Rexroth
Bosch Group

Rexroth SIS Serial Interface

R911289718
Edition 02

Information

Title Rexroth SIS
Serial Interface

Type of Documentation Information

Document Typecode DOK-GENERL-SIS-DEFINIT-IF02-EN-P

Internal File Reference
- 289719_SIS_IF_en.doc
- Document number: 120-1300-B306-02/EN

Purpose of Documentation This documentation is for:
- Information about data exchange via the serial interface
- Selecting the services needed for an application of the serial interface
- Guide for initializing SIS communication
- Information about pin assignment and cables

Record of Revisions

Description	Release Date	Notes
DOK-GENERL-SIS-DEFINIT-IF01-EN-P	12.00	first edition
DOK-GENERL-SIS-DEFINIT-IF02-EN-P	04.04	first revision

Validity The specified data is for product description purposes only and may not be deemed to be guaranteed unless expressly confirmed in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.

Published by Bosch Rexroth AG
Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main
Telephone +49 (0)93 52/40-0 • Tx 68 94 21 • Fax +49 (0)93 52/40-48 85
http://www.boschrexroth.com/
Dept. BRC/ESP (MH)

Note This document has been printed on chlorine-free bleached paper.

Contents

1 Introduction 1-1

1.1 General Information ... 1-1
1.2 Basic Conditions ... 1-1
1.3 Hardware Settings ... 1-1
1.4 Protocol and Procedure Settings ... 1-2
1.5 User Interface Settings ... 1-2

2 Data Exchange via the Serial Interface 2-1

2.1 Telegrams ... 2-1
Telegram Directions and Types ... 2-1
Structure of the Command Telegram ... 2-2
Structure of the Response Telegram ... 2-3
Data Formats in the Telegrams ... 2-4
2.2 Telegram Header ... 2-4
Sequence and Meaning of the Individual Bytes of the Telegram Header ... 2-5
Example for the Telegram Headers and the Routing with two Sub-addresses in the Command Telegram ... 2-6
Aspects of the Specification of the Telegram Header ... 2-8
2.3 Telegram Content ... 2-11
User Data and User Data Header ... 2-11
Error Codes in the Status Byte ... 2-12
2.4 Transmission Sequence ... 2-13
Transmission of Short Data Records ... 2-14
Transmission of Long Data Records ... 2-15
Time-Out Periods ... 2-16
Behavior in Case of Error ... 2-16
2.5 Physical Data Transmission ... 2-17

3 User Identification via SIS 3-1

3.1 SIS Service 0x00 User Identification ... 3-1
Subservice 0x01 Read Out SIS Version ... 3-2
Subservice 0x02 Read Out FWA Number ... 3-2
Subservice 0x03 Read Out Unit Typecode ... 3-3
Subservice 0x04 Read Out Supported Baud Rates ... 3-4
3.2 Overview of SIS Service 0x00 User Identification ... 3-5

4 Termination of a Data Transmission via SIS 4-1

4.1 SIS Service 0x01 Termination of Data Transmission ... 4-1

5 Flash EPROM Programming via SIS 5-1

5.1 SIS Service 0x02 Flash Operation ... 5-1
Subservice 0x90 Shutdown ... 5-2
Subservice 0x91 Reboot ... 5-3
Subservice 0x92 Read Flash ... 5-4
Subservice 0x93 Find Header ... 5-5
Subservice 0x94 Erase Flash ... 5-5
Subservice 0x96 Program Flash ... 5-6
Subservice 0x97 Build Checksum ... 5-8
Subservice 0x9F Error Reset in the Slave System ... 5-9
5.2 Overview of SIS Service 0x02 Flash Operations ... 5-10
5.3 Error Codes in the SIS Service 0x02 Flash Operation ... 5-11

6 Initializing Communication via SIS 6-1

6.1 SIS Service 0x03 Initialization of SIS Communications ... 6-1
Subservice 0x01 Setting TrS ... 6-2
Subservice 0x02 Specifying TzA ... 6-3
Subservice 0x03 Setting Tmas ... 6-4
Subservice 0x06 Address Allocation for Multi-Cast Group(s) ... 6-4
Subservice 0x07 Setting the Baud Rate ... 6-5
Subservice 0x08 Timed Baud Rate Test ... 6-6
Subservice 0xFF Assuming the Set Values ... 6-8
6.2 Overview of SIS 0x03 Service Initialization of SIS Communication ... 6-9
6.3 Error Codes of the SIS 0x03 Service Initializing SIS Communication ... 6-9

7 Execution of Multiple SIS Services in a SIS Transmission 7-1

7.1 SIS Service 0x04 Executing a List of SIS Services ... 7-1
7.2 Example for SIS Service 0x04 Executing a List of SIS Services ... 7-3

8 Handling of SERCOS parameters and SERCOS phase 8-1

8.1 Telegram Content ... 8-1
User data header ... 8-1
User data ... 8-4
8.2 SIS Service 0x10 Read a SERCOS parameter ... 8-4
8.3 SIS Service 0x11 Read a segment of a SERCOS list ... 8-6
8.4 SIS Service 0x12 Read the current SERCOS phase ... 8-8
8.5 SIS Service 0x1D Switching the SERCOS phase ... 8-9
8.6 SIS Service 0x1E Write a segment of a SERCOS list ... 8-11
8.7 SIS Service 0x1F Write a SERCOS parameter ... 8-12
8.8 Overview of SIS services related to SERCOS interface ... 8-16
8.9 Error codes in SIS services related to SERCOS interface ... 8-17

9 Token Passing via SIS 9-1

9.1 SIS Service 0x0F Token Passing ... 9-1

10 Timing for the SIS Interface 10-1

10.1 Constraints on the Individual Users ... 10-1
Control Unit (BTV05) ... 10-1
Drive (DKC) ... 10-1
10.2 Sequence for First Initialization of the Bus Users ... 10-1
10.3 Synchronization of a User on the Operating Bus ... 10-1
10.4 Automatic Baud Rate Detection ... 10-2
10.5 Discussion of the Timing Values ... 10-2
Maximum Allowable Character Separation (TzA) ... 10-2
Master Repetition Period (TwM) ... 10-2
Default Period After Scanning Has Begun (TDF) ... 10-2

11 Pin Assignment and Cables 11-1

11.1 Serial Interface Pin Assignment ... 11-1
11.2 Cables for the Serial Interface ... 11-2
Requirements for the Individual Interfaces ... 11-2
List of New Cables ... 11-5
Example for a RS485 Bus ... 11-6
Examples for RS232 Connections ... 11-7
Designations for Rexroth Cables ... 11-8

12 Service & Support 12-1

12.1 Helpdesk ... 12-1
12.2 Service-Hotline ... 12-1
12.3 Internet ... 12-1
12.4 Vor der Kontaktaufnahme... - Before contacting us... ... 12-1
12.5 Kundenbetreuungsstellen - Sales & Service Facilities ... 12-2

1 Introduction

1.1 General Information

The future demands on control and automation technology require new, modular and universal concepts not only for functionality but also for communication between control or software modules and the operating or programming PC.

This communications concept is based on the following at all levels:

- Hardware
- Protocol and procedure
- User interface

1.2 Basic Conditions

The following basic conditions must be considered in specifying the communications concept.

Target hardware

The medium-term goal is:

- Unified, appropriately powerful target hardware with the pSOS+ operating system.
- The control components required for the application are implemented as software modules on the target hardware.

PC

The following applies for the PC:

- The external link to the PC can be implemented over various interfaces (RS-232, RS-422, RS-485, DPR, Ethernet, Profibus, Interbus, ...).
- The internal link to communications is implemented via the function interface.

Applications

The applications are

- programmed in high-level language, as possible,
- can be combined as a module with other applications.

1.3 Hardware Settings

The required hardware settings are:

- Physical interfaces
- Port assignment
- Handshake signals
- Monitoring
- Pin assignment
- Connectors

1.4 Protocol and Procedure Settings

Data is exchanged using telegram transmissions. The following settings are required:

Protocol

For the protocol:

- Telegram header
- Telegram content
- Checksum
- Data formats

Procedure

For the transmission procedure:

- Internal implementation of master/slave communication
- Data transmission sequence
- Sequence controls

1.5 User Interface Settings

The user interface settings are not included in this description.

These settings must be specified in the functional interface under the PC auxiliary conditions ("the internal link to communications is via the functional interface").

2 Data Exchange via the Serial Interface

Data is exchanged over the serial interface via telegrams.

2.1 Telegrams

The telegrams used for data exchange over the SIS are generally structured according to the following scheme:

- Telegram header
- User data header (depends on the SIS service)
- User data

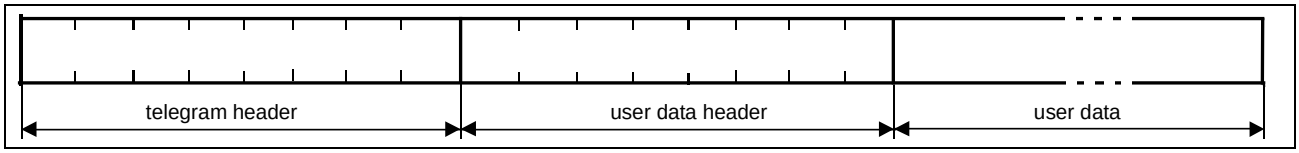


Fig. 2-1: Telegram structure for data exchange via SIS

However, the exact structure of the individual telegrams depends not only on the SIS service but also on the telegram direction and the telegram type.

Telegram Directions and Types

Telegram Directions

Two telegram directions are differentiated:

Telegram Direction	Sender of the Telegram
Command telegram	Master: the ‘active’ communications partner
Response telegram	Slave: the ‘passive’ communications partner

Fig. 2-2: Telegram directions

Theoretically, with a RS485 connection every bus user can be both a master - i.e. can send a command telegram that is then answered by the slave with a response telegram – as well as a slave, simultaneously. However, in practice there is a distinction between master partners (control devices) and slave partners (target devices).

Telegram Types in the Command Telegram

There are four types of command telegrams for the master:

Telegram Type	Data Direction
SEND telegram	Write access: data sent to a slave.
FETCH telegram	Read access: data requested from a slave.
Group message	Message: data sent to a group of slaves.
Broadcast message	Message: data sent to all slaves.

Fig. 2-3: Telegram types in the command telegram

Note: Group and broadcast messages are **not** answered by the slaves addressed.

Telegram Types in the Response Telegram

As appropriate, a slave that is addressed at its own address sends a response telegram as follows

Telegram Content	Telegram Type
Transmission status	with error-free SEND telegram
Transmission status and requested data	with error-free FETCH telegram
Transmission status and error code	with erroneous SEND or FETCH telegram

Fig. 2-4: Telegram content of the response telegram

Sequential Telegrams

Long data sets in the various forms of telegram directions and types may have to be subdivided among several subtelegrams and sent as sequential telegrams (see subsection on "Transmission Sequence", page 2-13).

Structure of the Command Telegram

Write Access

A SEND telegram is made up of:

- Telegram header
- User data header (depends on the SIS service)
- User data

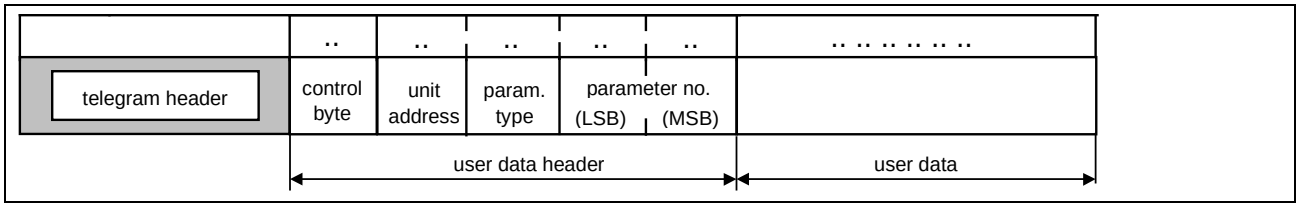


Fig. 2-5: Structure of the command telegram: write access

Read Access

A FETCH telegram is made up of:

- Telegram header
- User data header (depends on the SIS service)

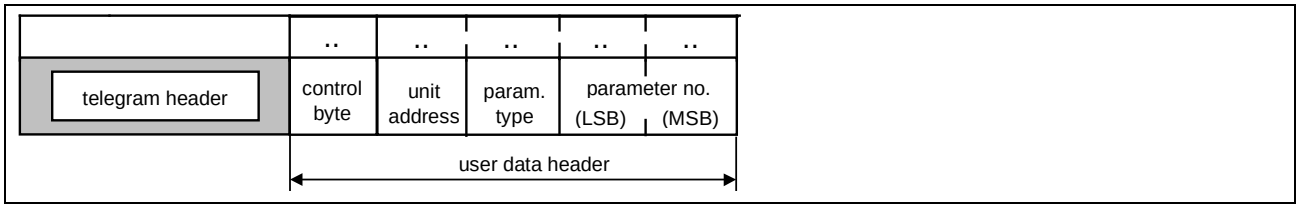


Fig. 2-6: Structure of the command telegram: read access

The user data header for specifying the read access can also be omitted if it is already set in the telegram header by the SIS service.

Structure of the Response Telegram

Write Access

The response telegram is made up of:

- Telegram header
- User data header (depends on the SIS service)
- Error code, if applicable

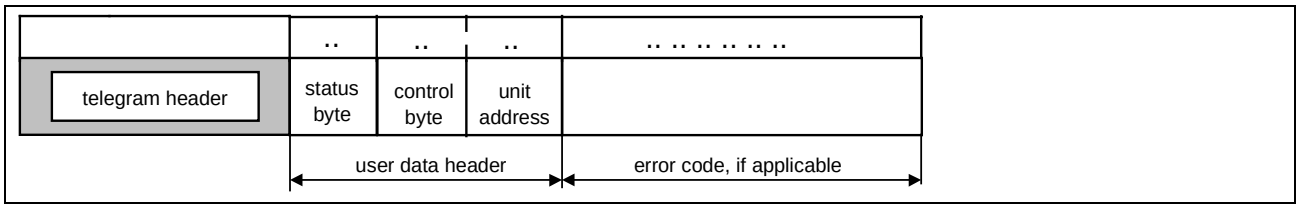


Fig. 2-7: Structure of the response telegram: write access

Read Access

The response telegram is made up of:

- Telegram header
- User data header (depends on the SIS service)
- User data or error code

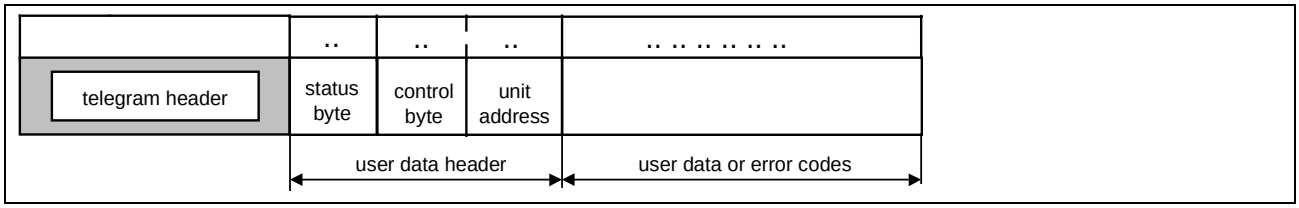


Fig. 2-8: Structure of the response telegram: read access

The response telegram from the slave has, apart from a few differences, the same telegram and the same user data header as the command telegram. This enables the sender of a command telegram to assign a unique response telegram.

Differences in the telegram header:

In the telegram header the telegram must be marked as a response telegram (see section 2.2, "Sequence and Meaning of the Individual Bytes of the Telegram Header").

Differences in the user data header:

A status byte for displaying the transmission status is shown in the first byte of the user data header for general SIS services.

Note: The user data depends on the SIS service and the status byte.

Status Byte for Special SIS Services

For special SIS services (as for general services), a transmission status should be issued in the first byte of the user data header.

Note: Historically and practically, however, this is not always possible. For such cases, a specification is yet to be determined about how to handle deviations from the general case (see section 2.2, "Sequence and Meaning of the Individual Bytes of the Telegram Header").

Data Formats in the Telegrams

The SIS telegram is a **binary telegram** with standardized, binary telegram header and content for all general SIS services.

Note: The telegram content may also consist of an ASCII data set for the special SIS services in the individual product groups.

Intel Format

The allocation of the individual bytes of data of the 'Word' or 'DWord' type corresponds to the **Intel** convention.

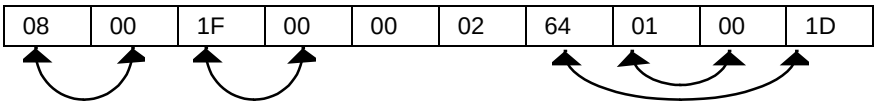
Example:

In a defined data structure the data words 0x0008 and 0x001F as well as the double word 0x1D000164 shall be transmitted.

The logical order of the data:

00	08	00	1F	00	02	1D	00	01	64
Word		Word		Byte	Byte	Long (Double Word)			

The order of bytes sent in **Intel-Format**:



IEEE Format

The **IEEE** format applies to the floating point display of data.

2.2 Telegram Header

The telegram header contains all information required for conducting orderly telegram traffic. In addition to general technical transmission conditions, it must also meet the following specific requirements:

- Addressing with up to four subaddresses
- Forwarding of the telegram to the next recipient in the simplest possible way (the existence of devices that cannot process such telegrams but will have to forward them should be taken into consideration).
- Data in the telegram header that may have to be changed or evaluated when forwarding a telegram should be in a fixed position in the telegram header.

Sequence and Meaning of the Individual Bytes of the Telegram Header

The telegram header is made up of at least 8 bytes and can be expanded to 16 bytes. It contains the following information:

Byte	Name	Meaning of the Individual Bytes
1	StZ	Start character: STX (0x02)
2	CS	This is the checksum byte. It is formed by addition of all of the following telegram characters and the start character StZ and the successive negation. This means that the sum of all telegram characters for a successful transmission is always 0.
3	DatL	This is the length of the following user data and of the variable section in the telegram header. **A minimum of 247 bytes** (255 - 7 {subaddresses} - 1 {sequential telegram number}) of user data can be transmitted in one telegram.
4	DatLW	This is the repetition of DatL. The telegram length is derived from DatLW and the fixed section of the telegram header (bytes 1 - 8), thus: telegram length = DatLW + 8.
5	Cntrl	Bit 0 - 2: Number of subaddresses in the address block (0 - 7), Bit 3: 'sequential telegram number' : 0 => not supported, 1 => additional byte Bit 4: 0 => command telegram, **1 => response telegram**, Bit 5: reserved, **(possibly for differences in the status byte of the response telegram)** Bit 6: 1 => The sender's system diagnostic signals a system warning. Bit 7: 1 => The sender's system diagnostic signals a system error
6	Service	This specifies the service that the sender requests from the recipient or that the recipient has executed. 0x00 ... 0x0F General services: 0x00 User identification 0x01 Data transmission aborted 0x02 Flash operation 0x03 Initializing SIS communication 0x04 Executing a list of SIS services 0x0F Token passing 0x10 ... 0x7F temporarily reserved 0x80 ... 0x8F Special services for ECODRIVE 0x90 ... 0x9F Special services for SYNAX 0xA0 ... 0xAF Special services for MT-CNC or. MTC200 0xB0 ... 0xBF Special services for ISP200 0xC0 ... 0xCF Special services for CLC-GPS 0xD0 ... 0xDF Special services for HMI-System 0xE0 ... 0xEF Special services for DISC 0xF0 ... 0xFF temporarily reserved
7	AdrS	Address of sender: • AdrS = 0 - 126 ==> specifies a single station, • AdrS = 127 ==> Special address for a SIS master in case of service or emergencies **(this address may not be used during active communication)**.

8	AdrE	Address of Recipient: • AdrE = 0 – 126 ==> specifies a single station, • AdrE = 128 ==> Special address for point-to-point communication **(the recipient's response is not dependent on its actual station number with this special address)**. • AdrE = 129 - 199 ==> reserved, • AdrE = 200 -253 ==> addresses logical groups, • AdrE = 254 ==> specifies a broadcast to all stations on a hierarchical level **(this address can only be listed once, as the last address in the list)**, • AdrE = 255 ==> specifies a global broadcast. Telegrams with AdrE = 200 – 255 are **not** answered with a response telegram.
9	AdrES1	Subaddress 1 of recipient, **if for bit 0 - 2 of byte Cntrl, the following applies: > 000**
10	AdrES2	Subaddress 2 of recipient, **if for bit 0 - 2 of byte Cntrl, the following applies: > 001**
11	AdrES3	Subaddress 3 of recipient, **if for bit 0 - 2 of byte Cntrl, the following applies: > 010**
12	AdrES4	Subaddress 4 of recipient, **if for bit 0 - 2 of byte Cntrl, the following applies: > 011**
13	AdrES5	Subaddress 5 of recipient, **if for bit 0 - 2 of byte Cntrl, the following applies: > 100**
14	AdrES6	Subaddress 6 of recipient, **if for bit 0 - 2 of byte Cntrl, the following applies: > 101**
15	AdrES7	Subaddress 7 of recipient, **if for bit 0 - 2 of byte Cntrl, the following applies: > 110**
16	PaketN	sequential telegram number (packet number) , **if for bit 3 in byte Cntrl is set**

Fig. 2-9: The SIS telegram header

Example for the Telegram Headers and the Routing with two Sub-addresses in the Command Telegram

In the following the telegram structure and the routing of telegrams with sub addresses are pointed out exemplarily.

The master sends a command telegram with 3 byte user data to the final recipient at second sublevel and receives a response telegram with 3 byte user data.

The addresses are:

- 10 sender address of the master
- 03 address of the router at main level
- 02 address of the router at first sublevel
- 08 address of the receiver at second sublevel

Command telegram of the master

The master sends its command telegram with all addresses to the router at main level.

Telegram header (standard)								Telegram header (expanded)		User data		
StZ	**CS**	**DatL**	**DatLW**	**Cntrl**	Service	Adr.S	Adr.E	Sub-addr. 1	Sub-addr. 2			
02	**E1**	**05**	**05**	**02**	03	10	03	02	08	F0	01	00

Fig. 2-10: Command telegram of the master

Routing at main level

The receiver at main level recognizes with the Cntrl byte that the telegram must be routed. The receiver

- saves locally at least the address of the transmitter (but it would be better to save the whole address list because of possible network configurations).
- shifts the first 6 telegram bytes by one byte backwards. So it automatically eliminates the old (deposited dark-grey) address of the transmitter. To the place there's now the own (deposited light-grey) address and the first sub address becomes the new receiver address. The expanded telegram header only consists of one sub address now.
- adjusts the Cntrl byte, the data length and the check sum to the new telegram (shortened by one byte).

Telegram header (standard)								(expan-ded)	User data		
StZ	**CS**	**DatL**	**DatLW**	**Cntrl**	Service	Adr.S	Adr.E	Sub-addr. 1			
02	**F4**	**04**	**04**	**01**	03	03	02	08	F0	01	00

Fig. 2-11: Command telegram of the router at main level

Routing at first sublevel

The receiver at first sublevel recognizes that this telegram must be routed. It executes the same routing algorithm than the router at main level. After this routing there's no expanded telegram header available any more.

Telegram header (standard)								User data		
StZ	**CS**	**DatL**	**DatLW**	**Cntrl**	Service	Adr.S	Adr.E			
02	**FA**	**03**	**03**	**00**	03	02	08	F0	01	00

Fig. 2-12: Command telegram of the router at first sub level

Response telegram at second sublevel

The receiver at second sublevel identifies itself with the Cntrl byte as final recipient and treats the received telegram. As answer, it sends the response telegram back to the master.

Telegram header (standard)								User data		
StZ	**CS**	**DatL**	**DatLW**	**Cntrl**	Service	Adr.S	Adr.E			
02	**E9**	**03**	**03**	**10**	03	02	08	00	01	F1

Fig. 2-13: Response telegram of the slave at second sub level

Routing at first sublevel

The router in the first sublevel recognizes with the Cntrl byte that a response telegram must be send back. The router

- compares the complete address list, if it has stored it before.
- shifts the first 6 bytes by one byte forward and enters the saved address (deposited light-grey) at a free position. The telegram header has been expanded by one byte again.
- adjusts the Cntrl byte, the data length and the check sum to the new telegram (expanded by one byte) and sends it to the participant at main level.

Telegram header (standard)								(expan-ded)	User data		
StZ	**CS**	**DatL**	**DatLW**	**Cntrl**	Service	Adr.S	Adr.E	Sub-addr. 1			
02	**E3**	**04**	**04**	**11**	03	03	02	08	00	01	F1

Fig. 2-14: Response telegram of the router at first sub level

Routing at main level

The receiver at main level also recognizes, that a response telegram must be routed. It executes the same routing algorithm than the router in the first sublevel. After this routing the expanded telegram header contains both subaddresses again.

Telegram header (standard)								Telegram header (expanded)		User data		
StZ	**CS**	**DatL**	**DatLW**	**Cntrl**	Service	Adr.S	Adr.E	Sub-addr. 1	Sub-addr. 2			
02	**D0**	**05**	**05**	**12**	03	10	03	02	08	00	01	F1

Fig. 2-15: Response telegram of the router at main level

Aspects of the Specification of the Telegram Header

Addressing the Telegram

It makes sense to enter all addresses in succession in the telegram header. The sender's address should be first, followed by the address of the main recipient. With subaddressing the subaddresses are in the sequence of the individual sublevels.

This specification is optimal for the proposed telegram forwarding path.

Routing

Every participant itself is responsible for the correct forwarding of command and response telegram.

Forwarding a Command Telegram

The algorithm for forwarding a command telegram influences the content and the length of the telegram header:

When the command telegram is forwarded, the address block is reduced step-by-step.

During this process, the recipient - unless it is the final recipient - notes the sender's address internally and copies its own and the additional addresses to the beginning of the address block. Its own address becomes the sender's address and the next address becomes the recipient's address.
When a response telegram is forwarded, the stored address of the former sender is added again.

Advantages:

- Minimizes telegram overhead.
- In all telegrams, at all levels, the addresses of the current sender and receiver are in the same position in the telegram header.

Disadvantages:

- Apart from the change required in a subaddress index and the telegram checksum, the length bytes must also be corrected and the addresses must be relocated.
- The management of forwarded telegrams and incoming response telegrams can be very demanding on resources for devices that may have to control a number of sublevels.

The address block is at the end of the telegram header, because it has a variable length in the case of subadressing.

Forwarding a Response Telegram

The collection of all subaddresses is not really necessary for forwarding a response telegram up to the point of initiating communication.

However, if an error occurs (addressed user does not answer), the subaddresses for the return path must be included in the telegram to enable the exact error position to be located.

When the response telegram is forwarded, the address block is reconstituted step-by-step.

Sequential Telegram Number

A sequential telegram number for differentiating every single transmission of a telegram transmitted cyclically is optional. Because this option influences the length of the telegram header, a sequential telegram number belongs in the flexible section of the telegram header and is attached after the address block.

Managing the Flexible Telegram Header and the Telegram Traffic

A control byte manages all telegram traffic. It provides the following information:

- Number of additional subaddresses (3 bits)
- Additional byte for a sequential telegram number (1 bit)
- Telegram property (1 bit)
- System status (3 bits)

The system status indicates that an error or warning occurred in the system from which the telegram was sent.

The control byte has a fixed position in the telegram header, because it is changed when being forwarded.

Services

The services are divided into

- General services
- Group-specific services

The general services must be supported by all SIS users. This particularly refers to the services involved in establishing SIS communications.

All new services must (so far as possible) be set up so they are generally available.

Note: Because the token is forwarded over its own service in multimaster systems, this byte should also have a fixed position at the beginning of the telegram header.

For reasons of compatibility with the currently available communications protocols used by the individual development groups and departments, group-specific services are provided. This makes it possible to place existing protocols (that cannot be converted to SIS within an acceptable period) in a SIS shell.

The appropriate development group is responsible for releasing the services within a group. Descriptions of the released group services must be established and maintained within the groups.

Modified Telegram Length

Because there may be devices that cannot process telegrams but have to forward them, the length information in the telegram header refers not only to the useful data but also to the variable section of the telegram header. This modified telegram length - it does not include the fixed section of the telegram header - has a fixed position in the fixed section of the telegram header.
Otherwise, the associated 'forwarding program' would have to be intelligent enough to be able to calculate the current length of the telegram header and therefore the current telegram length itself.

The actual telegram length is the result adding the modified telegram length and the (fixed number of) fixed data bytes of the telegram header.

Repeating the Modified Telegram Length

To increase data security, the length information is repeated in the variable Profibus protocol.

Telegram Checksum

The checksum calculation specified for the serial interface enables the telegram checksum to be posted at any position after a start byte. Because this checksum has to be changed according to the procedure described above for forwarding telegrams, it is placed at a fixed position at the beginning of the telegram header.

Start Character

The telegram header begins with a specified start character.

Response Time for Communication Request

No minimum answer period or time-outs are specified in the protocol definition. However, it must noted that the transmission path may not be blocked for any reason.

Therefore, for communication requirements whose processing cannot be completed quickly, corresponding states are added in the control byte of the telegram header.

For this purpose, the states "transmission request is being processed" and "transmission request cannot be accepted at present" seem to make sense.

The particular transmitting master has to decide which responses should follow these acknowledgments.

2.3 Telegram Content

The user data header and the user data combine to form the telegram content.

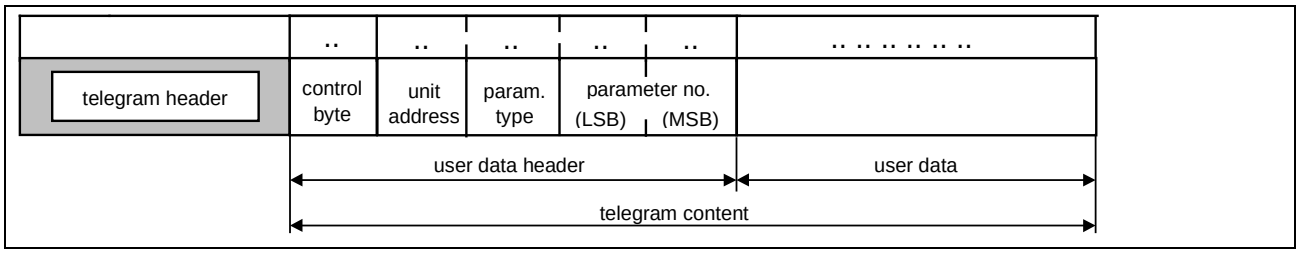


Fig. 2-16: Structure telegram content of the command telegram

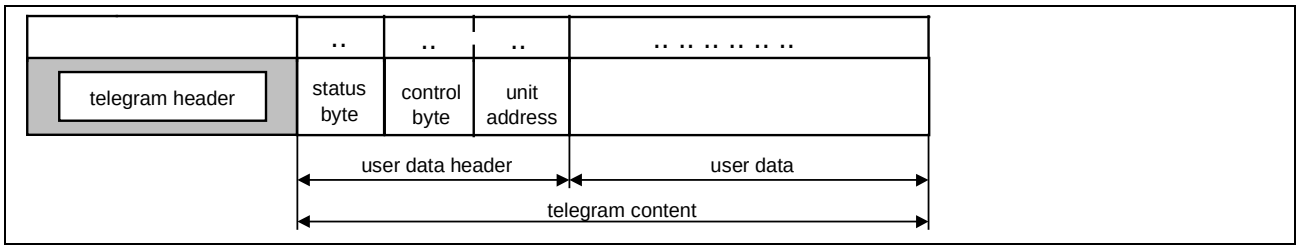


Fig. 2-17: Structure telegram content of the response telegram

User Data and User Data Header

User Data Header

The user data header specifies the SIS service more precisely. It contains information such as:

- Subservice
- Internal addresses
- Element of a SERCOS parameter
- Commands
- Control and status information

Note: The user data header may be omitted with read accesses if only the access is specified from the requested service (e.g. SIS service 0x01 data transmission aborted).

The exact description of the user data header may be found in the following chapters 3 "User Identification via SIS", page 3-1 to chapter 9 "Token Passing via SIS", page 9-1.

User Data

The user data is the actual data to be transmitted. User data is not included in the following cases:

- Read accesses in the command telegram
- Successful write accesses in the response telegram.

The exact description of the user data may be found in the following chapters 3 "User Identification via SIS", page 3-1 to chapter 9 "Token Passing via SIS", page 9-1.

Note: The user data is (as with the user data header) dependent on the SIS service.

Error Codes in the Status Byte

The first byte in the user data header of a response telegram is the status byte.

If the status byte has the value 0, the service requested by the command telegram service has been successfully executed, otherwise an error has occurred.

In the case of an error, two types of errors are distinguished:

- Telegram error
- Execution error

Telegram Error

In a telegram error the requested service is not executed. Only the data from the user data header in the response telegram, but no user data is sent. The error codes 0x0F0 - 0x0FF are available in the status byte to identify the various telegram errors.

Execution Error

If the requested service cannot be executed without error, a service-specific error code must be sent in the response telegram in place of the user data and the status byte must be set to the value 0x01 or 0x02.

List of Codes in the Status Byte

Code Number	Description	Error Type
0x00	Error-free transmission	-
0x01	"Error while executing the telegram": An error occurred during the execution of the requested service. The service-specific error code is in the user data of the response telegram.	Execution error
0x02	"Error in the (internal) transmission channel": An error occurred while accessing the (internal) transmission channel. The specific error code (see Fig. 2-19) is in the user data of the response telegram.	Execution error
0xF0	"Invalid service": The requested service is not specified or is not supported by the addressed user.	Telegram error
0xF1	"Invalid telegram": The telegram cannot be evaluated because, for example, a slave received a response telegram from the master or the start character was not found.	Telegram error
0xF2	"Telegram length error": The two length entries in the telegram do not match.	Telegram error
0xF4	"Checksum error": The transmitted checksum does not match the one calculated internally.	Telegram error
0xF8	"Invalid sequential telegram": Data in the user data header, the sender address or the service has changed in the sequential telegram.	Telegram error

Fig. 2-18: List of codes in the response telegram status byte

List of Specific Error Codes in the Transmission Channel

Error Codes	Error Message and Response
0x8001	"Transmission channel currently busy (BUSY)". The desired access is currently not possible, because the transmission channel is busy processing the request. *The command telegram is repeated until the transmission channel is available again.*
0x8002	"Transmission Channel Error" . The request cannot be forwarded to the desired user. *Possibly repeat the command telegram, to test for a long-term transmission channel problem.*
0x800B	"Transmission aborted (because of the higher priority of another request)". *Repeat the command telegram until it is executed without being aborted.*
0x800C	"Access denied (transmission channel is currently active)". A new request was started, before the last transmission was completed. *Repeat the command telegram until the active request is completed.*

Fig. 2-19: Specific error messages in the transmission channel

2.4 Transmission Sequence

The communication master initiates the data transmission by sending a command telegram. The communication slave processes the request. It sends a response telegram if it is not a communication to several slaves simultaneously.

Note: Command telegrams may only be sent by the master that has the send permit (token) in a multi-master system. The forwarding of this token to another master is described in the chapter on "Token Passing via SIS", page 9-1.

The data transmission sequence depends primarily on the following:

- The length of the user data
- The timing of the serial interface

Transmission of Short Data Records

A high proportion of data transmissions consist of short data records, which can be transmitted in one telegram.

Write Access Write access with only one telegram is executed as follows:

RS485 Sender		Hardware		RS485 Recipient
Interpreter <=>	Prot. Driver	<== RS485 ==>	Prot. Driver	<=> Interpreter
----------	----------	SEND telegram	----------	-------->
<---------	----------	Transmission status (+ error code)	----------	---------

Fig. 2-20: Write access with short data record

Read Access Read access with only one telegram is executed as follows:

RS485 Sender		Hardware		RS485 Recipient
Interpreter <=>	Prot. Driver	<== RS485 ==>	Prot. Driver	<=> Interpreter
----------	----------	FETCH telegram	----------	-------->
<---------	----------	Transmission status + (error code or requested data)	----------	---------

Fig. 2-21: Read access to a short data record

Transmission of Long Data Records

If the number of the variable telegram data exceeds 255 bytes, sequential telegrams are transmitted in all telegram directions and types to send or retrieve the complete data record.

Transmission with sequential telegrams is controlled by the mechanism installed in the user data header. This mechanism is in principle dependent on the selected SIS service.

Write Access

Write access with sequential telegrams is executed as follows:

RS485 Sender		Hardware		RS485 Recipient
Interpreter <=>	Prot. driver	<== RS485 ==>	Prot. driver	<=> Interpreter
----------	----------	SEND telegram	----------	-------->
<---------	----------	Transmission status (+ error code)	----------	---------
----------	----------	First sequential SEND telegram	----------	-------->
<---------	----------	Transmission status (+ error code)	----------	---------
. . .	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .
----------	----------	Last sequential SEND telegram	----------	-------->
<---------	----------	Transmission status (+ error code)	----------	---------

Fig. 2-22: Write access with long data record

Read Access Read access with sequential telegrams is executed as follows:

RS485 Sender		Hardware		RS485 Recipient
Interpreter <=>	Prot. driver	<== RS485 ==>	Prot. driver	<=> Interpreter
----------	----------	FETCH telegram	----------	-------->
<---------	----------	Transmission status + (error code or requested data)	----------	---------
----------	----------	First sequential FETCH telegram	----------	-------->
<---------	----------	Transmission status + (error code or requested data)	----------	---------
. . .	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .
----------	----------	Last sequential FETCH telegram	----------	-------->
<---------	----------	Transmission status + (error code or requested data)	----------	---------

Fig. 2-23: Read access to a long data record

Time-Out Periods

Some time-out periods for the sequence of a data transmission should be noted.

To configure these periods and to set default values, see the chapter on "Timing for the SIS Interface, page 10-1" and "Initializing Communication via SIS", page 6-1.

Behavior in Case of Error

Master If the master receives a response telegram that has an error on the hardware or protocol level, it first needs to allow the master repetition period (TwM) to expire. After that, it can - as also in the case of a response telegram that does not arrive - repeat the command telegram.

Note: Because of token passing, the number of repetitions cannot be specified.

Slave

If the slave receives a command telegram with hardware or protocol errors, the following applies:

- For a **point-to-point connection**, the slave lets the bus wait time go by. It then sends the response telegram with transmission status and error code.
- For **SIS communication via RS485** bus the slave does not answer, because the validity of the addresses – especially that of the sender – is not guaranteed.

Note: Because of optional multi-master operation, two monitoring processes that would be feasible for a single master system do not take place: on one hand the slave does not monitor the number of repetitions, on the other hand there is no time-out period for repetitions.

2.5 Physical Data Transmission

Electrical Basis

Physical data transmission on the serial interface can be as follows:

- RS485: Bus and point-to-point connection
- RS422: Bus (max. 10 users) and point-to-point connection
- RS232: Point-to-point connection with/without HW handshake

Port Assignment

The following settings are specified:

- Parity: **E**,
- Data Bits: **8**,
- Stop Bits: **1**.

A baud rate of at least **9600** must be supported.

Further information regarding the baud rates may be found in the chapters on "Timing for the SIS Interface, page 10-1" and "Initializing Communication via SIS", page 6-1.

3 User Identification via SIS

This service enables a communication master to obtain information about its communication slaves.

Required Service for all Users

This service is a required service for all SIS users, with the exception of subservice 0x02. Subservice 0x02 cannot yet currently be supported by most users.

3.1 SIS Service 0x00 User Identification

Command Telegram

The 0x00 service command telegram consists of:

- Telegram header
- Address (**byte 0** of the user data header)
- Subservice (**byte 1** of the user data header)

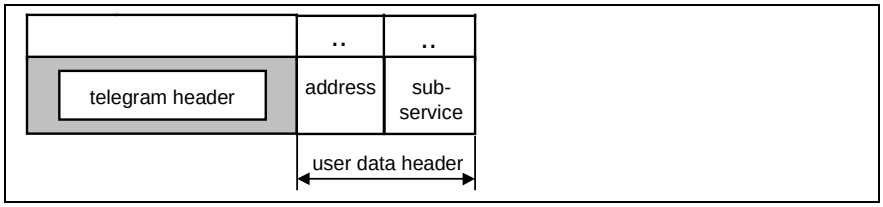


Abb. 3-1: Command telegram of service 0x00

Response Telegram

The 0x00 service response telegram after successful execution consists of:

- Telegram header
- Status (**byte 0** of the user data header) **= 0**
- Address (**byte 1** of the user data header)
- Subservice (**byte 2** of the user data header)
- User data

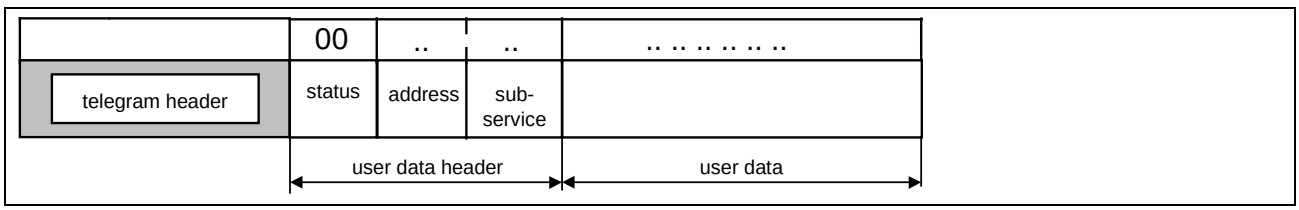


Fig. 3-2: Response telegram of service 0x00

The telegram header and the first 2 or 3 bytes of the user data header are fixed components of the command or response telegram.

The user data dependent on the subservice are explicitly listed in the following subsections.

Subservice 0x01 Read Out SIS Version

Function Reads out the SIS version to identify users.

Command Telegram User data:

None

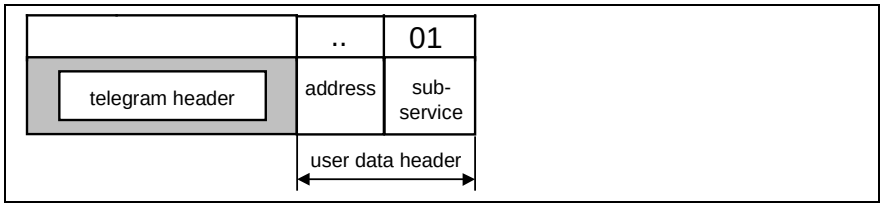


Fig. 3-3: Command telegram of subservice 0x01

Response Telegram User data:

Byte 3 - 7: SIS version string with format "**nnVmm**"

(nn = 2-digit version number, mm = 2-digit release date)

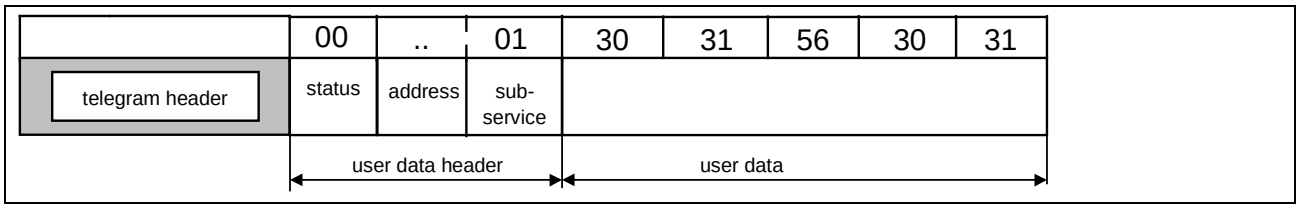


Fig. 3-4: Response telegram of subservice 0x00 (e.g., SIS version "01V01")

Error Codes Error code not present

Subservice 0x02 Read Out FWA Number

Function Readout of the FWA number (description of the product firmware) for user-identification.

Command Telegram User data:

None

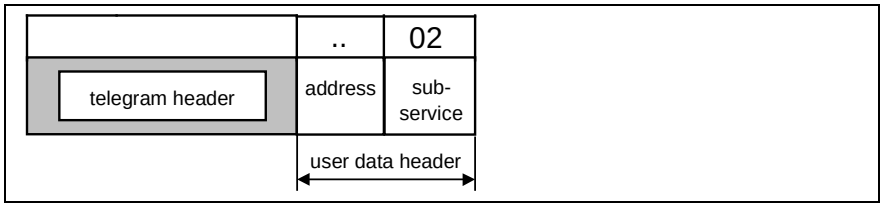


Fig. 3-5: Command telegram of subservice 0x02

Response Telegram User data:

Byte 3 - 42: max. 40-digit FWA number according to INN norm

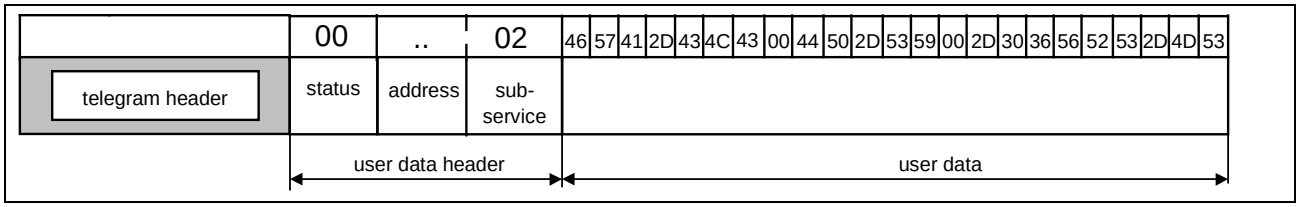


Fig. 3-6: Response telegram of subservice 0x02 (e.g., firmware "FWA-CLC*DP-SY*-06VRS-MS")

Error Codes Error code not present

Subservice 0x03 Read Out Unit Typecode

Function Read out the unit typecode (hardware description) for user-identification.

Command Telegram User data:

None

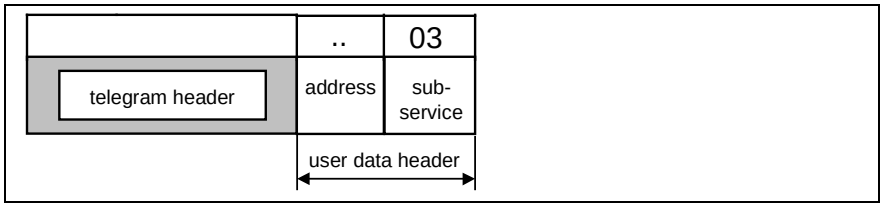


Fig. 3-7: Command telegram of subservice 0x03

Response Telegram User data:

Byte 3 - 42: max. 40-digit typecode according to INN norm

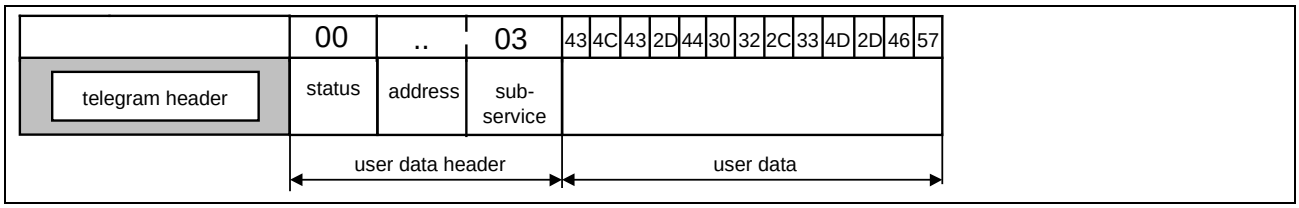


Fig. 3-8: Response telegram of subservice 0x03 (e.g., hardware "CLC-D02.3M-FW")

Error Codes Error code not present

Subservice 0x04 Read Out Supported Baud Rates

Function Read out supported user baud rates.

Command Telegram User data:

None

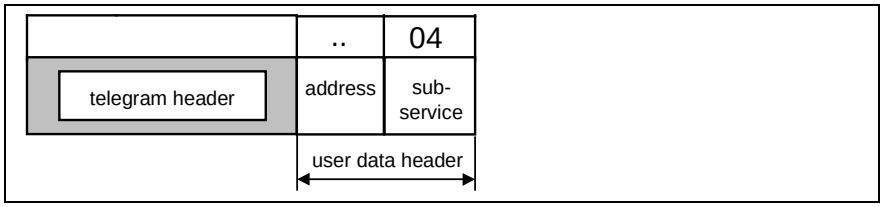


Fig. 3-9: Command telegram of subservice 0x04

Response Telegram User data:

Byte 2: Supported baud rates according to baud rate mask: For each supported baud rate higher than the default rate (9600), a bit is ORed into the user data byte according to the baud rate mask.

Baud Rate Mask	Baud Rate
00000000	9600
00000001	19200
00000010	38400
00000100	57600
00001000	115200

Fig. 3-10: Baud rate mask

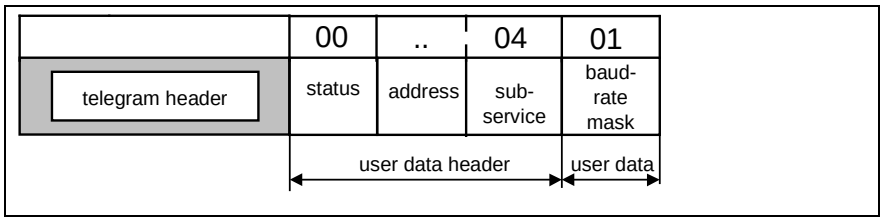


Fig. 3-11: Response telegram of subservice 0x04 (e.g., supporting default baudrate (9600) and 19200)

Error Codes Error code not present

3.2 Overview of SIS Service 0x00 User Identification

SIS Subservice	No.	Additional Data in the Command Telegram	User Data in the Response Telegram
Read out SIS version	0x01	-	Version in the format "nnVmm"
Read out FWA names	0x02	-	max. 40-digit FWA number according to INN Norm
Read out unit typecode	0x03	-	max. 40-digit typecode according to INN norm
Read out supported baud rates	0x04	-	Bit pattern of the supported baud rates

Fig. 3-12: Overview of SIS subservices with reference to 0x00 service

4 Termination of a Data Transmission via SIS

If a communication master is prevented by higher priority tasks or by disruption (e.g. abnormal termination) from continuing normal data transmission during a data transmission, it may terminate a transmission in progress.

Required Service for all Users

This service is a **required service** for all SIS users.

4.1 SIS Service 0x01 Termination of Data Transmission

Command Telegram

The 0x01 service command telegram consists of:

- Telegram header
- Device address (1 byte user data header)
- Service to be terminated (1 byte user date)

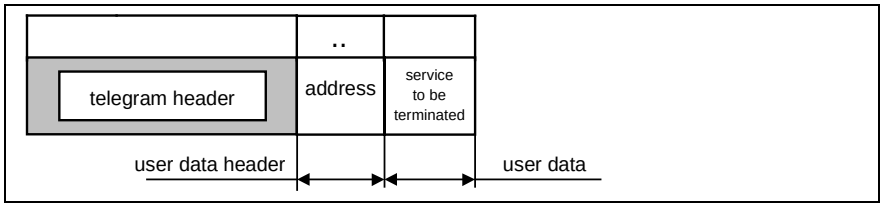


Fig. 4-1: Command telegram of service 0x01

The information on the device address and/or the service to be terminated is required if an SIS final user manages several devices and/or several services in parallel.

If the data transmission that is to be terminated can be uniquely assigned, the user data header and/or the user data are not required.

Note: Where several devices and/or services are managed in parallel, all parallel data transmissions can also be simultaneously terminated with the 0x01 service if the telegram header only is sent.

Response Telegram

The 0x01 service response telegram consists of:

- Telegram header
- Status byte (**byte 0** of the user data header)
- Device address (**byte 1** of the user data header)
- User data (in the case of execution error only)

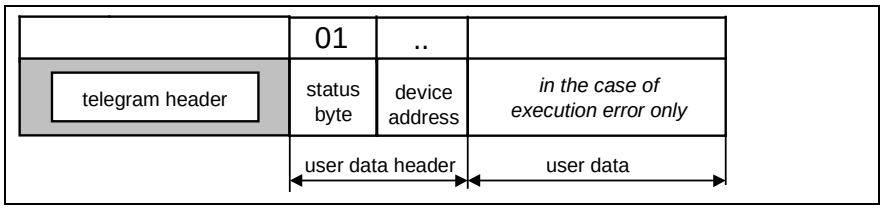


Fig. 4-2: Response telegram of service 0x01

If a device address is not given in the command telegram, it is also not included in the response telegram.

Note: If no sequential telegrams are processed and this service is sent in spite of that, the SIS user does not need to execute this service. In this case, user data are not sent in the response telegram because this service is permitted in every phase of data transmission.

5 Flash EPROM Programming via SIS

The 0x02 service is provided to implement the SIS loader; the information on a subservice in the second byte of the user data header specifies the flash operation that is to be executed.

The ‘Read Flash’ and ‘Program Flash’ subservices also enable data access to any storage media, such as RAM, dual port RAM or EEPROM, to be implemented.

5.1 SIS Service 0x02 Flash Operation

Command Telegram

The 0x02 service command telegram consists of:

- Telegram header
- Address (**byte 0** of the user data header)
- Subservice (**byte 1** of the user data header)
- Additional user data header bytes (dependent on the subservice)
- User data (write accesses only)

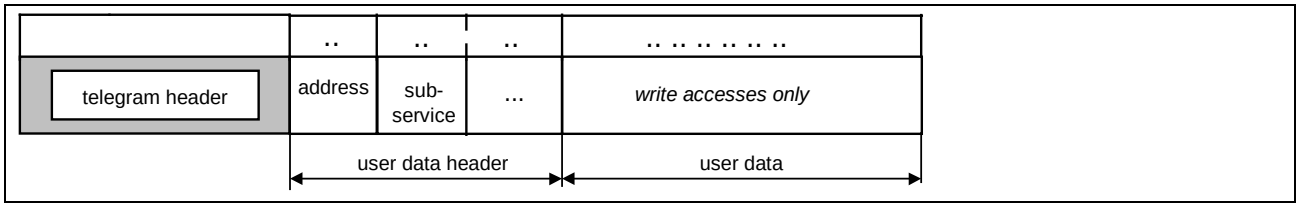


Fig. 5-1: Command telegram of service 0x02

Depending on the subservice, additional specifying data may have to be input in the user data header.

The user data follows the user data header in subservices with write access (e.g. ‘Program Flash’).

Response telegram after Successful Execution of the Service

The 0x02 service response telegram after successful execution consists of:

- Telegram header
- Status (**byte 0** of the user data header) **= 0**
- Address (**byte 1** of the user data header)
- Subservice (**byte 2** of the user data header)
- User data (read accesses only)

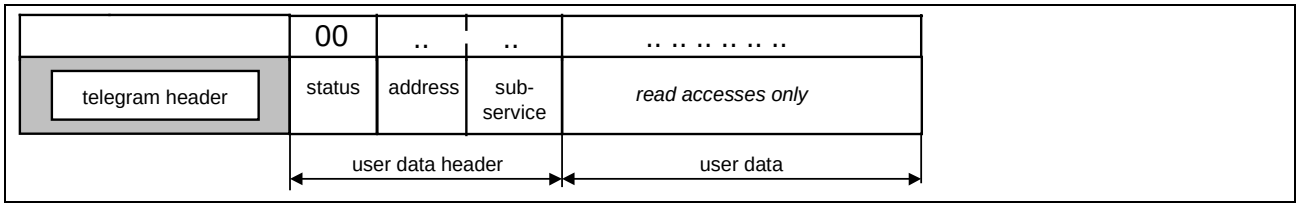


Fig. 5-2: Response telegram of service 0x02

The user data follows the user data header in subservices with read access (e.g. ‘Read Flash’).

Response telegram in Case of Error

In case of an error, an error byte is written into the response telegram in place of the user data.

- Telegram header
- Status (**byte 0** of the user data header) **= 1** (see Fig. 2-18)
- Address (**byte 1** of the user data header)
- Subservice (**byte 2** of the user data header)
- Error byte

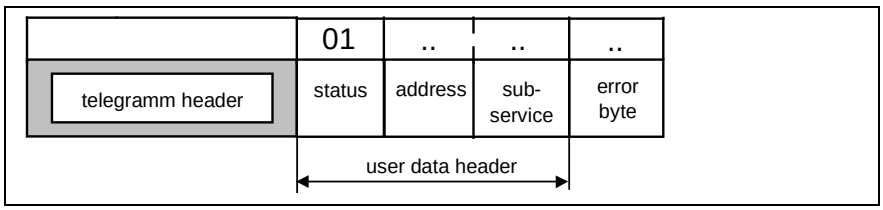


Fig. 5-3: Response telegram of service 0x02 in case of an error

Error Code

Because the error bytes depend on the subservice, it makes sense to generate a unique error code beyond the telegram structure. For this purpose, the subservice (user data header) and the error byte (user data) are combined into one sequential datum 2 bytes in length.

The telegram header and the first 2 or 3 bytes of the user data header are fixed components of the command or response telegram.

The data dependent on the subservice (expanded user data header and user data) are explicitly listed in the following subsections.

Subservice 0x90 Shutdown

Function

Shutdown, stop system, (optional restart), depending on the command selection in the firmware or branching in the load routine.

Command Telegram

Additional information in user data header:

Byte 2: Command selection
0 = branching in firmware
1 = branching in load routine

Byte 3 - 6: Start address

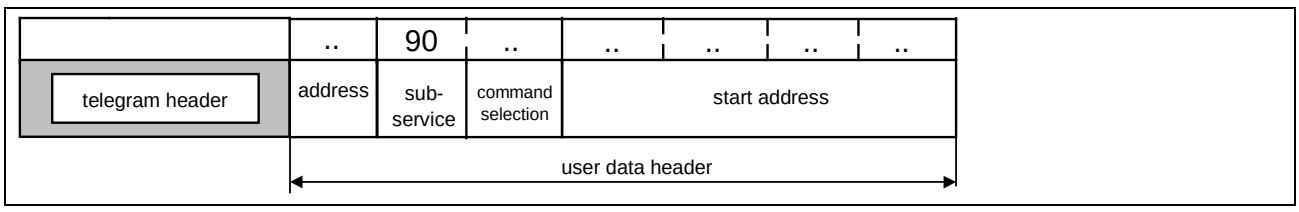


Fig. 5-4: Command telegram of subservice 0x90

Response Telegram User data:

None, error code, if applicable

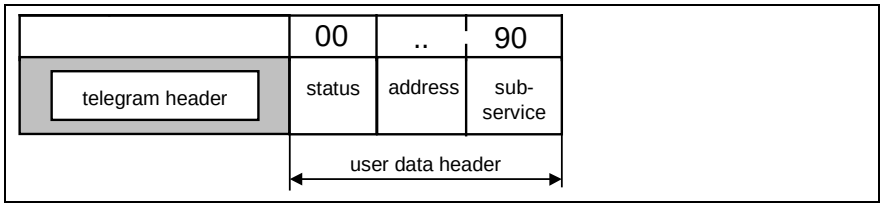


Fig. 5-5: Response telegram of subservice 0x90 in error-free case

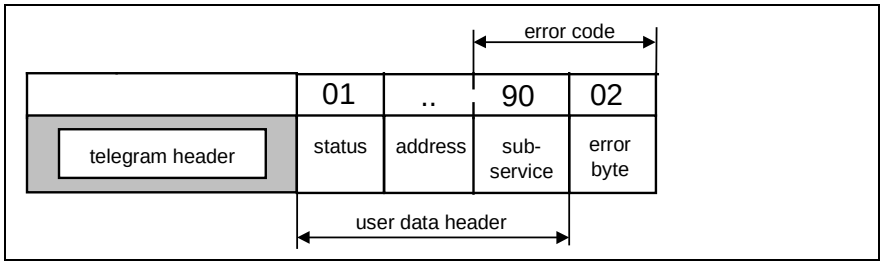


Fig. 5-6: Response telegram of subservice 0x90 in case of an error (e.g., 0x9002 firmware was deleted)

Error Codes

0x9002	Firmware was deleted
0x9003	Shutdown in phase 3 not permitted
0x9004	Shutdown in phase 4 not permitted

Subservice 0x91 Reboot

Function Shutdown, stop system, restart and branching in firmware.

Command Telegram Additional information in user data header:

Byte 2 - 5: Start address

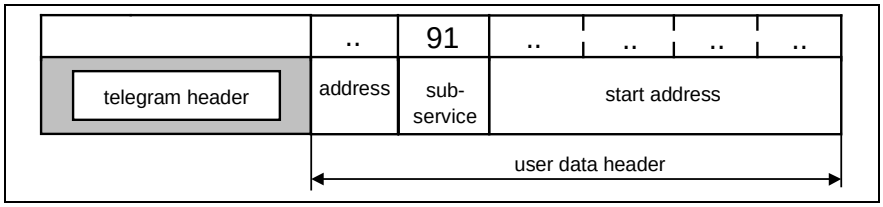


Fig. 5-7: Command telegram of subservice 0x91

Response Telegram User data:

None, error code, if applicable

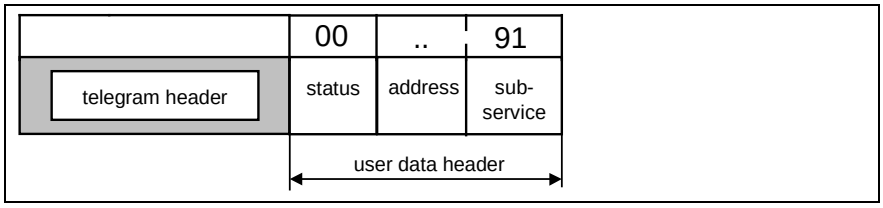


Fig. 5-8: Response telegram of subservice 0x91 in error-free case

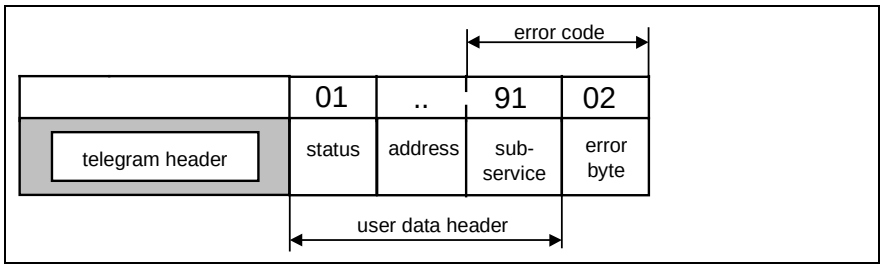


Fig. 5-9: Response telegram of subservice 0x91 in case of an error (e.g., 0x9102 firmware was deleted)

Error Codes

0x9102	Firmware was deleted
0x9103	Reboot in phase 3 not permitted
0x9104	Reboot in phase 4 not permitted

Subservice 0x92 Read Flash

Function Read out a memory range (Flash, RAM, DPR, EEPROM ...)

Command Telegram Additional information in user data header:

Byte 2 - 5: Source address

Byte 6: Length (max. 244 bytes)

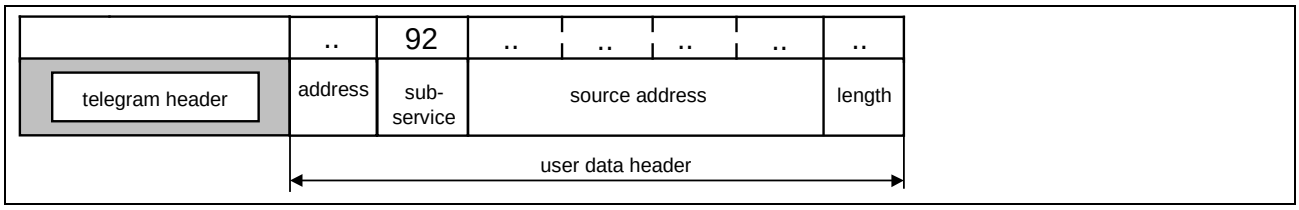


Fig. 5-10: Command telegram of subservice 0x92

Response Telegram User data:

Byte 3 - 246: Data read, or error code, if applicable

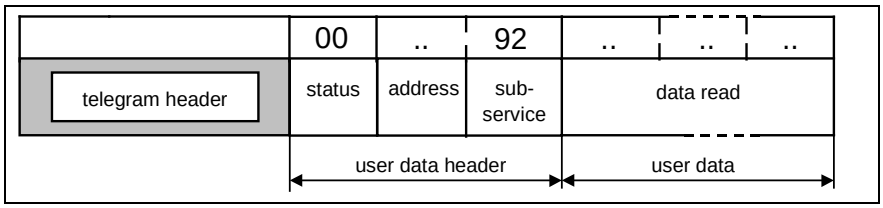


Fig. 5-11: Response telegram of subservice 0x92 in error-free case

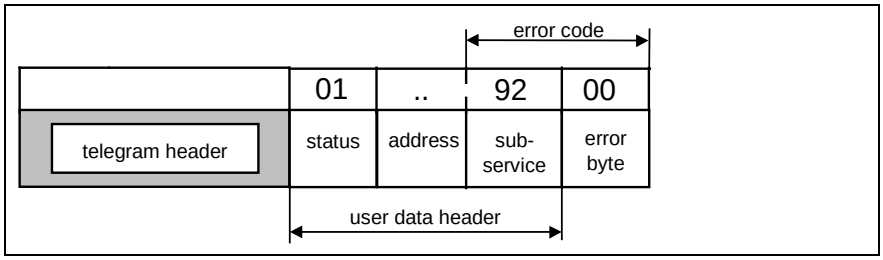


Fig. 5-12: Response telegram of subservice 0x92 in case of an error (e.g., 0x9200 error during read)

Error Codes

0x9200	Error during read

Subservice 0x93 Find Header

Function Determines the header address of the base module

Command Telegram Additional information in user data header:

None

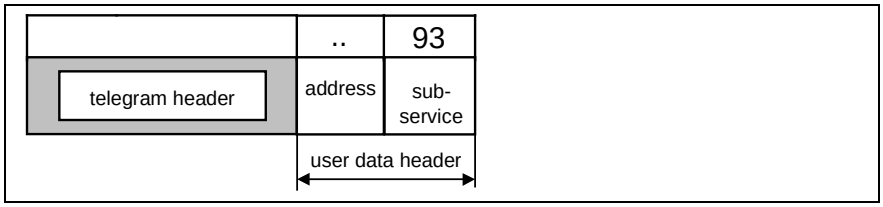


Fig. 5-13: Command telegram of subservice 0x93

Response Telegram User data:

Byte 3 - 6: Start address

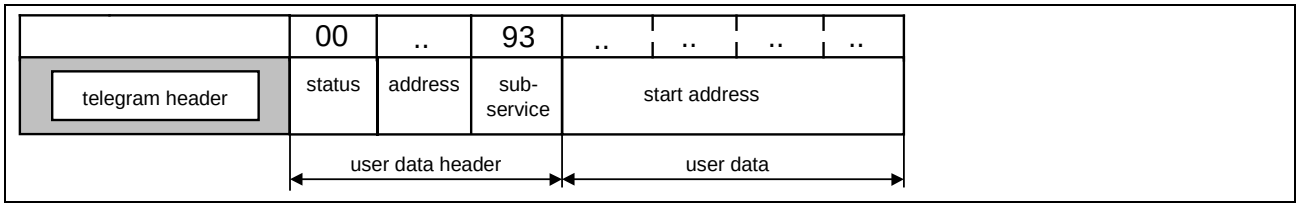


Fig. 5-14: Response telegram of subservice 0x93

Error Codes Error code not present

Subservice 0x94 Erase Flash

Function Initialization and start of the deletion process and check of the address range

Command Telegram Additional information in user data header:

- for complete deletion without loader and kernel

Byte 2 - 5: 0xFF FF FF FF (long)

Byte 6 - 9: 0xFF FF FF FF (long)

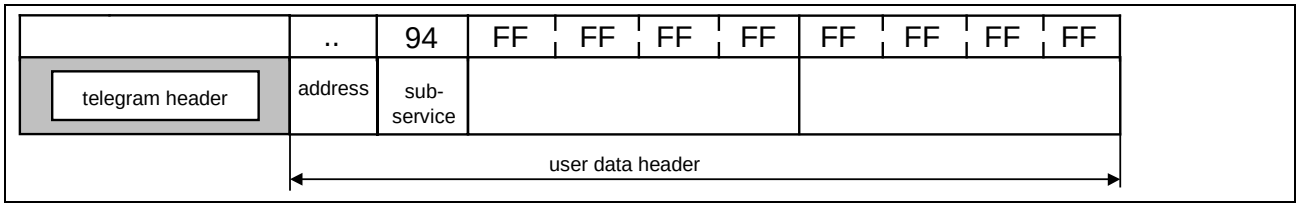


Fig. 5-15: Command telegram of subservice 0x94 (for complete deletion without loader and kernel)

- for sector deletion

Byte 2 - 5: Start address

Byte 6 - 9: Block length (long)

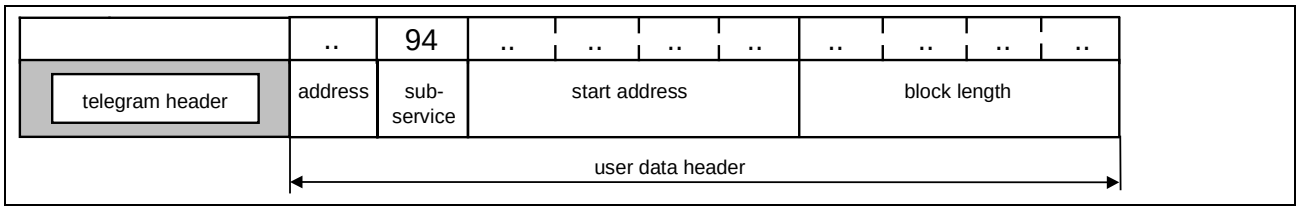


Fig. 5-16: Command telegram of subservice 0x94 (for sector deletion)

Response Telegram User data:

None, error code, if applicable

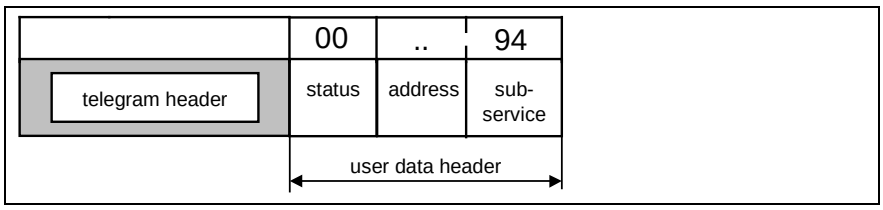


Fig. 5-17: Response telegram of subservice 0x94 in error-free case

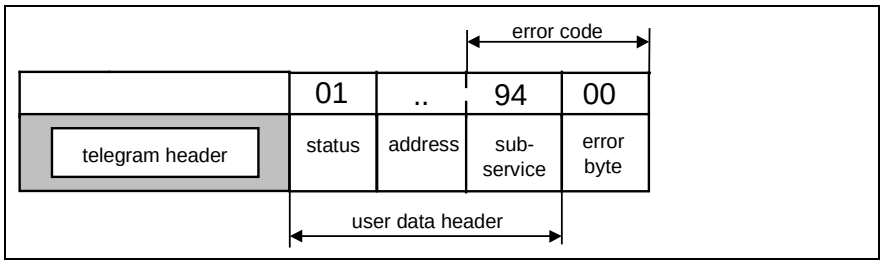


Fig. 5-18: Response telegram of subservice 0x94 in case of an error (e.g., 0x9400 time-out during deletion)

Error Codes

0x9400	Time-out during deletion
0x940A	Deletion possible in loader only

Subservice 0x96 Program Flash

Function Programming the software, writing data into a data storage medium.

Command Telegram Additional information in user data header:

Byte 2 - 5: Target address

Byte 6: Device type

User data:

Byte 7 - 246: Data (max. 240 byte code per telegram)

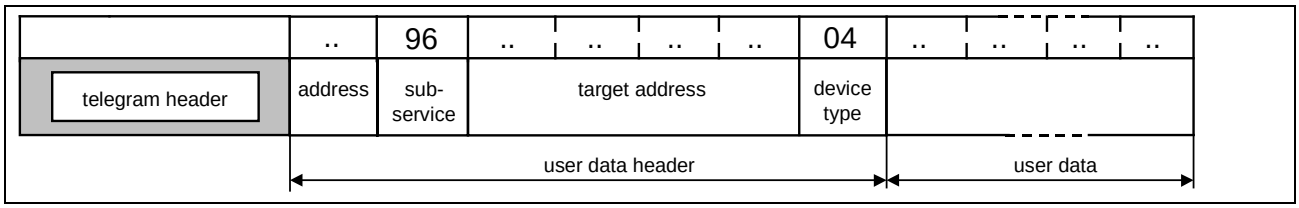


Fig. 5-19: Command telegram of subservice 0x96

Device type:

MEM_NO_TYPE	=	0x00	(not supported)
MEM_ROM	=	0x01	(not supported)
MEM_RAM	=	0x02	
MEM_DPR	=	0x03	
MEM_FLASH	=	0x04	

The device type must be firmly set to MEM_FLASH for flash programming.

Response Telegram

User data:

None, error code, if applicable

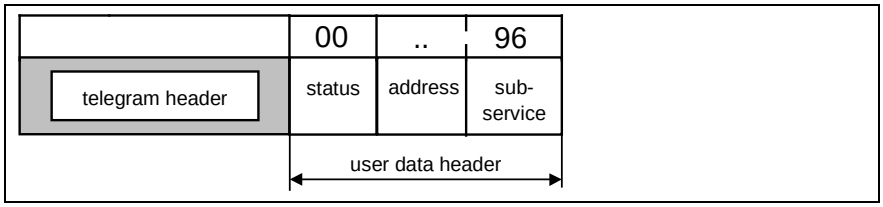


Fig. 5-20: Response telegram of subservice 0x96 in error-free case

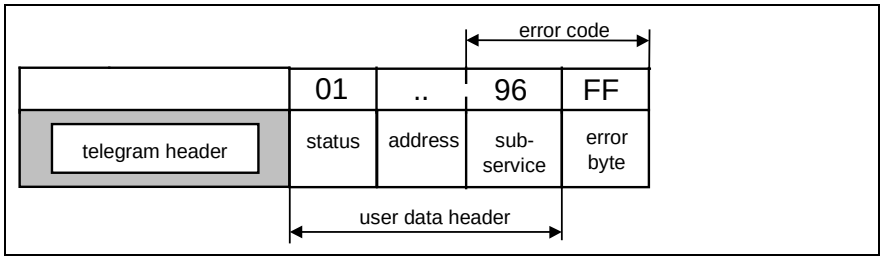


Fig. 5-21: Response telegram of subservice 0x96 in case of an error (e.g., 0x96FF error during write to RAM)

Error Codes

0x96FF	Error during write to RAM
0x96E0	Verify error during programming of Flash
0x96E1	Time-out during programming of Flash

Subservice 0x97 Build Checksum

Function

Internal calculation of the checksums of a module and comparison with those in the module.

The last 6 bytes of a module have a CRC32 checksum (4 bytes) of all user data (module length - 6 bytes) and the negated addition checksum (2 bytes) of all user data and the CRC32 checksum. The addition checksum of the full module length therefore always yields the value 0.

Command Telegram

Additional information in user data header:

Byte 2 - 5: Start address of the module

Byte 6 - 9: Length of the module (in Bytes)

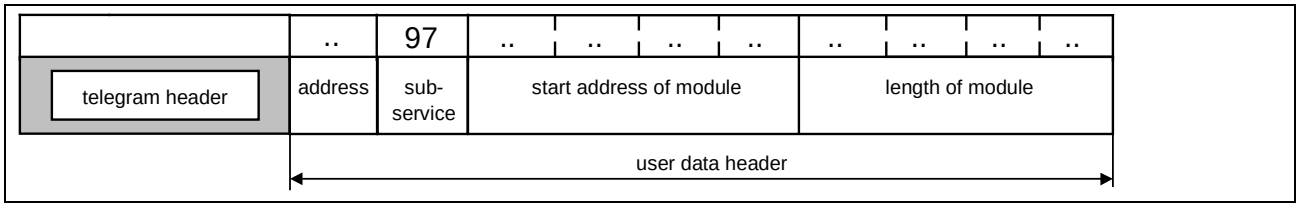


Fig. 5-22: Command telegram of subservice 0x97

Response Telegram

User data:

None, error code, if applicable

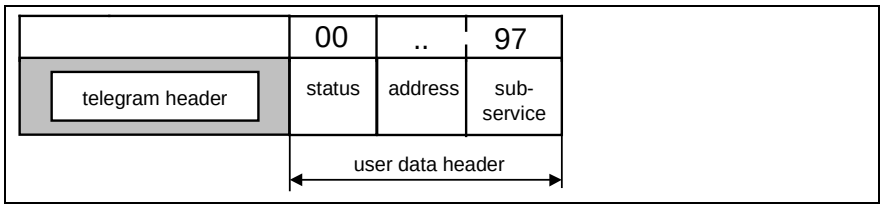


Fig. 5-23: Response telegram of subservice 0x97 in error-free case

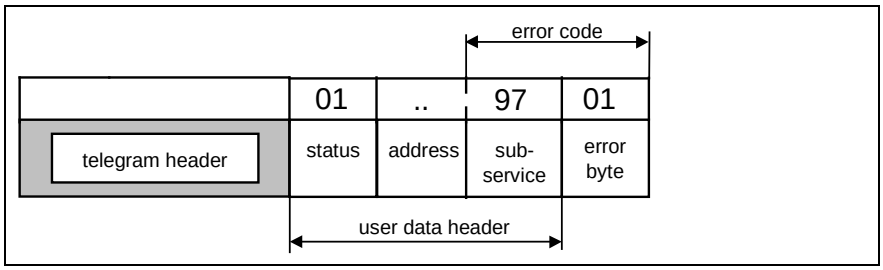


Fig. 5-24: Response telegram of subservice 0x97 in case of an error (e.g., 0x9701 addition checksum error)

Error Codes

0x9701	Addition checksum error
0x9702	CRC32 checksum error

Subservice 0x9F Error Reset in the Slave System

Function Deletes error and diagnosis messages displayed in the slave system after an error in flash operations (0x02 service).

Command Telegram Additional information in user data header:

None

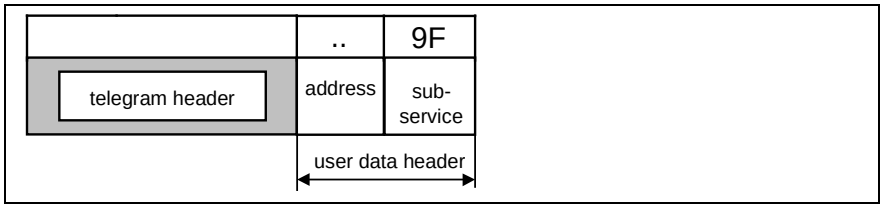


Fig. 5-25: Command telegram of subservice 0x9F

Response Telegram User data:

None

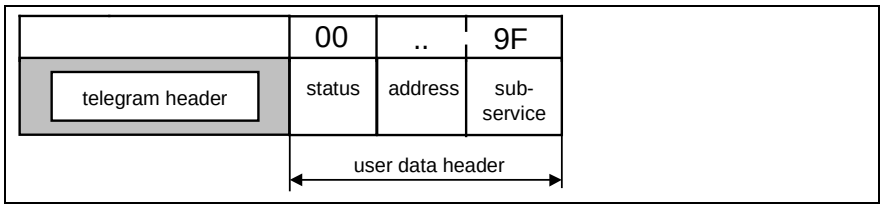


Fig. 5-26: Response telegram of subservice 0x9F

Error Codes Error code not present

5.2 Overview of SIS Service 0x02 Flash Operations

SIS Sub-Service	No.	Additional Data in the Command Telegram	User Data in the Response Telegram
Shutdown	0x90	Firmware/loader Start address	Error code, if applicable
Reboot	0x91	Start address	Error code, if applicable
Read Flash	0x92	Source address Length	Data, Error code, if applicable
Find Header	0x93	-	Header address of the base module
Erase Flash	0x94	Start address Block length	Error code, if applicable
Program Flash	0x96	Target address Device type Data	Error code, if applicable
Build Checksum	0x97	Start address Module length	Error code, if applicable
Error Reset in Slave System	0x9F	-	-

Fig. 5-27: Overview of SIS subservices with reference to 0x02 service

5.3 Error Codes in the SIS Service 0x02 Flash Operation

The error codes in an error response telegram are a combination of the subservice (user data header) and the error byte (user date).

Shutdown	**0x90**	0x9002	Firmware was deleted
		0x9003	Shutdown in phase 3 not permitted
		0x9004	Shutdown in phase 4 not permitted
Reboot	**0x91**	0x9102	Firmware was deleted
		0x9103	Reboot in phase 3 not permitted
		0x9104	Reboot in phase 4 not permitted
Read Flash	**0x92**	0x9200	Error during read
Find Header	**0x93**		Error code not present
Erase Flash	**0x94**	0x9400	Time-out during deletion
		0x940A	Deletion possible in loader only
Program Flash	**0x96**	0x96FF	Error during write to RAM
		0x96E0	Verify error during programming of Flash
		0x96E1	Time-out during programming of Flash
Build Checksum	**0x97**	0x9701	Addition checksum error
		0x9702	CRC32 checksum error
Error Reset in Slave	**0x9F**		Error code not present

6 Initializing Communication via SIS

Before data exchange over the serial interface can take place, SIS communication must first be initialized. All users must be informed of the timing and the baud rate.

The 0x03 service is provided for the first initialization. This service enables a communication master to configure the bus or the connection and to inform its communication slaves of the relevant monitoring periods and the desired baud rate.

Required Service for all Users

This service is a **required service** for all SIS users.

6.1 SIS Service 0x03 Initialization of SIS Communications

Command Telegram

The command telegram of the 0x03 service consists of:

- Telegram header
- Address (**byte 0** of the user data header)
- Subservice (**byte 1** of the user data header)
- User data

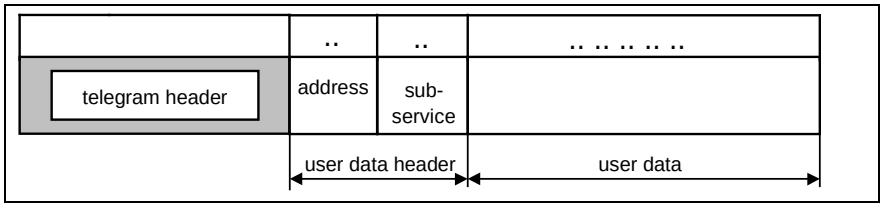


Fig. 6-1: Command telegram of service 0x03

The user data follow the user data header in subservices with write access (e.g. ‘specification of TrS’).

Response Telegram after Successful Execution of the Service

The 0x03 service response telegram after successful execution consists of:

- Telegram header
- Status (**byte 0** of the user data header) **= 0**
- Address (**byte 1** of the user data header)
- Subservice (**byte 2** of the user data header)

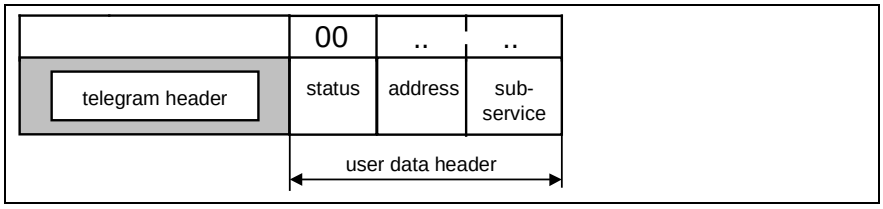


Fig. 6-2: Response telegram of service 0x03 in error-free case

Response Telegram in Case of Error

In case of error, an error code is written into the response telegram instead of the user data. The error code is always 1 byte in length.

- Telegram header
- Status (**byte 0** of the user data header) **= 1**
- Address (**byte 1** of the user data header)
- Subservice (**byte 2** of the user data header)
- Error code

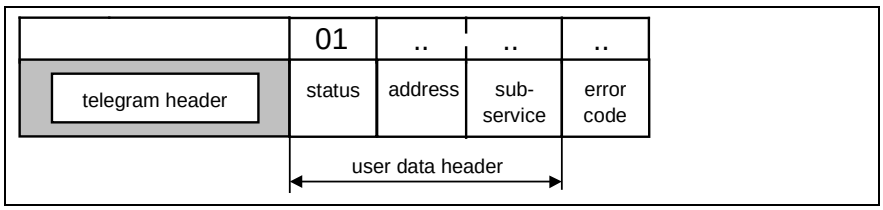


Fig. 6-3: Response telegram of service 0x03 in case of an error

Error Code

Because the error bytes depend on the subservice, it makes sense to generate a unique error code beyond the telegram structure. For this purpose, the subservice (user data header) and the error byte (user data) are combined into one sequential datum 2 bytes in length.

The telegram header and the first 2 or 3 bytes of the user data header are fixed components of the command or response telegram.

The user data dependent on the subservice are explicitly listed in the following subsections.

Note: In the 0x03 service, several subservices can be sent consecutively in one telegram because of the fixed user data lengths in the individual subservices.

Subservice 0x01 Setting TrS

Function

Sets the slave response period. The master will expect a response telegram within this period.

Command Telegram

User data:

Byte 2: Response period of the slave in [ms] - low byte

Byte 3: Response period of the slave in [ms] - high byte

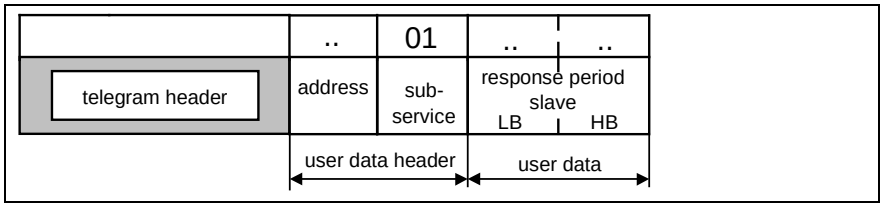


Fig. 6-4: Command telegram of subservice 0x01

Response Telegram User data:

None, error code, if applicable

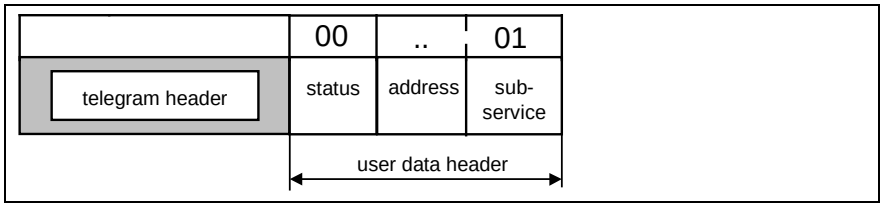


Fig. 6-5: Response telegram of subservice 0x01

Error Codes Error code not present

Subservice 0x02 Specifying TzA

Function Specifies the separation period between characters. The next character must be sent within this period.

Command Telegram User data:

Byte 2: Character separation period in [ms] - low byte

Byte 3: Character separation period in [ms] - high byte

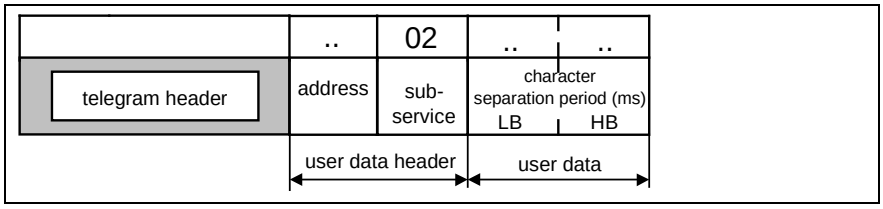


Fig. 6-6: Command telegram of subservice 0x02

Response Telegram User data:

None, error code, if applicable

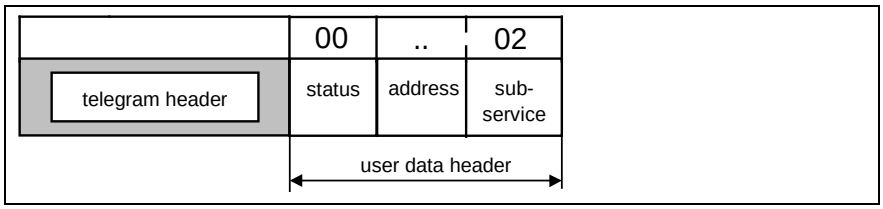


Fig. 6-7: Response telegram of subservice 0x02

Error Codes Error code not present

Subservice 0x03 Setting Tmas

Function

Sets the cycle period for the master control word (MSW). The master has to send the master control word (MSW) within this period when a machine movement is initiated. If the MSW is received later, the slave will terminate the movement and report an error.

Command Telegram

User data:

Byte 2: Cycle period of the MSW in [ms] - low byte

Byte 3: Cycle period of the MSW in [ms] - high byte

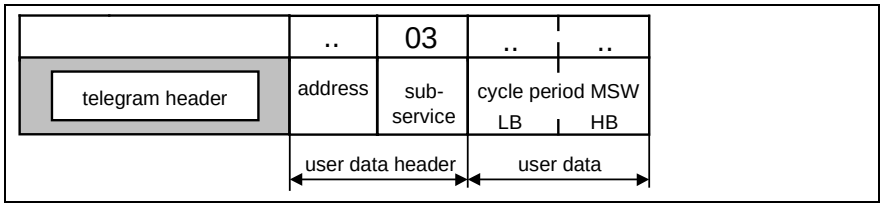


Fig. 6-8: Command telegram of subservice 0x03

Response Telegram

User data:

None, error code, if applicable

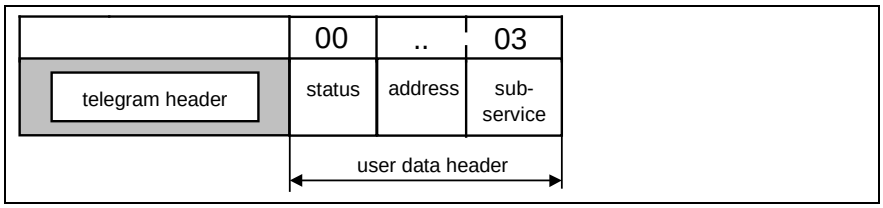


Fig. 6-9: Response telegram of subservice 0x03

Error Codes

Error code not present

Subservice 0x06 Address Allocation for Multi-Cast Group(s)

Subservice 0x06 is not yet specified. A service like this one becomes interesting when multiple users spread out randomly across a communication network must be supplied with communication telegrams at the same time. There are currently no applications for such a situation. Additionally, it could not be determined from discussions of this issue, to how many multicast groups a user should be assigned. It was determined that the number of assigned multicast groups should be very small, because each multicast address that must be supported increases the telegram processing time/effort for each user.

Subservice 0x07 Setting the Baud Rate

Function Sets the baud rate for all SIS users.

Note: When a SIS user is activated, it is always set to the default baud rate of 9600.

Command Telegram User data:

Byte 2: Baud rate mask

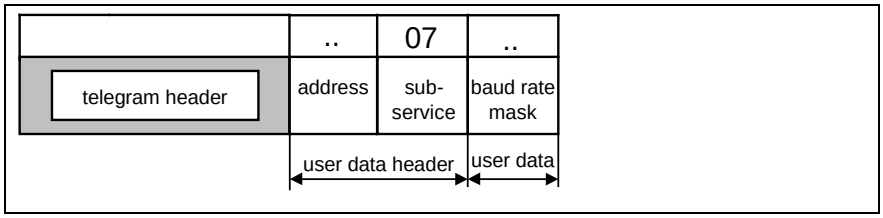


Fig. 6-10: Command telegram of subservice 0x07

Baud Rate Mask	Baud Rate
00000000	9600
00000001	19200
00000010	38400
00000100	57600
00001000	115200

Fig. 6-11: Baud rate mask

Response Telegram User data:

None, error code, if applicable

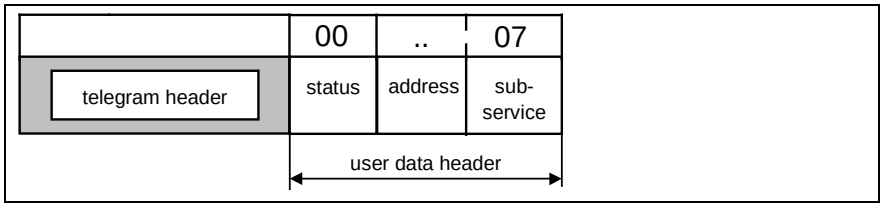


Fig. 6-12: Response telegram of subservice 0x07 in error-free case

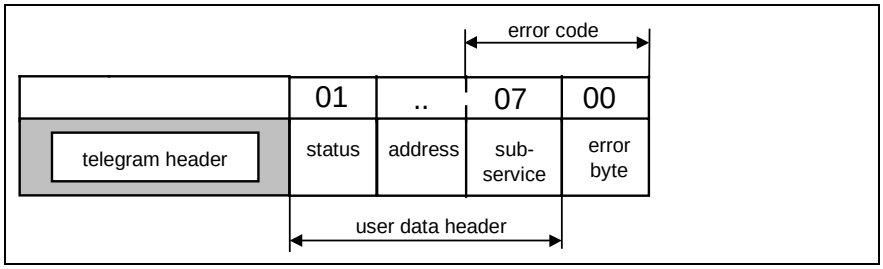


Fig. 6-13: Response telegram of subservice 0x07 in case of an error (e.g., 0x0700 baud rate not supported)

Error Codes 0x0700 Baud rate not supported

Examples of the 0x07 Subservice

Two sample telegrams for setting the baud rate are shown below:

Baud rate setting 9600

02 DB 03 03 00 03 10 03 (Telegram header)

00 07 00 (User data and header)

Telegram header								User data header		User data
StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Address	Sub-service	
02	D7	03	03	00	03	10	03	04	07	00

Fig. 6-14: Command telegram "baud rate setting 9600“

Baud rate setting 115200

02 D3 03 03 00 03 10 03 (Telegram header)

00 07 08 (User data and header)

Telegram header								User data header		User data
StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Address	Sub-service	
02	CF	03	03	00	03	10	03	04	07	08

Fig. 6-15: Command telegram "baud rate setting 115200“

Subservice 0x08 Timed Baud Rate Test

Function

Testing of the communication path to a SIS user with a pre-set baud rate during a specified time period.

Note: The SIS user answers using the (still) current baud rate and immediately switches to the new test baud rate. After the specified time period is expired, the SIS user returns to the original baud rate setting.

Command Telegram

User data:

Byte 2 - 3: Test duration in [ms]

Byte 4: Baud rate mask

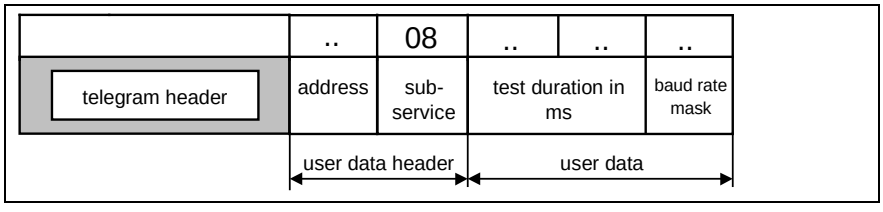


Fig. 6-16: Command telegram of subservice 0x08

Baud Rate Mask	Baud Rate
00000000	9600
00000001	19200
00000010	38400
00000100	57600
00001000	115200

Fig. 6-17: Baud rate mask

Response Telegram

User data:

None, error code, if applicable

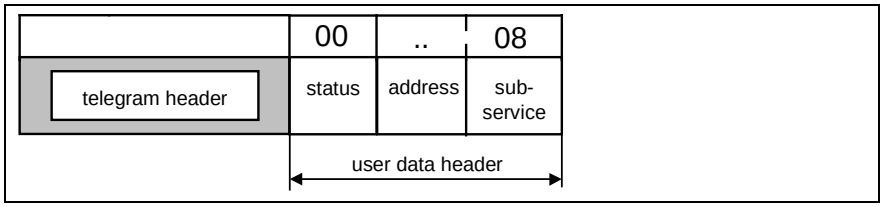


Fig. 6-18: Response telegram of subservice 0x08 in error-free case

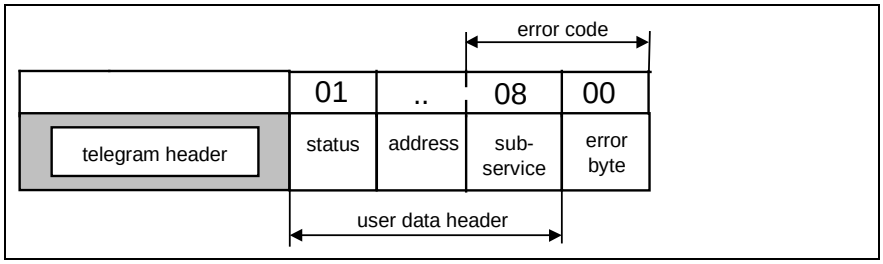


Fig. 6-19: Response telegram of subservice 0x08 in case of an error (e.g., 0x0800 baud rate not supported)

Error Codes

0x0800	Baud rate not supported
0x0801	Internal timer value for the test duration is too large

Examples of the 0x08 Subservice

Two sample telegrams for setting the baud rate are shown below:

Test using baud rate 9600 for 500 ms

02 E1 05 05 00 03 10 03 (Telegram header)

00 08 F4 01 00 (User data and header)

Telegram header								User data header		User data		
StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Address	Sub-service			
02	E1	05	05	00	03	10	03	00	08	F4	01	00

Fig. 6-20: Command telegram "test using baud rate 9600 for 500 ms“

Test using baud rate 115200 for 800 ms

02 AB 05 05 00 03 10 03 (Telegram header)

00 08 20 03 08 (User data and header)

Telegram header								User data header		User data		
StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Address	Sub-service	User data	User data	User data
02	AB	05	05	00	03	10	03	00	08	20	03	08

Fig. 6-21: Command telegram "test using baud rate 115200 for 800 ms“

Subservice 0xFF Assuming the Set Values

Function

This subservice enables the master to accept the prior values for the first initialization of the SIS communication as valid.

Note: It is recommended that this subservice should always be sent as a broadcast (address of the recipient: 0xFF) to enable all slaves to assume the new values simultaneously.

Command Telegram

User data:

None

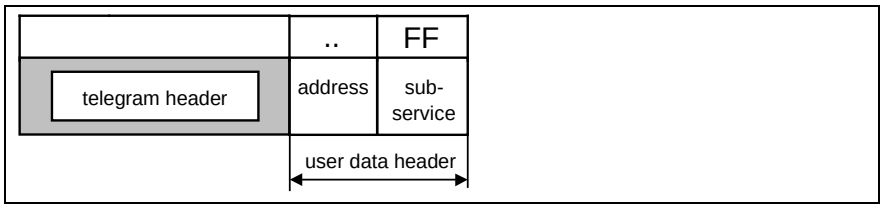


Fig. 6-22: Command telegram of subservice 0xFF

Response Telegram

User data:

None, error code, if applicable

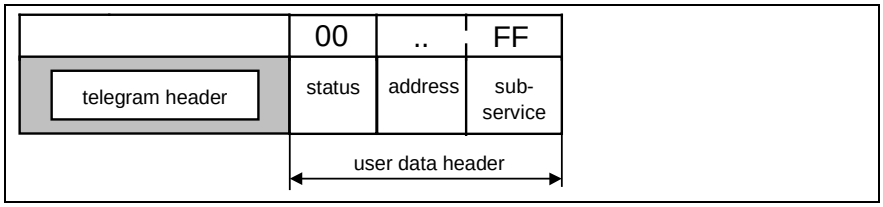


Fig. 6-23: Response telegram of subservice 0xFF

Note: When a point-to-point connection is first initialized, this subservice may also be sent to the address of the slave. In this case, the slave sends the response telegram in accordance with the default values for timing and baud rate.

Error Codes

Error code not present

Example of the 0xFF Subservice

Broadcast: assume settings

02 E9 02 02 00 03 10 FF (Telegram header)

00 FF (User Data header)

Telegram header								User data header	
StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Address	Sub-service
02	E9	02	02	00	03	10	FF	00	FF

Fig. 6-24: Command telegram "broadcast settings“

6.2 Overview of SIS 0x03 Service Initialization of SIS Communication

SIS Subservice	No.	Additional Data in the Command Telegram	User Data in the Response Telegram
Setting TrS	0x01	Response time of the slave (TrS)	-
Setting TzA	0x02	Character separation period (TzA)	-
Setting Tmas	0x03	Cycle time of the master control word (Tmas)	-
Address allocation for multicast group(s)	0x06	-	-
Setting the baud rate	0x07	Baud rate mask	Error code, if applicable
Timed baud rate test	0x08	Test duration in [ms], baud rate mask	Error code, if applicable
Assume the set values	0xFF	-	-

Fig. 6-25: Overview of SIS subservices with reference to the 0x03 service

6.3 Error Codes of the SIS 0x03 Service Initializing SIS Communication

The error codes in an error response telegram are made up of the subservice (user data header) and the error byte (user date).

Error Codes

So far, only the error codes for the 0x07 and 0x08 subservices have been specified.

Specifying the Baud Rate 0x07

0x0700 Baud rate not supported

Timed Baud Rate Test 0x08

0x0800 Baud rate not supported

0x0801 Internal timer value for the test duration is too large

7 Execution of Multiple SIS Services in a SIS Transmission

In this SIS service, a request for multiple SIS services in only one SIS telegram is supported. This way, dynamic (application-specific) or static data/request building blocks can be specified to minimize telegram traffic. For example, this service can be used for:

- transmitting multiple parameters in one telegram,
- application-specific data delivery for user-identification,
- application-specific initialization of SIS communication.

Note: The SIS service 0x04 may not be entered in its own list of requested SIS services, because this results in a circular reference in the request list!

7.1 SIS Service 0x04 Executing a List of SIS Services

Command Telegram

The 0x04 service command telegram consists of:

- Telegram header
- request: n1 — Data length of the first (1 byte user data header)
- First SIS service (1 byte user data header)
- first SIS service — User data/header of the ((n1 – 1) byte user data)
- request: n2 — Data length of the first (1 byte user data header)
- Second SIS service (1 byte user data)
- second SIS service — User data/header of the ((n2 – 1) byte user data)
- ...
- ...
- ...
- request: nm — Data length of the last (1 byte user data header)
- Last SIS service (1 byte user data)
- last SIS service — User data/header of the ((nm – 1) byte user data)

Input of the unit address is not necessary for service 0x04, because this information is already included in the individual SIS requests.

Note: Theoretically, different requests such as Write and Read Access can be listed together in the request list. Practically, however, this will make sense only in few applications.

Response Telegram

The 0x04 service response telegram consists of:

• Telegram header	
• **OK, 1 = Error**	Overall status byte: **0 =** (1 byte user data header)
• response: k1	Data length of the first (1 byte user data header)
•	First SIS service (1 byte user data)
• first SIS service	User data/header of the ((k_1 – 1) byte user data)
• response: k2	Data length of the second (1 byte user data header)
•	Second SIS service (1 byte user data)
• second SIS service	User data/header of the ((k_2 – 1) byte user data)
•	...
•	...
•	...
• response: km	Data length of the last (1 byte user data header)
•	Last SIS service (1 byte user data)
• last SIS service	User data/header of the ((k_m – 1) byte user data)

The overall status byte is used to quickly evaluate the response telegram (e.g. for a list of write access attempts).

Error Codes

SIS service 0x04 does not have its own error codes.

Even when an error occurs, the response telegrams of each individual SIS service are delivered in the response telegram of Service 0x04. The precise error analysis is provided via the status bytes and the error messages in the individual response telegrams.

7.2 Example for SIS Service 0x04 Executing a List of SIS Services

In the following example, the master wants to read the SIS Version, the unit typecode and the supported baud rate to check an SIS slave.

Command Telegram

In the list of SIS services to be executed, the subservices 0x01, 0x03 and 0x04 of the SIS Service 0x00 are entered:

02 8E 0C 0C 00 04 10 03	(Telegram header)
03 00 10 01	(Length: 3, SIS service: 0x00, unit address: 0x10, subservice: 0x01)
03 00 10 03	(Length: 3, SIS service: 0x00, unit address: 0x10, subservice: 0x03)
03 00 10 04	(Length: 3, SIS service: 0x00, unit address: 0x10, subservice: 0x04)

Telegram header								1. User data header				2. User data header				3. User data header			
StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Length	SIS service	Unit address	Sub-service	Length	SIS service	Unit address	Sub-service	Length	SIS service	Unit address	Sub-service
02	8E	0C	0C	00	04	10	03	03	00	10	01	03	00	10	03	03	00	10	04

Fig. 7-1: Command telegram of service 0x04

Response Telegram

The individual response telegrams of the listed SIS services are in the response telegram of SIS service 0x04:

02 F8 23 23 10 04 10 03	(Telegram header)
00	(Overall status byte)
09 00 00 10 01 30 31 56 30 31	(Length: 9, SIS service: 0x00, status byte: 0x00, unit address: 0x10, sub-service: 0x01, SIS version: "01V01")
11 00 00 10 03 43 4C 43 2D 44 30 32 2C 33 4D 2D 46 57	(Length: 17, SIS service: 0x00, status byte: 0x00, unit address: 0x10, sub-service: 0x03, unit typecode: "CLC-D02.3M-FW")
05 00 00 10 04 0F	(Length: 5, SIS service: 0x00, status byte: 0x00, unit address: 0x10, sub-service: 0x04, baud rates besides 9600: 19200, 38400, 57600, 115200)

Telegram header									1. User data header					1. User data				
StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Overall status byte	Length	SIS service	Status-byte	Unit address	Sub-service	SIS version				
02	F8	23	23	10	04	10	03	00	09	00	00	10	01	30	31	56	30	31

2. User data header					2. User data												
Length	SIS service	Status byte	unit address	Sub service	Unit typecode												
11	00	00	10	03	43	4C	43	2D	44	30	32	2C	33	4D	2D	46	57

3. User data header					3. User data
Length	SIS service	Status byte	unit address	sub-service	baud rate
05	00	00	10	04	0F

Fig. 7-2: Response telegram of service 0x04

8 Handling of SERCOS parameters and SERCOS phase

In this chapter some SIS services are introduced to handle the transfer of SERCOS parameters and the SERCOS phases (modes).

These services are strongly related to the groups specific SIS services 0x80 – 0x8F, 0x90 – 0x9F and 0xC0 – 0xCF; there are a few differences only.

The common SIS services are

- **0x10** Read access on a SERCOS parameter, supporting consecutive telegrams in case of (long) lists.
- **0x11** Read access on a list segment of SERCOS parameter, supporting no consecutive telegram.
- **0x12** Read access on the actual SERCOS phase.
- **0x1D** Switch of the SERCOS phase (Write access).
- **0x1E** Write access on a list segment of a SERCOS parameter, supporting no consecutive telegram.
- **0x1F** Write access on a SERCOS parameter, supporting consecutive telegrams in case of (long) lists.

8.1 Telegram Content

User data header

The user data header describes the type of request. The elements of the user data header are

- control byte
- unit address
- parameter number and type (command telegrams only)
- list offset (command telegrams only)
- data length (command telegrams only)
- status byte (response telegrams only)

Control byte

The control byte specifies how a data block element of a parameter is accessed. The transmission of a consecutive telegram is controlled with bit 2 (lists are written in several steps).

The control byte is read out of the command telegram and copied into the response telegram. The transmission of a consecutive telegram is controlled with bit 2 (lists read in several steps).

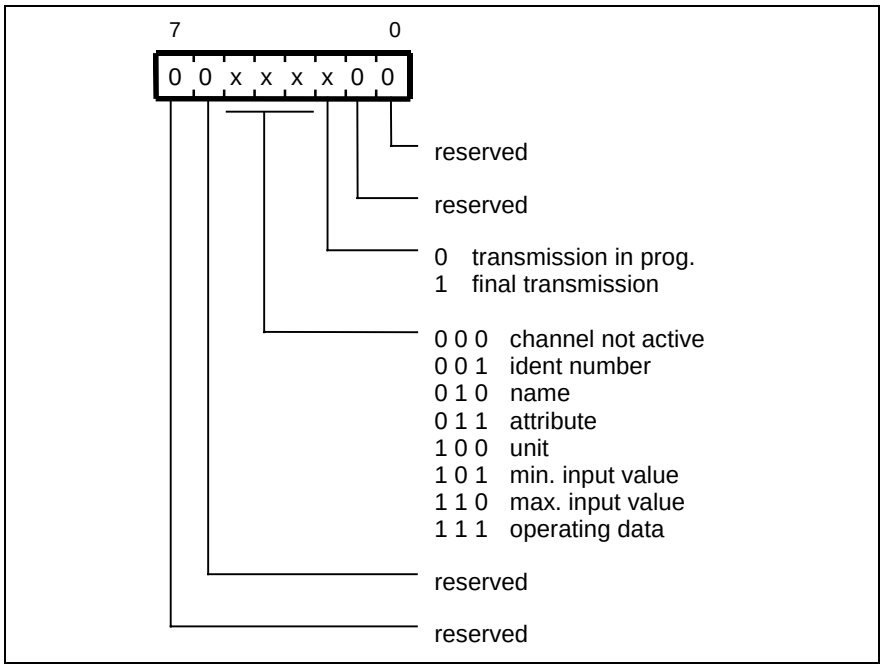


Fig. 8-1: Control byte structure

Write access of

- the operating data
- and the identical number (data status read, compare SERCOS interface specification 5.1.3.8)

is possible.

Read access of all elements that describe a parameter is possible. These elements are:

- the attribute,
- minimum and maximum input values,
- the name,
- the unit and
- the operating data.

Unit address

The unit address of a drive is read in the command telegram and copied into the response telegram.

The serial interface permits

- direct SIS communication with drives supporting SIS interface. In this case the unit address is the same as the SIS address of the receiver.
- accessing drive parameters via a motion control, in case of drives not supporting SIS interface. The SIS address is related to the motion control and the unit address to the drive.

Given SIS communication with a motion control as a SIS slave and a SERCOS master, then the SERCOS master must be informed as to which unit the request relates to. This unit can be the SERCOS master itself or any of the drives it controls.

The address set at the drive controller or "0" are transmitted.

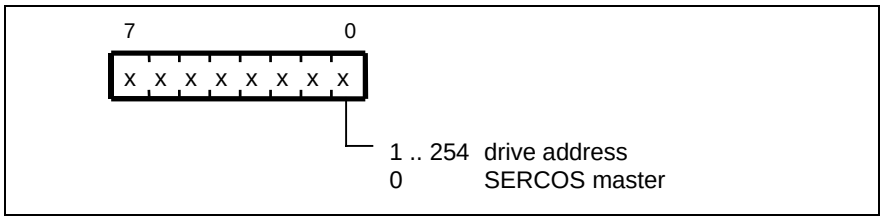


Fig. 8-2: Unit address

Parameter number and type

The parameter number has the form as defined in the SERCOS interface specification. To be able to also address control parameters, one byte is set ahead of the address to identify the parameter type.

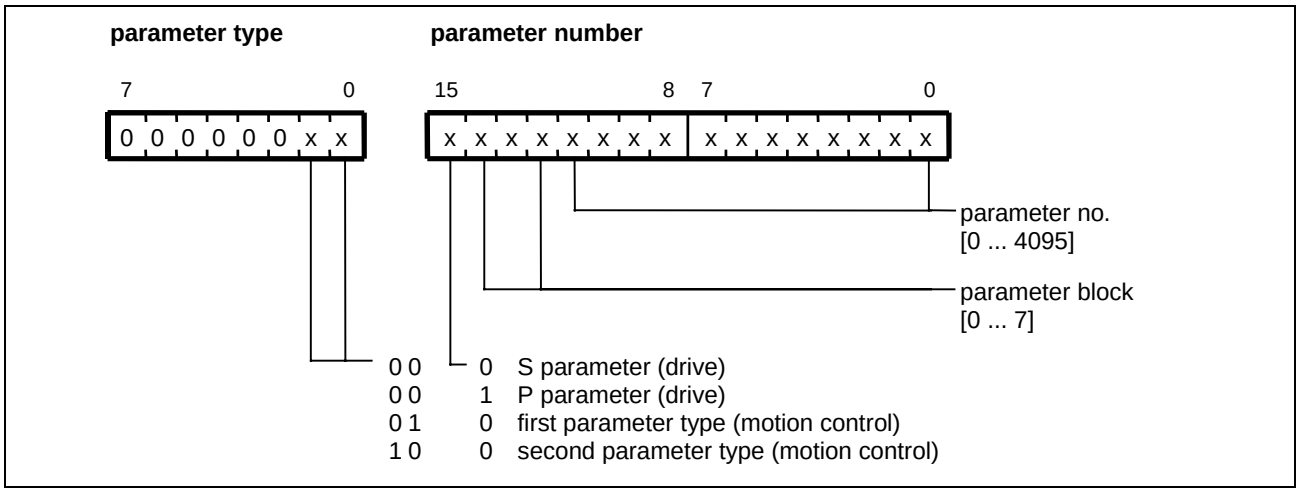


Fig. 8-3: Parameter identification (parameter transmission)

Status byte

The status byte supplies the results of a transmission in the form of a code number.

It generally applies:

Error type	Error code
Transmission error-free	0x00
Protocol error	0xF0 ... 0xFF
Execution error (see below)	0x01 ... 0xEF

Fig. 8-4: Error types

Whereby

- the Protocol errors are defined as in “Telegram Content“ in chapter 2
- and the Execution errors are defined as follows:

Execution error	Code number	Description of error
"Error during parameter transmission"	0x01	An error occurred while reading or writing a parameter (see below "Error codes in SIS services 0x10 – 0x1F")
"Error during phase switching"	0x02	The specified target phase was not achieved (see below " Error codes in SIS services 0x10 – 0x1F")

Fig. 8-5: Execution error

List offset The list offset is only specified with the transmission of a segment of a parameter list. It sets the number of bytes the desired element is to be shifted in contrast to the first element within the list.

Data length The list offset is only specified with the transmission of a segment of a parameter list. It specifies the data length of the list segment.

User data

User data are data to be transmitted. User data elements are:

- SERCOS phase (target phase or current phase)
- parameter value
- error word with execution error (response telegram only)

8.2 SIS Service 0x10 Read a SERCOS parameter

Command Telegram The command telegram of the 0x10 service consists of:

- Telegram header
- control byte (**byte 0** of the user data header)
- unit address (**byte 1** of the user data header)
- parameter number and type (command telegrams only) (**bytes 2 - 4** of the user data header)

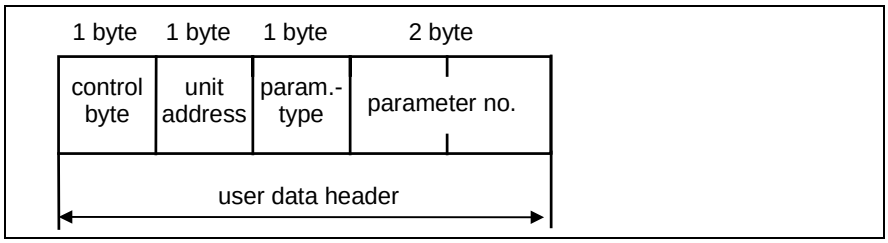


Fig. 8-6: Command telegram of service 0x10

Response Telegram after Successful Execution of the Service The 0x10 service response telegram after successful execution consists of:

- Telegram header
- Status (**byte 0** of the user data header) **= 0**
- control byte (**byte 1** of the user data header)
- unit address (**byte 2** of the user data header)
- parameter data (user data):
  max. user data length per each telegram = 236 bytes
  (255 - 16{protocol header, additional header} -3 {user data header})

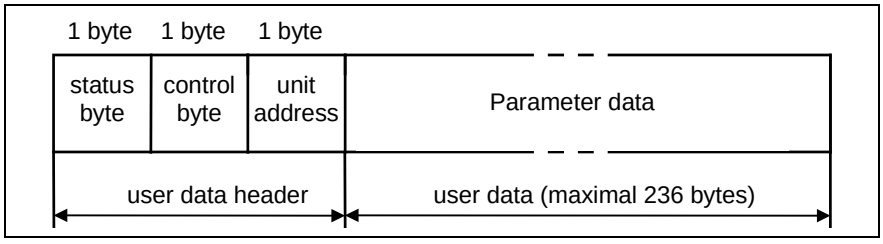


Fig. 8-7: Response telegram of service 0x10 in error-free case

Response Telegram in Case of Error

In case of error, an error code is written into the response telegram instead of the user data. The error code is always 2 bytes in length.

- Telegram header
- Status (**byte 0** of the user data header) **= 1**
- control byte (**byte 1** of the user data header)
- unit address (**byte 2** of the user data header)
- Error code

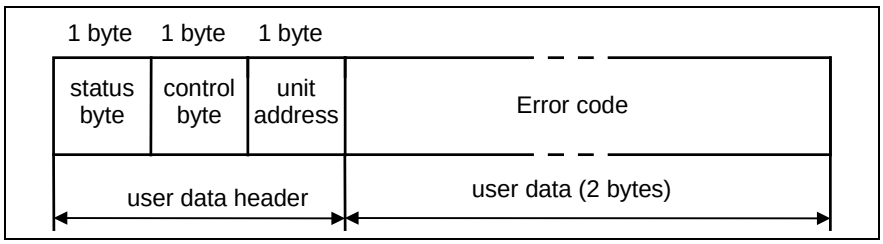


Fig. 8-8: Response telegram of service 0x10 in case of an error

Example of reading a parameter (service 0x10)

Parameters with a length exceeding maximum user data length of 236 bytes are read in steps. Bit 2 in the control byte identifies the current transmission step as a **running** or **final** transmission. A list of four byte data is read. The final data is 0x05F5E100.

The control byte for a transmission in several steps is listed below. The final data of the list is explicitly depicted to illustrate the Intel format of four byte data.

1st step:

Write-Request of the master with parameter request.

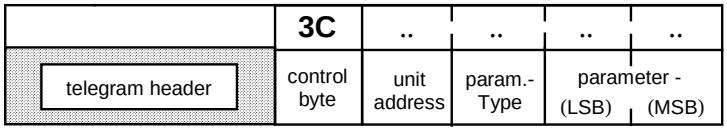


Fig. 8-9: First command telegram of service 0x10

After the first Read-Request of the master, the slave sends a response telegram:

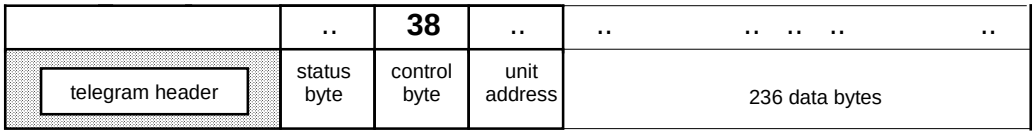


Fig. 8-10: First response telegram of service 0x10

2nd step:

Next Write-Request of the master with the parameter request.

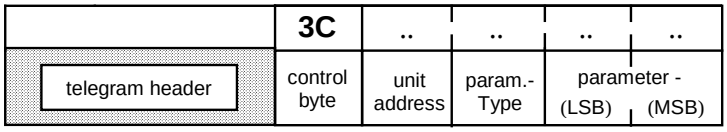


Fig. 8-11: Second command telegram of service 0x10

After the next Read-Request of the master, the slave sends the next response telegram (1st consecutive telegram):

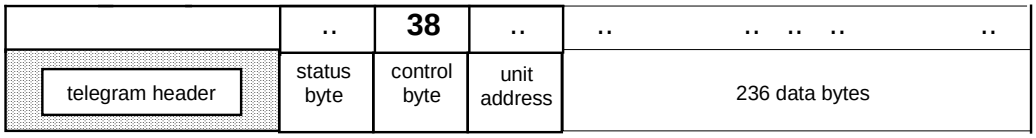


Fig. 8-12: Second response telegram of service 0x10

...

Final step

Final Write-Request of the master with the parameter request.

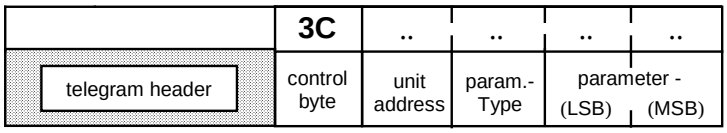


Fig. 8-13: Final command telegram of service 0x10

After the final Read-Request of the master, the slave sends the final response telegram (final consecutive telegram):

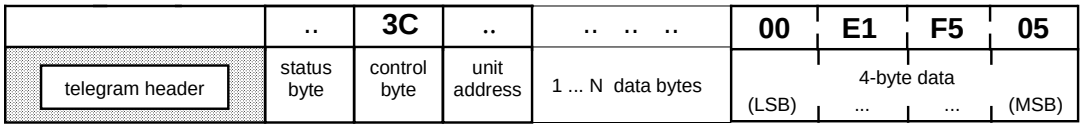


Fig. 8-14: Final response telegram of service 0x10

8.3 SIS Service 0x11 Read a segment of a SERCOS list

Command Telegram

The command telegram of the 0x11 service consists of:

- Telegram header
- control byte (**byte 0** of the user data header)
- unit address (**byte 1** of the user data header)
- parameter number and type (command telegrams only) (**bytes 2 - 4** of the user data header)
- list offset (**bytes 5 – 6** of the user data header)
- data length (**bytes 7 – 8** of the user data header)

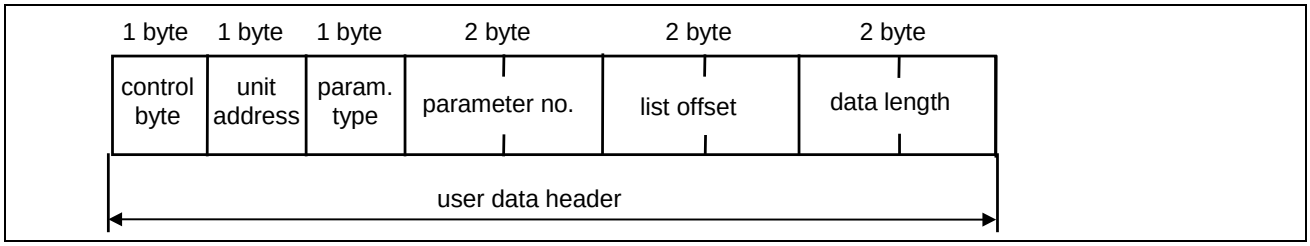


Fig. 8-15: Command telegram of service 0x11

Response Telegram after Successful Execution of the Service

The 0x11 service response telegram after successful execution consists of:

- Telegram header
- Status (**byte 0** of the user data header) **= 0**
- control byte (**byte 1** of the user data header)
- unit address (**byte 2** of the user data header)
- parameter data (user data):
  max. user data length per each telegram = 236 bytes
  (255 - 16{protocol header, additional header} - 3{user data header})

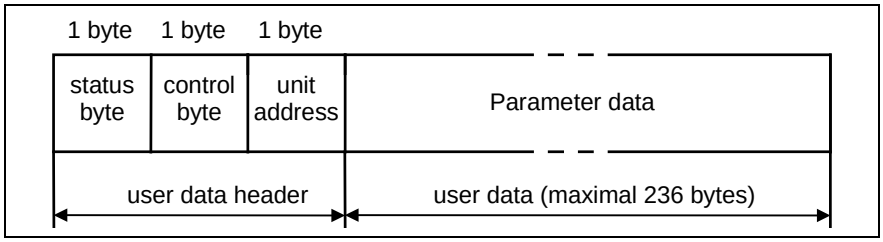


Fig. 8-16: Response telegram of service 0x11 in error-free case

Response Telegram in Case of Error

In case of error, an error code is written into the response telegram instead of the user data. The error code is always 2 bytes in length.

- Telegram header
- Status (**byte 0** of the user data header) **= 1**
- control byte (**byte 1** of the user data header)
- unit address (**byte 2** of the user data header)
- Error code

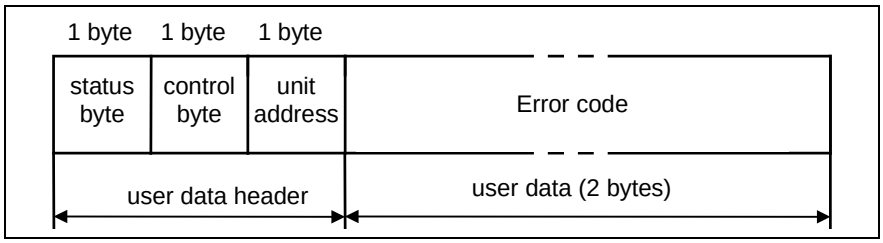


Fig. 8-17: Response telegram of service 0x11 in case of an error

Example of reading a list element (Service 0x11)

The eleventh element of a list of 4 byte data is to be read. The data is 0x05F5E100.

The following is the control byte for a transmission in one step. The list offset equals 40 bytes (= 10 elements). The data of the list is explicitly illustrated to clarify the Intel format of four-byte data.

Write-request of the master with parameter request:

	3C	..	..	..	..	0x28	0x00	0x04	0x00
telegram header	control byte	unit address	para.-type	parameter - (LSB)	parameter - (MSB)	list offset (LSB)	list offset (MSB)	data length (LSB)	data length (MSB)

Fig. 8-18: Command telegram of service 0x11

After the read request of the master, the slave sends the response telegram:

	..	3C	..	00	E1	F5	05
telegram header	status byte	control byte	unit address	(LSB)	4 byte date ...	...	(MSB)

Fig. 8-19: Response telegram of service 0x11

8.4 SIS Service 0x12 Read the current SERCOS phase

The following applies to reading the current SERCOS phase.

Command telegram

The command telegram does not contain user data. It consists only of:

- Telegram header

Response telegram

The response telegram consists of:

- Telegram header
- Status (**byte 0** of the user data header)
- current SERCOS phase
- error code (in case of error only)

Example when reading the current phase (Service 0x12)

Command telegram:

StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E
02	6C	00	00	00	92	00	00

Fig. 8-20: Command telegram 0x12 "Read phase“

Response telegram:

StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Status byte	User data
02	56	02	02	10	92	00	00	00	02

Fig. 8-21: Response telegram 0x12 " Phase 2 transition“

The current communication phase is transmitted in the user data (example: Phase "2").

After phase switching is activated (see example of phase switching: service 0x1D) the status byte displays the status of the progression. The master must repeat read access to the communication phase until the specified phase is reached or an error is signaled in the status byte.

If the phase switching cannot be executed, then "error with phase switching" is signaled in the status byte of the response telegram. The current SERCOS phase and the error code are transmitted in the user data. With running master axis, error code 0x8004 is set. The response telegram then looks like this:

	02	04	07	D0
telegram header	status byte	current phase	error code (LSB)	(MSB)

Fig. 8-22: Response telegram of service 0x12 in case of error

The error code in the response telegram illustrates the Intel format of 2-byte data.

8.5 SIS Service 0x1D Switching the SERCOS phase

The serial protocol can be used to switch between parameterization and operating mode. This progression is triggered by setting the *SERCOS* phase. It applies:

SERCOS phase 2 = parameterization mode

SERCOS phase 4 = operating mode

The following applies to switching the SERCOS phase.

Command telegram

The command telegram consists of:

- Telegram header
- New SERCOS phase

Response telegram

The response telegram consists of:

- Telegram header
- Status (**byte 0** of the user data header)
- current SERCOS phase
- error code (in case of error only)

Note: A successfully started phase transition cannot be aborted with a general service termination (0x01).

Example: Switching into operating mode (Service 0x1D)

Progression from parameterization into operating mode is done by setting SERCOS phase 4.

Command telegram:

StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	User data
02	5B	01	01	00	9D	00	00	04

Fig. 8-23: Command telegram 0x1D "phase 4 transition“

The command is acknowledged in the first response telegram, but not yet the execution of the progression.

Response telegram:

StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Status byte
02	4F	01	01	10	9D	00	00	00

Fig. 8-24: Response telegram 0x1D "Phase 4 transition“

Still after the first response telegram, the master detects with "polling". whether and when the progression is concluded. The current SERCOS phase is repeatedly read (see example for reading a current phase: Service 0x12), until the operating mode or the error is signaled during phase switching.

Example: Switching into parameterization mode (service 0x1D)

Switching from operating into parameterization mode is done by setting SERCOS phase 2.

Command telegram:

StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	User data
02	5D	01	01	00	9D	00	00	02

Fig. 8-25: Command telegram 0x1D " Phase 2 transition“

The command is acknowledged in the first response telegram, but not yet the execution of the progression.

Response telegram:

StZ	CS	DatL	DatLW	Cntrl	Service	Adr.S	Adr.E	Status byte
02	4F	01	01	10	9D	00	00	00

Fig. 8-26: Response telegram 0x1D "Phase 2 transition"

Still after the first response telegram, the master detects whether and when the progression is concluded by means of "polling". The current SERCOS phase is repeatedly read (see example for reading a current phase: service 0x12), until the operating mode or the error is signaled during phase switching.

If the set phase switching cannot be executed, then "error with phase switching" is signaled in the status byte of the response telegram. The current SERCOS phase and error code are transmitted in the user data. With a non-permitted phase (phase > 4), error code 0x8004 is set. The response telegram then looks like this:

	02	04	04	80
protocol header	status byte	current phase	error code (LSB)	(MSB)

Fig. 8-27: Response telegram of service 0x1D in case of error

The error code in the response telegram illustrates the Intel format of 2-byte data.

8.6 SIS Service 0x1E Write a segment of a SERCOS list

Command Telegram

The command telegram of the 0x1E service consists of:

- Telegram header
- control byte (**byte 0** of the user data header)
- unit address (**byte 1** of the user data header)
- parameter number and type (command telegrams only) (**bytes 2 - 4** of the user data header)
- list offset (**bytes 5 – 6** of the user data header)
- data length (**bytes 7 – 8** of the user data header)
- parameter data (segment of a list = user data):
  max. user data length per each telegram = 234 bytes
  (255 - 16{protocol header, additional header} - 5{user data header})

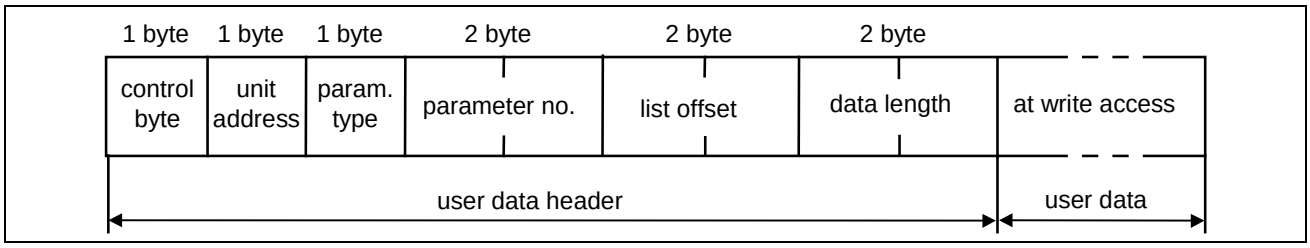


Fig. 8-28: Command telegram of service 0x1E

Response Telegram after Successful Execution of the Service

The 0x1E service response telegram after successful execution consists of:

- Telegram header
- Status (**byte 0** of the user data header) **= 0**
- control byte (**byte 1** of the user data header)
- unit address (**byte 2** of the user data header)
- data status in case of write access on the identical number (user data)

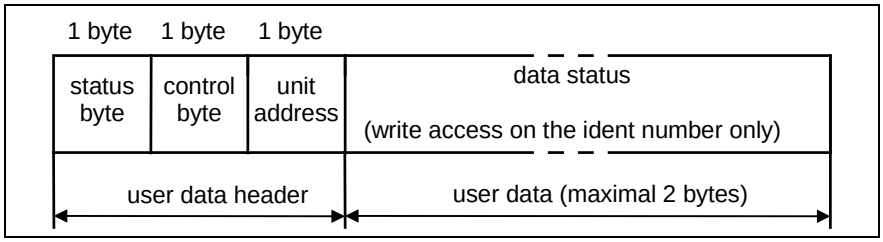


Fig. 8-29: Response telegram of service 0x1E in error-free case

Response Telegram in Case of Error

In case of error, an error code is written into the response telegram instead of the user data. The error code is always 2 bytes in length.

- Telegram header
- Status (**byte 0** of the user data header) **= 1**
- control byte (**byte 1** of the user data header)
- unit address (**byte 2** of the user data header)
- Error code

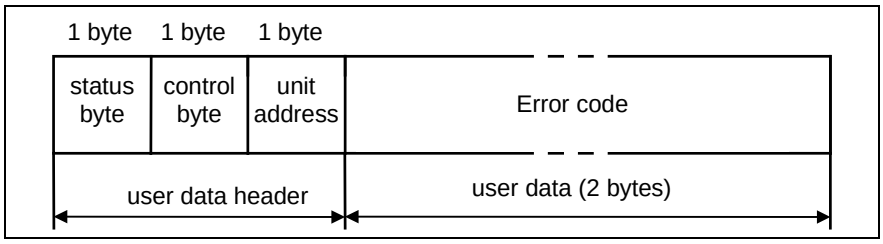


Fig. 8-30: Response telegram of service 0x1E in case of an error

Example of writing a list element (service 0x1E)

The second element of a line of 2-byte data is to be written. The data is 0x86A0. The list offset equals 2 bytes (= 1 element).

The following is the control byte for a transmission in one step. The final data of the list is explicitly illustrated to clarify the Intel format of the two byte data.

Write-request of master with the value for the list element:

	3C	..	..	..	..	02	00	02	00	A0	86
telegram header	control byte	unit address	para. type	parameter - (LSB)	(MSB)	list offset (LSB)	(MSB)	data length (LSB)	(MSB)	2 byte data (LSB)	(MSB)

Fig. 8-31: Command telegram of service 0x1E

Optionally, the transmission of the data is checked. The master sends a read request to the slave for this. The slave responds with a response telegram:

	..	3C	..
telegram header	status byte	control byte	unit address

Fig. 8-32: Response telegram of service 0x1E

8.7 SIS Service 0x1F Write a SERCOS parameter

Command Telegram

The command telegram of the 0x1F service consists of:

- Telegram header
- control byte (**byte 0** of the user data header)
- unit address (**byte 1** of the user data header)
- parameter number and type (command telegrams only) (**bytes 2 - 4** of the user data header)
- parameter data (segment of a list = user data):
  max. user data length per each telegram = 234 bytes
  (255 - 16{protocol header, additional header} - 5{user data header})

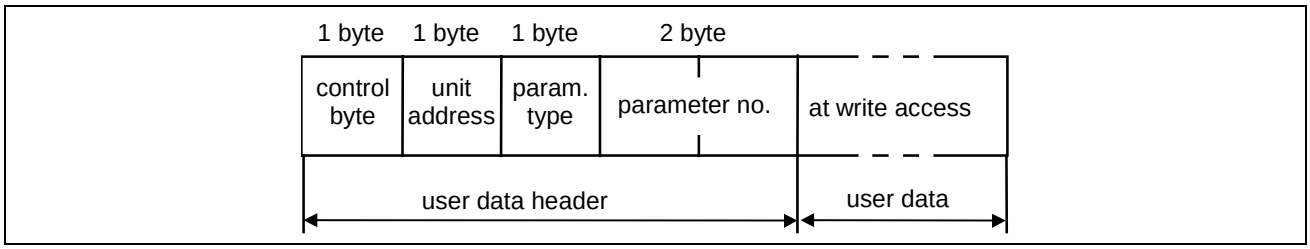


Fig. 8-33: Command telegram of service 0x1F

Response Telegram after Successful Execution of the Service

The 0x1F service response telegram after successful execution consists of:

- Telegram header
- Status (**byte 0** of the user data header) **= 0**
- control byte (**byte 1** of the user data header)
- unit address (**byte 2** of the user data header)
- data status in case of write access on the identical number (user data)

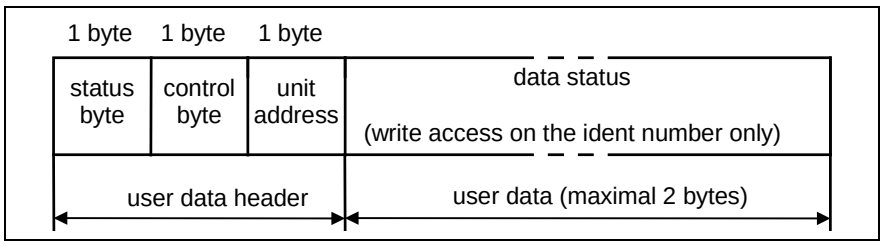


Fig. 8-34: Response telegram of service 0x1F in error-free case

Response Telegram in Case of Error

In case of error, an error code is written into the response telegram instead of the user data. The error code is always 2 bytes in length.

- Telegram header
- Status (**byte 0** of the user data header) **= 1**
- control byte (**byte 1** of the user data header)
- unit address (**byte 2** of the user data header)
- Error code

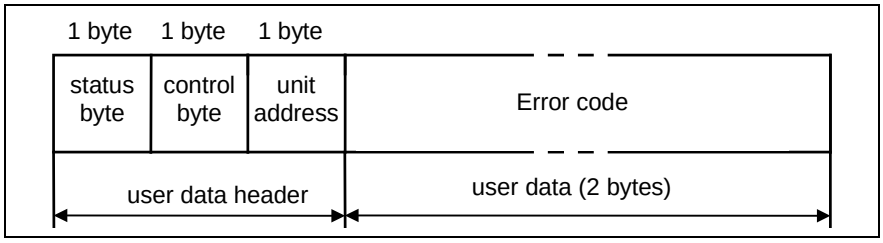


Fig. 8-35: Response telegram of service 0x1F in case of an error

Example for writing into a parameter (Service 0x1F)

Parameters with a length exceeding maximum user data length of 234 bytes are written consecutively. The transmission of such lists is performed in several steps. Bit 2 in the control byte identifies the current transmission steps as either **in progress** or the **final transmission**.

A list of 4 byte data is to be written. The final data is 0x000186A0.

The control byte for a transmission in several steps is listed below. The final data of the list is explicitly depicted to illustrate the Intel format of four byte data.

1st step:

Write-Request of the master with the first data block

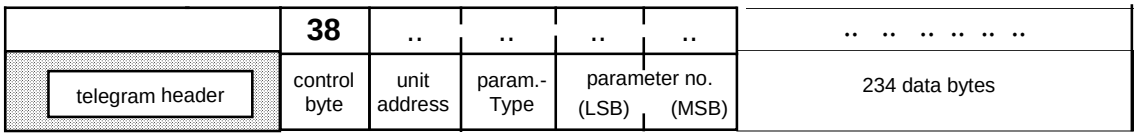


Fig. 8-36: First command telegram of service 0x1F

The transmission of data can optionally be checked. The master sends a Read-Request to this end. The slave responds with a response telegram.

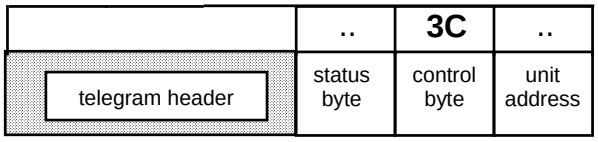


Fig. 8-37: First response telegram of service 0x1F

2nd step:

Write-Request of the master with additional data

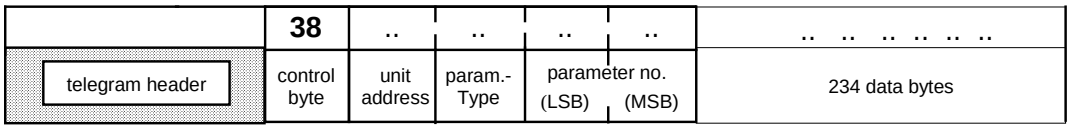


Fig. 8-38: Second command telegram of service 0x1F

The transmission of data can optionally be checked. The master sends a Read-Request for this purpose. The slave responds with a response telegram.

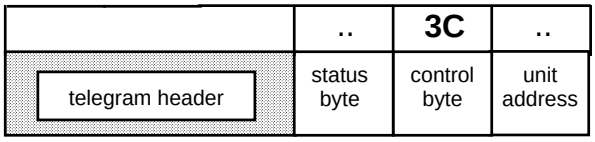


Fig. 8-39: Second response telegram of service 0x1F

Final step:

Write-Request of the master with the final data block (final consecutive telegram):

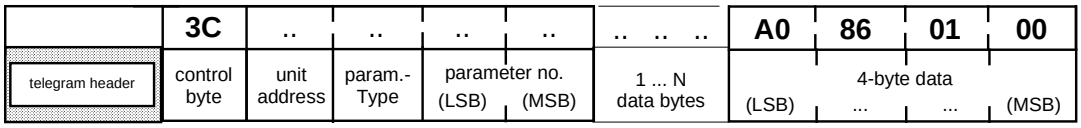


Fig. 8-40: Final command telegram of service 0x1F

After the concluding Read-Request, the master sends the response telegram to the slave:

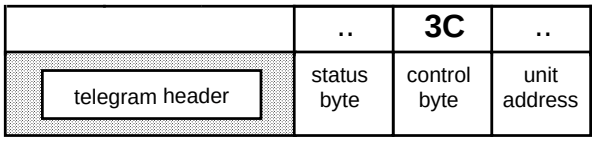


Fig. 8-41: Final response telegram of service 0x1F

Example for reading the data status of a parameter

To check a procedure command execution, it is necessary to read the data status. This is supplied in an SIS response telegram (as per SERCOS interface) when write requesting the ID number of a parameter in the form of user data (service 0x1F).

The write request is performed by repeating the parameter number as the 2-byte user data of the procedure command.

The following is an example of a check of the data status of the procedure command "C300 command set absolute measurement" (P-0-0012).

Command telegram:

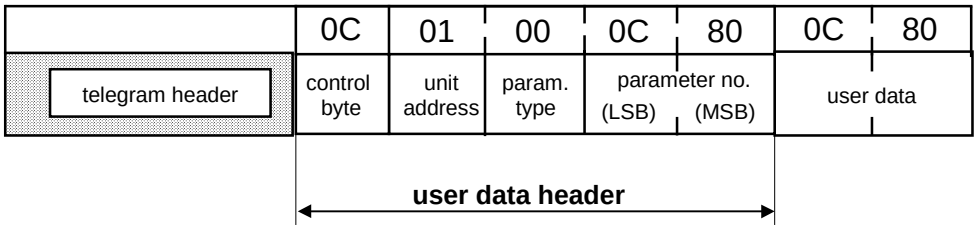


Fig. 8-42: Command telegram of service 0x1F

After the master's read request the slave sends the response telegram:

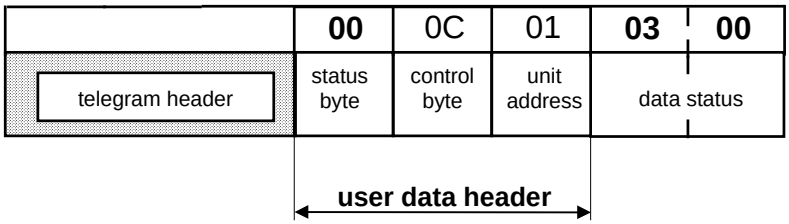


Fig. 8-43: Response telegram of service 0x1F

Data status 0x0003 shows that the procedure command is set, enabled and executed successfully.

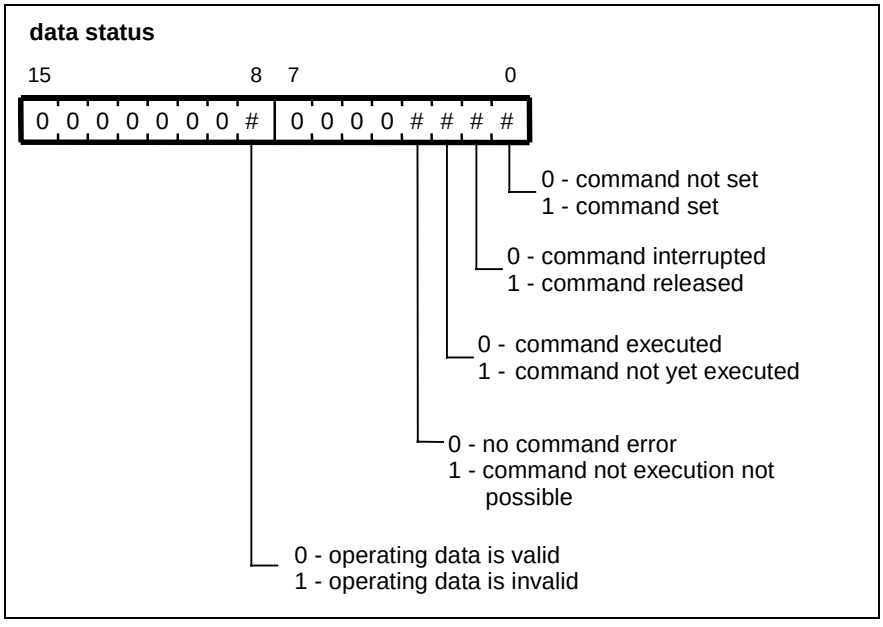


Fig. 8-44: Data status

Example of faulty parameter accessing

Write access to write-protected slave parameter "ELS master - actual position value" (C-0-0066).

The master is trying to write a value of 0 into the parameter. The slave acknowledges with error message 0x7004 ("data cannot be changed").

Command telegram:

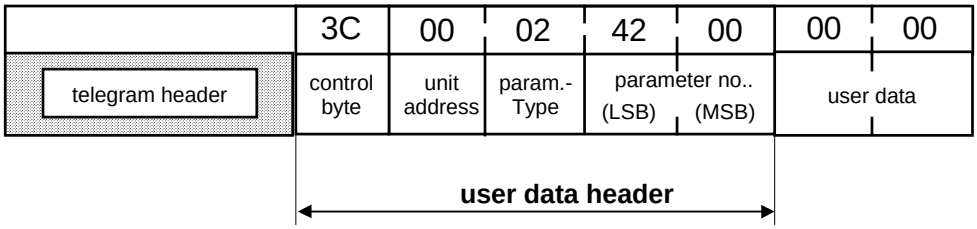


Fig. 8-45: Command telegram of service 0x1F

Response telegram:

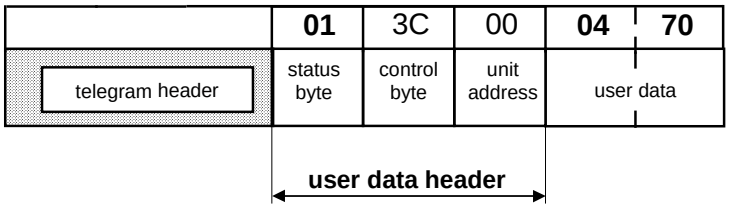


Fig. 8-46: Response telegram of service 0x1F

8.8 Overview of SIS services related to SERCOS interface

SIS Service	No.	User Data in the Command Telegram	User Data in the Response Telegram
Read a SERCOS parameter (support of consecutive telegrams)	0x10	---	Parameter data or Error code
Read a segment of a SERCOS list (single service only)	0x11	---	SERCOS list segment or Error code
Read the current SERCOS phase	0x12	---	Current SERCOS phase
Switching the SERCOS phase	0x1D	New SERCOS phase	Error code, if applicable
Write a segment of a SERCOS list (single service only)	0x1E	SERCOS list segment	Error code, if applicable
Write a SERCOS parameter (support of consecutive telegrams)	0x1F	Parameter data or Parameter number (read data status)	Data status or Error code, if applicable

Fig. 8-47: Overview of SIS subservices related to SERCOS interface

8.9 Error codes in SIS services related to SERCOS interface

Execution errors during parameter transmission

Errors occurring during the transmission of a parameter generate an error word in the response telegram with a specified error code. In case of a read access this error code will then be issued instead of the requested data in the user data field.

Error code	Error message in serial protocol
0x0000	no error
0x0001	service channel not open
0x0009	incorrect access to element 0
0x1001	no IDN available
0x1009	incorrect access to element 1
0x2001	no name available
0x2002	name transmission too short
0x2003	name transmission too long
0x2004	name cannot be changed
0x2005	name presently write-protected
0x3002	attribute transmission too short
0x3003	attribute transmission too long
0x3004	attribute cannot be changed
0x3005	attribute presently write-protected
0x4001	no unit available
0x4002	unit transmission too short
0x4003	unit transmission too long
0x4004	unit cannot be changed
0x4005	unit presently write-protected
0x5001	no minimum input value available
0x5002	minimum input value transmission too short
0x5003	minimum input value transmission too long
0x5004	minimum input value cannot be changed
0x5005	minimum input value presently write-protected
0x6001	no maximum input value available
0x6002	maximum input value transmission too short
0x6003	maximum input value transmission too long
0x6004	maximum input value cannot be changed
0x6005	maximum input value presently write-protected
0x7002	data transmission too short
0x7003	data transmission too long
0x7004	data cannot be changed
0x7005	data presently write-protected
0x7006	data smaller than minimum input value
0x7007	data greater than maximum input value
0x7008	data not correct
0x700C	"data exceeds numeric range" The transmitted value is smaller than zero or greater than the "modulo value" (S-0-0103) in the case of a modulo axis.
0x700D	"data length cannot presently be changed" The data length in current mode cannot be changed.

0x700E	"data length cannot be changed" The length of the data is permanently write protected.
0x700F	"list element not available“. List offset set in SIS services 0x91 or 0x9E exceeds range of list or does not show the start address of a list element.
0x8001	"service channel presently busy (BUSY)“ The desired access was not concluded within a timeout (programmed via C-0-0124) because the service channel, for example, was (still) busy. Data transmission is not conducted.

Fig. 8-48: Execution errors during parameter transmission

Execution errors when switching SERCOS phase

Errors occurring during phase switching generate an error word with a specific error code in the response telegram. It follows the current phase in the user data field.

Error code	Error message in serial protocol
0x8004	Incorrect phase specified via serial protocol
0xD005	"Phase switching still active" A phase switching presently not possible as one is still active
0xD006	"Phase switching with drive enable not possible" Set for at least one drive - "AF"
0xD007	"Phase switching with rotating master axis not permitted"

Fig. 8-49: Execution errors during phase switching

9 Token Passing via SIS

This service enables a communication master to forward the token to another communication master.

9.1 SIS Service 0x0F Token Passing

!!! This service is not yet specified !!!

10 Timing for the SIS Interface

10.1 Constraints on the Individual Users

Control Unit (BTV05)

- Communication cycle completed within 100 ms
- Connection not valid if bus inactive for more than 200 ms
- Parameterization with PC via bus not required, because a separate channel is provided for this!

Drive (DKC)

- Parameterization with PC via bus must be possible

10.2 Sequence for First Initialization of the Bus Users

The master informs the slaves of all important information in a first initialization service. The following could be considered:

- the Master Change-Over Time
- the Baud Rate.

However, to enable the master to have information on the users on the bus, a classification of users in accordance with their properties is proposed. This information must be available to the master, e.g. as a **config file**.

10.3 Synchronization of a User on the Operating Bus

Principle

A slave may not begin to transmit on its own initiative but only upon request from the master.

Because of this, if the TzA value is not correctly set, the new bus user will not detect a start character, because the inaction period on the bus is not long enough. If this is the case, the master will repeat the previously sent telegram once the slave response period (TrS) has expired. Therefore, in the second case the inaction on the bus has been long enough to enable the new user to detect the telegram start and also to evaluate this telegram.

Note: In the case described above, the prerequisite is that all users are set to the correct baud rate. However, this is considered in more detail in a separate section.

10.4 Automatic Baud Rate Detection

Principle

During the synchronization process, all baud rates must be scanned in sequence. This is the criterion for switching to the next baud rate: detection of an interface error (RS232 hardware error).

This routine will be terminated only when a valid telegram has been detected.

10.5 Discussion of the Timing Values

Maximum Allowable Character Separation (TzA)

If TzA is exceeded, the current communication (receive mode) is initially continued. If the character immediately following is a start character, a pointer is also set at this position on the input ring buffer, enabling an alternative evaluation from this position.

An initial test of whether the first 4 bytes from this position are the beginning of a new telegram (start byte and double length input) is executed.
If this is the case, the evaluation of the new telegram will be continued.

In all other cases, the evaluation of the telegram, which was terminated by the TzA being exceeded, will be continued beginning with the previously set pointer.

Note: There are particular problems in monitoring this period, particularly with PC applications.

Master Repetition Period (TwM)

The period TwM is valid if a slave answer is not received and for broadcast telegrams.

This period must be shorter than the station slot time (SSL) but no longer than the default period TDF.

Note: **Only the sender of a command telegram (master) waits for a response telegram with a time-out period of TwM**. All pass-through SIS users may neither repeat a command telegram nor send the master a response telegram containing a time-out error.

Default Period After Scanning Has Begun (TDF)

TDF is the period after which a user begins to scan the baud rate in the case of bus inaction (no characters received).

Note: This period input is not a component of the SIS, but is specified on level 7 (ISO model).

11 Pin Assignment and Cables

11.1 Serial Interface Pin Assignment

The 15-pin interface enables the control and data lines to be assigned exclusively to the individual interfaces.

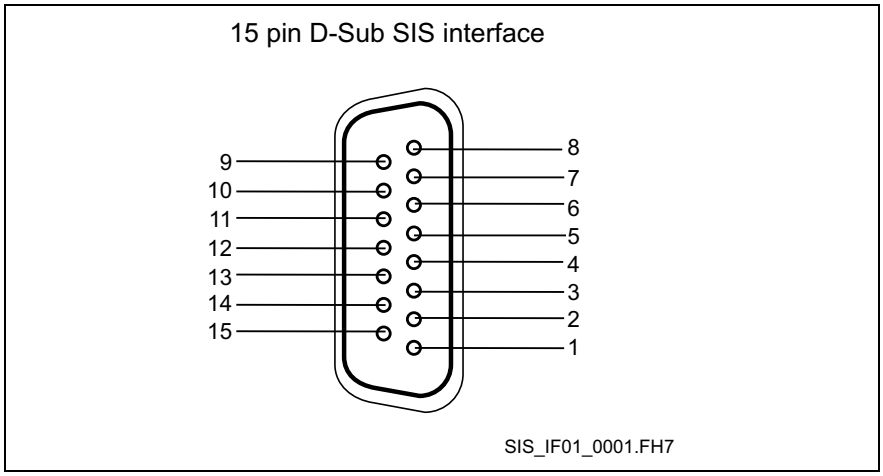


Fig. 11-1: 15 pin D-Sub SIS interface

Pin No.	Modem - 15 / RS232 - 15 / RS422 - 15 / RS485 - 15
1	(Protected Ground)
2	**T**ransmit **D**ata (RS232)
3	**R**eceive **D**ata (RS232)
4	RS485+ or. RxD+ (RS422)
5	RS485- or. RxD- (RS422)
6	**D**ata **S**et **R**eady (Modem)
7	Signal Ground
8	(**D**ata **C**arrier **D**etected) (Modem)
9	TxD+ (RS422)
10	GND
11	TxD- (RS422)
12	+5V
13	**R**equest **T**o **S**end (Modem)
14	**C**lear **T**o **S**end (Modem)
15	(**D**ata **T**erminal **R**eady) (Modem)

Fig. 11-2: Pin assignment of 15-Pin SIS interface

11.2 Cables for the Serial Interface

The cables provided by Rexroth should support approximately 90% of all applications. As required, the cables are designed specifically for certain interfaces. This particularly refers to the Rexroth only cables:

- For continuing a bus loop
- To adapt units with a Micro D-SUB design for the 15-pin interface

Requirements for the Individual Interfaces

RS232 and Modem

Only the connection cables need to be considered for the point-to-point connections.

All 9 lines must be looped through for modem operation:

- Two data lines
- Five control lines
- GND and screen

The two data lines are shielded. An overall shield is also planned.

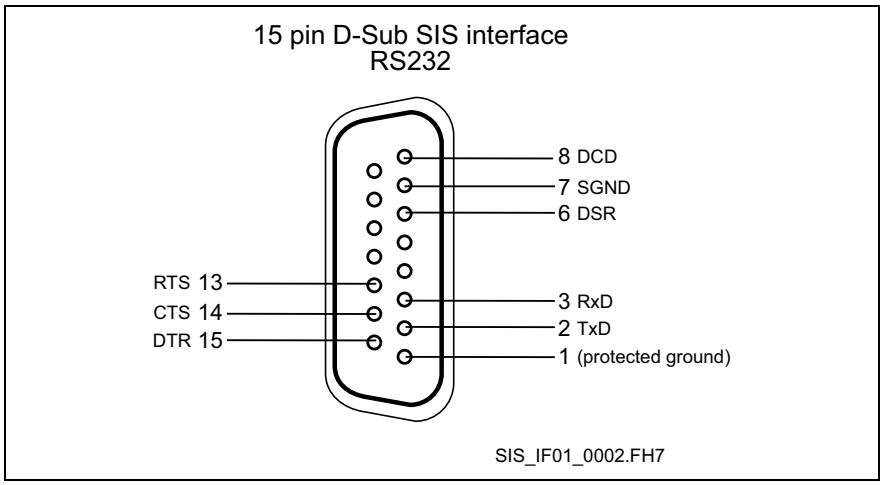


Fig. 11-3: Interface RS232

To connect an external unit to an Rexroth unit with a Micro D-SUB design of the 15-pin interface, the resistance must also be integrated.

Two new cables are therefore required for this interface.

RS485

In addition to the connector cables for this bus interface, cables for continuing the bus loop are also required.

Three lines are proposed for the connector cable:

- Two data lines for half duplex operation
- GND

The two data lines are shielded. An overall shield is also planned.

Pin assignment on the customer side is in accordance with DIN 19245, part 1.

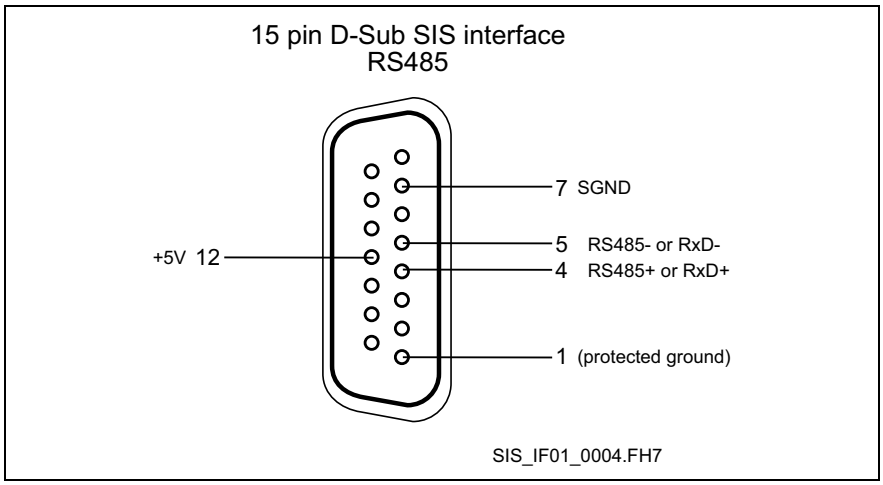


Fig. 11-4: Interface RS485

Pin No.	RS485 Basis	Signal	Meaning
1		SHIELD	Shield or protective ground
3	B/B’	RxD+/TxD+	Reception/transmission - data positive
5	C/C’	DGND	Data reference potential
6		VP	Power supply - plus
8	A/A’	RxD-/TxD-	Reception/transmission - data negative

Fig. 11-5: 9-Pin RS485 pin assignment in accordance with DIN 19245, part 1

To continue the bus, a Y cable and connector cable (for the Y cable) with 5 or 6 wires is planned.

- Two data lines for half duplex operation
- Two additional data lines for full duplex operation via RS422
- GND
- 5V, only for the D-SUB pins connection in the Y cable 02

Both pairs of data lines are individually shielded. An overall shield is also planned.

Therefore, three new cables are required for these interfaces.

RS422

This interface is already taken into account in the cables for looping a bus.

A connector cable for RS422 is currently not planned.

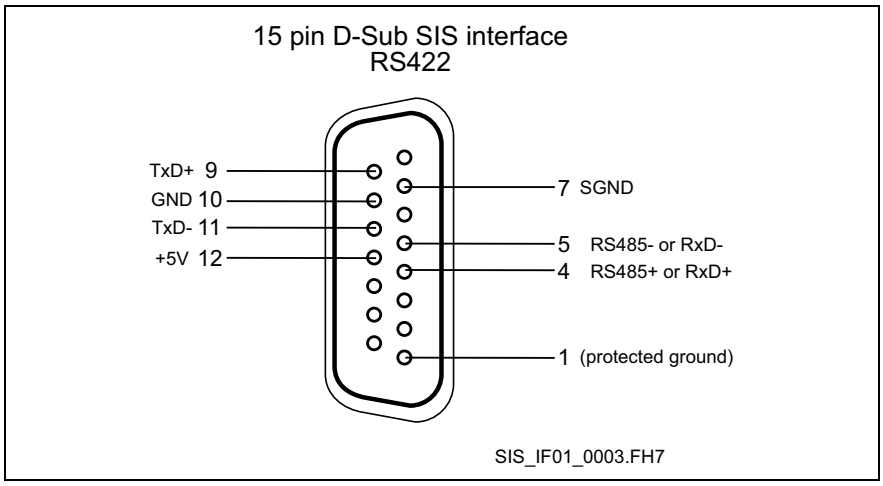


Fig. 11-6: Interface RS422

Micro D-SUB Adaptation

Only one cable is required for all interfaces for adaptation of units with micro D-SUB design. All 15 wires are looped through 1 : 1 with an overall shield.

Partially Ready-Made Cables

The range of cables available is enhanced by a partially ready-made cable. On the Rexroth side all 15 wires are ready-made, but the customer side remains unfinished.

Because of the possibility of incorrect configuration by the customer, these cables should not necessarily be offered as a standard solution.

List of New Cables

No.	Interface	Rexroth 1	Rexroth 2	Rexroth 3	Customer	Remarks
01	Optional	Micro D-SUB, female, 15-pin	D-SUB, female, 15-pin			Adapter for units with micro D-SUB; 15 wire, 1 shield
02	RS485, RS422	D-SUB, male, 15-pin, wire double assignment	D-SUB, female, 15-Pin	D-SUB, male, 15-pin, 6th wire for the 5V		Y cable for the RS485 or RS422 bus; 5/6 wire, 3 shields
03	RS485, RS422	D-SUB, male, 15-pin	D-SUB, female; 15-pin			Connector cables for the Y cables; 5 wire, 3 shields
04	RS232, modem	D-SUB, male, 15-pin			D-SUB, female, 9-Pin	RS232 connection; 9 wire, 2 shields
05	RS485	D-SUB, male, 15-pin			D-SUB, male, 9-pin (see DIN)	RS485 connection; 3 wire, 2 shields
06	RS232, modem	D-SUB, male, 15-Pin			D-SUB, Female, 9-Pin	RS232 Connection; 9 wire, 2 shields, resistor and bridges
07	Optional	D-SUB, male , 15-Pin			Unfinished	Optional configuration for customers possible
08	RS232					Accessory

Fig. 11-7: Cables for the 15-pin SIS interface

Note: Because of poor customer acceptance of the Y cable, a new solution is being sought for this cable.

Note: The list of cables will be revised soon.

Example for a RS485 Bus

The following diagram shows an RS485 bus to which drives of the DIAX04, DIAX03 and ECODRIVE (new) families and a PC are connected via an RS232/RS485 converter. In this example all RS485 cables and an RS232 cable are integrated as accessory parts.

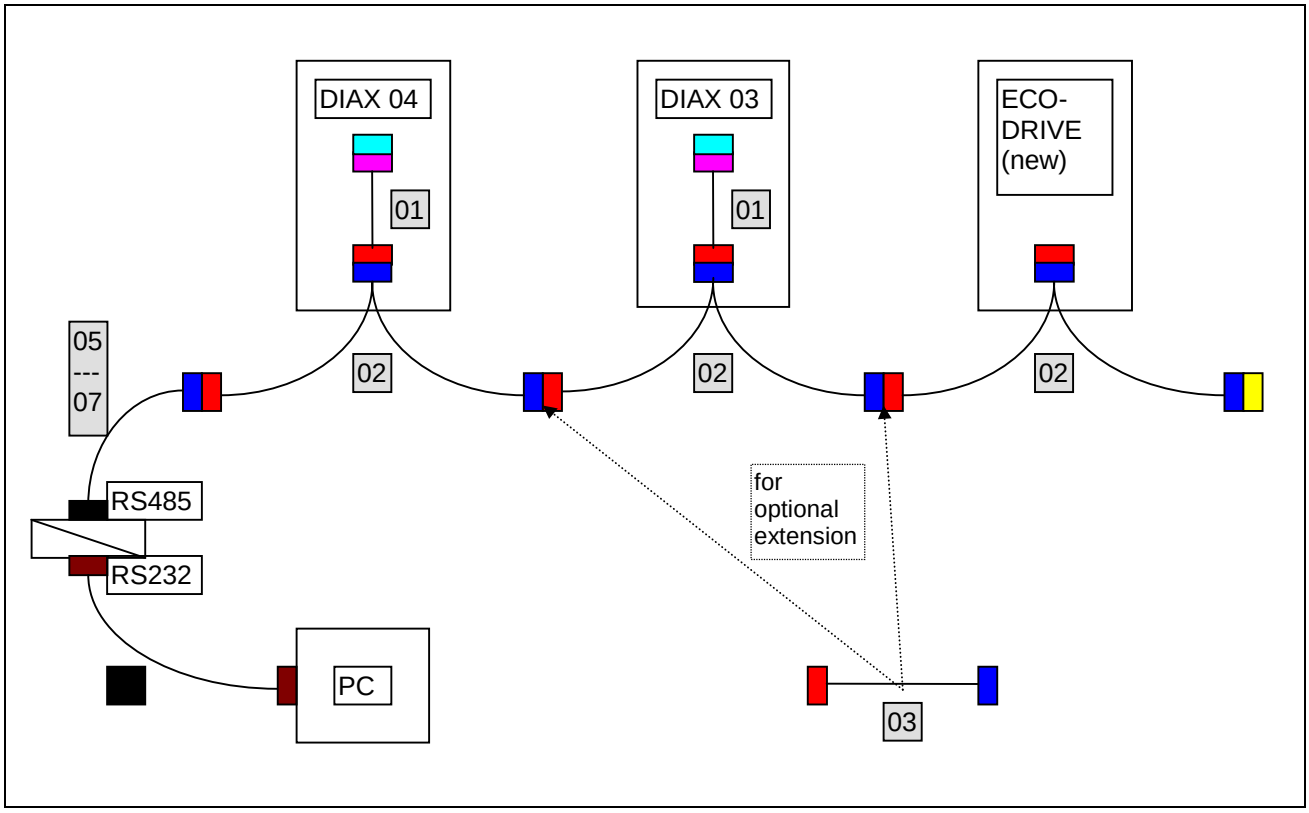


Fig. 11-8: Example: Cable configuration for RS485 bus (and RS422)

- Micro D-SUB, female, 15-pin
- Micro D-SUB, male, 15-pin
- D-SUB, female, 15-pin
- D-SUB, male, 15-pin
- Bus terminator, 15-pin, accessory part
- D-SUB, female, 9-pin
- D-SUB, male, 9-pin
- 01 Rexroth cable number
- Accessory cable number

Examples for RS232 Connections

The following two examples show the cable configurations for a RS232 connection (point-to-point).

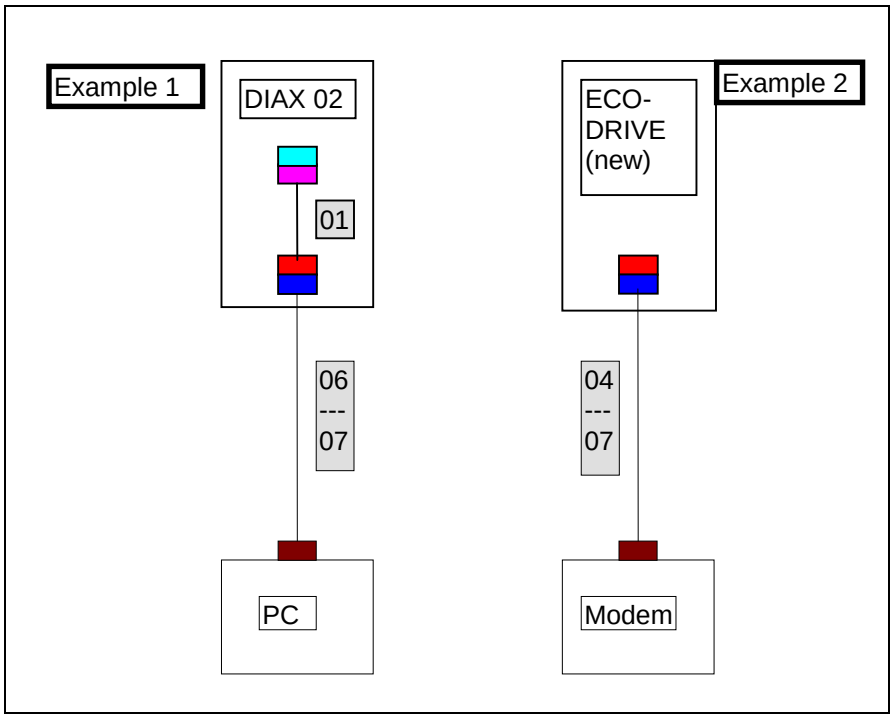


Fig. 11-9: Example: Cable configurations for RS232 and modem

- Micro D-SUB, female, 15-pin
- Micro D-SUB, male, 15-pin
- D-SUB, female, 15-pin
- D-SUB, male, 15-pin
- D-SUB, female, 9-pin
- 01 Rexroth cable number

Designations for Rexroth Cables

Three Rexroth cables have currently been released:

- IKB0003
- IKB0005
- IKB0012

IKB0003 The IKB0003 cable is the 15-pin adapter (1 – 1) from a standard D-Sub to a micro D-Sub (cable no. 01 in the 'List of New Cables', Fig. 11-7).

IKB0005 The IKB0005 cable establishes the RS485 connection between a PC or Interface (9-Pin) and the Rexroth side (15-Pin) – (Cable No. 05 in the 'List of New Cables', Fig. 11-7).

IKB0012 This cable has an identical configuration to IKB0005; on the PC side, it has a special connector for connection to the BTV.

12 Service & Support

12.1 Helpdesk

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main steht Ihnen mit Rat und Tat zur Seite. Sie erreichen uns

Our service helpdesk at our headquarters in Lohr am Main, Germany can assist you in all kinds of inquiries. Contact us

- telefonisch - by phone: **49 (0) 9352 40 50 60**
  über Service Call Entry Center Mo-Fr 07:00-18:00
  - via Service Call Entry Center Mo-Fr 7:00 am - 6:00 pm
- per Fax - by fax: **+49 (0) 9352 40 49 41**
- per e-Mail - by e-mail: **service.svc@boschrexroth.de**

12.2 Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service direkt ansprechbar unter

After helpdesk hours, contact our service department directly at

+49 (0) 171 333 88 26

oder - or **+49 (0) 172 660 04 06**

12.3 Internet

Unter **www.boschrexroth.com** finden Sie ergänzende Hinweise zu Service, Reparatur und Training sowie die **aktuellen** Adressen *) unserer auf den folgenden Seiten aufgeführten Vertriebs- und Servicebüros.

Verkaufsniederlassungen
Niederlassungen mit Kundendienst

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit unserem für Sie nächstgelegenen Ansprechpartner auf.

*) Die Angaben in der vorliegenden Dokumentation können seit Drucklegung überholt sein.

At **www.boschrexroth.com** you may find additional notes about service, repairs and training in the Internet, as well as the **actual** addresses *) of our sales- and service facilities figuring on the following pages.

sales agencies
offices providing service

Please contact our sales / service office in your area first.

*) Data in the present documentation may have become obsolete since printing.

12.4 Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

1. detaillierte Beschreibung der Störung und der Umstände.
2. Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.
3. Tel.-/Faxnummern und e-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind.

For quick and efficient help, please have the following information ready:

1. Detailed description of the failure and circumstances.
2. Information on the type plate of the affected products, especially type codes and serial numbers.
3. Your phone/fax numbers and e-mail address, so we can contact you in case of questions.

12.5 Kundenbetreuungsstellen - Sales & Service Facilities

Deutschland – Germany

vom Ausland: (0) nach Landeskennziffer weglassen!
from abroad: don't dial (0) after country code!

Vertriebsgebiet Mitte Germany Centre	**SERVICE** **CALL ENTRY CENTER** **MO – FR** **von 07:00 - 18:00 Uhr** **from 7 am – 6 pm**	**SERVICE** **HOTLINE** **MO – FR** **von 17:00 - 07:00 Uhr** **from 5 pm - 7 am** **+ SA / SO**	**SERVICE** **ERSATZTEILE / SPARES** verlängerte Ansprechzeit - extended office time - ♦ nur an Werktagen - only on working days - ♦ von 07:00 - 18:00 Uhr - from 7 am - 6 pm -
Rexroth Indramat GmbH Bgm.-Dr.-Nebel-Str. 2 / Postf. 1357 97816 Lohr am Main / 97803 Lohr **Kompetenz-Zentrum Europa** Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885	**Tel. +49 (0) 9352 40 50 60** service.svc@boschrexroth.de	**Tel.: +49 (0)172 660 04 06** **oder / or** **Tel.: +49 (0)171 333 88 26**	**Tel. +49 (0) 9352 40 42 22**
Vertriebsgebiet Süd Germany South	Vertriebsgebiet West Germany West	Gebiet Südwest Germany South-West	
Bosch Rexroth AG Landshuter Allee 8-10 80637 München Tel.: +49 (0)89 127 14-0 Fax: +49 (0)89 127 14-490	Bosch Rexroth AG Regionalzentrum West Borsigstrasse 15 40880 Ratingen Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406 +49 (0)2102 409-430	Bosch Rexroth AG Service-Regionalzentrum Süd-West Siemensstr.1 70736 Fellbach Tel.: +49 (0)711 51046–0 Fax: +49 (0)711 51046–248	
Vertriebsgebiet Nord Germany North	Vertriebsgebiet Mitte Germany Centre	Vertriebsgebiet Ost Germany East	Vertriebsgebiet Ost Germany East
Bosch Rexroth AG Walsroder Str. 93 30853 Langenhagen Tel.: +49 (0) 511 72 66 57-0 Service: +49 (0) 511 72 66 57-256 Fax: +49 (0) 511 72 66 57-93 Service: +49 (0) 511 72 66 57-783	Bosch Rexroth AG Regionalzentrum Mitte Waldecker Straße 13 64546 Mörfelden-Walldorf Tel.: +49 (0) 61 05 702-3 Fax: +49 (0) 61 05 702-444	Bosch Rexroth AG Beckerstraße 31 09120 Chemnitz Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333	Bosch Rexroth AG Regionalzentrum Ost Walter-Köhn-Str. 4d 04356 Leipzig Tel.: +49 (0)341 25 61-0 Fax: +49 (0)341 25 61-111

Europa (West) - Europe (West)

vom Ausland: (0) nach Landeskennziffer weglassen, Italien: 0 nach Landeskennziffer mitwählen
from abroad: don't dial (0) after country code, Italy: dial 0 after country code

Austria - Österreich

Bosch Rexroth GmbH
Electric Drives & Controls
Stachegasse 13
1120 Wien

Tel.: +43 (0)1 985 25 40
Fax: +43 (0)1 985 25 40-93

Austria – Österreich

Bosch Rexroth GmbH
Electric Drives & Controls
Industriepark 18
4061 Pasching

Tel.: +43 (0)7221 605-0
Fax: +43 (0)7221 605-21

Belgium - Belgien

Bosch Rexroth NV/SA
Henri Genessestraat 1
1070 Bruxelles

Tel: +32 (0) 2 582 31 80
Fax: +32 (0) 2 582 43 10
info@boschrexroth.be
service@boschrexroth.be

Denmark - Dänemark

BEC A/S
Zinkvej 6
8900 Randers

Tel.: +45 (0)87 11 90 60
Fax: +45 (0)87 11 90 61

Great Britain – Großbritannien

Bosch Rexroth Ltd.
Electric Drives & Controls
Broadway Lane, South Cerney
Cirencester, Glos GL7 5UH

Tel.: +44 (0)1285 863000
Fax: +44 (0)1285 863030
sales@boschrexroth.co.uk
service@boschrexroth.co.uk

Finland - Finnland

Bosch Rexroth Oy
Electric Drives & Controls
Ansatie 6
017 40 Vantaa

Tel.: +358 (0)9 84 91-11
Fax: +358 (0)9 84 91-13 60

France - Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
Avenue de la Trentaine
(BP. 74)
77503 Chelles Cedex

Tel.: +33 (0)164 72-70 00
Fax: +33 (0)164 72-63 00
Hotline: +33 (0)608 33 43 28

France - Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
ZI de Thibaud, 20 bd. Thibaud
(BP. 1751)
31084 Toulouse

Tel.: +33 (0)5 61 43 61 87
Fax: +33 (0)5 61 43 94 12

France – Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
91, Bd. Irène Joliot-Curie
69634 Vénissieux – Cedex

Tel.: +33 (0)4 78 78 53 65
Fax: +33 (0)4 78 78 53 62

Italy - Italien

Bosch Rexroth S.p.A.
Via G. Di Vittorio, 1
20063 Cernusco S/N.MI

Hotline: +39 02 92 365 563
Tel.: +39 02 92 365 1
Service: +39 02 92 365 326
Fax: +39 02 92 365 500
Service: +39 02 92 365 503

Italy - Italien

Bosch Rexroth S.p.A.
Via Paolo Veronesi, 250
10148 Torino

Tel.: +39 011 224 88 11
Fax: +39 011 224 88 30

Italy - Italien

Bosch Rexroth S.p.A.
Via Mascia, 1
80053 Castellamare di Stabia NA

Tel.: +39 081 8 71 57 00
Fax: +39 081 8 71 68 85

Italy - Italien

Bosch Rexroth S.p.A.
Via del Progresso, 16 (Zona Ind.)
35020 Padova

Tel.: +39 049 8 70 13 70
Fax: +39 049 8 70 13 77

Italy - Italien

Bosch Rexroth S.p.A.
Via Isonzo, 61
40033 Casalecchio di Reno (Bo)

Tel.: +39 051 29 86 430
Fax: +39 051 29 86 490

Netherlands - Niederlande/Holland

Bosch Rexroth Services B.V.
Technical Services
Kruisbroeksestraat 1
(P.O. Box 32)
5281 RV Boxtel

Tel.: +31 (0) 411 65 16 40
+31 (0) 411 65 17 27
Fax: +31 (0) 411 67 78 14
+31 (0) 411 68 28 60
services@boschrexroth.nl

Netherlands – Niederlande/Holland

Bosch Rexroth B.V.
Kruisbroeksestraat 1
(P.O. Box 32)
5281 RV Boxtel

Tel.: +31 (0) 411 65 19 51
Fax: +31 (0) 411 65 14 83
www.boschrexroth.nl

Norway - Norwegen

Bosch Rexroth AS
Electric Drives & Controls
Berghagan 1 or: Box 3007
1405 Ski-Langhus 1402 Ski

Tel.: +47 (0) 64 86 41 00

Fax: +47 (0) 64 86 90 62

Hotline: +47 (0)64 86 94 82
jul.ruud@rexroth.no

Spain - Spanien

Bosch Rexroth S.A.
Electric Drives & Controls
Centro Industrial Santiga
Obradors s/n
08130 Santa Perpetua de Mogoda
Barcelona

Tel.: +34 9 37 47 94 00
Fax: +34 9 37 47 94 01

Spain – Spanien

Goimendi S.A.
Electric Drives & Controls
Parque Empresarial Zuatzu
C/ Francisco Grandmontagne no.2
20018 San Sebastian

Tel.: +34 9 43 31 84 21
- service: +34 9 43 31 84 56
Fax: +34 9 43 31 84 27
- service: +34 9 43 31 84 60
sat.indramat@goimendi.es

Sweden - Schweden

Bosch Rexroth AB
Electric Drives & Controls
- Varuvägen 7
(Service: Konsumentvägen 4, Älfsjö)
125 81 Stockholm

Tel.: +46 (0)8 727 92 00
Fax: +46 (0)8 647 32 77

Sweden - Schweden

Bosch Rexroth AB
Electric Drives & Controls
Ekvändan 7
254 67 Helsingborg

Tel.: +46 (0) 42 38 88 -50
Fax: +46 (0) 42 38 88 -74

Switzerland East - Schweiz Ost

Bosch Rexroth Schweiz AG
Electric Drives & Controls
Hemrietstrasse 2
8863 Buttikon

Tel. +41 (0) 55 46 46 111
Fax +41 (0) 55 46 46 222

Switzerland West - Schweiz West

Bosch Rexroth Suisse SA
Av. Général Guisan 26
1800 Vevey 1

Tel.: +41 (0)21 632 84 20
Fax: +41 (0)21 632 84 21

Europa (Ost) - Europe (East)

vom Ausland: (0) nach Landeskennziffer weglassen
from abroad: don't dial (0) after country code

Czech Republic - Tschechien Bosch -Rexroth, spol.s.r.o. Hviezdoslavova 5 627 00 Brno Tel.: +420 (0)5 48 126 358 Fax: +420 (0)5 48 126 112	Czech Republic - Tschechien DEL a.s. Strojírenská 38 591 01 Zdar nad Sázavou Tel.: +420 566 64 3144 Fax: +420 566 62 1657	Hungary - Ungarn Bosch Rexroth Kft. Angol utca 34 1149 Budapest Tel.: +36 (1) 422 3200 Fax: +36 (1) 422 3201	Poland – Polen Bosch Rexroth Sp.zo.o. ul. Staszica 1 05-800 Pruszków Tel.: +48 22 738 18 00 – service: +48 22 738 18 46 Fax: +48 22 758 87 35 – service: +48 22 738 18 42
Poland – Polen Bosch Rexroth Sp.zo.o. Biuro Poznan ul. Dabrowskiego 81/85 60-529 Poznan Tel.: +48 061 847 64 62 /-63 Fax: +48 061 847 64 02	Romania - Rumänien East Electric S.R.L. Bdul Basarabia no.250, sector 3 73429 Bucuresti Tel./Fax:: +40 (0)21 255 35 07 +40 (0)21 255 77 13 Fax: +40 (0)21 725 61 21 eastel@rdsnet.ro	Romania - Rumänien Bosch Rexroth Sp.zo.o. Str. Drobety nr. 4-10, app. 14 70258 Bucuresti, Sector 2 Tel.: +40 (0)1 210 48 25 +40 (0)1 210 29 50 Fax: +40 (0)1 210 29 52	Russia - Russland Bosch Rexroth OOO Wjatskaja ul. 27/15 127015 Moskau Tel.: +7-095-785 74 78 +7-095 785 74 79 Fax: +7 095 785 74 77 laura.kanina@boschrexroth.ru
Russia - Russland ELMIS 10, Internationalnaya 246640 Gomel, Belarus Tel.: +375/ 232 53 42 70 +375/ 232 53 21 69 Fax: +375/ 232 53 37 69 elmis_ltd@yahoo.com	Turkey - Türkei Bosch Rexroth Otomasyon San & Tic. A..S. Fevzi Cakmak Cad No. 3 34630 Sefaköy Istanbul Tel.: +90 212 413 34 00 Fax: +90 212 413 34 17 www.boschrexroth.com.tr	Turkey - Türkei Servo Kontrol Ltd. Sti. Perpa Ticaret Merkezi B Blok Kat: 11 No: 1609 80270 Okmeydani-Istanbul Tel: +90 212 320 30 80 Fax: +90 212 320 30 81 remzi.sali@servokontrol.com www.servokontrol.com	Slowenia - Slowenien DOMEL Otoki 21 64 228 Zelezniki Tel.: +386 5 5117 152 Fax: +386 5 5117 225 brane.ozebek@domel.si

Africa, Asia, Australia – incl. Pacific Rim

Australia - Australien AIMS - Australian Industrial Machinery Services Pty. Ltd. 28 Westside Drive Laverton North Vic 3026 Melbourne Tel.: +61 3 93 14 3321 Fax: +61 3 93 14 3329 **Hotlines:** +61 3 93 14 3321 +61 4 19 369 195 enquires@aimservices.com.au	Australia - Australien Bosch Rexroth Pty. Ltd. No. 7, Endeavour Way Braeside Victoria, 31 95 Melbourne Tel.: +61 3 95 80 39 33 Fax: +61 3 95 80 17 33 mel@rexroth.com.au	China Shanghai Bosch Rexroth Hydraulics & Automation Ltd. Waigaoqiao, Free Trade Zone No.122, Fu Te Dong Yi Road Shanghai 200131 - P.R.China Tel.: +86 21 58 66 30 30 Fax: +86 21 58 66 55 23 richard.yang_sh@boschrexroth.com.cn gf.zhu_sh@boschrexroth.com.cn	China Shanghai Bosch Rexroth Hydraulics & Automation Ltd. 4/f, Marine Tower No.1, Pudong Avenue Shanghai 200120 - P.R.China Tel: +86 21 68 86 15 88 Fax: +86 21 58 40 65 77
China Bosch Rexroth China Ltd. 15/F China World Trade Center 1, Jianguomenwai Avenue Beijing 100004, P.R.China Tel.: +86 10 65 05 03 80 Fax: +86 10 65 05 03 79	China Bosch Rexroth China Ltd. Guangzhou Repres. Office Room 1014-1016, Metro Plaza, Tian He District, 183 Tian He Bei Rd Guangzhou 510075, P.R.China Tel.: +86 20 8755-0030 +86 20 8755-0011 Fax: +86 20 8755-2387	China Bosch Rexroth (China) Ltd. A-5F., 123 Lian Shan Street Sha He Kou District Dalian 116 023, P.R.China Tel.: +86 411 46 78 930 Fax: +86 411 46 78 932	China Melchers GmbH BRC-SE, Tightening & Press-fit 13 Floor Est Ocean Centre No.588 Yanan Rd. East 65 Yanan Rd. West Shanghai 200001 Tel.: +86 21 6352 8848 Fax: +86 21 6351 3138
Hongkong Bosch Rexroth (China) Ltd. 6th Floor, Yeung Yiu Chung No.6 Ind Bldg. 19 Cheung Shun Street Cheung Sha Wan, Kowloon, Hongkong Tel.: +852 22 62 51 00 Fax: +852 27 41 33 44 alexis.siu@boschrexroth.com.hk	India - Indien Bosch Rexroth (India) Ltd. Electric Drives & Controls Plot. No.96, Phase III Peenya Industrial Area Bangalore – 560058 Tel.: +91 80 51 17 0-211...-218 Fax: +91 80 83 94 345 +91 80 83 97 374 mohanvelu.t@boschrexroth.co.in	India - Indien Bosch Rexroth (India) Ltd. Electric Drives & Controls Advance House, II Floor Ark Industrial Compound Narol Naka, Makwana Road Andheri (East), Mumbai - 400 059 Tel.: +91 22 28 56 32 90 +91 22 28 56 33 18 Fax: +91 22 28 56 32 93 singh.op@boschrexroth.co.in	India - Indien Bosch Rexroth (India) Ltd. S-10, Green Park Extension New Delhi – 110016 Tel.: +91 11 26 56 65 25 +91 11 26 56 65 27 Fax: +91 11 26 56 68 87 koul.rp@boschrexroth.co.in
Indonesia - Indonesien PT. Bosch Rexroth Building # 202, Cilandak Commercial Estate Jl. Cilandak KKO, Jakarta 12560 Tel.: +62 21 7891169 (5 lines) Fax: +62 21 7891170 - 71 rudy.karimun@boschrexroth.co.id	Japan Bosch Rexroth Automation Corp. Service Center Japan Yutakagaoka 1810, Meito-ku, NAGOYA 465-0035, Japan Tel.: +81 52 777 88 41 +81 52 777 88 53 +81 52 777 88 79 Fax: +81 52 777 89 01	Japan Bosch Rexroth Automation Corp. Electric Drives & Controls 2F, I.R. Building Nakamachidai 4-26-44, Tsuzuki-ku YOKOHAMA 224-0041, Japan Tel.: +81 45 942 72 10 Fax: +81 45 942 03 41	Korea Bosch Rexroth-Korea Ltd. Electric Drives and Controls Bongwoo Bldg. 7FL, 31-7, 1Ga Jangchoong-dong, Jung-gu Seoul, 100-391 Tel.: +82 234 061 813 Fax: +82 222 641 295
Korea Bosch Rexroth-Korea Ltd. 1515-14 Dadae-Dong, Saha-gu Electric Drives & Controls Pusan Metropolitan City, 604-050 Tel.: +82 51 26 00 741 Fax: +82 51 26 00 747 eunkyong.kim@boschrexroth.co.kr	Malaysia Bosch Rexroth Sdn.Bhd. 11, Jalan U8/82, Seksyen U8 40150 Shah Alam Selangor, Malaysia Tel.: +60 3 78 44 80 00 Fax: +60 3 78 45 48 00 hockhwa@hotmail.com rexroth1@tm.net.my	Singapore - Singapur Bosch Rexroth Pte Ltd 15D Tuas Road Singapore 638520 Tel.: +65 68 61 87 33 Fax: +65 68 61 18 25 sanjay.nemade @boschrexroth.com.sg	South Africa - Südafrika TECTRA Automation (Pty) Ltd. 71 Watt Street, Meadowdale Edenvale 1609 Tel.: +27 11 971 94 00 Fax: +27 11 971 94 40 **Hotline:** +27 82 903 29 23 georgv@tectra.co.za
Taiwan Bosch Rexroth Co., Ltd. Taichung Branch 1F., No. 29, Fu-Ann 5th Street, Xi-Tun Area, Taichung City Taiwan, R.O.C. Tel : +886 - 4 -23580400 Fax: +886 - 4 -23580402 jim.lin@boschrexroth.com.tw david.lai@boschrexroth.com.tw	Taiwan Bosch Rexroth Co., Ltd. Tainan Branch No. 17, Alley 24, Lane 737 Chung Cheng N.Rd. Yungkang Tainan Hsien, Taiwan, R.O.C. Tel : +886 - 6 –253 6565 Fax: +886 - 6 –253 4754 charlie.chen@boschrexroth.com.tw	Thailand NC Advance Technology Co. Ltd. 59/76 Moo 9 Ramintra road 34 Tharang, Bangkhen, Bangkok 10230 Tel.: +66 2 943 70 62 +66 2 943 71 21 Fax: +66 2 509 23 62 Hotline +66 1 984 61 52 sonkawin@hotmail.com	

Nordamerika – North America

USA Headquarters - Hauptniederlassung Bosch Rexroth Corporation Electric Drives & Controls 5150 Prairie Stone Parkway Hoffman Estates, IL 60192-3707 Tel.: +1 847 6 45 36 00 Fax: +1 847 6 45 62 01 servicebrc@boschrexroth-us.com repairbrc@boschrexroth-us.com	USA Central Region - Mitte Bosch Rexroth Corporation Electric Drives & Controls Central Region Technical Center 1701 Harmon Road Auburn Hills, MI 48326 Tel.: +1 248 3 93 33 30 Fax: +1 248 3 93 29 06	USA Southeast Region - Südwest Bosch Rexroth Corporation Electric Drives & Controls Southeastern Technical Center 3625 Swiftwater Park Drive Suwanee, Georgia 30124 Tel.: +1 770 9 32 32 00 Fax: +1 770 9 32 19 03	**USA SERVICE-HOTLINE** - 7 days x 24hrs - **+1-800-REX-ROTH** **+1 800 739 7684**
USA East Region – Ost Bosch Rexroth Corporation Electric Drives & Controls Charlotte Regional Sales Office 14001 South Lakes Drive Charlotte, North Carolina 28273 Tel.: +1 704 5 83 97 62 +1 704 5 83 14 86	USA Northeast Region – Nordost Bosch Rexroth Corporation Electric Drives & Controls Northeastern Technical Center 99 Rainbow Road East Granby, Connecticut 06026 Tel.: +1 860 8 44 83 77 Fax: +1 860 8 44 85 95	USA West Region – West Bosch Rexroth Corporation 7901 Stoneridge Drive, Suite 220 Pleasant Hill, California 94588 Tel.: +1 925 227 10 84 Fax: +1 925 227 10 81	
Canada East - Kanada Ost Bosch Rexroth Canada Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Tel.: +1 905 335 5511 Fax: +1 905 335 4184 **Hotline:** +1 905 335 5511 michael.moro@boschrexroth.ca	Canada West - Kanada West Bosch Rexroth Canada Corporation 5345 Goring St. Burnaby, British Columbia Canada V7J 1R1 Tel. +1 604 205 5777 Fax +1 604 205 6944 **Hotline:** +1 604 205 5777 david.gunby@boschrexroth.ca	Mexico Bosch Rexroth Mexico S.A. de C.V. Calle Neptuno 72 Unidad Ind. Vallejo 07700 Mexico, D.F. Tel.: +52 55 57 54 17 11 Fax: +52 55 57 54 50 73 mariofelipe.hernandez@boschrexroth.com.mx	Mexico Bosch Rexroth S.A. de C.V. Calle Argentina No 3913 Fracc. las Torres 64930 Monterrey, N.L. Tel.: +52 81 83 65 22 53 +52 81 83 65 89 11 +52 81 83 49 80 91 Fax: +52 81 83 65 52 80 mario.quiroga@boschrexroth.com.mx

Südamerika – South America

Argentina - Argentinien Bosch Rexroth S.A.I.C. "The Drive & Control Company" Rosario 2302 B1606DLD Carapachay Provincia de Buenos Aires Tel.: +54 11 4756 01 40 +54 11 4756 02 40 +54 11 4756 03 40 +54 11 4756 04 40 Fax: +54 11 4756 01 36 +54 11 4721 91 53 victor.jabif@boschrexroth.com.ar	Argentina - Argentinien NAKASE Servicio Tecnico CNC Calle 49, No. 5764/66 B1653AOX Villa Balester Provincia de Buenos Aires Tel.: +54 11 4768 36 43 Fax: +54 11 4768 24 13 **Hotline:** +54 11 155 307 6781 nakase@usa.net nakase@nakase.com gerencia@nakase.com (Service)	Brazil - Brasilien Bosch Rexroth Ltda. Av. Tégula, 888 Ponte Alta, Atibaia SP CEP 12942-440 Tel.: +55 11 4414 56 92 +55 11 4414 56 84 Fax sales: +55 11 4414 57 07 Fax serv.: +55 11 4414 56 86 alexandre.wittwer@rexroth.com.br	Brazil - Brasilien Bosch Rexroth Ltda. R. Dr.Humberto Pinheiro Vieira, 100 Distrito Industrial [Caixa Postal 1273] 89220-390 Joinville - SC Tel./Fax: +55 47 473 58 33 Mobil: +55 47 9974 6645 prochnow@zaz.com.br
Columbia - Kolumbien Reflutec de Colombia Ltda. Calle 37 No. 22-31 Santafé de Bogotá, D.C. Colombia Tel.: +57 1 368 82 67 +57 1 368 02 59 Fax: +57 1 268 97 37 reflutec@neutel.com.co reflutec@007mundo.com			

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 (0)93 52-40-50 60
Fax +49 (0)93 52-40-49 41
service.svc@boschrexroth.de
www.boschrexroth.com

R911289718

Printed in Germany
DOK-GENERL-SIS-DEFINIT-IF02-EN-P