Manual

Protocols

PROFINET, PROFIsafe,
PROFIBUS DP

All of the technical specifications and information in this manual were prepared with great care and effective control measures were employed for their compilation. For questions, please contact HIMA directly. HIMA appreciates any suggestion on which information should be included in the manual.

Equipment subject to change without notice. HIMA also reserves the right to modify the written material without prior notice.

All the current manuals can be obtained upon request by sending an e-mail to: documentation@hima.com.

Contact

HIMA Paul Hildebrandt GmbH

P.O. Box 1261

68777 Brühl, Germany

Phone: +49 6202 709-0

Fax: +49 6202 709-107

E-mail: info@hima.com

Document designation	Description
HI 801 514 D, Rev. 11.04 (2020)	German original document
HI 801 523 E, Rev. 11.04.00 (2027)	English translation of the German original document

Table of Contents

1 Introduction

This manual describes the properties of the PROFINET, PROFIsafe and PROFIBUS DP protocols and their configuration in SILworX for the safety-related HIMA controller systems.

Knowledge of regulations and the proper technical implementation of the instructions detailed in this manual performed by qualified personnel are prerequisites for planning, engineering, programming, installing, starting up, operating and maintaining the HIMA controllers.

HIMA shall not be held liable for severe personal injuries, damage to property or the environment caused by any of the following: unqualified personnel working on or with the devices, de-activation or bypassing of safety functions, or failure to comply with the instructions detailed in this manual (resulting in faults or impaired safety functionality).

HIMA automation devices have been developed, manufactured and tested in compliance with the pertinent safety standards and regulations. They may only be used for the intended applications under the specified environmental requirements.

1.1 Structure and Use of This Manual

The manual contains the following chapters:

- Introduction
- Safety
- Product description
- PROFINET
- PROFIsafe
- PROFIBUS DP

Additionally, the following documents must be taken into account:

Name	Content	Document no.
HIMax system manual	Hardware description HIMax system	HI 801 001 E
HIMax safety manual	Safety function HIMax systems	HI 801 003 E
HIMatrix safety manual	Safety function HIMatrix systems	HI 800 023 E
HIMatrix compact system manual	Hardware description HIMatrix compact system	HI 800 141 E
HIMatrix modular system manual	Hardware description HIMatrix modular F 60 system	HI 800 191 E
HIQuad X system manual	Hardware description HIQuad X system	HI 803 211 E
HIQuad X safety manual	Safety function HIQuad X system	HI 803 209 E
Automation security manual	Description of automation security aspects related to the HIMA systems	HI 801 373 E
SILworX first steps manual	Introduction to SILworX.	HI 801 103 E

Table 1: Additional Applicable Manuals

All the current manuals can be obtained upon request by sending an e-mail to: documentation@hima.com. The documentation is available for registered HIMA customers in the download area https://www.hima.com/en/downloads/.

1.2 Target Audience

This document is aimed at the planners, design engineers, programmers and the persons authorized to start up, operate and maintain the automation systems. Specialized knowledge of safety-related automation systems is required.

1.3 Writing Conventions

To ensure improved readability and comprehensibility, the following writing conventions are used in this document:

Bold	To highlight important parts. Names of buttons, menu functions and tabs that can be clicked and used in the programming tool.
Italics	Parameters and system variables, references.
`Courier`	Literal user inputs.
RUN	Operating states are designated by capitals.
Chapter 1.2.3	Cross-references are hyperlinks even if they are not specially marked. In the electronic document (PDF): When the mouse pointer hovers over a hyperlink, it changes its shape. Click the hyperlink to jump to the corresponding position.

Safety notices and operating tips are specially marked.

1.3.1 Safety Notices

Safety notices must be strictly observed to ensure the lowest possible risk.

The safety notices are represented as described below.

- Signal word: warning, caution, notice.
- Type and source of risk.
- Consequences arising from non-observance.
- Risk prevention.

The signal words have the following meanings:

- Warning indicates hazardous situations which, if not avoided, could result in death or serious injury.
- Caution indicates hazardous situation which, if not avoided, could result in minor or moderate injury.
- Notice indicates a hazardous situation which, if not avoided, could result in property damage.

⚠ SIGNAL WORD

Type and source of risk!

Consequences arising from non-observance.

Risk prevention.

NOTICE

Type and source of damage!

Damage prevention.

1.3.2 Operating Tips

Additional information is structured as presented in the following example:

i	The text giving additional information is located here.

Useful tips and tricks appear as follows:

TIP	The tip text is located here.

1.4 Safety Lifecycle Services

HIMA provides support throughout all the phases of a plant's safety lifecycle, from planning and engineering through commissioning to maintenance of safety and security.

HIMA's technical support experts are available for providing information and answering questions about our products, functional safety and automation security.

To achieve the qualification required by the safety standards, HIMA offers product or customer-specific seminars at HIMA's training center or on site at the customer's premises. The current seminar program for functional safety, automation security and HIMA products can be found on HIMA's website.

Safety Lifecycle Services:	
Onsite+ / On-Site Engineering	In close cooperation with the customer, HIMA performs changes or extensions on site.
Startup+ / Preventive Maintenance	HIMA is responsible for planning and executing preventive maintenance measures. Maintenance actions are carried out in accordance with the manufacturer's specifications and are documented for the customer.
Lifecycle+ / Lifecycle Management	As part of its lifecycle management processes, HIMA analyzes the current status of all installed systems and develops specific recommendations for maintenance, upgrading and migration.
Hotline+ / 24 h Hotline	HIMA's safety engineers are available by telephone around the clock to help solve problems.
Standby+ / 24 h Call-Out Service	Faults that cannot be resolved over the phone are processed by HIMA's specialists within the time frame specified in the contract.
Logistics+/ 24 h Spare Parts Service	HIMA maintains an inventory of necessary spare parts and guarantees quick, long-term availability.

Contact details**:**	
Safety Lifecycle Services	https://www.hima.com/en/about-hima/contacts-worldwide/
Technical Support	https://www.hima.com/en/products-services/support/
Seminar Program	https://www.hima.com/en/products-services/seminars//

2 Safety

All safety information, notes and instructions specified in this document must be strictly observed. The product may only be used if all guidelines and safety instructions are adhered to.

The product is operated with SELV or PELV. No imminent risk results from the product itself. Use in the Ex zone is only permitted if additional measures are taken.

2.1 Intended Use

To use the HIMA controllers, all pertinent requirements must be met, see relevant manuals in Table 1.

2.2 Residual Risk

No imminent risk results from a HIMA system itself.

Residual risk may result from:

- Faults related to engineering.
- Faults in the user program.
- Faults related to the wiring.

2.3 Safety Precautions

Observe all local safety requirements and use the protective equipment required on site.

2.4 Emergency Information

A HIMA system is a part of the safety equipment of an overall system. If the controller fails, the system enters the safe state.

In case of emergency, no action that may prevent the HIMA system from operating safely is permitted.

2.5 Automation Security for HIMA Systems

Industrial controllers must be protected against IT-specific problem sources. Those problem sources are:

- Attackers inside and outside of the customer's plant
- Operating failures
- Software error.

All requirements for protection against manipulation specified in the safety and application standards must be met. The operator is responsible for authorizing personnel and implementing the required protective actions.

⚠ WARNING

Physical injury possible due to unauthorized manipulation of the controller!

Protect the controller against unauthorized access!

For example:

- **Changing the default settings for login and password!**
- **Controlling the physical access to the controller and PADT!**

Careful planning should identify the measures to implement. The required measures are to be implemented after the risk analysis is completed. Such measures can include:

- Meaningful allocation of user groups.
- Maintained network maps help to ensure that secure networks are permanently separated from public networks and, if required, only a well-defined connection exists (e.g., via a firewall or a DMZ).
- Use of appropriate passwords.

A periodical review of the security measures is recommended, e.g., every year.

The user is responsible for implementing the necessary measures in a way suitable for the plant!

For further details, refer to the HIMA automation security manual (HI 801 373 E).

3 PROFINET IO

PROFINET IO is the transmission protocol provided by PNO Germany and based on Ethernet technology.

As with PROFIBUS DP, the remote field devices for PROFINET IO are integrated in SILworX via a device description (GSDML file).

The PROFINET IO-Controller complies with Conformance Class A and supports non-real time (NRT) and real time (RT) communication with the PROFINET IO-Devices. Real time communication is used for time critical data exchange, and non-real time communication for non time critical processes, such as acyclic read/write operations.

A redundant PROFINET IO connection can be implemented by configuring a second PROFINET IO-Controller/Device and adjusting it in the user program.

3.1 PROFINET IO Function Blocks

To acyclically exchange data, function blocks with the same functionality as with SILworX are available in PROFIBUS DP.

The following PROFINET IO function blocks are available:

Function block	Function description
MSTAT 4.9.1	To control the controller state using the user program
RALRM 4.9.2	To read the alarm messages from the devices
RDREC 4.9.4	To read the data records of the devices
SLACT 4.9.5	To control the device states using the user program
WRREC 4.9.6	To write the data records of the devices

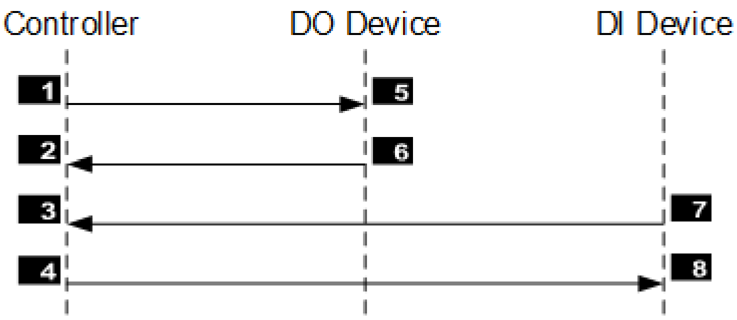
Table 2: Overview of PROFINET IO Function Blocks

The PROFINET IO function blocks are configured like the PROFIBUS DP function blocks. Refer to Chapter 4.9 for details.

3.2 Controlling the Consumer/Provider Status (IOxS)

The system variables described in this chapter, the Consumer/Provider status (IOxS) can be controlled via the user program. If the Consumer/Provider status should not be controlled via the user program, the output variables must be assigned a constant that is set to TRUE. The statuses are then set to GOOD as soon as the communication module obtains valid process values from the processor module.

The following picture shows how system variables are exchanged between the HIMA controller and a DO device or a DI device, respectively.



1	Valid Output Data	5	Output Data accepted by Device
2	Output Data Accepted by Device	6	Valid Output Data
3	Valid Input Data	7	Input Data Accepted by Device
4	Input Data Accepted by Controller	8	Valid Input Data

Figure 1: Controlling the Consumer/Provider Status (IOxS)

3.2.1 Control Variables in the HIMA Controller

The *Valid Output Data* 1 and *Input Data Accepted by Controller* 4 input variables can be used to control the Consumer/Provider status (IOxS) via the user program.

The *Output Data Accepted by Device* 2 and *Valid Input Data* 3 input variables can be used to read the Consumer/Provider status (IOxS) via the user program.

3.2.2 Control Variables in the HIMA DO Device

The *Valid Output Data* 6 output variable can be used to control the Consumer/Provider status (IOxS) via the user program.

The *Output Data Accepted by Controller* 5 input variable can be used to read the Consumer/Provider status (IOxS) via the user program.

3.2.3 Control Variables in the HIMA DI Device

The *Input Data Accepted by Device* 7 output variable can be used to control the Consumer/Provider status (IOxS) via the user program.

The *Valid Input Data* 8 input variable can be used to read the Consumer/Provider status (IOxS) via the user program.

3.3 PROFIsafe

Familiarity with the PROFIsafe specification from the PNO is assume!

PROFIsafe uses the PROFINET protocol to transfer safety-related data up to SIL 3, based on Ethernet technology.

The PROFIsafe protocol is superposed on the PROFINET protocol and contains safe user data, as well as data backup information. The safe PROFIsafe data is transferred together with the non-safety-related PROFINET data via the subsidiary PROFINET protocol.

Similarly to the black-channel principle, PROFIsafe uses unsafe data transmission channels (Ethernet) to transfer safe data. This allows the F-Host and F-Device to exchange safe PROFIsafe data.

3.3.1 PROFIsafe in Connection with HIMA Controllers

According to the PROFIsafe specification, the F-Host repeatedly sends a message packet until the F-Device acknowledges its receipt to the F-Host. Only then does the F-Host send a new message packet to the F-Device.

The current process value is transferred in every resent PROFIsafe message packet. It may thus happen that the same process signal has different values in the resent message packets.

In HIMA devices, PROFIsafe is implemented on the receiver side such that process values can only be adopted when the message packet is received for the first time.
The process values of the resent message packets (with identical progressive number of the message packet) are rejected.

If the connection is lost and after *F_WD_Time* has expired, the PROFIsafe process value variables adopt the initial value.

A given process value is only received on the opposite side (F-Host/F-Device) if the process value remains unchanged for at least the following time:

*2 * F_WD_Time + F_WD_Time2*

The PROFIsafe system must be configured so that the *SFRT* (Safety Function Response Time) is suitable for performing the corresponding safety function.

For the calculation formula, refer to Chapter 3.3.4.

---

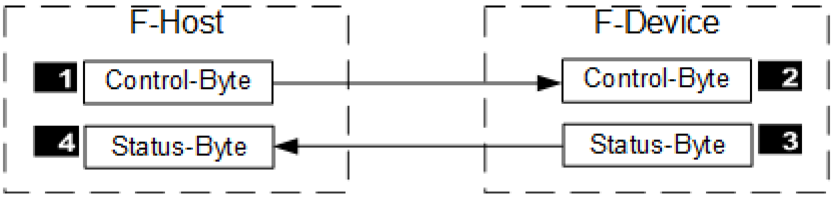
i The following conditions must be met to ensure a behavior consistent with PROFIsafe:
- The initial values of the process value variables must be set to 0.
- The *AutoAwaitFParamsOnConnLoss* parameter must be deactivated, see Chapter 3.10.

The SILworX default settings are configured to ensure a behavior consistent with PROFIsafe!

3.3.2 PROFIsafe Control Byte and Status Byte

Both system variables Control Byte and Status Byte are contained in each PROFIsafe submodule and are exchanged during communication between F-Host and F-Device; refer to Chapter 3.7.6.3 and Chapter 3.10.4 for details.

The PROFIsafe control byte is written to the F-Host and read in the F-Device.
The PROFIsafe status byte is written to the F-Device and read in the F-Host.



1 Writing the control byte
2 Reading the control byte
3 Writing the status byte
4 Reading the status byte

Figure 2: PROFIsafe Control Byte and Status Byte

---

i The Control Byte and Status Byte system variables have additional features in HIMA systems that deviate from the PROFIsafe specification, see Table 12 and Table 29.

3.3.3 F_WD_Time (PROFIsafe Watchdog Time)

The following inequality applies to functional PROFIsafe connection between HIMA F-Host and F-Device:

F_WD_Time >
3 ∗ CPU cycle time ∗ number of communication time slices +
2 ∗ PROFINET controller production interval[1)] +
1 ∗ DAT (F-Device acknowledgement time) +
2 ∗ internal F-Device bus time +
2 ∗ PROFINET device production interval[1)] +
2 ∗ Ethernet delay

[1)] In general, PROFINET controller and PROFINET device production intervals are identical and are calculated as follows: *Reduction factor ∗ SendClockFactor ∗ 31.25 µs*

---

i Refer to the device description provided by the F-Device manufacturer for the values of *DAT* (F-Device acknowledgment time) and the *internal device bus time*!

For HIMax and HIMatrix F-Devices, *DAT = DATout = DATin = 2 * WDT CPU*

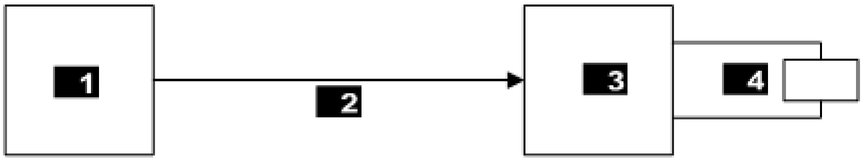
3.3.3.1 Remarks about F_WD_Time (PROFIsafe Watchdog Time)

1. DAT (F-Device acknowledgement time) is the time required by an F-Device to respond to a received PROFIsafe message. F-Devices are the safe units (in HIMA systems the CPU module) processing the F-Device stacks. In particular, if modular systems/devices are used, they do not include the time values for the non safety-related functions/components.
   This DAT definition differs from that provided in the PROFIsafe specification V2.5c, Chapter 9.3.3, in the following points:
   - DAT does not include the time values for the internal F-Device bus.
   - DAT does not include the portion of PROFINET device production intervals.
   - DAT does not include any delays, e.g., due to input or output value filters or the physical properties of the inputs and outputs.
   - DAT refers to DATin (input) or DATout (output), depending on the connection.
   - The corresponding maximum value must be used for all time parameters.
2. In HIMatrix and in HIMax, the internal F-Device bus time is:
   (Max. number of communication time slices – 1) ∗ WDT CPU.
3. Requirement: The F-Device runs cyclically and its DAT is:
   DAT = 2 ∗ max. cycle
   - F-Device **does not operate with** *communication time slices.*
     If
     HIMA CPU cycle time ∗ number of communication time slices is less than the *F-Device cycle time,*
     Delta = F-Device cycle time – HIMA CPU cycle time ∗ number of communication time slices must be added to the *HIMA CPU cycle time* specified in the *F_WD_Time* equation*.*
   - F-Device operates **with** *communication time slices*.
     If
     (HIMA CPU cycle time ∗ number of communication time slices) is less than the *F-Device cycle time ∗ number of F-Device communication time slices,*
     for the *HIMA CPU cycle time*
     *Delta = (F-Device cycle time * number of F-Device communication time slices) – (HIMA CPU cycle time ∗ number of communication time slices)*
     must be added to the calculation of the *F_WD_Time.*

3.3.4 SFRT (Safety Function Response Time)

3.3.4.1 Calculation of the SFRT between an F-Device and a HIMA F-Host

The maximum SFRT allowed for a PROFIsafe connection between an F-Device and a HIMA F-Host with local output is calculated as follows:



1 F-Device
2 PROFIsafe protocol
3 HIMA F-Host
4 Output

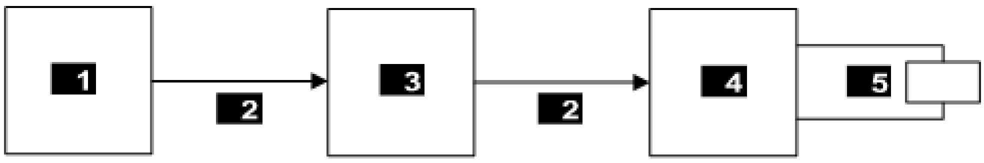
Figure 3: Response Time between an F-Device and a HIMA F-Host

SFRT <= MaxDataAgeIn + 2 ∗ F_WD_TIME + MaxDataAgeOut + Tu

Remarks:
With HIMax/HIMatrix and if data is used locally, i.e., if it is not output to a HIMax I/O, the fault tolerance time of the CPU module can be replaced by
2∗WDT CPU. See also the *Remark on SFRT calculations* in Chapter 3.3.4.3.

3.3.4.2 Calculation of the SFRT using an F-Device and a HIMA F-Host

The maximum SFRT allowed for a PROFIsafe connection between a HIMA F-Host and an F-Device with local output is calculated as follows:



1 F-Device
2 PROFIsafe protocol
3 HIMA F-Host
4 F-Device
5 Output

Figure 4: Response Time using 1 HIMA F-Host and 2 F-Devices

SFRT <= MaxDataAgeIn + 2 ∗ F_WD_TIME(input) + 2 ∗WDT CPU +
2 ∗ F_WD_TIME(output) + MaxDataAgeOut + Tu

3.3.4.3 Remark on the SFRT Calculations

1. Definition of **SFRT** in accordance with IEC 61784-3, Ed.2.
2. All additional delays in the user program must be added, e.g., due to TOF or TON function blocks, or in the modules, due to output filters, input filters, relays, etc.
3. **MaxDataAgeIn** is the maximum age of a process value that is read on a physical input and is added to a PROFIsafe message by an F-Device, <u>but only the portion that is not already contained in DATin</u>.

i In HIMatrix , MaxDataAgeIn must be set to 0 ms.
In a HIMax system, the MaxDataAgeIn value must be set as follows:

- *FTT CPU*[1] *– 2 ∗ WDT CPU – DATin,* for physical inputs.
- 0 ms, for data created by the user program.

4. **MaxDataAgeOut**
   is the worst case response time of an output F-Device or F-Host for
   a) outputting the received process values to a physical output,
   b) controlling the physical outputs after *F_WD_Time* has expired and
   c) deactivating the physical outputs if the devices fail.
   - In HIMatrix, *MaxDataAgeOut = 3 ∗ WDT CPU* (for physical outputs)
   - In HIMax, *MaxDataAgeOut = WDT CPU + FTT CPU*[1] (for physical outputs)
   - After *F_WD_TIME* has expired, the HIMax and HIMatrix respond without faults at the latest after *2 ∗ WDT CPU*.
     - If the F-Host/F-Device (HIMA CPU module) fails immediately prior to this response, the outputs of HIMatrix are de-energized once *WDT CPU* has expired. In the HIMax system, this occurs at the latest after *FTT CPU*[1] *– WDT CPU* since the output module has not received any message for the duration of a *WDT CPU*.
     - Assuming that only **one** fault occurs in HIMatrix or HIMax, *1 ∗ WDT CPU* can be subtracted from *MaxDataAgeOut*.
5. **Tu** is the minimum value of *DATin, DATout, WDT CPU.* Theoretically, it is possible to use half of the value of *DATin* and half of the value of *DATout*, but the manufacturer must specify to which inaccuracy degree *F_WD_Time* is monitored by the device. If the device runs cyclically,
   DAT = 2 ∗ max. device cycle.
   For HIMA HIMatrix and HIMax F-Devices, *DATout = DATin = 2 * WDT CPU.*
6. **F_WD_TIME**, see Chapter 3.3.3.

[1] Process safety time of the CPU module

3.4 Requirements for Safely Operating PROFIsafe

3.4.1 Addressing

The HIMA PROFIsafe network corresponds to the PROFINET Ethernet network that can be used to transfer the PROFIsafe messages. In this context, network refers to a logical network that can include multiple physical sub-networks.

A separation is suitable for a PROFIsafe network if PROFIsafe messages cannot override the network separation.
This would be the case, if an IP-based router is used and the networks are connected to different Ethernet interfaces of the router.

The PROFIsafe networks are not separated if, for instance, the networks are connected via one port router, switches, hubs or Ethernet bridges.

Even if managed switches are used and the PROFIsafe networks are separated, e.g., via port-based VLANs, one-to-one addressing should still be aimed for. This ensures that no connections between PROFIsafe networks are accidentally established during upgrade or maintenance.

The following conditions must be met for addressing the PROFIsafe devices:

- A one-to-one correspondence between the F-addresses of the PROFIsafe devices/modules in a PROFIsafe network must be ensured.
- Additionally, HIMA recommends selecting bijective F-addresses, even in separate PROFIsafe networks, to ensure addressing safety.
- When starting up and modifying safety functions, ensure that the safety functions use the proper inputs and outputs of the corresponding PROFIsafe devices throughout the entire PROFIsafe network.
- The PROFIsafe F-modules must be configured such that F-modules with identical input and output data lengths that operate in a given PROFIsafe network have different CRC1 signatures, e.g., by assigning suitable F-addresses or F_WD_Time. The CRC1 can be read in SILworX.
  Remark:
  This is ensured in any case for the affected F-modules if only one F-Host is used in a given PROFIsafe network and the F-Parameters of the F-Modules in a PROFIsafe network only differ in the F-address.
  To avoid accidental generation of the same CRC1 signature, make sure that, e.g., the parameters *F_WD_Time, F_Prm_Flag1/2* are identical for all F-modules and the F-modules do not use an iPar CRC.

Risk Associated with PROFIsafe Devices with Identical Input and Output Data Length

PROFIsafe device may only be operated if the F-INPUT data length does not equal the F-OUTPUT data length of the same PROFIsafe connection.
Otherwise, potential addressing faults in standard components and/or standard transmission technologies might not be detected and could cause safety-related malfunctions.

In HIMA F-modules configured for the HIMA controller, the F-input data length must differ from the F-output data length. To prevent the risk of a safety-related malfunction, only use F-Input modules or F-Output modules. Do not use F-input/output modules!

3.4.2 Network Aspects

The network used for transferring PROFIsafe messages must ensure sufficient availability and transmission quality.

i A safety reaction is triggered if reduced transmission quality is detected by PROFIsafe, but is not recognized by the standard transmission devices (Ethernet).

After a safety reaction due to reduced transmission quality, the problems must be resolved to once again ensure sufficient transmission quality. A PROFIsafe restart may only be acknowledged after the required measures were taken. To do so, the Operator Acknowledge or Reset signal must be used.

WARNING

Operator Acknowledge and Reset may only be used if dangerous states no longer exist.

A PROFIsafe network must be protected against unauthorized access and actions (e.g., DoS, hacker, etc.). The measures must be agreed upon together with the supervising authority. This is particularly important if wireless transmission technologies are used.
For further details, refer to the PROFIsafe Specification V2.5c, Table 23 and Table 24.

Availability with respect to added messages

Message packets can be stored, e.g., using network components such as switches, and can be added (sent) to a later point in time[1)]. These message packets cause a shutdown if they are older than the last message packet received by the PROFIsafe device (see Consecutive Number in Table 12 and Table 29).

[1)] Fault assumption for safety-related communication

3.5 PROFINET IO-Controller and PROFIsafe F-Host

This chapter describes the features of the PROFINET IO-Controller and PROFIsafe F-Host, and the menu functions and dialog boxes required to configure the PROFINET IO-Controller and PROFIsafe F-Host in SILworX.

3.5.1 System Requirements for the PROFINET IO-Controller

Equipment and system requirements

Element	Description
Controller	HIMax with X-COM 01 module. HIMatrix
CPU module	The Ethernet interfaces on the processor module may not be used for PROFINET IO.
COM module	Ethernet 10/100BaseT.
Activation	Software activation code is required for activation, refer to the communication manual (HI 801 101 E) for details.

Table 3: System Requirements and Equipment for the PROFINET IO-Controller

3.5.2 PROFINET IO-Controller and PROFIsafe F-Host Properties

Properties	Description
Safety-related	PROFINET: No PROFIsafe: Yes
Transfer rate	100 Mbit/s full duplex
Transport path	Ethernet interfaces on the COM module. The Ethernet interfaces in use can simultaneously be employed for other protocols.
Conformance class	The PROFINET IO-Controller meets the requirements for conformance class A.
Real time class	RT Class 1
Max. number of PROFINET IO-Controller	One PROFINET IO-Controller can be configured for each COM module.
Max. number of application relations (ARs) for PROFINET IO-Devices	A PROFINET IO-Controller can establish an application relation (AR) with a maximum of 64 PROFINET IO-Devices.
Number of communication relations (CRs for each AR)	Standard: 1 input CR, 1 output CR, 1 alarm CR
Max. process data length of a CR	Output: max. 1440 bytes Input: max. 1440 bytes
Transmit clocking	Possible at device level using the *Reduction Rate* setting.
The following features apply to PROFIsafe	
Max. number of F-Hosts (HIMax) Max. number of F-Hosts (HIMatrix)	1024 512
Max. process data length of a CR	Output: max. 123 bytes user data + 5 bytes[1)] Input: max. 123 bytes user data + 5 bytes[1)]
Max. user data size HIMax HIMatrix	 1024 x 123 bytes = 125 952 bytes 512 x 123 bytes = 62 976 bytes
[1)] 5 bytes management data (status/control bytes and CRC)	

Table 4: PROFINET IO-Controller Properties

3.6 PROFINET IO/PROFIsafe Example

This example shows how a PROFINET IO-Controller connected to a PROFINET IO-Device is configured in a HIMax controller.

The PROFINET IO-Device is equipped with one PROFINET IO module and one PROFIsafe module. In the PROFINET IO-Controller, the PROFINET IO-Device must be configured as it is actually structured.

3.6.1 Creating a PROFINET IO-Controller in SILworX

To create a new PROFINET IO-Controller

1. In the structure tree, select **Configuration, Resource, Protocols.**
2. Right-click Protocols and select **New**, **PROFINET IO-Controller** from the context menu to add a new PROFINET IO-Controller.
3. Right-click **PROFINET IO-Controller** and select **Properties** from the context menu.
4. Enter the controller's device name in the **Name** field.
5. Select **COM Module**.

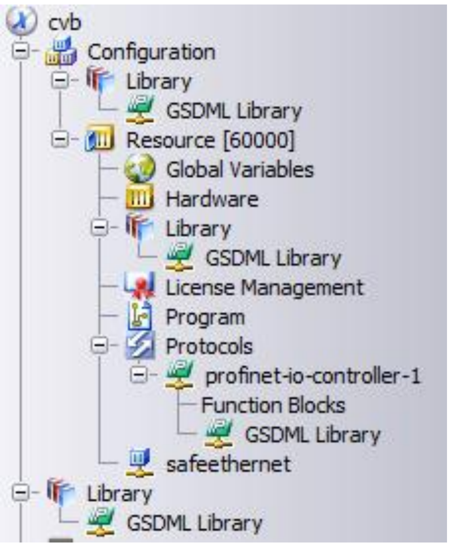


Figure 5: HIMax Structure Tree for the PROFINET IO-Controller

3.6.2 Creating the GSDML Library in SILworX

A GSDML library is used to manage the GSDML files that can be reused within a PROFINET IO-Controller.

A GSDML library can be created at project level, configuration level, resource level or directly below the PROFINET IO-Controller, see Figure 5.

If the GSDML library is created at project level, configuration level or resource level, it can be used by all PROFINET IO-Controllers included in it.

Creating the GSDML Library in SILworXat Configuration Level

1. If the project does not yet include a library, right-click **Project** in the structure tree and select **New**, **Library**.
2. In the structure tree, right-click **Project**, and select **New**, **GSDML Library**.

   ☑ A new GSDML library is created.

To read the GSDML file from an external data source (e.g., DVD, USB stick, Internet)

1. In the structure tree, select **Project**, **Library**, **GSDML Library**.
2. Right-click **GSDML Library** and select **New** from the context menu to open the **Add GSDML File** dialog box.
3. Click the **...** button and select the required GSDML files from the storage location.
4. Click **OK** to confirm.

3.6.3 Configuring the Device in the PROFINET IO-Controller

To create a HIMax PROFINET IO-Device within the PROFINET IO-Controller

1. Select **New**, **PROFINET IO-Device** from the context menu of the PROFINET IO-Controller.

To load the GSDML file for a new PROFINET IO-Device

1. In the structure tree**,** open **Configuration**, **Resource**, **Protocols**, **PROFINET IO-Device.**
2. Select **Properties** from the context menu and open the **Parameter** tab.
3. Enter the PROFINET IO-Device name in the **Name** field.
4. Enter the IP address of the PROFINET IO-Device in the IP Address field.
5. Select the GSDML library file specific to the PROFINET IO-Device from the drop-down menu for **GSDML File** and close **Properties**.

To select the device access point (DAP) for the PROFINET IO-Device

1. In the structure tree, select **Protocols**, **PROFINET IO-Controller**, **PROFINET IO-Device, DAP Module.**
2. Select **Edit** from the context menu and choose the appropriate DAP data record for the PROFINET IO-Device *(DAP = Device Access Point).*

i The GSDML file often contains several *DAPs* from the same manufacturer.

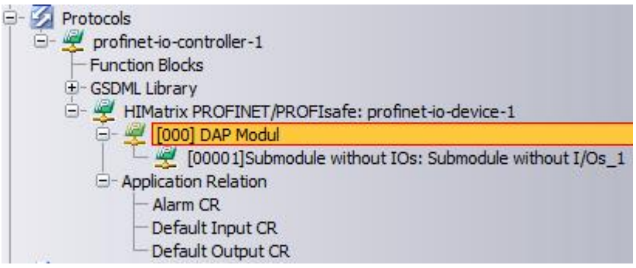


Figure 6: Device Access Point (DAP) for the PROFINET IO-Device

To configure the module slots

1. In the structure tree, open **Protocols**, **PROFINET IO-Device**.
2. Right-click **PROFINET IO-Device** and select **New** from the context menu to open the module list.
3. From the module list, select appropriate modules for the PROFINET IO-Device and click **Insert Module(s)** to confirm the action.

To number the PROFINET IO-Device modules

The **Device Access Point (DAP)** module is associated by default with slot 0. All other PROFINET IO-Device modules must be numbered.

1. Right-click **PROFINET IO-Device Module** and select **Properties** from the context menu.
2. In the **Slots** field, enter the device module slots as arranged in the actual PROFINET IO-Device.
3. Repeat these steps for any additional **PROFINET IO-Device modules**.

The **Model** and **Features** tabs display additional details of the GSDML file.

To configure the application relation

1. In the structure tree, open **PROFINET IO-Device, Application Relation.**
2. Right-click **Default Input CR** and select **Properties** from the context menu.
3. Adjust the reduction factor parameter, e.g., set it to 4.
4. Right-click **Default Output CR** and select **Properties** from the context menu.
5. Adjust the reduction factor parameter, e.g., set it to 4.

3.6.4 Configuring the Device Access Point (DAP) Module

To configure the Device Access Point (DAP) module

1. Select **[000] DAP Module, [xxxxx] DAP Submodule**.
2. Right-click [**xxxxx] DAP Submodule** and select **Edit** from the context menu.
3. If the Consumer/Provider status should not be controlled using a given user program logic, select the **System Variables** tab located in the **Edit** dialog box and assign a global variable with the initial value TRUE to the *Input Data Accepted by Controller* output variable.

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

Setting the header parameters, e.g., for a DAP with alarm settings.

To set the header parameters for the device access point (DAP) for the PROFINET IO-Device

1. In the structure tree, select **Protocols, PROFINET IO-Controller, PROFINET IO-Device, DAP Module, [xxxxx] DAP Submodule, Alarm Settings (Header): Parameters.**
2. Select **Properties** from the context menu.
3. Enter the header parameter name in the **Name** field.
4. Click the **Edit** button to open a dialog box for configuring the interfaces, diagnostics or alarms.

3.6.5 Configuring the PROFINET IO-Device Module

i The sum of the variables (in bytes), must be identical to the size of the module (in bytes).

To configure the PROFINET IO-Device module

1. Select **[001] PROFINET IO-Device Module, [xxxxx] PROFINET IO-Device Submodule**.
2. Right-click **[xxxxx] Submodule** and select **Edit** from the context menu.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Input Signals** area.
5. Right-click a free space in the **Inputs Signals** area and select **New Offsets** from the context menu to generate the new variable offsets.
6. If the Consumer/Provider status should not be controlled, select the **System Variables** tab located in the **Edit** dialog box and assign a global variable with the initial value TRUE to the output variables *Valid Output Data* and *Input Data Accepted by Controller*.

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

3.6.6 Configuring the PROFIsafe IO-Device Modules

i The sum of the variables (in bytes), must be identical to the size of the module (in bytes).

To configure the PROFIsafe IO-Device modules

1. Select **[001] PROFIsafe IO-Device Module, [xxxxx] PROFIsafe IO-Device Submodule.**
2. Right-click **[xxxxx] Submodule** and select **Edit** from the context menu.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Input Signals** area.
5. Right-click a free space in the **Inputs Signals** area and select **New Offsets** from the context menu to generate the new variable offsets.
6. If the Consumer/Provider status should not be controlled, select the **System Variables** tab located in the **Edit** dialog box and assign a global variable with the TRUE initial value to the output variables *Valid Output Data* and *Input Data Accepted by Controller*.

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

To configure the F parameters

1. Select **[001] PROFIsafe IO-Device Module, [xxxxx] PROFIsafe IO-Device Submodule, F Parameters.**
2. Right-click **F Parameters** and select **Properties** from the context menu.
3. Set the following parameters:
   - F_Dest_Add: Destination address of the device module
   - F_WD_Time: Watchdog time for the connection to this device module

To verify the PROFINET IO configuration

1. In the structure tree, open **Configuration, Resource, Protocols, PROFINET IO-Controller**.
2. Right-click **PROFINET IO-Controller** and select **Verification** from the context menu.
3. Thoroughly verify the messages displayed in the logbook and correct potential errors.

i Recompile the resource and load it into the controller to ensure that the new configuration can be used for communication with the PROFINET IO.

3.6.7 Identifying the PROFINET IO-Devices within the Network

To identify the PROFINET IO-Devices within the Ethernet network

1. Log-in to the communication module containing the **PROFINET IO-Controller**.
2. In the structure tree for to the online view, select **PROFINET IO-Controller, PROFINET IO Network Member**.
3. Select **Determine PROFINET IO Network Member** from the context menu.
   ☑ A list appears specifying all the PROFINET devices in the network of this PROFINET IO-Controller.

To configure the PROFINET IO-Device in SILworX

1. In the list, right-click the PROFINET IO-Device to be configured, to change the settings.
2. Name the device using the **Name the PROFINET IO-Device** context menu function.
   ☑ Make sure that the PROFINET IO-Device name complies with the project requirements. (Only lower case letters may be used!)
3. Use the **Network Settings** context menu function to set the IP address, subnet mask and gateway.

i The PROFINET IO-Device name and network settings must be configured in the PROFINET IO-Controller, or no communication is possible.

3.7 Menu Functions for the PROFINET IO-Controller

3.7.1 Example of Structure Tree for the PROFINET IO-Controller

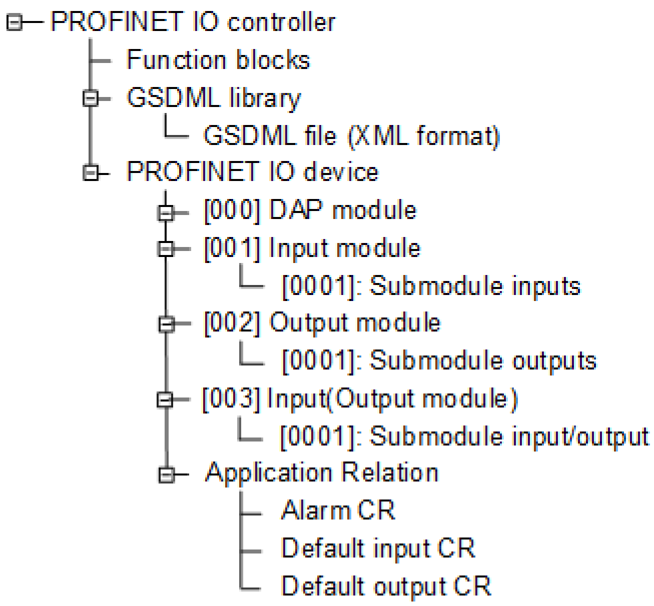


Figure 7: Structure Tree for the PROFINET IO-Controller

3.7.2 PROFINET IO-Controller

The **Properties** function of the context menu for the PROFINET IO-Controller is used to open the **Properties** dialog box. The dialog box contains the following tabs:

3.7.2.1 The PROFINET IO (Properties) Tab

Element	Description
Type	PROFINET IO-Controller
Name	Name of the PROFINET IO-Controller
Process Data Refresh Rate [ms]	Refresh rate in milliseconds at which the COM and CPU exchange protocol data. If the *Refresh Rate* is zero or less than the cycle time for the controller, data is exchanged as fast as possible. Range of values: 4…(2^{31}-1) Default value: 0
Force Process Data Consistency	Activated: Transfer of all of the protocol data from the CPU to the COM within a CPU cycle. Deactivated: Transfers all of the protocol data from the CPU to the COM, distributed over multiple CPU cycles, each with 1100 bytes per data direction. This may also cause the cycle time of the controller to be reduced. Default value: Activated
Module	Selection of the COM module within which the protocol is processed.
Activate Max. µP Budget	Activated: Use the µP budget limit from the *Max. µP Budget in [%]* field. Deactivated: Do not use the µP budget limit for this protocol.

Element	Description
Max. µP Budget in [%]	Maximum µP budget of the module that can be used for processing the protocols. Range of values: 1…100% Default value: 30%
RPC Port Server	Remote Procedure Call Port Range of values: 1024…65535 Default value: 49152 RPC port server and RPC port client must not be identical!
RPC Port Client	Remote Procedure Call Port Range of values: 1024…65535 Default value: 49153 RPC port server and RPC port client must not be identical!
F_Source_Add	Address of the controller (F-Host). Unique PROFIsafe controller/device addresses must be used in a PROFIsafe network. Also refer to *IEC 61784-3-3 V2.5c Chapter 9.7* for further details.

Table 5: The PROFINET IO-Device (Properties) Tab

3.7.3 PROFINET IO-Device (within the Controller)

The **Properties** function of the context menu for the PROFINET IO-Device is used to open the **Properties** dialog box, which contains the following tabs:

The Parameters (Properties) Tab

Element	Description
Name	Name of the PROFINET IO-Device
IP Address	IP address of the communication partner. Range of values: 0.0.0.0…255.255.255.255 Default value: 192.168.0.99 Do not use IP addresses already in use, refer to the chapter in the communication manual (HI 801 101 E) concerning the network ports and Ethernet communication for details.
Subnet Mask	Subnet mask for the addressed subnet containing the device. Range of values: 0.0.0.0…255.255.255.255 Default value: 255.255.255.0
GSDML File	GSDML stands for Generic Station Description Markup Language and refers to an XML-based description language. The GSDML file contains the PROFINET device master data.

Table 6: The Parameters (Properties) Tab

The Model and Features Tabs

The **Model** and **Features** tabs display additional details of the GSDML file, such as *Manufacturer Name, Device Description* or *Supported Factors*. This additional information is intended to support the users during the device configuration and cannot be changed.

3.7.4 DAP Module (Device Access Point Module)

Within a PROFINET IO-Device, a DAP module is always used for connecting to the bus. This DAP module should not be deleted.

The **Properties** function in the context menu for the DAP module is used to open the **Properties** dialog box, which contains the following tabs:

The Parameters (Properties) Tab

Element	Description
Name	Name for the DAP module
Slot	Not changeable Default value: 0

Table 7: The Parameters (Properties) Tab

The Model and Features Tabs

The **Model** and **Features** tabs display additional details of the GSDML file, such as *Module ID, Hardware/Software Version*. This additional information is intended to support the users during the device configuration and cannot be changed.

3.7.4.1 DAP-Submodule (Properties)

The **Properties** function in the context menu for the DAP submodule is used to open the **Properties** dialog box, which contains the following tabs:

The Parameters Tab

Element	Description
Name	Name of the input submodule
Subslot	Default value: 1
IO Data CR, Inputs	Selection of the communication relation (CR) to which the submodule inputs should be transferred. - None - Default Input CR
Input Data Accepted by Controller	Selection of the communication relation (CR) to which the submodule IO consumer status (CS) should be transferred. - None - Default Output CR

Table 8: The Parameters Tab

The Model and Features Tabs

The **Model** and **Features** tabs display additional details of the GSDML file, such as *Submodule ID* or *Data Length.* This additional information is intended to support the users during the device configuration and cannot be changed.

3.7.4.2 DAP Submodule (Edit)

The **Edit** function of the context menu for input submodules is used to open the **Edit** dialog box, which contains the following tabs:

The System Variables Tab

The **System Variables** tab provides the following system variables, which allow the PROFINET IO submodule state to be evaluated and controlled in the user program.

Element	Description	
Valid Input Data	TRUE	Valid input data GOOD
	FALSE	Invalid input data BAD
Input Data Accepted by Controller	TRUE	Valid input data GOOD
	FALSE	Invalid input data BAD

Table 9: The System Variables Tab

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

3.7.4.3 Header Parameters

Some devices contain so-called header parameters used to activate/deactivate parameters, such as Diagnosis, Alarm and Interfaces.

3.7.5 Input/Output PROFINET IO Modules

A PROFINET IO module can have multiple submodules. HIMax PROFINET IO-Controllers have one submodule in each input and output PROFINET IO module.

The PROFINET IO input modules are used to enter the HIMax PROFINET IO-Controller input variables that are sent by the PROFINET IO-Device.

The PROFINET IO output modules are used to enter the HIMax PROFINET IO-Controller output variables that are sent to the PROFINET IO-Device.

To create the required PROFINET IO module

1. In the structure tree, open **Configuration, Resource, Protocols, PROFINET IO-Device**.
2. Right-click **PROFINET IO-Device** and select **New** from the context menu.
3. Select the modules required.

The **Properties** function of the context menu for the input/output PROFINET IO modules is used to open the **Properties** dialog box, which contains the following tabs:

3.7.5.1 The Parameters Tab

Element	Description
Name	Name of the I/O PROFINET IO module.
Slot	0…32767 Default value: 1

Table 10: The Parameters Tab for the I/O PROFINET IO Module

The Model and Features Tabs

The **Model** and **Features** tabs display additional details of the GSDML file, such as *Module ID, Hardware/Software Version*. This additional information is intended to support the users during the device configuration and cannot be changed.

3.7.6 Submodule Input

The submodule parameters are used to define the communication relation of the module and its behavior after connection is interrupted.

3.7.6.1 Submodule Input (Properties)

The **Properties** function in the context menu for the *Input Submodules* is used to open the **Properties** dialog box, which contains the following tabs:

The Parameters Tab

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>Name</td><td>Name of the input submodule.</td></tr>
<tr><td>Subslot</td><td>Not changeable for HIMax PROFINET IO-Controller.
Default value: 1</td></tr>
<tr><td>IO Data CR, Inputs</td><td>Selection of the communication relation (CR) to which the submodule inputs should be transferred.
- None
- Default input CR</td></tr>
<tr><td>Input Data Accepted by Controller</td><td>Selection of the communication relation (CR) to which the submodule IO consumer status (CS) should be transferred.
- None
- Default output CR</td></tr>
<tr><td>Shared Input</td><td>Not applicable</td></tr>
<tr><td>Input Values if IO CR is Disconnected</td><td>Behavior of the input variables for this PROFINET IO submodule after the connection is interrupted.<table><tr><td>Retain Last Value</td><td>The input variables are frozen to the current value and used until a new connection is established.</td></tr><tr><td>Adopt Initial Values</td><td>The initial values are used for the input variables.</td></tr></table></td></tr>
</table>

Table 11: The Parameters Tab

The Model and Features Tabs

The **Model** and **Features** tabs display additional details of the GSDML, file such as *Submodule ID, Hardware/Software Version* or *Data Length.* This additional information is intended to support the users during the device configuration and cannot be changed.

3.7.6.2 Input Submodule (Edit)

The **Edit** function of the context menu for the input submodules is used to open the **Edit** dialog box. The dialog box contains the following tabs:

The System Variables Tab

The **System Variables** tab provides the following system variables, which allow the PROFINET IO submodule state to be evaluated in the user program.

Element	Description
Valid Input Data	TRUE Valid Input Data GOOD FALSE Invalid input data BAD
Input Data Accepted by Controller	TRUE Valid Input Data GOOD FALSE Invalid input data BAD
The following parameters are only available for PROFIsafe modules:	
Valid Output Data	TRUE Valid Output Data GOOD FALSE Invalid output data BAD
Output Data Accepted by Device	TRUE Valid Output Data GOOD FALSE Invalid output data BAD

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

Element	Description
PROFIsafe Control	With each message, PROFIsafe sends the PROFIsafe control byte from the controller to the device that can be set in the user program. See also Chapter 3.3.2. Bit 0 iPar_EN_C: To load new iParameters into the F-Device, the iPar_EN_C must be set to TRUE to unlock the F-Device. As long as iPar_EN_C is TRUE, failsafe values of 0 are exchanged between F-Host and F-Device. Bit 1 OA_C: Operator acknowledgment. After a PROFIsafe error (e.g., CRC error or timeout), bit must be set to TRUE for at least a PROFIsafe cycle. If a PROFIsafe connection should be (re-)started, the operator acknowledgment may only be sent if no dangerous states exist. Bit 2 Reserved Bit 3 Reserved Bit 4 Activate_FV_C: FALSE: Process values are exchanged between F-Host and F-Device. TRUE: Failsafe values of "0" are exchanged between F-Host and F-Device. Bit 5 Reserved Bit 6 Bit 7 Reset_Comm: PROFIsafe communication reset; the protocol stack is set to the initial state. The bit must be set to TRUE until the PROFIsafe status *Bit 2 Reset_Comm* has read back the TRUE value.
PROFIsafe RoundTrip Time last	For an F-Host, this is the time between a data message transmission (with consecutive number N) and the reception of the corresponding acknowledgment (with consecutive number N), measured in milliseconds.

Element	Description
PROFIsafe Status	Each message received by PROFIsafe on the host contains the PROFIsafe status byte, which can be evaluated in the user program. Bit 0 iPar_OK_S TRUE: New iParameters received FALSE: No change Bit 1 OA_Req_S Operator Acknowledge Requested. Bit 2 Reset_Comm is the Reset_Comm value read back from the host control byte. This bit indicates whether Reset_Comm has arrived. Bit 3 FV_activated_S Bit 4 Toggle_h Bit 5 Device_Fault TRUE: The F-Device reported a device fault. FALSE: The F-Device did not report any device faults. Bit 6 WD_timeout TRUE: Either the F-Device reported a watchdog timeout or the host timeout occurred on the F-Host. FALSE: No timeout occurred on the F-Device or on the F-Host. Bit 7 CRC TRUE: Either the F-Device reported a CRC error or a CRC error occurred on the F-Host. FALSE: No CRC fault occurred on the F-Device or on the F-Host.

Table 12: The System Variables Tab

The **Process Variables** tab is used to enter the input variables.

3.7.6.3 F-Parameters of Submodule Input (For PROFIsafe Modules Only)

To exchange process data safely, PROFIsafe F-Devices need normalized F-Parameters. The F-Device only establishes communication if valid F-Parameters were configured. Grayed-out parameters are disabled and, to some extent, preset by the GSDML file or automatically calculated.

Element	Description
Name	Module name.
Index	Module index.
F_Par_Version	Only V2 mode is supported. V1 mode is rejected. Determined through the GSDML file.
F_Source_Add	The F-source address of the F-Host must be unique within the PROFIsafe network! Range of values: 1…65534
F_Dest_Add	The F-destination address of the F-Device must be unique within the PROFIsafe network! Range of values: 1…65534
F_WD_Time	Watchdog time. Range of values: 1…65534 ms
F_iPar_CRC	The F_iPar_CRC of the F-Device is entered in this field.
F_SIL	The SIL is displayed in this field: 0 - SIL1 1 - SIL2 2 - SIL3 3 - NoSIL Determined through the GSDML file.
F_Check_iPar	The CRC iParameter is displayed in this field. Determined through the GSDML file.
F_Block_ID	Structure of the F-Parameters Determined through the GSDML file.
F_CRC_Length	Indicates if the 3-byte CRC or 4-byte CRC is used. Determined through the GSDML file.
F_Par_CRC	The CRC F-parameter (CRC1) is displayed in this field. Calculated based on the current F-Parameters.

Table 13: F-Parameters for Submodule Input (Properties)

3.7.7 Submodule Output

The submodule parameters are used to define the communication relation of the module and its behavior after connection is interrupted.

3.7.7.1 Submodule Output (Properties)

The **Properties** function on the context menu for the *Output Submodules* is used to open the **Properties** dialog box, which contains the following parameters:

The Parameters Tab

Element	Description
Name	Name of the output submodule
Subslot	Not changeable for HIMax PROFINET IO-Controller. Default value: 1
IO Data CR, Outputs	Selection of the communication relation (CR) to which the submodule outputs should be transferred. - None - Default Input CR
Output Data Accepted by Device	Selection of the communication relation (CR) to which the submodule IO consumer status (CS) should be transferred. - None - Default Output CR

Table 14: The Parameters Tab

The Model and Features Tabs

The **Model** and **Features** tabs display additional details of the GSDML file, such as *Submodule ID, Hardware/Software Version* or *Data Length.* This additional information is intended to support the users during the device configuration and cannot be changed.

3.7.7.2 Submodule Output (Edit)

The **Edit** function of the context menu for Output Submodule is used to open the **Edit** dialog box, which contains the following tabs:

The System Variables Tab

The **System Variables** tab provides the following system variables, which allow the PROFINET IO submodule state to be evaluated in the user program.

Element	Description	
Valid Output Data	TRUE	Valid Output Data GOOD
	FALSE	Invalid output data BAD
Output Data Accepted by Device	TRUE	Valid Output Data GOOD
	FALSE	Invalid output data BAD
The following parameters are only available for PROFIsafe modules:		
Valid Input Data	TRUE	Valid Input Data GOOD
	FALSE	Invalid input data BAD
Input Data Accepted by Controller	TRUE	Valid Input Data GOOD
	FALSE	Invalid input data BAD
For further PROFIsafe module parameters, see Table 12.		

Table 15: The System Variables Tab

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

The **Process Variables** tab is used to enter the output variables.

3.7.7.3 F-Parameters of Submodule Output (For PROFIsafe Modules Only)

For a description of the F-Parameters, refer to Chapter 3.7.6.3.

3.7.8 Submodule Inputs and Outputs

The submodule parameters are used to define the communication relation of the module and its behavior after connection is interrupted.

3.7.8.1 Submodule Inputs and Outputs (Properties)

The **Properties** function on the context menu for the *Input and Output Submodules* is used to open the **Properties** dialog box, which contains the following parameters:

The Parameters Tab

Element	Description
Name	Name of the input/output submodule
Subslot	Not changeable for HIMax PROFINET IO-Controller. Default value: 1
IO Data CR, Inputs	Selection of the communication relation (CR) to which the submodule inputs should be transferred. - None - Default Input CR
IO Data CR, Outputs	Selection of the communication relation (CR) to which the submodule outputs should be transferred. - None - Default Output CR
Input Data Accepted by Controller	Selection of the communication relation (CR) to which the submodule IO consumer status (CS) should be transferred. - None - Default Output CR
Output Data Accepted by Device	Selection of the communication relation (CR) to which the submodule IO consumer status (CS) should be transferred. - None - Default Input CR
Input Values if IO CR is Disconnected	- Retain Last Value - Adopt Initial Values

Table 16: The Parameters Tab

The Model and Features Tabs

The **Model** and **Features** tabs display additional details of the GSDML file, such as *Submodule ID, Hardware/Software Version* or *Data Length.* This additional information is intended to support the users during the device configuration and cannot be changed.

3.7.8.2 Submodule Inputs and Outputs (Edit)

The **Edit** function of the context menu for Input/Output Submodules is used to open the **Edit** dialog box, which contains the following tabs:

The System Variables Tab

The **System Variables** tab provides the following system variables, which allow the PROFINET IO submodule state to be evaluated in the user program.

Element	Description	
Valid Output Data	TRUE	Valid Output Data GOOD
	FALSE	Invalid output data BAD
Output Data Accepted by Device	TRUE	Valid Output Data GOOD
	FALSE	Invalid output data BAD
Valid Input Data	TRUE	Valid Input Data GOOD
	FALSE	Invalid input data BAD
Input Data Accepted by Controller	TRUE	Valid Input Data GOOD
	FALSE	Invalid input data BAD
For PROFIsafe module parameters, see Table 12.		

Table 17: The System Variables Tab

The **Process Variables** tab is used to enter the input and output variables in their corresponding area.

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

3.7.8.3 Submodule Input/Output F-Parameters (For PROFIsafe Modules Only)

For a description of the F-Parameters, refer to Chapter 3.7.6.3.

3.7.9 Application Relation (Properties)

An application relation (AR) is a logic construct for enabling data exchange between controller and device. In the following example (see Figure 8), data is transferred within the application relation via standard communication relations (CR): alarm CR, default input CR and default output CR. These communication relations are already configured by default in the input and output modules.

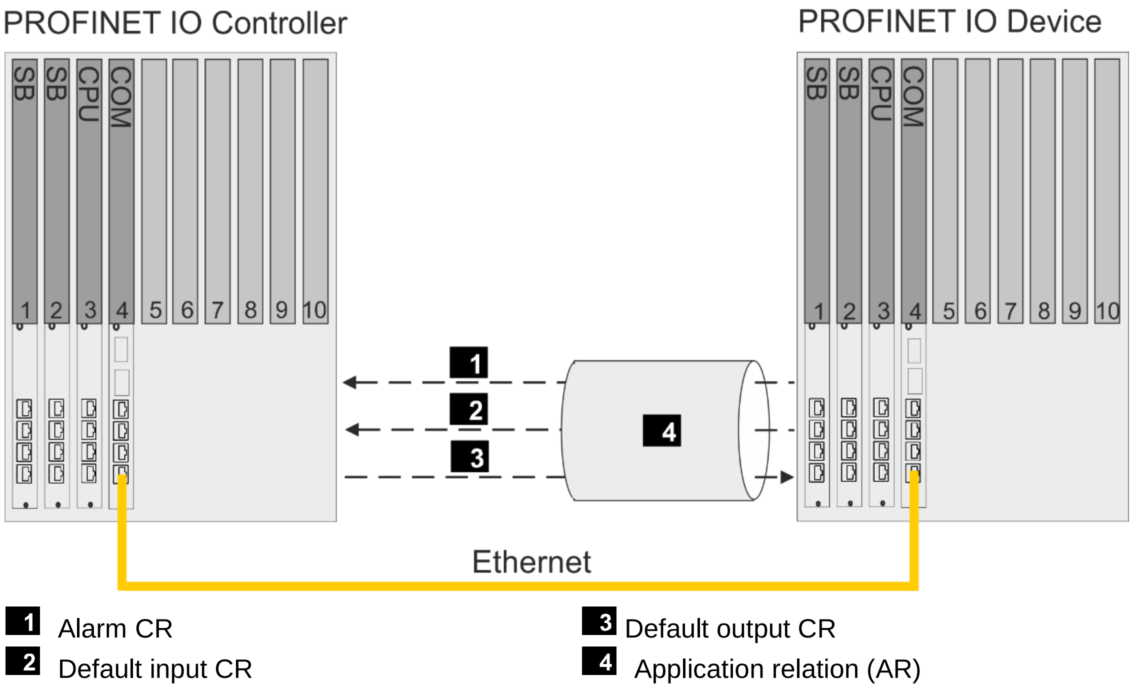


Figure 8: Communication via PROFINET IO/PROFIsafe

The **Properties** function of the context menu for *Application Relation* opens the **Properties** dialog box.

Element	Description
Name	Not changeable
AR UUID	Unique code used to identify the application relation (AR). Not changeable
Connection Establishment Timeout Factor	This parameter is used to calculate the maximum time that the PROFINET IO-Device allows the connection to take during the establishment of a connection between sending the response to the connect request and receiving a new request from the PROFINET IO-Controller. Range of values: 1…1000 (x 100 ms) Default value: 600
Supervisor may adopt the AR	Defines whether a PROFINET IO supervisor may adopt the application relation (AR). 0 Not allowed 1 Allowed Default value: 0

Table 18: Application Relation (Properties)

3.7.10 Alarm CR (Properties)

Several communication relations (CRs) can be established within an application relation.

The alarm CR is used by a PROFINET IO-Device to transmit alarms to the PROFINET IO-Controller.

The **Properties** function of the context menu for application relation opens the **Properties** dialog box, which contains the following tabs:

Element	Description
Name	Not changeable
VLAN ID, High Priority Alarms	Each virtual LAN (VLAN) is assigned a unique number to ensure separation. A device in the VLAN with ID=1 can communicate with any other device in the same VLAN, but not with a device in another VLAN (e.g., ID=2, 3, …). For the range of values, also refer to IEC 61158-6: 0x000 No VLAN 0x001 Standard VLAN 0x002…0xFFF See IEEE 802.1 Q Default value: 0
VLAN ID, Low Priority Alarms	Description, see VLAN ID, High Priority Alarms Default value: 0
Alarm Priority	Use User Priority: The priority assigned by the user is used. Ignore User Priority: The priority assigned by the user is ignored. The generated alarm has low priority.
Alarm Resends	Maximum number of device resends if the controller does not respond. Range of values: 3…15 Default value: 10
Alarm Timeout Factor	The RTA timeout factor is used to calculate the maximum device time that may elapse after sending a RTA data (alarm) frame and receiving the RTA ack frame. RTA timeout = RTA timeout factor x 100 ms Range of values: 1…65535 Default value: 5

Table 19: Alarm CR (Properties)

3.7.11 Input CR (Properties)

The input CR is used by a PROFINET IO-Device to transmit variables to the PROFINET IO-Controller.

The **Properties** function of the context menu for the default input CR opens the **Properties** dialog box. The dialog box contains the following parameters:

Element	Description
Name	Any unique name for an input CR. The default input CR cannot be changed.
Type	1 (not changeable)
Send Clock Factor	The send clock factor defines the send clock for the cyclic data transfer of an IO CR. **Send Clock = Send Clock Factor x 31.25 µs** Range of values: 1…128 Default value: 32
Reduction Factor	The reduction factor allows the actual cycle time needed for sending the data of an IO CR to be reduced to send clock. The actual data cycle time is calculated as follows: **Send Cycle = Reduction Factor x Send Clock** Range of values: 1…16384 Default value: 32 (depending on the device)
Watchdog Factor	From the perspective of an IO CR consumer, the watchdog factor is used to calculate the maximum time allowed between the reception of 2 frames: **Watchdog Time = Watchdog Factor x Send Cycle** Range of values: 1…7680 Default value: 3
VLAN ID	Each virtual LAN (VLAN) is assigned a unique number to ensure separation. A device in the VLAN with ID=1 can communicate with any other device in the same VLAN, but not with a device in another VLAN (e.g., ID=2, 3, ...). For the range of values, also refer to IEC 61158-6: 0x000 No VLAN 0x001 Standard VLAN 0x002…0xFFF See IEEE 802.1 Q Default value: 0

Table 20: Input CR (Properties)

3.7.11.1 Input CR (Edit)

The **Edit** function of the context menu for the default input CR opens the **System Variables** dialog box, and contains the following system variables:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>Data Status Input CR</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>State
With redundant connections, primary describes the leading channel.
1 = Primary
0 = Backup
With mono connections:
1 = Connected
0 = Not connected</td></tr>
<tr><td>1</td><td>Not used</td></tr>
<tr><td>2</td><td>Data Valid
Invalid is set during the start-up phase or if the application is not able to report errors via IOPS
1 = Valid
0 = Invalid</td></tr>
<tr><td>3</td><td>Not used</td></tr>
<tr><td>4</td><td>Process State
This has only an informational character and the actual data validity is reported via IOPS.
1 = Run
0 = Stop</td></tr>
<tr><td>5</td><td>Problem Indicator
Details on the diagnostic data of the alarm CR are provided when a problem is detected.
1 = Regular operation
0 = Problem detected</td></tr>
<tr><td>6</td><td>Not used</td></tr>
<tr><td>7</td><td>Not used</td></tr>
</table>
</td></tr>
</table>

Table 21: Input CR (Edit)

3.7.11.2 Output CR (Properties)

Several communication relations (CRs) can be established within an application relation.

The output CR is used by the PROFINET IO-Controller to transmit variables to the PROFINET IO-Device.

The **Properties** function of the context menu for the output CR opens the **Properties** dialog box, which contains the following parameters:

Element	Description
Name	Any unique name for an output CR. The default output CR cannot be changed.
Type	2 (not changeable)
Send Clock Factor	The send clock factor defines the send clock for the cyclic data transfer of an IO CR. **Send Clock = Send Clock Factor x 31.25 µs** Range of values: 1…128 Default value: 32
Reduction Factor	For setting the rate of transmission. The reduction factor allows the actual cycle time needed for sending the data of an IO CR to be reduced. The actual data cycle time is calculated as follows: **Send Cycle = Reduction Factor x Send Clock** Range of values: 1…16384 Default value: 32
Watchdog Factor	From the perspective of an IO CR consumer, the watchdog factor is used to calculate the maximum time allowed between the reception of 2 frames: **Watchdog Time = Watchdog Factor x Send Cycle** Range of values: 1…7680 Default value: 3
VLAN ID	Each virtual LAN (VLAN) is assigned a unique number to ensure separation. A device in the VLAN with ID=1 can communicate with any other device in the same VLAN, but not with a device in another VLAN (e.g., ID=2, 3, …). For the range of values, also refer to IEC 61158-6: 0x000 No VLAN 0x001 Standard VLAN 0x002…0xFFF See IEEE 802.1 Q Default value: 0

Table 22: Output CR (Properties)

3.8 PROFINET IO-Device

This chapter describes the characteristics of the PROFINET IO-Device and the menu functions and dialog boxes required to configure the PROFINET IO-Controller in SILworX.

3.8.1 System Requirements

Equipment and system requirements

Element	Description
Controller	HIMax with COM module. HIMatrix (Not applicable to HIMatrix F*02).
CPU module	The Ethernet interfaces on the processor module may not be used for PROFINET IO.
COM module	Ethernet 10/100BaseT
Activation	Software activation code is required for activation, refer to the communication manual (HI 801 101 E) for details.

Table 23: System Requirements and Equipment for the PROFINET IO-Controller

3.8.2 PROFINET IO-Device Properties

Element	Description
Safety-related	PROFINET: No PROFIsafe: Yes
Transfer rate	100 Mbit/s full duplex
Transport path	Ethernet interfaces on the COM module The Ethernet interfaces in use can simultaneously be employed for other protocols.
Conformance class	The PROFINET IO-Device meets the requirements for conformance class A.
Real time class	RT Class 1
Max. number of PROFINET IO-Devices	One PROFINET IO-Device can be configured for each COM module.
Max. number of application relations (ARs) to the PROFINET IO-Controller	A PROFINET IO-Device can establish a maximum of 5 application relations (ARs) to a PROFINET IO-Controller.
Max. number of communication relations (CRs for each AR)	Standard: 1 input, 1 output, 1 alarm
Max. process data length of all configured PROFINET IO modules.	Output: max. 1440 bytes Input: max. 1440 bytes
Data priorization	Possible at device level using the *Reduction Rate* setting.
The following features apply to PROFIsafe	
Max. number of F-Devices on each COM module (HIMax and HIMatrix)	63
Max. process data length of a CR	Output: max. 123 bytes user data + 5 bytes[1)] Input: max. 123 bytes user data + 5 bytes[1)]
Max. user data size HIMax: HIMatrix	 1024 x 123 bytes = 125 952 bytes 512 x 123 bytes = 62 976 bytes
[1)] 5 bytes management data (status/control bytes and CRC)	

Table 24: PROFINET IO-Controller Properties

3.9 PROFINET IO/PROFIsafe Example

The following chapter describes how to configure the PROFINET IO-Divice and the PROFIsafe device.

3.9.1 Configuring the PROFINET IO-Device in SILworX

To create a new PROFINET IO-Device

1. In the structure tree, open **Configuration, Resource, Protocols.**
2. Right-click **Protocols** and select **New, PROFINET IO-Device** from the context menu to add a new PROFINET IO-Device.
3. Right-click **PROFINET IO-Device** and select **Properties** from the context menu.
4. Enter the PROFINET IO-Device name in the **Name** field.
5. Select **COM Module**.

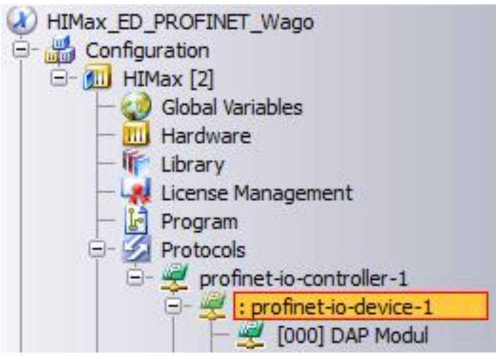


Figure 9: HIMax Structure Tree for the PROFINET IO-Device

To create the required PROFINET IO modules

1. In the structure tree, open **Configuration, Resource, Protocols, PROFINET IO-Device**.
2. Right-click **PROFINET IO-Device** and select **New** from the context menu.
3. For this example, select the following modules.

PROFINET IO/PROFIsafe module	Slot
In 1 Byte	1
Safe Out 1 Byte	2

To number the PROFINET IO-Device modules

1. Right-click the first **PROFINET IO-Device Module**, and then select **Properties** from the context menu.
2. Enter **1** in the **Slot** field.
3. Repeat these steps for any additional **PROFINET IO-Device modules** and number the modules consecutively.

i **Number** the modules as arranged in the actual PROFINET IO-Device.

The following step is only required for PROFIsafe modules!

4. Enter the PROFIsafe module address in the *PROFIsafe F_Destination_Address* field.

3.9.2 Configuring the PROFINET IO-Device Input Module

i The sum of the variables (in bytes), must be identical to the size of the module (in bytes).

To configure the input module [01] In 1 Byte

1. In the PROFINET IO-Device, select the input module **[01] In 1 Byte**.
2. Right-click **[01] In 1 Byte** and select **Edit** from the context menu.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Input Signals** area.
5. Right-click a free space in the **Input Signals** area to open the context menu.
6. Click **New Offsets** to generate the variable offsets.
7. If the Consumer/Provider status should not be controlled, select the **System Variables** tab located in the **Edit** dialog box and assign a global variable with the initial value TRUE to the output variable *Input Data Accepted by Device*.

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

3.9.3 Configuring the PROFIsafe Device Output Module

To configure the output module [02] Out 1 Byte

1. In the PROFINET IO-Device, select the output module **[02] Safe Out 1 Byte**.
2. Right-click **[02] Safe Out 1 Byte** and select **Edit** from the context menu.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Output Signals** area.
5. Right-click a free space in the **Output Signals** area to open the context menu.
6. Click **New Offsets** to generate the variable offsets.
7. If the Consumer/Provider status should not be controlled, select the **System Variables** tab located in the **Edit** dialog box and assign a global variable with the initial value TRUE to the output variables *Valid Output Variable* and *Input Data Accepted by Device*.

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

3.9.4 Verifying the PROFINET IO-Device Configuration

To verify the PROFINET IO-Device configuration

1. In the structure tree, select **Configuration, Resource, Protocols, PROFINET IO-Device.**
2. Right-click and select **Verification** from the context menu.
3. Thoroughly verify the messages displayed in the logbook and correct potential errors.

i Recompile the configuration of the PROFINET IO-Device together with the user program of the PROFINET IO-Device resource and transfer it to the controllers. The new configuration can only be used for communication with the PROFINET IO upon completion of this step.

3.10 Menu Functions for PROFINET IO-Device

3.10.1 The Properties Menu Function

The **Properties** function in the context menu for the PROFINET IO-Device is used to open the **Properties** dialog box.

Element	Description
Type	PROFINET IO-Device
Name	Any unique name for a PROFINET IO-Device.
Process Data Refresh Rate [ms]	Refresh rate in milliseconds at which the COM and CPU exchange protocol data. If the refresh rate is zero or lower than the cycle time for the controller, data is exchanged as fast as possible. Range of values: 4…(2^{31}-1) Default value: 0
Force Process Data Consistency	Activated: Transfer of all of the protocol data from the CPU to the COM within a CPU cycle. Deactivated: Transfers all of the protocol data from the CPU to the COM, distributed over multiple CPU cycles, each with 1100 bytes per data direction. This may also cause the cycle time of the controller to be reduced. Default value: Activated
Module	Selection of the COM module within which the protocol is processed.
Activate Max. µP Budget	Activated: Use the µP budget limit from the *Max. µP Budget in [%]* field. Deactivated: Do not use the µP budget limit for this protocol.
Max. µP Budget in [%]	Maximum µP budget of the module that can be used for processing the protocols. Range of values: 1…100% Default value: 30%
RPC Port Server	Remote Procedure Call Port Range of values: 1024…65535 Default value: 49152 RPC port server and RPC port client must not be identical!
RPC Port Client	Remote Procedure Call Port Range of values: 1024…65535 Default value: 49153 RPC port server and RPC port client must not be identical!
AutoAwait-FParams-OnConnLoss	**This parameter is only used for PROFIsafe modules.** Whenever the connection to the F-Host is lost, the F-Parameters must be reloaded into the F-Device. This parameter can be activated to simplify the PROFIsafe commissioning. Afterwards, the F-Device automatically sets the required F-Parameters at restart or after a connection loss. **After commissioning, it is essential that this parameter is deactivated to ensure a PROFIsafe-compliant behavior.** Activated: The F-Device automatically enters the *Waiting for F-Parameters* state. Deactivated: The user must send the online command to allow the F-Device to enter the *Wait for F-Parameters* state. Default value: Deactivated

Table 25: General Properties for the PROFINET IO-Device

3.10.2 PROFINET IO Modules

The following PROFINET IO modules are available in the PROFINET IO-Device.

PROFINET IO module	Max. size for the input variables	Max. size of the output variables
In 1 Byte	1 byte	
In 2 Bytes	2 bytes	
In 4 Bytes	4 bytes	
In 8 Bytes	8 bytes	
In 16 Bytes	16 bytes	
In 32 Bytes	32 bytes	
In 64 Bytes	64 bytes	
In 128 Bytes	128 bytes	
In 256 Bytes	256 bytes	
In 512 Bytes	512 bytes	
In 1024 Bytes	1024 bytes	
In-Out 1 Byte	1 byte	1 byte
In-Out 2 Bytes	2 bytes	2 bytes
In-Out 4 Bytes	4 bytes	4 bytes
In-Out 8 Bytes	8 bytes	8 bytes
In-Out 16 Bytes	16 bytes	16 bytes
In-Out 32 Bytes	32 bytes	32 bytes
In-Out 64 Bytes	64 bytes	64 bytes
In-Out 128 Bytes	128 bytes	128 bytes
In-Out 256 Bytes	256 bytes	256 bytes
In-Out 512 Bytes	512 bytes	512 bytes
Out 1 Byte		1 byte
Out 2 Bytes		2 bytes
Out 4 Bytes		4 bytes
Out 8 Bytes		8 bytes
Out 16 Bytes		16 bytes
Out 32 Bytes		32 bytes
Out 64 Bytes		64 bytes
Out 128 Bytes		128 bytes
Out 256 Bytes		256 bytes
Out 512 Bytes		512 bytes
Out 1024 Bytes		1024 bytes

Table 26: PROFINET IO Modules

3.10.3 HIMA PROFIsafe Modules

The following PROFIsafe modules are available in the PROFINET IO-Device:

PROFIsafe module	Max. size for the input variables	Max. size of the output variables
Safe In 1 Byte	1 byte	
Safe In 2 Bytes	2 bytes	
Safe In 4 Bytes	4 bytes	
Safe In 8 Bytes	8 bytes	
Safe In 16 Bytes	16 bytes	
Safe In 32 Bytes	32 bytes	
Safe In 64 Bytes	64 bytes	
Safe In 123 Bytes	123 bytes	
Safe In-Out 1 Byte	1 byte	1 byte
Safe In-Out 2 Bytes	2 bytes	2 bytes
Safe In-Out 4 Bytes	4 bytes	4 bytes
Safe In-Out 8 Bytes	8 bytes	8 bytes
Safe In-Out 16 Bytes	16 bytes	16 bytes
Safe In-Out 32 Bytes	32 bytes	32 bytes
Safe In-Out 64 Bytes	64 bytes	64 bytes
Safe In-Out 123 Bytes	123 bytes	123 bytes
Safe Out 1 Byte		1 byte
Safe Out 2 Bytes		2 bytes
Safe Out 4 Bytes		4 bytes
Safe Out 8 Bytes		8 bytes
Safe Out 16 Bytes		16 bytes
Safe Out 32 Bytes		32 bytes
Safe Out 64 Bytes		64 bytes
Safe Out 123 Bytes		123 bytes

Table 27: PROFIsafe Modules

To create a PROFINET or PROFIsafe module

1. In the structure tree, open **Configuration, Resource, Protocols, PROFINET IO-Device**.
2. Right-click PROFINET IO-Device and select **Insert Modules** from the context menu.
3. Select an appropriate module to transport the required process data.
4. Right-click the module selected, and then click **Edit**.
   - Enter the input and/or output variables in the **Process Variables** tab.
   - If the Consumer/Provider status should not be controlled, select the the **System Variables** tab and assign a global variable with the initial value TRUE to the output variables *Valid Output Variable* and *Input Data Accepted by Controller*.

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

The following setting is only required for PROFIsafe modules.

- Enter the *slot* and the *F_Destination_Address* in the **Properties** tab.

3.10.4 PROFINET IO and PROFIsafe Module

The module parameters are used to define the communication relations of the module and its behavior after connection is interrupted.

Properties

The **Properties** function of the context menu for the modules opens the **Properties** dialog box. The dialog box contains the following tabs:

Element	Description
Name	Name of the device module.
Slot	0…32767
Module ID	Unique number
Subslot	Number of sub-slots
Process data behavior	Process data value after the connection is interrupted - Retain last valid process data - Apply initial data
Length of IO input data	1…123
Length of IO output data	1…123
PROFIsafe F_Destination_Address	The F-destination address of the F-Device must be unique within the PROFIsafe network! Range of values: 1…65534

Table 28: General Properties for the Device Module

Edit

The **Edit** function of the context menu for the submodule is used to open the **Edit** dialog box.

The **System Variables** tab provides the following system variables, which allow the submodule state to be evaluated in the user program.

i These system variables can be used to control the Consumer/Provider status. Refer to Chapter 3.2 for details.

Element	Description	
Valid Output Data	TRUE	Valid output data GOOD
	FALSE	Invalid output data BAD
Output Data Accepted by Device	TRUE	valid output data GOOD
	FALSE	Invalid output data BAD
Valid Input Data	TRUE	Valid input data GOOD
	FALSE	Invalid input data BAD
Input Data Accepted by Device	TRUE	Valid input data GOOD
	FALSE	Invalid input data BAD

The following variables are only used for PROFIsafe modules:	
PROFIsafe Control	The PROFIsafe control byte sent by the controller read within the device, see also Chapter 3.3.2. Bit 0 iPar_EN_DC An enable from the controller releases the device to load new iParameters into the device. Bit 1 OA_Req_DC Operator acknowledgment from the host control byte. Bit 2 Reset_Comm is the Reset_Comm value read back from the host control byte. Bit 3 Activate_FV_DC FALSE: Process values are exchanged between F-Host and F-Device. TRUE: Failsafe values of "0" are exchanged between F-Host and F-Device. Bit 4 Toggle_d Toggle bit of the F-Device Bit 5 Cons_nr_R The consecutive number is adopted whenever there is a change of the Toggle_d bit between 2 consecutive control bytes, i.e., this does not depend on the occurrence of a fault. Bit 6 F_ParamValid TRUE: F parameters were set FALSE: Otherwise Bit 7 F_Param_ConfiguredTwice TRUE: The F-Device was configured more than once with different F-Parameters. FALSE: Otherwise
PROFIsafe F_iPar_CRC	iParameters are independent or technology-specific F-Device parameters. The iPar_CRC results from the configuration of the F-Device. i The users are responsible for setting the valid iPar_CRC after configuring the iParameters and moving to hot operation.
PROFIsafe F_SIL	0 SIL1 1 SIL2 2 SIL3 3 No SIL
PROFIsafe RoundTrip Time last	PROFIsafe must determine the RoundTripTimeLast on an F-Host. For an F-Host, this is the time between a data message transmission (with consecutive number N) and the reception of the corresponding acknowledgment (with consecutive number N), measured in milliseconds.

PROFIsafe Status	With each message, PROFIsafe sends the PROFIsafe status byte that can be set in the device user program from the device to the controller. See also Chapter 3.3.2. Bit 0 iPar_OK_DS New iParameters received Bit 1 Device_Fault_DS TRUE: Device fault FALSE: No device fault It is only taken into account from the *21* Await Message PROFIsafe state. Bit 2 Reserved Bit 3 Reserved Bit 4 FV_activated_DS Failsafe value activated Bit 5 Reserved Bit 6 Bit 7 Reset_Comm Protocol stack is reset to its initial state.

Table 29: Edit Dialog Box for Submodule

The **Process Variables** tab is used to enter the input variables.

4 PROFIBUS DP

PROFIBUS DP is an international, open fieldbus standard that is used when a fast response time is required for small amounts of data.

The HIMA PROFIBUS DP master and the HIMA PROFIBUS DP slave meet the criteria specified in the European EN 50170 standard [7] and the internationally binding standard IEC 61158 for PROFIBUS DP.

The HIMA PROFIBUS DP master can exchange data with the PROFIBUS DP slaves cyclically and acyclically.

Different function blocks are available in SILworX to acyclically exchange data. These function blocks are used to tailor the HIMA PROFIBUS DP master and the PROFIBUS DP slaves to best meet the project requirements.

A redundant PROFIBUS DP connection can only be implemented by configuring a second PROFIBUS DP master/slave and adjusting it in the user program.

- PROFIBUS DP master, see also Chapter 4.1.
- PROFIBUS DP slave, see also Chapter 4.13.

4.1 HIMA PROFIBUS DP Master

This section describes the characteristics of the HIMA PROFIBUS DP master and the menu functions and dialog boxes in SILworX required for configuring the HIMA PROFIBUS DP master.

Equipment and System Requirements

Element	Description
HIMA controller	HIMax with COM module. HIMatrix as of CPU OS V7 and COM OS V12 HIQuad X: Not applicable
COM module	The serial fieldbus interface (FB1 or FB2) used on the COM module must be equipped with an optional HIMA PROFIBUS DP master submodule.
Activation	Activation via fieldbus submodule, refer to the communication manual (HI 801 101 E) fo details.

Table 30: Equipment and System Requirements

i For the HIMax system, HIMA recommends operating the PROFIBUS DP using the FB1 fieldbus interface (maximum transfer rate 12 Mbit). The maximum transfer rate permitted for the FB2 fieldbus interface is 1.5 MBit.

PROFIBUS DP Master Properties:

Element	Description
Type of HIMA PROFIBUS DP master	DP-V1 Class 1 Master with additional DP-V2 functions
Transfer rate	9.6 kbit/s…12 Mbit/s
Bus address	0…125
Max. number of PROFIBUS DP master	Two PROFIBUS DP masters can be configured for each HIMatrix COM module or HIMax F30, F35, F60. Only one PROFIBUS DP master can be configured for each HIMatrix F20.
Max. number of PROFIBUS DP slaves	Up to 122 slaves can be configured for each resource (in all master protocol instances). However, a maximum of 31 slaves can be connected to a bus segment without repeaters.
Maximum process data length to a slave	DP output: max. 244 bytes DP input: max. 244 bytes

Table 31: PROFIBUS DP Master Properties

According to the standard, a total of three repeaters may be used so that a maximum of 122 bus stations are possible per serial interface on a master.

4.1.1 Creating a HIMA PROFIBUS DP Master

To create a new HIMA PROFIBUS DP Master

1. In the structure tree, open **Configuration, Resource, Protocols.**
2. Select **New**, **PROFIBUS DP Master** from the context menu of Protocols to add a new PROFIBUS DP master.
3. Select **Properties, General** from the context menu of the PROFIBUS DP master.
4. Select **Module** and **Interfaces.**

4.2 PROFIBUS DP: Example

In this example, a HIMA PROFIBUS DP master exchanges variables with a HIMA PROFIBUS DP slave.

The example shows how to create and configure a HIMA PROFIBUS DP master and a PROFIBUS DP slave.

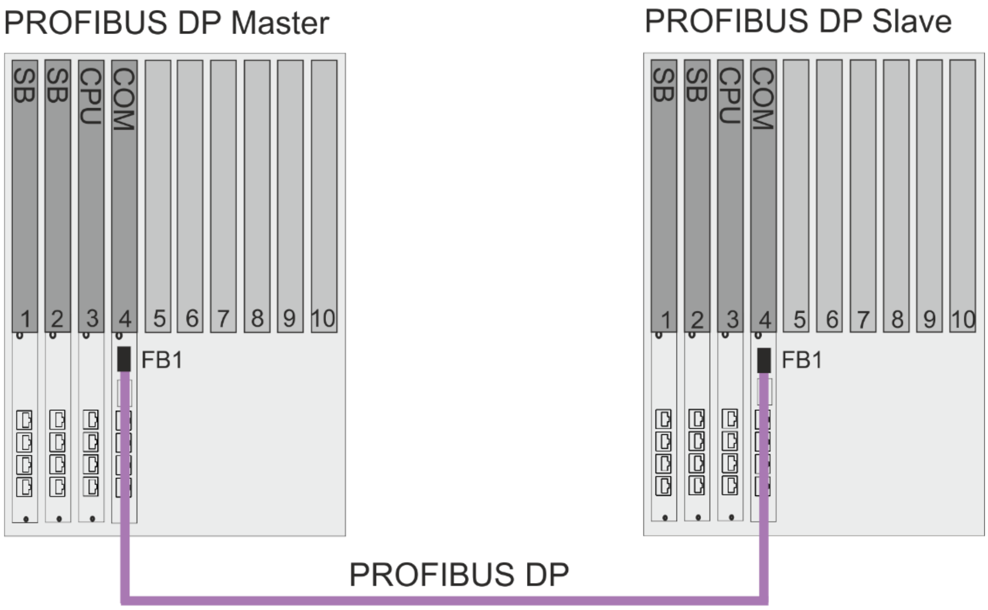


Figure 10: Communication via PROFINET IO

Fieldbus interface 1 on the COM modules of both HIMax controllers must be equipped with the corresponding PROFIBUS DP submodule, refer to the communication manual (HI 801 101 E).

For this example, the following global variables must be created in SILworX:

Global variable	Type
PB_Slave_Master1	UINT
PB_Slave_Master2	DWORD
PB_Slave_Master3	DWORD
PB_Slave_Master4	BYTE
PB_Master_Slave1	UINT
PB_Master_Slave2	BYTE

4.2.1 Configuring the PROFIBUS DP Slave

Configuration of the PROFIBUS DP slave.

To create a new HIMA PROFIBUS DP slave

1. In the structure tree, open **Configuration, Resource, Protocols.**
2. Select **New, PROFIBUS DP Slave** from the context menu of protocols to add a new PROFIBUS DP slave.
3. Select **Edit** from the context menu of the PROFIBUS DP slave.
4. In the **Properties** tab, select **COM Module** and **Interface** (e.g., FB1).

To assign variables in the HIMA PROFIBUS DP slave

1. Select **Edit** from the context menu of the PROFIBUS DP slave.
2. In the **Edit** dialog box, select the Process Variables tab.

i The start address of the input and output variables in the HIMA PROFIBUS DP slave always begin with 0. If the PROFIBUS DP master (from another manufacturer) expects a higher start address, dummy variables must be added to the reference variables.

Outputs in the HIMA PROFIBUS DP Slave

Name	Type	Offset	Global variable
PB_Slave_Master1	UINT	0	PB_Slave_Master1
PB_Slave_Master2	DWORD	2	PB_Slave_Master2
PB_Slave_Master3	DWORD	6	PB_Slave_Master3
PB_Slave_Master4	BYTE	10	PB_Slave_Master4

Table 32: Outputs for the HIMA PROFIBUS DP Slave

1. Drag the global variables to be sent from the Object Panel onto the **Output Variables** area.

i In this example, the output variables of the HIMA PROFIBUS DP slave are composed of **4 variables** with a total of **11 bytes**. The start address of the output variable with the lowest offset is **0**.

2. Right-click a free space in the **Output Variables** area to open the context menu.
3. Click **New Offsets** to generate the variable offsets.

Inputs in the HIMA PROFIBUS DP Slave

Name	Type	Offset	Global variable
PB_Master_Slave1	UINT	0	PB_Master_Slave1
PB_Master_Slave2	BYTE	2	PB_Master_Slave2

Table 33: Inputs in the HIMA PROFIBUS DP Slave

1. Drag the global variables to be received from the Object Panel onto the **Input Variables** area.

i In this example, the input variables of the HIMA PROFIBUS DP slave are composed of **2 variables** with a total of **3 bytes**. The start address of the input variable with the lowest offset is **0**.

2. Right-click a free space in the **Input Variables** area to open the context menu.
3. Click **New Offsets** to generate the variable offsets.

To verify the PROFIBUS DP slave configuration

1. In the structure tree, select **Configuration, Resource, Protocols, PROFIBUS DP Slave.**
2. Right-click and select **Verification** from the context menu.
3. Thoroughly verify the messages displayed in the logbook and correct potential errors.

i The configuration of the PROFIBUS DP slave must be recompiled together with the user program of the PROFIBUS DP slave resource and loaded to the controllers. The new configuration can only be used for communication with the PROFIBUS DP upon completion of this step.

4.2.2 Configuring the PROFIBUS DP Master

To create a new HIMA PROFIBUS DP master

1. In the structure tree, open **Configuration, Resource, Protocols.**
2. Select **New, PROFIBUS DP Master** from the context menu of protocols to add a new PROFIBUS DP master.
3. Select **Properties**, **General** from the context menu of the PROFIBUS DP master.
4. In the **General** tab, select **COM Module** and **Interfaces** (e.g., FB1).

i Perform these steps to configure the HIMax PROFIBUS DP slave from within the HIMax PROFIBUS DP master.

To create a HIMax PROFIBUS DP slave in the PROFIBUS DP master

1. Select **New**, **PROFIBUS DP Slave** from the context menu of the PROFIBUS DP master.

To read the GSD file for the new PROFIBUS DP slave

1. In the structure tree, open **Configuration**, **Resource**, **Protocols**, **PROFIBUS DP Master**, **PROFIBUS Slave.**
2. Select **Read GSD File** from the context menu of the PROFIBUS DP master and choose the GSD file for the PROFIBUS slave (e.g., hax100ea.gsd).

i The GSD files for HIMax controllers are available on the HIMA website at www.hima.com.

4.2.2.1 Creating the HIMax PROFIBUS DP Modules

The number of bytes to be actually transferred must be configured in the PROFIBUS DP master. To do this, add *Modules* until the physical configuration of the slave is achieved.

> i The number of modules used to achieve the necessary number of bytes is not important as long as the maximum of 32 modules is not exceeded.
> To avoid unnecessarily complicating the PROFIBUS DP master configuration, HIMA recommends keeping the number of selected modules to a minimum.

To create the required PROFIBUS DP Modules

1. In the structure tree, open **Configuration**, **Resource**, **Protocols**, **PROFIBUS DP Master**, **PROFIBUS Slave.**
2. On the menu bar, select **PROFIBUS DP Master**, **Add Modules.**
3. For this example, select the following modules to receive **11 bytes** from the PROFIBUS DP slave and to send **3 bytes**.

To number the PROFIBUS DP modules

1. Right-click the first **PROFINET DP Module**, and then select **Properties** from the context menu.
2. Enter **0** in the **Slot** field.
3. Repeat these steps for every further **PROFIBUS DP Module** and number the modules consecutively.

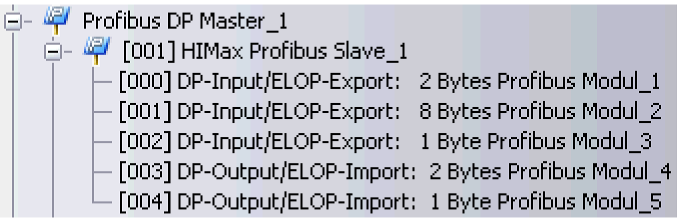


Figure 11: HIMax PROFIBUS DP Slave with Modules

> i **Number** the HIMax PROFIBUS DP modules without gaps and in ascending order, starting with **0**.
> The order in which the PROFIBUS DP modules are arranged is not important for operation. However, HIMA recommends organizing the DP input and output modules in an orderly manner to maintain a clear overview.

4.2.2.2 Configuring the Input and Output Modules

i The sum of the variables (in bytes), must be identical to the size of the module (in bytes).

To configure the input module [000] DP Input/ELOP Export: 2 bytes

1. In the PROFIBUS DP slave, select the input module **[000] DP Input/ELOP Export: 2 Bytes.**
2. Right-click the input module and select **Edit** from the context menu.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Input Signals** area of the input module **[000] DP-Input/ELOP-Export: 2 Bytes**.

Name	Type	Offset	Global Variable
PB_Slave_Master1	UINT	0	PB_Slave_Master1

Table 34: Variables of the Input Module [000] DP Input/ELOP Export: 2 Bytes

5. Right-click a free space in the **Input Signals** area to open the context menu.
6. Click **New Offsets** to generate the variable offsets.

To configure the input module [001] DP Input/ELOP Export: 8 bytes

1. In the PROFIBUS DP slave, select the input module **[001] DP Input/ELOP Export: 8 Bytes.**
2. Right-click the input module and select **Edit** from the context menu.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Input Signals** area of the input module **[001] DP-Input/ELOP-Export: 8 Bytes**.

Name	Type	Offset	Global variable
PB_Slave_Master2	DWORD	0	PB_Slave_Master2
PB_Slave_Master3	DWORD	4	PB_Slave_Master3

Table 35: Variables of the Input Module [001] DP Input/ELOP Export: 8 Bytes

5. Right-click a free space in the **Input Signals** area to open the context menu.
6. Click **New Offsets** to generate the variable offsets.

To configure the input module [002] DP Input/ELOP Export: 1 byte

1. In the PROFIBUS DP slave, select the input module **[002] DP Input/ELOP Export: 1 Byte**.
2. Right-click the input module and select **Edit** from the context menu.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Input Signals** area of the input module **[002] DP-Input/ELOP-Export: 1 Byte**.

Name	Type	Offset	Global variable
PB_Slave_Master4	BYTE	0	PB_Slave_Master4

Table 36: Variables of the Input Module [002] DP Input/ELOP Export: 1 Byte

5. Right-click a free space in the **Input Signals** area to open the context menu.
6. Click **New Offsets** to generate the variable offsets.

To configure the output module [003] DP Output/ELOP Import: 2 Bytes

1. In the PROFIBUS DP slave, select the output module **[003] DP Output/ELOP Import: 2 Bytes**.
2. Right-click the output module, then click Edit.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Output Signals** area of the output module **[003] DP-Output/ELOP-Import: 2 Bytes**.

Name	Type	Offset	Global variable
PB_Master_Slave1	UINT	0	PB_Master_Slave1

Table 37: Variables of the Output Module [003] DP Output/ELOP Import: 2 Bytes

5. Right-click a free space in the **Output Signals** area to open the context menu.
6. Click **New Offsets** to generate the variable offsets.

To configure the output module [004] DP Output/ELOP Import: 1 Byte

1. In the PROFIBUS DP slave, select the output module **[004] DP Output/ELOP Import: 1 Byte.**
2. Right-click the output module, then click Edit.
3. In the **Edit** dialog box, select the **Process Variables** tab.
4. Drag the appropriate variable from the Object Panel onto the **Output Signals** area of the output module **[004] DP-Output/ELOP-Import: 1 Byte**.

Name	Type	Offset	Global variable
PB_Master_Slave2	BYTE	0	PB_Master_Slave2

Table 38: Variables of the Output Module [004] DP Output/ELOP Import: 1 Byte

5. Right-click a free space in the **Output Signals** area to open the context menu.
6. Click **New Offsets** to generate the variable offsets.

4.2.2.3 Creating the User Data within the PROFIBUS DP Master

Creating the User Data within the PROFIBUS DP Master

1. In the structure tree, open **Configuration**, **Resource**, **Protocols**, **PROFIBUS DP Master**, **PROFIBUS DP Slave.**
2. Right-click the first **PROFINET DP module**, and then select **Properties** from the context menu.
3. Select the **Data** tab and click the **Edit** button next to the user data.

The group's *start address* and *number of variables* are defined in the 32 bytes long user data field, see also Chapter 4.8.

4. For this example, create the following user data:
   4, to ensure that **4 variables** are received by the PROFIBUS DP master.
   2, to ensure that **2 variables** will be sent by the PROFIBUS DP master.
   The start addresses of the input and output blocks begin with **0**.

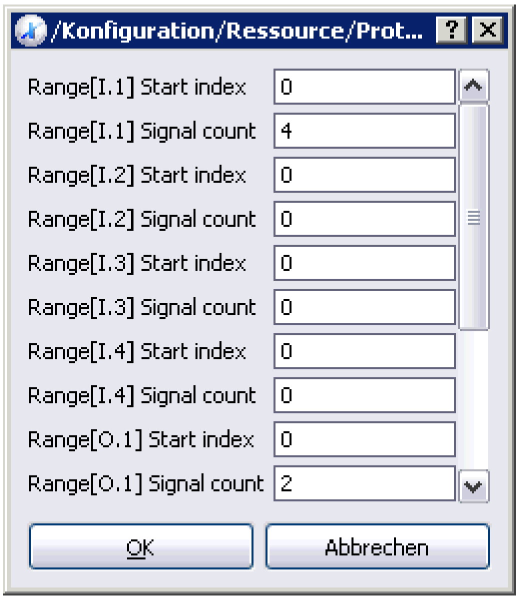


Figure 12: User Data Field

To verify the PROFIBUS DP slave configuration

1. In the structure tree, select **Configuration, Resource, Protocols, PROFIBUS DP Slave.**
2. Right-click and select **Verification** from the context menu.
3. Thoroughly verify the messages displayed in the logbook and correct potential errors.

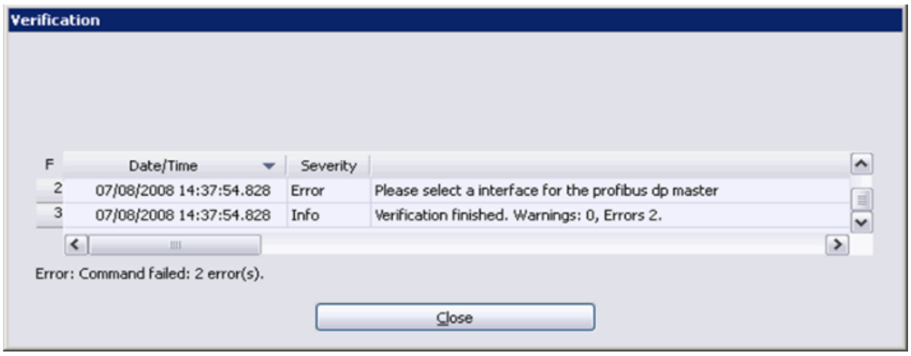


Figure 13: Verification Dialog Box

i The configuration of the PROFIBUS DP master must be recompiled together with the user program of the PROFIBUS DP master resource and transferred to the controller. The new configurations can only be used for communication with the PROFIBUS DP upon completion of this step.

4.2.2.4 Optimizing the PROFIBUS DP parameters

Using the default values for the PROFIBUS parameters, smooth PROFIBUS communication is generally not a problem. However, the settings should be further optimized to achieve faster data exchange rates and improve fault detection.

To determine the actual target rotation time TTR [ms]

1. Open the Control Panel associated with the HIMax PROFIBUS DP master.
2. In the structure tree associated with the Control Panel, click **PROFIBUS DP Master** and read the actual **Target Rotation Time TTR [ms].** Note down this value.

To determine the parameters required for the PROFIBUS DP slave

1. In the structure tree, open **Configuration**, **Resource**, **Protocols**, **PROFIBUS DP Master**, **PROFIBUS DP Slave.**
2. Right-click HIMax **PROFIBUS Slave**, and then click **Properties**.
3. Select the **Features** tab and read **Min. Slave Interval MSI [ms]** for this PROFIBUS DP slave. Note down this value.
4. Select the **Transfer Rate** tab and read **Max. Tsdr** for the transfer rate used. Note down this value.

To enter the parameters previously determined

1. Right-click **PROFIBUS Master** and select **Properties** from the context menu.
2. Select the **Timings** tab.

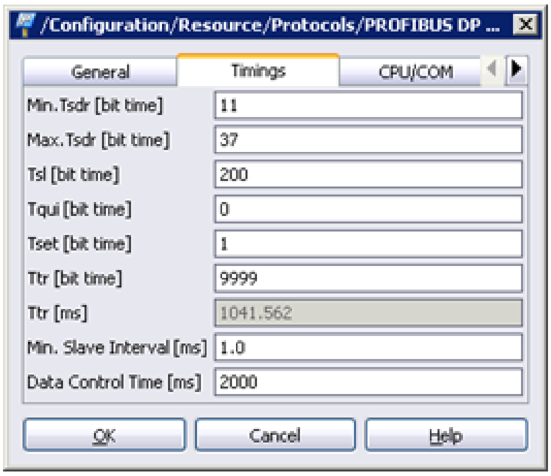


Figure 14: PROFIBUS DP Master Properties

3. Convert the **Max. Tsdr** that was previously noted down in **bit Time**.
4. Convert the **Target Rotation Time TTR [ms]** that was previously noted down in **bit Time**, add 1/3 safety margin and enter the resulting value in the **Target Rotation Time TTR [ms]** field.
5. Enter the **Min. Slave Interval MSI [ms]** that was previously noted down.

i If various slaves are configured, the largest values of the parameters MaxTsdr [bit time] and Min. Slave Interval [ms] must be used.

6. The data control time [ms] must be set to ≥ 6*Ttr [ms].

To enter the watchdog time for the PROFIBUS DP slave

1. In the structure tree, open **Configuration**, **Resource**, **Protocols**, **PROFIBUS DP Master**, **PROFIBUS DP Slave.**
2. Right-click HIMax **PROFIBUS Slave**, and then click **Properties**.

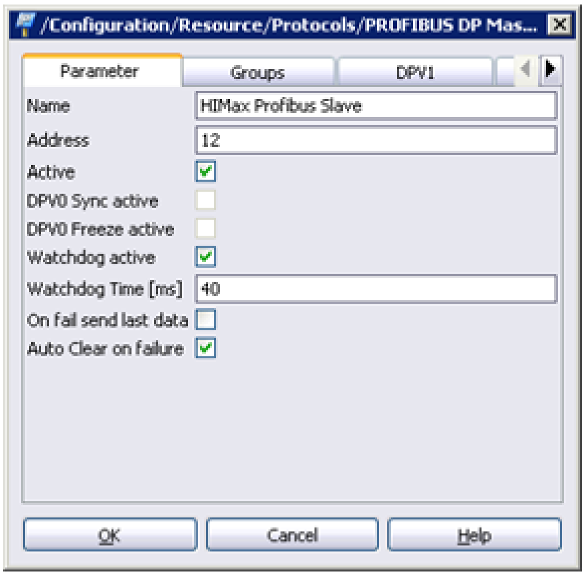


Figure 15: PROFIBUS DP Slave Properties

3. Select the **Parameter** tab and mark the **Watchdog Active** checkbox.
4. Enter the watchdog time [ms] ≥ 6*Ttr [ms] in the **Watchdog Time [ms]** field.

---

i The configurations of the PROFIBUS DP master and slave must be recompiled together with the user program of the PROFIBUS DP master and slave resources and transferred to the controllers. The new configurations can only be used for communication with the PROFIBUS DP upon completion of this step.

4.3 Menu Functions of the PROFIBUS DP Master

4.3.1 Edit

The **Edit** context menu function for the PROFIBUS DP master is used to open the **Edit** dialog box.

The **System Variables** tab provides the following system variables, which allow the PROFIBUS DP master state to be evaluated in the user program.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>Number of Errors</td><td>Number of errors since statistics reset.</td></tr>
<tr><td>Baud Rate</td><td>Baud rate (bit/s) used for the bus.</td></tr>
<tr><td>Bus Error</td><td>If a bus error occurs, an error code is set in the Bus Error system variable. An error code retains its value until the bus error has been removed.
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>0</td><td>OK, no bus error.</td></tr>
<tr><td>1</td><td>Address error
The master address is already available on the bus</td></tr>
<tr><td>2</td><td>Bus malfunction
Malfunction detected on the bus, (e.g., bus not properly terminated, multiple subscribers are sending data simultaneously).</td></tr>
<tr><td>3</td><td>Protocol error
An incorrectly coded packet was received.</td></tr>
<tr><td>4</td><td>Hardware fault
The hardware reported a fault, e.g., too short time periods.</td></tr>
<tr><td>5</td><td>Unknown error
The master changed the state for an unknown reason.</td></tr>
<tr><td>6</td><td>Controller reset
The controller chip is reset if a severe bus error occurs.</td></tr>
</table>
To evaluate the Bus Error status variable from within the user program, it must be connected to a variable.</td></tr>
<tr><td>Average Cycle Time</td><td>Measured average bus cycle time in milliseconds.</td></tr>
<tr><td>Last Cycle Time</td><td>Measured bus cycle time in milliseconds.</td></tr>
<tr><td>Master Status</td><td>Indicates the current protocol state.
0: OFFLINE
1: STOP
2: CLEAR
3: OPERATE
Connect the status variable Master State to a variable to evaluate it in the user program.</td></tr>
<tr><td>Maximum Cycle Time</td><td>Measured maximum bus cycle time in milliseconds.</td></tr>
<tr><td>Min. Slave Interval</td><td>Minimum slave interval measured for one of the slaves assigned to this master.</td></tr>
<tr><td>Minimum Cycle Time</td><td>Measured minimum bus cycle time in milliseconds.</td></tr>
<tr><td>Target Rotation Time</td><td>Target token rotation time</td></tr>
</table>

Table 39: System Variables in the PROFIBUS DP Master

4.3.2 The Properties Menu Function

The **Properties** function of the context menu for the PROFIBUS DP master is used to open the **Properties** dialog box.

The dialog box contains the following tabs:

4.3.2.1 The General Tab

Element	Description
Type	PROFIBUS DP master.
Name	Any unique name for a PROFIBUS DP master.
Module	Selection of the COM module within which the protocol is processed.
Activate Max. µP Budget	Activated: Use the µP budget limit from the *Max. µP Budget in [%]* field. Deactivated: Do not use the µP budget limit for this protocol.
Max. µP Budget in [%]	Maximum µP budget of the module that can be used for processing the protocols. Range of values: 1…100% Default value: 30%
Behavior on CPU/COM connection loss	If the connection of the processor module to the communication module is lost, the input variables are either initialized or are still used unchanged in the process module, depending on this parameter. For instance, if the communication module is removed when communication is running. **If a project created with a SILworX prior to V3 should be converted, this value must be set to Retain Last Value to ensure that the CRC does not change.** **For HIMatrix controllers with CPU OS prior to V8, this value must always be set to Retain Last Value.** Apply Initial Data — Input variables are reset to their initial values. Retain Last Value — The input variables retain the last value.
Address	Master station address. Only one master station address may be available on the bus. Range of values: 0…125 Default value: 0
Interface	COM interface that should be used for the master. Range of values: FB1, FB2

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>Baud rate</td><td>Baud rate (bit/s) used for the bus.
Possible values:
<table>
<tr><th>Value</th><th>Baud rate</th><th>FB1</th><th>FB2</th></tr>
<tr><td>9600</td><td>9.6 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>19200</td><td>19.2 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>45450</td><td>45.45 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>93750</td><td>93.75 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>187500</td><td>187.5 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>500000</td><td>500 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>1500000</td><td>1.5 Mbit/s</td><td>X</td><td>X</td></tr>
<tr><td>3000000</td><td>3 Mbit/s</td><td>X</td><td>-</td></tr>
<tr><td>6000000</td><td>6 Mbit/s</td><td>X</td><td>-</td></tr>
<tr><td>12000000</td><td>12 Mbit/s</td><td>X</td><td>-</td></tr>
</table>
Default value: 9.6 kbit/s</td></tr>
</table>

Table 40: General Properties for the PROFIBUS DP Master

4.3.2.2 The Timings Tab

Element	Description
MinTsdr [bit time]	Min. Station Delay Time Minimum time period that a PROFIBUS DP slave must wait before it may respond. Range of values: 11…1023 Default value: 11
MaxTsdr [bit time]	Max. Station Delay Time Maximum time period that a PROFIBUS DP slave may need to respond. Max Tsdr ≥ Tsdr (of the connected slave with the highest Tsdr). The MaxTsdr values of the slaves are read from the GSD files and are specified in the **Baud Rates** tab in the slave's **Properties** dialog box. Range of values: 37…65535 Default value: 100
Tsl [bit time]	Slot Time Maximum time period that the master waits for a slave to respond. Tsl > MaxTsdr + 2*Tset +Tqui + 13 Range of values: 37…16383 Default value: 200
Tqui [bit time]	Quiet Time for Modulator Time that a station may need to switch from sending to receiving. Range of values: 0…493 Default value: 0
Tset [bit time]	Setup Time Response time to an event. Range of values: 1…494 Default value: 1
Ttr [bit time]	Configured token rotation time Maximum time available for a token rotation. A low estimate of the Ttr is obtained through a calculation. Refer to Chapter 4.4.4 for details. Range of values: 256…16777215 Default value: 9999
TTR [ms]	Actual token rotation time in ms
Min. Slave Interval [ms]	Minimum time between 2 cyclical requests of a slave. The master observes the Min. Slave Interval and does not fall below it. However, the PROFIBUS DP cycle can be extended if Isochronous Mode is inactive and the portion of acyclic telegrams increases within a cycle. The value for *Min. Slave Interval* of the slave is read from the GSD file and is located on the **Features** tab of the slave in the **Properties** dialog box. In Isochronous Mode, the value for *Min. Slave Interval* defines the time period for an isochronous cycle. Isochronous Mode is activated if Isochronous Sync Mode or Isochronous Freeze Mode is active. See also Refresh Rate between CPU and COM (CPU/COM tab). Range of values: 0…6553.5 (step size: 0.1 ms) Default value: 1.0

Element	Description
User Data Monitoring Time [ms]	Time span within which the master must report its current state on the bus Standard value: User data monitoring time = WDT of the slave Range of values: 0…65,5350 (step size: 10 ms) Default value: 2000

Table 41: Timings Tab in the Properties Dialog Box for the PROFIBUS DP Master

4.3.2.3 The CPU/COM Tab

The default values for the parameters provide the fastest possible exchange of PROFIBUS DP data between the COM module (COM) and the processor module (CPU) within the HIMax controller. These parameters should only be changed if it is necessary to reduce the COM and CPU loads for an application, and the process allows this change.

i Only experienced programmers should modify the parameters. Increasing the COM and CPU refresh rate means that the effective refresh rate of the PROFIBUS DP data is also increased. The system time requirements must be verified.

Take also the parameter *Min. Slave Interval [ms]* into account which defines the refresh rate of the PROFIBUS DP data from/to the PROFIBUS DP slave. This can be increased according to the CPU/COM refresh rate.

Element	Description
Process Data Refresh Rate [ms]	Refresh rate in milliseconds at which the COM and CPU exchange protocol data. If the *Refresh Rate* is zero or less than the cycle time for the controller, data is exchanged as fast as possible. Range of values: 0…(2^{31}–1) Default value: 0
Force Process Data Consistency	Activated: Transfer of all of the protocol data from the CPU to the COM within a CPU cycle. Deactivated: Transfers all of the protocol data from the CPU to the COM, distributed over multiple CPU cycles, each with 1100 bytes per data direction. This may also cause the cycle time of the controller to be reduced. Default value: Activated

Table 42: CPU/COM Tab in the Properties Dialog Box for the PROFIBUS DP Master

4.3.2.4 The Others Tab

Element	Description
Max. Number of Resends	Maximum number of resends attempted by a master if a slave does not respond. Range of values: 0…7 Default value: 1
Highest Active Address	Highest Station Address (HSA) Highest station address to be expected for one master. Masters having a station address beyond the HSA are not included into the token ring. Range of values: 0…125 Default value: 125
Isochronous Sync Mode	Isochronous Sync Mode allows both a clock-controlled synchronization of the master and the slaves and a simultaneous activation of the physical outputs of multiple slaves. If Isochronous Sync Mode is active, the master sends the Sync control command as a broadcast telegram to all slaves. As soon as the slaves supporting Isochronous Sync Mode receive the "Sync" control command, they synchronously switch the data from the user program to the physical outputs. The values of the physical outputs remain frozen up to the next Sync control command. The cycle time is determined by the Min. Slave Interval. Condition: Ttr < Min. Slave Interval Default value: Deactivated
Isochronous Freeze Mode	Isochronous Freeze Mode allows the input data from multiple slaves to be simultaneously accepted. If Isochronous Freeze Mode is active, the master sends the Freeze control command as a broadcast telegram to all slaves. As soon as the slaves supporting Isochronous Freeze Mode receive the Freeze control command , the physical inputs' variables are frozen to the current value. The master can thus read the values. The input data are only updated when the next Freeze control command is sent. The cycle time is determined by the Min. Slave Interval. Condition: Ttr < Min. Slave Interval Default value: Deactivated
Auto Clear on Error	If *Auto Clear on Error* is set in a slave that fails, the master enters the CLEAR state. Default value: Deactivated
Time Master	The master is also timing master and periodically sends the system time via the bus. Default value: Deactivated
Clock Sync Interval [ms]	Clock synchronization interval. Time interval within which the timing master sends the system time over the bus. Range of values: 0…65,5350 (step size: 10 ms) Default value: 0 (no timing master)

Table 43: Other Properties for the PROFIBUS DP Master

4.4 PROFIBUS DP Bus Access Method

The bus access method provides a defined time window to every station within which the station can perform its communication tasks.

4.4.1 Master/Slave Protocol

The bus assignment between a PROFIBUS DP master and a PROFIBUS DP slave is ensured by the master/slave method.

An active PROFIBUS DP master communicates with passive PROFIBUS DP slaves.

The PROFIBUS DP master with the token is authorized to send and may communicate with the PROFIBUS DP slaves assigned to it. The master assigns the bus to a slave for a certain time and the slave must respond within this time period.

4.4.2 Token Protocol

The bus assignment between automation devices (class 1 masters) and/or programming devices (class 2 masters) is ensured via token passing.

All PROFIBUS DP masters connected to a common bus form a token ring. As long as the active PROFIBUS DP master has the token, it assumes the master function on the bus.

In a token ring, the PROFIBUS DP masters are organized in ascending order according to their station addresses. The token is passed on in this order until it is received by the PROFIBUS DP master with the highest station address.

This master passes the token on to the master with the lowest station address to close the token ring.

The token rotation time corresponds to one token cycle through all the PROFIBUS DP masters. The token rotation time (Ttr) is the maximum time permitted for a token rotation.

4.4.3 Token Rotation Time (Ttr)

Default values for different transfer rates

While configuring the PROFIBUS DP master, take into account that some parameters set in the **Timings** tab depend on the baud rate set in the **General** tab. For the first (initial) configuration, use the default values specified in the following table. The values are optimized at a later point in time.

	9.6k	19.2k	45.45k	93.75k	187.5k	500k	1.5M	3M	6M	12M
MinTsdr	11	11	11	11	11	11	11	11	11	11
MaxTsdr	60	60	400	60	60	100	150	250	450	800
Tsl bit time	100	100	640	100	100	200	300	400	600	1000
Tqui bit time	0	0	0	0	0	0	0	3	6	9
Tset bit time	1	1	95	1	1	1	1	4	8	16

Table 44: Default Values for Token Rotation Time at Different Transfer Rates

All time values specified are expressed in Tbit (1 Tbit = 1/[bit/s]).

MinTsdr is at least 11 Tbits long since a character is composed of 11 bits (1 start bit, 1 stop bit, 1 parity bit, 8 data bits).

Transmission Time for a Character

Baud rate	Tbit bit = 1/baud rate	Time
9600 bit/s	1 / 9600 = 104.166 µs	11*104.166 µs = of 114.583 ms
6 Mbit/s	1/ $6*10^6$ = 166.667 ns	11*166.667 ns = 1.833 µs

Table 45: Transmission Time for a Character at Different Transfer Rates

4.4.4 Calculating the Token Rotation Time (Ttr)

Calculate the minimum token rotation time Ttr as follows:

Ttr_{min} = n * (198 + T1 + T2) + b * 11 + 242 + T1 + T2 + Tsl

Element	Description
n	Number of active slaves
b	Number of I/O data bytes of the active slaves (input plus output)
T0	35 + 2 * Tset + Tqui
T1	If T0 < MinTsdr: T1 = MinTsdr If T0 > MinTsdr: T1 = T0
T2	If T0 < MaxTsdr: T2 = MaxTsdr If T0 > MaxTsdr: T2 = T0
Tsl	Slot Time: Maximum time period that the master waits for a slave to respond
198	Twice a telegram header with variable length (for request and response)
242	Global_Control, FDL_Status_Req and token passing

Table 46: Elements Required for Calculating the Token Rotation Time

i

The estimate of the target token rotation time *Ttr* is only valid if only one master is operated on the bus, no resends are necessary and no acyclic data has to be transferred.

Never set *Ttr* to a value less than that calculated with the above formula. Otherwise fault-free operation can no longer be ensured. HIMA recommends using a value two or three times greater than the result.

Example of calculating the target token rotation time *Ttr*

The following configuration is available:

5 active slaves

(n = 5)

20 I/O data bytes per slave

(b = 100)

The following time constants for a transmission rate of 6 Mbit/s are taken from Table 46:

- MinTsdr = 11 T_{bit}
- MaxTsdr = 450 T_{bit}
- Tsl bit time = 600 T_{bit}
- Tqui bit time = 6 T_{bit}
- Tset bit time = 8 T_{bit}

T0 = 35 + 2 * Tset + Tqui

T0 = 35 + 2 * 8 + 6

T0 = 57 T_{bit}

As T0>MinTsdr: **T1 = T0 = 57 T_{bit}**

As T0<MaxTsdr: **T2 = MaxTsdr = 450 T_{bit}**

Use the computed values in the formula for the minimum token rotation time:

$$\mathbf{Ttr_{min} = n * (198 + T1 + T2) + b * 11 + 242 + T1 + T2 + Tsl}$$

Ttr_{min}= 5 (198+57+450)+100*11+242+57+450+600

Ttr_{min} [T_{bit}] = 5974 T_{bit}

Result:

Ttr_{min} [µs] = 5974 T_{bit} * 166.67 ns = 995.68 µs

---

i

Ttr is verified when it is entered into the dialog box.

If the value set for *Ttr* is lower than the value calculated by SILworX, an error message appears in the logbook. A minimum value for *Ttr* is also suggested.

If *Isochronous Sync Mode* or *Isochronous Freeze Mode* is activated, the cycle time is defined by the parameter *Minimum Slave Interval*. *Ttr* must then in any case be lower than the *Minimum Slave Interval*.

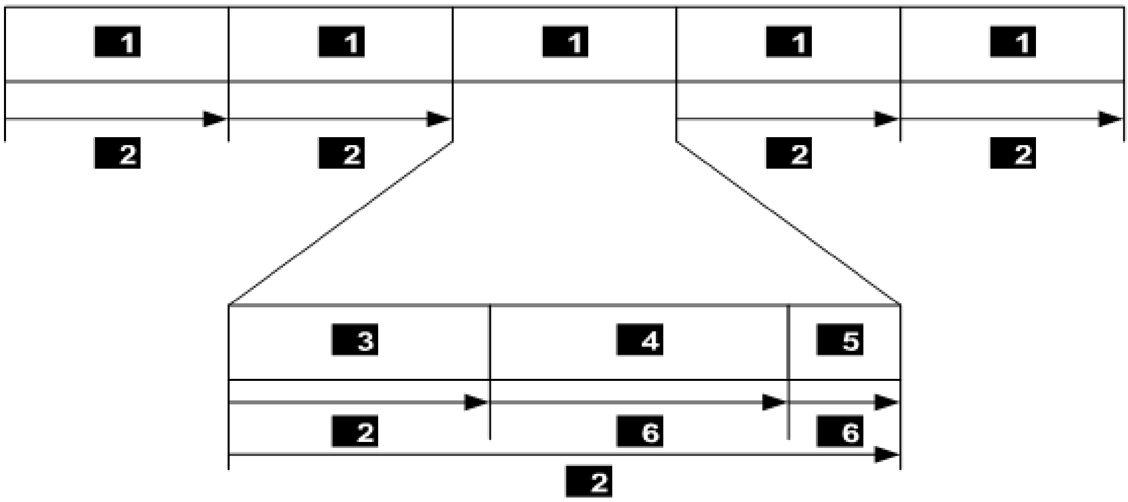
If this condition is not met in isochronous mode, an error message appears.

4.5 Isochronous PROFIBUS DP Cycle (DP-V2 and Higher)

The PROFIBUS DP cycle consists of 2 telegram phases: a fixed and cyclical phase and an event-driven and acyclic phase.

The acyclic phase of a PROFIBUS DP cycle can extend the corresponding PROFIBUS DP cycle. This effect is not wanted in specific applications and areas, such as drive technology.

To achieve a constant cycle time (t_{const}), Isochronous Mode must be activated in the master, where the *Min. Slave Interval [ms]* parameter defines the constant cycle time (t_{const}). Configured in this way, the isochronous PROFIBUS DP cycle offers clock accuracy with a difference of < 10 ms.



1 Isochronous DP cycle
2 t_{const}
3 Cyclic portion
4 Acyclic portion
5 Pause
6 $t_{variable}$

Figure 16: Isochronous PROFIBUS DP Cycle

To determine the cyclical phase, the minimum token rotation time must be calculated.

Additionally, a sufficiently large time interval (typically two to three times the minimum token rotation time *Ttr*) must be reserved for the acyclic phase. If the reserved time is not needed, a break is taken prior to starting the next cycle to ensure the cycle time remains constant. See also Chapter 4.4.3 concerning the token rotation time (*Ttr*).

i The master is configured entering the DP cycle time determined by the user into *Min. Slave Interval [ms].*

To operate in *Isochronous Mode*, one of the 2 parameters *Isochronous Sync Mode* or *Isochronous Freeze Mode* must be activated in the master.

On the bus, only one master may simultaneously operate in isochronous mode. Additional masters are not permitted.

4.5.1 Isochronous Mode (DP-V2 and Higher)

This function allows a clock-controlled synchronization in the master and the slaves, irrespective of congestion on the bus. The bus cycle is synchronized with a clock difference of less than 10 ms. Highly precise positioning processes can be thus implemented.

i To a certain degree, slaves (DP-V0 slaves) that do not support *Isochronous Mode* can also benefit from its advantages. To do so, the slaves must be assigned to Group 8 and the parameters *Sync* and/or *Freeze* must be activated.

Sync Mode and *Freeze Mode* are normally used simultaneously.

4.5.2 Isochronous Sync Mode (DP-V2 and Higher)

Isochronous Sync Mode *allows both a clock-controlled synchronization of the master and the slave and a simultaneous activation of the outputs of multiple slaves.*

4.5.3 Isochronous Freeze Mode (DP-V2 and Higher)

Isochronous Freeze Mode allows the input data from multiple slaves to be simultaneously accepted.

4.6 Menu Functions of the PROFIBUS DP Slave (in the Master)

4.6.1 Creating a PROFIBUS DP Slave (in the Master)

To create a PROFIBUS DP slave in the HIMA PROFIBUS DP master

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master.**
2. In the context menu for PROFIBUS DP master, click **New**, **PROFIBUS Slave** to add a new PROFIBUS slave.

4.6.2 Edit

The **Edit** function of the context menu for the PROFIBUS DP slave is used to open the **system variables** dialog box.

The **System Variables** tab provides the following system variables, which allow the PROFIBUS DP slave state to be evaluated in the user program.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>Activation Control</td><td>A change from 0 to 1 deactivates the slave.
A change from 1 to 0 activates the slave previously deactivated.
Activated = 0
Deactivated = 1</td></tr>
<tr><td>PNO Ident Number</td><td>16-bit unique number assigned by the PROFIBUS DP Nutzerorganisation e.V. (PNO) Germany to a product (field device) and identifying it.</td></tr>
<tr><td>Standard Diagnostics</td><td>Is used by the slave to inform the master about its current state. This variable always contains the last received standard diagnostics. The parameters comply with the diagnostic telegram in accordance with IEC 61158.</td></tr>
<tr><td>Connection Count</td><td>Is increased with every new connection.
Counts from the counter reset.</td></tr>
<tr><td>Connection State</td><td><table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Deactivated:
The parameter sets are loaded for these slaves, but the slaves are completely ignored. The input data is reset to its initial values, no activity related to these slaves is noted on the bus.</td></tr>
<tr><td>1</td><td>Inactive (not connected):
If a slave is no longer reachable, the input data is reset to the initial value.
One of the following options can be selected for each slave:
▪ The master continues to send output data to the slave
▪ The master attempts to re-configure the slave</td></tr>
<tr><td>2</td><td>Active (connected):
The slaves are exchanging I/O data with the CPU.</td></tr>
</table></td></tr>
<tr><td>Slave Alarm Counter</td><td>Number of alarms provided so far.
Counts from the counter reset.</td></tr>
<tr><td>Standard Diagnostic Count</td><td>Number of diagnostic messages provided so far.
Counts from the counter reset.</td></tr>
</table>

Table 47: System Variables in the PROFIBUS DP Slave

4.6.3 Properties

The **Properties** function of the context menu for the PROFIBUS DP slave is used to open the **Properties** dialog box. The dialog box contains the following tabs:

4.6.3.1 The Parameters Tab

Element	Description
Name	Name of the slave
Address	Address of the slave Range of values: 0…125 Default value: 0
Active	Slave State Only an active slave can communicate with a PROFIBUS DP master. Default value: Activated
DP-V0 Sync Active	*Sync Mode* allows the outputs of various DP-V0 slaves to be simultaneous activated. **Caution:** This field must be deactivated in DP-V2 slaves operating in *Isochronous Sync Mode*. Default value: Deactivated
DP-V0 Freeze Active	*Freeze Mode* allows the input data of multiple DP-V0 slaves to be simultaneously accepted. **Caution:** This field must be deactivated in DP-V2 slaves operating in *Isochronous Freeze Mode*. Default value: Deactivated
Watchdog Active	If the Watchdog Active checkbox is ticked, the slave detects a master's failure and enters the safe state. Default value: Deactivated
Watchdog Time [ms]	The Watchdog Active checkbox must be ticked. If master and slave do not exchange any data within this time interval, the slave disconnects itself and resets all DP output data to its initial values. 0 = Deactivated Standard value: Slave’s watchdog time > 6 * Ttr Range of values: 0…65535 Default value: 0
On Failure, Send Last Data	Activated: If a fault occurs, data is still sent, even without the slave’s acknowledgement. Deactivated: If a fault occurs, the connection is terminated and re-established. Default value: Deactivated
Auto Clear on Failure	This parameter is only effective if *Auto Clear on Error* is also activated in the master! Activated: The Auto Clear function for this slave is active. The master automatically switches from OPERATE to CLEAR state if no data exchange is possible with an Auto Clear slave. As soon as all Auto Clear slaves return to active, the master automatically switches to the OPERATE state. Deactivated: The Auto Clear function for this slave is not active. Default value: Activated

Table 48: The Parameters Tab in the PROFIBUS DP Slave

4.6.3.2 The Groups Tab

In this tab, the slaves can be organized into various groups. The global control commands, *Sync* and *Freeze*, can systematically address one or multiple groups.

Element	Description	
Member of Group 1	Member of group 1	Default value: Deactivated
Member of Group 2	Member of group 2	
Member of Group 3	Member of group 3	
Member of Group 4	Member of group 4	
Member of Group 5	Member of group 5	
Member of Group 6	Member of group 6	
Member of Group 7	Member of group 7	
Member of Group 8	Member of group 8	

Table 49: Groups Tab in the Properties Dialog Box for the PROFIBUS DP Slave

4.6.3.3 The DP-V1 Tab

This tab contains the parameters set with DP-V1 and higher. In DP-V0 slaves, no parameters can be selected in this tab. The **Supp.** column shows which parameters are supported by the slave.

Element	Description
DP-V1	If the DP-V1 mode is not activated, no DP-V1 features can be used. In this case, the slave acts like a DP-V0 slave. The configuration data may have to be changed (refer to the slave manual). Default value: Deactivated
Failsafe	If this mode is activated, a master in the CLEAR state does not send zeros as output data; rather, it sends an empty data packet (failsafe data packet) to the slave. The slave recognizes that it must place the safe output data on the outputs (the value of the safe output data is not necessarily zero). Default value: Deactivated
Isochronous Mode	This function allows a clock-controlled synchronization in the master and the slaves, irrespective of congestion on the bus. The bus cycle is synchronized with a clock difference of < 1 ms. Highly precise positioning processes can be thus implemented. Default value: Deactivated
Publisher Active	This function is required for the slave intercommunication. Direct and time-saving communication via broadcast between the slaves is thus ensured without detouring through the master. Default value: Deactivated
Prm Block Struct. Supp.	The slave supports structured configuration data (read only). Default value: Deactivated
Check Cfg Mode	Reduced configuration control: If Check Cfg Mode is activated, the slave can operate with an incomplete configuration. This field should be deactivated during start-up. Default value: Deactivated

Table 50: DP-V1 Tab in the Properties Dialog Box for the PROFIBUS DP Slave

4.6.3.4 The Alarms Tab

This tab is used to activate alarms. This, however, is only possible with DP-V1 slaves, if DP-V1 is activated and the slave supports alarms. The checkmarks in the **Supp.** column designate which alarms are supported by the slave. Mandatory alarms are noted in the **Required** column.

<table>
<tr><th>Element</th><th colspan="2">Description</th></tr>
<tr><td>Update Alarm</td><td>Alarm, if the module parameters changed.</td><td rowspan="6">Default value:
Deactivated</td></tr>
<tr><td>Status Alarm</td><td>Alarm, if the module state changed.</td></tr>
<tr><td>Vendor Alarm</td><td>Vendor specific alarm.</td></tr>
<tr><td>Diagnostic Alarm</td><td>Alarm, if specific events occur in a module, e.g., short circuits, over temperature, etc.</td></tr>
<tr><td>Process Alarm</td><td>Alarm, if important events occur in the process.</td></tr>
<tr><td>Pull & Plug Alarm</td><td>Alarm, if a module is removed or inserted.</td></tr>
</table>

Table 51: Alarms Tab in the Properties Dialog Box for the PROFIBUS DP Slave

4.6.3.5 The Data Tab

This tab specifies details about the supported data lengths and about the user data (extended configuration data).

Element	Description
Max. Input Len	Maximum length of the input data.
Max. Output Len	Maximum length of the output data.
Max. Data Len	Maximum total length of the input and output data.
User Data Size	Length of the user data.
User Data	Configuration data. HIMA does not recommend editing at this level. Use the dialog box for configuring the user parameters instead, Chapter 4.8.
Max. Diag. Data Len	Maximum length of the diagnostic data sent by the slave.

Table 52: Data Tab in the Properties Dialog Box for the PROFIBUS DP Slave

4.6.3.6 The Model Tab

This tab displays self-explanatory details.

Element	Description
Model	Manufacturer identification of the PROFIBUS DP slave.
Manufacturer	Manufacturer of the field device.
Ident Number	Slave identification provided by PNO Germany.
Revision	Revision of the PROFIBUS DP slave.
Hardware Release	Hardware revision status of the PROFIBUS DP slave.
Software Release	Software revision of the PROFIBUS DP slave.
GSD File Name	File name of the GSD file.
Info Text	Additional details about the PROFIBUS DP slave.

Table 53: The Model Tab in the Properties Dialog Box for the PROFIBUS DP Slave

4.6.3.7 The Features Tab

Element	Description
Modular Station	TRUE: Modular station FALSE: Compact station
First Configurable Slot Number	The modules (slots) must be numbered without gaps, starting with this value.
Max Modules	Maximum number of modules that can be installed in a modular station.
Support for Set Slave Add	The slave supports dynamic address allocation.
Min. Slave Interval [ms]	The minimum time period that must elapse between 2 cyclic calls of the slave.
Diag Update	Number of polling cycles until the slave's diagnosis mirrors the current state.
Support for WDBase1ms	The slave supports 1 ms as time unit for the watchdogs.
Support for DP-V0 Sync	The slave supports DP-V0 Sync.
Support for DP-V0 Freeze	The slave supports DP-V0 Freeze.
DP-V1 Data Types	The slave supports the DP-V1 data types.
Extra Alarm SAP	The slave supports SAP 50 for acknowledging the alarm.
Alarm Seq. Mode Count	Indicates the number of active alarms that the slave can simultaneously process. Zero means one alarm of each model.

Table 54: Features Tab in the Properties Dialog Box for the PROFIBUS DP Slave

4.6.3.8 The Baud Rates Tab

This tab specifies the baud rates that the slave supports and the corresponding *MaxTsdr*.

MaxTsdr is the time within which the slave must acknowledge a request from the master. The range of values is dependent on the slave and the transfer rate, and is between 15 and 800 Tbit.

Element	Description
9.6k	MaxTsdr = 60
19.2k	MaxTsdr = 60
31.25k	Not supported.
45.45k	MaxTsdr = 60
93.75k	MaxTsdr = 60
187.5k	MaxTsdr = 60
500k	MaxTsdr = 70
1.5M	MaxTsdr = 75
3M	MaxTsdr = 90
6M	MaxTsdr = 100
12M	MaxTsdr = 120

Table 55: Baud Rates Tab in the Properties Dialog Box for the PROFIBUS DP Slave

4.6.3.9 The Acyclic Tab

This tab contains some parameters for the acyclic data transmission.

Element	Description
Support for C1 Read/Write	The slave supports acyclic data transmission.
C1 Read/Write required	The slave requires the acyclic data transmission.
C1 Max Data Len[Byte]	Maximum length of an acyclic data packet.
C1 Response Timeout [ms]	Time out for the acyclic data transmission.

Table 56: Acyclic Tab in the Properties Dialog Box for the PROFIBUS DP Slave

4.7 Importing GSD File

The GSD file contains data for configuring the PROFIBUS DP slave.

To read the GSD file for the new PROFIBUS DP slave

1. In the structure tree, open **Configuration**, **Resource**, **Protocols**, **PROFIBUS DP Master**, **PROFIBUS Slave.**
2. Select **Read GSD File** from the context menu of the PROFIBUS DP master and choose the GSD file for the PROFIBUS slave (e.g., hax100ea.gsd).

i The GSD files for HIMA controllers are available on the HIMA website at www.hima.com.
The manufacturer of the field device is responsible for the correctness of the GSD file.

The GSD files for HIMax (hax100ea.gsd) and HIMatrix (hix100ea.gsd) include the following modules:

PROFIBUS DP master input modules	Type	Number
DP input/ELOP export	Byte	1
DP input/ELOP export	Bytes	2
DP input/ELOP export	Bytes	4
DP input/ELOP export	Bytes	8
DP input/ELOP export	Bytes	16
DP input/ELOP export	Word	1
DP input/ELOP export	Words	2
DP input/ELOP export	Words	4
DP input/ELOP export	Words	8
DP input/ELOP export	Words	16
PROFIBUS DP master output modules	Type	Number
DP output/ELOP import	Byte	1
DP output/ELOP import	Bytes	2
DP output/ELOP import	Bytes	4
DP output/ELOP import	Bytes	8
DP output/ELOP import	Bytes	16
DP output/ELOP import	Word	1
DP output/ELOP import	Words	2
DP output/ELOP import	Words	4
DP output/ELOP import	Words	8
DP output/ELOP import	Words	16

Table 57: GSD File of the HIMA PROFIBUS DP Slave

4.8 Configuring the User Parameters

The group's **Start Address** and **Number of Variables** are defined in the user data field.

Additionally, the number of bytes that should actually be transferred must be configured in the PROFIBUS DP master. This is done by choosing the PROFIBUS DP modules from the GDS file of the PROFIBUS DP slave, see also Chapter 4.2.2.

To open the Edit User Parameters dialog box

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master.**
2. Right-click **PROFINET Slave** and select **Properties** from the context menu.
3. Select the **Data** tab and click the **...** button next to the user data.

i The structure of the **Edit User Parameters** dialog box depends on the GSD file of the slave.

Structure of the 32-byte user data field

The 32-byte user data field is structured as follows:

The 32 bytes are allocated in **8 groups**, with **4 bytes each**.

Groups 1…4 define which and how many variables the PROFIBUS DP master receives from the PROFIBUS DP slave.

Groups 5…8 define which and how many variables the PROFIBUS DP master sends to the PROFIBUS DP slave.

The first 2 bytes of each group specify the start address of the first variables to be read or written to.

The last 2 bytes in each group specify the number of variables that should be received or sent.

To configure the user data in different groups

Usually, it is not necessary to allocate the variables (user data) into the various groups. It is enough to define the first variable group of the input and output variables, and to read or write the data *en bloc*.

In applications requiring that only selected variables are read and written, up to 4 variable groups can be defined for the input and output variables.

Example

The PROFIBUS DP master sends and receives the following variables from the PROFIBUS DP slave:

Group 1: **4** input variables from start address **0** and up.

Group 2: **6** input variables from start address **50** and up.

Group 4: **9** input variables from start address **100** and up.

Group 5: **2** output variables from start address **10** and up.

User data configuration in the PROFIBUS DP master:

Master import/slave export	Start address	Number of variables
Group 1 (bytes 0…3)	0.0	0.4
Group 2 (bytes 4...7)	0.50	0.6
Group 3 (bytes 8…11)	0.0	0.0
Group 4 (bytes 12…15)	0.100	0.9

Table 58: Example: Groups 1…4 of the User Data Field

Master export/slave import	Start address	Number of variables
Group 5 (bytes 16…19)	0.10	0.2
Group 6 (bytes 20…23)	0.0	0.0
Group 7 (bytes 24…27)	0.0	0.0
Group 8 (bytes 28...31)	0.0	0.0

Table 59: Example: Groups 1…4 of the User Data Field

Edit User Parameters dialog box of a HIMatrix or HIMax PROFIBUS DP Slave.

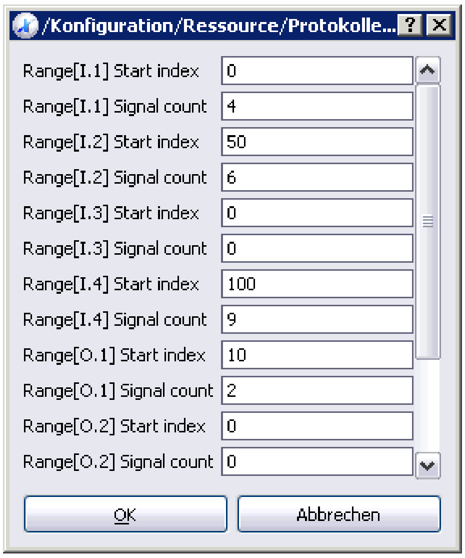


Figure 17: *Edit User Parameters* Dialog Box

4.9 PROFIBUS Function Blocks

The PROFIBUS function blocks are used to tailor the HIMA PROFIBUS DP master and the corresponding PROFIBUS DP slaves to best meet the project requirements.

The function blocks are configured in the user program such that the master and slave functions (alarms, diagnostic data, and states) can be set and read in the user program.

i Function blocks are required for special applications. They are not needed for the normal cyclic data traffic between master and slave!

For further details on the conceptual configuration of the PROFIBUS DP function blocks, refer to Chapter 5.3.

The following function blocks are available:

Function block	Function description	Suitable beginning with stage of extension DP
MSTAT 4.9.1	For controlling the master state using the user program.	DP-V0
RALRM 4.9.2	For reading the alarm messages of the slaves.	DP-V1
RDIAG 4.9.3	For reading the diagnostic messages from the slaves.	DP-V0
RDREC 4.9.4	For reading the acyclic data records of the slaves.	DP-V1
SLACT 4.9.5	For controlling the slave states using the user program.	DP-V0
WRREC 4.9.6	For writing the acyclic data records of the slaves.	DP-V1

Table 60: Overview of the PROFIBUS DP Function Blocks

i HIMA PROFIBUS DP masters operate with stage of extension DP-V1.

HIMA PROFIBUS DP slaves operate with stage of extension DP-V0.

For this reason, not all function blocks of the HIMA PROFIBUS DP master can be used to control a HIMA PROFIBUS DP slaves.

4.9.1 MSTAT Function Block

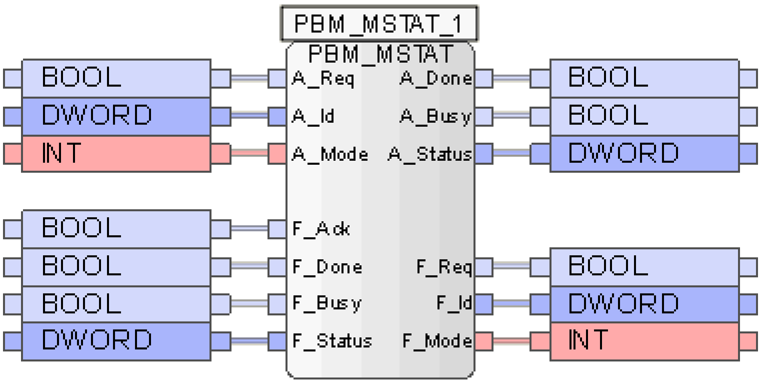


Figure 18: MSTAT Function Block

The user program uses the **MSTAT** function block (DP-V0 and higher) to control the PROFIBUS DP master. The PROFIBUS DP master can thus be set to one of the following operating states using a timer or a mechanical switch connected to a physical input:

- 0: OFFLINE
- 1: STOP
- 2: CLEAR
- 3: OPERATE

The **MSTAT** function block can be used for controlling the PROFIBUS slaves with the additional functions FREEZE, UNFREEZE, SYNC and UNSYNC. Additionally, it can be used for selecting the membership of the slave to be controlled.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs and outputs of the function block with prefix A

These inputs and outputs can be used to control and evaluate the function block using the user program. The prefix A means Application.

A-Inputs	Description	Type
A_Req	The rising edge triggers the function block.	BOOL
A_Id	Master ID (not used).	DWORD
A_Mode	**Modes of Operation for the PROFIBUS DP Master** The PROFIBUS DP master can be set to the following modes of operation: The least significant byte sets the modes of operation. (most significant byte=0): 0x00**00** - OFFLINE 0x00**01** - STOP 0x00**02** - CLEAR 0x00**03** - OPERATE	INT

A-Inputs	Description	Type
	Additional Information about PROFIBUS Slave Alternatively, additional functions can be set for the PROFIBUS slaves connected to this PROFIBUS master. The most significant byte set the additional functions: 0x**04**XX - Unfreeze 0x**08**XX - Freeze 0x**10**XX - Unsync 0x**20**XX - Sync The least significant byte describes the group(s) to which additional functions apply: 0xXX**00 -** for all slaves 0xXX**01** - Group 1 0xXX**02** - Group 2 0xXX**04** - Group 3 … 0xXX**80** - Group 8 Example: To send Sync+Freeze to all the groups, Mode must be set to 0x28ff.	

Table 61: A-Inputs for the MSTAT Function Block

Programming example for setting the A_Mode input

To provide a clearer exposition, recommends using an auxiliary function block for preparing the *A_Mode* input variables of the **MSTAT** function block.

The following figures show a programming example of how to implement this auxiliary function block in the user program.

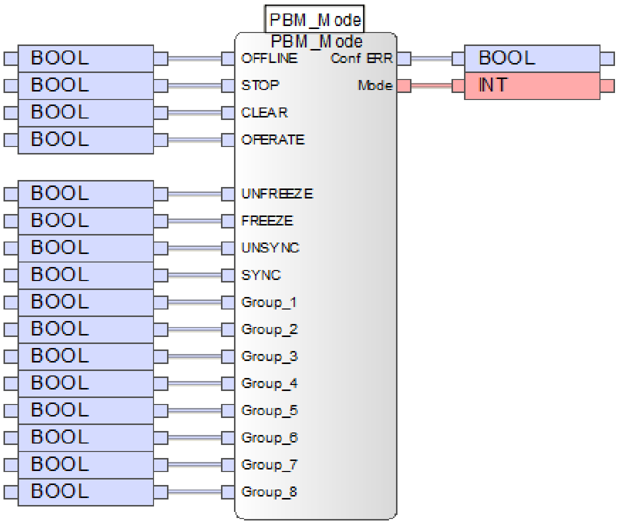


Figure 19: Programming Example for Setting the A_Mode Input

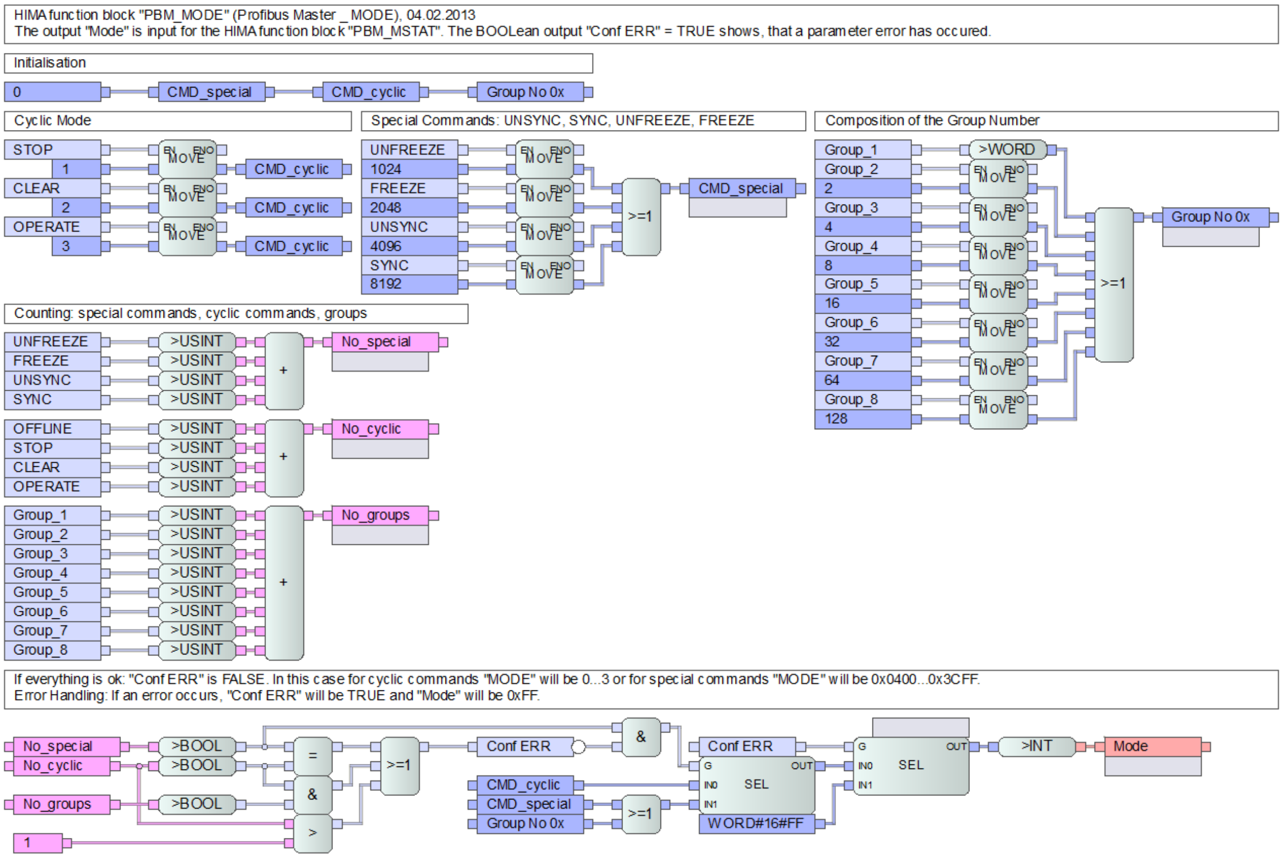


Figure 20: Suggestion for Programming the Logic of the PBM_Mode Auxiliary Function Block

A-Outputs	Description	Type
A_Done	TRUE: The PROFIBUS DP master has been set to the state defined on the A_Mode input.	BOOL
A_Busy	TRUE: The PROFIBUS DP master is still being set.	BOOL
A_Status	Status or error code, see Chapter 4.11.	DWORD

Table 62: A-Outputs for the MSTAT Function Block

Inputs and outputs of the function block with prefix F

These inputs and outputs of the function block establish the connection to the MSTAT function block in the structure tree. The prefix F means Field.

i Shared variables are used to connect the MSTAT function block (in the Blocks directory) to the MSTAT function block (in the user program). These must have been previously created in the Variable Editor.

Connect the *F-Inputs* of the **MSTAT** function block in the user program to the same variables that will be connected to the outputs of the **MSTAT** function block in the structure tree.

F-Inputs	Type
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Status	DWORD

Table 63: F-Inputs for the MSTAT Function Block

Connect the *F-Outputs* of the **MSTAT** function block in the user program to the same variables that will be connected to the inputs of the **MSTAT** function block in the structure tree.

F-Outputs	Type
F_Req	BOOL
F_ID	DWORD
F_MODE	INT

Table 64: F-Outputs for the MSTAT Function Block

To create the MSTAT function block in the structure tree

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master, Blocks, New.**
2. Select the **MSTAT** function block.
3. Right-click the **MSTAT** function block, and then click **Edit**.
   ☑ The window for assigning variables to the function blocks appears.

Connect the inputs of the **MSTAT** function block in the structure tree to the same variables that have been previously connected to the *F-Outputs* of the **MSTAT** function block in the user program.

Inputs	Type
M_ID	DWORD
MODE	INT
REQ	BOOL

Table 65: Input System Variables

Connect the outputs of the **MSTAT** function block in the structure tree to the same variables that have been previously connected to the *F-Inputs* of the **MSTAT** function block in the user program.

Outputs	Type
ACK	BOOL
Busy	BOOL
DONE	BOOL
STATUS	DWORD

Table 66: Output System Variables

To operate the MSTAT function block

1. In the user program, set the *A_Mode* input to the desired state.
   If *A_Mode* is not set, an error code is issued after step 2 on the *A_Status* output and the PROFIBUS DP master state is not set.
2. In the user program, set the *A_Req* input to TRUE.

i The function block responds to a rising edge at *A_Req*.

☑ The *A_Busy* output is TRUE until the MSTAT command has been processed. Afterwards, *A_Busy* is set to FALSE and *A_Done* is set to TRUE.

i If the preset mode could not be set successfully, an error code is output to *A_Status*.
The mode of the current master can be derived from the Master State variable. Refer to Chapter 4.10 for details.

4.9.2 RALRM Function Block

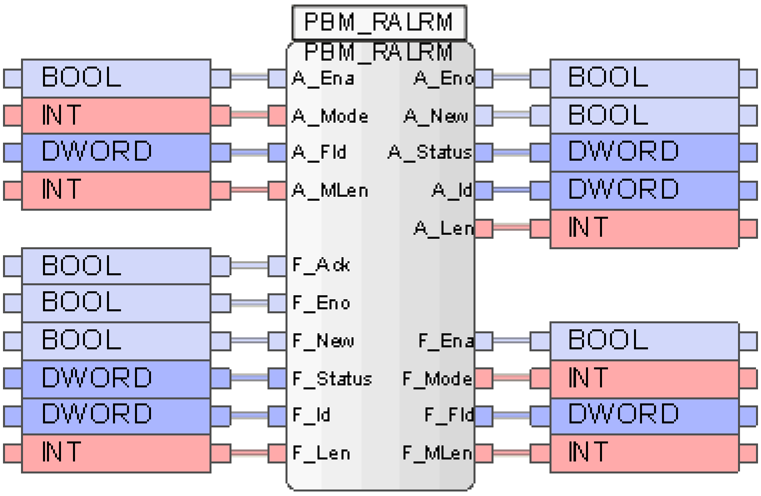


Figure 21: RALRM Function Block

The **RALRM** function block (DP-V1 and higher) is used to evaluate the alarms.

Alarms are a special type of diagnostic messages that are handled with a high priority. Alarms report important events to which the application must react (e.g., a WRREC). How the application reacts, however, depends on the manufacturer. Refer to the manual of the PROFIBUS DP slave for more information.

As long as the **RALRM** function block is active, it waits for alarm messages from the slaves. If an alarm is received, the *A_NEW* output is set to TRUE for at least one cycle and the alarm data can be read from an alarm telegram. Before the next alarm is received, *A_NEW* is set to FALSE for at least one cycle. All alarms are acknowledged implicitly. No alarms are lost.

If multiple **RALRM** function blocks are used, the user program must be configured such that only one **RALRM** function block is active at any given time.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs and outputs of the function block with prefix A

These inputs and outputs can be used to control and evaluate the function block using the user program. The prefix A means Application.

A-Inputs	Description	Type
A_Ena	TRUE enables the function block.	BOOL
A_Mode	Not used	INT
A_FId	Not used	DWORD
A_MLen	Maximum expected length of the received alarm data expressed in bytes.	INT

Table 67: A-Inputs for the RALRM Function Block

A-Outputs	Description	Type
A_Eno	TRUE: The function block is active. FALSE: The function block is not active.	BOOL
A_New	TRUE: New alarm was received. FALSE: No new alarm.	BOOL
A_Status	Status or error code, see Chapter 4.11.	DWORD
A_Id	Identification number of the slave triggering the alarm.	DWORD
A_Len	Length of the received alarm data in bytes.	INT

Table 68: A-Outputs for the RDIAG Function Block

Inputs and outputs of the function block with prefix F

These inputs and outputs of the function block establish the connection to the RALRM function block in the structure tree. The prefix F means Field.

i Shared variables are used to connect the **RALRM** function block (in the Blocks directory) to the **RALRM** function block (in the user program). These must have been previously created in the Variable Editor.

Connect the *F-Inputs* of the **RALRM** function block in the user program to the same variables that will be connected to the outputs of the **RALRM** function block in the structure tree.

F-Inputs	Type
F_Ack	BOOL
F_Eno	BOOL
F_New	BOOL
F_Status	DWORD
F_ID	DWORD
F_Len	INT

Table 69: F-Inputs for the RALRM Function Block

Connect the *F-Outputs* of the **RALRM** function block in the user program to the same variables that will be connected to the inputs of the **RLARM** function block in the structure tree.

F-Outputs	Type
F_Ena	BOOL
F_MODE	INT
F_FId	DWORD
F_Mlen	INT

Table 70: F-Outputs for the RALRM Function Block

To create the RALRM function block in the structure tree

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master, Blocks, New.**
2. Select the **RALRM** function block.
3. Right-click the **RALRM** function block, and then click **Edit**.
   ☑ The window for assigning variables to the function blocks appears.

Connect the inputs of the **RALRM** function block in the structure tree to the same variables that have been previously connected to the *F-Outputs* of the **RALRM** function block in the user program.

Inputs	Type
EN	BOOL
F_ID	DWORD
MLEN	INT
MODE	INT

Table 71: Input System Variables

Connect the outputs of the **RALRM** function block in the structure tree to the same variables that have been previously connected to the *F-Inputs* of the **RALRM** function block in the user program.

Outputs	Type
ACK	BOOL
ENO	BOOL
ID	DWORD
LEN	INT
NEW	BOOL
STATUS	DWORD

Table 72: Output System Variables

The Process Variables tab of the **RALRM** function block located in the structure tree contains variables that must be defined and whose structure must match the alarm data. If no variables are defined, alarm data can be requested but not read.

An alarm message contains at least 4 bytes. The first four bytes of the alarm message contain the standard alarm data.

To decode standard alarms, HIMA provides the **ALARM** auxiliary function block. Refer to Chapter 4.10 for details.

i If an alarm telegram contains more bytes than defined in the Data tab, only the preset number of bytes is accepted. The rest is cut off.

Alarm data	Description
Byte 0	Length of the alarm message expressed in bytes (4…126)
Byte 1	Identifier for the alarm type 1: Diagnostic alarm 2: Process alarm 3: Pull alarm 4: Plug alarm 5: Status alarm 6: Update alarm 31: Failure of a master's or a slave's extension 32…126: Manufacturer-specific Consult the device manufacturer specifications for details.
Byte 2	Slot number of the component triggering the alarm
Byte 3	0: No further information 1: Inbound alarm, slot malfunction 2: Outbound alarm, slot no longer malfunctioning 3: Outbound alarm, slot still malfunctioning
Bytes 4…126	Consult the device manufacturer specifications for details.

Table 73: Alarm Data

i The structure of the standard alarms (bytes 0…3) is normalized and identical for all manufacturers. Consult the manual of the PROFIBUS DP slave for more information on bytes 4…126, since their use is manufacturer-specific.

Devices built in accordance with the DP-V0 standard do not support alarm telegrams.

To operate the RALRM function block

1. At the user program input *A_Mlen*, define the maximum amount of alarm data to be expected, expressed in bytes. *A_Mlen* cannot be changed during operation.
2. In the user program, set the *A_Ena* input to TRUE.

i In contrast to other function blocks, the **RALRM** function block is only active as long as the *A_Ena* input is set to TRUE.

If the function block was started successfully, the *A_Eno* output is set to TRUE. If the function block could not be started, an error code is output to *A_Status*.

If a new alarm is received, the *A_New* output is set to TRUE for at least one cycle. During this time period, the alarm data of the slave triggering the alarm are contained in the outputs and can be evaluated.

Afterwards, the *A_New* output returns to FALSE for at least one cycle. The *A_Id* and *A_Len* outputs are reset to zero before the next alarm message can be received and evaluated.

4.9.3 RDIAG Function Block

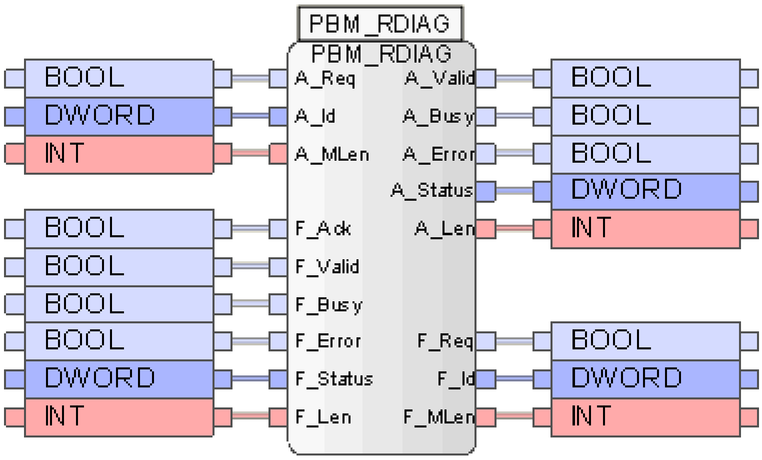


Figure 22: RDIAG Function Block

The **RDIAG** function block (DP-V0 and higher) is used for reading the current diagnostic message of a slave (6…240 bytes).

The number of **RDIAG** function blocks simultaneously active in the HIMA PROFIBUS DP master can be defined by the user.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs and outputs of the function block with prefix A

These inputs and outputs can be used to control and evaluate the function block using the user program. The prefix A means Application.

A-Inputs	Description	Type
A_Req	The rising edge triggers the reading request of a diagnostic message.	BOOL
A_Id	Slave identification number, see Chapter 4.10.	DWORD
A_MLen	Maximum expected length of the diagnostic message to be read, expressed in bytes.	INT

Table 74: A-Inputs for the RDIAG Function Block

A-Outputs	Description	Type
A_Valid	A new diagnostic message has been received and is valid.	BOOL
A_Busy	TRUE: Data is still being read.	BOOL
A_Error	TRUE: An error occurred during the reading process.	BOOL
A_Status	Status or error code, see Chapter 4.11.	DWORD
A_Len	Length of the read diagnostic data, expressed in bytes.	INT

Table 75: F-Outputs for the RDIAG Function Block

Inputs and outputs of the function block with prefix F

These inputs and outputs of the function block establish the connection to the **RDIAG** function block in the structure tree. The prefix F means Field.

i Shared variables are used to connect the **RDIAG** function block (in the Blocks directory) to the **RDIAG** function block (in the user program). These must have been previously created in the Variable Editor.

Connect the *F-Inputs* of the **RDIAG** function block in the user program to the same variables that will be connected to the outputs of the **RDIAG** function block in the structure tree.

F-Inputs	Type
F_Ack	BOOL
F_Valid	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Table 76: F-Inputs for the RDIAG Function Block

Connect the *F-Outputs* of the **RDIAG** function block in the user program to the same variables that will be connected to the inputs of the **RDIAG** function block in the structure tree.

F-Outputs	Type
F_Req	BOOL
F_ID	DWORD
F_Mlen	INT

Table 77: F-Outputs for the RDIAG Function Block

To create the RDIAG function block in the structure tree

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master, Blocks, New.**
2. Select the **RDIAG** function block.
3. Right-click the **RDIAG** function block, and then click **Edit**.
   ☑ The window for assigning variables to the function blocks appears.

Connect the inputs of the **RDIAG** function block in the structure tree to the same variables that have been previously connected to the *F-Outputs* of the **RDIAG** function block in the user program.

Inputs	Type
ID	DWORD
MLEN	INT
REQ	BOOL

Table 78: Input System Variables

Connect the outputs of the **RDIAG** function block in the structure tree to the same variables that have been previously connected to the *F-Inputs* of the **RDIAG** function block in the user program.

Outputs	Type
ACK	BOOL
Busy	BOOL
Error	BOOL
LEN	INT
Status	DWORD
VALID	BOOL

Table 79: Output System Variables

Diagnostic data

The structure of the variables defined in the **Data** tab must match the diagnostic data. A diagnostic message contains a minimum of 6 and a maximum of 240 bytes. The first 4 bytes of the diagnostic message contain the standard diagnostics.

To decode standard diagnostics, HIMA provides the **STDDIAG** auxiliary function block. Refer to Chapter 4.10 for details.

i If a diagnostic telegram contains more bytes than defined in the **Data** tab, only the preset number of bytes is accepted. The rest is cut off.

Diagnostic data	Description
Byte 0	Bytes 0…3 contain the standard diagnostics. Use the **STDDIAG** auxiliary function block to decode the standard diagnostics as a variable of type DWORD.
Byte 1	
Byte 2	
Byte 3	Bus address of the master to which a slave is assigned.
Byte 4	High byte (manufacturer ID)
Byte 5	Low byte (manufacturer ID)
Bytes 6…240	Consult the device manufacturer specifications for details.

Table 80: Diagnostic Data

i The HIMA slaves send a diagnostic telegram of six bytes in length. The meaning of these bytes is standardized.

For slaves from other manufacturers, only the first six bytes are functionally identical.

For further details on the diagnostic telegram, refer to the description of the slave provided by the manufacturer.

To operate the RDIAG function block

1. In the user program, enter the slave address at the *A_ID* input.
2. At the user program input *A_Mlen*, define the maximum amount of diagnostic data to be expected, expressed in bytes.
3. In the user program, set the *A_Req* input to TRUE.

i The function block responds to a rising edge at *A_Req*.

The *A_Busy* output is set to TRUE until the diagnostic request has been processed. Afterwards, *A_Busy* is set to FALSE and *A_Valid* or *A_Error* to TRUE.

If the diagnostic telegram is valid, the *A_Valid* output is set to TRUE. The diagnostic data can be evaluated using the variables defined on the *Data* tab. The *A_Len* output contains the number of bytes that were actually read out.

If the diagnostic telegram could not be read successfully, the *A_Error* output is set to TRUE and an error code is output to *A_Status*.

4.9.4 RDREC Function Block

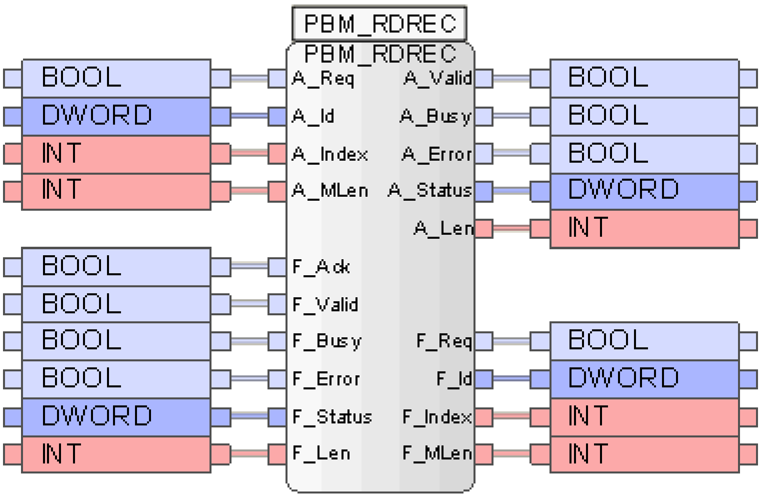


Figure 23: RDREC Function Block

The **RDREC** function block is used for acyclically reading a data record from a slave addressed on the *A_Index* input. Consult the slave's manual to find out which data can be read.

This functionality is optional and is only defined with DP-V1 and higher!

Up to 32 **RDREC** and/or **WRREC** function blocks can simultaneously be active in the HIMA PROFIBUS DP master.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs and outputs of the function block with prefix A

These inputs and outputs can be used to control and evaluate the function block using the user program. The prefix A means Application.

A-Inputs	Description	Type
A_Req	The rising edge triggers the reading request.	BOOL
A_Id	Slave identification number, see Chapter 4.10.	DWORD
A_Index	Number of the data record to be read. Consult the device manufacturer specifications for details.	INT
A_MLen	Maximum length of the data to be read in bytes.	INT

Table 81: A-Inputs for the RDREC Function Block

A-Outputs	Description	Type
A_Valid	A new data record was received and is valid.	BOOL
A_Busy	TRUE: Data is still being read.	BOOL
A_Error	TRUE: An error occurred FALSE: No error.	BOOL
A_Status	Status or error code, see Chapter 4.11.	DWORD
A_Len	Length of the read data record information in bytes.	INT

Table 82: A-Outputs for the RDREC Function Block

Inputs and outputs of the function block with prefix F

These inputs and outputs of the function block establish the connection to the **RDREC** function block in the structure tree. The prefix F means Field.

i Shared variables are used to connect the **RDREC** function block (in the Blocks directory) to the RDREC function block (in the user program). These must have been previously created in the Variable Editor.

Connect the *F-Inputs* of the **RDREC** function block in the user program to the same variables that will be connected to the outputs of the **RDREC** function block in the structure tree.

F-Inputs	Type
F_Ack	BOOL
F_Valid	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Table 83: F-Inputs for the RDREC Function Block

Connect the *F-Outputs* of the **RDREC** function block in the user program to the same variables that will be connected to the inputs of the **RDREC** function block in the structure tree.

F-Outputs	Type
F_Req	BOOL
F_ID	DWORD
F_Index	INT
F_MLEN	INT

Table 84: F-Outputs for the RDREC Function Block

To create the RDREC function block in the structure tree

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master, Blocks, New.**
2. Select the **RDREC** function block.
3. Right-click the **RDREC** function block, and then click **Edit**.
   ☑ The window for assigning variables to the function blocks appears.

Connect the inputs of the **RDREC** function block in the structure tree to the same variables that have been previously connected to the *F-Outputs* of the **RDREC** function block in the user program.

Inputs	Type
ID	DWORD
INDEX	INT
MLEN	INT
REQ	BOOL

Table 85: Input System Variables

Connect the outputs of the **RDREC** function block in the structure tree to the same variables that have been previously connected to the *F-Inputs* of the **RDREC** function block in the user program.

Outputs	Type
ACK	BOOL
Busy	BOOL
Error	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Table 86: Output System Variables

Data	Description
No predefined variables	A user-specific data structure can be defined in the *Process Variables* tab; however, the structure must match the data record structure. For further details on the record structure, refer to the operating instructions provided by the manufacturer of the slave.

Table 87: Data

To operate the RDREC function block

1. In the user program, enter the slave address at the *A_ID* input.
2. In the user program, set the slave-specific index for the data record at the *A_Index* input (see the manual provided by the manufacturer).
3. In the user program, set the length of the data record to be read, at the *A_Len* input.
4. In the user program, set the *A_Req* input to TRUE.

i The function block responds to a rising edge at *A_Req*.

The *A_Busy* output is set to TRUE until the data record request has been processed. Afterwards, *A_Busy* is set to FALSE and *A_Valid* or *A_Error* to TRUE.

If the data record is valid, the *A_Valid* output is set to TRUE. The data record can be evaluated using the variables defined in the *Data* tab. The *A_Len* output contains the actual length of the data record that has been read.

If the data record could not be read successfully, the *A_Error* output is set to TRUE and an error code is output to *A_Status*.

4.9.5 SLACT Function Block

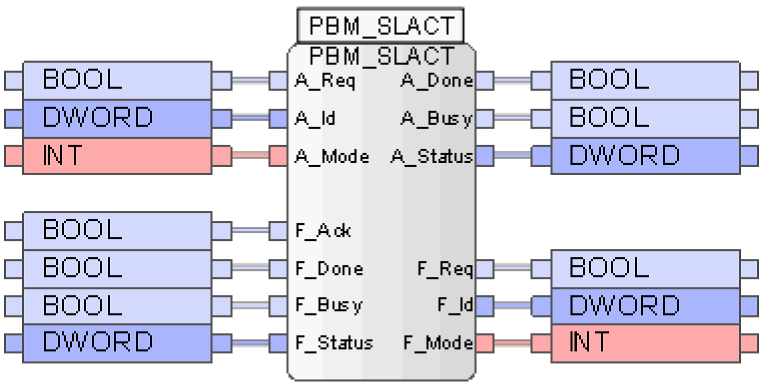


Figure 24: SLACT Function Block

The **SLACT** function block (DP-V0 and higher) is used for activating and deactivating a slave from within the user program of the PROFIBUS DP master. The slave can thus be set to one of the following states using a timer or a mechanical switch connected to a physical input of the PROFIBUS DP master:

- ≠ 0: Active
- = 0: Inactive

If various **SLACT** function blocks are used, the user program must be configured such that only one **SLACT** function block is active at a time.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs and outputs of the function block with prefix A

These inputs and outputs can be used to control and evaluate the function block using the user program. The prefix A means Application.

A-Inputs	Description	Type
A_Req	The rising edge triggers the function block.	BOOL
A_Id	Slave identification number, see Chapter 4.10.	DWORD
A_Mode	Target state for the PROFIBUS DP slave: ≠ 0: active (connected) = 0: Not active (deactivated)	INT

Table 88: A-Inputs for the SLACT Function Block

A-Outputs	Description	Type
A_Done	TRUE: The PROFIBUS DP slave has been set to the state defined on the A_Mode input.	BOOL
A_Busy	TRUE: The PROFIBUS DP slave is still being set.	BOOL
A_Status	Status or error code, see Chapter 4.11.	DWORD

Table 89: A-Outputs for the SLACT Function Block

Inputs and outputs of the function block with prefix F

These inputs and outputs of the function block establish the connection to the **SLACT** function block in the structure tree. The prefix F means Field.

i Shared variables are used to connect the **SLACT** function block (in the Blocks directory) to the **SLACT** function block (in the user program). These must have been previously created in the Variable Editor.

Connect the *F-Inputs* of the **SLACT** function block in the user program to the same variables that will be connected to the outputs of the **SLACT** function block in the structure tree.

F-Inputs	Type
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Status	DWORD

Table 90: F-Inputs for the SLACT Function Block

Connect the *F-Outputs* of the **SLACT** function block in the user program to the same variables that will be connected to the inputs of the **SLACT** function block in the structure tree.

F-Outputs	Type
F_Req	BOOL
F_ID	DWORD
F_MODE	INT

Table 91: F-Outputs for the SLACT Function Block

To create the SLACT function block in the structure tree

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master, Blocks, New.**
2. Select the **SLACT** function block.
3. Right-click the **SLACT** function block, and then click **Edit**.
   ☑ The window for assigning variables to the function blocks appears.

Connect the inputs of the **SLACT** function block in the structure tree to the same variables that have been previously connected to the *F-Outputs* of the **SLACT** function block in the user program.

Inputs	Type
ID	DWORD
MODE	INT
REQ	BOOL

Table 92: Input System Variables

Connect the outputs of the **SLACT** function block in the structure tree to the same variables that have been previously connected to the *F-Inputs* of the **SLACT** function block in the user program.

Outputs	Type
ACK	BOOL
Busy	BOOL
DONE	BOOL
STATUS	DWORD

Table 93: Output System Variables

To operate the SLACT function block

1. In the user program, set the *A_Mode* input to the desired state.
2. In the user program, enter the slave address ID at the *A_ID* input.
3. In the user program, set the *A_Req* input to TRUE.

i The function block responds to a rising edge at *A_Req*.

A_Busy output is set to TRUE until the *SLACT* command has been processed. Afterwards, *A_Busy* is set to FALSE and *A_Done* is set to TRUE.

If the slave mode could be set successfully, it is output to *A_Status*.

If the slave mode could not be set successfully, an error code is output to *A_Status*.

4.9.6 WRREC Function Block

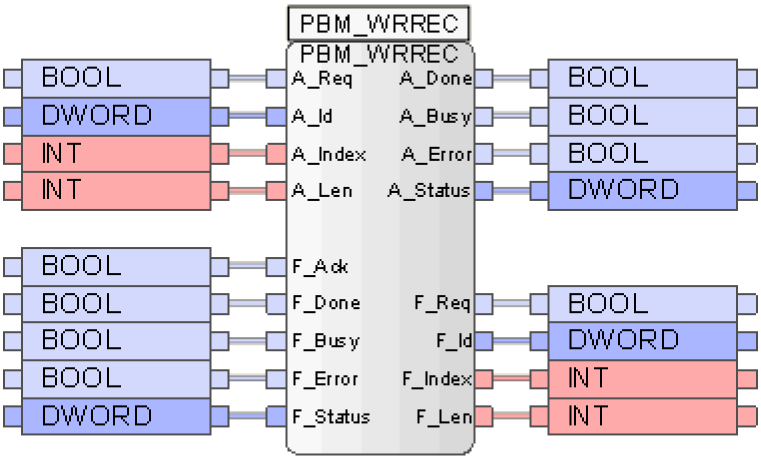


Figure 25: WRREC Function Block

The **WRREC** function block (DP-V1 and higher) is used for acyclically writing a data record to a slave addressed with *A_Index*. Consult the slave's manual to find out which data can be written.

Up to 32 **RDREC** and/or **WRREC** function blocks can simultaneously be active in the HIMA PROFIBUS DP master.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs and outputs of the function block with prefix A

These inputs and outputs can be used to control and evaluate the function block using the user program. The prefix A means Application.

A-Inputs	Description	Type
A_Req	The rising edge triggers the request for writing a data record.	BOOL
A_Id	Slave identification number, see Chapter 4.10.	DWORD
A_Index	Number of the data record to be written. Consult the device manufacturer specifications for details.	INT
A_Len	Length of the data record to be written in bytes.	INT

Table 94: A-Inputs for the WRREC Function Block

A-Outputs	Description	Type
A_Done	TRUE: The function block completed the writing process.	BOOL
A_Busy	TRUE: The function block has not yet completed the writing process	BOOL
A_Error	TRUE: An error occurred	BOOL
A_Status	Status or error code, see Chapter 4.11.	DWORD

Table 95: A-Outputs for the WRREC Function Block

Inputs and outputs of the function block with prefix F

These inputs and outputs of the function block establish the connection to the **WRREC** function block in the structure tree. The prefix F means Field.

i Shared variables are used to connect the WRREC function block (in the Blocks directory) to the WRREC function block (in the user program). These must have been previously created in the Variable Editor.

Connect the *F-Inputs* of the **WRREC** function block in the user program to the same variables that will be connected to the outputs of the **WRREC** function block in the structure tree.

F-Inputs	Type
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD

Table 96: F-Inputs for the WRREC Function Block

Connect the *F-Outputs* of the **WRREC** function block in the user program to the same variables that will be connected to the inputs of the **WRREC** function block in the structure tree.

F-Outputs	Type
F_Req	BOOL
F_ID	DWORD
F_Index	INT
F_Len	INT

Table 97: F-Outputs for the WRREC Function Block

To create the WRREC function block in the structure tree

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master, Blocks, New.**
2. Select the **WRREC** function block.
3. Right-click the **WRREC** function block, and then click **Edit**.
   ☑ The window for assigning variables to the function blocks appears.

Connect the inputs of the **WRREC** function block in the structure tree to the same variables that have been previously connected to the *F-Outputs* of the **WRREC** function block in the user program.

Inputs	Type
ID	DWORD
INDEX	INT
LEN	INT
REQ	BOOL

Table 98: Input System Variables

Connect the outputs of the **WRREC** function block in the structure tree to the same variables that have been previously connected to the *F-Inputs* of the **WRREC** function block in the user program.

Outputs	Type
ACK	BOOL
Busy	BOOL
DONE	BOOL
Error	BOOL
STATUS	DWORD

Table 99: Output System Variables

Data	Description
No predefined variables	A user-specific data structure can be defined in the *Process Variables* tab; however, the structure must match the data record structure. For further details on the record structure, refer to the operating instructions provided by the manufacturer of the slave.

Table 100: Data

To operate the WRREC function block, the following steps are essential

1. In the user program, enter the slave address at the *A_ID* input.
2. In the user program, set the slave-specific index for the data record at the *A_Index* input (see the manual provided by the manufacturer).
3. In the user program, set the length of the data record to be written, at the *A_Len* input.
4. In the user program, set the data record as defined in the Data tab.In the user program, set the data record as defined in the Data tab.
5. In the user program, set the *A_Req* input to TRUE.

i The function block responds to a rising edge at *A_Req*.

The *A_Busy* output is TRUE until the data record has been written. Afterwards, *A_Busy* is set to FALSE and *A_Done* is set to TRUE.

If the data record could not be written successfully, the *A_Error* output is set to TRUE and an error code is output to *A_Status*.

4.10 PROFIBUS Auxiliary Function Blocks

The auxiliary function blocks are used to configure and evaluate the inputs and outputs of the function blocks.

The following auxiliary function blocks are available:

Auxiliary function blocks	Function description
ACTIVE (see Chapter 4.10.1)	Slave active or inactive.
ALARM (see Chapter 4.10.2)	Decodes the alarm data.
DEID (see Chapter 4.10.3)	Decodes the identification number.
ID (see Chapter 4.10.4)	Generates a 4-byte identifier.
NSLOT (see Chapter 4.10.5)	Creates a continuous identification number for the slots.
SLOT (see Chapter 4.10.6)	Creates a SLOT identification number based on a slot number.
STDDIAG (see Chapter 4.10.7)	Decodes the standard diagnostics of a slave.
LATCH	Only used within other function blocks.
PIG	Only used within other function blocks.
PIGII	Only used within other function blocks.

Table 101: Overview of the Auxiliary Function Blocks

4.10.1 Auxiliary Function Block: ACTIVE

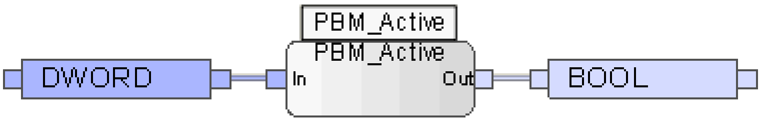


Figure 26: The ACTIVE Auxiliary Function Block

The ACTIVE auxiliary function block determines if the slave is active or inactive based on the standard diagnostics of a PROFIBUS DP slave.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs	Description	Type
IN	Slave standard diagnostics.	DWORD

Table 102: Inputs for the ACTIVE Auxiliary Function Block

Outputs	Description	Type
OUT	TRUE: Slave is active. FALSE: Slave is inactive.	BOOL

Table 103: Outputs for the ACTIVE Auxiliary Function Block

4.10.2 Auxiliary Function Block: ALARM

(Decode the alarm data)

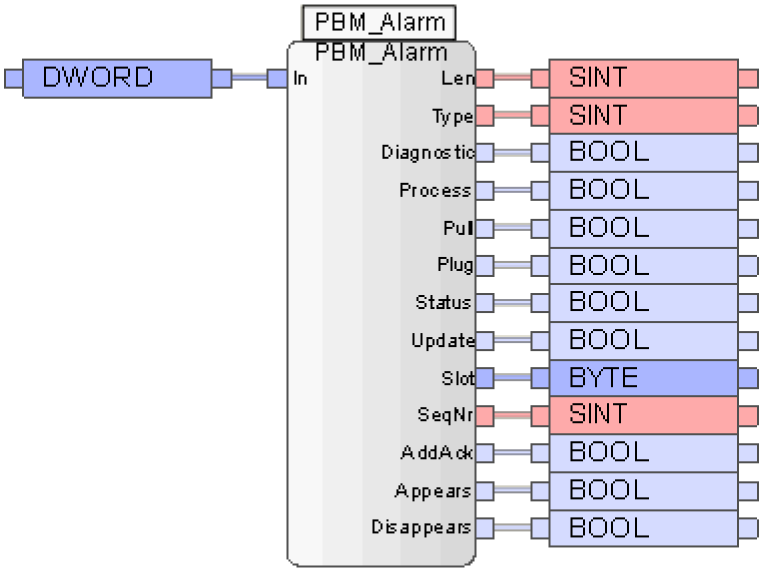


Figure 27: The ALARM Auxiliary Function Block

The **ALARM** auxiliary function block decodes the standard alarm data of a PROFIBUS DP slave.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs	Description	Type
IN	Standard alarm	DWORD

Table 104: Inputs for the ALARM Auxiliary Function Block

Output	Description	Type
LEN	Total length of the alarm message.	SINT
Type	1: Diagnostic alarm 2: Process alarm 3: Pull alarm 4: Plug alarm 5: Status alarm 6: Update alarm The other numbers are either reserved or manufacturer-specific. Consult the device manufacturer specifications for details.	SINT
Diagnostic	TRUE = Diagnostic alarm	BOOL
Process	TRUE = Process alarm	BOOL
Pull	TRUE = The module was pulled	BOOL
Plug	TRUE = The module was plugged in again	BOOL
Status	TRUE = Status alarm	BOOL
Update	TRUE = Update alarm	BOOL
Slot	Alarm-triggering module.	BYTE
SeqNr	Alarm sequence number.	SINT

<table>
<tr><th>Output</th><th>Description</th><th>Type</th></tr>
<tr><td>AddAck</td><td>TRUE means that the slave that triggered this alarm expects an additional acknowledgement from the application. For detailed information, refer to the slave-specific manual provided by the manufacturer.</td><td>BOOL</td></tr>
<tr><td>Appears
Disappears</td><td>
<table>
<tr><th>Output</th><th>Value</th><th>Description</th></tr>
<tr><td>Appears</td><td>TRUE</td><td rowspan="2">If both are FALSE, no error has occurred yet.</td></tr>
<tr><td>Disappears</td><td>FALSE</td></tr>
<tr><td>Appears</td><td>TRUE</td><td rowspan="2">An error occurred and is still present.</td></tr>
<tr><td>Disappears</td><td>FALSE</td></tr>
<tr><td>Appears</td><td>TRUE</td><td rowspan="2">An error occurred and is disappearing.</td></tr>
<tr><td>Disappears</td><td>FALSE</td></tr>
<tr><td>Appears</td><td>TRUE</td><td rowspan="2">If both are TRUE, the fault has disappeared, but the slave is still in a faulty state.</td></tr>
<tr><td>Disappears</td><td>FALSE</td></tr>
</table>
</td><td>BOOL</td></tr>
</table>

Table 105: Outputs for the ALARM Auxiliary Function Block

4.10.3 Auxiliary Function Block: DEID

(Decode the identification number)

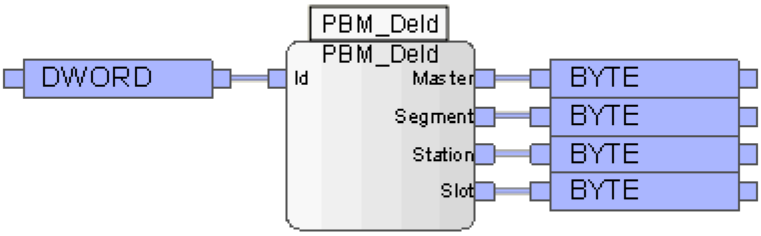


Figure 28: The DEID Auxiliary Function Block

The **DEID** auxiliary function block decodes the identification number and disjoints it into its 4 component parts.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs	Description	Type
ID	Identification number of the slave.	DWORD

Table 106: Inputs for the DEID Auxiliary Function Block

Outputs	Description	Type
Master	Master bus address.	BYTE
Segment	Segment	BYTE
Station	Slave bus address.	BYTE
Slot	Slot or module number.	BYTE

Table 107: Outputs for the DEID Auxiliary Function Block

4.10.4 Auxiliary Function Block: ID

(Generate the identification number)

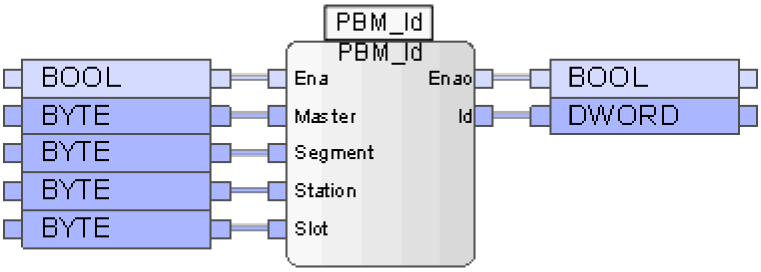


Figure 29: The ID Auxiliary Function Block

The **ID** auxiliary function block generates an identifier (identification number) used by other auxiliary function blocks based on 4 bytes.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs	Description	Type
Ena	Not used	BOOL
Master	Bus address	BYTE
Segment	Segment	BYTE
Station	Slave bus address	BYTE
Slot	Slot or module number	BYTE

Table 108: Inputs for the ID Auxiliary Function Block

Outputs	Description	Type
Enao	Not used	BOOL
ID	Identification number of the slave	DWORD

Table 109: Outputs for the ID Auxiliary Function Block

4.10.5 Auxiliary Function Block: NSLOT

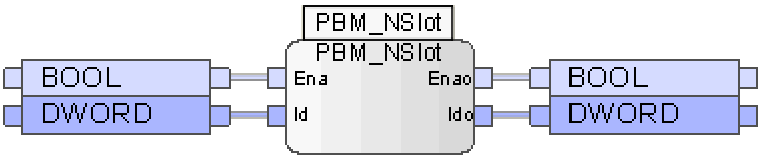


Figure 30: The NSLOT Auxiliary Function Block

The **NSLOT** auxiliary function block uses an identifier to generate a new identifier that addresses the next slot within the same slave. Ena must be set to TRUE to allow the auxiliary function block to run.

Enao is set to TRUE if the result at the Ido output is valid.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs	Description	Type
Ena	The auxiliary function block runs as long as Ena is TRUE	BOOL
ID	Identification number of the slave	DWORD

Table 110: Inputs for the NSLOT Auxiliary Function Block

Outputs	Description	Type
Enao	TRUE = The result is valid FALSE = No other slot numbers	BOOL
Ido	Identification number of the slave	DWORD

Table 111: Outputs for the NSLOT Auxiliary Function Block

4.10.6 Auxiliary Function Block: SLOT

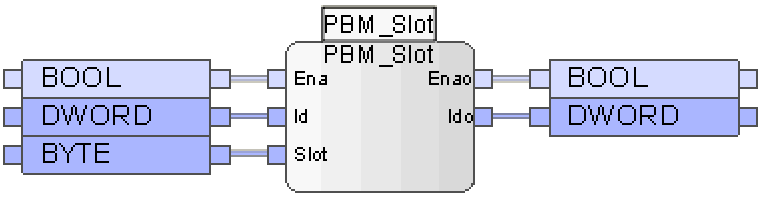


Figure 31: The SLOT Auxiliary Function Block

The **SLOT** auxiliary function block uses an identifier and a slot number to generate a new identifier that addresses the same slave as the previous identifier but with the new slot number.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs	Description	Type
Ena	Not used	BOOL
ID	Logical address of the slave component (slave ID and slot number)	DWORD
Slot	New slot or module number	BYTE

Table 112: Inputs for the SLOT Auxiliary Function Block

Outputs	Description	Type
Enao	Not used	BOOL
Ido	Identification number of the slave	DWORD

Table 113: Outputs for the SLOT Auxiliary Function Block

4.10.7 Auxiliary Function Block: STDDIAG

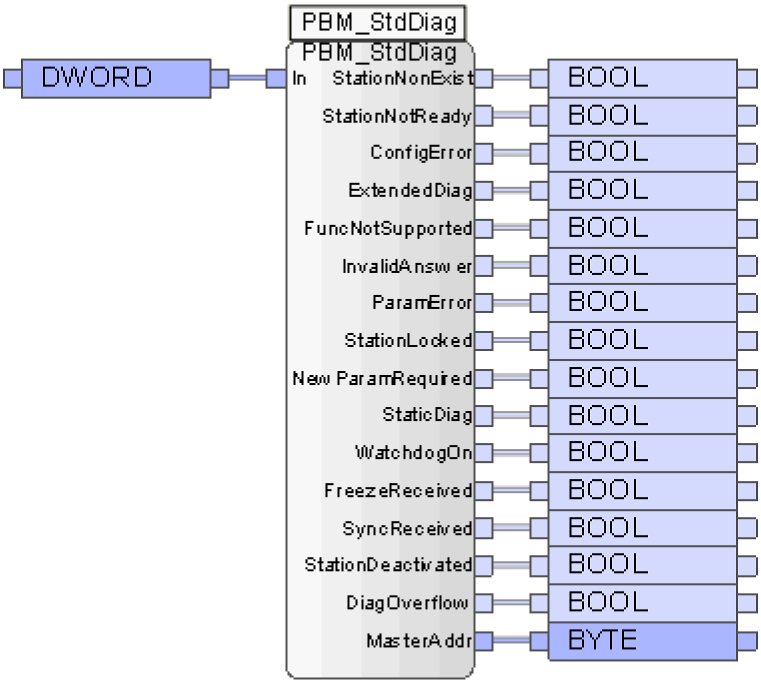


Figure 32: The STDDIAG Auxiliary Function Block

The **STDDIAG** auxiliary function block decodes the standard diagnostics of a PROFIBUS DP slave.

The outputs of type BOOL in the **STDDIAG** auxiliary function block are set to TRUE if the corresponding bit has been set in the standard diagnostics.

i To configure the function block, drag it from the function block library onto the user program, see also Chapter 5.3.

Inputs	Description	Type
IN	Slave standard diagnostics	DWORD

Table 114: Inputs for the STDDIAG Auxiliary Function Block

Outputs	Description	Type
StationNonExist	The slave does not exist	BOOL
StationNotReady	Slave is not ready	BOOL
ConfigError	Configuration error	BOOL
ExtendedDiag	Extended diagnostics follow	BOOL
FuncNotSupported	Function not supported	BOOL
InvalidAnswer	Invalid reply from slave	BOOL
ParamError	Parameter error	BOOL
StationLocked	Slave locked by another master	BOOL
NewParamRequired	New configuration data required	BOOL
StaticDiag	Static diagnosis	BOOL
WatchdogOn	Watchdog activated	BOOL
FreezeReceived	Freeze command received	BOOL
SyncReceived	Sync command received	BOOL
StationDeactivated	The slave was deactivated	BOOL
DiagOverflow	Diagnostic overflow	BOOL
MasterAddr	Master bus address	BYTE

Table 115: Outputs for the STDDIAG Auxiliary Function Block

To read out the standard diagnostics of the PROFIBUS DP slave

1. In the structure tree, open **Configuration, Resource, Protocols, PROFIBUS DP Master.**
2. Right-click **PROFIBUS Slave** and select **Edit**.
3. Drag the global variable of type DWORD onto the *Standard Diagnostics* field.
4. Connect this global variable with the input of the **STDDIAG** function block.

4.11 Error Codes of the Function Blocks

If a function block cannot properly execute a command, an error code is output to **A_Status**. The following table describes the meaning of the error codes.

Error code	Symbol	Description
16#40800800	TEMP_NOT_AVAIL	The service is temporarily not available.
16#40801000	INVALID_PARA	Invalid parameter.
16#40801100	WRONG_STATE	The slave does not support DP-V1.
16#40808000	FATAL_ERR	A fatal program error occurred.
16#40808100	BAD_CONFIG	Configuration error occurred in the data area.
16#40808200	PLC_STOPPED	The controller was stopped.
16#4080A000	READ_ERR	Error while reading a record.
16#4080A100	WRITE_ERR	Error while writing a record.
16#4080A200	MODULE_FAILURE	The error cannot be specified in greater detail.
16#4080B000	INVALID_INDEX	The index is invalid.
16#4080B100	WRITE_LENGTH	Invalid write length.
16#4080B200	INVALID_SLOT	Invalid slot number.
16#4080B300	TYPE_CONFLICT	Invalid type.
16#4080B400	INVALID_AREA	Invalid read/write area.
16#4080B500	STATE_CONFLICT	Master in invalid state.
16#4080B600	ACCESS_DENIED	Slave is not active (or similar).
16#4080B700	INVALID_RANGE	Invalid read/write area.
16#4080B800	INVALID_PARAMETER	Invalid parameter value.
16#4080B900	INVALID_TYPE	Invalid parameter type.
16#4080C300	NO_RESOURCE	Slave is not available.
16#4080BA00	BAD_VALUE	Invalid value.
16#4080BB00	BUS_ERROR	Bus error.
16#4080BC00	INVALID_SLAVE	Invalid slave ID.
16#4080BD00	TIMEOUT	Timeout occurred.
16#4080C000	READ_CONSTRAIN	Read constraint.
16#4080C100	WRITE_CONSTRAIN	Write constraint.
16#4080C200	Busy	A function block of this type is already active.
16#4080C300	NO_RESOURCE	Slave is not active.

Table 116: Error Codes of the Function Blocks

4.12 Control Panel (PROFIBUS DP Master)

The Control Panel can be used to verify and control the settings for the PROFIBUS DP master. It also displays details on the current status of the master or slave associated with it (e.g., cycle time, bus state, etc.).

To open the Control Panel for monitoring the PROFIBUS DP master

1. Right-click the **Hardware** structure tree element and select **Online** from the context menu.
2. In the **System Login** window, enter the access data to open the online view for the hardware.
3. Double-click **COM Module** and select the **PROFIBUS DP Master** structure tree node.

4.12.1 Context Menu (PROFIBUS DP Master)

The following commands can be chosen from the context menu of the selected PROFIBUS DP master:

Offline

Switches off the selected PROFIBUS DP master. If the master is switched off, it is completely disabled.

Stop

Stops the selected PROFIBUS DP master. The PROFIBUS DP master still participates in the token protocol, but does not send any data to the slaves.

Clear

By clicking the CLEAR button, the selected PROFIBUS DP master adopts a safe state and exchanges safe data with the slaves. The output data sent to the slaves only contains zeros. Failsafe slaves receive failsafe telegrams containing no data. The PROFIBUS DP master ignores the input data from the slaves, and uses the initial values in the user program instead.

Operate

Start the selected PROFIBUS DP master. The PROFIBUS DP master cyclically exchanges I/O data with the slaves.

Reset Statistics

The *Reset Statistics* button is used to reset the statistical data such as min. or max. cycle time to zero.

4.12.2 Context Menu (PROFIBUS DP Slave)

The following commands can be chosen from the context menu of the selected PROFIBUS DP slave:

Activate

Activates the selected slave which can now exchange data with the PROFIBUS DP master.

Deactivate

Deactivates the selected slave. The communication is terminated.

4.12.3 View Box (PROFIBUS Master)

The view box displays the following values of the selected PROFIBUS DP master.

Element	Description
Name	Name of the PROFIBUS DP master.
Master State	Indicates the current protocol state, see Chapter 4.12.4. 0 = OFFLINE 1 = STOP 2 = CLEAR 3 = OPERATE 100 = UNDEFINED
Bus State	Bus error code 0…6: 0 = OK 1 = Address error: The master address is already available on the bus. 2 = Bus malfunction: A malfunction was detected on the bus, e.g., bus was not properly terminated, and multiple subscribers are sending data simultaneously. 3 = Protocol error: An incorrectly coded packet was received. 4 = Hardware fault: The hardware reported a fault, e.g., too short time periods. 5 = Unknown error: The master changed the state for an unknown reason. 6 = Controller Reset: In the event of severe bus disturbances, the controller chip adopts an undefined state and is reset. An error code retains its value until the bus error has been removed.
Fieldbus Interface	FB1, FB2
µP Load (planned) [%]	Load of the COM module planned for this protocol.
µP Load (actual) [%]	Actual load of the COM module for this protocol.
Baud Rate [bps]	Baud rate of the master The master can communicate using all baud rates specified in the standard. Cycle times can be set up to a lower limit of 2 ms.
Fieldbus Address	Master bus address (0…125).
PNO-IdentNo	16-bit unique number assigned by the PROFIBUS DP Nutzerorganisation e.V. (PNO) Germany to a product (field device) and identifying it.
Number of Bus Errors	Number of bus errors, so far.
MSI [ms]	Min. Slave Interval in ms, resolution 0.1 ms
TTR [ms]	Target token rotation time in ms, resolution 0.1 ms.

Element	Description
Last Cycle Time [ms]	Last PROFIBUS DP cycle time [ms].
Minimum Cycle Time [ms]	Minimum PROFIBUS DP cycle time [ms].
Average Cycle Time [ms]	Average PROFIBUS DP cycle time [ms].
Maximum Cycle Time [ms]	Maximum PROFIBUS DP cycle time [ms].

Table 117: View Box for the PROFIBUS Master

4.12.4 PROFIBUS DP Master State

The master status is displayed in the view box of the Control Panel and can be evaluated in the user program using the Master Connection State status variable.

State of the master	State of the master
OFFLINE	The master is switched off; no bus activity.
STOP	The master participates in the token protocol, but does not send any data to the slaves.
Clear	The master is in the safe state and exchanges data with the slaves. ▪ The output data sent to the slaves only contains zeros. ▪ Failsafe slaves receive failsafe telegrams containing no data. ▪ The input data of the slaves are ignored and initial values are used instead.
Operate	The master is operating and cyclically exchanges I/O data with the slaves.
Undefined	The firmware for the PROFIBUS DP master module is being updated.

Table 118: PROFIBUS DP Master State

4.12.5 Behavior of the PROFIBUS DP Master

Behavior of the PROFIBUS DP master according to the controller operating state.

State of the controller	Behavior of the HIMA PROFIBUS DP master
STOP [1])	If the controller is in STOP, the master is in the OFFLINE state.
RUN	If the controller is in RUN, the master tries to enter the OPERATE state.
STOP	If the controller enters the STOP state, the master adopts the CLEAR state. If the master is already in STOP or OFFLINE, it stays in this state.
[1]) After powering up the controller or loading the configuration.	

Table 119: Behavior of the PROFIBUS DP Master

4.12.6 Function of the FBx LED in the PROFIBUS Master

The FBx LED of the corresponding fieldbus interface indicates the state of the PROFIBUS DP protocol. The states of the FBx LED are specified in the following table:

FBx LED	Description
OFF	No configuration or invalid configuration of the PROFIBUS DP master.
Blinking Every 2 seconds	Valid configuration. The PROFIBUS DP master is in the OFFLINE or STOP state.
ON	The PROFIBUS DP master is in the OPERATE or CLEAR state, if all activated slaves are connected.
Blinking, every second	At least one slave failed.

Table 120: The FBx LED

4.12.7 Function of the FAULT LED in the PROFIBUS DP Master (HIMax only)

The FAULT LED of the corresponding fieldbus interface indicates that the PROFIBUS DP protocol has failed. The states of the FAULT LED are specified in the following table:

FAULT LED	Color	Description
OFF	Red	The PROFIBUS DP protocols is not disturbed.
Blinking	Red	The protocol is disturbed. ▪ At least one slave failed. ▪ Bus error detected. ▪ Calculating time budget was exceeded. If no faults occur for a period longer than 5 s, the state changes to *Protocol not disturbed*.

Table 121: The FAULT FBx

4.13 HIMA PROFIBUS DP Slave

This chapter describes the characteristics of the HIMA PROFIBUS DP slave and the menu functions and dialog boxes in SILworX required for configuring the HIMA PROFIBUS DP slave.

Equipment and System Requirements

Element	Description
HIMA controller	HIMax with X-COM 01 module. HIQuad X with F-COM 01 module HIMatrix as of CPU OS V7 and COM OS V12
COM module	The serial fieldbus interface (FB1 or FB2) used on the COM module must be equipped with an optional HIMA PROFIBUS DP slave submodule. For further details on the interface assignment, refer to the communication manual (HI 801 101 E).
Activation	Activation via the plug-in module, refer to the communication manual (HI 801 101 E) for details.

Table 122: Equipment and System Requirements for the HIMA PROFIBUS DP Slave

i For the HIMax system, HIMA recommends operating the PROFIBUS DP using the FB1 fieldbus interface (maximum transfer rate 12 Mbit). The maximum transfer rate permitted for the FB2 fieldbus interface is 1.5 MBit.

PROFIBUS DP Slave Properties

<table>
<tr><th>Element</th><th colspan="2">Description</th></tr>
<tr><td>Type of HIMA PROFIBUS DP slave</td><td colspan="2">DP-V0</td></tr>
<tr><td>Transfer rate</td><td colspan="2">9.6 kbit/s…12 Mbit/s</td></tr>
<tr><td>Bus address</td><td colspan="2">0…125</td></tr>
<tr><td>Max. number of slaves</td><td colspan="2">One HIMA PROFIBUS DP slave can be configured for each COM module.</td></tr>
<tr><td rowspan="2">Process data volume of a HIMA PROFIBUS DP slave</td><td>HIMax and HIMatrix</td><td>DP output: max. 192 bytes
DP input: max. 240 bytes
Total: max. 256 bytes</td></tr>
<tr><td>HIQuad X</td><td>DP output: max. 244 bytes
DP input: max. 244 bytes
Total: max. 488 bytes</td></tr>
<tr><td>Protocol watchdog</td><td colspan="2">If the COM is in RUN and the connection to the PROFIBUS DP master is lost, the DP slave detects this once the watchdog timeout has expired (the watchdog timeout must be set in the master). In this case, the DP output data (or input data from the perspective of the resource) are reset to their initial value and the Data Valid flag (status variable of the DP slave protocol) is set to FALSE.</td></tr>
</table>

Table 123: Properties of the HIMA PROFIBUS DP Slave

4.13.1 Creating a HIMA PROFIBUS DP Slave

To create a new HIMA PROFIBUS DP slave

1. In the structure tree, open **Configuration**, **Resource**, **Protocols.**
2. Select **New, PROFIBUS DP Slave** from the context menu of protocols to add a new PROFIBUS DP slave.
3. Select **Edit** from the context menu of the PROFIBUS DP slave.
4. In the **Properties** tab, select **Module** and the **Interface**.

4.14 Menu Functions of the PROFIBUS DP Slave

4.14.1 Edit

The **Edit** dialog box for the PROFIBUS DP master contains the following tabs:

Process Variables

The send and receive variables are created in the **Process Variables** tab.

Input Variables

The variables that the current controller should receive are entered in the *Input Signals* area.

Any variables can be created in the *Input Signals* area. Offsets and types of the received variables must be identical to offsets and types of the send variables of the communication partner.

Output variables

The variables for cyclic data exchange sent by this controller are entered in the *Output Signals* area.

Any variables can be created in the *Output Signals* area. Offsets and types of the received variables must be identical to offsets and types of the receive variables of the communication partner.

System Variables

The variables that should be read in the controller are defined in the **System Variables** tab.

The **System Variables** tab provides the following system variables, which allow the PROFIBUS DP slave state to be evaluated in the user program.

Element	Description
Current Baud Rate	Baud rate currently used by the PROFIBUS DP slave protocol.
Data Valid	If the status variable *Data Valid* is set to TRUE, the slave received valid import data from the master. The status variable is set to FALSE if the watchdog time within the slave has expired. Default value: FALSE Notes: ▪ A global variable that is assigned the system variable *Data Valid* may not have the initial value TRUE. ▪ If the master did not activate the slave's watchdog and the connection is lost, the *Data Valid* status variable retains the value TRUE since the PROFIBUS DP slave has no means to recognize that the connection was lost. This fact must be taken into account when using this variable!
Error Code	If an error occurred in the PROFIBUS DP slave protocol, the error is transferred in this variable. The last occurred error is displayed. Possible (hexadecimal) value: 0x00: No error. 0xE1: Faulty configuration by the PROFIBUS DP master. 0xD2: Faulty configuration by the PROFIBUS DP master. Default value: 0x00

Element	Description
Master Address	This is the address of the PROFIBUS master that configured its own PROFIBUS DP slave. Possible (decimal) values: 0-125: Master address 255: The slave is currently not assigned to any master. Default value: 255
Protocol State	Describes the state of the PROFIBUS DP slave protocol. Possible (hexadecimal) value: 0xE1: The controller is disconnected from the bus or not active. 0xD2: The controller waits for a configuration from the master. 0xC3: The controller cyclically exchanges data with the PROFIBUS DP master. Default value: 0xE1
Slave Address	Address of the controller's PROFIBUS DP slave. This address was previously configured by the user using the PADT. Possible (decimal) values: 0-125: Slave address
Watchdog Time	Watchdog time in milliseconds configured in the master, see Chapter 4.6.3.

Table 124: System Variables in the PROFIBUS DP Slave

4.14.2 Properties

The **Properties** tab for the HIMA PROFIBUS DP slave contains the following parameters for configuring the PROFIBUS DP slave.

The default values for *Within one cycle* and *Process Data Refresh Rate [ms]* provide a fast means of exchanging PROFIBUS DP data between the COM module (COM) and the PROFIBUS DP slave hardware of the HIMA controller.

These parameters should only be changed if it is necessary to reduce the COM load for an application, and the process allows this change.

i

Only experienced programmers should modify the parameters.

Increasing the refresh rate for the COM and PROFIBUS DP hardware means that the effective refresh rate of the PROFIBUS DP data is also increased. The system time requirements must be verified.

The **Min. Slave Interval [ms]** parameter defines the minimum refresh rate of the PROFIBUS DP data between PROFIBUS DP master and PROFIBUS DP slave, see Timings Tab, Chapter 4.3.2) .

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>Type</td><td>PROFIBUS DP Slave.</td></tr>
<tr><td>Name</td><td>Name of the PROFIBUS DP Slave</td></tr>
<tr><td>Module</td><td>Selection of the COM module within which the protocol is processed.</td></tr>
<tr><td>Activate Max. µP Budget</td><td>Activated
Use the µP budget limit from the Max. µP Budget in [%] field.
Deactivated:
Do not use the µP budget limit for this protocol.</td></tr>
<tr><td>Max. µP Budget in [%]</td><td>Maximum µP budget of the module that can be used for processing the protocols.
Range of values: 1…100%
Default value: 30%</td></tr>
<tr><td>Behavior on CPU/COM connection loss</td><td>If the connection of the processor module to the communication module is lost, the input variables are either initialized or are still used unchanged in the process module, depending on this parameter.
For instance, if the communication module is removed when communication is running.
If a project created with a SILworX prior to V3 should be converted, this value must be set to Retain Last Value to ensure that the CRC does not change.
For HIMatrix controllers with CPU OS prior to V8, this value must always be set to Retain Last Value.

Apply Initial Data: Input variables are reset to their initial values.
Retain Last Value: The input variables retain the last value.</td></tr>
<tr><td>Station Address</td><td>Slave station address.
Only one slave station address may be available on the bus.
Range of values: 1…125
Default value: 0</td></tr>
<tr><td>Interface</td><td>Fieldbus interface that should be used for the PROFIBUS DP slave.
Range of values: FB1, FB2
Default value: None</td></tr>
<tr><td>Baud Rate [bps]</td><td>Baud rate used for the bus.
Possible values:
<table>
<tr><th>Value</th><th>Baud rate</th><th>FB1</th><th>FB2</th></tr>
<tr><td>9600</td><td>9.6 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>19200</td><td>16.2 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>45450</td><td>45.45 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>93750</td><td>93.75 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>187500</td><td>187.5 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>500000</td><td>500 kbit/s</td><td>X</td><td>X</td></tr>
<tr><td>1500000</td><td>1.5 Mbit/s</td><td>X</td><td>X</td></tr>
<tr><td>3000000</td><td>3 Mbit/s</td><td>X</td><td>-</td></tr>
<tr><td>6000000</td><td>6 Mbit/s</td><td>X</td><td>-</td></tr>
<tr><td>12000000</td><td>12 Mbit/s</td><td>X</td><td>-</td></tr>
</table></td></tr>
<tr><td>Process Data Refresh Rate [ms]</td><td>Refresh rate in milliseconds at which the COM and the PROFIBUS DP slave hardware exchange protocol data.
Range of values: 4…1000
Default value: 10</td></tr>
</table>

Element	Description
Force Process Data Consistency	Activated: Transfer of all of the protocol data from the CPU to the COM within a CPU cycle. Deactivated: Transfers all of the protocol data from the CPU to the COM, distributed over multiple CPU cycles, each with 1100 bytes per data direction. This may also cause the cycle time of the controller to be reduced. Default value: Activated

Table 125: Slave Properties: General Tab

4.15 Control Panel (PROFIBUS DP Slave)

The Control Panel can be used to verify and control the settings for the PROFIBUS DP slave. Details on the slave's current status (e.g., cycle time, bus state, etc.) are displayed.

To open the Control Panel for monitoring the PROFIBUS DP slave

1. Right-click the **Hardware** structure tree element and select **Online** from the context menu.
2. In the **System Login** window, enter the access data to open the online view for the hardware.
3. Double-click **COM Module** and select the **PROFIBUS DP Slave** structure tree node.

4.15.1 View Box (PROFIBUS DP Slave)

The view box displays the following values of the selected PROFIBUS DP slave.

Element	Description
Name	Name of the PROFIBUS DP slave.
Fieldbus Interface	Subordinated slave's fieldbus interface.
Protocol State	Connection State 0 = Deactivated 1 = Inactive (connection attempt) 2 = Connected
Error State	See Chapter 4.14.1.
Timeout	Watchdog time in milliseconds configured in the master. See Chapter 4.6.3.
Watchdog Time [ms]	Is set in the master. See Chapter 4.6.3.
Fieldbus Address	See Chapter 4.14.2.
Master Address	Address of the PROFIBUS DP master.
Baud Rate [bps]	Current baud rate. See Chapter 4.14.2.
µP Budget (planned) [%]	Load of the COM module planned for this protocol.
µP Budget (actual) [%]	Actual load of the COM module for this protocol.

Table 126: View Box of the PROFIBUS DP Slave

4.16 Function of the FBx LED in the PROFIBUS Slave

The FBx LED of the corresponding fieldbus interface indicates the state of the PROFIBUS DP protocol. The states of the FBx LED are specified in the following table:

FBx LED	Color	Description
OFF	Yellow	The PROFIBUS DP protocol is not active! This means that the controller is in the STOP state or no PROFIBUS DP slave is configured.
Blinking, every 2 seconds	Yellow	No data traffic! The PROFIBUS DP slave is configured and ready.
ON	Yellow	The PROFIBUS DP protocol is active and is exchanging data with the PROFIBUS DP master.

Table 127: The FBx LED

4.17 Function of the FAULT LED in the PROFIBUS Slave (HIMax Only)

The FAULT LED of the corresponding fieldbus interface indicates that the PROFIBUS DP protocol has failed. The states of the FAULT LED are specified in the following table:

FAULT LED	Color	Description
OFF	Red	The PROFIBUS DP protocols is not disturbed.
Blinking	Red	The protocol is disturbed. ▪ The configurations of the PROFIBUS DP master and slave are faulty. ▪ Calculating time budget was exceeded. If no faults occur for a period longer than 5 s, the state changes to *Protocol not disturbed*.

Table 128: The FAULT FBx

5 General

This chapter describes parameters that are relevant for all communication protocols.

5.1 Maximum Communication Time Slice

The maximum communication time slice is the time period in milliseconds (ms) per CPU cycle assigned to the processor module for processing the communication tasks. Even if the protocol processing could not be completed within one communication time slice, the CPU still executes the safety-relevant monitoring for all protocols within one CPU cycle.

i If not all upcoming communication tasks can be processed within one CPU cycle, the whole communication data is transferred over multiple CPU cycles. The number of communication time slices is then greater than 1.

For calculating the maximum response time, the number of communication time slices must be equal to 1.

5.1.1 Determining the Maximum Duration of the Communication Time Slice

For a first estimate of the maximum duration of the communication time slice, the sum of the following times must be entered in the *Max. Com. Time Slice [ms]* system parameter located in the properties of the resource.

- For each COM module: 3 ms.
- For each redundant safe**ethernet** connection: 1 ms.
- For non-redundant safe**ethernet** connection: 0.5 ms.
- For each kilobyte user data of non-safety-related protocols, e.g., Modbus: 1 ms.

HIMA recommends comparing the value estimated for *Max. Com. Time Slice [ms]* with the value displayed in the Control Panel and, if necessary, correcting it in the properties of the resource. This can be done during an FAT (factory acceptance test) or SAT (site acceptance test).

To determine the actual duration of the maximum communication time slice

1. Operate the HIMA system under full load (FAT, SAT):
   All communication protocols are in operation (safe**ethernet** and standard protocols).
2. Open the **Control Panel** and select the **Com. Time Slice** structure tree folder.
3. Read the vale displayed for *Maximum Com. Time Slice Duration per Cycle [ms].*
4. Read the value displayed for *Maximum Number of Required Com. Time Slice Cycles*.

The duration of the communication time slice must be set so that, when using the communication time slice, the CPU cycle cannot exceed the watchdog time specified by the process.

5.2 Load Limitation

A computing time budget expressed in % *(µP budget)* can be defined for each communication protocol. It allows the available computing time to be distributed among the configured protocols. The sum of the computing time budgets configured for all communication protocols on a CPU or COM module may not exceed 100%.

The defined computing time budgets of the individual communication protocols are monitored. If a communication protocol has already achieved or exceeded its budget and no reserve computing time is available, the communication protocol cannot be processed.

If sufficient additional computing time is available, it is used to process the communication protocol that has already achieved or exceeded its budget. It can therefore happen that a communication protocol uses more computing time budget than has been allocated to it.

It is possible that more than 100% computing time budget is displayed online. This is not a fault; the computing time budget exceeding 100% indicates the additional computing time used.

i The additional computing time budget is not a guarantee for a certain communication protocol and can be revoked from the system at any time.

5.3 Configuring the Function Blocks

The fieldbus protocols and the corresponding function blocks operate on the COM module of the HIMA system. In the SILworX structure tree, these function blocks must therefore be created as child element of **Configuration, Resource, Protocols**.

To control the function blocks on the COM module, function blocks can be created in the SILworX user program (see Chapter 5.3.1). These can be used as standard function blocks.

Shared variables are used to connect the function blocks in the SILworX user program to the corresponding function blocks in the SILworX structure tree. These must have been previously created in the Global Variable Editor.

5.3.1 Purchasing Function Block Libraries

The function block libraries for PROFIBUS DP and TCP Send/Receive must be added to the project using the *Restore* function (context menu of the project).

The function block library is available from HIMA support upon request.

5.3.2 Configuring the Function Blocks in the User Program

Drag the required function blocks onto the user program. The inputs and outputs must be configured as described for the individual function block.

Upper Part of the Function Block

The upper part of the function block corresponds to the user interface used by the user program to control the function block.

The variables used in the user program are connected at this level. The prefix A means Application.

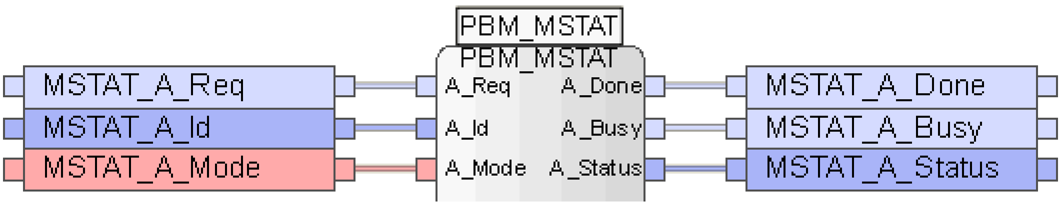


Figure 33: PNM_MSTST Function Block (Upper Part)

Lower Part of the Function Block

The lower part of the function block represents the connection to the function block (in the SILworX structure tree).

The variables that must be connected to the function block located in SILworX structure tree are connected here. The prefix F means Field.

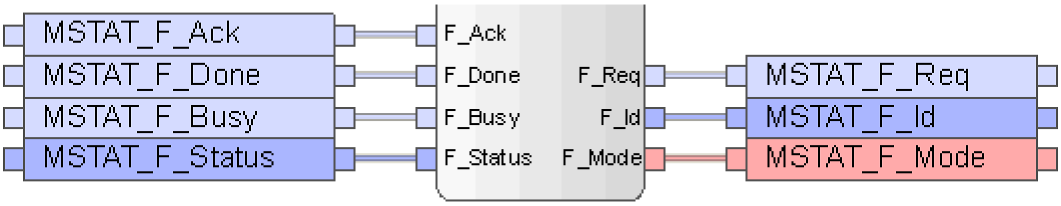


Figure 34: PNM_MSTST Function Block (Lower Part)

5.3.3 Configuring the Function Blocks in the SILworX Structure Tree

To create the function block in the SILworX structure tree

1. In the structure tree, open **Configuration**, **Resource**, **Protocols**, e.g., **PROFIBUS Master**.
2. Right-click **Function Blocks** and select **New**.
3. In the SILworX structure tree, select the suitable function block.

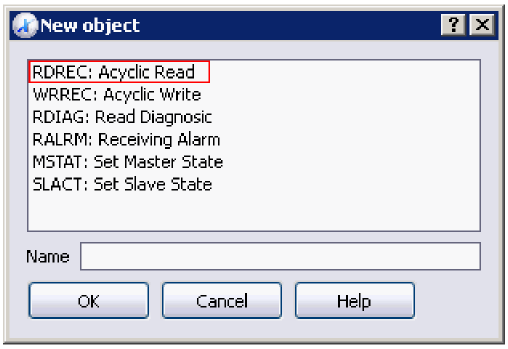


Figure 35: Selecting Function Blocks

The inputs of the function block (checkmark in the Input Variables column) must be connected to the same variables that are connected in the user program to the *F_Outputs* of the function block.

The outputs of the function block (no checkmark in the Input Variables column) must be connected to the same variables that are connected in the user program to the *F_Inputs* of the function block.

System Variables

F	Name	Data type	Input variable	Global Variable
1	ACK	BOOL	☑	MSTAT_F_Ack
2	BUSY	BOOL	☑	MSTAT_F_Busy
3	DONE	BOOL	☑	MSTAT_F_Done
4	M_ID	DWORD	☐	MSTAT_F_Id
5	MODE	INT	☐	MSTAT_F_Mode
6	REQ	BOOL	☐	MSTAT_F_Req
7	STATUS	DWORD	☑	MSTAT_F_Status

Figure 36: System Variables of the MSTAT Function Block

Appendix

Glossary

Term	Description
ARP	Address resolution protocol, network protocol for assigning the network addresses to hardware addresses.
Bit variable	Variable that is addressed bit by bit.
CENELEC	Comité Européen de Normalisation Électrotechnique (European Commitee for Electrotechnical Standardization).
COM	Communication module.
Connector board	Connector board for the HIMax module.
CPU	Processor module.
CRC	Cyclic redundancy check.
Data view	The global variables for output and output data are assigned to a data view to allow access to Modbus sources.
EN	European standard.
Export area	The export area is the process data volume that is written to by the system (a user program, hardware input or another protocol) and is read by the Modbus master.
FB	Fieldbus.
FBD	Function block diagrams.
ICMP	Internet control message protocol, network protocol for status or error messages.
IEC	International electrotechnical commission.
Import area	Process data volume that is written to by the Modbus master and can be used as input data for the system (in a user program, hardware output or another protocol).
Interference-free	Supposing that two input circuits are connected to the same source (e.g., a transmitter). An input circuit is termed "interference-free" if it does not distort the signals of the other input circuit.
KE	Communication end point.
MAC address	Media access control address, hardware address of one network connection.
NSIP	Not safety-related protocol.
PADT	Programming and debugging tool (acc. to IEC 61131-3), PC with SILworX.
PE	Protective ground.
PELV	Protective extra low voltage.
PES	Programmable electronic system.
R	Read.
R/W	Read/Write.
Rack ID	Base rack identification (number).
Register variable	Variable that is addressed word by word.
SB	System bus.
SFF	Safe failure fraction, i.e., portion of faults that can be safely controlled.
SIF	Safety-instrumented function.
SIL	Safety integrity level (in accordance with IEC 61508).
SILworX	Programming tool for HIMax, HIQuad X und HIMatrix.
SIP	Safety-instrumented protocol.
SNTP	Simple network time protocol (RFC 1769).
SRS	System.Rack.Slot.
SW	Software.
TMO	Timeout.
W	Write.
WD	Watchdog.
WDT	Watchdog time.

Index of Figures

Index of Tables

MANUAL
PROFI* Protocols

HI 801 523 E

For further information, please contact:

HIMA Paul Hildebrandt GmbH
Albert-Bassermann-Str. 28
68782 Brühl, Germany

Phone +49 6202 709-0
Fax +49 6202 709-107
E-mail info@hima.com

Learn more about HIMA solutions online:

www.hima.com/en/

www.hima.com