SIEMENS

Universal Serial Interface Protocol USS® Protocol

Specification

Ausgabe / Edition 09.94

Bestell-Nr. / Order-No. E20125-D0001-S302-A1-7600

Authors:

Walter Möller-Nehring, Siemens AG, ASI 1 D SP, Erlangen
Wolfgang Bohrer, Siemens AG, ASI 1 D SP, Erlangen

We have checked the contents of this document to ensure that they coincide with the described hardware and software. However, differences cannot be completely excluded, so that we do not accept and guarantee for complete conformance. However, the information in this document is regularly checked and necessary corrections will be included in subsequent editions. We are grateful for any recommendations for improvement.

Contents

Section A: Protocol specification

Section B: Physical interface and bus structure

Section C: Defining the net data for drive applications

Appendix

Editions

Edition	Order No.	Date	Status
1st Edition, Section A	E31930 - T9011 - X - A1	January '92	Published
1st Edition, English Section A	E31930 - T9011 - X - A1-7600	January '92	Published
1st Edition Applications	E31930-T9012-X-A1	April '92	Published
Errors, to the 1st Edition Applications	E31930-T9012-X-A2	October '92	Published
Function expansion for SIMOVERT Master Drive and general update. Not published as application	E31930-T9012-X-AXX	August '93	Preliminary
Summary of the documents E31930-T9011 and E31930-T9012	(d) E20125-D001-S302-A1 (e) E20125-D001-S302-A1-7600	September '94	New Edition

Brief description of the changes:

August 1993:
- Specification for SIMOVERT Master Drives.
- The restriction regarding a fixed telegram length for task telegrams from the master to the slave has been withdrawn. When a variable telegram length has been parameterized, this is possible in both data transfer directions (Master $\rightarrow$ Slave $\rightarrow$ Master).
- Expansion in the index word (IND) in the high byte for text transfer for SIMOVERT FC/VC/SC
- The text character sequence has been changed for data transfer via the bus.

September 1994:
- Section C has been updated: Point 3: Parameterization
- New in Section C: Point 6: Useful data transfer format
- Supplement, Section B: Physical interface and bus structure
- The master can now be located at any position on the bus.
- Recommendation for display parameters for BUS- and interface diagnostics.
- Assignment of the most significant bit in the address byte for special telegrams
- Broadcast definition, Section C
- Bits 11-15 in the index have been defined, converter-specific
- Section B: Designations (A) and (B) have been replaced by RS485P and RS485N

Note

These instructions do not purport to cover all details or variations in equipment, nor to provide for every possible contingency to be met in connection with installation, operation or maintenance.
Should further information be desired or should particular problems arise which are not covered sufficiently for the purchaser's purposes, the matter should be referred to the local Siemens sales office.

The contents of this Instruction Manual shall not become part of or modify any prior or existing agreement, committment or relationship. The sales contract contains the entire obligation of Siemens. The warranty contained in the contract between the parties is the sole warranty of Siemens. Any statements contained herein do not create new warranties or modify the existing warranty.

Literature

/1/: PROFIBUS profile: Variable-speed drives
VDI / VDE Guideline 3689
Draft, January 1993

/2/: RS485 Recommended Standard
EIA 485: STANDARD FOR ELECTRICAL CHARACTERISTICS
OF GENERATOR AND RECEIVER FOR USE IN
BALANCED DIGITAL MULTIPOINT SYSTEMS
EIA Standard April 1983

Files: All documents have been generated using Winword 2.0b and Designer 3.1 PLUS OLE.

Deckblat.doc	Titelseite ohne Bild
Ussbild.drw	Designer 3.1 Zeichnung für Titelseite
Vorspann.doc	Dieses Dokument: Copyright, Inhaltsverzeichnis, Änderungsstand, Literatur
KapitelA.doc	Kapitel A: Protokollspezifikation
KapitelB.doc	Kapitel B: Physik
KapitelC.doc	Kapitel C: Festlegung der Nutzdaten
Anhang.doc	Anhang
Ruecken.doc	Letzte Seite
Tabellen.doc	3 Tabellen im Querformat für Kapitel C
Brodcast.drw	Designer 3.1 Zeichnung für Anhang Thema Broadcast

A: Protocol specification

1. Introduction

The USS® protocol (Universal Serial Interface Protocol) defines an access technique according to the master-slave principle for communications via a serial bus. This also includes, as sub-quantity, the point-to-point connection.

Essential features of the USS® protocol are:
- It supports a multi-point-capable coupling, e.g. EIA RS 485 hardware
- Master-slave access technique
- Single master system
- Max. 32 nodes (max. 31 slaves)
- Simple, reliable telegram frames
- Easy to implement
- Operation with either variable or fixed telegram lengths.

One master and a maximum of 31 slaves can be connected to the bus. The individual slaves are selected by the master via an address character in the telegram. A slave itself can never transmit without first being requested to do so, and direct message transfer between the individual slaves is not possible. Communications is realized in the half-duplex mode.

The master function cannot be transferred to another node (single-master system).

A bus configuration for a drive application is illustrated in the following diagram.

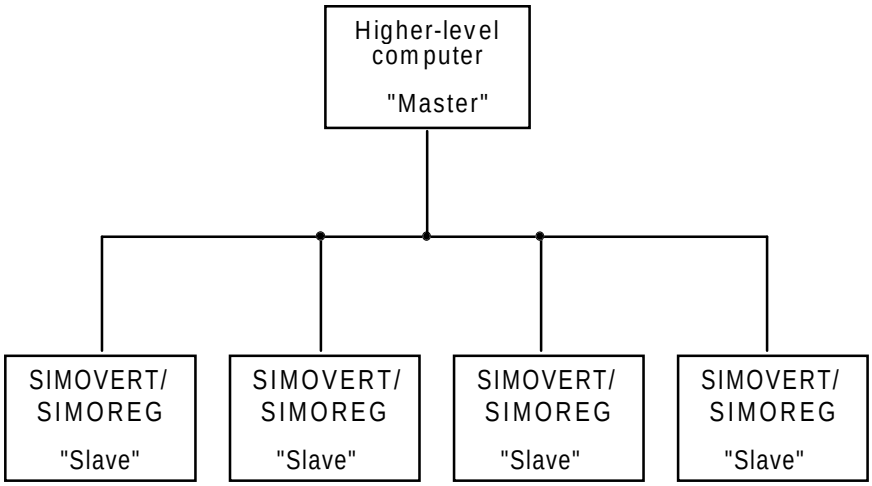


Fig. 1.1: Serial coupling between SIMOREG- / SIMOVERT drive converters (slaves) with a higher-level computer as master.

The electrical and mechanical interface characteristics (hardware) are not part of the protocol specification. The definitions and recommendations regarding the data transfer technology, the data transfer technique and bus configuration are described in Section B: ”Physical interface and bus configuration” of this specification.

2. Telegram transfer

Generally, a differentiation can be made between cyclic and acyclic telegram transfer. In drive technology, only cyclic telegram transfer is used. The master station is responsible for cyclic telegram transfer, whereby all slave nodes are addressed, one after the other, in identical time intervals.

2.1. Cyclic telegram transfer

Drive technology requires defined response times for the open-loop and closed-loop control tasks, and thus a rigidly cyclic telegram transfer:

The master continually transmits telegrams (task telegrams) to the slaves and waits for a response telegram from the addressed slave.

A slave must send a response telegram, if
- it received a telegram, error-free and
- it was addressed in this telegram.

A slave cannot send, if these conditions are not fulfilled or the slave was addressed in the broadcast mode.

For the master, a connection is established to the appropriate slave if it receives a response telegram from the slave after a defined processing time (response delay time).

In cyclic telegram transfer, the slave nodes must monitor telegram transfer for failure.

Some of the net data, which are included in the cyclic telegram, are provided for service and diagnostics. The net data transfer for data technology is described in Section C: ”Defining the net data for drive applications”.

2.2. Acyclic telegram transfer

Generally, telegram transfer is cyclic.

Cyclic and acyclic telegram transfer cannot be used simultaneously.

Service and diagnostic tasks can also be run in acyclic operation.
In acyclic operation, the master sends telegrams to the slaves at irregular intervals. The slave responds to the conditions defined for cyclic operation.

For acyclic telegram transfer, slaves cannot monitor for telegram failure.

3. Broadcast

In the broadcast mode, the master transmits a telegram to all slaves on the bus. In this case, the "Broadcast bit" in the task telegram, in the address byte, is set to logical 1 (refer to Section 4.3, assigning the address byte (ADR)). The address bits are ineffective. The individual slaves may not transmit a response telegram after receiving a broadcast telegram.

The use of a broadcast telegram requires other definitions at the application level (common telegram length, assignment of the net data contents to nodes, etc.). For a definition of the net data contents for broadcast, refer to Section C, Point 5: Process data - Broadcast.

4. Telegram structure

Each telegram (Fig. 4.1) starts with the STX start character (= 02 hex), followed by the length specification (LGE) and the address byte (ADR). The net characters then follow. The telegram is terminated by the BCC (block check character).

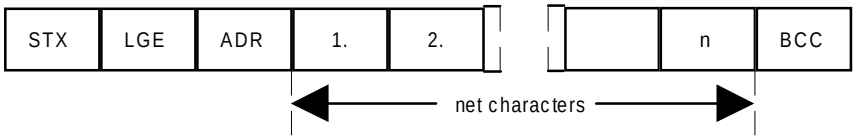


Fig. 4.1: Telegram structure

For word information (16 bit) in the net data block, the first byte is always the high byte (first character) and then the low byte (second character). The corresponding is valid for double words: First, the high word is sent, followed by the low word.

The identification of tasks in the net characters is not part of the protocol. The contents of the net data for drive converters is handled in Section C.

4.1. Data coding

The information is coded as follows:

- STX (start of text): ASCII characters: 02 hex
- LGE (telegram length): 1 byte, includes the telegram length as binary number, refer to Section 4.2
- ADR (address byte): 1 byte contains the slave address and the telegram type, binary coded; refer to Section 4.3
- Net characters: Each one byte, the contents are dependent on the task
- BCC: 1 byte block check character, generation, refer to Section 4.4

4.2. Telegram length (LGE)

The telegram length is variable.

The telegram length is specified in the 2nd telegram byte.

Depending on the configuration, fixed telegram lengths can be defined.

For fixed telegram lengths, different telegram lengths can be used for each slave node on the bus.

The maximum total length of a telegram is 256 bytes.

The actual length of the total telegram is two characters longer than LGE, as the first two characters (STX and LGE) are not counted.

Only the net characters (quantity n), address byte (ADR) and the block check character (BCC) are included in the telegram length. Thus, the telegram length is obtained from:

LGE = n + 2. { $1 \leq LGE \leq 254$ }

A maximum of n = 252 net characters (252 net data bytes) can be transferred per telegram.

4.2.1. Variable telegram length

For variable length telegrams, the number of net characters is dependent on the particular task (master → slave).

4.2.2. Fixed telegram length

For telegram transfer with a previously defined fixed length, the number net characters within a telegram is fixed, e.g. 6-word telegram, i.e. 12 net characters.

The protocol must be limited to a fixed length at the user level when configuring the bus system. (also refer to Section C)

Different telegram lengths can be defined for slaves connected to a bus.

4.3. Assigning the address byte (ADR)

In addition to the node numbers, additional information is coded in the address byte:
The individual bits in the address byte are assigned as illustrated.

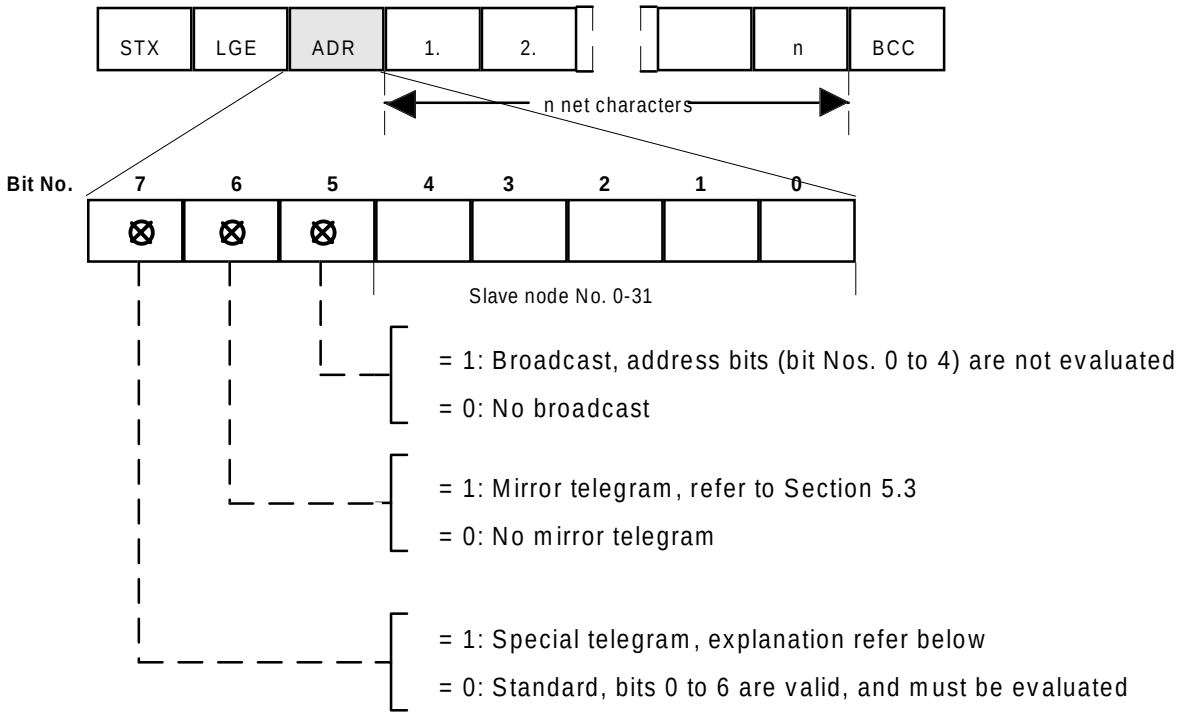


Fig. 4.2: Assignment of the address bytes (ADR)

Bit 7	Bit 6	Bit 5	Significance
0	0	0	Standard data transfer for converters. Node numbers (bits 0 to 4 are evaluated)
0	1	0	Mirror telegram: The node number is evaluated and the addressed slave returns the telegram, unchanged, to the master (refer to 5.3 Mirror telegram)
0	0	1	Broadcast: The node number is not evaluated. (refer under 3. Broadcast)
1	x	x	Special telegram: The telegram is rejected by all slaves, where no special telegram is defined. It is not permissible that the telegram is evaluated (refer to 5.4 Special telegram)

Table 4.1: Truth table of the possible combinations of bits 5, 6 and 7 in the address byte (ADR)

It is not permissible that masters transmit non-defined combinations and that the slaves respond to these.

4.4. BCC generation

The following example shows how the BCC is generated:

BCC = 0, before the first character of a telegram is received (STX)

BCC	0 0 0 0 0 0 0 0

After the first character has been received: $BCC_{new} = BCC_{old}$ EXOR "first character"
(EXOR = exclusive OR logic operation)

BCC_{old}	=	0 0 0 0 0 0 0 0
	EXOR	
"first character"	=	0 0 0 0 0 0 1 0 ($\hat{=}$ STX)
BCC_{new}	=	0 0 0 0 0 0 1 0

After each additional character has been received, this is EXOR'd with BCC_{old} EXOR, in order to regenerate BCC_{new}, e. g.:

BCC_{old}	=	0 0 0 0 0 0 1 0
	EXOR	
"second character"	=	1 1 0 1 0 1 1 0
BCC_{new}	=	1 1 0 1 0 1 0 0

The result is the BCC after the last net character.

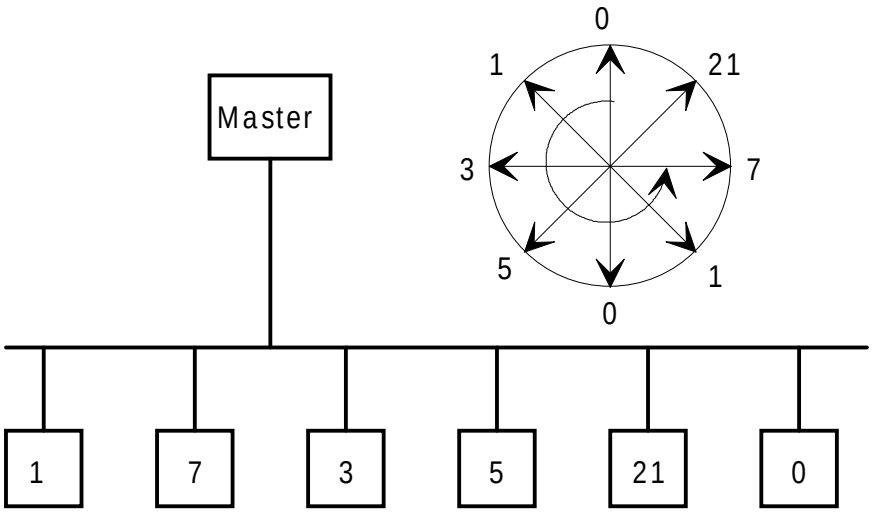
5. Data transfer procedure

The master implements cyclic telegram transfer. The master consecutively addresses all slave nodes using a task telegram. The addressed nodes return a response telegram. In accordance with the master-slave procedure, the slave must send the response telegram to the master after having received a specific task telegram, before the master addresses the next slave.

5.1. Data transfer handling

The sequence in which the addressed slave nodes can be specified, for example, by entering the node numbers (ADR) in a circulating list. If some slaves must be addressed in a faster cycle than others, then a node number can occur several times in the circulating list. A **point-to-point connection** can be implemented via the circulating list; in this case, only one node is entered in the circulating list.

Example of a configuration



Example for the circulating list

Nodes 0 and 1, are addressed twice as frequently as the other nodes.

Fig. 5.1: Circulating list

5.1.1. Cycle time

Cycle time is defined by the time for data transfer with the individual nodes.

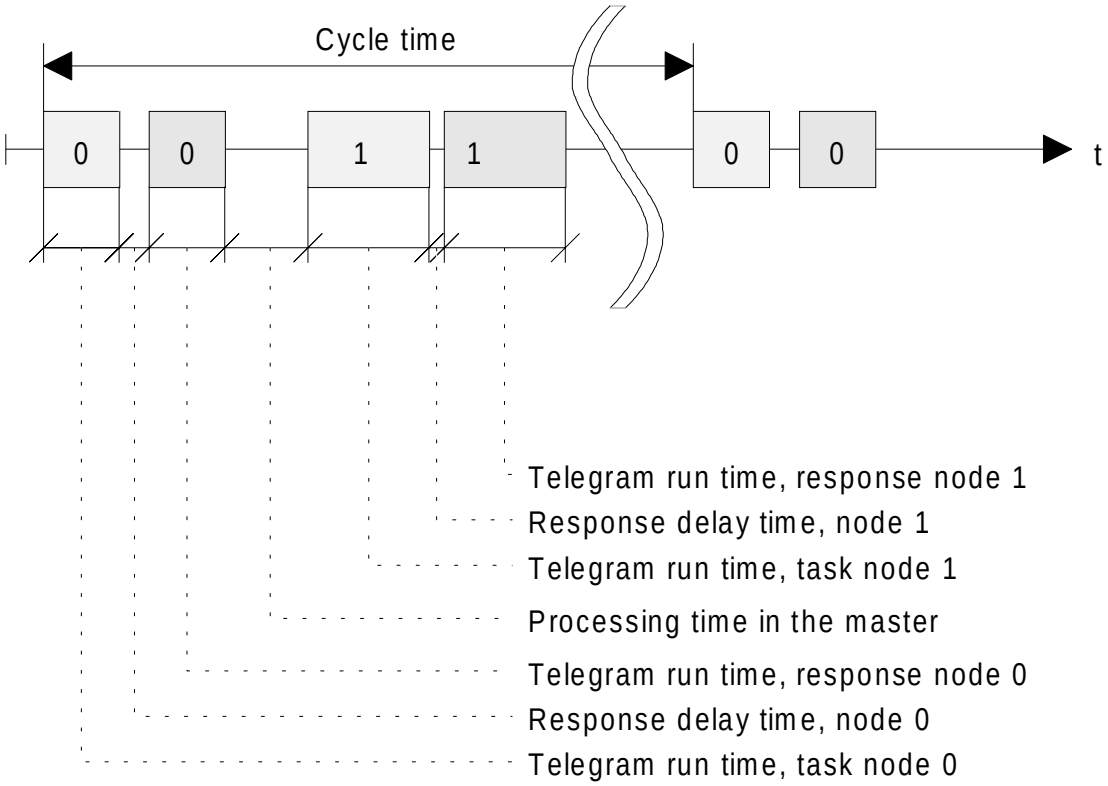


Fig. 5.2: Cycle time

The cycle time cannot be precisely defined, due to the inconsistent response delay- and processing times. A master can implement a fixed cycle time, by determining a maximum cycle time for one configuration, and then defining this as absolute cycle time. After data has been transferred with the last node, a master must wait, until the defined cycle time has expired before he can start to address the nodes again.

5.1.2. Start interval

The start character STX (=02 hex) is, by itself, not sufficient for the slaves to clearly identify the start of a telegram, as the bit combination 02/hex can also occur in the net characters. Thus, before STX, a character-less **start delay** of at least **2 character run times** (refer to Section 6) is specified for the master. The start interval or delay is part of the task telegram.

Only an STX with preceding start interval identifies a valid telegram start.

Baud rate in bit/s	Start interval in ms
9600	2.30 ms
19200	1.15 ms
38400	0.58 ms
187500	0.12 ms

Table 5.1: Minimum start interval for various baud rates

Data transfer is always realized in the schematic illustrated in Fig. 5.3 (half-duplex operation).

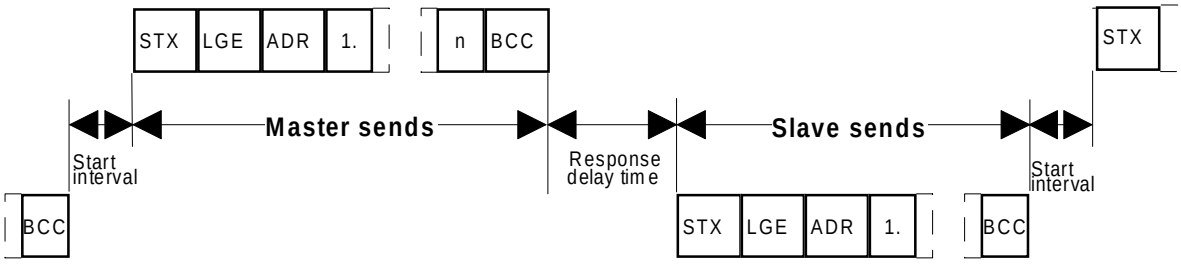


Fig. 5.3: Send sequence

5.2. Monitoring mechanisms and error responses

When a telegram is received, the correct telegram start must first be identified (start interval + STX), and then the length is evaluated (LGE). The telegram is rejected, if the length information does not correspond to the selected value for a fixed telegram length or if it is invalid for a variable telegram length.

Times have to be monitored before and during telegram reception (refer below).

The *block check character* (BCC) is generated during reception, and after the complete telegram has been read-in, compared with the received BCC. The telegram is not evaluated if these do not coincide.

If a character frame error or a parity error has not occurred in any of the received characters, the node number (ADR) of the received telegram can be evaluated.

If the address byte (ADR) does not coincide with the node number (for the slave), or the expected slave node number (for the master), then the telegram is rejected. (refer to 4.3 for the evaluation of ADR)

5.2.1. Time monitoring

The **master** must monitor the following times:
– Response delay time (slave processing time)
– Residual run time of the response telegram (refer to Section 6)

The **slave** monitors the following times:
– Start delay
– Residual run time of the task telegram (refer to Section 6)

5.2.1.1. Response delay time

The time interval between the last character of the task telegram (BCC) and the start of the response telegram (STX) is known as the **response delay time** (Fig. 5.3). The maximum permissible response delay time is **20 ms; however it may not be less than the start interval**. If node x does not respond within the maximum possible response delay time, the "node x does not transmit" error message is stored in the master. The master then transmits the telegram for the next slave node.
The "node x does not transmit" error message is only deleted after an error-free telegram from node x. Node x is not deleted from the circulating list.

5.2.1.2. Telegram residual run time

The monitoring of the telegram residual run time is dependent on the negotiated telegram length.

- Variable telegram length
  The monitoring of the max. telegram residual run time is first loaded after STX has been received, with the value, which is obtained for a telegram with 252 pieces of net data. After LGE has been received (Section 4.2), i. e. the next character, this monitoring time is loaded with the correct value.
- Fixed telegram length
  The monitoring of the maximum telegram residual run time can be directly started with the correct value (it is not necessary to evaluate LGE).

5.2.2. Processing received telegrams

Only telegrams which have been received error-free, are processed. The following receive errors are identified (this is true for both the master and slave):

- Parity errors
- Character frame errors
- Incorrect telegram length (LGE)
- Incorrect BCC
- Telegram residual run time exceeded
- Connection interrupted:
  - Slave: The master activities are not monitored at the protocol level, the user level can provide a monitoring function which can be parameterized (bus monitoring time).
  - Master: The slave does not respond within the maximum permissible response delay time.

Slaves do not send a response telegram when addressed with an incorrectly received telegram.

5.2.3. Diagnostic resources for checking the receive function

The interface software can provide information regarding communications status for communication interface diagnostics.

The diagnostic information should be able to be displayed for slaves, on the converter operator control panel (diagnostic parameters USS® interface).

The following diagnostic information is recommended:

The received telegram can then be viewed, character for character in an indexed diagnostics parameter field.

Additional, 16-bit diagnostic parameters:

1. Interface hardware version
2. Interface software release
3. Reserve
4. One 16-bit word for the error status
   Bits 15 to 8: reserved for use later
   - Bit 7: Master: Response delay time expired
     Slave: Monitoring time for cyclic telegrams, set at the user level, expired
   - Bit 6: Incorrect telegram length (LGE)
   - Bit 5: Residual telegram run time expired
   - Bit 4: Incorrect *block check character* (BCC)
   - Bit 3: Telegram start not identified (first character, no STX)
   - Bit 2: Parity error
   - Bit 1: Buffer overflow
   - Bit 0: Character frame error
5. Number of telegrams received error-free per minute
6. Number of rejected telegrams per minute
7. Counter: Error-free telegrams
8. Counter: Rejected telegrams
9. Counter: Character frame error
10. Counter: Overflow error
11. Counter: Parity error
12. Counter: Telegram start not identified
13. Counter: Telegram residual run time expired
14. Counter: BCC error
15. Counter: Incorrect telegram length
16. Counter: Monitoring time expired

Additional counters and fields can be provided.

5.3. Mirror telegram

The master can request a so-called mirror telegram from the slave.

The sequence is then as follows:
The master sends a telegram to the appropriate slave node. This telegram is different from the normal telegram in the fact that bit No. 6 of the address byte is set (=log. 1). The slave sends (mirrors) this telegram directly back to the master without making any changes. In this case, the conditions specified in Section 5.1 are valid.

Data transfer between the master and slave can be checked using the mirror telegram. This is beneficial during step-by-step start-up or troubleshooting.

5.3.1. Special telegram

The USS® protocol can also be used for special applications, which require a net data structure different from that specified in the converter profile (refer to Section C).

In order to permit mixed operation including applications with converter profile and special applications on one bus with a master, these special telegrams must be uniquely identifiable in order to be rejected by slaves with converter profile.

The structure of the special telegram frame corresponds completely to all other telegrams (STX, LGE, ADR, net data, BCC, max. net data length = 252 bytes).

The nodes with special telegram handling are addressed as defined under 4.3 (assigning address byte (ADR)). An address (bits 0 to 4) must be unique on the bus. This means, that a node with special telegram handling, can be addressed both with standard telegrams according to the converter profile (bit 7=0) as well as special application (bit 7=1).

A broadcast, which is only addressed to nodes with special telegram handling, is possible using the broadcast bit (bit 5).

Bus nodes, which cannot handle special telegrams (bit 7=1), must not be able to respond to normal telegrams (bit 7=0).

A mirror telegram is only possible with bit 7=0.

Bit 7	Bit 6	Bit 5	Significance
1	0	0	Special telegram: The telegram is received and responded to by the nodes addressed with bits 0 to 4 (nodes with special telegram processing capability).
1	0	1	Special telegram with broadcast: The node number is not evaluated. All nodes with special telegram processing capability receive the telegram, but do not send a response telegram back to the master.
0	x	x	Refer to 4.3

Table 5.2: Truth table of possible combinations of bits 5, 6 and 7 in the address byte (ADR) for special telegrams (bit 7=log. 1)

It is not permissible that non-defined combinations result in a response.

6. Definitions

6.1. Character run time

The character run time is the time required to transfer a character (11-bit character frame). This time is a function of the baud rate.

$$tz = \frac{11 \bullet 1000}{\text{transfer rate}} [\text{ms}]$$

6.2. Compressed telegram residual run time

The compressed telegram residual run time is defined as the run time which is required in order to consecutively transmit LGE, ADR, n net characters and BCC as a block (i. e. the stop bit of a character is immediately followed by the start bit of the next character).

The compressed telegram residual run time is thus obtained as follows:

(n + 3) * character run time.

6.3. Maximum telegram residual run time

The maximum telegram residual run time, includes, as illustrated in Fig. 6.1, in addition to the compressed telegram residual run time, character delay times. The sum of the character delay times is equal to 50 % of the compressed telegram residual run time.

The maximum telegram residual run time is thus obtained as follows:

1.5 * compressed telegram residual run time

corresponding to:

1.5 * (n + 3) * character run time

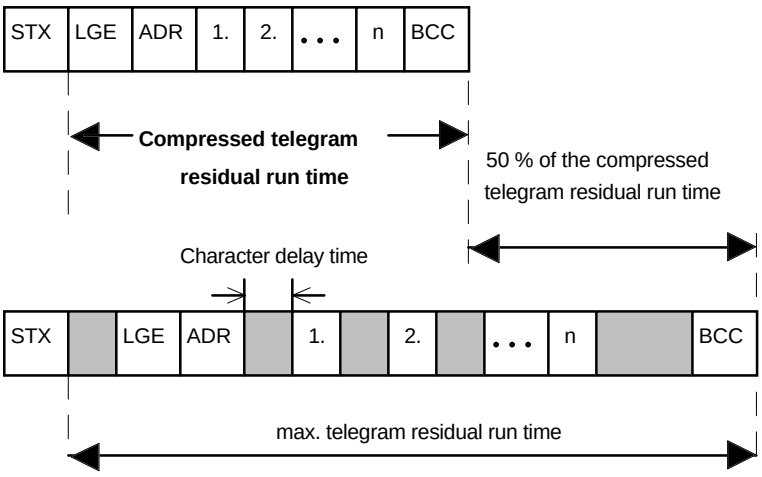


Fig. 6.1: Telegram residual run time

The delay time between two characters (= character delay time) must be less than the start delay (interval) and can be distributed as required between the characters. It is not necessary to monitor the character delay time.

B: Physical interface and bus structure

The data transfer medium and the physical bus interface are essentially defined by the bus system application.

It is possible to use various physical interfaces for the USS® protocol. However, when selecting the physical interface for a particular application, the required data transfer security and reliability should be noted.

The basis for the USS® protocol physical interface is the "recommended standard RS-485 according to /2/.

For point-to-point connections, a sub-quantity of EIA RS-232 (CCITT V.24), TTY (20 mA current loop) or fiber-optic cable can be used as physical interface.

This section describes how a USS® field bus must be structured in order to guarantee reliable data transport by the data transfer medium in standard applications. For special application conditions, other effects must be taken into account, which would require additional measures or restrictions, which are not handled in this specification.

1. Topology

The USS® bus is based on a line topology without drop lines.

Both ends of the line terminate at a node.

The maximum cable length, and thus the maximum distance between the master and the last slave is limited by the cable characteristics, ambient conditions and data transfer rate. For a data transfer rate < 100 kbit/s, a maximum length of 1200 m is possible. (EIA standard RS-422-a, December 1978, Appendix, Page 14)

The maximum number of nodes is limited to 32 (1 master, 31 slaves).

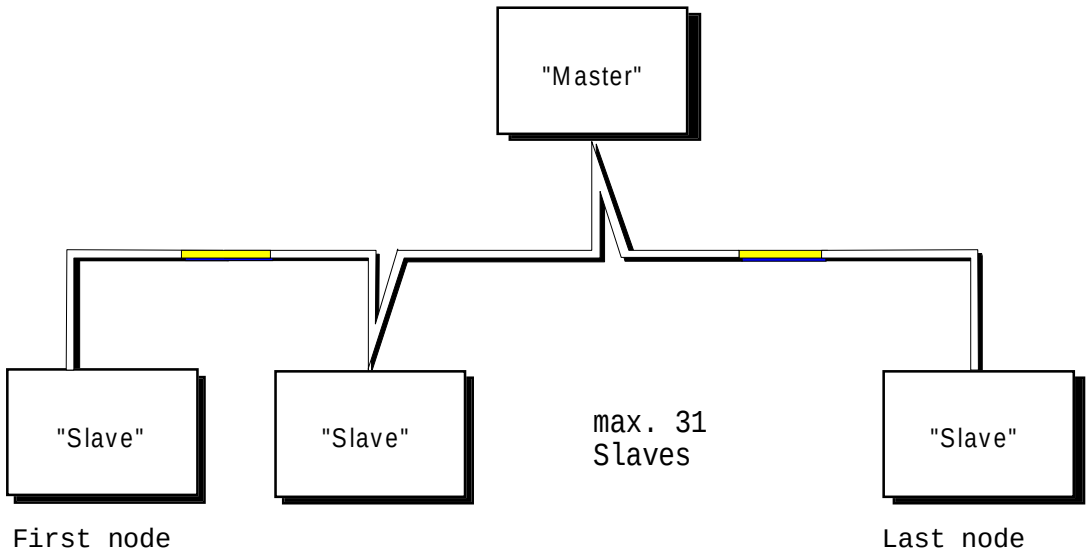


Fig. 1.1: Topology, USS® bus

Point-to-point connections are treated just like bus connections. One node has the master function and the other the slave function.

2. Data transfer technology

Data transfer is realized according to the EIA 485 Standard (/2/).

RS232 or TTY can be used for point-to-point couplings.

Data transfer is always half-duplex, i. e. sending and receiving are alternately realized and must be controlled by the software. The half-duplex technique allows the same cables to be used for both data transfer directions. This permits simple and favorably-priced bus cabling, operation in noisy environments and a high data transfer rate.

2.1. Cable characteristics

Screened, twisted two-core conductor cables are used for the bus cable.

Note:
All of this information is only a recommendation. Other measures or restrictions may be required depending on the actual requirements and application and conditions on site.

Structure:

Conductor cross-section:	2 x ≈ 0.5 mm²
Conductor:	≥ 16 x ≤ 0.2 mm
Twisting:	≥ 20 twists / m
Overall screening:	braided, tinned copper wire, diameter ≥ 1.1 mm² 85 % optical coverage
Overall diameter:	≥ 5 mm
External sheath:	depending on the requirements regarding flammability, debris remaining after burning, etc.

Thermal/electrical characteristics:

Conductor resistance (20°C):	≤ 40 Ω/km
Insulation resistance (20°C):	≥ 200 MΩ/km
Operating voltage (20°C):	≥ 300V
Test voltage (20°C):	≥ 1500V
Temperature range:	-40C ≤ T ≥ 80°C
Load capability:	≥ 5A
Capacitance:	≤ 120 pF/m

Mechanical characteristics:

Single bending:	≤ 5 x external diameter
Repeated bending:	≤ 20 x external diameter

Recommendations:

1. Standard, without any special requirements:

Two-core, flexible, screened conductor in accordance with VDE 0812 with colored PVC sheath.

PVC insulation, resistant to oil and petroleum products.
Type: LIYCY 2 x 0.5 mm²
For example: Metrofunk Kabel-Union GmbH 12111 Berlin Postfach 41 01 09
Tel 030-831 40 52, Fax: 030-792 53 43

2. Halogen-free cables (no gaseous hydrochloric acid when burning):
Halogen-free, highly flexible cables, resistant to high temperatures and cold. Sheath manufactured from an ASS special mixture with a silicon basis:
Type: ASS 1 x 2 x 0.5 mm²
From: Metrofunk Kabel-Union GmbH 12111 Berlin Postfach 41 01 09
Tel 030-831 40 52, Fax: 030-792 53 43

3. Recommended, if halogen- and silicon-free cables are required:
Type: BETAflam G-M/G-G-B1 flex 2 x 0.5 mm²
From: . Studer-Kabel-AG, CH 4658 Däniken

2.2. Cable length

The cable length is dependent on the data transmission rate and the number of connected nodes. The following cable length are possible under the cable specifications, specified in 2.1:

Data transfer rate	Max. node number	Max. cable length
9.6 kbit/s	32	1200 m
19.2 kbit/s	32	1200 m
38.4 kbit/s	32	1200 m
187.5 kbit/s	30	1000 m

Table 2.1: Cable lengths as a function of the data transmission rate

2.3. Interface characteristics

The following text discusses how the physical interface is implemented according to EIA RS 485.

The interface can be configured, either non-floating or floating with respect to the internal electronics per supply voltage.

The bus can be configured using only two-core cables as a result of the master-slave bus access technique.

This requires, that at any particular time, only one sender can access the bus. All other nodes must switch the senders to a high-ohmic state ("monitoring").

The logical 0- and 1-statuses for RS 485 technology are identified by the polarity of the voltage difference between the two data lines.

The EIA RS-485 and EIA RS-422 Standards are exclusively valid for the characteristics of the send- and receive blocks. The values taken from these standard, are characteristic values, as to how the correct signal shape can be tested on an installed bus.

A sender must generate, under test conditions and according to RS-485 Standard (the bus is terminated with 54 Ω), a specific voltage difference V_0 between the RS485P and RS485N signal lines:

Logical 1 status:	$1.5\ V \leq V_0 \leq 5\ V$	RS485P is positive with respect to RS485N
Logical 0 status:	$-5\ V \leq V_0 \leq -1.5\ V$	RS485P is negative with respect to RS485N

The rate of rise and rate of fall times, i. e. the time to change between two logical statuses must not exceed, on the bus:

0.3 / data transfer rate

If none of the nodes sends, the voltage difference is defined to be a positive level by the basis network (refer to Section 4.5, Bus termination).

Data transfer is asynchronous.

The deviation of the receive- and send clock from the nominal value is a maximum of $\pm$ 0.3%.

The telegram characters are sent and received as bit-serial UART characters.

The time between the stop bit of a character and the start bit of the next character within a telegram must be less than two character run times.

2.4. Data transfer rate

The following data transfer rates are used for the USS® interface:

300 bit / s
600 bit / s
1200 bit / s:
2400 bit / s
4800 bit / s
9600 bit / s
19200 bit / s
38400 bit / s
57600 bit / s
76800 bit / s
93750 bit / s
115200 bit / s
187500 bit / s

The data transfer rates which are highlighted are recommended for USS®, and should be implemented on all interfaces.

3. Data transfer techniques

The usual UART blocks are used for the serial asynchronous data transfer, as is generally used for digital data transfer.

3.1. Bit coding

The characters are transferred, as they are coded from the UART block. The code is designated as non-return to-zero code (NRZ code). A bit consists of a square wave pulse, whose width corresponds to the clock (1/data transfer rate).

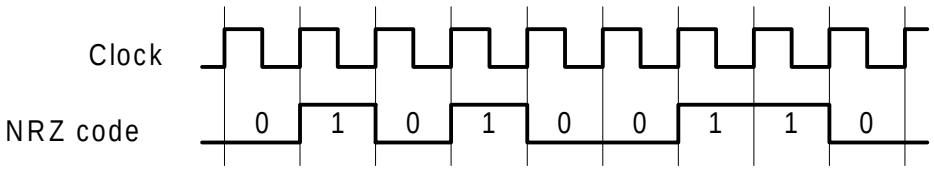


Fig. 3.1: NRZ code

3.2. Character frames

Each transferred character starts with a start bit and ends with a stop bit. 8 data bits are transferred. Each character (byte) has a parity bit (even parity, i. e. the number of logical ones in the data bits, including the parity bit, is an even number). An error message is generated if the character frame is not maintained (refer to Fig. 3.1).

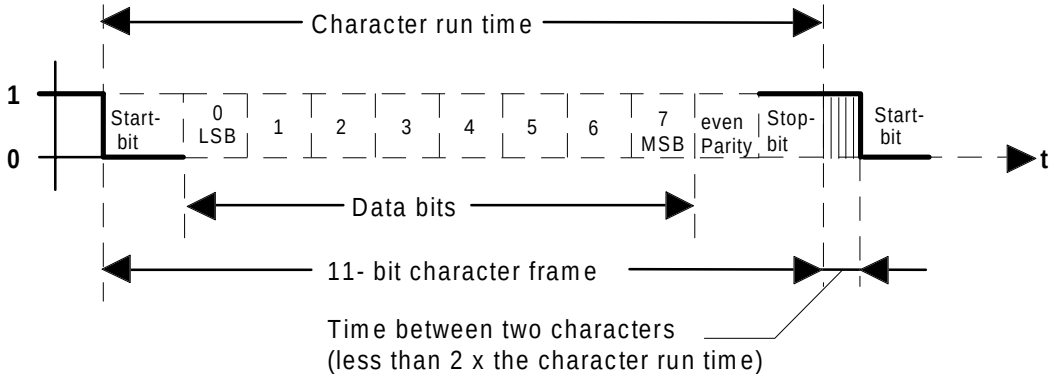


Fig. 3.2: Character frame

The start bit is always a logical 0, the 8 data bits can have any bit pattern, the parity bit is either 1 or 0 and the stop bit is always a logical 1. The signal level remains a logical 0 up to the start bit of the next character in the same telegram.

LSB is the least significant bit and MSB the most significant bit.

4. Configuration guidelines

4.1. Cable routing

When routing the USS® bus cable, it should be ensured that the system is configured in accordance with EMC guidelines. The following points should be specially observed:

- The bus cable may be routed in one bundle or cable duct with other data cables (PG, printer, SINEC L2-DP), unscreened cables for DC voltages ≤ 60 V and unscreened cables for AC voltages ≤ 25 V.
- The bus cable may not be routed together with unscreened cables for DC voltages > 60 V and ≤ 400 V together in a bundle or cable duct.
- A minimum clearance of 10 cm must be maintained between the bus cable and unscreened cables for AC voltages > 25 V and ≤ 400 V.
- A minimum clearance of 50 cm must be maintained between the USS® bus cable and data network cables (e. g. SINEC H1).
- A minimum clearance of 1m must be maintained to noise sources (transformers, contactors, motors, electrical welding equipment).
- The cables must be routed through the shortest possible path between 2 nodes.
- Potential bonding cables and signal cables should be routed as close together as possible.
- Cable extensions via terminals or connectors must be avoided.
- Cables should be routed close to grounded surfaces.

4.2. Potential bonding

If bus nodes are connected through non-floating interfaces, or if nodes are grounded at different plant sections, or if the difference between the 0 V potentials of the interface electronics ≥ 7 V, data transfer could be disturbed and the boards could be damaged.

In all other cases, potential bonding should be realized between the converter ground potentials.

Cross section of the copper potential bonding cable:

- 16 mm² for cable lengths up to 200 m
- 25 mm² for cable lengths above 200 m

The potential bonding cable should be connected, for all nodes, to ground through the largest possible surface area.

(also refer to Fig. 4.2: Screening and potential bonding)

4.3. Screening

Screening damps magnetic, electrical and electromagnetic noise fields. Noise currents are fed to ground through the screen braiding and the housing ground.

- The screens of all nodes should be connected to the housing ground/ground (grounded screen bar) through the largest surface area.
- If there is no potential bonding (only in exceptional cases), then the screen may only be connected at one end.
- The screen should be connected with the metallic connector housing.
- The connector housing in the interface electronics must not be in contact with the housing / electronics power supply ground (otherwise, do not connect the screen with the connector housing).
- It is not permissible to connect the screen through a pin (e. g. pin 1), as otherwise the noise currents would be fed to ground through the interface electronics (the interface electronics would be destroyed).

(also refer to Fig. 4.2: Screening and potential bonding)

4.4. Termination technology, connector assignments

The connector / terminal design and assignment is not specified. Each individual case should be evaluated (space requirement, accessibility etc..). However, it is recommended that, if at all possible, SINEC L2-DP termination technology is used.

Pin assignment of the bus interface with 9-pin SUB-D connector:

Socket connectors are provided on the interface side and plug connectors on the cable side.

PIN 1	-	Free
PIN 2	-	Free
PIN 3	RS485P	Receive- and transmit signal (+)
PIN 4	(-)	Reserve (for SINEC L2-DP direction signals for repeaters)
PIN 5	0 V	Data reference potential
PIN 6	5 V	Power supply voltage
PIN 7	-	Free
PIN 8	RS485N	Receive- and transmit signal (-)
PIN 9	(-)	Reserve (for SINEC L2-DP direction signals for repeaters)

Assignment of the bus interface for terminal connection:

Terminal 1	RS485P	Connection cable 1, positive signal level
Terminal 2	RS485N	Connection cable 1, negative signal level
Terminal 3	RS485P	Connection cable 2, positive signal level
Terminal 4	RS485P	Connection cable 2, negative signal level
Terminal 5	0Vext	
Terminal 6	5Vext	optional, is only required for an external basis network

Terminal 1 and terminal 2 are internally connected;
Terminal 2 and terminal 4 are internally connected

Cable 1: Cable to the previous node on the bus

Cable 2: Cable to the following node on the bus

If it is **not** permissible that the bus is interrupted if the connector is withdrawn from the interface, then core RS485P from cable 1 and 2 must be connected under terminal 1 and cores RS485N from cable 1 and 2, under terminal 2.

4.5. Bus termination

The bus cable must be terminated at both ends.

A 150 Ω resistor must be connected between data signal lines RS485P and RS485N at the first and last node.

The bus terminating resistor on the interface board can be activated with the jumper settings. The bus terminating resistor is not activated when the board is supplied.

If the bus terminating resistor cannot be mounted on the interface board, then it must be mounted in a connector housing.

Basis network

If none of the nodes transmits, then the bus is at an undefined potential, because all transmitters are switched to a high-ohmic condition. To suppress signal noise in this status, the bus has a basis network, so that a defined positive signal level is obtained.The basis network should be connected at both nodes at which the bus is terminated.

The location of the basis network resistors is the same as for the bus terminating resistor.

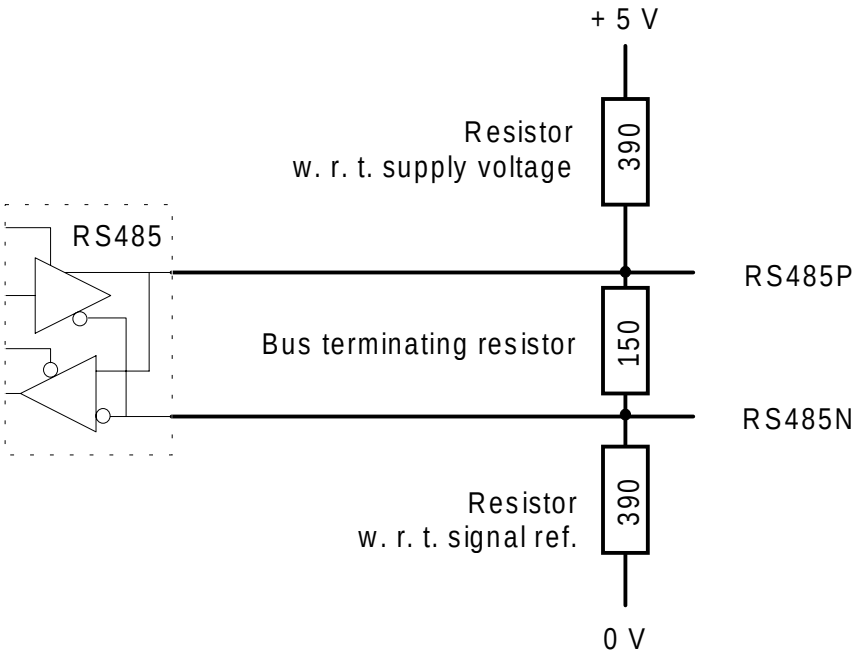


Fig. 4.1: Basis network and bus termination

390 Ω resistors are recommended if the interface supply voltage is 5 V (for 15 V, approx. 1 kΩ with respect to the supply voltage and 390 Ω with respect to 0 V).

The interface must provide the power supply voltage at a pin or terminal. The voltage source must be able to drive a 10 mA short-circuit current.

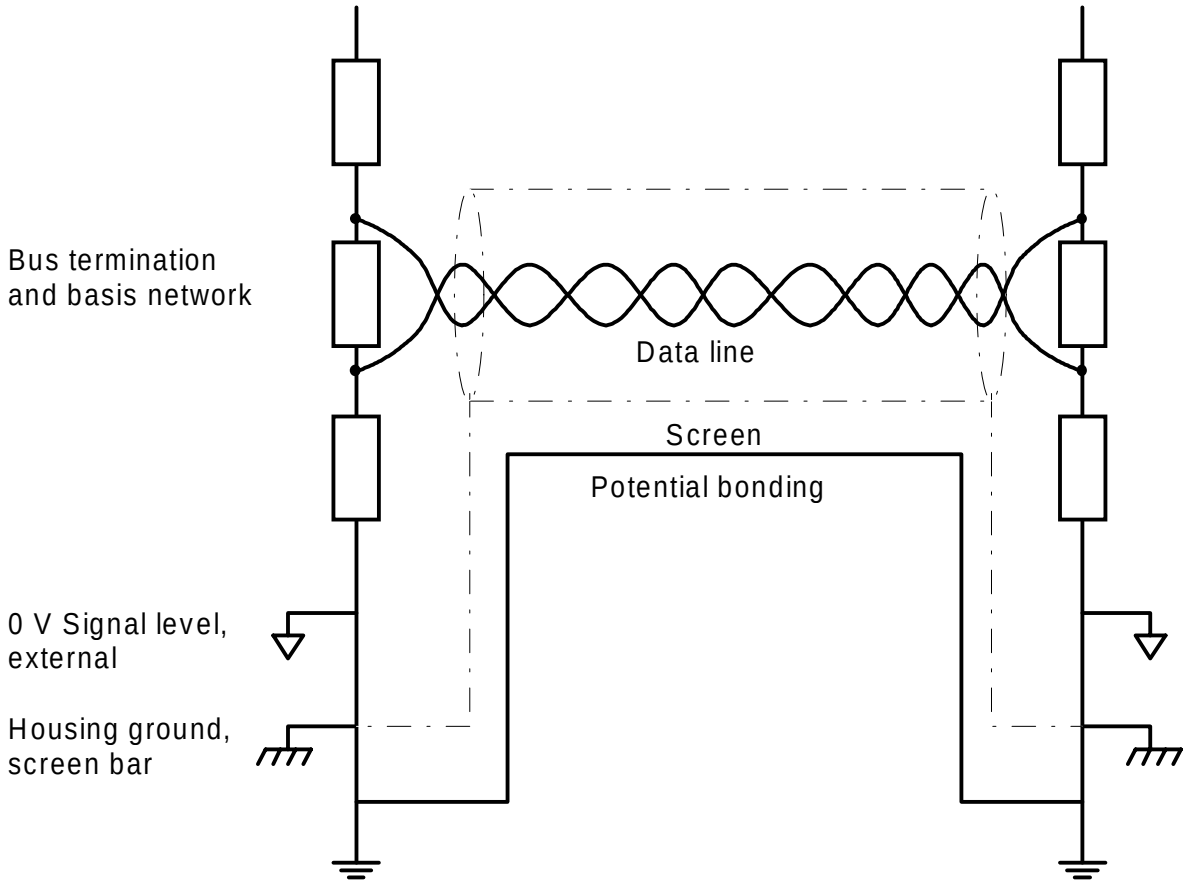
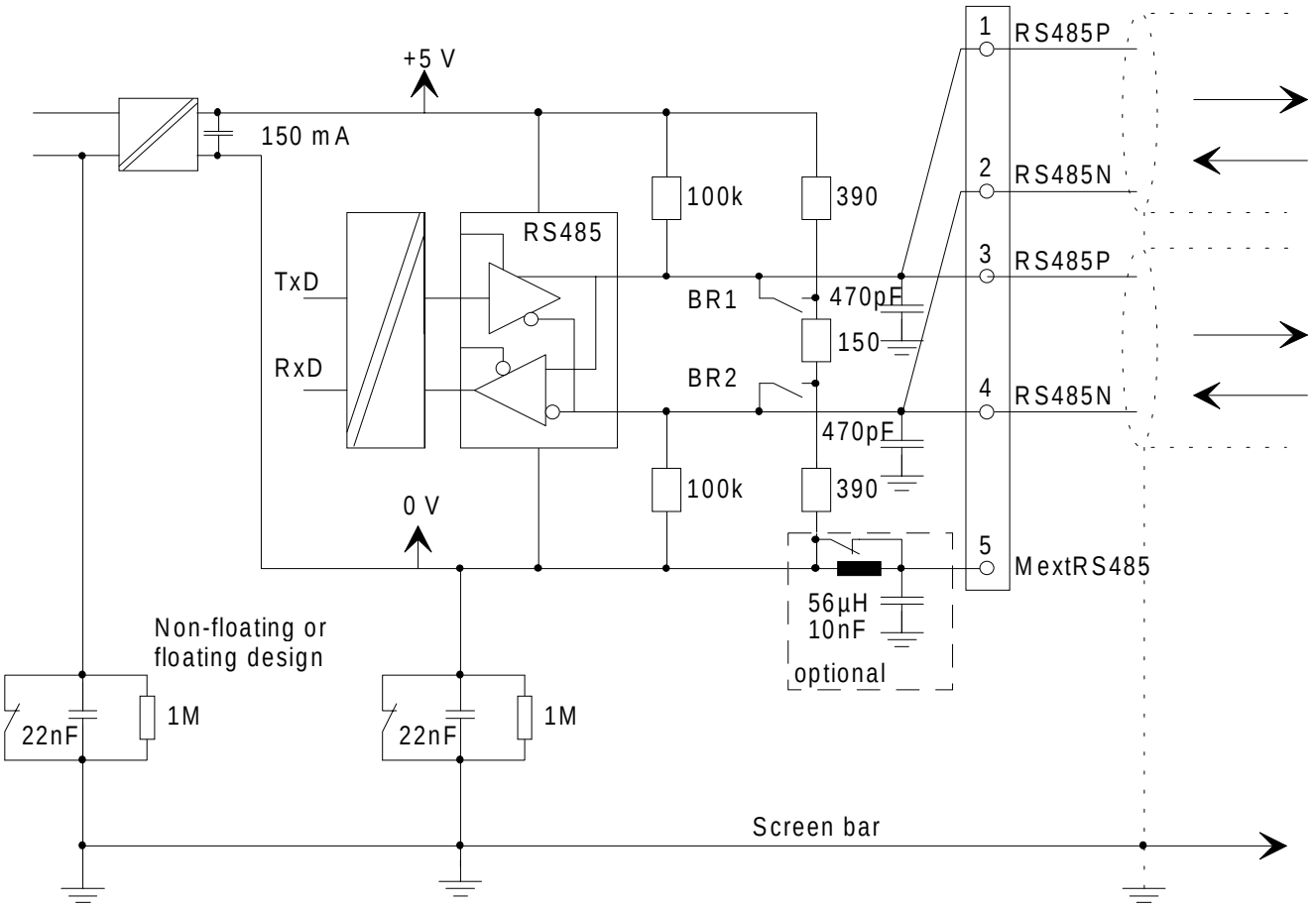


Fig. 4.2: Screening and potential bonding

4.6. Recommended circuit

A recommended circuit for a floating USS® interface is shown in the following diagram, whose connections are fed to terminals:



BR1 and BR2 are closed for bus termination; open when supplied.
Electrical isolation is optional.

Fig. 4.3: EIA RS485 interface structure, 2-wire, floating

C: Defining the net data for drive applications

1. Introduction

The USS® protocol allows the user to configure a serial bus coupling between a high-level master and several slave systems. Master systems can, for example, be PLCs or PCs. The SIMOVERT and SIMOREG drives are always slaves on the bus.

The USS® protocol allows the user to implement automation tasks requiring a cyclic (time) telegram transfer (fixed telegram length required), is well as visualization tasks. In this case, the variable telegram length protocol is advantageous, as texts and parameter descriptions can be transferred with one telegram without breaking up the information.

However, **drive automation tasks** (open-loop and closed-loop control), require cyclic telegram transfer, which can only be realized if the telegrams are a fixed length. The selected telegram length may not be changed during operation. A fixed telegram length limits the number of characters in the telegram net data block.

The subsequent section describes, in detail, the **structure of the net data blocks contained in the telegram** as well as the necessary settings (parameterization) of the interface, through which data transfer is to be realized. Communications can be realized through a basic converter interface or through a separate interface board.

The structure of the net data block in the telegram is independent of the specification of the protocol with which the net data is transferred. The structure including the contents of the net data essentially correspond to the definitions for cyclic data transfer via PROFIBUS (PROFIBUS profile "variable-speed drives" /1/). Thus, the user is guaranteed, that he can access the process data (= control / status words and setpoint / actual values) with the same access mechanisms, independent as whether this is realized using USS® or PROFIBUS-DP/FMS.

The subsequent descriptions are independent of any particular converter and are intended to provide the user with guidelines as to how the USS® protocol is to be handled. Documentation of the USS® protocol for the individual converters should be created for a specific converter application. This application should cover how the bus is to be structured and how the protocol is to be parameterized; further, it should be defined which net data contents the converter „understands“.

The specification must be observed for all implemented systems.

2. General structure of the net data block

The net data block is subdivided into two areas, the

- PKW (parameter ID value) area and the
- PZD (process data) area

- The **PKW area** refers to the handling of the parameter ID value (PKW) interface. The PKW interface does not involve a physical interface, but defines a mechanism which handles parameter transfer between two communication partners. This means, reading and writing parameter values, parameter definitions and associated texts as well as handling parameter changes using parameter change reports. All tasks, which are realized through the PKW interface, are operator control and visualization tasks, service and diagnostics.
- The **PZD area** contains the signals required for the **automation**:
  Control word (s) and setpoint (s) from the master to the slave
  Status word (s) and actual value (s) from the slave to the master.

The net data block is created from the combination of both areas. This structure is valid for the **task telegram** (master → slave) as well as for the **response telegram** (slave → master).

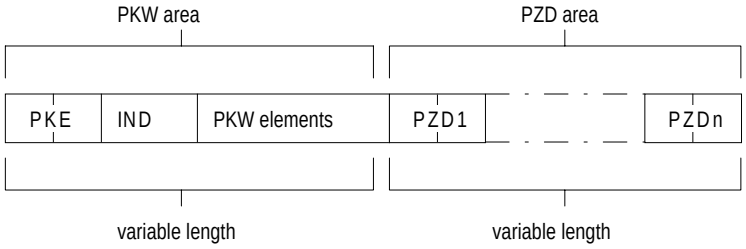


Fig. 2.1: Net data block

A **task telegram** is the transfer of a complete net data block from the master to the slave.

A **response telegram** is the transfer of the complete net data block from the slave to the master.

PKW area:

Using so-called "tasks" and "responses", in this area, access to the parameters accessible in a converter and via a USS® interface are covered.

PKE = Parameter ID (PKE); is used to identify and initiate tasks and responses for processing parameters and is always **one word** long (≙ 16 bits). The parameter number is also contained in the PKE.

IND = The index is always one word long. The significance of the index is described in detail in Section 4.2.2.

PWE element = Information, associated with a task or a response (defined in the PKWE) such as parameter values, texts or parameter description data are transferred. This area can vary in length depending on the task/response. If only PZD data are to be transferred in the net data block, the number of PKW elements can be set to 0 (PKE + index + PWE elements)!

PZD area:

Process data are **continuously** transferred between the master and slaves in this area. At the start of communications, it is configured as to which process data are exchanged with a slave. For example, the current setpoint is transferred to slave x in the second PZD (= PZD 2). This setting remains for the complete data transfer.

PZD1-PZDn = Process data (= control / status word (s) and setpoint(s) / actual value (s));
the control / status word (s), setpoints and actual values required for the automation are transferred in this area.
The length of the PZD area is defined by the number of PZD elements and their size (e. g. word, double word). Contrary to the PKW area, which can be variable, the length of this area must always be permanently agreed upon between the communication partners! The maximum number of PZD words per telegram is limited to 16 words. If only PKW data are to be transferred in the net data block, then the number of PZD can also be 0!
Depending on the data transmission direction, the control word or the status word is always transmitted in PZD1. The main setpoint or the main actual value is always transmitted in PZD2, corresponding to the data transmission direction. Additional setpoints and actual values or control / status words are sent in the following process data PZD 3 to PZD n.

The length of the individual areas must be negotiated between the communication partners, refer to Section 7. For this purpose, various parameters should be provided in the basic converter with which the PKW and PZD components for the protocol at the serial interfaces can be set.

3. Parameterization of the USS® protocol at a serial interface

Every serial interface, on which the USS® protocol is to be implemented, must have, in edition to parameters for the bus address, baud rate and telegram failure time, two parameters with which the length of the PKW- and PZD areas can be independently set. The appropriate interface can either be provided at the basic converter or on a communications board. The latter option requires an appropriate mechanism, so that the required parameterization can be made on the communications board. For example, the relevant parameters can be transferred from the basic converter to the communications board via a dual port RAM coupling.

3.1. Parameter setting for 6SE21, 6SE30 (Micro Master) SIMOVERT converters and SIMOREG K 6RA24

Baud rate

PNU: Converter-specific		
Designator: Converter-specific		
Type: 02		
Function: Baud rate		
Parameter value (PWE)		
1 ≙ 300 Baud 2 ≙ 600 Baud 3 ≙ 1200 Baud 4 ≙ 2400 Baud 5 ≙ 4800 Baud 6 ≙ 9600 Baud 7 ≙ 19200 Baud 8 ≙ 38400 Baud 9 ≙ 93750 Baud 10 ≙187500 Baud		

→ Parameter number (PNU), e. g. P783 for SIMOREG K 6RA24

→ Is used for a plain text display on the operator control panel (e. g. baud rate)

→ Data type: refer to /1/

Note: Special baud rates, beyond these are storeed at > 20!

Bus address

PNU:	Converter-specific
Designator:	Converter-specific
Type:	O2
Function:	Bus address
PWE:	0 - 31

→ e. g. BUS_ADR

Telegram failure time

In order to monitor the time taken by the master to address the slave on the basic converter, the maximum time between two valid telegrams addressed to the basic converter should be parameterized using this parameter. The time units used are "seconds" (sec). The parameter should be set to 0 if monitoring is not required.

PNU:	Converter-specific
Designator:	Converter-specific
Type:	O2
Function:	Telegram failure time in seconds for the basic converter interface.
PWE:	0 - 32 (0 : No monitoring) (1 : Factory setting)

→ e. g. TLG_AUS

Note:

The slave time monitoring only starts after its power supply has been switched-on and after the first fault-free task telegram has been received.

Number of PKW elements

The number of PKW elements in the PKW area of the net data block is defined per parameter. The specification always refers to PKW elements in the single-word format.

PNU:	Converter-specific		
Designator:	Converter-specific	→	e. g. PKW_ANZ
Type:	O2		
Function:	PKW number		
PWE:	0 → 0 words 3 → constant, 3 words 4 → constant, 4 words 127 → variable length		

Caution:

If only telegrams with constant net data quantities are to be used, 126 must not be exceeded when the number of PKW and number of PZD are added. According to the USS®-protocol specification, a maximum of 252 bytes (126 words) is permitted. If telegrams with variable PKW components are used, these parameters (PKW_ANZ) must be set to 127, independent of parameter PZD_ANZ.

Number of PZD elements

The quantity of process data contained in the net data block can be influenced using this parameter. The specification always refers to a PZD element in the single-word format.

PNU:	Converter-specific		
Designator:	Converter-specific	→	e. g. PZD_ANZ
Type:	O2		
Function:	PZD number		
PWE:	0 - 16 (words)		

3.2. Parameter setting for SIMOVERT Master Drives

Contrary to the previous parameter definitions the parameter, baud rate, bus address, PKW_ANZ and PZD _ANZ parameter, are the "array" data type with index 1 to 3 for the particular USS® protocol on the basic converter SST1, or SST2 on the SCB board (**S**erial **C**ommunications **B**oard).

PNU: P683	
Designator: SST/SCB bus address	
Type: Array, data type 02	→ Data type: Refer to /1/
Function: Bus address for USS® at the interface on the basic converter and on the SCB communications board	
Index	Parameter value (PWE)
1 ≙ SST 1 2 ≙ SCB 3 ≙ SST 2	0 to 30

PNU: P684		
Designator: SST/SCB baud rate		
Type: Array, data type 02	→ Data type: Refer to /1/	
Function: Baud rate for USS® at the interface on the basic converter and on the SCB communications board		
Index	Parameter value (PWE)	
1 ≙ SST 1 2 ≙ SCB 3 ≙ SST 2	1 ≙ 300 baud 2 ≙ 600 baud 3 ≙ 1200 baud 4 ≙ 2400 baud 5 ≙ 4800 baud 6 ≙ 9600 baud 7 ≙ 19200 baud 8 ≙ 38400 baud	Example: The 38.4 kbit/s baud rate at interface 1 of the basic converter is set to parameter value = 8 by setting parameter P684 with index = 1.

PNU: P685	
Designator: SST/SCB PKW number	
Type: Array, data type 02	→ Data type: Refer to /1/
Function: Number of words (16 bit) in the PKW area in the net data block at the interfaces on the basic converter and on the SCB communications boards	
Index	Parameter value (PWE)
1 ≙ SST 1 2 ≙ SCB 3 ≙ SST 2	0 →0 words 3 → constant, 3 words 4 → constant, 4 words 127 →variable length

PNU: P686	
Designator: SST/SCB PZD number	
Type: Array, data type 02	→ Data type: Refer to /1/
Function: Number of words (16 bit) in the PZD area of the useful data area at the interfaces of the basic converter and on the SCB communication boards	
Index	Parameter value (PWE)
1 ≙ SST 1 2 ≙ SCB 3 ≙ SST 2	0 to 16

PNU: P687	
Designator: SCom/SCB Tlg off	
Type: Array, with data type 02	→ Data type: Refer to /1/
Function: Telegram failure time for USS® at the interfaces of the basic converter and the SCB communications boards	
Index	Parameter value (PWE)
1 ≙ SST 1 2 ≙ SCB 3 ≙ SST 2	0 to 6500 [ms]

4. PKW area

4.1. Structure of the PKW area (parameter ID value)

The structure of the PKW area, is, as far as the sequence of its elements is concerned, always the same and only differ from the standard structure by the number of its parameter values (PWE).

The parameter area can be set with either a fixed length (3 words or 4 words long) or with variable length, via parameter PKW_ANZ, refer to Section 3.1.

If there is no PKW area in the net data block, **PKW_ANZ must be 0**, which means that converter **parameterization is not possible via this interface!**

4.1.1. PKW area for a fixed telegram length

Standard structure for parameter values in the single-word format (16 bit):

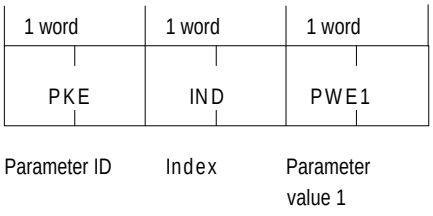


Fig. 4.1: Structure of the PKW area 3 words

If the PKW area is not set to 0, refer to the interface parameterization, then this area must be 3 words long. For a fixed telegram length, the number of words in the PKW area for both the task telegram (master to slave) as well as the response telegram (slave to master) are always constant and the same size. For parameterization of the fixed PKW area, refer to Section 3.1; in this case, PKW_ANZ must be set to 3.

Note:

It is necessary to define PKW_ANZ and PZD_ANZ for a fixed telegram length, in order to be able to parameterize a correct and optimum telegram monitoring "telegram failure time".

Standard structure for parameter values as double-word format (32 bit):

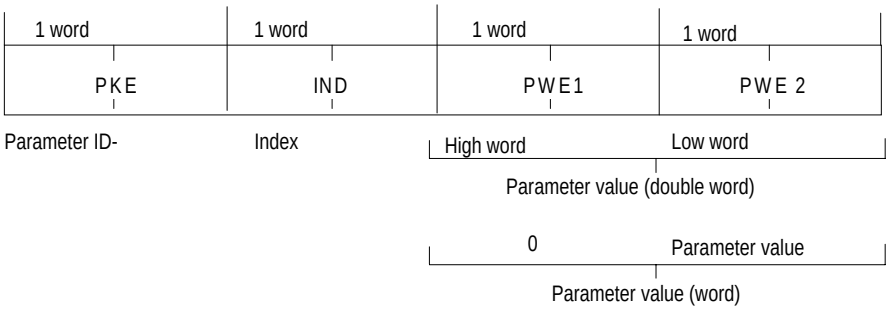


Fig. 4.2: Structure of the PKW area, 4 words

In this case, if the PKW area is not parameterized to 0, the number of words in the PKW area should be parameterized to 4. This is valid for both the task as well as for the response telegram.

4.1.2. PKW area with variable telegram length

Standard structure:

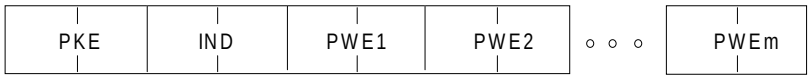


Fig. 4.3: Structure of the PKW area, variable length

with: 1 word ≤ m ≤ 108 words (maximum), if there are 16 PZD words (maximum) in the net data block.
1 word ≤ m ≤ 124 words (maximum), if there is no PZD.

Telegram transfer with variable telegram length means that the slave responds with a telegram in response to a telegram from the master; the length of the telegram from the slave no longer coincides with the length of the task telegram from the master.
The length and assignment of elements PWE 1 to PWE m in the response telegram are dependent on the task issued by the master. Parameter PKW_ANZ must be set to 127 in order that the slaves can respond appropriately to tasks which require a variable telegram length.

The master can only access the slave with a variable telegram length when parameterized for variable telegram length (PKW_ANZ = 127). Whereby, "variable" generally refers to a variable PKW area. The PZD-area length must be the same for task- and response telegrams. In this case, it must be checked as to whether the telegram failure time setting is practical.

4.2. Description of the individual PKW elements

4.2.1. Parameter ID (PKE)

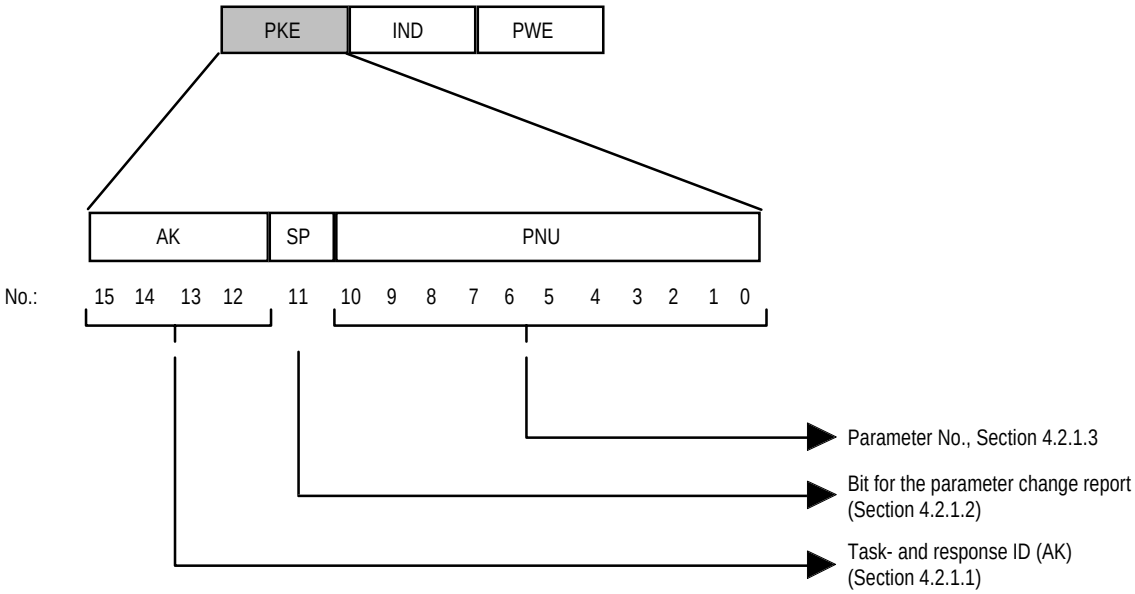


Fig. 4.4: Parameter ID structure

4.2.1.1. Task- and response ID

The tasks are issued, coded from the master to the slave in the task ID (AK). The slave processes the task and formulates the appropriate response, which is then issued in coded form (response ID) to the master. The tasks / response IDs are defined so that a task and a response are clearly defined by the PKE (AK + PNU), and for specific tasks / responses, additionally using the index word (IND).

Task ID (master → slave)

Bit No. 15	14	13	12	Function	Description
0	0	0	0	No task	No task for the PKW interface
0	0	0	1	Request PWE	Requests a parameter value (PWE)
0	0	1	0	Change PWE (word)	Writes a parameter value, word format (16 bit)
0	0	1	1	Change PWE (double word)	Writes a parameter value, double-word format (32 bit)
0	1	0	0	Request PBE element 1)	Reads an element from the parameter description (PBE). IND defines which element is to read. The complete PBE is requested if IND = 255
0	1	0	1	Change PBE element 1)	Writes an element from PBE; as for reading from PBE
0	1	1	0	Request PWE (array) 1)	Reads a parameter value from a one-dimensional field (≙ array). The position within the field, from which the value is to be read, is contained in IND. For example, IND = 4, then the PWE is transferred which is at the 4th position in the array.
0	1	1	1	Change PWE (array word) 1)	Writes a value (as word) to a specific position in a one-dimensional field (array); as for reading.
1	0	0	0	Change PWE (array double word) 1)	Writes a value as double word, such as ID 0111.
1	0	0	1	Request the number of array elements	Reads the number of elements of a field.
1	0	1	0	Reserve	
1	0	1	1	Change PWE (array double word), and store in the EEPROM 1), 2)	Writes a parameter value (double word) to a position in an array into the EEPROM
1	1	0	0	Change PWE (array word), and store in the EEPROM 1), 2)	Writes a parameter value (word) at a specific position in an array into the EEPROM
1	1	0	1	Change PWE (double word), and store in EEPROM 2)	Writes a parameter value (double word) into the EEPROM
1	1	1	0	Change PWE (word), and store in the EEPROM 2)	Writes a parameter value (word) into the EEPROM.
1	1	1	1	Request or change text 1), 2) 3)	Reads or writes a text

1) For these tasks, to make them completely clear, the value is required, which is located in IND in the net data block, refer to Section 4.2.2.
2) These IDs are only valid for the USS protocol. In the cyclic task IDs of the PROFIBUS profile /1/, these IDs are not available.
3) The "change text" supplement is only valid for SIMOVERT Master Drives

Table 4.1: Task IDs

Response IDs (slave → master)

Bit No. 15	14	13	12	Function	Description
0	0	0	0	No response	No response
0	0	0	1	Transfer PWE (word)	Transfers a parameter value (PWE) as word (16 bit)
0	0	1	0	Transfer PWE (double word)	Transfers a parameter value (PWE) as double word (32 bit)
0	0	1	1	Transfer PBE element 1)	Transfers an element from the parameter description (PBE). The particular PB element which is to be transferred is located in IND. The complete PBE is transferred if IND = 255.
0	1	0	0	Transfer PWE (array word) 1)	Transfers a parameter value (PWE), from a location specified in IND, within a one-dimensional field (^ array)
0	1	0	1	Transfer PWE (array double word) 1)	As previously, only PWE in a double-word format
0	1	1	0	Transfer the number of array elements	Transfers the number of elements of a field
0	1	1	1	Task cannot be executed (with error number)	The slave cannot execute the task. Refer to the error number for the reason
1	0	0	0	No PKW parameter change rights	Parameter values, parameter definitions or associated texts cannot be changed and can only be read from the interface on which the protocol is executed.
1	0	0	1	Parameter change report (word)	)
1	0	1	0	Parameter change report (double word)	)
1	0	1	1	Parameter change report (array word) 1)	) Refer to Section 4.2.1.2
1	1	0	0	Parameter change report (array double word) 1)	)
1	1	0	1	Reserve	
1	1	1	0	Reserve	
1	1	1	1	Transfer text 1) 2)	Text is transferred

1) For these tasks, to make them completely clear, the value is required which is located in the IND value in the net data block, refer to Section 4.2.2.
2) These IDs are only valid for the USS protocol. The IDs are not available in the cyclic response IDs of the PROFIBUS profile /1/.

Table 4.2: Response IDs

Relationship between the issued task and the associated response

Task ID					Response ID (positive)				
ID				Function	ID				Function
0	0	0	0	No task	0	0	0	0	No response
0	0	0	1	Request PWE	0	0	0	1	Transfers PWE (word)
					0	0	1	0	Transfers PWE (double word)
0	0	1	0	Change PWE (word)	0	0	0	1	Transfers PWE (word)
0	0	1	1	Change PWE (double word)	0	0	1	0	Transfers PWE (double word)
0	1	0	0	Request PBE element	0	0	1	1	Transfers PBE element
0	1	0	1	Change PBE element	0	0	1	1	Transfers PBE element
0	1	1	0	Request PWE (array)	0	1	0	0	Transfers PWE (array word)
					0	1	0	1	Transfers PWE (array double word)
0	1	1	1	Change PWE (array word)	0	1	0	0	Transfers PWE (array word)
1	0	0	0	Change PWE (array double word)	0	1	0	1	Transfers PWE (array double word)
1	0	0	1	Request number of array elements	0	1	1	0	Transfers the number of array elements
1	0	1	1	Change PWE (array double word) and store in the EEPROM	0	1	0	1	Transfers PWE (array double word)
1	1	0	0	Change PWE (array word) and store in the EEPROM	0	1	0	0	Transfers PWE (array word)
1	1	0	1	Change PWE (double word) and store in the EEPROM	0	0	1	0	Transfers PWE (double word)
1	1	1	0	Change PWE (word) and store in the EEPROM	0	0	0	1	Transfers PWE (word)
1	1	1	1	Request or change text	1	1	1	1	Transfers text
1	0	1	0	Reserve	0	1	1	1	Task cannot be executed

Table 4.3: Task ID and associated response IDs

If tasks cannot be executed, the task receiver sends the response ID "task cannot be executed" and transfers the appropriate error ID in the parameter value (PWE):

Error ID	Description
0	illegal PNU
1	Parameter cannot be changed
2	Lower or upper value limit violated
3	erroneous IND
4	No array
5	incorrect data type
6	Setting not permitted
7	Descriptive element not be changed
:	:
100	Reserved
> 100	Error IDs greater than 100 can be assigned for a specific converter.

Error IDs 0 to 100 are the same as the error IDs of the PROFIBUS profile "variable-speed drives" /1/.

Additional defined error numbers are documented in /1/.

Error IDs, stored in a unit, can be taken from the Instruction Manual

Task / response processing

The task / response processing defines the timing and functional sequence of data transfer for the PKW interface between the master and the slaves.

- A task or a response for the PKW interface consists of information for the task ID, the parameter number, parameter index and the parameter value. If individual information is not required, then these are preset with 0.
- **One** task or **one** response always only refers to **one** parameter value (exception: Index value 255).
- The master can only issue **one task to an interface** and it must wait for the appropriate response ID. The master must repeat its task as long as it waits for the response ID!
- The task must be completely transmitted in one telegram; split task telegrams are not permitted. The same is true for the response!
- Each task change signifies a new task, which must have an associated response. The task ID "no task" should be handled the same as for any other task ID and must be responded to with the response ID "no response"!
- If information is not required from the PKW interface in the cyclic mode (only PZD data are important), then the "no task" task must be issued.
- If there are considerable periods of time between the cyclic telegram sequence and the response in the drive converter, the response to "old task" is transmitted in the transition phase between "old task" and "new task" until the "new task" is identified and the associated response can be issued.
  For responses which contain parameter values, the slave always responds **with the actual value** when the response telegram is repeated.
- When communications are first established between the master and slave (the first time that the slave responds), the slave can, in the transition phase in which a response is being prepared in the converter, only respond with the ID "no response".
- If the master does not receive a response ID to his task from the addressed slave, then the master must respond appropriately.
- If the master does not have the parameter change rights, all change and EEPROM tasks from the drive converter are not processed, and are responded to with the ID "no parameter change rights". All read tasks are processed.
- The slave does not expect an acknowledgement from the master that the response has been received.
- Identification of the response to a task issued by the master:
  The master can identify the correct response in the response telegram by evaluating the response ID (Table 4.2), the parameter number (PNU) and if required, the value in IND.
- Identifying a new task in the slave:
  Every task, which is generated by the master after a valid response for the old task has been received, is identified as a new task.
- The master transmits a broadcast telegram,
  it is not permissible that the slaves send a response telegram to the master as result of this broadcast telegram.

4.2.1.2. Parameter change report

Bit No. 11 is defined as toggle bit in the PKE for processing parameter change reports. A slave sends a parameter change report to the master when a parameter value (PWE) is changed, if this change is not realized via the interface which is used for communications between the master and slave.

All parameters are defined (PBE) as either active or passive parameter using a bit, so that a parameter change report is not sent at each parameter value change. This bit is included in the description in the "ID" element (refer to Fig. 6.1). A parameter change report is only sent when the PWE changes for active parameters!

The following processing mechanism is executed for a parameter change report:
The standard task / response sequence, as described in Section 4.2.1.1, is immediately interrupted by the slave by transmitting, in the response telegram, the response ID "parameter change report" (IDs 9,10,11 and 12 from Table 4.2) with the appropriate **p**arameter **nu**mber (PNU) and the changed parameter value (PWE). Simultaneously, the slave changes the parameter change report toggle bit (No. 11), and in so doing, the slave indicates that the available parameter change report is new for the master. The parameter change report is transmitted to the master until the master acknowledges that the parameter change report has been definitely received by changing the toggle bit (bit 11). After this, the slave continuous with the **interrupted response processing**, or it transmits the next available parameter change report. This means, that if several parameter change reports are available, the next parameter change report can only be sent after the previous parameter change report has been acknowledged.

Changing all parameter values in an array:

If all values of a parameter array are changed using a telegram via an interface (index low byte = 255), and if only fixed telegram lengths are possible at a second interface, only the appropriate response ID (AK 9 to 12), the parameter number and the index low byte = 255 are transferred as parameter change report via this interface. In this case, the changed parameter values are **not** transferred.

Example:

Data transfer between the master and slave when two parameter change reports are present (SPM 1, SPM 2); **bit 11** of the parameter ID is observed (in brackets):

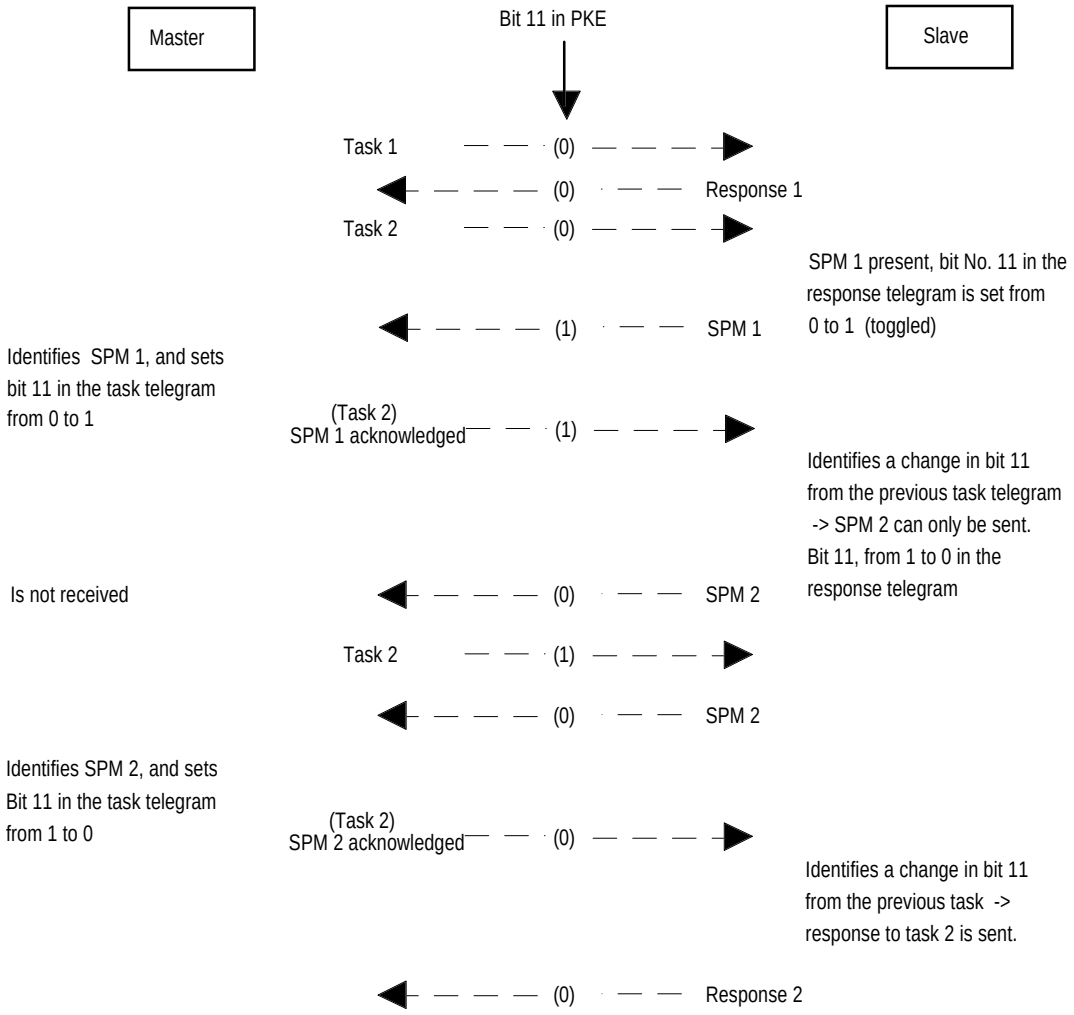


Fig. 4.5: Data sequence for parameter change reports

4.2.1.3 Parameter number (PNU)

The parameter number (PNU) is contained in bits 0 to 10 in the parameter ID (PKE). The assignment of a function / significance to a PNU (e. g. the integral-action time TN for the armature current is stored under parameter number P156 for 6RA24 SIMOREG K), is dependent on the drive converters and must be taken from the relevant Instruction Manual.

4.2.2. Index (IND)

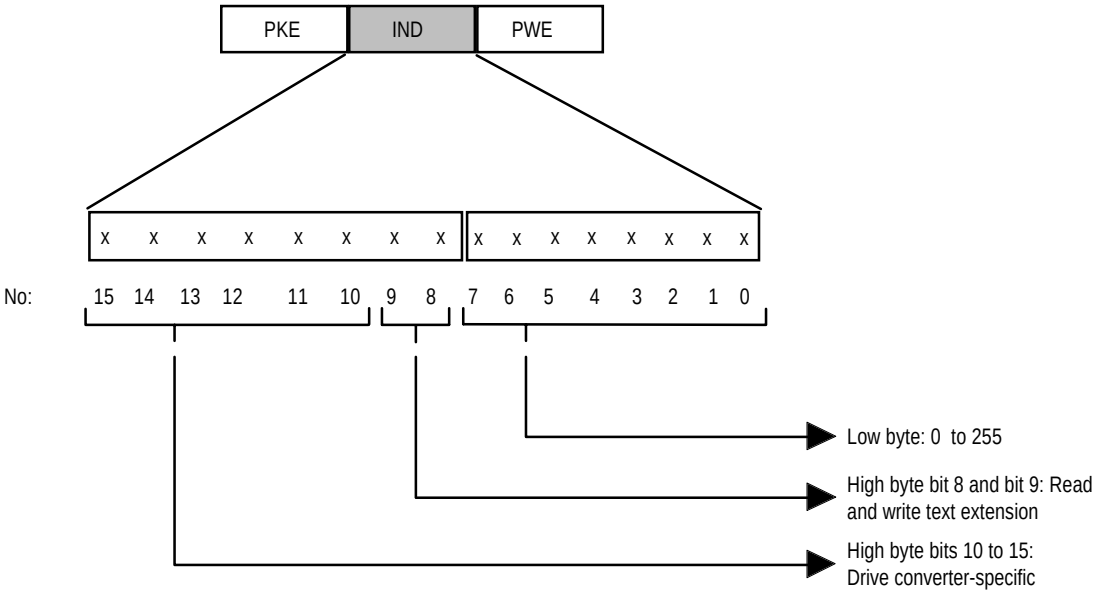


Fig. 4.6: Index structure

Using the IND for the task / response ID from Section 4.2.1.1.

The index is used for the following tasks:

- Reading and writing the parameter description
- Reading and writing values which are located in an array (= one-dimensional field)
- Reading and writing texts

In all other cases, bits 0 to 9 are set to logical 0.

The drive converter-specific bits 10 to 15 may only be evaluated by drive converters, which are defined for these bits. For drive converters, where no function is stored under bits 10 to 15, then the bits are set to logical 0. The master must take this into account when issuing tasks to the appropriate slaves.

Task ID (master → slave):
- Reading and writing the parameter description
- Reading and writing values from an array

Index low byte: 0 - 254

PKE					IND	
Task ID bit:				**Function**	**Low byte**	**Function**
15	**14**	**13**	**12**			
0	1	0	0	Request PBE element	y (≤ 254)	Read the yth element
0	1	0	1	Change PBE element	y (≤ 254)	Change the yth element
0	1	1	0	Request PWE (array)	y (≤ 254)	Read the parameter value from the yth position in the array
0	1	1	1	Change PWE (array word)	y (≤ 254)	Write the PWE in single word format to the yth position in the array
1	0	0	0	Change PWE (array double word)	y (≤ 254)	Write the PWE in double-word format to the yth position in the array
1	1	0	0	Change PWE (array word) and store in the EEPROM	y (≤ 254)	Write the PWE (word) to the yth position in the array in the EEPROM
1	0	1	1	Change PWE (array double word) and store in the EEPROM	y (≤ 254)	Write the PWE (double word) to the yth position in the array in the EEPROM

Table 4.4: Task ID and index for indexed parameters

Index low byte: 255

The index low byte = 255 has a special significance. For this index value, the task is referred to the complete array or the complete parameter description.
Error-free processing of such a task assumes that the system has been parameterized for a variable telegram length (PKW_ANZ = 127).
If the length (in words) of a parameter description or an array exceeds the maximum possible length of the PKW area, refer to Section 4.1.2, then the task should be responded to with response ID 7 (task cannot be executed), and an error message (= error number).
The procedure is the same, if the low byte = 255, parameterized for fixed telegram length.

Task ID (master → slave)

- Reading and writing a text (≙ text element)

  Only valid for SIMOVERT Master Drives

Texts are structured in the drive converter according to the PROFIBUS-profile definitions "variable-speed drives" /1/. Further, for SIMOVERT Master Drives, an extension over /1/ is defined, which allows, for an indexed parameter (= array), both a text field with text elements for each index, as well as a text field with text elements, for each parameter value, can be stored. According to /1/, it is only possible that only one text field can exist for a parameter. In order to be able to read or write these text elements, or to be able to select which of the two text fields is to be addressed, the high byte in the index (IND) is used.

Error-free processing with task ID always requires parameterization for a variable telegram length (PKW_ANZ = 127). If a fixed telegram length is parameterized for this task, then it must be also responded to with response ID 7 and error number.

Index low byte: 1 - 254 and index high byte, bits 8 and 9

PKE					**IND**		
Task ID bit: 15	**14**	**13**	**12**	**Function**	**Index high byte Bit No.: 9**	**8**	**Index low byte**	**Note**
1	1	1	1	Reads the yth text element in a text field which exists for parameters, data type: - Array - Unsigned 8/16/32 - V2	0	0	1 ≤ y ≤ 254	The yth text element is described in /1/ Section 5.1.2.4.
1	1	1	1	Writes the yth text element in the text field which exists for data types: - Array - Unsigned 8/16/32 - V2	0	1	1 ≤ y ≤ 254	"as above"

Table 4.5: Task ID and index when reading and writing texts

Contents of the text elements for parameters, data type:

- Array: Text, which describes the significance of the individual array indices.
- Unsigned 8/16/32: Text, which describes the significance of the individual parameter values of this parameter.
- V2: Text, which describes the significance of the bit values "0" and "1" of every bit in the V2 parameter.

Index low byte: 255

The special setting, low byte = 255, has no significance! A task so issued must be responded to with response ID 7 and error number.

A "second" additional text field is defined for parameters, data type array, in which text elements are stored, which describe the significance of the individual parameter values.

PKE		IND		
Task ID bit: **15 14 13 12**	**Function**	**Index high byte** **Bit No.:** **9 8**	**Index low byte**	**Note**
1 1 1 1	Reads the yth text element from the "second" text field to a parameter, data type array.	1 0	1 ≤ y ≤ 254	y = parameter value+1
1 1 1 1	Writes the yth text element from the "second" text field to a parameter, data type array.	1 1	1 ≤ y ≤ 254	y = parameter value+1

Table 4.6: Task ID and index when reading and writing the text elements of an array

Task ID (master → slave)

Index high byte, bits 10-15:

These bits can be used converter-specific, and must be described in the associated Instruction Manual of the drive converter. "0" should be sent as default value for drive converters, which do not use these bits.

Converter family	Significance of bits 0-15
SIMOVERT P 6SE21	None
Micro Master	None
SIMOVERT P 6SE12 /6SE35 with communications board C51/61	None
SIMOVERT K 6RA24	None
SIMOREG K 6RA23	None
SIMOVERT Master Drives	None
SIMOVERT PM	Addressing the infeed unit and the inverter
SIMOVERT P 6SE48	Characteristic number

Table 4.7: Significance of the converter-specific bits 10 to 15 in the index

Response ID (slave → master):

PKE					IND	
Response ID bit: 15	**14**	**13**	**12**	**Function**	**Index low byte**	**Function**
0	0	1	1	Transfer PBE element	y (≤ 254)	Transfer yth element
					255	Transfer complete PBE
0	1	0	0	Transfer PWE (array word)	y (≤ 254	Transfer the PWE, which is located at the yth position in the array (word format)
					255	Transfer the complete array (word format)
0	1	0	1	Transfer PWE (array double word)	y (≤ 254)	As above, only PWE as double word
					255	Transfer complete array (double word format)
1	0	1	1	Parameter change report (array word)	y (≤ 254)	Transfer a changed PWE as word format with a parameter change report, whereby PWE is located at the yth position in an array
					255	Transfer complete array, if variable telegram length is active with this interface
1	1	0	0	Parameter change report (array double word)	y (≤ 254)	As above, only PWE as double word format
					255	Transfer complete array, if variable telegram length is active at this interface

Table 4.8: Response ID and index for indexed parameters

Transfer response ID for text element, SIMOVERT Master Drives:

PKE					IND		
Response ID bit: 15	**14**	**13**	**12**	**Function**	**Index high byte Bit No.: 9**	**8**	**Index low byte**	**Function**
1	1	1	1	Transfer text element	x	x	1 ≤ y ≤ 254	Transfer text elements according to the particular task

Table 4.9: Response ID and index when reading and writing texts

Note: x: 0 or 1, depending on the particular task.

4.2.3. Parameter value (PWE)

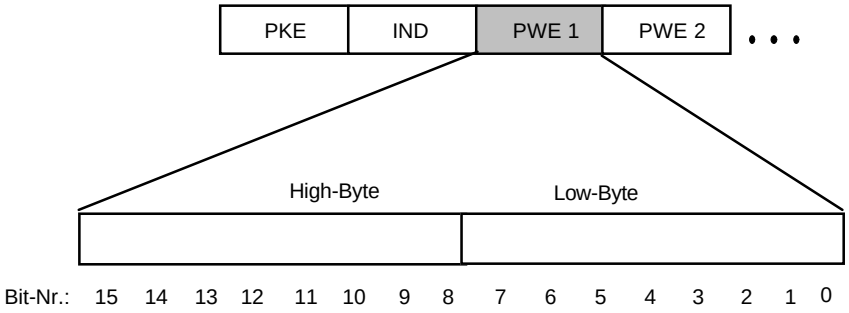


Fig. 4.7: Structure, parameter value

The assignment of PWE is dependent on the particular task, respective of the associated response.

Task telegram (master → slave)

PKE					PWE	Length of the PKW range in the net data block	
Task ID bit: 15	14	13	12	Function	Contents	Fixed (min.)	Variable necessary
0	0	0	0	No task	x ≙ not assigned	3 words	-
0	0	0	1	Request PWE	x	3 words	
0	0	1	0	Change PWE (words)	Value	3 words	
0	0	1	1	Change PWE (double word)	PWE 1 = high word PWE2 = low word	4 words	
0	1	0	0	Request PBE element	x	3 words / 4words	Yes, if
0	1	0	1	Change PBE element	PBE element	3 words / 4 words	index low byte = 255
0	1	1	0	Request PWE (array)	x	3 words	Yes, if
0	1	1	1	Change PWE (array word)	Value	3 words	index low byte = 255
1	0	0	0	Change PWE (array double word)	PWE 1 = high word PWE 2 = low word	4 words	Yes, if index low byte = 255
1	0	0	1	Request number of array elements	x	3 words	-
1	1	0	0	Store changed value (array word) in the EEPROM	Value	3 words	Yes, if index low byte = 255
1	1	0	1	Store changed value (double word) in the EEPROM	PWE 1 = high word PWE 2 = low word	4 words	-
1	1	1	0	Store changed value (word) in the EEPROM	Value	3 words	
1	1	1	1	Request or change text	- X = for "request" - text element: for "change"	No	Yes

Table 4.10: Parameter block length in the task telegram

Response telegram (slave → master)

PKE					PWE	Length of the PKW range in the net data block	
Response ID bit: 15	14	13	12	Function	Contents	Fixed (min.)	Variable required
0	0	0	0	No response	x ≙ not assigned	3 words	-
0	0	0	1	Transfer PWE (word)	Value	3 words	-
0	0	1	0	Transfer PWE (double word)	PWE 1 = high word PWE 2 = low word	4 words	-
0	0	1	1	Transfer PBE element IND ≤ 254	Element	3 words / 4 words	-
				IND = 255	All elements	No	Yes
0	1	0	0	Transfer PWE (array word) IND = y	Value (word) from the yth position in the array	3 words	Yes, if index low byte = 255
0	1	0	1	Transfer PWE (array double word) IND = y	Value (double word) from the yth position in the array	4 words	Yes, if index low byte = 255
0	1	1	0	Transfer the number of array elements	Number of array elements	3 words	-
0	1	1	1	Task cannot be executed	Error number	3 words	-
1	0	0	0	No PKW change rights	x	3 words	-
1	0	0	1	Parameter change report (word)	Value (word)	3 words	-
1	0	1	0	Parameter change report (double word)	Value (double word)	4 words	-
1	0	1	1	Parameter change report (array word) IND = y	Value (word) from the yth position in the array	3 words	-
1	1	0	0	Parameter change report (array double word) IND = y	Value (double word) from the yth position in the array	4 words	-
1	1	1	1	Transfer text	Text elements	No	Yes

Table 4.11: Length of the parameter block in the response telegram

Note:

If PKW_ANZ is set to 4, word formats are sent in the low word (PWE2). In this case, the high word (PWE1) is 0.

If PKW_ANZ is set to 127 (variable length), word formats are sent in PWE1, double-word formats in PWE1 (high word) and PWE2 (low word).

5. PZD area

The process data (PZD) area is independent of the PKW area of the second section in the net data block.

5.1. Structure of the PZD area

The PZD-area structure is always the same as far as the sequence of its elements (≙ words), and only differs from the standard structure by the number of transferred setpoints / actual values.

Standard structure:

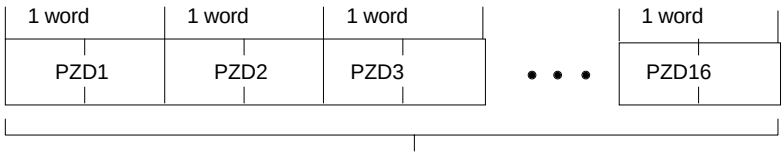


Fig. 5.1: Structure of the process data area PZD

	PZD1	**PZD2**	**PZD3 ... PZD16**
Task telegram (master → slave)	Control word	Main setpoint	Setpoints / supplementary control words
Response telegram (slave → master)	Status word	Main actual value 1)	Actual values 1) / supplementary status words

Table 5.1: Structure of the process data area PZD

Note:

The received PZD must always be processed with high priority in the master and slave. PZD processing has priority over PKW processing, and always transfers the most current data available at the interface.

1) The setpoint-actual value assignment can be selected as required, i. e. for example, the speed setpoint is transferred in the task telegram in PZD2, then the speed actual value can be signaled back in the response telegram in PZD2 (this makes sense from a technological perspective), or also a different actual value such as torque actual value, position actual value or current actual value.

5.2. Description of the individual PZD elements

5.2.1. The control word and status word

The control and status words are identical with the definition from the PROFIBUS "drive technology" profile /1/.

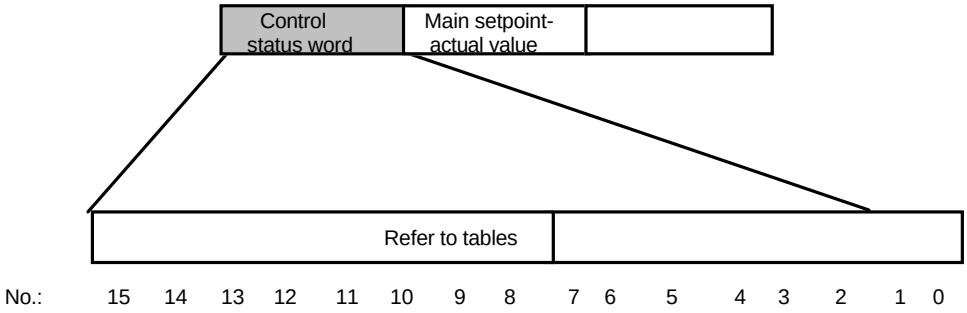


Fig. 5.2: Structure of the control- and status word

The control and status words, which are specified by a high-level automation system via the "USS® bus" correspond exactly, for bits 0 to 10, to the definitions which are specified for transmission via the PROFIBUS, in the "variable-speed drives" profile /2/. The other remaining bits are assigned, converter-specific.

Control word (bits 0 to bit 2)

Bit	Value	Significance	Comments
0	1	ON	Ready; voltage available at the converter, i. e. main contactor in (if available); the field is established; if progression is not realized within a delay time, which can be parameterized (t_{delay}), then the "switch-on inhibit" status is entered. t_{delay} = 0 ... 20 min, 20 min is interpretted as being infinite Converter-specific: Version 1: Field is established (standstill field); pulses are inhibited Version 2: DC link is charged; inverter pulses are inhibited. Version 3: Rectifier and inverter pulses are inhibited, commutating capacitors are not charged, nor post-charged Version 4: Field-, rectifier- and inverter pulses are inhibited.
	0	OFF 1	Shutdown (depending on the status of control word bits 0,1 and 2 return to the status "switch-on inhibit", "not ready to switch-on" or "ready to switch-on"); deceleration along the ramp-function generator ramp, or at the DC link voltage limit; at n/f = 0 and I = 0, supply is disconnected: Main contactor out (if available)
1	1	Operating condition	All "OFF 2" commands are canceled.
	0	OFF 2	Voltage disconnected: Converter-specific: Version 1: Shift pulses to the firing angle limit α_{max}; inhibit pulses at I =0. Version 2: Inhibit pulses. Version 3: Shift pulses to the firing angle limit α_{max}; inhibit the rectifier and inverter pulses at I = 0. Version 4: As version 3, additionally excitation and excitation contactor out. The main contactor is then switched-out (if available) and the drive goes into the switch-on inhibit condition; the motor coasts down.
2	1	Operating condition	All "OFF3" commands are canceled.
	0	OFF 3	Fast stop, if necessary cancel operating inhibit, fast as possible deceleration, e. g. along the current limit or at the DC link limit at n/f = 0; inhibit rectifier pulses, then the power is disconected (contactor out) and the drive goes into the switch-on inhibit condition.

Table 5.2: Assignment of control word bits 0 to 2

Control word (bit 3 to bit 7)

Bit	Value	Significance	Comments
3	1	Enable operation	Enable electronics + pulses Converter-specific: Version 1: Enable excitation. Version 2: If configured, de-energization time delay then the inverter pulses are enabled and the excitation current impressed Version 3: If configured, de-energization time delay, then the r rectifier and inverter pulses are enabled. Commutating capacitors pre-charge. Version 4: Pulses enabled for rotor positioning, then the rectifier and converter pulses are enabled. The drive then accelerates to the setpoint.
	0	Inhibit operation	Converter-specific: Version 1: Shift pulses to the firing angle limit α_{max}; inhibit pulses at I = O, and set excitation to the standstill excitation level. Version 2: Inhibit inverter pulses. Version 3: Shift rectifier pulses to the firing angle limit α_{max}; inhibit rectifier and inverter pulses at I = 0. Version 4: Shift the rectifier pulses to the firing angle limit α_{max}; inhibit rectifier, inverter and excitation pulses at I = 0. The drive coasts down (ramp-function generator to 0, or tracking) and the drive goes into the "ready" status (refer to the control word, bit 0).
4	1	Operating condition	
	0	Inhibit ramp-function generator	Ramp-function generator output is set to 0. The main contactor remains in, the converter is not isolated from the supply, drive decelerates along the current limit or at the DC link limit.
5	1	Enable ramp-function generator	
	0	Hold ramp-function generator	The setpoint from the ramp-function generator is held.
6	1	Enable setpoint	Selected value at the input of the ramp-function generator is switched-in.
	0	Inhibit setpoint	Selected value at the input of the ramp-function generator is set to 0.
7	1	Acknowledge	Group signal is acknowledged at the rising edge; converter is in the "fault" condition until the fault is removed, and then goes into the "switch-on inhibit" condition.
	0	No significance	

Table 5.3: Assignment of control word bits 3 to 7

Control word (bit 8 to bit 15)

Bit	Value	Significance	Comments
8 1)	1	Inching 1 ON	Prerequisite: Operation is enabled and n (set) = 0. Drive accelerates as fast as possible to inching setpoint 1
	0	Inching 1 OFF	Drive brakes as fast as possible, if "inching 1" was previously ON, and goes into the condition "operation enabled" at n/f = 0 and I = 0"
9 1)	1	Inching 2 ON	Prerequisite: Operation is enabled and n (set) = 0. Drive accelerates as fast as possible to inching setpoint 2
	0	Inching 2 OFF	Drive brakes as fast as possible, if "inching 2" was previously ON, and goes into "operation enabled" at n/f = 0 and I = 0
10	1	Control from the PLC	Control via interface, process data valid
	0	No control	Process data invalid, i. e. the "old" process data are retained
11-15		Converter-specific	Significance not specified

Table 5.4: Assignment of control word bits 8 to 15

1) The assignment of the inching function to bits 8 and 9 is optional.

Explanations:

AG	PLC	SM	Synchronous motor
ASM	Induction motor	SPM	Parameter change report
ER	Excitation	UZK	DC link
GR	Rectifier	WR	Inverter
HLG	Ramp-function generator		
α_{max}	Max. firing angle		

Version 1: DC converter
Version 2: PWN voltage-source DC link converter
Version 3: Current-source DC link converter with induction motor
Version 4: Current-source DC link converter with synchronous motor (converter-fed motor)

Status word (bit 0 to bit 7)

Bit	Value	Significance	Comments
0	1	Ready to switch-on	Power supply switched-on, electronics initialized, main contactor, if available, dropped-out, pulses inhibited
	0	Not ready to switch-on	
1	1	Ready	Refer to control word, bit 0
	0	Not ready	
2	1	Operation enabled	Refer to control word, bit 3
	0	Operation inhibited	
3	1	Fault	Drive faulted and thus non-operational, goes into the switch-on inhibit status after acknowledgement if the fault has been removed. Fault numbers are located in the fault parameters
	0	Fault-free	
4	1	No OFF 2	
	0	OFF 2	"OFF 2" command available
5	1	No OFF 3	
	0	OFF 3	"OFF 3" command available
6	1	Switch-on inhibit	Re-closure only using "OFF 1" and then "ON"
	0	No switch-on inhibit	
7	1	Alarm	Drive operational again; alarm in the maintenance-service parameter; no acknowledgement
	0	No alarm	No alarm present or alarm has disappeared.

Table 5.5: Assignment of the status word bits 0 to 7

Status word (bit 8 to bit 15)

Bit	Value	Significance	Comments
8	1	Setpoint-actual value monitoring in the tolerance range	Actual value within a tolerance bandwidth; dynamic violations permissible for $t < t_{max}$, e. g. $n = n_{set} \pm n$, $f = f_{set} \pm f$, etc. t_{max} can be parameterized
	0	Not in the tolerance range	
9	1	Control requested	The automation system is requested to take control.
	0	Local operation	Control is only possible at the converter
10	1	f or n reached	Actual value $\geq$ comparison value (setpoint), which can be set via the parameter number
	0	f/n fallen below	Actual value < comparison value
11-15		Converter-specific	Significance is not specified

Table 5.6: Assignment of status word bits 8 to 15

5.2.2. Setpoints / actual values

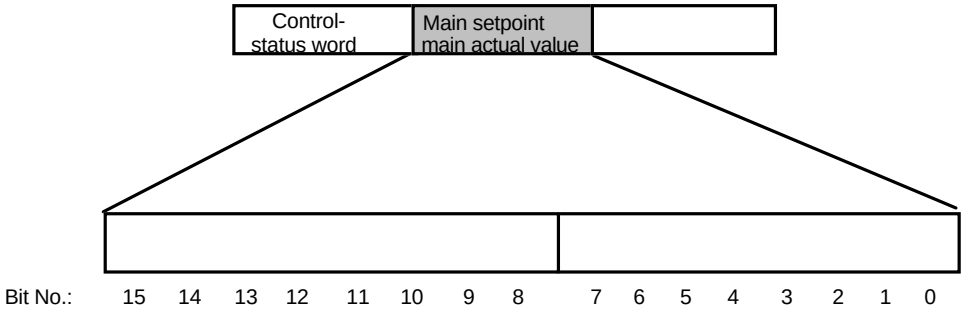


Fig. 5.3: Structure of the main setpoint and main actual value

Transfer of normalized setpoints and actual values. The normalization is dependent on the significance of the value and the particular converter type.

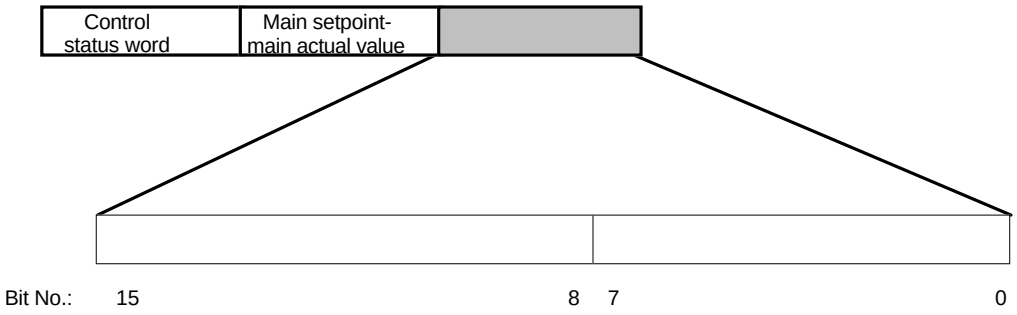


Fig. 5.4: Assignment of additional setpoints and actual values as well as, if necessary, additional control- and status words.

5.2.3. Broadcast mechanism

For SIMOVERT Master Drives, a mechanism has been defined for transferring process data, which allows the master to simultaneously transfer control word (s) and setpoints for all drives connected at the bus, in a telegram. Using masking, it is possible to define which setpoints and which control word bits should actually be influenced by the broadcast telegram. The masking is always sent in the particular broadcast telegram. In this case, a fixed length of 4 words is always used in the PKW area. This means, that "standard" PKW processing is not possible for the broadcast telegram. A detailed description of the broadcast mechanism is included in the Appendix.

6. Data transfer format for the net data

For net data, which consist of more than one byte, for data transfer via the bus, the most significant part is first transferred. This definition is identical with that for the transfer of net data via PROFIBUS. The following are valid:

Transfer of word formats:
The high byte is always transferred before the low byte
This is valid for all 16-bit data types: e. g. V2, unsigned 16, integer 16, etc. (refer to /1/)

Transfer of double-word formats:
A high word is always transferred before a low word. When transferring high and low words, the following data transfer regulations apply for word formats.
This is valid for all 32-bit data types: e. g. unsigned 32, integer 32, floating point, etc. (refer to /1/)

Transfer of byte formats:
In this case, there is no particular sequence. Data is transferred in the same sequence that it is stored in the converter "internal memory".
This is valid for data, type: Byte string (= octet string, according to /1/)

Transfer of texts:
Texts are comprised of individual characters. Each character has a byte format. There is no particular sequence for transferring the individual characters. The characters are transferred in the sequence in which they are stored in the converter "internal memory".
The data type for text characters is the visible string, refer to /1/.

The data transfer syntax is explained in the following example using the parameter description.
According to PROFIBUS profile /1/, a parameter description belongs to each converter parameter. The parameter description itself consists of several elements, e. g. the ID, normalization etc.. Depending on the converter degree of expansion, the parameter description is either completely or only partially present. Structure, scope and significance of the parameter description are explained in detail in /1/.

The complete parameter description of a parameter is illustrated in Fig. 6.1, as this is stored in the converter "internal memory". The example selected here shows how data is stored for SIMOVERT master drives. Data is stored in the converter conforming to the "intel" format, which means, that for example, a word, data type "unsigned 16" that the least significant byte is stored in the least significant address.

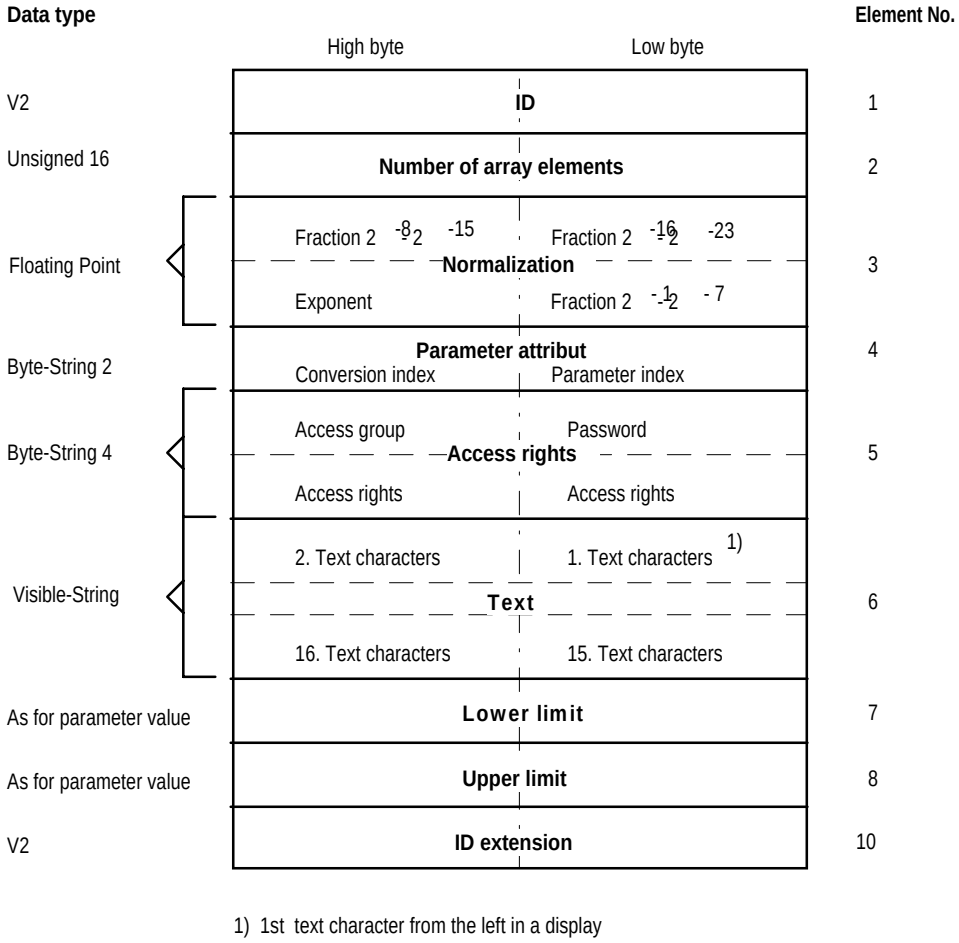


Fig. 6.1: Complete parameter description structure in the "internal memory" for SIMOVERT master drives

The individual **elements** of the parameter description are highlighted in Fig. 6.1. The appropriate data type of the particular element is shown on the left hand side. The element number in the parameter description is shown on the right hand side.

The parameter description transfer via the bus is illustrated in Fig. 6.2.
The parameter description can be read with the task "request PBE", refer to Table 4.1 in Section 4.2.1.1 from the master. In order to read the complete parameter description, the low byte must be set to 255 in IND; refer to Section 4.2.2. In order to be able to read only **one** element of the parameter description, the element number must be set in the low byte of IND. The "ID" element is number 1, the element "number of array element" is reserved for number 2, ..., number 9, the element "ID extension" is number 10 (also refer to Section 8, example 4). The slave transfers the parameter description in the PKW area of the net data telegram. If the PKW area length (in words) is defined to be less than (refer to Section 3) the length of the element to be transferred, the task is responded to by the slave with the negative response ID "task cannot be executed" (Table 4.2). Only the parameter description is shown in Fig. 6.2; the other net data components are not illustrated.

The data transfer sequence is shown in Fig. 6.2 from top to bottom. This means, that at first, the high byte of the ID is transferred, followed by the low byte etc.

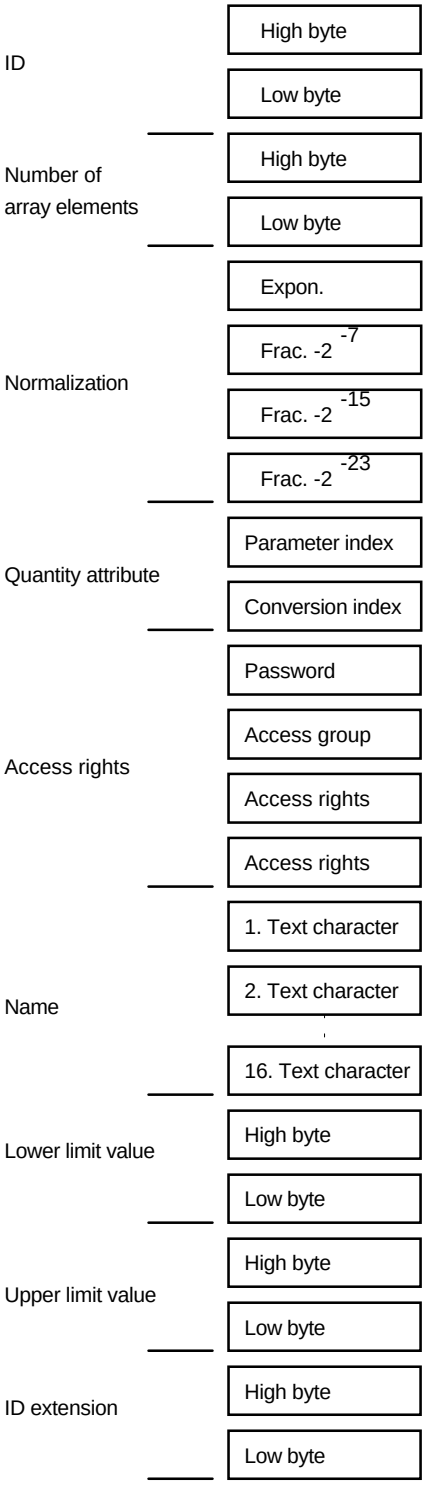


Fig. 6.2: Transfer sequence of the parameter description via the bus

7. Configuring the protocol on the bus system

As already explained in the introduction, it is possible to configure communications between master and slaves with a fixed or variable telegram length.

Fixed telegram length

When configuring communications with the USS® protocol for a fixed telegram length this means:

- The following is valid for communications between the master and a slave:
  The task- and response telegrams have the same length, i. e. the same length regarding the PKW- and PZD areas.
- This length must be set before the bus system is first commissioned, and may not be changed during operation.
- A fixed telegram length means that the net data block has a fixed size.
- The net data block size is set using two parameters, refer to Section 3.1.
  The size of the PKW area (in words) is set via parameter "PKW_ANZ", if PKW_ANZ is set to 3, the PKW area in the net data block always takes-up 3 words. The size of the PZD area (words) is appropriately set via parameter PZD_ANZ.
  For example, if PZD_ANZ = 2, then the PZD area takes-up 2 words in the net data block.
- If the master issues a task which should have a response as result, which would extend beyond the selected size of the PKW area, this task must be responded to with the response ID "task cannot be executed".

  Example: For PKW_ANZ = 3, the task "request PWE (double word)" cannot be executed. In this case, PKW_ANZ must be set to 4.
- Before setting the size of the net data block, it must be defined, which tasks are to be issued by the master. Based on this, the PKW area size must be defined. This means, that if double-word processing is used, then before first commissioning, the PKW area must be set to 4 words, even if mostly single-word processing is used.

Variable telegram length

Data transfer between master and slave with variable telegram length means:

- Variable telegram length from the master to the slave (task telegram) and
- Variable telegram length from the slave to the master (response telegram).

The following conditions apply:

Variable telegram length = variable PKW-area length (PKW-ANZ = 127)

- The PZD area (process data) must always be the same size for the task- and response telegram. This means, that the slave expects and transmits, independently of the actual parameterization of the PKW area, the number of process data, defined in parameter PZD_ANZ.
- For parameterization with variable telegram length in the slave, the "length byte" LGE in the telegram frame must always be evaluated. With this information, and with the fixed parameterization of the process data (parameter PZD_ANZ), the length of the received, variable task telegram can be clearly defined.
- An appropriate program on the master side must also be conceived, so that the variable response telegram from the slave can be identified and evaluated error-free. Due to the selected tasks, as well as information regarding the settings of the parameters of the individual slaves, the master knows whether the addressed slave responds with a variable telegram.
- From the configuring, it is possible, that slaves, which are parameterized with fixed - and slaves with variable telegram length, can communicate with a master via the same bus. However, this can result in increased software costs in the master.
- PKW-area structure
  For a variable telegram length, only the data which are actually necessary for the particular task or response, are transferred.
  Example:
  Master: Task, request text element y →
  PKW area: PKE, IND, PWE 1 (minimum 3 words)
  Slave: Transfer response, text element y (16 characters)
  PKW area: PKE, IND, PWE 1, PWE 2... PWE 8
- Monitoring the telegram failures
  It is difficult to set an optimum telegram failure time in the slave when configuring variable telegram length. This is even more critical, the more varied the possible task versions from the master to the slaves.
  Under worst case conditions (low baud rate, high number of nodes, long variable telegrams), the telegram failure time must be inactivated in the slaves.

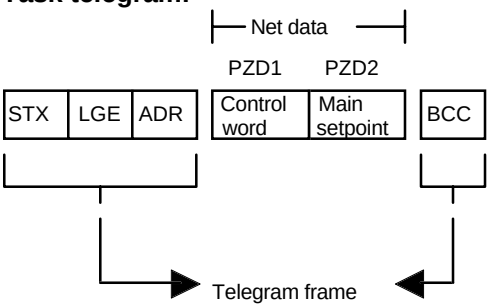
8. Examples

8.1. Fixed telegram length

Example 1: Transferring two words of process data (control word / status word, setpoint / actual value)

- Parameterization
  PKW_ANZ = 0
  PZD_ANZ = 2
- Task for PKW interface not possible
- Only PZD area with control word / status word and a setpoint / actual value in the telegram

Task telegram:



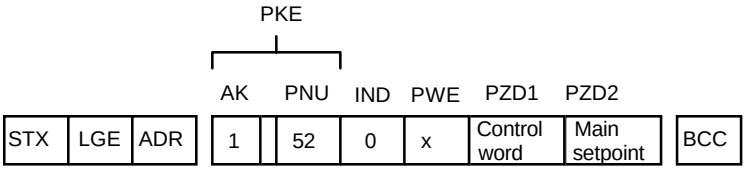
Response telegram:

Fig. 8.1: Telegram structure to example 1

Example 2: Transferring from a parameter (word format) and 2 words of process data

- Parameterization
  PKW_ANZ = 3
  PZD_ANZ = 2
- Task: Read value from parameter No. 52 (decimal); value = 4000 word format as hexadecimal value)
- Continuous transfer of the control / and status word and main setpoint / actual value.

Task telegram:



x ≙ unassigned, not relevant!

Response telegram:

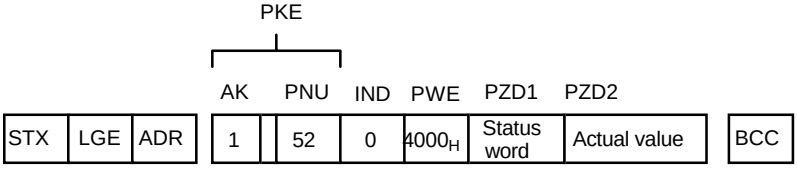


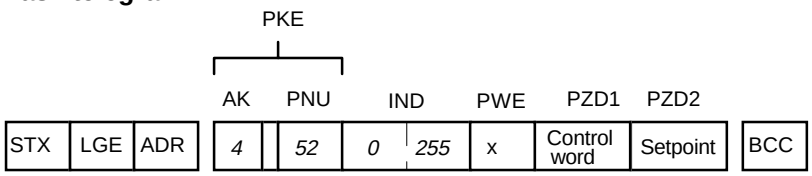
Fig. 8.2: Telegram structure to example 2

Example 3: Erroneous parameter read task

As for example 2, however

- Task: Read complete parameter description to parameter P 52

Task telegram:



Response telegram:

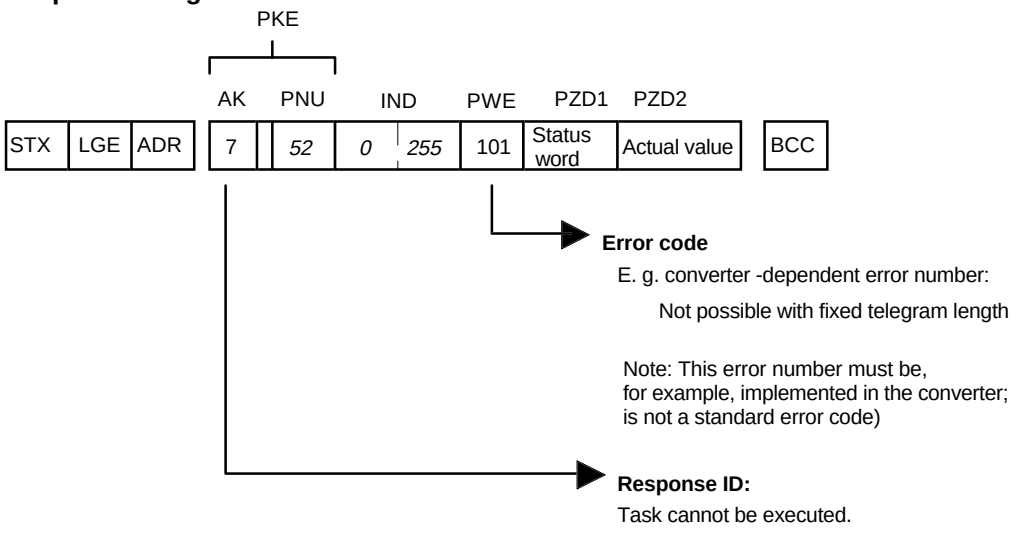


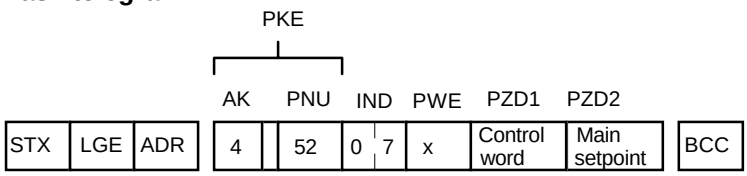
Fig. 8.3: Telegram structure to example 3

Example 4: Reading an element from the parameter description

As for example 2, however

- Task: Read "lower limit value" element (element No. 7) from the parameter description to parameter 52. Lower limit value = $5FFF_H$.

Task telegram:



x ≙ unassigned, not relevant!

Response telegram:

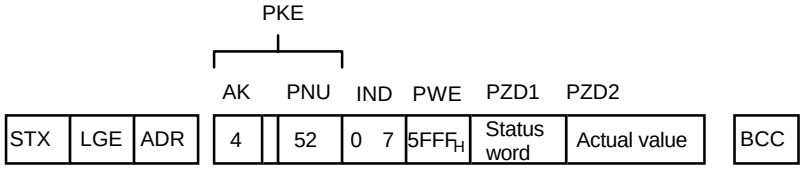
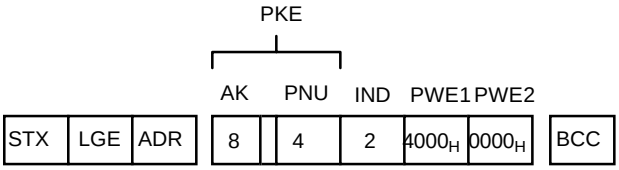


Fig. 8.4: Telegram structure to example 4

Example 5: Writing a double word parameter into a parameter field

- Fixed telegram length:
  PKW_ANZ = 4
  PZD_ANZ = 0
- Task: Writing a value (double word) = 4000 0000 (hex) to parameter No. 4 at the 2nd position of the uni-dimensional field, stored under PNU = 4 (= array).
- No PZD data in the telegram

Task telegram:



Response telegram:

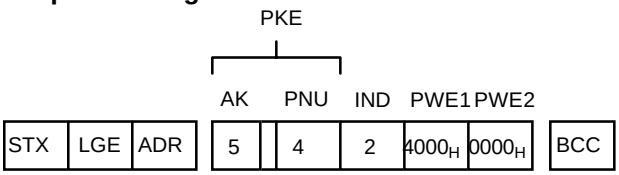


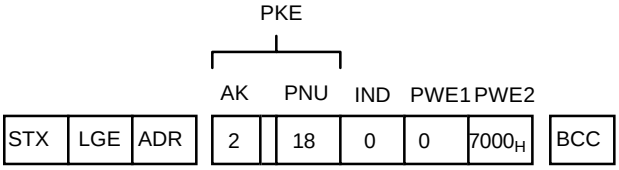
Fig. 8.5: Telegram structure to example 5

Example 6: Parameter in a single word format for 4 words PKW

As for example 5, however

- Task: Writing a value (word format) = 7000 (hex) to parameter No. 18.

Task telegram:



Response telegram:

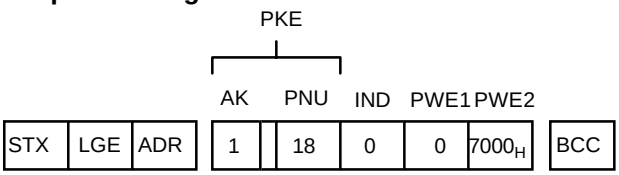


Fig. 8.6: Telegram structure to example 6

8.2. Variable telegram length

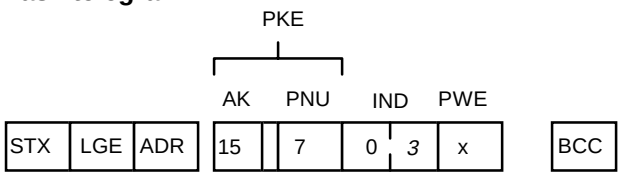
Examples for SIMOVERT Master Drives

Example 7: Read a text from a text array which belongs to parameter 7.

P 7 is "unsigned 16" data type (≙ 02) and has a value range from 1 to 10. P 7 has a text field, where every parameter value is assigned a "16 character" text element. Text field index = parameter value +1 (syntax according to /1/).

- Parameterization
  Slave: PKW_ANZ = 127
  PZD_ANZ = 0
- Task Read text element to parameter value 2 of P7
  3rd text element: 300␣ BAUD␣ ␣ ␣ ␣ ␣ ␣ ␣ ␣
  (According to /1/, always 16 characters)

Task telegram:



Response telegram:

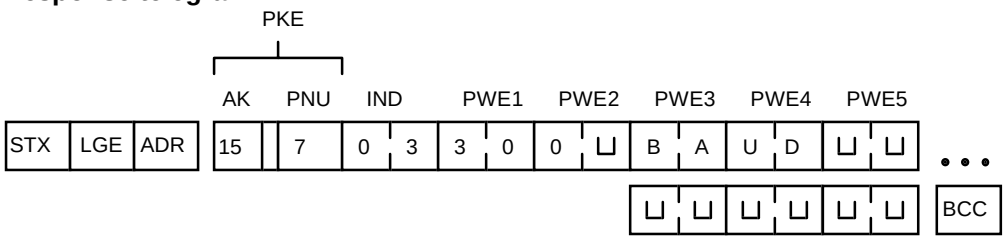


Fig. 8.7: Telegram structure to example 7

Example 8: Writing and reading text elements, indexed parameters (array)

The example parameter P 9 is an "array unsigned 16" data type (field with 02 types) with indices 1, 2 and 3. Each index has a value range between 1 and 20. The significance of the particular value range is the same for all indices.

2 text fields exist for this parameter:

- "first“ text field contains the significance of the indices
- "second“ text field contains the significance of the parameter values

The indices, which must be specified in the task telegram, are determined differently for the two fields:

- "first“ text fieldindex of the text filed = index of the parameter array
- "second“ text field index of the test field = parameter value + 1

1. Text field

Index	Text element to the index
1	BAUDRATE FOR SS1
2	BAUDRATE FOR SS2
3	BAUDRATE FOR SS3

2. Text field

Index	Value	Text element to param. value
2	1	300 BAUD
3	2	600 BAUD
4	3	1200 BAUD
..	..	..
21	20	1.5 MBAUD

Fig. 8.8 Text fields for indexed parameters

- Parameterization
  Slave: PKW_ANZ = 127
  PZD_ANZ = 0
- ID, parameter number and index are the same in the task- and response telegram

Possible tasks:	**AK**	**PNU**	**IND HB**	**IND LB**	**Task telegram PWE1 to PWE8**	**Response telegram PWE 1 to PWE 8**
Read text element for index 2	15	9	0	2	-	BAUDRATE FOR SS2
Write text element for index 2	15	9	1	2	BAUDRATE FÜR SS2	(= task telegram)
Read text element for parameter value 3	15	9	2	4	-	1200 BAUD
Write text element for parameter value 3	15	9	3	4	1200 BAUD	(= task telegram)

Task- and response telegram:

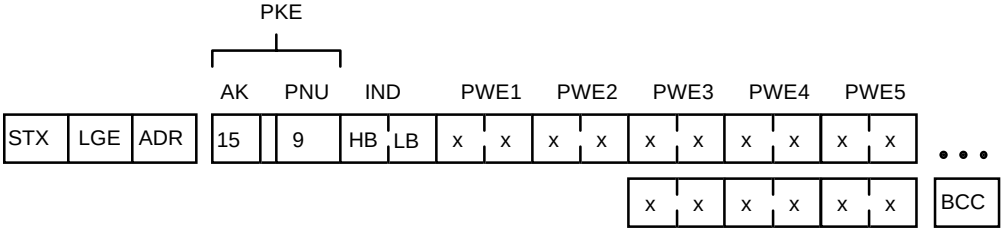
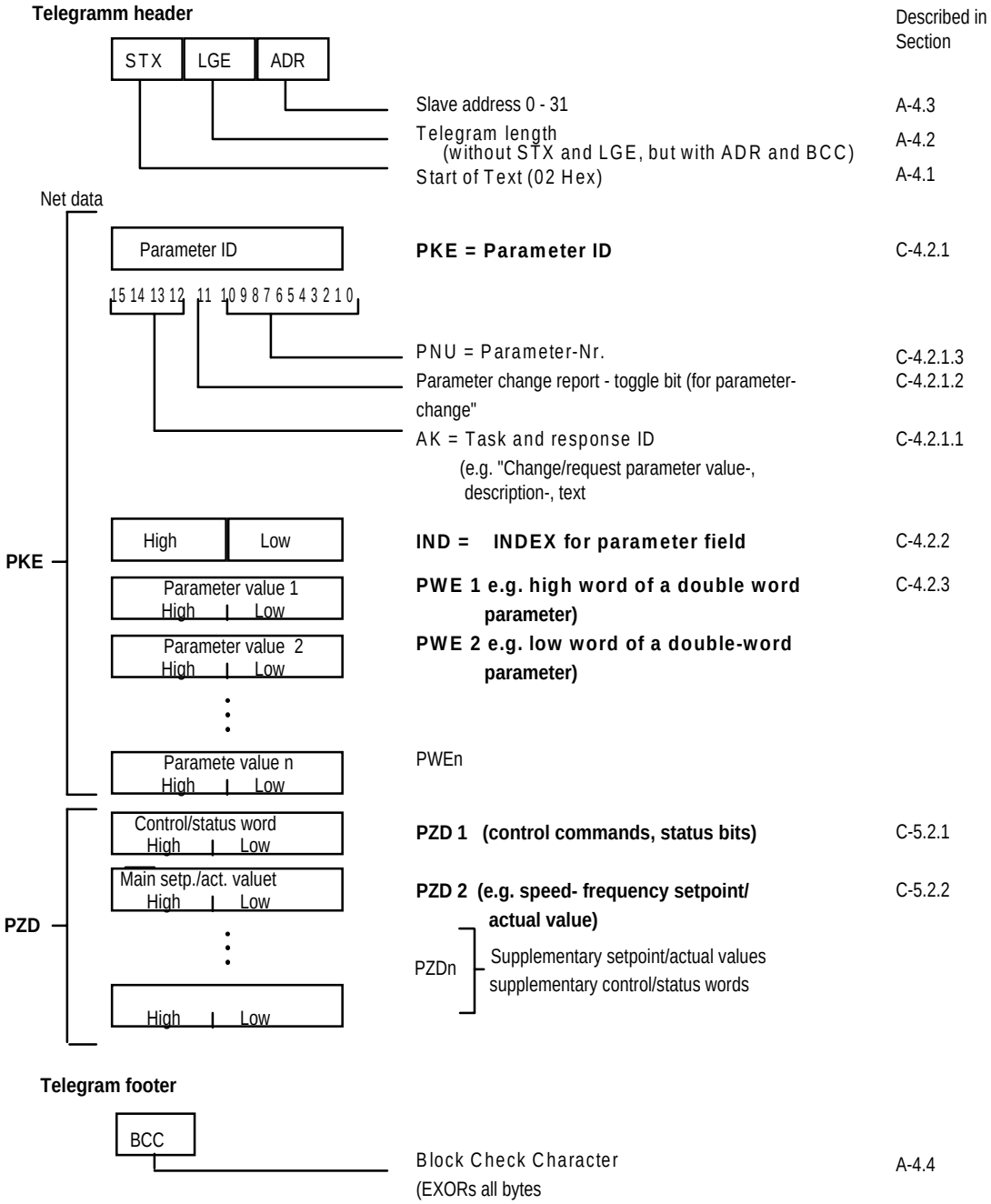


Fig. 8.9: Telegram structure to example 8

APPENDIX

Overview: Telegram structure for the USS®-protokoll



The Optional Broadcast Mechanism of the USS-Protocol

The Broadcast Mechanism is not obligatory for the USS slave interface. But if a Broadcast Mechanism is implemented for a slave communication interface, it must be realized according to this specification.

Whenever the term "Broadcast Telegram" is used in this document, it describes a telegram that has to be accepted by all drives connected via a Bus configuration; all drives fetch exactly the same identical net data from this telegram. A Broadcast Telegram has the same telegram length as a normal individual telegram. (A different type of Broadcast Telegram is defined in the PROFIBUS standard: A PROFIBUS Broadcast Telegram is a "multiple length" telegram whereby the individual information blocks required for all drives connected to the bus are contained in one extra long "group-telegram".)

It is a typical application of a Broadcast Telegram to enable the ramp generators of all drives at the same point of time by means of only one telegram in order to let all drives ramp up simultaneously.

Selective Reaction to a Broadcast Telegram

According to the experience with other protocols, a Broadcast Telegram from an external communication partner only makes sense if a 'selective reaction' to a Broadcast Telegram is made possible. This means that the drives must be told in the net data what REF VALUES and what Command Bits should be influenced by the Broadcast Telegram and what PROCESS DATA (PZD) should not be affected (the latter mentioned data must be dropped and instead the signals transmitted in the last "Non-Broadcast Telegram" remain effective).

The following construction is specified for this purpose: A Broadcast Telegram is marked by a "1" in bit 5 of the Address-Byte in the telegram header. In case of a Broadcast Telegram the PARAMETER DATA area (PKW) is "misused" for transmitting the information of how to react to the Broadcast Telegram instead of transmitting the normal PARAMETER DATA.

If the master station issues a Broadcast Telegram it is not allowed to enter a parameter into the PARAMETER DATA area - as it does with normal telegrams - but instead it has to enter a 4 Words "BROADCAST ENABLE ARRAY" into this space (refer to Fig. 2.5.1.6). This does not cause any problems because reading and writing parameters via a Broadcast Telegram is nonsense since: Broadcast Telegrams, as a general principle, are never answered by the drives.

The BROADCAST ENABLE ARRAY contains Enable-Bits for each of the PROCESS DATA WORDS 1 to 15 and for each of the max. possible 48 Command Bits in the Broadcast Telegram. If a PROCESS data block originates from a Broadcast Telegram, then a Command Bit or a REF VALUE is only accepted by the drive if the related Enable Bits are set to 1. If for a Command Bit or a REF VALUE the related bits in the BROADCAST ENABLE ARRAY are reset, the Command Bit or REF VALUE remains in the old state which has been set by the last Non-Broadcast-Telegram. The rules for processing the PROCESS DATA WORDS of a Broadcast Telegram according to the Bits in the BROADCAST ENABLE ARRAY are described below.

Rules for processing the PROCESS DATA WORDS of a Broadcast Telegram

The 16 PROCESS DATA WORDS 1 to 16 can be classified into the following 3 groups:

- PROCESS DATA WORDS 2, 3 and 6 to 15:

  These PROCESS DATA WORDS can contain REF VALUES only ; they cannot act as COMMAND WORDS. A PROCESS DATA WORD of this group is only picked up by the LOWER SIDE BOARD from a Broadcast Telegram if the corresponding enable bit in Word a of the BROADCAST ENABLE ARRAY is set to "1".

- PROCESS DATA WORDS 1, 4 and 5:

  These PROCESS DATA WORDS can act as COMMAND WORDS: WORD 1 is always a COMMAND WORD, according to the PROFIBUS-Profile (Lit. [2]). Words 4 and 5 may either be COMMAND WORDS or REF VALUES, e.g. depending on the software type of the TECH BOARD.
  For each of the 48 bits of these PROCESS DATA WORDS there is a corresponding bit in the words b, c and d of the BROADCAST ENABLE ARRAY.
  A PROCESS DATA Bit in the PROCESS DATA WORDS 1, 4 and 5 is only taken over from a Broadcast Telegram if

  I) the whole PROCESS DATA WORD is enabled via a "1" Information in the corresponding bit in Word a and additionally

  II) the PROCESS DATA Bit itself is enabled by the corresponding bit of word b, c or d respectively.

  As a general rule it is prescribed that in case of a Broadcast Telegram the BROADCAST ENABLE WORDS b, c and d must always be processed by the master station and the slave station irrespective of the fact whether the corresponding PROCESS DATA WORDS 1 4 and 5 are REF VALUES or COMMAND WORDS. This rule makes the implementation of a general-purpose PROCESS DATA receive function block easier (e.g. for a SIMADYN D fashioned TECH BOARD). As a consequence of this e.g. all bits in Bit mask d have to be set by the host system even if PROCESS DATA WORD 5 is a REF VALUE and is to be affected by the Broadcast Telegram.

- PROCESS DATA WORD 16

  The 16th PROCESS DATA WORD is not supported by the Broadcast mechanism; this means: Word 16 is not Broadcast-capable and never taken over from a Broadcast Telegram. The master station can modify this word via a Non-Broadcast Telegram only.

The rules for processing the PROCESS DATA WORDS of a Broadcast Telegram are summarized in the following table:

PROCESS DATA WORD	**BROADCAST ENABLE bits which must be taken into account**
WORD 1	Bit mask a (Bit 1) and Bit mask b
WORD 2	Bit mask a (Bit 2)
WORD 3	Bit mask a (Bit 3)
WORD 4	Bit mask a (Bit 4) and Bit mask c
WORD 5	Bit mask a (Bit 5) and Bit mask d
WORDS 6 to 15	Bit mask a (Bits 6 to 15)
WORD 16	not taken over from a Broadcast Telegram (not Broadcast-capable)

Example 1 of a Broadcast Telegram:

The host system wants to give the same Main Reference Value 'REF1.' to all drives connected to the bus via a Broadcast Telegram. All other PROCESS DATA of the telegram should be ignored by the drives.

⇨ In this case the host system has to transmit the following information in the BROADCAST ENABLE ARRAY:

```
Word a       = 0000 0000 0000 0101 (Broadcast Flag set and
                                    PROCESS DATA WORD 2
                                    enabled)
Words b to d = 0000 0000 0000 0000
```

Example 2 of a Broadcast Telegram:

The host system wants to start the ramp generators of all drives connected to the bus simultaneously via one Broadcast Telegram. All other PROCESS DATA must not be processed by the drives.

According to the PROFIBUS Profile (Lit [2]) the ramp generator can be enabled by setting bit 5 of COMMAND WORD 1 to a "1" value.

⇨ In this case the host system has to transmit the following information in the BROADCAST ENABLE ARRAY:

```
Word a       = 0000 0000 0000 0011 (Broadcast Flag set and
                                    PROCESS DATA WORD 1
                                    (= COMMAND WORD 1)
                                    enabled)
Word b       = 0000 0000 0010 0000 (Command Bit 5 enabled)
Words c to d = 0000 0000 0000 0000
```

Remarks on the Broadcast Telegram:

- Broadcast Telegrams are only possible with telegram types containing a PARAMETER DATA area (e.g. PROFIBUS Adjustable-Speed Drives Profile , PPO types 1, 2 or 5)
- For a double-word REF VALUE (32 bit information) the Enable Bits of both: High Word and LOW Word have commonly to be set or reset by the master station.

Drives and Standard Products Group
Motors and Drive Systems Division
Postfach 3269, D-91050 Erlangen

Germany

Siemens Aktiengesellschaft

We reserve the right to make changes

Order No. E20125-D0001-S302-A1-7600
Printed in the Fed. Rep. of Germany