TURCK

Industrial
Automation

piconet®–
USER MANUAL FOR PROFIBUS-DP

Sense it! Connect it! Bus it! Solve it!

Edition 05/2011

TURCK
Industrial Automation

Prior to installation

- Disconnect the device from the power supply
- Protect against an accidental restart
- Verify safe isolation from the supply
- Cover or separate adjacent live components by a barrier.
- Follow the specific mounting instructions of the device
- Only appropriately qualified personnel according to EN 50 110-1/2- (VDC 0105 part 100) is permitted to work with the device/system.
- Please ensure that you are free of static charge prior to touching the device when carrying out installation.
- Connecting cables and signal lines must be installed in such a manner that inductive and capacitive interference do not impair the automation function.
- Install automation equipment and related operating elements in such a way that they are protected against unintentional operation.
- In order to ensure that a signal line or wire breakage cannot trigger an undefined state of the automation equipment, it is required to implement hard and software safety measures for reliable I/O interfacing.
- Safe electrical isolation of the low voltage for the 24 V supply must be ensured. Use power supply units compliant with IEC 60 364-4-41 or HD 384.4.41 S2 (VDE 0100 part 410) only.
- Fluctuations or deviations of the mains supply from the rated value may not exceed the tolerance limits specified in the technical data.
- Devices for cabinet or cubicle mounting as well as desktop or portable units may only be used and operated after enclosure.
- Measures have to be taken to ensure correct restart of a program which has been interrupted by voltage dips or failures. It must be ensured that such events cannot result in dangerous operating states, even if only present for a short time. If necessary, an emergency stop must be carried out.
- The electrical installation has to be carried out in compliance with the applicable regulations (e.g. concerning cable cross sections, fuses, PE connection etc.)
- Transport, installation, set-up and maintenance may only be carried out by qualified staff (observe IEC 60 364 or HD 384 or DIN VDE 0100 and national work safety regulations).
- It is required to keep all covers and doors closed during operation.

Table of Contents

1 How to work with this manual

1.1 Introduction

Attention
It is indispensable to lease this section because the safety in dealing with electrical equipment should not be left to chance.

This manual contains all information pertaining to safe and proper operation of *piconet*® I/O modules for PROFIBUS-DP. It was specially edited for qualified staff with the required specialised knowledge.

1.1.1 Correct usage to the intended purpose

Danger
The devices described in this manual may only be used in such applications described in the technical sections of the manual and only in conjunction with certified external devices and components.

Correct and safe operation of the devices relies on appropriate transport and storage, correct set-up and installation as well as careful operation and maintenance.

1.1.2 Projecting guidelines/Product installation

Danger
It is indispensable to observe the applicable safety and accident prevention regulations of the specific application.

1.2 Meaning of the symbols used

Danger

This sign is placed next to a warning indicating the presence of a hazard. This can relate to personal injury as well as to system damage (hardware and software).

The user should interpret this symbol as follows: exercise extreme caution.

Attention

This sign is placed next to a warning indicating a potential hazard.
This can relate to personal injury as well as to system (hardware and software) and equipment damage.

Note

This sign is located next to general hints providing important information on individual or stepwise work procedures.

These hints may facilitate work and possibly help to avoid excess work resulting from faulty proceedings.

1.3 Release status and versions

1.3.1 Documentation

This documentation pertains to the hardware and firmware status at the time of editing this manual. The features of the *piconet®* systems are continuously developed and improved.

Changes in the documentation can be taken from
Modification index of this manual.

1.3.2 Firmware and hardware status

Downward compatibility

The modules of the *piconet®* series are downward compatible.
Older module cannot, however, feature the same characteristics than newer module versions. However, existing characteristics have been retained so that older modules can always be replaced with new ones.

The documentation describes the differences between the modules.

The firmware and hardware status of the *piconet®* modules can be taken from the version number printed on the side of the module. The version number can be identified by the prefix "D".

Table 1-1: Firmware and hardware status

Indication on module	Explanation	Example
D. kkjjxyzu		D.22011501
kk	Calendar week	Calendar week 22
jj	Year	of the year 2001
x	Firmware bus board	Firmware bus, version 1
y	Hardware bus board	Hardware, version 5
z	Firmware I/O board	Firmware I/O, 0 (no firmware needed for this board)
u	Hardware I/O board	Hardware I/O, version 1

1.4 List of Revisions

In comparison to the previous manual edition, the following changes/ revisions have been made:

Table 1-2: List of revisions

Chapter	Subject/ Description	new	changed
9	– IP-Link diagnosis via acyclic services (S7 and VT250) (page 10-1)	X	

Note
The publication of this manual renders all previous editions invalid.

2 The *piconet*® system

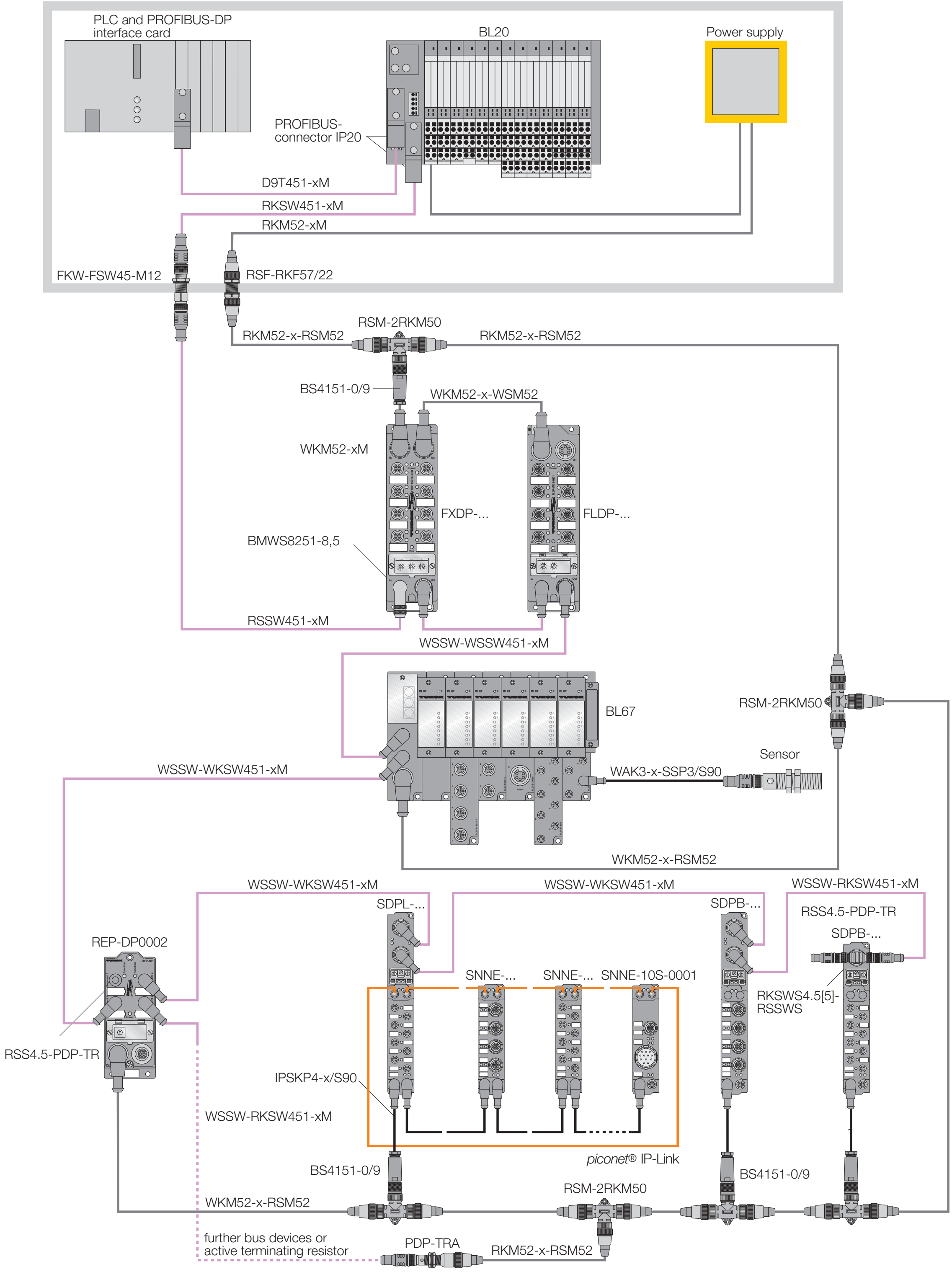
2.1 System overview

piconet®, the "smallest“ bus system within the TURCK programme, offers I/O modules with dimensions of only 210/175/126 x 30 x 26.5 mm, ideally suited for application in serial machine engineering:

- the coupling modules as the interface between the fieldbus and the *piconet*® fibre-optic network,
- the extension modules connected to the coupling module
  as well as
- the stand-alone modules for direct connection to the fieldbus.

All connections are screw connections and feature protection degree IP67.

Figure 2-1: System overview



2.1.1 Coupling and stand-alone modules

The stand-alone and coupling modules are connected directly to the respective fieldbus.

2.1.2 Extension network IP-Link

Up to 120 I/O **extension modules** can be interconnected via a **coupling module** and via the **IP-Link**, an extremely reliable internal bus based on fibre-optic transmission, and integrated into the higher level fieldbus as a single unit. Thus the user can determine any kind of I/O configuration that matches his application.

2.1.3 Combined modules

In order to obtain the highest flexibility, novel 16-channel digital combined I/O modules have been developed. The module channels can be used either as inputs or outputs. The separate circuitry ensures the isolated supply of the inputs and outputs. These combined modules enable precise modular construction of the system. The modules are powered via an M8 connector.

Note

A detailed description of the complete system, the individual *piconet*® modules and the extension network IP-Link is contained in the bus-independent manual "*piconet*® I/O Modules“, document number: D300778.

3 PROFIBUS-DP

3.1 System description

PROFIBUS is a manufacturer-independent and open fieldbus standard for a wide area of applications in factory and process automation. Manufacturer independence and openness are guaranteed by the international standards EN 50170 und EN 50254. PROFIBUS enables communication of devices of various manufacturers without requiring particular interface adaptations.

PROFIBUS-DP (Decentral Periphery) is designed for data transfer between the control and the input/output level. TURCK PDP stations and *piconet®* stations support PROFIBUS-DP.

PROFIBUS-DP is the speed-optimized PROFIBUS version, specially designed for communication between automation devices and decentralized peripheral devices. PROFIBUS-DP is suited to replace cost-intensive parallel signal transmission via digital and analogue sensors and actuators.

PROFIBUS-DP is based on DIN 19245, part 1 and part 4. During the course of European fieldbus standardization, PROFIBUS-DP has been integrated into the European fieldbus standard EN 50170.

3.1.1 DP-V0

DP-V0 contains the following basic functions of the DP communication protocol:

- cyclic exchange of I/O data between master and slaves,
- device, - module-, and channel-specific diagnostics,
- parameterization and configuration of slaves.

3.1.2 DP-V1

This is the first step in the further development of PROFIBUS-DP. DP-V1 is above all, designed for acyclic services (e.g. acyclic parameterization of devices) parallel to the cyclic user data traffic.

- acyclic parameterization, operation, monitoring and alarm handling
- indication of diagnostics using alarms,
- standardization of the first 3 bytes of the user parameter data.

Module addressing in DP-V1

In DP-V1, module addressing is done via slot-number and index, whereas the slot-number describes the module and the index the data block belonging to the module (max. 244 byte).

Modular slaves

In modular devices, the slot-number is assigned to the modules.

With *piconet®*, this is valid for the IP-Link coupling modules and the respective extension modules. The coupling module is slot-number 0, the addressing of the extension modules starts with 1.

Compact devices

The compact devices are considered as an entity of virtual modules.

For the *piconet*® stand-alone modules, addressing starts with slot-number 1, slot-number 0 does not exist.

Note

chapter 9 contains an application example which describes the addressing of modules in the IP-Link as well as the addressing of stand-alone modules.
In addition to that, the acyclic parameterization of modules is specified.

3.1.3 Master/slave-system

PROFIBUS-DP is a master/slave system, which consists of a master (usually integrated in the PLC) and up to 31 slaves per segment. During operation, the master constantly scans the connected slave stations. Several masters may be connected within a single network; this would then be classified as a multi master system. In this case they pass on their transmission permission (Token Passing).

PROFIBUS-DP uses a bit transmission layer (Physical Layer) based on the industrially proven RS485 standard.

System configuration and device types

PROFIBUS-DP is suited for both mono-master or multi-master system configuration. Thus a high level of flexibility in system configuration is guaranteed. The network comprises 126 devices max. (master or slaves).

Configurable system parameters include the number of stations, the assignment of the station address to the I/O addresses, data consistence of I/O data, format of diagnostic messages and the bus parameters used. Every PROFIBUS-DP system consists of different types of devices.

One distinguishes between three device types:

- DP master class 1 (DPM1)

This is a central control, which exchanges data in a defined message cycle with the remote stations (slaves). Typical devices are, for instance, programmable logic controllers (PLCs) or PCs.

- DP master class 2 (DPM2)

Devices of this type are engineering, configuration or operating devices. They are used during set-up, maintenance and diagnosis, to configure the connected devices, to evaluate parameters and to scan the device status.

- DP-slave

A PROFIBUS-DP slave is a peripheral device (I/Os, drives, transducers), which reads input data and provides output data to the periphery. Of course, there are also devices which provide only input or only output data. The input and output data volume depends on the specific device and may comprise up to 246 bytes input data and 246 bytes output data.

Single-Master systems

With mono-master systems merely a single master on the bus is active during bus operation. The PLC is the central control component. The slaves are coupled decentrally to the PLC via the transmission medium. With this type of system configuration the shortest bus cycle times are achieved.

Multi-Master systems

In multi-master operation there are several masters on the bus. These form independent sub-systems, consisting of one DPM1 each and the associated slaves, or additional configuration and diagnostic devices. The slave input and output data can be read by all DP masters. Writing of outputs is reserved to a single DP master (the assigned DPM1 during configuration). Multi-Master systems achieve an average bus cycle time. In time-critical applications you should monitor the bus cycle time via a connected diagnostic tool.

3.1.4 Topology

PROFIBUS-DP communicates via a shielded 2-wire cable according to the RS485 standard. The network topology accords to a line structure with active bus terminators on both ends.

3.1.5 Maximum system expansion

PROFIBUS-DP is suited for connection of a large number .
of I/O points. Up to 126 addressable bus nodes enable connection of thousands of analogue and digital I/O points within a network.

PROFIBUS-DP allows a maximum of 32 nodes per segment; please note that masters and repeaters always count as nodes. One segment is defined as the bus section between two repeaters. If no repeaters are used, the entire network corresponds to one segment.
Segments must comply with the specified maximum length and the specified transmission rates. Up to nine repeaters, type "REP-DP0002" may be connected within a network. The maximum length of a bus line within a segment and the number of repeaters are listed in the following table.

Table 3-1: Maximum system expansion, PROFIBUS-DP

Communication rate	Length of bus line	Max. no. of repeaters	Max. no. of nodes
9.6 kbps	1200 m	2	126
19.2 kbps	1200 m	2	126
93.75 kbps	1200 m	2	126
187.5 kbps	1000 m	2	126
500 kbps	400 m	4	126
1.5 Mbps	200 m	6	126
12 Mbps	100 m	9	126

Attention
The maximum number of 32 bus nodes may not be exceeded without a repeater.

Use of drop lines

Note
The length of drop lines may not exceed 6.6 m at a transmission speed of 1.5 Mbps. At a transmission speed of 12 Mbps it is not permitted to use drop lines.

3.1.6 Transmission rate/Cycle times

The transmission rate set by the PROFIBUS-DP master determines the system's transmission speed. The transmission speed can be adjusted in a range of 9,6 kbps up to 12 Mbps.

The transmission rate is automatically detected by the TURCK stations. Special settings on the stations are not required.

At 12 Mbps the typical response time accords to < 1 ms per 1000 I/O points.

3.1.7 Transmission cables

The bus nodes are interconnected via fieldbus cables, which accord to RS485 specifications and DIN19 245. The cables must thus have the following characteristics:

Table 3-2: Characteristics of PROFIBUS-DP transmission cables

Parameters	Cable type A (DIN 19 245 part 3)
Wave resistance	– 35 to 165 Ω (3 to 20 MHz) – 100 to 130 Ω (f >100 kHz)
Capacitance	< 30 nF/km
Loop resistance	< 110 Ω/km
Core diameter	> 0,64 mm
Core cross-section	> 0,34 mm^2
terminating resistors	220 Ω

Attention

The higher the transmission rate, the higher the number of bus nodes and the longer the transmission cables, the more important to observe these parameters.

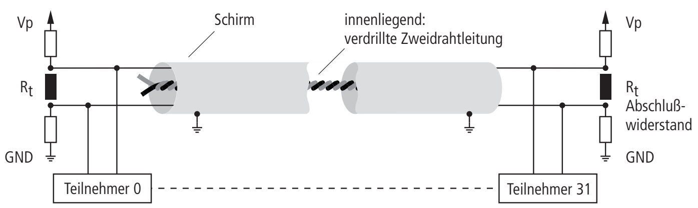


Figure 3-1: Schematic PROFIBUS-DP cable

Cable types

The bus cable of the PROFIBUS-DP network is a special shielded twisted data cable according to RS485 standards. The data transmission rate is 12 Mbps max.

The M12 bus connector of the station is shielded and mechanically reverse-keyed.

Note

Premoulded PROFIBUS-DP cables simplify network installation, shorten set-up times and reduce wiring errors. TURCK offers an extensive and varied product spectrum for this purpose.

The ordering information on the available cable types can be taken from the respective product catalog.

Installation guidelines

When mounting the modules and routing the cables please observe the technical guidelines of the PROFIBUS user organization concerning PROFIBUS-DP/FMS (see www.profibus.com).

Checking the PROFIBUS cabling

A PROFIBUS cable (or the cable segment if repeaters are used) can be tested with a few resistance measurements. For this the cable should be disconnected from all stations:

- Resistance between "A" and "B" at the beginning of the cable: approx. 110 Ω
- Resistance between "A" and "B" at the end of the cable: approx. 110 Ω
- Resistance between "A" at the beginning and "A" at the end of the cable: approx. 0 Ω
- Resistance between "B" at the beginning and "B" at the end of the cable: approx. 0 Ω
- Resistance between shield at the beginning and shield at the end of the cable: approx. 0 Ω

If these measurements are successful, then the cable can be used according to standards. However, if there are further disturbances on the bus, electromagnetic interference should be considered as cause. Please also observe the installation guidelines of the PROFIBUS user organization (www.profibus.com)

3.1.8 Diagnostic functions

The comprehensive diagnostic functions of PROFIBUS-DP allow fast error localization.

The PROFIBUS-DP diagnosis is divided into three levels:

Table 3-3: PROFIBUS-DP diagnostics

Type of diagnosis	Description
station-related diagnostics	Messages concerning the general operational readiness of a bus node. Example: "Excessive temperature" or "under-voltage"
Module-related diagnostics	These messages indicate that there is a diagnostic message within the a certain I/O area (e.g. 8 bit output module) of a bus node.
Channel-related diagnostics	Here the error cause of a single input/output bit, i.e. relating to a single channel, is indicated. Example: "Short-circuit at output 2"Module-related diagnostics

The PROFIBUS slaves of the *piconet*® series support the diagnostic functions of PROFIBUS-DP.

The evaluation of the diagnostic data via the control depends on the support of the master.

Note

Further information on diagnostics can be taken from the device descriptions of the master interfaces of the various manufacturers.

3.1.9 Sync and Freeze Mode

In addition to the node-specific user data traffic, which is automatically controlled by the DPM1, the DP master has the possibility to send control commands to a slave, a group of slaves, or simultaneously to all DP slaves. These control commands are transmitted as multicast messages.

The Sync and Freeze mode for synchronization of the DP slaves can be determined via the control commands. They enable event-controlled synchronization of the DP slaves.

Sync-Mode

The DP slaves initiate the Sync mode upon receipt of a Sync control command from the assigned DP master. In this mode, all addressed DP slaves "freeze" their present output status.
During the following user data transfer cycles, the output data are stored by the DP slaves, while the output states are retained. Only after receipt of the next Sync control command from the master, the stored output data are switched through to the outputs.
The Sync mode is terminated upon an Unsync control command.

Freeze-Mode

The Freeze control command induces the addressed DP slaves to assume the Freeze mode. In this mode, the momentary values of the input states are "frozen". Input data will only be updated upon receipt of the next freeze command from the DP master by the affected devices.
The Freeze mode is terminated upon an Unfreeze control command.

3.1.10 System performance

In order to achieve a relatively high level of device interchangeability, the PROFIBUS-DP system performance has also been standardized. It is largely determined by the operating status of the DPM1. This can be either controlled locally or via the bus from the configuration device.

One distinguishes between three major conditions:

Table 3-4: Operating modes

Operating mode	Description
Stop	There is no data transfer between the DPM1 and the DP slaves. The coupling module merely addresses the modules once after power-up (none of the I/O LEDs illuminate).
Clear	The DPM1 reads the input data of the DP slaves and retains the outputs of the DP slaves in the safe state (depending on the reaction to fieldbus errors, the green I/O LED illuminates and the outputs are set).
Operate	The DPM1 is in the data transfer phase. During cyclic data exchange the inputs of the DP slaves are read and the output information is transferred to the DP slaves (the green I/O LED illuminates).

The DPM1 sends its local status within a configurable time interval via a multimaster command to all assigned DP slaves. The system response to an error in the data transfer phase of the DPM1, e.g. a failure of a DP slave, is determined by the operating parameter "Auto-Clear". If this parameter is set to "True", then the DPM1 switches all outputs of the assigned DP slaves to the safe status, as soon as a DP slave is no longer capable of user data transfer. Then the DPM1 changes to the "Clear" state. If this parameter is set to "False", then the DPM1 will retain its operating condition also in the event of an error and the user can determine the system response.

Data transfer between DPM1 and the DP slaves

Data exchange between the DPM1 and the assigned DP slaves is automatically controlled by the DPM1 in a determined fixed order. During configuration of the bus system, the user assigns the DP slaves to the DPM1. It is also defined which DP slaves are to be included in or excluded from cyclic user data transfer.

Data exchange between DPM1 and the DP slaves can be divided into the phases parameterization, configuration and data transfer.

Prior to including a DP slave in the data transfer phase, the DPM1 checks during the parameterization and configuration phase, whether the programmed required configuration complies with the actual device configuration. This check is used to verify that the device type, the format and length information as well as the number of inputs and outputs accord. The user thus is securely protected against parameterization errors. Additionally to the user data transfer, which is automatically effected by the DPM1, it is also possible to send new parameters to the DP slaves upon request of the user.

Protective mechanisms

In the decentralized periphery it is required to provide the system with highly effective protective functions against faulty parametrization or failure of the transmission devices. PROFIBUS-DP applies certain mechanisms to monitor the DP master and the DP slaves. These can be described as time monitoring functions. The monitoring interval is determined during system configuration.

Table 3-5: Protective mechanisms

Protective mechanisms	Description
Of the DP Master	The DPM1 controls the user data transfer of the slaves via the Data_Control_Timer. Each assigned slaves has a monitoring timer of its own. The timer actuates if no user data are transferred correctly during a certain time interval. In this case the user is informed on this condition. If automatic error response (Auto_Clear = True) is enabled, the DPM1 terminates the "Operate" status, switches the outputs of the assigned slaves into the safe status and returns to the operating status "Clear".
Of the DP-slave	The slave carries out response monitoring to detect master or transmission errors. If there is no data exchange during the response monitoring interval with the associated master, the slave automatically switches the outputs into the safe status. In multimaster system operation, an additional access protection is required for the inputs and outputs of the slaves, in order to ensure that only the authorized master has direct access. The slaves provide an input and output image for all other masters so that this map can be read by any master, even without access token.

Ident. number

Each DP slave and each DPM1 must have an individual ident. number. It is needed so that the DP master can identify the connected devices directly without creating significant protocol overhead. The master compares the ident. numbers of the connected DP devices with the ident. numbers registered in the configuration data of the DPM2. User data transfer will only be started, if the right device types with the right station addresses are connected to the bus. This provides additional protection against configuration errors. The manufacturer specific ident. nos. are determined and assigned by the PROFIBUS user organization (PNO). The PNO governs the ident. no. together with the GSD files.

3.1.11 GSD files

Each PROFIBUS-DP module has a so-called GSD file (German abbr. for device data base file) that comprises detailed information on the module: I/O data volume, transmission rates, revision status etc. This GSD file is needed to configure the station within the PROFIBUS-DP system.

The GSD files can be downloaded via the TURCK web site under www.turck.com.

3.2 Acyclic services via DPV1

The following is basically valid:

- The execution of acyclic services is always possible via DP-V1 during active communication of a Master class 2 (DPM2) and *piconet*®
- For a communication between a DP Master, class 1 (DPM1) and *piconet*® the acyclic communication has to be activated using a configuration tool (acyclic services are deactivated per default).

3.2.1 DP-V1-functions

The stand-alone modules and the extension modules support the following functions of DP-V1 services.
DP-V1 services.

Note
The maximum length of the DP-V1 services is 48 bytes.

Reading the configuration (only IP-Link)

The configuration of the extension modules can be read from the IP-Link coupling module. For each connected extension module, one word (16 bits) is transferred. This word contains, for all analog and byte-oriented module, the modules type (for example 5109 = Incremental Encoder) and for all digital extension module the module size and the module type.

Table 3-6: Description of the digital modules

Bit	Description of the digital modules
0-1	if bit 4 = 0 number of outputs multplied with 2 bit if bit 4 = 1 number of outputs multplied with 8 bit
2-3	if bit 4 = 0 number of inputs multiplied with 2 bit if bit 4 = 1 number of inputs multiplied with 8 bit
4	0 bit data length 2, 1: bit data length 8
5-6	00 This is a combined module with 4 in- and 4 outputs. 01 This is a combined module with 8 in- and 8 outputs. 2 dedicated 3 dedicated
7-15	always = 0

The module configuration can be read using the following DP-V1-parameters (the modules are listed in the same order as they are connected to the OP-Link):

Table 3-7: Reading out the module configuration

Slot number	Index	Byte	Description
0	9	0-1	Description coupling module
0	9	2-3	Module 1
...	...	...	...
0	9	46-47	Module 23
0	10	0-1	Module 24
...	...	...	...
0	10	47-46	Module 47
0	11	0-1	Module 48
...	...	...	...
0	13	47-46	Module 95
0	14	0-1	Module 96
...	...	...	...
0	14	30-31	Module 119

This information can also be written via DP-V1 If the written configuration should not match the configuraiton on the table, a DP-V1 error message is generated.

IP-Link reset

If, via the GSD-file, the parameter "Restart after IP-Link error" is set to "manual reset", the IP-Link reset can be activated with the following DP-V1 parameters (After an IP-Link interruption and it's elimination, the IP-Link can be restarted):

Table 3-8: IP-Link reset

Slot number	Index	Byte	Description
0	99	0	2
0	99	1	1
0	99	2	0
0	99	3	0

Access to module registers

The registers of the extension modules can be red or written:

Table 3-9: Access to module registers

Slot number	Index		Description
1	0		Register 0 of module 1, channel 1
1	...		...
1	63		Register 63 of module 1, channel 1
1	64		Register 0 of module 1, channel 2
1	...		...
1	255		Register 63 of module 1, channel 4
2	0	IP-Link	Register 0 of module 2, channel 1 (from here on only for IP-Link)
2	...		...
...	...		...
255	255		Register 63 of module 255, channel 4

4 Connection of *piconet*® to PROFIBUS-DP

4.1 Connection modes

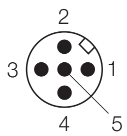
4.1.1 Fieldbus connection

The fieldbus conection of both stand-alone modules and coupling modules is established via reverse-keyed M12 x 1 connectors..

> **Note**
> The PROFIBUS-DP is routed forward either via an external T or Y piece, or, with the modules of new series (e.g.: SDPL-0404D-**1**003) via an integrated T piece.

- M12 male connector for the in-coming bus line (bus-in):

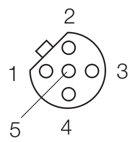
Figure 4-1: M12 male connector, reverse-keyed



1 = n. c.
2 = Line A
3 = GND
4 = Line B
5 = Shield (also on coupling nut)

- M12 female connector for the outgoing bus line (bus-out):

Figure 4-2: M12 female connector, reverse-keyed



1 = 5 VDC
2 = Line A
3 = GND
4 = Line B
5 = Shield (also on coupling nut)

4.1.2 Fieldbus termination

The bus is terminated via an external terminating resistor.

The *piconet*® modules are not capable of fieldbus termination.

> **Note**
> The bus termination must be accomplished externally via a connector with integrated terminating resistor (e.g. RSS4.5-PDP-TR, ident no.: 6601590 as passive terminating resistor or PDP-TRA, ident-nr.: 6825346 as active terminating resistor).

4.1.3 Service interface

The service interface is the interface between the *piconet*® modules and the configuration and diagnostic software I/O-Assistant.

Alongside various offline functions (project planning, plausibility check of stations, detailed project documentation), the software offers many online functions for system set-up, diagnosis and trouble shooting.

4.1.4 Nominal current consumption of *piconet*® modules connected to PROFIBUS-DP

It is important to consider the current consumption of the individual modules for power-feed through calculations, module protection and assessment of the voltage drop on the power cable.

The annex of the *piconet*® I/O module manual [TURCK documentation number: D300777 (German), D300778 (English)] contains tables with the nominal current consumption of the modules.

4.2 *piconet*® connection to the Siemens PLC, type S7

In order to describe the connection of the *piconet*® modules to an S7 type Siemens controller, the software package "SIMATIC Manager", version 5.1. with Service Pack 6 from the company Siemens is used.

4.2.1 Importing the GSD file

Prior to initial configuration of the *piconet*® system via the hardware configurator of the software, the *piconet*® GSD files must be imported into the software.

Two different procedures can be applied:

Prior to starting the software

- Copy the GSD/GSG files of the *piconet*® modules "TRCKFFxx.gsx" into the directory "Step7\S7data\GSD".
- Copy the Icon files (*.bmp) to the directory "Step7\S7data\NSBMP".
- Start the software "SIMATIC Manager".

Upon correct installation of the files, the *piconet*® modules will be registered automatically in the hardware overview which can be called up under the menu item "Insert → Hardware Catalogue".

After starting the software

If you have started to the software, please proceed as follows to import the above mentioned GSx files:

- Please open a new or existing project.
- Open the hardware configurator.
- Copy the required GSx file to the software via the menu item "Extras → Install New GSD File...".

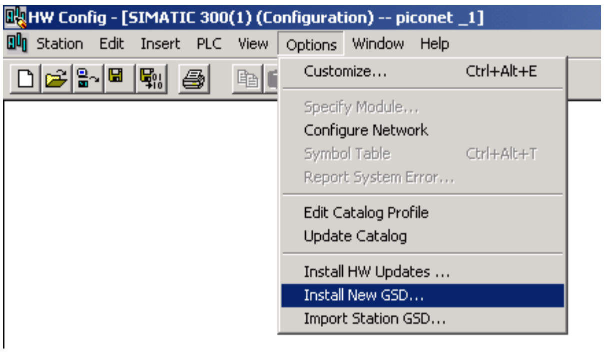


Figure 4-3: Importing a GSD file into the software via the menu item "Install New GSD File...".

- Select the GSD file from the according source directory.

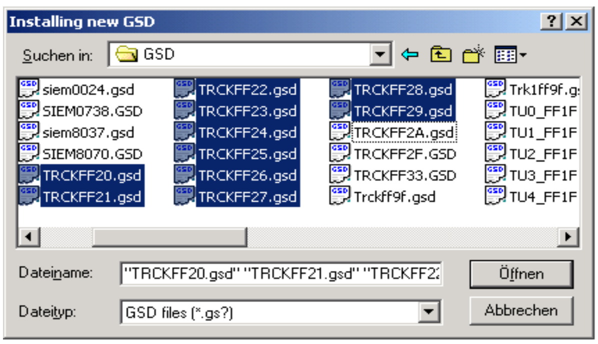


Figure 4-4: Selection of the GSD file from the according source directory.

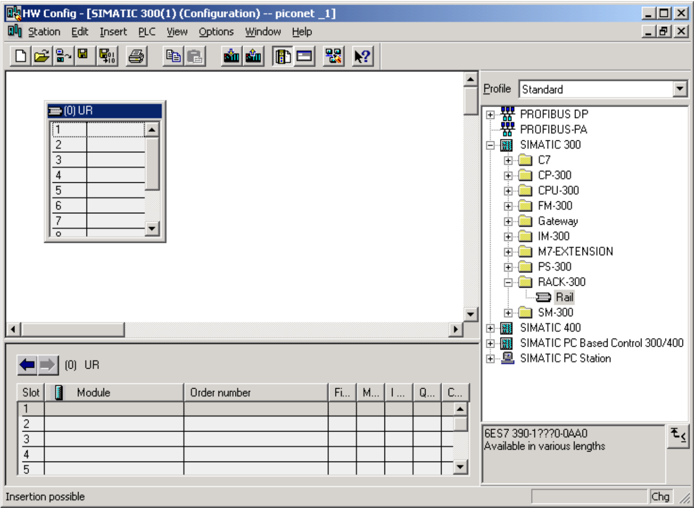
- After correct import and an update of the hardware catalogue via "Extras → Update catalogue", the modules will be displayed as separate entries in the hardware catalogue.

Note
The exact configuration procedure can be taken from the operating manual which is supplied together with the software.

Selection of a CPU

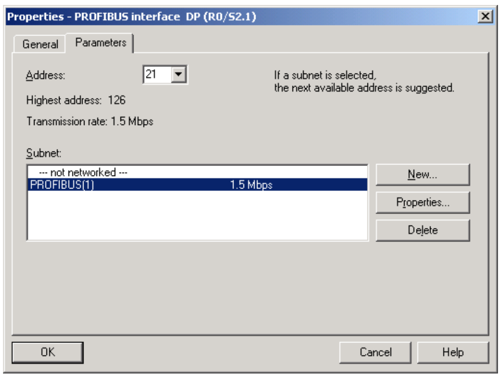
- First, please select a module rack. In this example the rack 300 is selected under "SIMATIC 300 → RACK 300".

Figure 4-5: Selection of a module rack



- Then the CPU type is determined.
- First enter the PROFIBUS address of the CPU and select the subnet "PROFIBUS" in the pop-up dialogue window. Via the button "Properties...." the subnet can be defined more precisely.

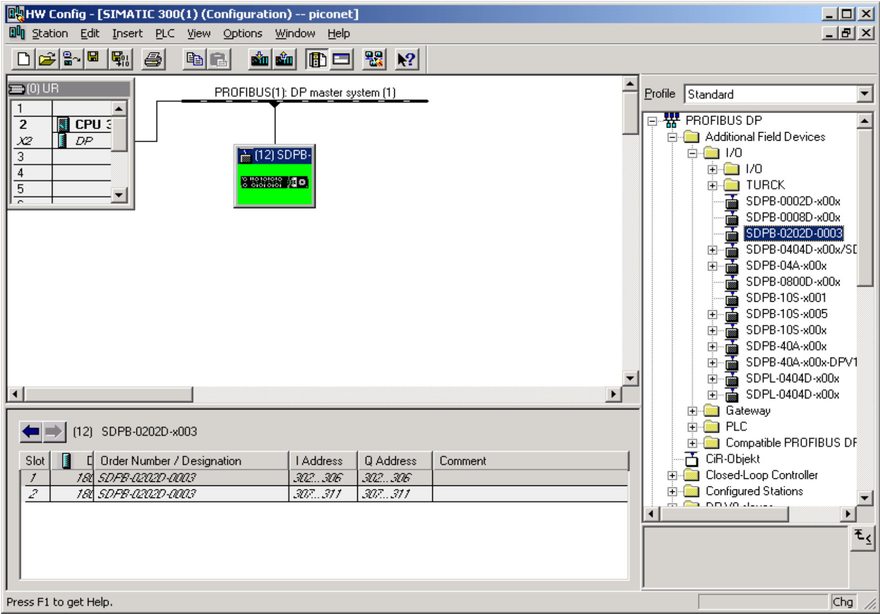
Figure 4-6: Selection of a subnet



4.2.2 Selection of the *piconet*® modules as slaves

- To register the *piconet*® modules as slaves, select the required entries in the hardware catalogue under "Further field devices → I/O".

Figure 4-7: Selection of the piconet® module as a slave



Configuration of the stand-alone modules

piconet® stand-alone module can be moved via drag & drop from the hardware catalogue to the PROFIBUS master system in any order.

IP-Link network configuration

When setting up an IP-Link network, first the coupling module is configured.

Then the extension modules are added to the coupling module. First the byte-oriented *piconet*® modules, i.e. all analogue modules and all technology modules, of the IP-Link ring must be selected.

Note

It must be ensured, that the order of the extension modules in the configuration software accords exactly to the physical order of the modules in the IP-Link network.

Table 4-1: Order in the IP-Link

Module order	
1	byte-oriented modules according to their physical order
2	digital input modules according to their physical order
3	digital output modules according to their physical order

After the byte-oriented modules have been configured, the digital modules are integrated into the IP-Link.

The input and output channels are always configured in steps of 8 to maintain a better overview and clearer assignment of the input and output bytes.

Note

To configure the digital modules in steps of 8, the byte alignment of the coupling module must be activated.

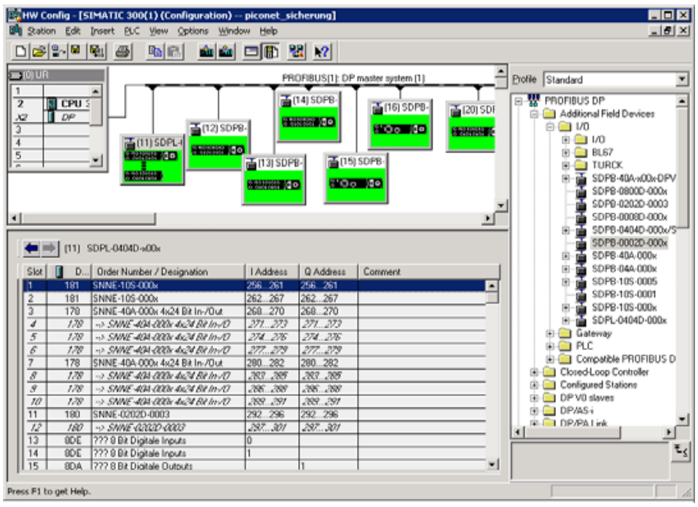


Figure 4-8: IP-Link network configuration

4.2.3 Consistent data transmission

The communication of the S7 controller with the *piconet*® modules in the complex mode requires consistent data transmission.

In order to ensure consistent data transmission, the Siemens function blocks SFC14 and SFC15 are used within an organisation block. These are standard program components of the Simatic Manager.

4.2.4 Consistent data transmission via Siemens function blocks SFC14 and SFC15

The function blocks SFC14 and SFC15 must be copied to the project and called up via the organisation block.

In this example, the function blocks are added to the OB1 of the project.

With the help of these two function blocks, the data for register communication are read and written consistently.

SFC14

The SFC14 is needed to read the module-specific settings and parameters in the register communication.

Figure 4-9: Function block SFC14

```
CALL  "DPRD_DAT"
 LADDR  :=W#16#116
 RET_VAL:=MW10
 RECORD :=P#M 100.0 BYTE 6
NOP   0
```

Call up the function block via the command *Call SFC14* .

Table 4-2: Parameters of the function block SFC14

Parameter name	Meaning	Notation
LADDR	Configured start address of the module from the input data memory of the controller.	The entry is written in an hexadecimal format. Example: W#16#14
RECORD	Target memory area of the CPU for the read user data. Here it is important to enter the data length of the user data (n byte).	Example: P#M 30.0 BYTE 3
RET_VAL	Target memory area of the CPU for a possible error code of the block.	e.g.: MW100

SFC15

The SFC15 is needed to write the module-specific settings and parameters to the register communication.

Figure 4-10: Function block SFC15

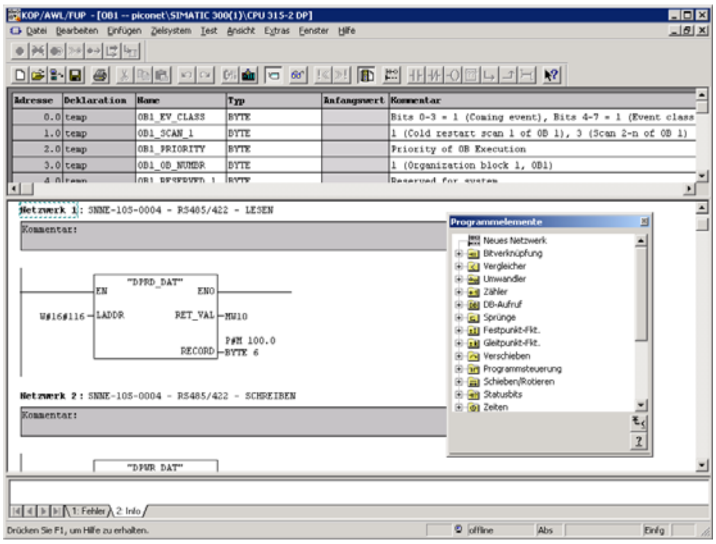
```
CALL  "DPWR_DAT"
 LADDR  :=W#16#116
 RECORD :=P#M 110.0 BYTE 6
 RET_VAL:=MW12
NOP   0
```

Call up the function block via the command *Call SFC14* .

Table 4-3: Parameters of the function block SFC15

Parameter name	Meaning	Notation
LADDR	Configured start address of the module from the output data memory of the controller.	The entry is written in an hexadecimal format. Example: W#16#14
RECORD	Target memory area of the CPU for writing user data. Here it is important to enter the data length of the user data (n byte).	Example: P#M 50.0 BYTE 3
RET_VAL	Target memory area of the CPU for the error code of the block.	e.g.: MW120

Figure 4-11: SFC14 in OB1



Note

The annex of this manual contains a detailed description of the connection of a *piconet*® network to the Siemens controller S7.
The chapter explains configuration, set-up and parameterisation (via GSD file or register communication) of various modules using several application examples.

5 Data mapping of the *piconet*® modules

5.1 Data mapping: Coupling modules

First all byte-oriented modules are registered in the process image. The are registered according to the order in which they are physically installed within the IP-Link ring. Next come the bit-oriented digital modules according to the order in which they are physically installed within the IP-Link ring.

5.1.1 Mapping of process data via byte alignment

Byte alignment is activated and de-activated via the coupling module. It is permitted to map the user data as shown in the following tables.

Note

Byte alignment can only be activated for PROFIBUS-DP modules. For DeviceNet™ only the setting “not active” is admissible.

This function is offered by all SDPL-0404D-x00x modules with the software version "1“ (Dwwxx1yzz). This setting refers to the entire modular extension network (coupling module SDPL-0404D-x00x including extensions SNNE-0404D-000x).

- **Byte alignment not active** (Default setting):
  The coupling module SDPL-0404D-x00x and the extension module SNNE-0404D-000x map 4 bits input and 4 bits output data each.

Table 5-1: Process image without byte alignment

C = female connector
P = Pin

Byte 0	Connection	Bit no. 7	6	5	4	3	2	1	0
IN	M8 x 1	Is used by the next bit-oriented extension module on the IP-Link.				C3P4	C2P4	C1P4	C0P4
	M12 x 1					C1P2	C1P4	C0P2	C0P4
OUT	M8 x 1					C7P4	C6P4	C5P4	C4P4
	M12 x 1					C3P2	C3P4	C2P2	C2P4

- **Byte alignment active**:
  The coupling module SDPL-0404D-x00x and the extension module SNNE-0404D-000x map 1 byte input and 1 byte output user data each.

Table 5-2: Process image with byte alignment

C = female connector
P = Pin
idle = not used

Byte 0	Connection	Bit no. 7	6	5	4	3	2	1	0
IN	M8 x 1	idle	idle	idle	idle	C3P4	C2P4	C1P4	C0P4
	M12 x 1	idle	idle	idle	idle	C1P2	C1P4	C0P2	C0P4
OUT	M8 x 1	C7P4	C6P4	C5P4	C4P4	idle	idle	idle	idle
	M12 x 1	C3P2	C3P4	C2P2	C2P4	idle	idle	idle	idle

5.2 Data mapping: Stand-alone and extension modules

5.2.1 Digital input modules

Table 5-3: Input data in the process image

C = female connector
P = Pin

Bit	7	6	5	4	3	2	1	0
M8 x 1	C7P4	C6P4	C5P4	C4P4	C3P4	C2P4	C1P4	C0P4
M12 x 1	C3P2	C3P4	C3P4	C1P2	C1P2	C1P4	C0P2	C0P4

5.2.2 Digital output modules

Table 5-4: Output data in the process image

C = female connector
P = Pin

Bit	7	6	5	4	3	2	1	0
M8 x 1	C7P4	C6P4	C5P4	C4P4	C3P4	C2P4	C1P4	C0P4
M12 x 1	C3P2	C3P4	C3P4	C1P2	C1P2	C1P4	C0P2	C0P4

Data mapping for SNNE-0016D-0001

- Compact mapping (Motorola format)
  Only the user data are mapped.

Table 5-5: Compact mapping (Motorola format)

C = Connector
P = Pin
SCx = short circuit ch.x

	Word	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Output	n	Low	C0P8	C0P7	C0P6	C0P5	C0P4	C0P3	C0P2	C0P1
		High	C0P16	C0P15	C0P14	C0P13	C0P12	C0P11	C0P10	C0P9

- Complex mapping (Motorola format), Word-Alignment not active, data are mapped with control and status byte (24DI/24DO).

Table 5-6: Complex mapping (Motorola format)

C = Connector
P = Pin
SCx = short circuit ch.x
x = reserved

	Word	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input	n	Low	RA	ERR	UV	OCG2	OCG1	-	DS	-
		High	SC8	SC7	SC6	SC5	SC4	SC3	SC2	SC1
	n + 1	Low	SC16	SC15	SC14	SC13	SC12	SC11	SC10	SC9
		-	-	-	-	-	-	-	-	-
Output	n	Low	RA	RE	SDS	-	-	-	-	-
		High	C0P8	C0P7	C0P6	C0P5	C0P4	C0P3	C0P2	C0P1
	n + 1		C0P16	C0P15	C0P14	C0P13	C0P12	C0P11	C0P10	C0P9
		-	-	-	-	-	-	-	-	-

- Complex mapping (Motorola format), Word-Alignment active, data are mapped with control and status byte (32DI/32DO).

Table 5-7: Complex mapping (Motorola format)

C = Connector
P = Pin
SCx = short circuit ch.x
idle = reserved

	Wort	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input	n	Low	RA	ERR	UV	OCG2	OCG1	-	DS	-
		High	SC8	SC7	SC6	SC5	SC4	SC3	SC2	SC1
	n + 1	Low	SC16	SC15	SC14	SC13	SC12	SC11	SC10	SC9
		idle								
Output	n	Low	RA	RE	SDS	-	-	-	-	-
		High	RA	RE	SDS	-	-	-	-	-
	n + 1		C0P8	C0P7	C0P6	C0P5	C0P4	C0P3	C0P2	C0P1
		idle								

Status byte

DS = Default Status
The outputs have assumed the default status as defined in register R33.

OCGx = Over Current Group x
Short circuit on one of the outputs of group x. The output concerned is indicated in the input byte .

Note
The fault display must be reset when the fault is rectified by resetting bit 6 of the control byte.

UV = Under voltage
Load voltage UL < 18 V, UL-LED lit red, the outptus remain switched off

Error = error bit
Fault has occured

RA = Register Access
Acknowledgement for the process data operation.

Control byte

SDS = Set Default Status
Sets the output to the predefined default status defined in register R33

RE = Reset Error
Resets the indicated faults in the status byte.

RA = Register Access
Register communication switched off (Process data operation)

5.2.3 Digital combined modules

- 4 digital inputs and 4 digital outputs

  1 Module parameter "byte alignment" = not active (default) and previous byte fully used.
  4 bits input and 4 bits output data are mapped:

Table 5-8: Data in the process image

C = female connector
P = Pin

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input	Byte 0 (M8 x 1)	Is used by the next bit-oriented extension module on the IP-Link.				C3P4	C2P4	C1P4	C0P4
	Byte 0 (M12 x 1)					C1P2	C1P4	C0P2	C0P4
Output	Byte 0 (M8 x 1)					C7P4	C6P4	C5P4	C4P4
	Byte 0 (M12 x 1)					C3P2	C3P4	C2P2	C2P4

2 Module parameter "byte alignment" = not active (default) and byte half used.
4 bits input and 4 bits output data are mapped:

Table 5-9: Data in the process image

C = female connector
P = Pin

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input	Byte 0 (M8 x 1)	C3P4	C2P4	C1P4	C0P4	Is used by the physically preceding bit-oriented extension module on the IP-Link.			
	Byte 0 (M12 x 1)	C1P2	C1P4	C0P2	C0P4				
Output	Byte 0 (M8 x 1)	C7P4	C6P4	C5P4	C4P4				
	Byte 0 (M12 x 1)	C3P2	C3P4	C2P2	C2P4				

3 Module parameter "Byte alignment" = active.
1 byte input and 4 bits output data are mapped:

Table 5-10: Data in the process image

C = female connector
P = Pin

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input	Byte 0 (M8 x 1)	idle	idle	idle	idle	C3P4	C2P4	C1P4	C0P4
	Byte 0 (M12 x 1)	idle	idle	idle	idle	C1P2	C1P4	C0P2	C0P4
Output	Byte 0 (M8 x 1)	C7P4	C6P4	C5P4	C4P4	idle	idle	idle	idle
	Byte 0 (M12 x 1)	C3P2	C3P4	C2P2	C2P4	idle	idle	idle	idle

- 8 digital combined inputs/outputs (M8):

Table 5-11: Data in the process image

C = female connector
P = Pin

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input	Byte 0	C7P4	C6P4	C5P4	C4P4	C3P4	C2P4	C1P4	C0P4
Output	Byte 0	C7P2	C6P2	C5P2	C4P2	C3P2	C2P2	C1P2	C0P2

- 8 digital combined inputs/outputs (IP20 terminals):

Table 5-12: Data in the process image

C = female connector
P = Pin

		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Input	Byte 0	C0P8	C0P7	C0P6	C0P5	C0P4	C0P3	C0P2	C0P1
Output	Byte 0	C1P8	C1P7	C1P6	C1P5	C1P4	C1P3	C1P2	C1P1

5.2.4 Analogue input modules

Note
The data mapping of all analogue input modules is identical.

- Compact evaluation in the MOTOROLA format (Default mapping)

Table 5-13: Compact evaluation; MOTOROLA format

DB = Data byte

Address	Input data		Output data	
Word	High Byte	Low Byte	-	-
0	Channel 1/ DB 0	Channel 1/ DB 1	-	-
1	Channel 2/ DB 0	Channel 2/ DB 1	-	-
2	Channel 3/ DB 0	Channel 3/ DB 1		
3	Channel 4/ DB 0	Channel 4/ DB 1	-	-

- Compact evaluation in the INTEL format

Table 5-14: Compact evaluation, INTELINTEL format

DB = Data byte

Address	Input data		Output data	
Word	High Byte	Low Byte	-	-
0	Channel 1/DB 1	Channel 1/ DB 0	-	-
1	Channel 2/ DB 1	Channel 2/ DB 0	-	-
2	Channel 3/ DB 1	Channel 3/ DB 0		
3	Channel 4/ DB 1	Channel 4/ DB 0	-	-

- Complex evaluation in the MOTOROLA format

Table 5-15: Complex evaluation; MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	Channel 1/ DB 1	SB/ Channel 1/	Channel 1/ DB 1	CB/ Channel 1
1	SB/ Channel 2/	Channel 1/ DB 0	CB/ Channel 2	Channel 1/ DB 0
2	Channel 2/ DB 0	Channel 2/ DB 1	Channel 2/ DB 0	Channel 2/ DB 1
3	Channel 3/ DB 1	SB/ Channel 3	Channel 3/ DB 1	CB/ Channel 3
4	SB/ Channel 4	Channel 3/ DB 0	CB/ Channel 4	Channel 3/ DB 0
5	Channel 4/ DB 0	Channel 4/ DB 1	Channel 4/ DB 0	Channel 4/ DB 1

- Complex evaluation in the INTEL format

Table 5-16: Complex evaluation; INTEL format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	Channel 1/ DB 0	SB/ Channel 1	Channel 1/ DB 0	CB/ Channel 1
1	SB/ Channel 2	Channel 1/ DB 1	CB/ Channel 2	Channel 1/ DB 1
2	Channel 2/ DB 1	Channel 2/ DB 0	Channel 2/ DB 1	Channel 2/ DB 0
3	Channel 3/ DB 0	SB/ Channel 3	Channel 3/ DB 0	CB/ Channel 3
4	SB/ Channel 4	Channel 3/ DB 1	CB/ Channel 4	Channel 3/ DB 1
5	Channel 4/ DB 1	Channel 4/ DB 0	Channel 4/ DB 1	Channel 4/ DB 0

5.2.5 Analogue output modules

Note
The data mapping of all analogue output modules is identical.

- Compact evaluation in the MOTOROLA format (default mapping)

Table 5-17: Compact evaluation, INTEL format

DB = Data byte

Address	Input data		Output data	
Word	-	-	High Byte	Low Byte
0	-	-	Channel 1/ DB 1	Channel 1/ DB 0
1	-	-	Channel 2/ DB 1	Channel 2/ DB 0
2	-	-	Channel 3/ DB 1	Channel 3/ DB 0
3	-	-	Channel 4/ DB 1	Channel 4/ DB 0

- Compact evaluation in the INTEL format

Table 5-18: Compact evaluation, INTEL format

DB = Data byte

Address	Input data		Output data	
Word	-	-	High Byte	Low Byte
0	-	-	Channel 1/ DB 1	Channel 1/ DB 0
1	-	-	Channel 2/ DB 1	Channel 2/ DB 0
2	-	-	Channel 3/ DB 1	Channel 3/ DB 0
3	-	-	Channel 4/ DB 1	Channel 4/ DB 0

- Complex evaluation in the MOTOROLA format

Table 5-19: Complex evaluation; MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	**Input data**		**Output data**	
Word	High Byte	Low Byte	High Byte	Low Byte
0	Channel 1/ DB 1	SB/ Channel 1	Channel 1/ DB 1	CB/ Channel 1
1	SB/ Channel 2	Channel 1/ DB 0	CB/ Channel 2	Channel 1/ DB 0
2	Channel 2/ DB 0	Channel 2/ DB 1	Channel 2/ DB 0	Channel 2/ DB 1
3	Channel 3/ DB 1	SB/ Channel 3	Channel 3/ DB 1	CB/ Channel 3
4	SB/ Channel 4	Channel 3/ DB 0	CB/ Channel 4	Channel 3/ DB 0
5	Channel 4/ DB 0	Channel 4/ DB 1	Channel 4/ DB 0	Channel 4/ DB 1

- Complex evaluation in the INTEL format

Table 5-20: Complex evaluation; INTEL format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	**Input data**		**Output data**	
Word	High Byte	Low Byte	High Byte	Low Byte
0	Channel 1/ DB 0	SB/ Channel 1	Channel 1/ DB 0	CB/ Channel 1
1	SB/ Channel 2	Channel 1/ DB 1	CB/ Channel 2	Channel 1/ DB 1
2	Channel 2/ DB 1	Channel 2/ DB 0	Channel 2/ DB 1	Channel 2/ DB 0
3	Channel 3/ DB 0	SB/ Channel 3	Channel 3/ DB 0	CB/ Channel 3
4	SB/ Channel 4	Channel 3/ DB 1	CB/ Channel 4	Channel 3/ DB 1
5	Channel 4/ DB 1	Channel 4/ DB 0	Channel 4/ DB 1	Channel 4/ DB 0

5.3 Technology modules

5.3.1 Sxxx-10S-0001, Incremental encoder interface

This module can only be operated in the complex mode.

- Complex evaluation in the MOTOROLA format
  (Default mapping)

Table 5-21: Complex evaluation in the MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	**Input data**		**Output data**	
Word	High Byte	Low Byte	High Byte	Low Byte
0	DB 1	SB	Reg 1	CB
1	DB 2	DB 0	reserved	Reg 0
2	DB 3	DB 4	reserved	reserved

- Complex evaluation in the INTEL format

Table 5-22: Complex evaluation in the INTEL format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	**Input data**		**Output data**	
Word	High Byte	Low Byte	High Byte	Low Byte
0	DB 0	SB	Reg 0	CB
1	DB 2	DB 1	reserved	Reg 1
2	DB 4	DB 3	reserved	reserved

Data byte 0:
least significant byte of the counter word (read/set)
Data byte 1:
most significant byte of the counter word (read/set)
Data byte 2:
Status of A, B, C (latch), gate and latch input
Data byte 3:
least significant byte of the latch word (read)/least significant byte of the period

Data byte 4:
most significant byte of the latch word (read)/ most significant byte of the period

Reg0:
least significant byte for register communication
Reg1:
most significant byte for register communication

5.3.2 Sxxx-10S-0002, RS232 interface

This module can only be operated in the complex mode.

- Complex evaluation in the INTEL/MOTOROLA format

Table 5-23: Complex evaluation; INTEL/ MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	DB 0	SB	DB 0	CB
1	DB 2	DB 1	DB 2	DB 1
2	DB 4	DB 3	DB 4	DB 3

5.3.3 Sxxx-10S-0004, RS422/485 interface

This module can only be operated in the complex mode.

- Complex evaluation in the INTEL/MOTOROLA format

Table 5-24: Complex evaluation; INTEL/ MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	DB 0	SB	DB 0	CB
1	DB 2	DB 1	DB 2	DB 1
2	DB 4	DB 3	DB 4	DB 3

5.3.4 Sxxx-10S-0005, SSI interface

- Compact evaluation in the MOTOROLA format (default mapping)

Table 5-25: Compact evaluation in the MOTOROLA format

DB = Data byte

Address	Input data		Output data	
Word	High Byte	Low Byte	-	-
0	DB 2	DB 3	-	-
1	DB 0	DB 1	-	-

■ Compact evaluation in the INTEL format

Table 5-26: Compact evaluation in the INTEL format

DB = Data byte

Address	Input data		Output data	
Word	High Byte	Low Byte	-	-
0	DB 1	DB 0	-	-
1	DB 3	DB 2	-	-

■ Complex evaluation in the MOTOROLA format

Table 5-27: Complex evaluation in the MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	DB 3	SB	Reg 1	CB
1	DB 1	DB 2	reserved	Reg 0
2	reserved	DB 0	reserved	reserved

■ Complex evaluation in the INTEL format

Table 5-28: Complex evaluation in the MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	DB 0	SB	Reg 0	CB
1	DB 2	DB 1	reserved	Reg 1
2	reserved	DB 3	reserved	reserved

Data bytes 0 to 3:
Contain the data of the encoder

Reg0:
least significant byte for register communication

Reg1:
most significant byte for register communication

5.3.5 Sxxx-0002D-0002, pulse width output 24 VDC/ 2.5 A

This module can only be operated in the complex mode.

- Complex evaluation in the MOTOROLA format (Default mapping)

Table 5-29: Complex evaluation; MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	Channel 1/ Reg 1	SB/ Channel 1	Channel 1/ DB 1	CB/ Channel 1/
1	SB/ Channel 2	Channel 1/ Reg 0	CB/ Channel 2/	Channel 1/ DB 0
2	Channel 2/ Reg 0	Channel 2/ Reg 1	Channel 2/ DB 0	Channel 2/ DB 1

- Complex evaluation in the INTEL format

Table 5-30: Complex evaluation; INTEL format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	Channel 1/ Reg 0	SB/ Channel 1	Channel 1/ DB 0	CB/ Channel 1
1	SB/ Channel 2	Channel 1/ Reg 1	CB/ Channel 2	Channel 1/ DB 1
2	Channel 2/ Reg 1	Channel 2/ Reg 0	Channel 2/ DB 1	Channel 2/ DB 0

Channel x Reg0:
Channel x, least significant byte for register communication

Channel x Reg1:
Channel x, most significant byte for register communication

5.3.6 Sxxx-0002D-0003, up/down counter 24 VDC

This module can only be operated in the complex mode.

- Complex evaluation in the MOTOROLA format (Default mapping)

Table 5-31: Complex evaluation; MOTOROLA format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	Channel 1/ DB 3	SB/ Channel 1	Channel 1/ DB 3	CB/ Channel 1
1	Channel 1/ DB 1	Channel 1/ DB 2	Channel 1/ DB 1	Channel 1/ DB 2
2	SB/ Channel 2	Channel 1/ DB 0	CB/ Channel 2	Channel 1/ DB 0
3	Channel 2/ DB 2	Channel 2/ DB 3	Channel 2/ DB 2	Channel 2/ DB 3
4	Channel 2/ DB 0	Channel 2/ DB 1	Channel 2/ DB 0	Channel 2/ DB 1

- Complex evaluation in the INTEL format

Table 5-32: Complex evaluation; INTEL format

DB = Data byte
SB = Status Byte
CB = Control Byte

Address	Input data		Output data	
Word	High Byte	Low Byte	High Byte	Low Byte
0	Channel 1/ DB 0	SB / Channel 1	Channel 1/ DB 0	CB/ Channel 1
1	Channel 1/ DB 2	Channel 1/ DB 1	Channel 1/ DB 2	Channel 1/ DB 1
2	SB/ Channel 2	Channel 1/ DB 3	CB/ Channel 2	Channel 1/ DB 3
3	Channel 2/ DB 1	Channel 2/ DB 0	Channel 2/ DB 1	Channel 2/ DB 0
4	Channel 2/ DB 3	Channel 2/ DB 2	Channel 2/ DB 3	Channel 2/ DB 2

6 Error treatment and diagnostics

6.1 Error diagnosis via LEDs

With the *piconet®* system, one distinguishes between the following kind of errors:

- PROFIBUS errors
- IP-Link or local module errors
  Please read the descriptions of the LED diagnostics of IP-Link errors and module errors in the *piconet®* I/O Manual.
  (Document numbers: German, D300777; English, D300778).

6.1.1 Flash codes

Table 6-1: LED flash codes

Flash sequence	Meaning
Fast flashing	Start
First slow sequence	Error code
Second slow sequence	Error argument
Third slow sequence (optional)	Error argument with more than 20 extensions

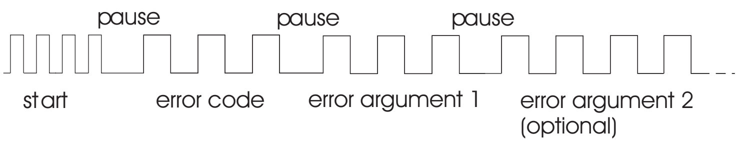


Figure 6-1: Flash codes

6.1.2 Diagnostic LEDs for PROFIBUS

Table 6-2: LEDs for PROFIBUS-DP

LED			Description	Remedy:
Green	**Red**	**I/O RUN**		
ON	ON	None Meaning	Module is waiting for communication	– Baud rate detected Start communication – Possibly the wrong ident. number
OFF	ON	None Meaning	module in baud rate search mode	– Start PROFIBUS – Connect and check bus line
OFF	OFF	None Meaning	Module is waiting for cyclic communication	– Start cyclic PROFIBUS communication
ON	Flashing	None Meaning	Baud rate detected, Start-up error	– Parameter or configuration error (see flash code)
ON	OFF	OFF	DP start-up okay, but still no Data_Exchange message	– Start cyclic PROFIBUS communication
ON	OFF	ON	Module is exchanging data	– No error

Parameter data errors

- **Error code 1**
  The reserved and fixed User_Prm_Data have a wrong value.
  → The error argument indicates the wrong byte (byte 0 starts with error argument 1).
- **Error code 3**
  The combination of some functions, via the User_Prm_Data, is not admissible.
  → The error argument provides detailed information:

Table 6-3: Error arguments parameter error code 3

Argument	Description
1	In the Synchronous Mode it is not permitted to set the response to PROFIBUS errors to "retain outputs". The watchdog of the sub-bus system responds faster than the PROFIBUS watchdog.
2 to 7	reserved
8	The Synchronous Mode is not permitted if there are no outputs.
9 to 11	reserved
12	The Fast-FreeRun Mode is not admissible if the Synchronous Mode is activated

- **Error code 9**
  Error code 9 describes start-up errors

Table 6-4: Error arguments parameter error code 9

Argument	Description
1	DP input data too long
2 ^	DP output data too long
3	DP CfgData data too long
4	Extension module type not recognised
5	Configuration data of the DP interface are faulty

Configuration data errors

- **Error code 2**
  Error during comparison of configuration data.
  → The error argument indicates the first faulty byte (starting with "1").
- **Error code 5**
  Error in the length of the digital output data.
  → The error argument indicates the expected number of bytes.
- **Error code 6**
  Error in the length of the digital input data.
  → The error argument indicates the expected number of bytes.
- **Error code 7**
  Error code 7 describes CfgData errors

Table 6-5: Error arguments parameter error code 7

Argument	Description
1	Length of CfgData not correct
2	Syntax of CfgData not correct
3	Length of input data too long
4	Length of output data too long

6.1.3 Trouble shooting

Should you have problems with the module, the Status LEDs will give information on the probable error cause.

Avoidance of typical errors

1 Checking the PROFIBUS cabling

A PROFIBUS cable (or the cable segment if repeaters are used) can be tested with a few resistance measurements.
For this the cable should be disconnected from all stations:

Table 6-6: Checking the PROFIBUS cabling

Measuring points	Typical resistance
between "A" and "B" at the beginning of the cable	approx. 110 Ω
between "A" and "B" at the end of the cable	approx. 110 Ω
between "A" at the beginning and "A" at the end of the cable	approx. 0 Ω
between "B" at the beginning and "B" at the end of the cable	approx. 0 Ω
between shield at the beginning and shield at the end of the cable	approx. 0 Ω

Note
If these measurements are successful, the cable is error-free.
Should there be further bus errors, these can usually be ascribed to EMI.
Please also observe the installation guidelines of the PROFIBUS user organisation (www.profibus.com)

2 Configuration sequence

When configuring the IP-Link network with digital and analogue modules, first register the analogue modules according to their physical order within the IP-Link in the control software. Then enter the digital modules in accordance to their physical order within the IP-Link. Then the inputs and outputs are mapped.

3 Configuration errors with the PROFIBUS coupling module

With *piconet*® modules with less than 8 bits process data, the process data are rounded off to 8 bits. These modules thus occupy 8 bits in the respective process data area.

Note
To configure the digital modules in steps of 8, the byte alignment of the coupling module must be activated.

Please note, that 4 inputs and 4 outputs must configured, for instance, for the module SxxB-0404D-xxxx.

4 IP-Link errors

If the IP-Link error LED illuminates or flashes irregularly, there is no message transmission or the telegrams are faulty. The modules amplify every telegram but they cannot correct an error occured once. The error must thus be traced backward from the last module to the first error-free module. The connection from this module to the next module is the source of error.

This error is usually due to IP-Link lines with too high damping. A possible cause could be the faulty assembly of cables and connectors.

Note

The most simple test is visual inspection:
When held against a not too bright light source, the respective connector surface should provide a uniformly lightened image (check both sides).

In case of doubt, re-assemble the connector. Please take care not to grind the optical fibre too far down (see also: D300778) *piconet*® I/O Manual, chapter 2, section “IP-Link connector“).

Note

During connector assembly please ensure that the specifications on insulation stripping are observed.

The assembled cable can be tested simply when connected between coupling and extension module. If the cable is correctly connected, there will be no faulty telegrams.

6.2 Error diagnosis via software

6.2.1 General information

In the event of an error, the *piconet*® modules automatically send diagnostic data to the DP master.

Note
Please note, that the parameter "IP module diagnostics" of the diagnostic-capable modules must be enabled.

Generally, DP masters have the possibility to check whether diagnostic data have changed via a flag in the PLC.

Diagnostic data can then be read via a function block. If more diagnostic data apply than can be transferred, then this wil be indicated by the diagnostic data.

6.2.2 General structure of diagnostic messages

Table 6-7: Structure of the diagnostic message

Diagnostic bytes	Diagnostic contents
0 to 5	*Standard PROFIBUS-DP diagnosis* (according to PROFIBUS-DP standard)
6 to 61	*Manufacturer-specific diagnosis*
– 6 and 7	Length of diagnostic data Diagnostic code (defined by PROFIBUS-DP standard)
– 8 to 15	reserved
– 16 to 61	Channel-specific diagnostics

6.2.3 Diagnostic telegram of the stand-alone modules

DP diagnosis accord. to DP standard

Table 6-8: DP diagnosis Stand-alone modules

Byte	Bit	Description
0	0	No slave response (set internally by the DP master)
	1	Slave in start-up mode (evaluation of parameters and configuration)
	2	Configuration error
	3	Ext_Diag_Data present (from byte 6 on)
	4	Function is not supported
	5	Faulty slave response (set internally by the DP master)
	6	Parameterisation error
	7	Slave is exchanging data with other master (set internally by the DP master)
1	0	Slave must be re-parameterised
	1	Slave with static diagnostics
	2	1 (accord. to PNO specification)
	3	DP watchdog is active
	4	Slave is in Freeze Mode
	5	Slave is in Sync Mode
	6	reserved
	7	Slave is de-activated (set internally by the DP master)
2	0-6	reserved
	7	Too many Ext_Diag_Data
3		Station address of master with which data are transferred
4, 5		Ident. number

Manufacturer-specific diagnosis

Table 6-9: Manufacturer-specific diagnosis Stand-alone modules

Byte	Bit	Description
6		Length of Ext_Diag_Data including length byte
7		0x81 (code DPV1 diagnostic format)
8 to 15		0x00

■ **Channel-specific diagnostics**

Table 6-10: Channel-specific diagnostics Stand-alone modules

Byte	Bit	Description
16	0-5	0x00
	6/ 7	Faulty channel number: – 0x00 = error in channel 0 – 0x40 = error in channel 1 – 0x80 = error in channel 2 – 0xC0 = error in channel 3
17	0-5	Status byte of channel
	6	General error bit (set in the event of an error)
18	0-5 6-7	Bit assignment analogous to byte 16 for another faulty channel
19	0-5 6-7	Bit assignment analogous to byte 17 for another faulty channel
...		
60	0-5 6-7	Bit assignment analogous to byte 16 for another faulty channel
61	0-5 6	Bit assignment analogous to byte 17 for another faulty channel

Note
The status byte is transferred from the module to the controller in the complex mapping mode. It contains various status bits depending on the module type (e.g. process data smaller/larger limit value x, over-range, under-range etc.)

6.2.4 Diagnostic message of the coupling modules

The DP master usually enables polling of a diagnostic data change via a flag in the PLC.

Diagnostic data can then be read via a function block. With the Siemens controller S5 the diagnostic data are read with the function module FB IM308C and with the Siemens controller S7 with the function block SFC13.

If more diagnostic data apply than can be transferred, then this will be indicated by the diagnostic data.

DP diagnosis accord. to DP standard

Table 6-11: DP diagnosis coupling modules

DP diagnosis

Byte	Bit	Description
0	0	No slave response (set internally by the DP master)
	1	Slave in start-up mode (evaluation of parameters and configuration)
	2	Configuration error
	3	Ext_Diag_Data present (from byte 6 on)
	4	Function is not supported
	5	Faulty slave response (set internally by the DP master)
	6	Parameterisation error
	7	Slave is exchanging data with other master (set internally by the DP master)
1	0	Slave must be re-parameterised
	1	Slave with static diagnostics
	2	1 (accord. to PNO specification)
	3	DP watchdog is active
	4	Slave is in Freeze Mode
	5	Slave is in Sync Mode
	6	reserved
	7	Slave is de-activated (set internally by the DP master)
2	0-6	reserved
	7	Too many Ext_Diag_Data
3		Station address of master with which data are transferred
4, 5		Ident. number

Manufacturer-specific diagnosis

Note

The values in bytes 6 to 10 and bytes 14 and 15 are presented in a hexadecimal format. The values in byte 11 to byte 13 are decimal.

Table 6-12: Manufacturer-specific diagnosis Coupling modules

Byte	Bit/ Value	Description
6		Length of Ext_Diag_Data including length byte
7		0xA1
8 to 10		0x00
11	0	No error
	1	Start IP-Link error diagnostics
	32	General IP-Link error
12	0	No error
	3	IP-Link: Interruption – Error argument: Fault location → In case of wire-break: against counting direction Fibre-optic cabling.
	4	IP-Link: Timeout errors – Error argument: Fault location
	5	Error when reading the register of the complex modules – Error argument: Problematic module
	11	Extension module is not processing the synchronisation telegram – Error argument: Problematic module
	12	More than 120 modules connected – Error argument: Number of modules connected in excess
	13	Module type is not supported – Error argument: Problematic module
13	0 to 120	– Error argument: IP-Link error (e.g. wire break) behind the (n-) extension module. Counting begins with **n = 1** with the last connected extension module against the direction of the fibre-optic cabling **backwards** in direction of the coupling module – Error argument **n = 0** accords to IP-Link error between the last coupling module and the last connected extension module.
14	0x00	
15	0x00	

Channel-specific diagnostics

Note
Channel-specific diagnostics are only available for coupling modules with firmware status "D.kkjj**2**yzu“ (firmware "B3“)!

Table 6-13: Channel-specific diagnostics Coupling modules

A *During connection of a valve terminal of the company Festo please observe the following note!*

Byte	Bit	Description
16	0 - 7	Faulty extension module number (1-120 only complex modules are counted in the counting direction "with the light")
17 **A**	0 -5	Status byte of channel
	6 -7	Faulty channel number (0-3)
18	0-7	Bit assignment analogous to byte 16 for another faulty module
19	0-5 6-7	Bit assignment analogous to byte 17 for another faulty module
...		
60	0-7	Bit assignment analogous to byte 16 for another faulty module
61	0-5 6-7	Bit assignment analogous to byte 17 for another faulty module

Attention
When connecting a Festo valve terminal please observe the following:
If byte 16 indicates a valve terminal error via indication of the extension module number this always signals an under-voltage of U_L (U_L< 20.4 V). Byte 17 is always = 0!
Please check the load voltage of the valves.

7 User parameters of the *piconet*® modules

7.1 Introduction

The chapter describes the bus-specific bit assignment of theuser parameters of the individual *piconet*® modules.

Attention

Alongside the parameter, which are defined via the GSD file of the modules and that can be be modified with the configuration software, many modules have parameters that can only be changed via the register communication.

In this case the parameters are contained in the register descriptions of the respective modules in the I/O Module Manual of the product series *piconet*® – TURCK document number: D300777 (German), D300778 (English).

Note

A more detailed description of the individual parameters, their settings and their impact on the system is contained in the annex of the I/O Module Manual, – TURCK document number: D300777 (German), D300778 (English).

7.2 Module-independent user parameters

The following parameters are identical for all PROFIBUS-DP nodes of the *piconet*® system:

Table 7-1: Module-independent parameters

A *Default setting*

Byte	Bit	Parameters
7	1	– *IP module diagnosis*
		0 = is not active **A**
		1 = is active
9	3 to 0	– *Data format*
		0011 = INTEL
		1011 = MOTOROLA **A**
	6 and 5	– *Updating of process image*
		01 = Synchron
		11 = FreeRun **A**
10	1 and 0	– *Reaction on DP error*
		00 = IL-cycle stopped **A**
		01 = IL outputs to 0
		11 = Outputs remain equal

7.3 IP-Link coupling module, SDPL-0404D-x00x

The modules feature the following module-specific parameters:

Table 7-2: Parameters SDPL-0404D-x00x

A *Default setting*

Byte	Bit	Parameters
10	3 and 2	– *IP-Link error reaction*
		00 = leave DataExchange**A**
		01 = DP inputs to 0
		10 = DP inputs remain equal
	5	– *SDPL-0404D-xxxx BYTE-Align*
		0 = is not active **A**
		1 = is active
7	0	– *Behavior on IP-Link error*
		manual reset
		automatic reset

7.4 Digital stand-alone modules

Note
The digital stand-alone modules of the *piconet*® systes do not have any other adjustable parameters in addition to the module-independent parameters.

7.5 Digital extension modules

Note
The digital extension modules of the *piconet*® system are not parameterisable.

7.6 Analogue input modules Sxxx-40A-000x

7.6.1 Sxxx-40A-0004

The modules feature the following module-specific parameters:

Table 7-3: Parameters Sxxx-40A-0004

A *Default setting*

Byte	Bit	Parameters
0	0	– *Cold junction compensation*
		0 = each channel individual **A**
		1 = channel 1 for all channels
1	0 to 3	– *Channel 1 thermocouple*
		0000 = thermocouple type L
		0001 = thermocouple type K **A**
		0010 =thermocouple type J
		0011 = thermocouple type E
		0100 =thermocouple type T
		0101 =thermocouple type N
		0111 = thermocouple type B
		1000 = thermocouple type R
		1001 = thermocouple type S
		1010 = reserved
		1011 = reserved
		1100 = reserved
		1101 = -30...+30 mV
		1110 = -60...+60 mV
		1111 = -120...+120 mV
1	5	– *Channel 1 Siemens additional bit*
		0 = is not active **A**
		1 = is active
	6	– *Channel 1 Reference junction*
		0 = is active
		1 = is not active **A**
2	0 to 6	Parameters of channel 2
3	0 to 6	Parameters of channel 3
4	0 to 6	Parameters of channel 4

TURCK Industrial Automation

7.6.2 Sxxx-40A-0005

The modules feature the following module-specific parameters:

Table 7-4: Parameters Sxxx-40A-0005

A *Default setting*

Byte	Bit	Parameters
1	4	– *Channel 1 overflow offset*
		0 = is not active
		1 = is active **A**
	6	– *Channel 1 threshold 1*
		0 = is not active **A**
		1 = is active
	7	– *Channel 1 threshold 2*
		0 = is not active **A**
		1 = is active
2		Bit assignment of channel 2 identical to channel 1
3		Bit assignment of channel 3 identical to channel 1
4		Bit assignment of channel 4 identical to channel 1
6, 7		– *Channel 1/ threshold 1*
8, 9		– *Channel 1/ threshold2*
10, 11		– *Channel 2/ threshold 1*
12, 13		– *Channel 2/ threshold 2*
14, 15		– *Channel 3/ threshold 1*
16, 17		– *Channel 3/ threshold 2*
18, 19		– *Channel 4/ threshold 1*
20, 21		– *Channel 4/ threshold 2*

7.6.3 Sxxx-40A-0007

The modules feature the following module-specific parameters.

Table 7-5: Parameters Sxxx-40A-0007

A *Default setting*

Byte	Bit	Parameters
1	0	– *Channel 1 current modus*
		0 = 0...20 mA **A**
		1 = 4...20 mA
	4	– *Channel 1 overflow offset*
		0 = is not active
		1 = is active **A**
	6	– *Channel 1 threshold 1*
		0 = is not active **A**
		1 = is active
	7	– *Channel 1 threshold 2*
		0 = is not active **A**
		1 = is active
2		Bit assignment of channel 2 identical to channel 1
3		Bit assignment of channel 3 identical to channel 1
4		Bit assignment of channel 4 identical to channel 1
6, 7		– *Channel 1/ threshold 1*
8, 9		– *Channel 1/ threshold 2*
10, 11		– *Channel 2/ threshold 1*
12, 13		– *Channel 2/ threshold 2*
14, 15		– *Channel 3/ threshold 1*
16, 17		– *Channel 3/ threshold 2*
18, 19		– *Channel 4/ threshold 1*
20, 21		– *Channel 4/ threshold 2*

7.6.4 Sxxx-40A-0009

The modules feature the following module-specific parameters:

Note
The adjustments, which are made in byte 1, apply to all other channels of the module.

Table 7-6: Parameters Sxxx-40A-0009

A *Default setting*

Byte	Bit	Parameters
1	3 ... 0	– *Channel 1 RTD* **A**
		0000 = Pt 100 0001 = Ni 100 0010 = Pt 1000 0011 = Pt 500 0100 = Pt 200 0101 = Ni 1000 0110 = Ni 120 0111 = RSNE1000 1000 = 10-5000 Ohm 1001 = 10-1200 Ohm
	4	– *Channel 1 Siemens additional bit*
		0 = is not active **A**
		1 = is active
	5	– *Channel 1 overrange protection*
		0 = is not active
		1 = is active **A**
	6	– *Channel 1 3-wire*
		0 = is not active **A**
		1 = is active
	7	– *Channel 1 2-wire*
		0 = is not active **A**
		1 = is active
2		Bit assignment of channel 2 identical to channel 1
3		Bit assignment of channel 3 identical to channel 1
4		Bit assignment of channel 4 identical to channel 1

7.7 Analogue output modules Sxxx-04A-000x

Sxxx-04A-0007

The modules feature the following module-specific parameters:

Table 7-7: Parameters Sxxx-04A-0007

A *Default setting*

Byte	Bit	Parameters
0	3	– Channel 1 watchdog
		0 = is active **A**
		1 = is not active
1	3	– Channel 2 watchdog
		0 = is active **A**
		1 = is not active
2	3	– Channel 3 watchdog
		0 = is active **A**
		1 = is not active
3	3	– Channel 4 watchdog
		0 = is active **A**
		1 = is not active

Sxxx-04A-0009

The modules feature the following module-specific parameters:

Table 7-8: Parameters Sxxx-04A-0009

A *Default setting*

Byte	Bit	Parameters
0	3	– Channel 1 watchdog
		0 = is active **A**
		1 = is not active
	5	– Current mode for all channels
		0 = 0...20 mA **A**
		1 = 4...20 mA
1	5	– Channel 2 watchdog
		0 = is active **A**
		1 = is not active
2	6	– Channel 3 watchdog
		0 = is active **A**
		1 = is not active
3	7	– Channel 4 watchdog
		0 = is active **A**
		1 = is not active

7.8 Technology modules

7.8.1 Sxxx-10S-0001, Incremental encoder interface

This module cannot be clear-text parameterised via the GSD file at present.

The module must be parameterised via the register communication of via the software "I/O ASSISTANT". For this, please refer to chapter 3 of the *piconet*® I/O module manual – TURCK documentation number: D300777 (German), D300778 (English) –.

7.8.2 Sxxx-10S-0002, RS232 interface

The modules feature the following module-specific parameters:

Table 7-9: Parameters Sxxx-10S-0002

A *Default setting*

Byte	Bit	Parameters
0	2, 1, 0	– Baud rate
		000 = reserved 001 = reserved 010 = reserved 011 = 1200 Baud 100 = 2400 Baud 101 = 4800 Baud 110 = 9600 Baud **A** 111 = 19200 Baud
1	2, 1, 0	– Data structure
		000 = reserved 001 = 7 data, even 010 = 7 data, odd 011 = 8 data, no **A** 100 = 8 data, even 101 = 8 data, odd 110 = reserved 111 = reserved
	3	– Stop bits
		0 = 1 stop bit **A**
		1 = 2 stop bits
2		reserved
3	2	– Status one cycle later
		0 = is not active **A**
		1 = is active

Byte	Bit	Parameters
3	3	– XON/ XOFF send
		0 = is not active **A**
		1 = is active
	4	– XON/ XOFF receive
		0 = is not active **A**
		1 = is active
	5	reserved
	6	– send 16 byte
		0 = is not active **A**
		1 = is active

7.8.3 Sxxx-10S-0004, RS422/485 interface

The modules feature the following module-specific parameters:

Table 7-10: Parameters Sxxx-10S-0002

A*Default setting*

Byte	Bit	Parameters
0	2, 1, 0	– Baud rate
		000 = reserved 001 = reserved 010 = reserved 011 = 1200 Baud 100 = 2400 Baud 101 = 4800 Baud 110 = 9600 Baud **A** 111 = 19200 Baud
1	2, 1, 0	– Data structure
		000 = reserved 001 = 7 data, even 010 = 7 data, odd 011 = 8 data, no **A** 100 = 8 data, even 101 = 8 data, odd 110 = reserved 111 = reserved
	3	– Stop bits
		0 = 1 stop bit **A**
		1 = 2 stop bits
2		reserved

Byte	Bit	Parameters
3	0	– Half duplex
		0 = is not active **A**
		1 = is active
	2	– State one cycle later
		0 = is not active **A**
		1 = is active
3	3	– XON/ XOFF send
		0 = is not active **A**
		1 = is active
	4	– XON/ XOFF receive
		0 = is not active **A**
		1 = is active
	5	– RS422 mode
		0 = is not active **A**
		1 = is active
	6	– send 16 bytes
		0 = is not active **A**
		1 = is active

7.8.4 Sxxx-10S-0005, SSI interface

This module cannot be clear-text parameterised via the GSD file at present.

The module must be parameterised via the register communication of via the software "I/O ASSISTANT". For this, please refer to chapter 3 of the *piconet*® I/O module manual – TURCK documentation number: D300777 (German), D300778 (English) –.

7.8.5 Sxxx-0002D-0002, pulse width output 24 VDC/ 2.5 A

This module cannot be clear-text parameterised via the GSD file at present.

The module must be parameterised via the register communication of via the software "I/O ASSISTANT". For this, please refer to chapter 3 of the *piconet*® I/O module manual – TURCK documentation number: D300777 (German), D300778 (English) –.

TURCK Industrial Automation

7.8.6 Sxxx-0002D-0003, up/down counter 24 VDC

The modules feature the following parameters:

Table 7-11: Parameters Sxxx-10S-0002

A *Default setting*

Byte	Bit	Parameters
Channel 1		
20	0	– Channel 1 watchdog
		0 = is active **A**
		1 = is not active
	1	– Channel 1 set counter
		0 = positive signal **A**
		1 = positive edge
	2	– Channel 1 set output
		0 = is not active **A**
		1 = is active
	3	– Channel 1 reset output
		0 = is not active **A**
		1 = is active
	4	– Channel 1 reset counter
		0 = is not active **A**
		1 = is active
	5	– Channel 1 pulse mode
		0 = is not active **A**
		1 = is active
	6	– Channel 1 disable counter
		0 = Gate 0 **A**
		1= Gate 1
20	7	Channel 1 timer basis
		0 = 1 µs/ Digit 250 µs – 65 ms **A**
		1 = 64 µs/ Digit 10 µs – 4 ms
21	0	– Channel 1 set back with reset
		0 = is not active **A**
		1 = is active
24 to 27		– Channel 1 power-on level
28 to 31		– Channel 1 power-off/pulse

B *Parameter setting identical to channel 1 of the module*

Byte	Bit	Parameters
32 to 35		– Channel 1 reset level
Channel 2		
22 and 23		The assignment of the parameter bytes 22 and 23 of the module accords to the assignment of the parameter bytes 20 and 21 for channel 1 of the module.
36 to 39		The assignment of the parameter bytes 36 to 39 of the module accords to the assignment of the parameter bytes 24 to 27 for channel 1 of the module.
40 to 43		The assignment of the parameter bytes 40 to 43 of the module accords to the assignment of the parameter bytes 28 to 31 for channel 1 of the module.
44 to 47		The assignment of the parameter bytes 44 to 47 of the module accords to the assignment of the parameter bytes 32 to 35 for channel 1 of the module.

8 Application example: *piconet*® with S7

8.1 General information

The following chapter contains a detailed description of the connection of a *piconet*® network to the controller S7.

The Siemens controller Simatic S7 together with the CPU 315-2AG10-0AB0 and the Simatic Manager V 5.1, Service Pack 6. is used in our example.

The *piconet*® network contains the following modules:

Table 8-1: Model network

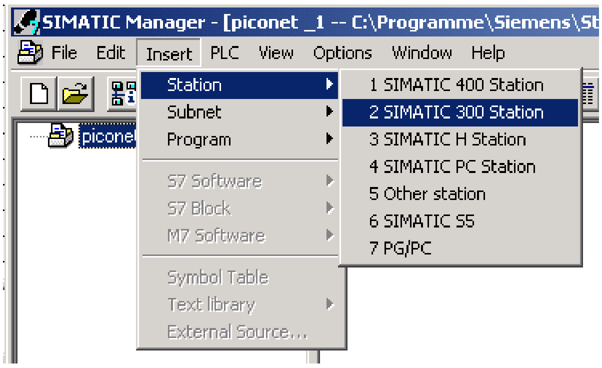
Module	DP address	Designation	Function
A	11	SDPL-0404D-x00x	4DI/ 4 DO, coupling module for PROFIBUS-DP
A_1	(IP-Link)	SNNE-10S-0004	RS422/485-Interface, extension module
A_2	(IP-Link)	SNNE-10S-0002	RS232 interface, extension module
A_3	(IP-Link)	SNNE-40A-0009	Pt100, extension module
A_4	(IP-Link)	SNNE-40A-0004	Thermoelement, extension module
A_5	(IP-Link)	SNNE-0202D-0003	Up/down counter, extension module
A_6	(IP-Link)	SNNE-0808D-0001	8 DI / 8 DO, Extension module
A_7	(IP-Link)	SNNE-0008D-0001	8 DO, Extension module
B	12	SDPB-0202D-0003	Up/down counter, stand-alone module
C	13	SDPB-10S-0004	RS422/485-Interface, Stand-alone module
D	14	SDPB-0002D-0002	PWM module, Stand-alone module
E	15	SDBP-10S-0005	SSI encoder interface, Stand-alone module
F	16	SDBP-10S-0001	incremental encoder interface, Stand-alone module

8.2 Creating a new Simatic project

1 Create a new project in the Simatic Manager "File → New..."

2 Select a Simatic station via „Insert → Station". In our example a Simatic 300 station is used.

Figure 8-1: Selection of the Simatic station



8.3 Importing the GSD file

Prior to initial configuration of the *piconet*® system via the hardware configurator of the software, the *piconet*® GSx files must be imported into the software.

Two different procedures can be applied:

8.3.1 Prior to starting the software

- Copy the GSx files of the *piconet*® modules into the directory "Step7\S7data\GSD“.
- Copy the Icon files (*.bmp) to the directory "Step7\S7data\NSBMP“.
- Start the software "SIMATIC Manager".

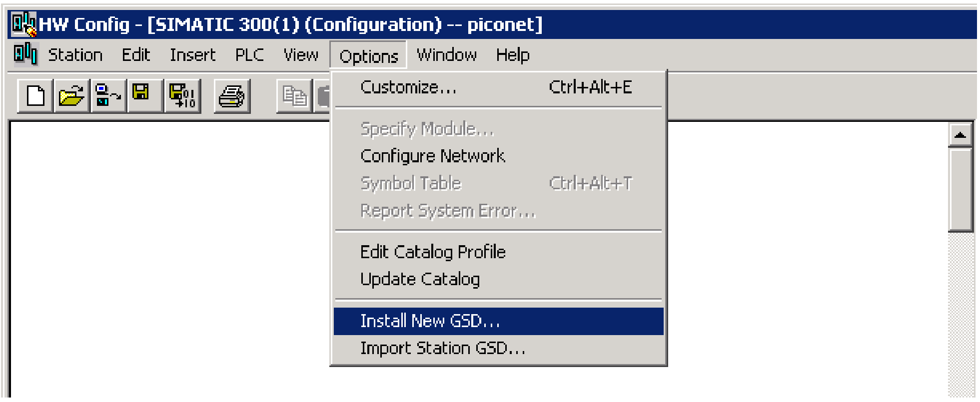
Upon correct installation of the files, the *piconet*® modules will be registered automatically in the hardware overview which can be called up under the menu item "Insert → Hardware Catalogue".

8.3.2 After starting the software

If you have started to the software, please proceed as follows to import the above mentioned GSx files:

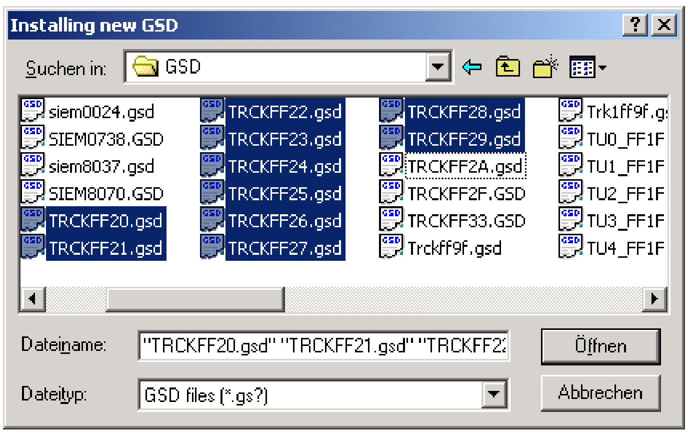
- Please open a new or existing project.
- Open the hardware configurator.
- Copy the required GSx file to the software via the menu item "Extras → Install New GSD File...".

Figure 8-2: Importing a GSD file into the software via the menu item "Install New GSD File...".



- Select the GSD file from the according source directory.

Figure 8-3: Selection of the GSD file from the according source directory



- After correct import and an update of the hardware catalogue via "Extras → Update catalogue", the modules will be displayed as separate entries in the hardware catalogue.

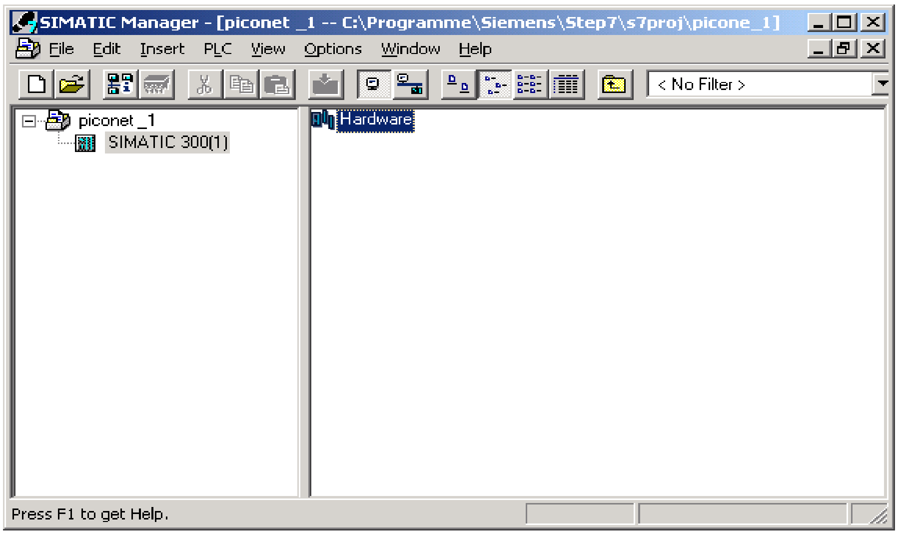
Note

The exact configuration procedure can be taken from the operating manual which is supplied together with the software.

8.4 Hardware configuration

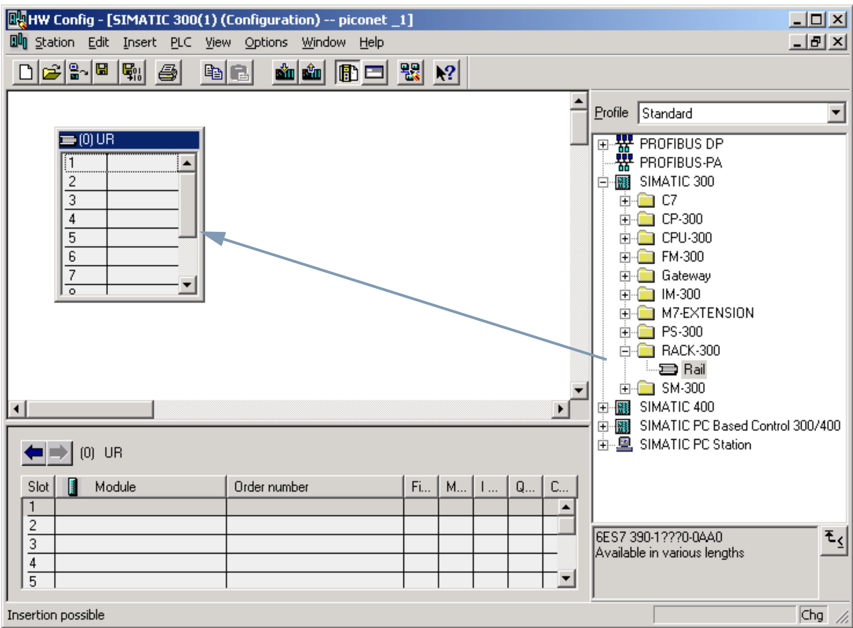
1 Open the hardware configuration via the folder "Hardware".HW Config) of the project.

Figure 8-4: Opening the hardware Configuration



2 In the hardware catalogue first select „Simatic 300 → RACK-300“ to select the type of rack. Simply drag it to the upper part of the station window.

Figure 8-5: Selection of the Mounting rack

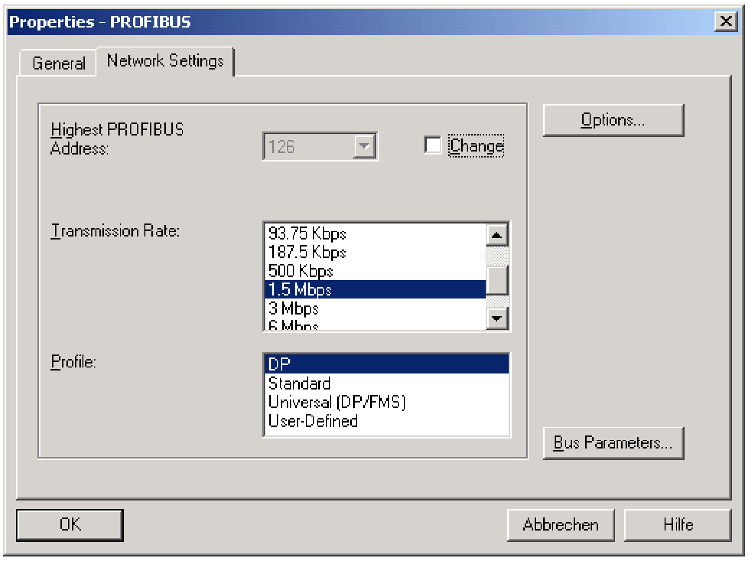


3 Then the CPU type is determined. In our example a CPU 315-2AG10-0AB0 is used. Drag the according entry from the hardware catalogue to the 2nd position of the mounting rack in the station window.

4 In the pop-up dialogue window "Properties - PROFIBUS interface DP" please define the PROFIBUS address of your control system and then select the subnet. Possibly a new subnet must be added via the button "New...".

5 The characteristics of the subnet, such as the transmission speed and profile, can be determined via "Properties..." in the window "Properties - PROFIBUS".
These can differ system-specifically from the specifications made here:

Figure 8-6: Properties of the subnet



8.4.1 Configuration of a *piconet*® network

1 After selecting the CPU and configuring the PROFIBUS-DP, the fieldbus nodes are configured. These will now appear after correct installation of the GSD files under "Additional field devices → I/O" in the hardware catalogue.

Configuration of the IP-Link

2 The first module in our model network is the coupling module with the hardware address 11.

3 Then the extension modules are added to the coupling module.

4 First the byte-oriented *piconet*® modules, i.e. all analogue modules and all technology modules, of the IP-Link ring must be selected.

Attention

It must be ensured, that the order of the extension modules in the configuration software accords exactly to the physical order of the modules (see Table 67: "Model network") in the IP-Link network!

5 After the byte-oriented modules have been configured, the digital modules are integrated into the IP-Link.

6 Then the input and output channels of the modules are configured in steps of 8.

Note

To configure the digital modules in steps of 8, the byte alignment of the coupling module must be activated.

Note

When configuring the digital inputs and outputs, please include the 4 inputs and 4 outputs of the coupling module.

Figure 8-7: Configuration of the model network

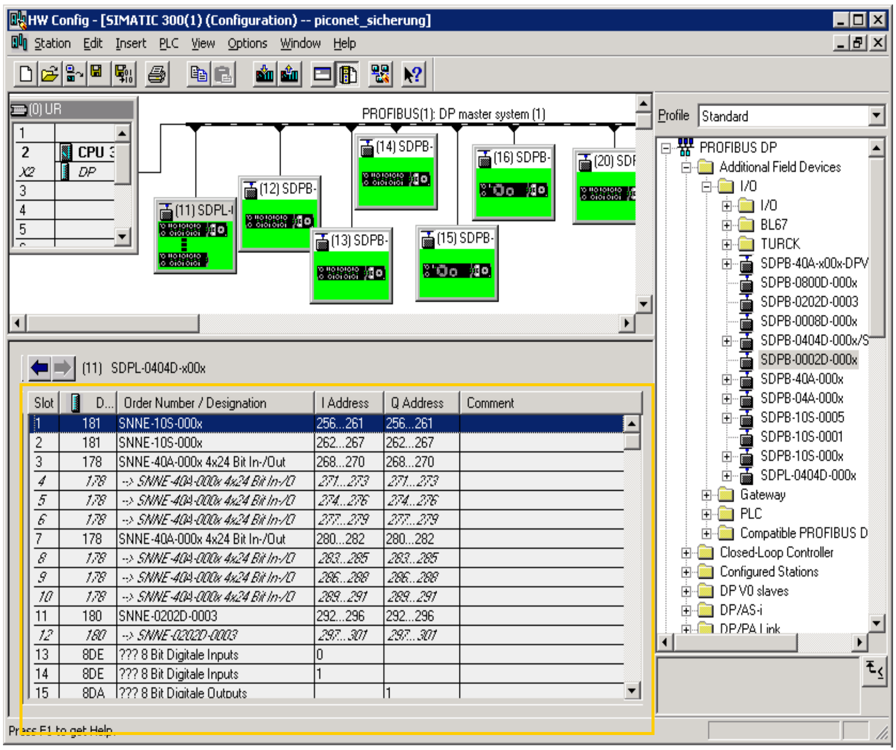


Figure 8-8: Configuration of the digital modules in steps of 8

(11) SDPL-0404D-x00x

Slot	D...	Order Number / Designation	I Address	Q Address	Comment
1	181	SNNE-10S-000x	256...261	256...261	
2	181	SNNE-10S-000x	262...267	262...267	
3	178	SNNE-40A-000x 4x24 Bit In-/Out	268...270	268...270	
4	*178*	*-> SNNE-40A-000x 4x24 Bit In-/O*	*271...273*	*271...273*	
5	*178*	*-> SNNE-40A-000x 4x24 Bit In-/O*	*274...276*	*274...276*	
6	*178*	*-> SNNE-40A-000x 4x24 Bit In-/O*	*277...279*	*277...279*	
7	178	SNNE-40A-000x 4x24 Bit In-/Out	280...282	280...282	
8	*178*	*-> SNNE-40A-000x 4x24 Bit In-/O*	*283...285*	*283...285*	
9	*178*	*-> SNNE-40A-000x 4x24 Bit In-/O*	*286...288*	*286...288*	
10	*178*	*-> SNNE-40A-000x 4x24 Bit In-/O*	*289...291*	*289...291*	
11	180	SNNE-0202D-0003	292...296	292...296	
12	*180*	*-> SNNE-0202D-0003*	*297...301*	*297...301*	
13	8DE	??? 8 Bit Digitale Inputs	0		
14	8DE	??? 8 Bit Digitale Inputs	1		
15	8DA	??? 8 Bit Digitale Outputs		1	

This results in the following allocation of the process data bytes in the IP-Link network of the sample station (compare Figure 8-8:).

Table 8-2: Process data bytes of the IP-Link

IP-Link network			**Process data**	
Module	physical order in the IP-Link	Designation	Input bytes (no.)	Output bytes (no.)
A		SDPL-0404D-x00x	0	0
A_1	1	SNNE-10S-0004	256 ... 261	256 ... 261
A_2	2	SNNE-10S-0002	262 ... 267	262 ... 267
A_3	3	SNNE-40A-0009	268 ... 279	268 ... 279
A_4	4	SNNE-40A-0004	280 ... 291	280 ... 291
A_5	5	SNNE-0202D-0003	292 ... 301	292 ... 301
A_6	6	SNNE-0808D-0001	1	1
A_7	7	SNNE-0008D-0001	-	2

Attention

Only if the parameter „SDPL-0404-xxxx Byte-Align“ is activated in the coupling module, all digital modules will occupy a full byte input or output data.
Please also read Kapitel 4, section Mapping of process data via byte alignment.

Data mapping without byte alignment

Table 8-3: Process image without byte alignment

Byte	**Input data**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	**A_6,** Bit 3	**A_6,** Bit 2	**A_6,** Bit 1	**A_6,** Bit 0	**A,** Bit 3	**A,** Bit 2	**A,** Bit 1	**A,** Bit 0
1	-	-	-	-	**A_6,** Bit 7	**A_6,** Bit 6	**A_6,** Bit 5	**A_6,** Bit 4
	Output data							
0	**A_6,** Bit 3	**A_6,** Bit 2	**A_6,** Bit 1	**A_6,** Bit 0	**A,** Bit 3	**A,** Bit 2	**A,** Bit 1	**A,** Bit 0
1	**A_7,** Bit 3	**A_7,** Bit 2	**A_7,** Bit 1	**A_7,** Bit 0	**A_6,** Bit 7	**A_6,** Bit 6	**A_6,** Bit 5	**A_6,** Bit 4
2					**A_7,** Bit 7	**A_7,** Bit 6	**A_7,** Bit 5	**A_7,** Bit 4

- **Data mapping with byte alignment**

For this please activate the parameter "SDPL-0404D-xxxx Byte Align“ in the coupling module.

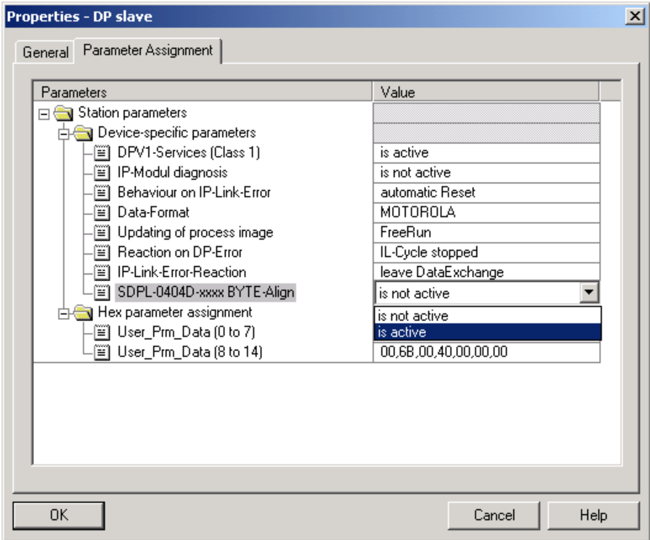


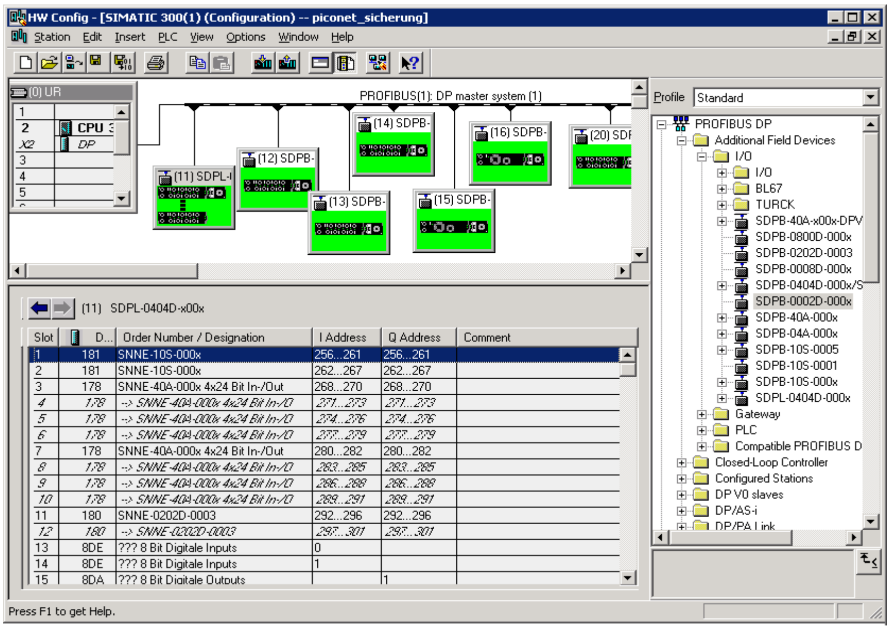
Figure 8-9: Activation of the byte alignment

Table 8-4: Process image with byte alignment

Byte	Input data							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	**idle**	**idle**	**idle**	**idle**	**A,** Bit 3	**A,** Bit 2	**A,** Bit 1	**A,** Bit 0
1	**A_6,** Bit 7	**A_6,** Bit 6	**A_6,** Bit 5	**A_6,** Bit 4	**A_6,** Bit 3	**A_6,** Bit 2	**A_6,** Bit 1	**A_6,** Bit 0
	Output data							
0	**A,** Bit 3	**A,** Bit 2	**A,** Bit 1	**A,** Bit 0	**idle**	**idle**	**idle**	**idle**
1	**A_6,** Bit 7	**A_6,** Bit 6	**A_6,** Bit 5	**A_6,** Bit 4	**A_6,** Bit 3	**A_6,** Bit 2	**A_6,** Bit 1	**A_6,** Bit 0
2	**A_7,** Bit 7	**A_7,** Bit 6	**A_7,** Bit 5	**A_7,** Bit 4	**A_7,** Bit 3	**A_7,** Bit 2	**A_7,** Bit 1	**A_7,** Bit 0

7 All other modules of the network are configured according to the coupling module on PROFIBUS-DP.

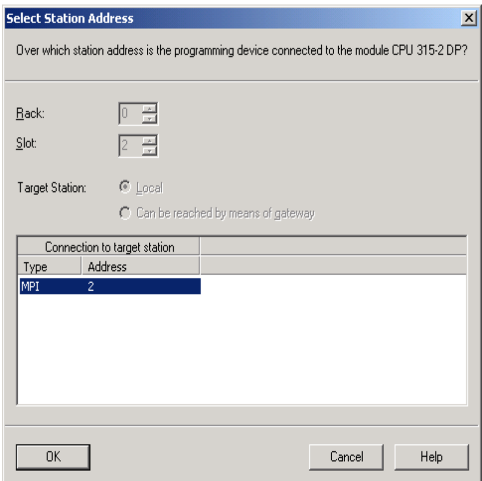
Figure 8-10: Model network



8 After having completed the network configuration it is stored and loaded to the controller via the command "PLC → Download".

9 For this select the destination station in the pop-up window and in the following window the node address for connection to the destination station. In this case it is the MPI with the node address 2.

Figure 8-11: Connection to the destination station



10 The destination station must be stopped and restarted to load the data.

8.5 Consistent data transmission

The communication of the S7 controller with the *piconet*® modules in the complex mode requires consistent data transmission.

In order to ensure consistent data transmission, the Siemens function blocks SFC14 and SFC15 are used within an organisation block. These are standard program components of the Simatic Manager.

8.5.1 Consistent data transmission via Siemens function modules SFC14 and SFC15

The function blocks SFC14 and SFC15 must be copied to the project and called up via the organisation block.

With the help of these two function blocks, the data for register communication are read and written consistently.

In this example, the function blocks are added to the OB1 of the project.

SFC14

The SFC14 is needed to read the module-specific settings and parameters in the register communication.

Figure 8-12: Function block SFC14

```
CALL  "DPRD_DAT"
 LADDR  :=W#16#116
 RET_VAL:=MW10
 RECORD :=P#M 100.0 BYTE 6
NOP   0
```

Call up the function block via the command *Call SFC14* .

Table 8-5: Parameters of the function block SFC14

Parameter name	Meaning	Notation
LADDR	Configured start address of the module from the input data memory of the controller.	The entry is written in an hexadecimal format. Example: W#16#14
RECORD	Target memory area of the CPU for the read user data. Here it is important to enter the data length of the user data (n byte).	Example: P#M 30.0 BYTE 3
RET_VAL	Target memory area of the CPU for a possible error code of the block.	e.g.: MW100

SFC15

The SFC15 is needed to write the module-specific settings and parameters to the register communication.

Figure 8-13: Function block SFC15

```
CALL  "DPWR_DAT"
 LADDR  :=W#16#116
 RECORD :=P#M 110.0 BYTE 6
 RET_VAL:=MW12
NOP   0
```

Call up the function block via the command *Call SFC14* .

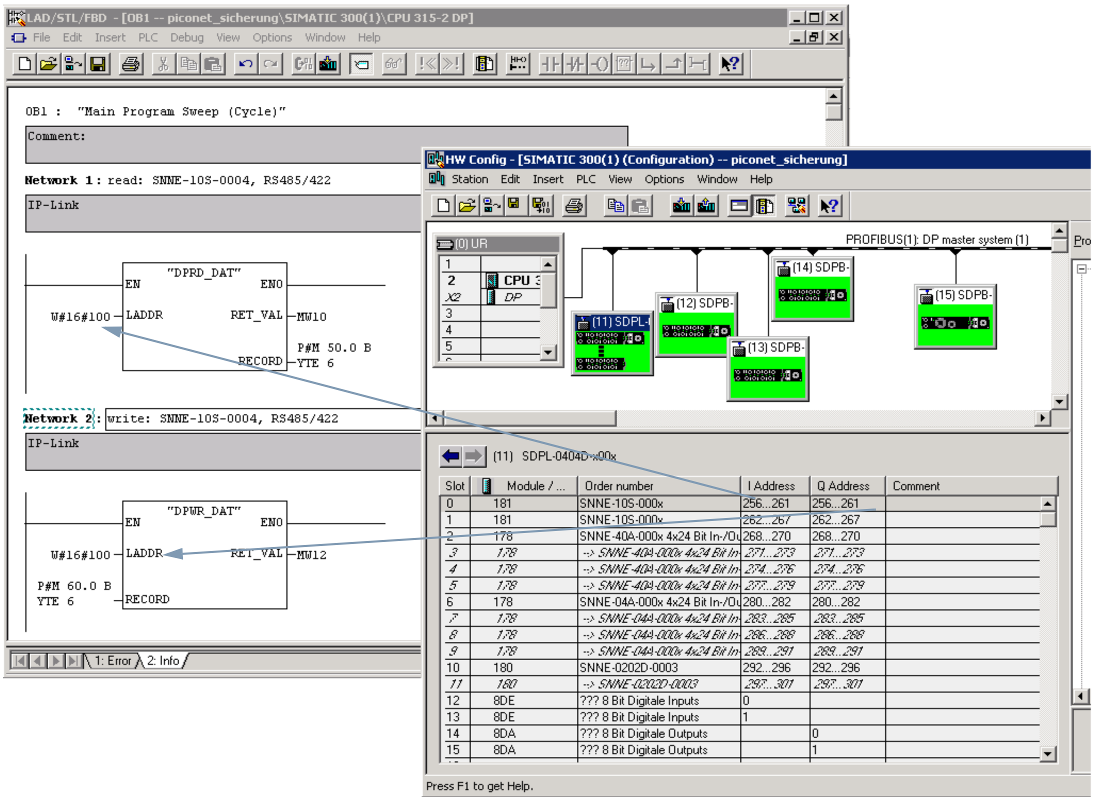
Table 8-6: Parameters of the function block SFC15

Parameter name	Meaning	Notation
LADDR	Configured start address of the module from the output data memory of the controller.	The entry is written in an hexadecimal format. Example: W#16#14
RECORD	Target memory area of the CPU for writing user data. Here it is important to enter the data length of the user data (n byte).	Example: P#M 50.0 BYTE 3
RET_VAL	Target memory area of the CPU for the error code of the block.	e.g.: MW120

The function blocks are stored in OB1 for each module. Here the "I addresses" and the "O addresses are assigned to certain MBs in the memory of the PLC.

The example (Figure 8-14:) shows the call-up for the first extension module, Modul A_1 (SNNE-10S-0004), in the IP-Link with the start address 256 dec. or 100 hex. in the input (I address) and output data area (O address) of the PLC.

Figure 8-14: SFC14 und SFC 15 for the module SNNE-10S-0004



The 6 input data bytes of the module are written to the flag area from MB 50.0 on (MB 50 up to MB 55), whereas the 6 output data bytes are written to the flag area from MB 60.0 on (MB 60 to MB 65).

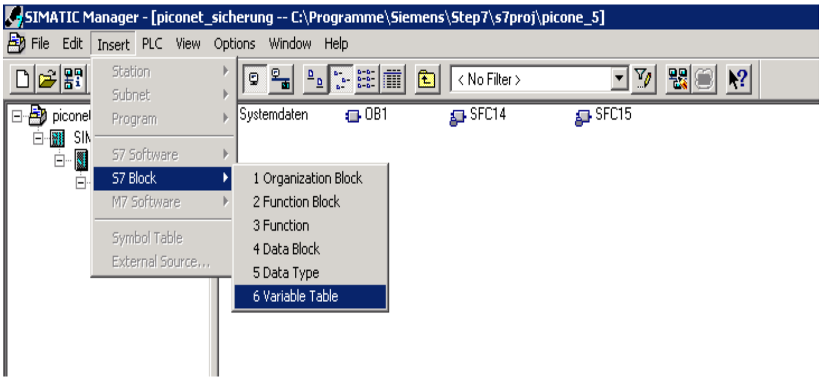
8.6 Module access via a table of variables

The variables, which are to be monitored and controlled (incl. the associated format specifications), are arranged in a table of variables.

This table enables direct monitoring and control of the network modules.

1 Click on „Insert → S7 block" in the Simatic Manager to add a variable table to your project:

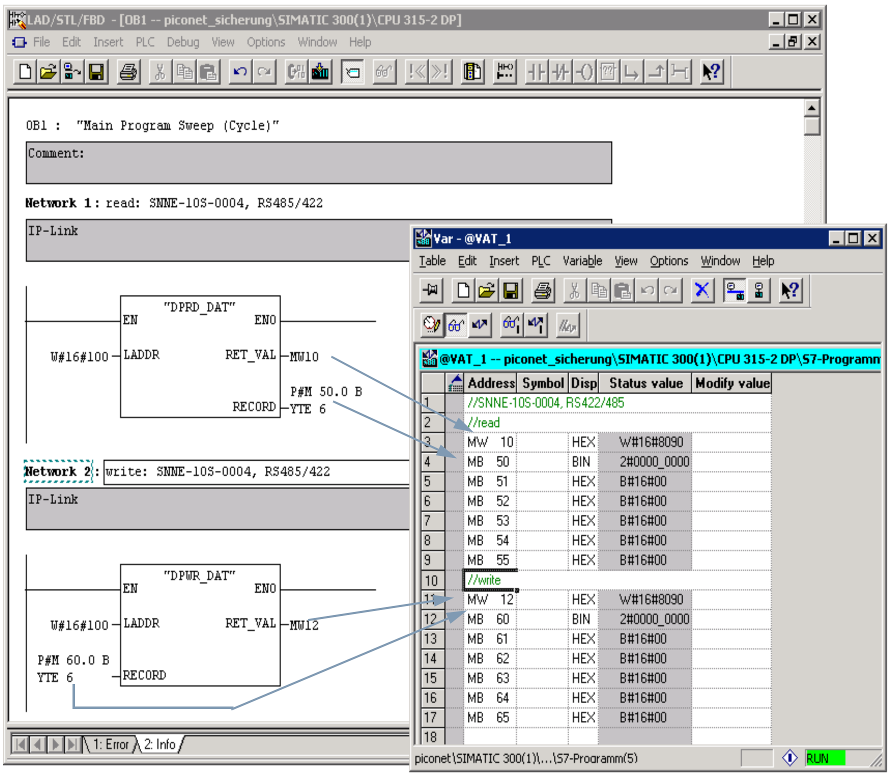
Figure 8-15: Inserting a table of variables



2 The structure of the variable table enables call-up, control and monitoring of all module data bytes.

3 The first flag word in the areas "//Read" and "//Write" of our example always contains the error word "RET_VAL" of the function blocks SFC 14 and SFC15.
The first MB in the "//Read" area is always the status byte, whereas the first MB in the "//Write" area always contains the control byte of the respective module.
Then come the input and output data of the modules.

Figure 8-16: SFCs in OB1 and in the table of variables



8.7 Parameterisation of the modules

The *piconet*® coupling module and most stand-alone modules allow clear-text parameterisation via the hardware configurator of the Simatic software with the help of the GSx files.

Some few byte-oriented stand-alone modules as well as all byte-oriented extension modules can only be parameterised via the control byte and/or the register communication or the software "I/O-ASSISTANT".

Attention

Generally parameterisation via GSx files has priority over parameterisation via register communication.
Following a power reset, the parameters set via register communication are overwritten by the PLC with the parameters from the GSx files.
Modules that enable clear-text parameterisation via GSx files should thus not be parameterised via register communication.

Note

The *piconet*® I/O module manual – TURCK document number D300777 (German) and D300778 (English) – contains a detailed description of the register communication as well as precise specifications of the register allocation of the individual *piconet*® modules.

8.8 Application example: counter module

In the following example, a *piconet*® counter module is to fulfil the following functions:

- Upwards counting
- Setting the output after 25 counted pulses
- Resetting the output after 50 counted pulses
- Resetting the counter to 0 after 100 counted pulses

8.8.1 Parameterisation of a counter as a stand-alone module via GSx file

The counter can be parameterised as a stand-alone module via clear-text parameterisation in the hardware configurator and the GSx file parameters.

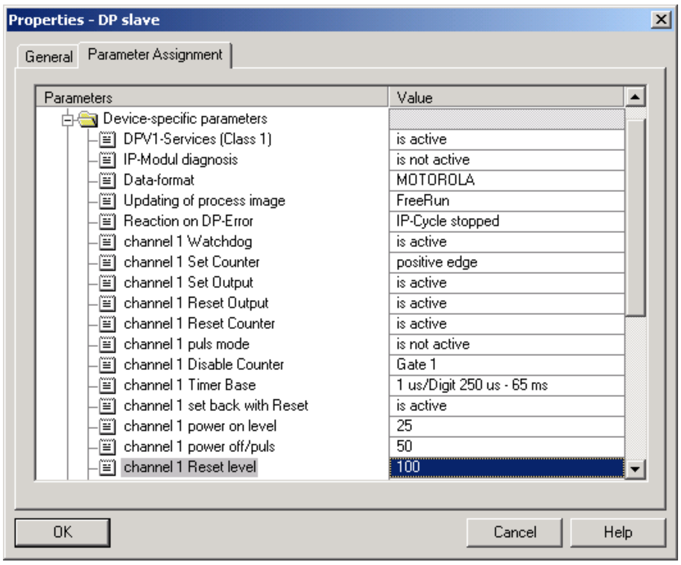
Attention
Modules that enable clear-text parameterisation via GSx files should thus not be parameterised via register communication. See Seite 1-20: “Parameterisation of the modules“.

Procedure

1 Double-click on the module to open the dialogue "Properties DP slave". In the register "Parameterisation" you will find the device-specific parameters of the module.

2 The following parameter settings are made:

Figure 8-17: Parameterisation via the hardware configurator



3 After having completed module configuration, store and load the modified parameters to the controller via the command "PLC → Download".

8.8.2 Parameterisation of a counter as an extension module via register communication

The module must be parameterised via the register communication or via the software "I/O ASSISTANT". All settings are made directly in the pre-defined registers.

Access to the registers of the *piconet*® modules is enabled via an upload of the register communication, a write or read command and the entry of the register number in the control byte of the respective module.

Writing a password to register 31

The password 0x1235 is written to register 31 of the counter module. The command to write to the register and thus to enable register communication via entry of the password is given by the control byte of the module.

Table 8-7: Writing the register 31 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	1	1	1	1	1

Bit 7 = 1 → Register communication is activated

Bit 6 = 1 → Register is written

Bit 5 up to Bit 0 = 31 (dec.) → Register number

The data bytes of the output data map contain the values which are to be transmitted to the respective register.

The password 0x1235 is written to data byte 0 (in our example: MB 119) and data byte 1 (in our example: MB 118) of the module. Data byte 0 presents the least significant byte and data byte 1 the most significant byte.

In our example 0x12 is written to MB 118 (data byte 1) and 0x35 to MB 119 (data byte 0).

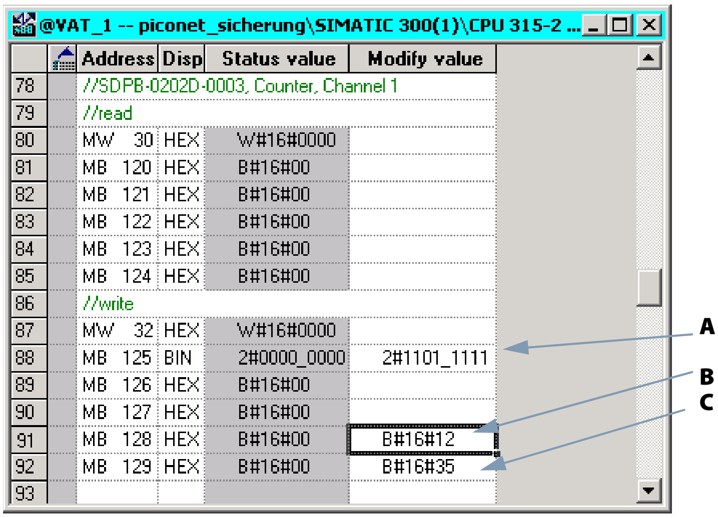


Figure 8-18: Password in register 31

A *Register 31*
B *Data byte 1*
C *Data byte 0*

Note

The mapping tables in chapter 5 of the manual show the assignment of the input and output data to the data bytes.

8.8.3 Activation of the switch-on, switch-off and reset thresholds

The counter function to set and reset the output and to reset the counter is activated in the feature register (register 32) of the module – see register assignment of the counter: chapter 12, "Technology modules" in the *piconet®* I/O module manual, TURCK document number: D300777 (German), D300778 (English) – .

Writing the feature register (register 32)

The parameters in register 32 are transferred to the module via the control byte as follows:

Table 8-8: Writing the register 32 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	0	0	0

Bit 7	= 1	→	Register communication is activated
Bit 6	= 1	→	Register is written
Bit 5 up to Bit 0	= 32 (dec.)	→	Register number

In our example, the configuration of the feature register of the counter module results in the following settings:

- Low byte → Data byte 0 (in the example MB 119):
  Value: 0x78

Table 8-9: Register 32, low byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	1	1	1	0	0	0

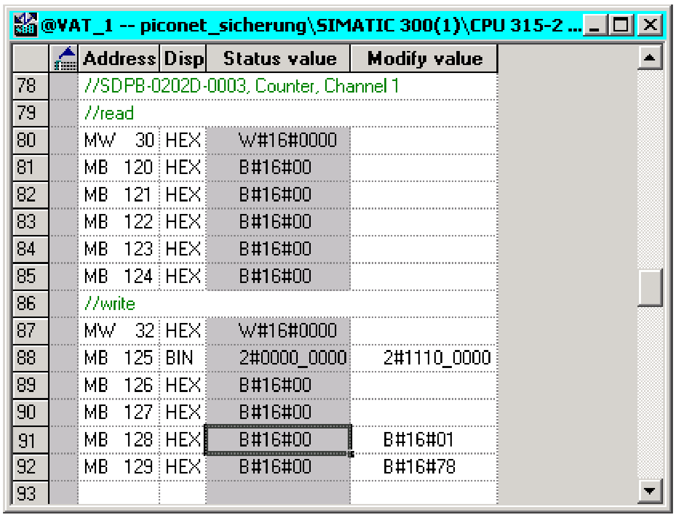
Bit 7	= 0	→	Pulse mode is activated.
Bit 6	= 1	→	Function to reset counter active
Bit 5	= 1	→	Function ro reset output active
Bit 4	= 1	→	Function to set output active
Bit 3	= 1	→	The counter is set upon a positive edge from CNT_SET Bit in the control byte
Bit 2	= 0	→	Watchdog active
Bit 1, Bit 0	= 0	→	reserved

- High byte → Data byte 1 (in the example MB 118): Value: 0x01

Table 8-10: Register 32, High byte	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
	0	0	0	0	0	0	0	1

Bit 15 up to Bit 11	= 0	→	reserved
Bit 10	= 0	→	Reset of output upon counter reset
Bit 9	= 0	→	Timer basis (pulse length register 41): 1 µs/digit (250 µs – 65 ms)
Bit 8	= 1	→	counter is inhibited if input gate = high (1)

Figure 8-19: Writing the register 32



Note

The settings made in register 32 are only activated after a power reset of the module.

8.8.4 Setting the switch-on, switch-off and reset thresholds

The threshold values are set in registers 35 to 40. The thresholds are stored permanently in the EEPROM of the module.

Note

Important for threshold value selection:
Switch-on threshold < Switch-off threshold < Reset threshold

Table 8-11: Treshold register

Register	Designation	Memory
R0	Switch-on threshold, least significant word	RAM
R1	Switch-on threshold, most significant word	RAM
R2	Switch-off threshold, least significant word	RAM
R3	Switch-off threshold, most significant word	RAM
R4	Reset threshold, least significant word	RAM
R5	Reset threshold, most significant word	RAM
....		
R35	Switch-on threshold, least significant word	EEPROM
R36	Switch-on threshold, most significant word	EEPROM
R37	Switch-off threshold, least significant word	EEPROM
R38	Switch-off threshold, most significant word	EEPROM
R39	Reset threshold, least significant word	EEPROM
R40	Reset threshold, most significant word	EEPROM
....		

Note

The settings in register 35 to 40 will only be activated after a power reset of the module and stored in the RAM of the module and thus in registers 0 to 5.

Attention

The threshold values should not be entered directly in registers 0 to 5.
In case of a power reset of the module, the actual changes in the RAM could be overwritten by invalid values of registers 35 to 40 stored in the EEPROM.

- **Switch-on threshold** (25 pulses):

Table 8-12: Writing the register 35 (control byte)

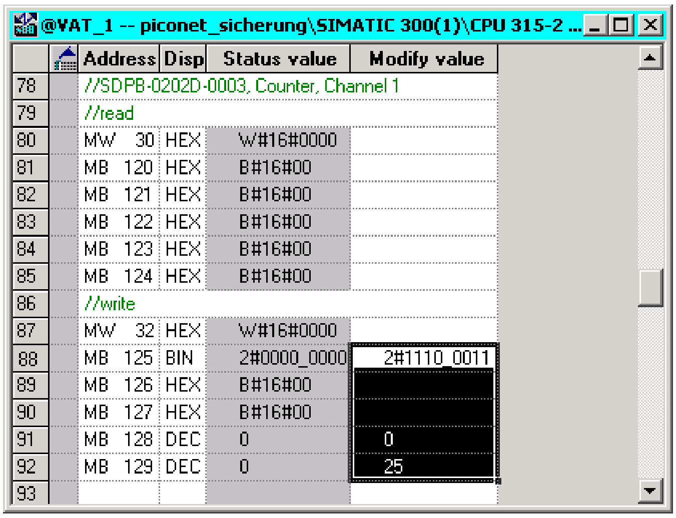
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	0	1	1

→ Data byte 0 (MB 119) = 25 dec. (0x19 hex.)

Table 8-13: Data byte 0 (MB 119)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	1	1	0	0	1

Figure 8-20: Switch-on threshold



- **Switch-off threshold** (50 pulses):

Table 8-14: Writing the register 37 (control byte)

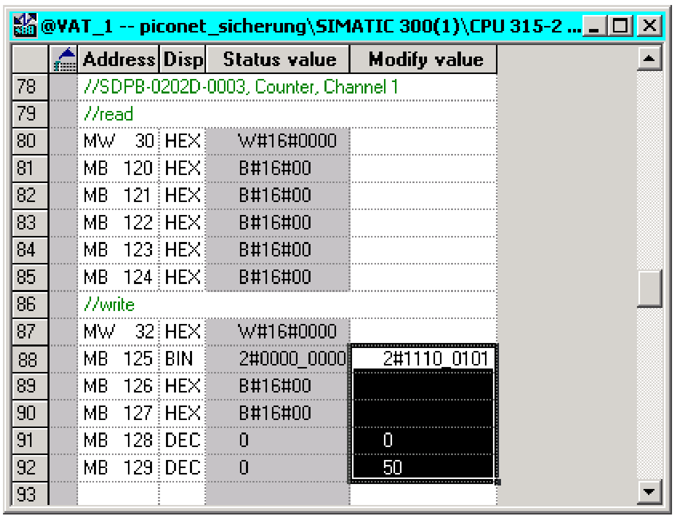
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	1	0	1

→ Data byte 0 (MB 119) = 50 dec. (0x32 hex.)

Table 8-15: Data byte 0 (MB 119)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	1	1	0	0	1	0

Figure 8-21: Switch-off threshold



- **Reset threshold** (100 pulses):

Table 8-16: Writing the register 39 (control byte)

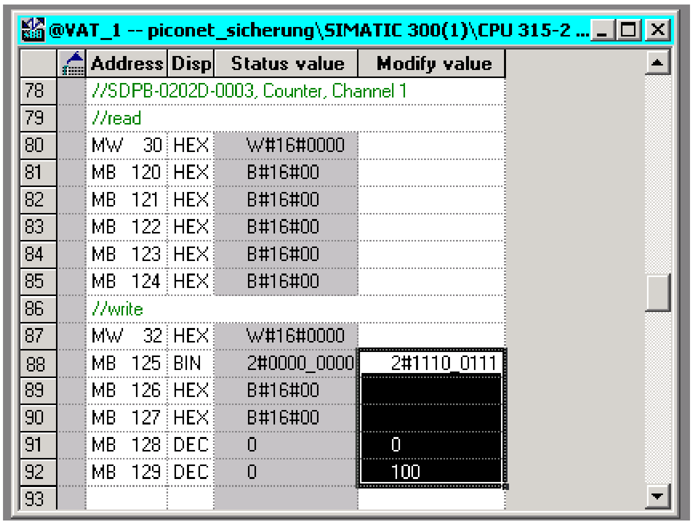
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	1	1	1

→ Data byte 0 (MB 119) = 100 dec. (0x64 hex.)

Table 8-17: Data byte 0 (MB 119)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	1	0	0	1	0	0

Figure 8-22: Reset threshold



Attention
Carry out a power reset to transfer settings from registers 35 to 40 to registers 0 to 5 of the module!

8.8.5 Enabling the internal counter functions

1 Prior to taken the counter in operation, the internal functions of the counter, i.e. the parameter settings, must first be enabled via the enable bit „EN_A“, bit 0 in the control byte.

2 For this the control byte is overwritten via the table of variables with 2#0000_0001.

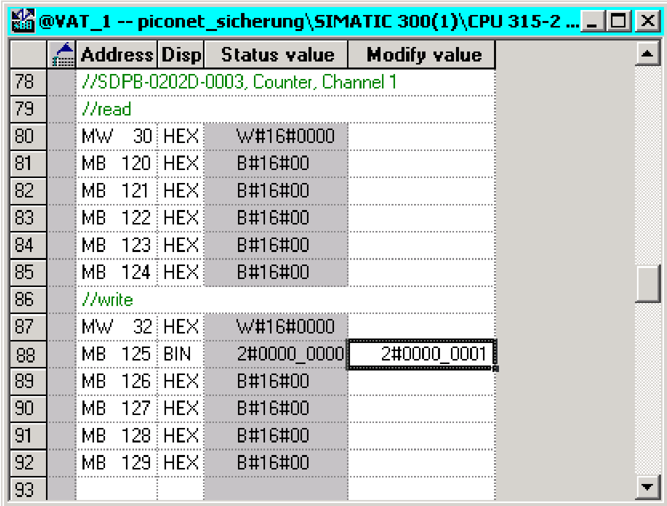


Figure 8-23: Enable via bit 0 of the control byte

8.8.6 Monitoring the counting procedure via the table of variables

The counter counts upwards. The counted pulses are indicated in data byte 0 of the counter, in this case MB 124.

Reaching the switch-on threshold

1 The counter counts up to 25 as pre-set and sets the output.

2 The LED Q1 at the connector "D" of the counter module illuminates green.

3 Bit 2 of the status byte (in our example: MB 120) is set.

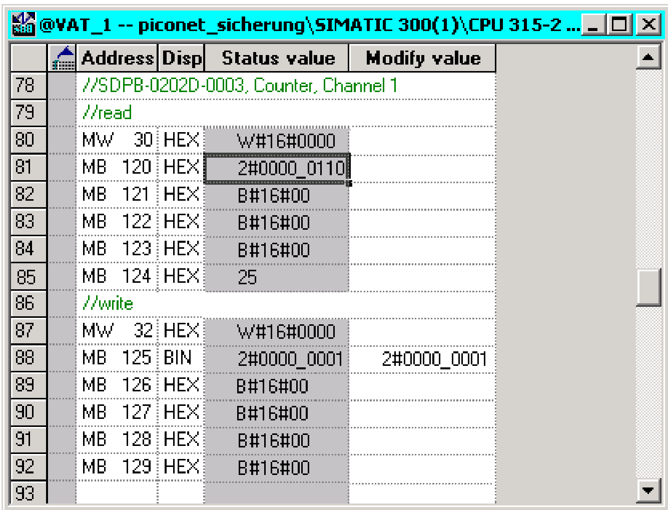


Figure 8-24: Setting the output

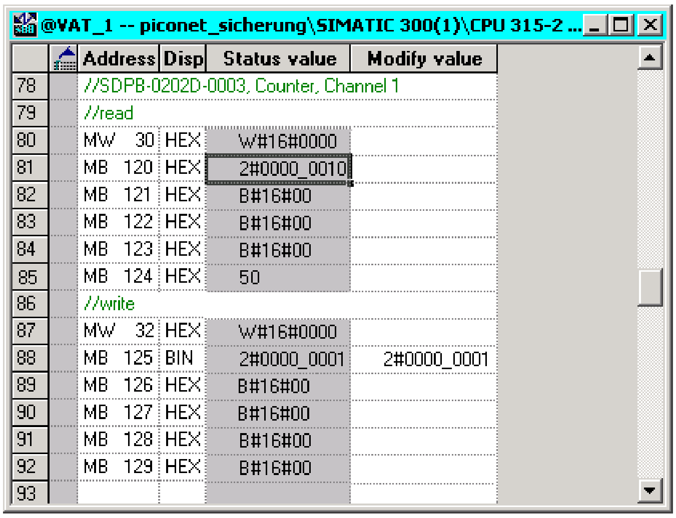
Reaching the switch-off/pulse value

4 The counter continues to count as pre-set. After 50 counted pulses the output is reset.

5 The LED Q1 at the connector "D" of the counter module turns off.

6 Bit 2 is reset again in the status byte (MB120).

Figure 8-25: Resetting the output



@VAT_1 -- piconet_sicherung\SIMATIC 300(1)\CPU 315-2 ...

	Address	Disp	Status value	Modify value
78	//SDPB-0202D-0003, Counter, Channel 1			
79	//read			
80	MW 30	HEX	W#16#0000	
81	MB 120	HEX	2#0000_0010	
82	MB 121	HEX	B#16#00	
83	MB 122	HEX	B#16#00	
84	MB 123	HEX	B#16#00	
85	MB 124	HEX	50	
86	//write			
87	MW 32	HEX	W#16#0000	
88	MB 125	BIN	2#0000_0001	2#0000_0001
89	MB 126	HEX	B#16#00	
90	MB 127	HEX	B#16#00	
91	MB 128	HEX	B#16#00	
92	MB 129	HEX	B#16#00	
93				

Reaching the reset threshold

7 After 100 counted pulses the count in data byte 0 (here: MB 124) is reset to "0".

8 The counter starts counting upwards from "0" on.

8.9 Application example: SSI module

The SSI module must be parameterised via the register communication or via the software "I/O ASSISTANT".

The multiturn encoders cannot be parameterised directly via the *piconet*® module. Configuration can only be carried out, e.g. für Kübler encoders, by the free "parameterisation tool for multiturn encoders", i.e. the software Ezturn®.

Note

When using the *piconet*® SSI module Sxxx-10S-x005, please ensure that the settings of the communication parameters (output format, single-/ or multi-turn, baud rate, data length etc) of the module and the used SSI encoder accord with each other.

This applies to all parameter settings of the feature register (register 32, the baud rate register (register 33) and the data length register (register 34).

Note

Detailed information on the module register assignment is contained in chapter 12 of the *piconet*® I/O module manual. „*piconet*® I/O Modules",TURCK document number: D300777 (German), D300778 (English).

8.9.1 Application example: Incremental encoder

In the following example the *piconet*® incremental encoder module is to fulfil the following functions:

- Setting the counter value to "0"
- Storing the count with an activated latch input,
- Inhibiting the counter with a low level at the gate input
- Switching from encoder interface mode to counter mode.

8.9.2 Parameterisation of an incremental encoder via GSx file

The incremental encoder cannot be parameterised via the GSx file at present.

The module can be parameterised either via the control byte and/or register communication or via the software "I/O-ASSISTANT".

8.9.3 Parameterisation of an incremental encoder via the control byte

Im many cases the incremental encoder can be parameterised directly via the control byte. Parameterisation via register communcation is thus not necessary.

Certain parameters must, however, be set by the register communication. All settings are made directly in the pre-defined registers.

Setting the counter value

The counter is set with a rising edge of the bit "CNT_SET", Bit 2 in control word" of the module to the value, which is determined by the process data in Reg0 and Reg1 (see "Mapping" in the bus-specific *piconet*® manual).

Setting the bit "CNT_SET“

This bit is set without register access simply via the control byte:

Table 8-18: Control byte

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RegAccess	-	-	-	-	CNT_ SET	EN_LAT_EXT/ RD_PERIOD	EN_LATC

Table 8-19: Setting "CNT_SET“

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	0	0	0	0	1	0	0

The value set in Reg 0 (MB 192) and Reg 1 (MB 191), in our example "0" is accepted.

Note

Please note that the bit "CNT_SET“ requires a reset prior to re-activation.

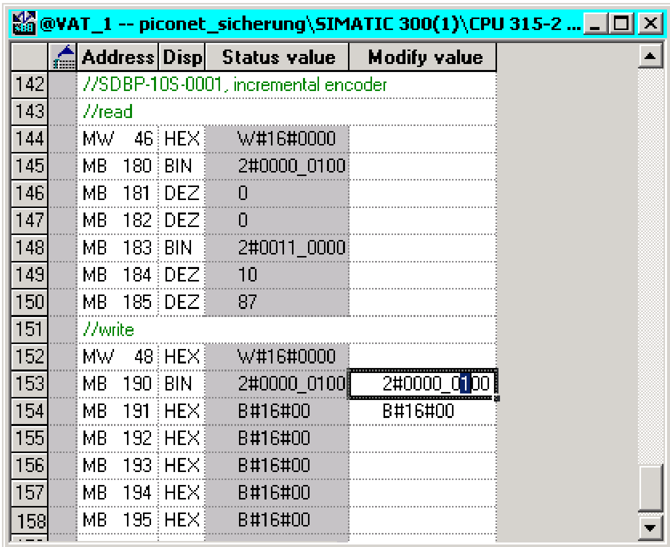


Figure 8-26: Setting the counter

The module counts from "0" on upwards or downwards, depending on the rotary direction of the connected encoder.

Data byte D2

The status of the input channels A, B and C as well as the inputs "Gate“ and "Latch“ is shown in data byte D2:

Table 8-20: Data byte D2

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	-	-	INPUT_A	INPUT_B	INPUT_C	INPUT_ERR	LATCH	Gate

Table 8-21: Description

Name	Description
INPUT_A	Status of input channel A
INPUT_B	Status of input channel B
INPUT_C	Status of input channel C
INPUT_ERR	Status of alarm channel
LATCH	Status of LATCH input at M12 connector
GATE	Status of GATE input at M12 connector

Storing the counter value

Data bytes D3 and D4 store the actual counter value upon activation of the external latch input.

Bit 1 „EN_LAT_EXT“ of the control byte is set to activate the external latch input.

Upon receipt of the first external latch pulse at the latch input and a valid "EN_LAT_EXT“ bit the counter value is saved. The following pulses have no influence on the latch register if the bit is set.

This bit is set without register access simply via the control byte in process data operation:

Table 8-22: Control byte

Bit	7	6	5	4	3	2	1	0
Name	RegAccess	-	-	-	-	CNT_SET	EN_LAT_EXT/ RD_PERIOD	EN_LATC

Table 8-23: Setting “EN_LAT_EXT“

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	1	0

If an edge change from "0" to "1" takes place at the external latch input during the counting procedure, the actual counter value is written to data bytes D3 and D4.

Bit 1 "LATCH“ in data byte D2 (see page 8-29) is set for a short moment to indicate activation of the latch input.

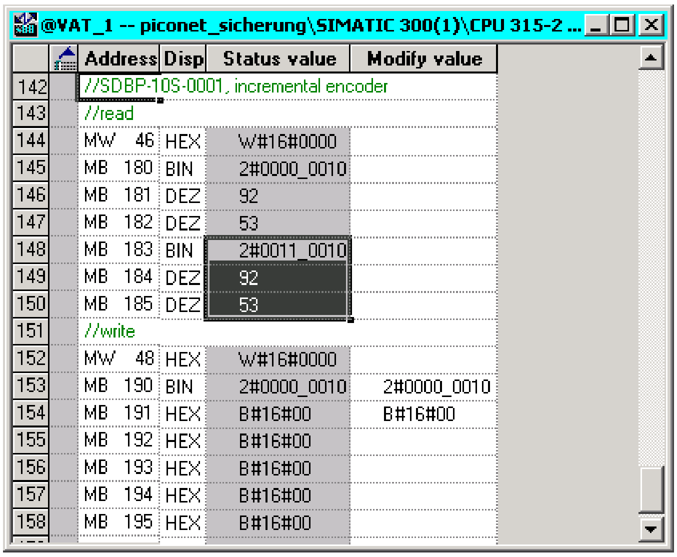


Figure 8-27: Storing the counter value

8.9.4 Parameterisation of an incremental encoder via register communication

In order to parameterise the module via register communication the password for register access must first be written to register 31 of the module.

Writing a password to register 31

The password 0x1235 is written to register 31 of the incremental encoder. The command to write to the register and thus to enable register communication via entry of the password is given by the control byte of the module.

Table 8-24: Writing the register 31 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	1	1	1	1	1

Bit 7 = 1 → Register communication is activated

Bit 6 = 1 → Register is written

Bit 5 up to Bit 0 = 31 (dec.) → Register number

The data bytes of the output data map contain the values which are to be transmitted to the respective register.

The password 0x1235 is written to Reg 0 (in our example: MB 192) and Reg 1 (in our example: MB 191) of the module, while Reg 0 presents the least significant byte and Reg 1 the most significant byte.

In our example 0x12 is written to MB 191 (Reg 1) and 0x35 to MB 192 (Reg 0).

Note

The mapping tables in chapter 5, „Data mapping of the *piconet*® modules" show the assignment of the input and output data to the data bytes.

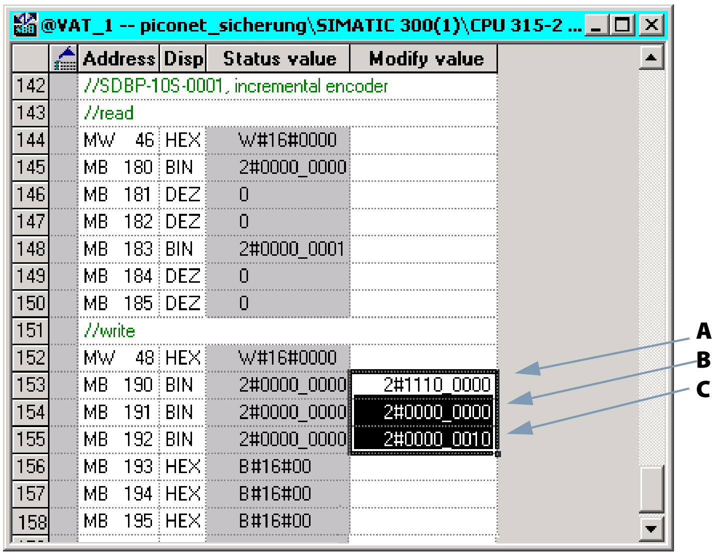


Figure 8-28: Password in register 31

A *Register 31*
B *Reg 1*
C *Reg 0*

Disabling the counter

The counter of the incremental encoder can be disabled either via a high or a low level at the gate input, depending on parameterisation.

In our example the counter is to be inhibited with a low level at the gate input.

The module is parameterisated via the feature register (R32).

Access to the registers of the module is enabled via call-up of the register communication, a write or read command and entry of the register number in the control byte of the respective module.

Writing the feature register (register 32)

The parameters in register 32 are transferred to the module via the control byte as follows:

Table 8-25: Writing the register 32 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	0	0	0

Bit 7	= 1	→	Register communication is activated
Bit 6	= 1	→	Register is written
Bit 5 up to Bit 0	= 32 (dec.)	→	Register number

In our example, the configuration of the feature register of the incremental encoder results in the following settings:

- Low byte → Data byte 0 (in the example MB 119):
  Value: 0x02

Table 8-26: Register 32, low byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	**1**	0

Bit 7	= 0	→	reserved
Bit 6, Bit 5	= 0	→	reserved
Bit 4	= 0	→	External latch function active
Bit 3, Bit 2	= 0	→	Status input (active low) is inserted in the status byte, bit 5
Bit 1	= 1	→	Counter is inhibited with a **low level** at the gate input
Bit 0	= 0	→	reserved

- High byte→ Reg 1 (in the example MB 191): Value: 0x80

Table 8-27: Register 32, High byte

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
1	0	0	0	0	0	0	0

Bit 15 = 0 → Encoder interface

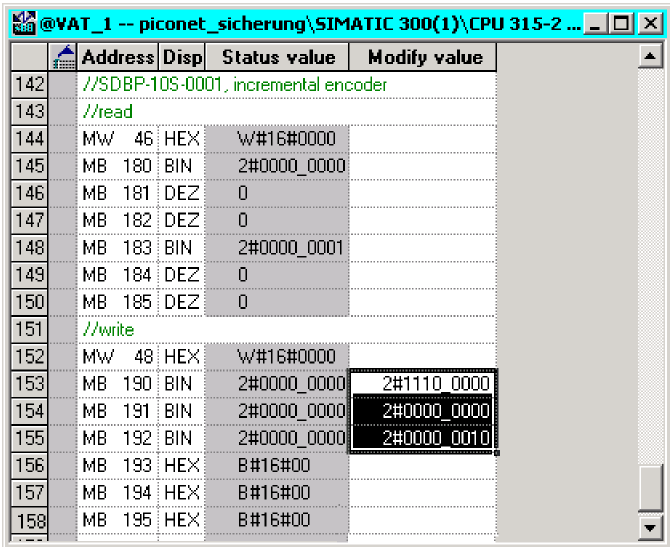
Bit 14

up to Bit 12 = 0 → reserved

Bit 11, Bit 10 = 0 → 4-fold evaluation of the encoder signals A, B, C

Bit 9, Bit 8 = 1 → reserved

Figure 8-29: Writing the register 32


@VAT_1 -- piconet_sicherung\SIMATIC 300(1)\CPU 315-2 ...

	Address	Disp	Status value	Modify value
142	//SDBP-10S-0001, incremental encoder			
143	//read			
144	MW 46	HEX	W#16#0000	
145	MB 180	BIN	2#0000_0000	
146	MB 181	DEZ	0	
147	MB 182	DEZ	0	
148	MB 183	BIN	2#0000_0001	
149	MB 184	DEZ	0	
150	MB 185	DEZ	0	
151	//write			
152	MW 48	HEX	W#16#0000	
153	MB 190	BIN	2#0000_0000	2#1110_0000
154	MB 191	BIN	2#0000_0000	2#0000_0000
155	MB 192	BIN	2#0000_0000	2#0000_0010
156	MB 193	HEX	B#16#00	
157	MB 194	HEX	B#16#00	
158	MB 195	HEX	B#16#00	

Note

The settings made in register 32 will only be activated after a power reset of the module, which will then return
to the process data exchange mode.

The counter is inhibited until a high level applies to the gate input.

After a new signal change at the input from "High"→ to "Low" the counter is re-inhibited.

Switching from: Encoder interface mode to counter mode

Changing from the encoder interface mode to the counter mode is also accomplished via the feature register (R32) of the module.

Writing the feature register (register 32)

The parameters in register 32 are transferred to the module via the control byte as follows:

Table 8-28: Writing the register 32 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	0	0	0

Bit 7 = 1 → Register communication is activated

Bit 6 = 1 → Register is written

Bit 5
up to Bit 0 = 32 (dec.) → Register number

In our example, the configuration of the feature register of the incremental encoder results in the following settings:

- Low byte → Data byte 0 (in the example MB 119):
  Value: 0x00

Table 8-29: Register 32, low byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0

Bit 7 = 0 → reserved

Bit 6, Bit 5 = 0 → reserved

Bit 4 = 0 → External latch function active

Bit 3, Bit 2 = 0 → Status input (active low) is inserted in the status byte, bit 5

Bit 1 = 0 → Counter is inhibited with a high level at the gate input

Bit 0 = 0 → reserved

- High byte → Reg 1 (in the example MB 191):
  Value: 0x80

Table 8-30: Register 32, High byte

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
1	0	0	0	0	0	0	0

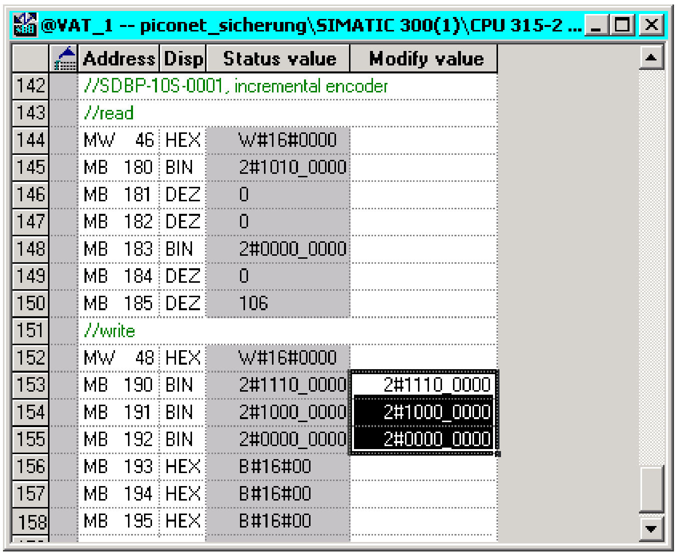
Bit 15 = 1 → Counter mode

Bit 14
up to Bit 12 = 0 → reserved

Bit 11, Bit 10 = 0 → 4-fold evaluation of the encoder signals
A, B, C

Bit 9, Bit 8 = 1 → reserved

Figure 8-30: Writing the register 32



Note

The settings made in R32 will only be activated after a power reset of the module, which will then return to process data exchange and the counter will continue to count.

9 Application example: acyclic services in *piconet*® with S7 (DP-V1)

9.1 General

The following chapter describes the application of acyclic services via DP-V1 which are used for example for reading-out the IP-Link configuration or for executing the parameter changes in *piconet*® stand-alone and extension modules, all this done parallel to the cyclic PROFIBUSP data exchange.

A Siemens PLC Simatic S7 with CPU 315-2AG10-0AB0 as well as the Simatic Manager V5.2 are used.

Note

For detailed information about dealing with the Siemens hard- and software, please read the respective manufacturer manuals/ documentation.

9.1.1 Example network

The network, described in this example corresponds to that in chapter 7 of this manual.

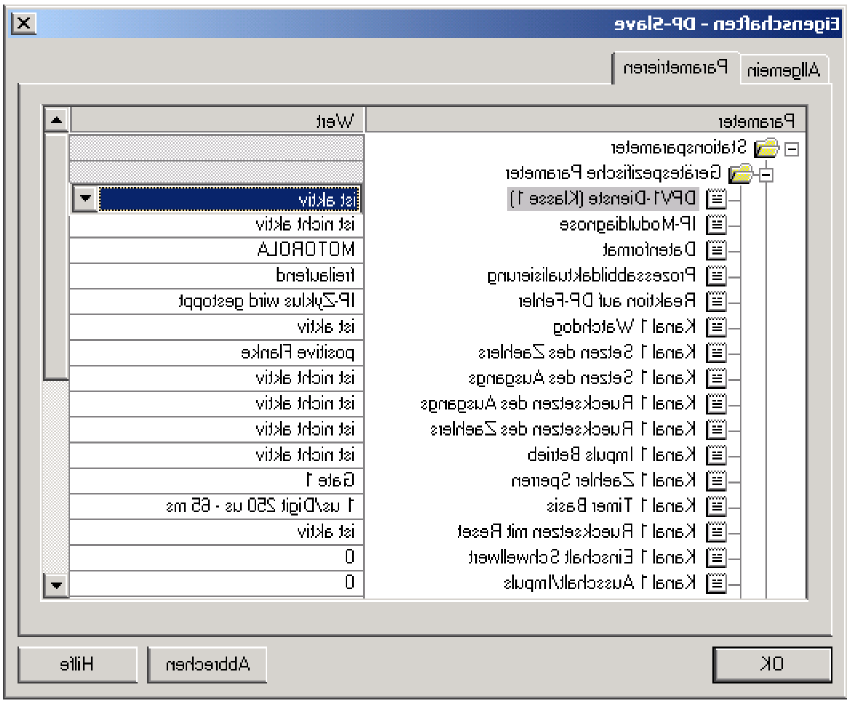
For creating a project in the Simatic Manager, please follow the steps page 8-3 to page 8-12 described in the last chapter.

9.1.2 Configuration of the *piconet*® slaves

In order to execute DP-V1-services, the piconet® modules have to be configured.

Activate the parameter „DP-V1-Services (Class 1)“ at every stand-alone or IP-Link coupling module.

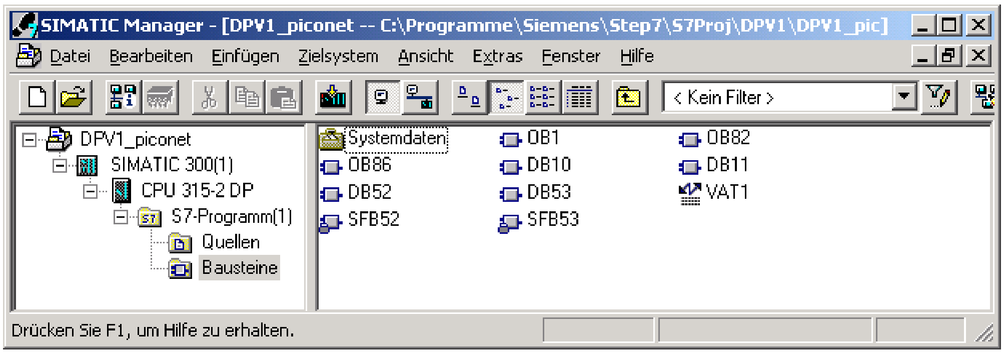
Figure 9-1: Activating the DP-V1 services



Load the changed hardware configuration into the PLC using the PLC → "Download..." command.

9.1.3 Structure of the S7 program

Figure 9-2: S7-program



1 In order to execute the DV-V1 services, the system function blocks SFB 52 and SFB53 with the corresponding data blocks DB52 and DB53 have to be called in OB1 of the program.

2 **SFB52**

Figure 9-3: SFB52

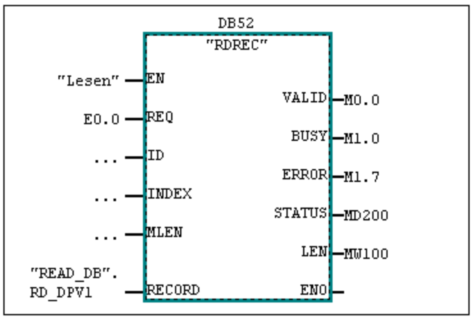


Table 9-1: Parameters of SFB52

Parameters	Data type	Description
Inputs		
EN	BOOL	Setting the parameter to "true" starts the function block.
REQ	BOOL	REQ = 1: Execute data transfer. Can be realized through a signal change at an additional digital input.
ID	DWORD	Logic address of the DP slave (station or module).
INDEX	INT	Number of the data set (slot number or register number)
MLEN	INT	Maximum length of the data to be read in bytes.
Outputs		
VALID	BOOL	New data set has been received and is valid.
BUSY	BOOL	BUSY = 1: The read operation is still running.
ERROR	BOOL	ERROR = 1: Error during read operation.
STATUS	DWORD	Call ID (bytes 2 and 3) or error code.
RECORD	ANY	Target area for the read data.

SFB53

Figure 9-4: SFB53

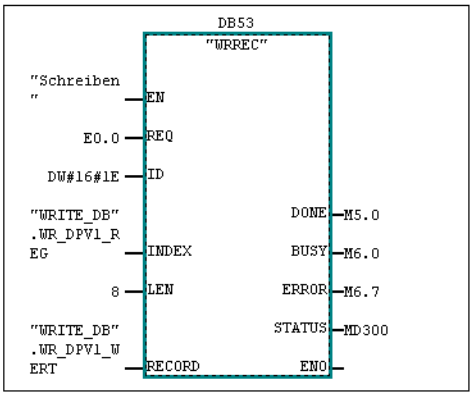


Table 9-2: Parameters of SFB53

Parameters	Data type	Description
Inputs		
REQ	BOOL	REQ = 1: Execute data transfer. Can be realized through a signal change at an additional digital input.
ID	DWORD	Logic address of the DP slave (station or module).
INDEX	INT	Number of the data set (slot number or register number)
LEN	INT	Maximum length of the data to be transferred in bytes.
Outputs		
DONE	BOOL	Data set has been transferred.
BUSY	BOOL	BUSY = 1: The write operation is still running.
ERROR	BOOL	ERROR = 1: Error during write operation.
STATUS	DWORD	Call ID (bytes 2 and 3) or error code.
RECORD	ANY	Data set

3 In this example the data blocks DB10 and DB11 are used to handle the in- and output bytes of the respective module.

4 OB82 and OB83 are used to avoid that the CPU changes to operation mode STOP in case of an error.

5 The variable table VAT1 serves for monitoring and controlling the module's in- and output data .

9.1.4 Reading out the IP-Link configuration

The example shows the possibility of reading out the IP-Link configuration via acyclic services.

The IP-Link configuration is read-out of the IP-Link-coupling module. For each connected extension module, one word is transferred. This word contains, for all analog and byte-oriented module, the modules type (for example 5109 = Incremental Encoder) and for all digital extension module the module size and the module type.

Note
The access to the coupling module and to the module configuration of the IP-Link is done via the module's diagnostic address (here: 2041dez = 7F9hex).

The IP-Link configuration with, in this example, a maximum of 23 connected modules is read out using Slot 0, Index 9 (see also chapter 3, Acyclic services via DPV1).

Table 9-3: Reading out the module configuration

Slot number	Index	Byte	Description
0	9	0-1	Description coupling module
	9	2-3	Module 1
	...	...	...
	9	46-47	Module 23

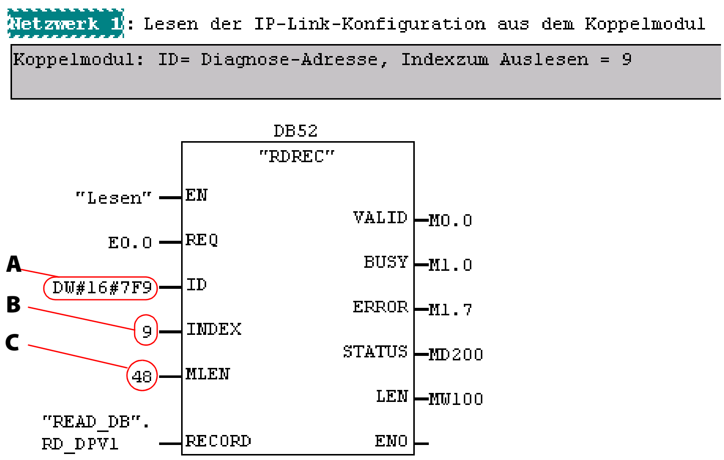


Figure 9-5: Reading out the module configuration via SFB52

A = 7F9 → logic address of the coupling module In this case - for reading the IP-Link configuration - the module's diagnostic address has to be set, **not** the process data address.

B = setting the index-no.

C = the max. number of expected data bytes is 48 Byte.

Note
Load the changed OB1 to the PLC using "PLC → Download".

Reading out the IP-Link configuration via variable table

Set M40.1 to "true" in order to read out the IP-Lin-configuration. The function block is started.

A signal change at input E0.0 enables the data transfer.

Figure 9-6: Reading out the IP-Link configuration via variable table

VAT1 -- @DPV1_piconet\SIMATIC 300(1)\CPU 315-2 DP\S7-Programm(1) ...

	Operand	Symbol	Anzei	Statuswert	Steuerwert
1	M 40.0	"Schreiben"	BOOL	false	false
2	DB11.DBW 4	"WRITE_DB".WR_DPV1_REG	DEZ	0	0
3	DB11.DBB 0	"WRITE_DB".WR_DPV1_WERT[0]	HEX	B#16#00	B#16#00
4	DB11.DBB 1	"WRITE_DB".WR_DPV1_WERT[1]	HEX	B#16#00	B#16#00
5	DB11.DBB 2	"WRITE_DB".WR_DPV1_WERT[2]	HEX	B#16#00	B#16#00
6	DB11.DBB 3	"WRITE_DB".WR_DPV1_WERT[3]	HEX	B#16#1E	B#16#1E
7	M 40.1	"Lesen"	BOOL	true	true
8	DB10.DBB 0	"READ_DB".RD_DPV1[0]	HEX	B#16#36	
9	DB10.DBB 1	"READ_DB".RD_DPV1[1]	HEX	B#16#01	
10	DB10.DBB 2	"READ_DB".RD_DPV1[2]	HEX	B#16#0A	
11	DB10.DBB 3	"READ_DB".RD_DPV1[3]	HEX	B#16#00	
12	DB10.DBB 4	"READ_DB".RD_DPV1[4]	HEX	B#16#86	
13	DB10.DBB 5	"READ_DB".RD_DPV1[5]	HEX	B#16#17	
14	DB10.DBB 6	"READ_DB".RD_DPV1[6]	HEX	B#16#72	
15	DB10.DBB 7	"READ_DB".RD_DPV1[7]	HEX	B#16#17	
16	DB10.DBB 8	"READ_DB".RD_DPV1[8]	HEX	B#16#82	
17	DB10.DBB 9	"READ_DB".RD_DPV1[9]	HEX	B#16#0C	
18	DB10.DBB 10	"READ_DB".RD_DPV1[10]	HEX	B#16#F0	
19	DB10.DBB 11	"READ_DB".RD_DPV1[11]	HEX	B#16#0C	
20	DB10.DBB 12	"READ_DB".RD_DPV1[12]	HEX	B#16#11	
21	DB10.DBB 13	"READ_DB".RD_DPV1[13]	HEX	B#16#00	
22	DB10.DBB 14	"READ_DB".RD_DPV1[14]	HEX	B#16#14	
23	DB10.DBB 15	"READ_DB".RD_DPV1[15]	HEX	B#16#00	
24	DB10.DBB 16	"READ_DB".RD_DPV1[16]	HEX	B#16#00	
25	DB10.DBB 17	"READ_DB".RD_DPV1[17]	HEX	B#16#00	
26	DB10.DBB 18	"READ_DB".RD_DPV1[18]	HEX		

In this example, the module configuration can be read from bytes „DB10.DBB 0“ to „DB10.DBB 14":

Table 9-4: Reading out the module configuration

Byte	Value	Description
0 1	B#16#36 B#16#01	Coupling module: Module type: $0136_{hex} = 310_{dez}$ → SDPL-0404D-000x (PROFIBUS-DP-coupling module)
2 3	B#16#0A B#16#00	In- and outputs of the coupling module: Evaluation (see page 3-11): – As byte 3 = 00, the module is a bit-oriented module. – Byte 2 = $0A_{hex} = 000\mathbf{00101}_{bin}$ Bit 1 and 0 = $01_{bin} = 2_{dec.}$ → 2 outputs Bit 3 and 2 = $01_{bin} = 2_{dec.}$ → 2 inputs, – As Bit 4 = 0, which means "number of in-and outputs multiplied with 2 Bit", the module provides 4 in- and 4 outputs.
4 5	B#16#86 B#16#17	1 Extension module: Module type: $1786_{hex} = 6022_{dez}$ → SNNE-10S-0004 (RS485/422-module)
6 7	B#16#72 B#16#17	2 Extension module: Module type: $1772_{hex} = 6002_{dez}$ → SNNE-10S-0003 (RS232-module)
8 9	B#16#82 B#16#0C	3 Extension module: Module type: $0C82_{hex} = 3202_{dez}$ → SNNE-40A-0009 (PT100-module)
10 11	B#16#F0 B#16#0C	4 Extension module: Module type: $0CF0_{hex} = 3312_{dez}$ → SNNE-40A-0004 (thermo-module)
12 13	B#16#DE B#16#05	5 Extension module: Module type: $05DE_{hex} = 1502_{dez}$ → SNNE-0202D-000x (counter-module)
14 15	B#16#35 B#16#00	6 Extension module: Evaluation (see page 3-11): – As byte 3 = 00, the module is a bit-oriented module. – Byte 2 = $35_{hex} = 0\mathbf{0110101}_{bin}$ Bit 1 and 0 = $01_{bin} = 2_{dec.}$ → 2 outputs Bit 3 and 2 = $01_{bin} = 2_{dec.}$ → 2 inputs, – Bit 4 = 1, which means: number of in- and outputs × 8 Bit → Module type: SNNE-0808D-000x (combi module with 8 in- and 8 outputs) – Bit 6 and 5 = always „01“ for 8/8-combi modules
16 17	B#16#11 B#16#00	7 Extension module: Evaluation (see page 3-11): – As byte 3 = 00, the module is a bit-oriented module. – Byte 2 = $11_{hex} = 000\mathbf{1}000\mathbf{01}_{bin}$ Bit 1 and 0 = $01_{bin} = 2_{dec.}$ → 2 outputs – Bit 4 = 1, which means: number of in- and outputs × 8 Bit → Module type: SNNE-0808D-000x (8-channel output module)
17 18	B#16#00 B#16#00	no further modules in the network
...	...	...

9.1.5 IP-Link reset

If, via the GSD-file, the parameter "Restart after IP-Link error" is set to "manual reset", the IP-Link reset can be activated with the following DP-V1 parameters (After an IP-Link interruption and it's elimination, the IP-Link can be restarted):

Table 9-5: IP-Link reset

Slot number	Index	Byte	Description	
0	99	0	2	See Figure 9-8: VAT1 – IP-Link reset
0	99	1	1	
0	99	2	0	
0	99	3	0	

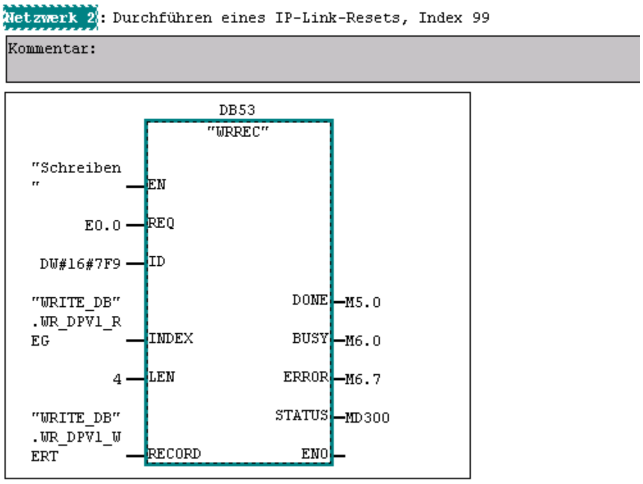
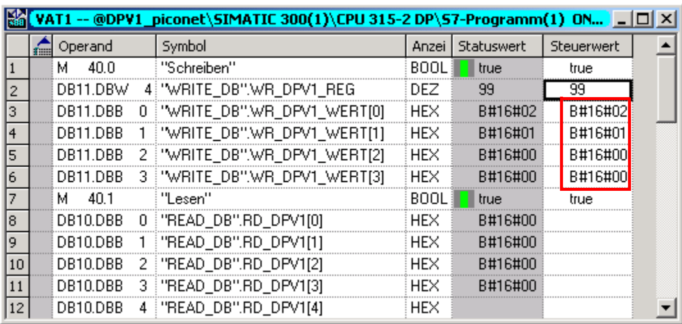


Figure 9-7: OB 1 – IP-Link reset

During the IP-Link reset, the index-no. 99 and the entries necessary for the reset are transferred to the SFB53 via the variable table. In this case, the activation of the function block with "true" at DB11.DBW 4 as well as the enable via a positive signal at E0.0 is important.



VAT1 -- @DPV1_piconet\SIMATIC 300(1)\CPU 315-2 DP\S7-Programm(1) ON...

	Operand	Symbol	Anzei	Statuswert	Steuerwert
1	M 40.0	"Schreiben"	BOOL	true	true
2	DB11.DBW 4	"WRITE_DB".WR_DPV1_REG	DEZ	99	99
3	DB11.DBB 0	"WRITE_DB".WR_DPV1_WERT[0]	HEX	B#16#02	B#16#02
4	DB11.DBB 1	"WRITE_DB".WR_DPV1_WERT[1]	HEX	B#16#01	B#16#01
5	DB11.DBB 2	"WRITE_DB".WR_DPV1_WERT[2]	HEX	B#16#00	B#16#00
6	DB11.DBB 3	"WRITE_DB".WR_DPV1_WERT[3]	HEX	B#16#00	B#16#00
7	M 40.1	"Lesen"	BOOL	true	true
8	DB10.DBB 0	"READ_DB".RD_DPV1[0]	HEX	B#16#00	
9	DB10.DBB 1	"READ_DB".RD_DPV1[1]	HEX	B#16#00	
10	DB10.DBB 2	"READ_DB".RD_DPV1[2]	HEX	B#16#00	
11	DB10.DBB 3	"READ_DB".RD_DPV1[3]	HEX	B#16#00	
12	DB10.DBB 4	"READ_DB".RD_DPV1[4]	HEX		

Figure 9-8: VAT1 – IP-Link reset

Note

The coupling module will always execute an IP-Link reset as long as the input signal will be reset.

9.1.6 Register communication via DP-V1-service

Setting and activating switch-on, - off and reset values in the *piconet*® counter extension module will be used as examples for explaining the register communication via DP-V1.

The counter module fulfills the following functions:

- Setting an output after 10 counted pulses
- Resetting the output after 15 counted pulses
- Resetting the counter to 0 after 20 counted pulses

Writing the password into register 31

The password 0x1235 will be written into register 31 of the counter module.

In the example network, the counter module is the fifth module n the IP-Link. module n the IP-Link. The communication is done via it's process data address defined in the hardware configuration (292_{dez}).

This means results in the following entries fir OB1:

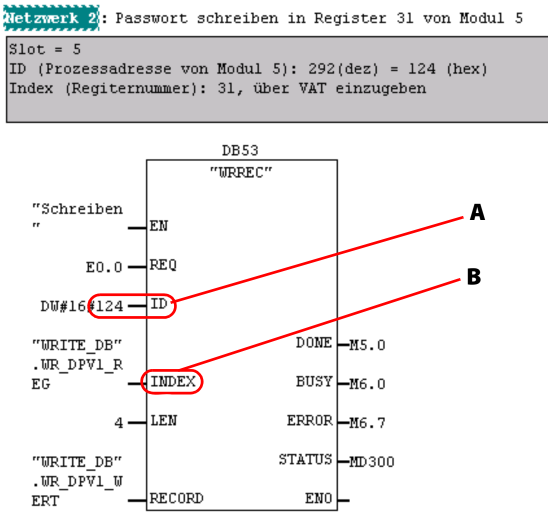
ID: $292_{dez} = 124_{hex}$

Index: Register number → Index = 31 (to be entered in VAT1).

Figure 9-9: OB1 – write password

A process data address of the counter module

B will be entered to VAT1 in DB11.DBW 4



In order to take over the changes done in OB1, the data block has to be sent again to the PLC.

The command for writing the register and the enabling of the register communication by entering the password is then done via variable table:

Figure 9-10: Entering the password

VAT1 -- @DPV1_piconet\SIMATIC 300(1)\CPU 315-2 DP\S7-Programm(1) ON...

	Operand	Symbol	Anzei	Statuswert	Steuerwert
1	M 40.0	"Schreiben"	BOOL	false	true
2	DB11.DBW 4	"WRITE_DB".WR_DPV1_REG	DEZ	32	31
3	DB11.DBB 0	"WRITE_DB".WR_DPV1_WERT[0]	HEX	B#16#02	B#16#35
4	DB11.DBB 1	"WRITE_DB".WR_DPV1_WERT[1]	HEX	B#16#00	B#16#12
5	DB11.DBB 2	"WRITE_DB".WR_DPV1_WERT[2]	HEX	B#16#00	B#16#00
6	DB11.DBB 3	"WRITE_DB".WR_DPV1_WERT[3]	HEX	B#16#00	B#16#00
7	M 40.1	"Lesen"	BOOL	false	true
8	DB10.DBB 0	"READ_DB".RD_DPV1[0]	HEX	B#16#00	
9	DB10.DBB 1	"READ_DB".RD_DPV1[1]	HEX	B#16#00	
10	DB10.DBB 2	"READ_DB".RD_DPV1[2]	HEX	B#16#00	
11	DB10.DBB 3	"READ_DB".RD_DPV1[3]	HEX	B#16#00	
12	DB10.DBB 4	"READ_DB".RD_DPV1[4]	HEX		

Activating the switch-on, -off and reset-threshold values

Activating the counter functions for setting and resetting the output as well as for resetting the counter is done in the module's feature register (register 32) - see for the counter's register assignment : Chapter 12 "Technology modules" in manual „*piconet*® I/O-modules´", TURCK-documentation number: D300778 (English) –.

The assignment of the Feature register defines the following settings for this example:

- Low-byte → data byte „DB11.DBB 1"
  Value: **0x78**:

Table 9-6: Register 32, Low-byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	1	1	1	0	0	0

Bit 7	= 0	→	pulse mode inactive.
Bit 6	= 1	→	function for the counter reset active
Bit 5	= 1	→	function for resetting the output active
Bit 4	= 1	→	function for resetting the output active
Bit 3	= 1	→	Setting the counter is done with a positive edge at bit CNT_SET in the control-byte
Bit 2	= 0	→	watchdog active
Bit 1, Bit 0	= 0	→	reserved

- High-byte → data byte „DB11.DBB 0"
  Value: **0x01**

Table 9-7: Register 32, High-byte

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0	0	0	0	0	0	0	1

Bit 15 to Bit 11	= 00	→	reserved
Bit 10	= 0	→	resetting the output is done with a counter reset
Bit 9	= 0	→	timer base (pulse length register 41): 1 µs/Digit (250 µs - 65 ms)
Bit 8	= 1	→	counter is disabled if input Gate = high (1)

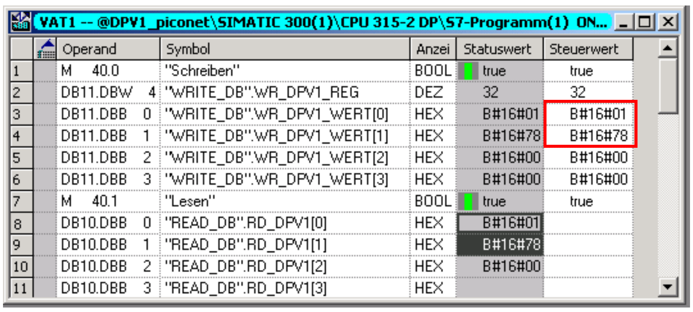


Figure 9-11: Writing register 32

Note

The settings done in register 32 will only be valid after a voltage reset of the module.

Note

In order to control the changes in register 32, a read command with „true“ in M40.1 (line 7 in the figure above), the actual register value can be read out. If this value fits to the value entered before, the write command was successful.

9.1.7 Enabling the internal counter functions

Before enabling the counter, the internal counter functions (which means the parameter settings) have to be enabled using the enable-bit „EN_A“, bit 0 in the module's control byte.

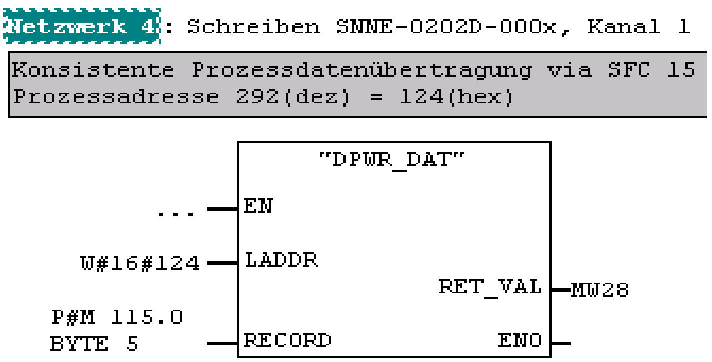
The access to the control byte is done via the process data, which makes a consistent data transfer necessary (see also chapter 8, Consistent data transmission, page 8-13).

To achieve this, function block SFC15 is added to the project.

SFC15

In register communication, SFC15 serves for writing module-specific settings and parameters.

Figure 9-12: function block SFC15



Add the function block SFC15 using the *Call SFC15* command.

Table 9-8: Parameters of function block SFC15

Parameter name	Meaning	Notation
LADDR	Configured start address from the process image output area of the PLC to which the data will be written.	Example: W#16#124
RECORD	Source area for the user data to be written. The entry of the user data data length (in byte) is important.	Example: P#M 115.0 BYTE 5
RET_VAL	If an error occurs while the function is active, the return value contains an error code.	for example: MW 28

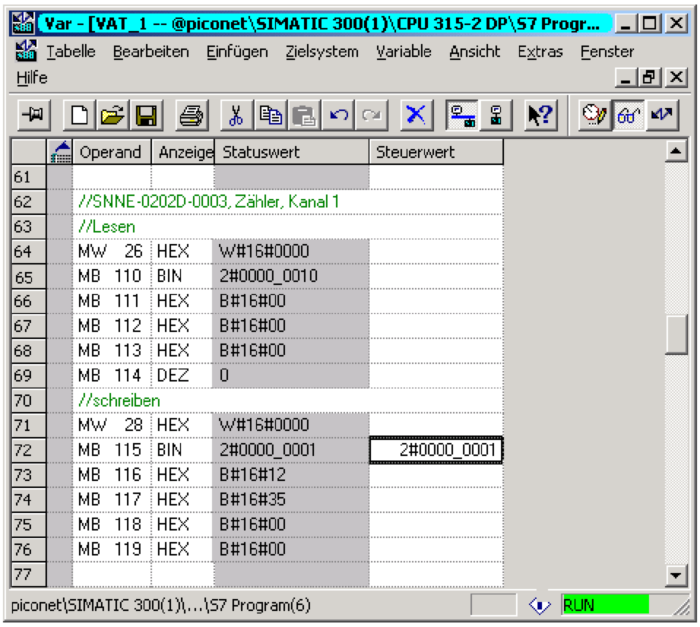
In order to have access to the module's control byte, the following entries have to be added to the variable table:

The first byte in the "//write"-area is always the control byte of the respective module.

The modules' in- and output data follow this byte.

6 The control byte is written with 2#0000_0001 using the variable table.

Figure 9-13: Enabling via bit Bit 0 of the control bytes

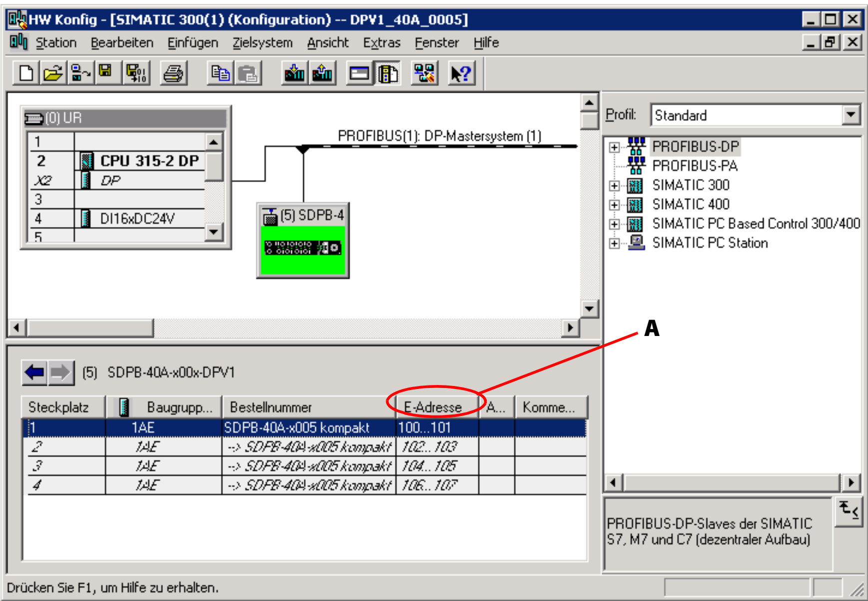


9.1.8 Configuration

The function blocks used work with the module's logical address.

Figure 9-14: logical address of the module

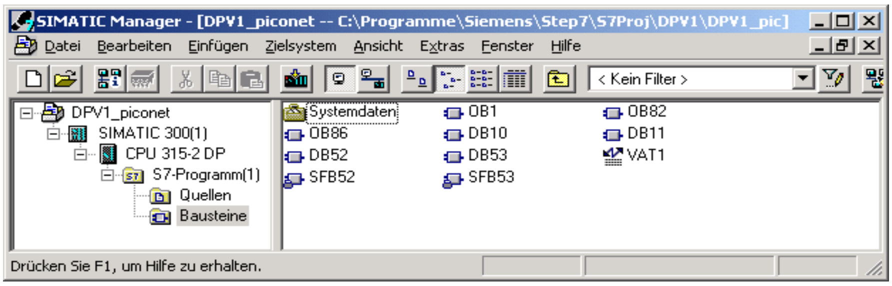
A *logical address*



9.1.9 Programming

The following function blocks are necessary for programming the PLC:

Figure 9-15: Function blocks of the program



Writing the password into register 31

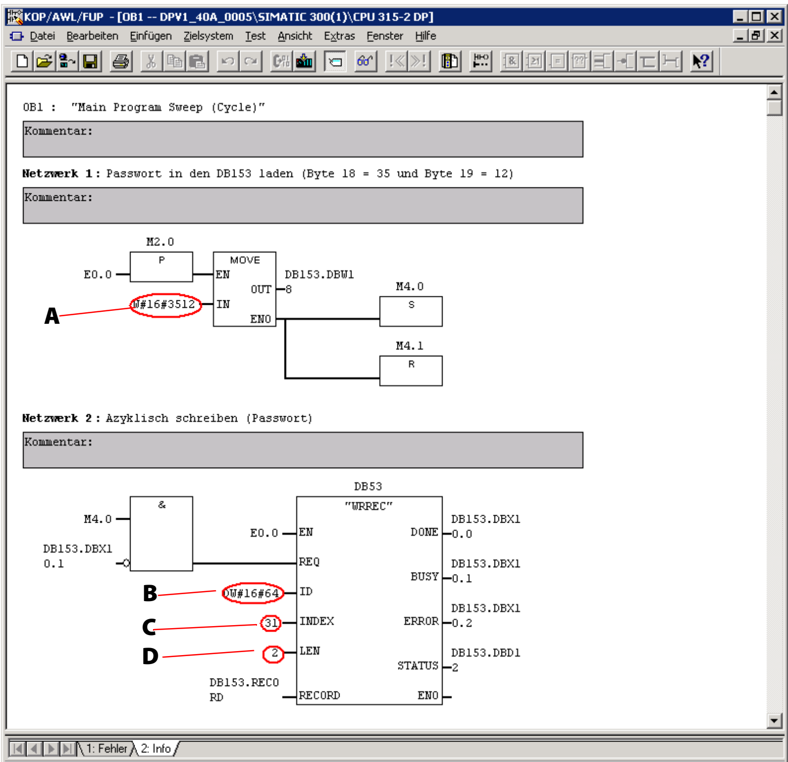
Input parameters:

Table 9-9: Input parameters

Parameters	Meaning
ID	Logical address of channel 1 $100_{dez} = 64_{hex}$
INDEX	Register number (see section DP-V1-functions (page 3-11)): Index = 31
LEN	Length of the data to be transferred: (Register = 2 byte)
RECORD	Memory area for the read data

Figure 9-16: Password in register 31,

A *Password 1235*
B *logical address of the module of the module*
C *register-number*
D *Length of the data to be transferred: 2 byte*



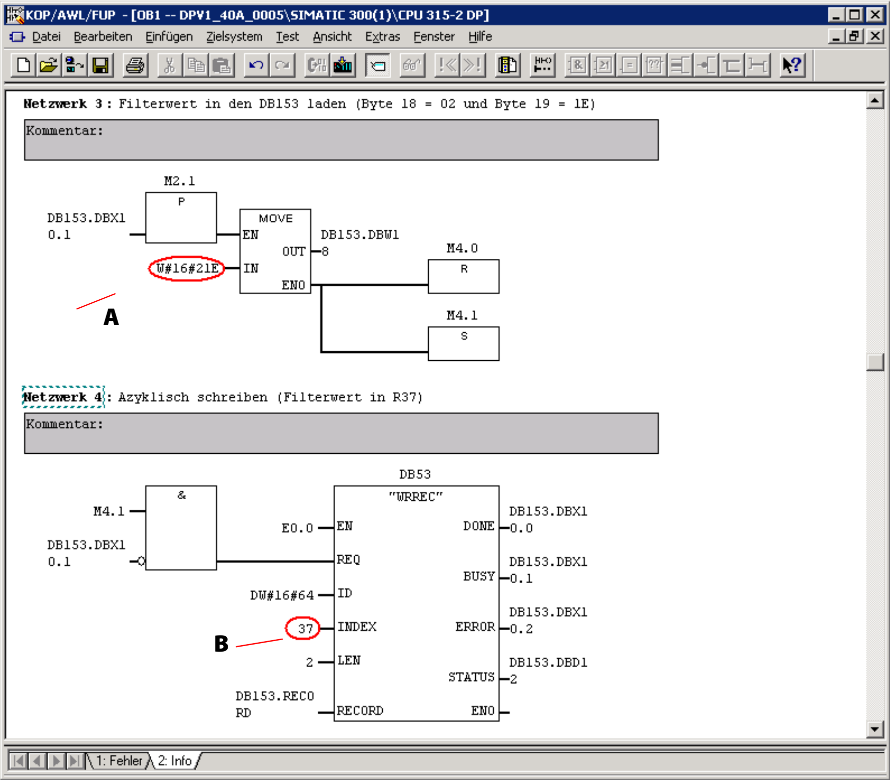
Setting the cycle time

In networks 3 and 4, the cycle time of 20 ms to be set (value = 0x021E) will be written to register 37 by means of SFB52 and DFB53.

Figure 9-17: Cycle time in register 37,

A cycle time 20 ms

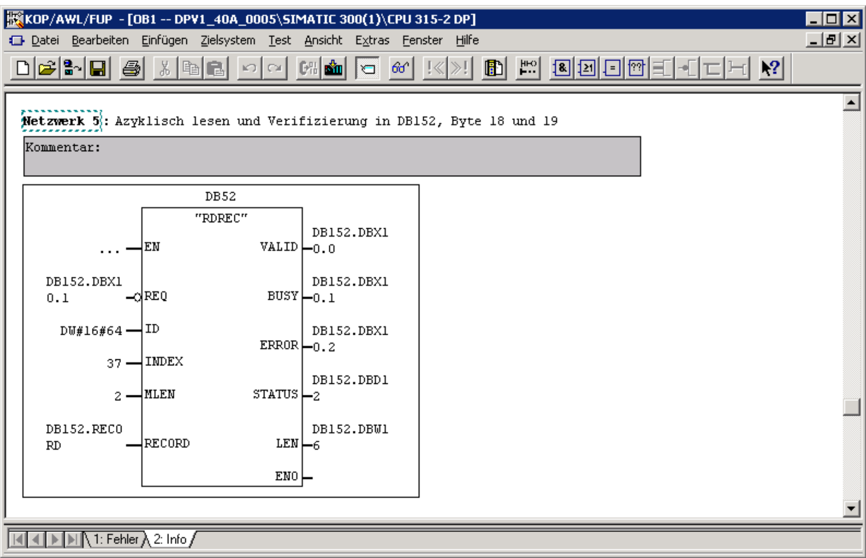
B register-number



Checking the parameter changes

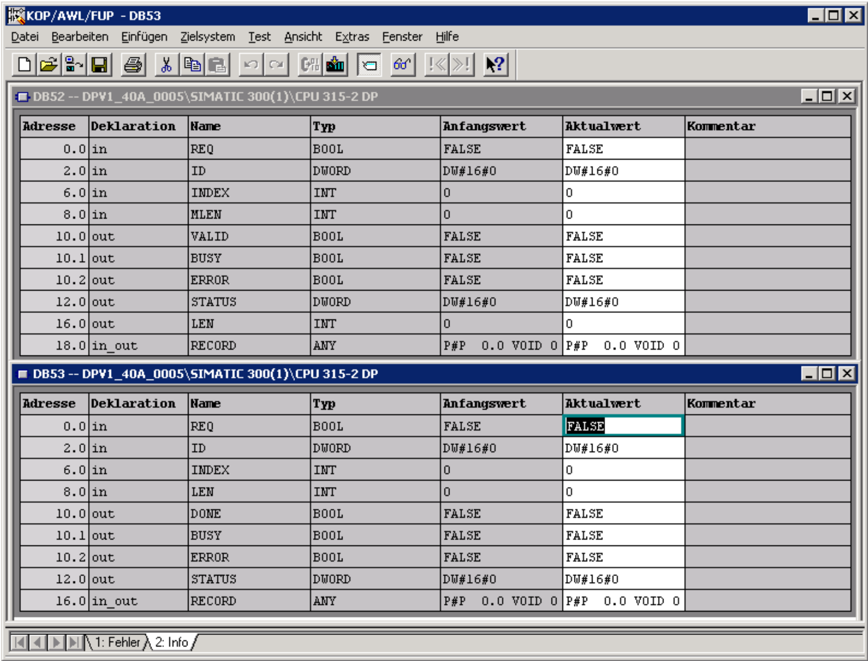
Checking the parameter changes is done by means of SFB52 "RDREC". To do so, the new value in registers 37 is read out using the SFB in network 5.

Figure 9-18: Checking the parameter changes



9.1.10 Structure of the data blocks

Figure 9-19:
DB52 and DB53



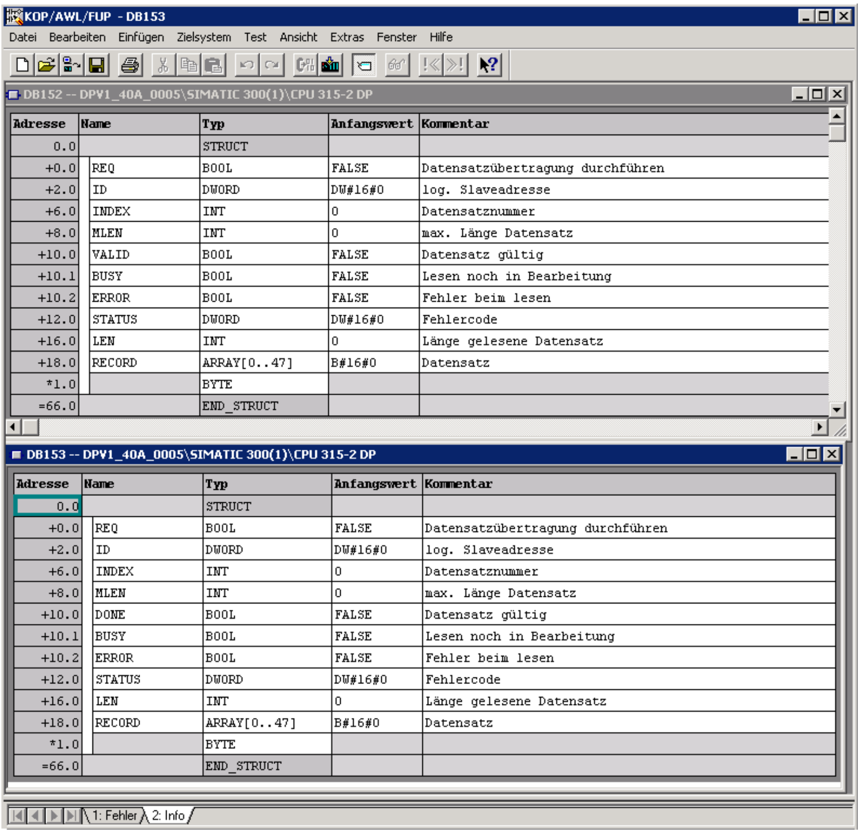
DB52:

Adresse	Deklaration	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	in	REQ	BOOL	FALSE	FALSE	
2.0	in	ID	DWORD	DW#16#0	DW#16#0	
6.0	in	INDEX	INT	0	0	
8.0	in	MLEN	INT	0	0	
10.0	out	VALID	BOOL	FALSE	FALSE	
10.1	out	BUSY	BOOL	FALSE	FALSE	
10.2	out	ERROR	BOOL	FALSE	FALSE	
12.0	out	STATUS	DWORD	DW#16#0	DW#16#0	
16.0	out	LEN	INT	0	0	
18.0	in_out	RECORD	ANY	P#P 0.0 VOID 0	P#P 0.0 VOID 0	

DB53:

Adresse	Deklaration	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	in	REQ	BOOL	FALSE	FALSE	
2.0	in	ID	DWORD	DW#16#0	DW#16#0	
6.0	in	INDEX	INT	0	0	
8.0	in	LEN	INT	0	0	
10.0	out	DONE	BOOL	FALSE	FALSE	
10.1	out	BUSY	BOOL	FALSE	FALSE	
10.2	out	ERROR	BOOL	FALSE	FALSE	
12.0	out	STATUS	DWORD	DW#16#0	DW#16#0	
16.0	in_out	RECORD	ANY	P#P 0.0 VOID 0	P#P 0.0 VOID 0	

Figure 9-20:
DB152 and DB153



DB152:

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	REQ	BOOL	FALSE	Datensatzübertragung durchführen
+2.0	ID	DWORD	DW#16#0	log. Slaveadresse
+6.0	INDEX	INT	0	Datensatznummer
+8.0	MLEN	INT	0	max. Länge Datensatz
+10.0	VALID	BOOL	FALSE	Datensatz gültig
+10.1	BUSY	BOOL	FALSE	Lesen noch in Bearbeitung
+10.2	ERROR	BOOL	FALSE	Fehler beim lesen
+12.0	STATUS	DWORD	DW#16#0	Fehlercode
+16.0	LEN	INT	0	Länge gelesene Datensatz
+18.0	RECORD	ARRAY[0..47]	B#16#0	Datensatz
*1.0		BYTE		
=66.0		END_STRUCT		

DB153:

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	REQ	BOOL	FALSE	Datensatzübertragung durchführen
+2.0	ID	DWORD	DW#16#0	log. Slaveadresse
+6.0	INDEX	INT	0	Datensatznummer
+8.0	MLEN	INT	0	max. Länge Datensatz
+10.0	DONE	BOOL	FALSE	Datensatz gültig
+10.1	BUSY	BOOL	FALSE	Lesen noch in Bearbeitung
+10.2	ERROR	BOOL	FALSE	Fehler beim lesen
+12.0	STATUS	DWORD	DW#16#0	Fehlercode
+16.0	LEN	INT	0	Länge gelesene Datensatz
+18.0	RECORD	ARRAY[0..47]	B#16#0	Datensatz
*1.0		BYTE		
=66.0		END_STRUCT		

9.2 Application example counter module

In the following example a *piconet*®-counter module is to be parameterized as follows:

- count upwards
- Setting an output after 25 counted pulses
- Resetting the output after 50 counted pulses
- Resetting the counter to 0 after 100 counted pulses

9.2.1 Parameterization of a counter as stand-alone module via GSx-file

The parameterization of the counter being a stand-alone module can be done using the GSx-parameters via the textual parameterization in the hardware configurator.

Attention
For modules, which provide the textual parameterization via GSx-file, a parameterization via register-communication should be avoided. See page 7-23 „Parameterization of the modules".

Instructions:

1 Open the dialog box "properties DP-slave" by double clicking the module. In the register "Parameter Assignment" all device-specific parameters can be found.

2 The following parameters are set:

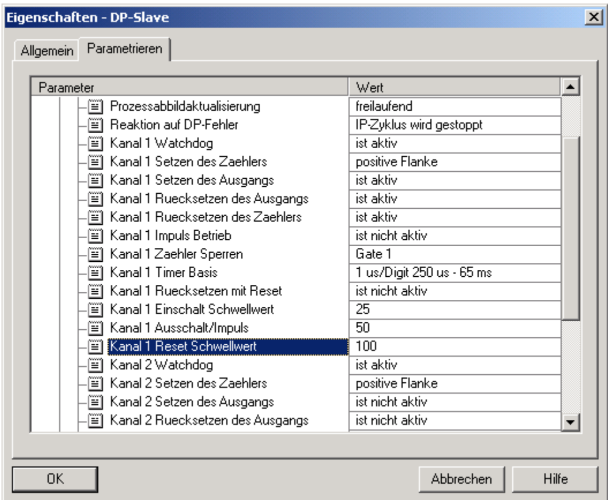


Figure 9-21: Parameterization in the Hardware Configurator

3 Parameterize the module accordingly and load the changed settings to the PLC using the "PLC → Download..." command.

9.2.2 Parameterization of a counter as extension module via register communication

The parameterization of the extension module can only be done via register communication or using the software I/O ASSISTANT All settings are done directly in the predefined registers.

The access to the registers of the *piconet*®-modules is done by activating the register communication, a write or read command and by entering the register number into the control byte of the respective module.

Writing the password into register 31

The password 0x1235 will be written into register 31 of the counter module. The command for writing the register and the enabling of the register communication by entering the password is given via the module's control byte.

Table 9-10: Writing register 31 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	1	1	1	1	1

Bit 7	= 1	→	register communication activated
Bit 6	= 1	→	register is written
Bit 5 to Bit 0	= 31 (dec.)	→	register number

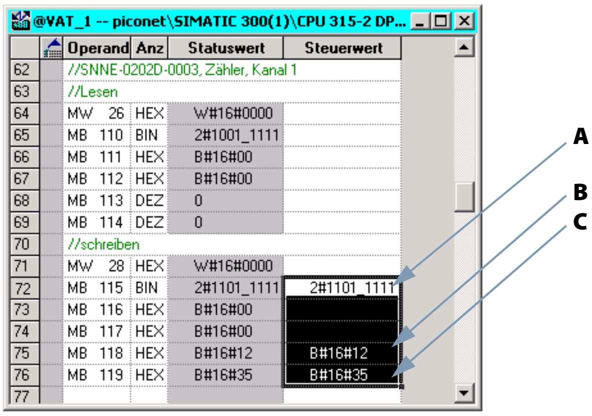
The data bytes of the output data image contain the values which have to be transmitted to the respective registers.

The password 0x1235 is written to the module's data byte 0 (in the example: marker byte 119) and data byte 1 (in the example: marker byte 118), whereas data byte 0 is the low- and data byte 1 the high-byte.

In this case the 0x12 has thus to be written into memory byte 118 (data byte 1) and the 0x35 into memory byte 119 (data byte 0).

Figure 9-22: Password in register 31,

A *register 31,*
B *data byte 1*
C *data byte 0*



Note

The mapping tables in chapter 5 of the manual show the assignment of in- and output data to the data bytes.

Industrial Automation

9.2.3 Activating the switch-on, -off and reset-threshold values

Activating the counter functions for setting and resetting the output as well as for resetting the counter is done in the module's feature register (register 32) - see for the counter's register assignment : Chapter 12 "Technology modules" in manual „*piconet*® I/O-modules´", TURCK-documentation number: D300778 (English) –.

Writing the feature register (register 32)

The settings in register 32 are transmitted to the module via the control byte as follows.

Table 9-11: Writing register 32 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	0	0	0

Bit 7	= 1	→	register communication activated
Bit 6	= 1	→	register is written
Bit 5 to Bit 0	= 32 (dec.)	→	register number

The assignment of the Feature register defines the following settings for this example:

- Low -byte → data byte 0 (in the example: memory byte 119)
  Value: 0x78:

Table 9-12: Register 32, Low-byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	1	1	1	0	0	0

Bit 7	= 0	→	pulse mode inactive.
Bit 6	= 1	→	function for the counter reset active
Bit 5	= 1	→	function for resetting the output active
Bit 4	= 1	→	function for resetting the output active
Bit 3	= 1	→	Setting the counter is done with a positive edge at bit CNT_SET in the control-byte
Bit 2	= 0	→	watchdog active
Bit 1, Bit 0	= 0	→	reserved

- High-byte → data byte 1 (in the example: memory byte 118)
  Value: 0x01

Table 9-13: Register 32, High-byte

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0	0	0	0	0	0	0	1

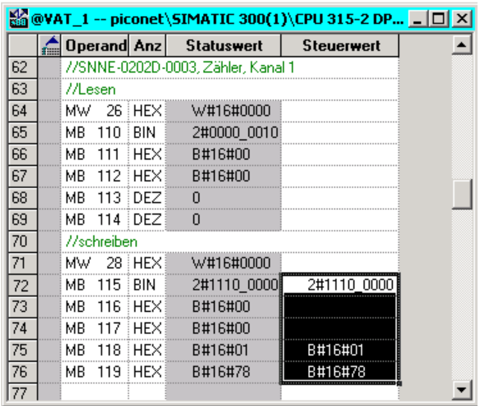
Bit 15
to Bit 11= 00 → reserved

Bit 10 = 0 → resetting the output is done with a counter reset

Bit 9 = 0 → timer base (pulse length register 41)
: 1 µs/Digit (250 µs - 65 ms)

Bit 8 = 1 → counter is disabled if input Gate
= high (1)

Figure 9-23: Writing register 32



Note
The settings done in register 32 will only be valid after a voltage reset of the module.

9.2.4 Activating the switch-on, -off and reset-threshold values

Setting the threshold values is done in registers 35 to 40. The threshold values will be permanently stored to the module's EEPROM.

Note
Important for the threshold values: switch-on value < switch-off value < reset value

Table 9-14: threshold value register register

register	Designation	Memory
R0	Switch-on threshold value, low-byte	RAM
R1	Switch-on threshold value, high-byte	RAM
R2	Switch-off threshold value, low-byte	RAM
R3	Switch-off threshold value, high-byte	RAM
R4	Switch-off threshold value, low-byte	RAM
R5	Reset threshold value, high-byte	RAM
....		
R35	Switch-on threshold value, low-byte	EEPROM
R36	Switch-on threshold value, high-byte	EEPROM
R37	Switch-off threshold value, low-byte	EEPROM
R38	Switch-off threshold value, high-byte	EEPROM
R39	Reset threshold value, low-byte	EEPROM
R40	Reset threshold value, high-byte	EEPROM
....		

Note
The changes in registers 35 to 40 will only be valid and copied to the module's RAM and thus to register 0 to 5 after a power reset.

Attention
Setting the threshold values should not be done directly in registers 0 to 5. In case of a power reset, the actual changes in the RAM could be overwritten with invalid EEPROM-values from registers 35 to 40.

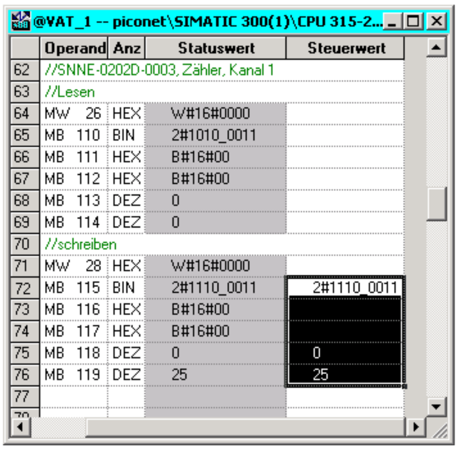
■ **Switch-on threshold value** 25 pulses

Table 9-15: Writing register 35 (control byte)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	1	1	1	0	0	0	1	1

→ data byte 0 (memory byte 119) = 25 dec. (0x19 hex.)

Table 9-16: Data byte 0 (memory byte 119)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	0	0	0	1	1	0	0	1

Figure 9-24: Switch-on threshold

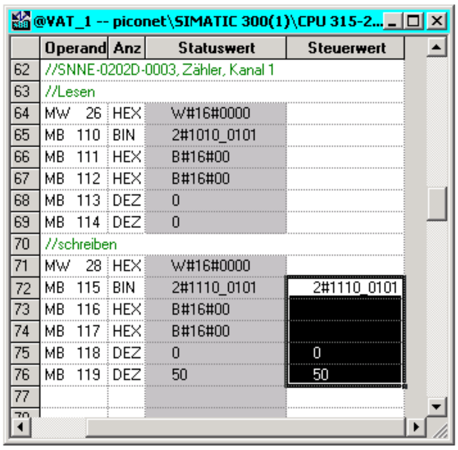


■ **Switch-off threshold value** (50 pulses)

Table 9-17: Writing register 37 (control byte)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	1	1	1	0	0	1	0	1

→ data byte 0 (memory byte 119) = 50 dec. (0x32 hex.)

Table 9-18: Data byte 0 (memory byte 119)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	0	0	1	1	0	0	1	0



	Operand	Anz	Statuswert	Steuerwert
62	//SNNE-0202D-0003, Zähler, Kanal 1			
63	//Lesen			
64	MW 26	HEX	W#16#0000	
65	MB 110	BIN	2#1010_0101	
66	MB 111	HEX	B#16#00	
67	MB 112	HEX	B#16#00	
68	MB 113	DEZ	0	
69	MB 114	DEZ	0	
70	//schreiben			
71	MW 28	HEX	W#16#0000	
72	MB 115	BIN	2#1110_0101	2#1110_0101
73	MB 116	HEX	B#16#00	
74	MB 117	HEX	B#16#00	
75	MB 118	DEZ	0	0
76	MB 119	DEZ	50	50
77				

Figure 9-25: Switch-off threshold

- **Reset threshold value** 100 pulses

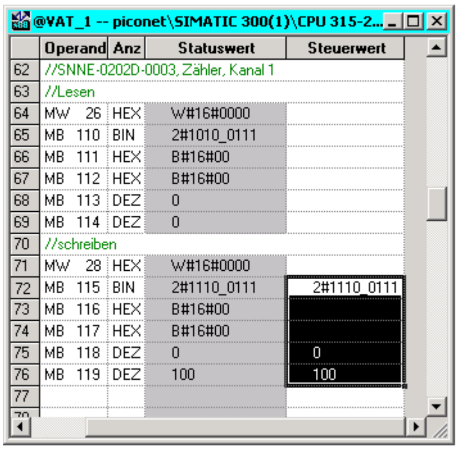
Table 9-19: Writing register 39 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	1	1	1

→ data byte 0 (memory byte 119) = 100 dec. (0x64 hex.)

Table 9-20: Data byte 0 (memory byte 119)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	1	0	0	1	0	0



	Operand	Anz	Statuswert	Steuerwert
62	//SNNE-0202D-0003, Zähler, Kanal 1			
63	//Lesen			
64	MW 26	HEX	W#16#0000	
65	MB 110	BIN	2#1010_0111	
66	MB 111	HEX	B#16#00	
67	MB 112	HEX	B#16#00	
68	MB 113	DEZ	0	
69	MB 114	DEZ	0	
70	//schreiben			
71	MW 28	HEX	W#16#0000	
72	MB 115	BIN	2#1110_0111	2#1110_0111
73	MB 116	HEX	B#16#00	
74	MB 117	HEX	B#16#00	
75	MB 118	DEZ	0	0
76	MB 119	DEZ	100	100
77				

Figure 9-26: Reset threshold value

Attention

Please execute a power reset in order to overtake the settings from register 35 to 40 to register 0 to 5.

9.2.5 Enabling the internal counter functions

1. Before enabling the counter, the internal counter functions (which means the parameter settings) have to be enabled using the enable-bit „EN_A“, bit 0 in the module's control byte.
2. The control byte is written with 2#0000_0001 using the variable table.

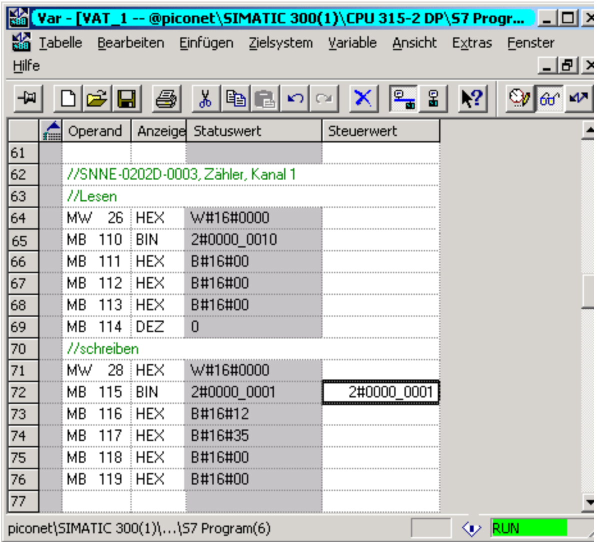


Figure 9-27: Enabling via bit Bit 0 of the control bytes

9.2.6 Monitoring the count operation via the variable table

The counter counts upwards. The counted pulses are monitored in data byte 0, in this case MB124, of the counter.

Reaching the switch-on threshold value´

1. As defined, the counter counts to 25 and sets the output.
2. The LED Q1 at connector "D" of the counter module lights up green.
3. Bit 2 of the status byte (in the example MB 110) is set.

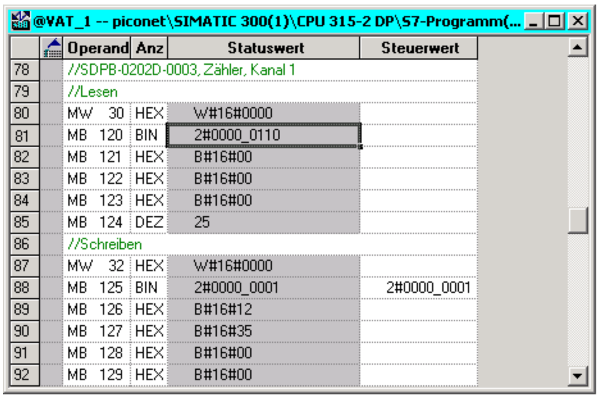
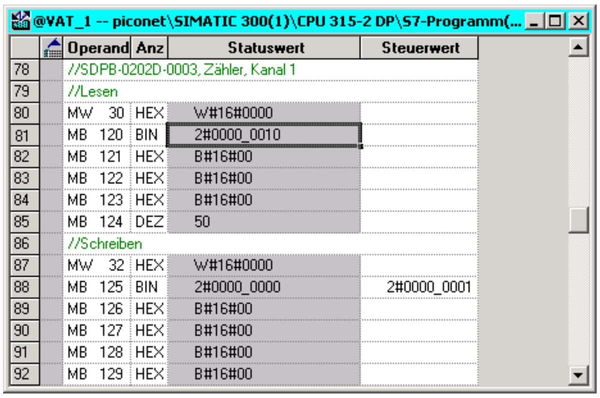


Figure 9-28: Setting the output

Reaching the switch-off/ pulse value

4. The counter continues counting as defined. Having reached the 50 counted pulses, the output is reset.
5. The LED Q1 at connector "D" of the counter module is switched-off.

6 In the status byte (MB110), bit2 is reset.



@VAT_1 -- piconet\SIMATIC 300(1)\CPU 315-2 DP\S7-Programm(...

	Operand	Anz	Statuswert	Steuerwert
78	//SDPB-0202D-0003, Zähler, Kanal 1			
79	//Lesen			
80	MW 30	HEX	W#16#0000	
81	MB 120	BIN	2#0000_0010	
82	MB 121	HEX	B#16#00	
83	MB 122	HEX	B#16#00	
84	MB 123	HEX	B#16#00	
85	MB 124	DEZ	50	
86	//Schreiben			
87	MW 32	HEX	W#16#0000	
88	MB 125	BIN	2#0000_0000	2#0000_0001
89	MB 126	HEX	B#16#00	
90	MB 127	HEX	B#16#00	
91	MB 128	HEX	B#16#00	
92	MB 129	HEX	B#16#00	

Figure 9-29: Resetting the output

Reaching the reset threshold value

7 After counting the 100 pulses , the counter value in data byte 0 (here: MB 124) is reset to "0".

8 The counter restarts counting upwards starting at "0".

9.3 Application example, SSI module

The parameterization of the extension module can only be done via register communication or using the software I/O ASSISTANT.

The multi turn-encoder cannot be parameterized through the *piconet*®-module. The configuration can of Kuebler encoders can for example be done using the special software tool for parameterizing multi turn encoders, Ezturn®.

Note

When using the *piconet*® SSI-module Sxxx-10S-x005, please check that the communication parameter (output format, single/ or multi turn, baud rate , data length, etc.) are the same for both devices, module and SSI-encoder.

This effects all parameter settings of the feature register (register 32), of the baud rate register (register 33) and of the data length register (register 34).

Note

Detailed information about the module's register assignments can be found in chapter 12 of the *piconet*® I/O manual "*piconet*® I/O modules", TURCK-documentation number: D300778 (English).

9.4 Application example incremental-encoder

In the following example a *piconet*®-incremental-encoder module is to be parameterized as follows:

- Setting the counter value to "0"0
- Storing the count value with activated latch-input
- Disabling the counter with a low-signal at the gate-input
- Switching from the encoder to the counter mode

9.4.1 Parameterization of an incremental encoder via GSx-file

Parameterization of the incremental encoder module via GSX-file is not possible at the moment.

The module can either be parameterized via the control byte and/or via register communication or via the software I/O-ASSISTANT 3.0 .

9.4.2 Parameterization of an incremental encoder via the control byte

In many cases, the parameterization of the incremental encoder can be done directly via the control byte. A parameterization via register communication is then not necessary.

Certain parameterizations have nevertheless to be done using the register communication. All settings are done directly in the predefined registers.

Setting the counter value

With a rising edge at bit "CNT_SET", bit 2 in the control word", the counter will be set to the value defined via the process data in Reg0 and Reg1 (see "Mapping" in the bus specific *piconet*®-manual).

Setting the bit "CNT-SET"

Setting the bit is done simply via the control byte, without register access.

Table 9-21: Control byte

Bit	7	6	5	4	3	2	1	0
Name	RegAccess	-	-	-	-	CNT_SET	EN_LAT_EXT/ RD_PERIOD	EN_LATC

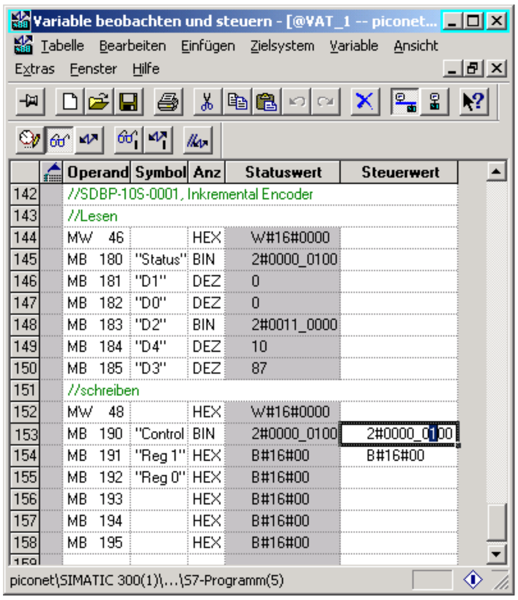
Table 9-22: Setting "CNT_SET"

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	1	0	0

The value defined in Reg0 (memory byte 192) and Reg1 (memory byte 191), in this example "0", is overtaken.

Note
Please ensure, that bit "CNT_SET" has to be reset before activating it again.

Figure 9-30: Setting the counter



Depending on the sense of rotation of the connected encoder, the module counts up- or downwards starting at "0".

Data byte D2

The status of the input channels A, B and C as well of inputs "Gate" and "Latch" are monitored in data byte 2.

Table 9-23: Data byte D2

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	-	-	INPUT_A	INPUT_B	INPUT_C	INPUT_ERR	LATCH	Gate

Table 9-24: Description

Name	**Description**
INPUT_A	Status of input channel A
INPUT_B	Status of input channel B
INPUT_C	Status of input channel C
INPUT_ERR	Status of the alert-channel
LATCH	Status of the LATCH input at the M12 connector
GATE	Status of the GATE input at the M12 connector

Storing the counter value

Data byte D3 and D4 store the actually present count value when activating the Latch-input.

For activating the external Latch-input, bit 1 "EN_LAT_EXT" of the control byte is set.

After EN_LAT_EXT is true, the first latch-signal at the latch-input, stores the counter value- The following pulses do not influence the latch register if this bis is set.

Setting the bits is done is done simply via the control byte in process data exchange, without register access.

Table 9-25: Control byte

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RegAccess	-	-	-	-	CNT_SET	EN_LAT_EXT/ RD_PERIOD	EN_LATC

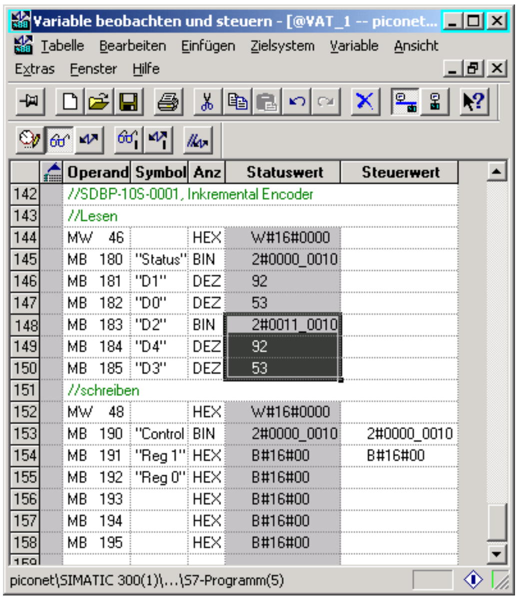
Table 9-26: Setting "CNT_SET"

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	0	0	0	0	0	1	0

If now, during the count operation, a signal change from "0" to "1" occurs at the latch-input, the count value actually read at this point of time is written to data byte D3 and D4.

The activation of the latch-input is confirmed by a short setting of bit 1 "LATCH" in data byte D2 (see page 9-28).

Figure 9-31: Storing the counter value



9.4.3 Parameterization of an incremental encoder via register communication

For the module parameterization via register communication, at first, the password for the register access has to be written to register 31.

Writing the password into register 31

The password 0x1235 will be written into register 31 of the incremental encoder module. The command for writing the register and the enabling of the register communication by entering the password is given via the module's control byte.

Table 9-27: Writing register 31 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	0	1	1	1	1	1

Bit 7	= 1	→	register communication activated
Bit 6	= 1	→	register is written
Bit 5 to Bit 0	= 31 (dec.)	→	register number

The data bytes of the output data image contain the values which have to be transmitted to the respective registers.

The password 0x1235 is written to the module's data byte 0 (in the example: marker byte 192) and data byte 1 (in the example: marker byte 191), whereas Reg0 is the low- and Reg1 the high-byte.

In this case the 0x12 has thus to be written into memory byte 191 (Reg1) and the 0x35 into memory byte 192 (Reg0).

Note

The mapping tables in chapter 5 Data mapping of the piconet® modules show the assignment of in- and output data to the data bytes.

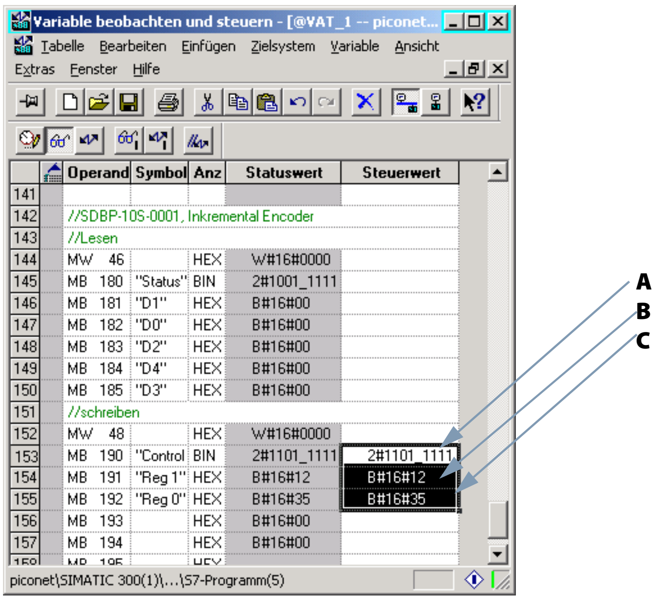


Figure 9-32: Password in register 31,

***A** register 31,*
***B** Reg 1*
***C** Reg 0*

Disabling the counter

The counter of the incremental encoder can, depending on the parameterization, be disabled either with a high or a low signal at the gate-input.

In this case, the disabling of the counter is done with a low signal at the gate-input.

The parameterization is done via the module's Feature Register (R32).

The access to the module registers is done by activating the register communication, a write or read command and by entering the register number into the control byte of the respective module.

Writing the feature register (register 32)

The settings in register 32 are transmitted to the module via the control byte as follows.

Table 9-28: Writing register 32 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	0	0	0

Bit 7 = 1 → register communication activated

Bit 6 = 1 → register is written

Bit 5 to Bit 0 = 32 (dec.) → register number

The assignment of the Feature register of the incremental encoder defines the following settings for this example:

- Low -byte → data byte 0 (in the example: memory byte 119)
  Value: 0x02

Table 9-29: Register 32, Low-byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	**1**	0

Bit 7 = 0 → reserved

Bit 6, Bit 5 = 0 → reserved

Bit 4 = 0 → external latch function active

Bit 3, Bit 2 = 0 → status-input (active-low) is mirrored to the status byte, bit 5.

Bit 1 = 1 → the counter is disabled with a **low signal** at the gate-input

Bit 0 = 0 → reserved

- High-byte → Reg1 (in the example: memory byte 191)
  Value: 0x80

Table 9-30: Register 32, High-byte

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
1	0	0	0	0	0	0	0

Bit 15 = 0 → encoder interface

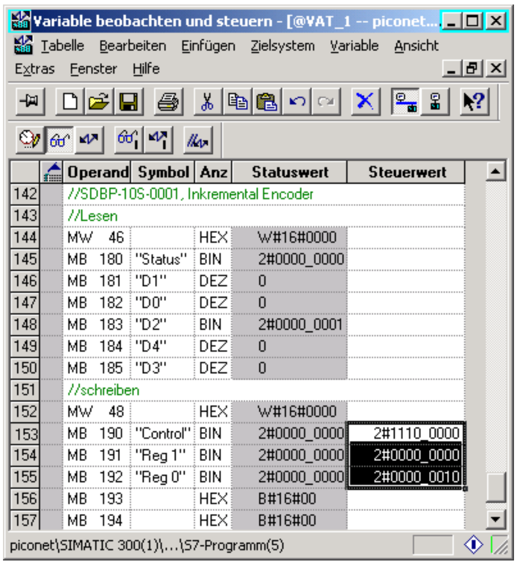
Bit 14

to Bit 12= 00 → reserved

Bit 11, Bit 10 = 0 → quadruple evaluation of the encoder signals A,B,C

Bit 9, Bit 8 = 1 → reserved

Figure 9-33: Writing register 32



Note

The settings done in R32 become only valid after a power reset at the module. The module now changes
to process data exchange.

The counter will now be disabled as long as a high signal at the gate-input occurs.

With a new signal change at the input "High" → "Low", the counter will again be disabled.

Switching: Encoder interface mode to counter mode

Switching from the encoder interface mode to the counter mode is also done using the module's feature register R32.

Writing the feature register (register 32)

The settings in register 32 are transmitted to the module via the control byte as follows.

Table 9-31: Writing register 32 (control byte)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	1	1	0	0	0	0	0

Bit 7 = 1 → register communication activated

Bit 6 = 1 → register is written

Bit 5
to Bit 0 = 32 (dec.) → register number

The assignment of the Feature register of the incremental encoder defines the following settings for this example:

- Low -byte → data byte 0 (in the example: memory byte 119)
  Value: 0x00

Table 9-32: Register 32, Low-byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0

Bit 7 = 0 → reserved

Bit 6, Bit 5 = 0 → reserved

Bit 4 = 0 → external latch function active

Bit 3, Bit 2 = 0 → status-input (active-low) is mirrored to the status byte, bit 5.

Bit 1 = 0 → the counter is disabled with a high-signal at the gate-input

Bit 0 = 0 → reserved

- High-byte → Reg1 (in the example: memory byte 191)
  Value: 0x80

Table 9-33: Register 32, High-byte

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
1	0	0	0	0	0	0	0

Bit 15 = 1 → counter-mode

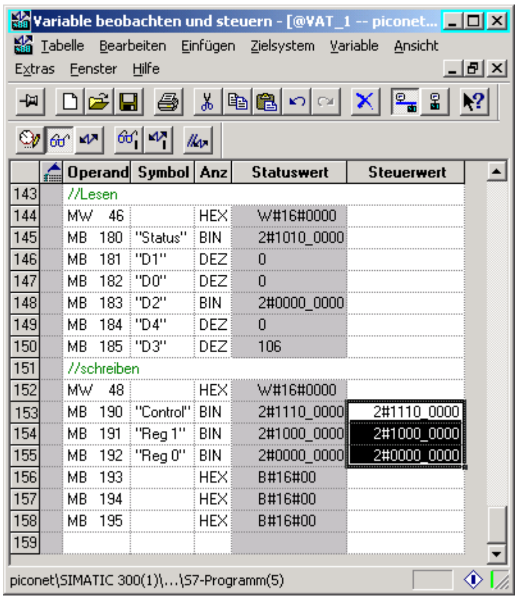
Bit 14

to Bit 12= 00 → reserved

Bit 11, Bit 10 = 0 → quadruple evaluation of the encoder signals A,B,C

Bit 9, Bit 8 = 1 → reserved

Figure 9-34:
Writing register 32



Note

After a power reset of the module, the settings done in R32 become valid, the module changes to the process data exchange and the counter continues counting.

10 IP-Link diagnosis via acyclic services (S7 and VT250)

10.1 General

This chapter describes, in order to localize an error in a IP-Link line, the reading of error counters from register tables (see General structure of the data in the register tables (page 10-3) of the coupling module and the IP-Link extension modules by means of acyclic PROFIBUS-DPV1-services.

If the IP-Link-communication is disturbed, but not totally interrupted, the diagnostics messages of the IP-Link may be read out of the register tables from the coupling or the extension modules.

The access to these register tables is done via acyclic services by means of respective function blocks (seeFunction blocks for acyclic services (page 10-2)).

10.1.1 Important information

Date of manufacturing

Only modules manufactured in 2011 or later off the possibility of reading the IP-Link error counter.

GSD files

Please use only the newest GSD-files (GSD revision ≥ V 3).

10.1.2 Function blocks for acyclic services

These function blocks are used for acyclic data exchange in the PLCs and are provided by the PLC manufacturers.

Table 10-1: Function blocks for acyclic services

PLC	function block/ structure	
S7 with Step 7	SFB52 „RDREC“	Acyclic reading of data from defined registers.
	SFB53 „RWRREC“	Acyclic writing of defined registers.
VT250 with CoDeSys V3	DPV1_C1_M_Read (from the DP-Master library IIoDrvDPV1C1.library))	Acyclic reading of data from defined registers.
	DPV1_C1_M_Write (from the DP-Master library IIoDrvDPV1C1.library)	Acyclic writing of defined registers.

Note

The usage of these function blocks is described later on in this chapter by means of examples with S7 and VT250 (see also Reading IP-Link errors using the S7 with Step7 (example) (page 10-6) and Reading IP-Link errors using VT250 with CoDeSys V3 (page 10-11).

10.1.3 General structure of the data in the register tables

Table 10-2: Register with IP-Link diagnostic messages

Registertable Index-no. (DPV1)		Diagnostic content
dec.	hex	
90	5A	**Error counter for general IP-Link errors** In this register, offset 005, occurred IP-Link errors are counted.
50 - 60	32 - 3C	**Error counter in the extension modules - error localization** Reading the error position using the error counter of the extension modules. This is only possible if the IP-Link communication is still running.

Register table 90 (general IP-Link errors)

Register table 90 contains, with an offset of 5 registers, an general IP-Link error counter.

Sporadically occurring errors do not cause problems in the communication. This error counter will only be reset by a Power ON/OFF.

Table 10-3: Structure of register table 90

Byte -no.	word- -no.	Word-no. at Siemens	register -no.	Content
0	0	0	0	Offset
1	1	0	0	
2	2	2	1	
3	3	2	1	
4	4	4	2	
5	5	4	2	
6	6	6	3	
7	7	6	3	
8	8	8	4	
9	9	8	4	
10	10	10	5	IP-Link error counter
11	11	10	5	

Register tables 50 - 60 (error counter of the extension modules)

If the coupling module recognizes an error, it tries to localize this error by reading the extension modules' error counter.

this is not possible, if the IP-Link is interrupted or if the communication is heavily disturbed. In this case, the IP-Link error position is either shown optically via LED blinking codes or read via PROFIBUS-DP diagnostics. The error position is localized by counting backwards starting at the coupling module (see Error diagnosis via LEDs (page 6-2))..

Note

If, in register table 50, the maximum number of error streams is reached (255/0×FF in low and high byte), the error counter stops at these values.
It can only be reset to zero by means of a power reset!

But, if the communication is still running, then the error counter for each extension module can be read from register tables 50 to 60. Register table 50 contains the error counters for modules 1 to 22 of the IP-Link-line, register table 51 contains the error counters for modules 23 to 46 etc..

Table 10-4: Structure of register tables 50 - 60

Table	Byte -no.	Word- -no.	Word-no. at Siemens	Register -no.	Content (Error counter of)	
50	0	0	0	0	Offset	
	1	0	0	0		
	2	1	2	1		
	3	1	2	1		
	4	2	4	2	extension module 1	Low-byte
	5	2	4	2		High-byte
	6	3	6	3	extension module 2	Low-byte
	7	3	6	3		High-byte
	8	4	8		extension module 3	Low-byte
	9	4	8			High-byte
	...	...	...	...	...	
	46	23	46	23	extension module 22	Low-byte
	47	23	46	23		High-byte
51	0	0	0	24	extension module 23	Low-byte
	1	0	0	24		High-byte
	2	1	2	25	extension module 24	Low-byte
	3	1	2	25		High-byte
	...	...	...	...	...	
	46	23	46	47	extension module 46	Low-byte
	47	23	46	47		High-byte

The position of the error counter in the register table corresponds to the position of the extension module in the IP-Link-line (offset = 4 bytes). An error at the second extension module would thus be monitored in bytes 6 and 7, an error in extension module 3 in bytes 8 and 9. (Step7 - reading the IP-Link error counter from register table 50 (page 10-10) or CoDeSys V3 - reading the IP-Link error counter from register table 50 (page 10-16)).

Note

If register table 90 shows an error (error counter counts upwards), but register table 50 shows no error, then, the error cause has to be found between the last extension module and the coupling module.

Note

Register table 50 is not actualized during an IP-Link interruption, because reading those values directly from the extension modules is not possible.

10.2 Reading IP-Link errors using the S7 with Step7 (example)

10.2.1 Example project

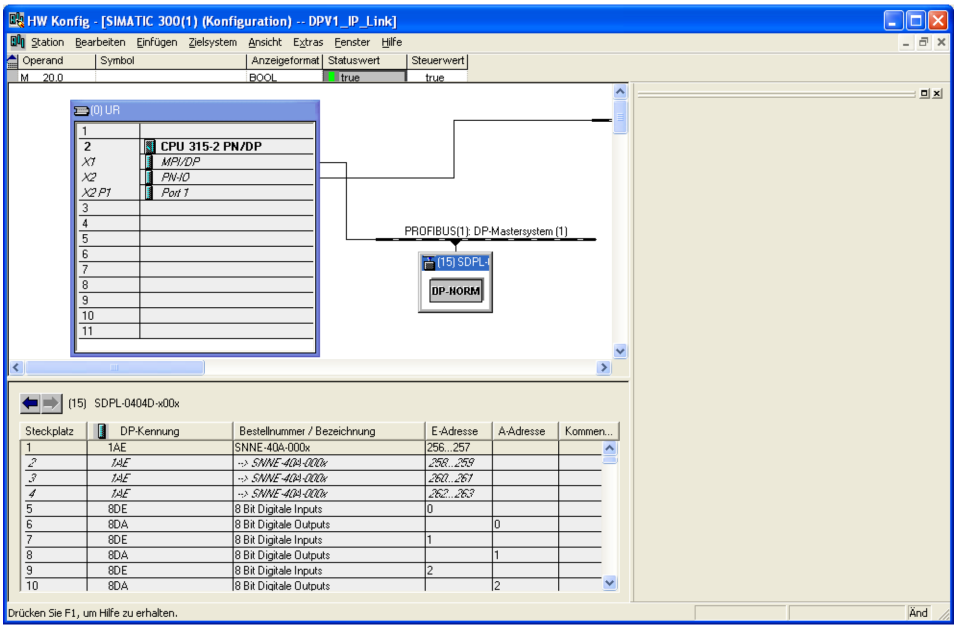
Example network

- S7 CPU315-2 PN/DP (6ES7 315-2EH13-0AB0, (V 2.6)
- *piconet*®-station (DP-address 15)
  - coupling module SDPL-0404D-1003, 4 digital inputs and 4 digital outputs
  - IP-Link-extension module 1: SNNE-0808D-0001, 8 digital inputs and 8 digital outputs
  - IP-Link-extension module 2: SNNE-0800D-0007, 8 digital inputs
  - IP-Link-extension module 3: SNNE-40A-0004, 4 analog inputs (TC)
  - IP-Link-extension module 4: SNNE-0008D-0006, 8 digital outputs

10.2.2 Configuration of the station in Step7

When configuring the *piconet*®-station, please observe that analog channels always have to be configured before the digital channels (see figure)!

Figure 10-1: Configuration of the station in Step7

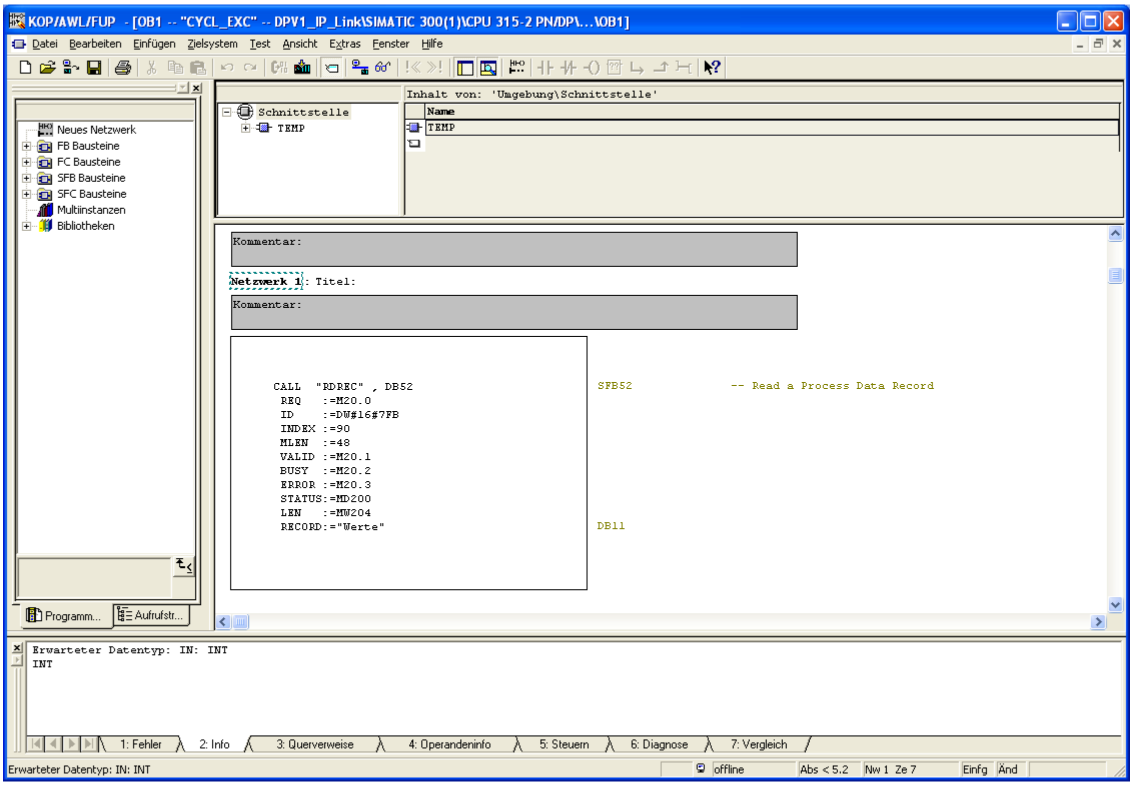


10.2.3 Reading the diagnosis via acyclic services

Using the function block SFB52

For acyclic reading of the IO-Link diagnosis, SFB52 is called in OB1 of the project (see also section Structure of the S7 program (page 9-3)).

Figure 10-2: SFB52



The variables of the function block are defined as follows:

Table 11: Input data SFB52

Parameter name	Meaning
REQ	REQ = 1, starts the data transmission.
ID	Logical address of the respective BL××-I/O-module, taken from the hardware configuration. When establishing a connection to the coupling module, the logical address is the "Diagnostic Address" assigned in the hardware configuration. Note: If the module to be addressed is an output module, bit 15 has to be set (e.g. for address 5: ID:=DW#16#8005). If the module concerned is a combination module, the lowest address has to be chosen.
INDEX	Number of the module's index to be read (no. of the register table to be read, see also Step7 - reading the IP-Link error counter from register table 90 (page 10-9)and Step7 - reading the IP-Link error counter from register table 50 (page 10-10)).
MLEN	Maximum length of the data to be read.

Table 12: Output data SFB52	Parameter name	Meaning
	VALID	New data set was received and valid.
	BUSY	BUSY = 1: The read process is not yet terminated.
	ERROR	ERROR = 1: Error occurred while reading.
	STATUS	Error code of the function block (see Siemens-online help for SFB54 "RALRAM")
	LEN	Length of the read data.
	RECORD	Target area for the read data record (here in this example DB11).

The following variables are important for reading out the IP-Link diagnosis:

- **REQ**:
  Start of the data set transmission
- **ID**:
  Definition of the diagnosis address of the *piconet*®-coupling module (here address 2043 = $7FB_{hex}$)
- **INDEX**:
  Number of the register table to be read (register table 90 or register table 50 - 60, see General structure of the data in the register tables (page 10-3)).
- **MLEN**:
  Length of data to be read.

Variable table

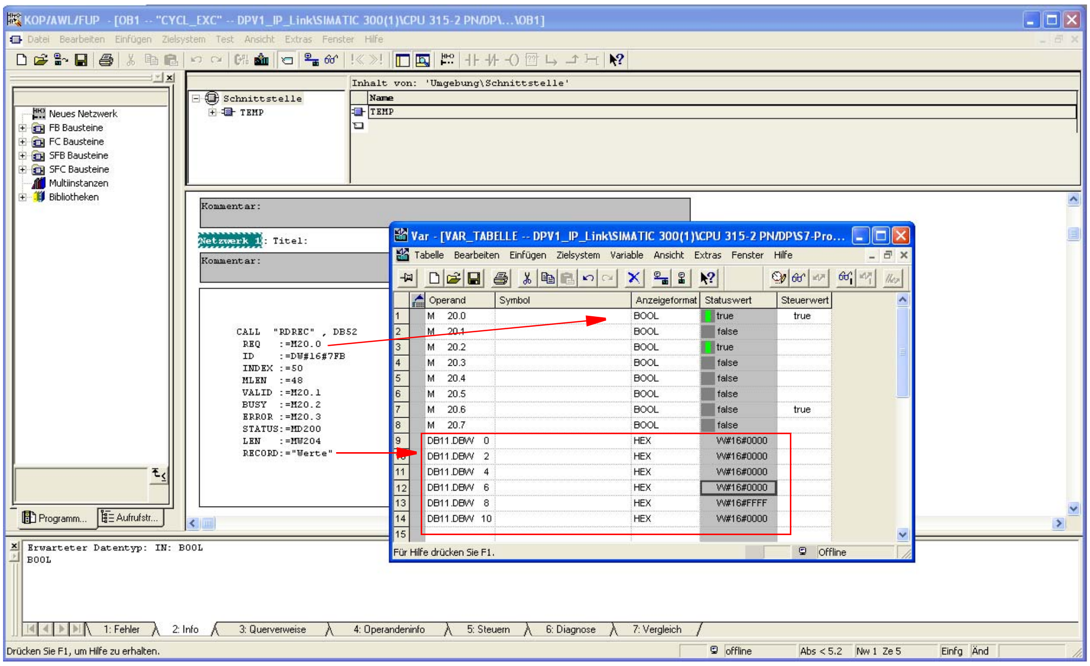


Figure 10-3: Example of variable table

Step7 - reading the IP-Link error counter from register table 90

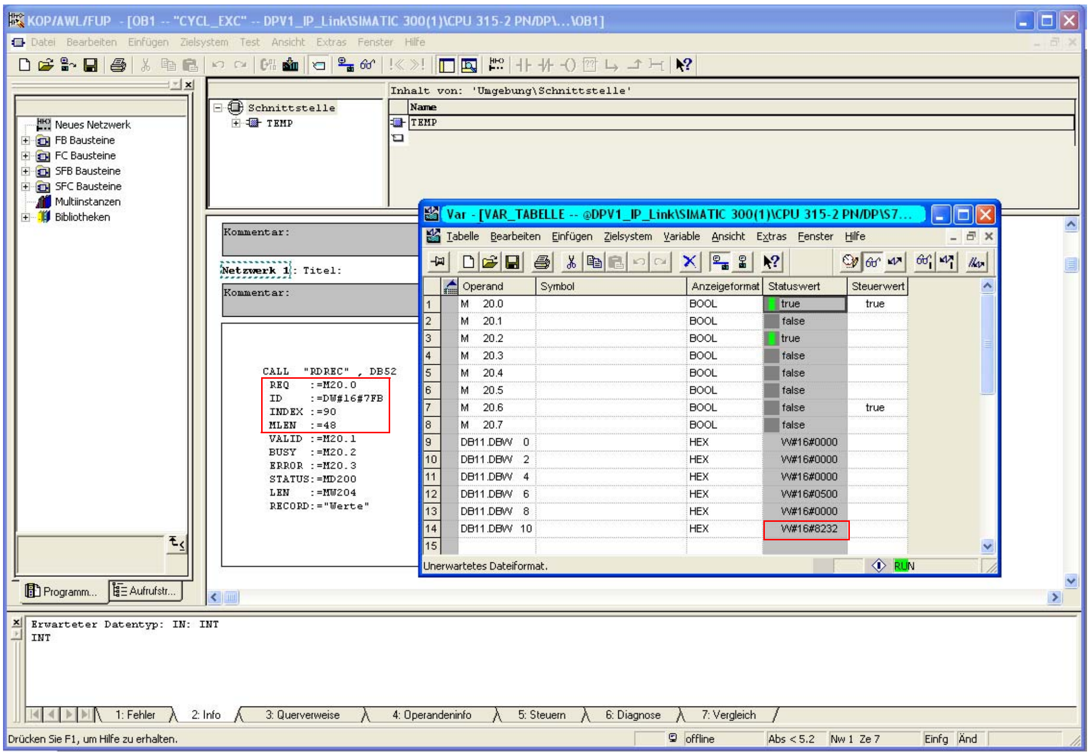
For reading register table 90 the following entries have to be done:

ID = 7FB (diagnosis address of the coupling module)

INDEX = 90

MLEN = 48

Table 10-1: Reading register table 90



The read are displayed in DB11.

The error counter in DB11.DBW10 shows that error streams arrive at the coupling module.and that the IP-Link communication is disturbed.

Step7 - reading the IP-Link error counter from register table 50

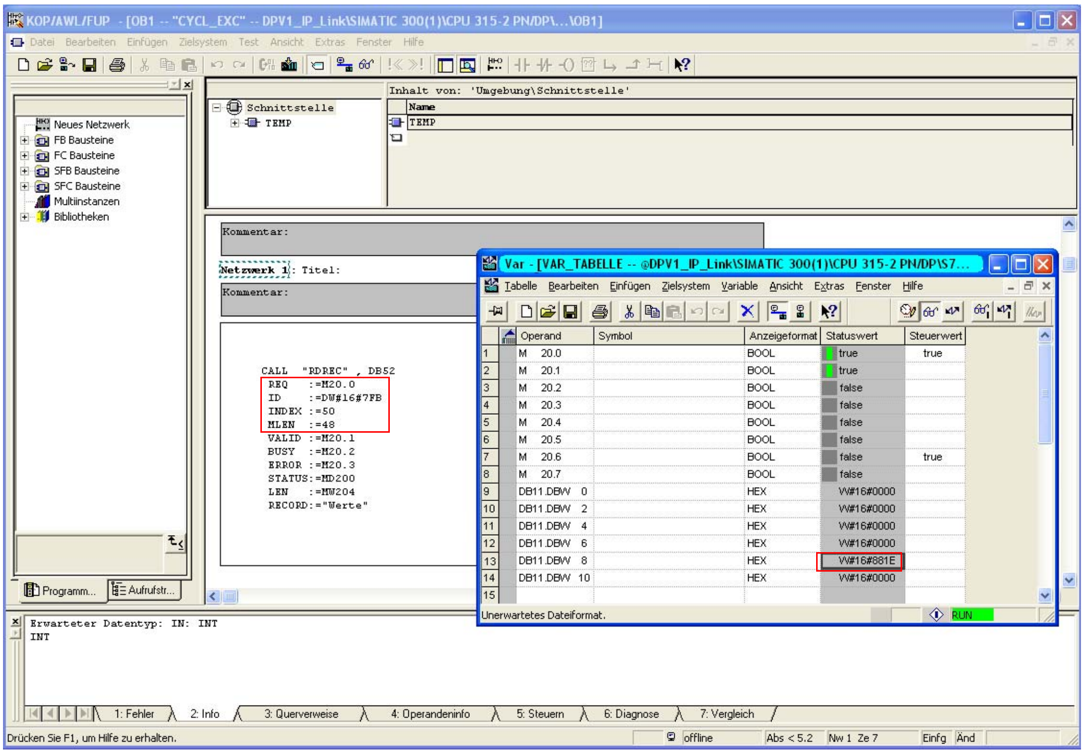
For reading register table 50 the following entries have to be done:

ID = 7FB (diagnosis address of the coupling module)

INDEX = 50

MLEN = 48

Table 10-2: Reading register table 50



The read are displayed in DB11.

The error counter in DB11.DBW8 shows that error streams arrive at the extension module.(see also Register tables 50 - 60 (error counter of the extension modules) (page 10-4)) and that the IP-Link communication is disturbed at this position.

10.3 Reading IP-Link errors using VT250 with CoDeSys V3

10.3.1 Example project

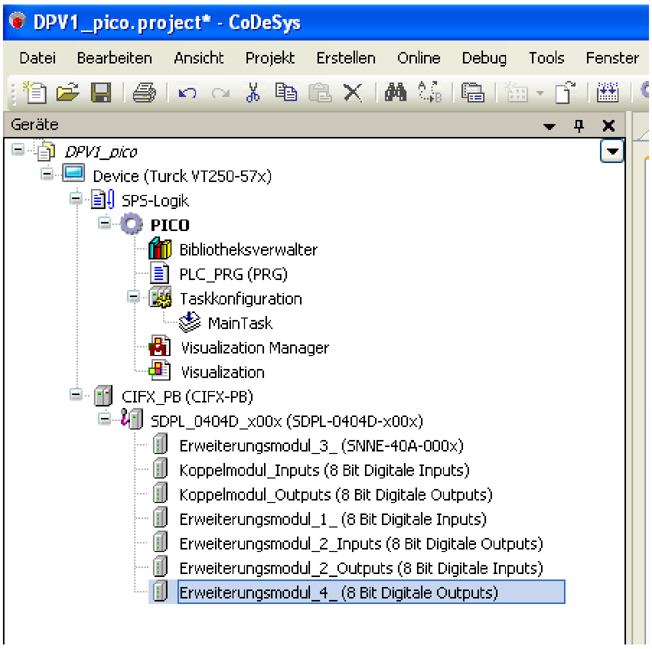
Example network

- VT250-57P (firmware VT250-57P-DPM-V1.0.5)
- *piconet*®-station (DP-address 15)
  - coupling module SDPL-0404D-1003, 4 digital inputs and 4 digital outputs
  - IP-Link-extension module 1: SNNE-0808D-0001, 8 digital inputs and 8 digital outputs
  - IP-Link-extension module 2: SNNE-0800D-0007, 8 digital inputs
  - IP-Link-extension module 3: SNNE-40A-0004, 4 analog inputs (TC)
  - IP-Link-extension module 4: SNNE-0008D-0006, 8 digital outputs

10.3.2 Configuration of the station in CoDeSys V3

When configuring the *piconet*®-station, i9nd CoDeSys, please observe that analog channels always have to be configured before the digital channels (see figure)!

Figure 10-4: Configuration of the station in CoDeSys V3



10.3.3 Reading the diagnosis via acyclic services

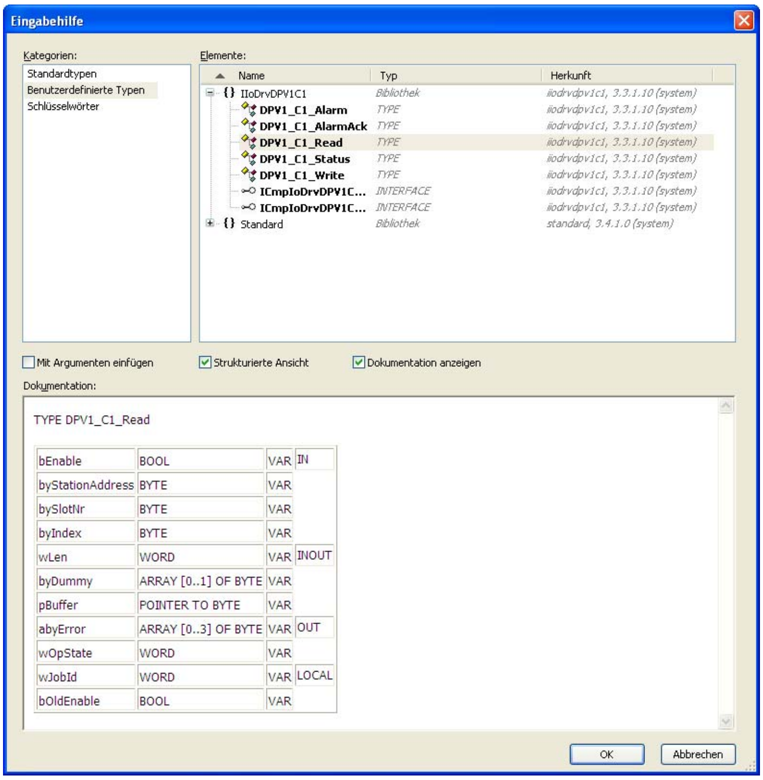
Using the library "IIoDrvDPV1C1.library“ of CoDeSys V3

By means of the library "IIoDrvDPV1C1.library“, the VT250 works as PROFIBUS-DPV1-Master, class 1.

The library is generated automatically when using the DP-Master.

For acyclic reading of the IP-Link diagnosis, the function "IoDrvDPV1_C1_M_Read“ with the structure "DPV1_C1_M_Read“ is used.

Figure 10-5: DPV1_C1_M_Read



The variables of the function block are defined as follows:

Table 10-3: Description of DPV1_C1_M_Read

Variable	Data type	Description
bEnable	BOOL	Enable Flag of the service (input)
byStationAddress	BYTE	station address of the slave (input)
bySlotNr	BYTE	slot number of the slave (input)
byIndex	BYTE	Index number of the slave (input)
wLen	WORD	Length of the data to be read
byDummy	ARRAY [0..1] OF BYTE	reserved (in-/output)
pBuffer	POINTER TO BYTE	Pointer to the memory area to which the read data have to be written (in-/ output).
abyError	ARRAY [0..3] OF BYTE	reserved (in-/output)

Table 10-3: Description of DPV1_C1_M_Read

Variable	Data type	Description
wOpState	WORD	Operation status (output)
wJobId	WORD	Internal variable which must not be used (local).
bOldEnable	BOOL	Internal variable which must not be used (local).

The following variables are important for reading out the IP-Link diagnosis:

- **bEnable**:
  Activating the read command
- **byStationAddress**:
  DP-address of the *piconet*® coupling module (here addr. 15)
- **bySlotNr.**:
  Slot-no. of the coupling module (always 0)
- **byIndex**:
  Number of the register table to be read (register table 90 or register table 50 - 60, see General structure of the data in the register tables (page 10-3)).
- **wLen**:
  Length of data to be read.

Variable declaration

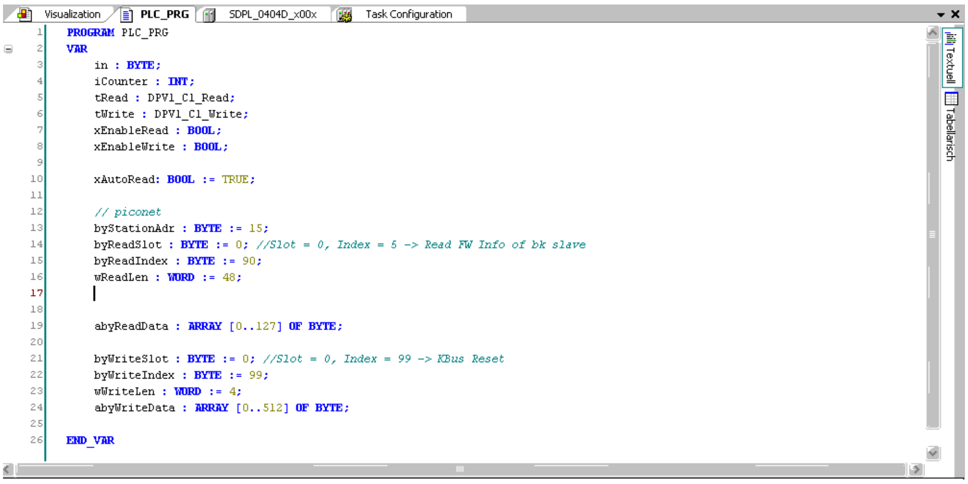
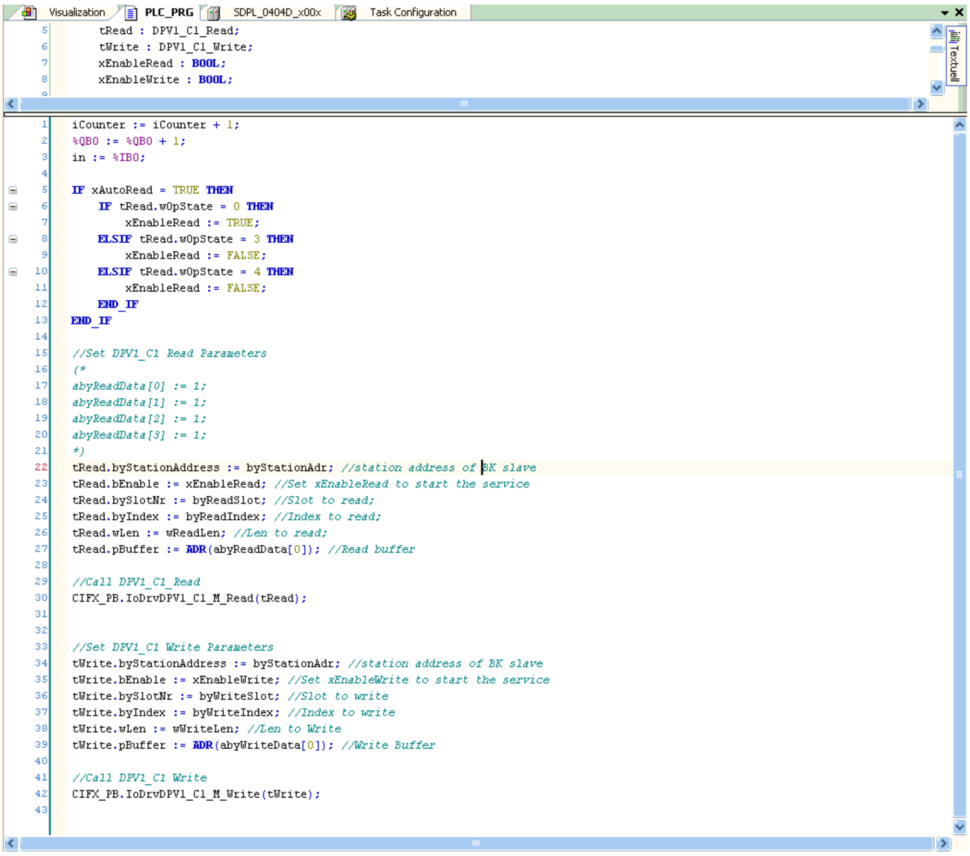


Figure 10-6: Example of variable declaration

Example program

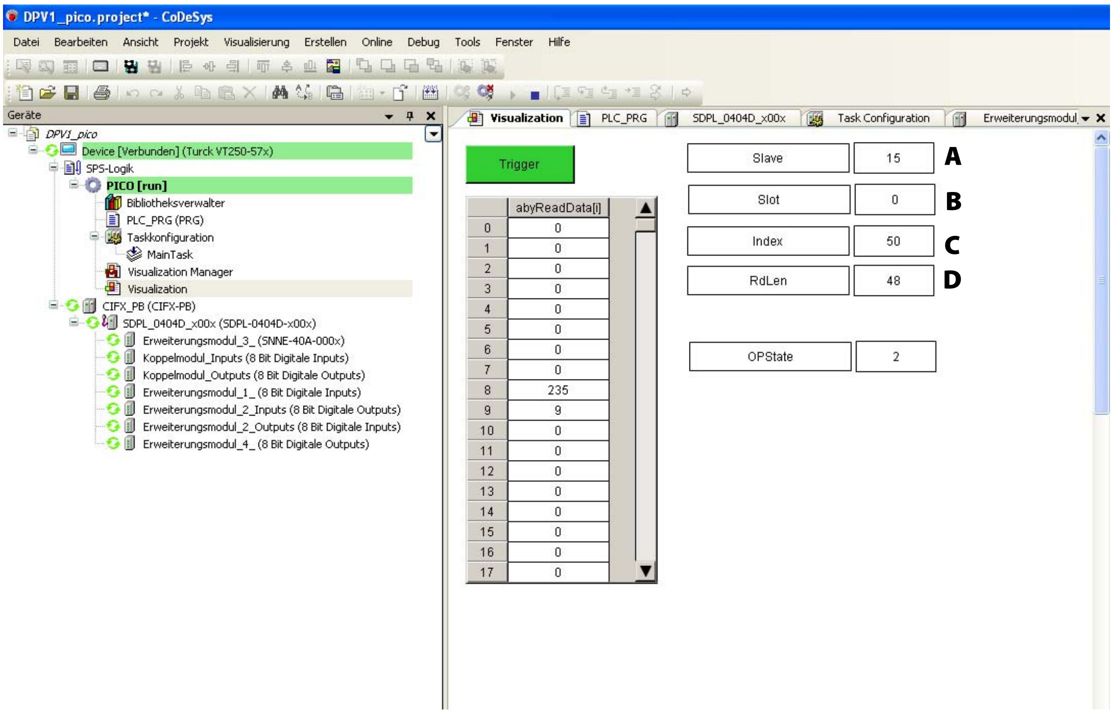
Figure 10-7:
Example program



Example visualization

Figure 10-8:
Example program

A DP-address (byStation Address)
B Slot-no. of the coupling module (bySlotNr)
C No. of the register table to be read (byIndex)
D LwLen



CoDeSys V3 - reading the IP-Link error counter from register table 90

For reading register table 90 the following entries have to be done:

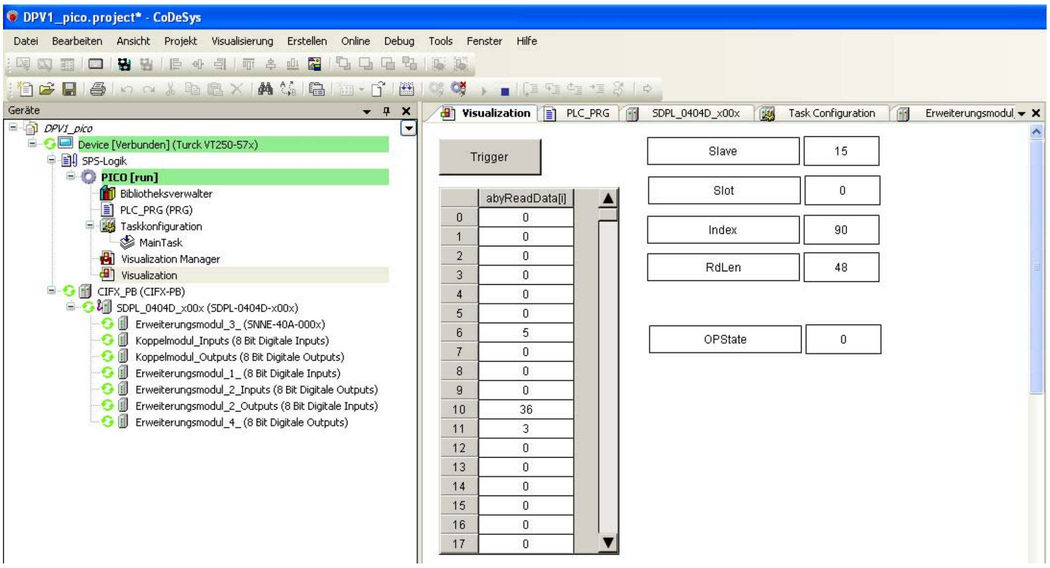
Slave = 15

Slot = 0

Index = 90

RdLen = 48

Table 10-4: Reading register table 90



The read are displayed in ."abyReadData“.

The error counter bytes 10 and 11 shows that error streams arrive at the coupling module.and that the IP-Link communication is disturbed.

CoDeSys V3 - reading the IP-Link error counter from register table 50

For reading register table 50 the following entries have to be done:

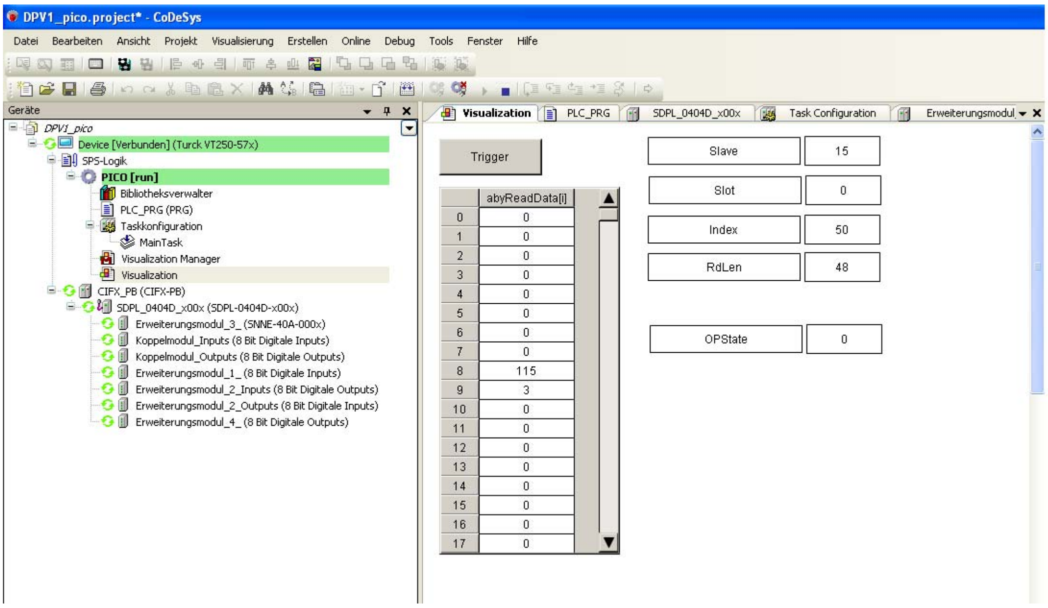
Slave = 15

Slot = 0

Index = 50

RdLen = 48

Table 10-5: Reading register table 50



The read are displayed in ."abyReadData".

The error counter in byte 8 and 9 shows that error streams arrive at the extension module.(see also Register tables 50 - 60 (error counter of the extension modules) (page 10-4)) and that the IP-Link communication is disturbed at this position.

11 Glossary

A

Acknowledge

Acknowledgement of the receiver to confirm receipt of the signal.

Address

A number for identification, e.g. for a memory location, a system or a module within a network.

Addressing

Allocation or setting of an address, e.g. for a module within a network.

Active conductive part

Conductor or conductive component tHat is energized during operation.

Analogue

A value – e.g. a voltage - that is infinitely proportional. With analogue signals the value of the signal can assume any value within certain limits.

Automation device

A device for connection of inputs and outputs, which is connected within a technical process. Programmable logic controllers (PLC) belong to a certain category of automation devices.

B

Baud

Unit of measure for the transmission speed of data. One baud accords to one step per second. If one bit is transferred per step, then the baud rate is identical to the transmission rate in bit per second.

Baud rate

see "Baud"

Bidirectional

Working in both directions.

Binary code

Coding method with which the contents to be coded is reproduced in form of logical binary characters (0 and 1) or character strings. Binary codes are suited for coding of numerical and alpha-numerical characters.

Bus

A group signal line for data transfer, e.g. between the central processing unit (CPU), memory and I/O level. A bus can consist of several parallel lines for data transfer, addressing, control and power supply.

Bus cycle time

The time interval in which the master adresses and communicates with all slaves within the bus system, i.e. the time in which the master writes the slave outputs and reads the slave inputs.

Bus line

Smallest unit connected to the bus; consisting of a PLC, a coupling element to couple the modules to the bus and a module.

Bus system

The entirety of components that communicate via a bus.

C

Capacitive coupling

A capacitive (electrical) coupling occurs between two conductors with different potentials. Typical sources of interference are, for example, parallely routed signal lines, contactors and static discharge.

Coding element

A two-part component for clear assignment of electronic and base module.

Command-capable modules

Command-capable modules are modules with an internal command rountine, which are capable of executing certain commands (e.g. output of substitute values).

Configuration

Systematic arrangement of the I/O modules of a station.

CPU

The abbreviation for "Central Processing Unit". Central unit for data processing, the core component of the computer.

D

Digital

A value – e.g. of a voltage – which can only assume a certain condition, usually defined as 0 and 1.

DIN

The abbreviation for "Deutsches Institut for Normung e.V."

E

Earth

An electrotechnical term used to signify conductive earth whose electrical potential is always zero. The electrical potential of the earth can be unequal zero in the proximity of earthing systems. In this case the term "ground reference plane" would be used.

Earth electrode

One or several components which are in direct and good contact with the earth.

EIA

The abbreviation for "Electronic Industries Association“. An association of companies belonging to the electronic industries in the USA.

Electrical equipment

All devices that are used for generation, conversion, transmission, distribution and use of electrical energy, such as conductors, cables, machines, control sevices etc.

EMC

The abbreviation for "Electromagnetic Capability". The term EMC describes the capability of an electrical apparatus to function correctly within a certain environment without having a negative influence on the environment.

Equipotential bonding

Adaptation of the electrical levels of the frame of electrical apparatus and external conductive components via an electrical connection.

ESD

The abbreviation for "Electro Static Discharge".

Exposed conductive part

An exposed conductive part is electrically isolated from the active conductive part but can become energised in the event of an error.

F

Fieldbus

Data network at the sensor/actuator level. The fieldbus connects the field devices. The fieldbus is characterised by the high transmission reliability and real time performance.

Field supply

Supply of voltage to power the field devices as well as the signal voltage.

Force mode

A software mode, in which it is possible to set certain fixed variables of input and output modules to simulate certain system conditions.

Full duplex

Also called duplex. Physical or logical connection of two terminal points to establish a data transmission channel. Data can be sent or received simultaneously in both directions. Full duplex cables have two wires. In full duplex operation either both channels or only one channel are used. If data are transferred via a single channel, this takes place in the multiplex mode. That means that data are transferred alternately but with a very high frequency so that the impression of simultaneous data transfer is given.

G

Galvanic coupling

Galvanic coupling generally occurs if two current circuits share a common line. Typical interference sources are, for example, starting motors, static discharge, clocked devices and a potential difference between component housings and the mutual power supply.

GND

The abbreviation for "Ground" (zero potential).

Gray-Code

Binary code for reproduction of integers. Two consecutive values only differ in a single bit.

Ground

All linked inactive parts of an electrical apparatus, which will not assume a touch voltage even in the event of an error.

Ground reference plane

Ground potential in the proximity of earthing systems. In contrast to the "earth", whose potential is always zero, it can have a different a potential than zero.

Grounding strip

Usually a flexible braided conductor that connects the inactive parts of the electrical equipment, e.g. the door of a switching cabinet with the switching cabinet corpus.

Grouping

A power supply module forms a new potential group. Thus the load and sensor supply can be fed separately.

GSD

The German term for device data base file (DDBF). The GSD file contains standardised descriptions of PROFIBUS modules. GSD files are used to simplify configuration of the DP master and the DP slaves.

H

Half duplex

Physical or logical connection of two terminal points to establish a data transmission channel. In contrast to full duplex operation, data can be transferred in both directions but, however, not simultaneously. Both terminal stations are equipped with a switch to toggle between sending and receiving data.

Hexadecimal

Numerical system with 16 as basis. One counts from 0 to 9 and then continues with the letters A, B, C, D, E and F.

Hysteresis

An encoder can stop at a certain point and then "swing" around this position. This will lead to a fluctuation of the count at a certain value. If a reference value is within this range of fluctuation, the associated output will switch on and off in the rhythm of the oscillation.

I

I/O

The abbreviation for "Input/Output".

Impedance

The resistance of a component or a circuit of several components for an AC current of a certain frequency.

Inductive coupling

An inductive (magnetic) coupling between two current-carrying conductors. The magnetic effect caused by the currents induces an interference voltage. Typical sources of interference are, for example, transfomers, motors, parallely routed power cables and high-frequency signal lines.

L

Lightning protection

All measures, that can help protect a system against damage caused by excessive voltages due to lightning.

Low impedance connection

Connection with a low AC resistance.

LSB

The abbreviation for "Least Significant Bit". The bit with the lowest significance.

M

Master

A bus station or a bus node which controls the communication between the other bus devices.

Master Slave Mode

An operating mode in which one station or node controls the communication over the bus as a master.

Mode

The operating mode of a system, component etc.

Module bus

The module bus is the internal bus of a BL67 station. The BL67 modules communicate via the module bus with the gateway. It is independent of the fieldbus.

MSB

The abbreviation for "Most Significant Bit". The bit with the highest significance.

Multimaster Mode

An operating mode in which all stations or node have equal rights to communicate over the bus.

N

NAMUR

The abbreviation for "Normen-Arbeitsgemeinschaft für Mess- und Regeltechnik". NAMUR sensors are special versions of 2-wire sensors. Due to their special construction, i.e. low internal resistance, only very few components, short housings, Namur initiators are particularly interference immune and provide a high level of operational safety.

O

Overhead

System administration time required by the the system for each transmission cycle.

P

Parameterisation

Determination of parameters of the individual bus stations, or their modules via the configuration software of the DP master.

PLC

The abbreviation for "Programmable Logic Controller"

Potential-free

Galvanic isolation of the reference potentials of the control and load current circuits of I/O modules.

Potential-bound

Electrical connection of the reference potentials of the control and load current circuits of I/O modules.

PROFIBUS-DP

PROFIBUS bus system with DP protocol. DP stands for "Decentralised Periphery"
The PROFIBUS-DP is based on DIN 19245 part 1+4 and was integrated in the European fieldbus standard EN 50170.
It is designed for fast cyclic data transfer between the central DP master and the remote peripheral components, i.e. the DP slaves. Consistent usage is realised by a multimaster concept.

PROFIBUS-DP address

Every PROFIBUS-DP module has a unique address via which it can be addressed by the master.

PROFIBUS-DP Master

As the central bus component, the PROFIBUS-DP master controls the access of all PROFIBUS-DP slaves to the PROFIBUS.

PROFIBUS-DP Slave

PROFIBUS-DP slaves are addressed by the PROFIBUS-DP master and exchange data with the master upon receipt of a master poll request.

Protective earth conductor

A conductor needed for protection against dangerous shock currents, signified by the abbreviation PE for "Protective Earth".

R

Radiated coupling

Radiative coupling occurs if an electromagnetic wave meets a conductor structure. The wave will induce currents and voltages in the conductor. Typical sources of interference are, for instance, sparking gaps (sparking plugs, collectors of electro-motors) and emitters (e.g. radio interference), which are operated near the affected conductor structure.

Reference potential

Potential, which serves as a reference for the measurement or assessment of the voltage of all connected circuits.

Response time

In a bus system this term is used to define the time interval between sending a read command and the receipt of a response. If referring to an input module, it describes the time interval between a signal change at the module input and the signal output to the bus system.

Repeater

Amplifier for signals transferred via the bus.

RS 485

Serial interface according to EIA standards for fast data transfer via several transmitters.

S

Shield

This term is used to describe the conductive sheath of cables, casings and cabinets.

Shielding

The entirety of all measures and equipment used to connect the system parts to the shield.

Serial

This term is used to define a data transmission mode with which data are transferred consecutively - bit by bit - via a cable.

Short-circuit proof

Property of electrical apparatus. A short-circuit proof apparatus withstands the thermal and dynamic stress which can occur at its place of installation due to a short-circuit.

Slave

A bus station or a bus node that is subordinate to the master.

Station

A functional unit or module assembly consisting of several components.

T

Terminating resistor

Resistor on both ends of the bus line to prevent disturbing signal reflections and to adjust bus lines. Terminating resistors must always be the physically last unit at the end of a bus segment.

Topology

Geometric construction of a network or arrangement of circuits.

U

UART

The abbreviation for "Universal Asynchronous Receiver/Transmitter". UART is a logic circuit which is used to transform an asynchronous serial data string into a bit parallel data string or vice versa.

Unidirectional

Working in one direction.

12 Index

TURCK

Industrial
Automation

www.turck.com

D300776 0511

Hans Turck GmbH & Co. KG
45472 Mülheim an der Ruhr
Germany
Witzlebenstraße 7
Tel. +49 (0) 208 4952-0
Fax +49 (0) 208 4952-264
E-Mail more@turck.com
Internet www.turck.com