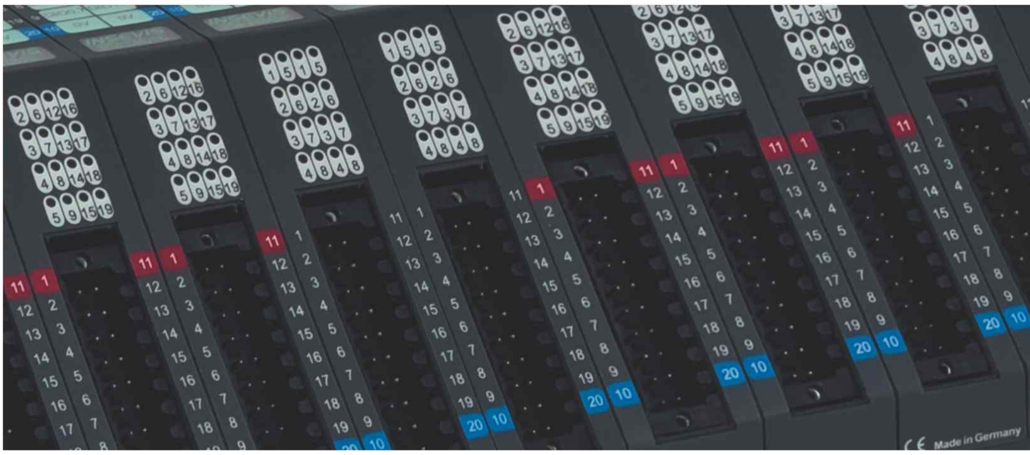
INSEVIS



# S7-Compact-PLC Manual

S7-Panel-PLC

S7-Compact-PLC

S7-Panel-HMI

Periphery

Software

Energy Management

## Index of contents

# Changes to older versions of the manual

**Rev. 02 / 2012:**
new: SFB130 (Modbus), SFC75 (Profibus DP-Slave), SFC254 (RAM to ROM)
changed: SFB125/126 (TCP)-description, SFB127/128 (UDP)-description,
Profibus-description, SFB/SFC overview

**Rev. 03 / 2012:**
changed: SFB60/61 (UART) parameter description, alarm handling, ConfigStage completely reworked

**Rev. 04 / 2012:**
changed: description of Profibus-signals made compatible to Siemens-manuals,
explanation of cycling time added (ConfigStage-CPU settings)

**Rev. 01 / 2013:**
new: Assigning an IP-address with ConfigStage 1.0.14.5 and PLC-firmware (operating system/OS) 2.0.35
changed: description of Ethernet UDP-Receive (SFB128), Profibus-Slave Subnets

**Rev. 02 / 2013:**
new: ServiceStage added

**Rev. 3 / 2013:**
new: Memory organization added, chapter CAN: more information

**Rev. 04 / 2013:**
new: New description for ConfigStage: PDO-/ SDO-Mapping, address overview list (from ConfigStage 1.0.14.7)

**Rev. 05 / 2013:**
new: New PLC-function: SFC206: read/ write recipe record into PLC with S7, SFC207: archiving DBs
(with PLC-OS 2.0.41 and HMI-OS 1.0.11)

**Rev. 06 / 2013:**
new: Service Stage 1.0.1.1: Know-how-protection available, ConfigStage 1.14.9 with EDS-Import,
SFC208 Backup&Restore (with OS-firmware 2.1.0 and S7-Lib 2_1_0)
changed: description of CAN SFB114

**Rev. 07 / 2013:**
changed: description of embedding S7-lib into S7-project moved to system functions

**Rev. 08 / 2013:**
changed: description of SFB60/61 UART improved, ConfigStage 1.0.14.10 „Change device“ function added

**Rev. 01 / 2014:**
new: from PLC-firmware 2.1.5 SFC209 READFILE, SFC2010 LOAD_DBL, SFC2011 LOAD_DBW new
changed: Backup&Restore- resume in chapter System functions added

**Rev. 02 / 2014:**
changed: S7-program lines as sample inserted for all SFCs and Ethernet-SFBs, Hints for VIDEOS added

**Rev. 03 / 2014:**
changed: Improved description of SFB 120 Get and SFB 121 Put in one connection,
Improved description of SFB 208 (Saving password file)

**Rev. 01 / 2015:**
changed: Description of TSAPs improved, are changeable now, with firmware 2.2.1 and ConfigStage 1.0.14.15

**Rev. 02 / 2015:**
changed: System functions: better overview OBs, SFBs, SFCs, explanation how to create a WLD-file
UART: Description of RS232 receive data with defined end code

**Rev. 03 / 2015:**
new: CPU-T-inserted, adaptions (extensions) for CPU-T functionality inserted,
changed: New ServiceStage functionality Firmware update by ETH, description: how to make a WLD,
(with PLC-Firmware 2.3.1, ConfigStage 1.0.14.21, VisuStage 2.0.4.7, ServiceStage 1.0.1.5, RemoteStage 1.0.4.13)

**Rev. 01 / 2016:**
new: Profinet handling added, UART-, CAN-, Ethernet chapter debugged,
changed: System functions: Description of SFC208 increased, image inserted for manual backup

**Rev. 02 / 2016:**
new: SFC 214 LCD_BRIG added in system functions
changed: description for manual backup changed

**Rev. 03 / 2016:**
new: new up- and downloads in command line mode of RemoteStage 1.0.4.17, online-backup in Service Stage 1.0.1.7
from PLC firmware 2.3.9 (all CPUs) or HMI firmware 1.2.7 CPU-V/-P and 1.2.6 CPU-T

# Changes to older versions of the manual

**Rev. 04 / 2016:**
changed: RemoteStage: Description at download recipes and data archives into remote-device improved
ServiceStage: Description at Online Backup expanded

**Rev. 05 / 2016:**
changed: Description of Hardware-reset improved, ServiceStage: Description at Online Backup expanded once more
Design changed

**Rev. 01 / 2017:**
new: SFC215 to change user level by S7-program, SFC76 to set address AND baud rate of Profibus-DP-slave
ConfigStage 1.0.14.32: Webserver for all CPU-T-PLCs, RemoteStage 1.0.4.23,
firmwares: Compact-/Panel-PLCs with CPU-V/-P 2.4.5, Compact-/Panel-PLCs with CPU-T 2.4.8

**Rev. 02 / 2017:**
new: Ethernet; info for communication assignments for S7-1200/1500 – settings in TIA-Portal explained

# 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.

⚠ Warning

## 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.

⚠ Attention

## Data security

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

## Copyright

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

## Trade Marks

INSEVIS refers that all trade marks of particular companies used in own documentation as e.g.
- STEP®, SIMATIC® and other as reserved trade mark of Siemens AG.
- CANopen® and other as reserved trade mark of CAN in Automation eG
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 new 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

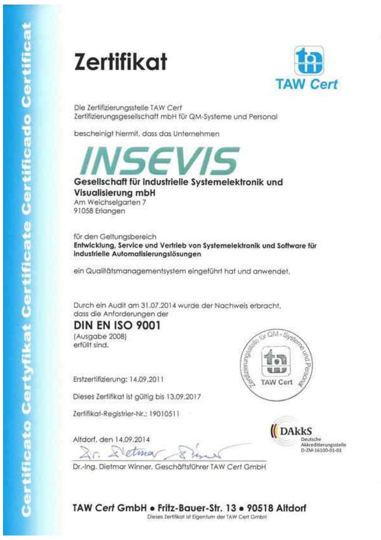
INSEVIS-products does not base on Windows or Linux, they have an own firmware. Thereby the hard- and software can be exactly designed for a perfect co-ordination with this firmware and a low power consumption. Booting times of less than 4 seconds and completely no software licenses and a current drain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

Zertifizierungsstelle für QM-Systeme und Personal TAW Cert

Erstzertifizierung: 14.09.2011

Dieses Zertifikat ist gültig bis 13.09.2017

Zertifikat-Registrier-Nr.: 19010511

DAkkS Deutsche Akkreditierungsstelle D-ZM-16100-01-01

Altdorf, den 14.09.2014
Dr. Dietmar Winner
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.

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

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

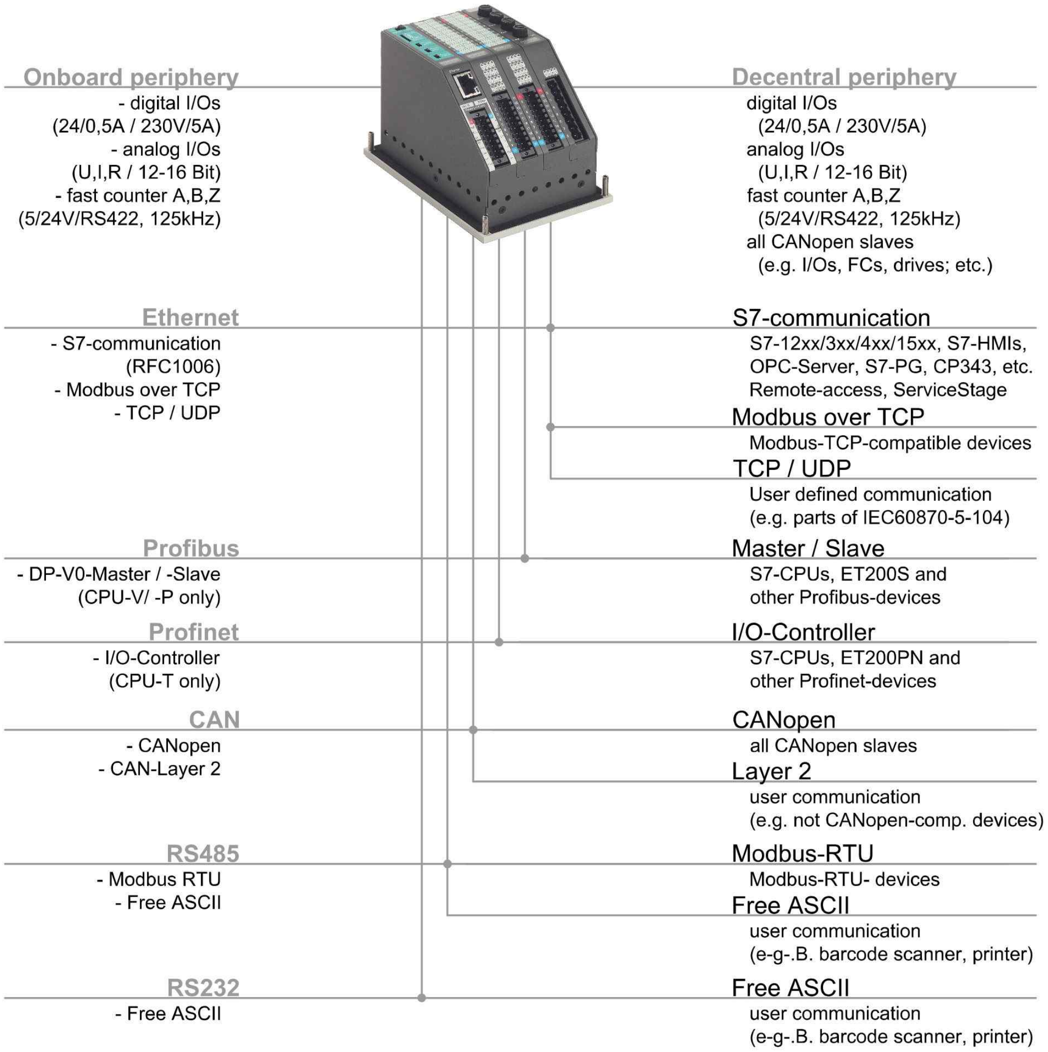
These products were put into comprehensive validation tests before they were produced in selected and certified production lines.

**INSEVIS - Made in Germany**

# Product family S7-Compact-PLC

## Communication overview

INSEVIS-S7-Compact-PLCs contain versatile possibilities for a connection to diverse 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.



Onboard periphery
- digital I/Os (24/0,5A / 230V/5A)
- analog I/Os (U,I,R / 12-16 Bit)
- fast counter A,B,Z (5/24V/RS422, 125kHz)

Decentral periphery
digital I/Os (24/0,5A / 230V/5A)
analog I/Os (U,I,R / 12-16 Bit)
fast counter A,B,Z (5/24V/RS422, 125kHz)
all CANopen slaves (e.g. I/Os, FCs, drives; etc.)

Ethernet
- S7-communication (RFC1006)
- Modbus over TCP
- TCP / UDP

S7-communication
S7-12xx/3xx/4xx/15xx, S7-HMIs, OPC-Server, S7-PG, CP343, etc. Remote-access, ServiceStage

Modbus over TCP
Modbus-TCP-compatible devices

TCP / UDP
User defined communication (e.g. parts of IEC60870-5-104)

Profibus
- DP-V0-Master / -Slave (CPU-V/ -P only)

Master / Slave
S7-CPUs, ET200S and other Profibus-devices

Profinet
- I/O-Controller (CPU-T only)

I/O-Controller
S7-CPUs, ET200PN and other Profinet-devices

CAN
- CANopen
- CAN-Layer 2

CANopen
all CANopen slaves

Layer 2
user communication (e.g. not CANopen-comp. devices)

RS485
- Modbus RTU
- Free ASCII

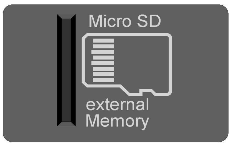
Modbus-RTU
Modbus-RTU- devices

Free ASCII
user communication (e-g-.B. barcode scanner, printer)

RS232
- Free ASCII

Free ASCII
user communication (e-g-.B. barcode scanner, printer)

## External memory card



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

## Most important properties at a glance

### S7-Programming

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

### Backup & Restore

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

### Data archiving

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

### Gateway functionality

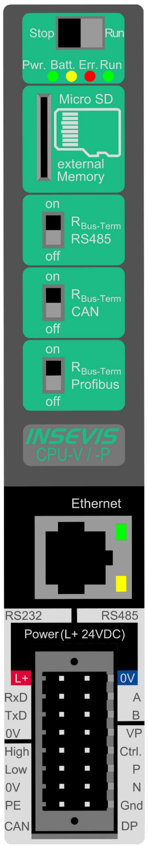
Ethernet with TCP, UDP, RFC1006 or Modbus TCP, Profinet IO Controller, CANopen® or Layer2, free ASCII on RS232 and RS485 and Modbus RTU. INSEVIS-S7-PLCs are communication talents.

# Technical data of INSEVIS-S7-CPUs

## CPU-V

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

### CPU-V



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

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

## Most important properties at a glance

### S7-Programming

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

### 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

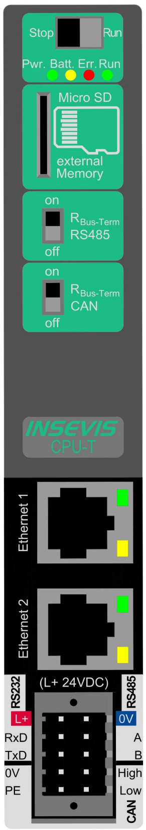
Ethernet with TCP, UDP, RFC1006 or Modbus TCP,
Profibus-DP V0 Master/Slave, CANopen® or Layer2,
free ASCII on RS232 and RS485 and Modbus RTU
INSEVIS-S7-PLC - a communication talent

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

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

## Most important properties at a glance

### S7-Programming

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

### 2 Ethernet ports

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

### Know-how-protection

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

### Gateway functionality

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

# Decentral touch panel units for the Compact-PLCs

## General

By using TFT- displays a high brilliance of 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,
- 10,2“ with WVGA- resolution (800x480pixel, 16:9-format) → for front panel use

## With CPU -T

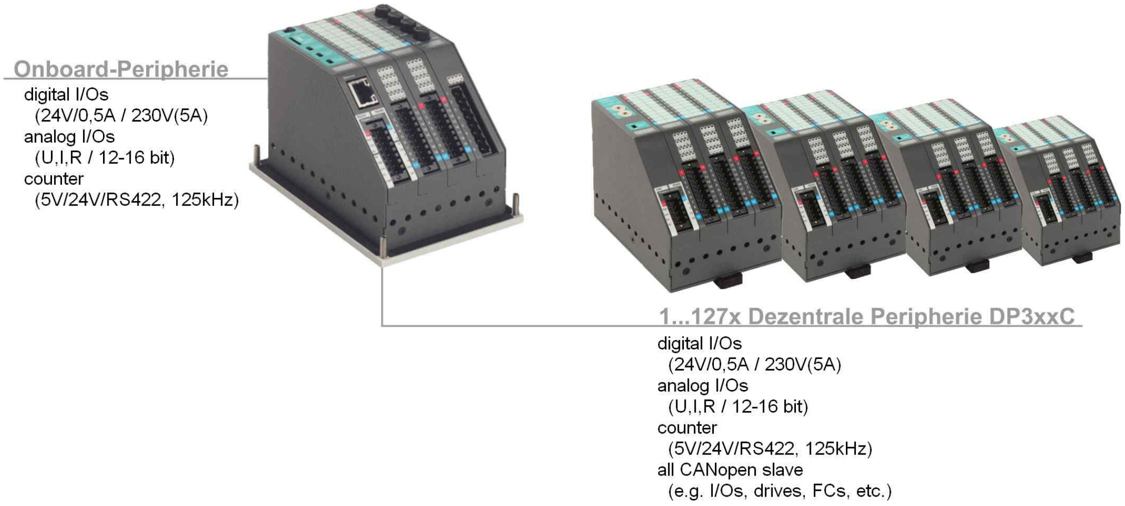
The S7-Panel-HMI family with the CPU-T consist of 4 different display diagonals:

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

# Periphery in the Compact-PLCs

## Communication to PLC

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



## Periphery modules

**Decentral head stations DP303C/DP307C/DP311C**
with 3 / 7 / 11 periphery slots
**Periphery module DI16**
(16 digital inputs 24V)
**Periphery module DIO16**
(16 digital in- or outputs 24V / 0,5A)
**Periphery module DO4-R**
(4 relay outputs 230V / 3A)
**Periphery module MIO84**
(8 digital- and 4 analog in- or outputs)
**Periphery module AI4O4**
(4 analog in- and 4 analog outputs)
**Periphery module AI8O2**
(8 analogin- and 2 analog outputs)
**Periphery module RTD8O2**
(8 analog in- and 2 analog outputs)
**Function module DIO8-Z**
(2 Encoder inputs A,B,Z and 2 digital in- or outputs)

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

Data valid for all periphery and function modules:

| Property | Technical data |
|---|---|
| Operating temperature range<br>Storing temperature range | -20°C ... +60°C<br>(without condensation)<br>-30°C ... +80°C |
| Dimension (W x H x D)<br>Weight | 20 x 108 x 70 mm<br>ca. 150 g |
| Wire length<br>unshielded (max.)<br>shielded (max.) | 30 m<br>100 m |
| Connection technology | Cage clamp technology for cross section up to max. 1,5mm$^2$ |

**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 in the Compact-PLCs

## Configuration

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

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

## Remote access

With the free-of-charge software „RemoteStage“ data archives can be read in from the Micro-SD-card in the Compact-PLC. This program is a portable solution but with lots of functions. It communicates via TCP/IP with the Compact-PLC. These archive data can be converted into csv-format and stored anywhere. With an SFC213 READ_CSV modified csv-files can be read bach to the Compact-PLC, what offerst lots of solutions.
This procedure is also possible as command line in any batch process. Multiple RemoteStages can be operated in one PC to get a kind of master display with multiple remote screens.

## Service tool

The „ServiceStage“ is made to have easy service access to the PLCs. It allows to identify a S7-CPU by its IP-address and to analyze it by diagnostic buffer, to update the user-programs and to set the protection levels. This software is free of licenses, offers lots of service functions easily to be found even if you are not working every day with this software tool. Installed in a minute, useful service functions, absolutely easy to understand and to use. These functions can be solved with the „ServiceStage“

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

# Accessories for S7-Compact-PLCs

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

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

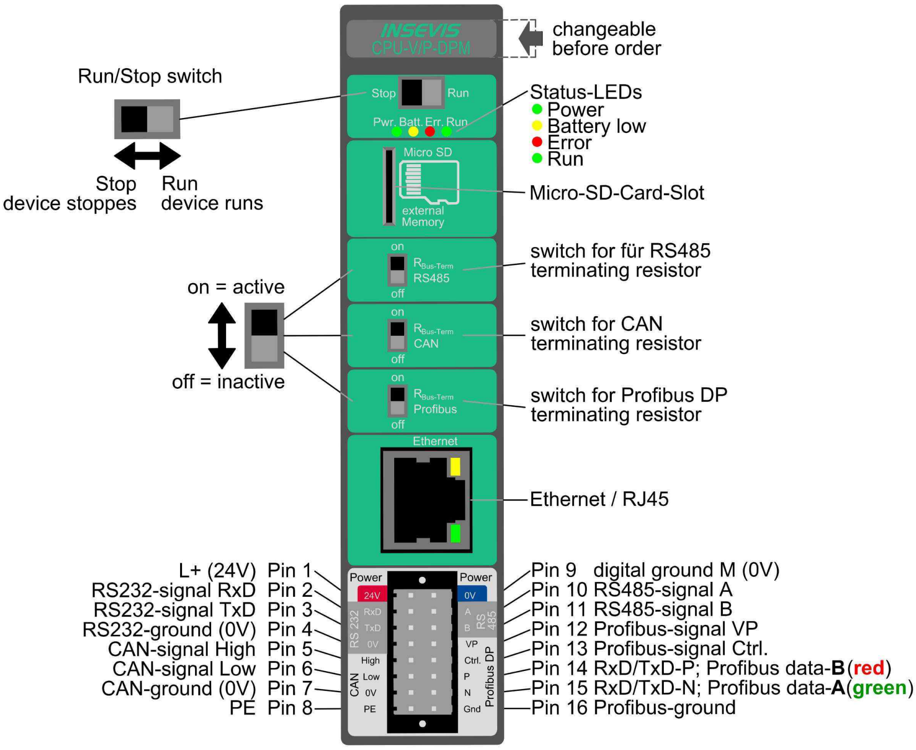
| Figure of accessories | accessories | Article-no. | MOQ (pcs) |
|---|---|---|---|
| **Connectors**<br> <br>E-CON16<br> <br>E-CONS16<br>(pin marked connectors for max. 1,5mm² cross sections) | **For CPU V:**<br>Connector 2x8pin, lift arms<br>Connector 2x8pin, bolt flanges<br>Adapter for Profibus SUB-D 9 | <br>E-CON16-00<br>E-CONS16-00<br>E-AD-DP12-00 | <br>1<br>1<br>1 |
| <br>E-CON10<br> <br>E-CONS10<br>(pin marked connectors for max. 1,5mm² cross sections) | **For CPU T:**<br>Connector 2x5pin, lift arms<br>Connector 2x5pin,  bolt flanges | <br>E-CON10-00<br>E-CONS10-00 | <br>1<br>1 |
| **External memory**<br> <br>**Hint:**<br>S7-program runs without these cards, these parts are used for archiving / recipes / updating only | Micro SD-card 1GB (external memory)<br>Micro SD-card 2GB (external memory)<br>Micro SD-card 4GB (external memory)<br>Micro SD-card 8GB (external memory) | E-MSD1-00<br>E-MSD2-00<br>E-MSD4-00<br>E-MSD8-00 | 1<br>1<br>1<br>1 |
| **Customized labeling**<br>**Hint:**<br>[1] Single fix costs only, no run-time costs per single PLC | 3D-doming labels with customer logo (front)<br>OEM-firmware with integrated customer logo<br>Inserting stripes H with customer logo (rear)<br>Inserting stripes V with customer logo (rear) | E-LAB3D-00<br>SW-BS-OEM [1]<br>E-LABH-00<br>E-LABV-00 | 100<br>1<br>100<br>100 |
| **Software**<br>**Hint:**<br>required, otherwise only old versions do run<br>Software available for free download at INSEVIS web sites | ConfigStage<br>RemoteStage<br>ServiceStage | -<br>-<br>- | free download<br>free download<br>free download |
| **Spare parts**<br>**Hint:**<br>A mounting set with grounding terminal is part of every delivery | Additional grounding terminal set<br>(with 10pcs of it) | E-MNT-00 | 10 |

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

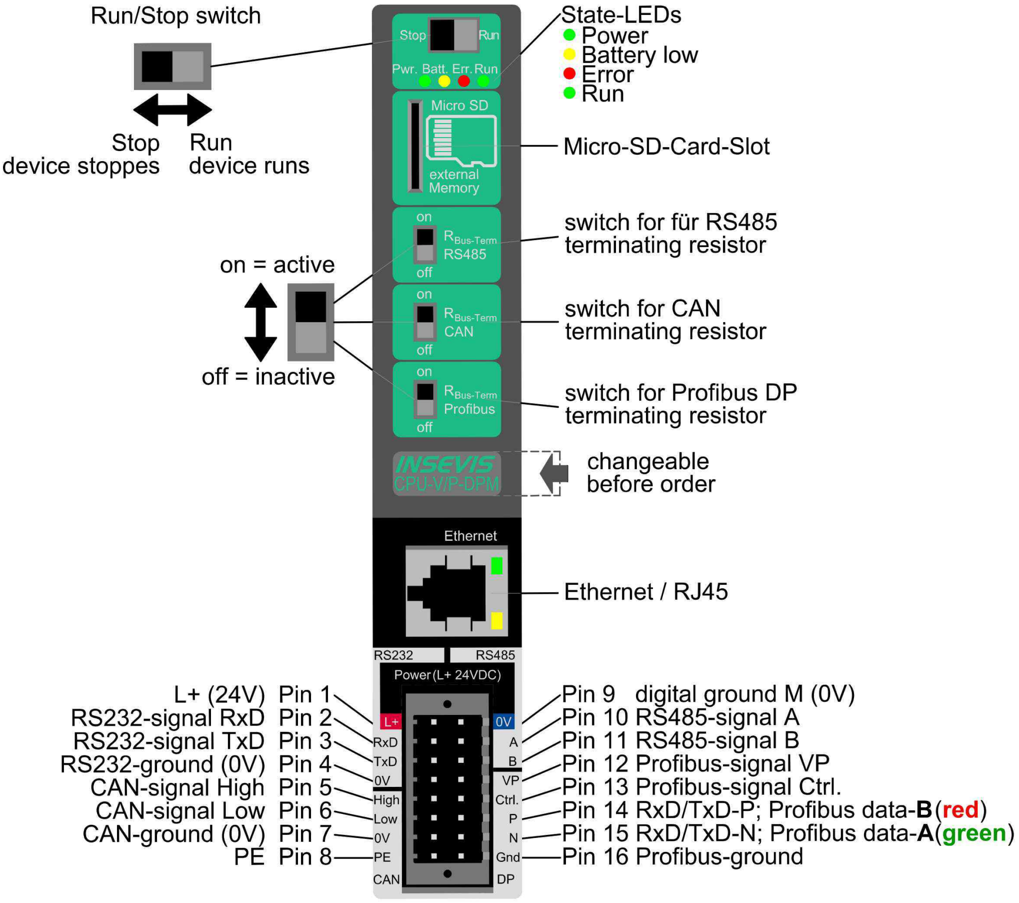
# Wiring of the S7-Compact-PLCs

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

*Wiring in basic devices (without periphery slots) CC300V*



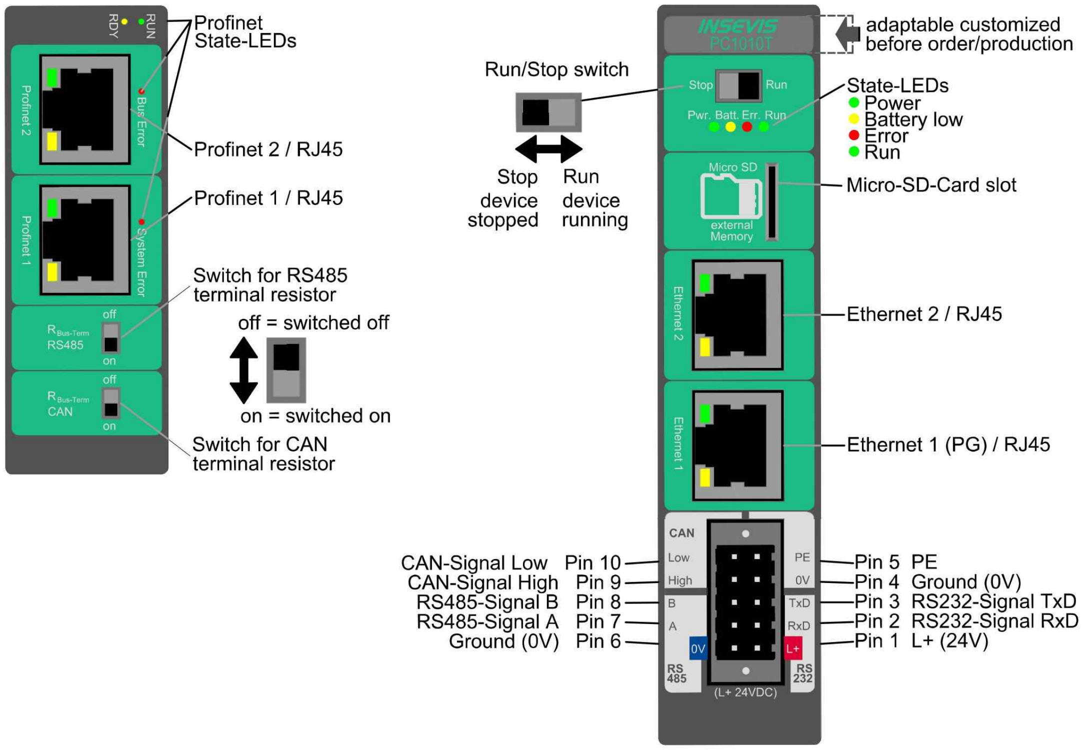
*Wiring in devices with periphery slots CC303V CC307V CC311V*



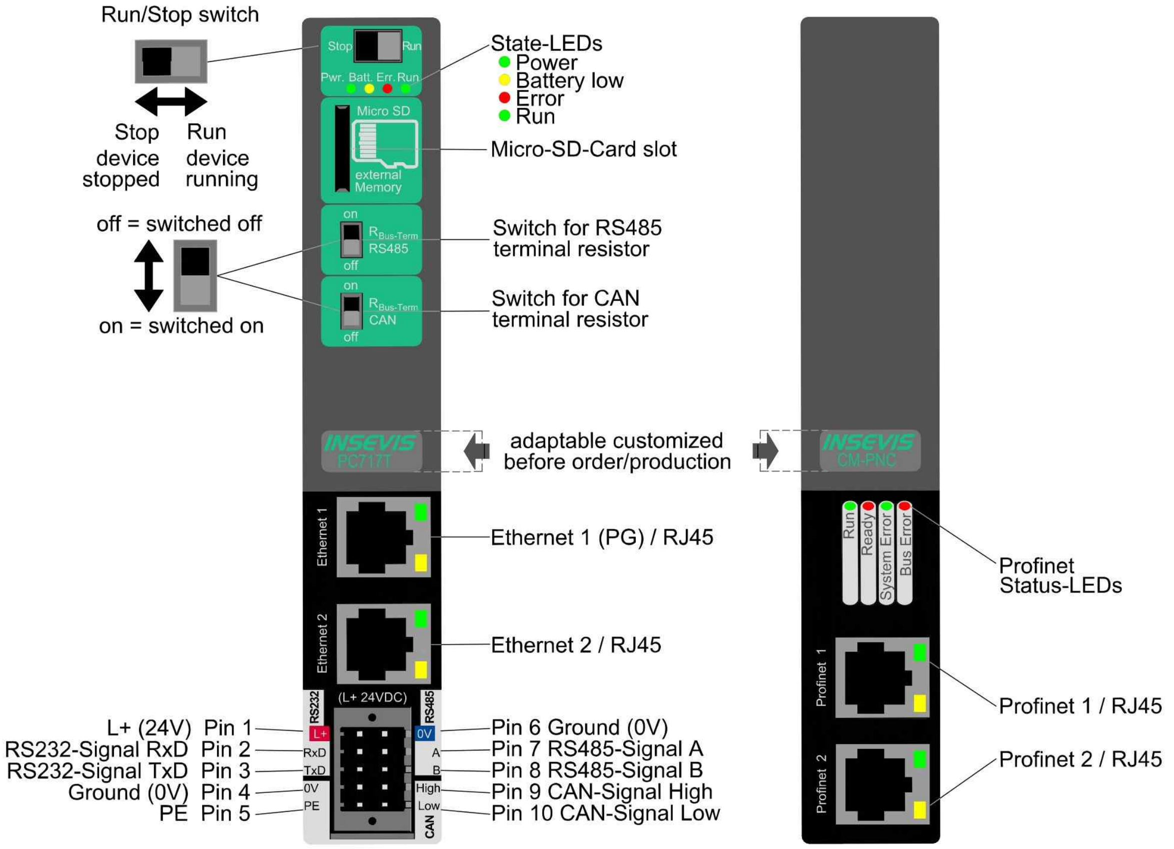
# Wiring of the S7-Compact-PLCs

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

*Wiring in basic devices (without periphery slots) CC300T*



*Wiring in devices with periphery slots CC303T CC307T CC311T*

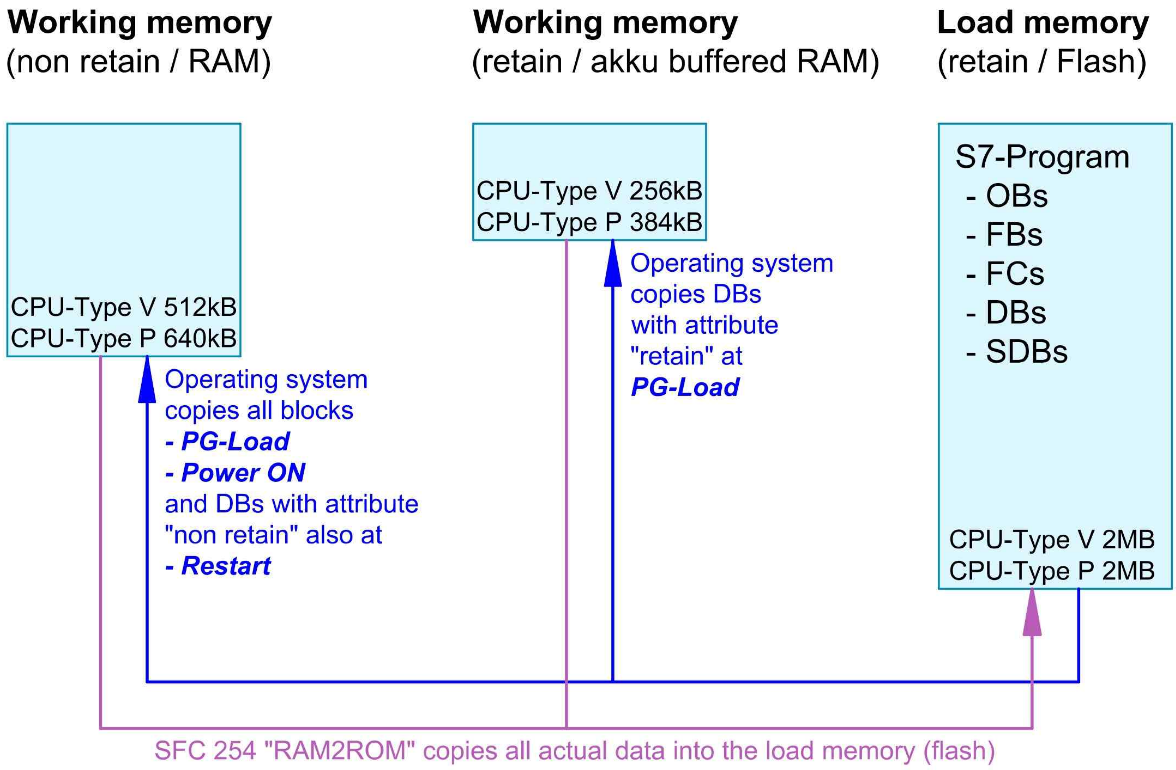


## Memory organization

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 changes into RUN-state.



# Communication

## Ethernet

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

> The INSEVIS- PLCs will be deliverd with default **IP-address 192.168.80.50.**
>
> Change this IP-adress of your PLC, before you set up your communication.
>
> Attention

## Set up IP-address in the SIMATIC® -Manager

**Essential equipment**

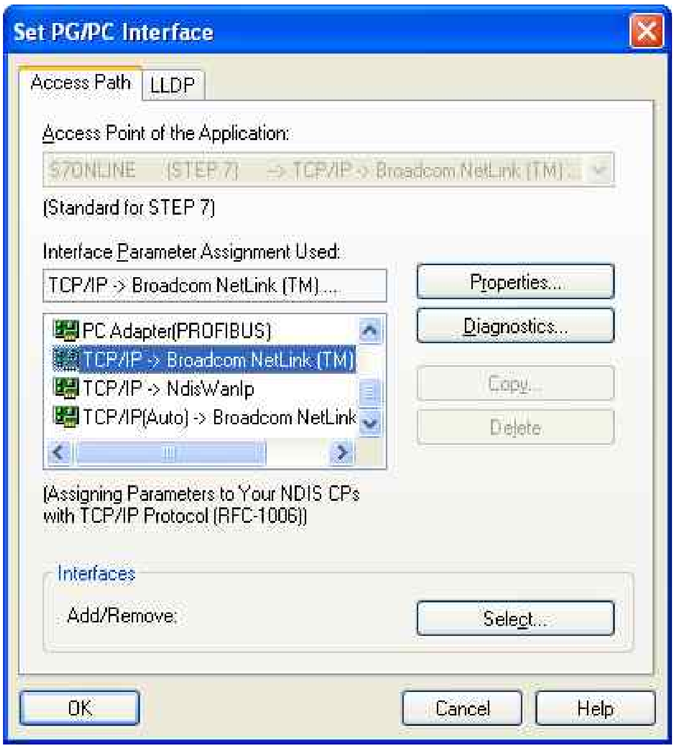
PC with SIMATIC®-Manager from
Version 5.4 SP5 (PG)
INSEVIS-PLC (AG)
Power supply 24V
Patch cable or crossover patch cable

**Procedure**

connect PG with AG
AG Power on
Start SIMATIC® Manager

**Set up PG/PC-interface**

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



Set PG/PC Interface

Access Path | LLDP

Access Point of the Application:
S7ONLINE (STEP 7) --> TCP/IP -> Broadcom NetLink (TM)
(Standard for STEP 7)

Interface Parameter Assignment Used:
TCP/IP -> Broadcom NetLink (TM)....

PC Adapter(PROFIBUS)
TCP/IP -> Broadcom NetLink (TM)
TCP/IP -> NdisWanIp
TCP/IP(Auto) -> Broadcom NetLink

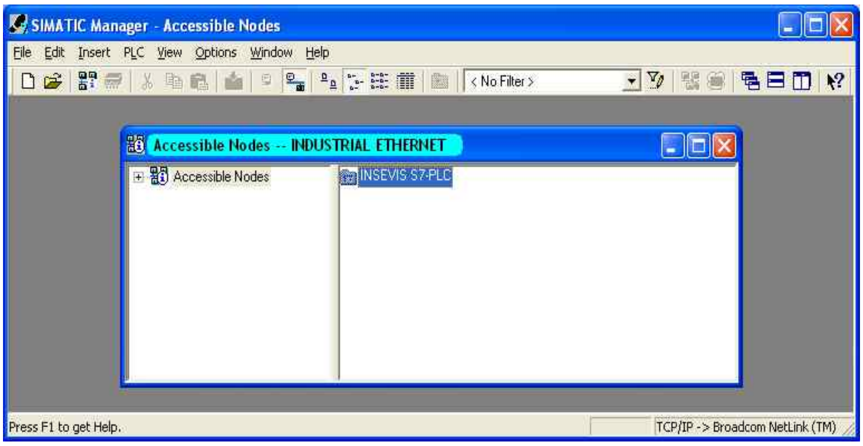
(Assigning Parameters to Your NDIS CPs with TCP/IP Protocol (RFC-1006))

Interfaces
Add/Remove:

**Address the PLC**

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

**Note:** There is a VIDEO available for that item at the INSEVIS-service web site.



SIMATIC Manager - Accessible Nodes

Accessible Nodes -- INDUSTRIAL ETHERNET

Accessible Nodes
INSEVIS S7-PLC

Press F1 to get Help.
TCP/IP -> Broadcom NetLink (TM)

# 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**
Bevor using the new IP-address, it mus be searched & found again:
→ PLC
→ Display Accessible Nodes
OR:
→ referesh 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.

**Note:** There is a VIDEO available for that item at the INSEVIS-service web site

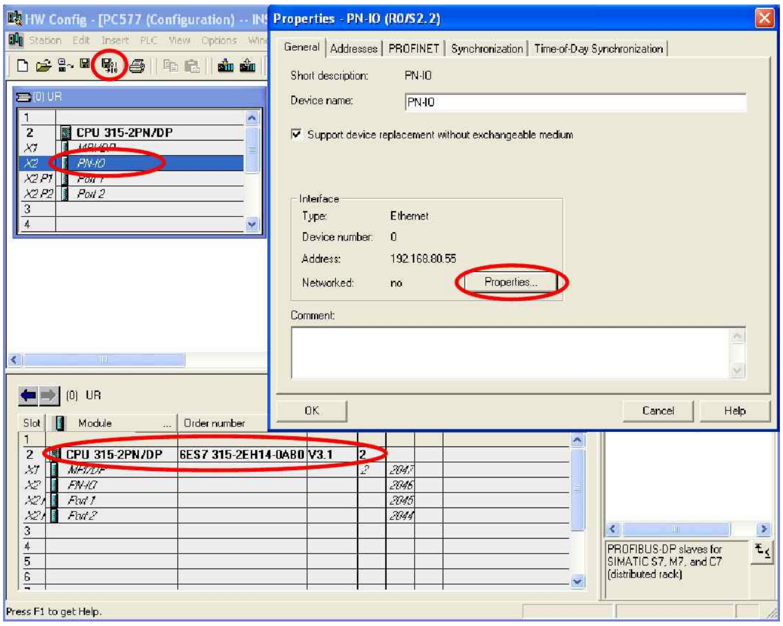
## Set up the CPU and the Ethernet connection in the SIMATIC®-Manager

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.*

Selection of the Ethernet communication in the SIMATIC- Manager from Siemens:

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



HW Config - [PC577 (Configuration) -- IN

| | |
|---|---|
| 1 | |
| 2 | CPU 315-2PN/DP |
| X1 | MPI/DP |
| X2 | PN-IO |
| X2 P1 | Port 1 |
| X2 P2 | Port 2 |
| 3 | |
| 4 | |

Properties - PN-IO (R0/S2.2)

Short description: PN-IO
Device name: PN-IO
☑ Support device replacement without exchangeable medium

Interface
Type: Ethernet
Device number: 0
Address: 192.168.80.55
Networked: no

Comment:

(0) UR

| Slot | Module ... | Order number | | | |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | CPU 315-2PN/DP | 6ES7 315-2EH14-0AB0 | V3.1 | 2 | |
| X1 | MPI/DP | | | 2 | 2047 |
| X2 | PN-IO | | | | 2046 |
| X2 P1 | Port 1 | | | | 2045 |
| X2 P2 | Port 2 | | | | 2044 |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |

Press F1 to get Help.

**Note:** There is a VIDEO available for that item at the INSEVIS-download web site

**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.

| 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

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.

⚠ Attention

| 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<br>W#16#.. | Description |
|---|---|---|---|
| 1 | 0 | 0 | Job has successfully completed. |
| 0 | 0 | 7000 | Job has not been accepted, because:<br>• this is the first call with REQ=0 or because<br>• 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 Put and Get in one connection**

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

If you call the SFB 121 „Put" and SFB 120 „Get" in one connection, the SFB 120 „Get" 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.

⚠ Attention

| 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:<br>• this is the first call with REQ=0 or because<br>• 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
```

## 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”.

| | |
|---|---|
| **Connection with passive establishment: the local port number must be assigned and unique.** | ⚠ Attention |

## 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.<br>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.<br>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<br>e.g: port $2000_{\text{dec}}$, enter as W#16#7D0 |
| IPADDR | INPUT | DWORD | Partner IP Address<br>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<br>e.g: port 2000$_{dec}$, output as W#16#7D0 |
| IPADDR | OUTPUT | DWORD | Partner IP Address<br>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.<br><br>Bit 0-3: Send job status<br>Bit 0: 1=Connection accepts Send job<br>Bit 1: 1=Send job is in Process<br>Bit 2: 1=Send job has completed successfully<br>Bit 3: 1=Send job has failed<br><br>Bit 4-7: Receive job status<br>Bit 4: 1=Connection accepts Receive job<br>Bit 5: 1=Receive job is in Process<br>Bit 6: 1=Receive job has completed successfully<br>Bit 7: 1=Receive job has failed<br><br>Bit 8-12: Connection status<br>1 Active establishment is in process<br>2 Connection terminating<br>3 Connection established/Connected with Partner<br><br>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.

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

⚠ Attention

## 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<br>e.g: port $2000_{dec}$, enter as W#16#7D0 |
| R_ADDR | INPUT | DWORD | Partner IP Address<br>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.<br>Only Input image (I), Output image (Q), Bit Memory (M), Data Block (DB) area allowed. |

DONE, BUSY, ERROR and STATUS parameters meaning

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

## S7-program sample for using the SFB 127

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

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

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

## Receiving data via UDP with SFB 128 "URECV"

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

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

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

…

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

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

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

The parameters ID, LEN, R_PORT, R_ADDR, DATA are evaluated and starts the receiving on EN_R = TRUE.
The new job can only be accepted again after the last job is completed and EN_R = TRUE.
Job completion is signaled by NDR or ERROR parameter.

# Communication - Ethernet

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

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

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

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

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

Attention

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

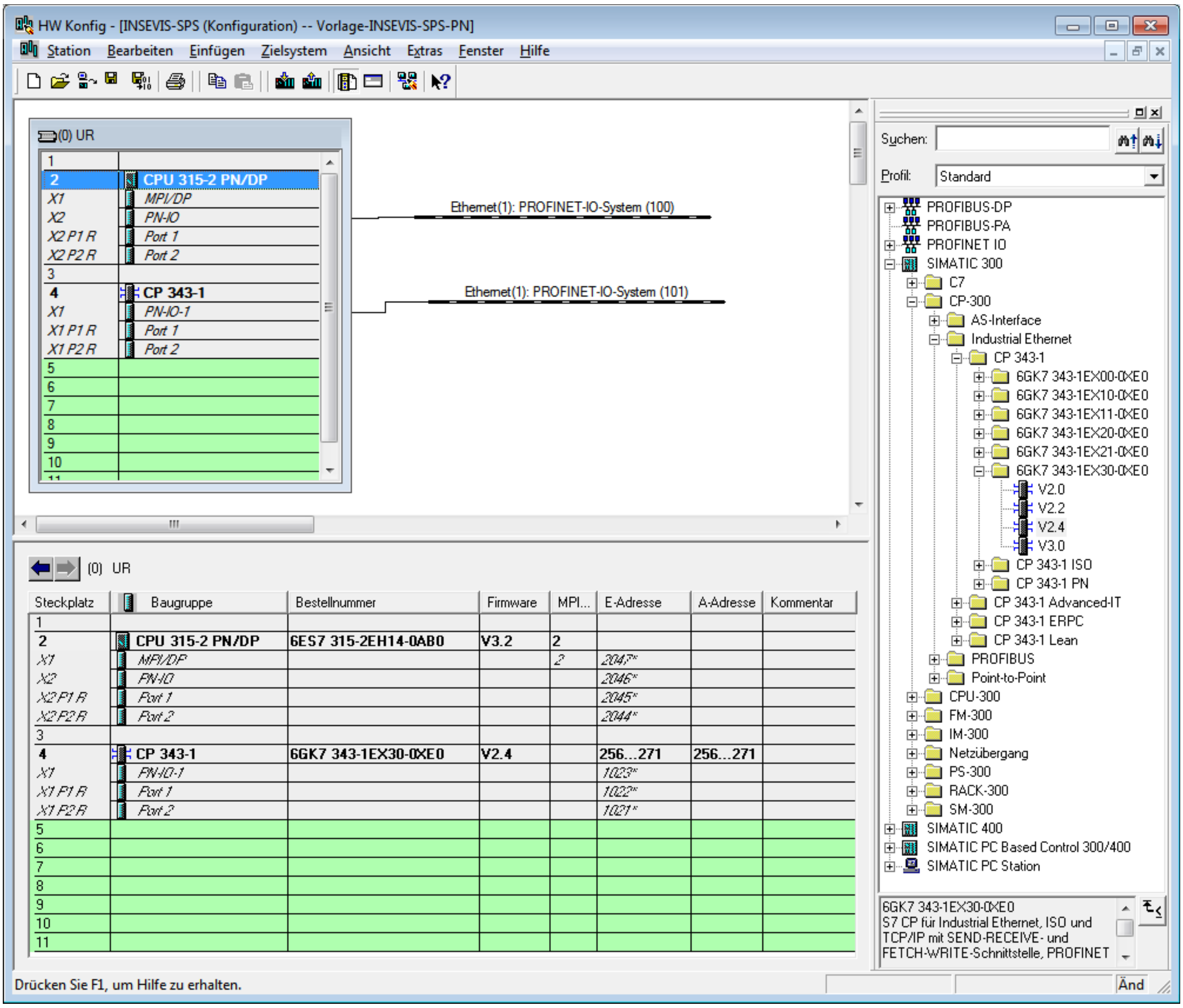
### Connecting

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

## Configuration

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

The configuration of the Profinet in INSEVIS-projects is simple:
- Select CPU 315-2 PN/DP (Siemens-order-no. 6ES7 315-2EH14-0AB0) and place it on slot 2
- Select  CP 343-1 (Siemens-order-no. 6GK7 343-1EX30-0XE0 with Version V2.4) and place it on slot 4.


HW Konfig - [INSEVIS-SPS (Konfiguration) -- Vorlage-INSEVIS-SPS-PN]
Station
Bearbeiten
Einfügen
Zielsystem
Ansicht
Extras
Fenster
Hilfe
Ethernet(1): PROFINET-IO-System (100)
Ethernet(1): PROFINET-IO-System (101)
Suchen:
Profil:
Standard

(0) UR

| | |
|---|---|
| 1 | |
| 2 | CPU 315-2 PN/DP |
| X1 | MPI/DP |
| X2 | PN-IO |
| X2 P1 R | Port 1 |
| X2 P2 R | Port 2 |
| 3 | |
| 4 | CP 343-1 |
| X1 | PN-IO-1 |
| X1 P1 R | Port 1 |
| X1 P2 R | Port 2 |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |

(0) UR

| Steckplatz | Baugruppe | Bestellnummer | Firmware | MPI... | E-Adresse | A-Adresse | Kommentar |
|---|---|---|---|---|---|---|---|
| 1 | | | | | | | |
| 2 | CPU 315-2 PN/DP | 6ES7 315-2EH14-0AB0 | V3.2 | 2 | | | |
| X1 | MPI/DP | | | 2 | 2047* | | |
| X2 | PN-IO | | | | 2046* | | |
| X2 P1 R | Port 1 | | | | 2045* | | |
| X2 P2 R | Port 2 | | | | 2044* | | |
| 3 | | | | | | | |
| 4 | CP 343-1 | 6GK7 343-1EX30-0XE0 | V2.4 | | 256...271 | 256...271 | |
| X1 | PN-IO-1 | | | | 1023* | | |
| X1 P1 R | Port 1 | | | | 1022* | | |
| X1 P2 R | Port 2 | | | | 1021* | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |

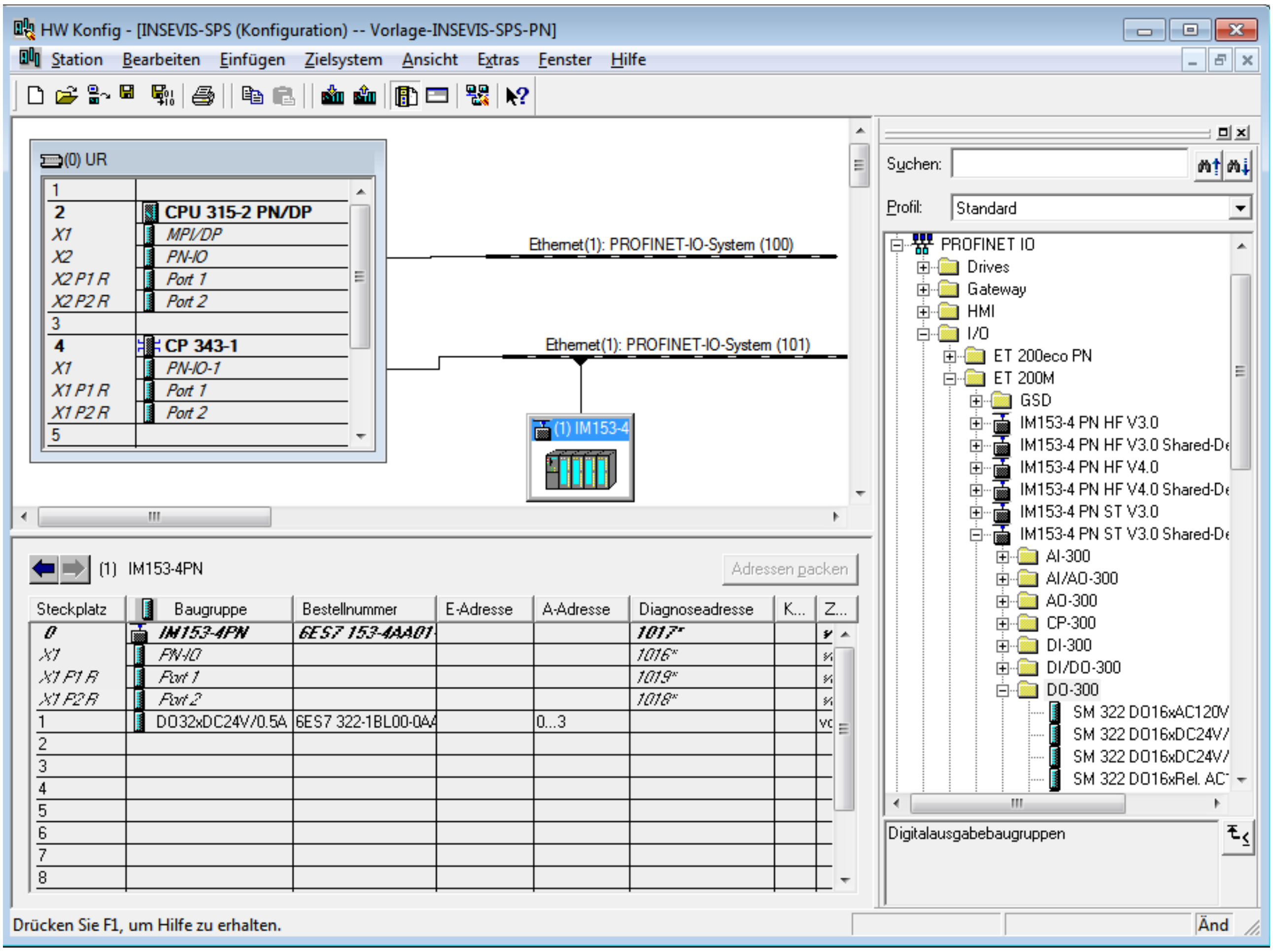
- PROFIBUS-DP
- PROFIBUS-PA
- PROFINET IO
- SIMATIC 300
  - C7
  - CP-300
    - AS-Interface
    - Industrial Ethernet
      - CP 343-1
        - 6GK7 343-1EX00-0XE0
        - 6GK7 343-1EX10-0XE0
        - 6GK7 343-1EX11-0XE0
        - 6GK7 343-1EX20-0XE0
        - 6GK7 343-1EX21-0XE0
        - 6GK7 343-1EX30-0XE0
          - V2.0
          - V2.2
          - V2.4
          - V3.0
        - CP 343-1 ISO
        - CP 343-1 PN
      - CP 343-1 Advanced-IT
      - CP 343-1 ERPC
      - CP 343-1 Lean
    - PROFIBUS
    - Point-to-Point
  - CPU-300
  - FM-300
  - IM-300
  - Netzübergang
  - PS-300
  - RACK-300
  - SM-300
- SIMATIC 400
- SIMATIC PC Based Control 300/400
- SIMATIC PC Station

6GK7 343-1EX30-0XE0
S7 CP für Industrial Ethernet, ISO und TCP/IP mit SEND-RECEIVE- und FETCH-WRITE-Schnittstelle, PROFINET

Drücken Sie F1, um Hilfe zu erhalten.

# 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.


HW Konfig - [INSEVIS-SPS (Konfiguration) -- Vorlage-INSEVIS-SPS-PN]
Station Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe
(0) UR
Ethernet(1): PROFINET-IO-System (100)
Ethernet(1): PROFINET-IO-System (101)
(1) IM153-4
Suchen:
Profil: Standard
PROFINET IO
Drives
Gateway
HMI
I/O
ET 200eco PN
ET 200M
GSD
IM153-4 PN HF V3.0
IM153-4 PN HF V3.0 Shared-De
IM153-4 PN HF V4.0
IM153-4 PN HF V4.0 Shared-De
IM153-4 PN ST V3.0
IM153-4 PN ST V3.0 Shared-De
AI-300
AI/AO-300
AO-300
CP-300
DI-300
DI/DO-300
DO-300
SM 322 DO16xAC120V
SM 322 DO16xDC24V/
SM 322 DO16xDC24V/
SM 322 DO16xRel. AC
Digitalausgabebaugruppen
(1) IM153-4PN
Adressen packen
Drücken Sie F1, um Hilfe zu erhalten.
Änd

| | |
|---|---|
| 1 | |
| 2 | CPU 315-2 PN/DP |
| X1 | MPI/DP |
| X2 | PN-IO |
| X2 P1 R | Port 1 |
| X2 P2 R | Port 2 |
| 3 | |
| 4 | CP 343-1 |
| X1 | PN-IO-1 |
| X1 P1 R | Port 1 |
| X1 P2 R | Port 2 |
| 5 | |

| Steckplatz | Baugruppe | Bestellnummer | E-Adresse | A-Adresse | Diagnoseadresse | K... | Z... |
|---|---|---|---|---|---|---|---|
| 0 | IM153-4PN | 6ES7 153-4AA01- | | | 1017* | | v |
| X1 | PN-IO | | | | 1016* | | v |
| X1 P1 R | Port 1 | | | | 1019* | | v |
| X1 P2 R | Port 2 | | | | 1018* | | v |
| 1 | DO32xDC24V/0.5A | 6ES7 322-1BL00-0A | | 0...3 | | | vc |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |

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

This means in detail, that:

- Is is not neccessary to call a function for process data exchange between CPU and CP.
- The alarm-OBs will be called by CPU by itself.

## 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.

## 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":<br>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<br>0 = RS232, 1 = RS485 |
| DONE | OUTPUT | BOOL | 4.0 | FALSE: inactive or sending is still executed.<br>TRUE: task is executed faultless<br>The parameter is only set for one call. |
| ERROR | OUTPUT | BOOL | 4.1 | Status parameter<br>The parameter is only set for one call. |
| STATUS | OUTPUT | WORD | 6 | 0303hex = ANY- pointer to data area not valid<br>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<br>0 = RS232, 1 = RS485 |
| NDR | OUTPUT | BOOL | 4.0 | FALSE: Job has not been started or is still being executed<br>TRUE: Job was completed successfully, data accepted<br>The parameter is only set for one call. |
| ERROR | OUTPUT | BOOL | 4.1 | Job completed with error.<br>The parameter is only set for one call. |
| STATUS | OUTPUT | WORD | 6 | 0303hex = ANY- pointer to data area not valid<br>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<br>(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.<br>(not conform to Modbus timing rules)<br>TRUE: task is executed faultless<br>The parameter is only set for one call. |
| ERROR | OUTPUT | BOOL | 4.1 | „Error“ failure state in STATUS<br>The parameter is only set for one call. |
| STATUS | OUTPUT | WORD | 6 | 0303hex = ANY- pointer to data area not valid<br>051Dhex = reset by input R |
| SD_1 | INOUT | ANY | 8 | ANY- pointer in data area of send data<br>- UID (1 byte target address)<br>- PDU: Modbus-command (1 Byte)<br>... Modbus-address, number ...<br>(specific for each commando)<br>-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<br>(Not necessary because of compatibility reasons) |
| LADDR | INPUT | WORD | 2 | UART-index<br>only 1 = RS485 |
| NDR | OUTPUT | BOOL | 4.0 | TRUE: task is executed faultless, Data assumed<br>FALSE = still receiving |
| ERROR | OUTPUT | BOOL | 4.1 | „Error“ failure state in STATUS<br>The parameter is only set for one call. |
| STATUS | OUTPUT | WORD | 6 | 0303hex = ANY- pointer to data area not valid<br>051Dhex = reset by input R<br>0000hex = ok<br>0001hex = blocked by an other job<br>0002hex = nothing received<br>0003hex = received < 2 bytes<br>0004hex = CRC Error, data ignored<br>0005hex = Parity error<br>0006hex = receiving in progress |
| RD_1 | INOUT | ANY | 8 | ANY- pointer in data area of receive data<br>- UID (1 byte target address)<br>- PDU: Modbus-command (1 Byte)<br>... Modbus-address, number ...<br>(specific for each commando)<br>-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.

## 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,<br>FALSE = Query the UART-configuration, |
| UART | INPUT | WORD | 2 | 0 = RS232, 1 = RS485 |
| DONE | OUTPUT | BOOL | 4.0 | TRUE = configured successfully<br>FALSE = configuration failed |
| ERROR | OUTPUT | BOOL | 4.1 | TRUE = configuration failed |
| STATUS | OUTPUT | WORD | 6 | Statuscode:<br>10hex = Index UART not valid<br>20hex = Baud rate BAUD not valid<br>30hex = Data format DATA_FMRT not valid<br>40hex = Modbus-Server Unit-Identifier UID not valid |
| BAUD | INOUT | DINT | 8 | Baud rate in bit/s, e.g. "9600"<br>Value must be > 0 |
| DATA_FMRT | INOUT | BYTE | 12 | 0 = 8 data bit, no parity<br>1 = 8 data bit, even parity<br>2 = 8 data bit, odd parity |
| UID | INOUT | BYTE | 13 | 0 = configuration ASCII or Modbus-RTU client<br>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).

## Profibus DP

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

Attention

**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 colour code: **red** |
| N | Signal RxD/TxD−N; PROFIBUS data line A; Siemens colour code: **green** |
| GND | potential seperated 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 spefic deviations at OB 86 and SFC 11 are described in Chapter „System functions“.

**Note:** There is a VIDEO available for that item at the INSEVIS-service web site.

## 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

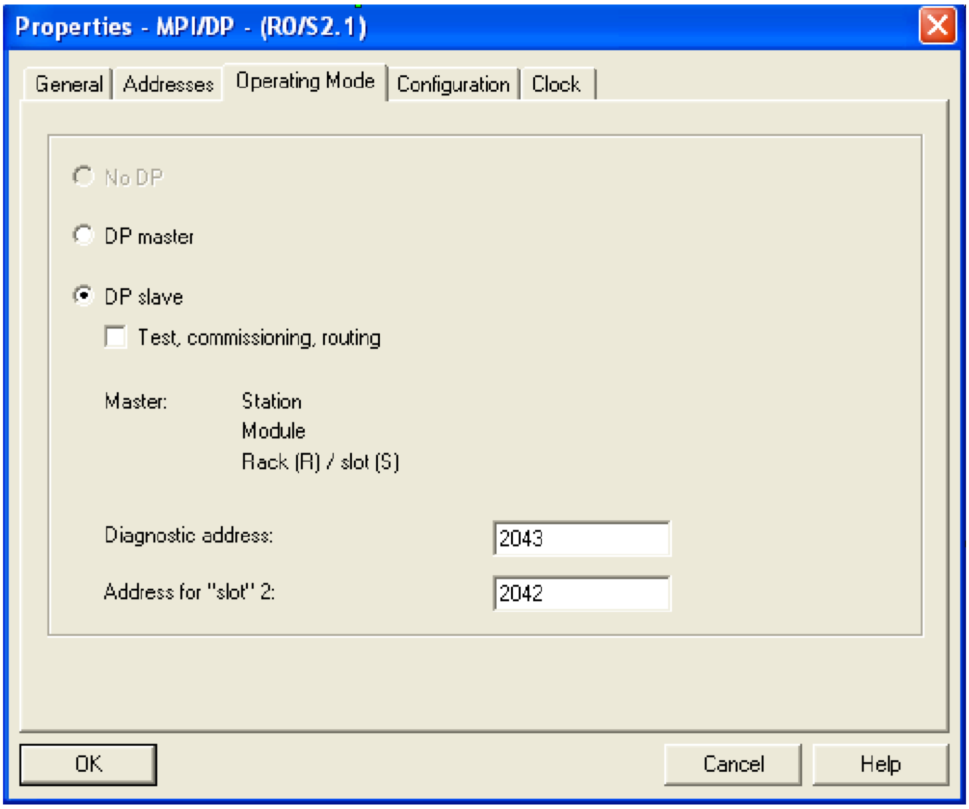
The Hardware configurator of Simatic is used to configure the DP-slave as same as configuring the integrated DP interface of CPU315-2PN/DP as DP-slave.

- Set integrated MPI/DP interface type as **PROFIBUS** under Properties->”General” tab
- Set DP-Slave-Adress under „Properties Profibus interface“ tab
  (opens automaticelly, 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

# Communication - Profibus DP-Slave

**Note:** There is a VIDEO available for that item at the INSEVIS-service web site.

- Set the interface as "DP-slave" under Properties->"Operating Mode" tab



Properties - MPI/DP - (R0/S2.1)

○ No DP

○ DP master

● DP slave

□ Test, commissioning, routing

Master: Station
Module
Rack (R) / slot (S)

Diagnostic address: 2043

Address for "slot" 2: 2042

**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)

# 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.

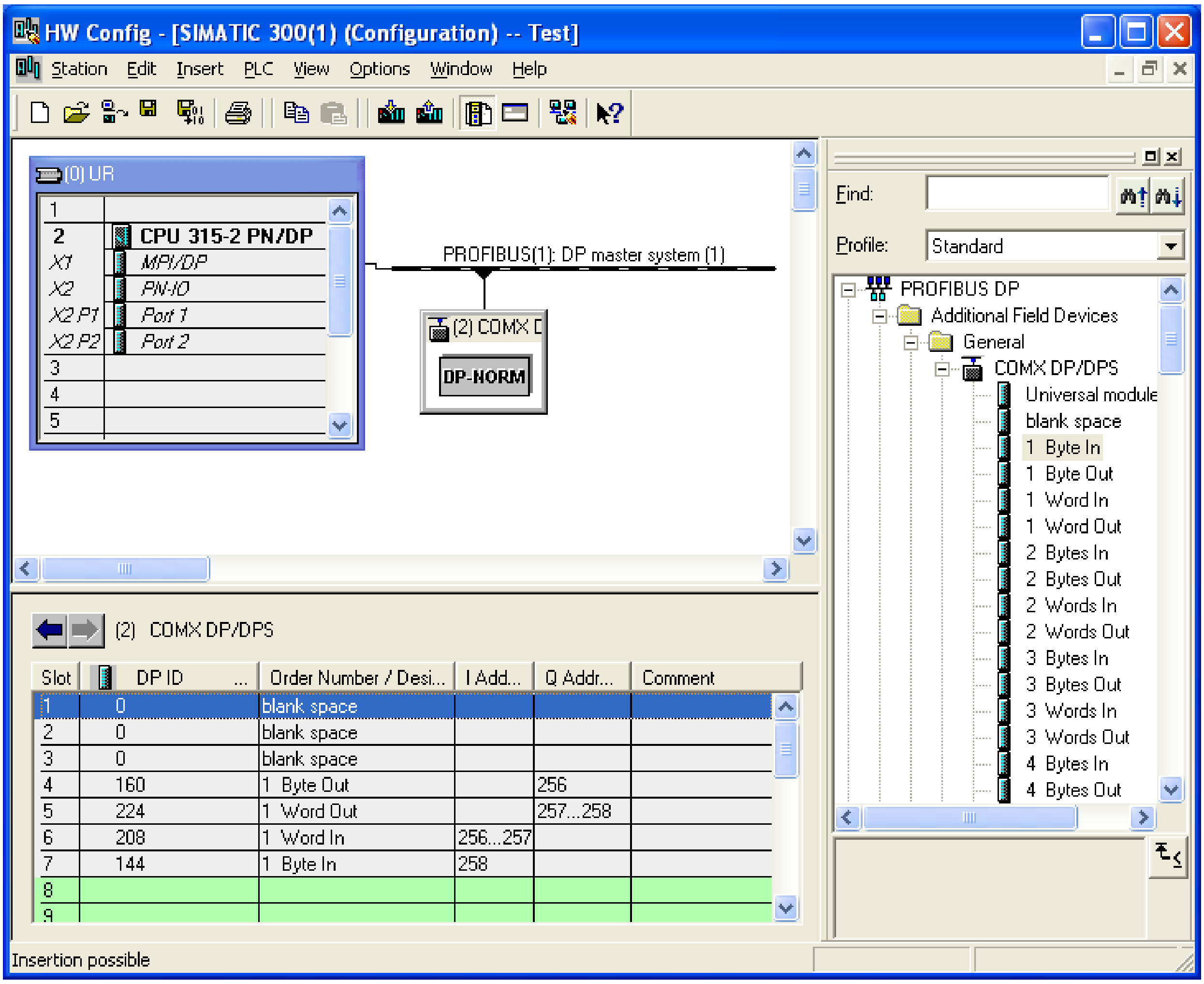
The input / output virtual slots under Properties->”Configuration” tab in Hardware Configurator.
**Note:**

- 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.

## DP Master configuration for INSEVIS Profibus DP-slave

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

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



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

(0) UR

| | |
|---|---|
| 1 | |
| 2 | CPU 315-2 PN/DP |
| X1 | MPI/DP |
| X2 | PN-IO |
| X2 P1 | Port 1 |
| X2 P2 | Port 2 |
| 3 | |
| 4 | |
| 5 | |

PROFIBUS(1): DP master system (1)

(2) COMX DP/DPS

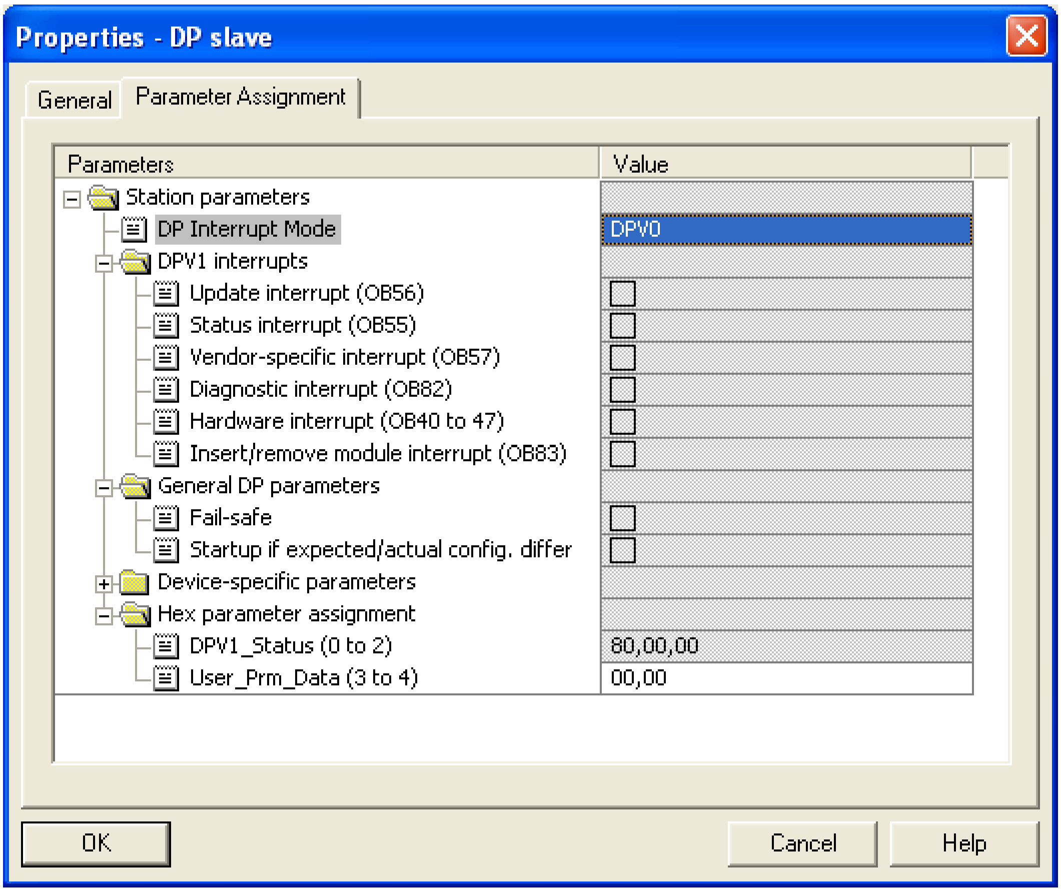
| Slot | DP ID | Order Number / Desi... | I Add... | Q Addr... | Comment |
|---|---|---|---|---|---|
| 1 | 0 | blank space | | | |
| 2 | 0 | blank space | | | |
| 3 | 0 | blank space | | | |
| 4 | 160 | 1 Byte Out | | 256 | |
| 5 | 224 | 1 Word Out | | 257...258 | |
| 6 | 208 | 1 Word In | 256...257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

Insertion possible

**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 completey, what is more complicated)
- Maximum 21 slots can be configured (exclusive first 3 blank space)

**Set DP-Slave parameter in properties->”Parameter Assignment” tab**



| Parameters | Value |
|---|---|
| Station parameters | |
| DP Interrupt Mode | DPV0 |
| DPV1 interrupts | |
| Update interrupt (OB56) | □ |
| Status interrupt (OB55) | □ |
| Vendor-specific interrupt (OB57) | □ |
| Diagnostic interrupt (OB82) | □ |
| Hardware interrupt (OB40 to 47) | □ |
| Insert/remove module interrupt (OB83) | □ |
| General DP parameters | |
| Fail-safe | □ |
| Startup if expected/actual config. differ | □ |
| Device-specific parameters | |
| Hex parameter assignment | |
| DPV1_Status (0 to 2) | 80,00,00 |
| User_Prm_Data (3 to 4) | 00,00 |

**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<br>...<br>Byte x - 1 | Module diagnostic (Identification related diagnostics)<br>(length depends on the number of the configured slots ) |
| Byte x<br>…<br>Byte y | Module status (Device related diagnostics)<br>(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<br>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<br>**0:** DP-Slave CPU in RUN<br>**1:** DP-Slave CPU is in STOP | Analyse Slave CPU's Diagnose buffer |
| 4 | **1:** Function is not supported | Check the configuration |
| 5 | Always **0** | |
| 6 | **1:** DP-Slave type inconsistent | Check parameter assignment |
| 7 | **1:** DP-Slave configured by other DP-Master | |

Station status byte 2

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

Station status byte 3

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

Master PROFIBUS address

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

Manufacturer ID

| Byte 4 | Byte 5 |
|---|---|
| **$0C_H$** | **$0F_H$** |

# Communication - Profibus DP-Slave

Module diagnostics (Identification related diagnostics)

| | |
|---|---|
| Byte 6 | Bit 6..7 = **01**: Code for module diagnostics<br>Bit 0..5 = Length of module diagnostics inclusive byte 6.<br>(dependent on the number of configured slots up to 4 byte) |
| Byte 7 | Bit 0 = Always **0**<br>Bit 1 = **1:** Slave CPU in STOP<br>**0:** Slave CPU in RUN<br>Bit 2 = Always **0**<br>Bit 3 = Entry for slot 4<br>Bit 4 = Entry for slot 5<br>Bit 5 = Entry for slot 6<br>Bit 6 = Entry for slot 7<br>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<br>Bit 0..5 = Length of module status inclusive byte x.<br>(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<br>Bit 2..3 = CPU status, **00:** Ok, **01:** Fault<br>Bit 4..5 = Always 0<br>Bit 6..7 = Slot 4 status, **00:** Ok, **01:** Fault |
| Byte x + 5 | Bit 0..1 = Slot 5 status<br>…<br>Bit 6..7 = Slot 8 status |
| … | … |
| Byte y | Bit 0..1 = Slot 21 status<br>Bit 2..3 = Slot 22 status<br>Bit 4..5 = Slot 23 status<br>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 assinged with the orignally 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<br>0000$_H$ No error<br>7000$_H$ First call with REQ=FALSE, No assignment<br>7001$_H$ First call with REQ=TRUE, Assignment is in process<br>7002$_H$ Intermediate call, Parameters are evaluated<br>8090$_H$ LADDR parameter invalid, must be fixed **W#16#7FF**<br>8091$_H$ ADDR parameter invalid<br>809b$_H$ Interface is not configured as DP-Slave<br>80C3H SFC called in lower priority level |
| BUSY | OUTPUT | BOOL | TRUE=Job is in process |

## 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<br>B#16#00 9.6 kbit/s<br>B#16#01 19.2 kbit/s<br>B#16#02 93.75 kbit/s<br>B#16#03 187.5 kbit/s<br>B#16#04 500 kbit/s<br>B#16#06 1.5 Mbit/s<br>B#16#07 3 Mbit/s<br>B#16#08 6 Mbit/s<br>B#16#09 12 Mbit/s<br>B#16#0a 31.25 kbit/s<br>B#16#0b 45.45 kbit/s<br>B#16#0f Auto |
| RET_VAL | OUTPUT | INT | Error code<br>$0000_H$ No error<br>$7000_H$ First call with REQ=FALSE, No assignment<br>$7001_H$ First call with REQ=TRUE, Assignment is in process<br>$7002_H$ Intermediate call, Parameters are evaluated<br>$8090_H$ LADDR parameter invalid, must be fixed **W#16#7FF**<br>$8091_H$ ADDR parameter invalid<br>$8092_H$ BAUD parameter invalid<br>$809b_H$ Interface is not configured as DP-Slave<br>80C3H SFC called in lower priority level |
| BUSY | OUTPUT | BOOL | TRUE=Job is in process |

# 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.**

## CANopen® Configuration with the ConfigStage

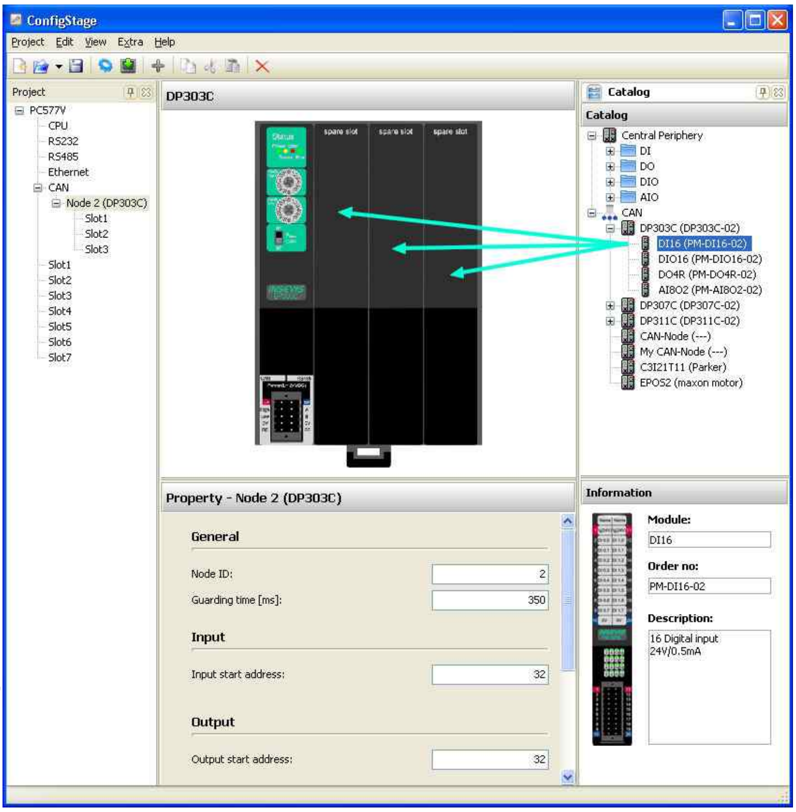
The settings are easy to do in the configuration software ConfigStage and
→ explained in the chapter „ConfigStage“ below the issue „CAN-settings“ in detail.

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.



Property - Node 2 (DP303C)

General

Node ID: 2

Guarding time [ms]: 350

Input

Input start address: 32

Output

Output start address: 32

Information

Module: DI16

Order no: PM-DI16-02

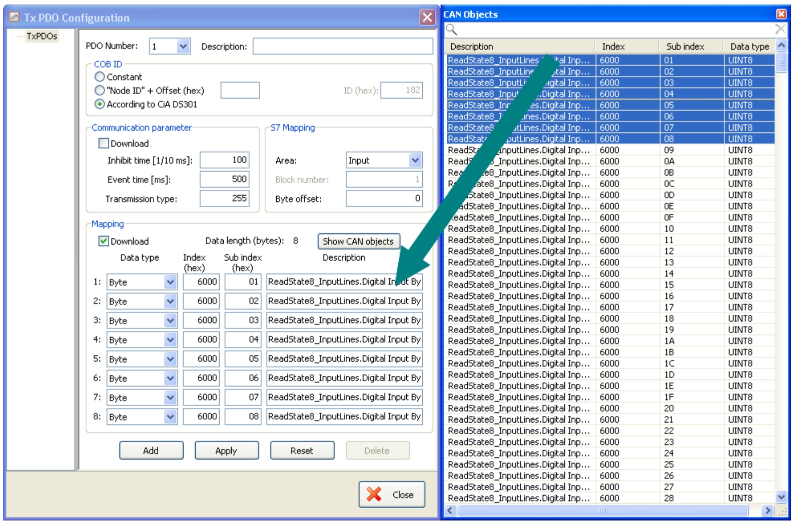
Description: 16 Digital input 24V/0.5mA

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.



PDO Number: 1 Description:

COB ID
○ Constant
○ "Node ID" + Offset (hex)
● According to CiA DS301
ID (hex): 182

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

S7 Mapping
Area: Input
Block number: 1
Byte offset: 0

Mapping
☑ Download
Data length (bytes): 8

| | Data type | Index (hex) | Sub index (hex) | Description |
|---|---|---|---|---|
| 1: | Byte | 6000 | 01 | ReadState8_InputLines.Digital Input By |
| 2: | Byte | 6000 | 02 | ReadState8_InputLines.Digital Input By |
| 3: | Byte | 6000 | 03 | ReadState8_InputLines.Digital Input By |
| 4: | Byte | 6000 | 04 | ReadState8_InputLines.Digital Input By |
| 5: | Byte | 6000 | 05 | ReadState8_InputLines.Digital Input By |
| 6: | Byte | 6000 | 06 | ReadState8_InputLines.Digital Input By |
| 7: | Byte | 6000 | 07 | ReadState8_InputLines.Digital Input By |
| 8: | Byte | 6000 | 08 | ReadState8_InputLines.Digital Input By |

| Description | Index | Sub index | Data type |
|---|---|---|---|
| ReadState8_InputLines.Digital Inp... | 6000 | 01 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 02 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 03 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 04 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 05 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 06 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 07 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 08 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 09 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 0A | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 0B | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 0C | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 0D | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 0E | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 0F | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 10 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 11 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 12 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 13 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 14 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 15 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 16 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 17 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 18 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 19 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 1A | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 1B | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 1C | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 1D | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 1E | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 1F | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 20 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 21 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 22 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 23 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 24 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 25 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 26 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 27 | UINT8 |
| ReadState8_InputLines.Digital Inp... | 6000 | 28 | UINT8 |

## 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<br>1: 29 bit |
| DLC | INPUT | BYTE | 5 | „Data Length Code“<br>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:<br>Bit 0: Buffer overflow e.g.<br>- due to „flood“ of transmit buffer by user program<br>- caused by wiring error no telegrams can be sent<br>Bit 1: Parameter error COB-ID<br>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“<br>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<br>1: 29 bit |
| DLC | OUTPUT | BYTE | 5 | „Data Length Code“<br>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<br>.0 Buffer empty: no data received<br>.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<br>0: SDO download, write CANopen variable<br>1: SDO upload, read CANopen variable |
| SEG | INPUT | BOOL | 0.2 | Segmented SDO transfer for objects > 4 byte<br>0: expedited SDO (1..4 byte)<br>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<br>0: OK<br>1: Parameter error<br>2: no SDO channel available<br>3: overflow of send buffer<br>4: received SDO error response<br>5: Error object index / subindex<br>6: Unexpected answer<br>7: Faulty length at seg. SDO<br>8: Faulty length at exp. SDO<br>9: Timeout |
| DATA | INOUT | ARRAY[1..6] of BYTE<br><br>Area pointer | 10 | Expedited SDO:<br>User data 1..4 byte<br><br>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<br>Node: node-ID<br>Object<br>Subindex<br>Data byte<br>Return values:<br>Done, Error, Error code | Request<br>Node: node-ID<br>Object<br>Subindex<br>Data byte<br>Return values:<br>Done, Error, Error code | Request<br>Node: node-ID<br>Object<br>Subindex<br>Data word<br>Return values:<br>Done, Error, Error code |

| | | |
|---|---|---|
| **FB110 SDO download variable word**<br>Request<br>Node: node-ID<br>Object<br>Subindex<br>Data word<br>Return values:<br>Done, Error, Error code | **FB111 SDO upload variable double word**<br>Request<br>Node: node-ID<br>Object<br>Subindex<br>Data double word<br>Return values:<br>Done, Error, Error code | **FB112 SDO download variable double word**<br>Request<br>Node: node-ID<br>Object<br>Subindex<br>Data double word<br>Return values:<br>Done, Error, Error code |
| **FB113 SDO transfer string**<br>Request<br>Upload-Control Flag (0=download, 1=upload)<br>Node: node-ID<br>Object<br>Subindex<br>S7-Area pointer in data<br>Length of data field / of data to be sent / of received data in byte<br>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<br>0: SDO download, writing CANopen-variable |
| SEG | INPUT | BOOL | 0.2 | Segmented SDO transfer for objects > 4 Byte<br>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<br>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<br>0: OK<br>1: Parameter error<br>2: no SDO chanal available<br>3: Send buffer overflow<br>4: SDO-error received response<br>5: Error object-index/ -subindex<br>6: Unexpected answer<br>9: Timeout |
| DATA | INOUT | ARRAY[1..6] of BYTE | 10 | Internal used,<br>Watching of actual user data possible,<br>Return of SDO-error code when abort |
| LEN | INOUT | WORD | 16 | Index of the actual processed list entry (via all nodes)<br>Done: Total number of processed entries (all nodes) |
| INTERN | STATIC | ARRAY[1..7] of BYTE | 18..24 | Internal variables |

# CAN

## 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<br>00 hex BOOTUP = at last received bootup telegram from node<br>01 hex UNKNOWN = configured but no response from node<br>02 hex MISSED = no more answer from node<br>03 hex UNAVAILABLE = no Guarding because of communication errors<br>04 hex STOPPED<br>05 hex OPERATIONAL<br>7F hex PREOPERATIONAL<br>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<br>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.

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.

```
// 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 bootup, state  == PREOPERATIONAL
      U(
      L     #CAN_State.NODESTATE[2]     // Node 2
      L     B#16#7F                     // 7F = PREOPERATIONAL
      ==I
      )
// 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

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

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

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

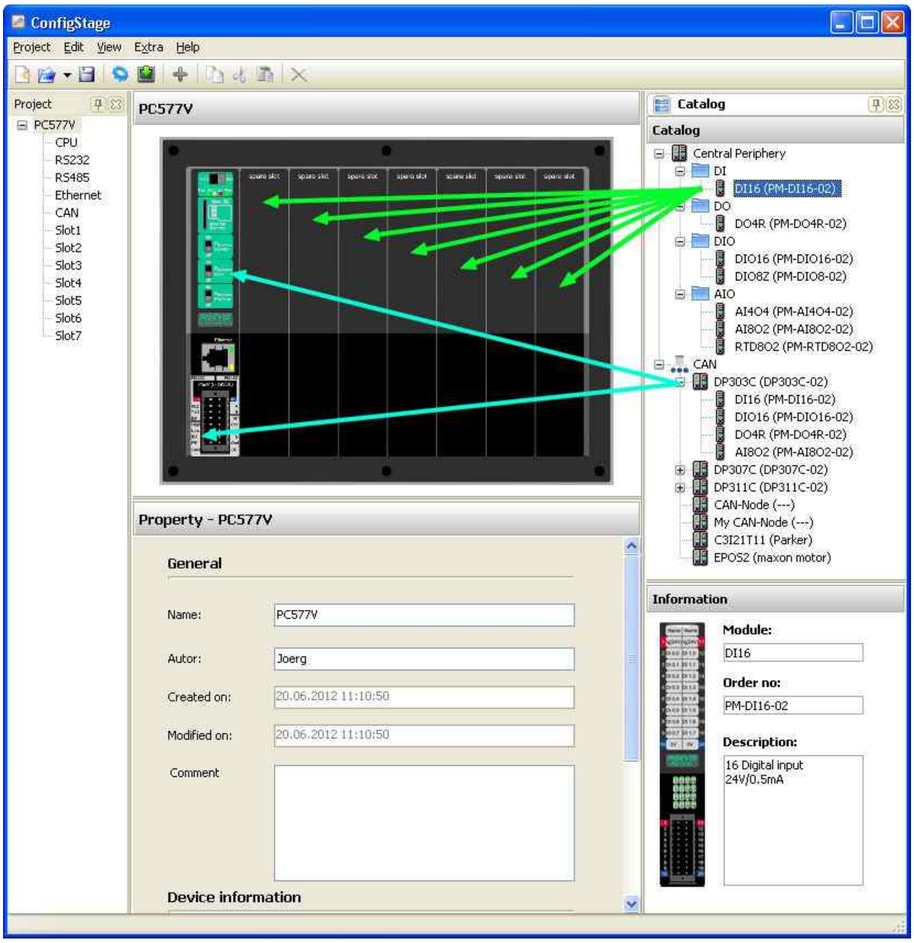
# Configuration with the Software „ConfigStage“

It is very easy to configure and parametrize all INSEVIS- PLCs and internal or external CAN-Periphery with the 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.

## Basic settings

| | |
|---|---|
| Middle | PLC you selected (from rear side) without any periphery |
| Left | self constructing project tree with the configurable functions of the CPU,<br>free slots and the periphery already configured |
| Right | catalog area with all periphery modules and decentral CAN-peripheries available<br>- periphery modules will be moved by drag`n drop to the preferred slot<br>- 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 |



Name: PC577V

Autor: Joerg

Created on: 20.06.2012 11:10:50

Modified on: 20.06.2012 11:10:50

Comment:

Module: DI16

Order no: PM-DI16-02

Description: 16 Digital input 24V/0.5mA

**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.

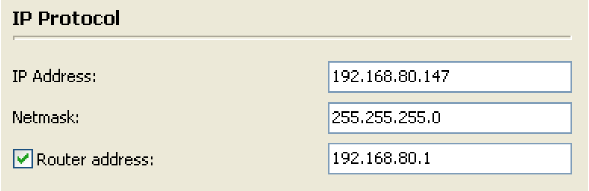
# Configuration with the Software „ConfigStage“

## Assign the IP-address

With ConfigStage 1.0.14.5 this software can **CHANGE** the IP-address of the target device.

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

(This needs the operating system 2.0.35 or higher in the INSEVIS-PLCs).



IP Protocol

IP Address: 192.168.80.147

Netmask: 255.255.255.0

☑ Router address: 192.168.80.1

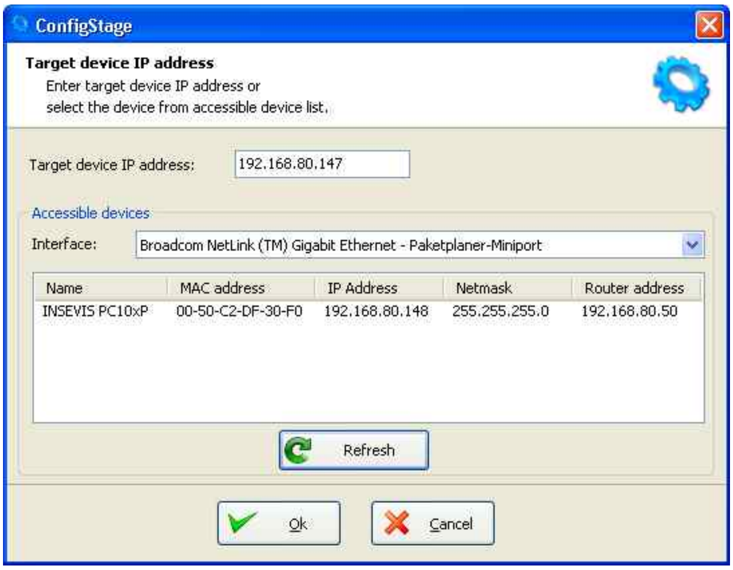
The IP-address of the target-device will be entered in the download-dialog box (F12).

In this sample a new IP-address was choosen in the new configuration (192.168.80.147, see below).

This new configuration must be downloaded to the old IP-adress 192.168.80.148 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.
After the download the device got its new configuration and is available with the new IP-address 192.168.80.147.



ConfigStage

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

Target device IP address: 192.168.80.147

Accessible devices

Interface: Broadcom NetLink (TM) Gigabit Ethernet - Paketplaner-Miniport

| Name | MAC address | IP Address | Netmask | Router address |
|---|---|---|---|---|
| INSEVIS PC10xP | 00-50-C2-DF-30-F0 | 192.168.80.148 | 255.255.255.0 | 192.168.80.50 |

## 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:
byte

| | | |
|---|---|---|
| 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 |

.

Attention

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

### Different addressing in PLCs and addressing of decentral periphery

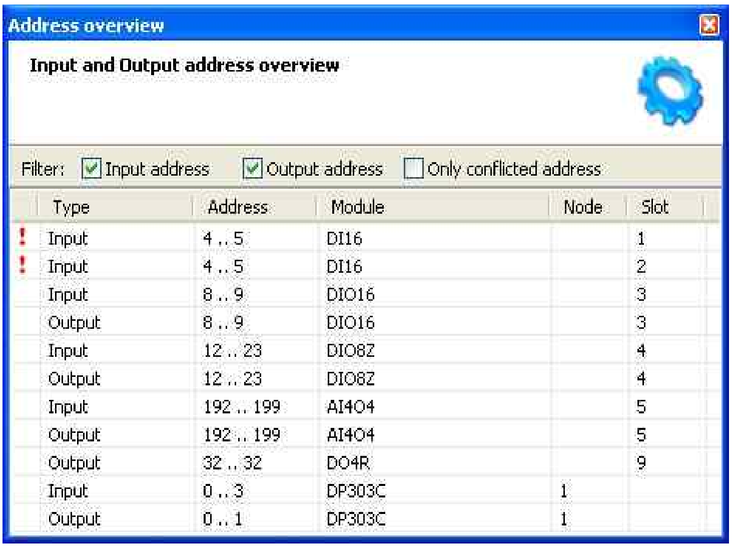
If decentral periphery is used and/or other should be used other address areas in the PLCs, these will be configured with the software tool „ConfigStage“ and the set up date will be stored in the system data blocks (SDBs).

# Configuration with the Software „ConfigStage“

From version 1.0.14.7 the used periphery address area can be displayed comfortable in an own window.

The pull down muenu „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.

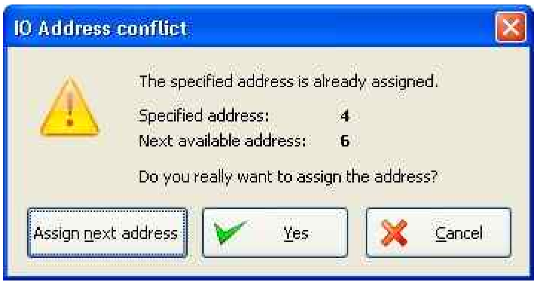


Address overview

Input and Output address overview

Filter: ☑ Input address ☑ Output address □ Only conflicted address

| | Type | Address | Module | Node | Slot |
|---|---|---|---|---|---|
| ! | Input | 4 .. 5 | DI16 | | 1 |
| ! | Input | 4 .. 5 | DI16 | | 2 |
| | Input | 8 .. 9 | DIO16 | | 3 |
| | Output | 8 .. 9 | DIO16 | | 3 |
| | Input | 12 .. 23 | DIO8Z | | 4 |
| | Output | 12 .. 23 | DIO8Z | | 4 |
| | Input | 192 .. 199 | AI4O4 | | 5 |
| | Output | 192 .. 199 | AI4O4 | | 5 |
| | Output | 32 .. 32 | DO4R | | 9 |
| | Input | 0 .. 3 | DP303C | 1 | |
| | Output | 0 .. 1 | DP303C | 1 | |



IO Address conflict

The specified address is already assigned.

Specified address: 4

Next available address: 6

Do you really want to assign the address?

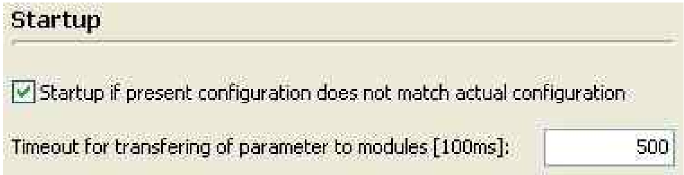
This conflict is shown in the address overview in red.

## 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.

### 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



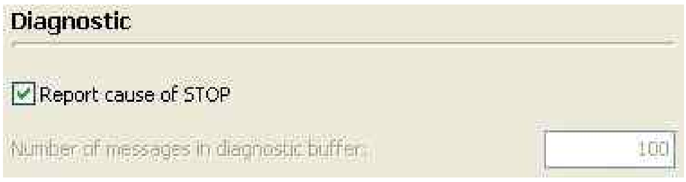
Startup

☑ Startup if present configuration does not match actual configuration

Timeout for transfering of parameter to modules [100ms]: 500

### Diagnostic

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



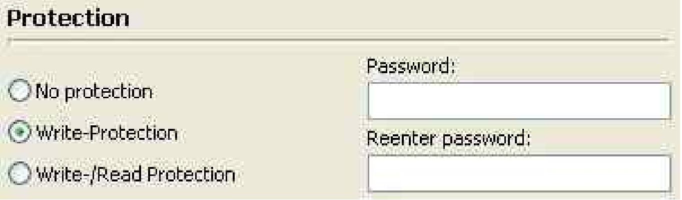
Diagnostic

☑ Report cause of STOP

Number of messages in diagnostic buffer: 100

### Know-how protection

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



Protection

○ No protection

● Write-Protection

○ Write-/Read Protection

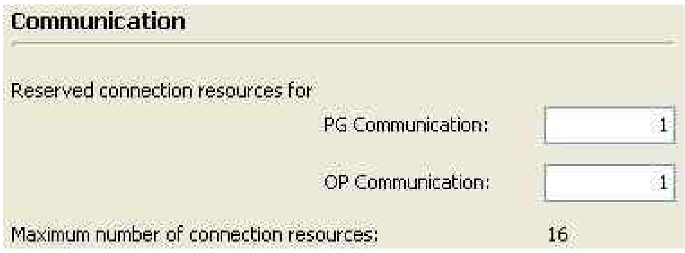
Password:

Reenter password:

### 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.



Communication

Reserved connection resources for

PG Communication: 1

OP Communication: 1

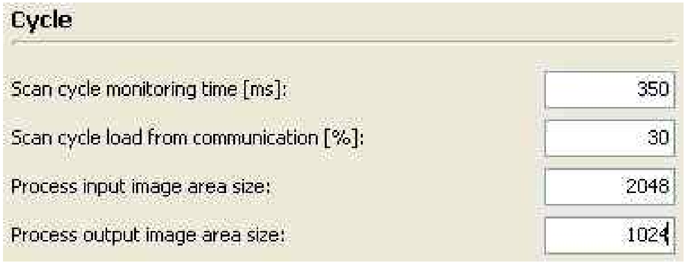
Maximum number of connection resources: 16

More at **Information about TSAPs**

# Configuration with the Software „ConfigStage“

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

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


Cycle

Scan cycle monitoring time [ms]: 350

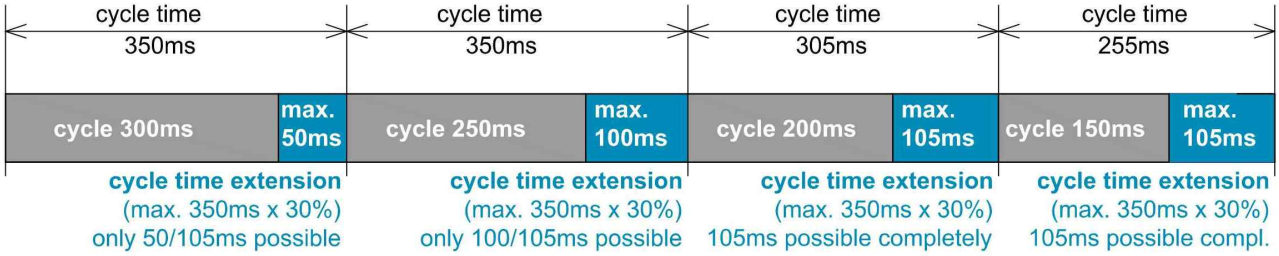
Scan cycle load from communication [%]: 30

Process input image area size: 2048

Process output image area size: 1024

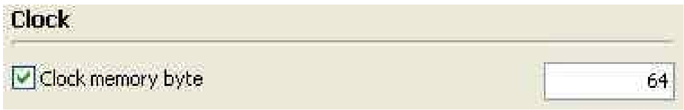
The declaration of the cycle time in the PG is carried out from control point to control point. This contains communication and visualization. Independent from the S7-program an unsteady cycle time can occur.

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).



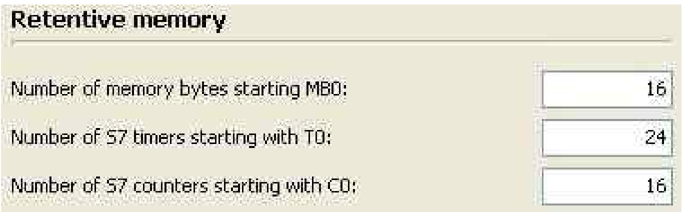
In this sample the cycle time can be extended max. by 30% of 350ms (= 105ms). If this time is not needed for communication (visualization, etc), it is available for other tasks.

**Clock**
check memory byte and insert its
number from value 0


Clock

☑ Clock memory byte 64

**Retentive memory**
<u>Merkerbytes:</u>
*total 2.048 (thereof 0..15 preset)*
<u>Timer and counter:</u>
*total 256 (no preset)*

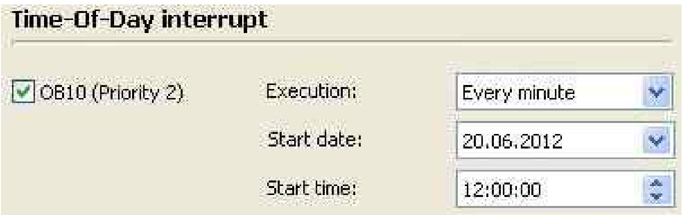

Retentive memory

Number of memory bytes starting MB0: 16

Number of S7 timers starting with T0: 24

Number of S7 counters starting with C0: 16

**Time-Of-Day interrupt**
<u>Box checked:</u> Time-Of-Day interrupt activated
Insert interval areas, starting date and starting time


Time-Of-Day interrupt

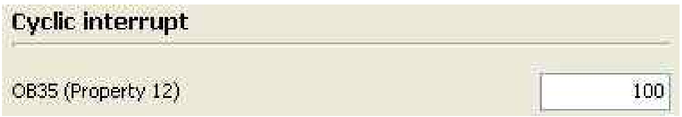
☑ OB10 (Priority 2)

Execution: Every minute

Start date: 20.06.2012

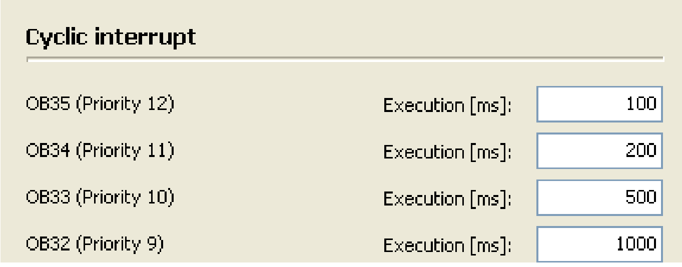
Start time: 12:00:00

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


Cyclic interrupt

OB35 (Property 12) 100

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


Cyclic interrupt

OB35 (Priority 12) Execution [ms]: 100

OB34 (Priority 11) Execution [ms]: 200

OB33 (Priority 10) Execution [ms]: 500

OB32 (Priority 9) Execution [ms]: 1000

# Configuration with the Software „ConfigStage“

## Communication settings

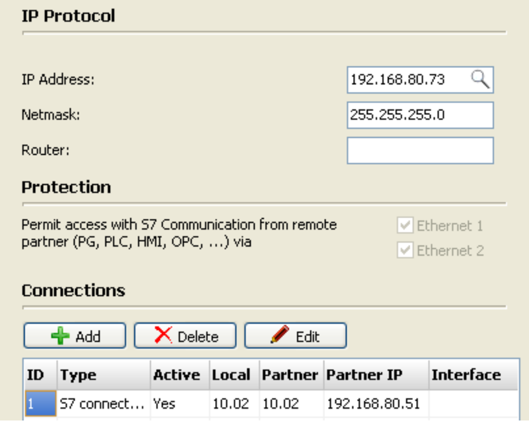
### RS232 and RS485

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



**IP Protocol**

IP Address: 192.168.80.73
Netmask: 255.255.255.0
Router:

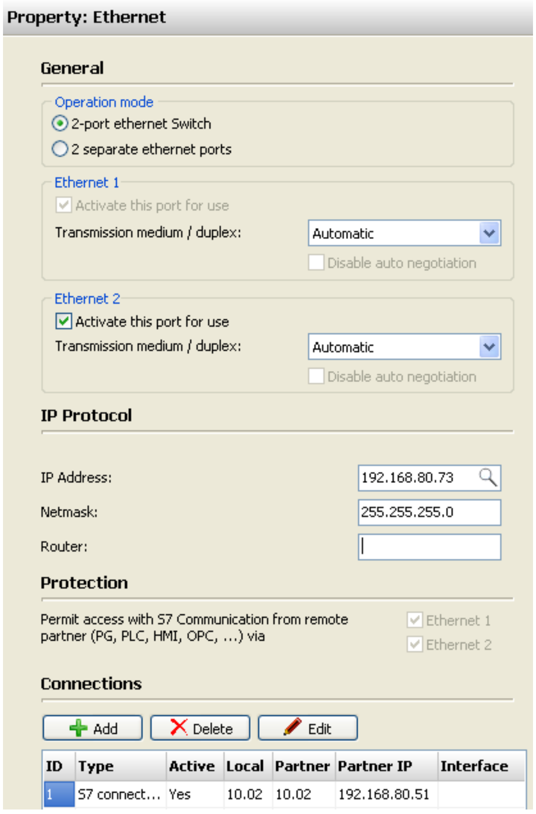
**Protection**

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

**Connections**

| ID | Type | Active | Local | Partner | Partner IP | Interface |
|---|---|---|---|---|---|---|
| 1 | S7 connect... | Yes | 10.02 | 10.02 | 192.168.80.51 | |

**CPU-T:** 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).



**Property: Ethernet**

**General**

Operation mode
● 2-port ethernet Switch
○ 2 separate ethernet ports

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

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

**IP Protocol**

IP Address: 192.168.80.73
Netmask: 255.255.255.0
Router:

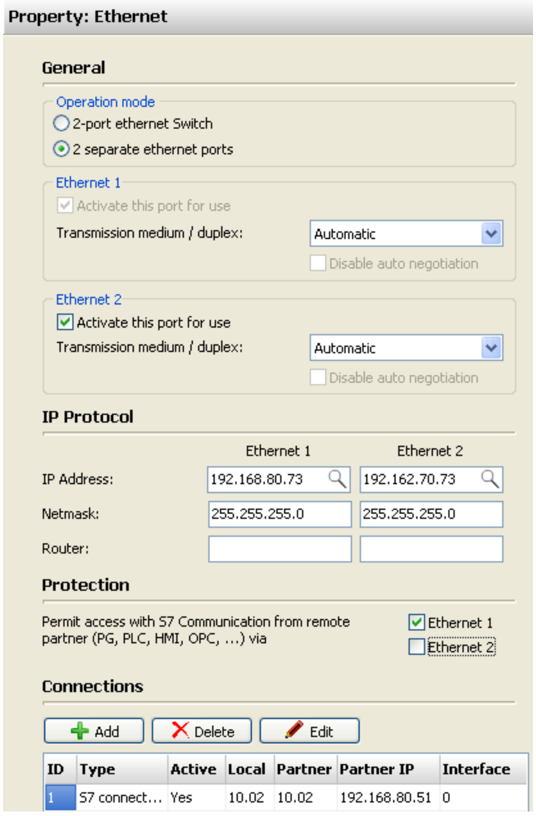
**Protection**

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

**Connections**

| ID | Type | Active | Local | Partner | Partner IP | Interface |
|---|---|---|---|---|---|---|
| 1 | S7 connect... | Yes | 10.02 | 10.02 | 192.168.80.51 | |

Both ports can be used



**Property: Ethernet**

**General**

Operation mode
○ 2-port ethernet Switch
● 2 separate ethernet ports

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

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

**IP Protocol**

| | Ethernet 1 | Ethernet 2 |
|---|---|---|
| IP Address: | 192.168.80.73 | 192.162.70.73 |
| Netmask: | 255.255.255.0 | 255.255.255.0 |
| Router: | | |

**Protection**

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

**Connections**

| ID | Type | Active | Local | Partner | Partner IP | Interface |
|---|---|---|---|---|---|---|
| 1 | S7 connect... | Yes | 10.02 | 10.02 | 192.168.80.51 | 0 |

Port 2 can not more communicate by S7-communication (protection)

**Hint:**

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

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

**parameters at S7-connection (Active)**

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

**parameters at INSEVIS-Panel-HMI**

- Local TSAP*

**parameters at TCP Send/Receive**

- Local port,
- Partner-port,
- Partner-IP-address

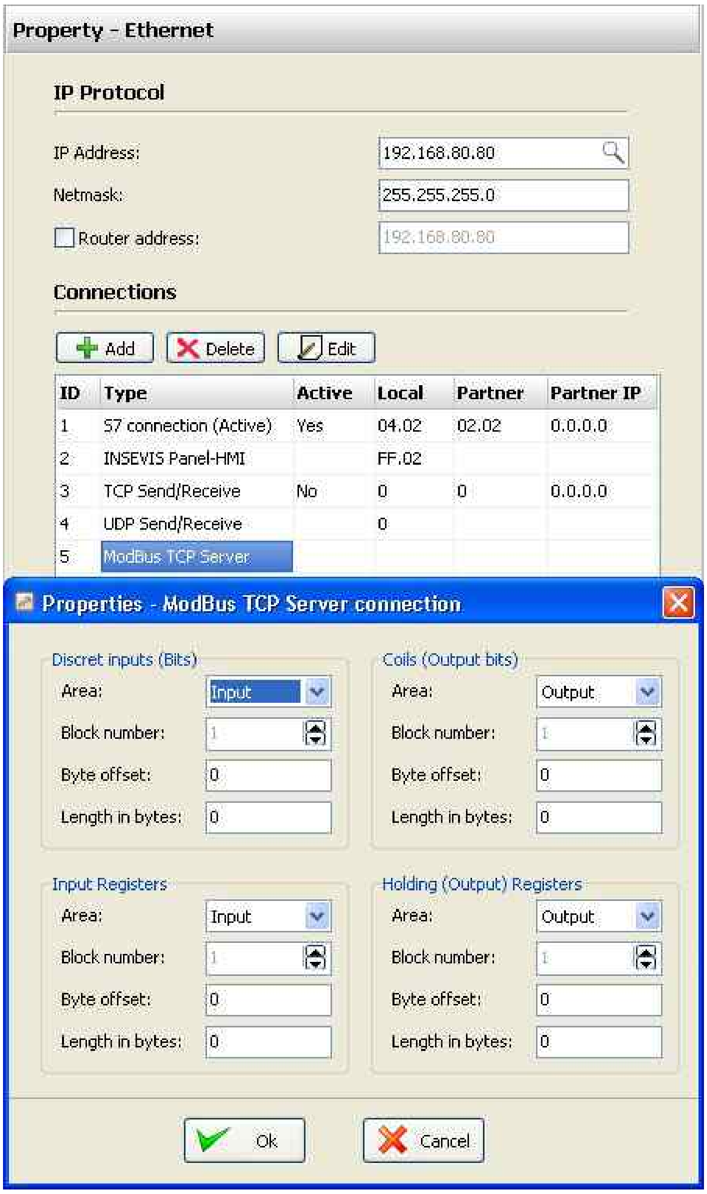
**parameters at UDP Send/Receive**

- Local port

**parameters at Modbus-TCP (Server)**
Assign of S7-operand areas for Modbus-

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

(As sample again the „Mapping“ at Modbus-Server, but here as Modbus TCP)



Property - Ethernet

IP Protocol

IP Address: 192.168.80.80

Netmask: 255.255.255.0

□ Router address: 192.168.80.80

Connections

| ID | Type | Active | Local | Partner | Partner IP |
|---|---|---|---|---|---|
| 1 | S7 connection (Active) | Yes | 04.02 | 02.02 | 0.0.0.0 |
| 2 | INSEVIS Panel-HMI | | FF.02 | | |
| 3 | TCP Send/Receive | No | 0 | 0 | 0.0.0.0 |
| 4 | UDP Send/Receive | | 0 | | |
| 5 | ModBus TCP Server | | | | |

Properties - ModBus TCP Server connection

Discret inputs (Bits)
Area: Input
Block number: 1
Byte offset: 0
Length in bytes: 0

Coils (Output bits)
Area: Output
Block number: 1
Byte offset: 0
Length in bytes: 0

Input Registers
Area: Input
Block number: 1
Byte offset: 0
Length in bytes: 0

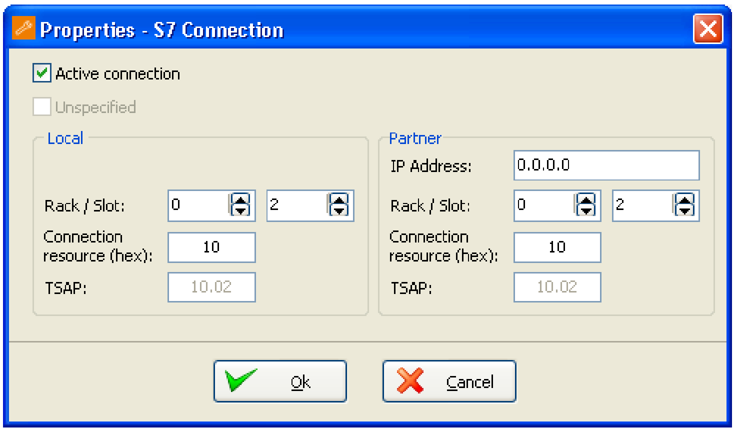
Holding (Output) Registers
Area: Output
Block number: 1
Byte offset: 0
Length in bytes: 0

# 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)


Properties - S7 Connection

☑ Active connection

☐ Unspecified

Local

Rack / Slot: 0 2

Connection resource (hex): 10

TSAP: 10.02

Partner

IP Address: 0.0.0.0

Rack / Slot: 0 2

Connection resource (hex): 10

TSAP: 10.02

**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.

## CAN configuration

### Decentral INSEVIS periphery

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

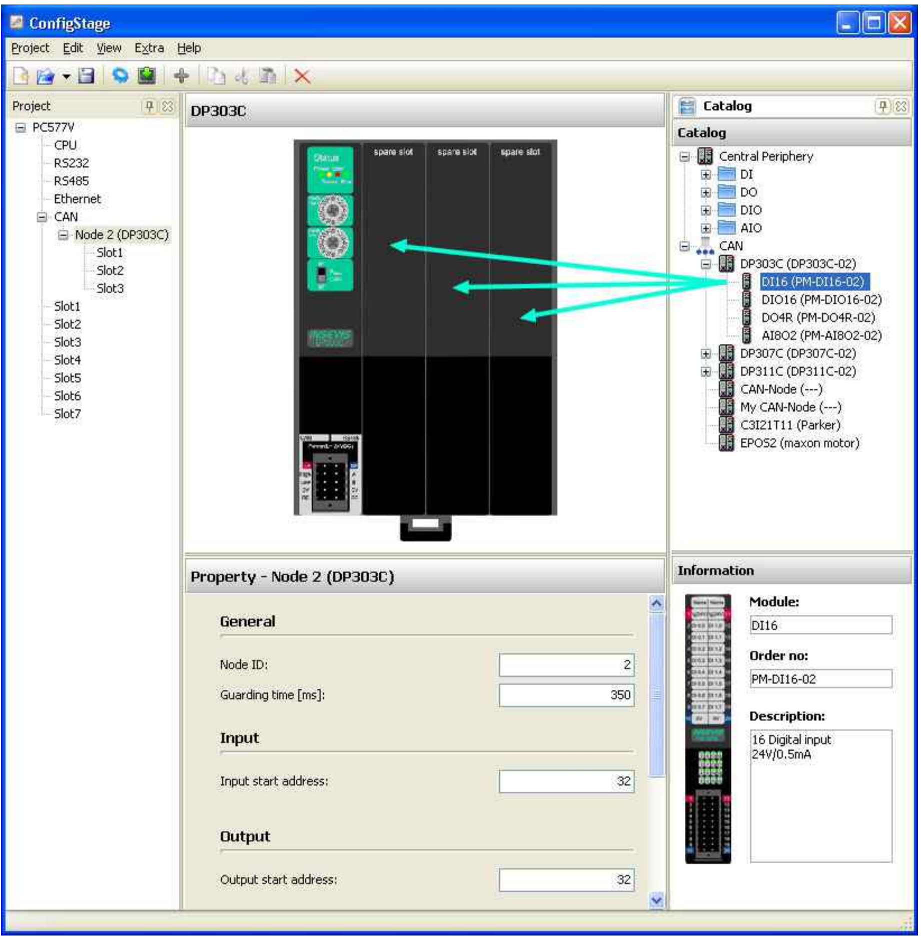
After having placed your INSEVIS head station on the CPU, this module appears in the project tree and in the display.

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.



Property - Node 2 (DP303C)

General

Node ID: 2

Guarding time [ms]: 350

Input

Input start address: 32

Output

Output start address: 32

Module: DI16

Order no: PM-DI16-02

Description: 16 Digital input 24V/0.5mA

# Configuration with the Software „ConfigStage“

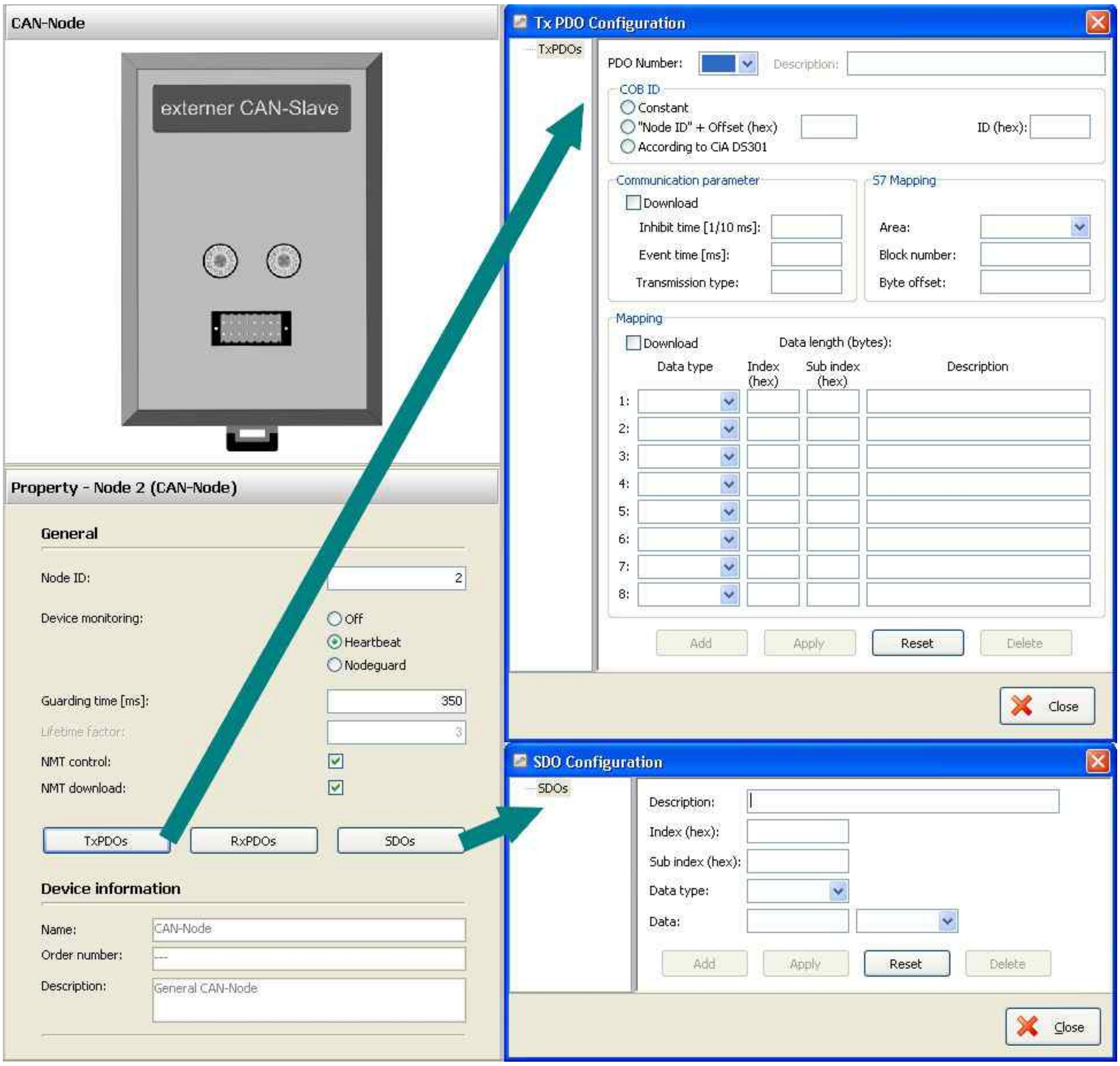
## Configure decentral external periphery manually

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



Property - Node 2 (CAN-Node)

General

Node ID: 2

Device monitoring: ○ Off ● Heartbeat ○ Nodeguard

Guarding time [ms]: 350

Lifetime factor: 3

NMT control: ☑

NMT download: ☑

TxPDOs RxPDOs SDOs

Device information

Name: CAN-Node

Order number: ---

Description: General CAN-Node

For the process data are available each **32 Tx- and Rx-PDOs**. If more as **4 PDOs** of it will be used, PDO-identifiers of other node-IDs will be used for it. These node-IDs should not be assigned again. (Only CiA-conform TxPDO-identifier are valid.) Received Tx-PDOs will be buffered and transferred synchronous to the control point to the process image, RxPDOs will be send event- and time-controlled in the control point always.
An 8-byte-data field on a selectable S7-operand is to assign to every PDO.

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

# Configuration with the Software „ConfigStage“

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

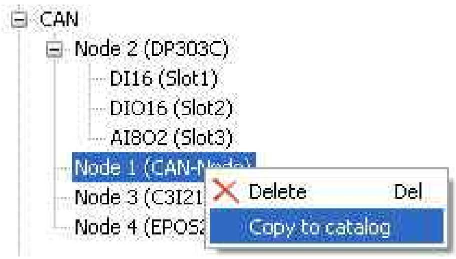
Entering of single CAN-objects (Index, Subindex) in the mapping field is optional, as long the download of the mapping is not activated. Always assign the data type to define the telegrams length and to allow a real byte swapping.
With activated download of complete mapping-parameter the PLC configures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

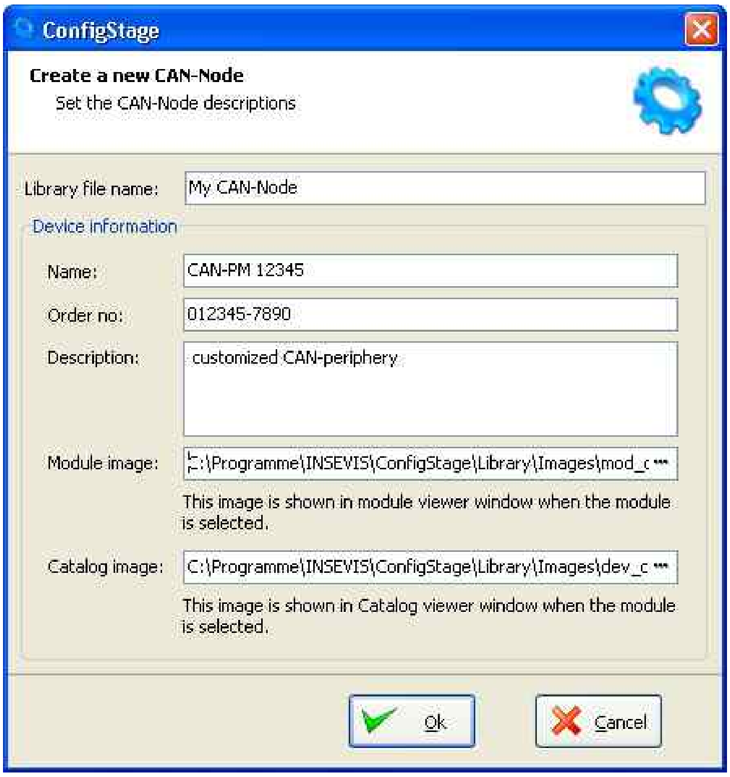
**Optional:** If this „mapping“ is made, the new configured CAN-node (CAN-slave) can be stored locally in the ConfigStage to use this part as a library element with pre-defined values.

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.



**Create a new CAN-Node**

Set the CAN-Node descriptions

Library file name: My CAN-Node

Device information

Name: CAN-PM 12345

Order no: 012345-7890

Description: customized CAN-periphery

Module image: C:\Programme\INSEVIS\ConfigStage\Library\Images\mod_c ...

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

Catalog image: C:\Programme\INSEVIS\ConfigStage\Library\Images\dev_c ...

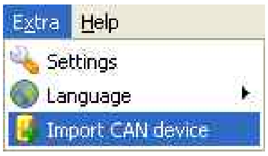
This image is shown in Catalog viewer window when the module is selected.

.

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

## Configure decentral external peripherie by EDS-file

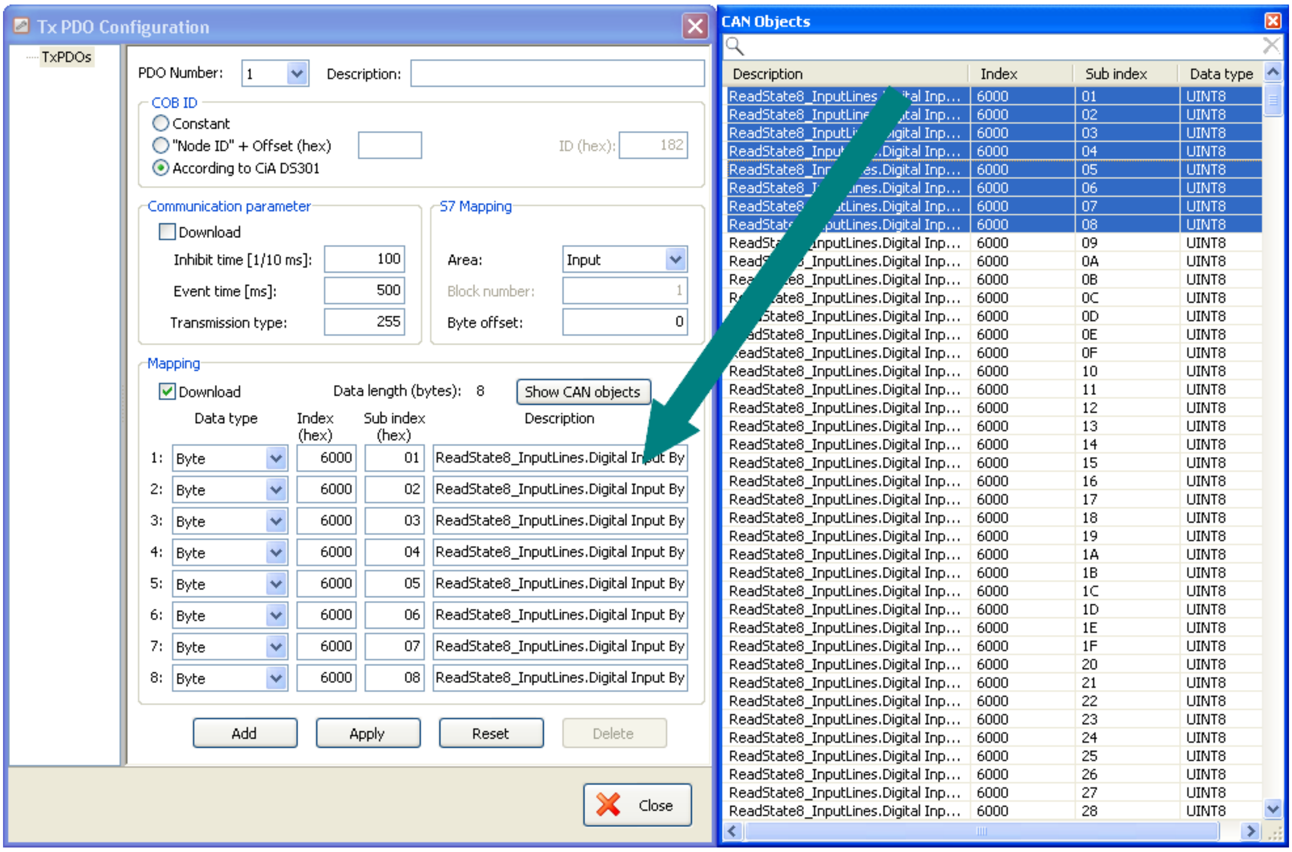
You need a EDS-file of the external CAN-slave you want to configure. Import it like shown here:



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.



PDO Number: 1 Description:

COB ID
○ Constant
○ "Node ID" + Offset (hex)
● According to CiA DS301
ID (hex): 182

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

S7 Mapping
Area: Input
Block number: 1
Byte offset: 0

Mapping
☑ Download
Data length (bytes): 8

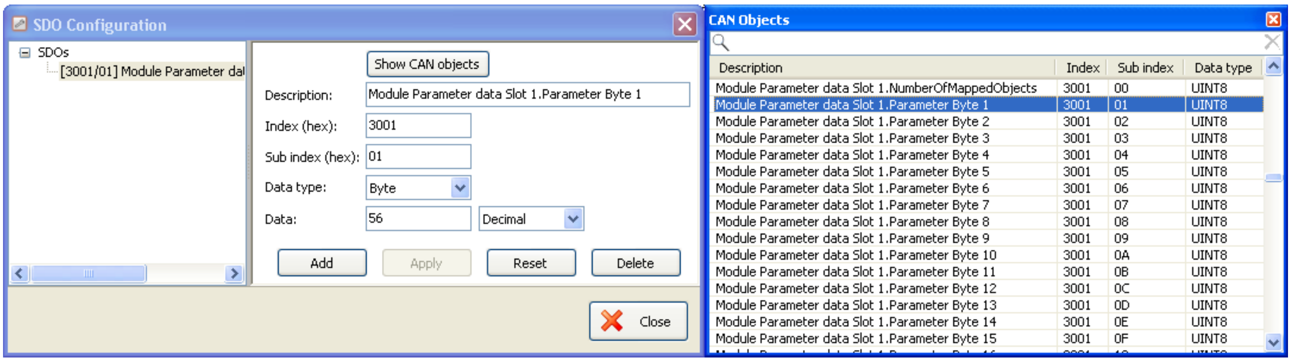
| | Data type | Index (hex) | Sub index (hex) | Description |
|---|---|---|---|---|
| 1: | Byte | 6000 | 01 | ReadState8_InputLines.Digital Input By |
| 2: | Byte | 6000 | 02 | ReadState8_InputLines.Digital Input By |
| 3: | Byte | 6000 | 03 | ReadState8_InputLines.Digital Input By |
| 4: | Byte | 6000 | 04 | ReadState8_InputLines.Digital Input By |
| 5: | Byte | 6000 | 05 | ReadState8_InputLines.Digital Input By |
| 6: | Byte | 6000 | 06 | ReadState8_InputLines.Digital Input By |
| 7: | Byte | 6000 | 07 | ReadState8_InputLines.Digital Input By |
| 8: | Byte | 6000 | 08 | ReadState8_InputLines.Digital Input By |

| Description | Index | Sub index | Data type |
|---|---|---|---|
| ReadState8_InputLines.Digital Inp… | 6000 | 01 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 02 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 03 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 04 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 05 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 06 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 07 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 08 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 09 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 0A | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 0B | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 0C | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 0D | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 0E | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 0F | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 10 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 11 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 12 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 13 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 14 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 15 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 16 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 17 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 18 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 19 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 1A | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 1B | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 1C | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 1D | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 1E | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 1F | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 20 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 21 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 22 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 23 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 24 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 25 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 26 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 27 | UINT8 |
| ReadState8_InputLines.Digital Inp… | 6000 | 28 | UINT8 |

# Configuration with the Software „ConfigStage“

### SDO- configuration

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



[3001/01] Module Parameter dat

Description: Module Parameter data Slot 1.Parameter Byte 1
Index (hex): 3001
Sub index (hex): 01
Data type: Byte
Data: 56 Decimal

| Description | Index | Sub index | Data type |
|---|---|---|---|
| Module Parameter data Slot 1.NumberOfMappedObjects | 3001 | 00 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 1 | 3001 | 01 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 2 | 3001 | 02 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 3 | 3001 | 03 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 4 | 3001 | 04 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 5 | 3001 | 05 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 6 | 3001 | 06 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 7 | 3001 | 07 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 8 | 3001 | 08 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 9 | 3001 | 09 | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 10 | 3001 | 0A | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 11 | 3001 | 0B | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 12 | 3001 | 0C | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 13 | 3001 | 0D | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 14 | 3001 | 0E | UINT8 |
| Module Parameter data Slot 1.Parameter Byte 15 | 3001 | 0F | UINT8 |

## 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

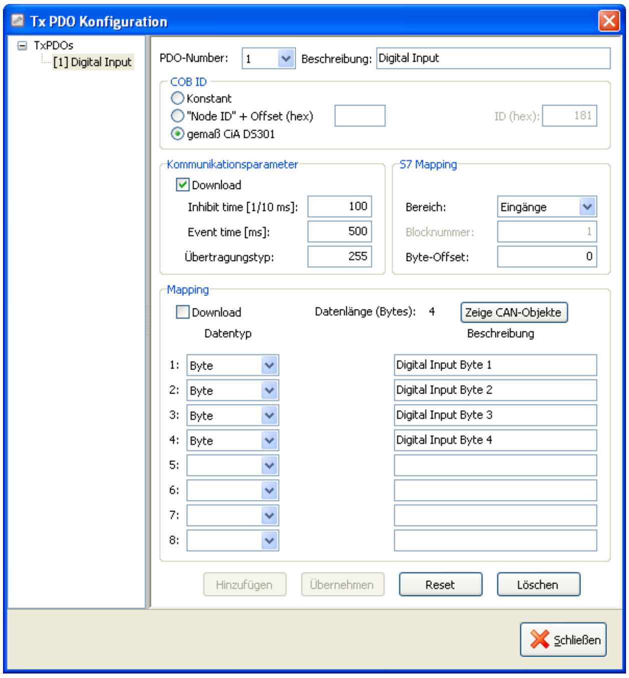
# Configuration with the Software „ConfigStage“

## 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



Tx PDO Konfiguration

TxPDOs – [1] Digital Input

PDO-Number: 1 Beschreibung: Digital Input

COB ID: ○ Konstant ○ "Node ID" + Offset (hex) ● gemaß CiA DS301 ID (hex): 181

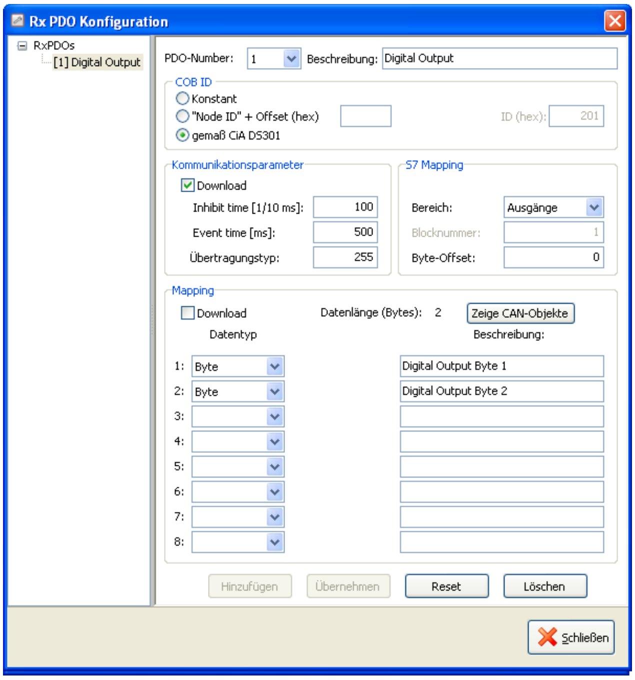
Kommunikationsparameter: ☑ Download; Inhibit time [1/10 ms]: 100; Event time [ms]: 500; Übertragungstyp: 255

S7 Mapping: Bereich: Eingänge; Blocknummer: 1; Byte-Offset: 0

Mapping: □ Download Datenlänge (Bytes): 4

| | Datentyp | Beschreibung |
|---|---|---|
| 1: | Byte | Digital Input Byte 1 |
| 2: | Byte | Digital Input Byte 2 |
| 3: | Byte | Digital Input Byte 3 |
| 4: | Byte | Digital Input Byte 4 |
| 5: | | |
| 6: | | |
| 7: | | |
| 8: | | |

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



Rx PDO Konfiguration

RxPDOs – [1] Digital Output

PDO-Number: 1 Beschreibung: Digital Output

COB ID: ○ Konstant ○ "Node ID" + Offset (hex) ● gemaß CiA DS301 ID (hex): 201

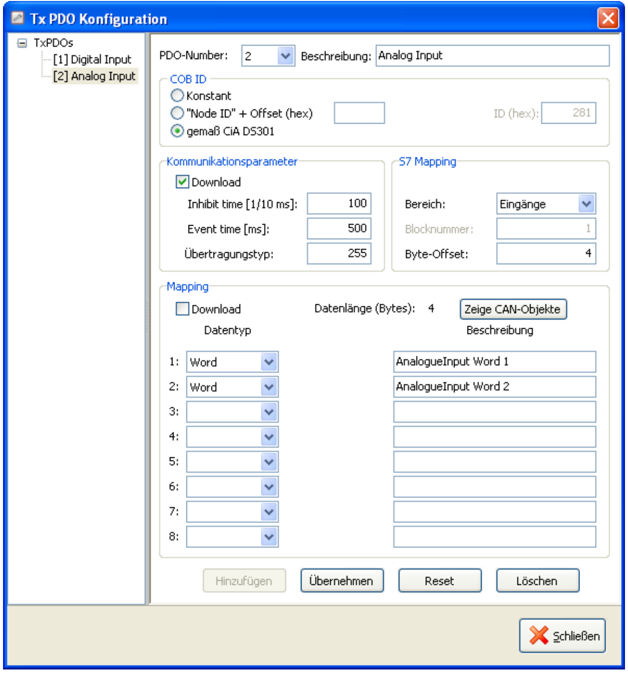
Kommunikationsparameter: ☑ Download; Inhibit time [1/10 ms]: 100; Event time [ms]: 500; Übertragungstyp: 255

S7 Mapping: Bereich: Ausgänge; Blocknummer: 1; Byte-Offset: 0

Mapping: □ Download Datenlänge (Bytes): 2

| | Datentyp | Beschreibung: |
|---|---|---|
| 1: | Byte | Digital Output Byte 1 |
| 2: | Byte | Digital Output Byte 2 |
| 3: | | |
| 4: | | |
| 5: | | |
| 6: | | |
| 7: | | |
| 8: | | |

- analog inputs in TxPDO2 to 4



Tx PDO Konfiguration

TxPDOs – [1] Digital Input, [2] Analog Input

PDO-Number: 2 Beschreibung: Analog Input

COB ID: ○ Konstant ○ "Node ID" + Offset (hex) ● gemaß CiA DS301 ID (hex): 281

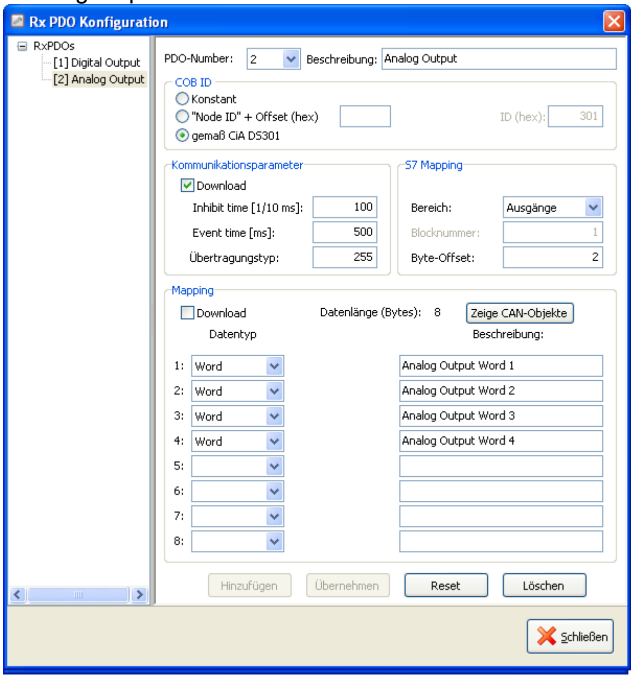
Kommunikationsparameter: ☑ Download; Inhibit time [1/10 ms]: 100; Event time [ms]: 500; Übertragungstyp: 255

S7 Mapping: Bereich: Eingänge; Blocknummer: 1; Byte-Offset: 4

Mapping: □ Download Datenlänge (Bytes): 4

| | Datentyp | Beschreibung |
|---|---|---|
| 1: | Word | AnalogueInput Word 1 |
| 2: | Word | AnalogueInput Word 2 |
| 3: | | |
| 4: | | |
| 5: | | |
| 6: | | |
| 7: | | |
| 8: | | |

- analog outputs in RxPDO2 to 4



Rx PDO Konfiguration

RxPDOs – [1] Digital Output, [2] Analog Output

PDO-Number: 2 Beschreibung: Analog Output

COB ID: ○ Konstant ○ "Node ID" + Offset (hex) ● gemaß CiA DS301 ID (hex): 301

Kommunikationsparameter: ☑ Download; Inhibit time [1/10 ms]: 100; Event time [ms]: 500; Übertragungstyp: 255

S7 Mapping: Bereich: Ausgänge; Blocknummer: 1; Byte-Offset: 2

Mapping: □ Download Datenlänge (Bytes): 8

| | Datentyp | Beschreibung: |
|---|---|---|
| 1: | Word | Analog Output Word 1 |
| 2: | Word | Analog Output Word 2 |
| 3: | Word | Analog Output Word 3 |
| 4: | Word | Analog Output Word 4 |
| 5: | | |
| 6: | | |
| 7: | | |
| 8: | | |

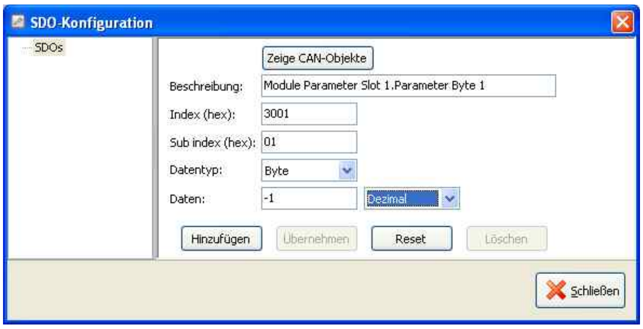
Therewith the in- and outputs can be used on the configured S7-address (in this sample from EB0, AB0) as usually.

Attention

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

# Configuration with the Software „ConfigStage“

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



SDO-Konfiguration

SDOs

Beschreibung: Module Parameter Slot 1.Parameter Byte 1

Index (hex): 3001

Sub index (hex): 01

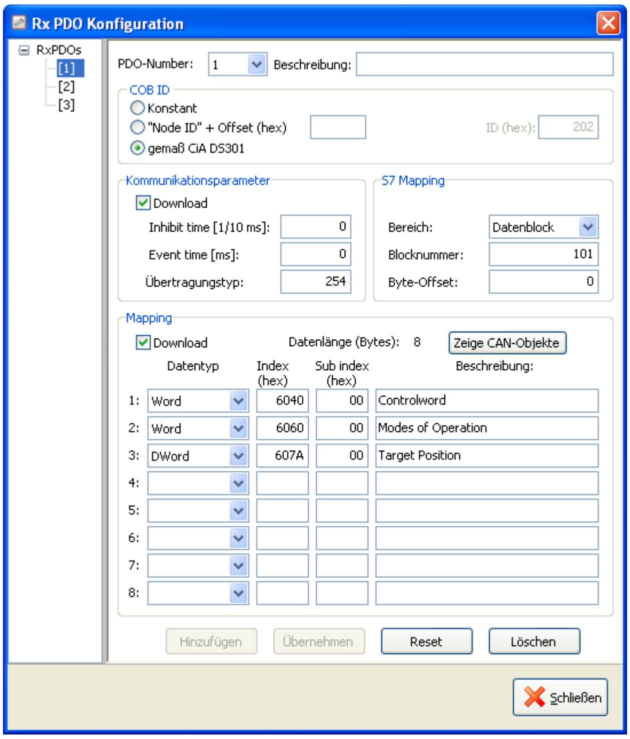
Datentyp: Byte

Daten: -1 Dezimal

**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



Rx PDO Konfiguration

RxPDOs [1] [2] [3]

PDO-Number: 1 Beschreibung:

COB ID

○ Konstant

○ "Node ID" + Offset (hex) ID (hex): 202

● gemaß CiA DS301

Kommunikationsparameter

☑ Download

Inhibit time [1/10 ms]: 0

Event time [ms]: 0

Übertragungstyp: 254

S7 Mapping

Bereich: Datenblock

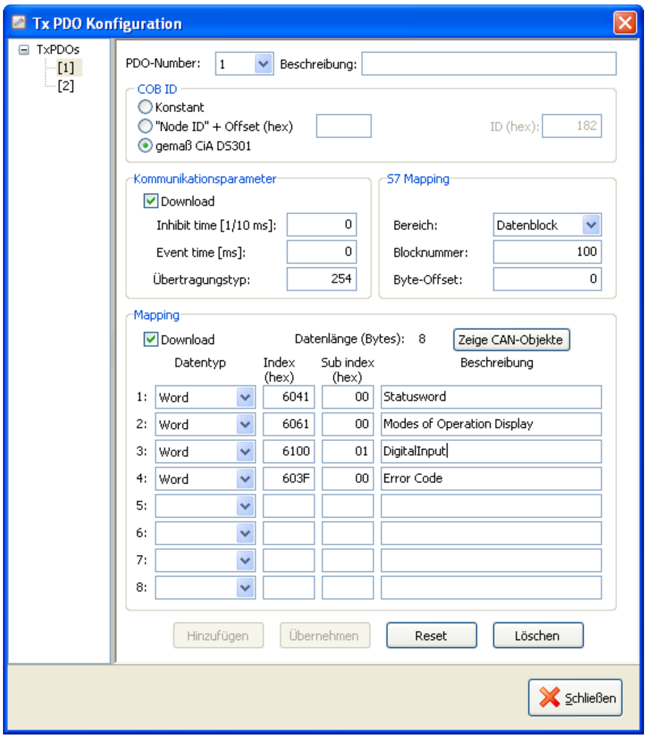
Blocknummer: 101

Byte-Offset: 0

Mapping

☑ Download Datenlänge (Bytes): 8

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



Tx PDO Konfiguration

TxPDOs [1] [2]

PDO-Number: 1 Beschreibung:

COB ID

○ Konstant

○ "Node ID" + Offset (hex) ID (hex): 182

● gemaß CiA DS301

Kommunikationsparameter

☑ Download

Inhibit time [1/10 ms]: 0

Event time [ms]: 0

Übertragungstyp: 254

S7 Mapping

Bereich: Datenblock

Blocknummer: 100

Byte-Offset: 0

Mapping

☑ Download Datenlänge (Bytes): 8

| | Datentyp | Index (hex) | Sub index (hex) | Beschreibung |
|---|---|---|---|---|
| 1: | Word | 6041 | 00 | Statusword |
| 2: | Word | 6061 | 00 | Modes of Operation Display |
| 3: | Word | 6100 | 01 | DigitalInput |
| 4: | Word | 603F | 00 | Error Code |
| 5: | | | | |
| 6: | | | | |
| 7: | | | | |
| 8: | | | | |

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)<br>*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 ) |

### 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 |

## 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<br>*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<br>Block processing of new alarm- and asynchronous error events |
| SFC 40 | at CPU-T only / not available for CPU-V/P<br>Release processing of new alarm- and asynchronous error events |
| SFC 41 | at CPU-T only / not available for CPU-V/P<br>Retard processing of new alarm- and asynchronous error events with higher priority |
| SFC 42 | at CPU-T only / not available for CPU-V/P<br>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 file on Micro-SD-card |
| SFC 208 BACKUP | Backup |
| SFC 209 READFILE | Read data from archive file on Micro-SD-card |
| SFC 210 COPY_DBL | Copy selected DBs from RAM to ROM |

# System functions

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

| SFC | **Function** *(more information at single SFC-chapters)* |
|---|---|
| SFC 211 COPY_DBW | Copy selected DBs from ROM to RAM |
| SFC 213 READ_CSV | Read data from CSV-formatted archive file (SD) |
| SFC 214 LCD_BRIG | Change LCD-backlight brightness |
| SFC 215 LOGIN | Change user level at runtime |
| SFC 254 RAM2ROM | Copy RAM to ROM |

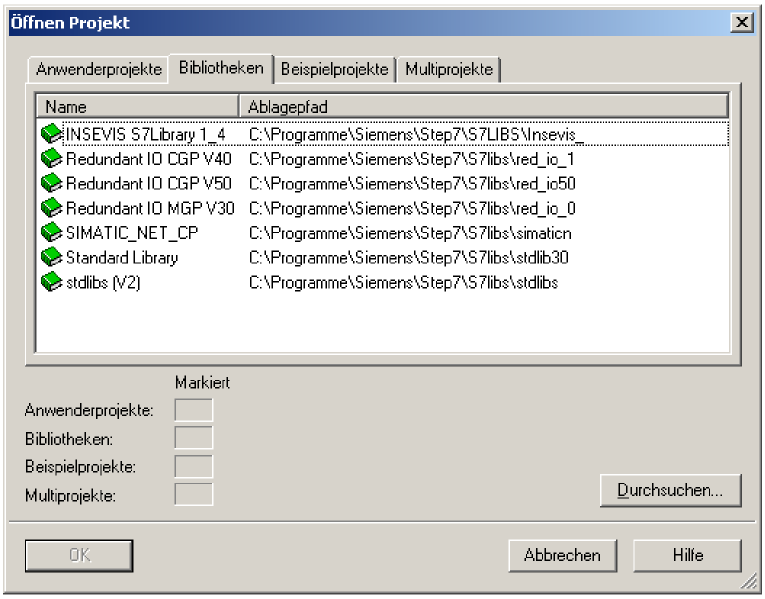
## Import of INSEVIS-blocks into the S7-Project

Attention

All additional INSEVIS- blocks are available as **S7-Library** in the internet at www.insevis.de/de/service for free download. It will be offered always the newest library with the most functions, what needs the regarding firmware in the PLC. (e.g. Insevis_S7-library_from_2_1_0.zip - works from firmware 2.1.0 and higher).

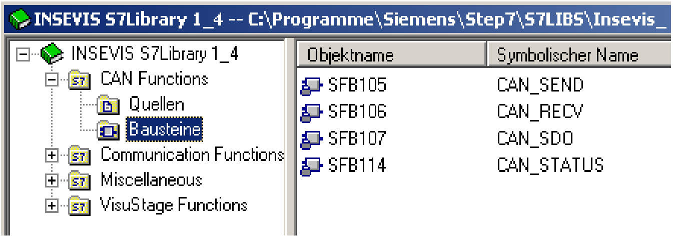
### Installation of the INSEVIS-S7-library in the Simatic-manager

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“



| Name | Ablagepfad |
|---|---|
| INSEVIS S7Library 1_4 | C:\Programme\Siemens\Step7\S7LIBS\Insevis_ |
| Redundant IO CGP V40 | C:\Programme\Siemens\Step7\S7libs\red_io_1 |
| Redundant IO CGP V50 | C:\Programme\Siemens\Step7\S7libs\red_io50 |
| Redundant IO MGP V30 | C:\Programme\Siemens\Step7\S7libs\red_io_0 |
| SIMATIC_NET_CP | C:\Programme\Siemens\Step7\S7libs\simaticn |
| Standard Library | C:\Programme\Siemens\Step7\S7libs\stdlib30 |
| stdlibs (V2) | C:\Programme\Siemens\Step7\S7libs\stdlibs |

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



| Objektname | Symbolischer Name |
|---|---|
| SFB105 | CAN_SEND |
| SFB106 | CAN_RECV |
| SFB107 | CAN_SDO |
| SFB114 | CAN_STATUS |

Additional hints for Win-S7 are to be find in the documents „First Steps“at the service area of the INSEVIS website.

**Note:** There is a VIDEO available for that item at the INSEVIS-service web site

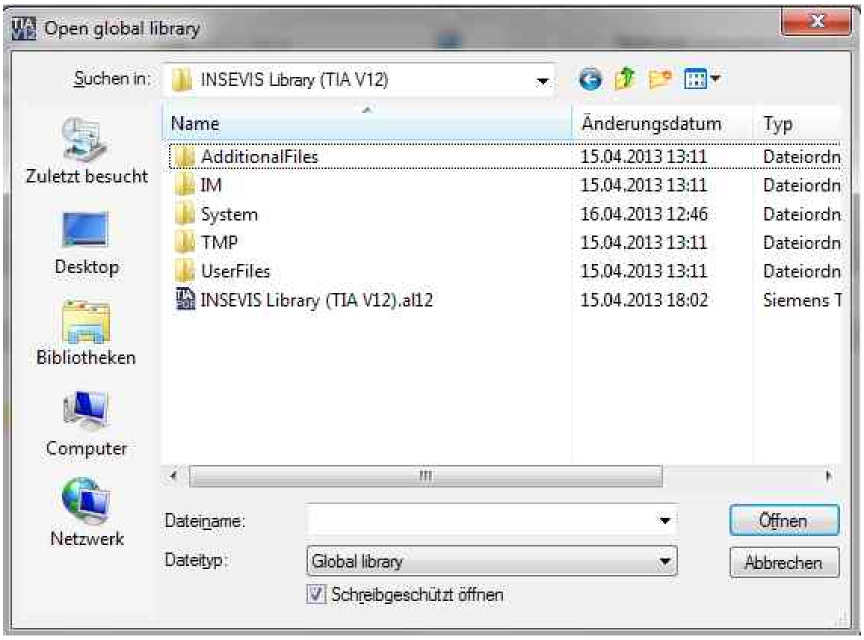
**Installation of the INSEVIS-S7-library in the TIA-Portal**

1. Download latest version  of INSEVIS library for TIA-portal

2. store it and extract it in any folders

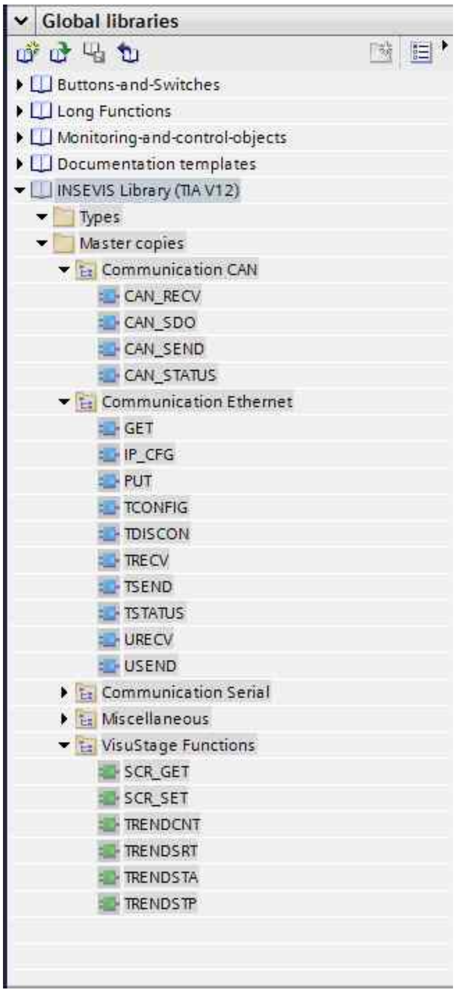
3. Open library “***INSEVIS Library (TIA V2).al12***“ from the referring folder in TIA-Portal as “Global library“.→

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


Open global library
Suchen in: INSEVIS Library (TIA V12)
Zuletzt besucht
Desktop
Bibliotheken
Computer
Netzwerk
Name
Änderungsdatum
Typ
AdditionalFiles 15.04.2013 13:11 Dateiordn
IM 15.04.2013 13:11 Dateiordn
System 16.04.2013 12:46 Dateiordn
TMP 15.04.2013 13:11 Dateiordn
UserFiles 15.04.2013 13:11 Dateiordn
INSEVIS Library (TIA V12).al12 15.04.2013 18:02 Siemens T
Dateiname:
Dateityp: Global library
Öffnen
Abbrechen
☑ Schreibgeschützt öffnen

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

   *→ For more details to „use a library element“ use TIA-Portal online help*



## 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

With the **SFC 208** BACKUP all data (S7-Program, visualization, process data) will be combined to a binary (encryptable) an written to the Micro-SD-card. For that process the CPU will be switched into STOP mode and return, when it is ready.
With this file the end user can create a spare device with similar data (need a similar hardware) without any PC-programs. To secure data a password file can be activated.

→ sample: **Create a spare machine**

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

### Manual program-backup for manufacturers

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

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

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

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

### S7-Program-restore for both kinds of backup

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

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

## 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<br>FALSE= Overwrite existing file |
| CSV | INPUT | BOOL | TRUE= Convert data into text and write as text record (CSV) format.<br>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,<br>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.

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

### Meaning of CSV_STRUCT

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

| First byte | Second byte | Format |
|---|---|---|
| 9 | Data is DATE | |
| | 0 | D#YYYY-MM-DD (e.g D#2013-07-24) |
| | 1 | As decimal integer, 0..65535 = number of days since 1990.01.01 |
| | 2 | As hexadecimal W#16#0 .. W#16#FFFF |
| | 3 | As hexadecimal 0x0000 .. 0xFFFF |
| 10 | Data is TIME_OF_DAY (TOD) | |
| | 0 | TOD#HH:MM:SD.MS (e.g TOD#12:45:23.102)<br>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)<br>d=day<br>h=hour<br>m=minute<br>s=second<br>MS=millisecond<br>if one of time unit is 0, then its not displayed.<br>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)<br>h=hour<br>m=minute<br>s=second<br>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<br>YYYY – year 1990 – 2089<br>MM – month<br>DD – day<br>HH – hour<br>MI – minute<br>SS – second<br>MS – millisecond<br>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 |
| W#16#8000 | 0 | 0 | 1 | File system is not ready (e.g SD card not inserted) |
| W#16#8001 | 0 | 0 | 1 | Could not write to file |
| W#16#8002 | 0 | 0 | 1 | Invalid data type ID number in CSV_STRUCT |
| W#16#8003 | 0 | 0 | 1 | CSV_STRUCT – size is must be even number |
| W#16#822A | 0 | 0 | 1 | Invalid format in FILENAME parameter |
| W#16#8xyy | 0 | 0 | 1 | General error code (S7 compatible) |

## S7-program sample for using the SFC207

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

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

      U     M   1701.3                         // IF Archive data is in process; then
      BEB                                      // 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)<br>Logical base address = 1...16 (connection-ID-No. to the Panel-HMI) |
| ATTRIBUTE | INPUT | BYTE | Attributes<br>Bit0: 1= Backup WITH actual data of Retain-DBs<br>0= Backup WITHOUT actual data of Retain-DBs<br>Bit1: 1= Backup WITH remanent S7-memory<br>0= Backup WITHOUT remanent S7-memory<br>(remanent merker, S7-Timer, counter, operating hours)<br>Bit2: 1= Backup WITH VisuStage RunTime memory<br>0= Backup WITHOUT VisuStage RunTime memory<br>(Message, trend actual buffer, user PINs)<br>Bit3: 1= Newstart PLC after backup end<br>0= NO newstart, PLC stays in STOP mode<br>Bit7: 1= Decrypt backup file WITH password<br>0= Decrypt backup file WITHOUT password |
| PASSWORD | INPUT | ANY | Pointer to Keyword for encryption. Data type must be BYTE or CHAR.<br>Keyword must be min. 5 and max. 16 characters long. |
| RETVAL | OUTPUT | WORD | Status code of job W#16#...<br>7000 = Backup is requested, and in process<br>7001 = HMI- connection is in invalid state (e.g not connected)<br>7002 = Job is not accepted, due to previous job is in process<br>7003 = Connection resource is temporary locked<br>7004 = REQ parameter is FALSE (Backup is not accepted)<br>8001 = Invalid connection ID number<br>8003 = SD card not inserted<br>842A = PASSWORD parameter is not BYTE or CHAR type<br>842B = PASSWORD parameter length is invalid<br>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

# System functions

**Note:**

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

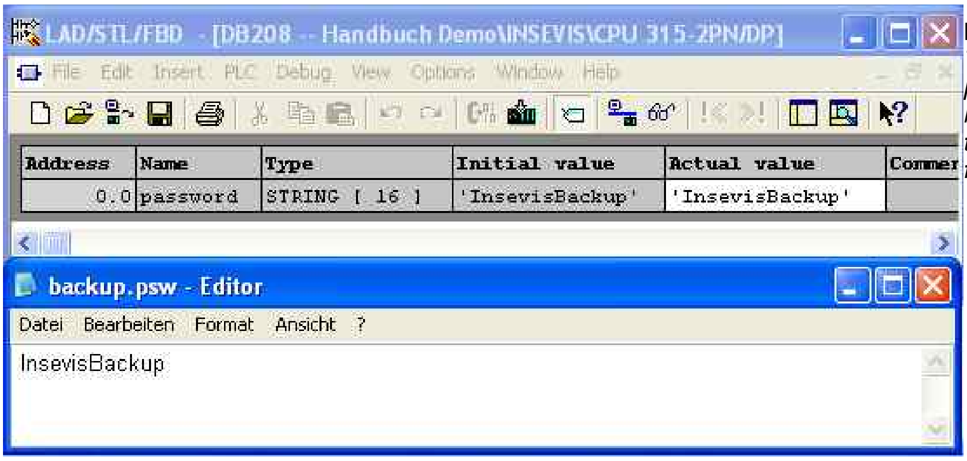
| | |
|---|---|
| EVENT ID | 16#BF00 |
| OB | 16#00 |
| PK | 16#FF |
| DatID1/2 | Result<br>16#0002 = Successful<br>16#0001 = Backup failed |

- 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
```


LAD/STL/FBD - [DB208 -- Handbuch Demo\INSEVIS\CPU 315-2PN/DP]
File Edit Insert PLC Debug View Options Window Help
backup.psw - Editor
Datei Bearbeiten Format Ansicht ?

| Address | Name | Type | Initial value | Actual value | Commen |
|---|---|---|---|---|---|
| 0.0 | password | STRING [ 16 ] | 'InsevisBackup' | 'InsevisBackup' | |

InsevisBackup

**Hint:**
*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) |
| 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
```

## 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..)<br>Copying is not started. |
| W#16#8002 | 0 | Invalid data block number (HIGH_NR is smaller than LOW_NR) |
| W#16#822A | 0 | LOW_NR is invalid. |
| W#16#832A | 0 | HIGH_NR is invalid. |
| W#16#8xyz | 0 | General error code (S7 compatible) |

## S7-program sample for using the SFC210

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

```
      UN    M   1502.0            // If request is not set, then
      BEB                         // end the function

      CALL  SFC  210
       REQ    :=M1502.0           // TRUE = Request to copy data block(s) to load memory
       LOW_NR :=W#16#D1           // Data block number 209 (dec)
       HIGH_NR:=W#16#D1           // only one data block
       RET_VAL:=MW1000            // Return / result code
       BUSY   :=M1702.0           // TRUE = Request is in process

      U     M   1702.0            // if Copy request is in process, then
      BEB                         // end the function
      R     M   1502.0            // reset the request signal

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

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

# System functions

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

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

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

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

| RET_VAL | BUSY | Description |
|---|---|---|
| W#16#0000 | 0 | Successfully copied |
| W#16#7000 | 0 | First call with REQ=0, Copying is not started |
| W#16#7001 | 1 | First call with REQ=1, Copying is started |
| W#16#7002 | 1 | Intermediate call (REQ irrelevant): Copying is in process |
| W#16#8001 | 0 | Other memory manipulation function is active. (e.g PG Block download, delete etc..)<br>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 (e.g. SFC207 “ARCHIVE” …) |
| W#16#8000 | 0 | 0 | 1 | SD card is not inserted |
| W#16#8001 | 0 | 0 | 1 | File is not exist in SD |
| W#16#8002 | 0 | 0 | 1 | Invalid data type in CSV_STRUCT parameter |
| W#16#8003 | 0 | 0 | 1 | Invalid length in CSV_STRUCT parameter |
| W#16#8004 | 0 | 0 | 1 | S7-Process data area should start from even offset (DATA parameter) |
| W#16#8005 | 0 | 0 | 1 | File is empty |
| W#16#8006 | 0 | 0 | 1 | Error occurred on reading the file |
| W#16#8007 | 0 | 0 | 1 | Invalid line / record separator. It should be CRLF. |
| W#16#8008 | 0 | 0 | 1 | Invalid character. It should be only ASCII characters. |
| W#16#8009 | 0 | 0 | 1 | Could not convert the field data. (e.g syntax error ...) |
| W#16#800A | 0 | 0 | 1 | Too big file. Not enough system resource to read file. |
| W#16#822A | 0 | 0 | 1 | FILENAME parameter is invalid. |
| W#16#8xyz | 0 | 0 | 1 | General error code (S7 compatible) |

# System functions

Notes:

- 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
```

## 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<br>0 Panel-PLC (intern LCD)<br>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<br><br>W#16#0000 Copying is in process, if BUSY=TRUE<br>Copying is done, if BUSY=FALSE<br>W#16#7000 No copying in process, if REQ = FALSE<br>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.

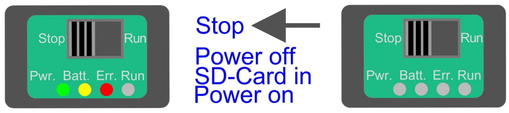
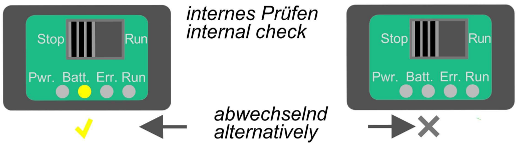
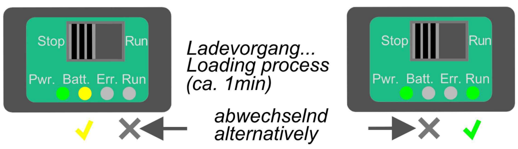
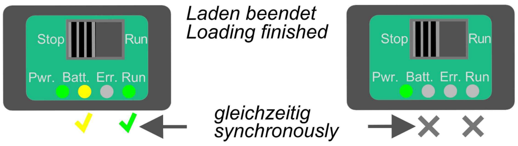
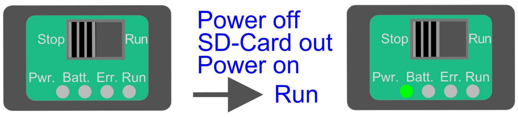
```
      UN    M   1502.2                 // If request is not set, then
      BEB                              // end the function

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

## General reset by hardware

General resetting is possible in 2 ways:

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

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

**Procedure for general reset by hardware:**

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

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

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

Within the next second

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

Within the next second

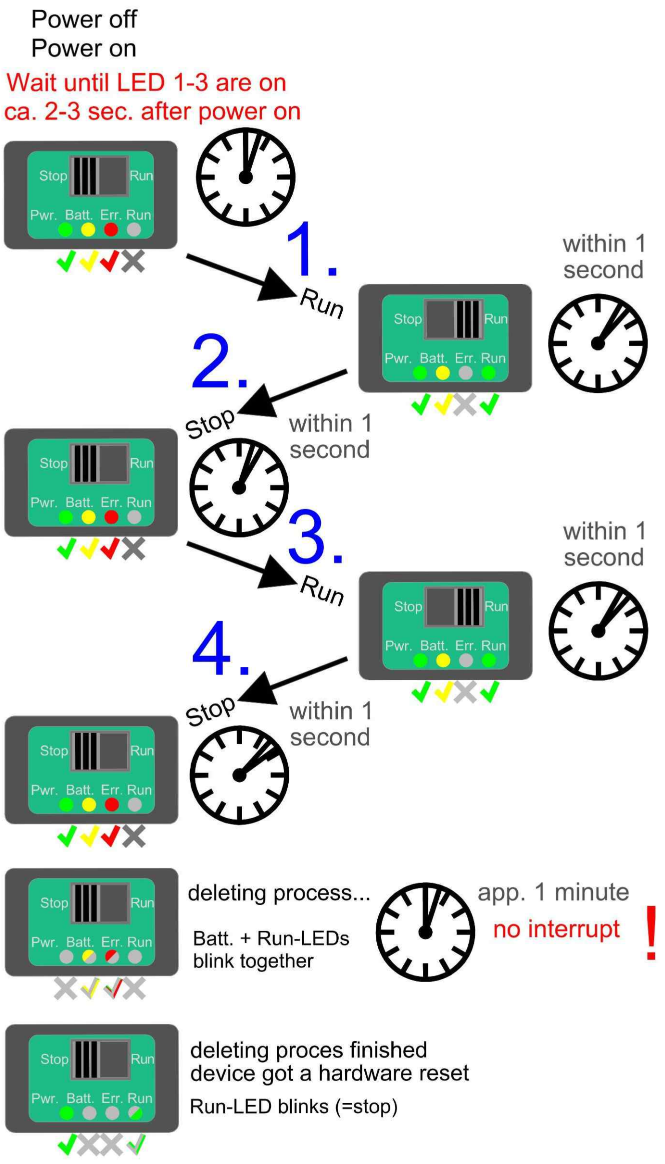
3 Run/Stop-switch in Run
*red „Error"-LED now is off*
*green „Run"-LED now glows*

Within the next second

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

Deleting in operation
**DO NOT power off device**
LEDs „Batt" (yellow) and
„Err." (red) blink together

Deleting finished
„Run"-LED blinks
(device is in Stop mode)



**Hint for RESTORE after Hardware-Reset:**

After this operation the PLC tries to read data from the micro-SD-card. RUN- and BATT-LEDs are blinking together during this time.

If there is no or an empty micro-SD-card plugged in, the user data and the process data will be written over with„0" (general reset).

During the first 3 seconds* is a value for the CPU-V/-P. The larger CPU-T needs longer for booting process, than the red LEDs start after the progress bar has finished (if you have a display, ~ 6-8 seconds), please be patient – you will catch this moment surely.

# Remote access with the software „RemoteStage“

## General

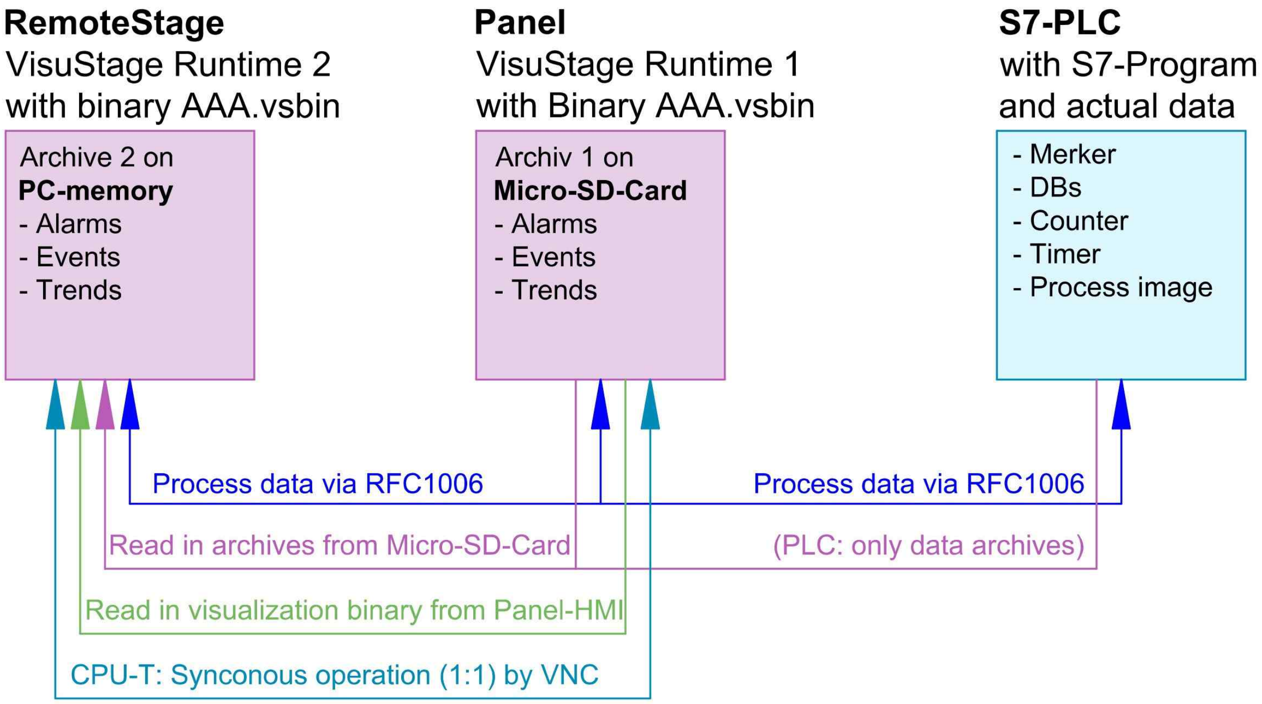
The RemoteStage is a multiple instanciable software, what not needs to be installed from version 1.0.3.3 any more.
It can be kept in a folder on the remote-PC or on a mobile data carrier as **remotestage.exe** (and if you use German program language with the file **remotestage.deu** as well).

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 useres know-how and allows to forward these files towards the final user or operator.

With version 1.0.4.0 it is possible to download the visualization binary directly by Ethernet from the Panel-PLC / Panel-HMI (See right button). This helps to reduce the organization effords by far. In VisuStage you may assign a PIN-code to protect your 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 remote visualization and account to external S7-PLCs as well. This is a function nearly similar to WinCC runtime of a 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:

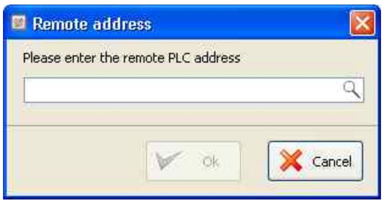
***[path of RemoteStage]\remotestage.exe*** ***/r=[IP-address of the remote-PLC in the same sub net]***
***/V=[path of visualization binary]\complete file name]***
***/start***

### 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.)*

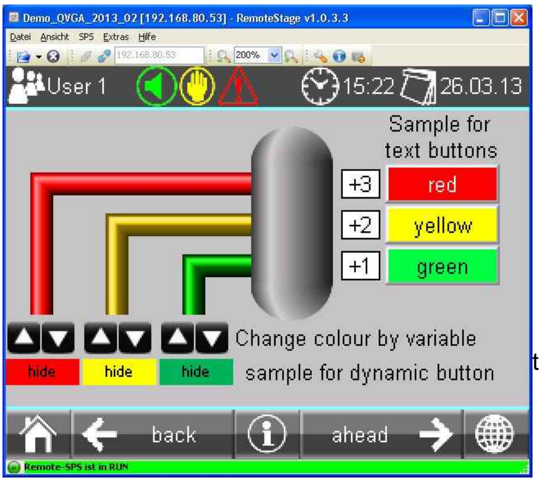
This address will be linked with this binary. So it is not necessary to re-type it again.


Remote address

Please enter the remote PLC address

# Remote access with the software „RemoteStage“

## Using and setting up the remote screen


Demo_QVGA_2013_02 [192.168.80.53] - RemoteStage v1.0.3.3
Datei Ansicht SPS Extras Hilfe
192.168.80.53
200%
User 1
15:22
26.03.13
Sample for text buttons
+3
red
+2
yellow
+1
green
Change colour by variable
hide
hide
hide
sample for dynamic button
back
ahead
Remote-SPS ist in RUN

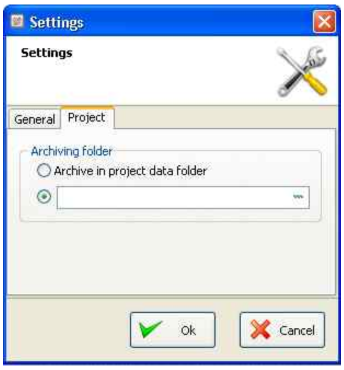
The PopUp-menues are self explaining.
The actual remote state is shown on the bottom of the screen.

| | |
|---|---|
| **Yellow**: | connecting (set up connection) |
| **Green**: | connected (online) and PLC in RUN |
| **Red**: | connected (online) and PLC in STOP |

The switching between online/offline can be done by the green button beside the IP-address line or by the menu „PLC“.

**Hint:** Referring the system data (like buzzer, archives and time) always the PC-system data will be used, not these of the remote device.

## Storing the remote (on the PC) archived data


Settings

Settings

General | Project

Archiving folder

○ Archive in project data folder

●

Ok | Cancel

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 visualization binary is kept) or
- at another path.

## Converting archive data into csv-files

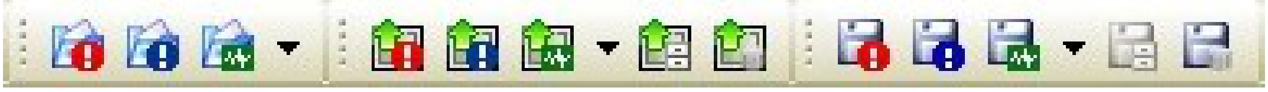
These data can be archived by these products on the Micro-SD-card:

| | |
|---|---|
| **Messages** (alarms, events) | by all Panel-PLCs and Panel-HMIs<br>→ by activating the archive funtion „archiving“ in the VisuStage-project |
| **Trends** (with all channels) | by all Panel-PLCs and Panel-HMIs<br>→ by activating the archive funtion „archiving“ in the VisuStage-project |
| **Recipes** (with all records) | by all Panel-PLCs and Panel-HMIs<br>→ will always be stored / archived on the Micro-SD-card |
| **Archiv**-(data blocks) | by all Panel-PLCs and Panel-HMIs<br>→ will be configured by SFCs in the S7-program |

Most of the data is stored as a binary on the Micro-SD-card and can be converted into csv-format by the RemoteStage only. Therefore you may open a special data converting mode within the RemoteStage program by this icon (right).

**CAUTION:** Always open the binary suiting to your visualization on the remote device, or upload it as binary!

In the newly appeared PC-window you may find different groups of icons with different functions:



Displays archive by-inserting the Micro-SD-card in PC-card slot
left->right
- Alarm archive
- Event archive
- Trend archive

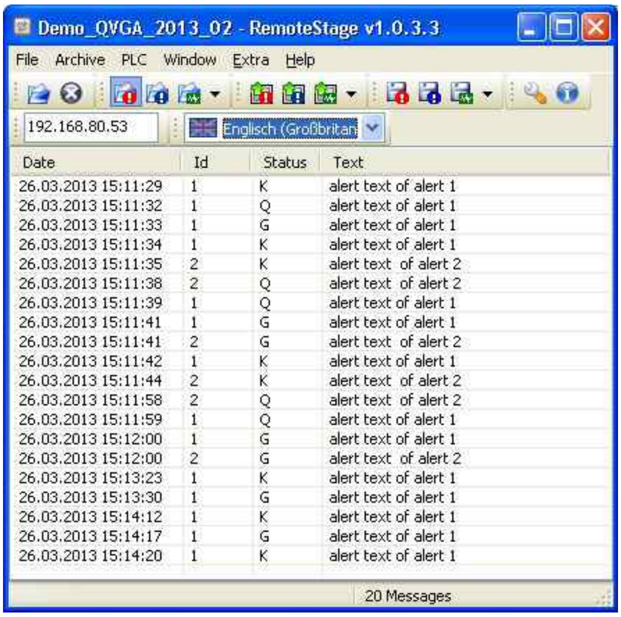
Read in the archives over Ethernet by IP-address directly into PC
left->right:
- Alarm archive
- Event archive
- Trend archive
- Recipe archive
- Data(DB-)archive

Export / save the archive data to thePC / into the PC-network
left->right:
- Alarm archive
- Event archive
- Trend archive
- Recipe archive
- Data(DB-)archive

# Remote access with the software „RemoteStage“

### Sample for message archive viewer and csv-export

*Below: View of the event archive in the RemoteStage*



Demo_QVGA_2013_02 - RemoteStage v1.0.3.3

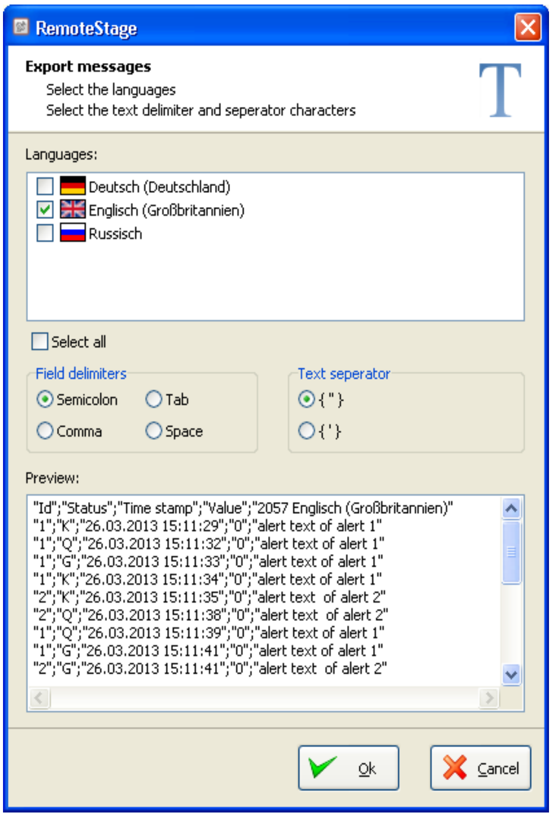
192.168.80.53 · Englisch (Großbritan

| Date | Id | Status | Text |
|---|---|---|---|
| 26.03.2013 15:11:29 | 1 | K | alert text of alert 1 |
| 26.03.2013 15:11:32 | 1 | Q | alert text of alert 1 |
| 26.03.2013 15:11:33 | 1 | G | alert text of alert 1 |
| 26.03.2013 15:11:34 | 1 | K | alert text of alert 1 |
| 26.03.2013 15:11:35 | 2 | K | alert text of alert 2 |
| 26.03.2013 15:11:38 | 2 | Q | alert text of alert 2 |
| 26.03.2013 15:11:39 | 1 | Q | alert text of alert 1 |
| 26.03.2013 15:11:41 | 1 | G | alert text of alert 1 |
| 26.03.2013 15:11:41 | 2 | G | alert text of alert 2 |
| 26.03.2013 15:11:42 | 1 | K | alert text of alert 1 |
| 26.03.2013 15:11:44 | 2 | K | alert text of alert 2 |
| 26.03.2013 15:11:58 | 2 | Q | alert text of alert 2 |
| 26.03.2013 15:11:59 | 1 | Q | alert text of alert 1 |
| 26.03.2013 15:12:00 | 1 | G | alert text of alert 1 |
| 26.03.2013 15:12:00 | 2 | G | alert text of alert 2 |
| 26.03.2013 15:13:23 | 1 | K | alert text of alert 1 |
| 26.03.2013 15:13:30 | 1 | G | alert text of alert 1 |
| 26.03.2013 15:14:12 | 1 | K | alert text of alert 1 |
| 26.03.2013 15:14:17 | 1 | G | alert text of alert 1 |
| 26.03.2013 15:14:20 | 1 | K | alert text of alert 1 |

20 Messages

(Displays all archived alarm messages of the Panel-PLC / Panel-HMI with the IP-address 192.168.80.50)

*Below: Export of the event archive as csv-file*



RemoteStage

**Export messages**

Select the languages

Select the text delimiter and seperator characters

Languages:

- □ Deutsch (Deutschland)
- ☑ Englisch (Großbritannien)
- □ Russisch

□ Select all

Field delimiters: ● Semicolon ○ Tab ○ Comma ○ Space

Text seperator: ● {"} ○ {'}

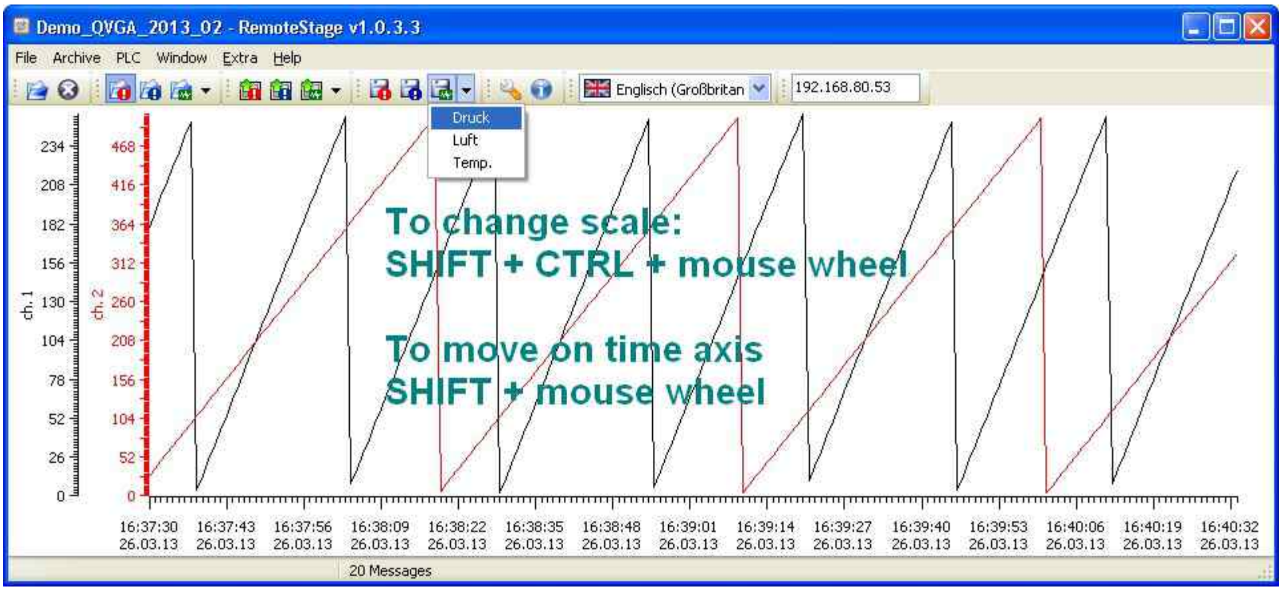
Preview:

```
"Id";"Status";"Time stamp";"Value";"2057 Englisch (Großbritannien)"
"1";"K";"26.03.2013 15:11:29";"0";"alert text of alert 1"
"1";"Q";"26.03.2013 15:11:32";"0";"alert text of alert 1"
"1";"G";"26.03.2013 15:11:33";"0";"alert text of alert 1"
"1";"K";"26.03.2013 15:11:34";"0";"alert text of alert 1"
"2";"K";"26.03.2013 15:11:35";"0";"alert text  of alert 2"
"2";"Q";"26.03.2013 15:11:38";"0";"alert text  of alert 2"
"1";"Q";"26.03.2013 15:11:39";"0";"alert text of alert 1"
"1";"G";"26.03.2013 15:11:41";"0";"alert text of alert 1"
"2";"G";"26.03.2013 15:11:41";"0";"alert text  of alert 2"
```

**CAUTION:**
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.

### Sample for trend archive viewer and csv-export

*Below: View of the trend „Druck“ archive in the RemoteStage*



Demo_QVGA_2013_02 - RemoteStage v1.0.3.3

To change scale:
SHIFT + CTRL + mouse wheel

To move on time axis
SHIFT + mouse wheel

Above figure displays all archived trend with name „Druck“of the Panel-PLC / Panel-HMI with the IP-address 192.168.80.30. The time will be displayed in the Language format you assigned on your PC (here in German = DE)

**CAUTION:**
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.

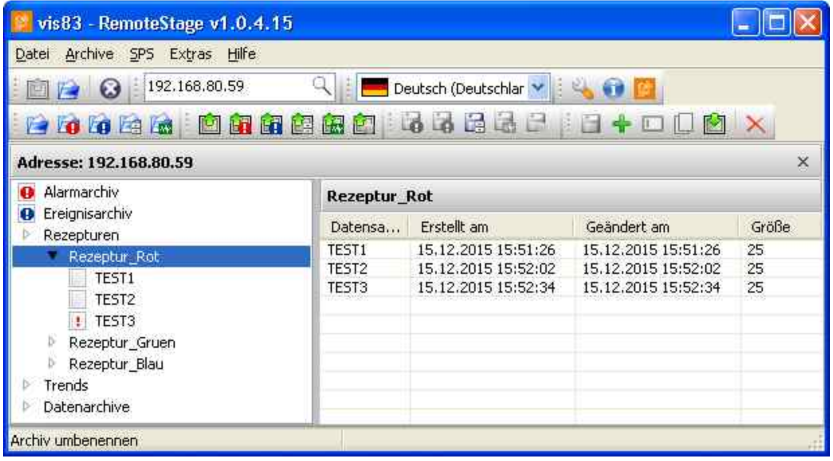
# Remote access with the software „RemoteStage“

## Recipe import, display, converting, editing and export

Recipe archive data from Micro-SD-card can be

- read in / uploaded (→ Select the whole root-folder of the Micro-SD-card !),
- displayed (self explaining)
- converted into csv
- modified with RemoteStage itself or a calculating PC-program and
- written back / downloaded into the PLC

*Left below: list of archived recipes and its records*



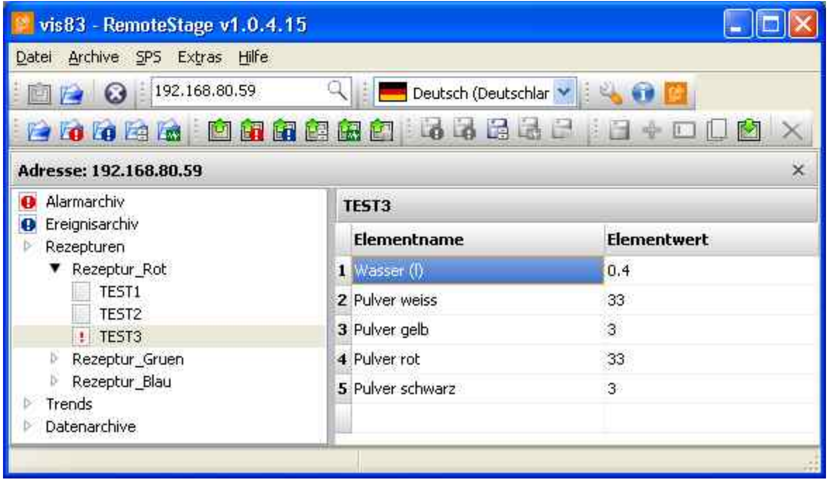
Adresse: 192.168.80.59

- Alarmarchiv
- Ereignisarchiv
- Rezepturen
  - Rezeptur_Rot
    - TEST1
    - TEST2
    - ! TEST3
  - Rezeptur_Gruen
  - Rezeptur_Blau
- Trends
- Datenarchive

Rezeptur_Rot

| Datensa... | Erstellt am | Geändert am | Größe |
|---|---|---|---|
| TEST1 | 15.12.2015 15:51:26 | 15.12.2015 15:51:26 | 25 |
| TEST2 | 15.12.2015 15:52:02 | 15.12.2015 15:52:02 | 25 |
| TEST3 | 15.12.2015 15:52:34 | 15.12.2015 15:52:34 | 25 |

*right below: list of records and its elements*



Adresse: 192.168.80.59

- Alarmarchiv
- Ereignisarchiv
- Rezepturen
  - Rezeptur_Rot
    - TEST1
    - TEST2
    - ! TEST3
  - Rezeptur_Gruen
  - Rezeptur_Blau
- Trends
- Datenarchive

TEST3

| | Elementname | Elementwert |
|---|---|---|
| 1 | Wasser (l) | 0.4 |
| 2 | Pulver weiss | 33 |
| 3 | Pulver gelb | 3 |
| 4 | Pulver rot | 33 |
| 5 | Pulver schwarz | 3 |

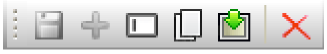
The record „TEST3“ was modifyed in the RemoteStage and this is shown by an „!“ .
After downloading into the remote device this sign disappears.

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.

With these icons record can be renamed, copied, downloaded into the remote device and deleted there.



**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

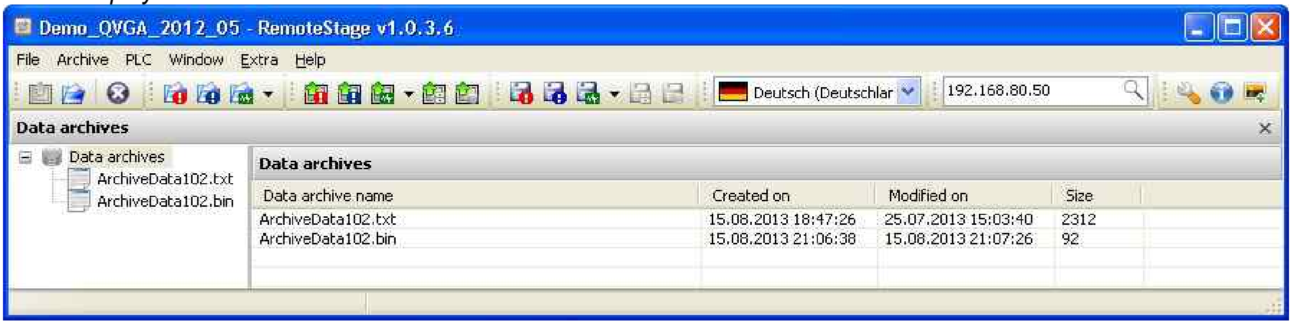
Any data can be archived with the **SFC 207** ARCHIVE onto the Micro-SD-card (binary or csv-data)and read back with the **SFC 209** READFILE (see system functions).

### Data archive display and storing

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

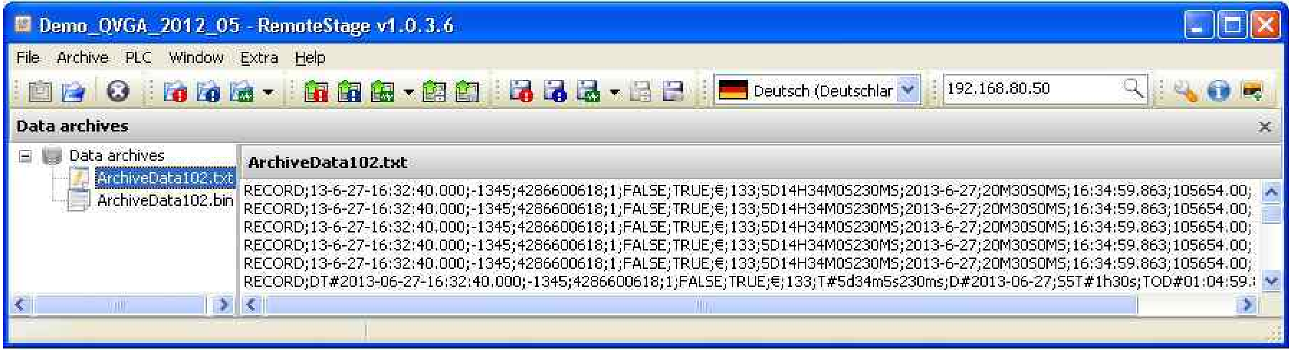
The format of the csv-file will be assigned by the S7-program (SFC207) and is not configurable by the RemoteStage. This can be done later on with any table calculation program by customer (because only he knows the format).

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



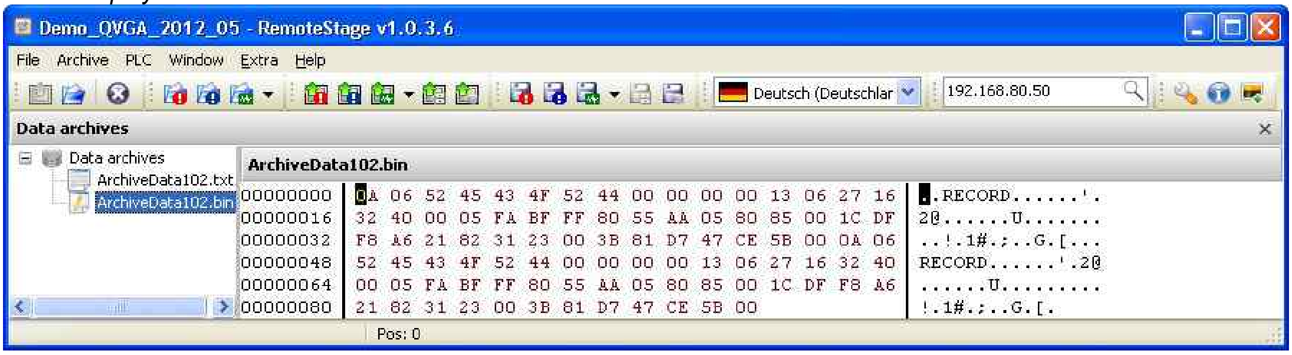
| Data archive name | Created on | Modified on | Size |
|---|---|---|---|
| ArchiveData102.txt | 15.08.2013 18:47:26 | 25.07.2013 15:03:40 | 2312 |
| ArchiveData102.bin | 15.08.2013 21:06:38 | 15.08.2013 21:07:26 | 92 |

*below: display of the data archives in txt-format available on the Micro-SD-card*



```
RECORD;13-6-27-16:32:40.000;-1345;4286600618;1;FALSE;TRUE;€;133;5D14H34M0S230MS;2013-6-27;20M30S0MS;16:34:59.863;105654.00;
RECORD;13-6-27-16:32:40.000;-1345;4286600618;1;FALSE;TRUE;€;133;5D14H34M0S230MS;2013-6-27;20M30S0MS;16:34:59.863;105654.00;
RECORD;13-6-27-16:32:40.000;-1345;4286600618;1;FALSE;TRUE;€;133;5D14H34M0S230MS;2013-6-27;20M30S0MS;16:34:59.863;105654.00;
RECORD;13-6-27-16:32:40.000;-1345;4286600618;1;FALSE;TRUE;€;133;5D14H34M0S230MS;2013-6-27;20M30S0MS;16:34:59.863;105654.00;
RECORD;13-6-27-16:32:40.000;-1345;4286600618;1;FALSE;TRUE;€;133;5D14H34M0S230MS;2013-6-27;20M30S0MS;16:34:59.863;105654.00;
RECORD;DT#2013-06-27-16:32:40.000;-1345;4286600618;1;FALSE;TRUE;€;133;T#5d34m5s230ms;D#2013-06-27;S5T#1h30s;TOD#01:04:59.
```

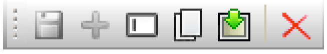
*below: display of the data archives as binaries available on the Micro-SD-card*



```
00000000  0A 06 52 45 43 4F 52 44 00 00 00 00 13 06 27 16  ..RECORD......'.
00000016  32 40 00 05 FA BF FF 80 55 AA 05 80 85 00 1C DF  2@......U.......
00000032  F8 A6 21 82 31 23 00 3B 81 D7 47 CE 5B 00 0A 06  ..!.1#.;..G.[...
00000048  52 45 43 4F 52 44 00 00 00 00 13 06 27 16 32 40  RECORD......'.2@
00000064  00 05 FA BF FF 80 55 AA 05 80 85 00 1C DF F8 A6  ......U.........
00000080  21 82 31 23 00 3B 81 D7 47 CE 5B 00              !.1#.;..G.[.
```

Pos: 0

With these icons data archives will be renamed, copied and - from Remote Stage V 1.0.4.18- downloaded into remote device or deleted.



# Remote access with the software „RemoteStage“

## Command line of RemoteStage (Viewing mode)

RemoteStage can be inserted in „planned tasks“ of the remote PC. Here are available these special functions.

### Automatic connecting to remote panel after starting PC

With his function a binary of the visualization is called and linked with the predefined IP-address of the remote panel (if this is available online only)

```
remotestage.exe /start /r=<remote> /v=<project binary>
```

**Hint:** If the binary name contains space-signs it is necessary to write it in “ like „C:\Test 1 2 3.vsbin“)

### Automatic full screen after starting PC

With his function a binary of the visualization is called and the RemoteStage changes into full screen at the remote PC after automatic starting (Screen stays dark, because no IP-address was assigned here...)

```
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.

```
remotestage.exe /v=d:\visu\process.vsbin /r=192.168.80.50 /p=4500
```

**Hint:**
The default port number for TCP is port 102 at the PLCs.

# Remote access with the software „RemoteStage“

## Command line of RemoteStage (Archive mode)

RemoteStage can be used in batch process from command line with command line parameters, to save archives somewhere with a self defined name.

In the PC-OS Windows® batch files can be done **time controlled** in the task manager
Start 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

```
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)

***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 number which is configured in VisuStage. It is only used in "trend_archive" mode.

### Sample

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 can execute from command

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

```
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:

```
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:

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

# Remote access with the software „RemoteStage“

## Write back (download) recipe archive into remote device

With this function the uploaded (and maybe modified) recipe data will be written back (downloaded) into the remote device.
(from rev. 1.0.4.17 and PLC firmware 2.3.9 (all CPUs) or HMI firmware 1.2.7 CPU-V/-P and 1.2.6 CPU-T)

**Attention:** The structure of the receipes in the remote device may not be modified between upload and download.
(No adding, deleting elements or changing the row of order)

## Command line

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

## Parameters

| | |
|---|---|
| ***project binary*** | Binary 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

```
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

```
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

```
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:

```
C:\>remotestage.exe /mode=data_archive /r=192.168.80.50
/archive=Test-DB /o=D:\archive\Testdaten.csv
```

## Write back (download) data (DB-) archive into remote device

With this function the uploaded (and maybe modified) data (DB-) archives will be written back (downloaded) into the remote device. (from rev. 1.0.4.17 and PLC firmware 2.3.9 (all CPUs))

**Attention:** The programmer itself is responsible to keep the structure of the data (DB-) archives. If they will be modified, all referring reports must be adapted too!

### Command line

```
remotestage.exe /mode=data_download /archive:=<archive file(s)>
```

### Parameters

***archive file(s)*** S7 User data archive file(s) to download to remote PLC

### Samples

Following command line downloads ONE archive file "default_cfg.csv" from the folder D:\configs into the remote device with the IP address 192.168.80.50

```
remotestage.exe /mode=data_download /archive=d:\configs\default_cfg.csv /r=192.168.80.50
```

Following command line downloads ALL archive files from the folder D:\configs into the remote device with the IP address 192.168.80.50

```
remotestage.exe /mode=data_download /archive=d:\configs\*.* /r=192.168.80.50
```

# Remote access with the software „RemoteStage“

## Return codes

Return codes of remotestage.exe indicate the result of operation.

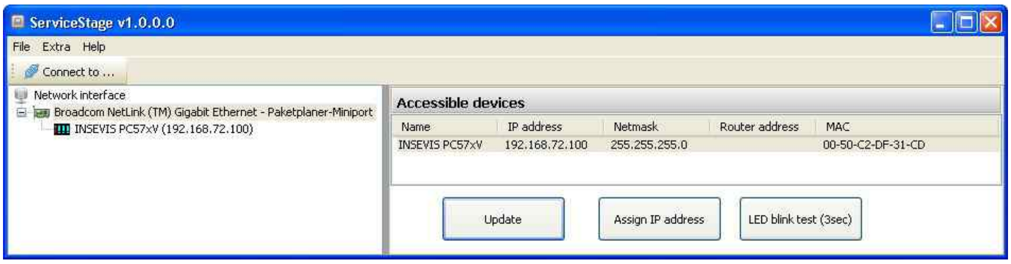
| Return code | Meaning |
|---|---|
| 0 | Successfully read out and saved. |
| 1 | Invalid parameter or mandatory parameters are missing |
| 2 | Project binary file is invalid or not found |
| 3 | Remote IP address is invalid |
| 4 | Could not connect to remote device |
| 5 | SD card not inserted on remote device |
| 6 | Archive file not found on remote device |
| 7 | Remote device does not support archive file read service |
| 8 | Invalid archive data |
| 9 | Not enough system resource |
| 11 | Not enough system resource on remote device |
| 14 | Could not write to in give <output file> |
| 15 | Connection is broken |

Sample of RemoteStage usage in batch process:

```
@remotestage /mode=alarm_archive /r=192.168.80.50 /o=d:\archive\alarm.csv /v=d:\visu\myvisu.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 and to change their state or program data. 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.



Network interface
- Broadcom NetLink (TM) Gigabit Ethernet - Paketplaner-Miniport
  - INSEVIS PC57xV (192.168.72.100)

Accessible devices

| Name | IP address | Netmask | Router address | MAC |
|---|---|---|---|---|
| INSEVIS PC57xV | 192.168.72.100 | 255.255.255.0 |  | 00-50-C2-DF-31-CD |

Therefore you choose the network adapter of your PC (This one, what makes the network access to the CPUs). Select one of the CPUs displayed in the left window and press **„Update“**.
The **„LCD-blink test“** helps to identify the selected CPU. (yellow battery-LED blinks for 3 seconds). Also you can assign a new IP-address, Net mask and Router-IP-address of the selected CPU here.

## Function overview

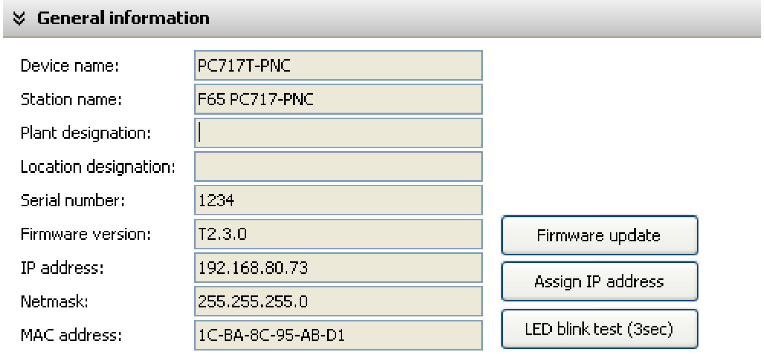
### General information

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.

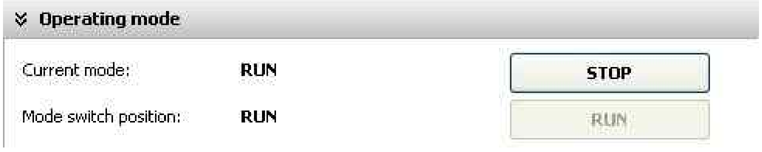
**Only for CPU-T:**
Firmwareupdate possible by Ethernet



| | |
|---|---|
| Device name: | PC717T-PNC |
| Station name: | F65 PC717-PNC |
| Plant designation: | |
| Location designation: | |
| Serial number: | 1234 |
| Firmware version: | T2.3.0 |
| IP address: | 192.168.80.73 |
| Netmask: | 255.255.255.0 |
| MAC address: | 1C-BA-8C-95-AB-D1 |

### Operating mode

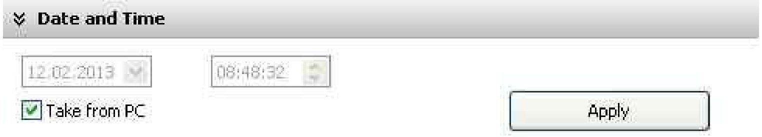
Here you can change the operating mode of you CPU between RUN and STOP (not for Panel-HMI). This change must be confirmed in an extra window.



| | |
|---|---|
| Current mode: | RUN |
| Mode switch position: | RUN |

### 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“**.

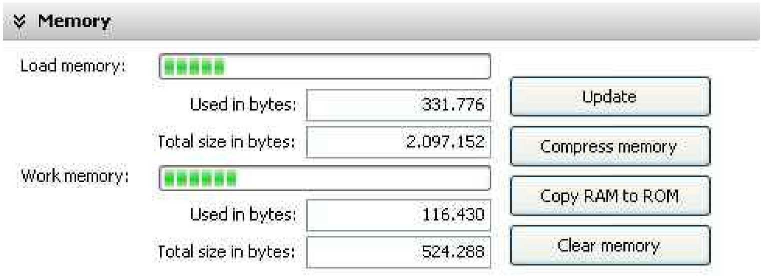


12.02.2013 08:48:32
☑ Take from PC

# 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.



Load memory:
Used in bytes: 331.776
Total size in bytes: 2.097.152

Work memory:
Used in bytes: 116.430
Total size in bytes: 524.288

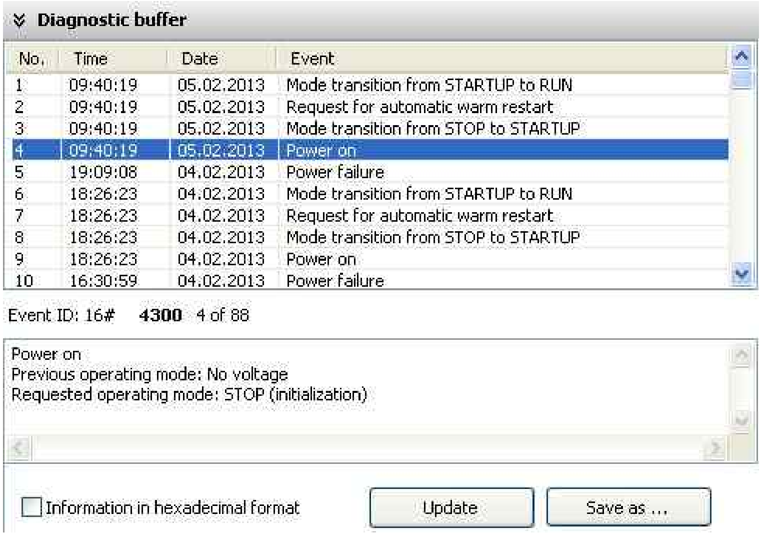
## Diagnostic buffer

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).



| No. | Time | Date | Event |
|---|---|---|---|
| 1 | 09:40:19 | 05.02.2013 | Mode transition from STARTUP to RUN |
| 2 | 09:40:19 | 05.02.2013 | Request for automatic warm restart |
| 3 | 09:40:19 | 05.02.2013 | Mode transition from STOP to STARTUP |
| 4 | 09:40:19 | 05.02.2013 | Power on |
| 5 | 19:09:08 | 04.02.2013 | Power failure |
| 6 | 18:26:23 | 04.02.2013 | Mode transition from STARTUP to RUN |
| 7 | 18:26:23 | 04.02.2013 | Request for automatic warm restart |
| 8 | 18:26:23 | 04.02.2013 | Mode transition from STOP to STARTUP |
| 9 | 18:26:23 | 04.02.2013 | Power on |
| 10 | 16:30:59 | 04.02.2013 | Power failure |

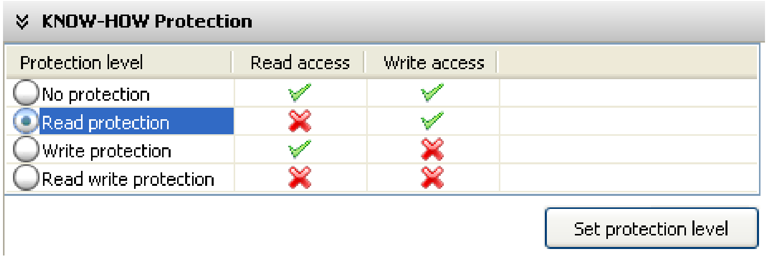
Event ID: 16# **4300** 4 of 88

Power on
Previous operating mode: No voltage
Requested operating mode: STOP (initialization)

☐ Information in hexadecimal format

## Know-how-protection

With version 1.0.1.1 can be set different levels of to protectc the S7-program against unauthorized reading ord writing. Therefore the PLC will be switchet to STOP-mode.



| Protection level | Read access | Write access |
|---|---|---|
| ○ No protection | ✓ | ✓ |
| ● Read protection | ✗ | ✓ |
| ○ Write protection | ✓ | ✗ |
| ○ Read write protection | ✗ | ✗ |

**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.

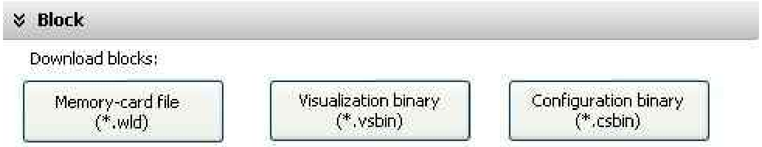
**Read/Write protection:** The S7-program blocks (DBs) can no more be read or manipulated. Only the Panel-HMIs can still read out the process data to visualize it. This protection level can only be reset by general reset by hardware directly on the device. Than the S7-program is deleted in the PLC and the unprotected original program may be downloaded into the PLC again.

# Working with the software „ServiceStage“

## Block Update

This function makes it possible to download
- the S7-user program as MemoryCard-file ***.wld**
- the INSEVIS-binary for visualization file as ***.vsbin** and
- the INSEVIS-configuration file as ***.csbin**

via the network into the CPU


Block
Download blocks:
Memory-card file (*.wld)
Visualization binary (*.vsbin)
Configuration binary (*.csbin)

By this software **„ServiceStage“** it is possible to give away your data in a wld (S7) or binary (INSEVIS) format to those, who care for diagnostics, maintenance or update. You can keep your source data in your company and must not fear to be copied.

* 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

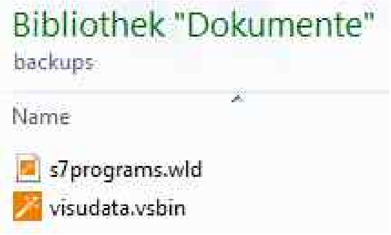
**ATTENTION:** When inserting system data,
- they shall be uploaded from a PLC with included ConfigStage-configuration (Menue: „PLC → Upload Station to PG“), because the are “ConfigStage-configured” already or
- the system data of the SimaticManager overwrite the ConfigStage-system system configuration.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

Attention

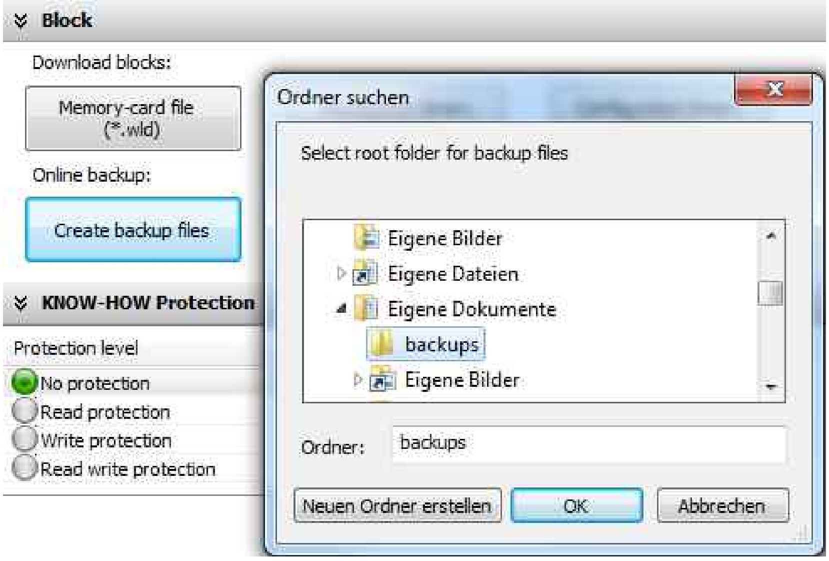
**ATTENTION:** When using Block Upload for restore
- With firmware 2.3.6 a Restore can only bedone by WLD- and VSBIN-file. CSBIN-file will be ignored. So the WLD-file MUST CONTAIN these system data, who where downloaded with CSBIN before.
- If the WLD-file does not contain system data, program stops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

## Online Backup

This function creates an WLD-file with all the S7-system data and all OB, DB, FB, SFB and SFC and copies this file with the visualization binary file into a folder „backups“.



With these data S7-programs and visualizations can be updatet by a Micro-SD-card only – complete without any PC. (more at „System functions“, chapter „Data backup and restore“)



Select root folder for backup files

Ordner: backups

Attention

**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.

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 9001:2008