BAUMÜLLER

Option board PROFIBUS-DP Slave for Ωmega Drive-Line II

Technical Description
and Operating Instructions

Edition: September 2001

E	5.01013.03

BAUMÜLLER

PROFIBUS-DP-SLAVE OPTION BOARD FOR ΩMEGA DRIVE-LINE II

Technical Description and Operating Instructions

Edition: September, 1st 2001

Document no. 5.01013.03

This operation manual is intended as a complement to the technical description and the operation manual of the apparatus.

BEFORE CARRYING OUT COMMISSIONING, CAREFULLY READ AND OBSERVE THE OPERATING INSTRUCTIONS AND SAFETY INFORMATION

This document contains all the information necessary to correctly use the products it describes. It is intended for specially trained, technically qualified personnel who are well-versed in all warnings and commissioning activities.

The equipment is manufactured using state-of-the-art technology and is safe in operation. It can safely be installed and commissioned and functions without problems if the safety information is followed.

You may not carry out commissioning until it has been established that the machine into which this component is to be installed complies with the specifications of the EC machine guidelines.

This technical description/these operating instructions invalidate all previous descriptions of the corresponding product. Within the scope of further development of our products, Baumüller GmbH reserves the right to change their technical data and handling.

Manufacturer and Supplier's Address: Baumüller Nürnberg GmbH
Ostendstr. 80
D-90482 Nürnberg
Tel. ++49 (0)9 11/54 32 - 0
Fax ++49 (0)9 11/54 32 - 1 30

Country of Origin: Germany

Date of Manufacture: Determined from the serial number on the equipment

TABLE OF CONTENTS

Table of Contents

1 SAFETY INFORMATION

General Information

These operating instructions contain all the information necessary for correct operation of the products described. The document is intended for specially trained, technically qualified personnel who are well-versed in all warnings and commissioning activities.

The equipment is manufactured using state-of-the-art technology and is safe in operation. It can safely be installed and commissioned and functions without problems if the safety information in these operating instructions is followed.

Danger Information

On the one hand, the information below is for your own personal safety and on the other to prevent damage to the described products or to other connected equipment.

In the context of the operating instructions and the information on the products themselves, the terms used have the following meanings:

DANGER

This means that **death, severe personal injury**, or **damage to property will** occur unless appropriate safety measures are taken.

WARNING

This means that **death, severe personal injury**, or damage to property **may** occur unless appropriate safety measures are taken.

NOTE

This draws your attention to **important information** about the product, handling of the product or to a particular section of the documentation.

Qualified Personnel

In the context of the safety-specific information in this document or on the products themselves, qualified personnel are considered to be persons who are familiar with setting up, assembling, commissioning and operating the product and who have qualifications appropriate to their activities:

- Trained or instructed or authorized to commission, ground and mark circuits and equipment in acordance with recognized safety standards.
- Trained or instructed in accordance with recognized safety standards in the care and use of appropriate safety equipment.

Appropriate Use

WARNING

You may only use the equipment/system for the purposes specified in the operating instructions and in conjunction with the third-party equipment and components recommended or authorized by BAUMÜLLER NÜRNBERG GmbH.

For safety reasons, you must not change or add components on/to the equipment/system.

The machine minder must report immediately any changes that occur which adversely affect the safety of the equipment/system.

2 TECHNICAL DATA

2.1 General

The option board PROFIBUS-DP-Slave for the Ωmega Drive-Line II permits Ωmega Drive-Line II linkup to the popular field bus standard PROFIBUS-DP.

PROFIBUS-DP is a very powerful serial field bus system facilitating the networking of various sensors and actuators, offering transmission rates of 9.6 kBit/s to 12 MBit/s as well as extensive diagnostics options and error detection mechanisms. PROFIBUS-DP features are defined in European Standard EN 50170.

The option board PROFIBUS-DP-Slave allows complete drive control via this field bus standard and the transmission of status and control word as well as up to 8 setpoints and actual values. Service data communication is also possible for diagnostic or parameterisation tasks.

NOTE

Under the asynchronous PROFIBUS-DP bus system, exact definition of value transmission at certain times is possible to a limited extent only, although a time frame may be specified within which the setpoints and actual values are refreshed via the bus system. This time frame and the maximum possible bus speed are dependent on bus devices and control, so that it is therefore necessary to check suitability of the selected system before using bus systems to control specific system functions.

2.2 Technical data of the option board PROFIBUS-DP-Slave

2.2.1 Scope of supplies

- Option board PROFIBUS-DP-Slave for the Ωmega Drive-Line II, permanently installed therein
- This technical description and operating instructions
- Diskette with:
  - Device description file (GSD file) BNF_00D0.GSD (V1.41 of 22.11.1999)
  - Bitmaps BNF_NORN.BMP and BNF_ERRN.BMP

2.2.2 Functional scope

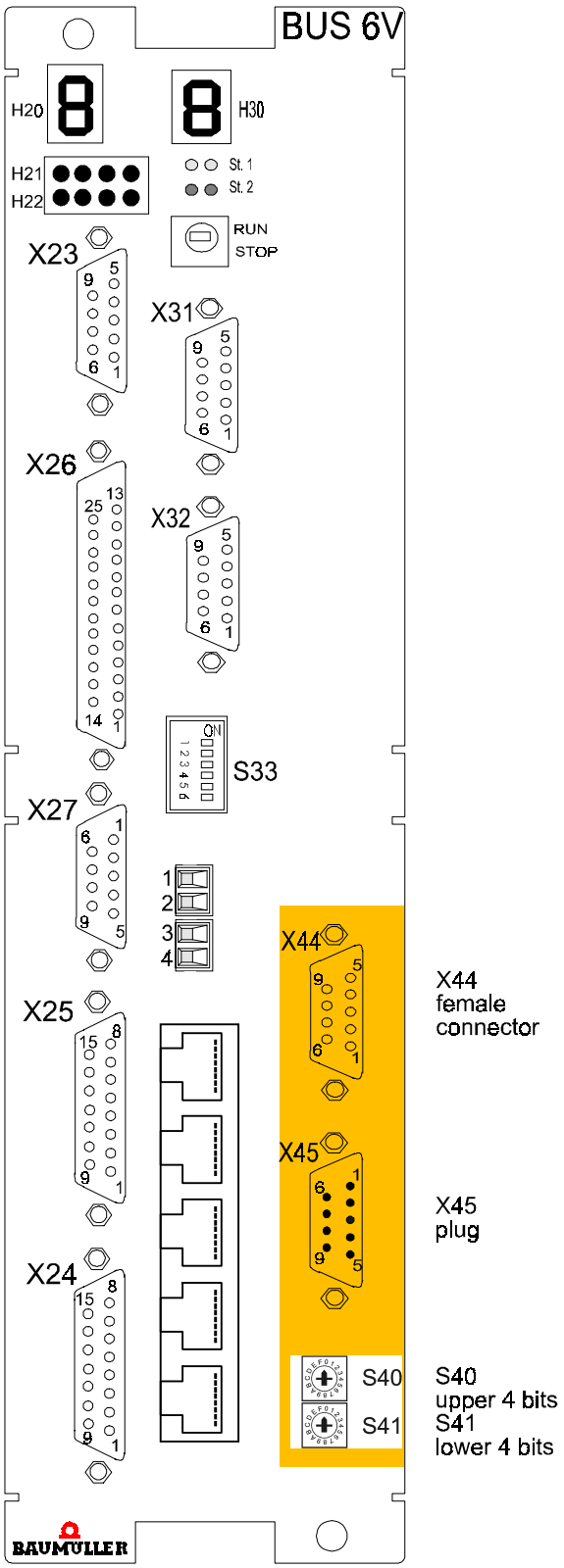
- PROFIBUS-DP slave interface
- Transmission of max. 8 cyclical setpoints (32-bit format)
- Transmission of max. 8 cyclical actual values (32-bit format)
- Complete access (read and write) to all controller parameters via service data communication (32-bit format)
- Configuration setting via the Ωmega Drive-Line II in the PROPROG wt II project
- Autocheck of setpoint / actual value configuration on each power-on
- Automatic Baud rate detection up to 12 MBaud
- Automatic detection of bus errors with parameterisable drive error response
- Automatic detection of a communication failure on the PROFIBUS-DP with programmable drive error response

2.3 Abbreviations

The following abbreviations are used in several locations for better transparency of this description:

AO	Address data outputs (for configuration)
AI	Address data inputs (for configuration)
BAPS interface	Internal interface between V controller and Ωmega Drive-Line II
FB	Function block
GSD file	Geräte-Stamm-Datei [device description file], electronic documentation of PROFIBUS-DP device characteristics for master configuration
IND	Index, defines which information of a parameter is to be accessed via service data communication.
IPM	Input process map
Net data block	Useful data in the PROFIBUS-DP telegram, i. e. without the extensions by device address, checksum etc. required for transmission.
Nibble	The 4 higher-order and / or 4 lower-order bits of a byte.
OPM	Output process map
PG	Siemens programmer
PKE	Parameter code, parameter number and command code for service data communication
PKW	Parameter code value, data in the PROFIBUS-DP telegram for service data transmission
PLC	Programmable logic controller
POU	Program organisation unit
PPO	Definitions for 16-bit data transmission via PROFIBUS-DP, not supported by this board
PWE	Parameter value, value associated with the parameter during service data communication
PZD	Process data area, data in the PROFIBUS-DP telegram for process data transmission (cyclical transmission)

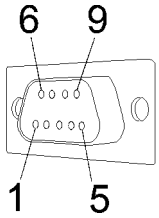
2.4 Sample configuration



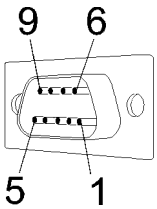
3 INSTALLATION

3.1 Pin assignment

Female connector X44



Plug X45



Pin no.	Assignment
1	Not assigned
2	Not assigned
3	Data line B
4	Request To Send RTS
5	Data ground
6	5 V_{DC}
7	Not assigned
8	Data line A
9	Not assigned

The option board PROFIBUS-DP-Slave offers a 9pole SUB-D plug and a 9pole SUB-D female connector for PROFIBUS-DP connection. Pin assignment is as per PROFIBUS standard EN 50170, so that all required signals are connected correctly when standard plugs are used.

Pre-assembled PROFIBUS-DP plugs with integrated terminating resistors and, if necessary, integrated PG interface are recommended for cabling. Connection is then exclusively at the 9pole SUB-D female connector X44. Use only cables for PROFIBUS-DP networking that have been approved explicitly for this application.

The terminating resistor must be activated at both the first and last device on the PROFIBUS-DP.

To facilitate bus analysis via a bus analyser, allow at least one bus plug with additional PG linkup socket at the PROFIBUS-DP.

3.2 Rotary switch

S40

S41

Setting PROFIBUS-DP device addresses 0…255

S40: Upper 4 bits

S41: Lower 4 bits

The rotary switches S40 and S41 assist setting the PROFIBUS-DP device address. This address must be set prior to Ωmega Drive-Line II power-on as it is accepted only during initialisation.

3.3 Commissioning

Commissioning the Ωmega Drive-Line II with option board PROFIBUS-DP-Slave is divided into commissioning the V controller, setting the interface parameters at the controller and PLC ends, and programming the Ωmega Drive-Line II.

3.3.1 Commissioning the V controller

Set and optimise the V controller in the required operating modes as a first commissioning step. Commissioning completed, make sure to back the set data up on the PC to avoid their loss or corruption due to wrong PROFIBUS-DP operations.

3.3.2 Commissioning the interface

The following settings must be performed for option board PROFIBUS-DP-Slave operation:

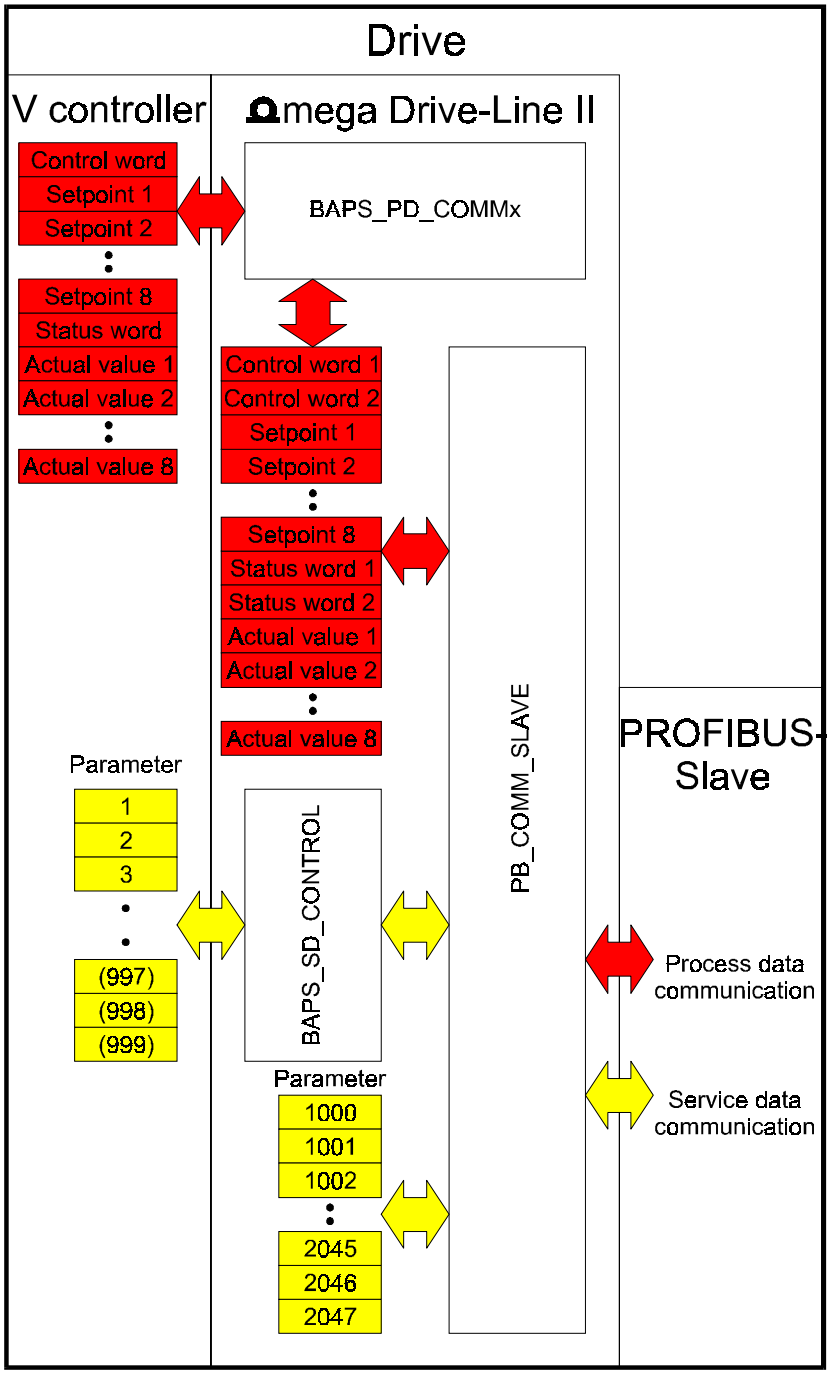
- Setting the PROFIBUS-DP device address via rotary switch S40 / S41 (refer “Rotary switch" on page 13)
- Setting the required transmission data volume (number of setpoints / actual values) in the master (GSD file)
- Setting communication source and monitoring in the controller with WinBASS
- Setting the setpoints and actual values to be transmitted in the PROPROG wt II project (PB_COMM_SLAVE)
- Saving the data record in the controller as boot data record
- Restarting complete system (controller, Ωmega Drive-Line II and PROFIBUS-DP master)

The option board PROFIBUS-DP-Slave will now be initialised on system restart as a consequence of the settings made in the PROPROG wt II project.

4 USE WITHIN THE PROPROG WT II PROJECT

4.1 General

The function block PB_COMM_SLAVE from the library PROFIBUS_DLII_20bd00 (or higher) is available for data exchange between option board PROFIBUS-DP and the Ωmega Drive-Line II. This function block is used for the initialisation of the PROFIBUS-DP, that uses process data and service data communication. The function block should be requested in the cyclical program or an event task.



Overview

A PB_CTRL_BMSTRUCT data-type global variable is declared. This variable and its structural elements allow access to the option board PROFIBUS-DP.

In the PROPROG wt II project, a

```
PB_CTRL_BMSTRUCT data-type global variable
```

is generated and linked to the basic address of the PROFIBUS-DP slave interface of the option board PROFIBUS-DP-Slave,

```
%MB3.3000000
```

Example

```
_PB_Base AT %MB3.3000000 : PB_CTRL_BMSTRUCT;
```

where:

`_PB_BASE`	the name of the variable with data type identifier '_' for struct
`PB_CTRL_BMSTRUCT`	the data type
`%MB3.3000000`	the basic address of the PROFIBUS-DP slave interface on the option board PROFIBUS-DP-Slave.

The basic address of the option board PROFIBUS-DP-Slave is independent of the assigned slot:

Address	AT %MB3.3000000

4.2 Initialisation

The number of cyclical setpoints and actual values (double words) to be transmitted via the PROFIBUS-DP is specified at inputs si_NR_WR_VALUE and si_NR_RD_VALUE (permissible values are SINT#1 to SINT#8). The option board PROFIBUS-DP-Slave accepts these values during the first run of the function block (x_EN = TRUE) after switching on the power supply. After successful initialisation, output x_OK = TRUE, and the number of initialised setpoints and actual values are made available at outputs si_INIT_NR_WR_VALUE and si_INIT_NR_RD_VALUE. On bus startup, these values are compared to the configuration in the PROFIBUS-DP master. Message 3 at output b_MESSAGE indicates a no-match (after expiry of t_TIMEOUT_PB_DP, refer “Communication on the PROFIBUS-DP" on page 20).

NOTE

In this case, the option board parameters of the V controller have no impact on the PROFIBUS-DP configuration, the latter being dependent merely on function block PB_COMM_SLAVE.

4.3 Process data communication

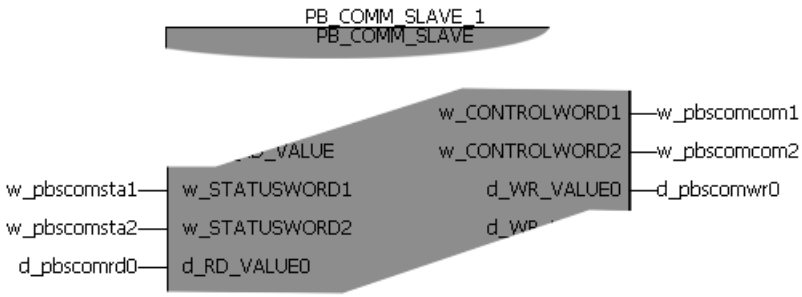
The two control words and the setpoints transmitted by the PROFIBUS-DP master to the option board PROFIBUS-DP-Slave are available at outputs w_CONTROLWORD1/w_CONTROLWORD2 and d_WR_VALUE0 to d_WR_VALUE7. The status words and actual values to be transmitted to the PROFIBUS-DP master must be specified at inputs w_STATUSWORD1/w_STATUSWORD2 and d_RD_VALUE0 to d_RD_VALUE7.

All values will be processed on function block PB_COMM_SLAVE request.

If process data are to be passed from / to the V controller, it is necessary to establish BAPS communication with the help of function blocks BAPS_INIT and BAPS_PD_COMMx (from the library SYSTEM1_DLII_20bd00 or higher) between Ωmega Drive-Line II and V controller.

Example:

Control word, status word and the first setpoint and actual value are to be transmitted between PROFIBUS-DP master and V controller.



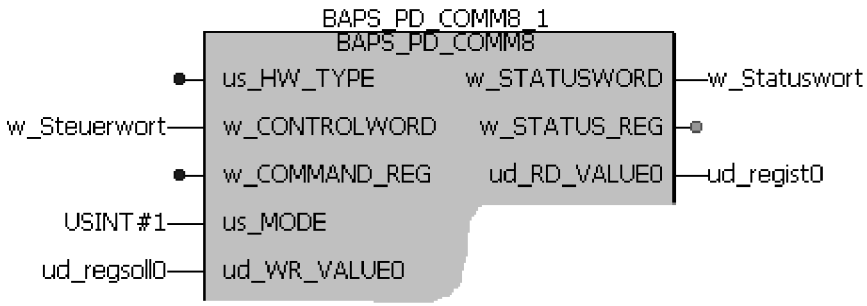
LD w_pbscomcom1

ST w_Steuerwort

LD d_pbscomwr0

DWORD_TO_UDINT

ST ud_regsoll0



LD w_Statuswort

ST w_pbscomsta1

LD ud_regist0

UDINT_TO_DWORD

ST d_pbscomrd0

4.4 Service data communication

4.4.1 Service data communication with Ωmega Drive-Line II

1048 double word parameters are available for the Ωmega Drive-Line II. The PROFIBUS-DP master can write these with the parameter numbers PNU1000 to PNU2047 and read them (refer “Net data block" on page 35). Access from the Ωmega Drive-Line II is via a PBOMEGA_BMARRAY data-type array to be linked to input a_OMEGA_SD_DATA. The data type PBOMEGA_BMARRAY is a field with 1048 entries of data type DINT, with the index starting at 1000 and ending at 2047.

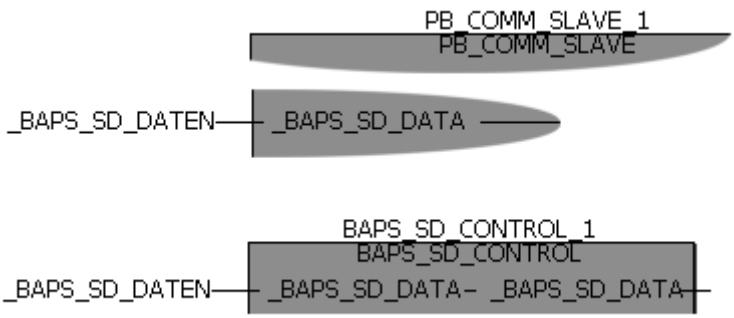
Example:

```
VAR
a_ParValues     :     PBOMEGA_BMARRAY:
END_VAR
```

The values of the parameters for service data communication between PROFIBUS-DP master and Ωmega Drive-Line II (parameters 1000 to 2047) are then entered into the field elements a_ParValues[1000] to a_ParValues[2047], for example, the parameter number matching the index.

4.4.2 Service data communication with the V controller

Direct access to all V controller parameters is possible from the PROFIBUS-DP master via service data communication, merely requiring specification of the parameter number of the V controller as parameter number [PNU] in the master. Within the Ωmega Drive-Line II, the function blocks BAPS_PAR_READ, BAPS_PAR_WRITE and BAPS_SD_CONTROL are responsible for service data communication with the V controller. The first two function blocks are already integrated in function block PB_COMM_SLAVE, so that only function block BAPS_SD_CONTROL is still required. Only a BAPS_BMSTRUCT data-type structure must be linked to input _BAPS_SD_DATA of function block PB_COMM_SLAVE and the same structure to input _BAPS_SD_DATA of function block BAPS_SD_CONTROL to allow service data communication between PROFIBUS-DP master and V controller, so that a PROFIBUS-DP master access is simply passed on. Also refer “Net data block" on page 35.



4.5 Monitoring

4.5.1 Process data communication

A TIME data-type variable must be linked to input t_TIMEOUT_CYCLIC to monitor process data communication between the PROFIBUS-DP slave interface on the option board PROFIBUS-DP-Slave and the Ωmega Drive-Line II. If a process data communication process is started and not completed within the specified time, the error output b_ERR is set with the associated value (refer “Error evaluation" on page 21). A default assignment of t_TIMEOUT_CYCLIC = TIME#1000ms is applied if the variable is not linked. Monitoring is off at t_TIMEOUT_CYCLIC = TIME#0ms.

NOTE

Monitoring in this case covers only the process data communication between the PROFIBUS-DP slave interface on the option board PROFIBUS-DP-Slave and the Ωmega Drive-Line II, not the communication via the PROFIBUS-DP.

4.5.2 Service data communication

A TIME data-type variable must be linked to input t_TIMEOUT_SD_DATA to monitor service data communication between the PROFIBUS-DP slave interface on the option board PROFIBUS-DP-Slave and the Ωmega Drive-Line II and between Ωmega Drive-Line II and V controller. If a data exchange process is started and not completed within the specified time, the error output b_ERR is set with the associated value (refer “Error evaluation" on page 21). A default assignment of t_TIMEOUT_SD_DATA = TIME#5000ms is applied if the variable is not linked. Monitoring is off at t_TIMEOUT_SD_DATA = TIME#0ms.

NOTE

Monitoring in this case covers only the service data communication between the PROFIBUS-DP slave interface on the option board PROFIBUS-DP-Slave and Ωmega Drive-Line II and between Ωmega Drive-Line II and V controller, not the communication via the PROFIBUS-DP.

4.5.3 Communication on the PROFIBUS-DP

A TIME data-type variable must be linked to input t_TIMEOUT_PB_DP to monitor communication on the PROFIBUS-DP. Output b_MESSAGE displays 16#02 if no data are exchanged via the PROFIBUS-DP for a time longer than the specified period. A default assignment of t_TIMEOUT_PB_DP = TIME#1000ms is applied if the variable is not linked. Monitoring is off at t_TIMEOUT_PB_DP = TIME#0ms.

The output b_MESSAGE may display the following values:

b_Message	Description
16#00	No warning, no error
16#01	General error
16#02	Field bus error
16#03	Setpoint / actual value configuration error

4.6 Error evaluation

Any errors are specified in more detail at output b_ERR of function block PB_COMM_SLAVE.

Output b_ERR Bit number	**Description**
0	Reserved
1	No access to option board, function block not processing data
2	Reserved
3	Service data access on option board timeout (refer “Service data communication" on page 19), function block not processing service data
4	Service data access on V controller timeout (refer “Service data communication" on page 19), function block not processing service data
5	Setpoint number error, function block not processing cyclical data. (SINT#1 ≤ si_NrWriteValue ≤ SINT#8 not met)
6	Actual value number error, function block not processing cyclical data. (SINT#1 ≤ si_NrReadValue ≤ SINT#8 not met)
7	Cyclical data transmission timeout (refer “Process data communication" on page 19), function block not processing cyclical data

Errors no longer present may be acknowledged by setting input x_QUIT of function block PB_COMM_SLAVE.

5 PROFIBUS-DP SETTINGS

5.1 Setting the PROFIBUS-DPdevice address

To achieve unequivocal identification within the bus system, each device must be set an address for initialisation on the PROFIBUS-DP.

In the case of the option board PROFIBUS-DP-Slave, this is done with the help of the two rotary switches S40 and S41. As soon as the device is switched on, the address set at the switches is accepted, leading to initialisation on the bus.

The address setting must be hexadecimally coded, with the lower nibble (4 bit) to be set at the lower rotary switch (S41) and the upper nibble at the upper rotary switch (S40). The address is accepted after reactivating the 24 V supply. In other words, the device must be restarted after a change to the setting for the modification to become effective.

Address setting must also observe the maximum possible slave address limits on the PROFIBUS-DP master.

Example

The PROFIBUS-DP device address 25 is to be set:

The decimal 25 corresponds to a hexadecimal 19, i. e., the lower nibble is hexadecimal 9, the upper hexadecimal 1.

- Set upper rotary switch (S40) to 1.
- Set lower rotary switch (S41) to 9.

5.2 Settings on the master

The PROFIBUS-DP master must be configured via the software accompanying delivery. For this purpose, the following PROFIBUS-DP slave information must be made available to the PROFIBUS-DP master:

- Device address of the PROFIBUS-DP slave at the PROFIBUS-DP.
- Input and output data of the PROFIBUS-DP slave (observe maximum input and output data limits at consistent transmission in the PROFIBUS-DP master).
- Special slave characteristics

NOTE

Any changes to the settings at the PROFIBUS-DP master require a complete system restart for reinitialisation.

5.2.1 Device description file (GSD file)

The PROFIBUS-DP master must have various pieces of information about the individual PROFIBUS-DP slaves to utilise the functions at the PROFIBUS-DP. For this purpose, the GSD file (device description file) accompanying delivery is required for PROFIBUS-DP master system configuration. The GSD file was generated in accordance with the standard *EN 50170 Volume 2 PROFIBUS* and contains defined information on each device for the configuration of data transmission and bus system startup behaviour.

It is recommended to copy the GSD file and the two bitmap files from the enclosed diskette into the relevant directories for the PROFIBUS-DP master configuration tool.

5.3 Defining the address data

5.3.1 Data exchange principles

Data exchange at the PROFIBUS-DP requires definition of the volume of the data to be transmitted. The transmission data volume (also address data) is dependent on the number of data to be transmitted and must be configured accordingly in the PROFIBUS-DP master. The transmission data volume may not be modified during operation; the bus system must be restarted for any changes to be accepted.

The transmission data volume is dependent on the number of cyclical setpoints and actual values and on whether or not service data communication is to be applied.

The required transmission data volume may be defined as follows:

- Always define a 2-word block each for the input and output data for the transmission of control and status words.
- Each cyclical actual value requires the configuration of a 2-word block as input data, each cyclical setpoint a 2-word block as output data at the PROFIBUS-DP master.
  PPO types (predefined transmission types for PROFIBUS-DP) can not be used as these operate on a 1-word format base, and a 2-word = 32 bit specification is required.
- A 4-word block must be defined as input and output data for the application of service data communication.

The complete address data, consisting of service data and cyclical data, is also referred to as net data block. For information on the net data block makeup refer chapter “Makeup, net data block" on page 35.

5.3.2 Block-consistent data transmission

Data transmission in the case of the PROFIBUS-DP is usually effected at a 2-word consistency, which means that all data within these 2 words are guaranteed to originate from the same PROFIBUS-DP master cycle. This may lead to problems with service data communication in some master systems as data consistency is not ensured across the complete 4-word service data block. The option board PROFIBUS-DP-Slave therefore also supports block-consistent data transmission, i. e., data consistency across the complete transmission data volume.

Application of consistent data transmission also requires the PROFIBUS-DP master to support the necessary input and output transmission data. The different PROFIBUS-DP masters vary in this respect, with the Siemens S7, for example, being capable of a mere 32-byte block-consistent transmission data volume. The associated system functions of the master interface must also be activated to ensure consistent transmission by the master.

5.3.3 Settings, I/O data, standard modules

The required transmission data volume is determined via setting codes in the PROFIBUS-DP master. To simplify these settings, frequently required settings are offered as standard modules in the GSD file. These standard modules always include status and control word as well as a defined number of setpoints and actual values (also refer “Standard modules without service data communication" on page 28 to “Standard modules, block-consistent transmission" on page 29), but **no** service data communication.

The master must be parameterised as follows if these standard modules are used (without service data communication):

Module / identifier	Designation	Input address	Output address	Comments
0	x actual value(s), x setpoint(s)	xxxx	xxxx	Enter required standard module
1				
2				
3				

It is possible to configure a service data module in addition to a standard module. In such case, the PROFIBUS-DP master must be set as follows:

Module / identifier	Designation	Input address	Output address	Comments
0	Service data module	xxxx	xxxx	Enter service data module
1	x actual value(s), x setpoint(s)	xxxx	xxxx	Enter required standard module
2				
3				

It is important in this context to parameterise the service data module prior to the standard module (also refer “Makeup, net data block" on page 35).

NOTE

If service data communication is to be applied, it is generally recommended to use block-consistent transmission.

Standard modules are available for block-consistent transmission as well and always include status and control word, a defined number of setpoints and actual values as well as service data communication. Note also that the modules for block-consistent transmission may not be combined with additional individual modules.

Application of the standard modules ‘Complete address data consistency’ requires the PROFIBUS-DP master to be set as follows:

Module / identifier	Designation	Input address	Output address	Comments
0	Data block x actual value / setpoint	xxxx	xxxx	Enter required standard module
1				
2				
3				

Application of consistent data transmission also requires activation of the associated system functions in the PROFIBUS-DP master so that the data may be transmitted in accordance with the settings at the slave.

Chapter “I/O data identifiers" on page 28 includes a list of the individual modules.

5.3.4 Settings, I/O data, general

The transmission data volume may be set via standard modules as well as via individual modules. In the latter case, the required volume may be configured exactly to the necessary number of setpoints and actual values, if at the expense of complete-data consistency.

The following formula may be applied to determine the required I/O data:

Formula for the calculation of the complete I/O data

Input data = 4AI word service data + 2AI status word + (number of cyclical actual values x 2AI word)
Output data = 4AO word service data + 2AO status word + (number of cyclical setpoints x 2AO word)

The PROFIBUS-DP master must be parameterised as follows for this application:

Module / identifier	Designation	Input address	Output address	Comments
0	Service data module	xxxx	xxxx	Enter service data module
1	Status / control module	xxxx	xxxx	Enter status / control module
2	x actual value(s)	xxxx		Select required number of actual values from list
3	x setpoint(s)		xxxx	Select required number of setpoints from list

NOTE

Configuration must adhere to the sequence of the individual modules.

If service data communication is required, the service data module must always be entered prior to the status / control word module and the setpoint / actual value modules. The status / control word module must always exist and have been entered prior to the setpoints / actual values. At least 1 cyclical setpoint and 1 cyclical actual value must be parameterised per device. For the identifiers of the individual modules refer to the overview in chapter “I/O data identifiers" on page 28.

The settings thus defined enable the option board PROFIBUS-DP-Slave to perform a configuration check on power-on and to detect a parameterisation error.

The transmission format of the individual values is generally defined at 32 bits. The control and status word is also defined at 32 bits by the option board PROFIBUS-DP-Slave.

5.4 I/O data identifiers

Setting the transmission data volume in the PROFIBUS-DP master is via the identifiers of the individual modules. For easier configuration, these are entered with their designations in the GSD file accompanying the delivery and, as a rule, are also displayed with their designations by the PROFIBUS-DP master operating software.

Information contained in the GSD file can not influence the arrangement of the individual selection options in the respective configuration interface of the PROFIBUS-DP master system, so that, on each new configuration of a device, the selection may appear freshly sorted on the PC.

The identifiers below along with the associated designations are therefore intended to give a better overview of the possible settings, and for control purposes. The actual identifier is in column *PROFIBUS-DP identifier*, and contains the information required for the PROFIBUS-DP master to set the system configuration.

5.4.1 Standard modules without service data communication

Designation	Service data	Status and control word	Address data	Identifier PROFIBUS-DP
1 actual value, 1 setpoint	No	Yes	4 AI/AO	0x73 (115_dez.)
2 actual values, 2 setpoints	No	Yes	6 AI/AO	0x75 (117_dez.)
4 actual values, 4 setpoints	No	Yes	10 AI/AO	0x79 (121_dez.)
8 actual values, 1 setpoint	No	Yes	18 AI, 4 AO	0xC0, 81, 67
8 actual values, 8 setpoints	No	Yes	18 AI, 18 AO	0xC0, 81, 81

5.4.2 Individual modules for configuration

Designation	Address data	Identifier PROFIBUS-DP
Service data	4 AI/AO	0x73 (115_dez.)
Status and control word	2 AI/AO	0x71 (113_dez.)
1 actual value	2 AI	0x51 (81_dez.)
2 actual values	4 AI	0x53 (83_dez.)
3 actual values	6 AI	0x55 (85_dez.)
4 actual values	8 AI	0x57 (87_dez.)
5 actual values	10 AI	0x59 (89_dez.)
6 actual values	12 AI	0x5B (91_dez)
7 actual values	14 AI	0x5D (93_dez.)
8 actual values	16 AI	0x5F (95_dez.)
1 setpoint	2 AO	0x61 (97_dez.)
2 setpoints	4 AO	0x63 (99_dez.)
3 setpoints	6 AO	0x65 (101_dez.)
4 setpoints	8 AO	0x67 (103_dez.)
5 setpoints	10 AO	0x69 (105_dez.)
6 setpoints	12 AO	0x6B (107_dez.)
7 setpoints	14 AO	0x6D (109_dez.)
8 setpoints	16 AO	0x6F (111_dez.)

5.4.3 Standard modules, block-consistent transmission

Designation	Service data	Status and control word	Address data	Identifier PROFIBUS-DP
1 actual value, 1 setpoint	Yes	Yes	8 AI/AO	0xF7 (247_dez.)
2 actual values, 2 setpoints	Yes	Yes	10 AI/AO	0xF9 (249_dez.)
4 actual values, 4 setpoints	Yes	Yes	14 AI/AO	0xFB (251_dez.)
5 actual values, 5 setpoints	Yes	Yes	16 AI/AO	0xFD (253_dez)
6 actual values, 6 setpoints	Yes	Yes	18 AI/AO	0xFF (255 _dez)
7 actual values, 7 setpoints	Yes	Yes	20 AI, 20 AO	0xC0,209, 209
8 actual values, 1 setpoint	Yes	Yes	22 AI, 8 AO	0xC0, 213, 199
8 actual values, 8 setpoints	Yes	Yes	22 AI, 22 AO	0xC0, 213, 213

NOTE

For block-consistent transmission, these must also be supported by the PROFIBUS-DP master, with the maximum data blocks varying in size. Note also that the modules for block-consistent transmission may not be combined with additional individual modules.

5.5 Example, address allocation

The following example illustrates the configuration of the transmission data volume.

The requirement is for a configuration of 3 cyclical setpoints and 1 cyclical actual value with service data communication. The address data range is automatically configured as follows by the PROFIBUS-DP master control:

Input data in the master (response service data and cyclical actual values);

4 AI service data (PKE, IND, PWE 1, PWE 2)

2 AI status word

2 AI actual value 1

The input data at the PROFIBUS-DP master therefore consists of 8 input words (8 AI).

Output data in the PROFIBUS-DP master (command service data and cyclical setpoints):

4 AO service data area

2 AO control word

6 AO setpoints 1 to 3

The output data at the PROFIBUS-DP master therefore consists of 12 output words (12 AO).

Setting the I/O data for this parameterisation

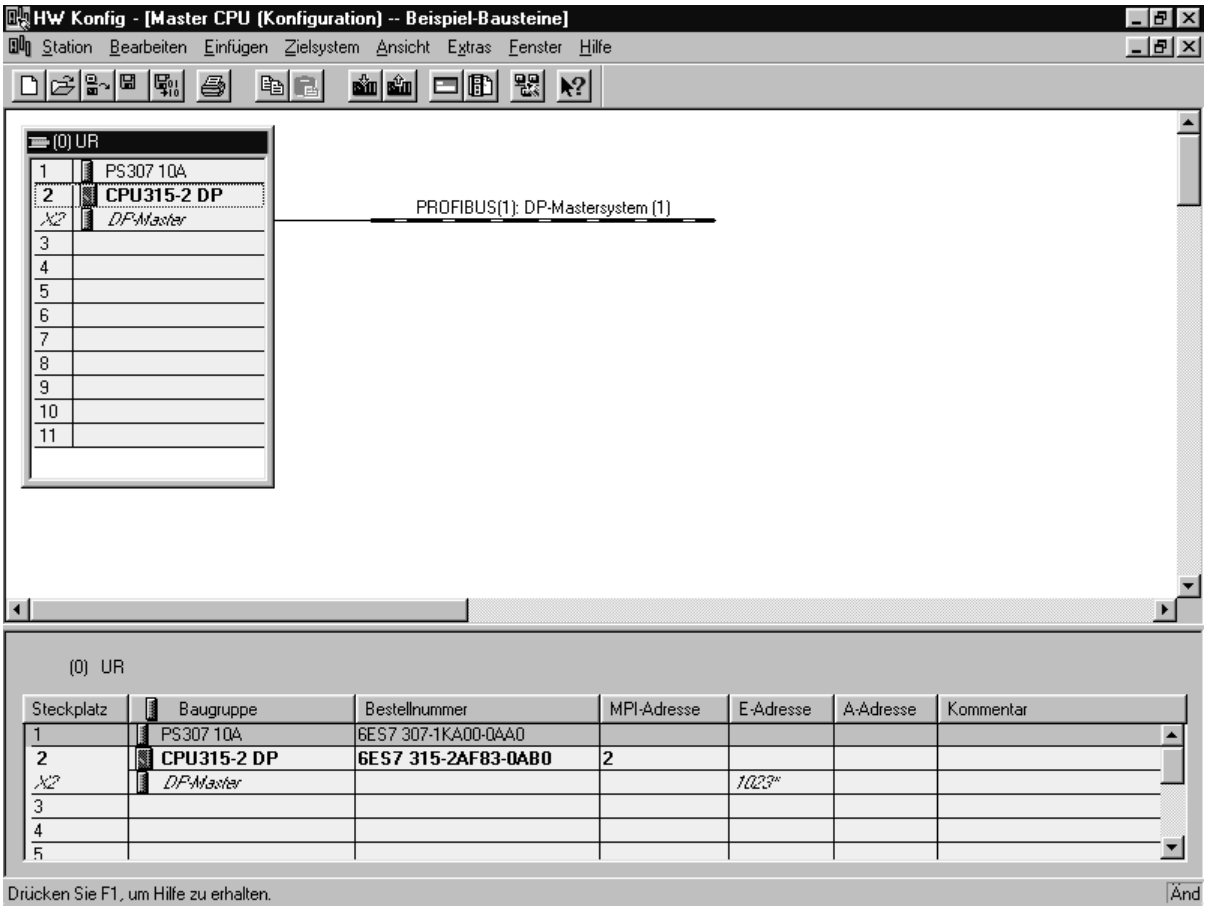
Module / identifier	Designation	Input address	Output address	Comments
0	Service data module	xxxx	xxxx	Enter service data module
1	Status / control module	xxxx	xxxx	Enter status / control module
2	1 actual value	xxxx		Enter 1 cyclical actual value
3	3 setpoints		xxxx	Enter 3 cyclical setpoints

5.6 Sample configuration using an S7

The chapter below describes the configuration of option board PROFIBUS-DP-Slave in a setup using a Siemens S7 as the PROFIBUS-DP master.

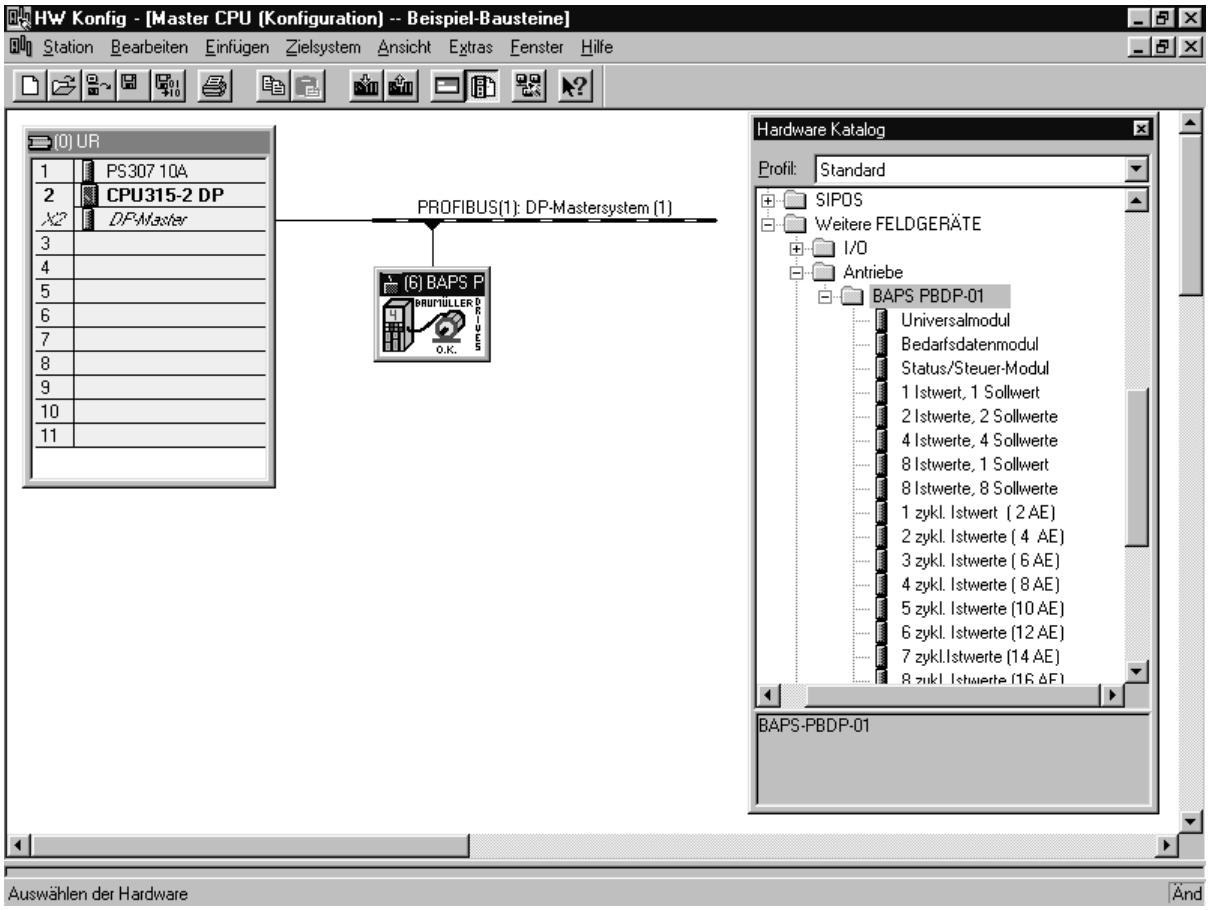
The option board PROFIBUS-DP-Slave must first be integrated into the PLC programming interface before it can be used (also refer Siemens description for the respective programming interface version used).

Then, the PROFIBUS-DP master system (an S7-300 DP in this case) is configured accordingly to enable PROFIBUS-DP communication:

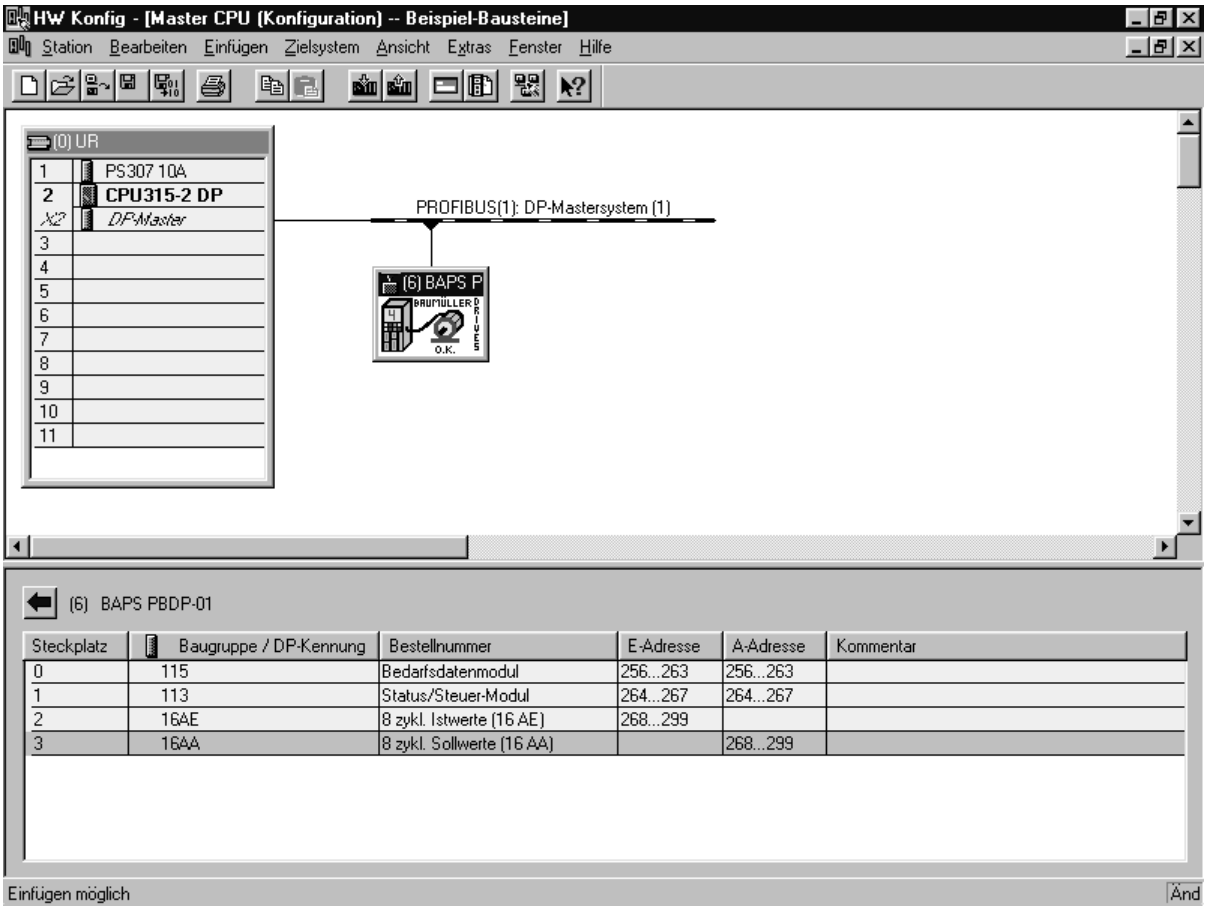


After successfully importing the Baumüller GSD file, activating the interface's hardware catalog allows location of the module *BAPS-PBDP-01* in the PROFIBUS component directory under *Other field devices -> drives*.

The *PROFIBUS* communication path must be activated to set up a Baumüller PROFIBUS user (PROFIBUS-DP bold underlined), so that a PROFIBUS device may be set up by double-clicking the entry *BAPS-PBDP-01*. This double-click activates a window for device bus address setting. Entry and confirmation of the address calls up the following window in the hardware configuration of the programming interface:



The device must now be selected with the mouse to set the configuration of the Baumüller PROFIBUS user -> window with *slot, module / DP identifier, ...* is activated. To enter the necessary data (sample parameterisation 8 setpoints / actual values and service data), select*slot 0* with the mouse to install this communication component by double-clicking *Service data module* in the hardware catalog. Then set the *status / control module*, the *8 cyclical actual values* and the *8 cyclical setpoints* in the same manner. The user may freely select the settings of the addresses of the individual communication components.



After all the necessary parameterisations have been completed, the resulting configuration needs to be transferred into the target system. Before the data are transferred, you may perform a consistency check in the interface to locate any parameterisation errors.

The PROFIBUS-DP should start without communication problems after a system (master and slaves) restart. Should the S7 display any errors, the cause may be located and eliminated via the online diagnostic function of the hardware configuration.

If consistent data transmission is applied, the associated system functions in the S7 must be used for consistent data transmission.

For further information refer chapter “Block-consistent data transmission" on page 25.

6 NET DATA BLOCK

This chapter describes the makeup of the net data block and illustrates service data communication flow and protocol.

6.1 Makeup, net data block

The net data block (corresponds to the address data in the PROFIBUS-DP master and) consists of two areas.

- PKW [Parameter code value] = Service data communication area
- PZD [Process data area] = Process data communication area

The parameter code value area is the data area for service data communication and exists only if that was configured in the PROFIBUS-DP. The size of the parameter code value area is defined at 4 words and can not be changed.

The process data area is for process data communication with the Ωmega Drive-Line II, in other words, the high-speed data transmission of status and control word as well as setpoints and actual values. The size of this area is dependent on the selected configuration (also refer chapter “Defining the address data" on page 25.

A sample configuration with service data, status and control word as well as 2 cyclical setpoints and actual values is described below.

Sample net data block for service data, status / control word, 2 cyclical setpoints and actual values:

PKW area				**PZD area**					
PKE	IND	PWE 1	PWE 2	PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6
0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
4 AI for response service data				Status word 1	Status word 2	Cycl. actual value 1		Cycl. actual value 2	
4 AO for command service data				Control word 1	Control word 2	Cycl. setpoint 1		Cycl. setpoint 2	
				6 AI / AO for command / response PZD					

Setting the I/O data for this parameterisation

This configuration requires the following configuration in the PROFIBUS-DP master:

- 4 AI/AO for service data
- 6 AI for cyclical status word and 2 cyclical actual values
- 6 AO for cyclical control word and 2 cyclical setpoints

and may be achieved either via standard modules or by combining individual modules. The various options are briefly illustrated below under A), B) and C).

A) Using the standard modules for address data configuration

Module / identifier	Designation	Input address	Output address	Identifier PROFIBUS-DP
0	Service data module	xxxx	xxxx	16#73
1	2 actual values, 2 setpoints (including status / control word)	xxxx	xxxx	16#75

B) Setting the address data via individual modules

Module / identifier	Designation	Input address	Output address	Identifier PROFIBUS-DP
0	Service data module	xxxx	xxxx	16#73
1	Status / control module	xxxx	xxxx	16#71
2	2 actual values	xxxx		16#53
3	2 setpoints		xxxx	16#63

C) Block-consistent transmission

Module / identifier	Designation	Input address	Output address	Identifier PROFIBUS-DP
0	Data block 2 actual values / setpoints (includes status / control word and service data)	xxxx	xxxx	16#F9

On the above settings

All 3 parameterisations configure the same address data for data transmission. The above examples illustrate the board configuration options.

Parameterisations A and B are functionally identical, although parameterisation B allows easier change to a new and even different number of cyclical setpoints or actual values as only one entry with address data definition needs to be changed.

Option C, however, also differs in function as data transmission is block-consistent. This setup must also be supported by the associated PROFIBUS-DP master.

Allocating addresses in an S7 for the above example

The address allocation shown here is a mere example and, as a rule, looks slightly different in every application case. The intent is to illustrate the relationship between the net data block configured for the PROFIBUS-DP and the input and output addresses of the control.

Input data: Actual values and response service data

PKW area				**PZD area**					
PKE	IND	PWE 1	PWE 2	PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6
0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
4 AI for response service data				Status word 1	Status word 2	Cycl. actual value 1		Cycl. actual value 2	
ED10		ED14		ED18		ED22		ED26	
EW10	EW12	EW14	EW16						

Output data: Setpoints and request service data

PKW area				**PZD area**					
PKE	IND	PWE 1	PWE 2	PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6
0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
4 AI for response service data				Control word 1	Control word 2	Cycl. setpoint 1		Cycl. setpoint 2	
AD10		AD14		AD18		AD22		AD26	
AW10	AW12	AW14	AW16						

6.2 PKW area (service data)

The parameter code value area of the net data block is for the transmission of service data communication and exists only if service data communication was configured in the PROFIBUS-DP master. The size of this area is defined at 4 words and can not be changed.

6.2.1 Parameter code value area makeup

The complete parameter code value area is divided into the areas PKE (parameter code, for parameter number and command and / or response code) IND (index, for access to parameter-specific information) and PWE 1 as well as PWE 2 (for the value to be transmitted). The makeup is the same for master to slave and slave to master data transmission.

The individual sub-areas are described in the chapters below. The following table gives a subdivision overview.

1 word	1 word	1 word	1 word
PKE	IND	PWE 1	PWE 2
Command / response code	Index	High-word value	Low-word value

6.2.2 Parameter code [PKE] makeup

PKE		
AK	R	PNU
15 14 13 12	11	10 9 8 7 6 5 4 3 2 1 0

The *parameter code [PKE]* in the parameter code value [PKW] area consists of the *parameter number PNU* (bits 0 to10), a spare bit *R* (bit 11) and the command and / or response code *[AK]* (bits 12 to 15). This parameter code [PKE] makeup is what allows every parameter in the controller to be read or written.

From the PROFIBUS-DP master end, the transmit and receive parameter codes differ only in the meaning of the command / response code [AK] field. During a transmit, it displays the command code, and the controller response code during a receive.

These command and response codes define the exact meaning of the other fields in the PKW area.

It is important for service data communication flow that the PCO field be the last to be written with the new command, i. e. after the other fields (IND and PWE) have been entered. On command completion, the parameter code field [PKE] must be the first to be set to 0 as this is the only way to ensure that no invalid command will be transmitted.

6.2.3 Command codes, PROFIBUS-DP master to drive

The PROFIBUS-DP master defines the requested action via the command codes. The following table gives an overview of the permissible command codes. For a detailed description with examples refer chapters from “Service data communication" on page 43.

Parameter code [PKE] (without PNU [parameter number])	Function	Description
16#0000 [1] [2]	No command	No command for service data
16#1xxx [1] [2]	Request PWE	Read parameter from controller
16#2xxx [2]	Change PWE (word)	Write parameter word format to controller
16#3xxx [1] [2]	Change PWE (double word)	Write parameter double word to controller
16#4xxx [2]	Request parameter description element [PBE]	Read parameter description

[1]) Permissible codes for Ωmega Drive-Line II parameters

[2]) Permissible codes for V controller parameters

6.2.4 Response code, drive to PROFIBUS-DP master

The drive uses appropriate defined response codes to confirm each command with a response after processing.

Parameter code [PKE] (without PNU [parameter number]	Function	Description
16#0000 [1] [2]	No response	No response
16#1xxx [2]	Transmit PWE (word)	Transmit parameter word format
16#2xxx [1] [2]	Transmit PWE (double word)	Transmit parameter double word
16#3xxx [2]	Transmit parameter description element [PBE]	Transmit element from parameter description
16#7xxx [1] [2]	Cannot execute command	PWE error code -> error list

[1]) Permissible codes for Ωmega Drive-Line II parameters

[2]) Permissible codes for V controller parameters

Once the command has been successfully executed, the relevant response code is transmitted, and the value is signalled back in the associated input data as a mirror value.

If a command cannot be executed, the response code ‘Cannot execute command’ is transmitted, and an error code is returned in PWE 2 that can be evaluated by the PROFIBUS-DP master.

6.2.5 Error codes in PWE 2

If the drive cannot execute a command, it returns the response code *16#7xxx* for *Cannot execute command*, and an error is signalled in the PWE 2 area:

PWE 2 value	Description
16#0000	No error
16#0001	Cannot change parameter
16#0002	Value outside MIN / MAX limit
16#0003	Wrong index entry
16#0004	No array
16#0005	Wrong data type
16#0006	No setting permitted
16#0007	Cannot change description element
16#0008	Impermissible parameter number
16#0009 to 16#0079	Reserved
16#0080	Received uninterpretable command
16#0081	No configuration / initialisation performed
16#0082	Error reading actual value
16#0083	Error writing setpoint
16#0084	Actual value configuration error
16#0085	Setpoint configuration error
16#0086 to 16#0100	Reserved
16#0101	Unspecified error
16#0102	Service not implemented
16#0103	Parameter format too large for PKW area
16#0104	Parameter information does not exist (request via index)

6.2.6 Contents of the IND (index) field

Index value	Read action
16#0001	Parameter attribute
16#0002	Parameter info
16#0005	Parameter minimum value
16#0006	Parameter maximum value

The above contents of the *IND* (index) field is valid only in combination with the *Request parameter description element [PBE]* command codes. In this command, the value in IND must be used to define in more detail the element to be read from the parameter description.

NOTE

For commands other than *Request parameter description element [PBE]* , the field IND must always be zero !

6.2.7 Reading parameter attribute feedback

When reading the parameter attribute via the command *Request parameter description element [PBE]*, the drive returns the attribute information in the element PWE 2 in a special code. The individual bits are described in the table below.

Bit	Description	Definition	Comments
0 - 1	Data length of an element in bytes	00 1 Byte 01 2 Byte 10 4 Byte 11 8 Byte	
2 -3	Data type of an element	00 SIGNED 01 UNSIGNED 10 FLOAT	Integer with polarity Integer without polarity Floating-point number
4 - 5	Number of elements	00 FIXED 01 VARIABLE	One element Variable number of elements
6 - 7	Element type	00 DATA 01 COMMAND	
8 - 11	Display format	0000 BIN 0001 DEC 0010 HEX 0100 NORM 0101 FIX 0110 SCI 0111 ENG 1000 ASCII	Binary Decimal Hexadecimal Floating point without exponent Floating point with fixed exponent Floating point, scientific Floating point, engineering ASCII character
12 - 15	Positions after comma	0000 0 1111 15	No positions after comma 15 positions after comma

6.2.8 Reading parameter info feedback

When reading the parameter information via the command *Request parameter description element [PBE]*, the drive returns further information on the parameter in the element PWE 2. This information, too, is bit-coded. For a description refer table below.

Bit	Description	Definition	Comments
0 – 2	Write protection	000 UNPROTECTED 001 PROTECTED	Not protected Write-protected
3	Save mode	0 NONSTORE 1 STORE	Parameter value is not stored. Parameter value is stored in data record.
4 – 15	Not assigned		

6.3 Service data communication flow

Service data communication has been provided for comprehensive parameterisation and diagnostic capability. The drive processes service data commands at low priority so that no guaranteed processing or refresh time may be specified.

Service data communication flow must follow the sequence below:

1. Enter values in PWE and IND
2. Then (!) enter parameter number and command code in PKE [parameter code].
3. Wait for drive response (or board error acknowledgement)
4. Read values received from drive
5. Complete command by resetting PKE , PWE and IND to zero

Service data communication is subject to the following principles:

- Only ever one command must be active per PROFIBUS-DP slave, i. e., service data communication flow must be as follows:
  1. Activate service data function from PROFIBUS-DP master.
  2. Wait for drive response.
  3. Reset service data function by writing the address data *Service data* with zero.
  - A current command can always be reset with the function *No command* (write the address data Service data with zero).
- Steps must be taken to ensure in the PROFIBUS-DP master by the program or via *Control / observe variable* that the value in PKE is entered only after valid values appear in PWE and IND.
- The PROFIBUS-DP master must trap any response codes other than expected.
- The drive expects no acknowledgement from the PROFIBUS-DP master as to whether the controller response arrived.
- Service data communication is asynchronous in relation to cyclical setpoint and actual value communication, with the option board PROFIBUS-DP-Slave attempting to execute the requested function within a specific period of time. If the function can currently not be executed, the system returns an error message, so that a new attempt to execute the function may be started.

6.4 Service data communication

6.4.1 Reading parameter (word or double word)

Makeup of the transmitted PKW area:

PKE	IND	PWE 1	PWE 2
1xxx	0000	0000	0000

- Command code for parameter value read: 1
- Parameter number in PNU [parameter number] field of PKE [parameter code].
- Index field (IND) always 0.

Error-free execution returns the following responses as a function of the parameter format (word or double word):

For word parameters, the response code is 1, and the parameter contents are in PWE 2.

PKE	IND	PWE 1	PWE 2
1xxx	0000	0000	xxxx

For double word parameters, the response code is 2, and the parameter contents are in PWE 1 (high word) and PWE 2 (low word).

PKE	IND	PWE 1	PWE 2
2xxx	0000	0000	xxxx

Should an error occur during execution, the system returns response code 7 (Cannot execute command), and the error cause is specified in more detail in the field PWE 2.

PKE	IND	PWE 1	PWE 2
7xxx	0000	0000	xxxx

Possible error causes

- The parameter to be read is parameterised as a cyclical setpoint or actual value, and service data communication is currently not possible.
- Service data communication not enabled at controller end ▸ Check P126 in the V controller.
- Function blocks PB_COMM_SLAVE and / or BAPS_SD_CONTROL not run.
- Previous service data communication was not reset with *No command* (PKE = 0).

Example

The parameter 218 L Rev actual value is to be read via the service data function. For this purpose, the following settings are made in the service data output data:

PKE	IND	PWE 1	PWE 2
10DA	0000	0000	0000

The following response is returned in the input data on *Error-free read*:

PKE	IND	PWE 1	PWE 2
20DA	0000	xxxx	xxxx

PWE 1 and PWE 2 show the double word value of parameter 218 L Rev actual value.

Errors return the following:

PKE	IND	PWE 1	PWE 2
70DA	0000	0000	0082

PWE 2 shows the error code 16#0082. The actual value cannot be read.

6.4.2 Writing parameter (word) (V controller parameters only)

Makeup of the transmitted PKW area:

PKE	IND	PWE 1	PWE 2
2xxx	0000	0000	xxxx

- Command code for word parameter write: 2
- Parameter number in PNU field of PKE.
- Index field (IND) always 0.
- Value to be written in PWE 2

Error-free execution is confirmed by response code 1 (Transmit PWE (word)) and the written value in PWE 2.

PKE	IND	PWE 1	PWE 2
1xxx	0000	0000	xxxx

Should an error occur during execution, the system returns response code 7 (Cannot execute command), and the error cause is specified in more detail in the field PWE 2.

PKE	IND	PWE 1	PWE 2
7xxx	0000	0000	xxxx

Possible error causes

- The parameter to be written is parameterised as a cyclical setpoint or actual value, and service data communication is currently not possible.
- Service data communication not enabled at controller end ▸ Check P126 in the V controller.
- Function blocks PB_COMM_SLAVE and / or BAPS_SD_CONTROL not run.
- Previous service data communication was not reset with *No command* (PKE = 0).

Example

Parameter *P127 M Communication monitoring* is to be set to value 4 (Activate process data communication monitoring).

PKE	IND	PWE 1	PWE 2
207F	0000	0000	0004

The following response is returned on *Error-free write*:

PKE	IND	PWE 1	PWE 2
107F	0000	0000	0004

PWE shows the copy of the written value.

Errors return the following:

PKE	IND	PWE 1	PWE 2
707F	0000	0000	0083

PWE 2 shows the error code 16#0083. The setpoint cannot be written.

6.4.3 Writing parameter (double word)

Makeup of the transmitted PKW area:

PKE	IND	PWE 1	PWE 2
3xxx	0000	xxxx	xxxx

- Command code for double word parameter write: 3
- Parameter number in PNU field of PKE.
- Index field (IND) always 0.
- Value to be written in PWE 1 (high word) and PWE 2 (low word)

Error-free execution is confirmed by response code 2 (Transmit PWE (double word)) and the written value in PWE 1 and PWE 2.

PKE	IND	PWE 1	PWE 2
2xxx	0000	xxxx	xxxx

Should an error occur during execution, the system returns response code 7 (Cannot execute command), and the error cause is specified in more detail in the field PWE 2.

PKE	IND	PWE 1	PWE 2
7xxx	0000	0000	xxxx

Possible error causes:

- The parameter to be written is parameterised as a cyclical setpoint or actual value, and service data communication is currently not possible.
- Service data communication not enabled at controller end ▸ Check P126 in the V controller.

- Function blocks PB_COMM_SLAVE and / or BAPS_SD_CONTROL not run.
- Previous service data communication was not reset with *No command* (PKE = 0).

6.4.4 Requesting parameter description element [PBE] (V controller parameters only)

Makeup of the transmitted PKW area:

PKE	IND	PWE 1	PWE 2
4xxx	xxxx	0000	0000

- Command code for parameter description element request: 4
- Parameter number in PNU field of PKE.
- Index field (IND) to requested description element.

Error-free execution is confirmed by response code 3 (Transmit parameter description element [PBE]) and the requested element in PWE 2.

PKE	IND	PWE 1	PWE 2
3xxx	xxxx	0000	xxxx

Should an error occur during execution, the system returns response code 7 (Cannot execute command), and the error cause is specified in more detail in the field PWE 2.

PKE	IND	PWE 1	PWE 2
7xxx	xxxx	0000	xxxx

Possible error causes

- The parameter to be written is parameterised as a cyclical setpoint or actual value, and service data communication is currently not possible.
- Service data communication not enabled at controller end ▸ Check P126 in the V controller.
- Function blocks PB_COMM_SLAVE and / or BAPS_SD_CONTROL not run.
- Previous service data communication was not reset with *No command* (PKE = 0).

Example

The maximum value of parameter *126 M Communication source* is to be read.

PKE	IND	PWE 1	PWE 2
407E	0006	0000	0000

The following response is returned on error-free read:

PKE	IND	PWE 1	PWE 2
307E	0006	0000	000F

PWE shows the maximum value of parameter 126: 16#000F.

Errors return the following:

PKE	IND	PWE 1	PWE 2
707E	xxxx	xxxx	xxxx

PWE 2 shows the related error code.

6.5 PROFIBUS-DP diagnostic function

The integrated diagnostic functions of the PROFIBUS-DP allow status information to be read out from the PROFIBUS-DP slave. Note that process data communication is halted for this function, and no further setpoints and actual values are transmitted via the PROFIBUS-DP.

NOTE

If communication monitoring is activated in the Ωmega Drive-Line II, activation of this diagnostic function triggers an error response as the cyclical transmission of setpoints and actual values is halted.

7 PROBLEMS, CAUSES AND POSSIBLE SOLUTIONS

7.1 Power-on sequence

Problem	Possible cause and elimination
Controller signals process data communication timeout Error 16#0003	The process data communication monitoring time set at the controller has expired. a) Ωmega Drive-Line II did not pass on any further values to the controller within the set monitoring time ▸ Check BAPS POUs b) Different PROFIBUS-DP master and slave configurations: ▸ Check POU for error message. ▸ Adapt PROFIBUS-DP master / slave settings where parameterisation differs. c) PROFIBUS-DP communication disabled in the V controller: ▸ Check parameter P126 M communication source.
Controller signals service data communication timeout Error 16#0004	The service data communication monitoring time set at the controller has expired. Service data monitoring is usually not appropriate as service data communication must be performed within the specified time period. If service data communication is requested, the option board PROFIBUS-DP-Slave monitors communication performance, and an error is returned as response code if a timeout period is exceeded. ▸ Deactivate service data communication monitoring.
No setpoint or actual value transmission	Different PROFIBUS-DP master and slave configurations: ▸ Check PWE in the service data area for error entry or ▸ Check POUs for error message. ▸ Set the address data and the device number in the hardware. ▸ Check PROFIBUS-DP master configuration. ▸ Check BAPS parameters in the controller: P 175 = 1; P 176 = 2; P 177 = 2; P 178 = 11 ▸ Check POU setting.
PROFIBUS-DP master bus error	▸ Check device number at the slave and in the PROFIBUS-DP master hardware configuration. ▸ Activate diagnostic function in the PROFIBUS-DP master system for troubleshooting, and follow instructions.

7.2 Process data communication

Problem	Possible cause and elimination
Controller error, cyclical data timeout (16#0003)	The monitoring time set at the controller has expired: Ωmega Drive-Line II did not pass on any further values to the controller within the set monitoring time ▸ Check BAPS POUs The PROFIBUS-DP did not pass on any further telegrams to the PROFIBUS-DP slave within the set monitoring time. ▸ Activate diagnostic function in the PROFIBUS-DP master hardware configuration to locate error cause. ▸ Check PROFIBUS-DP plugs at PROFIBUS-DP master and all PROFIBUS-DP slaves (plugged and both terminating resistors on ?). PROFIBUS-DP communication disabled in the V controller: ▸ Check parameter *P126 M communication source*. PROFIBUS-DP master operating status was changed ▸ Restart PROFIBUS-DP master.
PROFIBUS-DP master setpoint sometimes differs from expected setpoint	If the output process map (OPM) is applied directly as setpoint specification, it is possible that a wrong setpoint is passed to the controller due to program errors in the PROFIBUS-DP master. As a consequence of the system characteristics of some PROFIBUS-DP master systems (e. g. S7), the OPM data can not be output as status. ▸ For error elimination, either link the outputs to intermediate flags first before transfer into the output process map (OPM), or record and evaluate the telegrams from the PROFIBUS-DP master to the PROFIBUS-DP slave with the help of a PROFIBUS-DP analyser.
Refresh times differ from information in this description	Check data transmission with the help of bus analysis auxiliary programs as the option board merely represents one PROFIBUS-DP slave device at the PROFIBUS-DP and signals telegrams only on request from the PROFIBUS-DP master. ▸ Check the settings in the PROFIBUS-DP master for waiting periods during bus communication or PROFIBUS-DP master operating modes.

7.3 Service data communication

Problem	Possible cause and elimination
Error message, service data timeout (16#0004)	The monitoring time set at the controller has expired: The user program did not execute any service data communication within the set period of time. Service data communication time monitoring is usually inappropriate as service data communication is executed during communication standby time.
Unexpected drive response due to service data communication	▶ Check the command and response codes in the service data area for communication checkup. Where the service data (PWE area) specification differs from the returned value, ▶ follow the correct sequence in activating the function. Variations between specified and returned values despite correct activation sequence may be due to inconsistent value transmission by the PROFIBUS-DP master. ▶ Use area-consistent data transmission instead.
Specified value does to match return value in the response telegram	Where the PROFIBUS-DP slave returns a response other than specified by the PROFIBUS-DP master during service data communication, the causes may be as follows: Previous service data communication was not completely reset. ▶ Check user program in the control. Select the consistent data transmission mode as the applied PROFIBUS-DP master and the associated user program do not guarantee consistent transmission of the service data area. ▶ Check user program in the PROFIBUS-DP master control. ▶ Set consistent data transmission mode.
Return *Cannot execute command* (response PKE 16#7xxx)	The service data function could not be executed, error code refer PWE entry, service data. Service data communication disabled in the V controller: ▶ Check parameter P126 M communication source. Error code, cannot change parameter (16#0001) ▶ Parameter for display in the V controller only. Error code, impermissible parameter number (16#0008) ▶ Check whether the parameter number exists in the controller /Ωmega Drive-Line II.

7.4 Miscellaneous

Problem	Possible cause and elimination
Remote bit in the status word is not set	POUs not processed
GSD file can not be read in	▶ Substitute entry 'Slave_Family' by ' ; ' as some interfaces cannot read this information out for drive directory generation. The module is generated in the folder Miscellaneous after another read-in. ▶ Observe restrictions in the PROFIBUS-DP master interface with respect to GSD file read-in.

8 PARAMETER DESCRIPTIONS

Observe the following important peculiarities that some V controller parameters display in combination with the option board PROFIBUS-DP-Slave:

8.1 Parameters P 41 and P 247, delta phi 16

These parameters may be used to specify an angle difference directly with respect to the encoder position actual value. The specification may be via the connection of a digital input as the value is written once to the associated 16-bit parameter only upon a status change.

Cyclical specification via the option board PROFIBUS-DP-Slave directly to these parameters is not possible as the option board continuously writes the current values to the controller, with the latter providing for an offset upon each setpoint refresh. Cyclical data exchange with these parameters helps the controller detect a new offset value upon each refresh via the option board PROFIBUS-DP-Slave. This value is then added to the actual position of the associated encoder.

Application of this function via the PROFIBUS-DP therefore requires service data communication to be used as follows only:

The required value is written to the parameter via the service data function and then overwritten again with zero. Function execution can be checked and controlled via the service data function feedbacks.

8.2 Parameters P 42 and P 248, delta phi 32

The information given above for parameters P 41 and P 247 also applies for these parameters. These parameters may be written to in the way described above.

8.3 Parameters P1000 to P2047

Double word parameters in the Ωmega Drive-Line II

The only possible command codes (refer 5.2.3) are

- Request PWE
- Change PWE (double word)

9 ADDITIONAL INFORMATION

9.1 References

- Siemens ‘SIMATIC NET’ manual
  Industrial communications networks
  PROFIBUS networks
- Baumüller PROFIBUS protocol description
- Baumüller V controller description

9.2 On option board PROFIBUS-DP-Slave versions

9.2.1 Version 2.04

Version 2.04 of the option board PROFIBUS-DP-Slave supports area-consistent data transmission. For this purpose, additional PROFIBUS-DP codes have been included into the GSD file, that are not supported by older board versions.

The time slot method for data exchange with the controller and the BAPS transmission mode *2 setpoints and 2 actual values per cycle* are supported.

The board’s considerably improved processing speed is most noticeable in its faster evaluation capability.

Operation on the Ωmega Drive-Line II is possible from this version.

9.2.2 Version 2.02

Version 2.02 of the option board PROFIBUS-DP-Slave does not support all codes entered in the new GSD file valid with version 2.04. The new codes for area-consistent transmission are not supported. Otherwise, the new GSD file may also be used to configure version 2.02 PROFIBUS-DP-Slave option boards.

The evaluation capability of option board PROFIBUS-DP-Slave (version 2.02) does not match the one mentioned in this description.

Only the time slot method is supported for transmission to the controller.

9.2.3 Upgrading from version 1.x to version 2.x

The following aspects must be taken into consideration when upgrading from option boards PROFIBUS-DP-Slave V1.x to V2.x:

1. The new board functions require the new GSD file with new bitmaps. Keep the old GSD file for parameterising the old board as the two GSD files are not compatible.
2. New net data block definition
   A 32-bit data format was generally defined for option board PROFIBUS-DP-Slave control in the V controller due to the varying address formats of the PROFIBUS-DP master interfaces. This means that the same address data must be used at each CPU, allowing software sharing among the individual control systems of the PROFIBUS-DP master.

PLC system	PROFIBUS basic address
S7 300	Addresses divisible by 2
S7 400	Addresses divisible by 4

NOTE

The data format can not be changed as the controller system expects a 32-bit data format for position specifications.

3. The required address data must be adapted if the new board is used;
   ▸ Insert spare word after control / status word = 1-word PZD 2 (refer “Sample configuration using an S7" on page 31).
4. The board will now check the configurations in PROFIBUS-DP master and PROFIBUS-DP slave for consistency on power-on.
   ▸ Errors are indicated in the *controller BAPS parameters* or in the service data communication PWE area.
5. Extended service data area error list ▸ Error evaluation must be adapted as necessary.
6. The parameter *P179 BAPS Z Error response* has been outmoded for the new board. Monitoring is now via the parameter Communication monitoring.
7. Faster communication between option board PROFIBUS-DP-Slave and controller may lead to unexpected program responses of the control system, so that it is absolutely vital to check the interlocks and controls via the PLC program.

WARNING

Malfunctioning and personal injury may result from ignoring the above when exchanging the PROFIBUS-DP-Slave option boards.

9.3 Plugs and cables for PROFIBUS networking

Due to the standardised cabling of PROFIBUS systems, there are various manufacturers offering products for overall system networking, leaving to ensure only that the components and cables used conform to the PROFIBUS standard.

Selection of Siemens connector plugs

Cable exit	Vertical	Adjustable 0° - 30°	30° angle
Transmission rates	9,6 kBit/s...12 MBit/s	9,6 kBit/s...12 MBit/s	9,6 kBit/s...1,5 MBit/s
Terminating resistor	Integrated, add-on	Integrated, add-on	----------
Order reference numbers	6ES7 972-	6ES7 972	6ES7 972
Without	0BA10-0XA0	0BA20-0XA0	0BA30-0XA0
With PG jack	0BB10-0XA0	0BB20-0XA0	----------

NOTE

For easier error analysis, there should be at least one plug with PG jack per PROFIBUS ring for problem-free connection of a PROFIBUS analyser.

9.4 V controller status word

A quick and brief overview of the V controller status word (for further details refer to the V controller operating instructions).

Bit no.	Description	Comment
0	Ready to switch on	Device control status machine
1	Switched on	Device control status machine
2	Operation enabled	Device control status machine
3	Fault	Device control status machine
4	Voltage disabled	Bit = 0: Request *Disable voltage* is present
5	Quick stop	Device control status machine
6	Switch-on disabled	Device control status machine
7, 8	Spare	Spare
9	**Remote**	**Bit = 1: Controller can be approached via option board PROFIBUS-DP-Slave (enable process data communication)**
10	Setpoint reached	Description dependent on operating mode
11	Spare	Spare
12, 13	Dependent on operating mode	
14	Freely configurable	Parameterisation via P134
15	Freely configurable	Parameterisation via P135

9.5 Change to Baumüller system

In the operating mode **Velocity specification 1** of the V controller, the functions are designed as per DRIVECOM principle to possibly allow the use of existing software.

However, changes between different operating modes may lead to problems in the higher-level control system program as the functions in the control word are triggered by a High or Low-active bit dependent on the operating mode. ▸ Refer V controller description.

9.6 GSD file (device description file)

The GSD or type file contains all characteristics of the PROFIBUS-DP slave device to be configured via the PROFIBUS-DP master operating programs. The GSD file format is defined in the standard *EN 50170, volume 2, PROFIBUS*, so that every PROFIBUS-DP master can read the information from the GSD file.

NOTE

Where there occur any problems reading the GSD file into PLC interfaces, contact the PLC manufacturer first for known problems with his interface on GSD file read-in.

Current GSD file:

```
;===================================================================
;
; 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 file for BAPS module PROFIBUS-DP PBDP-01
;===================================================================
;
; Name  : BNF_00D0.GSD
; Version : V 1.41 (22.11.1999)
;
;===================================================================
;
; Modifications:
;
; a) Extended parameterisation, consistent data transmission
;    Complete data
;
;===================================================================
;
#Profibus_DP
; Unit-Definition-List:
GSD_Revision=1
Vendor_Name="BAUMÜLLER"
Model_Name="BAPS PBDP-01"
Revision="REV 1.00"
```

```
Ident_Number=0x00D0
Protocol_Ident=0
Station_Type=0
Hardware_Release="REV 1.10"
Software_Release="REV 2.04"
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=2
24V_Pins=0
Implementation_Type="SPC3"
Bitmap_Device="bnf_nor"
Bitmap_SF="bnf_err"
Bitmap_Diag="bnf_err"
;
; Slave-Specification:
OrderNumber="BAPS-PBDP-01"
;
Freeze_Mode_supp=1
Sync_Mode_supp=1
Auto_Baud_supp=1
Set_Slave_Add_supp=0
Min_Slave_Intervall=1
Max_Diag_Data_Len=16
Slave_Family=1 ; drives; if GSD cannot be read in, substitute this line with ';'.
;
; UserPrmData: Length and Preset:
User_Prm_Data_Len=5
User_Prm_Data=0x00,0x00,0x00,0x00,0x00
;
Modular_Station=1
Max_Module=4
Max_Input_Len=44
Max_Output_Len=44
Max_Data_Len=88
;
; Module definition list:
;
; Special modules
;
Modules='Service data module' 0x73
EndModule
Modules='Status / control module' 0x71
EndModule
;
; Standard modules, word consistency
;
```

```
Modules='1 actual value, 1 setpoint'   0x73
EndModule
Modules='2 actual values, 2 setpoints' 0x75
EndModule
Modules='4 actual values, 4 setpoints' 0x79
EndModule
Modules='8 actual values, 1 setpoint'  0xC0,81,67
EndModule
Modules='8 actual values, 8 setpoints' 0xC0,81,81
EndModule
;
; Standard modules, complete address data consistency
;
Modules='Data block 1 actual value / setpoint'  0xF7
EndModule
Modules='Data block 2 actual values / setpoints' 0xF9
EndModule
Module='Data block 3 actual values / setpoints' 0xFB
EndModule
Modules='Data block 4 actual values / setpoint'  0xFD
EndModule
Modules='Data block 5 actual values / setpoints' 0xFF
EndModule
Modules='Data block 6 actual values / setpoints' 0xC0,209,209
EndModule
Modules='Data block 7 actual values / setpoints' 0xC0,211,211
EndModule
Modules='Data block 8 actual values / 1 setpoint' 0xC0,213,199
EndModule
Modules='Data block 8 actual values / setpoints' 0xC0,213,213
EndModule
;
; User Configuration
;
; Modules, word-organised, with transmit data
;
Modules = '1 cycl. actual value  ( 2 AI)'  0x51
EndModule
Modules = '2 cycl. actual values ( 4 AI)'  0x53
EndModule
Modules = '3 cycl. actual values ( 6 AI)' 0x55
EndModule
Modules = '4 cycl. actual values ( 8 AI)'  0x57
EndModule
Modules = '5 cycl. actual values (10 AI)'  0x59
EndModule
Modules = '6 cycl. actual values (12 AI)' 0x5B
EndModule
Modules = '7 cycl. actual values (14 AI)'  0x5D
EndModule
Modules = '8 cycl. actual values (16 AI)' 0x5F
EndModule
;
; Modules, word-organised, with receive data
;
Modules = '1 cycl. setpoint  ( 2 AO)'  0x61
EndModule
Modules = '2 cycl. setpoints ( 4 AO)'  0x63
EndModule
Modules = '3 cycl. setpoints ( 6 AO)'  0x65
EndModule
Modules = '4 cycl. setpoints ( 8 AO)' 0x67
EndModule
```

Modules = ‘5 cycl. setpoints (10 AO)’ 0x69
EndModule
Modules = ‘6 cycl. setpoints (12 AO) ‘ 0x6B
EndModule
Modules = ‘7 cycl. setpoints (14 AO)’ 0x6D
EndModule
Modules = ‘8 cycl. setpoints (16 AO)’ 0x6F
EndModule

9.7 Bitmaps

NOTE

The bitmap names can not be changed as these are integrated into some PROFIBUS-DP master system configuration interfaces together with the GSD file.

A) Bitmap BNF_NORN.BMP

B) Bitmap BNF_ERRN.BMP

10 FUNCTION BLOCKS

10.1 PB_COMM_SLAVE

The function block PB_COMM_SLAVE communicates with option board PROFIBUS-DP-Slave and V controller via the BAPS interface.

NOTE

The function block PB_COMM_SLAVE uses the library BM_TYPES_20bd01 or higher.

Parameter input	Data type	Description
x_EN	BOOL	Enable
x_QUIT	BOOL	Error acknowledgement
t_TIMEOUT_PB_DP	TIME	Monitoring time for PROFIBUS-DP
t_TIMEOUT_CYCLIC	TIME	Monitoring time for process data communication
t_TIMEOUT_SD_DATA	TIME	Monitoring time for service data communication
si_NR_WR_VALUE	SINT	Number of setpoints
si_NR_RD_VALUE	SINT	Number of actual values
w_STATUSWORD1	WORD	Status word 1
w_STATUSWORD2	WORD	Status word 2
d_RD_VALUE0	DWORD	Actual value 0
d_RD_VALUE1	DWORD	Actual value 1
d_RD_VALUE2	DWORD	Actual value 2
d_RD_VALUE3	DWORD	Actual value 3
d_RD_VALUE4	DWORD	Actual value 4
d_RD_VALUE5	DWORD	Actual value 5
d_RD_VALUE6	DWORD	Actual value 6
d_RD_VALUE7	DWORD	Actual value 7
_BAPS_SD_DATA	BAPS_BMSTRUCT	Data for BAPS service data communication
_BASE	PB_CTRL_BMSTRUCT	Data for PROFIBUS-DPcommunication
a_OMEGA_SD_DATA	PBOMEGA_BMARRAY	Data for Ωmega service data communication

Parameter output	Data type	Description
x_OK	BOOL	Initialisation complete
b_MESSAGE	BYTE	Message
b_ERR	BYTE	Error number
si_INIT_NR_WR_VALUE	SINT	Number of initialised setpoints
si_INIT_NR_RD_VALUE	SINT	Number of initialised actual values
w_CONTROLWORD1	WORD	Status word 1
w_CONTROLWORD2	WORD	Status word 2

Parameter output	Data type	Description
d_WR_VALUE0	DWORD	Setpoint 0
d_WR_VALUE1	DWORD	Setpoint 1
d_WR_VALUE2	DWORD	Setpoint 2
d_WR_VALUE3	DWORD	Setpoint 3
d_WR_VALUE4	DWORD	Setpoint 4
d_WR_VALUE5	DWORD	Setpoint 5
d_WR_VALUE6	DWORD	Setpoint 6
d_WR_VALUE7	DWORD	Setpoint 7
_BAPS_SD_DATA	BAPS_BMSTRUCT	Data for BAPS service data communication
_BASE	PB_CTRL_BMSTRUCT	Data for PROFIBUS-DPcommunication
a_OMEGA_SD_DATA	PBOMEGA_BMARRAY	Data for Ωmega service data communication

Description of function block PB_COMM_SLAVE

Serial communication via the PROFIBUS-DP allows process data communication (process data transmission, cyclical communication) and service data communication (service data transmission). Process data transmission is in the PZD area, service data transmission in the PKW area of the PROFIBUS-DP protocol.

The PROFIBUS-DP slave interface on the option board PROFIBUS-DP-Slave receives setpoints, control word and parameter commands from a PROFIBUS-DP master and transmits actual values, status word and responses to parameter commands to the PROFIBUS-DP master.

The function block PB_COMM_SLAVE accepts the setpoints, control word and parameter commands from the PROFIBUS-DP slave interface on the option board PROFIBUS-DP-Slave.

- Setpoints and status words are made available at the outputs.
- The function block passes the parameter commands for parameters 0 to 999 on to the V controller directly via the BAPS.
- The function block outputs (parameter write) and / or expects (parameter read) the parameter variables for parameters 1000 to 2047 at the input / output a_OMEGA_SD_DATA.

The function block PB_COMM_SLAVE

- also accepts the actual values and
- the status words from the inputs as well as
- the responses to parameter commands for parameters 0 to 999 (from the V controller, directly via the BAPS)

and passes these data to the PROFIBUS-DP slave interface at the option board PROFIBUS-DP-Slave.

Setpoints, control words, actual values and status words are passed to and / or from the V controller via explicit use of function block BAPS_PD_COMM2, BAPS_PD_COMM24 or BAPS_PD_COMM8 via the BAPS.

Input x_EN:

TRUE at input x_EN enables the function block.

Input x_QUIT:

TRUE at input x_QUIT acknowledges the present errors.

Input t_TIMEOUT_PB_DP:

Input t_TIMEOUT_PB_DP may be used to specify a timeout period for the PROFIBUS-DP. Output b_MESSAGE signals accordingly if no data transmission occurs at the PROFIBUS-DP for a period longer than the time set. If not connected, the input is pre-initialised with 1000 ms.

Monitoring is off at t_TIMEOUT_PB_DP = TIME#0ms.

Input t_TIMEOUT_CYCLIC:

Input t_TIMEOUT_CYCLIC may be used to specify a timeout period for process data communication. Output b_ERR indicates an error if no process data communication occurs between PROFIBUS-DP master and Ωmega Drive-Line II (as PROFIBUS-DP slave) for longer than the set period of time (refer table 9).

If not connected, the input t_TIMEOUT_CYCLIC is pre-initialised with 1000 ms.

Monitoring is off at t_TIMEOUT_CYCLIC = TIME#0ms. Monitoring in this case covers only the process data communication between PROFIBUS-DP master and Ωmega Drive-Line II (as PROFIBUS-DP slave), not the communication on the PROFIBUS-DP.

Input t_TIMEOUT_SD_DATA:

Input t_TIMEOUT_SD_DATA may be used to specify a timeout period for service data communication. Output b_ERR indicates an error if a parameter command is started and not completed within the set period of time (refer table 9).

If not connected, the input is pre-initialised with 5000 ms.

Monitoring is off at t_TIMEOUT_SD_DATA = TIME#0ms. Monitoring in this case covers only the service data communication between PROFIBUS-DP master and Ωmega Drive-Line II (as PROFIBUS-DP slave), not the communication on the PROFIBUS-DP.

Input si_NR_WR_VALUE:

The number of required setpoints to be transmitted via the PROFIBUS-DP, must be specified at input si_NR_WR_VALUE. A setpoint requires two words on the bus.

Input si_NR_RD_VALUE:

The number of required actual values must be specified at input si_NR_RD_VALUE.

Example:

si_NR_WR_VALUE := SINT#2

si_NR_RD_VALUE := SINT#2

→ PZD area ranges from PZD1 to PZD6 (refer table 2)

Inputs w_STATUSWORD1 / w_STATUSWORD2:

The status words to be transmitted to the PROFIBUS-DP master are specified at inputs w_STATUSWORD1 and w_STATUSWORD2.

Inputs d_RD_VALUE0 to d_RD_VALUE7:

The actual values to be transmitted to the PROFIBUS-DP master are specified at inputs d_RD_VALUE0 to d_RD_VALUE7.

Input / output _BAPS_SD_DATA:

A BAPS_BMSTRUCT data-type global variable must be linked to _BAPS_SD_DATA.

Example:

```
_BAPS_SD_DATEN : BAPS_BMSTRUCT;
```

where:

`_BAPS_SD_DATEN`	the name of the variable with data type identifier '_' for struct
`BAPS_BMSTRUCT`	the data type.

This variable is linked to the input _BAPS_SD_DATA of function block BAPS_SD_CONTROL (from the library SYSTEM1_DLII_20bd00 or higher).

Input / output _BASE :

A PB_CTRL_BMSTRUCT data-type global variable must be linked to _BASE.

This variable must be linked to the basic address of the PROFIBUS-DP slave interface via the global variable declaration.

Example:

Option board PROFIBUS-DP-Slave for Ωmega Drive-Line II

```
_PB_BASE AT %MB3.3000000 : PB_CTRL_BMSTRUCT;
```

where:

`_PB_BASE`	the name of the variable with data type identifier '_' for struct
`PB_CTRL_BMSTRUCT`	the data type
`%MB3.3000000`	the basic address of the PROFIBUS-DP slave interface on the option board PROFIBUS-DP-Slave

Input / output a_OMEGA_SD_DATA:

A PBOMEGA_BMARRAY data-type variable is linked to a_OMEGA_SD_DATA. The data type PBOMEGA_BMARRAY is a field with 1048 entries of data type DINT, with the index starting at 1000 and ending at 2047.

```
PBOMEGA_BMARRAY : ARRAY [1000..2047] OF DINT;
```

Example:

```
a_ParValues : PBOMEGA_BMARRAY;
```

where:

`a_ParValues`	the name of the variable with data type identifier 'a' for array
`PBOMEGA_BMARRAY`	the data type.

The values of the parameters for service data communication between PROFIBUS-DP master and Ωmega Drive-Line II (parameters 1000 to 2047) are then entered into the field elements `a_ParValues[1000]` to `a_ParValues[2047]`, for example, the parameter number matching the index.

Output x_OK:

TRUE at output x_OK indicates that initialisation of the PROFIBUS-DP slave interface has been completed.

Output b_MESSAGE :

The output b_MESSAGE makes available messages of the PROFIBUS-DP slave interface.

Table 1:

b_MESSAGE	Description
0	No warning, no error
1	General error
2	Field bus error
3	Setpoint / actual value configuration error
4	Diagnostic function active
5 to 127	Reserved
-1	General warning
-2	Wire breakage warning
-3	Timeout warning
-4 to - 128	Reserved

Output b_ERR :

Output b_ERR makes available error messages (description refer: Error evaluation). b_ERR = 16#00 on no errors.

Output si_INIT_NR_WR_VALUE :

The output si_INIT_NR_WR_VALUE makes available the number of initialised setpoints.

Output si_INIT_NR_RD_VALUE :

The output si_INIT_NR_RD_VALUE makes available the number of initialised actual values.

Outputs w_CONTROLWORD1 / w_CONTROLWORD2:

Outputs w_CONTROLWORD1 and w_CONTROLWORD2 make the control words available.

Outputs d_WR_VALUE0 to d_WR_VALUE7:

Outputs d_WR_VALUE0 to d_WR_VALUE7 make the cyclical setpoints available.

Cyclical data transmission

PZDarea (Process data area):

Table 2:

PZD 1	**PZD 2**	**PZD 3**	**PZD 4**	**PZD 5**	**PZD 6**	**PZD 7**	**PZD 8**	**...**	**PZD 17**	**PZD 18**
0000	0000	0000	0000	0000	0000	0000	0000	...	0000	0000
w_STATUS-WORD1	w_STATUS-WORD2	d_RD_VALUE0		d_RD_VALUE1		d_RD_VALUE2		...	d_RD_VALUE7	
w_CONTROL-WORD1	w_CONTROL-WORD2	d_WR_VALUE0		d_WR_VALUE1		d_WR_VALUE2		...	d_WR_VALUE7	

Service data transmission

PKWarea (Parameter Code Value):

Table 3:

PKE Parameter code	**IND** **IND**ex	**PWE 1** Parameter value HighWord	**PWE 2** Parameter value LowWord
0000	0000	0000	0000
Response service data			
Command service data			

Table 4:

PKE **P**arameter code		
AK Command / response code	**R** **R**eserved	**PNU** **P**arameter**NU**mber
15 14 13 12	11	10 9 8 7 6 5 4 3 2 1 0

Service data transmission with Ωmega Drive-Line II :

The function block PB_COMM_SLAVE outputs (parameter write) and / or expects (parameter read) the parameter variables for parameters 1000 to 2047 at the input / output a_OMEGA_SD_DATA.

These parameters can be approached by the PROFIBUS-DP master if the parameter number PNU is greater than or equal 1000. Maximum PNU = 2047.

Read and write access is possible in double word format only (PWE 1: High-word, PWE 2: Low-word).

Command codes for service data transmission with Ωmega Drive-Line II

Table 5:

AK Command / response code				
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Definition**
0	0	0	0	No command
0	0	0	1	Read parameter
0	0	1	1	Write parameter (DOUBLE WORD)

Response codes for service data transmission with Ωmega Drive-Line II

Table 6:

AK Command **/** response code				
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Definition**
0	0	0	0	No command
0	0	1	0	Transmit parameter (DOUBLE WORD)
0	1	1	1	Cannot execute command

Example: a_OMEGA_SD_DATA

AK = 0011 (write double word)

PNU = 1000

PWE 1 = 0x1234

PWE 2 = 0x5678

→ a_OMEGA_SD_DATA [1000] = 0x12345678

AK = 0001 (read parameter)

PNU = 1000

→ PWE 1 = 0x1234

PWE 2 = 0x5678

Service data transmission with the controller:

A BAPS_BMSTRUCT data-type variable must be linked to input _BAPS_SD_DATA to allow service data transmission with the controller. The same variable must be linked to the input _BAPS_SD_DATA of function block BAPS_SD_CONTROL (from the library SYSTEM1_DLII_20bd00 or higher). This function block must also be requested in the project. Access to controller parameters is via specification of the parameter number as PNU. Access to option board parameters is possible, but has no impact on cyclical setpoint and actual value transmission. Setpoints are always available at outputs d_WR_VALUE0 to d_WR_VALUE7, actual values must be applied to inputs d_RD_VALUE0 bis d_RD_VALUE7. The user must establish process data communication between V controller and function block PB_COMM_SLAVE (with the function block BAPS_PD_COMM2, BAPS_PD_COMM24 or BAPS_PD_COMM8).

Command codes for service data transmission with the controller

Table 7:

AK Command **/** response code				
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Definition**
0	0	0	0	No command
0	0	0	1	Read parameter
0	0	1	0	Write parameter (WORD)
0	1	0	0	Read parameter description

Response codes for service data transmission with the controller

Table 8:

AK Command **/** response code				
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Definition**
0	0	0	0	No response
0	0	0	1	Transmit parameter (WORD)
0	0	1	0	Transmit parameter (DOUBLE WORD)
0	1	1	1	Transmit parameter description
0	1	1	1	Cannot execute command

Example: P003 = 0x0064 (controller parameter 3)

AK = 0001 (read parameter)

PNU = 3

→ PWE 1 = 0x0000

PWE 2 = 0x0064

Error evaluation:

Table 9:

b_ERR	**Description**
Bit 0	Wrong TYPE (function block not processing data)
Bit 1	No access to option board (function block not processing data)
Bit 2	Reserved
Bit 3	Service data access on option board timeout (function block not processing service data)
Bit 4	Service data access on controller timeout (function block not processing service data)
Bit 5	Setpoint number error (function block not processing cyclical data)
Bit 6	Actual value number error (function block not processing cyclical data)
Bit 7	Cyclical data transmission timeout (function block not processing cyclical data)

10.2 BYTES_TO_DWORD

The function BYTES_TO_DWORD generates a double word from 4 bytes.

Parameter input	Data type	Description
b_X_LOW_WORD_LOW_BYTE	BYTE	Byte 0 (bits 0 to 7)
b_X_LOW_WORD_HIGH_BYTE	BYTE	Byte 1 (bits 8 to 15)
b_X_HIGH_WORD_LOW_BYTE	BYTE	Byte 2 (bits 16 to 23)
b_X_HIGH_WORD_HIGH_BYTE	BYTE	Byte 3 (bits 24 to 31)

Parameter output	Data type	Description
BYTES_TO_DWORD	DWORD	Output value

Description:

The function BYTES_TO_DWORD generates a double-word-format value from the values (data type BYTE) applied to the inputs,. where

- b_X_LOW_WORD_LOW_BYTE = Bits 0 to 7 of the output value,
- b_X_LOW_WORD_HIGH_BYTE = Bits 8 to 15 of the output value,
- b_X_HIGH_WORD_LOW_BYTE = Bits 16 to 23 of the output value and
- b_X_HIGH_WORD_HIGH_BYTE = Bits 24 to 31 of the output value.

10.3 BYTES_TO_WORD

The function BYTES_TO_WORD generates a word from 2 bytes.

Parameter input	Data type	Description
b_X_LOW_BYTE	BYTE	Byte 0 (bits 0 to 7)
b_X_HIGH_BYTE	BYTE	Byte 1 (bits 8 to 15)

Parameter output	Data type	Description
BYTES_TO_WORD	WORD	Output value

Description:

The function BYTES_TO_WORD generates a word-format value from the values (data type BYTE) applied to the inputs, where

- b_X_LOW_BYTE = Bits 0 to 7 of the output value and
- b_X_HIGH_BYTE = Bits 8 to 15 of the output value.

10.4 DWORD_TO_BYTES

The function block DWORD_TO_BYTES divides the contents of a double word across four bytes before output.

Parameter input	Data type	Description
d_X	DWORD	Input value

Parameter output	Data type	Description
b_Y_LOW_WORD_LOW_BYTE	BYTE	Byte 0 (bits 0 to 7) of d_X
b_Y_LOW_WORD_HIGH_BYTE	BYTE	Byte 1 (bits 8 to 15) of d_X
b_Y_HIGH_WORD_LOW_BYTE	BYTE	Byte 2 (bits 16 to 23) of d_X
b_Y_HIGH_WORD_HIGH_BYTE	BYTE	Byte 3 (bits 24 to 31) of d_X

Description:

The function block WORD_TO_BYTES generates four bytes from the value (data type DWORD) applied to input d_X, where:

- b_Y_LOW_WORD_LOW_BYTE = Bits 0 to 7 of d_X,
- b_Y_LOW_WORD_HIGH_BYTE = Bits 8 to 15 of d_X,
- b_Y_HIGH_WORD_LOW_BYTE = Bits 16 to 23 of d_X and
- b_Y_HIGH_WORD_HIGH_BYTE = Bits 24 to 32 of d_X.

10.5 WORD_TO_BYTES

The function block WORD_TO_BYTES outputs the contents of a word in two bytes.

Parameter input	Data type	Description
w_X	WORD	Input value

Parameter output	Data type	Description
b_Y_LOW_BYTE	BYTE	Byte 0 (bits 0 to 7) of w_X
b_Y_HIGH_BYTE	BYTE	Byte 1 (bits 8 to 15) of w_X

Description:

The function block WORD_TO_BYTES generates the Low byte and the High byte from the value (data type WORD) applied to input w_X, where

- b_Y_LOW_BYTE = Bits 0 to 7 of w_X and
- b_Y_HIGH_BYTE = Bits 8 to 15 of w_X.

11 INDEX

W