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.

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

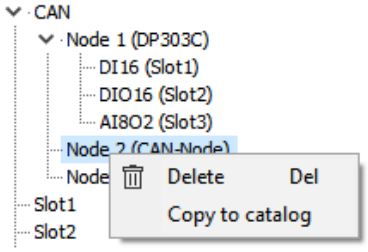
Entering of single CAN-objects (Index, Subindex) in the mapping field is optional, as long the download of the mapping is not activated. Always assign the data type to define the telegrams length and to allow a real byte swapping.
With activated download of complete mapping-parameter the PLC configures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

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

Create library elements of your CAN-slaves

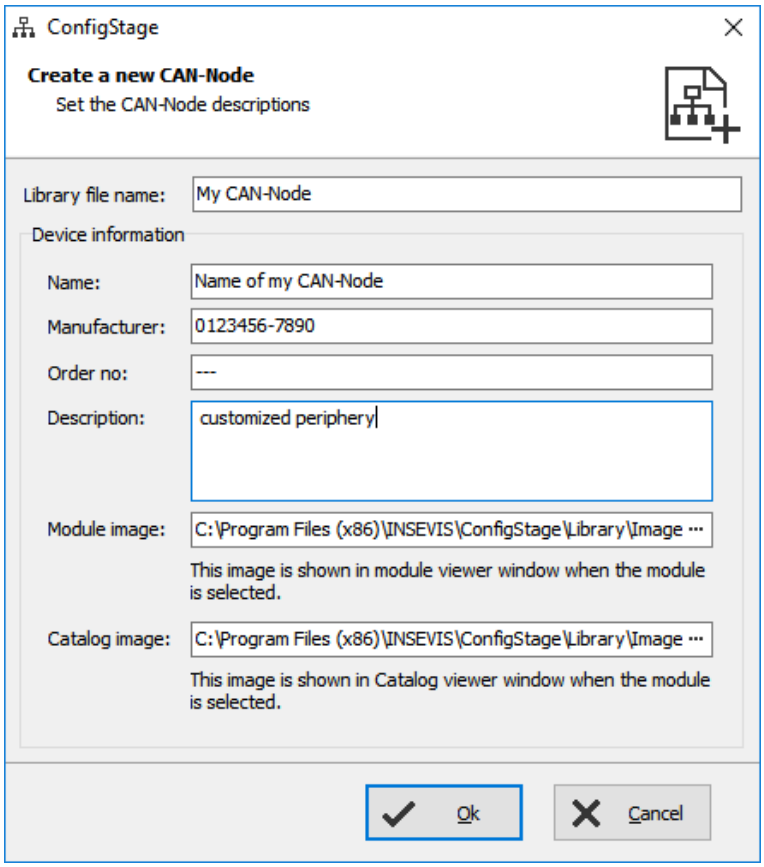
Optional: If this „mapping“ is made, the new configured CAN-node (CAN-slave) can be stored locally in the ConfigStage to use this part as a library element with pre-defined values.

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.

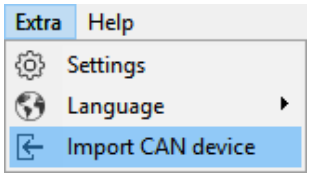


The nomenclature **TX** (Transmit) and **RX** (Receive) are at CANopen®-PDOs always from the CAN-nodes view (slave). It means, with a TX-PDO a node sends its input data to the PLC and with a RX-PDO it receives its output data.

Configuration with the Software „ConfigStage“

Configure decentral external peripherie by EDS-file

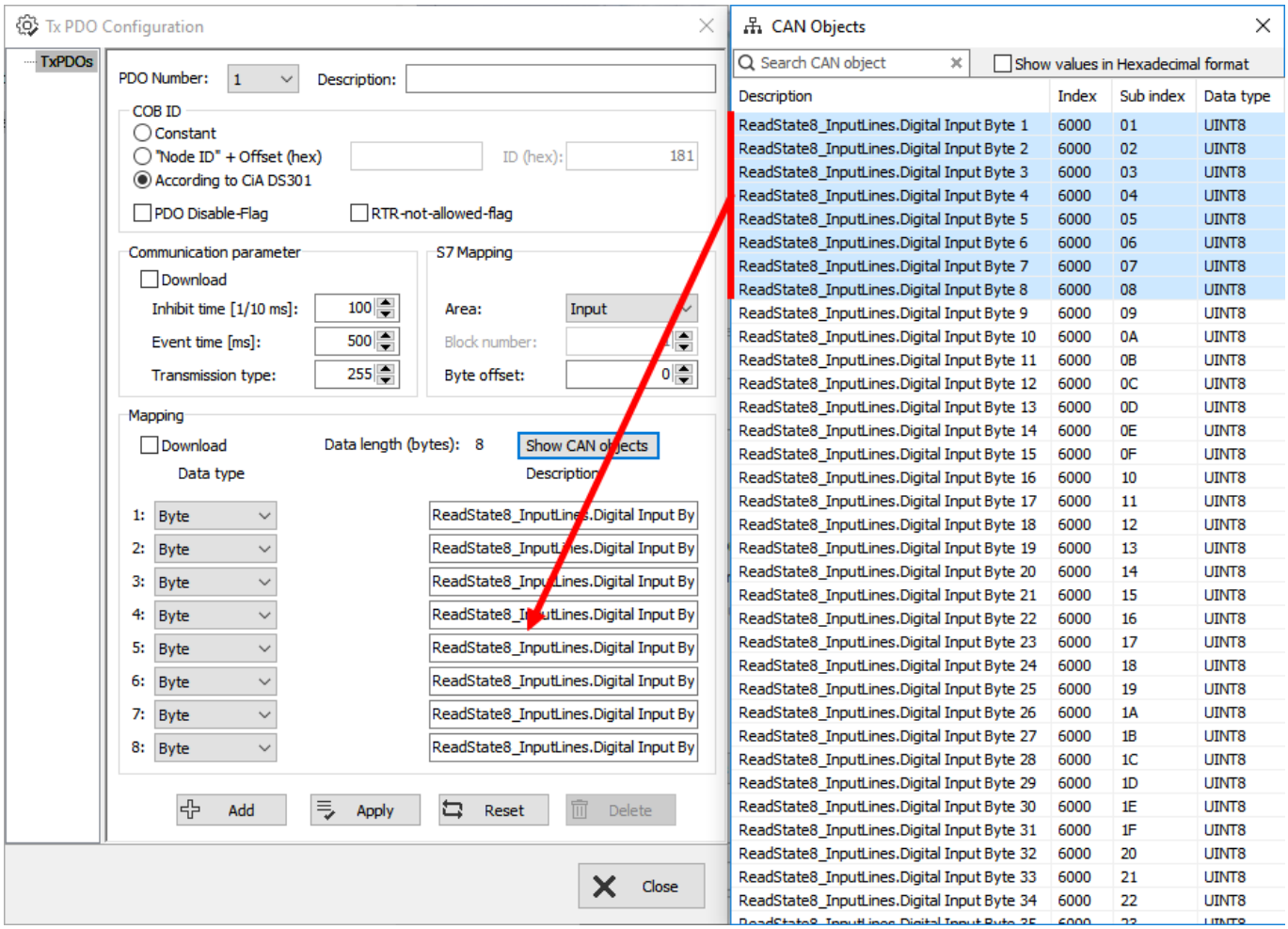
You need a EDS-file of the external CAN-slave you want to configure. Import it like shown here:



Than 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.

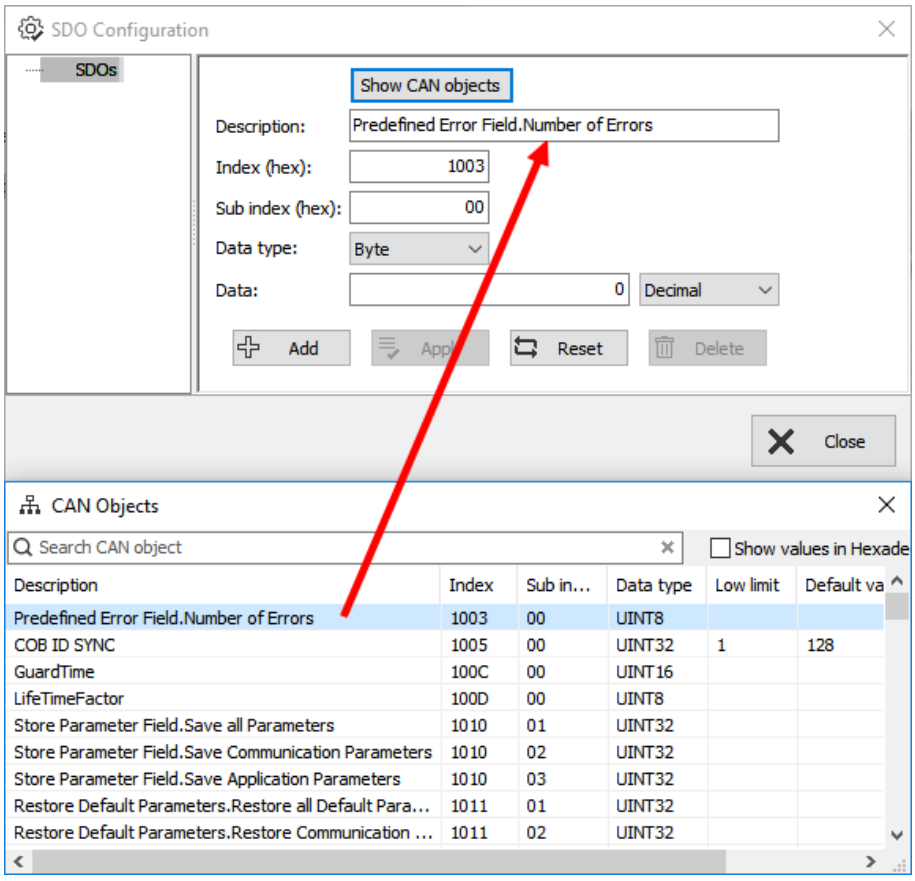
PDO configuration

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.



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.

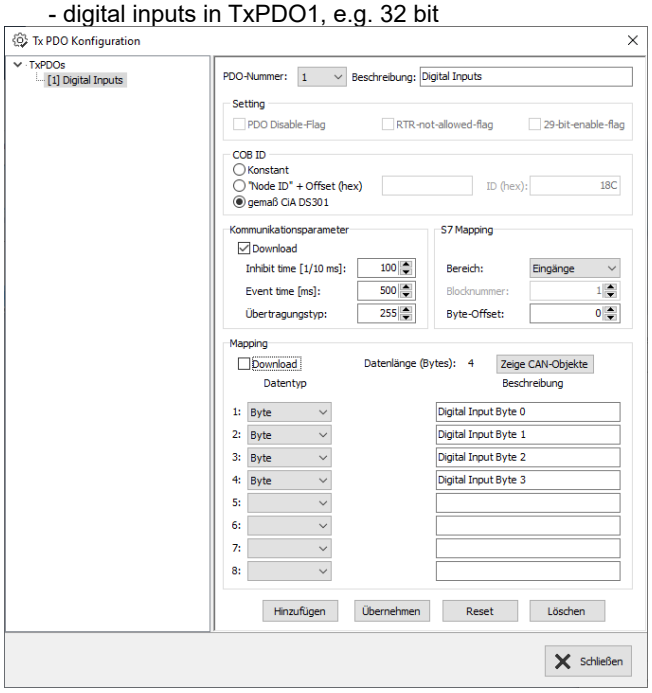
The complete description with demonstration program and data blocks is available at our website

Configuration with the Software „ConfigStage“

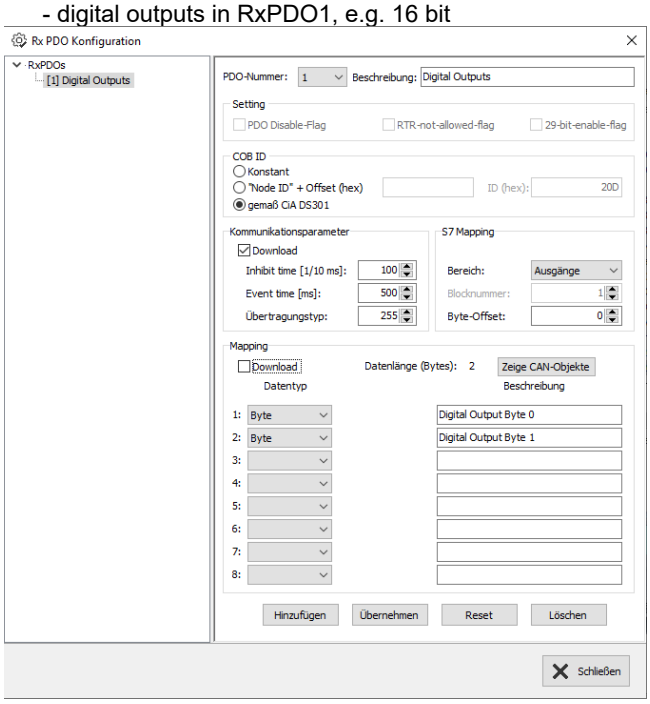
Samples for external CANopen devices

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

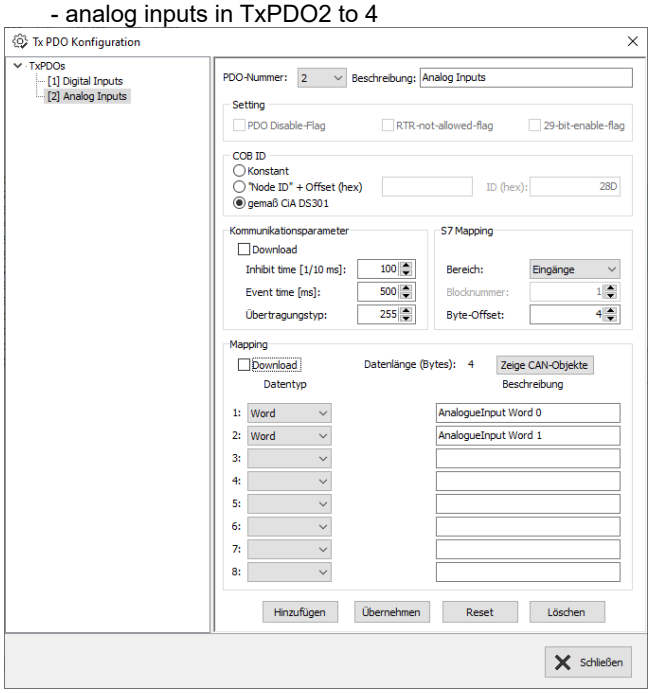
- digital inputs in TxPDO1, e.g. 32 bit



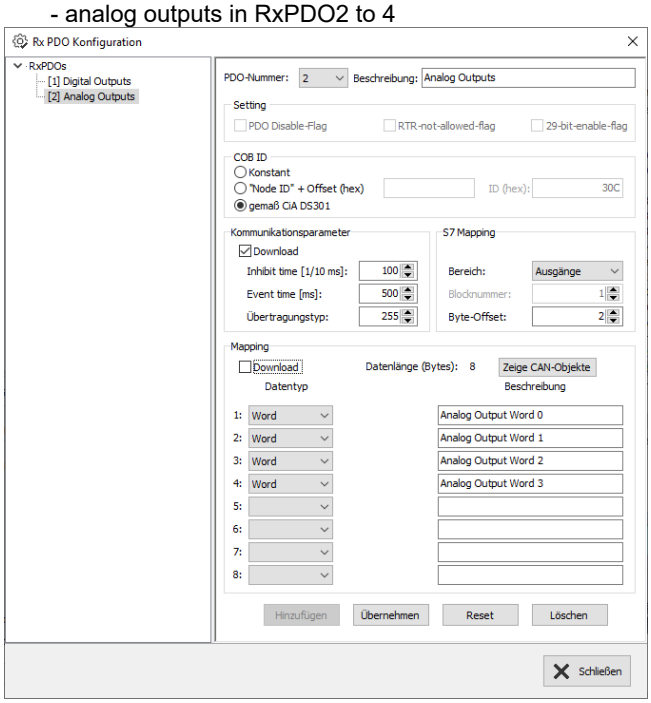
- digital outputs in RxPDO1, e.g. 16 bit



- analog inputs in TxPDO2 to 4



- analog outputs in RxPDO2 to 4



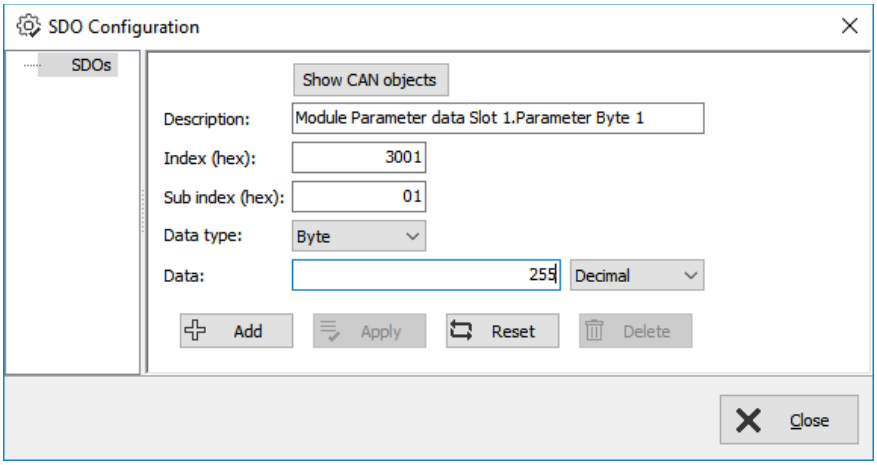
(Images in German language but self explaining)

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“

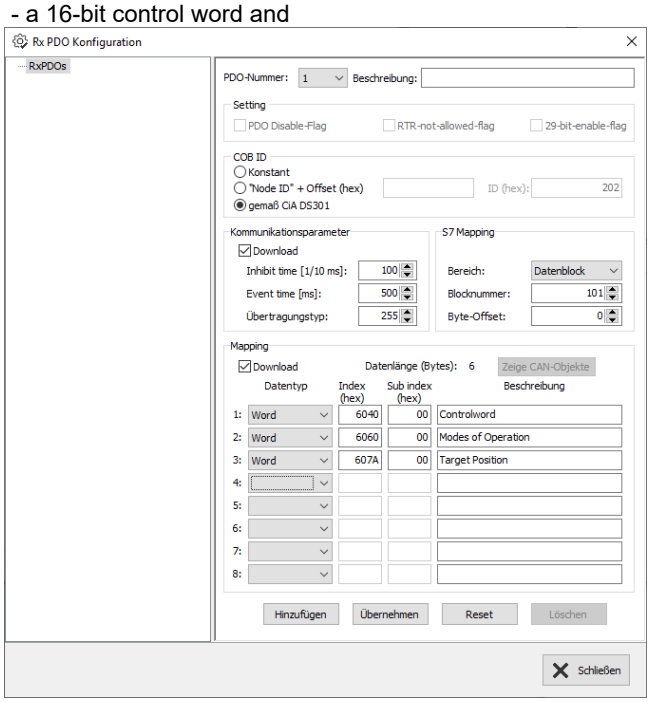
To transfer parameters (e.g. metering ranges for analog modules), SDOs need to be configured specific referring to the vendor.



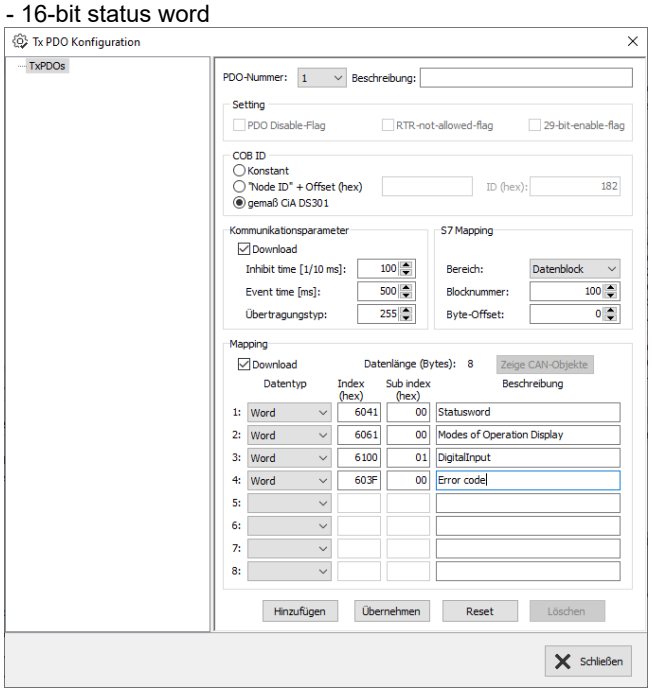
Sample configuration CANopen-drive controller ref. to DS402

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



- 16-bit status word



(Images in German language but self explaining)

Assign these both words to S7-operands by the PDO-mapping and than 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 S7-315-2PNDP with Firmware 3.2.7 (6ES7 315-2EH14-0AB0) at INSEVIS-CPUs 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) *Local data „OB86_MDL_ADDR“ (LW6) has a fix value 2047 (W#16#7FF)*
OB 100	Restart
OB 121	Programming error
OB 122	Periphery access error

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

OB	Function *(more information at single OB-chapters)*
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)
SFB 52 / 53 / 54	Read / write asynchronous data record / read alarm (available in CPU-T (for Profinet) only)
SFB 81	Read dynamic parameter (available in CPU-T (for Profinet) only)

System function blocks (SFB) - what are not available at INSEVIS-CPUs 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 TRECV	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 *Parameter „LADDR“ must be 2047 (W#16#7FF)*
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 *(only supported the SSL_ID W#16#0092, W#16#0292, W#16#0692 for Profibus DP master system)*
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-CPUs 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 interrups
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 *(more information at single SFC-chapters)*
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
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

System functions

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

SFC	Function *(more information at single SFC-chapters)*
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

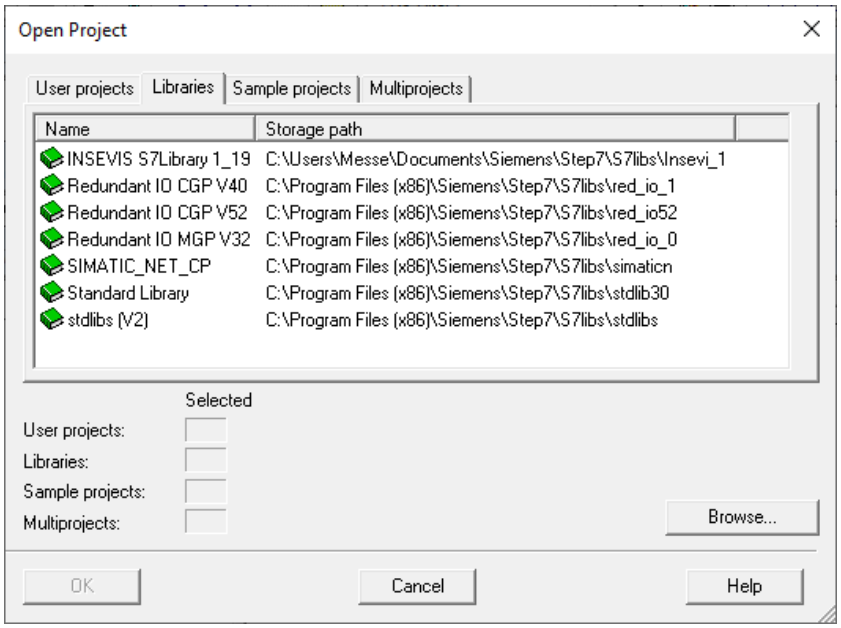
Import of INSEVIS-blocks into the S7-Project

ATTENTION:

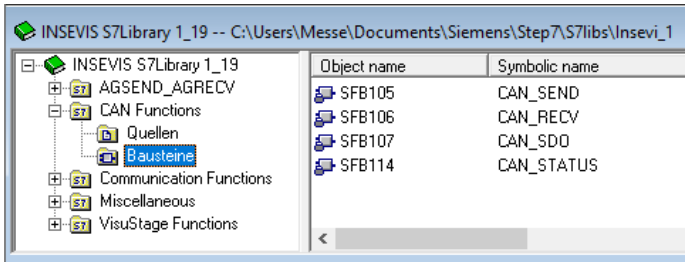
All additional INSEVIS- blocks are available as **S7-Library** for free download at the INSEVIS-web sites in the download area. It will be offered always the newest library with the most functions, what needs the referring firmware in the PLC. (e.g. Insevis_S7-library_from_2_1_5.zip - works from firmware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

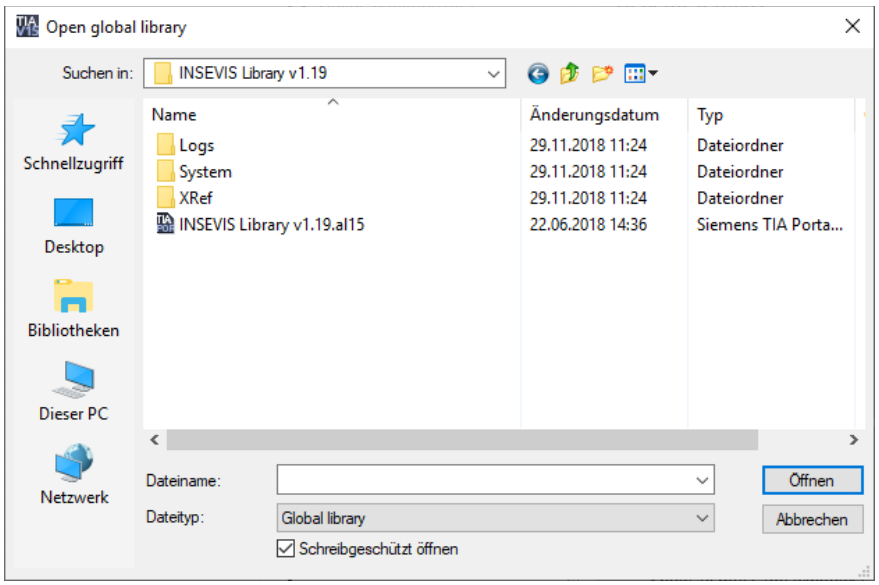


4. Copy the selected SFB´s by drag´n drop into your project



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

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

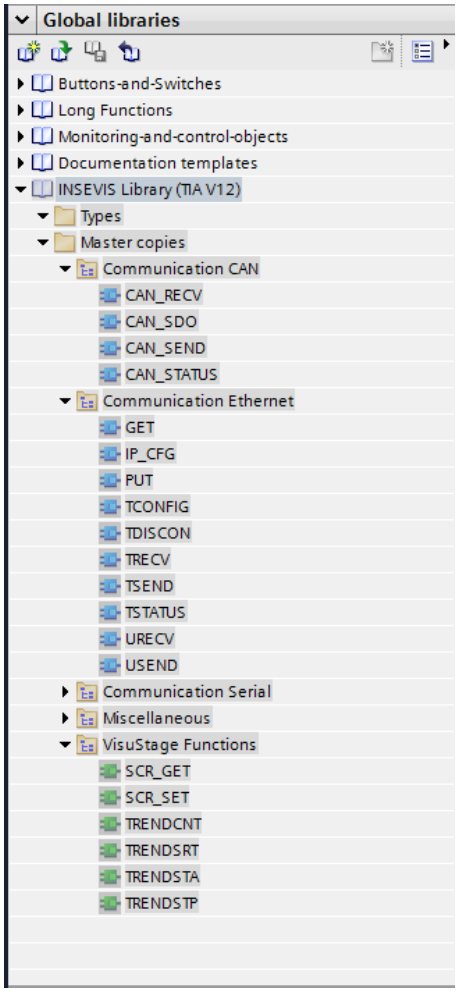


1. Download latest version of INSEVIS library for TIA-portal
2. store it and extract it in any folders
3. Open library "***INSEVIS Library v1.19al15***" from the referring folder in TIA-Portal as "global library".→

   → *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 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 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 → from device to Micro-SD-Card

With the **SFC 208** BACKUP ALL data (S7 program, visualization -incl. PINs changed by operators), process data) are written to the micro SD card in a binary file (encrypted on request). For this purpose the PLC is set to STOP until the backup is done. This allows even an end user to create a copy of the original device without any PC programs at identical devices, which also contains already recorded process data. Of course, only with the permission of the machine builder and his password file..

→ Application example: **Providing a redundant replacement machine**

- A customer wants to have an alternative or replacement machine that can always continue to operate with the same current parameters of the original machine when maintenance work is carried out on the original machine.
- For this purpose a backup is created e.g. at the end of each production cycle with the SFC208.
- If the original machine is switched off/maintained, the alternative/replacement machine can continue to work with the process data of the last production cycle by means of a restore.

Manual program backup for device manufacturers → from ServiceStage to PC.

Manually and completely without SFC for programmers with access to source data:

- Using the ServiceStage to execute the "OnlineBackup" function.
- ServiceStage creates a directory "Backups" on the PC and copies the file "S7-programs.wld" containing ALL S7 program blocks, data blocks and hardware configuration data (i.e. the ConfigStage configuration) to this directory.
- The binary file of the visualization with the name "visudata.vsbin" is copied into the same directory, if this access was explicitly released in the VisuStage project. Changed PINs are NOT stored as well.

Creating a manual programme restore on an SD card

Manually create a micro SD card with the desired source data:
Create the "Backups" folder on the SD card
The desired "S7programs.wld" system data and all OB, DB, FB, SFB and SFC
and
save the desired visualisation binary file a "Visudata.vsbin" in the folder

→ Application example: **Import standard program for series devices**

- A programmer hands over in such a way a micro SD card with the program components to his customers.
- After hardware reset, each PLC loads its program data from this micro SD card without a PC.

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)

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 carrage 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 (Data type)	Second byte (Format)	Format (Discription)
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 (Data type)	Second byte (Format)	Format (Discription)
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

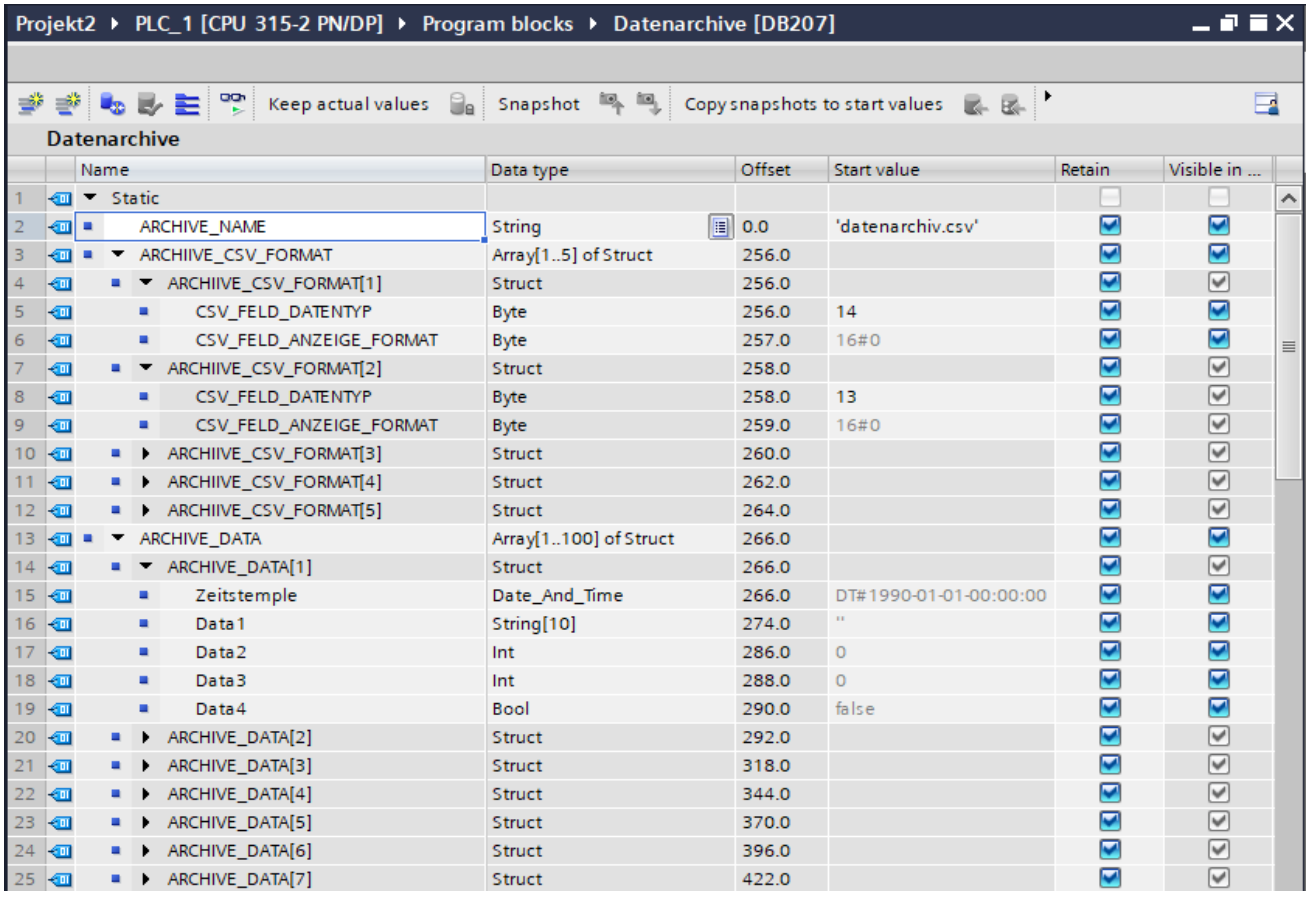
Status code meaning

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 (Always use one of SFC 207-209, 213 and 220-225 only)
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.

The example shows a configuration of the CSV archive file with the following structure:
1. date&time (1st byte =14 / 2nd byte=0 / expanded)
2. string (1st byte =13 / 2nd byte=0 / expanded)
3. integer (1st byte =5 / 2nd byte=0 / not expanded)
4. integer (1st byte =5 / 2nd byte=0 / not expanded)
5. bool (1st byte =1 / 2nd byte=0 / not expanded)



	Name	Data type	Offset	Start value	Retain	Visible in ...
1	Static					
2	ARCHIVE_NAME	String	0.0	'datenarchiv.csv'	☑	☑
3	ARCHIIVE_CSV_FORMAT	Array[1..5] of Struct	256.0		☑	☑
4	ARCHIIVE_CSV_FORMAT[1]	Struct	256.0		☑	☑
5	CSV_FELD_DATENTYP	Byte	256.0	14	☑	☑
6	CSV_FELD_ANZEIGE_FORMAT	Byte	257.0	16#0	☑	☑
7	ARCHIIVE_CSV_FORMAT[2]	Struct	258.0		☑	☑
8	CSV_FELD_DATENTYP	Byte	258.0	13	☑	☑
9	CSV_FELD_ANZEIGE_FORMAT	Byte	259.0	16#0	☑	☑
10	ARCHIIVE_CSV_FORMAT[3]	Struct	260.0		☑	☑
11	ARCHIIVE_CSV_FORMAT[4]	Struct	262.0		☑	☑
12	ARCHIIVE_CSV_FORMAT[5]	Struct	264.0		☑	☑
13	ARCHIVE_DATA	Array[1..100] of Struct	266.0		☑	☑
14	ARCHIVE_DATA[1]	Struct	266.0		☑	☑
15	Zeitstemple	Date_And_Time	266.0	DT#1990-01-01-00:00:00	☑	☑
16	Data1	String[10]	274.0	''	☑	☑
17	Data2	Int	286.0	0	☑	☑
18	Data3	Int	288.0	0	☑	☑
19	Data4	Bool	290.0	false	☑	☑
20	ARCHIVE_DATA[2]	Struct	292.0		☑	☑
21	ARCHIVE_DATA[3]	Struct	318.0		☑	☑
22	ARCHIVE_DATA[4]	Struct	344.0		☑	☑
23	ARCHIVE_DATA[5]	Struct	370.0		☑	☑
24	ARCHIVE_DATA[6]	Struct	396.0		☑	☑
25	ARCHIVE_DATA[7]	Struct	422.0		☑	☑

System functions

```
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                                    // end this function
R     M   1501.3                       // 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
```

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 (Always use one of SFC 207-209, 213 and 220-225 only) 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 programer 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

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

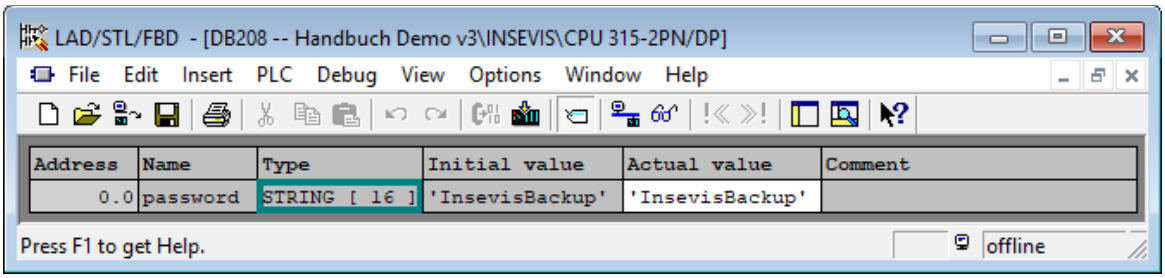
- Backup result is indicated by OB100_STRT_INFO parameter of OB100 by next NEWSTART.

  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.

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) (Always use one of SFC 207-209, 213 and 220-225 only)
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)

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 ther Micro-SD-Karte does **not allow** some signs in the file- or directory name

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

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 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.
- 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 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 : Coping 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
```

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
```

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 overs 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 (Always use one of SFC 207-209, 213 and 220-225 only)
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 ther Micro-SD-Karte does not allow some signs in the file- or directory name

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

These signs will be ignored.

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
```

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

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 (Always use one of SFC 207-209, 213 and 220-225 only)
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 exits (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
```

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 (Always use one of SFC 207-209, 213 and 220-225 only)
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
```

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 (Always use one of SFC 207-209, 213 and 220-225 only)
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
```

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 (Always use one of SFC 207-209, 213 and 220-225 only)
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 (Always use one of SFC 207-209, 213 and 220-225 only)
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
// ...
      BEA                             // end of function
```

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 (Always use one of SFC 207-209, 213 and 220-225 only)
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
```

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 overs 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
```

```
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.

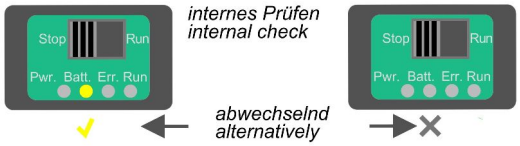
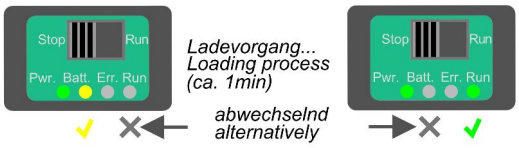
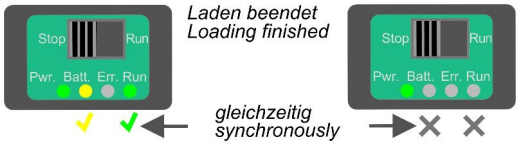
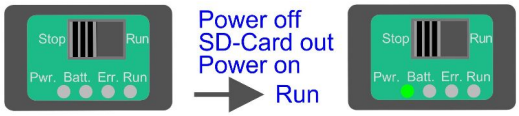
```
      UN    M   1502.2                 // If request is not set, then
      BEB                               // end the function

      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
```

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)	
Internal check *(depends on existing s oftware version)* LED Batt. (yellow) blinks while internal CPU check runs	2) 	
Loading process in operation *(Do not switch off the device, no interrupt in power supply!!)* „Power"-LED shines permanently LEDs „Batt." (yellow) and „Run" (green) blink alternatively	3) 	a) b)
Loading process finished „Power"-LED shines permanently LEDs „Batt." (yellow) and „Run" (green) blink together	4) 	
initialize the new OS - switch device power OFF - pull out the micro-SD-card - switch device power ON	5) 	

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

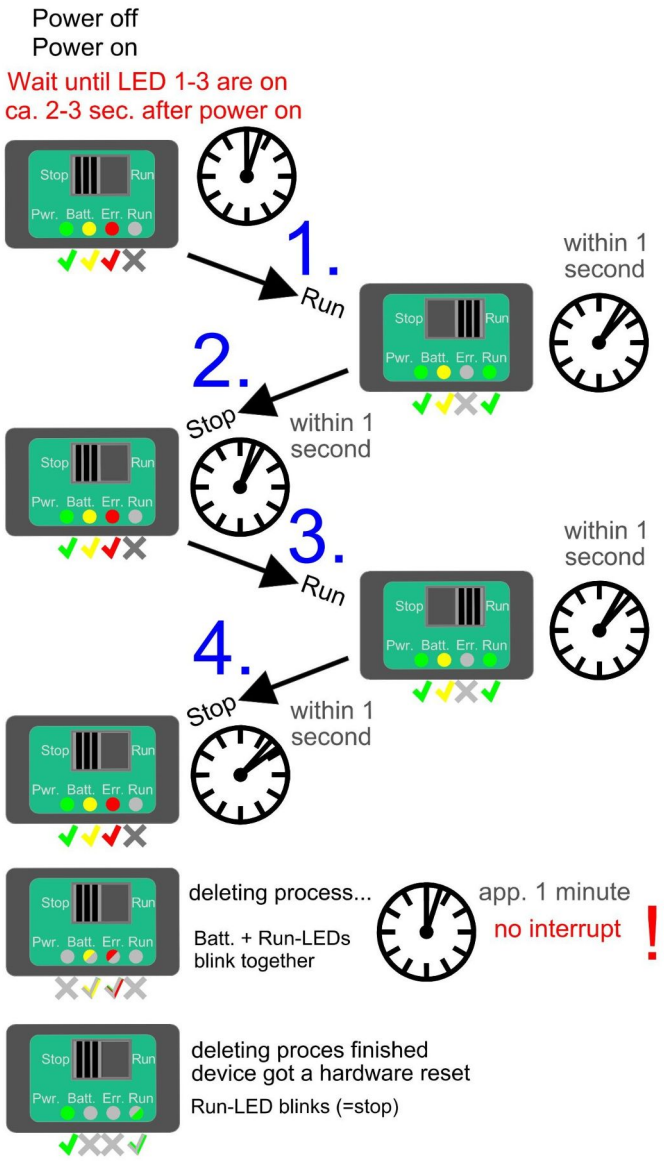
3 Run/Stop-switch in Run
red „Error“-LED now is off
green „Run“-LED now glows

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.
(A subfolder "Backups" with the files "S7programs.wld" and "Visudata.vsbin" must be present on the micro SD card).

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.

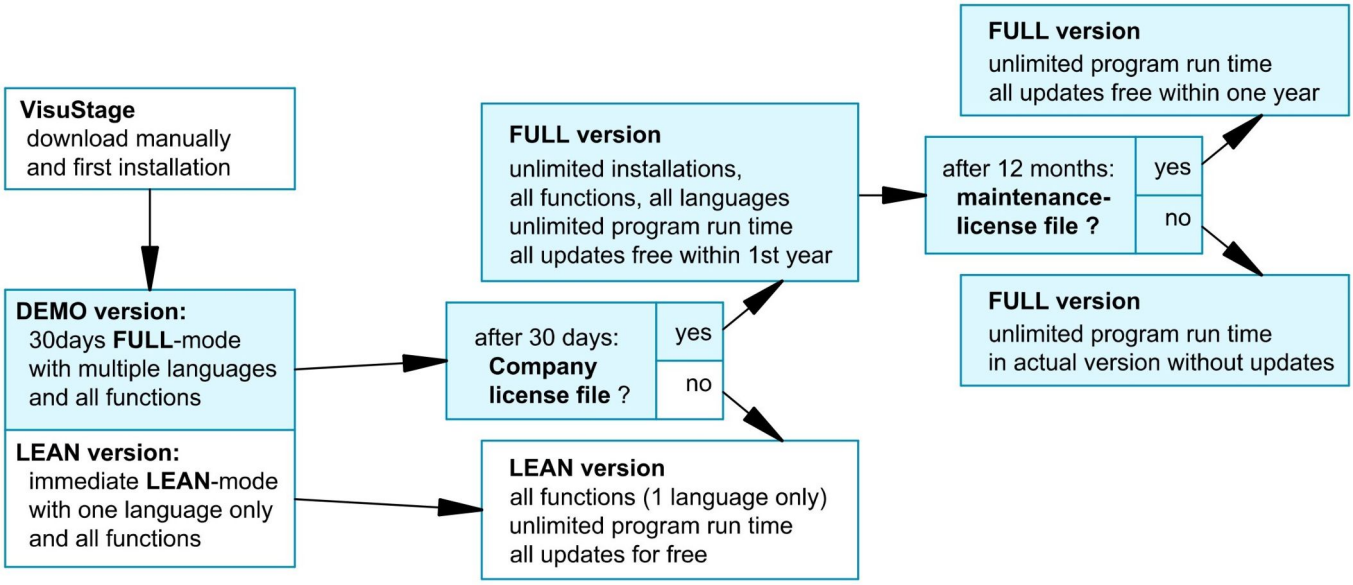
Visualization with the software „VisuStage“

Download and licensing

The actual version of VisuStage is free to download at INSEVIS websites in the download area. INSEVIS- programs do not check the internet for new versions, so pleas visit INSEVIS websites for new features by yourself. This software can be used with operating system Windows 10 only.

At the first installation VisuStage offers 3 modes: FULL-Mode with license, DEMO-mode with 30 days free FULL-mode without license and LEAN-mode without license. You can add a license later on at the folder “Help” → “Info” to upgrade a DEMO or LEAN version into a FULL version.

This company license can be multiple times used within one company for multiple installations. (Open visustage.exe once as administrator (by right mouse key) to not have to enter the license path every time).



System functions

The S7 Operating system provides the system functions (SFC) to control the visualization by your S7 program. These SFCs are described in the referring PLC-manual (Compact or Panel-PLC) at the chapter “System functions”. To use these SFCs you need the **S7-Library** from INSEVIS. It can be downloaded at download-area of INSEVIS websites for free.

Communication between PLC and external Panel-HMI

In a Panel-PLC these SFCs will be used from a common CPU for PLC- and HMI-functions as well. Only if you use PLCs with external HMI please note followig:

ATTENTION:
Only single SFC-jobs can be processed.
If the parameter "BUSY" of the corresponding SFC returns the value FALSE, the job is finished and a new SFC can be processed,
The configuration of connection parameters of INSEVIS-Panel to INSEVIS-PLC or external PLC is explained at the chapter “Resource Partner-PLC-device” in this manual

Use these data in the S7-program (e.g. LADDR := W#16#1 for the panel-connection with ID-No. 1)

Here the PLC can change the screens in an external panel (e.g. with the connection-ID-No. 1

```
CALL „SFC 201
LADDR     :=  W#16#1
SCREEN    :=  MW1006
RET VAL   :=  MW1008
BUSY      :=  #busy
```

More easy meanwhile: write/read your scree-ID into/from a variable (see more at the “Screen” resource)

General settings of the program shell

Top: **Standard functions bar** (self explaining or explained by tool-tips)

Alignment functions (2 and more objects)

FIRST mark the Master-element,
THAN mark the others, what shall be aligned to the master
than choose the alignment property you want (left, horizontal centered, right, top, bottom centered, bottom)

Size functions

FIRST mark the Master-element,
THAN mark the others, what shall be aligned to the master
than choose the size item (width, height)
or assign multiple dimension properties to different selected objects (right)
with key-combination of CTRL+arrow-keys size of marked objects can be changed pixelwise

Alignment functions (1 or 3 and more objects)

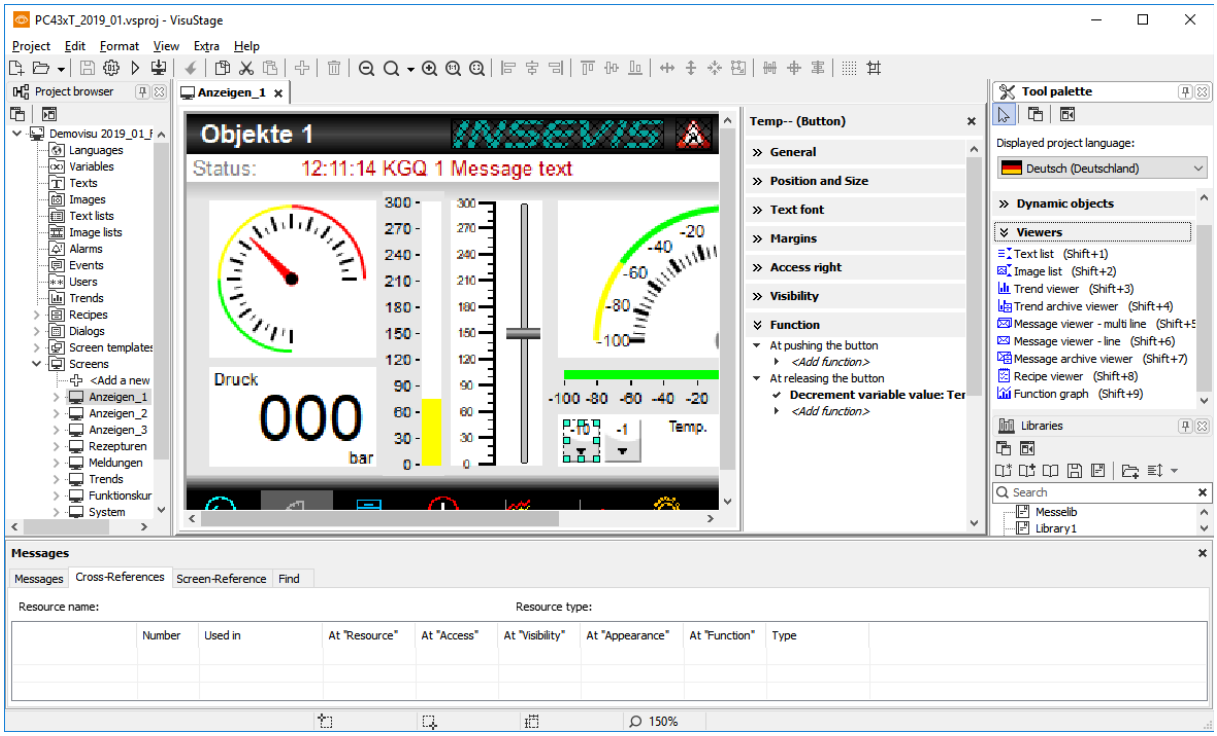
FIRST mark the Master-element,
THAN mark the others, what shall be aligned to the master
FINALLY the element, what should mark the outer border of the grid
than choose the alignment property you want (same horizontal distance, center a single object, same vertical distance)

Grid-/ Snap functions (self explaining)

At Extra / Settings assign grid size
(de-) activate grid and snap by mouse click

General settings (without figure, self explaining)
It is recommended to hide the panels bezel to save window-space on your VisuStage-PC.
Grid and snap functions make it easier to design new screens

Top: General functions and drop down menus,
Left: Project browser with resource settings (here the resources will be set up and parametrized) and screens
Middle: Window of the projected screen, display of the objects parameters of the the selected object
Right: Tool bar with project language, function catalog with all objects what are available, Library catalog
Bottom: Message window (for messages, references, queries and filter results), to be switched on/off by F6-key



Visualization with the software „VisuStage“

General functionality

VisuStage imports S7-PLC-variables from Simatic®-Manager or TIA® Portal and uses it in different visualization objects in different screens. It can synchronize these S7-variables to the S7-project automatically.

The visualization source file will be compiled into a binary, what will be transferred to the Panel-HMI or Panel-PLC by Ethernet. The firmware in the Panel-HMI/-PLC creates the images and touch fields at the touch display there. No source date can be read back from the Panel-HMI/-PLC.

There is possible to get a free remote visualization of all S7-PLCs on a PC-screen by RemoteStage, what requires the visualization binary.

The following descriptions refer to the actual sample project what is available at the download area at INSEVIS websites. It requires the newest VisuStage-version and the newest operating systems at the devices. It is hardly recommended to use this project together with this documentation to understand it better.

Recommended procedures

A systematic working procedure reduces errors and decreases the programming efforts

- Import your S7-variables from the S7-symbol table or directly from your S7-DBs including symbols and synchronize it with your S7-project
- Assign objects for multiple screen use in different template screens, what you can use in every single screen.
- If you have finished entering all texts in your project language, you may export them into a *.csv-file and forward it to a translator into any other language you need. It is only necessary to keep the *.csv structure then you can re-import this file again - and you have your next language texts.
- Collect all special pictures for your visualization theme before you start. (We strongly recommend the file format "png", because it works with a good compression and it allows definition of transparency by an alpha-channel. This is very important, if you want to see your background color on rounded corners and not any image colors.)
- Use the grid- and snap-function.

Save your resources

It is very important to have an eye on the memory size, required by the visualization. In general it is better to tune your visualization with some easy hints, before you decrease the PLC performance with an unsuitable visualization.

- Multiple use of buttons
  (if you use exactly the same buttons (resolution, image) you save lots of memory size because they will stored only one time as 1 object. It dies not matter, if symbols or texts belonging to this items or not.)
- Do not overlap dynamic objects
  (This function is not allowed by program cycle. You better navigate your images by dialog boxes with coords.)
- Better to have less screens with more common objects than to have lots of nearly empty screens.
  (Every new screen is stored as image and requires memory space.

Keep the overview

You need to know, what resources are „active“ and what are „blind passengers“.

- Filters reduce the number of displayed resources to get a bettor overview.
- With „cross reference“-function (right mouse button in the resource-menu) the use of variables, texts, images, text lists, image lists and trends can be displayed. Delete resources you do not use.
- Delete unused resources with the „CleanUp“-function at the rider „Edit“

Transfer your work to other projects

When you invested much time to create a very special head- or bottom line or navigation:

- Use the library function to export often used groups to you PC for a further use in other projects.
- Use the screen template function for Headlines or footes, etc.

VIDEO-Tutorials available You find detailed explanations in the VisuStage-Playlist at the INSEVIS YouTube®-channel „INSEVIS EN“. This manual refers to e referencial visualization, what is available for free download at the download area at INSEVIS- web sites.

General project formats and settings

File formats

VisuStage creates 3 file formats:

- ***.vsproj** contains the source code of the visualization
- ***.res** contains the resources of the visualization
- ***.bin** contains the compiled binaries to be downloaded into the panel

Hint:
The vsproj and res-files must be in the same directory in order to open the project (vsproj)...
When passing on the source data, send the vsproj and res files!
If no source data is to be passed on, then pass on the vsbin (sufficient for updating).

At the menu „Project“ basic settings will be made.

General data

- manual changeable

Version counter

- manual assignment of a 3 digits compile-no.
  or
- automatic counter (can be displayed by text field)

Screen rotations

- Only for CPU-P and -T versions

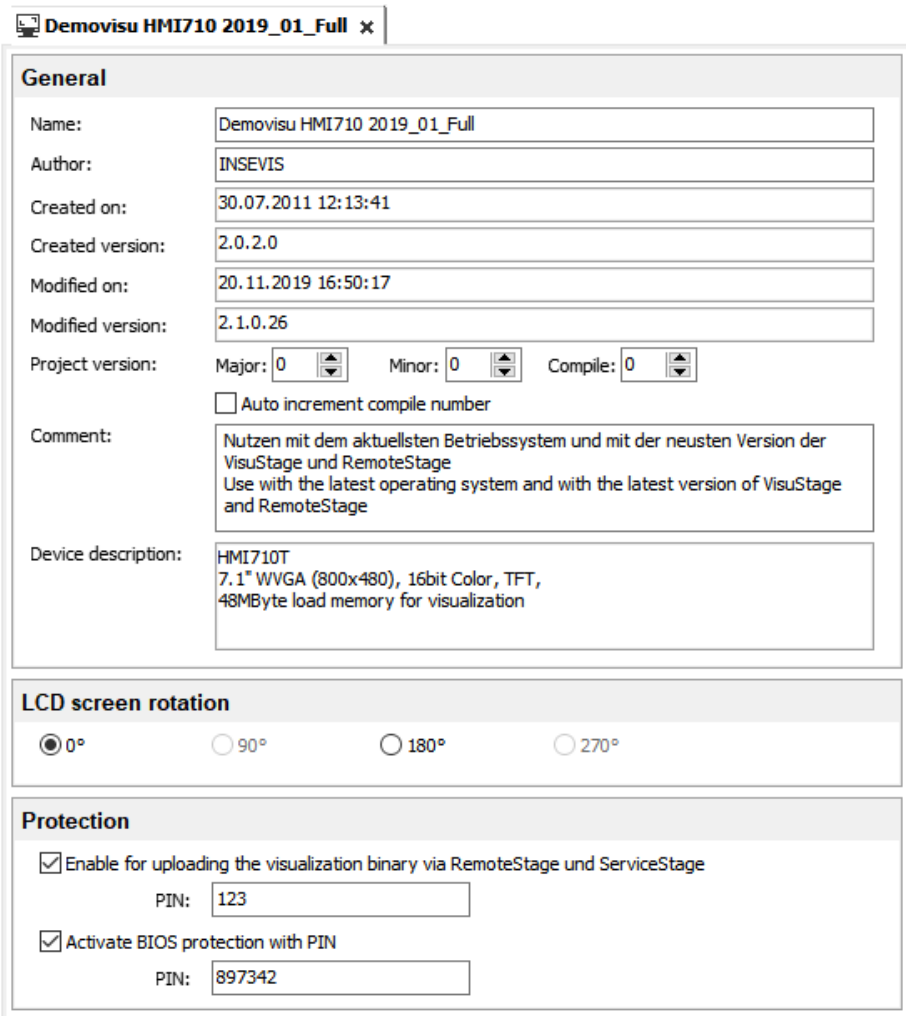
Data protection at upload

Secures the the vsbin file for upload (must be switched on, in each case with/without entering a PIN)

- into a remote PC (via RemoteStage)
- to a remote PC (via ServiceStage)
- to the micro SD card (for backup)

and (only for HMIs)

- Secures the the BIOS screen by entering a PIN (for access see chapter BIOS)



Conversion of third-party visualisations

TIA WinCC visualisations can to a large extent be automatically converted into VisuStage format. This saves the manual import of variables, the creation and linking of pages (menus) and the creation and linking of static and dynamic objects.

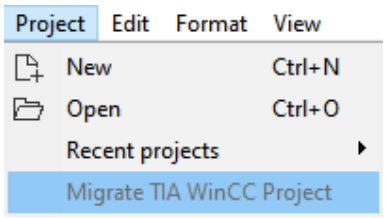
Only list objects such as recipe and trend management and the fault message system must then be added manually, as these cannot be converted.

Requirements

This function is integrated from VisuStage version 2.1.0.36.
However, on the same computer as the VisuStage, the Siemens software tools

- TIA Portal 14-16.1 and from VisuStage version 2.1.0.37: TIA V17
- TIA Openness *

must be installed on the same computer as VisuStage in order for the conversion function to be activated (not greyed out).

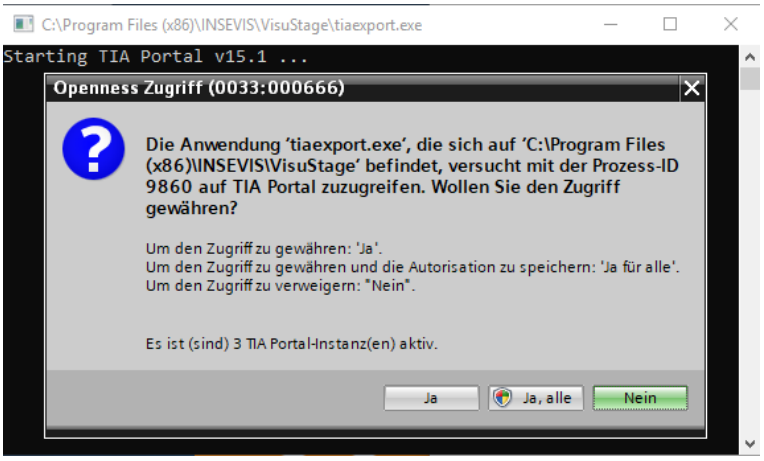


Hint for the correct settings when using the Openness software:
The Windows user must be a member of the "Siemens TIA Openness" group.
Depending on the Windows operating system, please set this assignment as administrator!

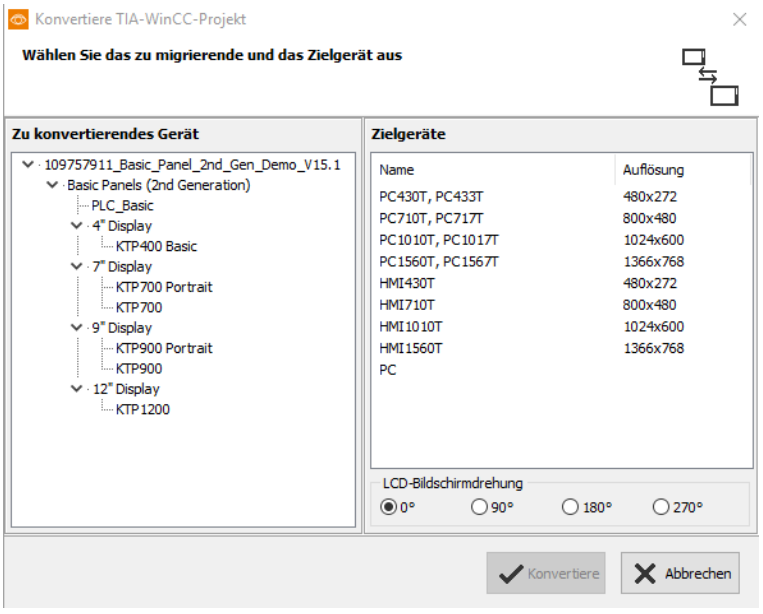
See here for more: https://support.industry.siemens.com/cs/mdm/109773802?c=101778035467&lc=en-DE

Procedure

- A project browser opens under Project → Convert TIA-WinCC Project.
- Open the TIA project with the visualisation to be converted.
- The file "tiaexport.exe" integrated in the VisuStage is opened and automatically started in a DOS window.
  (The DOS window may be hidden, but can be seen in the task bar).
- This program starts the TIA Portal.
- The Openness Firewall recognises the access, which must be granted with "Yes, all"



- A new screen opens for selecting the source and target devices.
- (Make sure that the resolutions of both devices are identical so that the entire visualisation is visible).
- Start the conversion by pressing "Convert".
- The progress of the conversion is displayed in the DOS window.
- (Depending on the size of the visualisation, the conversion may take some time).



Visualization with the software „VisuStage“

Cross reference list: The following TIA WinCC objects are converted into VisuStage screen objects

TIA WinCC „Basic objects“	VisuStage „Static objects“
Line	Line
Rectangle	Rectangle
Ellipse	Ellipse
Circle	Ellipse
Text field	Text
Graphic view	Image

TIA WinCC „Elements“	VisuStage „Dynamic objects“ /„Viewers“
I/O field	I/O field
Button (Mode=“Text“ and Label=“Text“)	Button
Button (Mode=“Text“ and Label=“Text list“)	Text button
Button (Mode=“Graphic“ and Graphic=“Graphic“)	Button
Button (Mode=“Graphic“ and Graphic=“Graphic list“)	Dynamic button
Button (Mode=“Graphics and text“)	Button
Button (Mode=“Invisible“)	Touch area
Symbolic I/O field	Text list
Graphic I/O field (Mode=“Two states“)	State area
Graphic I/O field (except „Two states“ Mode)	Image list
Date/time field	Time/Date field
Bar	Progress bar
Switch (Mode=“Switch“)	Text list
Switch (Mode=“Switch with text“)	Text list
Switch (Mode=“Switch with graphic“)	State area
Slider	Slider
Gauge	Gauge

The following TIA WinCC objects are not supported during conversion.
Either they must be created manually in the VisuStage project or they are not available due to the system.

- „Polyline“ ..... replace by multiple „Linies“
- „Polygon“ ..... replace by „Images“
- „Symbol library“ ..... replace by „Image list“ with single state images
- „Clock“ ..... replace by a digital display at „Date/Time field“
- „Alarm view“ ..... replace by „Message viewer“
- „Trend view“ ..... replace by „Trend viewer“
- „User view“ ..... replace by „Text list viewer“ with the user names
- „Watch table“ ..... this function is currently not available
- „SmartClient view“ ..... this function is currently not available
- „Recipe view“ ..... replace by „Recipe viewer“
- „f(x) trend view“ ..... replace by „Function graph“
- „System diagnostics view“ ..... this function is currently not available
- „Media player“ ..... this function is currently not available
- „GRAPH overview“ ..... this function is currently not available
- „ProDiag overview“ ..... this function is currently not available
- „Critera analysis view“ ..... this function is currently not available
- „Camera view“ ..... this function is currently not available
- „PDF view“ ..... this function is currently not available

Visualization with the software „VisuStage“

Following „Event-Press“ and „Event-Release“ functions are converted into VisuStage button functions:

TIA WinCC functions	VisuStage functions
ActivateScreen	Change screen
ActivatePreviousScreen	Change screen
SetLanguage	Select language
SetTag	Assign value to variable
IncreaseTag	Increment variable value
DecreaseTag	Decrement variable value
SetBrightness	Increment brightness
SetBit	Set a bit
SetBitInTag	Set a bit
SetBitWhileKeyPressed	Set a bit
ResetBit	Reset a bit
ResetBitInTag	Reset a bit
InvertBit	Invert a bit
InvertBitInTag	Invert a bit
ClearAlarmBuffer	Clear the alarm buffer
ClearAlarmBufferProTool (class number = 0)	Clear the alarm buffer
ClearAlarmBufferProTool (class number = 1)	Clear the event buffer
Logoff	Log-Out the user
ActivateCleanScreen	Activate Clean screen
ClearLog	Delete the alarm archive

Creation and management of resources

Resource Device Settings

Assign the target IP-address

- The connection between the PLC and the PC with visualization software „VisuStage“is done by Ethernet S7-communication Put/Get. That´s why it is necessary to enter the IP-address of the target Panel-PLC/HMI to identify the device whereto the visualization should be send.
- **AT HMIs** the IP address of the connected PLC (partner PLC device) is entered, both must be in the same IP address range!

VNC-Server (CPU-T devices only)

- Select to start the VNC-Server always or controlled by variable
- Static / dynamic port number (default port 5900 recommended!)
- Password settings (Needed in VNC-Client/Viewer)
- Content in VNC-Client/Viewer
  - passive (1:1 display only)
  - active (controls the HMI)
  - changeable by variable (e.g. key switch)

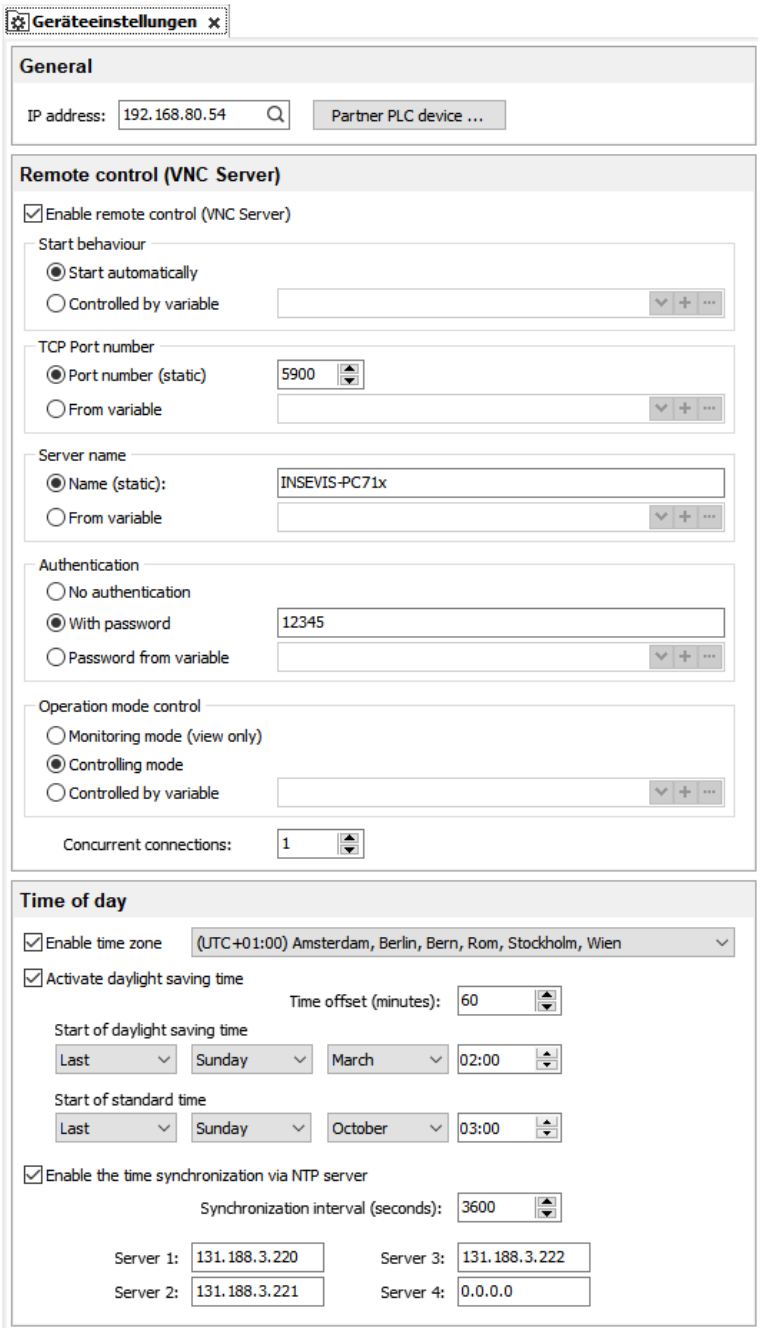
NTP-Server, time zones and daylight saving time (CPU-T devices only)

Activation of a time zone **(for HMIs only, for PLCs the configuration is done in the ConfigStage)**

- The respective changeover dates are entered here.
- The system then adjusts itself automatically. If the unit was switched off during the changeover and the battery had enough float voltage for the real-time clock, the unit adjusts the time itself to the current time during the boot process as of HMI firmware 1.6.3.
- If the unit was switched off for longer that the real-time clock became voltage-free, the time must be set manually.

Using NTP Servers

- NTP servers transmit the Coordinated Universal Time (UTC), which is the basis for calculating your own time.
  In the illustration, 3 TimeServers of the FAU University Erlangen-Nuremberg were addressed (131.188.3.220/1/ 2 correspond to the ntp0/1/2 of the fau.de).



ATTENTION:
Using a VNC-Client/-Viewer as app makes possible a mobile application. The settings of operational and security functions depend on each app.
Every user is responsible for data security by itself! Do not use VNC-Server without authentification and care for access of identified and known devices by yourself
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 installing 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.

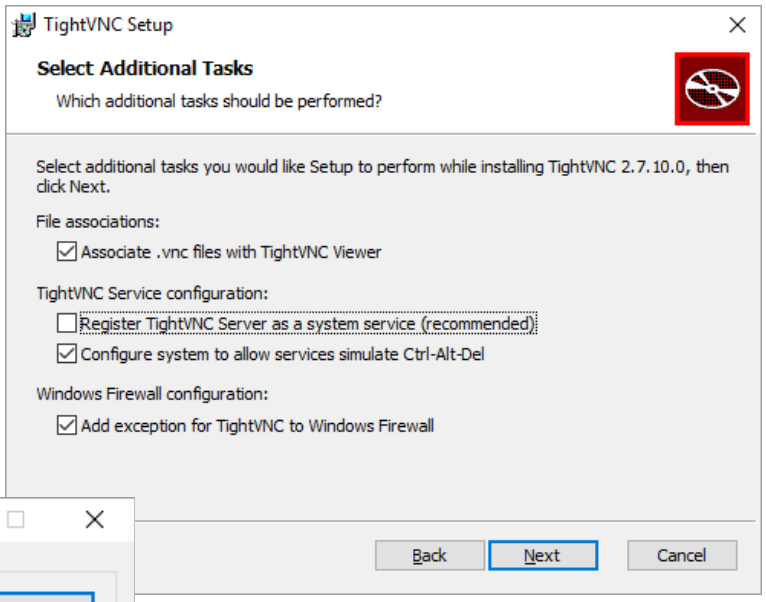
Configure a VNC-Viewer/-Client

When select a VNC-Viewer/-Client care for following issues:

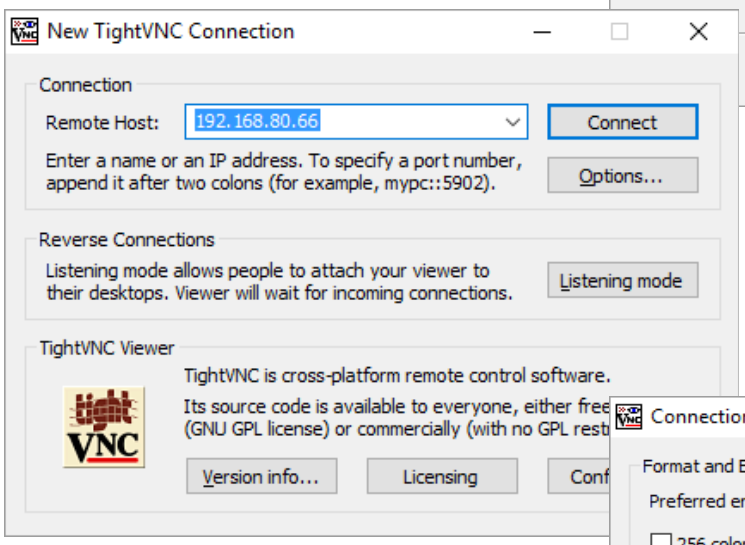
- Supporting RFB V 3.8 with encryption ZRLE, RAW-protocol
- Deselecting of „Clipboard transfer“ (copy & paste for texts)

See a setup and configuration sample for the free **Tight VNC-Viewer** to get the most important settings from installation to remote visualization

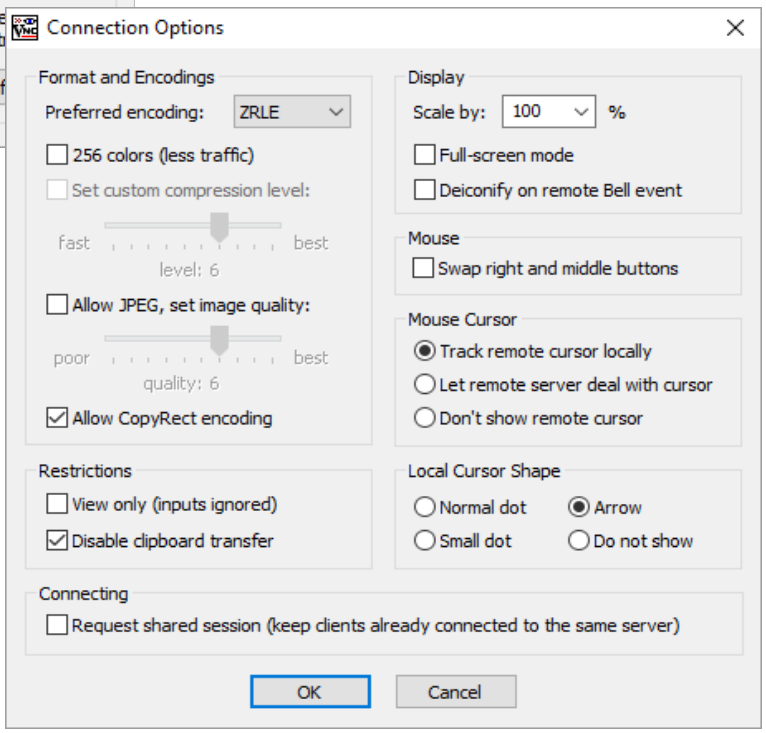
Deselect server function during program setup



Hint: Images are made at installation of Tight VNC-Viewer. This is a free software easy to install and configure, so that we can recommend it.

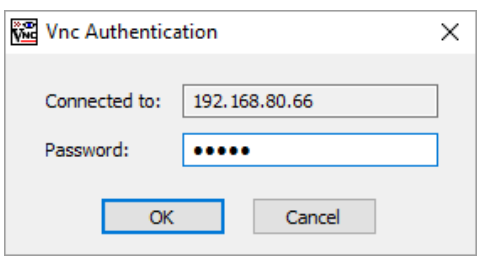


Assign IP-address of the remote devise and go to „Options“



assign “ZLRE-Encoding”
deselect „256 colors”
deselect „Clipboard Transfer“
deselect „Shared Session“

Insert password what was assigned in VisuStage



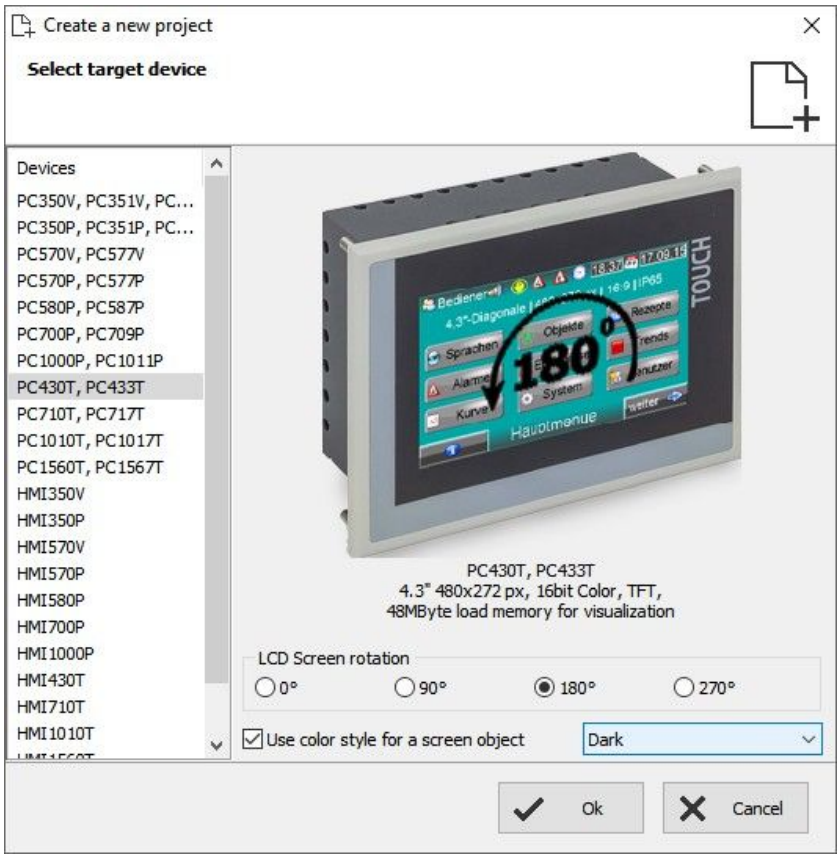
Change screen orientation or resolution

At the menu „Project/change target device“ it is possible to adapt the project to another target system.

This could be:

- other Panel-PLCs or Panel-HMIs from INSEVIS)
- a free resolution for using PC-monitors as remote panel
- other orientations of the same device (in 90-degree-steps for CPU-P-versions only)

This procedure is only made arithmetical and needs manual reconditioning to get an attractive visualization and to reduce the binaries memory size (e.g. font sizes, bottom sizes, etc.).

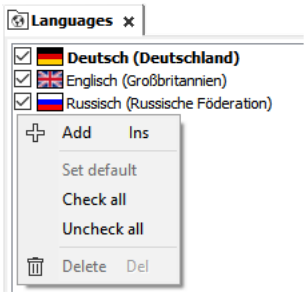


Create and administrate resources

Resource languages

All languages installed on your PC are available for VisuStage-projects. These will be stored as bmp. If an other PC without these installed languages will be used to compile and download to the Panel-PLC/HMI, the missing languages can not not be displayed correctly.

- Add / delete a language: → by right mouse key
- Select the project language: → by right mouse key opens up a pull down menu
- Select a reference: language (marked bright) → by right mouse key „Set default“



Hint:
Text- and image resources are connected to the language, where they were assigned/imported. If this language is deleted, all (text- and image) resources will be deleted after confirming it in a confirmation window.

Resource variables

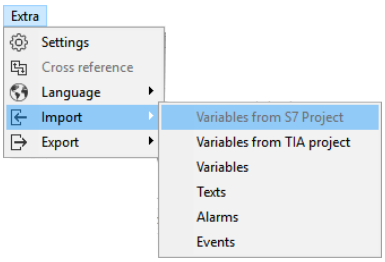
Beside the manually input of variables there are 2 better opportunities to import S7-variables from the S7-program

1) Import the variables by the system data file (*.sdf) of Simatic®-Manager (without symbols)

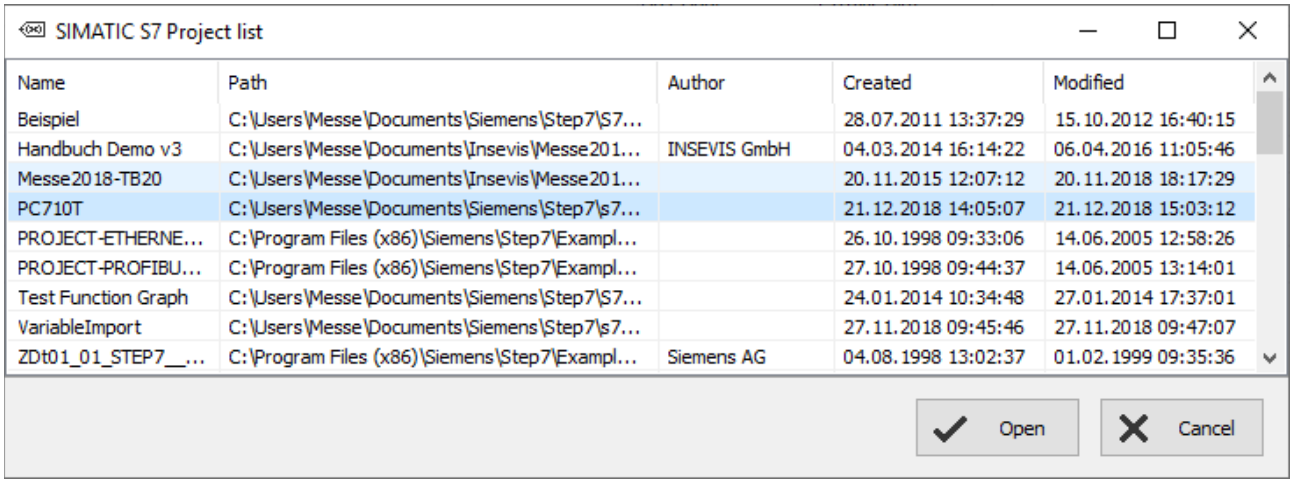
- Create a S7- symbol table by the Simatic®-Manager
- Import it as a variables list **(importable in sdf-file format)** by:
  → Extra
  → Import
  → Variables

2.) Import the variables directly from the DBs of the S7-project (including symbols)

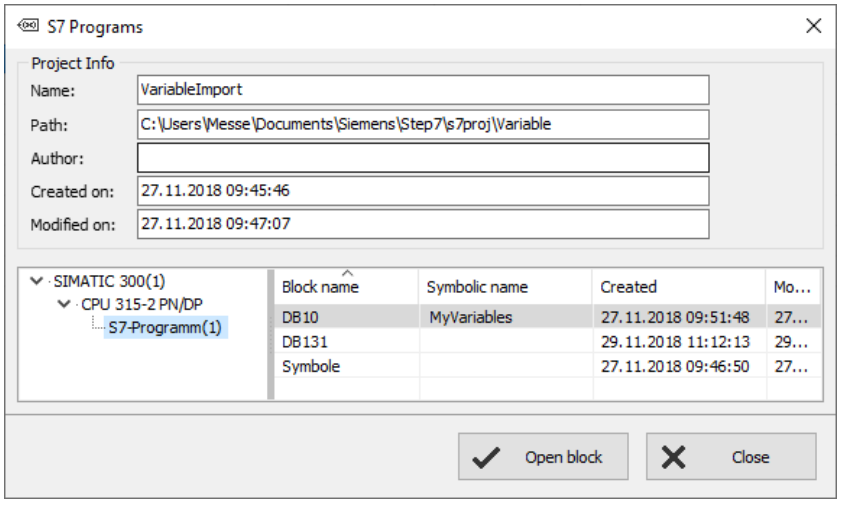
- Start it by
  → Extra
  → Import
  → Variables from S7-Project
  (Simatic®-Manager must be installed on this PC)



- Mark and open the corresponding project



- Choose the required data blocks from the S7-program list **(no instance data blocks).**



- Mark the required variables (also multiple) and import it.

VIDEO-Tutorials available Therefore is a **VIDEO** available in the VisuStage-Playlist at the INSEVIS YouTube®-channel „INSEVIS EN“.

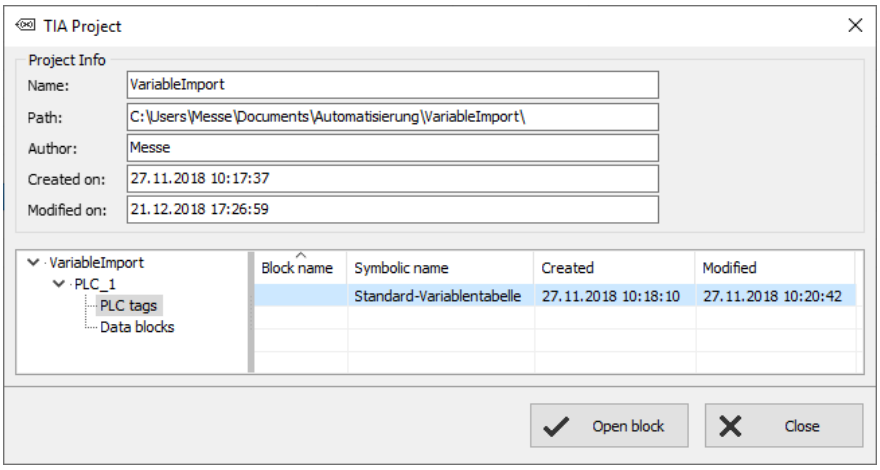
3.) Import the variables directly from the TIA®-project (including symbols)

3.1) Import the variables directly from S7-3xx-CPUs from TIA®-project

With VisuStage version 2.1.0.27 projects from TIA®-Portal versions 13...16 can be imported.
TIA®-Portal does not need to be installed on this PC, just only TIA®-project must be closed while importing.

- Start it by
  → Extra
  → Import from TIA®-project

Than a window opens to select the TIA®-project.

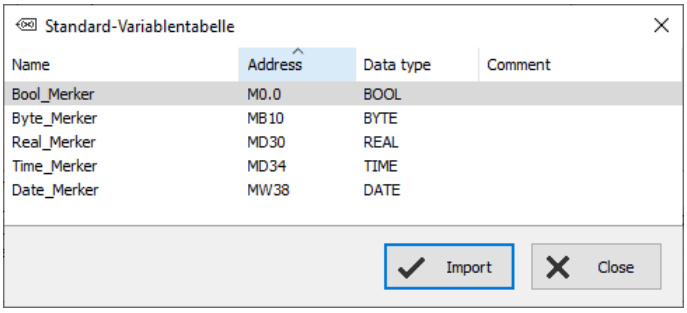


SELECTION

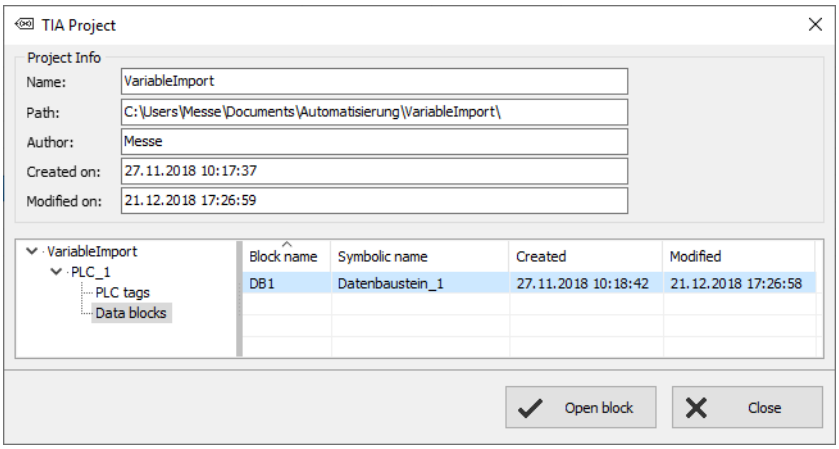
→ of the **Station** (here INSEVIS SPS)
→ and than of the **CPU** (here PLC_1)
and than

EITHER
of the **symbol table** (here SPS-Tags)

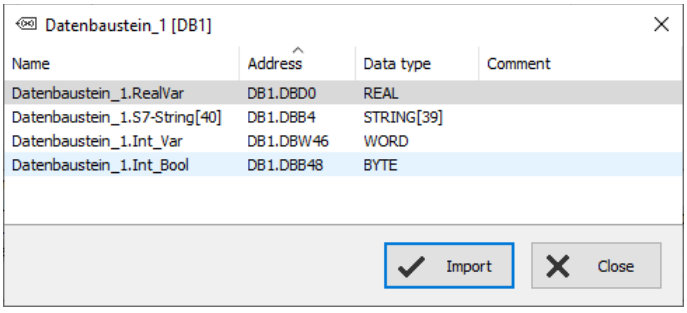
1. In the right window all variable tables will be displayed
2. Open up a variable table and see all existing variable lists.
3. Selection of the variable lists by marking (multiple marking by Shift or Ctrl)
4. Import the selected variable lists



OR
of the **data blocks** (here Data Blocks)



1. In the right window all data blocks will be displayed
2. Selection of the data block by marking (only single blocks)
3. Open a data block
4. Selection of variable from a data block by marking (multiple marking by Shift or Ctrl)
5. Import the selected variables from the data blocks



3.2) Import the variables directly from S7-12xx and 15xx CPUs from TIA®-project

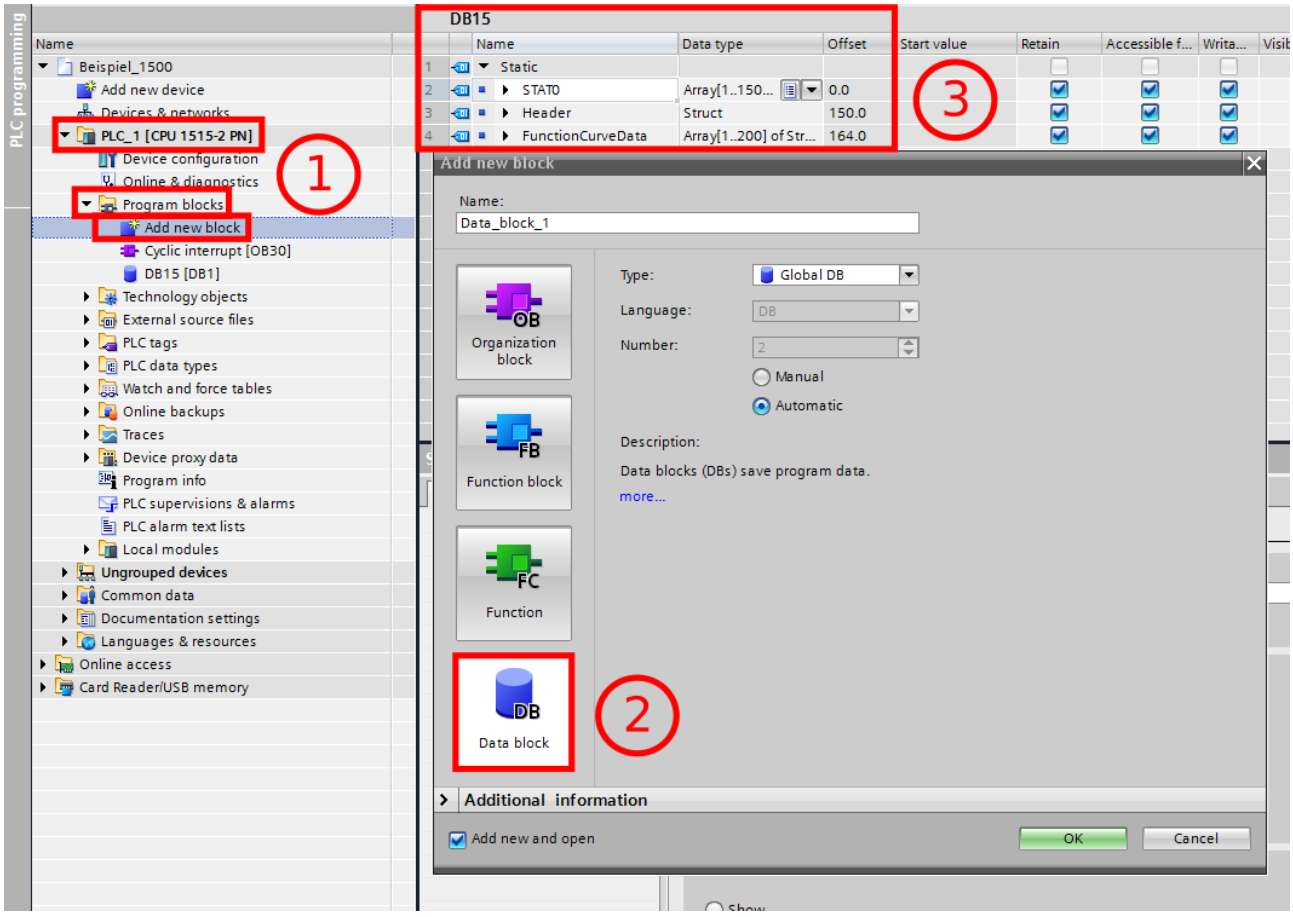
To import variables from projects which are based on CPUs from the 1200 and 1500 series they have to be stored inside a data block.

To achive this execute the following steps in your TIA project:

1. Select your CPU -> Program blocks -> Add new block

2. Select in the opened menue the type "Data block" and enter a name.
   Press OK to create the data block

3. In this new data block you can now create variables with name, data type and offset which can later be imported into the VisuStage.

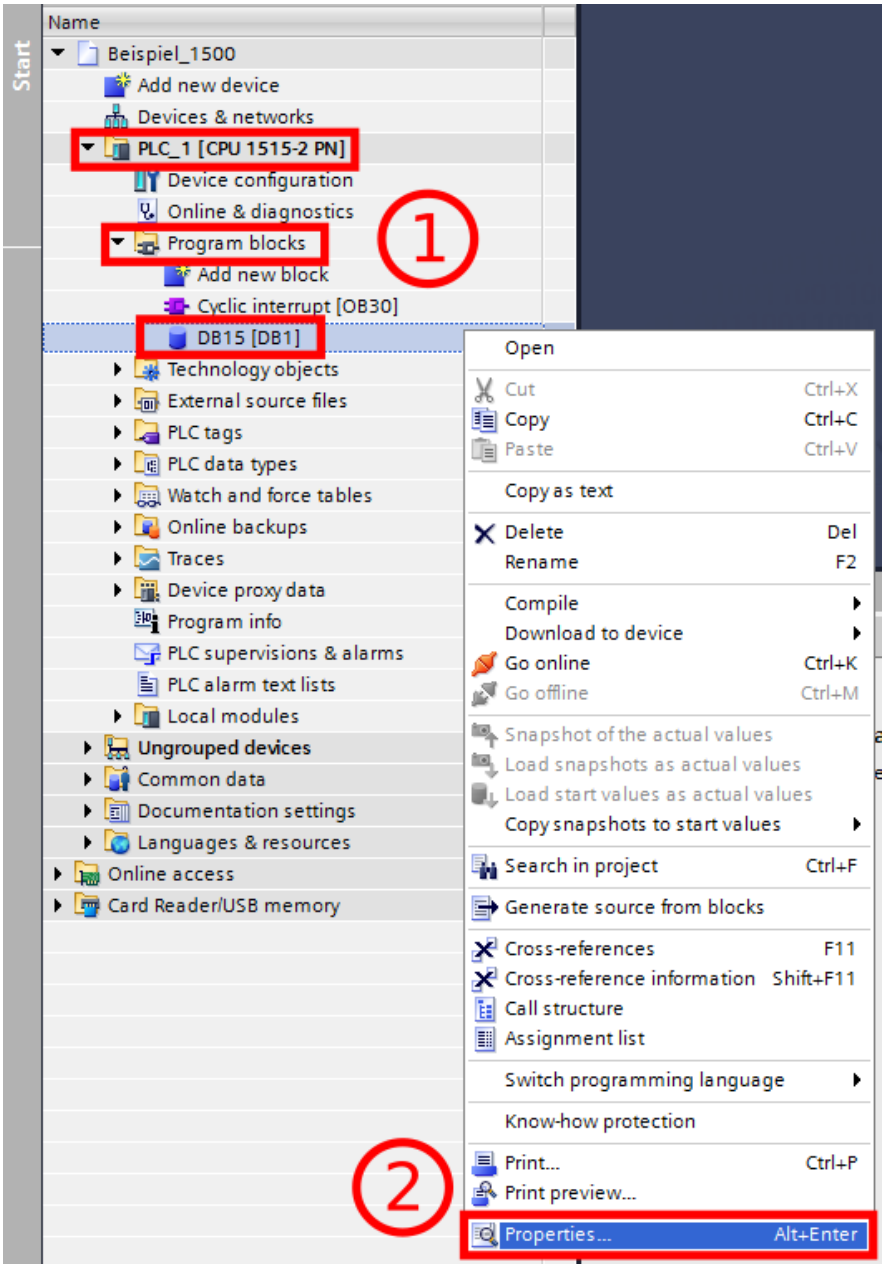


Visualization with the software „VisuStage“

To ensure a flawless communication between RemoteStage and S7-CPUs 12xx / 15xx, the block access must not be optimized. (only for S7-12xx and 15xx)

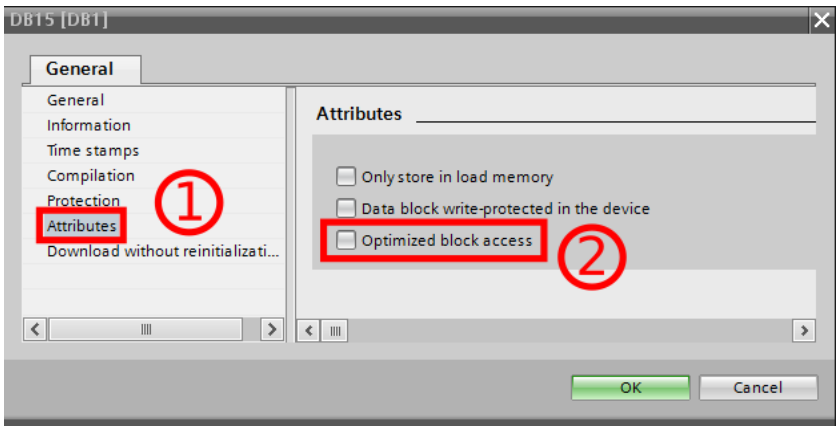
To achive this execute the following steps in your TIA project:

1. Select your CPU -> Program blocks
2. Do a right click on the concerning block and select "Properties..."



Execute the following steps in the opened window:

1. Select "Attributes"
2. Deactivate the option "Optimized block access"



In following cases VisuStage wants a confirmation before it starts a variable import:
- A variable with the same name exists already, but with another address or data type
- A variable with the same address exists already, but with another name or data type
- A variable with the same name and address exists already, but with another data type

ATTENTION:
When VisuStage overwrites the variables, the connections between variables and its objects will be kept.
When deleting the variables manually these connections will be deleted as well.

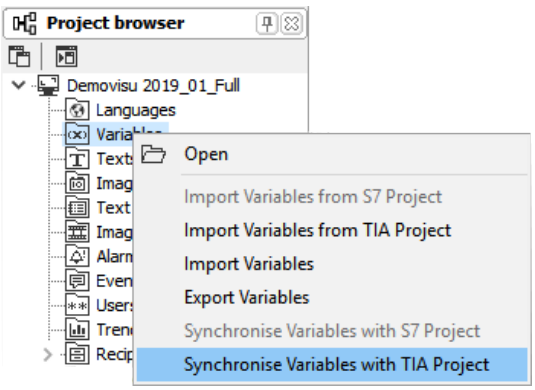
The **Export** of the variables is done in the same formats as well:
- sdf- file format (System data format, variables only), or in
- sym- file format (Symbol table format with variables and data blocks)

Synchronization of variables with TIA®-Portal/ Simatic®-Manager

By inserting, deleting and other manipulating of S7-variables in programming systems TIA®-Portal and SimaticManager the absolute addresses in referring S7-programs change. To synchronize these changes easily with the variables in the VisuStage-project as function „Synchronize with S7 /TIA® Project “ is available.

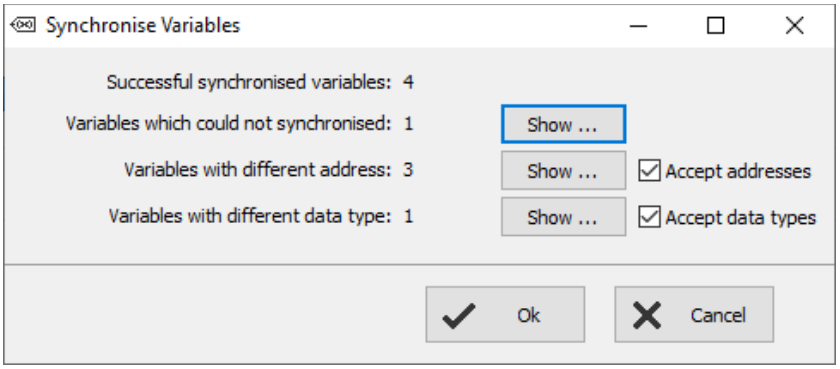
Go to resource „Variables“ or
directly into table of variables

- right mouse key opens Pull-Down menue
- if Simatic-®Manager is installed on this PC, synchronization with S7-Projects is possible
- to synchronize with TIA® projects insert the path to this project (It must be compiled and closed before)
- all variables of the VisuStage project will be detected and synchronized with S7/TIA now



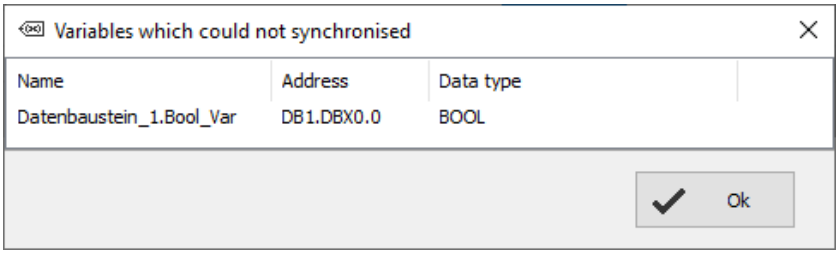
As result a synchronization table will be displayed, where

In the 1st line
- Successfully synchronized variables (without differences – where nothing was changed) are mentioned/displayed only.
- There is nothing to do for that variables, just for your information.



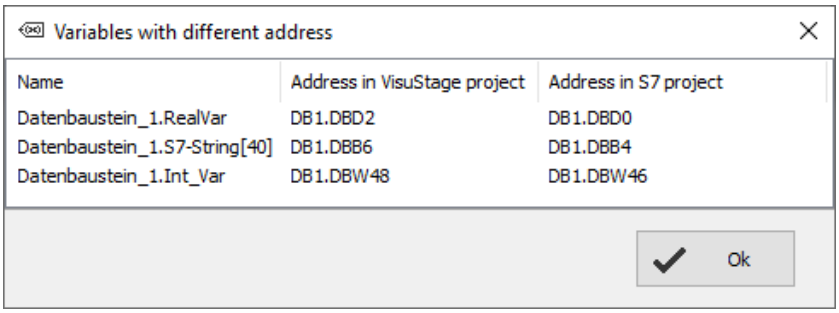
In the 2nd line:

- Not synchronized variables (where the symbolic variables name was not available in the S7-/TIA®-project) are mentioned/displayed only.
- These must be **deleted manually**, to keep the overview (First delete the connection in the visualization, otherwise you will earn compile errors)



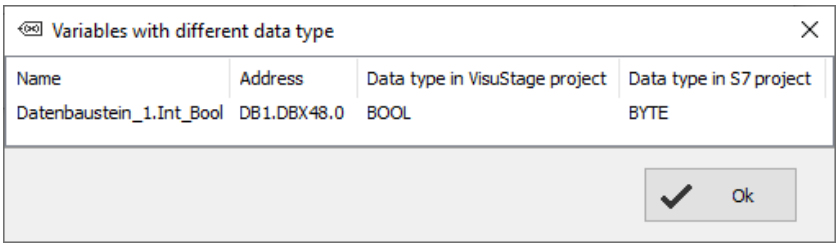
In the 3rd line:

- variables where at least the absolute address (or also the data type) was changed in the S7-/TIA®-project will be displayed.
- This is a proposal for modification only and **must be accepted** by checking a hook in the check box, than all these variables will be synchronized with referring variables in VisuStage-project automatically.



In the 4th line:

- variables where only the data type was changed in the S7-/TIA®-project will be displayed.
- This is a proposal for modification only and **must be accepted** by checking a hook in the check box, than all these variables will be synchronized with referring variables in VisuStage-project automatically.



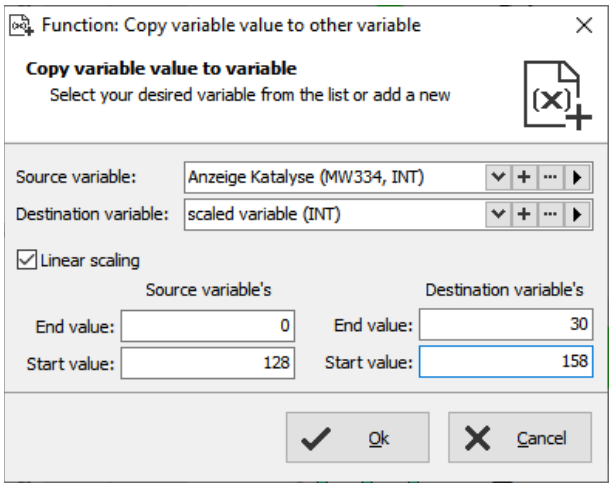
If there was created new variables in S7-/TIA®-project, import them or add it manually new.

Internal variables

These variables are available internally to the visualisation and are listed in a separate tab.

Using the key function "Copy variable value", the values of these variables can be transferred to PLC variables and vice versa.

With indirect variables, e.g. linear scaling is possible through this function (see picture on the right).



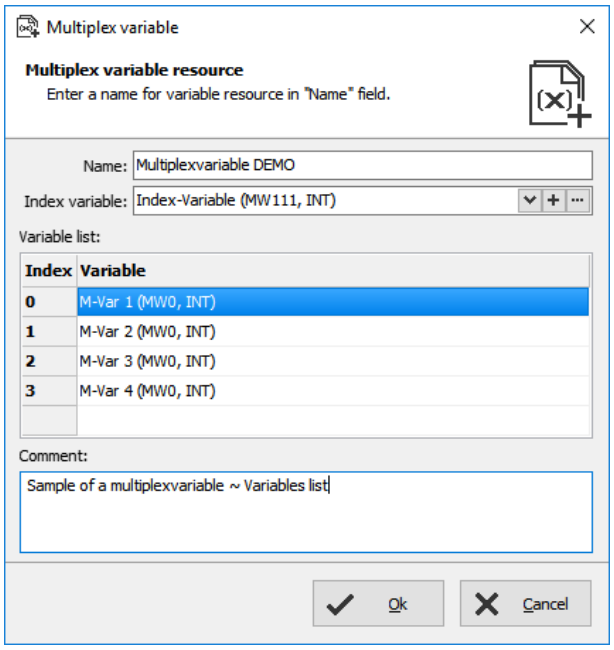
Multiplex variables (variables list)

With multiplex variables it is possible to use one I/O-field multiple times. Depending on the value of the index variable only the pre-assigned and referring the multiplex variables will be showed at run time.

At the resource “Variable”, at the rider “Multiplex variables” can be created (see sample right beside). It work similar to text- and image lists, but only with variables.

Only whole-number variables will be accepted as index variable, the variables in the variable list are free (STRNG-types are not to mix with other types). It also can be DBs, Merkers, inputs or outputs.

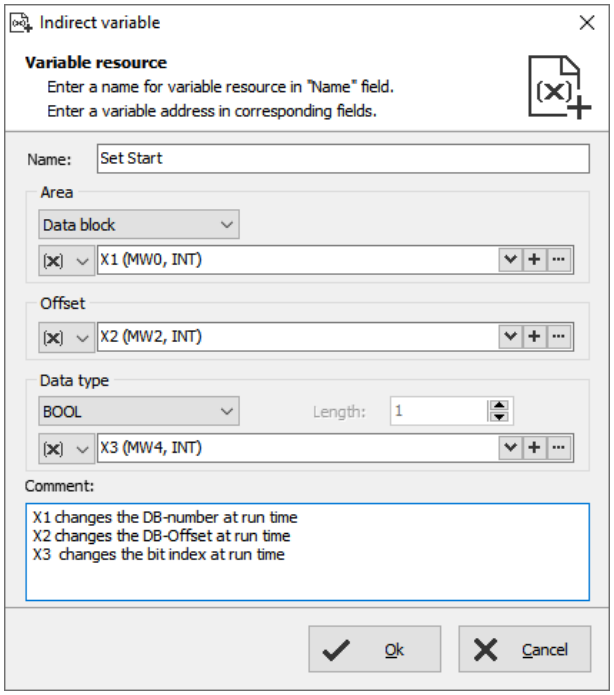
Index	Variable
0	Variable (DB1.DBW0, INT)
1	Variable1 (DB4.DBW10, INT)
2	Variable2 (DB10.DBW0, INT)



Indirect variables

Alternatively to the Multiplex variables (quasi the list of variables) the number of variables can be reduced by far, if the address range can be manipulated individually at run time by one or more variables.

The indirect variables allow by help of one or more variables a multitude of memory cells in the address range of the PLC. These variable memory areas can be written or read out, without defining a fixed variable for each before.



Sample with data type BOOL

Fix addressed
(512 variables)

DB 0 DBX 0 . 0
DB 0 DBX 0 . 1
…
DB 0 DBX 0 . 0
DB 0 DBX 0 . 7

DB 0 DBX 1 . 0...7
DB 0 DBX 2 . 0...7
…
DB 0 DBX 6 . 0...7
DB 0 DBX 7 . 0...7

DB 0 DBX 0...7 . 0...7
DB 1 DBX 0...7 . 0...7
…
DB 6 DBX 0...7 . 0...7
DB 7 DBX 0...7 . 0...7

indirectly addressed
(4 variables)

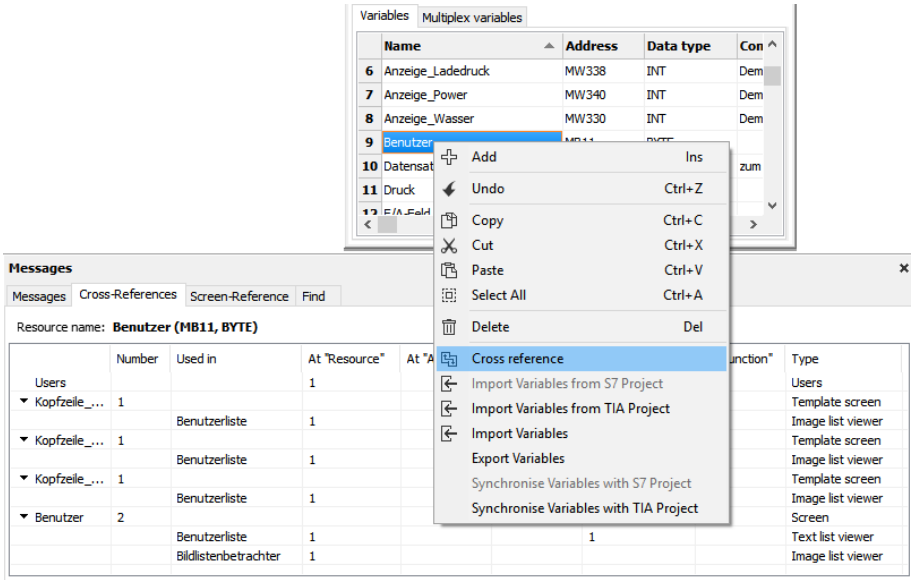
DB X1 DBX X2 . X3
X1 = 0...7
X2 = 0...7
X3 = 0...7

Visualization with the software „VisuStage“

Cross references for variables

To keep the overview about the variables used in the project, there is a function „Cross reference“ (mark a variable and use right mouse button). All variables will be displayed referring to their usage in resources or in menu objects.

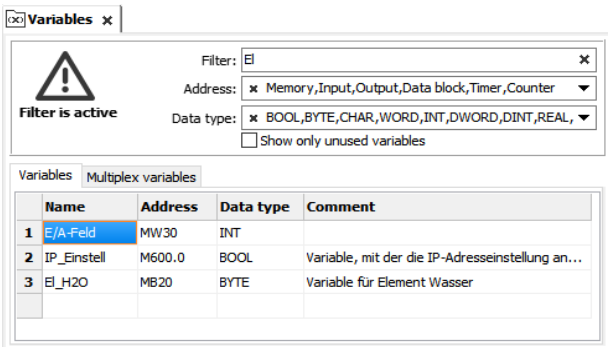
By a double click on an item you get automatically into the menu, where it is used. For sample the “Benutzer” variable was used. Close/open the message window by key „F6“.



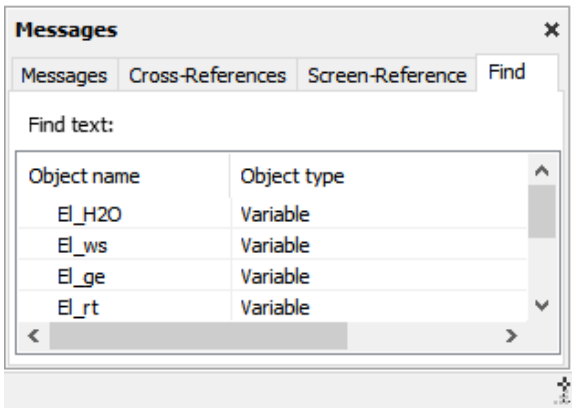
Set filter or find variables

Variables can be filtered for:

- Name
- Address- and
- Data type

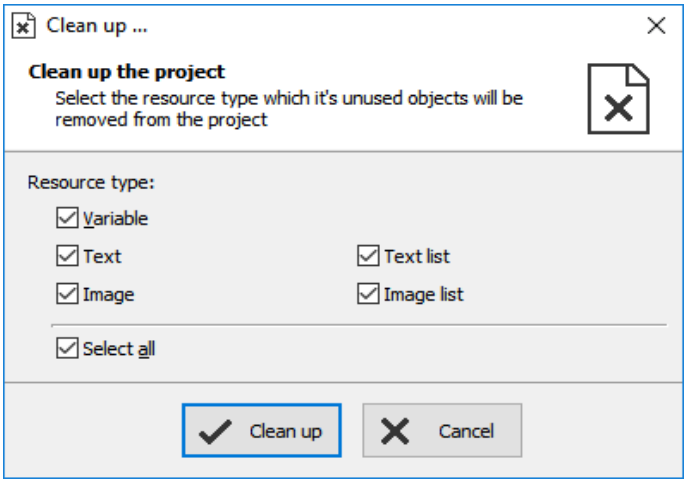


The result is shown in the message window (show/hide with key „F6“)



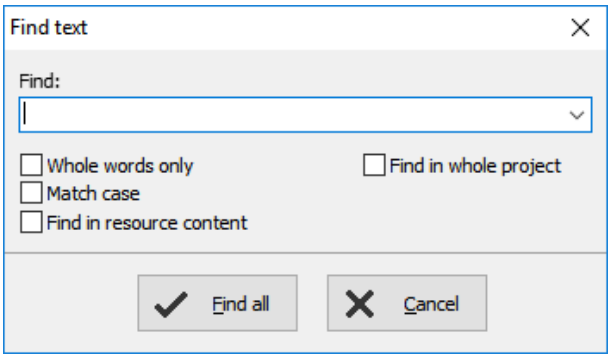
Clean up unused variables

Delete unused resources with the „CleanUp“-function at the rider „Edit“ and keep your project small and simple.



Find variables by text phrases

Find resources with the „Find“-function <CTRL+F> at the rider „Edit“ to keep an overview.

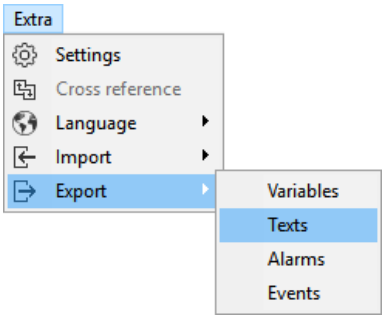


Resources Texts and Text lists

Texts

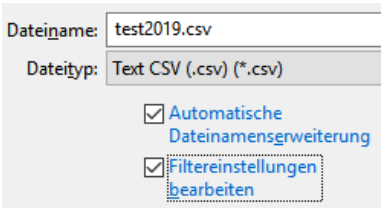
Will be created in the project language and can be imported or exported as a csv-file
(This helps for external translation.)

- Export by:→ Extra → Export → Texts
- Choose the languages to be exported
- select your csv-format and save the csv-file.
- Open/import it with a table calculation program (csv-settings must be the same!)
- Edit the text while keep the structure as it is (no cells may be removed or added!)
- Import by → Extra → Import → Texts



In the first lines there are system texts. Is a text created in the project language only and not in another language, it will be displayed in all other language as it was created in the project language (see line 5).

Hint:
If import is failed, your table calculating program removed the field delimiters or text separators. To prevent this, do „Save as“ and activate the „Assign filter settings“ function.



Hint: at some Windows7 installations it is necessary to check “Quote / Set All text cells”

Text lists

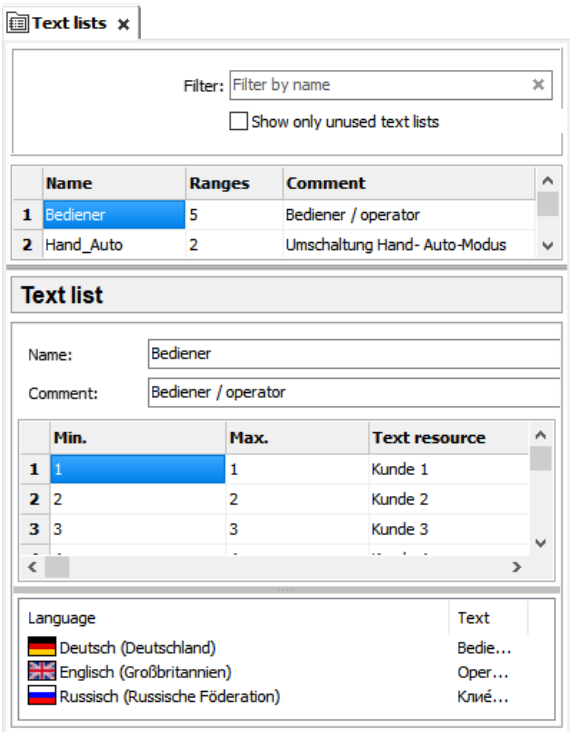
(when different texts should be displayed depending on a variable, e.g. in the user administration - operator, master etc.)

Will be created manually

- first define the list,
- than add existing text resources or create new ones
- than add limiting values **(Min/Max)** for each element of the text list

or

- mark multiple texts in resource “texts”
- press right mouse button and select “create text list”
- than add limiting values **(Min/Max)**
- and add a **name**

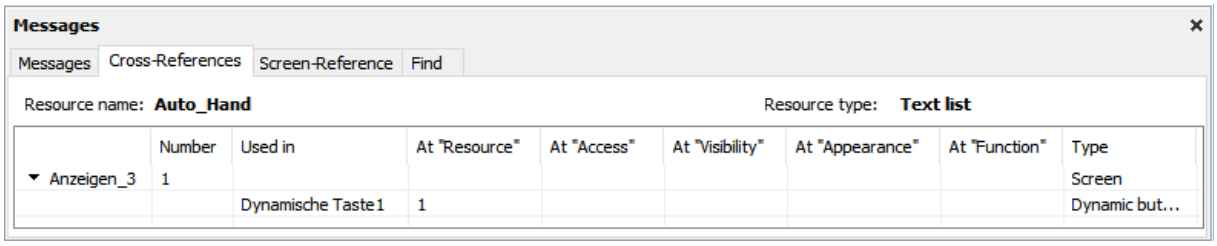


ATTENTION:
If there is active another value as these what are assigned in line 1...n, than ALWAYS this text resource will be displayed, what is assigned FIRST (in first line / line 1).

Cross references for texts / text lists

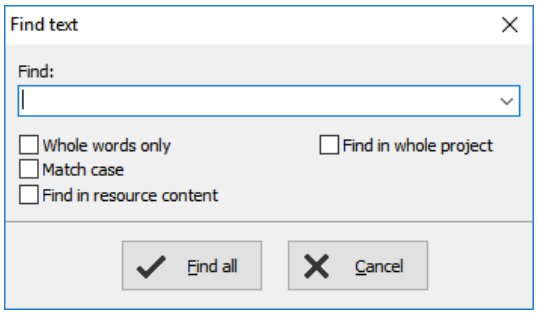
To keep the overview about the texts/ text lists used in the project, there is a function „Cross reference“ in the resource pool. (Go to resources Text/ text list, mark a text/ text list and use right mouse button). All texts / text lists will be displayed referring to their usage in resources or in menu objects.

By a double click on an item you get automatically into the menu, where it is used. Close/open the message window by „F6“.



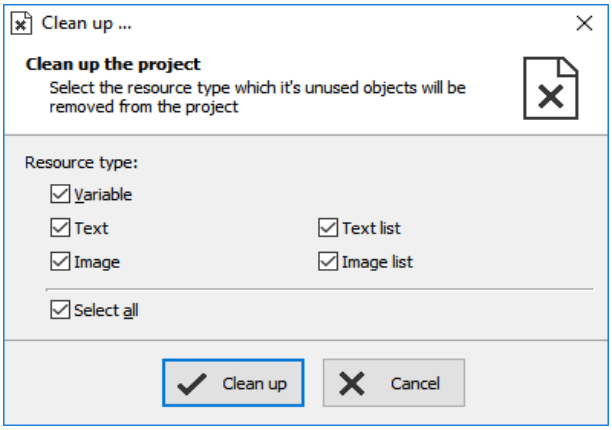
Find texts / text lists by text phrases

Find resources with the „Find“-function <CTRL+F> at the rider „Edit“ to keep an overview.



Clean up unused texts / text lists

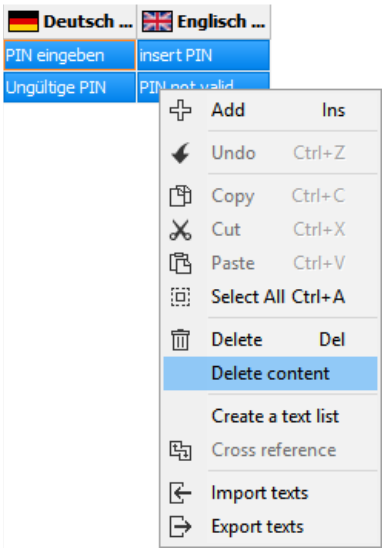
Delete unused resources with the „CleanUp“-function at the rider „Edit“ and keep your project small and simple.



Delete single text contents in text resource list

It is possible to delete single text contents in the text resource list.
Mark the desired cells by mouse and click right mouse key.
With “Delete content” the content of these cells will be deleted.

Alternatively:
If you press the “delete”-button of your keyboard the marked text resources will be deleted completely.



Resources Images and Image lists

Images

INSEVIS delivers some sample images with the VisuStage.
Creating own images is a child´s play by any free software.

- Recommended file format: PNG because of the transparent alpha-channel.
  Substitute a color by transparency and save the image as PNG (Interlaced / Automatic recommended).
- Images of the project-language will be used for all other languages as well
  → shown half transparency
- It is possible to use other images in other languages instead the image of the project language
  → by right mouse key assign a new one and it will be shown with full color

Image lists

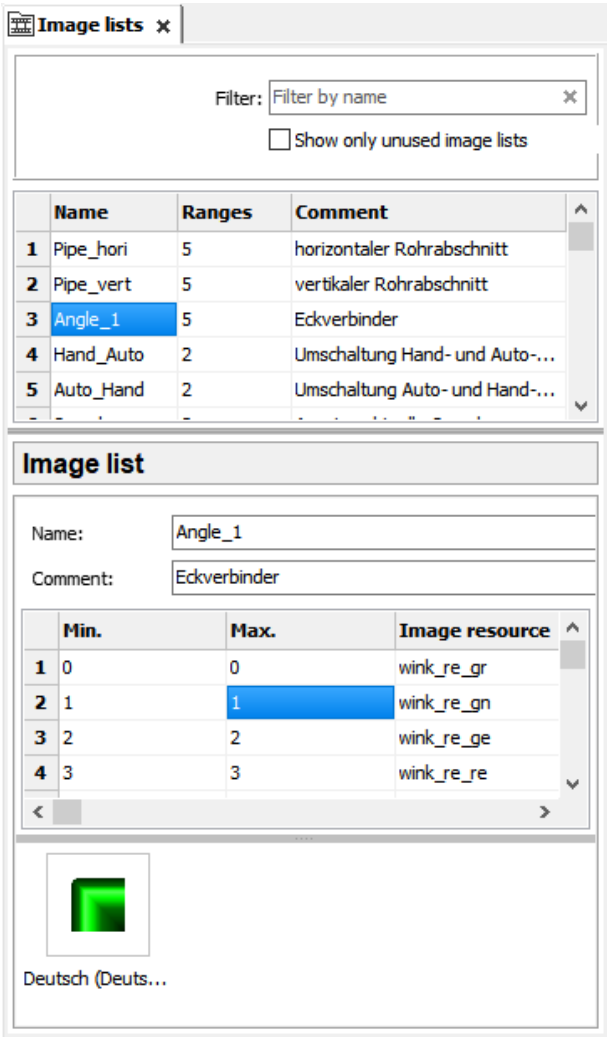
These lists are the best tool for color changes of an object or toggling images caused by changing variable values.

Will be created manually

- right mouse key in the field **Name**,
  (add the new name of the image list)
- Add limiting values for each element of the image list
- Assign an existing image resource,
  (or add a new one in this field directly)

or

- mark multiple images in resource “images”
- press right mouse button and select “create imagelist”
- than add limiting values **(Min/Max)**
- and add a **name**

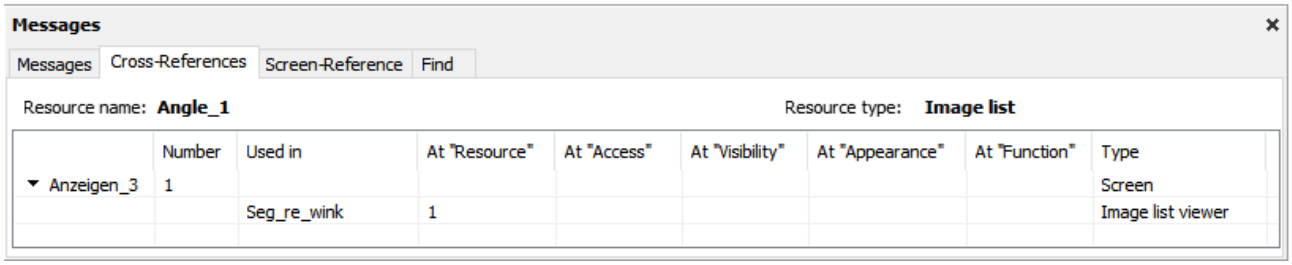


ATTENTION:
If there is active another value as these what are assigned in line 1...n, than ALWAYS this text resource will be displayed, what is assigned FIRST (in first line / line 1).

Cross references for images / image lists

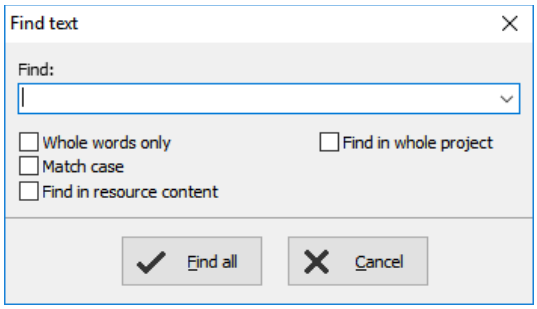
To keep the overview about the images / image lists used in the project, there is a function „Cross reference“ (mark a image / image list and use right mouse button). All images / image lists will be displayed referring to their usage in resources or in menu objects.

By a double click on an item you get automatically into the menu, where it is used.
Close/open the message window by key „F6“.



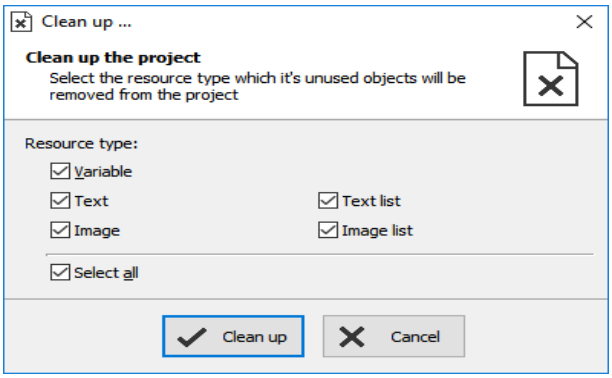
Find images/image lists by text phrases

Find resources with the „Find“-function <CTRL+F>
at the rider „Edit“ to keep an overview.



Clean up unused images/image lists

Delete unused resources with the „CleanUp“-function
at the rider „Edit“ and keep your project small and simple.



Multiple replacement of resources

If you organize your symbolic stringently you can swap multiple resources

- Variables,
- Texts,
- Text lists,
- Images,
- Image lists.

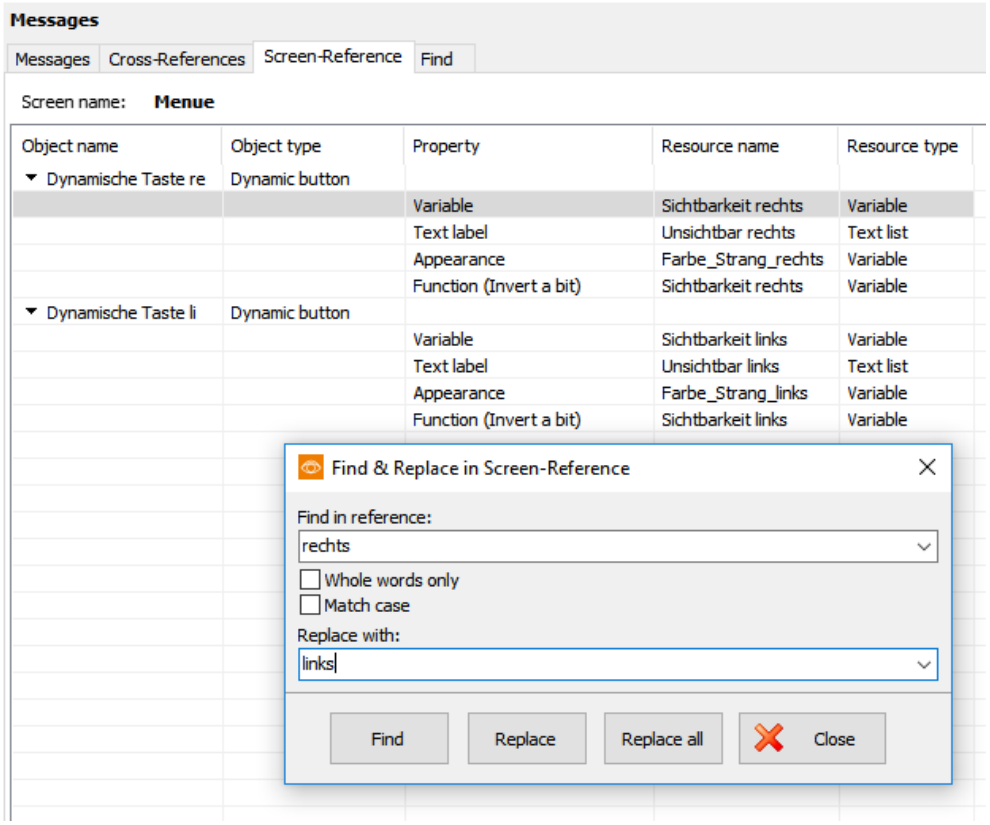
If they have similar symbolic phrases in their name.

In the sample on the right a „Dynamische Taste re“ is described with different properties by different resources „*** rechts“.

After copying an object like this it is necessary to swap all resources at the new copied object step by step. This needs time while multiple copying.

The function „Find & Replace“ allows a fast swapping.

Click with the right mouse key directly on the screen background (not on an object!), than select “Show references” and the message window opens up (close with F6, refresh by closing and re-opening)



There all resources such as with „rechts“ in their name will be replaced into others like such as with „links“ in their name.
(… but only, if you organized you symbolic stringently ...)

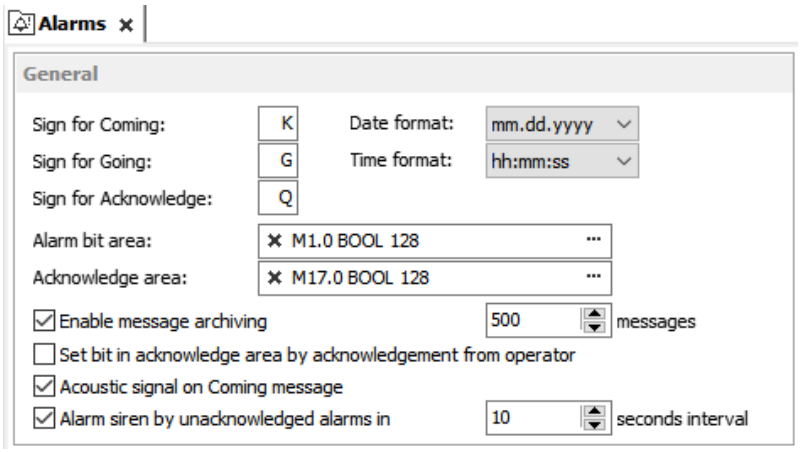
Resource Messages (Alarms and Events)

Alarms / Events

It can be processed
- with CPU-V 128 alarms and 128 events
- with CPU-P/T 1024 alarms and 1024 events

It can be archived messages
(each: alarms and events). in total:
500 for (CPU-V/-P-devices)
or 50.000 (for CPU-T-devices)

(These archives can be deleted from Micro-SD-card
by a VisuStage button function)



- Can be imported or exported as a csv-file (This helps for external translation)
- Import by → Extra → Import → Alarms / Events
- A new creation proceeds in project languages set default before.
- To archive it enable message archiving by a check mark and insert a micro-SD-card (Archiving is done in a binary fil format. RemoteStage converts this binaries into csv, alt automated in batch files)
- Acoustic signal means a single sound by an internal buzzer
- Alarm siren means an one-second-sound of the internal buzzer, what is played once between 10 and 600 seconds.

Alarm-(Event-) Bit area (Trigger bit address area)
is 128 bit at CPU-type V and 1024 bit at CPU-type P
and must be in a row without gaps!
Each bit belongs to a message (alarm/ event) (bit 0 to Alarm/Event 1, bit 1 to Alarm/ Event 2, etc...)

The message state can be controlled by the S7-Program by a bit from this area.
With bit=1 this message appears as „coming“.
With bit=0 this message will be interpreted as„going“.

Acknowledge bit area (only for alarms)
is 128 bit at CPU-type V and 1024 bit at CPU-type P. too
and must be in a row without gaps!.

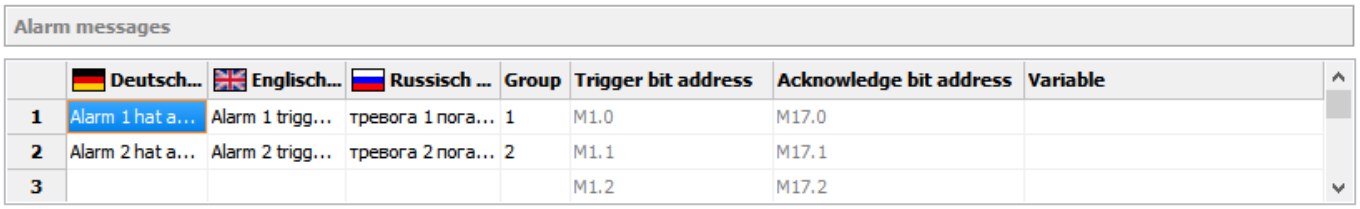
Each bit belongs to an alarm (bit 0 to alarm 1, bit 1 to alarm 2, etc...)

The alarm state can be controlled by the S7-Program by a bit from this area.
With bit=1 this alarm appears as „acknowledged“.
With bit=0 this message will be interpreted as „not acknowledged“.

Both areas may not overlap each other (error message appears at compiling)

ATTENTION:
The block for Trigger- and Acknowledge areas **MUSST EXIST** in the PLC and must have at least that size 128/1024Bit, otherwise messages will not be displayed!
If the acknowledge bit is **activated**, it must be **reset** in the PLC after a while, otherwise the next of these alarms appears as „acknowledged“.
If it is **not activated** and you create and use a visualization with the RemoteStage there will be set no bit in the PLC when you acknowledge the alarm on the RemoteStage-visualization and on any other panels/HMI this alarm stays **not acknowledged** there.

Alarm/ Event-Message box



- Alarms of the same **Group** are acknowledged together / Alarms of Group 0 are acknowledged **automatically**.
- **Trigger bit / Acknowledge bit areas** are displayed for better information only and are not changeable in this view!
- An additional **variable** *(only for Panel-PLCs not for Panel-HMIs)* offers to display a dynamic optional error code as decimal number.
  (e.g. an calculated value of an special factor for customers needs as decimal number)
  → This is not a trigger variable for the message!
- Events can be displayed ascending or descending in the archive viewer
- By “Variable” a variables value (INT or REAL with (fix) 2 decimal places) can be displayed at the end of the message line.
- Like at the text resources it is possible to delete single text contents in the alarm / event message box.
  → Mark the desired cells by mouse and click right mouse key.
  → With “Delete content” the content of these cells will be deleted.
  Alternatively:
  → If you press the “delete”-button of your keyboard
  → the marked text resources will be deleted completely.

Resource User administration

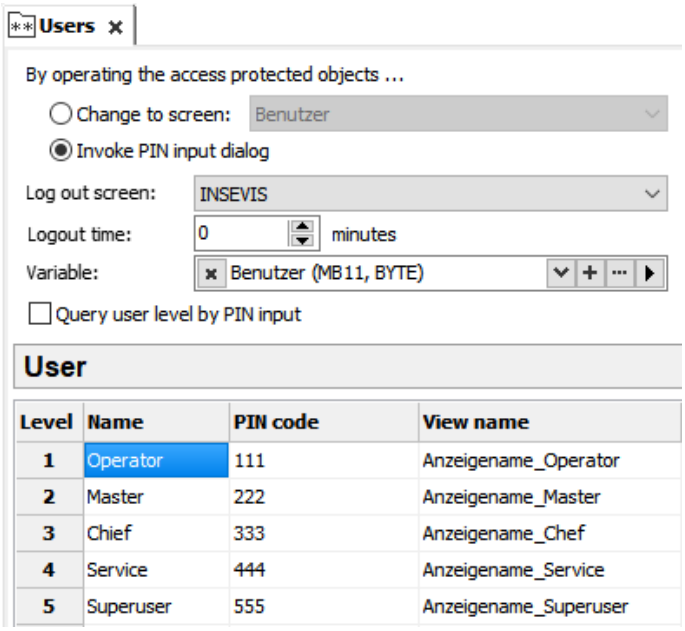
There can administrated be up to 9 users with passwords by 8 digit - PIN-codes. Users with level 9 have all access rights, those with level 1 have least access rights.

In the upper mask can be assigned

- the target screens, whereto the project should change after login/ logout or if there shall be called the PIN-input dialog directly after touching a button with access rights.and
- an automatic logout time (in minutes). (With logout time 0 is assigned no automatic logout.) and
- a variable, what informs the PLC, what user level is active yet and
- an additional query to identify the user level before PIN input

In the lower mask can be assigned

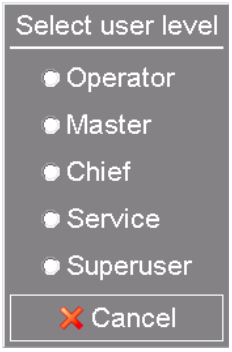
- the user names and PIN-codes of each levels (1...9).
- A new user will be added by entering its name and PIN-code
- The “view name” (multilingual resources, what displays the present user name by using system variable $(CurrentUserViewName) in text fields (only CPU-T-devices) see sample visualization



Level	Name	PIN code	View name
1	Operator	111	Anzeigename_Operator
2	Master	222	Anzeigename_Master
3	Chief	333	Anzeigename_Chef
4	Service	444	Anzeigename_Service
5	Superuser	555	Anzeigename_Superuser

“Query user level by PIN input

Just before entering PIN code it is possible to activate a identification mask, where the user enters his level.



All lines can be displayed single-lingual or -by selecting different text resources- multi lingual in every project language. To be assigned at the resource “Dialogs / user level select”

Hint:

- The target menu screen after successfully log in will be assigned at the dynamic object „**PIN-input field**“
- With the **PIN-Change dialog- function** a PIN can be changed by the operator (his level stays the same). If this PIN is forgotten, a new upload of the visualization binary can set it back to delivery state.
- For a **user-level-referring screen** change use the function „Change screen with PIN input“ on a button.

Change user level by SFC215 "LOGIN"

The SFC215 "LOGIN" enables the user level change by S7-User program at Runtime.

Parameter	Declaration	Data type	Description
LADDR	INPUT	WORD	Logical base address = 0 (for all Panel-PLC) Logical base address = 1..16 (Connection ID number to Panel-HMI)
LEVEL	INPUT	BYTE	User level (1..9) 0 = Logout current user level
RET_VAL	OUTPUT	WORD	Error code W#16#0000 – No error W#16#8002 – VisuStage is not running W#16#8005 – Level is invalid or not configured W#16#8xyy – General error codes (compatible to STEP 7) only for Panel-HMI: W#16#7000 – Job is in process W#16#7001 – Invalid connection state e.g not connected W#16#7002 – Job is not accepted, because another job is proceeded actually W#16#7003 – Connection resource occupied (temporarily locked) W#16#8001 – Invalid connection ID number or connection is not configured
BUSY	OUTPUT	BOOL	TRUE: Job is in processing

S7-User program example to change user level 7

Target: Change to user level 7

```
UN    M   1502.4                 // if change request is not set, then
BEB                               // end the function

CALL  SFC  215
 LADDR  :=MW2000                  // VisuStage connection ID number.
                                  // In Panel-PLC (e.g PC710T) must be W#16#0.
 LEVEL  :=B#16#7                  // User level, could be 0..9
                                  // 0= Logout current user level
 RET_VAL:=MW1000                  // Return / result code
 BUSY   :=M1702.3                 // TRUE = Reading is in process

U     M   1702.3                  // Job is in process
BEB
R     M   1502.4                  // reset request signal

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

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

BEA                               // end of function
```

Resource Partner-PLC (for HMI only)

INSEVIS HMIs run with data types of S7-300-family only. They use Ethernet S7-communication (Put/Get) to communicate. The HMI communicates with its partner device as active partner by via Ethernet S7-communication (put/get). In that communication INSEVIS-HMIs are the ACTIVE part. The referring partner-device will be identified by IP-address and TSAP (made of rack-no. / slot-no. of CPU / and hex no. of this connection resource). One HMI can only communicate with ONE partner-device (CPU).

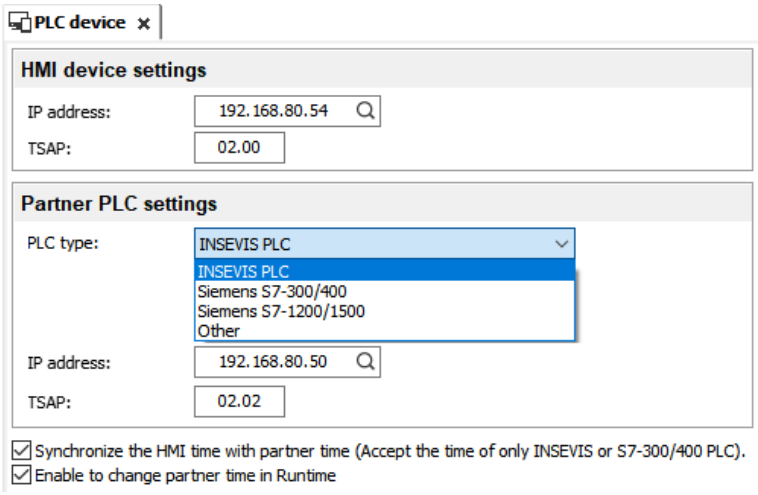
Connection setting in the HMI

- Either assign it in the BIOS of the Panel-HMIs (put out the Ethernet-cable, wait for the logo in the screen and than press left/right/left side of the screen) or
- in VisuStage-project at the resource partner device

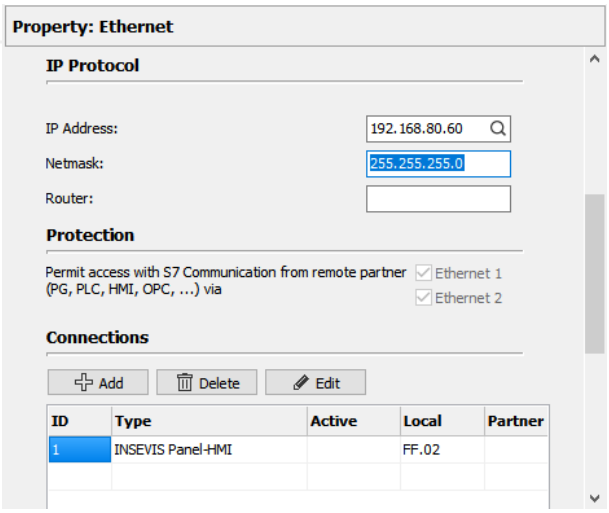
Connection settings in the Partner-device

- If general no connections must be configured in the partner device (PLC) for the HMI because the HMI is the actcive part of S7-communication (put/get) and uses the connection resource no.* 02 (reserved for OP-communication). (Screen- and language can be changed from PLC by using variables with screen- or language-ID-no.)
- If the **PLC uses SFCs for changing screens or handle trends** in the Panel-HMI shall change screens and languages at the INSEVIS-HMI, an Ethernet connection with connection resource no.* > 03 (hex), best FF must be assigned (e.g. in ConfigStage or in Siemens-programming tool)

below: assignment of connection parameter at HMI´s view in the VisuStage (TSAP can be entered by selecting a partner-PLC automatically by rules below)



below: assignment of connection parameter at PLC´s view in the ConfigStage



For INSEVIS-Panels and Siemens-CPUs

1. Assign the IP-address of the partner-PLC into the field „Partner-PLC“ in VisuStage-project or in HMI-BIOS
2. Configure the TSAP by assigning **CPU rack-no. and -slot-no. and connection resource no.* (hex)**
   for Siemens-CPUs 300/ 400 it is **0, 2, 02,**
   for Siemens-CPUs 1200/1500 it is **0, 1, 02** → allow “Put/Get” communication
   → do not use optimized DBs

*** Please do not confuse the connection resource no. 04...FF (hex) with the connection-ID-no. (01...16 (dec))!**

Partner-PLC Synchronization (for HMI only)

When assigning partner device a time synchronization between panel and PLC is highly recommended.

1st check box: The HMI will be synchronized to the PLC-time
2nd check box: The PLC-time can be changed by the panel at runtime

Attention:
If you do not synchronize panel and PLC the difference between both devices may lead to unexpected program behavior.

Resource Dialogs

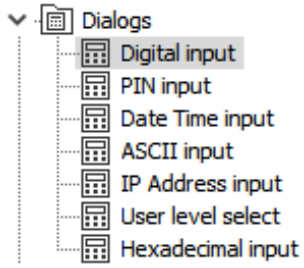
The dimension and the design of the virtual keyboards appearing at

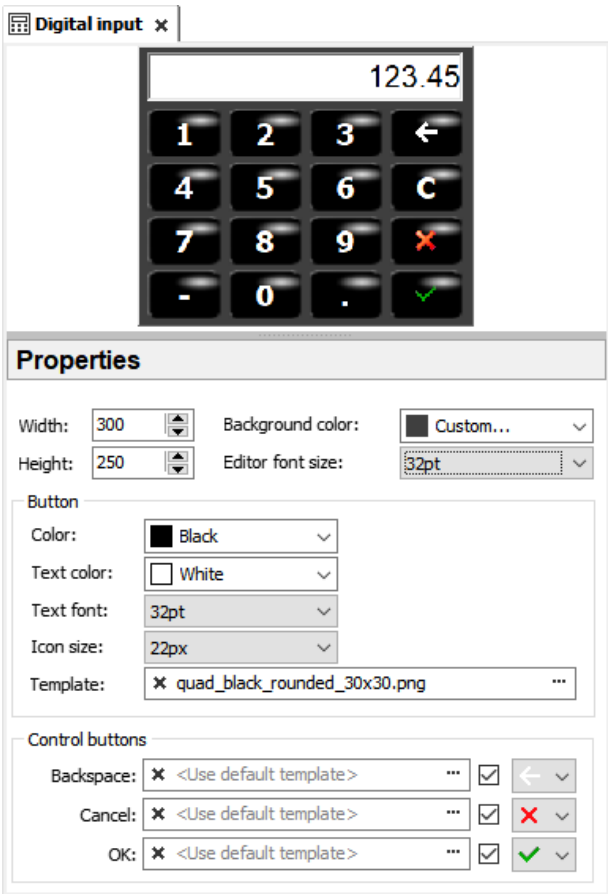
- I/O-field,
- Login field,
- Date-/Time field
- Stringinput field
- IP-address field
- User level select

can be adapted to ones needs in the resource „Dialogs“.

The size of the single keys is generated by the size of the whole object (Do not assign a size larger than the LCD-resolution!)

All other settings are self explaining.

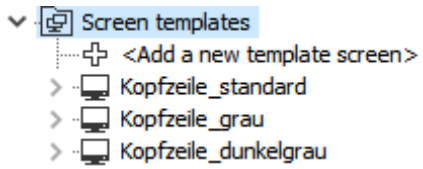




Hints:
At small panel diagonals it is very important benefit for the operator to easy enter his values that there is available a fitting virtual keyboard. This is not available at most other HMIs and a strong benefit for your solution.

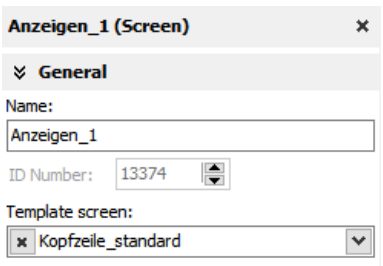
At the resource “user level select “ all lines (Header, single user levels and the cancel-button) can be displayed single-lingual or -by selecting different text resources- multi lingual in every project language.

Resource Screen templates



Screen templates (no image) can be created as a special resource and
can be uses once in every screen.

Therefore you activate this template screen in the settings of a screen you selected. Than choose one of the screen templates you made before.



Every modification in the screen templates will be taken over automatically into every screen this template will be used. Use this e.g. for top or bottom menus, headlines and so on.

Resources Trends

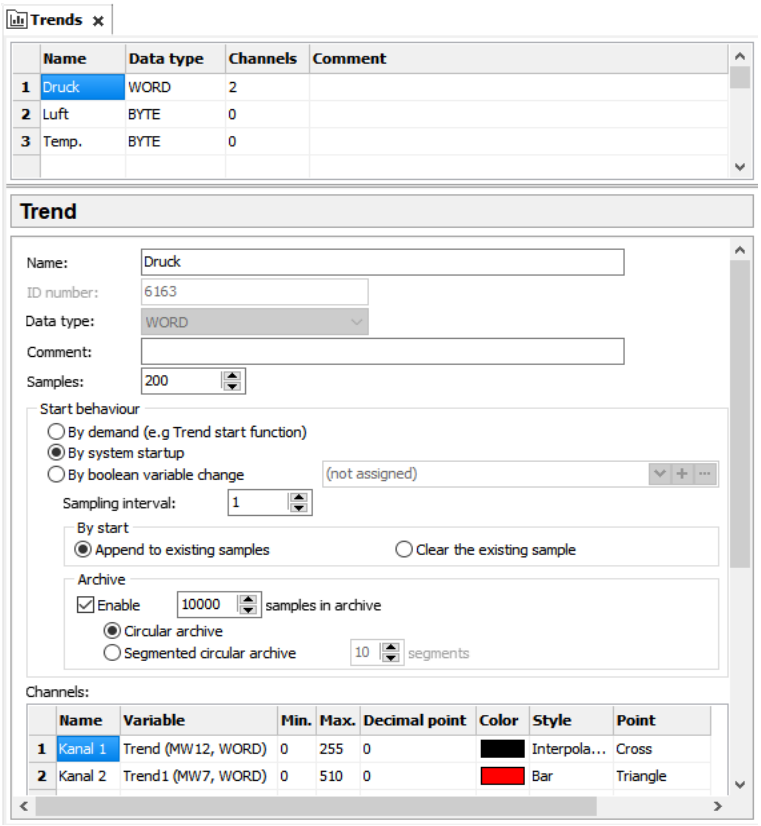
There can be processed 4 trends with 16 channels each. This represent values by time, smallest time base is 1 sec. All 16 channels of one trend need to have the same data format.

Trends will get a name - here “Druck” (pressure), what will be used for displaying and storing at RemoteStage as file name.

Samples: here you enter the number of samples in the circular buffer in runtime (100...200)

Start behavior: Beside S7-SFCs there are 3 additional ways to start a trending

- manually / by demand by assigning a function “Trend Start” to a button
  → Configuration at the function (limited archiving functionality to stay compatible to old versions)
- with each system start
  → Configuration at the trend resource settings
- depending on a variable
  → Configuration at the trend resource settings



If **Archive** was activated, you can

- enter the number of samples you want to archive in the **Circular archive** (max. 65635 samples per each trend channel) or
- combine up to 10 circular archives on the Micro-SD-card to an **Segmented circular archive** to 656.350 samples per each trend channel to display it in the trend archive viewer.

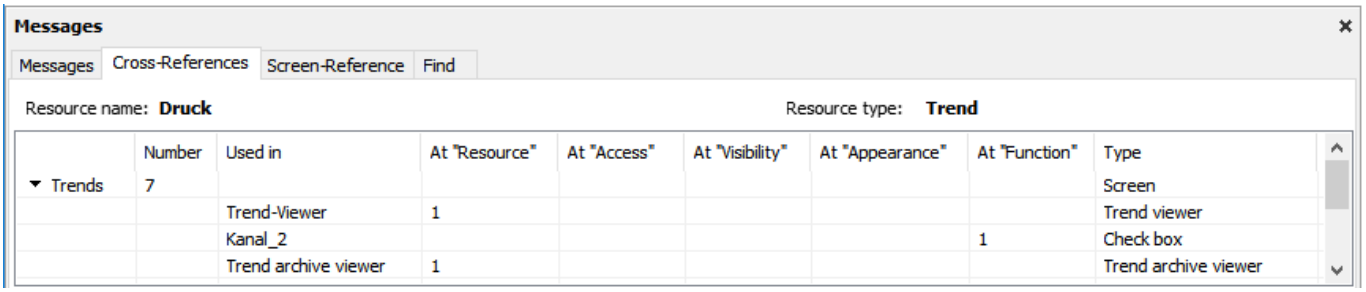
When number of samples exceed these limits, they will be overwritten in the FIFO system.

To configure the trend channel values, assign min/max values, decimal point, color, style (interpolated – shortest connection between 2 sample values, bars (vertically), stepped) and design of the sample point markers (diverse).

Archives will be archived to the external Micro-SD-card only. Each trend will be saved as own file. This binary will be converted by the software „RemoteStage“ on the PC into a csv- format either manually or in “Planned Tasks” of your Windows-PC. These archives can be deleted from Micro-SD-card by an special SFC, assigned as function to a button.

Cross references for trends

To keep the overview about the trends started, stopped, continued or displayed) in the project, there is a function „Cross reference“ (mark a trend and use right mouse button). All trend handling will be displayed referring to their usage in resources or in menu objects. By a double click on an item you get automatically into the menu, where it is used. Close/open the message window by key „F6“.



Starting the trend sampling with SFC 202 "TRENDSRT"

To start the configured Trend use SFC202 "TRENDSRT". The sampling interval and number of samples are assigned to the trend.
After starting, operating system samples the configured operand values in given interval and stores to Micro-SD card if „ARCHIVE“ parameter set to „TRUE“. If „CYCLIC“ parameter is „FALSE“, the „COUNT“ samples are one time sampled and trend automatically changes to „STOP“ state.

Trends are written to „**trend_xxx.bin**“ whereas xxx is Trend ID number. This archive file is based on „Ring-Buffer“ mechanism with „COUNT“ samples.

(Trend data can be get out of the PLC by SD-card only. In future versions a download by Ethernet will be possible.

Parameter	Declaration	Data type	Description
LADDR	INPUT	WORD	Logical base address = 0 (for all Panel-PLCs) Logical base address = 1...16 (connection-ID-No. to the Panel-HMI)
TREND	INPUT	INT	Trend ID number, (from VisuStage configured @ resources)
INTERVAL	INPUT	WORD	Sampling interval in seconds. 1..65535 second are allowed.
COUNT	INPUT	WORD	Number of samples to sample / to archive. 1..65535 are valid.
CYCLIC	INPUT	BOOL	TRUE: It cyclically samples the COUNT samples. FALSE: One time samples the COUNT samples, then stops the sampling
ARCHIVE	INPUT	BOOL	TRUE: Sampled values are stored in microSD card FALSE: No archive
RET_VAL	OUTPUT	WORD	Error code W#16#0000 – No error W#16#8002 – VisuStage is not running/configured W#16#8003 – Invalid „TREND“ ID number W#16#8004 – Invalid state / Already started. W#16#8005 – Invalid „INTERVAL“ value W#16#8006 – Invalid „COUNT“ value W#16#8xyy – General error codes (compatible to STEP®7) only for Panel-HMI: W#16#7000 – Job is in process W#16#7001 – Invalid connection state, e.g not connected W#16#7002 – Job not accepted, because another job is proceeded actually W#16#7003 – Connection resource occupied (temporarily locked) W#16#8001 – Invalid connection ID number or connection is not configured
BUSY	OUTPUT	BOOL	Busy flag, TRUE Job is in processing

If a trend is started once and the PLC has got a power OFF, the trend starts after power ON by itselfe automatically.

→ Here you will find more for parameterizing of the objects Trend viewer and trend archive viewer

S7-program sample for using the SFC202

Purpose: Start trend (ID number =6163) with configuration:

- 1s interval,
- cyclically,
- with archiving,
- ring buffer size 65535 samples (set of values)

```
      UN    M   1500.2       // if request is not set, then
      BEB                    // end the function

      CALL  SFC  202
       LADDR   :=W#16#0      // VisuStage connection ID number.
                             // In Panel-PLC (e.g PC700P) must be W#16#0.
       TREND   :=6163        // ID number of trend, which is configured in VisuStage
       INTERVAL:=W#16#1      // every 1s samples the values
       COUNT   :=W#16#FFFF   // max. no. of samples (set of values) 65535 (dez) => FFFF (hex)
       CYCLIC  :=TRUE        // cyclically samples as ring buffer.
                             // Buffer size is given in COUNT parameter
       ARCHIVE :=TRUE        // enables the archiving in micro SD card
       RET_VAL :=MW1000      // Result / return code
       BUSY    :=M1700.2     // TRUE = Request is in process

      U     M   1700.2       // If Start the trend is in process, then
      BEB                    // end this function
      R     M   1500.2       // reset request signal

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

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

      BEA                    // end of function
```

Stopping the trend sampling with SFC 203 "TRENDSTP"

The trend sampling can be stopped by SFC 204 “TRENDSTP“ function call.

Parameter	Declaration	Data type	Description
LADDR	INPUT	WORD	Logical base address = 0 (for all Panel-PLCs) Logical base address = 1...16 (connection-ID-No. to the Panel-HMI)
TREND	INPUT	INT	Trend ID number, (from VisuStage configured @ resources)
RET_VAL	OUTPUT	WORD	Error code W#16#0000 – No error W#16#8002 – VisuStage is not running/configured W#16#8003 – Invalid „TREND“ ID number W#16#8xyy – General error codes (compatible to STEP®7) only for Panel-HMI: W#16#7000 – Job is in process W#16#7001 – Invalid connection state, e.g not connected W#16#7002 – Job not accepted, because another job is proceeded actually W#16#7003 – Connection resource occupied (temporarily locked) W#16#8001 – Invalid connection ID number or connection is not configured
BUSY	OUTPUT	BOOL	Busy flag, TRUE Job is in processing

S7-program sample for using the SFC202

Purpose: Stop the trend with ID number 6163

```
      UN    M   1500.3          // if request is not set, then
      BEB                       // end the function

      CALL  SFC  203
       LADDR  :=W#16#0          // VisuStage connection ID number.
                                // In Panel-PLC (e.g PC700P) must be W#16#0.
       TREND  :=6163            // ID number of trend, which is configured in VisuStage
       RET_VAL:=MW1000          // Result / return code
       BUSY   :=M1700.3         // TRUE = Request is in process

      U     M   1700.3          // If top the trend is in process, then
      BEB                       // end this function
      R     M   1500.3          // reset request signal

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

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

      BEA                       // end of function
```

Continue the trend sampling with SFC 204 "TRENDCNT"

The previously stopped trend sampling could be continued by calling SFC 204 “TRENDCNT“.

The sampling interval must be new assigned. The number of samples and sampling mode (cyclical overwriting, archiving) are kept as last assigned.

Parameter	Declaration	Data type	Description
LADDR	INPUT	WORD	Logical base address = 0 (for all Panel-PLCs) Logical base address = 1...16 (connection-ID-No. to the Panel-HMI)
TREND	INPUT	INT	Trend ID number, (from VisuStage configured @ resources)
INTERVAL	INPUT	WORD	Sampling interval in seconds. 1..65535 second are allowed.
RET_VAL	OUTPUT	WORD	Error code W#16#0000 – No error W#16#8002 – VisuStage is not running/configured W#16#8003 – Invalid „TREND“ ID number W#16#8004 – Trend was not started or Sampling count reached to zero. W#16#8xyy – General error codes (compatible to STEP®7) only for Panel-HMI: W#16#7000 – Job is in process W#16#7001 – Invalid connection state, e.g not connected W#16#7002 – Job not accepted, because another job is proceeded actually W#16#7003 – Connection resource occupied (temporarily locked) W#16#8001 – Invalid connection ID number or connection is not configured
BUSY	OUTPUT	BOOL	Busy flag, TRUE Job is in processing

→ Here you will find more for parameterizing of the objects Trend viewer and trend archive viewer

S7-program sample for using the SFC204

Purpose: Continue / resume the trend with ID number 6163

```
      UN    M   1500.4          // if request is not set, then
      BEB                       // end the function

      CALL  SFC  204
       LADDR   :=W#16#0         // VisuStage connection ID number.
                                // In Panel-PLC (e.g PC700P) must be W#16#0.
       TREND   :=6163           // ID number of trend, which is configured in VisuStage
       INTERVAL:=W#16#1         // every 1s samples the values
       RET_VAL :=MW1000         // Result / return code
       BUSY    :=M1700.4        // TRUE = Request is in process


      U     M   1700.4          // If Resume the trend is in process, then
      BEB                       // end this function
      R     M   1500.4          // reset request signal

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

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

      BEA                       // end of function
```

Query the trend status with SFC 205 "TRENDSTA"

To query the trend status use SFC205 “TRENDSTA” function. It reports about trend sampling and/or trend archiving status.

Parameter	Declaration	Data type	Description
LADDR	INPUT	WORD	Logical base address = 0 (for all Panel-PLCs) Logical base address = 1...16 (connection-ID-No. to the Panel-HMI)
TREND	INPUT	INT	Trend ID number, (from VisuStage configured @ resources)
RET_VAL	OUTPUT	WORD	Error code W#16#0000 – No error W#16#8002 – VisuStage is not running/configured W#16#8003 – Invalid „TREND“ ID number W#16#8xyy – General error codes (compatible to STEP®7) only for Panel-HMI: W#16#7000 – Job is in process W#16#7001 – Invalid connection state, e.g not connected W#16#7002 – Job not accepted, because another job is proceeded actually W#16#7003 – Connection resource occupied (temporarily locked) W#16#8001 – Invalid connection ID number or connection is not configured
BUSY	OUTPUT	BOOL	Busy flag, TRUE Job is in processing
TREND_STATUS	OUTPUT	BYTE	Trend status code B#16#00 – Trend sampling is not initiated B#16#01 – Trend sampling is initiated and running B#16#02 – Trend sampling has finished/stopped
ARCHIVE_STATUS	OUTPUT	BYTE	Archiving status code B#16#00 – Trend archiving is not initiated B#16#01 – Trend archiving is initiated and running B#16#02 – Trend archiving has finished successfully B#16#03 – SD card is not inserted or not recognized B#16#04 – Trend archiving stopped: Could not create file B#16#05 – Trend archiving stopped: Could not read file B#16#06 – Trend archiving stopped: Could not set file size B#16#07 – Trend archiving stopped: Could not write header to file B#16#08 – Trend archiving stopped: Error on file seek B#16#09 – Trend archiving stopped: Could not write samples to file

→ Here you will find more for parameterizing of the objects Trend viewer and trend archive viewer

S7-program sample for using the SFC205

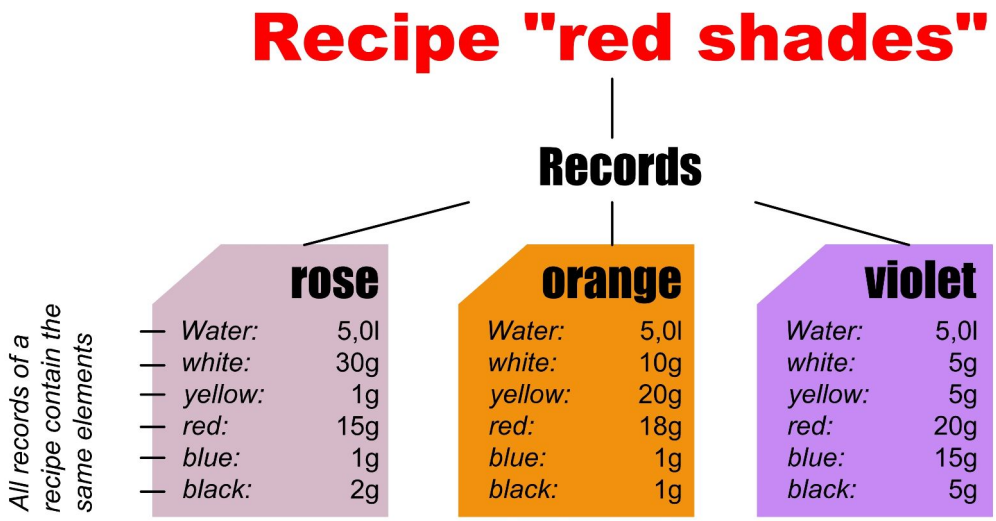
Purpose: Check the status of trend with ID number 6163

```
      UN    M   1500.5          // if request is not set, then
      BEB                       // end the function

      CALL  SFC  205
       LADDR          :=W#16#0  // VisuStage connection ID number.
                                // In Panel-PLC (e.g PC700P) must be W#16#0.
       TREND          :=6163    // ID number of DRUCK trend, which is configured in VisuStage
       RET_VAL        :=MW1000  // Result / return code
       BUSY           :=M1700.5 // TRUE = Request is in process
       TREND_STATUS   :=MB2008  // Trend status code
       ARCHIVE_STATUS:=MB2009   // Trend archiving status code

      U     M   1700.5          // If Start the trend is in process, then
      BEB                       // end this function
      R     M   1500.5          // reset request signal

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


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

      BEA                       // end of function
```

Resource recipe administration and recipe viewer

INSEVIS-Panels/HMIs and Panel-PLCs may administrate
- with CPU type V up to 64 recipes with each up to 128 records with each up to 128 elements and
- with CPU type P up to 64 recipes with each up to 256 records with each up to 256 elements.

Structure of a randomized sample for color mixing

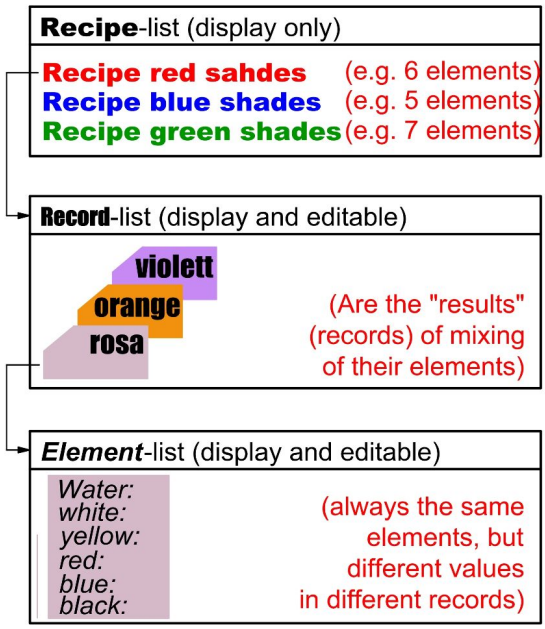


The different kinds of color shades (**red shade**, **blue shade**, **green shade**) are **recipes**.
Each recipe contains of always the same **elements** (e.g. **red shades** contains of *water, white, yellow, red, blue, black).*
Depending on the relation of the elements several **records** of the recipe **red shades** are generated (e.g. the colors **rose**, **orange**, **violet**).

The user selects a recipe from the **recipe list** first.
(in this sample the recipe **red shade**)
There can not be created new recipes at runtime, because new elements are needed for that. But for those elements some new machine parts must be integrated and wired (e.g. hatches, scales and so on...) first.

The record list shows existing records. A record can be added, selected for further operation, saved as or deleted here.
(in the sample te records (colors) **rose**, **orange**, **violet**)
If you create a recipe in the VisuStage, you assign a value for each element and so the first record is created automatically, buit not named and stored. Save it with a new name and so create your records.

Select a record to change its elements in the element list. The values of all elements (assigned to this recipe in the VisuStage) can be changed and stored here
(in the sample the elements *water, white, yellow, red, blue, black)*



INSEVIS-specific hints for recipe administration
1. INSEVIS- Panel-PLCs store the records in the Micro-SD-Card.
2. There is only one recipe viewer allowed and useful in a screen window.
3. If the visualization is changed and download again, the recipe structure wil overwritten by itself. But the record data on the Micro-SD-Card will be kept. So all stored records of the recipes can be kept while you update your visualization.
4. Because of the data storage on the Micro-SD-card (external memory) there no simulation of the recipe administration with SimuStage is possible.
5. The access administration is made by a button with own access rights, who changes the screen into the menu of the recipe viewer.
6. With SFC206 you can combine e.g. a bar code reader input with a special recipe record. The reference table (what bar code uses what recipe record) is to program in S7.
7. Export of recipe records is available with RemoteStage 1.0.3.7

Development of recipes in a randomized sample for color mixing

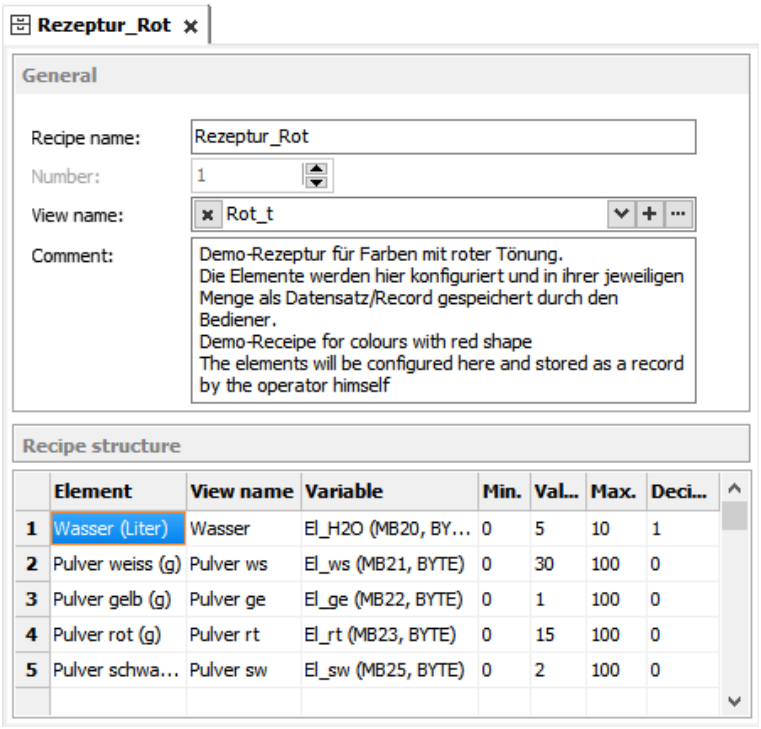
Choose „Add a new recipe“, parameterize it in a new window as written below:

General settings

Recipe name:	internal name of the recipe (independ of project language)
View name:	displayed name of the recipe (depends of project language)
Comment:	Insert your additional comments here
(Number)	automatic increased internal number of the recipe (not changeable)

Recipe structure

Element	internal name of the element (independ of project language)
View name:	displayed name of the element (depends of project language), Here the physical user unit can be displayed too (e.g. kg, sec., cm, etc.)
Variable	Assigning of a variable to this element (if you show user units, than scale this variable to fit with the displayed unit. So the operator can work very comfortable)
Min.	Input of the minimal value of the element (If the operator inserts a lower value, it will replaced by the the minimum value automatically)
Value	Input of the element value (thereby the elements of the first record will be pre-assigned and kept in the memory until the record is first time saved as...)
Max.	Input of the maximal value of the element (If the operator inserts a higher value, it will replaced by the the maximum value automatically)
Decimal point:	Insert the number of decimal places of the value



	Element	View name	Variable	Min.	Val...	Max.	Deci...
1	Wasser (Liter)	Wasser	El_H2O (MB20, BY...	0	5	10	1
2	Pulver weiss (g)	Pulver ws	El_ws (MB21, BYTE)	0	30	100	0
3	Pulver gelb (g)	Pulver ge	El_ge (MB22, BYTE)	0	1	100	0
4	Pulver rot (g)	Pulver rt	El_rt (MB23, BYTE)	0	15	100	0
5	Pulver schwa...	Pulver sw	El_sw (MB25, BYTE)	0	2	100	0

Hint:
The FAT16/32 file system of the Micro-SD-card does not accept some characters in the file- or folder names

These are: \ / : * ? " < > | This characters will be ignored.

Recipe handling with SFC 206 "RECIPE"

With the SFC206 “RECIPE” recipe records can be
- read in from Micro-SD-card and written to the PLC and
- written from the PLC to the Micro-SD-card.

It means, you can combine e.g. a bar code reader input with a special recipe record. The reference table (what bar code uses what recipe record) is to program in S7.

Parameter	Declaration	Data type	Description
REQ	INPUT	BOOL	TRUE assign and start job
LADDR	INPUT	WORD	Logical address 0 Panel-SPS (intern VisuStage RT) 1..16 HMI- Connection ID number
RECIPE	INPUT	INT	Recipe ID number (from VisuStage configured @ resources)
REC_NAME	INPUT	STRING	Recipe record name
FUNCTION	INPUT	BYTE	Function number 1 Read record from SD and write to PLC (SD → PLC) 2 Read record from PLC and write to SD (PLC → SD) 3 Read record from PLC and overwrite to SD (PLC → SD) 4 Delete record from SD 5 Check record existence in SD 6 Get number of records in SD (Output parameter „RETVAL“ returnes number of available records)
RETVAL	OUTPUT	WORD	Status code of job
BUSY	OUTPUT	BOOL	TRUE= Job is in process
DONE	OUTPUT	BOOL	TRUE= Job is done successful
ERROR	OUTPUT	BOOL	TRUE= Job is failed, error code in RETVAL

Status code meaning

RETVAL	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	1	Connection is in invalid state (e.g. not connected)
W#16#7002	0	0	1	Job is not accepted, due to previous job is still in process
W#16#7003	0	0	1	Connection resource is temporary locked
W#16#7004	0	0	0	REQ Parameter is FALSE
W#16#8001	0	0	1	Invalid connection ID number
W#16#8002	0	0	1	Visualization is not running (on Panel-HMI) or recipe not configured
W#16#8003	0	0	1	Invalid RECIPE ID number
W#16#8004	0	0	1	File system is not ready (eg. SD card is not inserted)
W#16#8005	0	0	1	Could not access to record file in SD card
W#16#8006	0	0	1	Invalid file content
W#16#8007	0	0	1	Record file exists in SD card
W#16#8008	0	0	1	Recipe path not found in SD card
W#16#842A	0	0	1	Invalid format in REC_NAME parameter
W#16#852B	0	0	1	Invalid function ID number in FUNCTION parameter
W#16#8xyy	0	0	1	General error code (S7 compatible)

S7-program sample for using the SFC206

Purpose: Create (overwrite if record file exists) a new recipe record file in micro SD card.

```
      CALL  SFC  206
       REQ     :=M1500.6              // TRUE = Request to create a new recipe record
       LADDR   :=W#16#0               // VisuStage connection ID number.
                                      // In Panel-PLC (e.g PC700P) must be W#16#0.
       RECIPE  :=1                    // Recipe number, in VisuStage configured
       REC_NAME:=DB205.RECORD_NAME    // Recipe record name
       FUNCT   :=B#16#3               // Write Record to SD card,
                                      // if record exists, overwrite it
       RET_VAL :=MW1000               // Result / return code
       BUSY    :=M1700.6              // TRUE = Request is in process
       DONE    :=M1002.0              // TRUE = Successful created
       ERROR   :=M1002.1              // TRUE = Error occurred

      U     M   1700.6                // If Recipe record is in process, then
      BEB                             // end this function
      R     M   1500.6                // reset request signal

      U     M   1002.0                // If successful done, then
      BEB                             // end the function

      UN    M   1002.1                // if no error occurred, then
      BEB                             // end the function

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

       BEA                            // end of function
```

Hint:
The FAT16/32 file system of the Micro-SD-card does not accept some characters in the file- or folder names

These are: \ / : * ? " < > | This characters will be ignored.

Hint to function 6
The input parameter „REC_NAME“ can carry wildcards as:

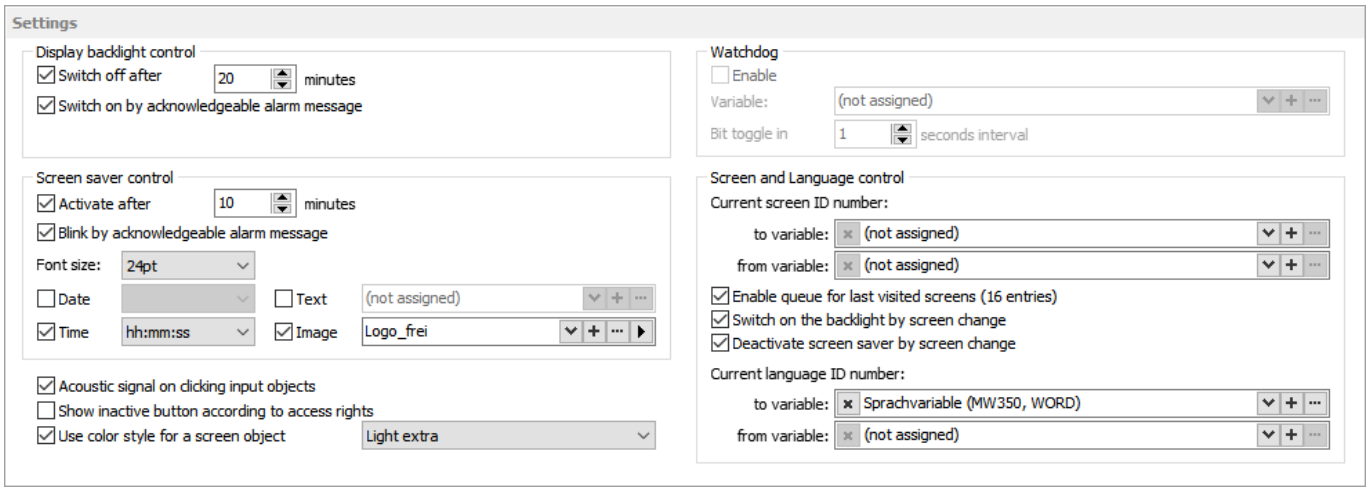
*	any ASCII character(s)
?	only one ASCII character
<ASCII character>	specific ASCII character

For sample:

- REC_NAME = “**ab***“ returns the number of all records, what start with **ab**
  (no matter what and how much character follow)
- REC_NAME = “**a?b**“ returns the number of all records with 3 characters
  (what start with **a** and end with **b** and contain on character in the middle (no matter what))
- **REC_NAME = “*“ or an empty S7-String return the number of all records**

Ressource Menues

The "Menus" resource shows the overview of the created menus (screens) and general settings about them as follows:



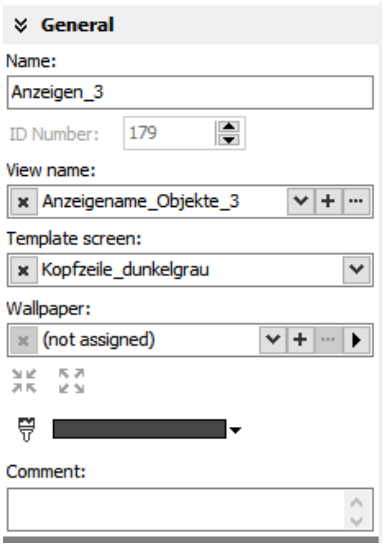
- **Display backlight** - Autonomous switch off / - Alarm message after switch on
- **Screen saver**
  - Autonomous switch on
  - To be switched off by a programmed screen change by S7-program
  - Blinking, when alarm message is active
  - Design of a display of date/time, image or text, what should be displayed during the screen saver timer period
- **Acoustic signals** by an integrated buzzer
- **Displaying inactive buttons** when no access rights exist
- **Using of color schemes** to get a modern design (or to switch it off)
- **Watchdog**
  to control the connections to the PLC with a toggeling (live-) bit (for Panels-HMI only)
- **Control** of
  screens and languages by variables (writing actual screen/language ID to variable or change screen/language by its ID from variable)
- **Activation** of a log of the last 16 screens visited before (for button function “Change Screen” → <Previous screen>)

Hint:
Make the dimension of the screen saver ***much smaller*** that the whole resolution is to let the screen saver „walk“.

Visualization with the software „VisuStage“

Basic functions

There can be selected any objects by left mouse click and be copied/ cut out/ grouped/ ungrouped by right mouse key. Copied objects will be pasted in the same screen, they will get an X-Y-offset of 5 pixels to keep orientation. When haviing marked multiple different object, common properties can be edited at once.



Background image

Hit an empty space (with no objects) in your screen to get to the general settings.

The name is the name of the screen (to be seen in the screen-tree on the left side)
The ID-number of the screen will be used by the PLC, when a screen change is to be ordered by the PLC (of course not available in the HMIs). This ID is not to be changed manually.

By the view name a a multi-languaged text resource will be used from a text field from the screen template with the system variable $(CurrentScreenViewName) to display the present screen name in multiple languages .

A template screen can be selected and assigned.

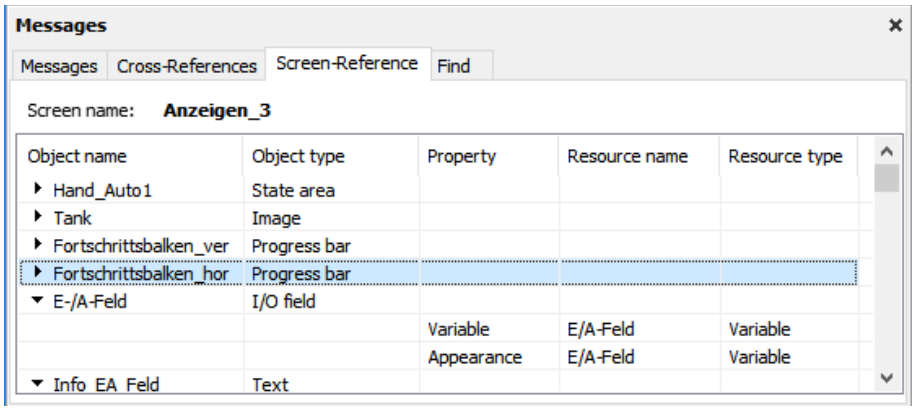
A background color or a background image can be selected. (Image needs more memory than a simple color.)

In this configuration no wallpaper but a gray background color is defined.

Display of used resources in a screen

The function „Screen-reference“ displays all resources used in the marked screen.

- Mark a screen in the left screen tree by left mouse key
- Right mouse key opens up a pull-down menu, select „Screen-reference“
- In the message window all resources of this screen will appear

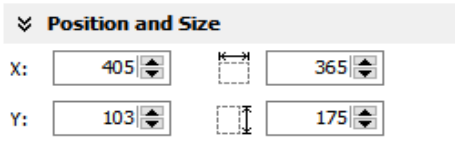


Clean Screen

The Clean screen will be started with a function (to be chosen by the rider „Function“) and there also the time period will be assigned.

Placing an object

Select the required objects at the catalog tree (right) and add them to the screen in a window by left mouse key.
Make the fine placement by the box „Position and Size“ in pixels or by key-combination SHIFT+arrow-keys (8 pixels each click)
Note that the zero-point (X;Y = 0;0) is the **TOP LEFT CORNER**

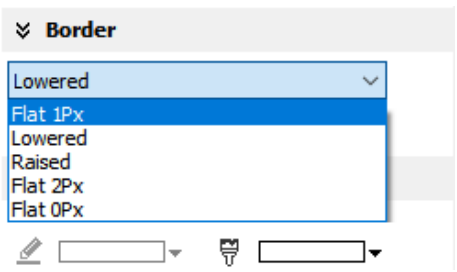


Text font

Here you can assign the font and size as well as the color and the style (Italian, Bride) of text

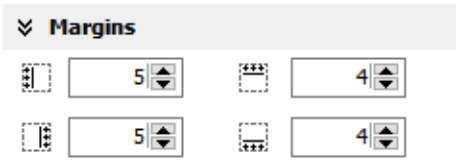
Border

Assign here the outline and mostly the rear color of your object
- flat = assign frame and background color without 3D-effect
- lowered = assign frame and background color with 3D-effect “low”
- raised = assign frame and background color with 3D-effect “high”
- none = transparent



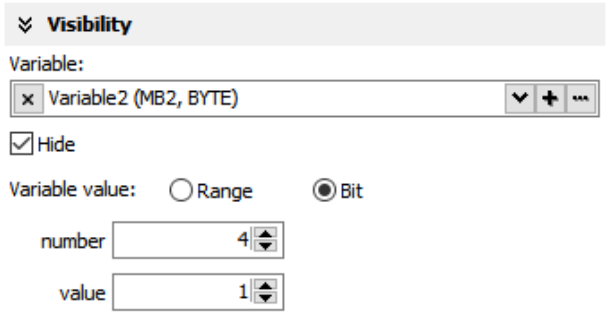
Margins

Assign the distance between texts/labels or symbols on a button to the frame of this object (in pixels).

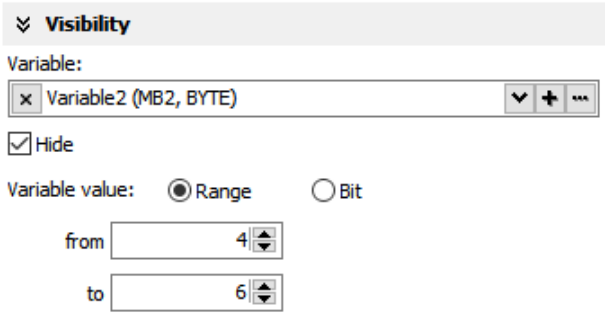


Visibility

The visibility of objects can be controlled by a variable and is to be assigned easily



In this sample an object is hidden,
when the 4th bit (counting from right)
will have the binary value 1,
in all other cases it is visible

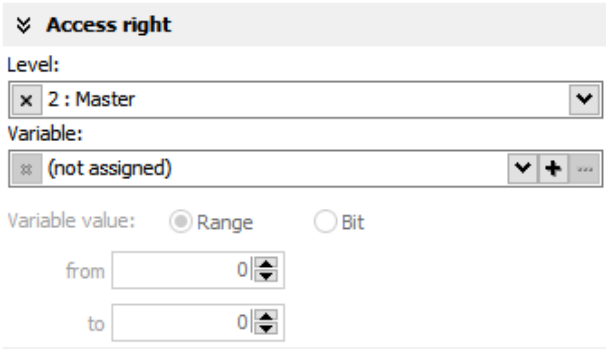


In this sample an object is hidden,
when the variables value is between 4 and 6,
in all other cases it is visible

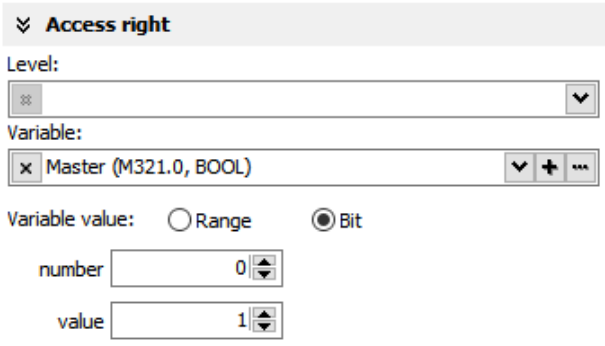
Hint: Depending on type of variables, some inputs can stay empty (e.g. the masking, if a boolean variable was selected).

Access rights

These rights are to be assigned by user level and by a variable.



In this sample an object is accessible,
for every user below the level 2 (Master)



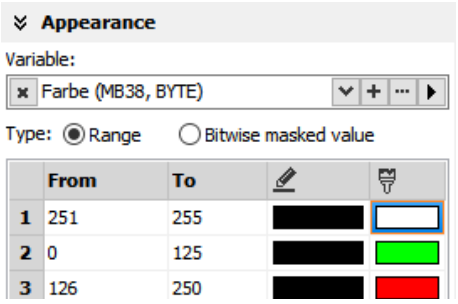
In this sample an object is accessible,
when the variables value is 1

Group / Ungroup objects

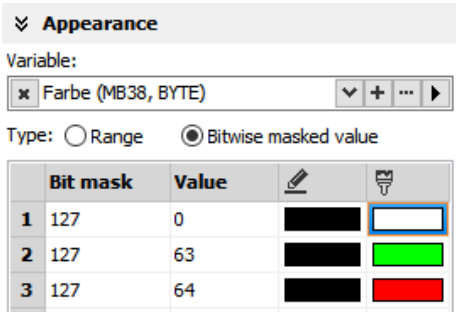
Group objects is necessary to fix multiple objects in an preassigned position by the right mouse key. It makes sense e.g. to create library elements. The group can be split by right mouse key again.

Appearance

The appearance of most of the objects can be controlled based by value of a variable.



In the area (0...125) the object will displayed in green and in the area (126...250) in red. The frame will be displayed in black.



Here the object becomes green only if in the lowest 7Bit (refers to bit mask 127) the value of 63 is reached and red If the value of 64 is reached. The frame is shown in black.

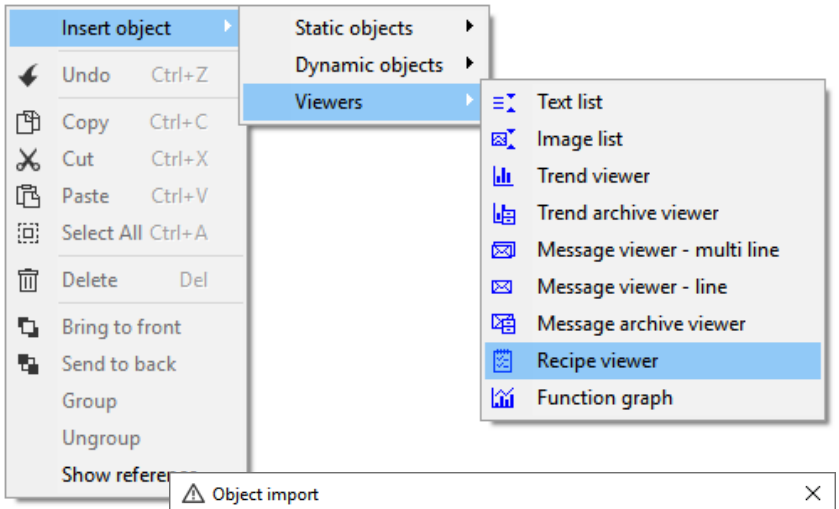
Out of the selected area or values the color of the first line is displayed always. In above samples it is withe with black frame. Here it is recommended to create an additional line for that purpose.

ATTENTION:
You need to define another color valid for values out of desired area or values in the 1st line.
If you do not add this line, the color of the first line will be displayed for values out of desired area or values.
This can cause wrong color information, what can cause an incorrect operation by operator!

Insert objects

Objects can be inserted by

- selecting in the right catalog window and clicking with left mouse key in the menue window
- a key combination without mouse in the menue window (to see right beside the object in the catalog-window)
- by right mouse key in the menue window by a popup window (see image right)



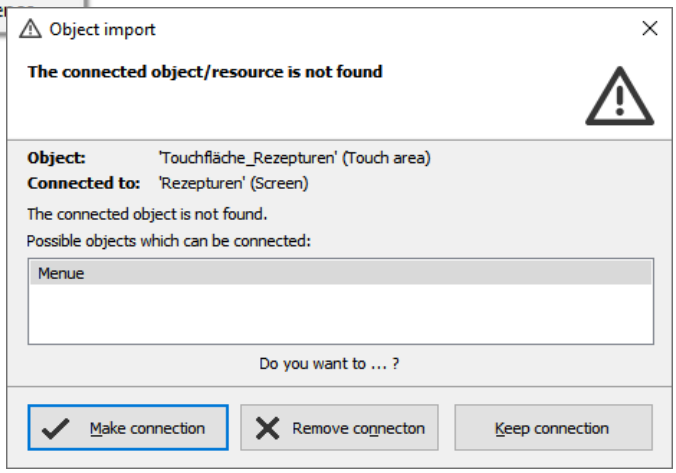
Copy objects

Instead of systematic saving groups of objects into libraries (recommended!) there is a “cheap´n dirty” version too by simple copying of selected object from one VisuStage into another open VisuStage.

- Mark the requested objects
- Move it by drag´n drop into the other VisuStage

Variables will be copied too and

- if identical to existing ones the original variables will be used
- if symbols (name) and or absolut address differ to original ones, a ne variable will be crated



If there are variables, what are linked to any objects (like screen changes) a message box appears and informs about every single variable and wants a decision how to act.
The PLC can request what screen menu is active ad can force the VisuStage to change into another screen.

Query the current the screen ID number with SFC 200 "SCR_GET"

The currently shown screen ID number can be retrieved by using SFC 200 “SCR_GET“

Parameter	Declaration	Data type	Description
LADDR	INPUT	WORD	Logical base address = 0 (for all Panel-PLCs) Logical base address = 1...16 (connection-ID-No. to the Panel-HMI)
RET_VAL	OUTPUT	WORD	Error code W#16#0000 – No error W#16#8002 – VisuStage is not running/configured W#16#8xyy – General error codes (compatible to STEP®7) only for Panel-HMI: W#16#7000 – Job is in process W#16#7001 – Invalid connection state, e.g not connected W#16#7002 – Job not accepted, because another job is proceeded actually W#16#7003 – Connection resource occupied (temporarily locked) W#16#8001 – Invalid connection ID number or connection is not configured
BUSY	OUTPUT	BOOL	Busy flag, TRUE Job is in processing
SCREEN	OUTPUT	INT	Currently shown screen ID number, (from VisuStage configured @ resources)
LANG	OUTPUT	INT	Currently selected language ID number (referring to int´l. codes - you can find it in the csv-file, when using the text export)

S7-program sample for using the SFC200

Purpose: Query the actual screen

```
UN    M   1500.0                // if read request is not set, then
BEB                             // end the function

CALL  SFC  200
 LADDR  :=W#16#0                // VisuStage connection ID number.
                                // In Panel-PLC (e.g PC700P) must be W#16#0.
 RET_VAL:=MW1000                // Result / return code
 BUSY   :=M1700.0               // TRUE = Reading is in process
 SCREEN :=MW2002                // Current screen ID number
 LANG   :=MW2004                // Current language ID number

U     M   1700.0                // Reading is in process
BEB
R     M   1500.0                // reset request signal

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

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

BEA                             // end of function
```

Changing the screen with SFC 201 "SCR_SET"

Screen can be changed by using SFC 201 “SCR_SET“.

Parameter	Declaration	Data type	Description
LADDR	INPUT	WORD	Logical base address = 0 (for all Panel-PLCs) Logical base address = 1...16 (connection-ID-No. to the Panel-HMI)
SCREEN	INPUT	INT	Screen ID number, (from VisuStage configured @ resources)
RET_VAL	OUTPUT	WORD	Error code W#16#0000 – No error W#16#8002 – VisuStage is not running/configured W#16#8003 – Invalid Screen ID number W#16#8xyy – General error codes (compatible to STEP®7) only for Panel-HMI: W#16#7000 – Job is in process W#16#7001 – Invalid connection state, e.g not connected W#16#7002 – Job not accepted, because another job is proceeded actually W#16#7003 – Connection resource occupied (temporarily locked) W#16#8001 – Invalid connection ID number or connection is not configured
BUSY	OUTPUT	BOOL	Busy flag, TRUE Job is in processing

S7-program sample for using the SFC201

Purpose: Change to screen with ID number 16917

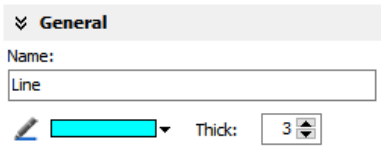
```
      UN    M   1500.1                  // if request is not set, then
      BEB                               // end the function

      CALL  SFC  201
       LADDR  :=W#16#0                  // VisuStage connection ID number.
                                        // In Panel-PLC (e.g PC700P) must be W#16#0.
       SCREEN :=16917                   // Screen ID number to change
       RET_VAL:=MW1000                  // Return / result code
       BUSY   :=M1700.1                 // TRUE = Request is in process

      U     M   1700.1                  // request is in process
      BEB
      R     M   1500.1                  // reset request signal

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

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

      BEA                               // end of function
```

Creating a visualization screen

Static objects

Lines, rectangles, texts, images

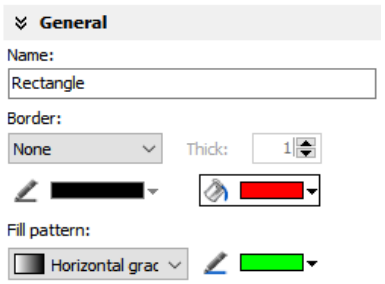
Lines



Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign appearance: (see basic funtions)
Parameterize the object: (register “General”)

- this sample shows a 3px wide line.
- When changing target screens a miss calculation of width can happen what causes overlapping failure (reported in message window)
- As of version 2.2.10, new line types and arrows can be used.

Rectangles

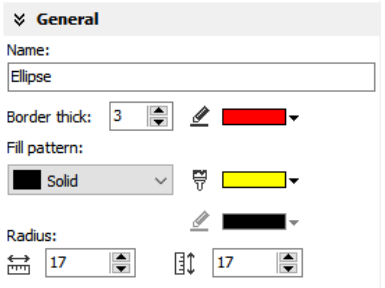


Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign appearance: (see basic funtions)
Parameterize the object: (register “General”)

- this sample shows a rectangle without border, with a horizontal color gradient from red to green, all other properties are self explaining

From VisuStage version 2.2.1.0 it is possible to round the corners with a radius.
In the settings, under the item Corners, a radius can be configured for all four corners.

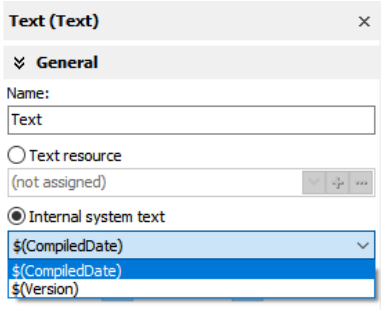
Ellipses



Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign appearance: (see basic funtions)
Parameterize the object: (register “General”)

- this sample shows 17px diameter yellow circle with a 3px red outline
- all other properties are self explaining

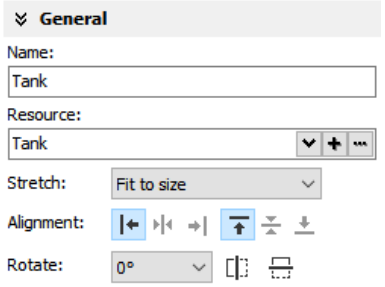
Texts



Place an object: (see basic funtions)
Assign visibility, fonts: (see basic funtions)
Assign appearance: **(see basic funtions)**
Parameterize the object: (register “General”)

- Use an existing **text resource** or create a new one directly here **OR** use the **Internal system text** to display compile-no and -date
- Assign font, bride and italic style
- Assign horizontal/vertical alignment
- Check “Wrap words” if this is a multiline text

Images



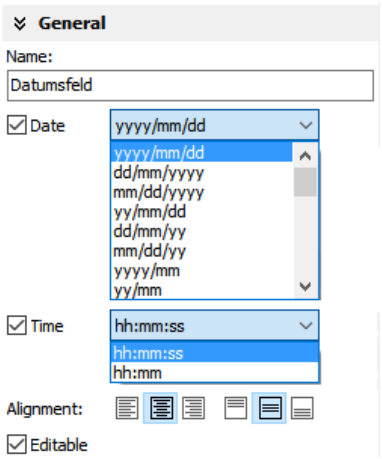
Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign appearance: (see basic funtions)
Parameterize the object: (register “General”)

- use an existing **resource** or create a new one directly here
- in the **„Stretch“**-box choose **„Fit to size“** to see your image in the objects rectangle you did draw
- assign horizontal/vertical **orientation** if object is not stretched
- **rotate** and **mirror** object as you want to

Dynamic objects

Date/Time fields

Date/Time fields are used to **display** (and, when „Editable“ is checked) to **edit** time and date by an automatic displayed virtual keyboard to type in the values.



Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign margins: (see basic funtions)
Assign border: (see basic funtions)
Assign font: (see basic funtions)
Assign access rights: (see basic funtions)

Parameterize the object: (register “General”)
- Assign the **format of Date/Time field:**
  - Time format
  - Date format
- Assign horizontal and vertical alignment
- By checking **„Editable“** the object´s value is changeable by the operator on the touch panel

Hint:
- If a DateTime field is to narrow, than characters, what can not be displayed completely will not displayed at all
- At **TOD** (TimeOfDay) the time will be displayed in hh:mm:ss format.
  Use TOD for longer times.
- At **TIME**, **S5TIME** and **TIMER** time is displayed in ms Milliseconds).
- At **DATE** the date will be displayed in days (since 01.01.1990).

Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „ ► “ creates an error report, click on error and come directly to the failed object.

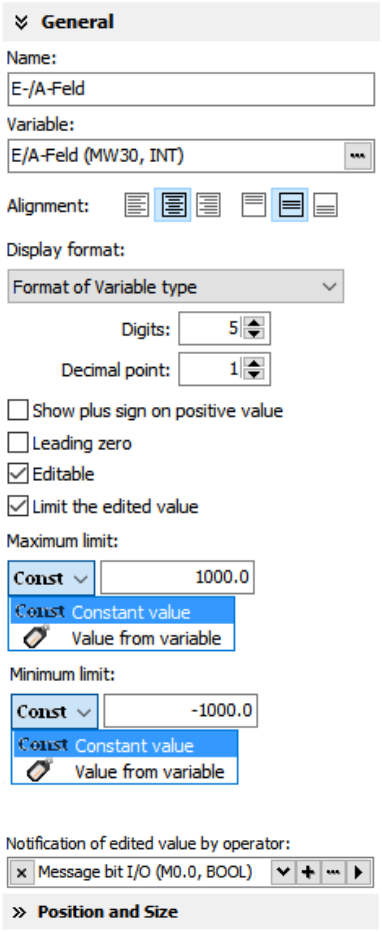
I/O fields

I/O fields are used to **display** (and, when „Editable“ is checked) to **edit** values with an automatic displayed virtual keyboard to type in the values.

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign margins: (see basic funtions)
Assign border: (see basic funtions)
Assign font: (see basic funtions)
Assign access rights: (see basic funtions)
Assign appeareance: (see basic funtions)

Parameterize the object: (register “General”)

- Assign the vertical and horizontal **alignment**
- Assign a (or create a new one) **variable** to be displayed in this field
- Assign the **format of I/O-field**
  - leading zero, „+“-sign, digits, decimal point
  - display format (if comma-key shall be used in keyboard dialog)
  - or use the format of the assigned variable (shown here)
- By checking **„Editable“** the object´s value is changeable by the operator on the touch panel
- **Limit** the max/min values (always depending on the variables type)



Notification of modification
Assign a bit to get informed, when the operator changes the value of the I/O-field at runtime.

Hint:
- At **STRNG** notice the S7-definitions (length-values in the first 2 bytes)!
  At INSEVIS the maximal length of the STRNG is 60 bytes
  BUT the referring DB MUST BE at least 60+2 bytes large!
- **Scaling:** by S7-program with the Siemens-FC105 SCALE
  (Result: REAL-value, what is not allowed in VisuStage to color the appearance of this fields),
- If this coloring/ appearance is needed, than us S7-commands Rounding (RND) or Cutting (TRUNC) to reach a valid value for this function
- If a DateTime field is to narrow, than characters, what can not be displayed completely will not displayed at all

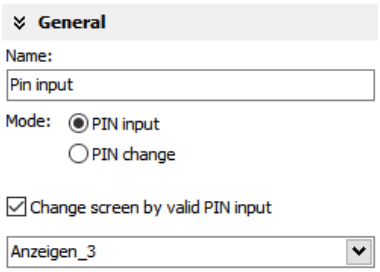
Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

PIN input fields

In the user administration can be assigned up to 9 levels with an up to 8-digit PIN- number to administrate the access rights.

A login-keyboard appears automatically after touching the PIN input field by the operator.
„INPUT“ is to enter your PIN to get into your access rights.
„CHANGE“ means, that PIN can be changed by authorized personally. After changing the new PIN is valid.

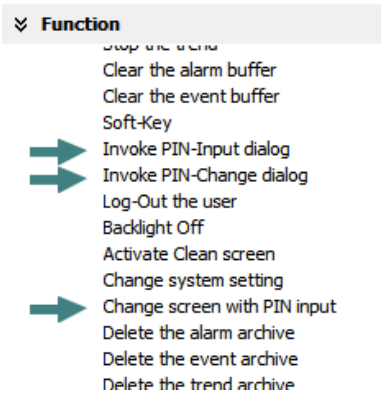
There are 3 ways to call this dialog:



1.) PIN-INPUT / CHANGE field (object)
The benefit of this way is, that the target screens, whereto it will be changed after successfully login/ logout are defined in the resource „User“

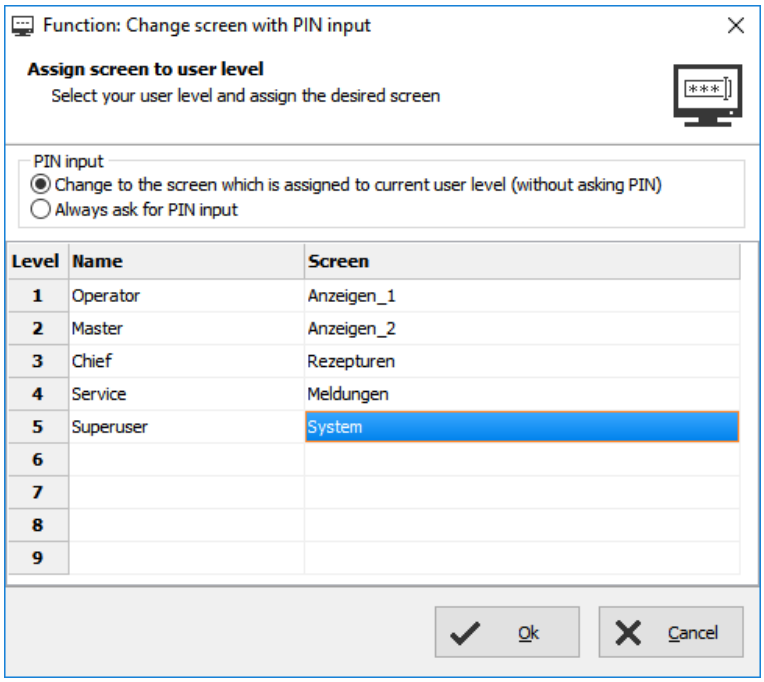
And here can be assigned a special screen change too.

2) Invoke PIN-INPUT / CHANGE dialog (function)
The benefit of this way is, that the login dialog can be assigned to any button or touch field as well. (See first 2 functions right)
The disadvantage is, that the screen is not automatically changeable with this function.



2) Change screen with PIN-input (function)
This is the right way to assign a different screen to change for every single user level.
(See 3rd function right)

With assigning this function you need to design the target screens for every user level.



Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

Diverse kinds of buttons

VisuStage contains standard, dynamic and text buttons.

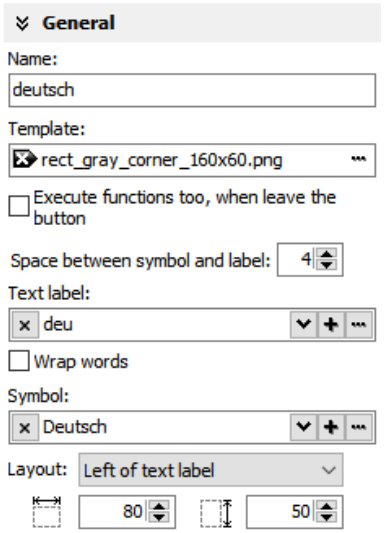
- **Standard buttons** are available as templates and contain 65000 colors.
- **Dynamic buttons** will be created in the VisuStage, contain 256 colors only but offer symbol change (by image list), multi lingual text change (by text list) and color change functions.
- **Text buttons** can proceed text changes with very large text lists

All buttons contain the dead-man´s function (executing the function too, when leave the button sidewards).

Attention
If you execute a function wen PRESSING and LEAVING the button, it is hardly recommended to check this box! Otherwise it may happen, that the operator leaves the button sideways without executing the assigned function for leaving the button and e.g. an drive does go on instead of stopping.

Standard buttons

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign margins: (see basic funtions)
Assign font: (see basic funtions)
Assign access rights: (see basic funtions)



Parameterize the object: (register “General”)

- Check / uncheck the execution when leaving the button sideways
- Select a **Label (Text)** from your resources or import it directly here to place it on the button
- Select a **Symbol** from your resources or import it directly here to place it on the button
- Assign the place, where the image/symbol is displayed referring to the text
- Assign the size of the symbols image

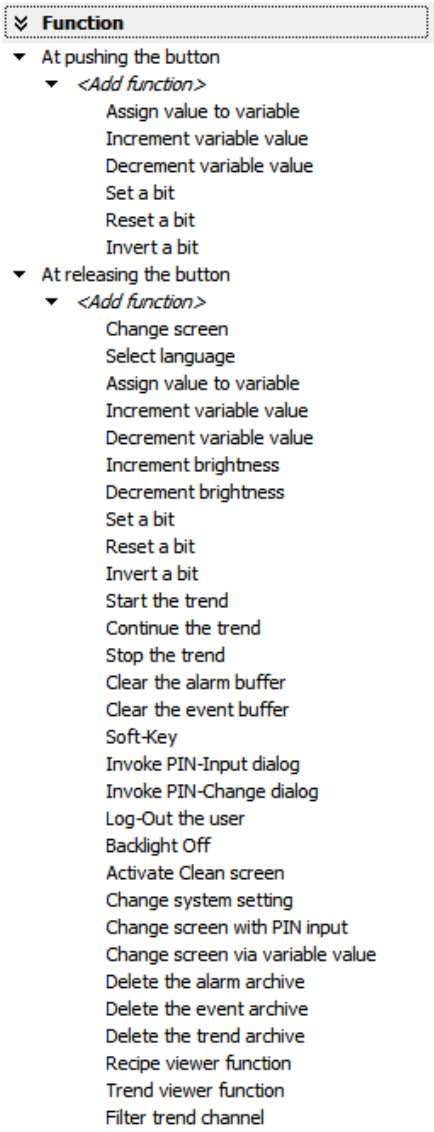
Use of selfmade buttons

- Copy your 3 buttons (all with the same name) each in one of those 3 directories
  -\inactiv (displayed when no access is allowed),
  -\pressed (displayed wen pressed)
  -\unpressed (displayed when access is allowed, but unpressed),
  of the main directory C:\Programs (x86)\INSEVIS\ VisuStage\Buttons.
- Choose the same color for the are in the middle, where your text and/or symbol will appear.

Attention
If you forward the source code of this VisuStage project (vsproj + vsbin) you need to add the 3 self made buttons and copy it in the same path like VisuStage Is installed there. Otherwise these buttons will not be displayed (and compiled to the binary).

Assign a function (register “Function”)

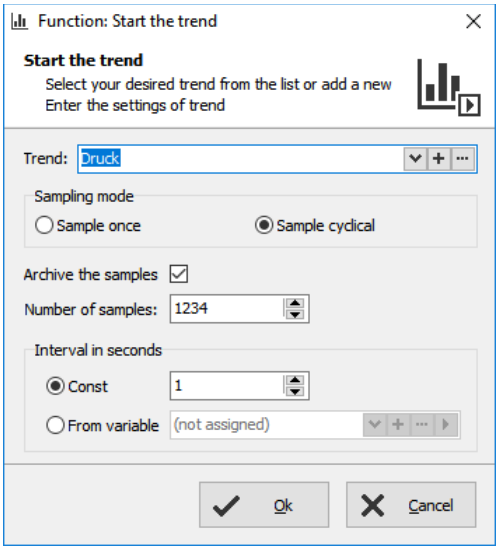
It is possible to assign multiple functions to one action (press or release).



- Decide, WHEN the function will be proceeded:
  - at PUSHING or
  - at RELEASING

  the button

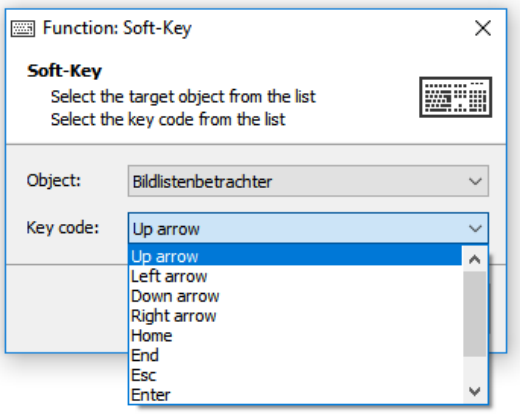
- After choosing a function (by double click) opens up another window to enter the required data for the selected function (like shown in sample below at “start Trend” function:)



- There can be assigned multiple functions to one object as well
- There can be assigned functions for pushing and releasing the button as well

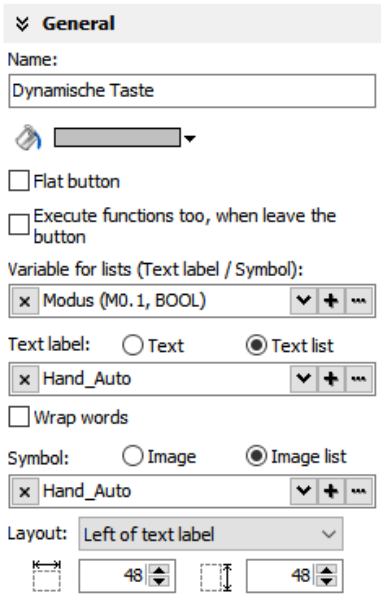
Use of Soft-Key function

- It is used to create own navigation buttons for viewer object and works in the same screen as the viewer object is.
- If there are more than one object with soft-key opportunities you have to select it first in the upper line.



Dynamic buttons

This buttons were used mostly for a mixture as state area, text- and image lists with a function to execute.)e.g. is in run / switch to stop, etc.). For this kind of button it is not possible to create own ones, because they were calculated at runtime.



Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign margins: (see basic funtions)
Assign font: (see basic funtions)
Assign access rights: (see basic funtions)
Assign appeareance: (see basic funtions)

Parameterize the object: (register “General”)

- **Assign a variable**, to control symbol- and text changes
- **Text changes:** (in the sample description / state of the buttons) choose (or create a new) a **text list**, what shell be shown on the dynamic button
- **Symbol changes:** (in the sample manually/automatic mode sign) choose (or create a new) an **image list**, what shell be shown on the dynamic button

Attention
If there is active another value as these what are assigned in line 1...n, than ALWAYS this appearance will be displayed, what is assigned FIRST (in first line / line 1).

Use of self-made buttons
- Dynamic buttons are to be created in the VisuStage software.
- There is not possible to add additional (own) button designs.

Text buttons

These objects are used like a dynamic button without symbol change, but can be used to proceed very large text lists. For this kind of button it is not possible to create own ones, because they were calculated at runtime.

Touch areas – invisible buttons

These objects will be placed above an other object (mostly an image) to add a special function by pressing it (like an invisible button over a special part of a machine).

Place an object: (see basic funtions)
Assign access right: (see basic funtions)
Assign appeareance: (see basic funtions)
Assign a function: (register “Function”)
like at the standard simple buttons.

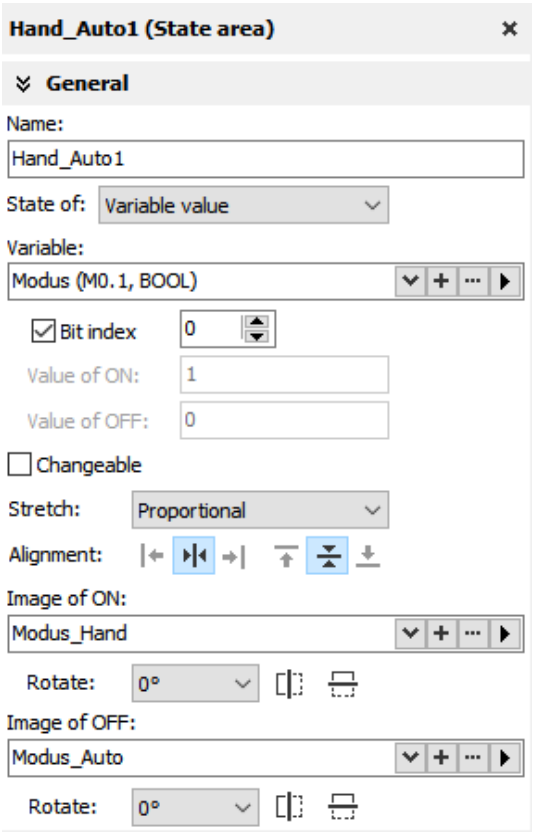
Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „►“ creates an error report, click on error and come directly to the failed object.

State areas

Place an object: (see basic funtions)
Assign access rights: (see basic funtions)
Assign visibility: (see basic funtions)

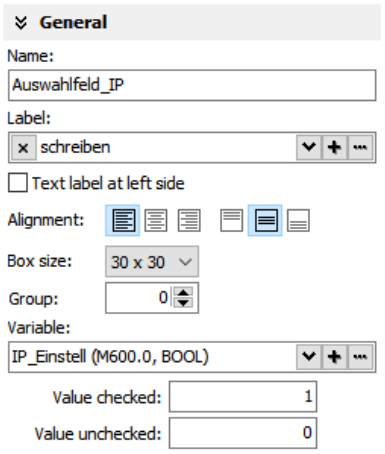
Parameterize the object: (register “General”)

- If you depend the state area on variables
  → use an existing resource or create a new one directly here
- If you use a system setting
  → select one of the system settings of the drop down menue
- In the „Stretch“-box choose
  - „Fit to size“ to see your image in the objects rectangle you did draw
  - “Proportional” to keep the images proportion (rest of the object stays empty)
- Assign an image resource or create a new one to be shown in ON-state and in OFF-state



Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- use PNG graphic format with a transparent Alpha-channel to let the background shine through the transparent areas of the image (e.g. on rounded corners).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

Radio boxes and Check boxes

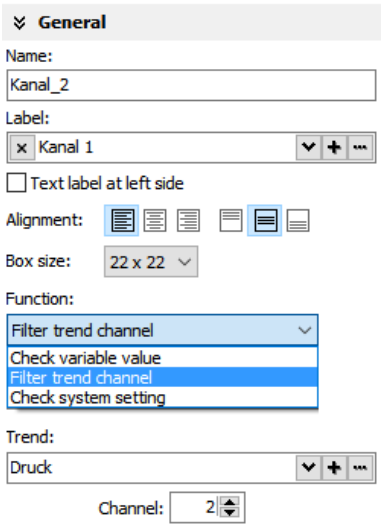


Radio boxes are used to visualize an **either-or-selection**
(e.g. at alarms coming and leaving)

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign margins: (see basic funtions)
Assign font: (see basic funtions)
Assign access rights: (see basic funtions)

Parameterize the object: (register “General”)

- Assign (or create a new one) a text („Label“) to describe the check item in the box
- Assign the alignment of the text
- Assign the dimension of the box size to click into
- Assign the group information (to what other radio boxes it should belong)
- Assign a variable value for UNCHECKED and one for CHECKED state



Check boxes can be checked and unchecked individually and **multiple** of it
(e.g. to filter trend channel or to change a state area).
Also these boxes are used to assign buzzer system functions.

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign margins: (see basic funtions)
Assign font: (see basic funtions)
Assign access rights: (see basic funtions)

Parameterize the object: (register “General”)

- Check,. If the text shall be left of the check box
- Assign (or create a new one) a text („Label“) to describe the check item in the box
- Assign the alignment of the text to the objects border
- Assign the dimension of the box size to click into
- Assign a variable, where the value is set by activating CHECKED / UNCHECKED (e.g. a bit with 0 / 1)
- Assign (or create a new one) a filter for a trend channel to show or
- Assign the state of a system property like e.g. the internal buzzer

Attention / objects in general:

- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- transparent areas of the image (e.g. on rounded corners).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

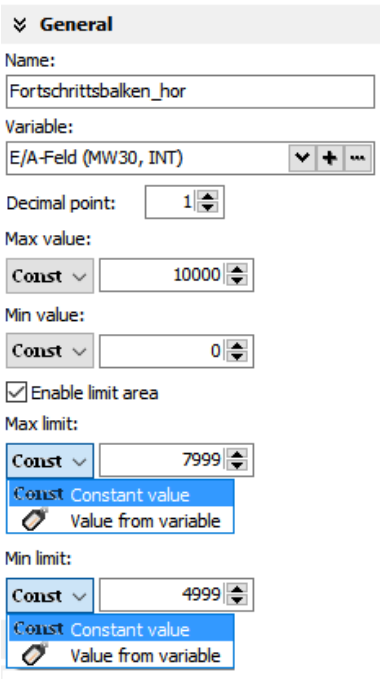
Progress bars

This item is uses to display analog values like dimensions, time, etc., where different colors depend on different limits assigned before.
The progress bar right was designed with these settings below:

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign margins: (see basic funtions)
Assign font: (see basic funtions)
Assign access rights: (see basic funtions)

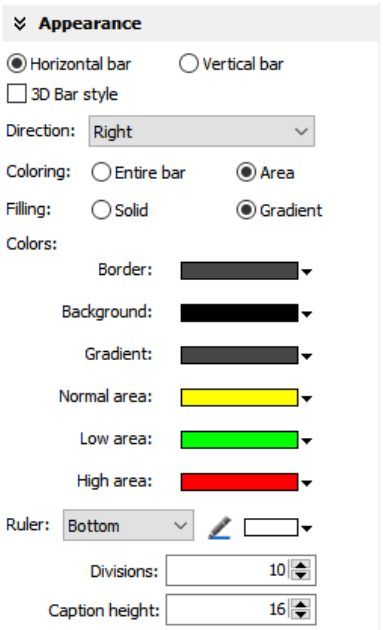
Parameterize the object: (register “General”)

- Assign a **variable** (or create a new one)
- Assign the **format** (minimal value, maximal value, decimal point)
  At INTEGER values type in the value w/o decimal point (here: 10000 = 1000.0)
- Assign a variable or a fixed value for Maxlimit and for Minlimit of the progress bar.
- If you want to color 3 parts of the progress bar, enable limit areas
  - by a fixed value or
  . By a variable



Color the object: (register „Appearance“)

- Choose a **bar style** (horizontal, vertical, 2D, 3D, color of background and fill color
- Decide, if the bar is always having ONE actual color (“Solid”)
  or 3 segments with a single color each (“Area”)
- If “Area” was selected: Assign direction of the colors in the bar
- Assign colors and of the 3 segments defined by the limits in “General”
- Design the **ruler** (position, color, divisions, capture height)
  → at unfavourable divisions rounding errors are possible!



Hint:
During first time dropping a progress bar into the screen the progress direction will be assigned referring to heigth / width as shown below:

- Is height > width = vertical direction
- Is height < width= horizontal direction

Attention / objects in general:

- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

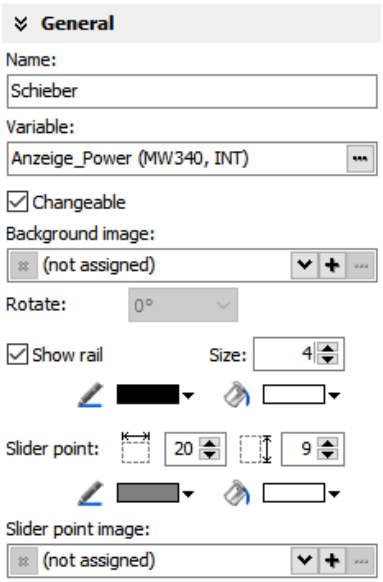
Sliders

This object is a good solution to display a variables value along a staight line or (if editable is activated) to set up a value roughly by hand.

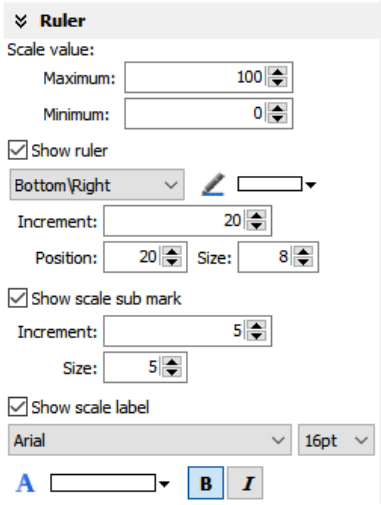
Place an object: (see basic funtions)
- Is height > width = vertical direction
- Is height < width= horizontal direction

Assign visibility: (see basic funtions)
Assign access rights: (see basic funtions)

Parameterize the object: (register “General”)
- Assign a **Variable** vor the value you want to display/edit
- If “**Changeable**” is activated the slider point is movable and changes the variables value
- It is possible to assign the sliding rail by VisuStage or to import an existing image as **Background image**
- It is possible to assign the sliding point by VisuStage or to import an existing image as **Slider point image**



Parameterize the scale: (register “Ruler”)
- Insert fixed values for the **Minimum / Maximum** of the rulers range
- Activate the display of the **Ruler**
- Select the rulers position referring the slider
- Assign the increments (values between the visible values on the ruler)
- **Position** = The rulers distance between the ruler and the sliding rail.
- **Size** = Width (at vertical) e.g. height (at horizontal) of the ruler
- Activate **Scale sub marks** (between the main marks = rulers increments)
  (it is better if these sub marks are a devisor of the main mark, e.g. 5 or 10, when the rulers increment is 20)
- Activate **Scale label** (self explaining)



Hint:
By the dimension of the objects when putting into the menue by mouse it will be set automatically, if it is a
- vertical (height > width) or an
- horizontal (height > width) slider.

Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „►“ creates an error report, click on error and come directly to the failed object.

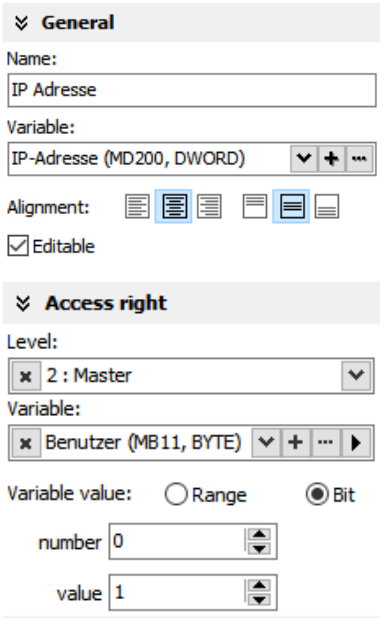
System data (IP-address field, languages, etc.)

While languages, brightness or screen savers are functions what can be assigned to buttons or touch areas, the IP-address-field is an own object.

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign margins: (see basic funtions)
Assign font: (see basic funtions)
Assign access rights: (see basic funtions)

Parameterize the object: (register “General”)

- Assign a (or create a new one) **variable** to control the system property
- By checking **„Editable“** the object´s value is changeable by the operator on the touch panel
- Assign the alignment of the box
- Assign access rights



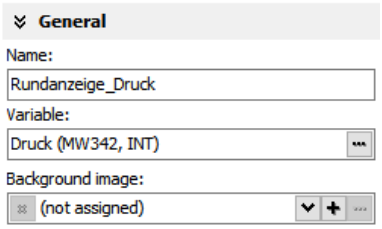
Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

Gauges

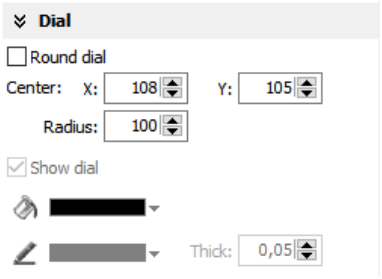
With this object a value can displayed in a gauge form with a pointer like in older meters.

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)

Parameterize the object: (register “General”)
- Assign a variable to display their value
- Activate a background image
  (If you do not want do create your gauge by VisuStage)



Parameterize the dial: (register “Dial”)
When round dial is activated, a circle is created by the smaller value ox X and Y
- When “Show dial” is activated this can be colored
  - Assign the background color of the dial
  - Assign the border color and thickness of the dial
- When “Round dial” is dectivated, you can set:
  - **Radius** of the circle segment (angle below „Scale“)
  - **X-Position** of the pointer rotation points (from the left object border)
  - **Y-Position** of the pointer rotation points (from the upper object border)

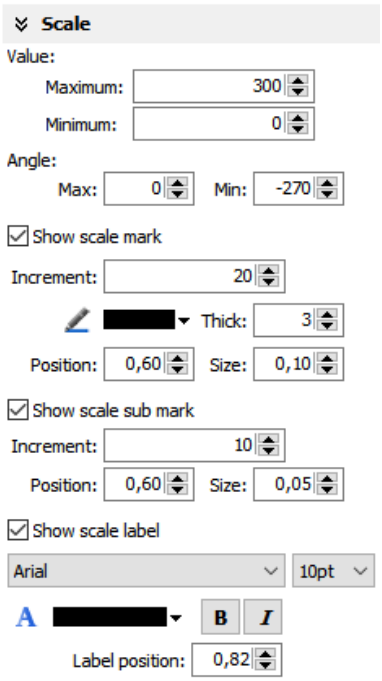


Scale configuration: (register “Scale”)
- Insert **Minimum / Maximum** of the scale value to be dislayed

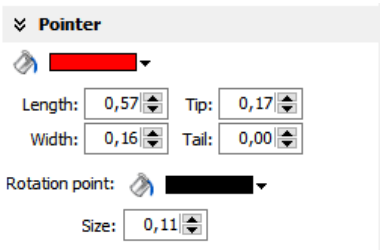
- Insert the **Angle** of the circle segment for the scale area
  - Start at **Min**, end at **Max,**
  - **Pointer always runs clockwise**
  - Information angle position: -270° = „6 o´clock“ / -180° = „9 o´clock“
    -90° = „12 o´clock“ / 0° = „3 o´clock“ / 90° = „6 o´clock“
    180° = „9 o´clock“ / 270° = „12 o´clock“

- Activate **Scale marks** (marks and sub marks)
  - Increments between main marks
  - Color, thickness, position (~Radius form pointer rotation point), size (~length)
- Activate **Scal sub marks**
  - Increments between sub marks
  - Color, thickness, position (~Radius of the pointer rotation point), size (~length)

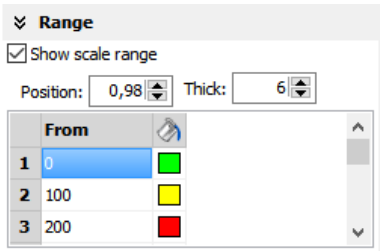
- Activate **Scale label**
  - Font, size, color, style (bride/italian)
  - Position (~ assign radius of the pointer rotation point



Pointer configuration: (register “Pointer”)
- Assign a **color** for the pointer
- Assign the **length** of the pointer (distance between pointer root to top))
- Assign the **width** of the pointer
- Design the shape of the **pointers tip** (0,01~ flat, >0,01 more and more narrow)
- Assign the distance between pointer root to the rotation point **(tail**)
- Color the **circle at the pointers rotation point**
  (The rotation circle always does have a 1 pixel-frame outside in the pointers color))
- Adjust the **size (diameter)** of the rotations points circle



Configure the color ring: (register „Range“)
Limits and areas can be colored by a ring inside/outside the scale.



- Activate **Scale range**
  - Assign the **position** (~ radius of this circle segment)
  - Assign the **thickness** of this ring
- **Coloring**
  - The color area do not have an upper limit.
  - The next color are starts at the lower limit of the next color.
  - the last color limit is the maximum limit of the scale

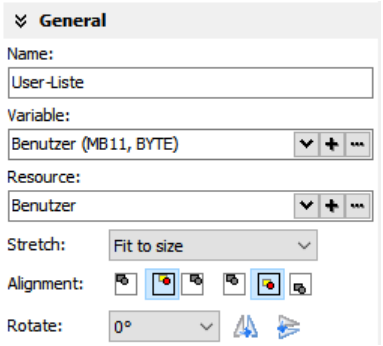
Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

Viewers

Image and text lists

Image list

Look at the color change of the pipes in visualization sample, screen “objects” 3. The pipes will be separated into segments. For these segments all colors are available as single images with the requested colors. In the resource „image list“ values will be assigned to these single images. In the object image list these values will be controlled by a variable (what itself is controlled by arrow buttons, what increase/decrease its value by „1“. This value is also shown in a I/O-field).



Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign appeareance: (see basic funtions)

Parameterize the object: (register “General”)

- Ín the **„Stretch“**-box choose **„Fit to size“** to see your image in the objects rectangle you did draw
- Assign horizontal/vertical **orientation** if object is not stretched
- **Rotate** and **mirror** object as you want to assign an **image list** resource (must be existing - created at the image lists item)
- Assign a **variable**, what controls the image lists value

Text lists

Look at the text change of the users in visualization sample, screen “users”).with displaying the actual logged-in user. There were assigned 3 user (user 1, .._2, .._3).

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Assign appeareance: (see basic funtions)

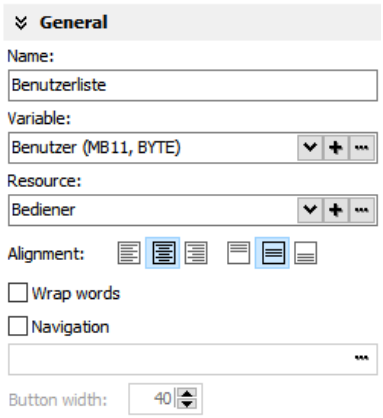
Parameterize the object: (register “General”)

- In the **„Stretch“**-box choose **„Fit to size“** to see your image in the objects rectangle you did draw
- Assign horizontal/vertical **orientation** if object is not stretched
- **Rotate** and **mirror** object as you want to assign an **image list** resource (must be existing - created at the image lists item)
- Assign a **variable**, what controls the image lists value

Color the object: (register „Appearance“):

Additional functions at text lists

- If there is a multiline text list, it is possible to assign **navigation buttons** for scrolling up-/down



Attention / objects in general:

- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

Trend viewer and trend archive viewer

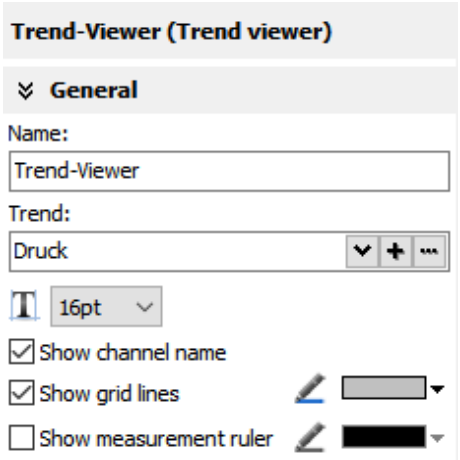
This object shows the samples of up to 16 values (= 1 trend) by time in a graphic way.
At CPU-T-devices there are available grid lines for better orientation, a higher scaled X-axis (more time values) and a movable measurement ruler to show archived values at different times.

Per trend there is shown only ONE vertical scale with the referring trend name above. To see the other scales click on the name and the scale and name of the next trend appears, and so on until the first trend comes again.

Place an object: (see basic funtions)
Assign border settings: (see basic funtions)

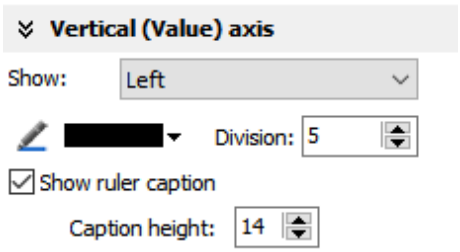
Parameterize the object: (register “General”)

- Assign a **trend resource**
  (use existing ressources)
- **Format the text** of the displayed channel name
- Check, if the **channel name** should be displayed
- Activate and color **grid lines**
- Activate and color the **measurement ruler**



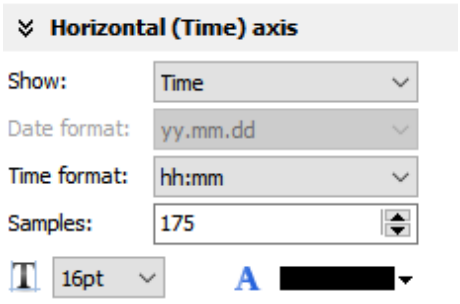
Parameterize the object: (register „X/Y-axis“)

- **Activate and color the Y-axis**
- Assign **scale divisions**
  (Rounding errors can appear!)
- Activate the rulers caption
- Assign caption height



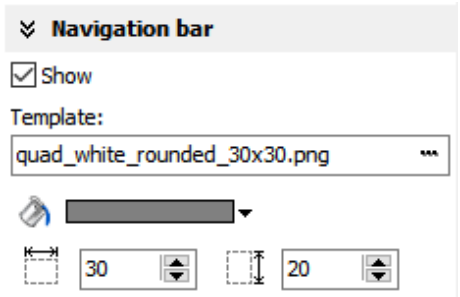
Activate and color the X(time)-axis

- **Assign time format**
  - Time + Date or
  - Time only or
  - Date only or
  - nothing
- Enter the **number of samples** to be displayed
  max. 200 in viewer, max. 65535 (expect longer displaying time)
  in archive viewer (values for CPU-T)
- Assign font, size and color of the X-axis



Navigation bar (for trend archive viewer only)
For CPU-V/-P-devices or for CPU-T-devices, when there shall not be an manual generated navigation, zoom bar

- Activate **navigation bar** for scrolling forward or backward
- Assign buttons and dimension it
- Assign background color for the bar
- 4 buttons with scroll functionality
  1st: back to first (oldest) sample
  2nd: one window back
  3rd: one window forward
  4th: forward to last (newest) sample



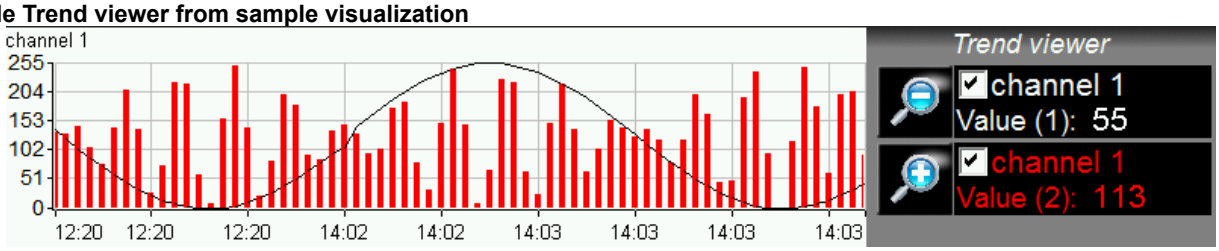
Information for time axis (X-axis) of CPU-V/-P-devices
Because the displayed time will be calculated in run-time, only 2 time stamps will be displayed:

- the time of the first shown sample (very left) and
- the time of the last shown sample (very right)

and nothing between it because it would overwrite each other.
100 samples will be displayed in the trend window. After the graph reaches the right border, the graph is moved 10 samples (10%) to the left (and the displayed times are changing).

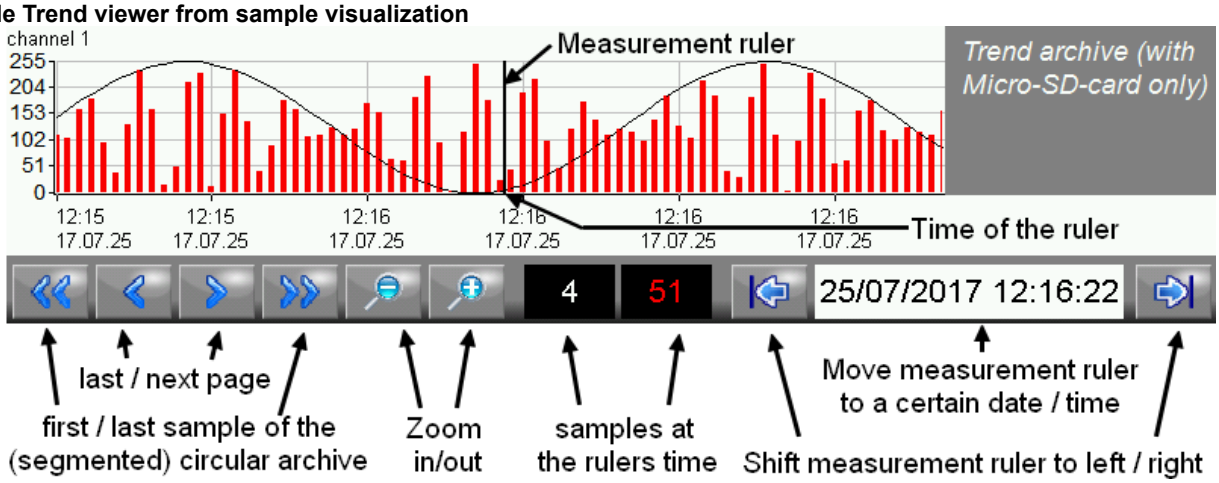
Visualization with the software „VisuStage“

Sample Trend viewer from sample visualization



Activating trend channel: by check box (function: Filter trend channel)
Display the actual sample value: by I/O-field (variables value)
Zoom in/out functions: as button function (Trend viewer function, zoom in/out)

Sample Trend viewer from sample visualization



Activating trend channel: by check box (function: filter trend channel)
Zoom in/out functions: as button function (trend archive viewer function, zoom in/out)

Manually designed navigation
First / last (actual) samples as button function (trend viewer function, zoom in/out)
Next sample page / sample page before as button function (trend viewer function, zoom in/out)
Values at the measurement ruler time as I/O-field (with values from trend archive viewer)
Move the m-ruler to a certain time as date/time-field (with source trend archive viewer)
(The measurement ruler can be moved by touching the touch display in the viewer)

Attention / objects in general:
- trends can be archived only, if there is a micro-SD-card inserted
- for CPU-V/-P-devices:
  trend viewer max. 100 samples will be displayed. When the samples reach the right border, they were shifted 10% to the left
  trend archive viewer: max. 800 samples can be displayed
- for the CPU-T-devices
- trend viewer: max 1000 samples can be displayed. When the last sample reaches the right border, the where shifted to left pixel by pixel.
- Trend archive viewer: max. 655.350 samples can be archived (=10 circular archives)
- there is shown only ONE vertical scale with the referring trend name above. To see the other scales click on the name and the scale and name of the next trend appears... until the first trend comes again.
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

Message viewing and archiving

There are different objects to display messages:

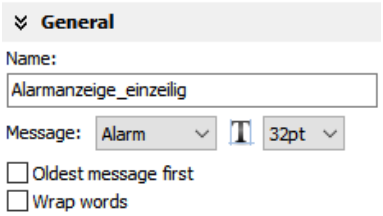
- blinking symbols (Message indicator)
- blinking text lines (Message indicator - text)
- oldest/youngest message only (Message viewer - line)
- viewer box for all actual messages (Message viewer - multi line)
- viewer for all archived messages (Message archive viewer)

For all standard parameters:

Place an object: (see basic funtions)
Assign frames: (see basic funtions)
Assign margins: (see basic funtions)
Assign visibility: (see basic funtions)

Message indicator symbol and text

This dynamic objects are used to inform the operator about an actual existing message (alarms/events) either with a blinking symbol or a blinking text line.



Indicator at alarms:

Blinking	if alarm has come and is not acknowledged
Display (without blinking)	if alarm is acknowledge but still there
Disappears	if alarm is acknowledged and no more there

Indicator at events:

Blinking	if event is there
Disappears	if event is no more there

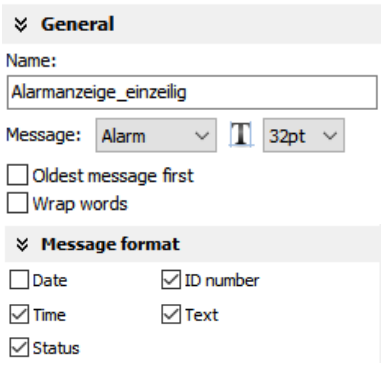
Placing as like at all other objects too, **Parameterization** is self explaining only **Assign** an event type - alarm or message.

Message viewer - line

This message line is made to **show only** the most actual message to be implemented as an information line only, with no opportunity to acknowledge or scroll.

Parameterize the object: (register “General”)

- Select “events” or ”alarms” to display
- Adjust, if oldest or newest message will be displayed
- Activate word wrapping

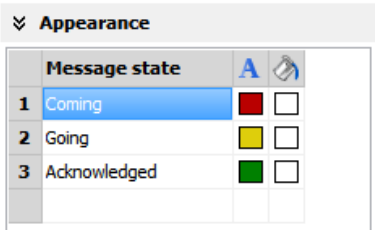


Formatting the object (register “Message format”)

- date
- time
- state
- ID-no. (Group)
- message text

Color the object (register „Appearance“)

All messages in the message viewers all single messages can be changed in color referring to their stat (Coming, Going, Acknowledged).



Visualization with the software „VisuStage“

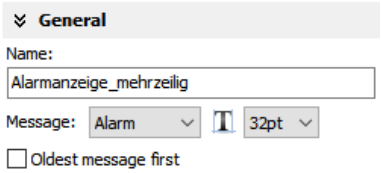
Message viewer multiline / Message archive viewer

This message viewer is made to show all **active messages**. It contains navigation functions (line **scroll**) and (for alarms) an **acknowledgment** function.
This message viewer is made to show all **archived messages** saved on the Micro-SD-card. It contains navigation functions (line and page scroll). To archive samples activate it by a hook as archivable.

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)
Formatting the object: (register “Message format” at the single line message viewer)
Color the object: (register „Appearance“ at the single line message viewer)

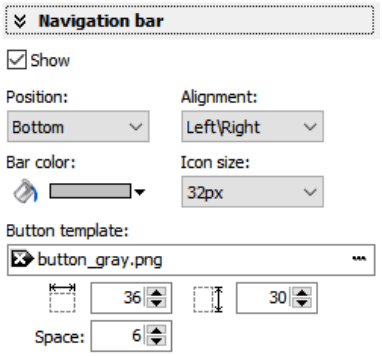
Parameterize the object: (register “General”)

- Select “events” or ”alarms” to display
- Adjust, if oldest or newest message will be displayed
- Activate word wrapping



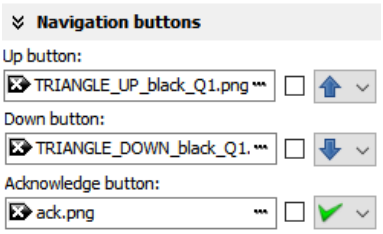
Formatting navigation bar: (register “Navigation bar”)

- **Activate** the bar
- **Color** the background
- Design the **placement** of the buttons
- Configure the **size of the icons** in the buttons
- Select a **button template**
- Design the **button size** width by height
- Adjust the **distance** between the buttons



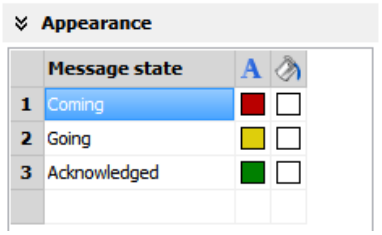
Designing navigation buttons: (register “Navigation buttons”)

- Select a **button template**
- Activate to insert a icon (Up/Down/Acknowledge)
- Select one of the icons from the list
- Scroll navigation (up & down):
  - Message viewer: line by line
  - Message archive viewer: line by line & page by page



Color lines and texts by status: (register “Appearance”)

- Depending on message status
  (such as Coming, Going, Acknowledged)
  different colors are possible for
  - Foreground (Text)
  - Background (field area)



Attention / objects in general:
- Dynamic objects shall **not overlap each other** (Exception: touch area may).
- Simulation with F9 or „▶“ creates an error report, click on error and come directly to the failed object.

Recipe viewer

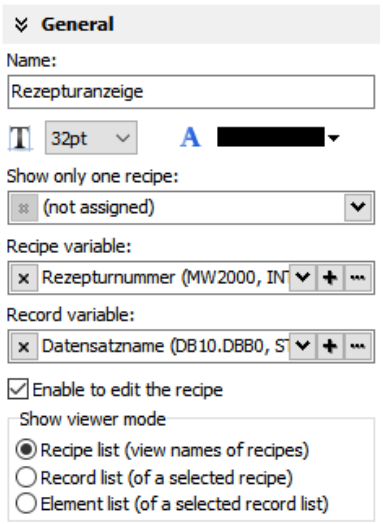
The resource “recipes” will displayed by this object in 3 different windows:

- Recipe viewer window
- Records viewer window (of each recipe)
- Element configuration window

Place an object: (see basic funtions)
Assign border settings: (see basic funtions)
Assign access right: (see basic funtions)

Parameterize the object: (register “General”)
Name:

- internal name of the recipe viewer (independent of project language)



Show only one recipe:

- reduces the list to only one (selected) recipe (e.g. if this machine is designed for only one recipe but the project contains more recipes for more machines)

Recipe variable: *(to display the actual used receipt name)*

- Assign it, if the internal number of the recipe shall be stored in a S7-variable, what will be linked with the receipt name
- (resource „View Name“ in a text list and can be displayed as active receipt name)

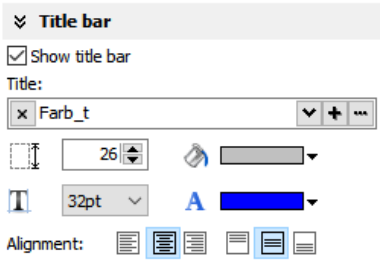
Record variable: *(to display the actual used receipt name)*

- Assign it, if the internal number of the record shall be stored in a S7-variable to be shown in a I/O-field to display the actual record name
- (At first simulation assign any value to this variable in the SimuStage, than you see the active record after press the button „Save to PLC“)

Enable to edit the recipe:

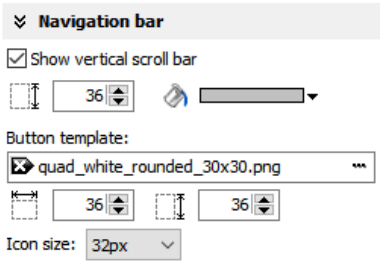
- Allows manipulation by the operator (changing the element values, storing as a new record, deleting records and so on)

Parameterization of the title bar
(general for recipe lists, record lists, element lists)



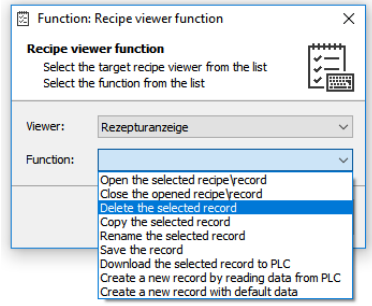
- It is recommended to use a title bar and to assign a title to the recipe list.
- The titles of the other lists bellow (record list and element list) will be assigned automatically by the inserted data.
- (Title of the record list in the sample „red shades“, element list in the sample Rose“)
- All other settings are self explaining settings to design the title bar

Parameterization of the navigation bar
(general for recipe lists, record lists, element lists)



- This activation is recommended, when more recipes, records or elements are used than fit into the screen window and scrolling is necessary.

- All settings are self explaining settings to design the scroll bar



Hint:
Instead of a navigation bar you can assign own buttons by using “Receipe Viewer Function” at releasing the button.

Parameterization of the operators control elements
(general for recipe lists, record lists, element lists)

By the register „Navigation buttons“ can be selected all function keys you need.
Their button design can be substituted by a customized button.

Recipe list:

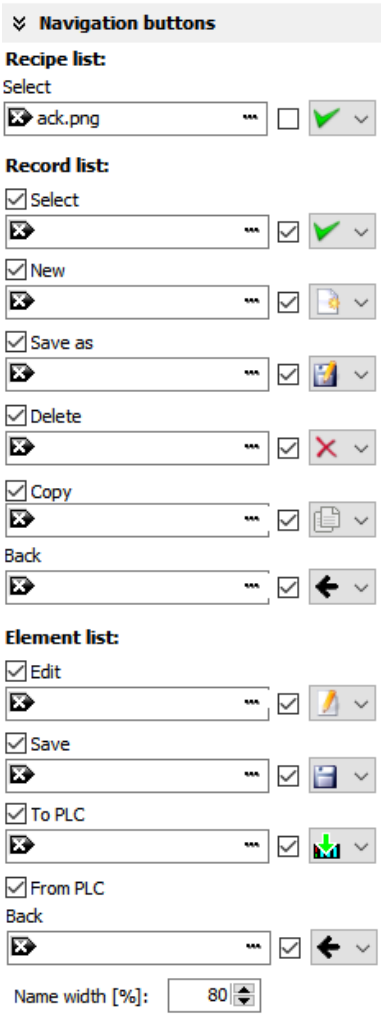
Select:	activates a recipe from the recipe list (will be highlighted), by touching directly too, this list can be hided, when you have 1 receipe only

Record list:

Select:	activates a record from the record list (will be highlighted), by touching directly too
New:	creates a new record with existing element structure
Save as:	saves the record with a new name in the Micro-SD-card (an string input field opens automatically)
Delete:	deletes an existing record
Copy:	copies an existing record
Back:	returns to recipe list

Element list:

Edit	opens up a virtual keyboard to edit the elements value
Save:	overwrites (in the Micro-SD-card) an existing value with actual element values
To PLC:	transfers the stored values and records into the PLC
From PLC:	reads variable for element values from the PLC
Back:	returns to record list



Assign a width of the recipe viewers window to the displayed elements name to have a efficient window design.

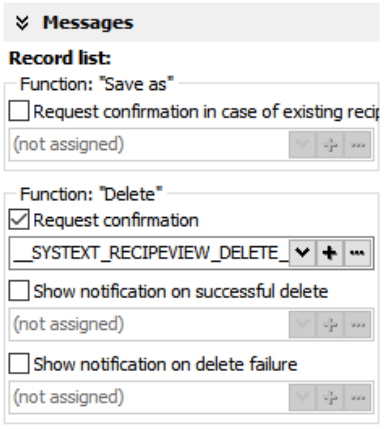
Different confirmation messages (register “Messages”)

In the 3 different viewing windows (recipe, record- and element-) it is useful to ask the usere “Do you really want to...?” BEFORE he is manipulating something wrong.
These message texts are system messages and can be activated here

At the text resources you will find these text phrases as system texts always on top of the texts and in the csv-file with the prefix „__SYSTEXT“.

These messages will be displayed BEFORE storing, deleting, data transfer to/from PLC

(These register is shown just for the record-list only, the others work similar to it.)



Views of recipe lists, record lists, element lists at runtime

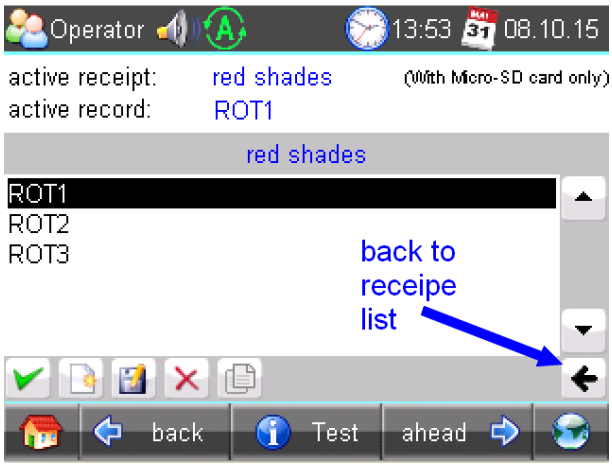
Recipe list

The recipe list contains all recipes, what are pre-assigned in the VisuStage before. It has an clear and an uniform element structure, what was pre-assigned in the software VisuStage before.

The assigned name for all recipes is displayed on the title list.

The choice of the a recipe is done by a touch on the name and a touch on the bottom left symbol to open it. Than the display changes into the next screen with the record list.

To exit the recipe administration, add a common button and assign the function „change screen“ whereto ever you want.



Record list

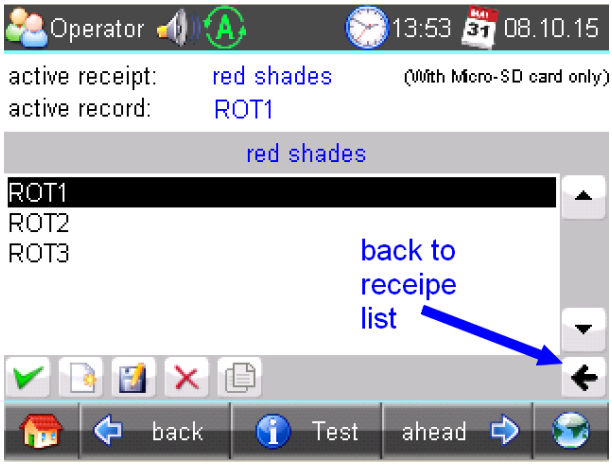
The record list contains all user-created records of a certain recipe (in this sample of the recipe „red colors“).

The regarding recipe name is displayed on the title list. Select it by touch on the name and press „Edit“ or create a new one. Than it switches directly into the element table to edit values and to store it in an automatically displayed string input field.

New from VS 2.0.1.5: The actual record will be displayed in the color of the recipes name in the title bar

The function buttons are displayed in the lower left bar:
Edit | New | Save | Delete
With the lower right „←“ this screen changes back to the recipe list.

Note: *Because recipe data are kept in the Micro-SD-card there is possible no simulation with these values (your PC has not such a memory extension).*



Element list

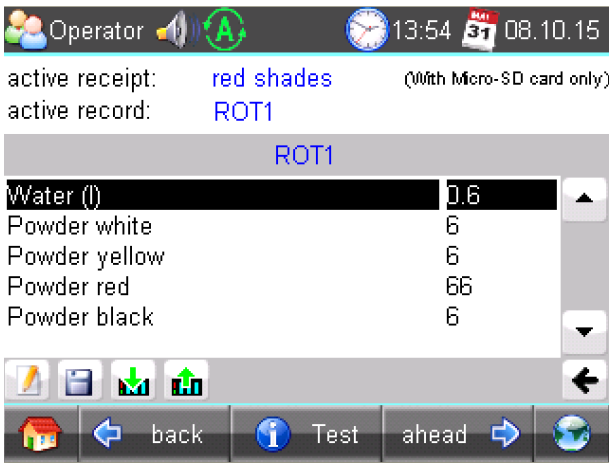
The element list contains all user-created values of a certain record (in this sample of the record „ROSE_1“)

The regarding record name is displayed on the title list.

By touching an elements value a numeric keyboard opens up automatically to type in the new value of the selected element.

The function buttons are displayed in the lower left bar:
Edit | Save | Read from PLC | Write to PLC
With the lower right „←“ this screen changes back to the record list.

Note: *Because recipe data are kept in the Micro-SD-card there is possible no simulation with these values (your PC has not such a memory extension).*

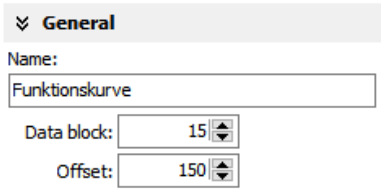


Function graph

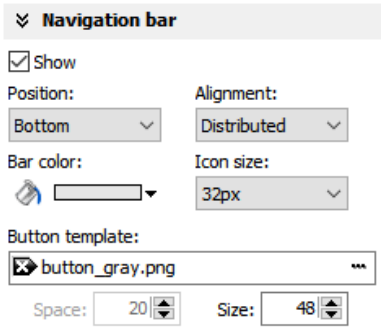
This viewer is able to display a number of x;y-values from a data block. There could displayed up to 2000 couples of x;y-values each as one pixel.

Place an object: (see basic funtions)
Assign visibility: (see basic funtions)

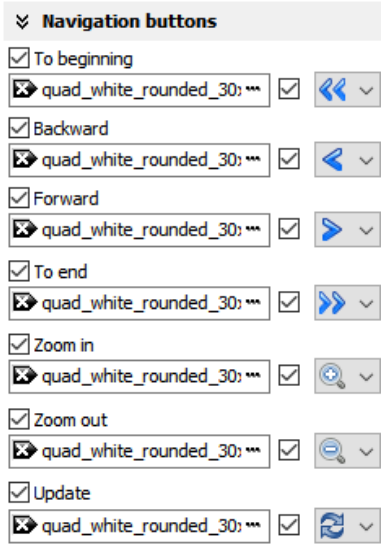
Parameterize the object: (register “General”)
- **Assign a data block and offset**
  This data block contains the values to be displayed



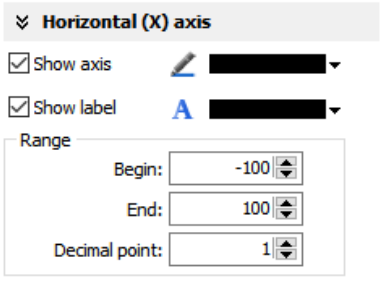
Adapt the navigation bar: (register “Navigation bar”)
- **Activate** the bar
- **Color** the background
- Design the **placement/alignment** of the buttons
- Configure the **size of the icons** in the buttons
- Select a **button template**
- Design the **button size** width by height
- Adjust the **distance** between the buttons



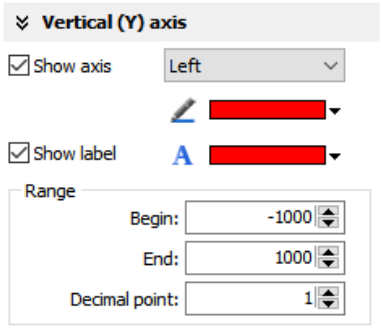
Designing navigation buttons: (register “Navigation buttons” / see right)
- Select a **button template**
- Activate to insert a icon into your buttom
- Select one of the icons from the list



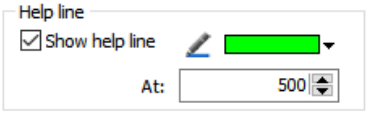
Parameterize the object: (register „X/Y-axis“ / see below)



Activate and color X-axis
Activate and color the label of the X-axis
Assign scale values
- Start value
- End value
- Decimal point



Activate and color Y-axis
Activate and color the label of the Y-axis
Assign scale values
- Start value
- End value
- Decimal point



Activate and assign help line
- Y-Value
- Color (one help line possible for an Y-value)

The data block with the x;y-values to display must be devided into two parts:

- Header — Information about trend coordinate structure.
- Data — Array of [X,Y] pairs. Data type must be defined in Header

Header structure

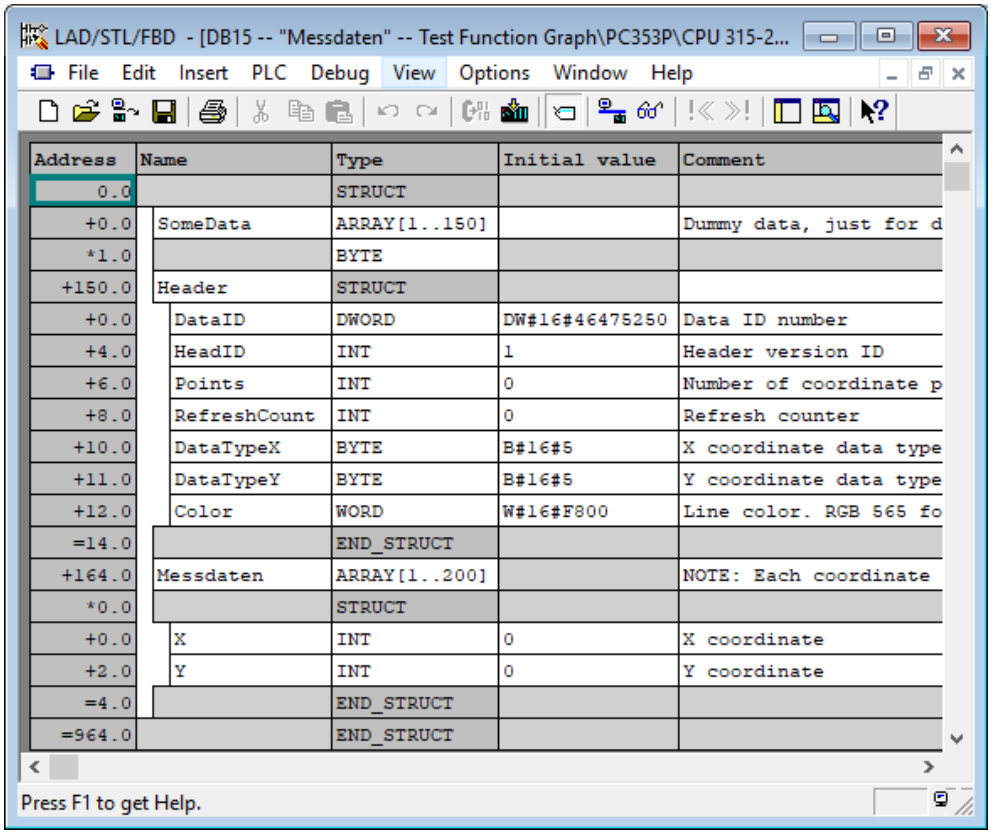
Offset	Name	Data type	Value	Notes
0	DataID	DWORD	DW#16#46475250	Constant value
4	HeadID	INT	1	Constant value
6	Points	INT		Number of [X,Y] pairs [0 .. 2000]
8	RefreshCount	INT		Refresh request counter. S7-User program should increment the value, in order to request refreshment in VisuStage.
10	DataTypeX	BYTE		Data type of X coordinate value 2 = BYTE 3 = CHAR 4 = WORD 5 = INT 6 = DWORD 7 = DINT 8 = REAL
11	DataTypeY	BYTE		Data type of Y coordinate value (see DataTypeX notes)
12	Color	WORD		RGB color value, format 565 Bit0..Bit4 Blue component Bit5..Bit10: Green component Bit11..Bit15: Read component

Data structure

Array of [X,Y] pairs (structure).
It's structure completely depending on user demands and must be correctly defined in Header structure.

For example:
The following data block contains trend coordinates with following property

- each coordinate value (16 bit integer, value range -32768 .. 32767)
- 200 coordinate pairs (sample, max. 2000)
- Line color is blue



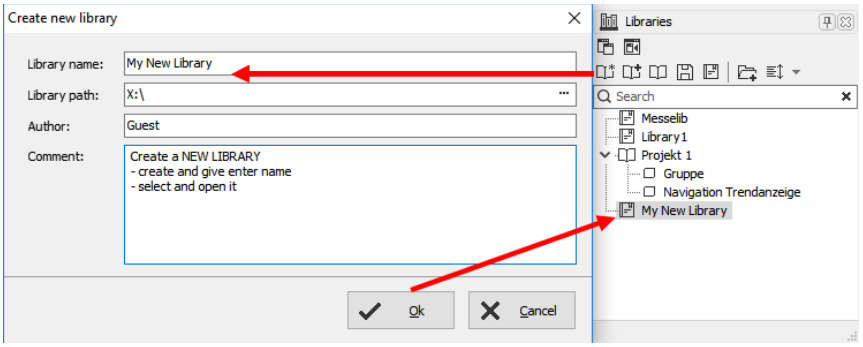
Working with library elements

Library elements were used to have available pre assigned elements for different projects. This is very useful as base for multiple visualizations and show a corporate design e.g. at navigation bars, state bars and login pages. For all, who want to copy (quick´n durty) only, find more about copying more in basic funtions.

Creating libraries and ~ elements

Single or multiple (grouped or ungrouped) objects can be added to a library, consisting of at least one element.

If a library element consists of multiple objects, it is better to group these objects before. So the positions between these objects are fixed.



To create a new library
- 1. select the "new library" button
- 2. assign name, path and comments
- 3. select the new created library and
- 4. open this new library

To add new library elements
- 1. open your library
- 2. move objects or groups directly on the library sign

Using library elements

The Library function is self explaining:
- either **CREATE** a new one,
- or **ADD** another library from any PC-directory (*.vsl) into this library list
- or **OPEN** a library from this list

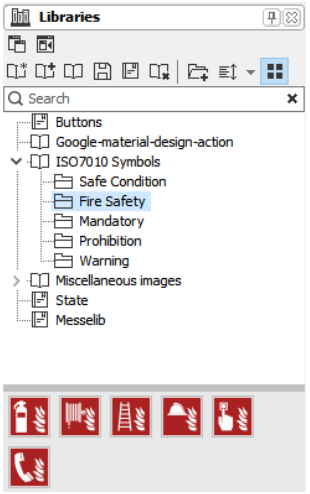
- Group some objects and name the group.
  Copy this group by drag´n drop into your open library
  or
- Click at a group in your open library and copy it by drag´n drop into your screen.

Variables, texts and images will be taken over 1:1 from the library element into the new screen/project. All other resources must be assigned new.

Library elements will be integrated in the visualization project (*.vsproj).

To send your libraries to others, always zip and send complete libraries (folders on the VisuStage-PC) including your library elements (*.vsl) .

The library offers images, button shapes and state areas in a SVG-format, who will be stored at the image-resorce of the project. To get a preview right in the library window, select the icon as shown in the figure to the right.

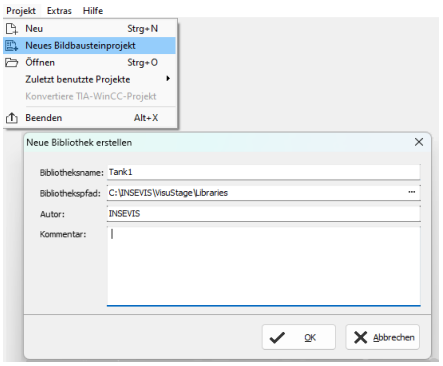


Hints:
The library objects in SVG-format allow a very simple color-change (at the rider „Appearance“), much easier then with image lists, what will simplify your visualization project by far.

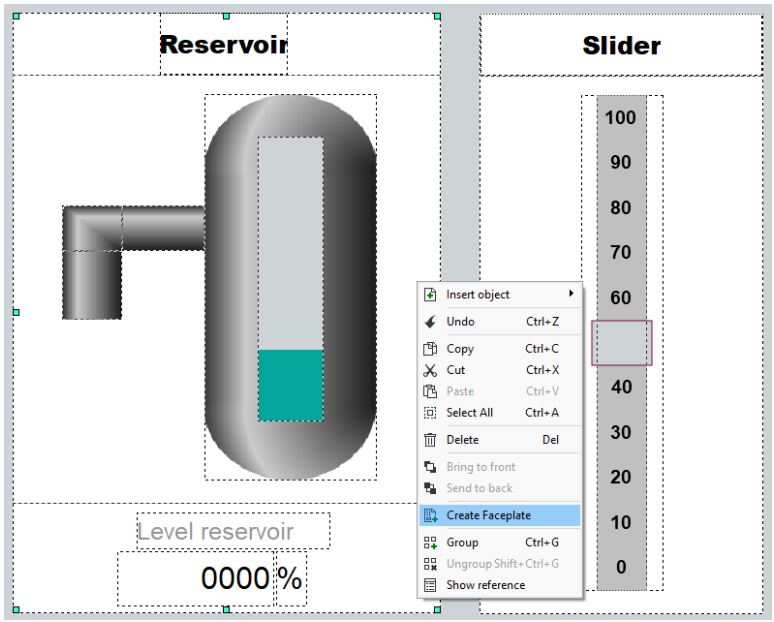
Also when scaling the size of the objects the quality loss is less than at bmp-, jpg- oder png-files.

Working with image blocks

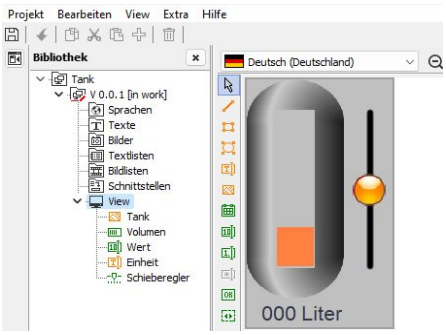
As of version 2.2.0.0, image blocks can be created.



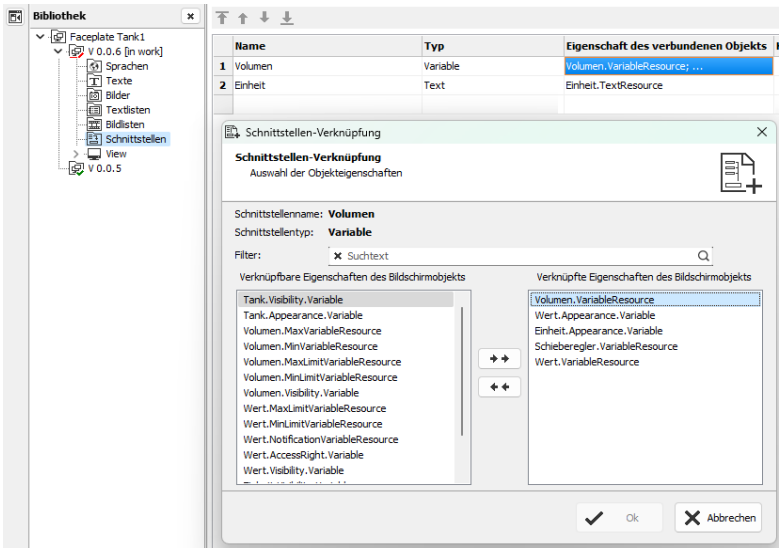
For this purpose, a new image block project is started in an empty VisuStage or an existing image block (attention: file format *.vsl) is opened in a library and a new release is created with the right mouse button (display as [in work]).



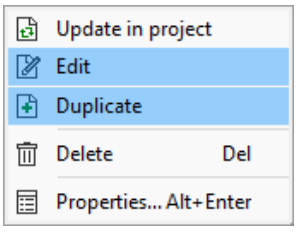
Another possibility to create image modules, which is especially useful for existing projects, is to select the desired objects in an existing view and then create an image module by right-clicking. This can then be stored under the desired name at the desired location. Afterwards, the image module that has just been created appears in the library.



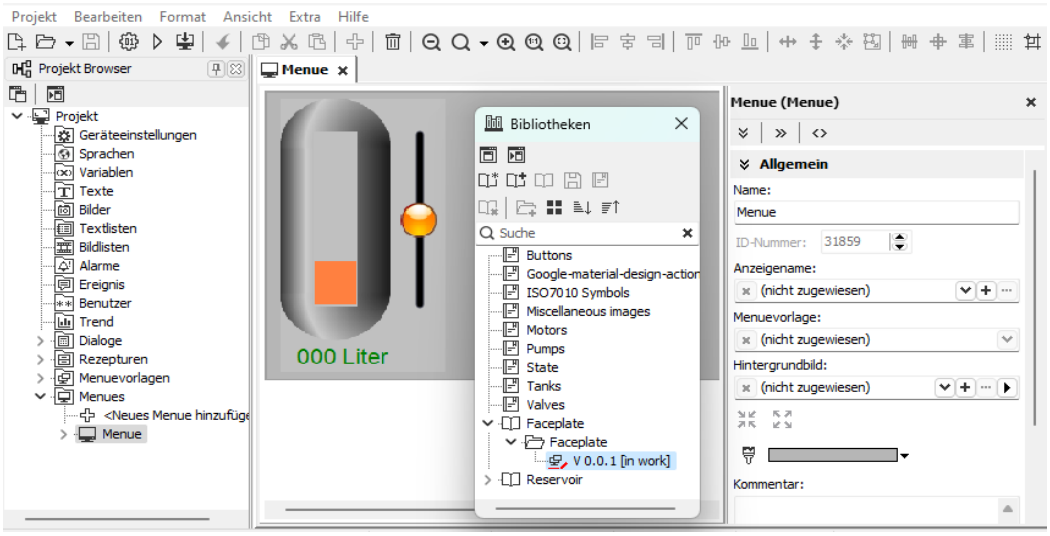
n the "View" resource, the image module is created with the familiar VisuStage objects (not all of them are useful for use in an image module and are therefore grayed out). The configuration takes place in the usual property windows as with the previous VisuStage also. However, no variables can be assigned (this is done in the resource Interfaces).



In the "Interfaces" resource, after adding an interface with the right mouse button, selected properties of the used individual image block objects are assigned as a property for the entire image block.
This is done in the "Property of connected object" column. These can be selected variables, texts or other properties. The image module is then integrated in the VisuStage project via these properties, which are now linked to the image module.



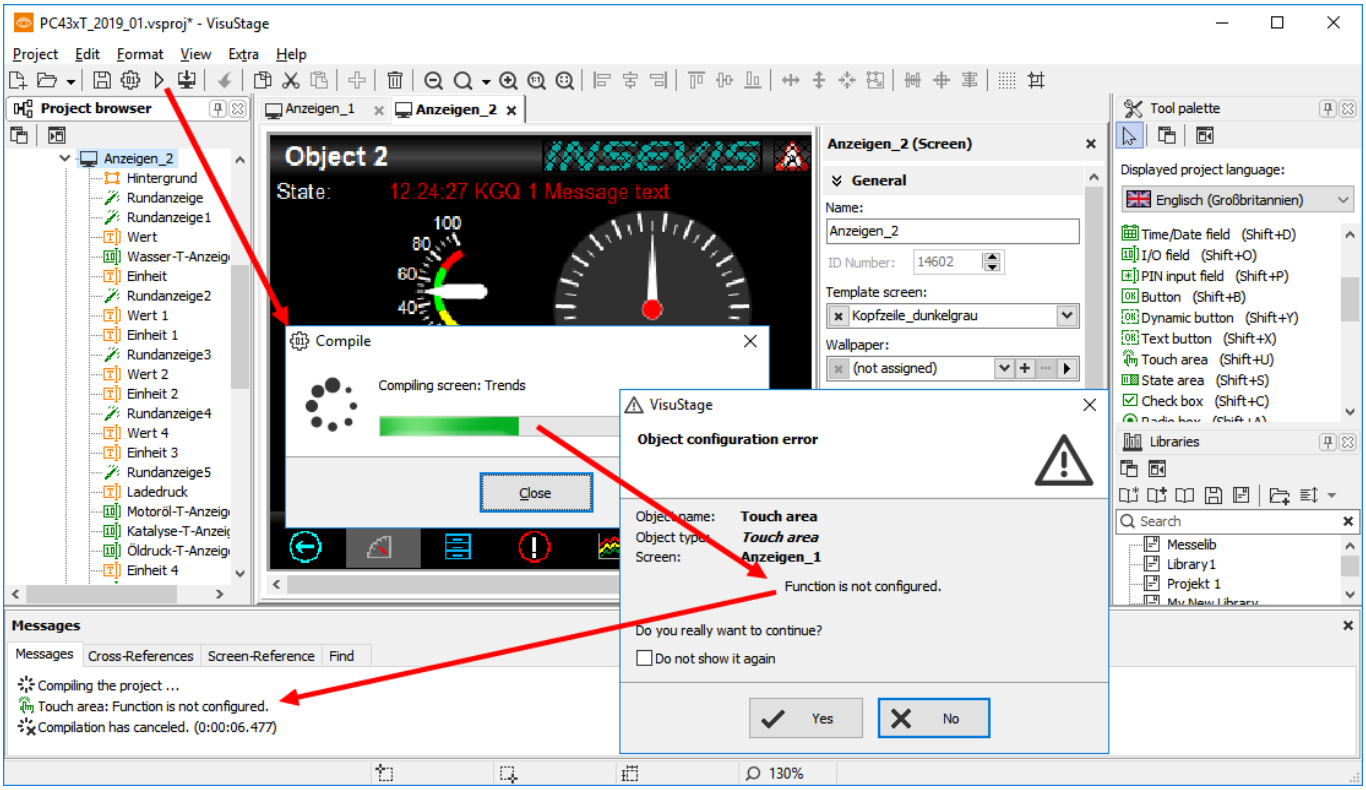
he image module can now also be opened, edited and duplicated directly from the library. This is done by right-clicking on the object, then selecting the desired option. (Changes made in one picture component affect all used picture components of this type. If picture components are to be excluded from this, a duplicate of the picture component must be created).



After creating the image module, this release is released (right mouse button) and the image module is saved (do not forget, otherwise the module remains locked in the [in work] state). Before release and saving, the function block is subjected to an automatic plausibility check. If errors occur, they are displayed and the module is not saved. Only after these errors have been corrected is the function module released and can be used. Now the project can be closed in VisuStage. The next time a VisuStage project is opened, the image module appears with its releases in the library at the bottom and the released versions can be dragged into a menu after opening the library. There, the linked properties of the image module must then be assigned to the resources of the visualisation project.

Simulation and download

Error treatment



- After finishing the visualization start the simulation by F9 or „▶“
- If there are errors, the compiler stops and creates a error message (object and error type).
- Double click this message and you will see the failed object.
- Correct the error and restart the simulation by „F9“ or „▶“

If all errors are cleared, the compiler creates a *.bin-file and the simulation tool „SimuStage“ starts automatically.

Visualization with the software „VisuStage“

Simulation and download

There are 2 ways to simulate your visualization:

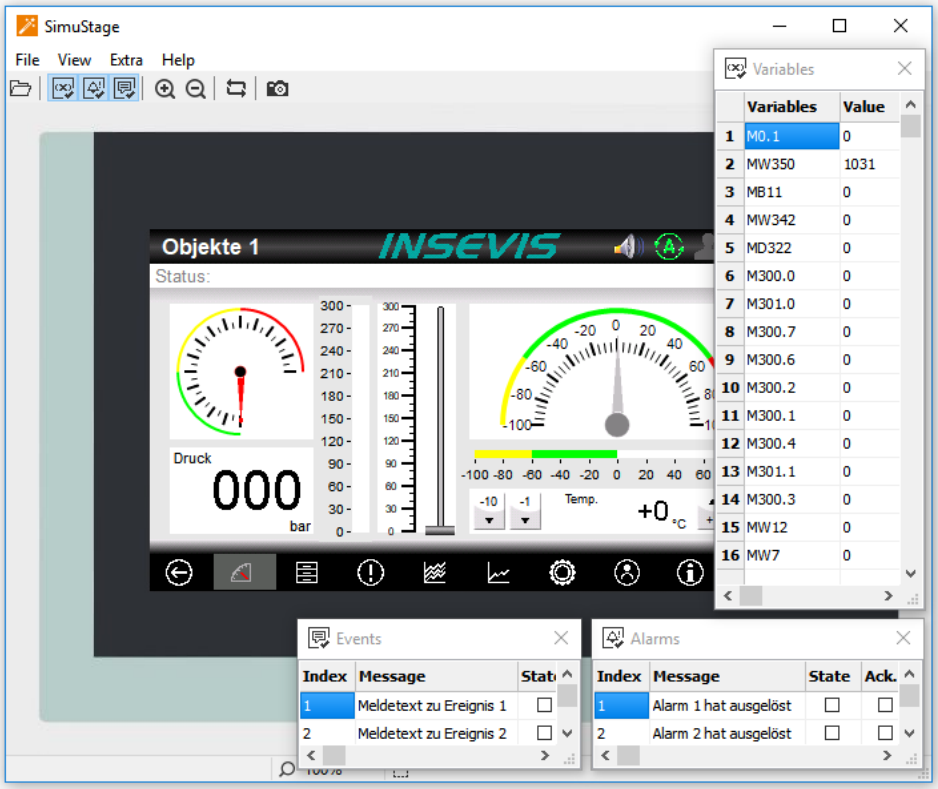
Manual simulation
Starts the integrated simulation tool SimuStage in a new window.
The operation by the user is by mouse pointer simulated like at the real touch panel.

Automatical simulation
Starts the SimuStage together with the Siemens-software S7-PLCSIM (what must be installed at the same PC).
The process data come from S7-PLCSIM (not from PLCSIM Advanced – this is for 1200/1500CPUs). Of course, S7-PLCSIM does not simulate INSEVIS-SFCs/SFBs.

Manual simulation
By manipulating variable values in the variable table you can cause changes manually.

Events can be simulated (Coming - Leaving - Acknowledge) by checking/unchecking boxes in the SimuStage.

It is also possible to generate screen shots of the simulated screen for creating any kind of documentation.

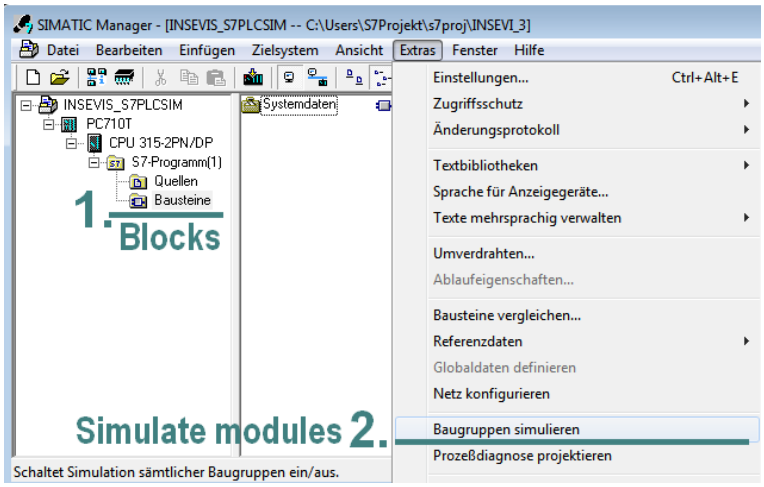


Automatical simulation
Start in the VisuStage by
“Project” → “Simulate with S7-PLCSIM”

The (already installed) S7-PLCSIM must be opened from the S7-programming tool.

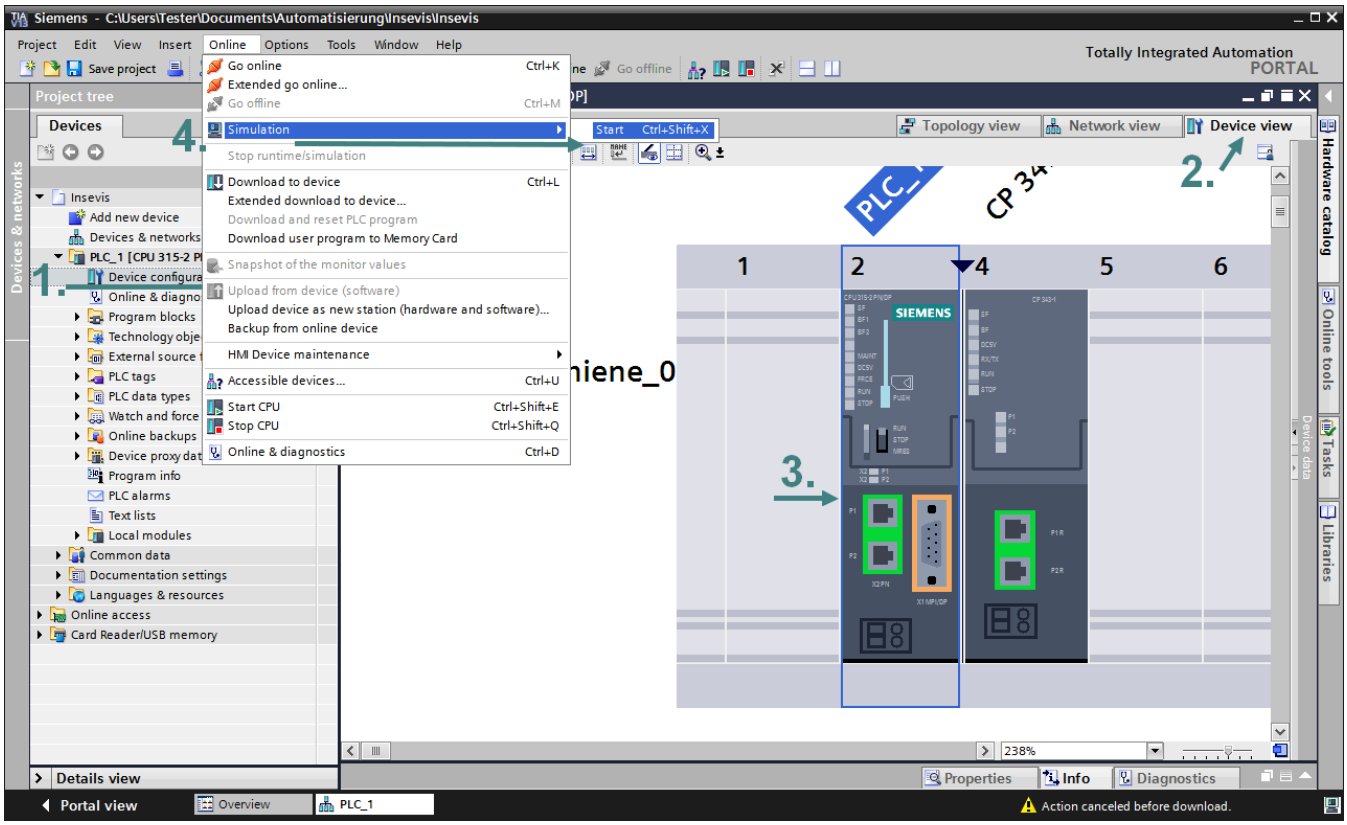
Here you find the call from Simatic®-Manager

1. select “Blocks”
2. click on “Simulate modules”



Visualization with the software „VisuStage“

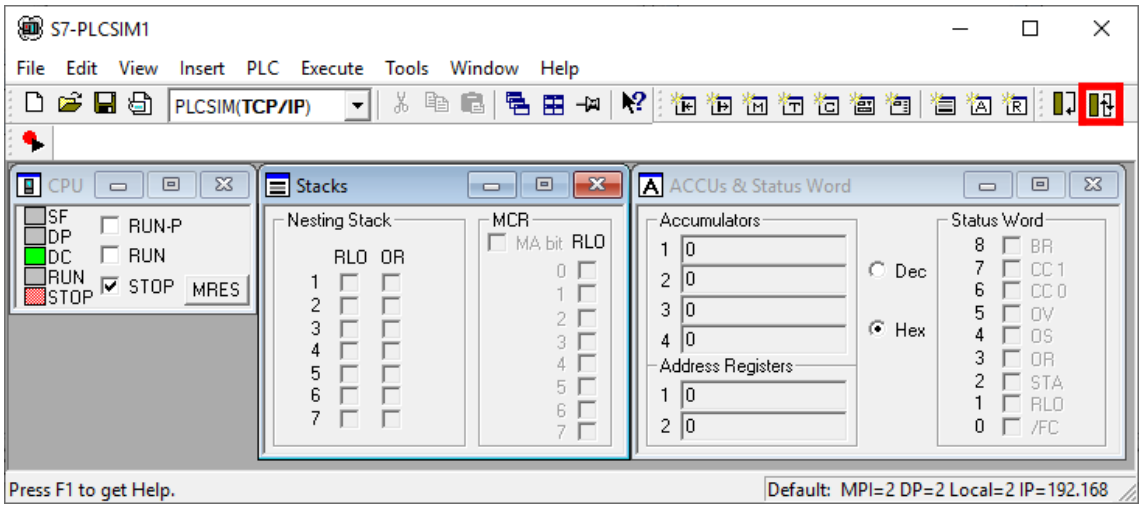
Here you find the call from Simatic®-Manager



1. Goto “Device configuration”
2. Select “Device view”
3. Select the CPU
4. select “Simulation” and “Start”

If the S7-PLCSIM was called, the S7 program is simulated **ONCE** only.

For a **CYCLIC** simulation use the right button (green frame)



Visualization with the software „VisuStage“

Download

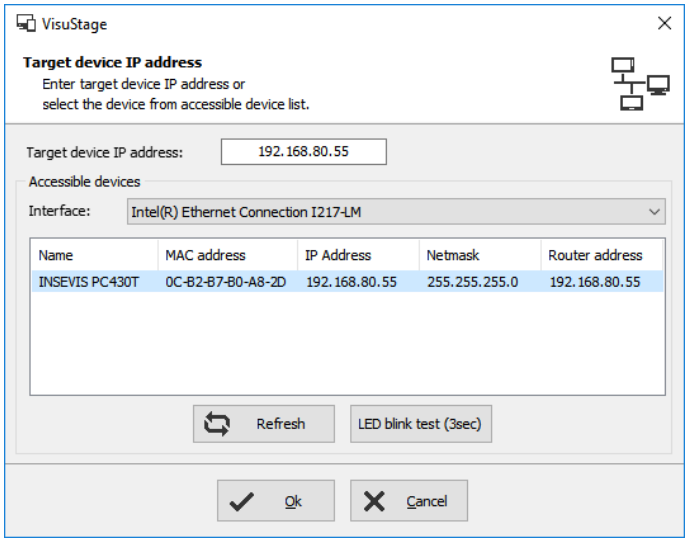
Transfer the visualization binary file to the target device

To transfer the project (start by key F12 or icon) use

- the IP-target-address from the project data
  or
- the IP-addresses in the network area you are connected to by using the magnifier icon (identification in networks with switches only, not with routers, because they filter the Ethernet-Layer 2, what is necessary for identification)

Alternatively download the compiled project as vsbin-file

- via ServiceStage or
- by a Micro-SD-card
  (with hard reset and clear all)



Attention
If some devices with **the same IP-address** should be connected, this will be stopped from Windows-OS. To switch off this function, please insert in the **cmd** this command: **arp -d** (this deletes in your PC the old MAC-address from the foregoing device, what was stored in your PC).

Remote access with the software „RemoteStage“

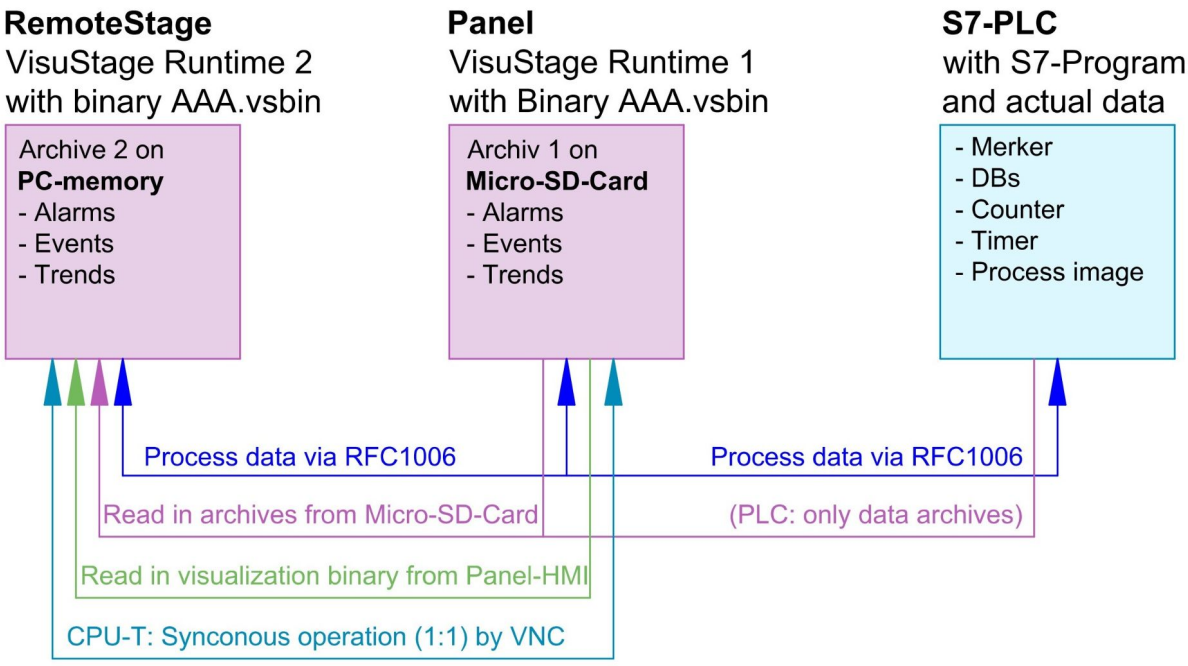
General

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 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.



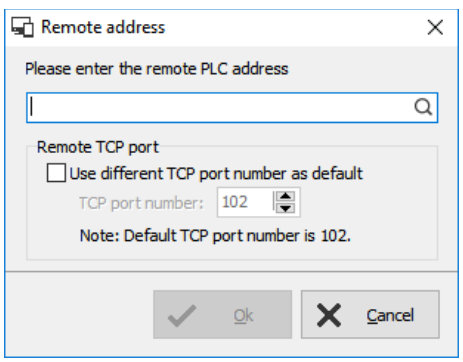
If the RemoteStage should be installed as an batch call on the end user´s PC this is all you need to write:

[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.)

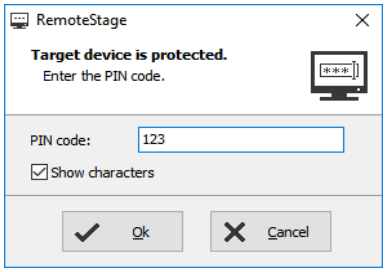
This address will be linked with this binary.
So it is not necessary to re-type it again.



Remote access with the software „RemoteStage“

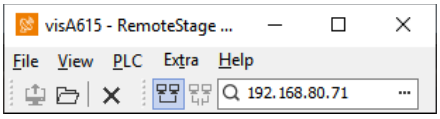
Insert PIN code

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 menue 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-menues are self explaining. It starts with a black screen and displays the remote visualization after pressing “connect” key. 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.

Hints: Following properties / functions / setting **will not be supported** by RemoteStage (viewing mode)

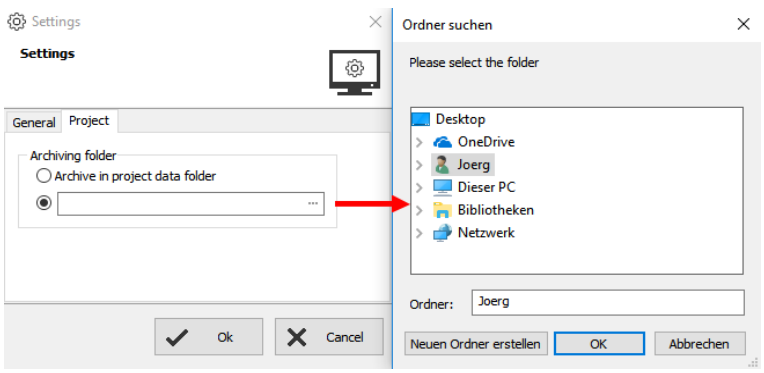
- "**display backlight control**" and "**Screen- and language control**" - from “Screens - Settings“
- "**Synchronize the HMI time with partner time**" – from resource „PLC-device settings“
- "**Enable to change partner time in runtime**" – from resource „PLC-device settings“
- “**Change system time**“by Screen-object "Time/Date field"
- Button-functions: „**increment / decrement brightness“, „backlight off“, „activate clean screen“**
- **VNC Server**

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.

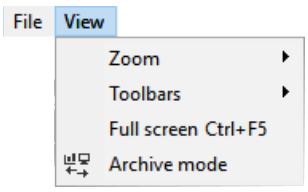
At the menu „Settings“ will be set up, if these date will be stored
- in the project folder (where the visu binary is kept) or
- at another path.



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 con be converted to csv-format by RemoteStage “Archive mode” (a 2nd instance of RemoteStage) only.

Open this 2nd instance in the “View”.menue (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 funtion „archiving“ in the VisuStage-project
Trends (with all channels)	by all Panel-PLCs and Panel-HMIs → by activating the archive funtion „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 nto the PC-network from left → right:
- Alarm archive
- Event archive
- Recipe archive
- Trend archive
- Data(DB-)archive

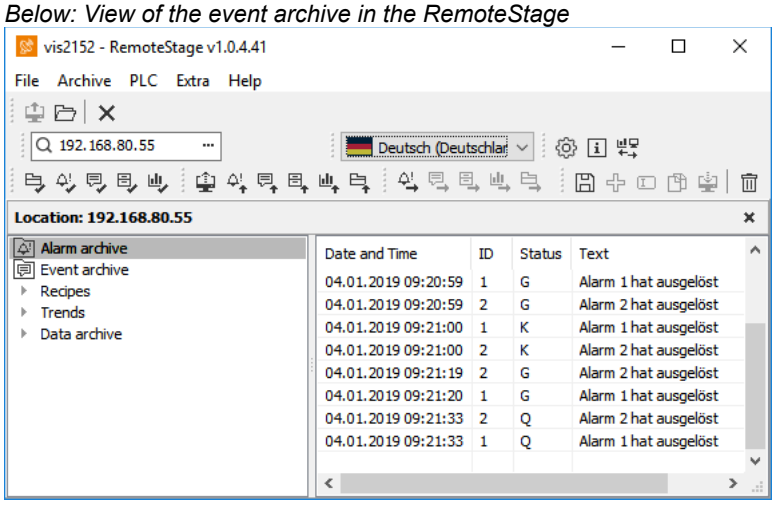
Editing and downloading of modified archives from left → right:
- Save archive
- Add new archive
- Rename archive
- Copy archive
- Download archive

* To open these files insert a nearly valid IP-address and open the binary of the visualization what fits to your project. Than the binaries can be converted into csv-files

Display and export event and alarm archives

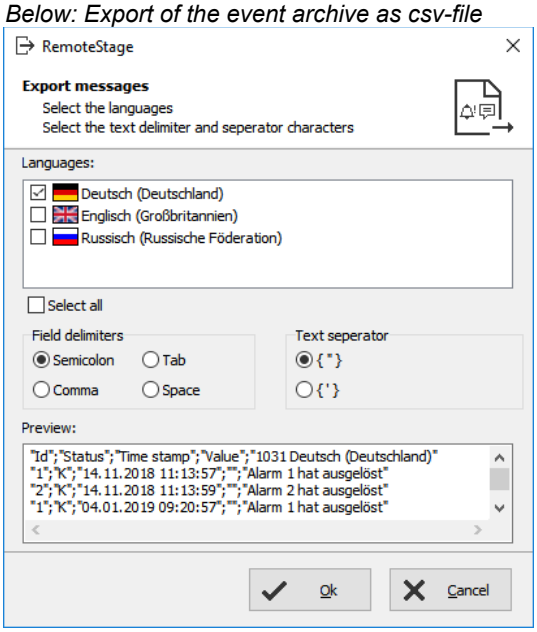
Archive data will be read in from Micro-SD-card, can be displayed and e.g manually saved as csv-file.

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.

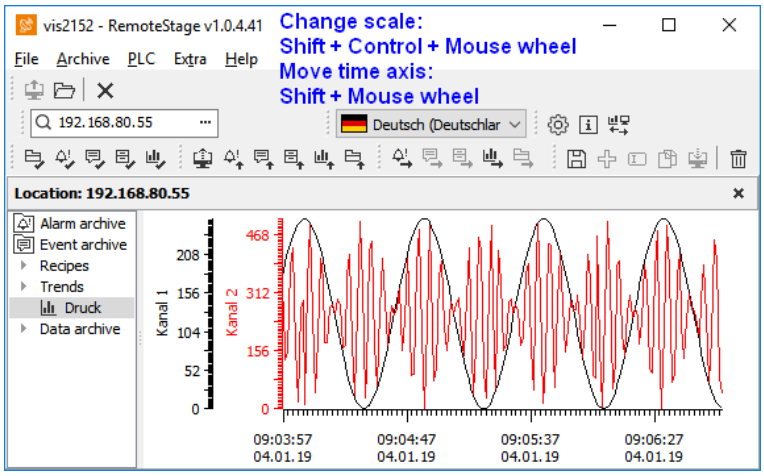
Remote access with the software „RemoteStage“

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

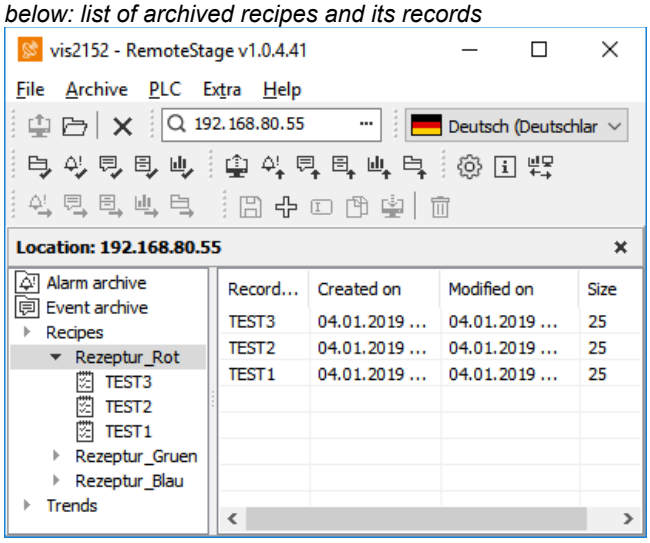
for FURTHER EDITING stored as csv-file at the PC

– converted into csv and stored, edited and written back / downloaded into the remote PLC or

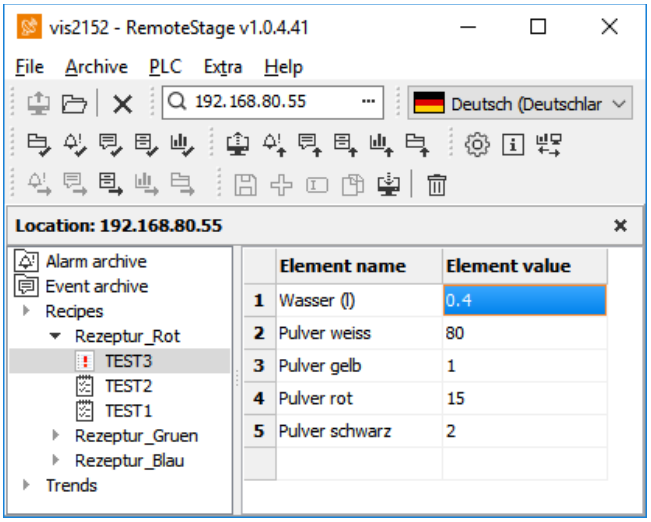
for COPYING THE WHOLE RECIPE SYSTEM saved as binary at the PC

– 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.

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 from the Micro-SD-card in the folder UserData (see system functions).

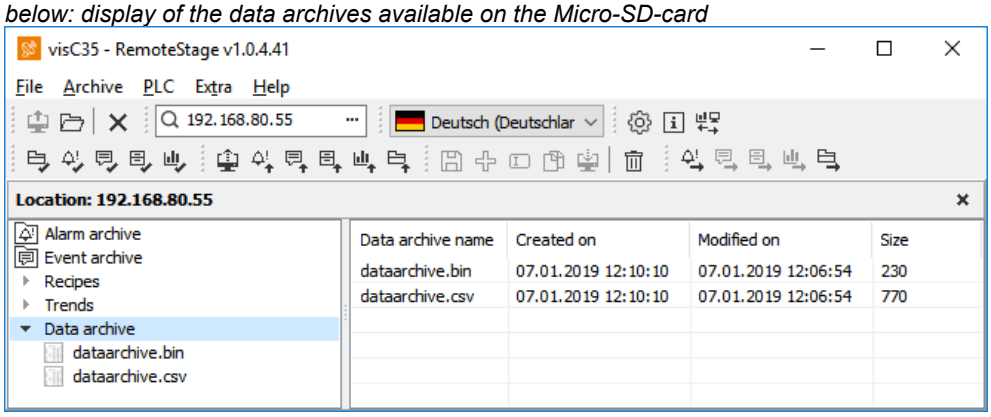
In the RemoteStage-shell no DB-archives can be written from PC to the Micro-SD-card of the PLC manually. This will be done by command lines in batch files (see samples in next pages).

Data archive display and storing

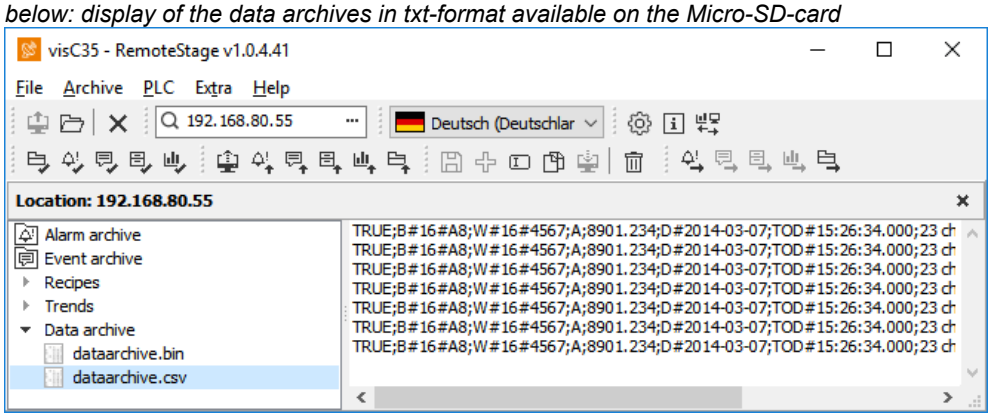
These data (DB) archived over S7-program to Micro-SD-card, can be read in and displayed and stored to the PC in the assigned format.

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).

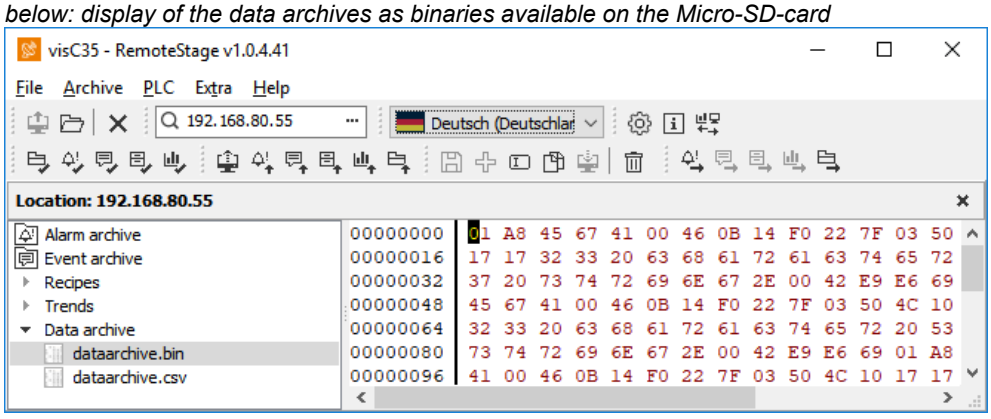
below: display of the data archives available on the Micro-SD-card



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



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 device 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

<path>\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 reconnect to remote device after disconnect

With his function the lost connection to the partner device will be reconnected after a configurable time in seconds.

Command line

<path>\remotestage.exe /restart=[seconds] /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“)
If the parameter is „restart=0 (default), the connection will be restarted only once. For all others the value is the time in seconds from loss of connection abort to the planned start of the new reconnection.

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

<path>\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

<path>\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.

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 menue →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 (no line feed please, write all in one line!)

<path>\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 ressources / 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** extention, 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 (supposed remotestage.exe is at C:\):

Command line (no line feed please, write all in one line!)

C:\remotestage.exe /mode=alarm_archvie /r=192.168.80.50
/o=d:\archive\alarm.csv /v=d:\visu\myvisu.vsbin

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 (supposed remotestage.exe is at C:\):

Command line (no line feed please, write all in one line!)

C:\remotestage.exe /mode=trend_archive /r=192.168.80.50
/o=D:\archive\trend6163.csv /t=6163 /v=D:\PC156xT_2017_03.vsbin

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 (no line feed please, write all in one line!)

<path>/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 Name of the record for saving of SPECIAL (single) records.
If this parameter is not assigned, RemoteStage reads in ALL records, what belong to the recipe *<recipe name>* and stores it in a folder assigned by the parameter "/o=" before.
→ To configure the format of the output file as csv, the parameter "/format=csv" must be set, otherwise all data will be stored a a binary.
In the csv-files this format is predefined: **;** (Semicolon) as field limiter and “ (quotation mark) as text separator.
If this parameter is assigned, the format of the output file MUST BE predefined by the extension of the record name.
→ If the file has an extension .csv or .txt, than it is a csv-format, otherwise the output file format is binary.

output file: Path and file name of the recipe record for the storing in the PC (optional parameter)
If the parameter "/o=" is not assigned, a folder "Recipes\<recipe name>“ will be created in the project folder (where the visualization binary is kept) automatically and the recipe record stored as <record name>.rec.
If file name includes space character, it should be quoted with **"** (quotation mark).

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 (supposed remotestage.exe is at C:\):

Command line (no line feed please, write all in one line!)

C:\remotestage.exe /mode=recipe /r=192.168.80.50 /v=d:\visu\myvisu.vsbin /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 (supposed remotestage.exe is at C:\):

Command line(no line feed please, write all in one line!)

C:\remotestage.exe /mode=recipe /r=192.168.80.50 /v=d:\visu\myvisu.vsbin /recipe=Rezeptur_Rot /o=D:\archive /format=csv

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 (no line feed please, write all in one line!)

<path>\remotestage.exe /mode=recipe_download /v=<project binary>
/recipe=<recipe name> /record=<record file(s)>

Parameters

project binary	Binary fle 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 (supposed remotestage.exe is at C:\):

Command line (no line feed please, write all in one line!)

C:\remotestage.exe /mode=recipe_download /v=d:\visu\process.vsbin
/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 (supposed remotestage.exe is at C:\):

Command line (no line feed please, write all in one line!)

C:\remotestage.exe /mode=recipe_download /v=d:\visu\process.vsbin
/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 SFC207to the requested target path into the PC. A modification of the csv-format is not possible.

Command line

C:\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 (supposed remotestage.exe is at C:\).:

Command line (no line feed please, write all in one 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

<path>\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 (supposed remotestage.exe is at C:\):

<path>\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 (supposed remotestage.exe ist at C:\):

<path>\remotestage.exe /mode=data_download /archive=d:\configs*.* /r=192.168.80.50

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.vsbin
@echo.
@goto res%ERRORLEVEL%

:res4
@echo Could not connect to remote device
@goto :EOF

:res0
@echo Successful read out and saved
@goto :EOF
```

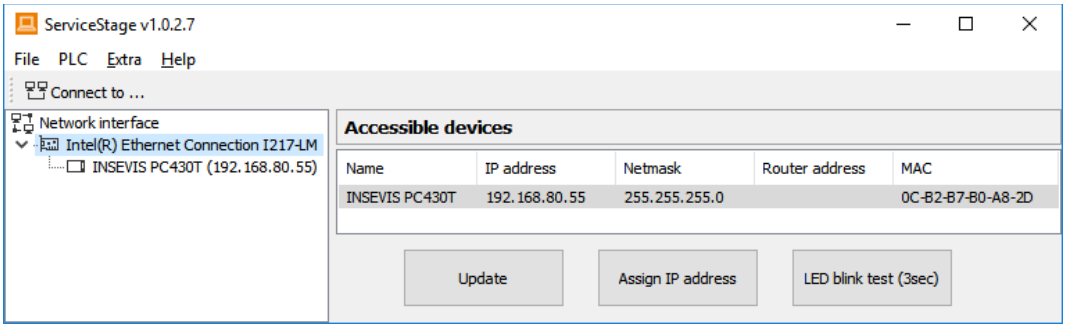
Working with the software „ServiceStage“

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. This software can be used with operating system Windows 10 only.

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

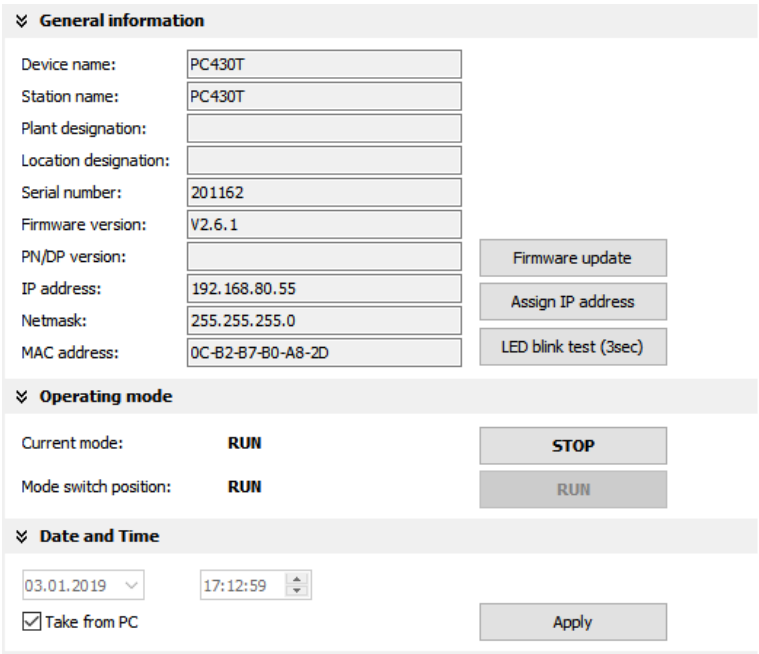
General information

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

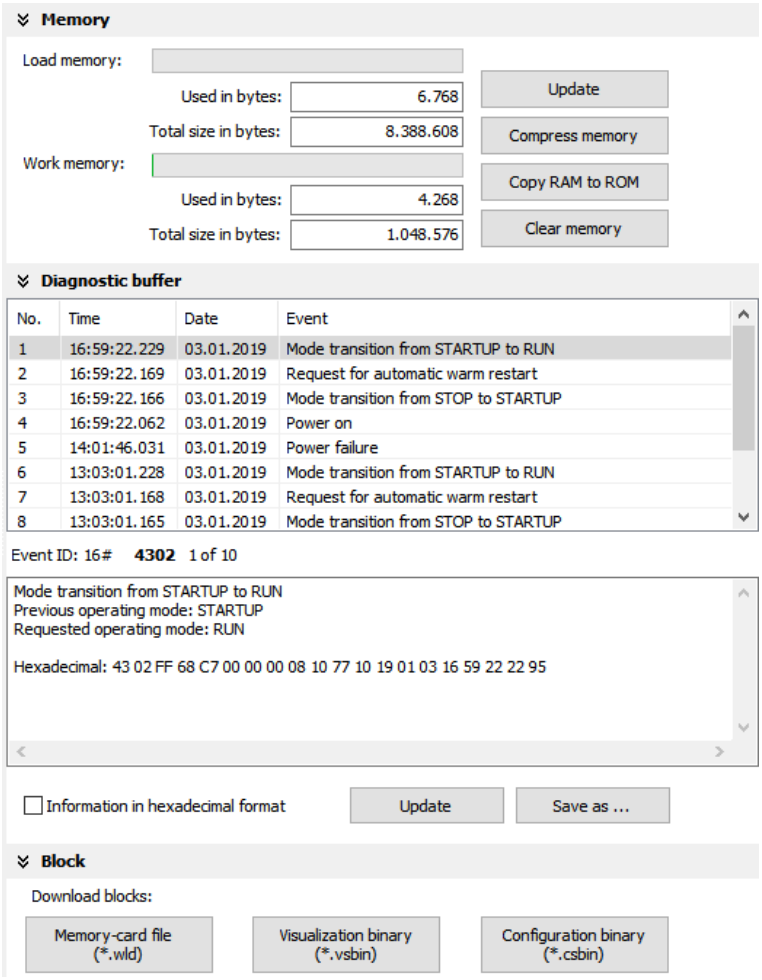
For manual adustment of system date and time.
By checking the box **„Take from PC“** these data will be taken over from your PC after pressing **„Apply“**.

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.

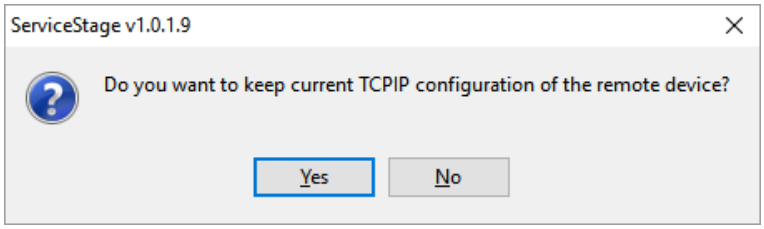
With **„Save as“** the displayed entries will be saved into a text file (*.txt).

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.



* 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 menue: „File → Memory Card File → New“) you can copy DIFFERENT DBs with or without their system dats 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.

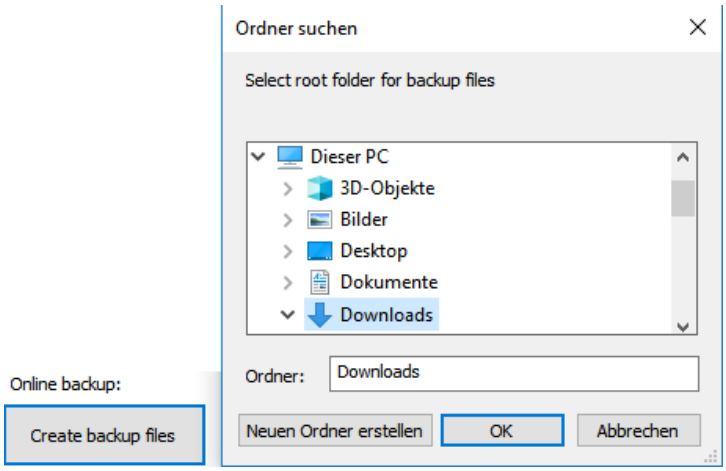
ATTENTION: When inserting system data by SimaticManagers/ TIA-Portal,
- they shall be uploaded from a PLC with included ConfigStage-configuration (Menue: „PLC → Upload Station to PG“), because the are “ConfigStage-configured” already or
- the system data of the Simatic-Manager/TIA-Portal overwrite the ConfigStage-system system configuration. In that case the *.csbin-file of the project must be downloaded again after the WLD-file was downloaded.

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“)



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.

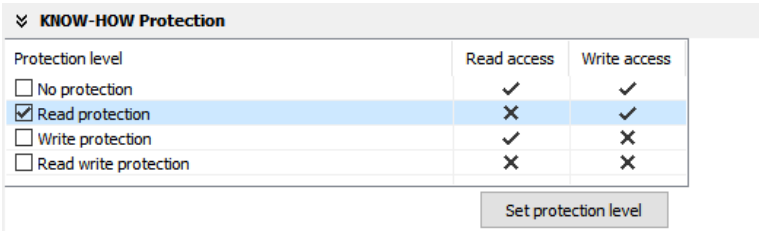
ATTENTION: When using Block Upload for backup / restore
- With firmware 2.3.6 a Restore can only be done by WLD- and VSBIN-file. The CSBIN-file will be ignored. So the WLD-file MUST CONTAIN these system data, who where downloaded with CSBIN before.
- If the WLD-file does not contain system data, program 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.

From firmware 2.7.5 (PLC CPU-T) the present protection level is displayed at the checkbox.



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 authentification by password, if it was configured so in the Simatic-Manager or TiA-Portal. So only authorized personnel can activate higher protection levels.

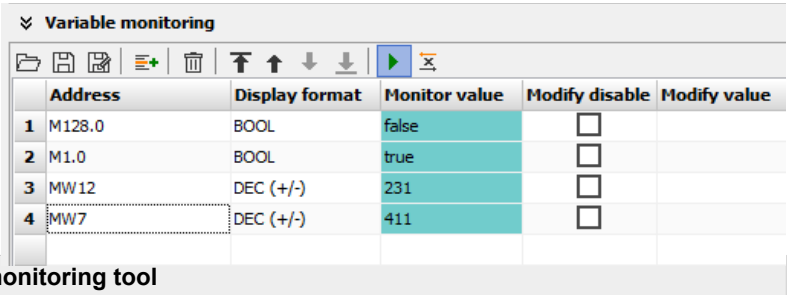
Read protection: The S7-program blocks (DBs) can still be manipulated from everyone, but not read any more. Only the Panel-HMIs can still read out the process data (DBs) to visualize it. This protection level can be reset by the PG-function „Clear All“ or by ServiceStage. Than the S7-program is deleted 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.

Variable monitoring

From ServiceStage V 1.0.3.3, the monitoring table function is available for quick monitoring of variables.
This can be used to monitor individual values.
These values can also be controlled. Variable tables (***.vartab files**) can also be imported so that large amounts of data can be quickly observed or used for quick function tests.
All functions of the observation table are described with a tooltip when the mouse pointer hovers over the button.



Note: The ServiceStage as a quick monitoring tool

In parallel to other programmes, you can monitor selected variables with the service stage. This works very efficiently and as a small window. A simplification if you need to keep an eye on data from several programmes at the same time.

INSEVIS

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 9001:2015