BAUMÜLLER

be in motion be in motion

BM4-O-PRO-01

Option Module PROFIBUS-DP-Slave for b maXX PLC
Application Manual

E	5.03058.02

Title	Application Manual
Product	Option Module PROFIBUS-DP-Slave for b maXX PLC BM4-O-PRO-01
Last Revision:	July 07, 2004

Binding nature

This Application Manual is a part of the unit/machine. This Application Manual must always be available to operators and be legible. If the unit/machine is sold, the owner must pass on this Application Manual together with the unit/machine.
After selling the unit/machine you must pass on this original and all the copies that you made to the purchaser. After disposing of the machine in any way, you must destroy this original and all the copies that you made.

When you pass on this Application Manual, all earlier revisions of the corresponding Application Manuals are invalidated.
Note that all the data/numbers/information that are quoted are **current values at the time of printing.** This information is **not legally binding** for dimensioning, calculation and costing.
Within the scope of further-development of our products, Baumüller Nürnberg Electronic GmbH & Co. KG reserve the right to change the technical data and handling.

We cannot guarantee this Application Manual is completely error-free unless this is expressly indicated in our General Conditions of Business and Delivery.

Manufacturer

Baumüller Nürnberg Electronic GmbH & Co. KG
Ostendstr. 80 - 90
90482 Nuremberg
Germany
Tel. +49 9 11 54 32 - 0 Fax: +49 9 11 54 32 - 1 30
www.baumueller.de

Hiermit zeigen wir an, dass die Baumüller Nürnberg Electronic GmbH & Co. KG im Wege der Verschmelzung mit Wirkung zum 01.12.2004 in der Baumüller Nürnberg GmbH aufgegangen ist. Ihr zukünftiger Ansprechpartner ist damit die

Baumüller Nürnberg GmbH, Ostendstrasse 80-90, 90482 Nürnberg.

Please notify that with effect from 01-12-2004 Baumüller Nürnberg Electronic GmbH & Co. KG merged with Baumüller Nürnberg GmbH. Your future business partner will be

Baumüller Nürnberg GmbH, Ostendstrasse 80-90, 90482 Nürnberg.

Par la présente, nous vous signalons qu'en voie de la fusion, la Baumüller Nürnberg Electronic GmbH & Co. KG a été intégrée à la Baumüller Nürnberg GmbH avec effet au 1 décembre 2004. Votre interlocuteur sera par conséquent la

Baumüller Nürnberg GmbH, Ostendstrasse 80 – 90, 90482 Nürnberg.

1

INTRODUCTION

This Operation Manual is an important constituent of your b maXX 4400 unit; accordingly and last but not the least, in the interest of your own safety, go through this documentation completely.

In this chapter we will describe the first steps.

1.1 First steps

1. The following hardware is required for programming the Option Module PROFIBUS-DP-Slave for b maXX PLC:
   Basic unit b maXX 4400,
   Option Module b maXX PLC and
   Option Module PROFIBUS-DP-Slave for b maXX PLC.
   The Hardware must be installed as indicated in the respective Operation Manual and be ready for operation.
   Furthermore, you require a PROFIBUS-Master as also a communication cable and where required additional PROFIBUS-Slaves (e. g. additional b maXX 4400 with Option Module PROFIBUS-DP-Slave or also an I/O-Module) for building up a fully functional PROFIBUS-Network .
2. In addition to the above, you require the following Software:
   PROPROG wt II for programming the b maXX PLC and the Option Module PROFIBUS-DP-Slave for b maXX PLC.

1.2 Terms used

We shall also be calling the Baumüller-Product „BM4-O-PRO-01-00-01“ („Option Module “PROFIBUS-DP-Slave für b maXX PLC) as „Plug-in Module “PROFIBUS-DP-Slave“. We shall also be calling the Option Module PROFIBUS-DP-Slave for b maXX PLC as „PROFIBUS-Slave“ and „participant“.

The term "b maXX" is also used for the product "Basic unit b maXX 4400".

The controller unit in the basic unit is also called as „b maXX controller“.

For a list of the abbreviations that are used, refer to ▹Abbreviations◃ from page 87 onward.

1.3 Conditions

This manual is based on the "Application Manual b maXX PLC" and presupposes the knowledge of the programming tools PROPROG wt II and of the related manual.

2

BASIC SAFETY INSTRUCTIONS

We have designed and manufactured each Baumüller plug-in module in accordance with the strictest safety regulations. Despite this, working with the plug-in module can be dangerous for you.

In this chapter, we will describe the risks that can occur when working with a Baumüller plug-in module. Risks are illustrated by icons. All the symbols that are used in this documentation are listed and explained.

In this chapter, we cannot explain how you can protect yourself from specific risks in individual cases. This chapter contains only general protective measures. We will go into concrete protective measures in subsequent chapters directly after information about the individual risk.

2.1 Hazard information and instructions

WARNING

The following **may occur**, if you do not observe this warning information:

- serious personal injury
- death

The hazard information is showing you the hazards which can lead to injury or even to death.

Always observe the hazard information given in this documentation.

Hazards are always divided into three danger classifications. Each danger classification is identified by one of the following words:

DANGER

- Considerable damage to property
- Serious personal injury
- Death **will** occur

WARNING

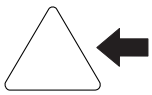
- Considerable damage to property
- Serious personal injury
- Death **can** occur

CAUTION

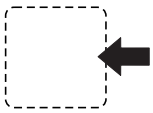
- Damage to property
- Slight to medium personal injury **can** occur

2.1.1 Structure of hazard information

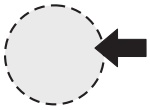
The following two examples show how hazard information is structured in principle. A triangle is used to warn you about danger to living things. If there is no triangle, the hazard information refers exclusively to damage to property.



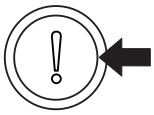
A triangle indicates that there is danger to living things.
The color of the border shows how severe the hazard is: the darker the color, the more severe the hazard is.



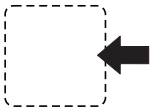
The icon in the rectangle represents the hazard.
The color of the border shows how severe the hazard is: the darker the color, the more severe the hazard is.



The icon in the circle represents an instruction. Users must follow this instruction.
(The circle is shown dashed, since an instruction is not available as an icon for each hazard advisory).



The circle shows that there is a risk of damage to property.



The icon in the rectangle represents the hazard.
The color of the border shows how severe the hazard is: the darker the color, the more severe the hazard is. (The rectangle is shown dashed, since the danger is not represented as an icon with every hazard advisory)

The text next to the icons is structured as follows:

THE SIGNAL WORD IS HERE THAT SHOWS THE DEGREE OF RISK

Here we indicate whether one or more of the results below occurs if you do not observe this warning.

- Here, we describe the possible results. The worst result is always at the extreme right.

Here, we describe the hazard.

Here, we describe what you can do to avoid the hazard.

2.1.2 Hazard advisories that are used

If a signal word is preceded by one of the following danger signs: ⚠ or ⚠ or ⚠, the safety information refers to injury to people.

If a signal word is preceded by a round danger sign: ⓘ, the safety information refers to damage to property.

2.1.2.1 Hazard advisories about injuries to people

To be able to differentiate visually, we use a separate border for each class of hazard information with the triangular and rectangular pictograms.

For danger classification **DANGER,** we use the ⚠ danger sign. The following hazard information of this danger classification is used in this documentation.

DANGER

The following **will occur**, if you do not observe this danger information:

- serious personal injury
- death

Danger from: ***electricity.*** *The hazard may be described in more detail here.*

Here, we describe what you can do to avoid the hazard.

DANGER

The following **will occur**, if you do not observe this danger information:

- serious personal injury
- death

Danger from: ***mechanical effects.*** *The hazard may be described in more detail here.*

Here, we describe what you can do to avoid the hazard.

For danger classification **WARNING,** we use the ⚠ danger sign. The following hazard information of this danger classification is used in this documentation.

WARNING

The following **may occur**, if you do not observe this warning information:

- serious personal injury
- death

Danger from: ***electricity.****The hazard may be described in more detail here.*

Here, we describe what you can do to avoid the hazard.

For danger classification **CAUTION,** we use the ⚠ danger sign. The following hazard information of this danger classification is used in this documentation.

CAUTION

The following **may occur**, if you do not observe this caution information:

- minor to medium personal injury.

Danger from: ***sharp edges.*** *The hazard may be described in more detail here.*

Here, we describe what you can do to avoid the hazard.

CAUTION

The following **may occur**, if you do not observe this danger information:

- environmental pollution.

Danger from: ***incorrect disposal.*** *The hazard may be described in more detail here.*

Here, we describe what you can do to avoid the hazard.

2.1.2.2 Hazard advisories about damage to property

If a signal word is preceded by a round danger sign: ⓘ, the safety information refers to damage to property.

CAUTION

The following **may occur**, if you do not observe this caution information:

- property damage.

Danger from: ***electrostatic discharge.*** *The hazard may be described in more detail here.*

Here, we describe what you can do to avoid the hazard.

2.1.2.3 Instruction signs that are used

carry safety gloves

carry safety shoes

2.2 Information signs

NOTE

This indicates particularly important information.

2.3 Legal information

This documentation is intended for technically qualified personnel that has been specially trained and is completely familiar with all warnings and maintenance measures.

The equipment is manufactured to the state of the art and is safe in operation. It can be put into operation and function without problems if you ensure that the information in the documentation is complied with.

Operators are responsible for carrying out servicing and commissioning in accordance with the safety regulations, applicable standards and any and all other relevant national or local regulations with regard to cable rating and protection, grounding, isolators, overcurrent protection, etc.

Operators are legally responsible for any damage that occurs during assembly or connection.

2.4 Appropriate Use

You must always use the plug-in module appropriately. Some important information is listed below. The information below should give you an idea of what is meant by appropriate use of the plug-in module. The information below has no claim to being complete; always observe all the information that is given in these operating instructions.

- You must only install the plug-in module in series b maXX 4400 units.
- Configure the application such that the plug-in module is always operating within its specifications.
- Ensure that only qualified personnel works with this plug-in module.
- Mount the plug-in module only in the specified slot/slots.
- Install the plug-in module as specified in the operating instructions.
- Ensure that connections always comply with the stipulated specifications.
- Operate the plug-in module only when it is in technically perfect condition.
- Always operate the plug-in module in an environment that is specified in the technical data.
- Always operate the plug-in module in a standard condition.
  For safety reasons, you must not make any changes to the plug-in module.
- Observe all the information on this topic if you intend to store the plug-in module.

You will be using the plug-in module in an appropriate way if you observe all the comments and information in the operating instructions.

2.5 Inappropriate Use

Below, we will list some examples of inappropriate use. The information below should give you an idea of what is meant by inappropriate use of the plug-in module. We cannon, however, list all possible cases of inappropriate use here. Any and all applications in which you ignore the information in this documentation are inappropriate; particularly, in the following cases:

- You installed the plug-in module in units that are not Series b maXX 4400.
- You ignored information in these operating instructions.
- You did not use the plug-in module as intended.
- You handled the plug-in module as follows
  - you mounted it incorrectly,
  - you connected it incorrectly,
  - you commissioned it incorrectly,
  - you operated it incorrectly,
  - you allowed non-qualified or insufficiently qualified personnel to mount the module, commission it and operate it,
  - you overloaded it,
  - You operated the module
    - with defective safety devices,
    - with incorrectly mounted guards or without guards at all,
    - with non-functional safety devices and guards
    - outside the specified environmental operating conditions
- You modified the plug-in module without written permission from Baumüller Nürnberg Electronic GmbH & Co. KG.
- You ignored the maintenance instructions in the component descriptions.
- You incorrectly combined the plug-in module with third-party products.
- You combined the drive system with faulty and/or incorrectly documented third-party products.
- Your self-written PLC software contains programming errors that lead to a malfunction.

Version 1.1 of Baumüller Nürnberg Electronic GmbH & Co. KG’s General Conditions of Sale and Conditions of Delivery dated 2/15/02 or the respective latest version applies in all cases. These will have been available to you since the conclusion of the contract at the latest.

2.6 Protective equipment

In transit, the plug-in modules are protected by their packaging. Do not remove the plug-in module from its packaging until just before you intend to mount it.

The cover on the b maXX units' controller sections provides IP20 protection to the plug-in modules from dirt and damage due to static discharges from contact. This means that you must replace the cover after successfully mounting the plug-in module.

2.7 Personnel training

WARNING

The following **may occur**, if you do not observe this warning information:

- serious personal injury
- death

Only qualified personnel are allowed to mount, install, operate and maintain equipment made by Baumüller Nürnberg Electronic GmbH & Co. KG.

Qualified personnel (specialists) are defined as follows:

Qualified Personnel

Electrical engineers and electricians of the customer or of third parties who are authorized by Baumüller Nürnberg Electronic GmbH & Co. KG and who have been trained in installing and commissioning Baumüller drive systems and who are authorized to commission, ground and mark circuits and equipment in accordance with recognized safety standards.

Qualified personnel has been trained or instructed in accordance with recognized safety standards in the care and use of appropriate safety equipment.

Requirements of the operating staff

The drive system may only be operated by persons who have been trained and are authorized.

Only trained personnel are allowed to eliminate disturbances, carry out preventive maintenance, cleaning, maintenance and to replace parts. These persons must be familiar with the Operating Instructions and act in accordance with them.

Commissioning and instruction must only be carried out by qualified personnel.

2.8 Safety measures in normal operation

- At the unit's place of installation, observe the applicable safety regulations for the plant in which this unit is installed.
- Provide the unit with additional monitoring and protective equipment if the safety regulations demand this.
- Observe the safety measures for the unit in which the plug-in module is installed.

2.9 Responsibility and liability

To be able to work with this PROFIBUS-DP-Slave option module in accordance with the safety requirements, you must be familiar with and observe the hazard information and safety instructions in this documentation.

2.9.1 Observing the hazard information and safety instructions

In these operating instructions, we use visually consistent safety instructions that are intended to prevent injury to people or damage to property.

WARNING

The following **may occur**, if you do not observe this warning information:

- serious personal injury
- death

Any and all persons who work on and with Series b maXX units must always have available the Operating Instructions and must observe the instructions and information they contain – this applies in particular to the safety instructions.

Apart from this, any and all persons who work on this unit must be familiar with and observe all the rules and regulations that apply at the place of use.

2.9.2 Danger arising from using this module

The PROFIBUS-DP-Slave option module has been developed and manufactured to the state of the art and complies with applicable guidelines and standards. It is still possible that hazards can arise during use. For an overview of possible hazards, refer to the chapter entitled ▹Basic Safety Instructions ◃ from page 7 onward.

We will also warn you of acute hazards at the appropriate locations in this documentation.

2.9.3 Warranty and Liability

All the information in this documentation is non-binding customer information; it is subject to ongoing further development and is updated on a continuous basis by our permanent change management system.

Warranty and liability claims against Baumüller Nürnberg Electronic GmbH & Co. KG are excluded; this applies in particular if one or more of the causes listed in ▹Inappropriate Use ◃ from page 12 onward or below caused the fault:

- Disaster due to the influence of foreign bodies or force majeure.

3

PROFIBUS

Information on data exchange via PROFIBUS is given in this chapter.

As PROFIBUS-DP-Slave is available the Option Module BM4-O-PRO-01-00-01.

Different devices of different manufacturers can be used as PROFIBUS-Master, e. g.:

- Simatic S7 with PROFIBUS-Master, manufacturer Siemens
- PROFIBUS-Master-PC-card, manufacturer Beckhoff

Your decision to use a particular PROFIBUS-Master is mainly depending on your application .

3.1 General information on PROFIBUS and the use of the Option Module PROFIBUS-DP-Slave

PROFIBUS is a very efficient serial field bus system, which enables networking (amongst) the most diverse devices. PROFIBUS guarantees a safe and dependable passage and continuity from the field level to the management (guide) level via a cell level Different versions of the PROFIBUS were developed for supporting the functions required in each case.

- PROFIBUS-FMS is used for the object oriented, universal data exchange. The cycle times of the bus are usually less than 100 ms. It is preferably used in the cell level and for conecting to the management level.
- PROFIBUS-DP is used for data exchange at fast speed. The cycle times of the bus are usually less than 10 ms. It is preferably used in the field level and for conecting to the cell level.
- PROFIBUS-PA is a variant of the PROFIBUS-DP with its built - in safty which can be used for supplying energy to the devices operating in the field via the bus. It is used in the field level and for conecting to the cell level.

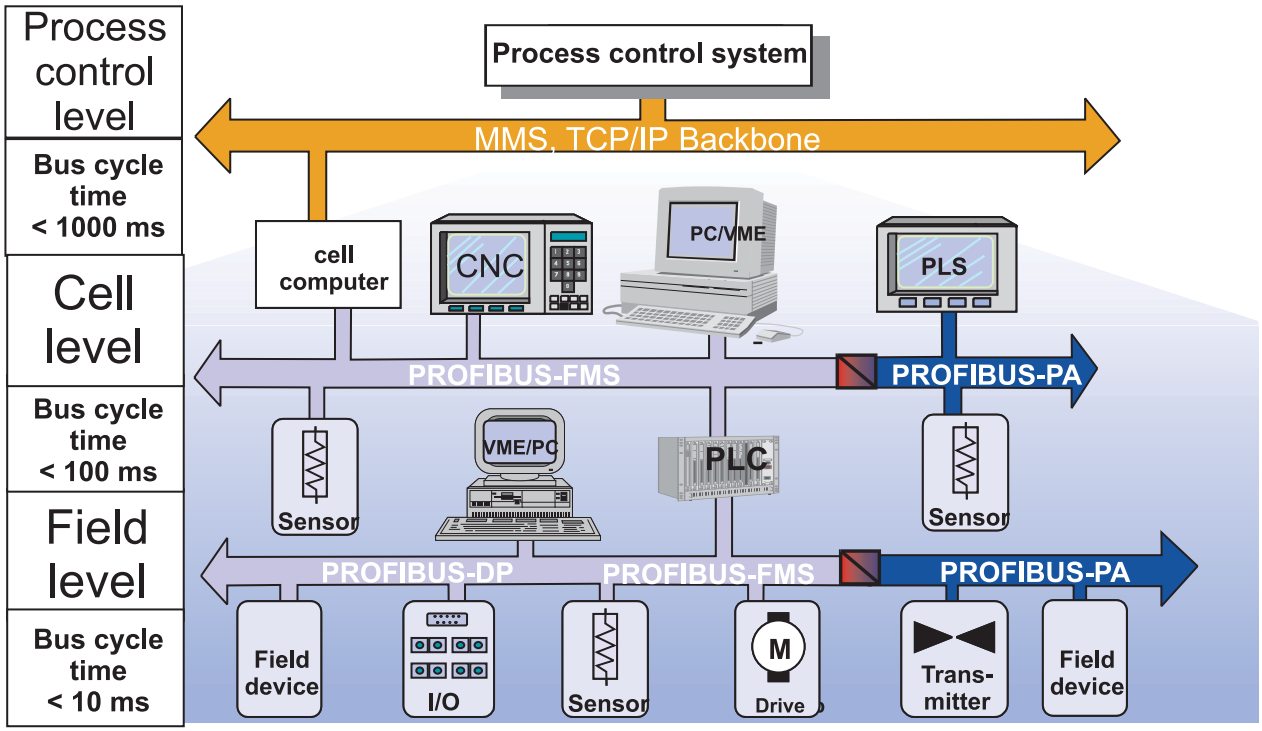
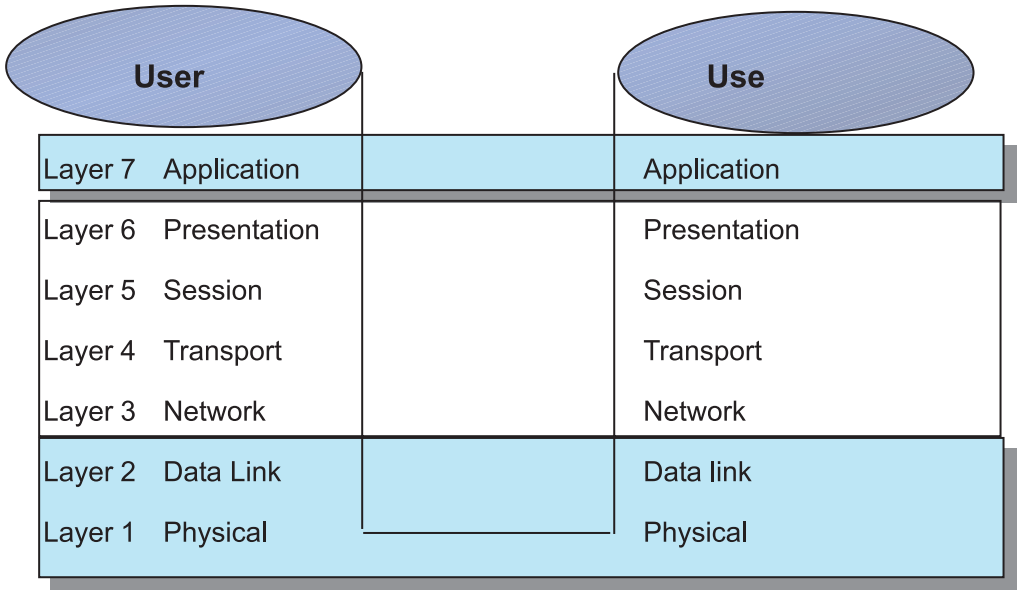


Figure 1: Hierarchy-Levels

PROFIBUS is defined in the European Standard EN 50170, Part 2 .

Like almost all the field buses, PROFIBUS supports the layers 1, 2 and 7 from the ISO-/OSI-Communication-Layer model:



Field busses have the requirement to ensure the work flow between sensor/actuator level and process control system.

Figure 2: ISO-/OSI-Layer model

For both the versions PROFIBUS-DP and PROFIBUS-FMS, the layers 1 and 2 are the identical in the ISO-/OSI-Model, i.e. the Physical layer as also the datalink layer are identical. The layers 3 to 7 are empty in PROFIBUS-DP, so that PROFIBUS-DP can be considered as the standardised application on layer 2.

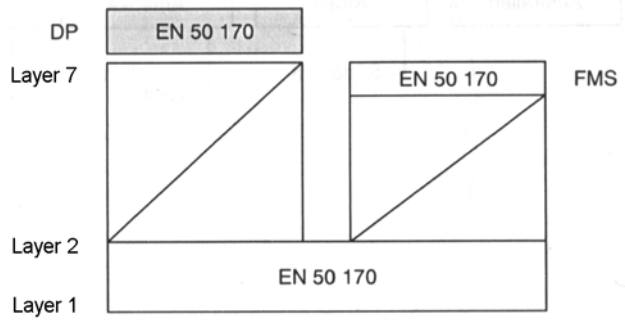


Figure 3: PROFIBUS in ISO-/OSI-Layer model

PROFIBUS-DP-Profiles have been defined within the framework of uniform representation of certain devices. Among others, there are such profiles for:

- Drives (variable speed drives ; PROFIdrive V3)
- Safe communications (PROFIsafe)
- Encoder
- Building Automation
- Numerical contols (NC - Controls)
- Human-Machine-Interfaces (HMI), (Mensch Maschine Interface - MMI)

Although the b maXX 4400 together with b maXX PLC and the Option Module PROFIBUS-DP-Slave is a drive, the PROFIBUS-Profile specification 3 PROFIdrive V3 is not supported.

Reason: The PROFIBUS-Master does not have a direct access to the drive. The PROFIBUS-Master accesses the data of the b maXX PLC (which is programmed in accordance with IEC 61131-3).

PROFIBUS-DP is available in three versions, which differ in their scope of functioning. These are the Versions V0, V1 and V2, whereby V1 and V2 are PROFIBUS-DP extensions.

The Option Module PROFIBUS-DP-Slave supports PROFIBUS-DP V0 fully.

Using the Option Module PROFIBUS-DP-Slave for the b maXX PLC, together with the Function blocks from the Library PROFIBUS_PLC01_20bd01 (or higher) for the programming surface PROPROG wt II, you can realise a data exchange with PROFIBUS-DP-Slave-functionalities.

Transmission speeds (Baud rates) available for das Option Module PROFIBUS-DP-Slave are:
9,6 kBit/s, 19,2 kBit/s, 31,25 kBit/s, 45,45 kBit/s, 93,75 kBit/s, 187,5 kBit/s,
500 kBit/s, 1,5 MBit/s, 3,0 MBit/s, 6,0 MBit/s and 12 MBit/s .

3.2 Fundamentals of PROFIBUS-DP

3.2.1 Physical structure of the Bus

Transmission medium/ransmission Interface

The physical structure of the PROFIBUS is defined in the Standard: IEC 61158. The data can be transmitted, firstly by means of RS485 via a twisted two wire cable (of the cable type A) and it is also possible to use fiber optic conductors as the medium for transmission.

The Option Module PROFIBUS-DP-Slave uses the RS485 and a twisted two wire cable (of the cable type A) for transmission of data. A 9-pole SUB-D-socket is available as connector to the PROFIBUS-DP (connector pin assignment, see Operating Instructions for PROFIBUS-DP-Slave for b maXX).

Parameters	Type of Cable:A
Wave resistance in Ohm	135..165 at 3..20 MHz
Working capacity in pF/m	<30
Loop resistance in Ohm/km	<110
Core diameter in mm	>0,64
Cross section of the core in mm^2	>0,34

Length extensions for a bus segment as indicated in the table are obtained with the cable parameters given on the previous page:

Transmission Speed	Cable Length
9,60 kBit/s	1200 m
19,20 kBit/s	1200 m
45,45 kBit/s	1200 m
93,75 kBit/s	1200 m
187,50 kBit/s	1000 m
500,00 kBit/s	400 m
1,50 MBit/s	200 m
3,00 MBit/s	100 m
6,00 MBit/s	100 m
12,00 MBit/s	100 m

Stub cables upto 1500 kBit/s < 6,6 m
Stub cables should not be used where the transmission speeds exceed 1500 kBit/s. If stub cables cannot be avoided, then they must not have any bus terminator.

Bus terminator

Bus terminator resistances are equal to the impedance of the bus line at the impedance of the bus-devices. The input circuit of the field units has a significantly higher resistance than the bus terminator. By terminating the bus cable at both the ends of a segment by the terminator resistances, it is ensured that

- a definite open-circuit potential is set on the cable,
- cable reflexions are minimised,
- an almost uniform load pattern is set on the bus.

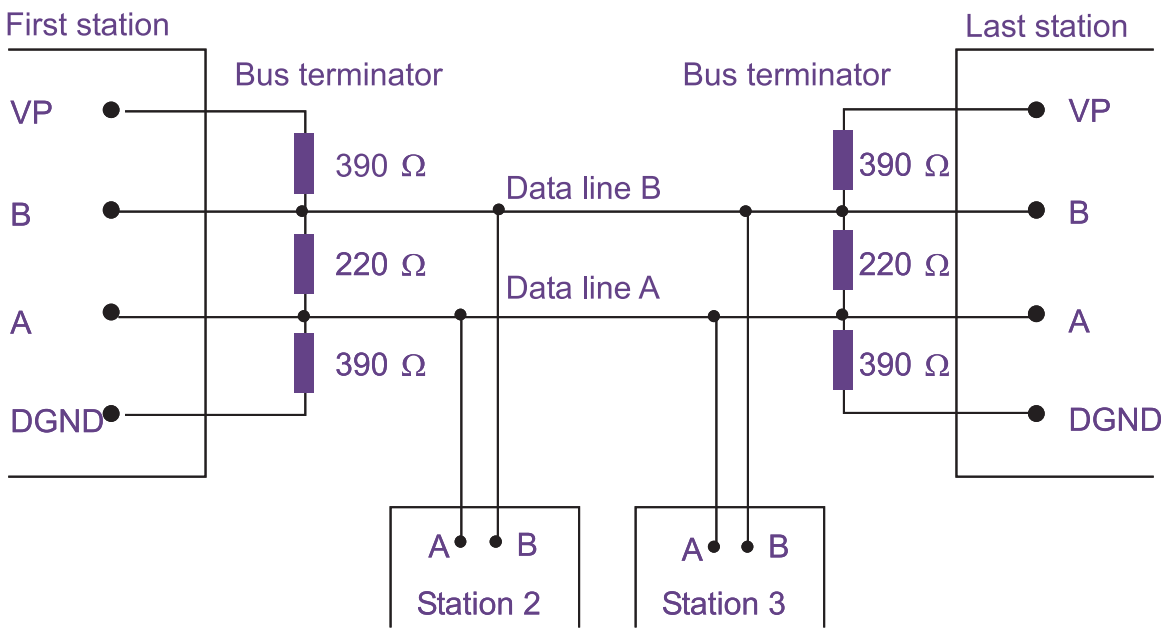


Figure 4: Bus Terminator

The data lines are connected through, in the PROFIBUS connecting slot; failure of any device thus does not affect other field units.

3.2.2 Network

A PROFIBUS network consists of:

- (at least) one or several PROFIBUS-Master(s)
- (at least) one or several PROFIBUS-Slave(s)
- the transmission medium
- two bus terminating resistor connector per segment
- where required, additional segments, which are connected via the repeater(s)

Maximum 126 devices are permitted in the overall PROFIBUS network, while a maximum of only 32 devices may be connected per segment.

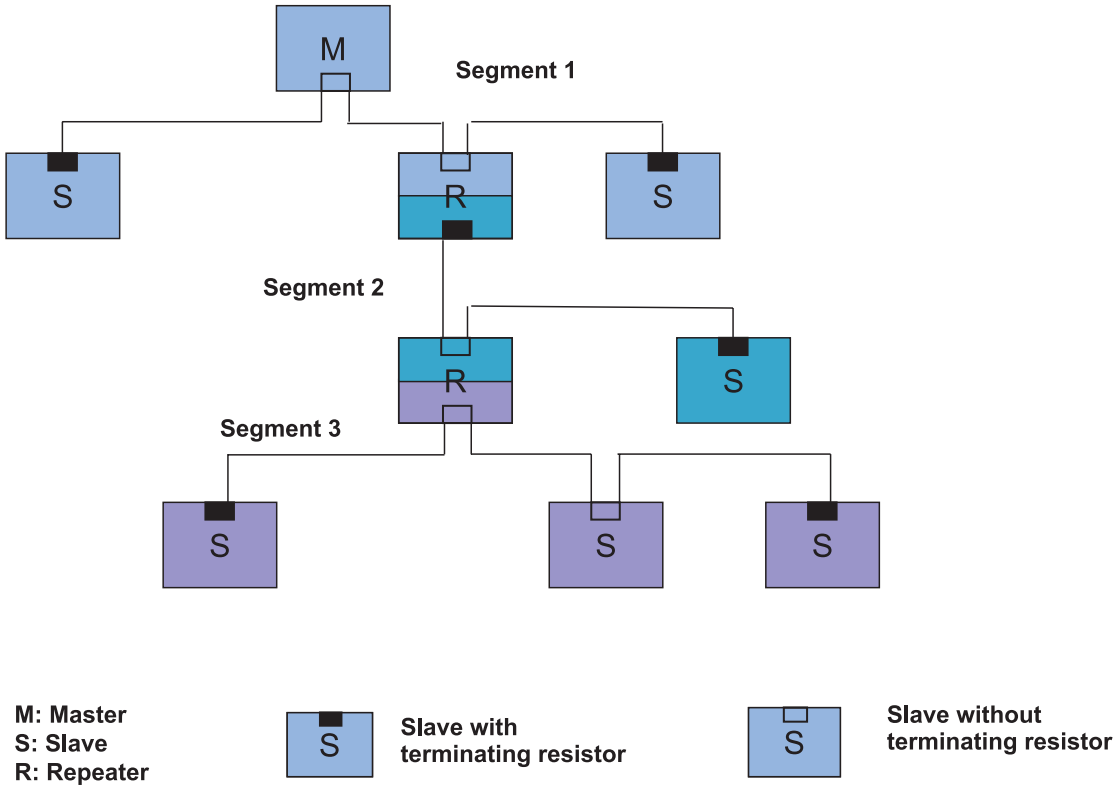


Figure 5: PROFIBUS network with segments (connected via the repeaters) and the bus terminator (in a tree-Topology)

The structure of the network can be as Bus-, Star-, Tree - or Line structure and combinations thereof. It must be noted thereby, that per segment only 32 devices (incl. Repeaters) are connected and the segments have to be connected via the repeaters. The devices to PROFIBUS are also called as node or bus-node.

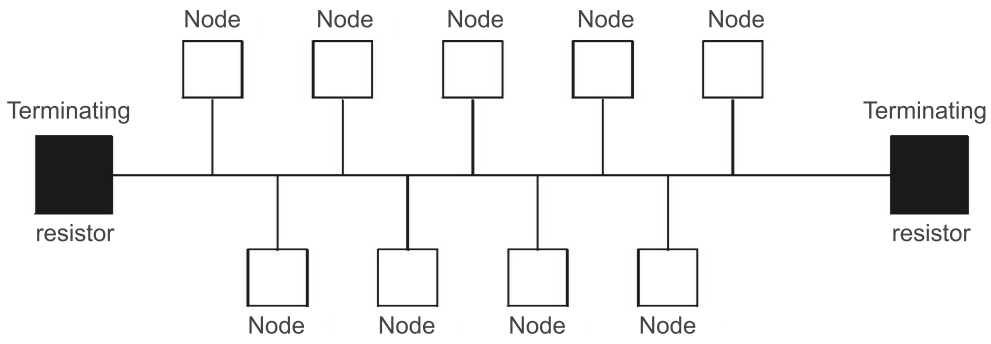


Figure 6: Bus-Topology

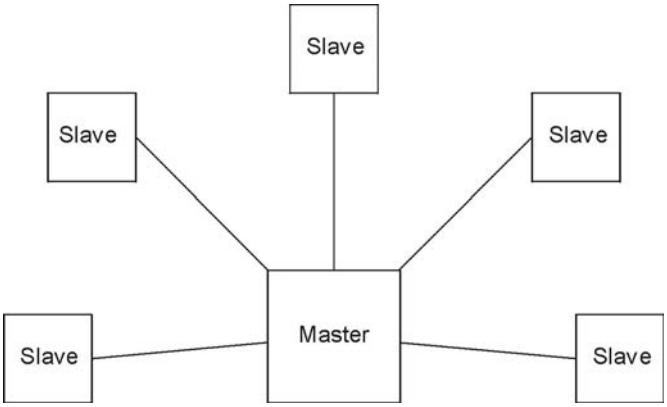


Figure 7: Star-Topology

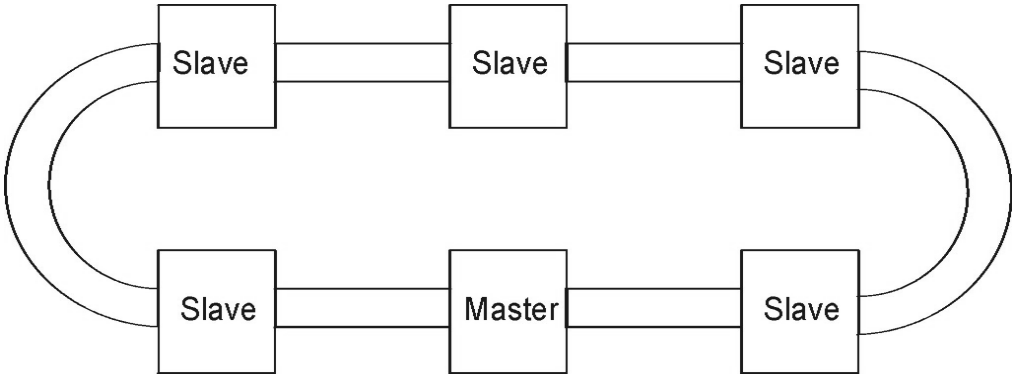


Figure 8: Ring-Topology

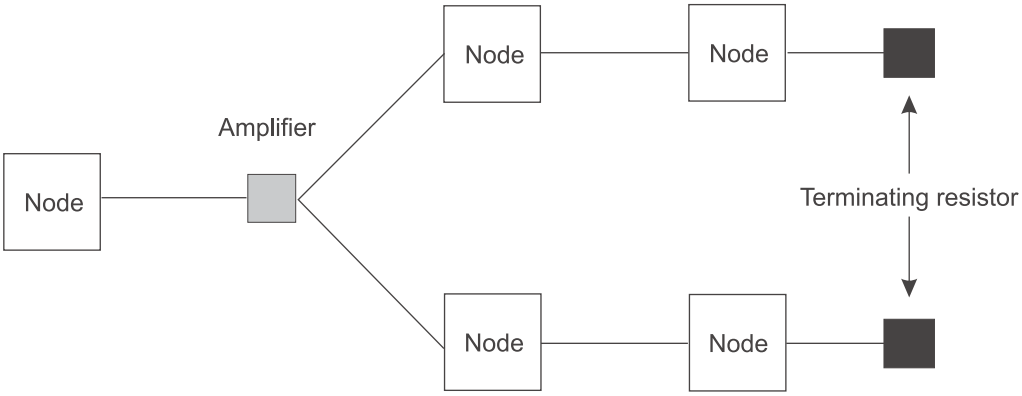


Figure 9: Tree-Topology

Bus-Topology is used in PROFIBUS-DP.

3.2.3 Signals

In PROFIBUS-DP, the exchange of data takes place in the NRZ-Code (non return to zero). While the bit duration does not change thereby the pattern of the signal from binary „0“ or „1“. A character consists of 11 Bits (1 start bit, 8 data bits, 1 parity bit, 1 stop bit). If nothing is being transmitted, then the line lies at „1“. The start bit pulls the line to „0“.

Figure 10: Bus signals in NRZ format

The data line „B“ is specified as the original line in PROFIBUS-DP and the data line „A“ as an inverted line.

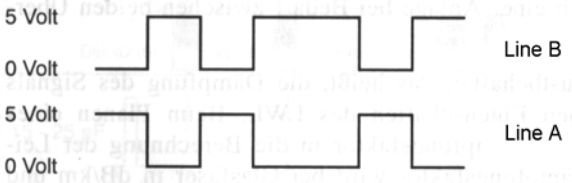


Figure 11: Difference signal on the data lines

The logical signal level „1“ means a positive signal of the line „B“ with respect to „A“ and „0“ means a negative signal of the line „B“ with respect to „A“.

3.2.4 Access

PROFIBUS is a master-slave system, in which however, several masters can be collected on the bus.

A token is exchanged among the masters. The master having the token is permited to communicate with the slaves. After a certain time, the master must pass on the token to the next master. This forms a logiccal token ring between the masters on the bus. This process is called as token passing.

One master communicates (when it has the token) with only one slave at a time. The slaves answer only on request by the master. A slave is permitted to give a data output only when the data is written by the master which has parametrized and configured the slave. The parametrization and configuration of the slaves by the master (or the masters) takes place when the bus is switched on.

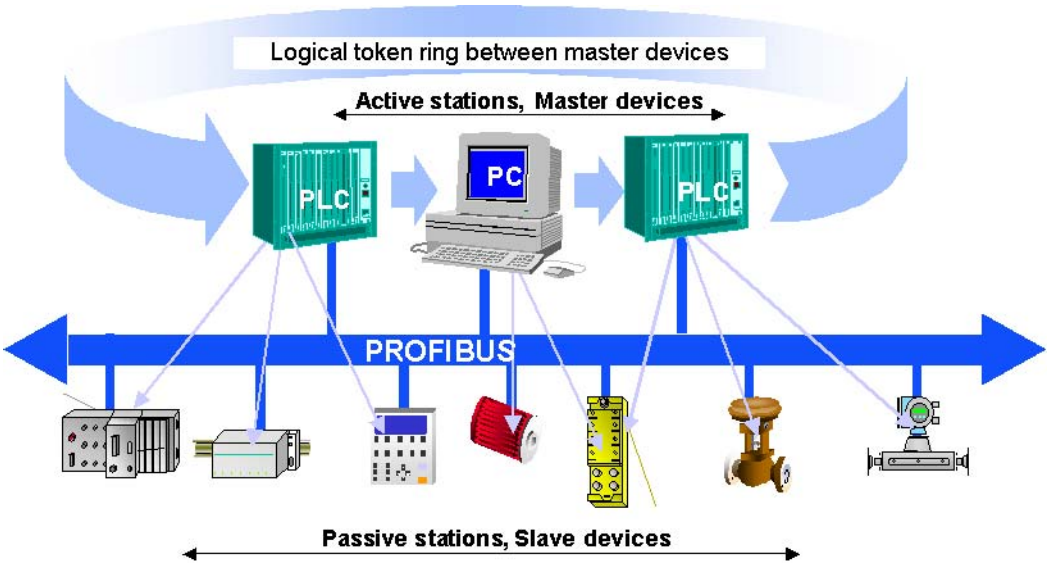


Figure 12: Overview of communication from several Masters to the respective Slaves

3.2.5 Errors

A Hamming distance (measure of error-detection) of HD = 4 is achieved in PROFIBUS due to a high level of error-detection and error-handling on Layer 2, i. e. maximum 3 Bit errors in a telegram occuring simultaneously are detected as error.

3.2.6 Data exchange

PROFIBUS uses different transmission services for data exchange:

The service „Send data with acknowledge" (SDA) is used by PROFIBUS-FMS. That means that data will sent to a master or to a slave and the receiver sends a short acknowledgement by way of reply.

The service „Send and request data with acknowledge" (SRD) is used by PROFIBUS-FMS and PROFIBUS-DP. The output data is sent by a master to a slave here in the form of a message cycle. The slave sends the data in the same message cycle, which, the master receives as the input data.

The service „Send data with no acknowledge" (SDN) is used by PROFIBUS-FMS and PROFIBUS-DP. A master can send broadcast- or multicast-telegrams. These telegrams are not acknowledged by the slaves.

The service „Cyclic send and request data" (CSRD) is used by PROFIBUS-FMS. The data is sent and received cyclic.

Only the services SRD and SDN are used in PROFIBUS-DP.

3.2.7 Structure of a telegram

A PROFIBUS telegram consists of a header and the data area. Information about the source, about the destination and about the type and length of the data in the data area are stored in the header. Furthermore, a check sum for the telegram is calculated and transmitted, using which the receiver checks whether the telegram has been transmitted correctly (whereby he also calculates the check sum).

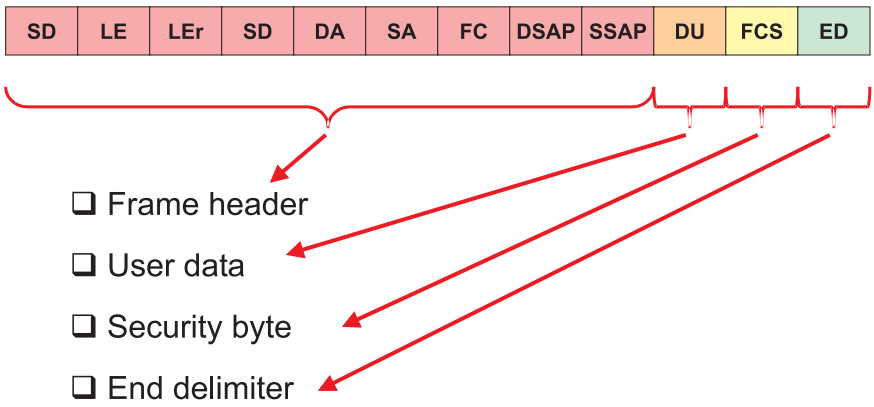


Figure 13: Structure of a PROFIBUS telegram

SD - Start delimiter
LE - Length of the data (DA, SA, FC, DSAP, SSAP, DU)
LEr - Length of the net data (repitition)
DA - Destination address
SA - Source address
FC - Function code
DSAP - Destination service access point
SSAP - Source service access point
DU - Data unit (net data block / data area)
FCS - Frame checking sequence (Check sum of DA, SA, FC, DSAP, SSAP, DU)
ED - End delimiter

3.2.8 Station address

In PROFIBUS, there are different ways of giving a slave a station address. A station address can be specified directly at the slave. The PROFIBUS-Master can also assign the PROFIBUS-Slave an address, via a suitable telegram. See ▹Setting the station address◃ from page 35 onward and ▹PROFIBUS_INIT_SL◃ from page 65 onward.

Setting the station address at the PROFIBUS-Slave:

The station address at the PROFIBUS-DP-Slave is set, as a rule, by using a rotary switch. In the case of an Option Module PROFIBUS-DP-Slave, the setting can be done by using a rotary switch or by initialisiation by the b maXX PLC. In this case, see ▹Setting the station address◃ from page 35 onward and ▹PROFIBUS_INIT_SL◃ from page 65 onward.

Setting the station address via a telegram sent by the PROFIBUS-Master:

The PROFIBUS-Master can set the station address of a slave by sending a telegram. It is differentiated thereby in

- one time setting of the station address
- setting the station address several times

One time setting of the station address:

In case of one time setting of the station address, the master sends a telegram with station address and the PROFIBUS-Slave saves this address internally in a Flash-module. After switching off and switching on once, the PROFIBUS-Slave reports at the bus with this address.

Setting the station address several times:

In case of setting the station address several times, the master sends a telegram with the station address after every bus start. The PROFIBUS-Slave then reports at the bus with this address. On switching off, the slave “loses” the address and at the time of the next bus start, must once again receive a telegram from the master for setting the bus address.

NOTE

In PROFIBUS, the station address is also called as the bus address, address at the bus, device-address, slave-address and module-address depending on the literature, while the station number is also termed as the device number, slave number and module number.

3.3 Data Exchange via the PROFIBUS by use of the Option Module PROFIBUS-DP-Slave

3.3.1 General points on data exchange

In the chapters that follow, it will be illustrated to you as to how an Option Module PROFIBUS-DP-Slave is to be used in a PROFIBUS-DP network.

The basic mechanisms as well as use of the function blocks from the library PROFIBUS_PLC01_20bd01 (or higher) will be shown to you.

The presentations will also be accompanied by an example of step by step building up of a project for PROPROG wt II .

Please refer to the relevant Operation Manual for commissioning the PROFIBUS-Master.

NOTE

The example project is in no way to be looked upon as a fully functional application. The data which are written via the examples of the PROFIBUS-DP-Slave to the b maXX PLC and further to the b maXX controller have no functional meaning from the point of view of the value. Accordingly, make absolutely sure that the drive is not in a ready to start condition!

Of course, the Option Module PROFIBUS-DP-Slave and the function blocks offer a lot of options than what can be presented within the scope of this manual. We cannot also go into all the details of the function blocks here. Details of the function blocks can be obtained from the relevant Online-Help in PROPROG wt II.

3.3.2 Overview of the data exchange

Although the b maXX 4400 together with the b maXX PLC and the Option Module PROFIBUS-DP-Slave is a drive, the PROFIBUS-Profile specification 3 PROFIdrive V3 is not supported.

Reason: The PROFIBUS-Master does not have a direct access to the drive. The PROFIBUS-Master accesses the data of the b maXX PLC (which is programmed in accordance with IEC 61131-3).

As PROFIBUS-Slave, the b maXX 4400 receives the data from the PROFIBUS-Master. In b maXX 4400, the Option Module PROFIBUS-DP-Slave receives the data and passes it on to the b maXX PLC. The data can be processed further in the b maXX PLC (e. g. by means of an electronic cam) and is then transfered to the b maXX controller. Actual control and regulation of the drive takes place here via the b maXX power unit.

In the reverse way, the data can be transmitted to the PROFIBUS-Master.

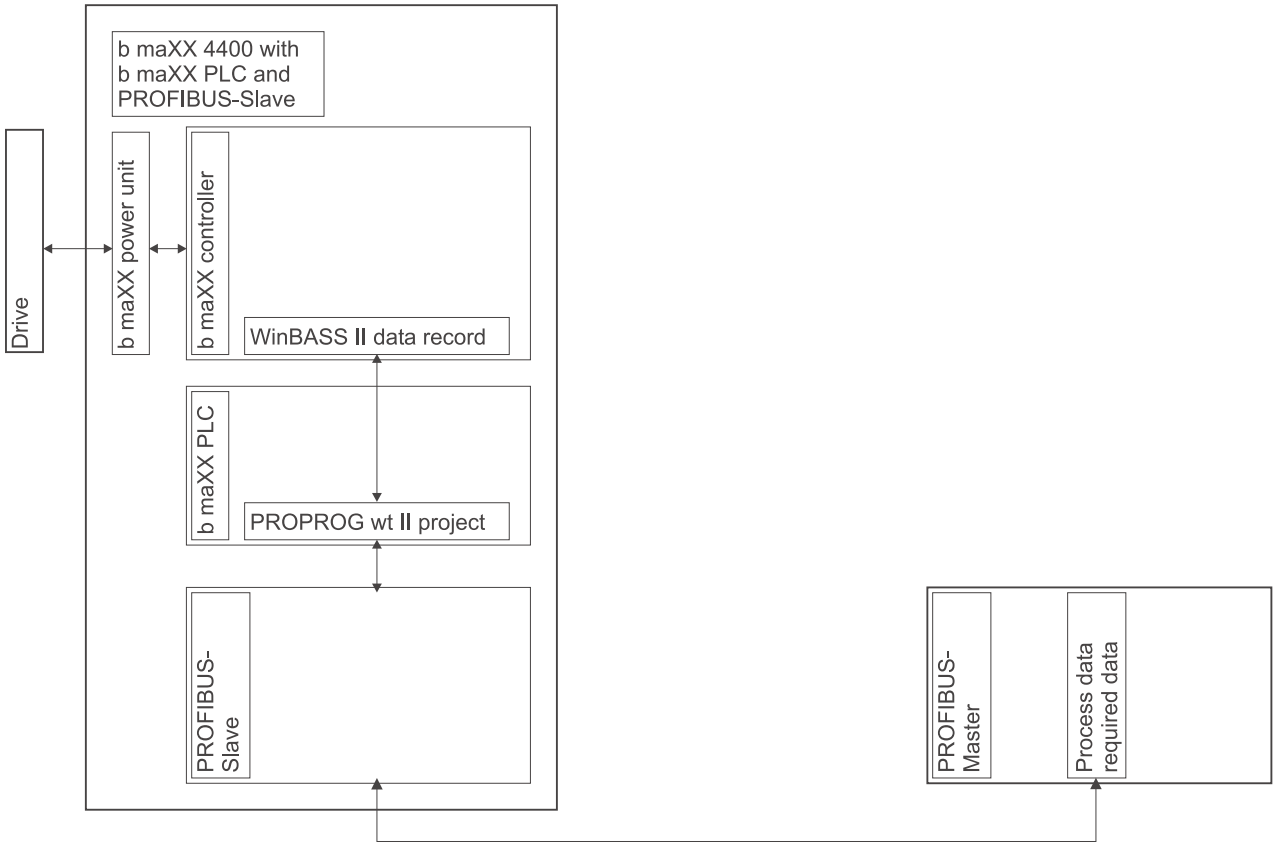


Figure 14: Data flow PROFIBUS-Master → PROFIBUS-Slave → b maXX PLC → b maXX controller → Power unit → Drive and back

Communication: PROFIBUS-Master → PROFIBUS-Slave

The PROFIBUS-Master communicates with the PROFIBUS-Slave via the PROFIBUS telegram. This PROFIBUS telegram contains the service data in the PKW-area and the process data (reference - and actual values) in the PZD-area (see ▹Data telegram having a variable length◃ from page 58 onward.

PROFIBUS-Slave

On the PROFIBUS-Slave, there takes place a conversion of the individual Bytes from the PROFIBUS telegram into the service data and process data.

Communication: PROFIBUS-Slave → b maXX PLC

The service data and process data are assigned to the variables in the PROPROG wt II project in the b maXX PLC, via the PROFIBUS-PLC-Slave configurator (in the b maXX configurator; see ▹Configuration of the devices using the b maXX configurator◃ from page 30 onward).

b maXX PLC

In the b maXX PLC, the data is processed further as a rule (using the variables) e. g. by means of electronic cam. The reference values resulting thereby are transfered to the b maXX controller. The programming of the application program in the b maXX PLC takes place via PROPROG wt II (see Manual PROPROG wt II and Application Manual b maXX PLC).

Communication between the b maXX PLC → b maXX controller

The reference values are transfered by the b maXX PLC to the b maXX controller. In the b maXX controller, these reference values are then the parameter like position setpoint. The programming of the communication between the b maXX PLC and the b maXX controller takes place via PROPROG wt II. The programming of the communication is described in the Application Manual for the b maXX PLC and is not explained in detail in this document.

b maXX controller

In b maXX controller, the control loop is carried out by using the reference values (of the b maXX PLC) and the actual values of the drive (or the b maXX power unit). The conditions for the control loop (e. g. torque limit) can be parametrised.

The parametrisation of the b maXX controller is carried out by using WinBASS II (see Parameter Manual for b maXX 4400).

Communication between the b maXX controller → b maXX power unit

The set values of the parameters of the b maXX controller are transferred to the b maXX power unit.

b maXX power unit / drive

In b maXX power unit, the set values of the parameters of the b maXX controller are converted in current and voltage, at which the motor is run. Actual values (e. g. actual value of the current) are also determined in b maXX power unit. Other actual values (e. g. actual values for the motor speed or the actual position) are also determined by the encoder on the motor and the encoder module on the b maXX controller.

The actual values can be reported in the reverse way, to the PROFIBUS-Master .

3.3.3 Steps to be carried out

Assembly, connection and the commissioning of

- drive,
- b maXX 4400,
- Option Module b maXX PLC (BM4-O-PLC-01),
- Option Module PROFIBUS-DP-Slave (BM4-O-PRO-01-00-01),
- as also of the PROFIBUS-Master used by you

must have been completed successfully.

The following steps must be carried out for being able to carry out data exchange by the PROFIBUS-DP-Slave, using the PROFIBUS network:

- Creation of a PROPROG wt II project for the b maXX PLC.
- Configuration of the devices by means of the b maXX configurator.
- Configuration of the Option Module PROFIBUS-DP-Slave by the PROFIBUS-PLC-Slave Configurator (in the b maXX Configurator).
  Result:
  - List of variables for the service data and the process data for the PROPROG wt II project,
  - GSD file for a PROFIBUS-Master configuration tool
- Programming of the application part in the PROPROG wt II project for the b maXX PLC.
  Result:
  - PROPROG wt II example-project with INIT, service data, process data, diagnosis.
- If required, integration of the GSD file in a PROFIBUS-Master configuration tool.

3.3.4 Creation of a PROPROG wt II project

3.3.4.1 Procedure for creation of a project

A PROPROG wt II project for the b maXX PLC01 is required for being able to use the Option Module PROFIBUS-DP-Slave with the PROFIBUS function blocks. If you have still not created a project for your application, then create this please with the template of the bmaXX4400 with configurator. For this purpose, you require a PROPROG wt II Version 3.0 from Build 262 onwards. The version no. of the PROPROG wt II can be found on the case of the installation CD for PROPROG wt II or in PROPROG wt II itself at Menu item **? \ Info**. Check whether the library BM_TYPES_20bd05 (or higher) is present in your PROPROG wt II project. If not, please integrate this library in your project. It contains important types of data for PROFIBUS. Integrate thereafter the Library PROFIBUS_PLC01_20bd01 in your project.

3.3.4.2 Example: Creation of the project „ProfibusSlave_Example"

The example project „ProfibusSlave_Example" has been created by using the template *bmaXX4400 with configurator* and the libraries BM_TYPES_20bd05 and PROFIBUS_PLC01_20bd01 integrated therein.

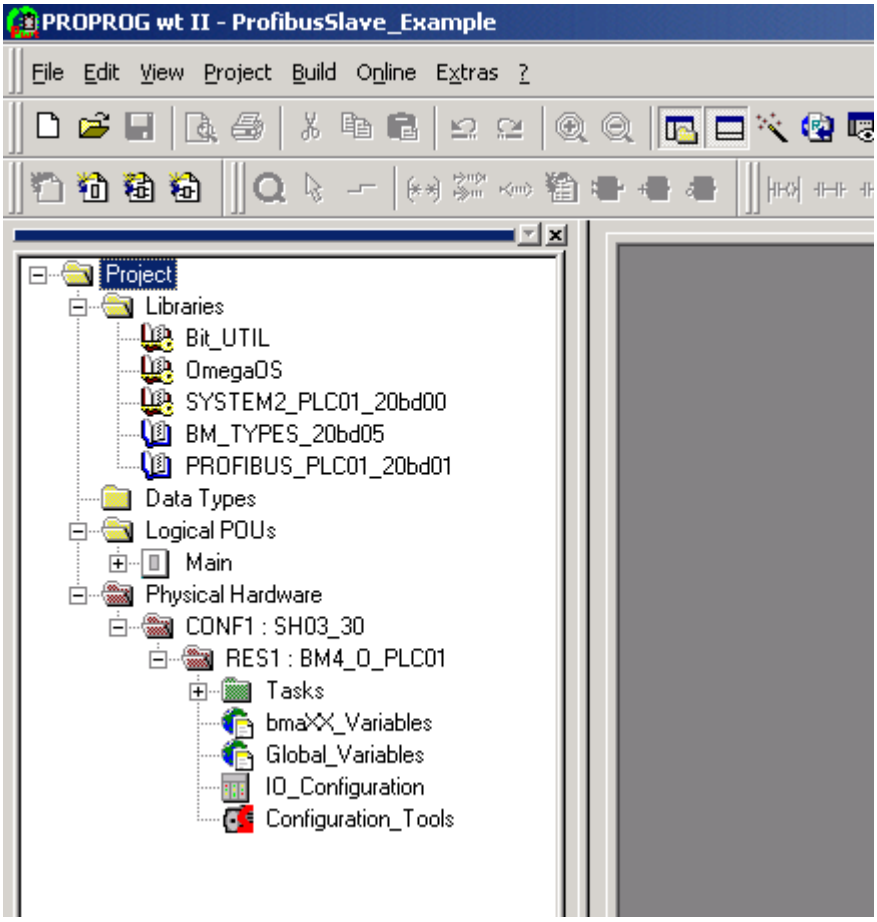


Figure 15: Example: Creation of the project „ProfibusSlave_Example"

3.3.4.3 Creating a global variable for data exchange

A global variable is required for the exchange of data between the Option Module BM4-O-PRO-01 and the function blocks. It has no further significance for the user. This global variable has already been created in your project, if the project has been created by using the template *bmaXX4400 with the configurator*. This global variable is connected to the input/output parameter _BASE on the function blocks for the PROFIBUS-Slave.

Depending on the slot used on the Option Module (Slots G to M), the global variables _ProfibusSlave_Ctrl_Slot_G upto _ProfibusSlave_Ctrl_Slot_M are available for you.

You can also find these in the worksheet: „Global_Variables":

```
(* Option module PROFIBUS-Slave (BM4-O-PRO-01) *)
   _ProfibusSlave_Ctrl_Slot_G    AT %MB3.2001792   :   PROFIBUS_PLC_SL_BMSTRUCT;
   _ProfibusSlave_Ctrl_Slot_H    AT %MB3.3001792   :   PROFIBUS_PLC_SL_BMSTRUCT;
   _ProfibusSlave_Ctrl_Slot_J    AT %MB3.4001792   :   PROFIBUS_PLC_SL_BMSTRUCT;
   _ProfibusSlave_Ctrl_Slot_K    AT %MB3.5001792   :   PROFIBUS_PLC_SL_BMSTRUCT;
   _ProfibusSlave_Ctrl_Slot_L    AT %MB3.6001792   :   PROFIBUS_PLC_SL_BMSTRUCT;
   _ProfibusSlave_Ctrl_Slot_M    AT %MB3.7001792   :   PROFIBUS_PLC_SL_BMSTRUCT;
```

Figure 16: Global variables for the Option Module PROFIBUS-DP-Slave depending on the slot

If the global variable _ProfibusSlave_Ctrl_Slot_G (upto _ProfibusSlave_Ctrl_Slot_M) of the data type PROFIBUS_PLC_SL_BMSTRUCT required for the slot of the Option Module is not present in the project, then create it, depending on the slot (Slot G to M). This variable must be declared as the global variable and placed at the base address for the PROFIBUS-Slave communication of the Option Module BM4-O-PRO-01 The base address depends on the slot :

Slot	Base address for Profibus-Slave communication
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

NOTE

The variable name is replaced by * in the illustration below. Furthermore, it is assumed that the Option Module PROFIBUS-DP-Slave for b maXX PLC has been plugged in the slot G.

As a result , the register *.u_BAUDRATE is accessed as follows :

```
_ProfibusSlave_Ctrl_Slot_G.u_BAUDRATE
```

where:

`_ProfibusSlave_Ctrl_Slot_G`	is the variable name with the data type short designation „_" for struct
`u_BAUDRATE`	the register in which the Option Module PROFIBUS-DP-Slave enters the baud rate recognised automatically .

3.3.5 Configuration of the devices using the b maXX configurator

In the PROPROG wt II project, which you have already created (under ▷Creation of a PROPROG wt II project ◁ on page 28), you will find the button for the configurators (Configuration_Tools). On giving a double click on these buttons, the configurators window gets opened, among other things, by the b maXX configurator (Drive_Configuration).

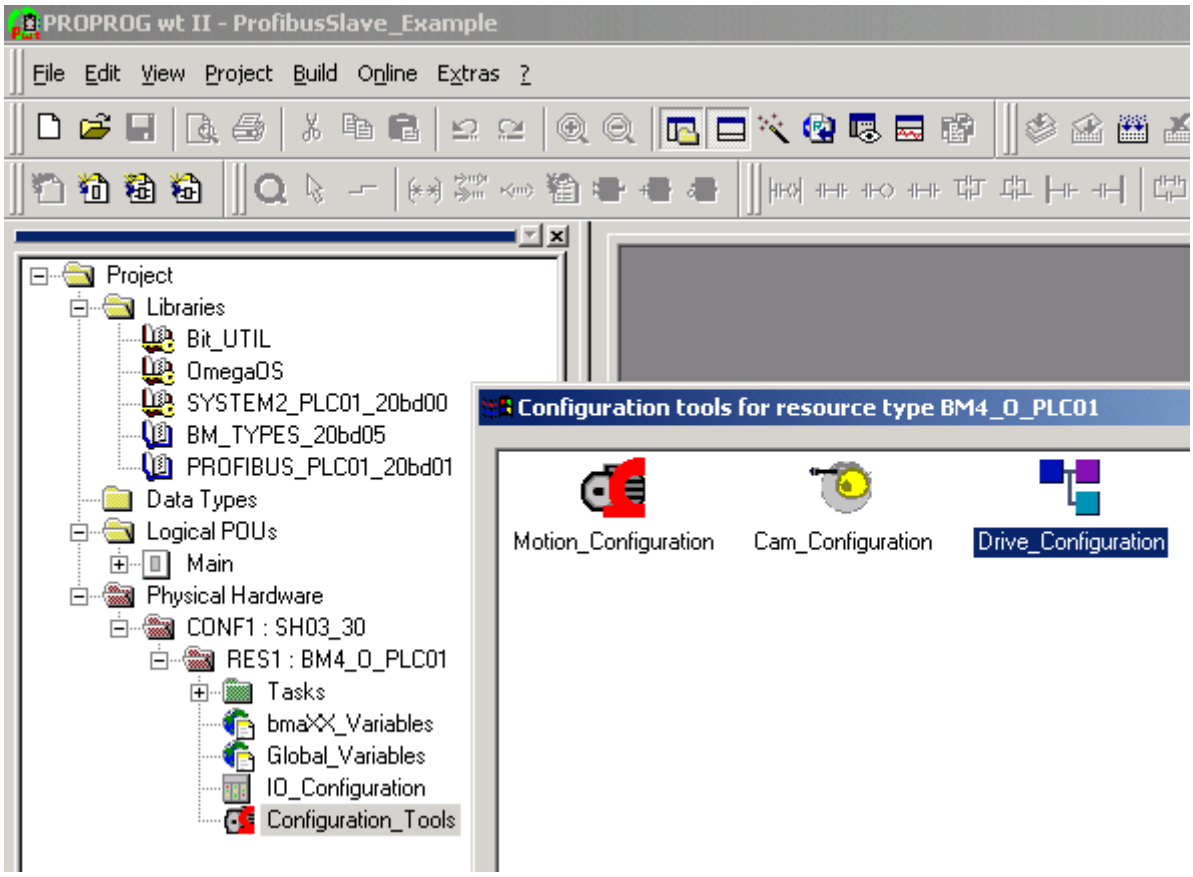


Figure 17: Opening the b maXX configurator

Click now on the button „Drive_Configuration" for opening the b maXX configurator.

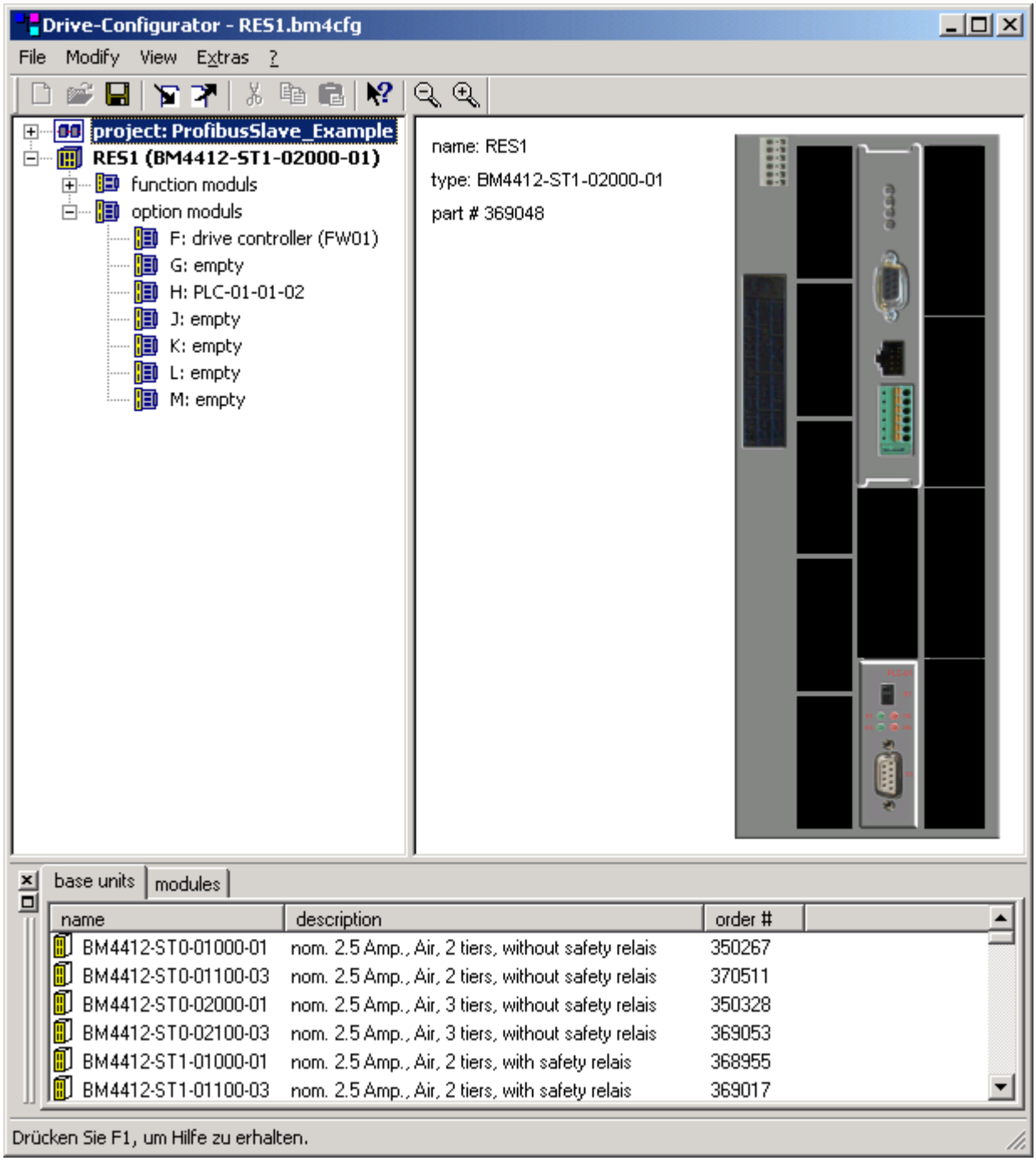


Figure 18: b maXX configurator

Description of the steps :

- Selection of the b maXX 4000 basic unit (incl. b maXX controller)
- Selection of the Option Module b maXX PLC
- Selection of the Option Module PROFIBUS-DP-Slave

3.3.5.1 Selection of the b maXX 4000 basic unit (incl. b maXX controller)

A b maXX basic unit incl. b maXX controller is set by default in the project tree of the b maXX configurator. The b maXX basic unit BM4412-ST1-02000-01 is set in Fig.18, under resource „RES1".

If you want to use any other b maXX basic unit, then select the point: „add/change" by using the context menu for the resource.

- Select your b maXX basic unit from the window „Select device type“.
- Confirm the selection with „OK".

The new b maXX basic unit (incl. b maXX controller) is thus accepted and the grafic matched.

The b maXX basic unit BM4412-ST0-01000-01 is used In the example project.

3.3.5.2 Selection of the Option Module b maXX PLC

- Open the entries of the Option Module (Click on „+" before „Option Modules") in the project tree of the b maXX configurator
- A b maXX PLC in slot „H" is set by default . The b maXX PLC BM4-O-PLC-01-01-02 is set in Fig.18 slot „H".
- If you want to use any other b maXX PLC, then select the point: „add/change“ by using the context menu for the slot “H”.
- Select your b maXX PLC from the window „Select Option Module for Slot H"
- Confirm the selection by clicking on „OK".

The new b maXX PLC is thus accepted and the graphic matched.

The b maXX PLC BM4-O-PLC-01-00-02 at slot „H" is used in the example project .

3.3.5.3 Selection of the Option Module PROFIBUS-DP-Slave

- Select the point „add/change“ by using the context menu of the slot „G".
- Select the Option Module PROFIBUS-DP-Slave for b maXX PLC (BM4-O-PRO-01-00-01) from the window „Select Option Module for Slot G".
- Confirm the selection by clicking on „OK".

The Option Module PROFIBUS-DP-Slave for b maXX PLC is thus accepted and the graphic matched.

The Option Module PROFIBUS-DP-Slave for b maXX PLC (BM4-O-PRO-01-00-01) at slot „G" is used in the example project.

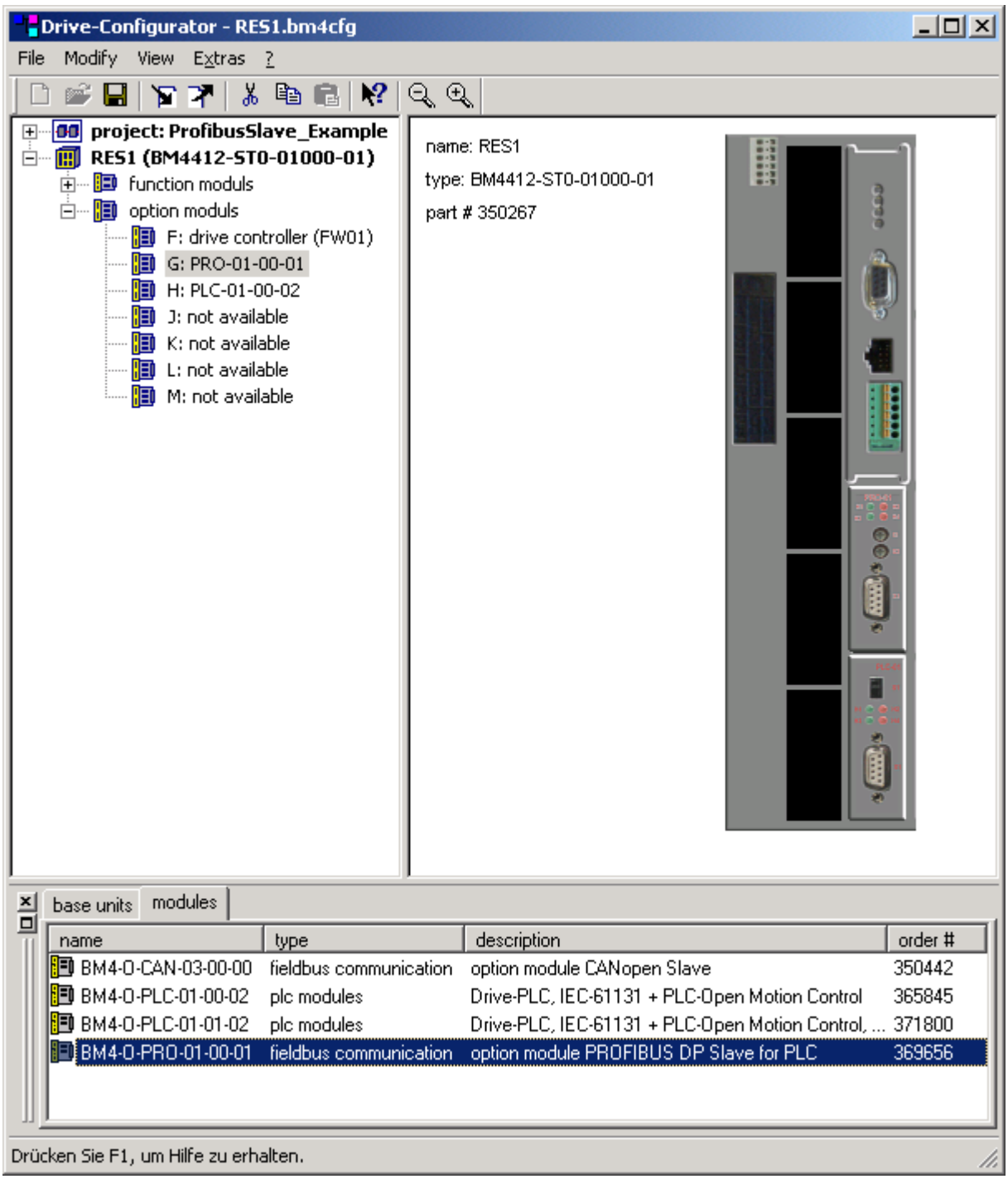


Figure 19: b maXX configurator after the configuration of the devices

Further with „Configuring the Option Module PROFIBUS-DP-Slave by using the PROFIBUS-PLC-Slave configurator“.

3.3.6 Configuring the Option Module PROFIBUS-DP-Slave by using the PROFIBUS-PLC-Slave configurator

The operation of the PROFIBUS-PLC-Slave configurator is illustrated in this chapter. The PROFIBUS-PLC-Slave configurator

- generates parts of the datasheet of the global variables „bmaXX_Variables" for the PROPROG wt II project
  (if required, first downloads parts of an already existing datasheet of the global variables „bmaXX_Variables" of the PROPROG wt II projects)
- generates the GSD file for the key data of the devices describing the device „b maXX 4400 with b maXX PLC and the Option Module PROFIBUS-DP-Slave" for the PROFIBUS-Master configurators.

If required, open the PROFIBUS-Slave-configurator from PROPROG wt II via Configuration_Tools → Drive_Configuration → „Option Module \ G: PRO-01-00-01" - context menu Point „Configure“.

3.3.6.1 Setting the station address

In this chapter, it is explained how you can set the station address of the Option Module PROFIBUS-DP-Slave by using the PROFIBUS-PLC-Slave configurator and the b maXX PLC.

There are several ways of setting the station address of the Option Module PROFIBUS-DP-Slave.

- Setting the station address by using the rotary switch.
- Setting the station address by initialisation by b maXX PLC
- Setting the station address using the PROFIBUS-Master (one time or several times), see ▹Station address ◃ on page 24 and ▹PROFIBUS_INIT_SL◃ from page 65 onward.

For these ways it is necessary to inform the b maXX PLC as to which settings it has to undertake during initalisation (using the function block PROFIBUS_INIT_SL). This setting is carried out in the EditBox „Station address".

Setting the station address:

This is governed by the Table below:

EditBox „Station address"	rotary switch on the PROFIBUS-Slave		Meaning
	decimal	hexadecimal	
0	-	-	not permitted as station address
1..125	-	-	Station address 1..125 via b maXX PLC, FB PROFIBUS_INIT_SL (Example 1)
126	-	-	One time setting of the station address (Example3a)
127	-	-	Setting the station address several times (Example 4)
activated CheckBox „from DIP-Switch" (Default)	0	00	not permitted as station address; error message on initialisation of the Option Module PROFIBUS-DP-Slave by the b maXX PLC on the FB PROFIBUS-Slave
	1..125	01..7D	Station address 1..125 by using the rotary switch (Example 2)
	126..255	7E..FF	One time setting of the station address (Example3b)
129..255	-	-	Reserved

- Click in the PROFIBUS-PLC-Slave configurator in the EditBox „Station address".
- Enter the desired station address 1 (..125) or 126 or 127 or activate CheckBox „from DIP-Switch". Other values are not permitted and result in an error message.

In our example project, station address „6" is set.

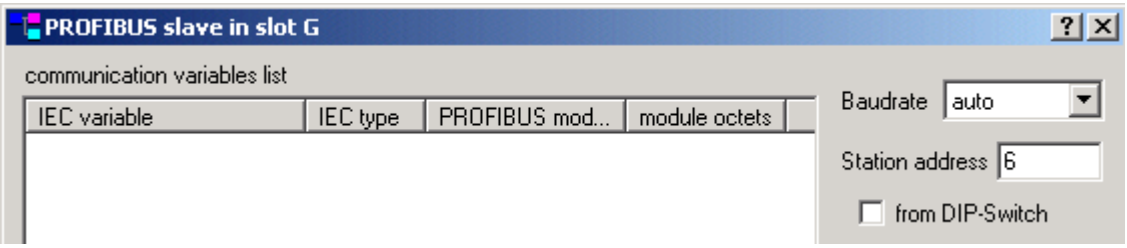


Figure 20: Setting the station address „6"

The station address is available after configuring the Option Module PROFIBUS-DP-Slave as a variable „us_PBS_<SlotName>_Station" in the datasheet for the global variabls „bmaXX_Variables" in the PROPROG wt II project.

In our example, the Option Module is pluged in Slot G and the name of the variable is us_PBS_G_Station.

Setting the station address - examples

Example 1: set the station address „6" by using the b maXX PLC:

- Click in the PROFIBUS-PLC-Slave configurator in the EditBox „Station address"
- Enter the value 6 in the EditBox „Station address"
  (The variable us_PBS_G_Station receives the value 6.)

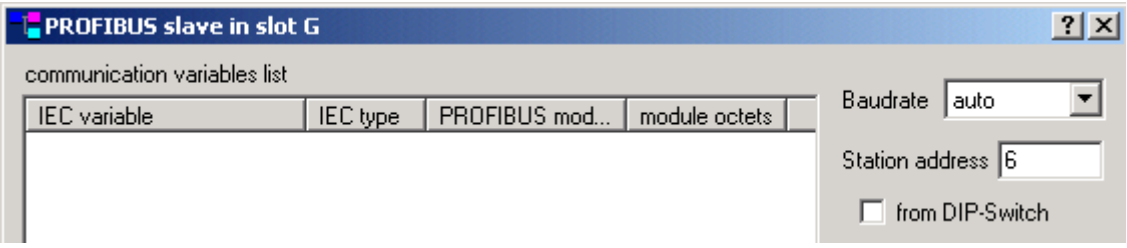
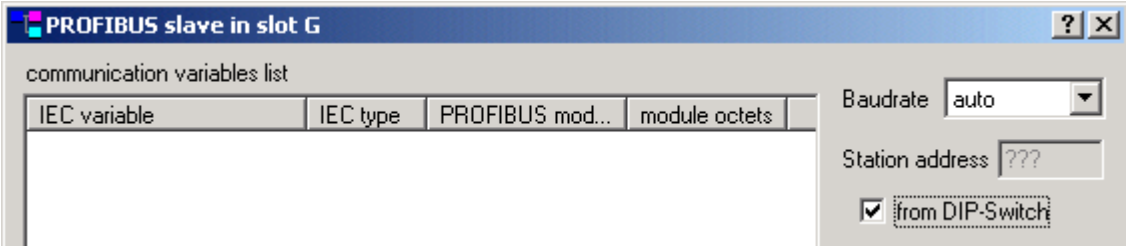


Figure 21: Setting the station address „6" by using the b maXX PLC (rotary switch inoperable)

Example 2: set the station address „6" by using the rotary switch:

- Activate the CheckBox „from DIP-Switch"
  (The variable us_PBS_G_Station receives the value 128)
- Set the station address at 6 (06 hex) by using the rotary switch



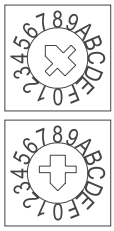


Figure 22: Setting the station address „6" by using the rotary switch (rotary switch „6")

Example 3a: One time setting of the station address by PROFIBUS-Master:

- Click in the PROFIBUS-PLC-Slave configurator in the EditBox „Station address"
- Enter the value 126 in the EditBox „Station address"
  (The variable us_PBS_G_Station receives the value 126.)
- Set the station address via the PROFIBUS-Master.

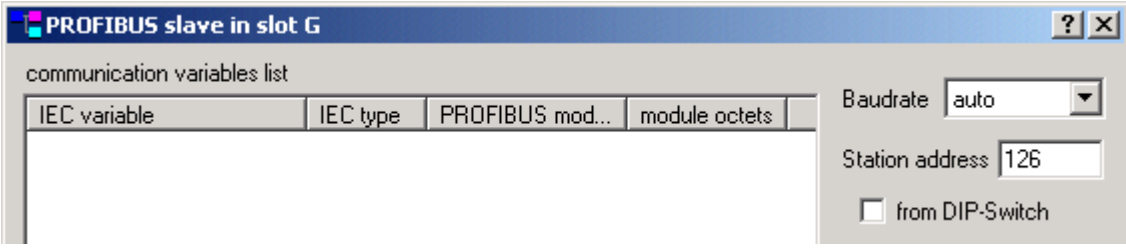


Figure 23: One time setting of the station address by PROFIBUS-Master (rotary switch inoperable)

The station address is stored in an internal Flash-Module and is available for the Option Module PROFIBUS-DP-Slave after switching off and again switching on.

Example 3b: One time setting of the station address by PROFIBUS-Master:

- Activate the CheckBox „from DIP-Switch"
  (The variable us_PBS_G_Station receives the value 128.)
- Set the station address at a value greater than 125 (greater than 7D hex) by using the rotary switch
- Set the station address via the PROFIBUS-Master

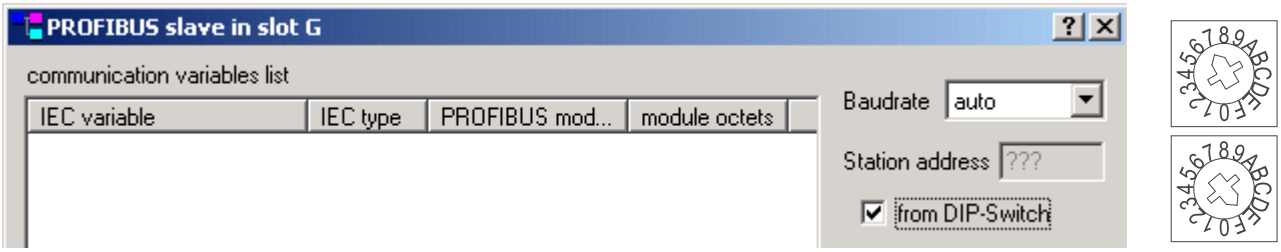


Figure 24: One time setting of the station address by the PROFIBUS-Master + rotary switch „126 (7E hex)"

The station address is stored in an internal Flash-Module and is available for the Option Module PROFIBUS-DP-Slave after switching off and again switching on.

Example 4: Setting the station address several times by the PROFIBUS-Master:

- Click in the PROFIBUS-PLC-Slave configurator in the EditBox „Station address"
- Enter the value 127 in the EditBox „Station address"
  (The variable us_PBS_G_Station receives the value 127.)
- Set the station address via the PROFIBUS-Master.

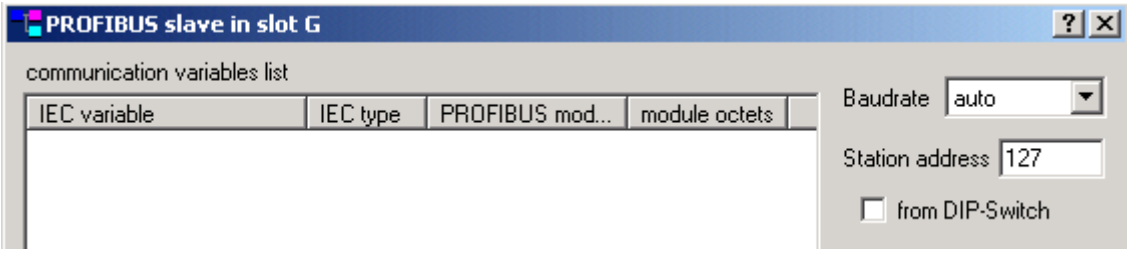


Figure 25: Setting the station address by the PROFIBUS-Master„127" (rotary switch inoperable)

The station address is not stored and gets deleted by switching off and again switching on. In this case, the PROFIBUS-Master must transmit the station address once again.

3.3.6.2 Creation of variables

In this chapter, it is explained, how to create a IEC 61131-3 variable and assign it to a PROFIBUS-module. This variable is then available to you in the PROPROG wt II project, in the datasheet for the global variables „bmaXX_Variables", for programming your application.

In case of the process data, an automatic assignment is effected between IEC 61131-3 variable and the data in the area of the process data (PZD-area) of a PROFIBUS-telegram.

In case of the service data, an automatic assignment is effected between IEC 61131-3 variable and a parameter number (= 4. Byte of the module recognition). The PROFIBUS-Master can access the IEC 61131-3 variable by communication of the service data and this parameter number.

INPUT module: The b maXX PLC passes on the data from the variables to the Option Module PROFIBUS-DP-Slave and, this in turn, transmits the data to the PROFIBUS-Master.
From the point of view of the PROFIBUS-Master, this data are actual values of the b maXX, the data are read.

OUTPUT module: The PROFIBUS-Master transmits the data to the Option Module PROFIBUS-DP-Slave which is passed on by it to the b maXX PLC. The data in the variables is available in the b maXX PLC.
From the point of view of the PROFIBUS-Master, this data are reference values of the b maXX, the data are written.

A variable is created by clicking on the button „Add“ in PROFIBUS-PLC-Slave configurator.

- Click on the button „Add"

The window „Communication variable settings“ gets opened.

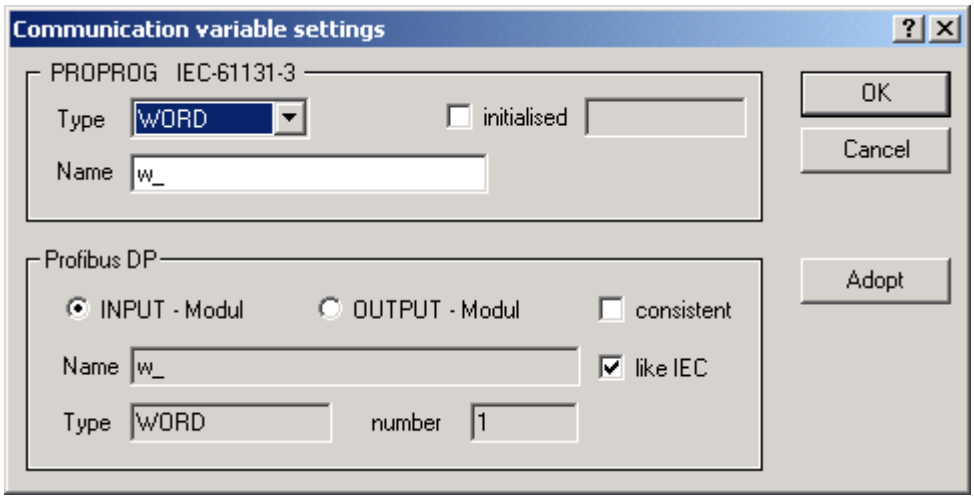


Figure 26: Opening of the window : „Communication-variable settings"

In the EditBox „Type", in the area PROPROG IEC 61131-3, select the data type of the variables. The IEC 61131-3 data types BOOL, SINT, USINT, INT, UINT, DINT, UDINT, BYTE, WORD, DWORD and REAL are available for selection. Select the data type UDINT for our example.

- Click in the EditBox „Type"
- Select the desired data type from the list.
  The data type UDINT has been selected for the example.

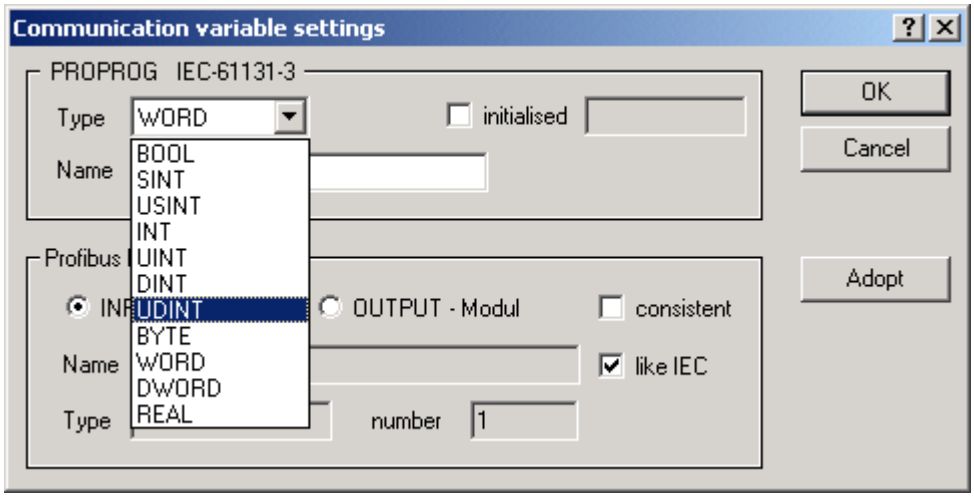


Figure 27: Selecting the data type

In the EditBox „Name", there appears then the corresponding short designation for the data type for the variable. You can specify now the name of the variable (as per the short designation for the data type) in the EditBox „Name".

- Specify the desired variable name in the EditBox „Name".
  The variable name „ud_MyReadVar1" is specified in the example.

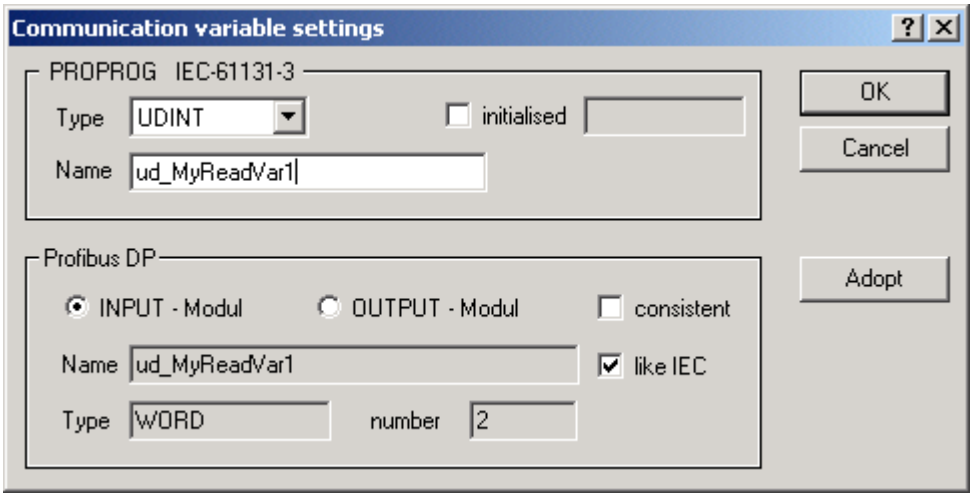


Figure 28: Specifying the variable name

You can specify the start value for your variable in the EditBox „initialised“

- Click on the CheckBox „initialised" (on the left of the EditBox „initialised") .
- Enter the start value for the variable in the EditBox „initialised“. The start value of “1000” is specified in the example.
- Click on the RadioButton „Input-Modul“ in the area PROFIBUS-DP.
- Activate the CheckBox „consistent“ for consistent transferring this module on to the PROFIBUS.

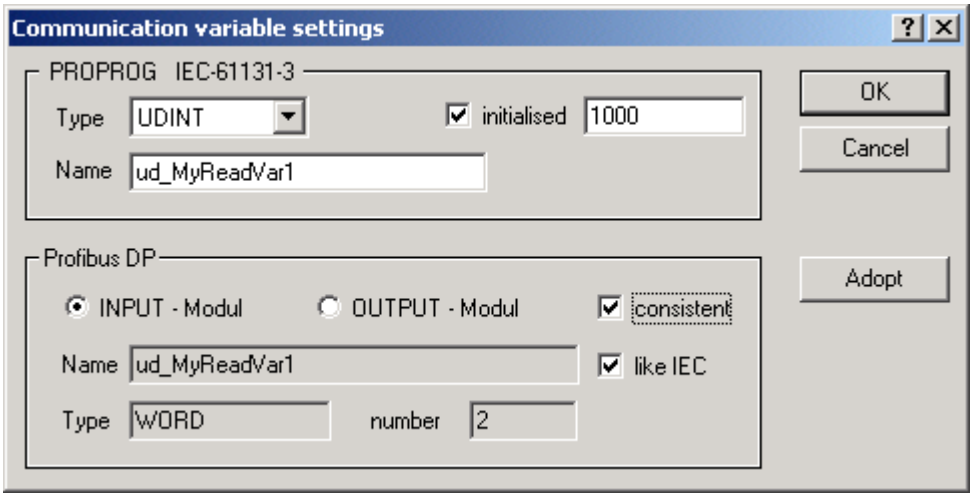


Figure 29: Specifying initialisation value

With that, creation of the variables is completed. Click on Button „OK"

The window „Communication variable settings" is closed and the new variable is displayed in the window „PROFIBUS slave in slot x" (x = G..M) .

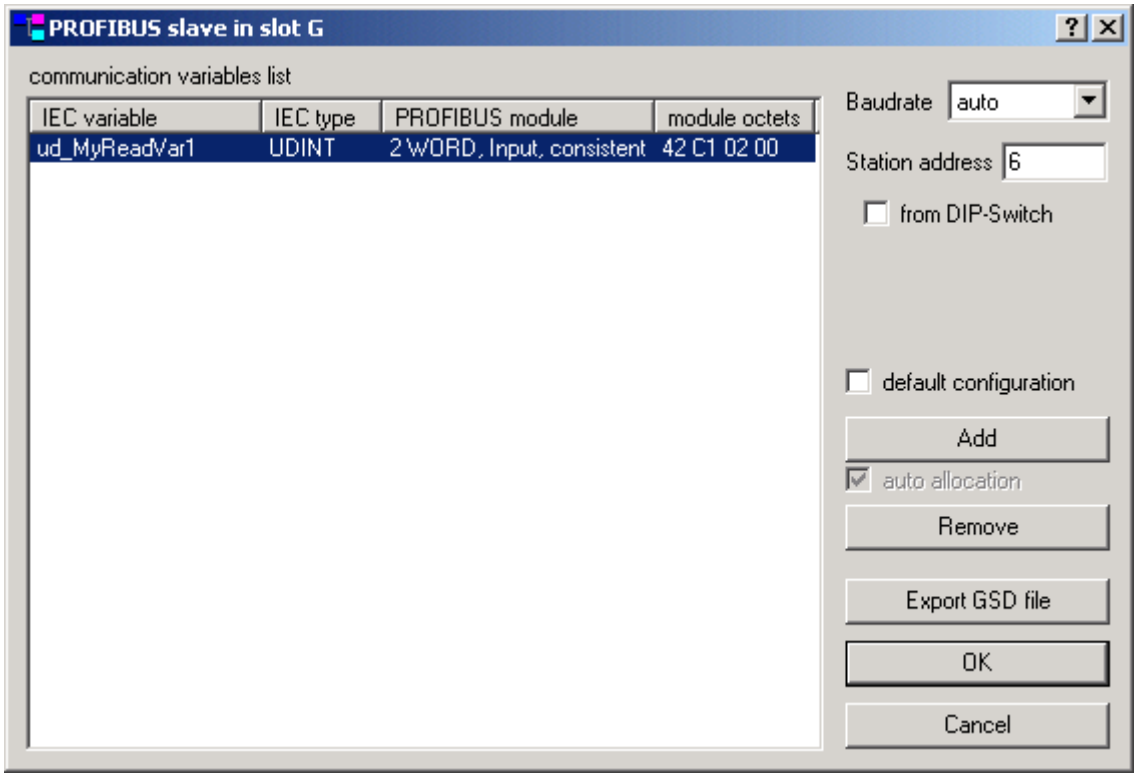


Figure 30: Variable „ud_MyReadVar1"

Below, we are creating still four more variables.

These are: the variable „ud_MyReadVar2" of the data type UDINT,

the variable„ud_MyWriteVar1" of the data type UDINT,

the variable„u_MyWriteVar2" of the data type UINT and

the variable „di_MyReadVar3" of the data type DINT.

- Click on the Button „Add"
- Click in the EditBox „Type" and select the data type UDINT from the list.
- Specify the name of the variable „ud_MyReadVar2" in the EditBox „Name".
- Click on the CheckBox (on the left of the EditBox „initialised") and enter the start value “2000” for the variable in the EditBox „initialised“
- Click on the RadioButton „Input-Modul“ in the area PROFIBUS-DP.
- Activate the CheckBox „consistent“ for consistent transferring this module on to the PROFIBUS

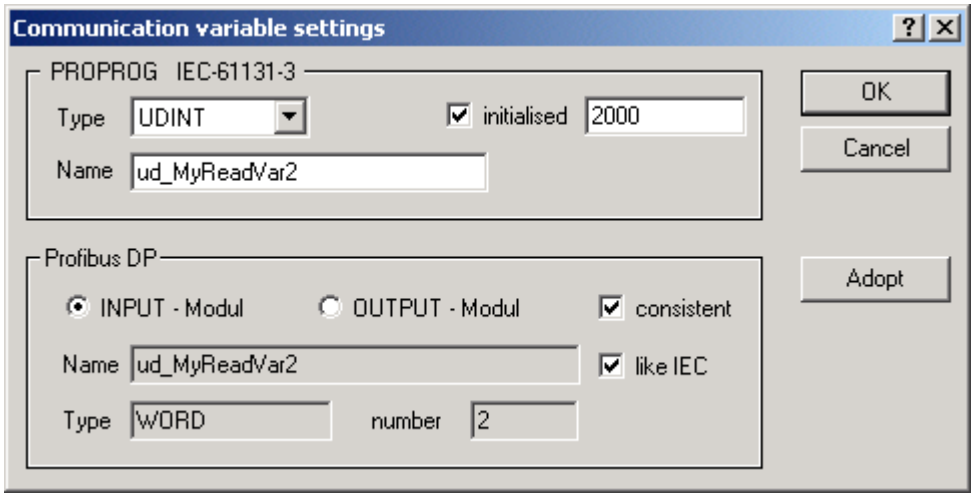


Figure 31: Variable „ud_MyReadVar2"

- Click on the Button „Adopt“.
- Select the data type UDINT from the list in the EditBox „Typ" (this data type should have already been entered by the PROFIBUS-PLC-Slave configurator
- Specify the name of the variable „ud_MyWriteVar1" in the EditBox „Name"
- Click on the CheckBox (on the left of the EditBox „initialised") and enter the start value “1500” for the variable in the EditBox „initialised“
- Click on the RadioButton „OUTPUT-Modul".
- Activate the CheckBox „consistent“ for consistent transferring this module on to the PROFIBUS.

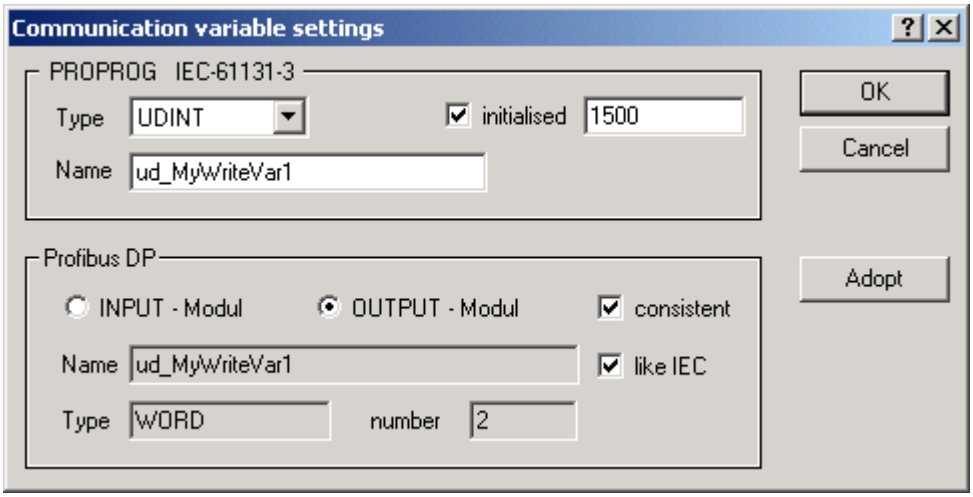


Figure 32: „ud_MyWriteVar1"

- Click on the Button „Adopt".
- Select the data type UINT from the list in the EditBox „Type"
- Specify the name of the variable „u_MyWriteVar2" in the EditBox „Name"
- This variable does not receive any start value.
- Click on the RadioButton „OUTPUT-Modul".
- Activate the CheckBox „consistent“ for consistent transferring this Module on to the PROFIBUS.

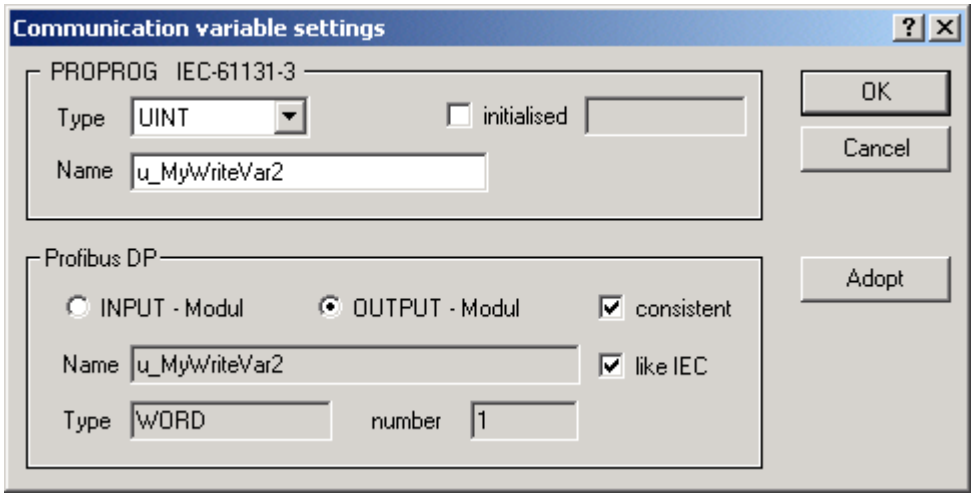


Figure 33: „u_MyWriteVar2"

- Click on the Button „Adopt“.
- Select the data type DINT from the list in the EditBox „Typ"
- Specify the name of the variable „di_MyReadVar3" in the EditBox „Name"
- Click on the CheckBox (on the left of the EditBox „initialised“) and enter the start value “3000” for the variable in the EditBox „initialised“

- Click on the RadioButton „INPUT-Module" .
- Activate the CheckBox „consistent“ for consistent transferring this Module on to the PROFIBUS.

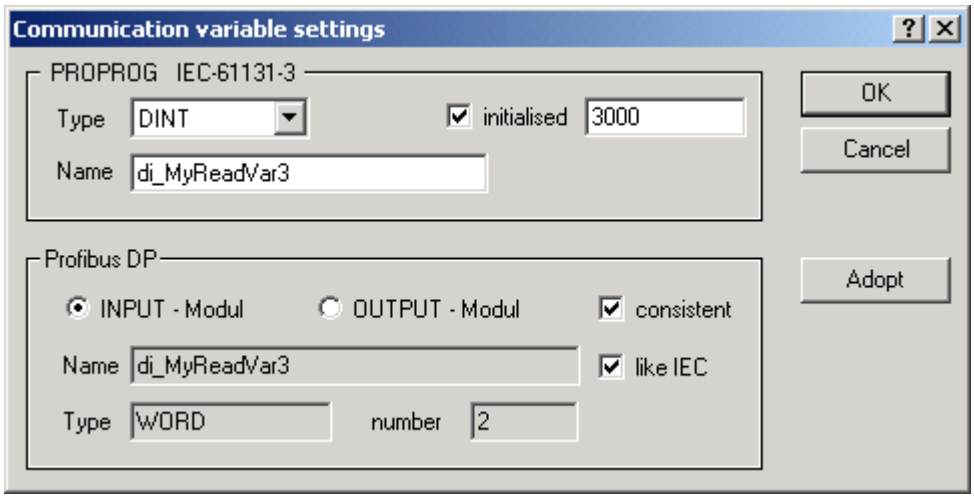


Figure 34: „di_MyReadVar3"

- Click on the Button „OK". The window „Communication variable settings“ is closed and the new variable is displayed in the window „PROFIBUS slave in slot x" (x = G..M) .

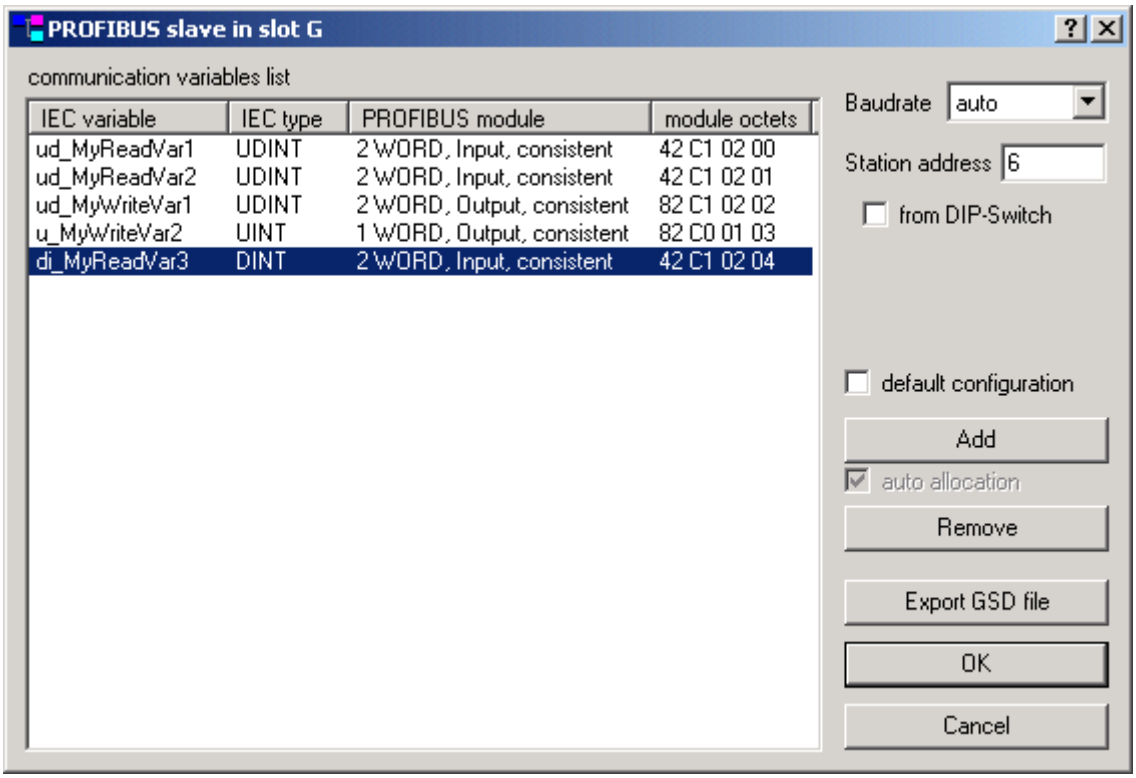


Figure 35: Variables created

With that, creation of the variables „ud_MyReadVar1", „ud_MyReadVar2", „ud_MyWriteVar1", „ud_MyWriteVar2" and „di_MyReadVar3" is completed.

Please read the section: ▹Generation of a GSD-file ◃ on page 45 for generating a GSD-file.

Terminate the PROFIBUS-PLC-Slave configurator by clicking on the Button „OK".

Terminate the b maXX configurator by using File \ Exit and confirm the query „...Save?" with „Yes".

The datasheet for the global variables „bmaXX_Variables" is generated now for the PROPROG wt II project and inserted in the project tree or updated. Variables for programming the application are available in PROPROG wt II project.

```
ud_MyReadVar1 AT %MD3.2002560 : UDINT := UDINT#1000; (* INPUT 42 C1 02 00 *)
ud_MyReadVar2 AT %MD3.2002564 : UDINT := UDINT#2000; (* INPUT 42 C1 02 01 *)
ud_MyWriteVar1 AT %MD3.2002568 : UDINT := UDINT#1500; (* OUTPUT 82 C1 02 02 *)
u_MyWriteVar2 AT %MW3.2002572 : UINT; (* OUTPUT 82 C0 01 03 *)
di_MyReadVar3 AT %MD3.2002576 : DINT := DINT#3000; (* INPUT 42 C1 02 04 *)
```

Figure 36: Datasheet „bmaXX_Variables" for global variables

After having created the variables covered in the section: “Creation of Variables" in the PROPROG wt II project, the GSD-file can be generated now. The example setting of variables can be found in ▹Example setting the PROPROG wt II project variables◃ from page 80 onward.

If you have not created any independent variables, then you can also press the Default-Button. In that case, the Default-Setting for the variables is used. The Default-Setting for the variables can be found in ▹Default setting PROPROG wt II project variables◃ from page 79 onward.

NOTE

Clicking on the CheckBox „default configuration" deletes the variables eventually present in the PROFIBUS-PLC-Slave configurator!

3.3.6.3 Generation of a GSD-file

The GSD-file (GSD [Geräte-Stamm-Datei] master file containing key data of the devices, electronic Datasheet) describes a PROFIBUS device on the PROFIBUS. Among other things, it contains the information necessary for communication with the PROFIBUS device. This GSD-file is required for being able to use the PROFIBUS device „b maXX 4400 together with b maXX PLC and Option Module PROFIBUS-DP-Slave" in a PROFIBUS-Master configurator. Extension of the name of the GSD-file is „gsd" (e. g. MyGSD-File.gsd).

NOTE

The PROFIBUS-Slave (device b maXX 4400 together with b maXX PLC and Option Module PROFIBUS-DP-Slave) is a device which can be programmed by the b maXX PLC in accordance with IEC 61131-3.

In the case of these devices, the contents of the GSD-file get changed as soon as

- other variables and/or correlation with the objects as fixed by the Default-Setting are used and
- other variables and/or correlation with the objects as were fixed in the last setting are used.

Change in the Default-Setting or that of the last setting for the variables and their correlation with the objects makes it mandatory to generate a new GSD-file (and integration of this GSD-file in the PROFIBUS-Master-configurator).

After having created the variables covered in the Section: “Creation of variables" in the PROPROG wt II project, the GSD-file can be generated now. The Example-GSD-file can be found in ▹Example GSD-File◃ from page 83 onward.

If you have not created any variables, then you can also press the Default-Button. In that case, the Default-Setting for the variables is used. The Default-GSD-file can be found in ▹Default GSD file ◃ on page 81.

NOTE

Clicking on the CheckBox „default configuration" deletes the variables eventually present in the PROFIBUS-PLC-Slave configurator!

NOTE

The Default GSD-file for the Option Module PROFIBUS-Slave for b maXX PLC can be found on the CD „PROFIBUS Version 1.0" (or higher), which also has the Library PROFIBUS_PLC01_20bd01" (or higher) on it.

For generating the GSD-file, click on the Button „Export GSD file" and specify a file name and the location of storage/saving for the GSD-file.

- Open the Profibus-Slave-configurator and create the desired variables (see ▹Creation of variables ◃ on page 38).
- Click on the Button „Export GSD file". The window „file save as..." gets opened.
- Search the location of saving in the window „file save as..." and specify the name of the GSD-file in the EditBox „filename".

The GSD-file is named as „MyGSDFile.gsd" in the Example.

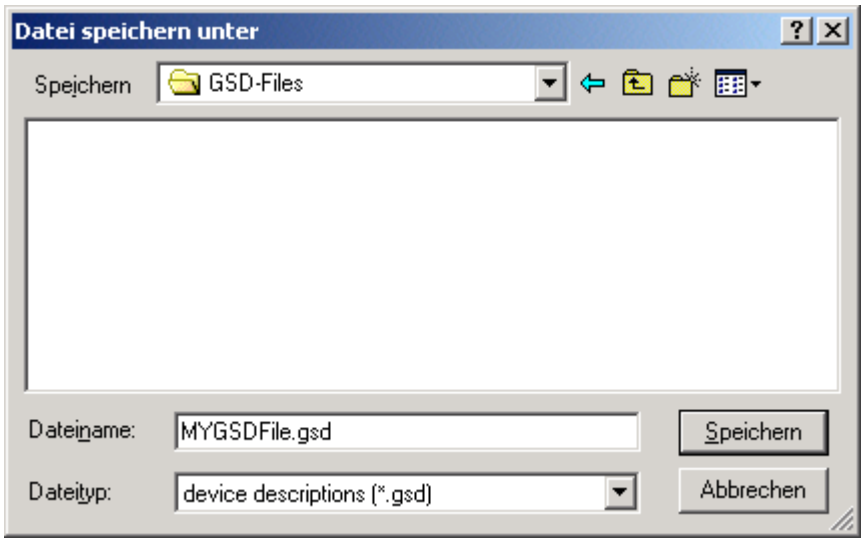


Figure 37: Exporting the GSD-file (2)

- Click on the Button „OK". The PROFIBUS-PLC-Slave configurator gets closed and the GSD-file is available for use in the PROFIBUS-Master configurators.

3.3.7 Programming of the Application in the PROPROG wt II Project

3.3.7.1 Initialisation of the Option Module PROFIBUS-DP-Slave

Procedure for initialisation of the Option Module PROFIBUS-DP-Slave

The initialisation of the Option Module PROFIBUS-DP-Slave is carried out by using the FB PROFIBUS_INIT_SL. The steps for using this function block are given below :

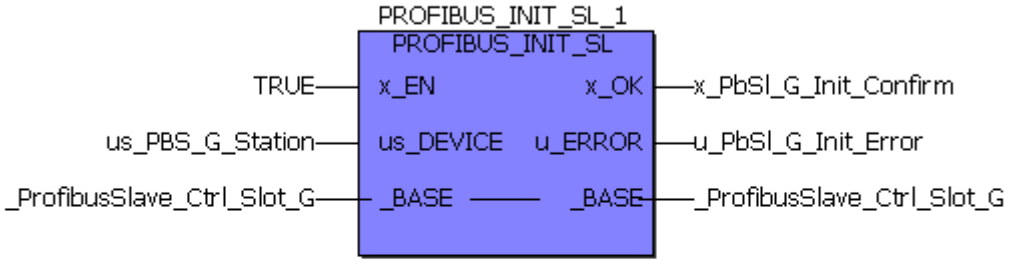
- Create a POE with SPS-Typ SH03-30 and processor type BM4_O_PLC01. This POE should be called later in a cold start- and hot start task.
- Place the FB PROFIBUS_INIT_SL in this POE.
- Connect the function block with variables of the correct data type.

At the input parameter us_DEVICE, connect the global variable „us_PBS_G_Station" from the datasheet „bmaXX_Variables" for the global variables. This variable was created by the PROFIBUS-PLC-Slave configurator, and so also the value of the variable (see ▹Setting the station address ◃ on page 35).

Example: Initialisation of the Option Module PROFIBUS-DP-Slave

The connection of the FB PROFIBUS_INIT_SL looks as follows:

```
(*Initialise the PROFIBUS-Slave in Slot G*)
```




```
(* - us_PBS_G_Station
     from global variables worksheet "bmaXX_Variables",

   - _ProfibusSlave_Ctrl_Slot_G
     from global variables worksheet "Global_Variables"*)
```

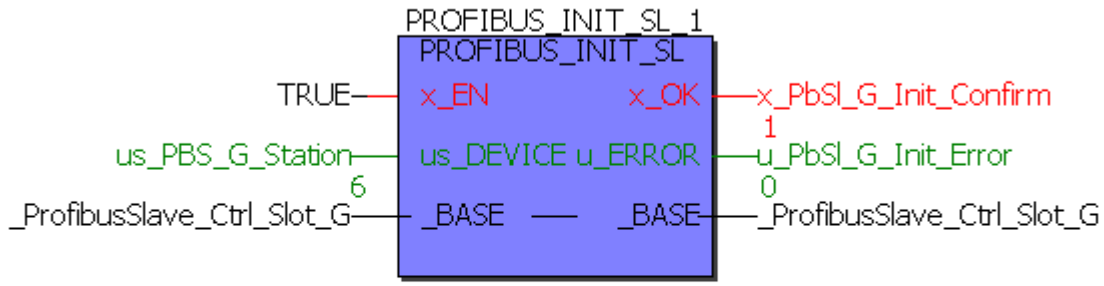
Figure 38: Initialisation of the PROFIBUS-DP-Slave with the FB PROFIBUS_INIT_SL

- Create a task for the cold start and the hot start of the PLC, if these should not be present in your project. Integrate the POE created for initialisation of the Option Module in both the tasks.
- Translate the project and load it as the boot project on the PLC. Switch off the device b maXX 4400 and switch it on again.

The FB PROFIBUS_INIT_SL reports a successful initialisation with x_OK = 1 and u_ERROR = 0.

Example:

```
(*Initialise the PROFIBUS-Slave in Slot G*)
```




```
(* - us_PBS_G_Station
     from global variables worksheet "bmaXX_Variables",

   - _ProfibusSlave_Ctrl_Slot_G
     from global variables worksheet "Global_Variables"*)
```

Figure 39: Initialisation of the PROFIBUS-DP-Slave with the FB PROFIBUS_INIT_SL - Online

Successful initialisation is also displayed by the LEDs on the Option Module. In this case, the Option Module PROFIBUS-DP-Slave is in parametrising-/configuring mode or already in the data exchange mode.

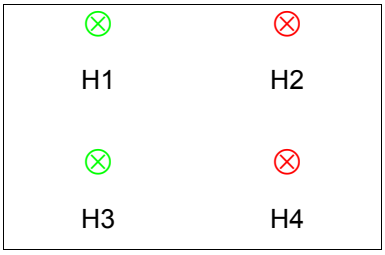


Figure 40: LEDs of the Plug-in Module BM4-O-PRO-01

H1 (green)	H2 (red)	H3 (green)	H4 (red)	Meaning
Off	On	Off	Off	Bus Error / Hardware Watchdog *)
On	Off	Off	Off	Data Exchange Mode
Off	Off	On	Off	Parametrising-/configuring mode
Off	On	Off	On	Configuration Error *)
On	On	On	On	Software Watchdog *)
Off	Blinks	Off	Blinks	(Synchronous Blinking) The blinking frequency of H2 (red) and H4 (red) indicates different types of errors *)

Blinking : about 200 ms on, then about 200 ms off

Further information in this regard can be found in the Operation Instructions for the Option Module PROFIBUS-DP-Slave for b maXX (5.03040), chapter „Finding and eliminating disturbances“.

See to it that all the PROFIBUS-devices have set or recognised the correct Baud rate. Refer to the related documentation for this.

3.3.7.2 Exchange of the service data

Definition

Communication of the service data corresponds to the client/server-communication model, i. e. the PROFIBUS-Master is the client which sends an order to a PROFIBUS-device for accepting or transmitting the data, in the form of the PPROFIBUS telegram. The PROFIBUS-device acts as the server, accepts the data and confirms this in the reply-telegram or sends the required data in the reply-telegram along with.

The Option Module PROFIBUS-DP-Slave is such a PROFIBUS-device and acts as result, as server of the data. 4 Byte data per order can be written or read by the Option Module PROFIBUS-DP-Slave.

Writing and reading the service data

Exchange of the service data on the PROPROG wt II side is restricted to wrting and reading of global variables created by you under ▹Creation of variables ◃ on page 38 by using the PROFIBUS-PLC-Slave configurator. The PROFIBUS-Master uses the 4th Byte of the module identification for addressing the global variables.

In this connection, please see ▹Figure34 on page 44 and ▹Figure35 on page 44 „Worksheet bmaXX_Variables for global variables".

NOTE

The variables provided for exchange of the service data, must not be used in the PROFIBUS-Master or as process data in these configuration tool (by selecting the module). Otherwise, the order for the service data of the masters is replied by an error message of the slaves.

Reading the service data

For reading the service data, the PROFIBUS-Master requests for the value of a variable in the form of a PROFIBUS telegram.

You have created the IEC 61131-3 variables in the chapter ▹Creation of variables◃ from page 38 onward. The configurator has deposited thereby some data in the GSD-file, for addressing of the variables. With the help of this data, the PROFIBUS-Master requests for the value of the variables („Parameter value" in the PKW-area in the PROFIBUS-Telegram). The Option Module PROFIBUS-DP-Slave converts this request in a read-access to the variable in the PROPROG wt II project and sends the reply to the master.

Writing the service data

For writing the service data, the PROFIBUS-Master sends the value of a variable in the form of a PROFIBUS telegram.

You have created the IEC 61131-3 variables in the chapter ▹Creation of variables◃ from page 38 onward. The configurator has deposited thereby some data in the GSD-file, for addressing of the variables. With the help of this data, the PROFIBUS-Master sends the value of the variables („Parameter value" in the PKW-area in the PROFIBUS-Telegram). The Option Module PROFIBUS-DP-Slave converts this write-request in a write-access to the variable in the PROPROG wt II project and sends an acknowledgement to the master.

In the example, the variable „ud_MyReadVar3" has been created in the PROFIBUS-PLC-Slave configurator. A PROFIBUS-Master can now have an access to the data from the GSD-file as it reads this variable and can thus read the contents of the variable „ud_MyReadVar3". Since the variable „ud_MyReadVar3" has been preassigned with a initial value of 3000, the value 3000 is bound to be displayed in the PROFIBUS-Master while reading the variables.

In this connection, also see chapter ▹Service data ◃ on page 60.

Furthermore, the variable „ud_MyWriteVar1" has been created in the PROFIBUS-PLC-Slave configurator. A PROFIBUS-Master can now have an access to this variable as it writes this variable and can thus write the contents of the variable „ud_MyWriteVar1".

In this connection, also see chapter ▹Service data ◃ on page 60.

3.3.7.3 Exchange of process data

Definition

Exchange of the process data on the PROPROG wt II side is restricted to reading and writing of global variables created by you under ▹Creation of variables ◃ on page 38 by using the PROFIBUS-PLC-Slave configurator.

Writing and reading the process data

For exchange of the process data, the PROFIBUS-Master sends the reference values in a PROFIBUS-telegram and requests for the actual values.

You have created the IEC 61131-3 variables in the chapter ▹Creation of variables◃ from page 38 onward. The configurator has deposited thereby some data in the GSD-file, for addressing of the variables. From this data, the master knows the locations of the reference or the actual values in the PROFIBUS telegram.

On the b maXX PLC side, the actual values are written in the Intel-Format to the Option Module PROFIBUS-DP-Slave. The Option Module PROFIBUS-DP-Slave writes the actual values in the Motorola-Format in the PROFIBUS-Telegram. This applies especially to the 32-Bit values. If required, conversion in the Intel-Format must take place again in the PROFIBUS-Master (e. g. where the PROFIBUS-Master is a PC). Further additional details on this can be obtained from the documentation of your PROFIBUS-Master.

The reference values of the PROFIBUS-Master, especially the 32-Bit reference values are expected in the Motorola-Format by the Option Module PROFIBUS-DP-Slave and are transferred to the b maXX PLC in the Intel-Format.

Consistency of the data

In case of consistency of the data, it must be differentiated between

- consistency of the data between PROFIBUS-Master and Option Module PROFIBUS-DP-Slave and
- consistency of the data between Option Module PROFIBUS-DP-Slave and b maXX PLC.

PROFIBUS-Master ↔ Option Module PROFIBUS-DP-Slave

The Option Module PROFIBUS-DP-Slave supports consistency among all the modules created, i. e. The data in the PROFIBUS-telegram is transmitted consistently.

Option Module PROFIBUS-DP-Slave ↔ b maXX PLC

The b maXX PLC supports consistency for 8-Bit- and 16-Bit-values.

If consistency is desired over a larger data area (e. g. 32-Bit-values or several values in any combination), then a Handshake-mechanism between the Option Module PROFIBUS-DP-Slave and b maXX PLC is necessary.

The following code in a POE (in Structured Text (ST)) is required for this purpose:

Worksheet of variables:

```
VAR
   u_tempAccessVar    :  UINT; (* temporary use *)
   u_tempMerker       :  UINT;
```

```
END_VAR
VAR_EXTERNAL
   _ProfibusSlave_Ctrl_Slot_G : PROFIBUS_PLC_SL_BMSTRUCT;
(* PROFIBUS-Slave BM4-O-PRO-01-00-01 in Slot G *)
END_VAR
```

Code-worksheet:

```
(* Run this code sequence only in one task ! *)

_ProfibusSlave_Ctrl_Slot_G.u_PLC_ACCESS  := UINT#1;  (* sperren
/ lock *)
u_tempAccessVar   := _ProfibusSlave_Ctrl_Slot_G.u_OMC_ACCESS;

IF  ((UINT_TO_WORD(u_tempAccessVar)  AND WORD#16#0001) =
WORD#16#0000) THEN
   (* wir dürfen zugreifen / we are allowed to access DPRAM *)
   IF (u_tempAccessVar <> u_tempMerker) THEN
   (* das Option Module hat data ausgetauscht / data exchange
done *)
      (* ----- Fügen Sie hier Ihre datazugriffe ein ----- *)
      (* ----- add your data access code here ----- *)
      ;
   END_IF;
END_IF;

u_tempMerker  := u_tempAccessVar;
_ProfibusSlave_Ctrl_Slot_G.u_PLC_ACCESS := UINT#0;  (* freigeben
/ unlock *)
```

You can copy this code in the Code-worksheet of your POE .

The data exchange (reading and writing of the variables) between the b maXX PLC and the Option Module PROFIBUS-DP-Slave then takes place for example as follows :

```
      (* ----- Fügen Sie hier Ihre datazugriffe ein ----- *)
      (* ----- add your data access code here ----- *)
      ud_MyWriteVar1 := ud_MyReadVar1  + UDINT#1;
```

NOTE

The code for a consistent exchange of data is used only in one POE and this POE must not be instantiated.
Otherwise, consistency of the data is not ensured/guaranteed.

3.3.7.4 Diagnostic message

Definition

In case of a diagnostic message, the Slave makes the diagnostic data available and informs the master in a process data reply telegram about the diagnostic data produced. The PROFIBUS-Master can read and evaluate the diagnostic data at any given point of time on receiving a request for diagnostics.

NOTE

The Master must request for the diagnostic data, so that the slave can send the diagnostic data.

Sending a diagnostic message

You can send a diagnostic message in the application program on the b maXX PLC from the Option Module PROFIBUS-DP-Slave to the master by means of the FBs SDIAG.

NOTE

The Option Module PROFIBUS-DP-Slave itself sends as much as a 7 Byte diagnostic data. The data which is connected to the FB SDIAG at the input-/output parameter a_ED_DATA, get connected to these 7 Byte. In the master, the data from a_ED_DATA[0] is thus displayed in the 8th Byte, that of a_ED_DATA[1] in the 9th Byte etc.

Please follow the steps given below for sending a diagnostic message, if you have not already integrated the FB SDIAG for sending the status data in your project:

- Create a POE with SPS-Type SH03-30 and processor type BM4_O_PLC01. This POE is to be called later in a cyclic task.
- Place the FB SDIAG in this POE.
- Connect the function block with variables of the correct data type.
- Create a cyclic task with medium to low priority, if your project should still not be containing such one. Integrate the POE created with the FB SDIAG in this task.
- Translate the project and load it as the (Boot-) project on the PLC.
- Start the project.

With x_REQ = 1, the diagnostic message is transferred to the Option Module PROFIBUS-DP-Slave. The diagnostic message is not sent by the slave to the master until a request for diagnostics is received from the master. The FB SDIAG reports about successful execution with x_OK = 1.

Example: Sending a diagnostic message

The diagnostic message 16#1234_5678_90AB_CDEF is to be transmitted.

This 8 Byte diagnostic data is entered in a_ED_DATA[0] to a_ED_DATA[7].

Since the diagnostic data must be lying in a_ED_DATA, x_ED_FLAG must be set at TRUE.

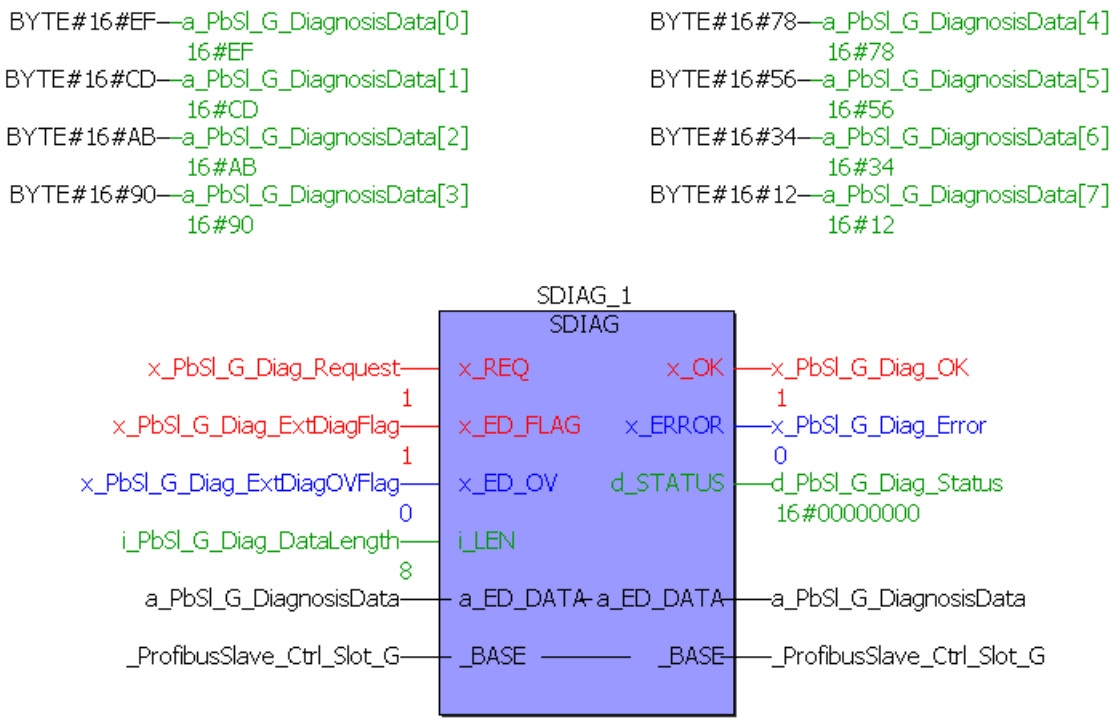
At i_LEN, INT#8 is specified for a data of 8 Byte -length.

x_REQ is now set at TRUE. This transfers the diagnostic data to the Option Module PROFIBUS-DP-Slave and this sends the data (after the master has requested for the diagnostic data).

Successful transmission of the data is displayed by x_OK = TRUE (only when the Master has requested for the diagnostic data!).

```
(*Send a diagnosis info via PROFIBUS-Slave in Slot G*)


        (*Diagnosis data: 16#1234_5648_90AB_CDEF*)
```




```
(* - _ProfibusSlave_Ctrl_Slot_G
     from global variables worksheet "Global_Variables"*)
```

Figure 41: Sending a diagnostic message of the Option Module PROFIBUS-DP-Slave

Display of the data from a_ED_DATA in a PROFIBUS-Master in the area „Extended Device Diagnostic“ may look something like this (noting that the data of a_ED_DATA[0] is displayed in the 8th Byte, that of a_ED_DATA[1] in the 9th Byte etc.):

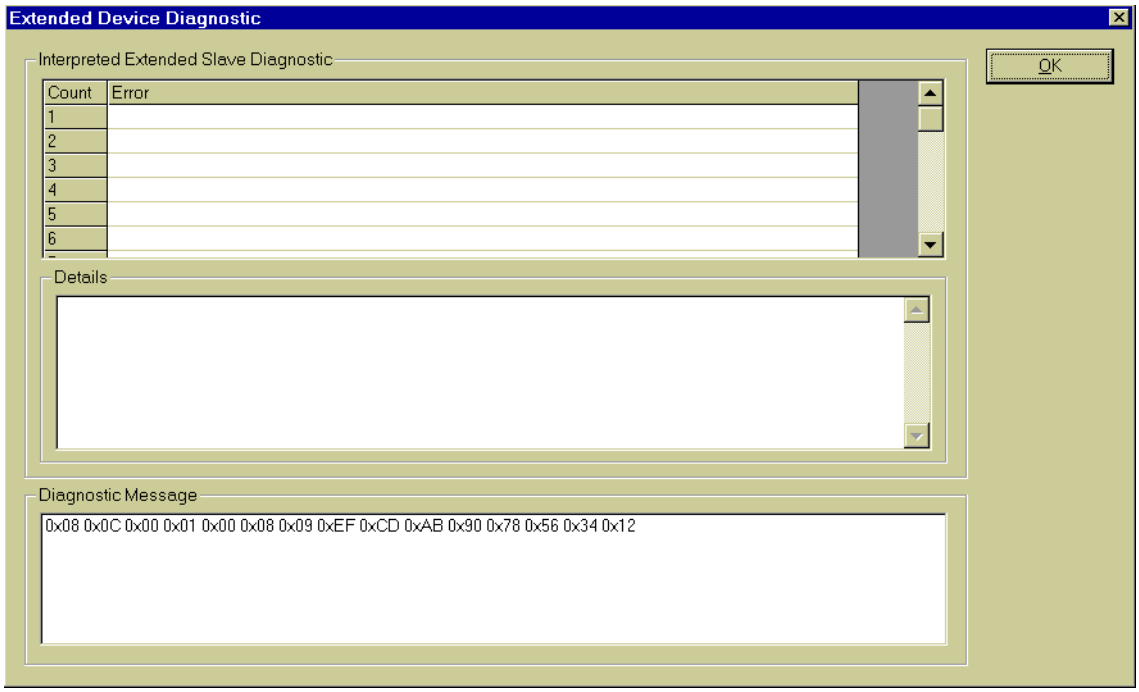


Figure 42: Diagnostic message in a PROFIBUS-DP-Master (SyCon - System Configurator, manufacturer Hilscher)

3.3.7.5 Status message

In case of a status message, the slave makes the status data available. The PROFIBUS-Master can read and evaluate the status data at any given point of time on receiving a request for status.

The status data is generated by the Option Module PROFIBUS-DP-Slave.

The user has no influence on the application program on the b maXX PLC to the status data or time of its transmission.

NOTE

The Master must request for the status data, so that the slave can send the status data.

Display of the status in a PROFIBUS-Master may look somewhat like this:

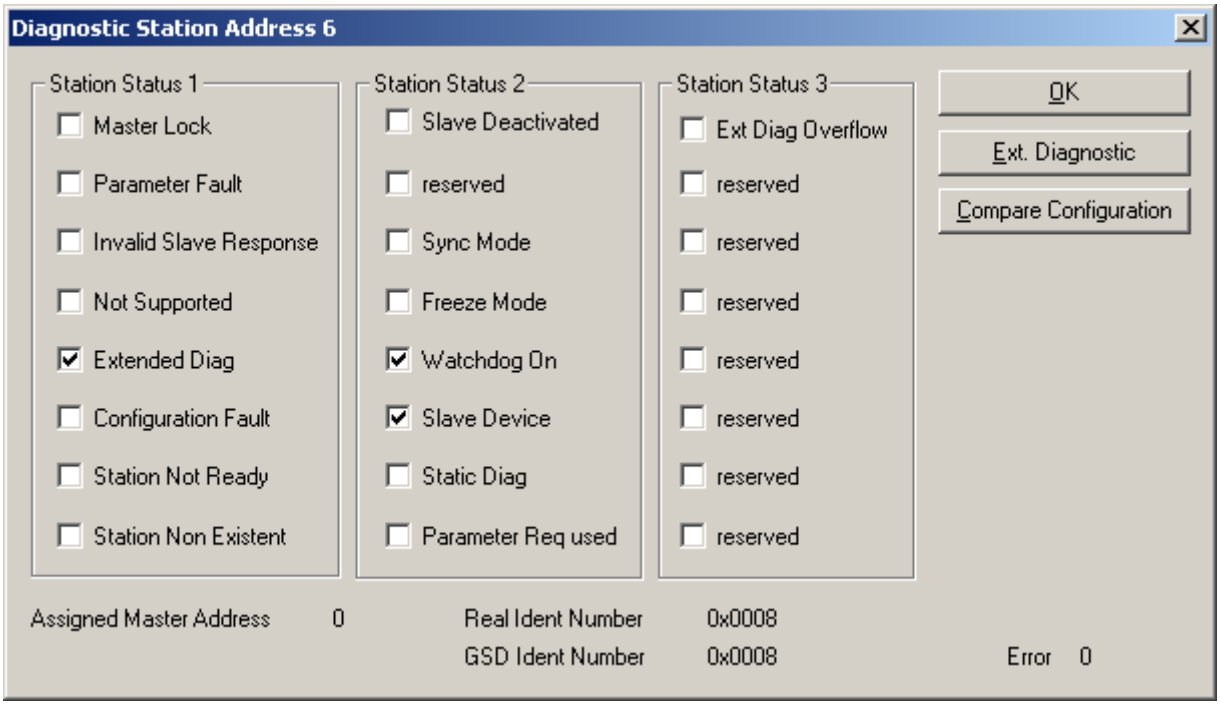


Figure 43: Status message in a PROFIBUS-DP-Master (SyCon - System Configurator, manufacturer Hilscher)

3.3.7.6 Reading out the current station address

The setting of the station address is effected by using:

- the PROFIBUS-Slave-configurator and the FB PROFIBUS_INIT_SL

or

- the rotary switch on the Option Module

or

- the PROFIBUS-Master

(see ▹Setting the station address◃ from page 35 onward).

The currently set station address can be read out in the DPRAM between the Option Module PROFIBUS-DP-Slave and the b maXX PLC.

(*.u_LOCAL_NODE_ID; * is the global variable _ProfibusSlave_Ctrl_Slot_G (to _ProfibusSlave_Ctrl_Slot_M), see ▹Creating a global variable for data exchange ◃ on page 29).

The Meaning of the values lying in *.u_LOCAL_NODE_ID can be obtained from the table ▹Setting the station address: ◃ on page 36.

Example: You want to read out the currently set station address (Address 6) of the Option Module PROFIBUS-DP-Slave on the b maXX PLC. For this, copy the station address of the DPRAM in the variable „u_PbSl_G_LocalNodeId":

```
u_PbSl_G_LocalNodeId:=_ProfibusSlave_Ctrl_Slot_G.u_LOCAL_NODE_ID
```

The Variable u_PbSl_G_LocalNodeId contains now the value 6 for the station address 6.

```
(*Read local node id of PROFIBUS-Slave in Slot G*)
```

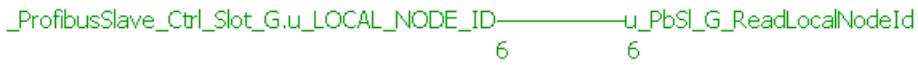


Figure 44: Reading the station address set

3.4 Structure of PROFIBUS telegrams

PROFIBUS uses 4 kinds of telegrams. These are

- token telegram
- data telegram without the data
- data telegram having a fixed length
- data telegram having a variable length

3.4.1 Token telegram

The Token telegram does not contain any useful data and is used only by the masters on the bus for passing on the token.

❑ Token telegram:

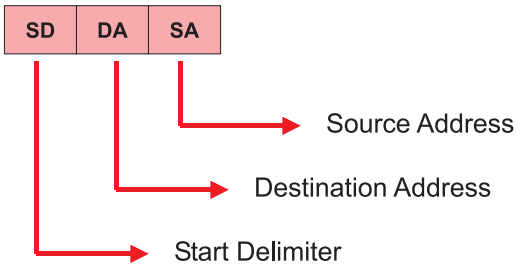


Figure 45: Token telegram

3.4.2 Data telegram without the data

The data telegram without the data is used by the master by way of a request for status while searching for new stations on the bus. Furthermore, a master uses this telegram when it is not sending any starting data to a slave.

❑ Data telegram without data:

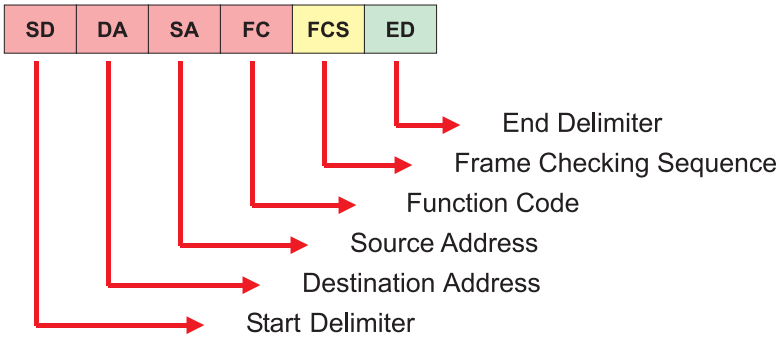


Figure 46: Data-Telegram without the data

3.4.3 Data telegram having a fixed length

The data telegram having a fixed length has 8 Byte data.

❑ Data telegram with fixed length:

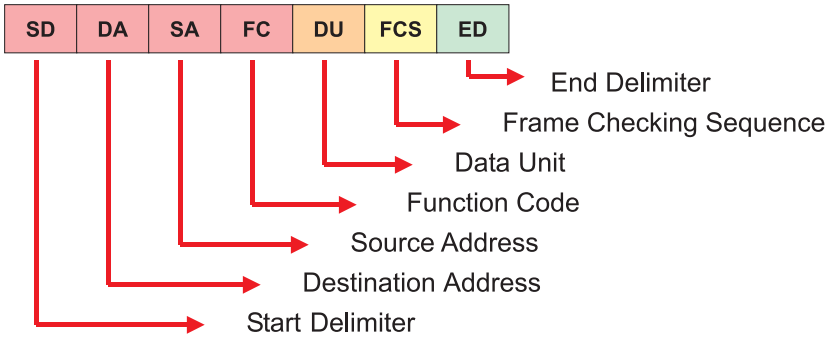


Figure 47: Data telegram having a fixed length

3.4.4 Data telegram having a variable length

The data telegram having a variable length is used for exchange of the process - and service data.

❑ Data telegram with variable length:

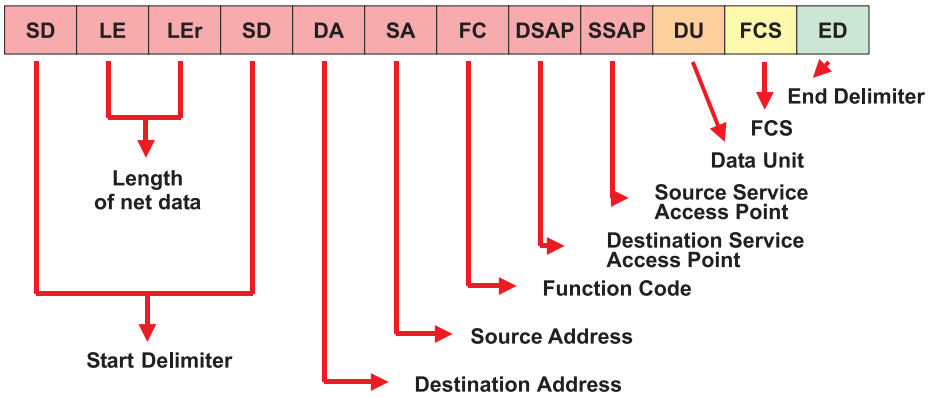


Figure 48: Data telegram having a variable length

In case of a data telegram having a variable length, the data (Data Unit) are divided in the PKW-area (for the service data; PKW = Parameter identifier value) and the PZD-area (for the process data). The length of the PKW-area is 8 Byte / 4 words, the length of the PZD-area is variable.

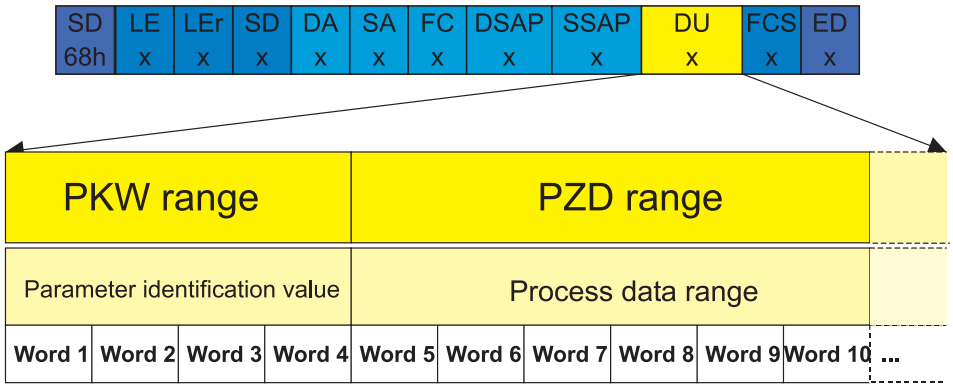


Figure 49: PKW- and PZD-area in the data unit (DU)

It is possible to use telegrams containing only the process - or the service data. A pure process data telegram contains in the data unit (DU) only the PZD-area, whereas a pure service data telegram contains in the data unit (DU) only the PKW-area.

3.4.4.1 Process data

Exchange of the process data between the master and the slave takes place only when the master has configured exchange of the process data at the start of the bus.

The process data area (PZD-area) consists a maximum of 244 Bytes. If there takes place simultaneously, exchange of the service data, then the process data area consists a maximum of 236 Bytes.

3.4.4.2 Service data

Exchange of the service data between the master and the slave takes place only when the master has configured exchange of the service data at the start of the bus.

The service data area (PKW-area) consists of a multipart identifier code for the service data (PKE, IND) and the value of the service data (PWE 1, PWE 2).

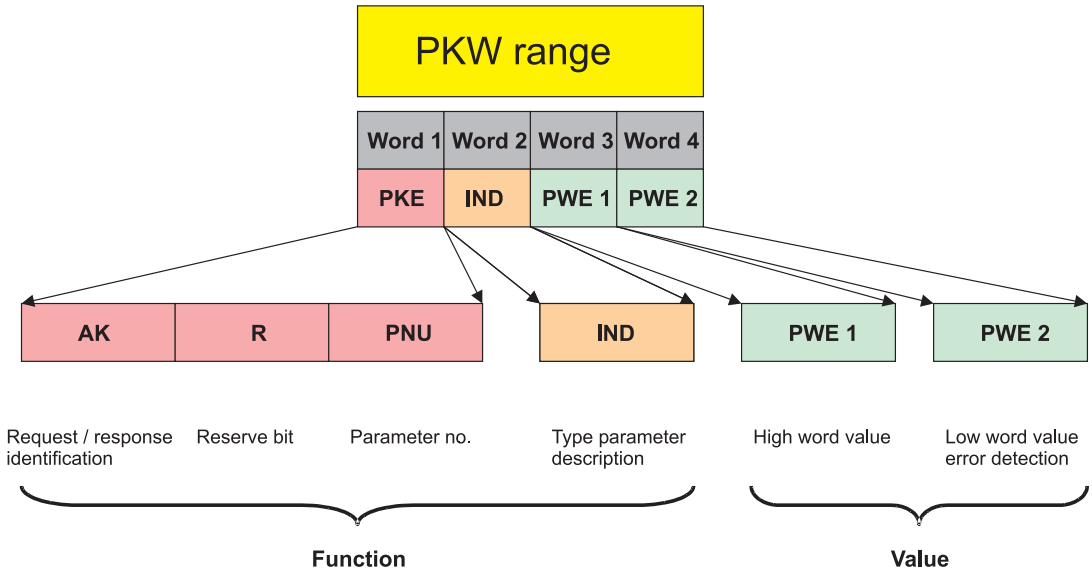


Figure 50: Division of the PKW-area

The first part of the identifier code for the service data, the word PKE, is divided in the job identification code (AK), a reserve bit (always 0) and the Parameter No. (0..2047).

In case of the Option Module PROFIBUS-Slave, create an assignment between the variable in the PROPROG wt II project and a parameter number for addressing these variables via the PROFIBUS-Master by using a PROFIBUS-Slave-configurator (in the b maXX 4400 configurator for devices)

NOTE

In case of the Option Module PROFIBUS-DP-Slave for b maXX PLC, the parameter numbers addressed by the PROFIBUS-Masters are not identical with the parameter numbers of the b maXX Controller.

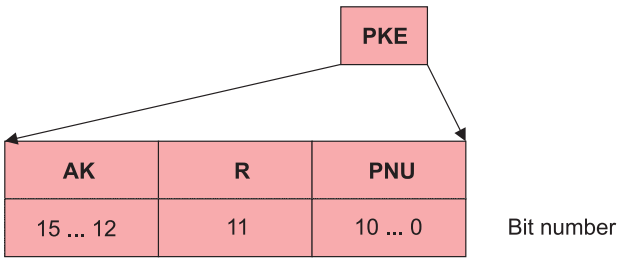


Figure 51: Division Word PKE from the PKW-area

Codes for functions job identification are listed in the table below.

PKE-AK	Function	Description
0_hex	No order	No order of service data
1_hex	PWE	Reading the service data
2_hex	Change the PWE (word)	Wrting the service data (in word format)
3_hex	Change the PWE (doubleword)	Wrting the service data (in doubleword format)
4_hex	Request for PBE-Element	Reading the parameter description not supported by the Option Module PROFI-BUS-Slave
5_hex	Reserved	-
6_hex	Reading the array	Reading the element of an array
7_hex	Change the array (word)	Writing the element of an array (in word format)
8_hex	Change the array (doubleword)	Writing the element of an array (in doubleword format)
9_hex..15_hex	Reserved	-

The second part of the identification for the service data, the word IND is always 0 in case of the Option Module PROFIBUS-DP-Slave, i. e. the value of the parameter is read or written. In case of arrays, the word IND contains the index of the array element (starting with 0).

The Option Module PROFIBUS-DP-Slave sends an identification code for reply in the reply-telegram, in the PKE-AK area.

Codes for functions job identification are listed in the table below.

PKE-AK	Function	Description
0_hex	No order	No order of service data
1_hex	Order executed (word)	Service data read or written (in word format)
2_hex	Order executed (in doubleword format)	Service data read or written (in doubleword format)
3_hex	Reserved	-
4_hex	Array-order executed (wortd)	Array-element read or written (in word format)
5_hex	Array-order executed (doubleword)	Array-element read or written (in doubleword format)
6_hex	Reserved	-

PKE-AK	Function	Description
7_hex	Error	An error has occured while processing the order for the service data. The error number is in PWE 2
8_hex..15_hex	Reserved	-

If an error occurs while processing the order for the service data, then 7 is entered in the reply identifier code (PKE-AK) and an error number is given as output in PWE 2. The error numbers are listed in the table below

Table: error numbers in PWE 2

PWE 2	Error	Description
0000_hex	-	Reserved
0001_hex	Read only	The variable is created as an INPUT-Module (actual value).
0002_hex	-	Reserved
0003_hex	Subindex	Array: the subindex does not exist
0004_hex	-	Reserved
0005_hex	Incorrect Type	The variable to be written (or read) has a format different than the one which has been specified via the identifier code (PKE-AK).
0006..000A_hex	-	Reserved
000B_hex	Address	The variable does not exist
000C..0015_hex	-	Reserved
0016_hex	Access Group	Unknown identifier code (PKE-AK)
0017..FFFF_hex	-	Reserved

NOTE

The variables which have been provided for exchange of the service data must not be used in the PROFIBUS-Master or in its configuration tool (by selection of the Module) as process data. Otherwise, the order for the service data of the Master is replied by an error message from the Slave.

Example 1:

You have created the variable „di_MyReadVar3" in the PROFIBUS-Slave-configurator. This variable received the module identifier code „42 C1 02 04". The 4th Byte is „04". Now, you want to read this Variable by using the PROFIBUS-Master.
Do the settings in the sequence indicated below for that:

IND: 0000 (hex)
PWE 1 0000 (hex)

PWE 2	0000 (hex)			
PKE-AK:	0001	(bin)	(1 hex)	Request for PWE
PKE-R:	0	(bin)		Bit-No. 11 (Reserved)
PKE-PNU:	000 0000 0100	(bin)	(4 hex)	4th Byte of the module identifier
PKE:	0001 0000 0000 0100	(bin)	(1004 hex)	

The PKW-area in the data section of the PROFIBUS telegram then looks as indicated below:

PKW				**PZD**	
PKE	IND	PWE 1	PWE 2	...	...
1004_h	0000_h	0000_h	0000_h	...	...

Reply of the Option Module PROFIBUS-DP-Slave:

Return message without any error:

PKW				**PZD**	
PKE	IND	PWE 1	PWE 2	...	...
2004_h	0000_h	xxxx	xxxx	...	...

In PWE 2 lies Lowword and in PWE 1 lies the Highword of the variable „di_MyReadVar3".

Return message with error (e. g. if the variable „di_MyReadVar3" is absent in the PROFIBUS-PLC-Slave configurator):

PKW				**PZD**	
PKE	IND	PWE 1	PWE 2	...	...
7004_h	0000_h	0000_h	xxxx	...	...

PKE-AK shows 7 (order not executable) and PWE 2 shows an error number (see ▹Table: error numbers in PWE 2◃ on page 62).

Example 2:

You have created the variable „ud_MyWriteVar1" in the PROFIBUS-Slave-configurator. This variable received the module identifier code „ud_MyWriteVar1". The 4th Byte is „02". Now, you want to write the value 1234 5678 (hex) on to this variable by using the PROFIBUS-Master.
Do the settings in the sequence indicated below for that:

IND:	0000 (hex)			
PWE 1	1234 (hex)			
PWE 2	5678 (hex)			
PKE-AK:	0011	(bin)	(3 hex)	Change PWE (Double word)
PKE-R:	0	(bin)		Bit-No. 11 (Reserved)
PKE-PNU:	000 0000 0010	(bin)	(02 hex)	4th Byte of the module identifier
PKE:	0011 0000 0000 0010	(bin)	(3002 hex)	

The PKW-area in the data section of the PROFIBUS telegram then looks as indicated below:

PKW				PZD	
PKE	IND	PWE 1	PWE 2	...	...
3002_h	0000_h	1234_h	5678_h	...	...

Reply of the Option Module PROFIBUS-DP-Slave:

Return message without any error:

PKW				PZD	
PKE	IND	PWE 1	PWE 2	...	...
2002_h	0000_h	1234_h	5678_h	...	...

The contents of the data are reflected and sent back.
In PWE 2 lies Lowword and in PWE 1 lies the Highword of the variable „ud_MyWriteVar1".

Return message with error (e. g. if the variable „ud_MyReadVar1" is absent in the PROFIBUS-PLC-Slave configurator):

PKW				PZD	
PKE	IND	PWE 1	PWE 2	...	...
7002_h	0000_h	0000_h	xxxx	...	...

PKE-AK shows 7 (order not executable) and PWE 2 shows an error number (see ▹Table: error numbers in PWE 2◃ on page 62).

3.4.4.3 Consistency of the Data

In case of consistency of the data, it must be differentiated between

- consistency of the data between PROFIBUS-Master and Option Module PROFIBUS-DP-Slave and
- consistency of the data between Option Module PROFIBUS-DP-Slave and b maXX PLC.

PROFIBUS-Master ↔ Option Module PROFIBUS-DP-Slave

The Option Module PROFIBUS-DP-Slave supports consistency among all the modules created, i. e. the data in the PROFIBUS-telegram is transmitted consistently.

Option Module PROFIBUS-DP-Slave ↔ b maXX PLC

The b maXX PLC supports consistency for 8-Bit- and 16-Bit-values.

If consistency is desired over a larger data area (e. g. 32-Bit-values or several values in any combination), then a Handshake between the Option Module PROFIBUS-DP-Slave and b maXX PLC is necessary. See ▹Consistency of the data◃ on page 51.

4

PROFIBUS-SLAVE FUNCTION BLOCKS

4.1 Function blocks for the PROFIBUS-Slave

The following function blocks are available for PROFIBUS-Slave:

Function block	Brief Description
PROFIBUS_INIT_SL	Initialises a PROFIBUS-Slave-Interface
SDIAG	Executes a Diagnosis-Send order

4.2 PROFIBUS_INIT_SL

Description

You may use this function block for PROFIBUS to initialize an option module PROFIBUS-DP-Slave for b maXX PLC (BM4-O-PRO-01-00-01) for data exchange.

NOTE

The FB PROFIBUS_INIT_SL uses the library BM_TYPES_20bd05 or above.

Input Parameters	Type of Data	Description
x_ENABLE	BOOL	Enable
us_DEVICE	USINT 1...127, 128	Node-ID for the PROFIBUS-Interface at option module PROFIBUS-DP-Slave
_BASE	PROFIBUS_PLC_SL_BMSTRUCT	PROFIBUS interface operating data

Output Parameters	Type of Data	Description
x_OK	BOOL	Confirms execution
u_ERROR	UINT	Error
_BASE	PROFIBUS_PLC_SL_BMSTRUCT	PROFIBUS interface operating data

The function block PROFIBUS_INIT_SL performs the initialization of the PROFIBUS interface at option module PROFIBUS-DP-Slave (BM4-O-PRO-01-00-01) for data exchange. The function block is requested in cold and warm start tasks and can be parameterized via input values. The node-ID for the option module is set at input **us_DEVICE**. The function block confirms the initialization of the PROFIBUS interface at output x_OK..

Input Parameter: **x_ENABLE**:

If input x_ENABLE is set to TRUE, the function block PROFIBUS_INIT_SL starts the initialization. After initialization is confirmed the function block sets the output x_OK to TRUE. If an error occurs the output u_ERROR shows them and x_OK is set to FALSE.

If input x_ENABLE is set to FALSE, the function block sets the output u_ERROR to 0 (Zero) and sets the output x_OK to FALSE.

Input Parameter: **us_DEVICE**:

There are various ways to set a node-ID to a Slave.

These are:

- Set the node-ID at the Slave
  - Set by rotary switch
    (Node-ID set by hardware rotary switch,
    changes will be valid after OFF/ON)
  - Set by function block PROFIBUS_INIT_SL, input us_DEVICE
    (Node-ID by software in b maXX PLC,
    changes will be valid after recall function block PROFIBUS_INIT_SL)
- Set the node-ID by message of PROFIBUS-Master
  - Set node-ID once only
    (The node-ID is set by the Master only once and is then stored in flash,
    not changeable, respectivly only after a special sequence)
  - Set node-ID more than once
    (The node-ID is set by the Master after every start of the PROFIBUS)

NOTE

According the PROFIBUS literature the node-ID is also called bus address, address, slave address, module address, station address, station number and module number.

The node-ID is set by the function block PROFIBUS_INIT_SL, input us_DEVICE.

The valid table for the node-ID is:

"us_DEVICE"	Rotary switch on the PROFIBUS-Slave		Meaning
	decimal	hexadecimal	
0	-	-	Reserved
1..125	-	-	Node-ID 1..125 by b maXX PLC, FB PROFIBUS_INIT_SL (The option module connects with node-ID 1..125 to the PROFIBUS)
126	-	-	Set node-ID only once (The option module responds with the node-ID, which was set by PROFIBUS-Master and which is stored in flash, to the PROFIBUS)
127	-	-	Set node-ID more than once (The option module responds with the node-ID, which was set by PROFIBUS-Master after every start of bus, to the PROFIBUS)
128 (Default)	0	00	Reserved
	1..125	01..7D	Node-ID 1..125 by rotary switch (The option module responds with node-ID 1..125 to the PROFIBUS)
	126..255	7E..FF	Set node-ID only once (The option module responds with the node-ID, which was set by PROFIBUS-Master and which is stored in flash, to the PROFIBUS)
129..255	-	-	Reserved

A node-ID was set already by configuration of the option module PROFIBUS-DP-Slave with the PROFIBUS-PLC-Slave configurator. The configurator generates the global variable worksheet "bmaXX_Variables". In this worksheet also the variable "us_DeviceNumber" is generated. Connect this variable to input us_DEVICE.

The default setting is us_DEVICE = 128, i. e. the node-ID will set by the rotary switches.

Input-/output Parameter:**_BASE**:

At _BASE, you must connect the global variable `_ProfibusSlave_Ctrl_Slot_G` (up-to `_ProfibusSlave_Ctrl_Slot_M`) according to the slot in which the option module is mounted.

If this global variable is not available in your project, you must create the global variable `_ProfibusSlave_Ctrl_Slot_G` (upto `_ProfibusSlave_Ctrl_Slot_M`) of data type `PROFIBUS_PLC_SL_BMSTRUCT` according to the slot in which the option module is mounted.

You must assign this variable via declaration of global variables to the base address of the PROFIBUS interface at option module PROFIBUS-DP-Slave.

The base address of the option module PROFIBUS-DP-Slave for b maXX PLC (BM4-O-PRO-01-00-01) according to the slot is:

Slot	Base address of the PROFIBUS interface at option module PROFIBUS-Slave
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Example:

Option module PROFIBUS-DP-Slave for b maXX PLC in Slot G of b maXX

```
_ProfibusSlave_Ctrl_Slot_G  AT  %MB3.2001792  :
                                    PROFIBUS_PLC_SL_BMSTRUCT;
```

where:

`_ProfibusSlave_Ctrl_Slot_G`	is the variable name with the data type short designation "_" for STRUCT
`PROFIBUS_PLC_SL_BMSTRUCT`	is the data type
`%MB3.2001792`	is the base address of the PROFIBUS interface at option module PROFIBUS-DP-Slave in slot G of b maXX

At each FB for the option module PROFIBUS-DP-Slave, part of the library PROFIBUS_PLC01_20bd02 (and higher), at **_BASE** the identical global variable must be linked. The variable is required for data exchange with the option module PROFIBUS-DP-Slave (BM4-O-PRO-01-00-01) and is of no further significance to the user.

Output Parameter: **x_OK**:

The output x_OK signalizes with TRUE that the function block PROFIBUS_INIT_SL initialized the PROFIBUS interface at option module PROFIBUS-DP-Slave.

The output x_OK is set to FALSE if input x_ENABLE is set to FALSE.

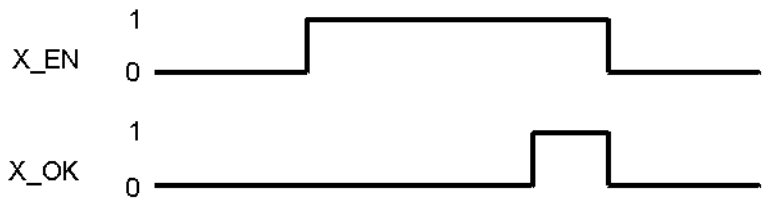


Figure 52: Timing diagram

Output Parameter: **u_ERROR**:

If an error occurs the FB PROFIBUS_INIT_SL signalizes this error at output u_ERROR, only if x_ENABLE = TRUE.

NOTE

The values are given as output only when x_ENABLE = TRUE .

u_ERROR	Meaning
0	Initialization ends with OK
1	PROFIBUS interface does not the handshake *)
2	Timeout at initialization (Time: 3 s) **)
3	Reserved
4	There is an invalid node-ID at input us_DEVICE (us_DEVICE <> 1..125, 126, 127, 128).
5 to 65535	Reserved

*) Check if the variable at _BASE is according to the slot, in which the option module PROFIBUS-DP-Slave is mounted, see "Input/Output _BASE".

**) Check if an option module PROFIBUS-DP-Slave (BM4-O-PRO-01-00-01) is mounted in the accorded slot (see Input/Output _BASE).

4.3 SDIAG

Description

You may use this function block for PROFIBUS to send a diagnostic message of a PROFIBUS-Slave.

NOTE

TheFB SDIAG uses the library BM_TYPES_20bd05 or above.

Input Parameters	Type of Data	Description
x_REQ	BOOL	Enable
x_ED_FLAG	BOOL	Extended diagnostic flag
x_ED_OV	BOOL	Extended diagnostic overflow flag
i_LEN	INT 3..60	Data length
a_ED_DATA	BYTE_60_BMARRAY	Diagnostic data
_BASE	PROFIBUS_PLC_SL_BMSTRUCT	PROFIBUS interface operating data

Output Parameters	Type of Data	Description
x_OK	BOOL	Confirms execution
x_ERROR	BOOL	Error
d_STATUS	DWORD	Status
a_ED_DATA	BYTE_60_BMARRAY	Diagnostic data
_BASE	PROFIBUS_PLC_SL_BMSTRUCT	PROFIBUS interface operating data

The function block SDIAG sends a diagnostic message via the option module PROFIBUS-DP-Slave to the PROFIBUS-Master. The slave informs the master (by a flag in the process data message) that diagnostic data is available. As a result the master requests the diagnosic data.

The user must serve the diagnostic data.

NOTE

The option module PROFIBUS-DP-Slave sends 7 Bytes Diagnosis (including header byte for a_ED_DATA) independent of the user. Diagnostic data, connected at a_ED_DATA, follows this 7 Bytes. In the master the Byte a_ED_DATA[0] will show in the 8th Byte, a_ED_DATA[1] will show in the 9th Byte and so on.

Input parameter **x_REQ**:

With x_REQ = TRUE the function block SDIAG transfers the diagnostic data to the option module. After sending the data the function block confirms with x_OK = TRUE, in case of an error the output x_ERROR is set to TRUE and an error number is set in d_STATUS.

With x_REQ = FALSE the function block SDIAG sets x_OK and x_ERROR to FALSE and d_STATUS to 16#0000_0000.

Input parameter **x_ED_FLAG**:

At x_ED_FLAG the extended diagnostic flag is to connect.
The x_ED_FLAG = TRUE means diagnostic data at a_ED_DATA is available.
The default setting is x_ED_FLAG = TRUE.

Input parameter **x_ED_OV**:

At x_ED_FLAG the extended diagnostic overflow flag is to connect.
The x_ED_OV means there where more diagnostic data available at the b maXX PLC than enlisted in a_ED_DATA. In this case the master knows about and can change the diagnostic request sequence for the slaves.

Input parameter **i_LEN**:

At i_LEN is set how many bytes are enlisted in a_ED_DATA. The minimum value is 3 (Byte), the maximum value is 60 (Byte). Other values result in an error at d_STATUS.

Input/output parameter **a_ED_DATA**:

A variable of the data type BYTE_60_BMARRAY is connected at a_ED_DATA. Data type BYTE_60_BMARRAY is a field of 60 entries of the data type BYTE:

```
BYTE_60_BMARRAY      :  ARRAY [0..59] OF BYTE;
```

Example:

```
a_DiagnosisData      :  BYTE_60_BMARRAY;
```

whereby:

`a_DiagnosisData`	is the variable name with the data type short designation "a" for ARRAY
`BYTE_60_BMARRAY`	is the data type

Access to the respective Bytes takes place in the pattern:
Variable name[Bytenumber]

Example: The value 16#EF should be entered in the first Byte (Byte 0) of the variable `a_DiagnosisData` and the value 16#CD should be entered in the second Byte (Byte 1) (in Structured Text (ST)):

```
a_DiagnosisData[0] := BYTE#16#EF;
a_DiagnosisData[1] := BYTE#16#CD;
```

Input/output parameter **_BASE**:

At _BASE, you must connect the global variable `_ProfibusSlave_Ctrl_Slot_G` (upto `_ProfibusSlave_Ctrl_Slot_M`) according to the slot in which the option module is mounted.

If this global variable isn't available in your project, you must create the global variable `_ProfibusSlave_Ctrl_Slot_G` (upto `_ProfibusSlave_Ctrl_Slot_M`) of data type `PROFIBUS_PLC_SL_BMSTRUCT` according to the slot in which the option module is mounted.

You must assign this variable via declaration of global variables to the base address of the PROFIBUS interface at option module PROFIBUS-DP-Slave.

The base address of the option module PROFIBUS-DP-Slave for b maXX PLC (BM4-O-PRO-01-00-01) according to the slot is:

Slot	Base address of the PROFIBUS interface at option module PROFIBUS-Slave
G	%MB3.2001792
H	%MB3.3001792
J	%MB3.4001792
K	%MB3.5001792
L	%MB3.6001792
M	%MB3.7001792

Example:

Option module PROFIBUS-DP-Slave for b maXX PLC in slot G of b maXX

```
_ProfibusSlave_Ctrl_Slot_G  AT  %MB3.2001792  :
                                    PROFIBUS_PLC_SL_BMSTRUCT;
```

where:

`_ProfibusSlave_Ctrl_Slot_G`	is the variable name with the data type short designation "_" for STRUCT
`PROFIBUS_PLC_SL_BMSTRUCT`	is the data type
`%MB3.2001792`	is the base address of the PROFIBUS interface at option module PROFIBUS-DP-Slave in slot G of b maXX

At each FB for the option module PROFIBUS-DP-Slave, part of the library PROFIBUS_PLC01_20bd01 (and higher), at _BASE the identical global variable must be linked. The variable is required for data exchange with the option module PROFIBUS-DP-Slave (BM4-O-PRO-01-00-01) and is of no further significance to the user.

Output parameter **x_OK**:

The output x_OK signalizes with TRUE that the option module PROFIBUS-DP-Slave sent the diagnostic data.

The output x_OK is set to FALSE if input x_REQ is set to FALSE.

Output parameter **x_ERROR**:

If an error occurs the output x_ERROR is set to TRUE and an error number is set at d_STATUS. The output x_ERROR is set to FALSE if input x_REQ is set to FALSE.

Output parameter **d_STATUS**:

The output d_STATUS cotaines four Bytes with different meaning.

d_STATUS			
Byte 3	Byte 2	Byte 1	Byte 0
Error_Code 1	Error_Code 1	Error_Decode	Function_Num

The byte Function_Num always is 16#00.

Error_Decode

Error_Decode	**Error_Code 1**	**Error_Code 2**	**Meaning**
16#00	16#00	16#00	No error
16#01..16#7F	-	-	Reserved
16#80	16#B6	16#00	Internal error (no access to send diagnostic data)
16#80	16#B7	16#00	Internal error (Parameter out of range)
16#80	16#C3	16#02	Internal error (the stack does not provide memory)
16#81	-	-	Reserved
16#82	16#01	16#01	Error at input x_ED_FLAG x_ED_FLAG = FALSE (set x_ED_FLAG to TRUE)
16#83	-	-	Reserved
16#84	16#01	16#01 16#02	Error at input i_LEN i_LEN < 3 (set i_LEN greater than 3) i_LEN > 60 (set i_LEN lower than 60)
16#85..16#FF	-	-	Reserved

Other values in Error Code 1 and Error Code 2 are reserved.

Example: Send diagnosis:

Transfer the diagnostic data 16#1234_5678_90AB_CDEF should be transferred to the master.

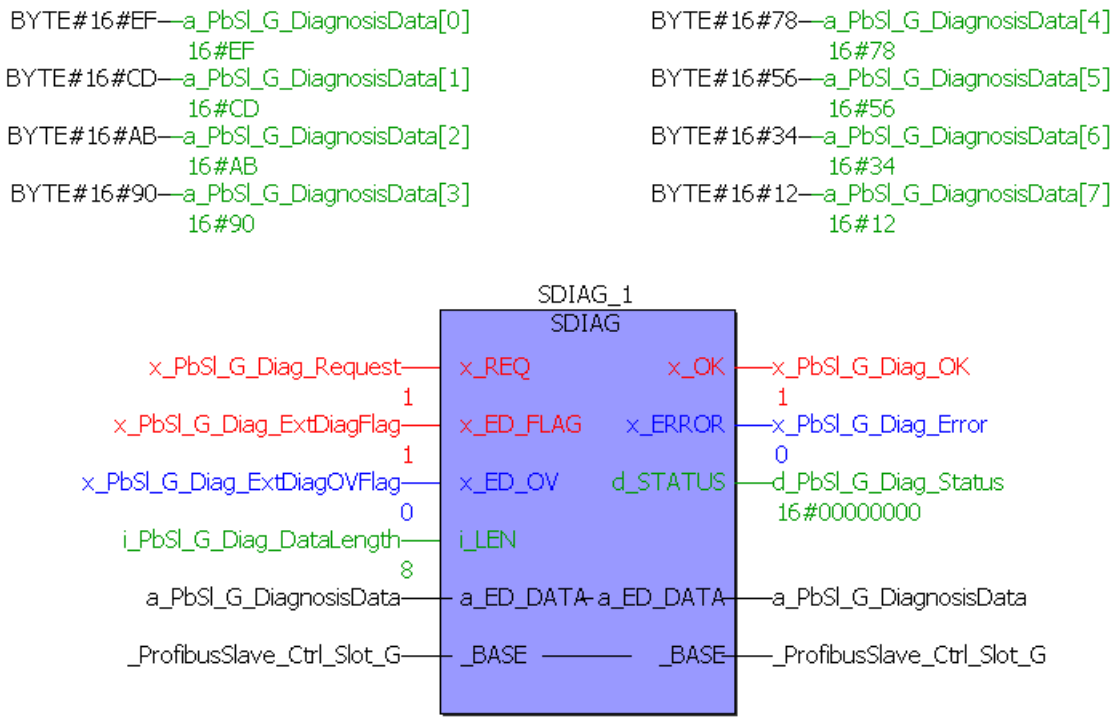
Enlist these 8 Byte diagnostic data in a_ED_DATA[0] to a_ED_DATA[7].
Set x_ED_FLAG to TRUE (x_ED_OV stays FALSE because there are no more data than enlistable in a_ED_DATA).
Set i_LEN to INT#8 for 8 Byte data length.

Now set x_REQ to TRUE. The diagnostic data is transferred to the option module. The option module informs the master that diagnostic data is available. As a result the master requests diagnostic data and the slave send it.

Once the data has been sent the function block confirms with x_OK = TRUE.

```
(*Send a diagnosis info via PROFIBUS-Slave in Slot G*)

(*Diagnosis data: 16#1234_5648_90AB_CDEF*)
```




```
(* - _ProfibusSlave_Ctrl_Slot_G
     from global variables worksheet "Global_Variables"*)
```

Figure 53: Sending the diagnosis message of the Option Module PROFIBUS-DP-Slave

In a PROFIBUS-Master the data could show like following (consider that the Byte a_ED_DATA[0] will show in the 8th Byte, a_ED_DATA[1] will show in the 9th Byte and so on):

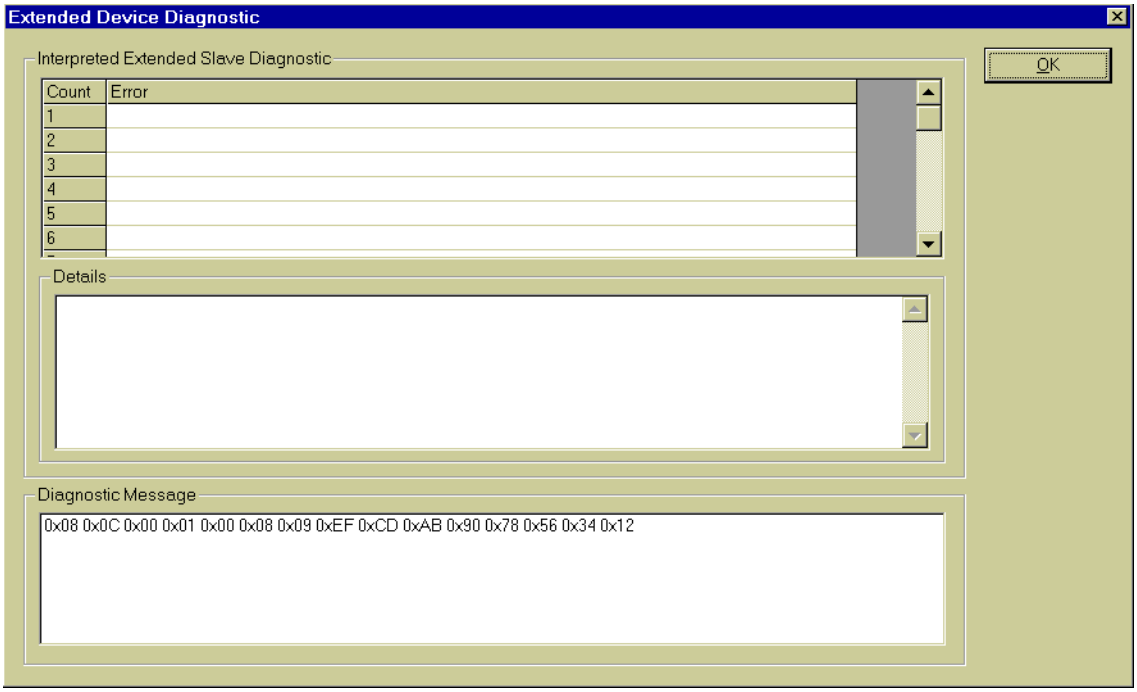


Figure 54: Diagnosis message in a PROFIBUS-DP-Master (SyCon - System Configurator made by Hilscher)

5

TABLES

5.1 Description of errors

The Option Module PROFIBUS-DP-Slave can recognise and display the errors on the PROFIBUS as also in the configuration. LEDs on the Option Module are used for this purpose

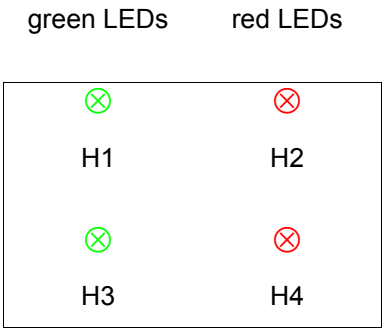


Figure 55: Arrangement of LEDs of the Option Module PROFIBUS-DP-Slave (BM4-O-PRO-01)

The Option Module PROFIBUS-DP-Slave for b maXX PLC signals the errors by the LED-combinations indicated below:

H1 (green)	H2 (red)	H3 (green)	H4 (red)	Meaning
Off	On	Off	Off	Bus Error / Hardware Watchdog *)
On	Off	Off	Off	*Data Exchange Mode*
Off	Off	On	Off	*Parametrisation- / Configuration mode*
Off	On	Off	On	Configuration error **)
On	On	On	On	Software Watchdog ***)
Off	Blinks	Off	Blinks	(Synchronous blinking) The blinking frequency of H2 (red) and H4 (red) represents different errors.****)

Blinking: about 200 ms on, then about 200 ms off

***) Bus Error**

The Option Module can monitor by Watchdog-time (set by the master via a PROFIBUS telegram), whether (or not) communication with the Master takes place within this time period. If after expiry of this period, there is no communication with the master, then this is displayed by the Option Module as "Bus Error / Watchdog". Furthermore, the reference and actual values of the Option Module are not updated.

The Option Module then causes automatic recognition of Baud rate to take place for restoring communication at the bus. As soon as the Option Module starts communicating again, the Option Module must be parameterised by the master. Thereafter, the Option Module goes in the "Data_Exchange" mode and the reference and actual values of the Option Module are updated again.

****) Configuration Error**

A configuration error is displayed if the Option Module PROFIBUS-DP-Slave has not been configured correctly by b maXX PLC. This, for instance, is the case, if the global worksheet of variables bmaXX_Variables" has not been generated by the PROFIBUS-PLC-Slave configurator or has not at all been called by the PROFIBUS-PLC-Slave configurator.

Configure the Option Module by executing the steps given under ▹Configuring the Option Module PROFIBUS-DP-Slave by using the PROFIBUS-PLC-Slave configurator◃ from page 35 onward.

*****) Software Watchdog**

The Software Watchdog is displayed if the Option Module PROFIBUS-DP-Slave is not functioning properly any more. In this case, the b maXX 4400 must be switched off and again switched on.

If this message occurs again, send the Option Module PROFIBUS-DP-Slave to Baumüller, describing the error(s) occuring thereby.

******) The blinking frequency of H2 and H4 has the following meaning**

Pattern	Meaning	Removal of Fault
1-time blinking, then pause	Hardware defective, or no PROFIBUS-Hardware on the Option Module	Send the Module to the manufacturer
2-times blinking, then pause	No PROFIBUS-Slave-Software on the Option Module	Send the Module to the manufacturer
3-times blinking, then pause	A new(er) PROFIBUS-Software required on the Option Module PROFIBUS-DP-Slave	Send the Module to the manufacturer
4-times blinking, then pause	Hardware-defect in the b maXX basic unit	Send the unit to the manufacturer

5.2 Default setting PROPROG wt II project variables

Contents of the global worksheet of variables „bmaXX_Variables":

```
(* Conf= CONF1 , Res= RES1 *)
(* Generated by system configurator tool. Do not modify ! *)
(* Automatisch generierter Code. Nicht manuell ändern ! *)

VAR_GLOBAL

(* -- BM4000 device -- )
<BM4Dev><Name>RES1</Name><ArtNr>369048</ArtNr><A>0</A><B>0</
B><C>0</C><D>0</D><E>0</E><F>362779</F><G>369656</G><H>365845</
H><J>0</J><K>0</K><L>0</L><M>0</M></BM4Dev>
( -- End BM4000 device -- *)

(* -- BM4000 device Slot G -- *)

(* -- Profibus -- *)
(* Slot G: Profibus Slave Module "OM_G" *)
   us_PBS_G_Station  AT %MB3.2001796 : USINT := 128; (* 1..125 *)
   us_PBS_G_Baudrate AT %MB3.2001798 : USINT := 0; (* 0=auto *)
   u_PBS_G_VarTableVersion AT %MW3.2001984 : UINT;
(* this BM4-PRO-01 uses Default Settings *)
   a_b_RefValue0_15 AT %MB3.2002560 : BYTE_16_BMARRAY;
   a_w_RefValue0_15 AT %MW3.2002576 : WORD_16_BMARRAY;
   a_d_RefValue0_7  AT %MD3.2002608 : DWORD_8_BMARRAY;
   a_d_RefValue8_15 AT %MD3.2002640 : DWORD_8_BMARRAY;
   a_b_ActValue0_15 AT %MB3.2002672 : BYTE_16_BMARRAY;
   a_w_ActValue0_15 AT %MW3.2002688 : WORD_16_BMARRAY;
   a_d_ActValue0_7  AT %MD3.2002720 : DWORD_8_BMARRAY;
   a_d_ActValue8_15 AT %MD3.2002752 : DWORD_8_BMARRAY;
(* -- End Profibus -- *)

(* -- BM4000 device Slot End -- *)

END_VAR
```

5.3 Example setting the PROPROG wt II project variables

Contents of the global worksheet of Variables „bmaXX_Variables":

```
(* Conf= CONF1 , Res= RES1 *)
(* Generated by system configurator tool. Do not modify ! *)
(* Automatisch generierter Code. Nicht manuell ändern ! *)

VAR_GLOBAL

(* -- BM4000 device -- )
<BM4Dev><Name>RES1</Name><ArtNr>350267</ArtNr><A>350324</A><B>0</B><C>0</
C><D>0</D><E>0</E><F>362779</F><G>369656</G><H>365845</H><J>0</J><K>0</K><L>0</
L><M>0</M></BM4Dev>
( -- End BM4000 device -- *)

(* -- BM4000 device Slot G -- *)

(* -- Profibus -- *)
(* Slot G: Profibus Slave Module "OM_G" *)
  us_PBS_G_Station  AT %MB3.2001796 : USINT := 6; (* 1..125 *)
  us_PBS_G_Baudrate AT %MB3.2001798 : USINT := 0; (* 0=auto *)
  u_PBS_G_VarTableVersion AT %MW3.2001984 : UINT := 100;
  w_PBS_G_Var1_ModuleID   AT %MW3.2001988 : WORD := 16#42c1;
  w_PBS_G_Var1_ModuleID2  AT %MW3.2001990 : WORD := 16#0200;
  u_PBS_G_Var1_Start      AT %MW3.2001992 : UINT := 2560;
  u_PBS_G_Var1_Number     AT %MW3.2001994 : UINT := 1;
  w_PBS_G_Var2_ModuleID   AT %MW3.2001996 : WORD := 16#42c1;
  w_PBS_G_Var2_ModuleID2  AT %MW3.2001998 : WORD := 16#0201;
  u_PBS_G_Var2_Start      AT %MW3.2002000 : UINT := 2564;
  u_PBS_G_Var2_Number     AT %MW3.2002002 : UINT := 1;
  w_PBS_G_Var3_ModuleID   AT %MW3.2002004 : WORD := 16#82c1;
  w_PBS_G_Var3_ModuleID2  AT %MW3.2002006 : WORD := 16#0202;
  u_PBS_G_Var3_Start      AT %MW3.2002008 : UINT := 2568;
  u_PBS_G_Var3_Number     AT %MW3.2002010 : UINT := 1;
  w_PBS_G_Var4_ModuleID   AT %MW3.2002012 : WORD := 16#82c0;
  w_PBS_G_Var4_ModuleID2  AT %MW3.2002014 : WORD := 16#0103;
  u_PBS_G_Var4_Start      AT %MW3.2002016 : UINT := 2572;
  u_PBS_G_Var4_Number     AT %MW3.2002018 : UINT := 1;
```

```
  w_PBS_G_Var5_ModuleID   AT %MW3.2002020 : WORD := 16#42c1;
  w_PBS_G_Var5_ModuleID2  AT %MW3.2002022 : WORD := 16#0204;
  u_PBS_G_Var5_Start      AT %MW3.2002024 : UINT := 2576;
  u_PBS_G_Var5_Number     AT %MW3.2002026 : UINT := 1;
  w_PBS_G_VarTableEnd     AT %MW3.2002028 : WORD := 16#0000;

  ud_MyReadVar1 AT %MD3.2002560 : UDINT := UDINT#1000; (* INPUT 42 C1 02 00 *)
  ud_MyReadVar2 AT %MD3.2002564 : UDINT := UDINT#2000; (* INPUT 42 C1 02 01 *)
  ud_MyWriteVar1 AT %MD3.2002568 : UDINT := UDINT#1500; (* OUTPUT 82 C1 02 02 *)
  u_MyWriteVar2 AT %MW3.2002572 : UINT; (* OUTPUT 82 C0 01 03 *)
  di_MyReadVar3 AT %MD3.2002576 : DINT := DINT#3000; (* INPUT 42 C1 02 04 *)
(* -- End Profibus -- *)

(* -- BM4000 device Slot End -- *)

END_VAR
```

5.4 Default GSD file

```
;=================================================================
;
; Baumüller Nürnberg Electronic GmbH & Co. KG
; Ostendstr. 80
; D - 90443 Nürnberg
; Tel:      ++49 (0)911 5432-0
; Fax:      ++49 (0)911 5432-417
; Internet: http://www.baumueller.de
;
;=================================================================
; GSD-File for bmaXX option module BM4-O-PRO-01-00-01 (PLC-Slave)
;=================================================================
;
; Name   : BM4OPRO1.GSD
; Status : V 1.04 (19.12.2003)
;
;=================================================================
;#Profibus_DP
; <Unit-Definition-List>
GSD_Revision=2
Vendor_Name="BAUMUELLER"
Model_Name="BM4-O-PRO-01 PLC"
Revision="REV 1.04"
Ident_Number=0x0008
Protocol_Ident=0
Station_Type=0
```

```
Hardware_Release="REV 1.01"
Software_Release="REV 1.01"
9.6_supp=1
19.2_supp=1
93.75_supp=1
187.5_supp=1
500_supp=1
1.5M_supp=1
3M_supp=1
6M_supp=1
12M_supp=1
MaxTsdr_9.6=60
MaxTsdr_19.2=60
MaxTsdr_93.75=60
MaxTsdr_187.5=60
MaxTsdr_500=100
MaxTsdr_1.5M=150
MaxTsdr_3M=250
MaxTsdr_6M=450
MaxTsdr_12M=800
Redundancy=0
Repeater_Ctrl_Sig=0
24V_Pins=0
Implementation_Type="DPC31"
Bitmap_SF="bmaXX_sf"
Bitmap_Device="bmaXX_ok"
Bitmap_Diag="bmaXX_er"
;
; Slave-Specification:
;
Freeze_Mode_supp=0
Sync_Mode_supp=0
Auto_Baud_supp=1
Set_Slave_Add_supp=1
Min_Slave_Intervall=6
Max_Diag_Data_Len=244
Slave_Family=10 ; 10=SPS 0=Allgemein
;
User_Prm_Data_Len=3
User_Prm_Data=0x00,0x00,0x00
;
Modular_Station=1
Max_Module=64
Max_Input_Len=112
Max_Output_Len=112
Max_Data_Len=224
;
; <Module-Definition-List>
Module="all modules" 0x2F,0x6F,0x6F,0x6F,0x1F,0x5F,0x5F,0x5F
EndModule
Module="service data PKW" 0x73
EndModule
Module="16 Byte b_RefValue0..15"  0x82,0x0F,0x00,0x00
EndModule
```

```
Module="16 Word w_RefValue0..16"  0x82,0x4F,0x01,0x01
EndModule
Module="8 DWord d_RefValue0..7"   0x82,0x4F,0x02,0x02
EndModule
Module="8 DWord d_RefValue8..15"  0x82,0x4F,0x02,0x03
EndModule
Module="16 Byte b_ActValues0..15" 0x42,0x0F,0x00,0x04
EndModule
Module="16 Word w_ActValues0..15" 0x42,0x4F,0x01,0x05
EndModule
Module="8 DWord d_ActValues0..7"  0x42,0x4F,0x02,0x06
EndModule
Module="8 DWord d_ActValues8..15" 0x42,0x4F,0x02,0x07
EndModule
```

5.5 Example GSD-File

```
; =============
; generated automatically
; created: February 16, 2004
; Project: ProfibusSlave_Example
; Cfg: CONF1
; Res: RES1
; Name: OM_G
;
; FileName: MyGSDFile.gsd
; =============
;
;=================================================================
;
; Baumüller Nürnberg Electronic GmbH & Co.
; Ostendstr. 80
; D - 90443 Nürnberg
; Tel:      ++49 (0)911 5432-0
; Fax:      ++49 (0)911 5432-417
; Internet: http://www.baumueller.de
;
;=================================================================
; GSD-Datei für bmaXX-Baugruppe BM4-O-PRO-01-00-01 (PLC-Slave)
;=================================================================
```

```
;
; Base  : BM4OPRO1.GSD
; Stand : V 1.04 (19.12.2003)
;
;===================================================================
;

#Profibus_DP
; <Unit-Definition-List>
GSD_Revision=2
Vendor_Name="BAUMUELLER"
Model_Name="BM4-O-PRO-01 PLC"
Revision="REV 1.04"
Ident_Number=0x0008
Protocol_Ident=0
Station_Type=0
Hardware_Release="REV 1.01"
Software_Release="REV 1.01"
9.6_supp=1
19.2_supp=1
93.75_supp=1
187.5_supp=1
500_supp=1
1.5M_supp=1
3M_supp=1
6M_supp=1
12M_supp=1
MaxTsdr_9.6=60
MaxTsdr_19.2=60
MaxTsdr_93.75=60
MaxTsdr_187.5=60
MaxTsdr_500=100
MaxTsdr_1.5M=150
MaxTsdr_3M=250
MaxTsdr_6M=450
MaxTsdr_12M=800
Redundancy=0
Repeater_Ctrl_Sig=0
```

```
24V_Pins=0
Implementation_Type="DPC31"
Bitmap_SF="bmaXX_sf"
Bitmap_Device="bmaXX_ok"
Bitmap_Diag="bmaXX_er"
;
; Slave-Specification:
;
Freeze_Mode_supp=0
Sync_Mode_supp=0
Auto_Baud_supp=1
Set_Slave_Add_supp=1
Min_Slave_Intervall=6
Max_Diag_Data_Len=244
Slave_Family=10 ; 10=SPS 0=Allgemein
;
User_Prm_Data_Len=3
User_Prm_Data=0x00,0x00,0x00
;
Modular_Station=1
Max_Module=64
Max_Input_Len=12
Max_Output_Len=6
Max_Data_Len=256
; <Module-Definition-List>
Module="Bedarfsdaten PKW" 0x73
EndModule
Module="ud_MyReadVar1" 0x42,0xC1,0x02,0x00
EndModule
Module="ud_MyReadVar2" 0x42,0xC1,0x02,0x01
EndModule
Module="ud_MyWriteVar1" 0x82,0xC1,0x02,0x02
EndModule
Module="u_MyWriteVar2" 0x82,0xC0,0x01,0x03
EndModule
Module="di_MyReadVar3" 0x42,0xC1,0x02,0x04
EndModule
```

A

APPENDIX A ABBREVIATIONS

AK **T**ask **I**dentification Code / **R**eply **I**dentification Code (PKE-AK)

BACI Baumüller Component Interface

bin binary

BUB Ballast-unit

BUC Baumüller Supply-/Feedback unit

BUG Baumüller Inverter Standard supply unit

BUM Baumüller Individual power unit

BUS Baumüller Power-Module

CE Communité Europeenne

CEN Comité Européen de Normalisation

CPU Central Processing Unit

CSRD **C**yclic **s**end and **r**equest **d**ata (Service provided with PROFIBUS-FMS)

DA **D**estination **a**ddress (Part of the PROFIBUS-Telgram-Header)

dez decimal

DIN **D**eutsches **I**nstitut für **N**ormung e.V., www.din.de

DINT Double integer (32 Bit, signed)

DP-RAM Dual-Port RAM

DSAP **D**estination **s**ervice **a**ccess **p**oint (part of the PROFIBUS-Telgram-Header)

DU **D**ata **u**nit (Net data block / Useful data area)
(part of the PROFIBUS-Telgram)

DWORD Double Word (32 Bit)

ED End delimiter
(part of the PROFIBUS-Telgram-Header)

EMC Electromagnetic compatibility

EN European Standard

ESD electrostatic sensitive device (components endangered by static electricity, EGB)

FB **F**unction **B**lock (PROPROG wt II, IEC 61131-3)

FC **F**unction **c**ode (Part of the PROFIBUS-Telgram-Header)

FCS **F**rame **c**hecking **s**equence (Check sum of DA, SA, FC, DSAP, SSAP, DU)
(part of the PROFIBUS-Telgram-Header)

GSD-file Geräte-Stamm-Datei
(contains the technical characteristics of a PROFIBUS-device)

hex hexadecimal

I/O Input/Output

I/O-module
Input-/Output-module

IEC **I**nternational **E**lectrotechnical **C**ommission, www.iec.ch

IND Index

INT Integer (16 Bit, signed)

ISO International Standard Organisation

LE Length - Length of the data (DA, SA, FC, DSAP, SSAP, DU)
(part of the PROFIBUS-Telgram-Header)

LED Light emitting diode

LEr **Le**ngth **r**epeat - Length of the net data (repitition)
(part of the PROFIBUS-Telgram-Header)

OSI **O**pen **S**ystem **I**nterconnect

PBE Parameter Description (**P**arameter **Be**schreibung)

PKE Parameter Identification Code (**P**arameter**ke**nnung)

PKW area
Parameter Identification Code / Range of parameter values (**P**arameter **K**ennung-/Parameter **W**ert-Bereich; Service data area in the PROFIBUS-telegram)

PLC Process loop control (SPS - Speicher programmierbare Steuerung)

PNO PROFIBUS user organisation e.V.

PNU **P**arameter **Nu**mber

PROFIBUS
Process **Fi**eld**bus**, DIN 19245

PROFIBUS-DP
PROFIBUS - **D**ecentralised **P**eriphery, this is the Bus from the Profibus-Master to the Profibus-Slaves

PROFIBUS-DPV0/V1
PROFIBUS-DP Version 0 / Version 1

PROFIBUS-FMS
PROFIBUS-**F**ieldbus **m**essage **s**pecification

PROFIBUS-PA
PROFIBUS-**P**rocess **a**utomation

PROFIdrive
PROFIBUS profile for drives with changeable speed

PROPROG wt II
Tool for programing the b maXX PLC

PWE Parameter Value (**P**arameter **We**rt)

PZD area
Processing data area (**Pro**ze**ssd**atenbereich; Processing data area in the PROFIBUS-telegram)

R **R**eserved

RAM Random Access Memory

RS485 Standardised serial Interface (IEC 61158)

SA **S**ource **a**ddress (part of the PROFIBUS-Telgram-Header)

SD **S**tart-**D**elimiter (part of the PROFIBUS-Telgram-Header)

SDA **S**end **d**ata with **a**cknowledge (service available with PROFIBUS-FMS)

SDN **S**end **d**ata with **n**o acknowledge (service available with PROFIBUS-FMS and PROFIBUS-DP)

SINT Short integer (8 Bit, signed)

SRD **S**end and **r**equest **d**ata with acknowledge (service available with PROFIBUS-FMS and PROFIBUS-DP)

SSAP **S**ource **s**ervice **a**ccess **p**oint (part of the PROFIBUS-Telgram-Header)

SyCon **Sy**stem **Con**figurator (Tool, made by Hilscher for parametrisation and configuring a PROFIBUS-Master as well as PROFIBUSes)

UDINT Unsigned double integer (32 Bit, unsigned)

UINT Unsigned integer (16 Bit, unsigned)

USINT Unsigned short integer (8 Bit, unsigned)

WinBASS II
Tool for parametrisation of the b maXX controller

Index

be in motion

Baumüller Nürnberg Electronic GmbH & Co. KG Ostendstraße 80-90 90482 Nuremberg Tel: +49(0)911-5432-0 Fax: +49(0)911-5432-130 **www.baumueller.de**