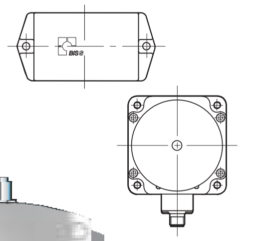
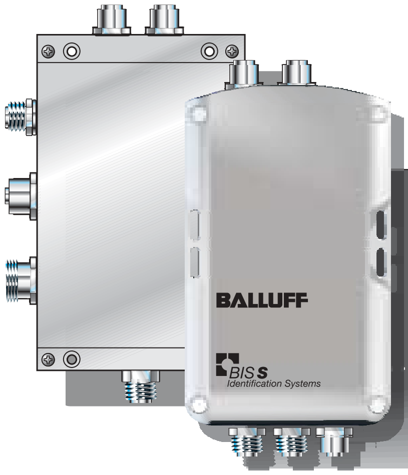
1

BALLUFF





Manual

Electronic Identification Systems BIS
Processor BIS S-60_2
Profibus DP

Deutsch – bitte wenden!

2

No. 828 318 D/E • Edition 0303
Subject to modification.

http://www.balluff.de

Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Germany
Phone +49 (0) 71 58/1 73-0
Fax +49 (0) 71 58/50 10
E-Mail: balluff@balluff.de

Contents

Safety Considerations

Approved Operation

Series BIS S-60_2 processors along with the other BIS S system components comprise an identification system and may only be used for this purpose in an industrial environment in conformity with Class A of the EMC Law.

Installation and Operation

Installation and operation should be carried out by trained personnel only. Unauthorized work and improper use will void the warranty and liability.

When installing the processor, follow the chapters containing the wiring diagrams closely. Special care is required when connecting the processor to external controllers, in particular with respect to selection and polarity of the signals and power supply.

Only approved power supplies may be used for powering the processor. See chapter 'Technical Data' for details.

Use and Checking

Prevailing safety regulations must be adhered to when using the identification system. In particular, steps must be taken to ensure that a failure of or defect in the identification system does not result in hazards to persons or equipment.

This includes maintaining the specified ambient conditions and regular testing for functionality of the identification system including all its associated components.

Fault Conditions

Should there ever be indications that the identification system is not working properly, it should be taken out of commission and secured from unauthorized use.

Scope

This manual applies to processors in the series BIS S-6002-019-050-03-ST11 and BIS S-6022-019-050-03-ST14.

Introduction
BIS S Identification Systems

This manual is designed to assist the user in setting up the control program and installing and starting up the components of the BIS S Identification System, and to assure rapid, trouble-free operation.

Principles

The BIS S Identification Systems belongs in the category of

non-contact systems for reading and writing.

This dual function permits applications for not only transporting information in fixed-programmed Data carriers, but also for gathering and passing along up-to-date information as well.

☞ If 2 read/write heads are connected to a BIS S-60_2 processor, both heads can be operated independently of each other. This means for example that you can read a Data carrier from one head while writing to another Data carrier at the other head.

Applications

Some of the notable areas of application include

– **for controlling material flow in production processes**
(e.g. in model-specific processes),
for workpiece conveying in transfer lines,
in data gathering for quality assurance,
for gathering safety-related data,
– **in equipment organization;**
– **in storage systems for monitoring inventory movement;**
– **in transporting and conveying systems;**
– **in waste management for quantity-based fee assessment.**

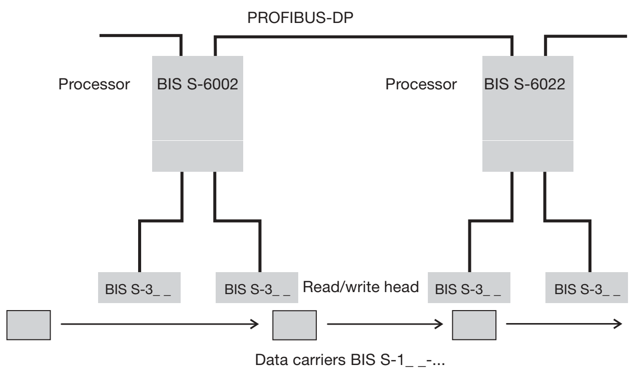
Introduction
BIS S Identification Systems

System Components

The main components of the BIS S Identification Systems are:

– **Processor,**
– **Read/Write Heads and**
– **Data carriers**

Configuration with BIS S-6002 and BIS S-6022 processor



Schematic representation of an Identification System (example)

BIS S-60_2 Processor
Basic knowledge for application

Selecting System Components

The **BIS S-6002** processor has a plastic housing.

The **BIS S-6022** processor has a metal housing.

Connection is made through round connectors. Two read/write heads can be cable connected.

Series BIS S-60_2 processors have in addition a digital input. The input has various functions depending on the configuration (see Parametering).

The read/write distances depend on which data carriers are used. Additional information on the read/write heads in series BIS S-3_ _ including all the possible data carrier/read-write head combinations can be found in the manuals for the respective read/write heads.

The system components are electrically supplied by the processor. The data carrier represents a free-standing unit and needs no line-carried power. It receives its energy from the read/write head. The latter constantly sends out a carrier signal which supplies the code head as soon as the required distance between the two is reached. The read/write operation takes place during this phase. Reading and writing may be dynamic or static.

BIS S-60_2 Processor
Basic knowledge for application

Control Function

The processor writes data from the host system to the Data carrier or reads data from the tag through the read/write head and prepares it for the host system. Host systems may include:

– **a host computer (e.g. industrial PC) or**
– **a programmable logic controller (PLC)**

Data checking

When sending data between the read/write head and the Data carrier a procedure is required for recognizing whether the data were correctly read or written.

The processor is supplied with standard Balluff procedure of double reading and comparing. In addition to this procedure a second alternative is available: CRC_16 data checking.

Here a test code is written to the Data carrier, allowing data to be checked for validity at any time or location.

Advantages of CRC_16	Advantages of double reading
Data checking even during the non-active phase (CT outside read/write head zone).	No bytes on the data carrier need to be reserved for storing a check code.
Shorter read times since each page is read only once.	Shorter write times since no CRC needs to be written.

Since both variations have their advantages depending on the application, the user is free to select which method of data checking he wishes to use (see Parametering on 23).

> It is not permitted to operate the system using both check procedures!

BUS interface PROFIBUS-DP

PROFIBUS-DP

Communication between the BIS S-60_2 processor and the host system is via PROFIBUS-DP.

The PROFIBUS-DP system consists of the components:

– the bus master and
– the bus modules/slaves (here the BIS S-60_2 processor).

☞

Important hints for use with PLC:

In some control systems the PROFIBUS-DP data area is not synchronously transmitted with the updating of the input/output content. If more than 2 bytes of data are sent, a mechanism must be used which guarantees that the data in the PLC and the data in the BIS S are always identical!

1st alternative: Synchronous data transmission as a setting on the Master
In this method the bus Master ensures that all the data necessary for the respective Slave are always sent contiguously. There is usually a special software function in the PLC which likewise controls access between the PLC and bus Master so that data are always sent contiguously.

2nd alternative: Set 2nd bit header
Data exchange between PLC and BIS is controlled by the so-called bit header. This is always the first byte of the respective read/write head in the data buffer. This bit header exists both in the input range (data from BIS to the PLC) and in the output range (data from the PLC to the BIS). IIf this bit header is also sent as the last byte, a comparison of these two bytes can be used to guarantee the consistency of the transmitted data.

In this method the PLC cycle is unaffected nor is the bus access time changed. All that is required is that a byte in the data buffer be used for the 2nd bit header instead of for user data.

This 2nd alternative is the Balluff recommended setting (factory default).

BUS interface PROFIBUS-DP

Unit's Master Data

For the correct parametering of the bus master as per type, CD ROM, containing the unit's master data in the form of a GSD file is included with the BIS S-60_2 processor.

Station Address

The Processor BIS S-60_2 is delivered with the station address 126. This has to be set individually before using in a bus system. See information on 🗋 11.

Input/Output Buffer

An input buffer and an output buffer are used for the data exchange with the control system. The size of these buffers has to be configured via the master.

☞ The possible settings are entered in the GSD file (and Type file). A minimum of 4 and a maximum of 128 bytes can be accommodated. However, it must be an even number.

Parametering Bytes
User-Parameter Bytes

Besides, in the case of the BIS S-60_2 processor, there are 6 further bytes (User-Parameter Bytes) which have to be set while parametering. The significance of the 6 bytes for parametering is described starting from 🗋 22.

☞ The preset is stored in the GSD file.

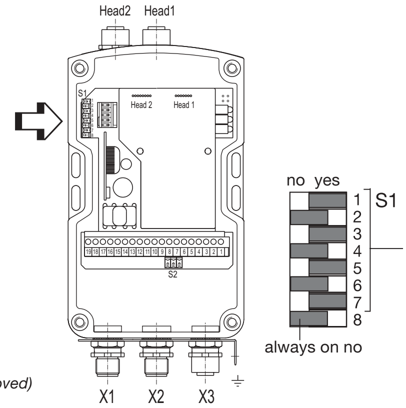
BUS interface PROFIBUS-DP

Station Address setting

The station address under which the unit is accessed on the bus can be assigned through the slide switch S1. Each address shall be assigned only once.

The slide switch S1 is binary coded. The setting of the station address is carried out according to the scheme shown in the table. Switch position: no = left, yes = right.

The address 85 is set in the following figure.



Slide switch S1 (with cover removed)

Station Address	Slide switch S1						
	7	6	5	4	3	2	1
	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	not allowed						
1	no	no	no	no	no	no	yes
2	no	no	no	no	no	yes	no
3	no	no	no	no	no	yes	yes
4	no	no	no	no	yes	no	no
5	no	no	no	no	yes	no	yes
...							
85	yes	no	yes	no	yes	no	yes
...							
123	yes	yes	yes	yes	no	yes	yes
124	yes	yes	yes	yes	yes	no	no
125	yes	yes	yes	yes	yes	no	yes
126	yes	yes	yes	yes	yes	yes	no
127	not allowed						

To open the cover of the processor, see 52 for BIS S-6002 or 62 for BIS S-6022.

Function Description
Communication with the processor

Basic Procedure

Communication between the host system and the processor takes place using a fixed protocol sequence. Data integrity from the control to the processor and vice-versa is indicated by a control bit. This bit is used to implement a handshake between the control and the processor.

Following is a simplified representation of the sequence of a job sent from the control to the processor:

1. The control sends a command designator to the processor together with the associated command parameters and sets a bit (AV bit). This bit indicates to the processor that the transmitted data are valid and that the job is now beginning.
2. The processor takes the job and sets a bit (AA bit), which indicates this to the control.
3. If an additional exchange of data between the control and the processor is required to carry out the job, each uses a bit (TI bit and TO bit) to indicate that the control / processor is now ready for additional data exchange or has accepted the received data.
4. Once the processor has carried out the job correctly, it sets a bit (AE bit).
5. Once the control has accepted all the important data, it indicates this to the processor by resetting the bit that was set at the beginning (AV bit).
6. The processor now in turn sets all the control bits that were set during the sequence (AA bit, AE bit) and is ready for the next job.

Please see also 25...31 and the examples on 32...47.

Function Description
Input and Output Buffers

Input and Output Buffers

In order to transmit commands and data between the BIS S-60_2 and the host system, the latter must prepare two fields. These two fields are:

– **the output buffer**
  for the control commands which are sent **to** the BIS Identification System and
  for the data to be written.

– **the input buffer**
  for the data to be read and
  for the designators and error codes which come **from** the BIS Identification System.

The possible setting values are stored in the GSD file.

The buffer size can be selected between 4 and 128 bytes in steps of 2 bytes. This must be given by the master during parametering. The total buffer size is divided into 2 ranges:

Buffer range 1 for Read/Write Head 1; size is specified in paramter byte 6.
Buffer range 2 for Read/Write Head 2; size = total buffer size – buffer size of Read/Write Head 1.
See 🗋 14 for example.

☞ If a buffer size of less than 8 bytes is set for a read/write head, a read/write request can be carried out without specifying the start address and the number of bytes. Automatic reading for Codetag present (see 🗋 26) remains active. This permits fast reading of small data quantities without placing an unnecessary load on the bus.

Buffer size – 1 = number of bytes read without double bit header;
Buffer size – 2 = number of bytes read with double bit header.

Please note the basic procedure on 🕮 12 and 25...31 and the examples on pages 🕮 32...47.

Function Description
Input and Output Buffers

Input and Output Buffers (continued)

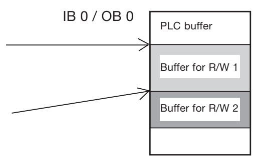
Example: The 82 bytes for the total buffer need to be distributed. An input/output buffer of 46 bytes is assigned to Read/Write Head 1. This results in an input/output buffer of 36 bytes for Read/Write Head 2.

Procedure: The buffer size for Read/Write Head 1 is set to 46 bytes. This means using the parameter byte 6 to enter Hex value 2E (corresponds to 46 decimal), which corresponds to binary 0**010111**0.

PLC Organisation: The buffer range starts at input byte IB 32 and output byte OB 32.

Result:

Read/Write Head 1: (R/W 1)	Subaddress 00	IB 32 and OB 32
	Input buffer	IB 32 to IB 77
	Output buffer	OB 32 to OB 77
Read/Write Head 2: (R/W 2)	Subaddress 00	IB 78 and OB 78
	Input buffer	IB 78 to IB 113
	Output buffer	OB 78 to OB 113



☞ Note that these buffers can be in two different sequences depending on the type of control.

The following description is based on sequence 1!

Sequence 1	Sequence 2
Subaddress 00	Subaddress 01
01	00
02	03
03	02
04	05
05	04
06	07
07	06

Please note the basic procedure on 🕮 12 and 25...31 and the examples on pages 🕮 32...47.

Function Description
Output buffer, configuration and explanation

Configuration of the output buffer for one (1) read/write head

The last byte can be arranged as a 2nd bit header through parametering (default).

Subaddress \ Bit No.	7	6	5	4	3	2	1	0	
00Hex = Bit Header	CT	TI				GR		AV	Bit Name
01Hex	Command Designator					or	Data		
02Hex	Start Address (Low Byte) or Program No.					or	Data		
03Hex	Start Address (High Byte)					or	Data		
04Hex	No. of Bytes (Low Byte)					or	Data		
05Hex	No. of Bytes (High Byte)					or	Data		
06Hex	Data								
...	Data								
Last Byte	2nd Bit Header (as above)					or	Data		

Description of Output Buffer

Sub-address	Bit Name	Meaning	Function Description
00Hex Bit Header	CT	Data carrier type 1	Select Data carrier type: 64 Byte block size — for Data carrier type: BIS S-1_ _-32, -42
	TI	Toggle-Bit In	Shows during a read action that the controller is ready for additional data.
	GR	Ground state	Causes the BIS system to go to the ground state for the respective read/write head. Any pending command is cancelled.
	AV	Command	Signals the identification system that a command for the respective read/write head is present.

Please note the basic procedure on 🕮 12 and 25...31 and the examples on pages 🕮 32...47.

(continued next 🗋)

Function Description
Output buffer, configuration and explanation

Description of Output Buffer (continued)

Sub-address	Meaning	Function Description
01Hex	Command designator	
	00Hex	No command present
	01Hex	Read data carrier
	02Hex	Write to data carrier
	06Hex	Store program in the EEPROM for the Mixed Data Access function
	07Hex	Store the start address for the Auto-Read function in the EEPROM
	12Hex	Initialize the CRC16 data check
	21Hex	Read for Mixed Data Access function (corresponding to the program stored in the EEPROM)
	22Hex	Write for Mixed Data Access function (corresponding to the program stored in the EEPROM)
or:	Data	for writing to the data carrier
or:	Program data	for writing to the EEPROM.

(continued next 🗋)

Please note the basic procedure on 🕮 12 and 25...31 and the examples on pages 🕮 32...47.

Function Description
Output buffer, configuration and explanation

Description of Output Buffer (continued)

Sub-address	Meaning	Function Description
02Hex	Start address (Low Byte)	Address at which reading from or writing to the data carrier begins. (The Low Byte includes the address range from 0 to 255).
or:	Start address (Low Byte)	Address for the Auto-Read function, starting at which the code tag is to be read. The value is stored in the EEPROM. (The Low Byte covers the address range from 0 to 255).
or:	Program No	Number of the program to be stored in the EEPROM in conjunction with command ID 06Hex for Mixed Data Access function (values between 01Hex and 0AHex are allowed!).
or:	Program No.	Number of the program stored in the EEPROM for read or write operations in conjunction with command ID 21Hex or 22Hex for the Mixed Data Access function.
or:	Data	for writing to the data carrier
or:	Program data	for writing to the EEPROM.
03Hex	Start address (High Byte)	Address for reading from or writing to the Data carrier (the High Byte is additionally used for the address range from 256 to 16.383).
or:	Start address (High Byte)	Address for the Auto-Read function, starting at which the code tag is to be read. The value is stored in the EEPROM (the High Byte is also required for the address range from 256 to 16.383).
or:	Data	for writing to the Data carrier
or:	Program data	for writing to the EEPROM.

(continued next 🗋)

Please note the basic procedure on 🕮 12 and 25...31 and the examples on pages 🕮 32...47.

Function Description
Output buffer, configuration and explanation

Description of Output Buffer (continued)

Sub-address	Meaning	Function Description
04Hex	No. of bytes (Low Byte)	Number of bytes to read or write beginning with the start address (the Low Byte includes from 1 to 256 bytes).
or:	Data	for writing to the data carrier
or:	Program data	for writing to the EEPROM.
05Hex	No. of bytes (High Byte)	Number of bytes to read or write beginning with the start address (the High Byte is additionally used for the range between 257 and 16.384 bytes).
or:	Data	for writing to the data carrier
or:	Program data	for writing to the EEPROM.
06Hex	Data	for writing to the data carrier
or:	Program data	for writing to the EEPROM.
...	Data	for writing to the data carrier
or:	Program data	for writing to the EEPROM.
Last byte		
	2nd Bit header	The data are valid if the 1st and 2nd bit header are identical.
or:	Data	for writing to the data carrier
or:	Program data	for writing to the EEPROM.

Please note the basic procedure on 🕮 12 and 25...31 and the examples on pages 🕮 32...47.

Function Description
Input buffer, configuration and explanation

Configuration of the input buffer for one (1) read/write head

The last byte can be arranged as a 2nd bit header through parametering (default).

Subaddress \ Bit No.	7	6	5	4	3	2	1	0	
00_{Hex} = Bit Header	BB	HF	TO	IN	AF	AE	AA	CP	Bit Name
01_{Hex}	Error Code					or		Data	
02_{Hex}	Data								
03_{Hex}	Data								
04_{Hex}	Data								
05_{Hex}	Data								
06_{Hex}	Data								
...	Data								
Last byte	2nd Bit Header (as above)					or		Data	

Description of Input Buffer

Sub-address	Bit Name	Meaning	Function Description
00_{Hex}	BB	Ready	The BIS Identification System is in the Ready state.
Bit Header	HF	Head Error	Cable break from read/write head or no read/write head connected.
	TO	Toggle-Bit Out	for read: BIS has new/additional data ready. for write: BIS is ready to accept new/additional data.

(continued on next 🗋)

Please note the basic procedure on 🕮 12 and 25...31 and the examples on pages 🕮 32...47.

Function Description
Input buffer, configuration and explanation

Description of Input Buffer (continued)

Sub-address	Bit Name	Meaning	Function Description
00_{Hex}	(continued)		
Bit Header	IN	Input	If the parameter "Input IN" is 1, this bit indicates the state of the Input.
	AF	Command Error	The command was incorrectly processed or aborted.
	AE	Command end	The command was finished without error.
	AA	Command start	The command was recognized and started.
	CP	Codetag Present	Data carrier present within the active zone of the read/write head.

In addition to the CP bit, the output signal **CT present** is available. This allows you to process the presence of a data carrier directly as a hardware signal.

Sub-address	Meaning		Function Description
01_{Hex}	Error code		Error number is entered if command was incorrectly processed or aborted. Only valid with AF bit!
	00_{Hex}		No error.
	01_{Hex}		Reading or writing not possible because no data carrier is present in the active zone of a read/write head.
	02_{Hex}		Read error.
	03_{Hex}		Data carrier was removed from the active zone of the read/write head while it was being read.
	04_{Hex}		Write error.

(continued on next 🗋)

Please note the basic procedure on 🕮 12 and 25...31 and the examples on pages 🕮 32...47.

Function Description
Input buffer, configuration and explanation

Description of Input Buffer (continued)

Sub-address	Meaning		Function Description
01Hex	Error code (continued)		
	05Hex		Data carrier was removed from the active zone of the read/write head while it was being written.
	07Hex		AV bit is set but the command designator is missing or invalid.
		or:	Number of bytes is 00Hex.
	09Hex		Cable break to select read/write head, or head not connected.
	0CHex		The EEPROM cannot be read/programmed.
	0DHex		Communication with the read/write head.
	0EHex		The CRC of the read data does not coincide with the CRC of the data carrier.
	0FHex		Contents of the 1st and 2nd bit header (1st and last bytes) of the output buffers are not identical (2nd bit header must be served).
	20Hex		Addressing of the read/write job is outside the memory range of the data carrier.
	21Hex		Invoking of a function which is not possible for the data carrier which is in front of the read/write head.
or:	Data		Data which was read from the data carrier.
02Hex	Data		Data which was read from the data carrier.
...	Data		Data which was read from the data carrier.
Last byte			
	2nd Bit header		The data are valid if the 1st and 2nd bit headers are in agreement.
or:	Data		Data which was read from the data carrier.

Please note the basic procedure on 12 and 25...31 and the examples on pages 32...47.

Function Description
Parametering the BIS S-60_2 processor

Parameters, Overview

There are 6 user parameter bytes stored on the Profibus master that can be used to activate and deactivate various functions. Setting is done directly by linking a device to the Profibus master. The parameter default settings are stored in the GSD file.

- **CRC_16** data check**:**
  If this function is activated, the correctness of the read or written data is ensured by a CRC_16 data check (see 8).
- **Simultaneous data transmission for both read/write heads:**
  With simultaneous data transmission shorter read/write times can be achieved depending on the amount of data to be read/written and the type of controller.
- **Dynamic operation on read/write head 1 or 2:**
  If dynamic operation is parametered, a read/write job can be sent even though there is no Data carrier in the active zone of the head. As soon as a Data carrier passes by the head, the command is immediately carried out.
- **"Auto-Read" for read/write head 1 or 2:**
  If this function is activated, the processor reads out the first (max. 31) bytes from the Data carrier starting at a defined start address as soon as the tag enters the active zone of the read/write head. The start address must first have been stored in the processor's EEPROM with the command ID 07Hex.
- **2nd bit header at end of in- and output buffer:**
  The 2nd bit header (factory setting) prevents data from being accepted by the bus as long as it is not fully updated.
- **Display state of the digital input in the bit header of the input buffer:**
  If this function is activated, the IN-bit displays the state of the digital input of the processor: IN = 0 → digital input low; IN = 1 → digital input high
- **Reset BIS S-60_2 processor through the digital input:**
  If this function is activated, the processor is reset when the digital input is set to high.

Please note the basic procedure on 12 and 25...31 and the examples on pages 32...47.

Function Description
Parametering, Parametering Bytes

Parametering Bytes
User-Parameter Bytes

> ⚠ For parametering all 6 bytes must always be transferred in Hex. Only the bits mentioned may be changed. No guaranty will be given for the proper functioning of the BIS S-60_2 if any of the other bits are changed.

The default values (factory setting) for the 6 bytes are:

	1st byte	**2nd byte**	3rd byte	**4th byte**	**5th byte**	**6th byte**
Hex	00	80	00	82	00	02
Binary	000**0**0**0**00	100**00**000	00000000	**10**0000**1**0	000**00**000	**00000010**
These are used for configuration:	**bit 3**, **bit 5**	**bit 4**, **bit 5**		**bit 7**, **bit 8**, **bit 2**	**bit 4**, **bit 5**	**bit 1...8**

Having the following functions:

The bits which serve for parametering have the following functions:

Bit state: 0 = no
1 = yes

1st byte, bit 5, Activate CRC-16 data checking

1st byte, bit 3, Activate simultaneous data transmission **for both read/write heads**

2nd byte, bit 5, Dynamic mode on read/write head 1 (for effects on read/write times, see 48/49)

2nd byte, bit 4, Activate Auto-Read function starting at specified address after CT-Present for Head 1 (the number of bytes read depends on the selected buffer size minus bit headers for Head 1)

4th byte, bit 8, Arrange a 2nd bit header at the end of the input and output buffers

> ☞ If this function is selected, then the minimum size of both buffers is 4 words (8 bytes) each.
>
> Please note the basic procedure on 12 and 25...31 and the examples on pages 32...47.

Function Description
Parametering, Parametering Bytes

Parametering Bytes
User-Parameter Bytes
(continued)

4th byte, bit 7, 0 = no, 1 = yes — Display state of the digital input in the bit header of the input buffers:

Input is Low: "IN" in the bit header of the input buffers = 0.
Input is High: "IN" in the bit header of the input buffers = 1.

4th byte, bit 2, 0 = no, 1 = yes — Reset the BIS S-60_2 processor through the digital input:

Input is Low: Do not reset.
Input is High: Reset.

Bit state: 0 = no
1 = yes

5th byte, bit 5 Dynamic mode on read/write head 2 (for effects on read/write times, see 48/49)

5th byte, bit 4 Activate Auto-Read function for Head 2 starting at specified address after CT-Present (the number of bytes read depends on the selected buffer size minus bit headers for Head 2)

6th byte, bit 1...8 No. of bytes in input and output buffer which shall be used for read/write head 1, see example on 14

The specification for the input and output buffer on the Master applies to both read/write heads, i.e. this buffer must be divided for both heads. The specification is done in Hex format and must be in a range between 02_{Hex} and 80_{Hex} (128 dec.).

> ☞ If only one read/write head (Head 1) will be used, you may enter the same value here as for the total buffer size. An entry of less than 2 bytes results in an undefined state.
>
> Please note the basic procedure on 12 and 25...31 and the examples on pages 32...47.

Function Description
Processing data carriers

Reading and writing

To carry out a read or write job, the Data carrier must be located in the active zone of the read/write head.

A read/write job has the following sequence (see examples on 32ff):

1. The host sends to the output buffer:
   – the command designator to subaddress 01_{Hex},
   – the start address for reading or writing to subaddress $02_{HEX}/03_{HEX}$,
   – the number of bytes for reading or writing to subaddress $04_{HEX}/05_{HEX}$,
   – the CT bit in the bit header according to the Data carrier type (block size),
   – and sets the AV bit in the bit header to high.
2. The processor:
   – takes the request (AA in the bit header of the input buffer to high),
   – begins to transport the data;
   read = from data carrier to input buffer,
   write = from output buffer to data carrier.
   (Larger data quantities are sent in blocks
   block size with 2nd bit header = buffer size – 2),
   block size without 2nd bit header = buffer size – 1).
   The toggle bits in the two bit headers are used as a kind of handshaking between the host and the BIS S-60_2 processor.
3. The processor has processed the command correctly (AE bit in the bit header of the input buffer). If an error occurred during execution of the command, an error number will be written to subaddress 01_{Hex} of the input buffer and the AF bit in the bit header of the input buffer will be set.

Function Description
Processing data carriers

Codetag Present

As soon as the data carrier enters the active one of the read/write head, the processor indicates this by setting the CP bit (Codetag Present).

> To accelerate the reading of small amounts of data, the ID system makes the first bytes of the data carrier available in the input buffer of the respective read/write head as soon as the tag is detected (30 bytes with 2nd bit header, 31 bites without 2nd bit header, or less if the buffer size has been set smaller).

The data are only valid after the rising edge of the CP bit in the bit header of the input buffer. They remain valid until the falling edge of the CP bit, or until the controller issues a new job.

Special characteristics

To adjust the read/write functions to the numerous possible applications, a few unique features have been implemented that the user can select and set when parametering or programming the processor. These are as follows:

Auto-Read

If the Auto-Read function is activated, the data are read as soon as a data carrier is recognized. No command from the controller is required. Since there is an in- and output buffer for each read/write head, the start address must be specified for each head using the command designator 07_{Hex}. The start addresses may be different. The number of bytes read is determined by the selected size of the input buffer, which is distributed over both heads when 2 are used.

This distinguishes the Auto-Read function from the standard setting for automatic reading, which always starts at Address 0 and includes a maximum number of 30 bytes with 2nd bit header or 31 bytes without 2nd bit header (or less if the buffer size has been set smaller).

Function Description
Processing data carriers

Reading and writing in dynamic mode

In normal operation a read/write job is rejected by the BIS S-60_2 processor by setting the AF bit and an error number if there is no data carrier in the active zone of the read/write head. If dynamic mode is configured, the processor accepts the read/write job and stores it. When a data carrier is recognized, the stored job is carried out.

Reading and writing with simultaneous data transmission

Reading without simultaneous data transmission: In the case of a read job the processor first reads our all requested data from the data carrier after receiving the start address and the desired number of bytes, and then sets the AE bit. Then the data read from the data carrier are written to the input buffer. In the case of larger data amounts this is done in blocks, controlled by the handshake with the toggle bits as described on 25.

Reading with simultaneous data transmission: In the case of a read job the processor begins by transmitting the data into the input buffer as soon as the first 30 bytes (with 2nd bit header, or 31 bytes without 2nd bit header, or less if the buffer size was set smaller) have been read from the data carrier beginning with the start address, and indicates this by inverting the TO bit. As soon as the controller inverts the TI bit, the processor sends the data, which have in the meantime been read, to the input buffer. This is repeated until the processor has read out all the desired data from the data carrier. Now the processor sets the AE bit and outputs the remaining data on the input buffer.

Writing without simultaneous data transmission: In the case of a write job the processor waits until it has received all the data that need to be written from the controller. Only then are the data written to the data carrier as described on 25.

Writing with simultaneous data transmission: In the case of a write job the processor begins to write the data to the data carrier as soon as it has received the first data to be written from the controller's output buffer. Once all the data have been written to the data carrier, the AE bit is set.

Function Description
Processing data carriers

Mixed Data Access

Small read/write programs can be stored in the BIS S-60_2 processor's EEPROM.

The Mixed Data Access function is useful when the required information is stored on the data carrier at various addresses. This function makes it possible to read out this "mixed", i.e. non-contiguously stored data from the data carrier in a single procedure and using just one command.

Up to 10 programs with up to 25 instructions can be stored. Each program instruction contains a "start address" and a "number of bytes" specification. The amount of data for reading may not exceed 2 kB.

Storing a program:
The command identifier 06_{Hex} is used to send the read/write program to the BIS S-60_2 processor. One program per command can be stored. All 25 program records plus an additional 2 bytes with $FF_{Hex}FF_{Hex}$ as a terminator must always be sent. This means a total of **104 bytes** of information per program must be sent (including the command identifier and program number).

> The individual program records must all be contiguous. They must be sent one after the other and be terminated with $FF_{Hex}FF_{Hex}$ as a terminator. It is recommended that the remaining, unused memory sector be filled with $FF_{Hex}FF_{Hex}$.
>
> If an address range is selected twice, the data will also be output twice.

Function Description
Processing data carriers

Mixed Data Access (cont.)

The following shows the structure of a program:

Program structure	Subaddress	Value	Range
Command designator	01_{Hex}	06_{Hex}	
1. Program record			
Program number	02_{Hex}	01_{Hex}	01_{Hex} to $0A_{Hex}$
1st data record:			
Start address Low Byte	03_{Hex}		
Start address High Byte	04_{Hex}		
Number of bytes Low Byte	05_{Hex}		
Number of bytes High Byte	06_{Hex}		
2nd data record:			
...			
25th data record:			
Start address Low Byte	03_{Hex}		
Start address High Byte	04_{Hex}		
Number of bytes Low Byte	05_{Hex}		
Number of bytes High Byte	06_{Hex}		
Terminator	FF_{Hex} FF_{Hex}		

To store a second program, repeat this process.

The procedure for writing these settings to the EEPROM is described in the 9th example on 42...44.

Replacing the EEPROM is described on 56 for BIS S-6002 and on 66 for BIS S-6022.

Function Description
Processing data carriers

Read from data carrier, with program Mixed Data Access

The command identifier 21_{Hex} can be used to read out the program records stored in the program from the data carrier. The user must document exactly which data are to be read from where and with what number of bytes for the respective program (see example 10 on 45)

Write to data carrier, with program Mixed Data Access

The command identifier 22_{Hex} can be used to write the program records stored in the program to the data carrier. The user must document exactly which data are to be written from where and with what number of bytes for the respective program (see example 11 on 46)

Function Description
Processing data carriers

CRC initialization

To be able to use the CRC check, the data carrier must first be initialized with the command identifier 12_{Hex} (see 🗋 32/33). The CRC initialization is used like a normal write job. The latter is rejected (with an error message) if the processor recognizes that the data carrier does not contain the correct CRC. Data carriers as shipped from the factory (all data are 0) can immediately be programmed with a CRC check.

If CRC-16 data checking is activated, a special error message is output to the interface whenever a CRC error is detected.

If the error message is not caused by a failed write request, it may be assumed that one or more memory cells on the data carrier is defective. That data carrier must then be replaced.

If the CRC error is however due to a failed write request, you must reinitialize the data carrier in order to continue using it.

The checksum is written to the data carrier as a 2-byte wide datum. Two bytes per page are 'lost', i.e., the page size becomes 62 bytes. This means that the actual usable number of bytes is reduced:

Data carrier type		Usable bytes
8192 bytes	=	7936 bytes
16384 bytes	=	15872 bytes

Function Description
Examples for protocol sequence

Example No. 1

For configuring with double bit header and 128-byte buffer size!

Initializing the Data carrier for the CRC_16 data checking

The processing of this command is similar to a write command. Start address and number of bytes have to correspond to the maximum number of data to be used.
In this example the complete memory range of a Data carrier with 8 kbytes shall be used (BIS S-1_ _-32/L with 64 byte block size). Because 2 bytes are used for the CRC only 7936 bytes can be used as data bytes, hence: start address = 0, number of bytes = 7936.

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 12_{Hex}
02_{Hex}	Start address 00_{Hex}
03_{Hex}	Start address 00_{Hex}
04_{Hex}	No. of bytes 00_{Hex}
05_{Hex}	No. of bytes $1F_{Hex}$
$00_{Hex}/7F_{Hex}$	Set AV-Bit, CT-Bit to 1

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/7F_{Hex}$	Set AA-Bit, invert TO-Bit

3.) Process subaddresses of the output buffer:

$01...7E_{Hex}$	Enter first 126 bytes of data
$00_{Hex}/7F_{Hex}$	Invert TI-Bit

4.) Process subaddresses of the output buffer:

$01...7E_{Hex}$	Copy first 126 data bytes

Process subaddress of the input buffer:

$00_{Hex}/7F_{Hex}$	Invert TO-Bit

5.) Process subaddresses of the output buffer:

$01...7E_{Hex}$	Enter the second 126 data bytes
$00_{Hex}/7F_{Hex}$	Invert TI-Bit

6.) Process subaddresses of the output buffer:

$01...7E_{Hex}$	Copy second 126 data bytes

Process subaddress of the input buffer:

$00_{Hex}/7F_{Hex}$	Invert TO-Bit

...To be continued until the complete memory range is written. See next 🗋.

Function Description
Examples for protocol sequence

Example No. 1 (continued)

For configuring with double bit header and 128-byte buffer size!

Host:

127.) Process subaddresses of the output buffer:

01...7E Hex	Enter the remaining data byte
00 Hex/7F Hex	Invert TI-Bit

129.) Process subaddresses of the output buffer:

00 Hex/7F Hex	Reset AV-Bit

BIS S-60_2 Identification System:

128.) Process subaddresses of the output buffer:

01...7E Hex	Copy the remaining data byte
Process subaddress of the input buffer:	
00 Hex/7F Hex	Set AE-Bit

130.) Process subaddresses of the input buffer:

00 Hex/7F Hex	Reset AA-Bit and AE-Bit

Function Description
Examples for protocol sequence

Example No. 2

For configuring with double bit header and 8-byte buffer size!

Read 17 bytes starting at data carrier address 10 (Data carrier type with 64 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01 Hex	Command designator 01 Hex
02 Hex	Start address Low Byte 0A Hex
03 Hex	Start address High Byte 00 Hex
04 Hex	No. of bytes Low Byte 11 Hex
05 Hex	No. of bytes High Byte 00 Hex
00 Hex/07 Hex	CT-Bit to 1 (64 Byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

00 Hex/07 Hex	Set AA-Bit
01...06 Hex	Enter first 6 bytes of data
00 Hex/07 Hex	Set AE-Bit

3.) Process subaddresses of the input buffer:

01...06 Hex	Copy first 6 data bytes
Process subaddress of the output buffer:	
00 Hex/07 Hex	Invert TI-Bit

4.) Process subaddresses of the input buffer:

01...06 Hex	Enter the second 6 data bytes
00 Hex/07 Hex	Invert TO-Bit

5.) Process subaddresses of the input buffer:

01...06 Hex	Copy second 6 data bytes
Process subaddress of the output buffer:	
00 Hex/07 Hex	Invert TI-Bit

6.) Process subaddresses of the input buffer:

01...05 Hex	Enter the remaining 5 data bytes
00 Hex/07 Hex	Invert TO-Bit

7.) Process subaddresses of the input buffer:

01...05 Hex	Copy the remaining 5 data bytes
Process subaddress of the output buffer:	
00 Hex/07 Hex	Reset AV-Bit

8.) Process subaddresses of the input buffer:

00 Hex/07 Hex	Reset AA-Bit and AE-Bit

Function Description
Examples for protocol sequence

Example No. 3
like 2nd example but with simultaneous data transmission

For configuring with double bit header and 8-byte buffer size!

Read 17 bytes starting at data carrier address 10, with simultaneous data transmission (data carrier type with 64 byte block size)**:**

While the read job is being carried out and as soon as the input buffer is filled, the first data are sent. The AE bit is not set until the “Read” operation is completed by the processor.

The reply “Job End” = AE bit is reliably set no later than before the last data are sent. The exact time depends on the requested data amount, the input buffer size and the timing of the controller. This is indicated in the following by the note *Set AE-Bit* (in italics).

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 01_{Hex}
02_{Hex}	Start address Low Byte $0A_{Hex}$
03_{Hex}	Start address High Byte 00_{Hex}
04_{Hex}	No. of bytes Low Byte 11_{Hex}
05_{Hex}	No. of bytes High Byte 00_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 1 (64 Byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit
$01...06_{Hex}$	Enter first 6 bytes of data
$00_{Hex}/07_{Hex}$	Invert TO-Bit
$00_{Hex}/07_{Hex}$	*Set AE-Bit*

3.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Copy first 6 data bytes
Process subaddress of the output buffer:	
$00_{Hex}/07_{Hex}$	Invert TI-Bit

4.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Enter the second 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TO-Bit
$00_{Hex}/07_{Hex}$	*Set AE-Bit*

Continued on next 🗋.

Function Description
Examples for protocol sequence

Example No. 3 (continued)

like 2nd example but with simultaneous data transmission

For configuring with double bit header and 8-byte buffer size!

Host:

5.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Copy second 6 data bytes
Process subaddress of the output buffer:	
$00_{Hex}/07_{Hex}$	Invert TI-Bit

7.) Process subaddresses of the input buffer:

$01...05_{Hex}$	Copy the remaining 5 data bytes
Process subaddress of the output buffer:	
$00_{Hex}/07_{Hex}$	Reset AV-Bit

BIS S-60_2 Identification System:

6.) Process subaddresses of the input buffer:

$01...05_{Hex}$	Enter the remaining 5 data bytes
$00_{Hex}/07_{Hex}$	Invert TO-Bit
$00_{Hex}/07_{Hex}$	*Set AE-Bit*

8.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AE-Bit

Function Description
Examples for protocol sequence

Example No. 4

For configuring with double bit header and 8-byte buffer size!

Read 30 bytes starting at data carrier address 10 with read error (data carrier type with 64 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 01_{Hex}
02_{Hex}	Start address Low Byte $0A_{Hex}$
03_{Hex}	Start address High Byte 00_{Hex}
04_{Hex}	No. of bytes Low Byte $1E_{Hex}$
05_{Hex}	No. of bytes High Byte 00_{Hex}
$00_{Hex}/07_{Hex}$	Set CT-Bit to 1 (64 Byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

If an error occurs right away:

$00_{Hex}/07_{Hex}$	Set AA-Bit
01_{Hex}	Enter error number
$00_{Hex}/07_{Hex}$	Set AF-Bit

3.) Process subaddress of the input buffer:

01_{Hex}	Copy error number

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

4.) Process subaddress of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AF-Bit

Function Description
Examples for protocol sequence

Example No. 5, like 4th example but with simultaneous data transmission

For configuring with double bit header and 8-byte buffer size!

Read 30 bytes starting at data carrier address 10, with read error and simultaneous data transmission (data carrier type with 64 byte block size)**:**

If an error occurs, the AF bit is set instead of the AE-Bit, with a corresponding error number. When the AF-BIT is set the job is interrupted and declared to be ended.

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 01_{Hex}
02_{Hex}	Start address Low Byte $0A_{Hex}$
03_{Hex}	Start address High Byte 00_{Hex}
04_{Hex}	No. of bytes Low Byte $1E_{Hex}$
05_{Hex}	No. of bytes High Byte 00_{Hex}
$00_{Hex}/07_{Hex}$	Set CT-Bit to 1 (64 Byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

If an error occurs right away:

$00_{Hex}/07_{Hex}$	Set AA-Bit
01_{Hex}	Enter error number
$00_{Hex}/07_{Hex}$	Set AF-Bit

3.) Process subaddress of the input buffer:

01_{Hex}	Copy error number

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

4.) Process subaddress of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AF-Bit

☞ An error can also occur after the data have already been sent (see 6th example on the next page).

Function Description
Examples for protocol sequence

Example No. 6, with simultaneous data transmission

For configuring with double bit header and 8-byte buffer size!

Read 30 bytes starting at data carrier address 10, with read error and simultaneous data transmission (data carrier type with 64 byte block size)**:**

If an error occurs after data have started to be sent, the AF-Bit is set instead of the AE-Bit along with the corresponding error number. The error message AF is dominant. It cannot be specified which data are incorrect. When the AF-BIT is set the job is interrupted and declared to be ended.

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 01_{Hex}
02_{Hex}	Start address Low Byte $0A_{Hex}$
03_{Hex}	Start address High Byte 00_{Hex}
04_{Hex}	No. of bytes Low Byte $1E_{Hex}$
05_{Hex}	No. of bytes High Byte 00_{Hex}
$00_{Hex}/07_{Hex}$	Set CT-Bit to 1 (64 Byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit
$01...06_{Hex}$	Enter the first 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TO-Bit

3.) Process subaddress of the input buffer:

$01...06_{Hex}$	Copy first 6 data bytes

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Invert TI-Bit

4.) Process subaddresses of the input buffer:

If an error has occurred:

01_{Hex}	Enter error number
$00_{Hex}/07_{Hex}$	Set AF-Bit

5.) Process subaddresses of the input buffer:

01_{Hex}	Copy error number

Process subaddress of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

6.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AF-Bit

Function Description
Examples for protocol sequence

Example No. 7

For configuring with double bit header and 8-byte buffer size!

Write 16 bytes starting at data carrier address 20 (data carrier type with 64 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 02_{Hex}
$02_{Hex}/03_{Hex}$	Start address 14_{Hex} / 00_{Hex}
$04_{Hex}/05_{Hex}$	No. of bytes 10_{Hex} / 00_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 1 (64 Byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit, invert TO-Bit

3.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Enter the first 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TI-Bit

4.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Copy the first 6 data bytes

Process subaddress of the input buffer:

$00_{Hex}/07_{Hex}$	Invert TO-Bit

5.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Enter the second 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TI-Bit

6.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Copy the second 6 data bytes

Process subaddress of the input buffer:

$00_{Hex}/07_{Hex}$	Invert TO-Bit

7.) Process subaddresses of the output buffer:

$01...04_{Hex}$	Enter the remaining 4 data bytes
$00_{Hex}/07_{Hex}$	Invert TI-Bit

8.) Process subaddresses of the output buffer:

$01...04_{Hex}$	Copy the remaining 4 data bytes

Process subaddress of the input buffer:

$00_{Hex}/07_{Hex}$	Set AE-Bit

9.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

10.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AE-Bit

Function Description
Examples for protocol sequence

Example No. 8
Address assignment for the Auto-Read function

For configuring with double bit header and 8-byte buffer size!

Programming start address 75 (data carrier type with 64 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 07_{Hex}
02_{Hex}	Start address Low Byte $4B_{Hex}$
03_{Hex}	Start address High Byte 00_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 1 (64 Byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Set AA-Bit and AE-Bit

3.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Reset AV-Bit

4.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset AA-Bit and AE-Bit

To ensure correct data output, use command identifier 07_{Hex} for each distributed buffer Head 1 and/or Head 2.

If the Auto-Read function is not activated, the processor runs in standard mode and sends starting with data carrier address 0 until the buffer is filled, but a maximum of 30 bytes for double bit header or 31 bytes for a single bit header.

Function Description
Examples for protocol sequence

Example No. 9
Store Mixed Data Access program

For configuring with double bit header and 8-byte buffer size!

Storing a program for reading out 3 data records:

1st data record	Start address	5	Number of bytes	7
2nd data record	Start address	75	Number of bytes	3
3rd data record	Start address	312	Number of bytes	17

Total number of bytes exchanged in the operation: 27 bytes

All 104 bytes are written for the programming.

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 06_{Hex}
02_{Hex}	Program number 01_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 1 (64 bytes block size), set AV-Bit

Host:

2.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Set AA-Bit, invert TO-Bit

3.) Process subaddresses of the output buffer:

01_{Hex}	1st start address	(Low Byte) 05_{Hex}
02_{Hex}		(High Byte) 00_{Hex}
03_{Hex}	1st number of bytes	(Low Byte) 07_{Hex}
04_{Hex}		(High Byte) 00_{Hex}
05_{Hex}	2nd start address	(Low Byte) $4B_{Hex}$
06_{Hex}		(High Byte) 00_{Hex}
$00_{Hex}/07_{Hex}$	Invert TI-Bit	

4.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Invert TO-Bit

Continued on next page.

Function Description
Examples for protocol sequence

Example No. 9
Store Mixed Data Access program
(continued)

For configuring with double bit header and 8-byte buffer size!

Host:

5.) Process subaddresses of the output buffer:

01$_{Hex}$ 02$_{Hex}$	2nd number of bytes	(Low Byte) 03$_{Hex}$ (High Byte) 00$_{Hex}$
03$_{Hex}$ 04$_{Hex}$	3rd start address	(Low Byte) 38$_{Hex}$ (High Byte) 01$_{Hex}$
05$_{Hex}$ 06$_{Hex}$	3rd number of bytes	(Low Byte) 11$_{Hex}$ (High Byte) 00$_{Hex}$
00$_{Hex}$/07$_{Hex}$	Invert TI-Bit	

BIS S-60_2 Identification System:

6.) Process subaddresses of the input buffer:

00$_{Hex}$/07$_{Hex}$	Invert TO-Bit

7.) Process subaddresses of the output buffer:

01$_{Hex}$/02$_{Hex}$	Terminator	FF$_{Hex}$/FF$_{Hex}$
03$_{Hex}$/04$_{Hex}$	(not used)	FF$_{Hex}$/FF$_{Hex}$
05$_{Hex}$/06$_{Hex}$	(not used)	FF$_{Hex}$/FF$_{Hex}$
00$_{Hex}$/07$_{Hex}$	Invert TI-Bit	

8.) Process subaddresses of the input buffer:

00$_{Hex}$/07$_{Hex}$	Invert TO-Bit

Fill all unused start addresses and number of bytes with FF$_{Hex}$!

Continued on next 🗋.

Function Description
Examples for protocol sequence

Example No. 9
Store Mixed Data Access program
(continued)

For configuring with double bit header and 8-byte buffer size!

Host:

35.) Process subaddresses of the output buffer:

01$_{Hex}$/02$_{Hex}$	Terminator	FF$_{Hex}$/FF$_{Hex}$
03$_{Hex}$/04$_{Hex}$	(not used)	FF$_{Hex}$/FF$_{Hex}$
05$_{Hex}$/06$_{Hex}$	(not used)	FF$_{Hex}$/FF$_{Hex}$
00$_{Hex}$/07$_{Hex}$	Invert TI-Bit	

BIS S-60_2 Identification System:

36.) Process subaddresses of the input buffer:

00$_{Hex}$/07$_{Hex}$	Set AE-Bit

37.) Process subaddresses of the output buffer:

00$_{Hex}$/07$_{Hex}$	Reset AV-Bit

38.) Process subaddresses of the input buffer:

00$_{Hex}$/07$_{Hex}$	Reset AA-Bit and AE-Bit

☞ We recommend that you carefully document which parameters are used for start addresses and number of bytes for writing/reading the desired data records.

The data are sequenced in the exact order specified in the program.

Function Description
Examples for protocol sequence

Example No. 10
Use Mixed Data Access program

For configuring with double bit header and 8-byte buffer size!

Read data carrier using Program No. 1 (data carrier type with 64 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 21_{Hex}
02_{Hex}	Program number 01_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 1 (64 byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit
$01...06_{Hex}$	Enter first 6 bytes of data
$00_{Hex}/07_{Hex}$	Set AE-Bit

3.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Copy first 6 data bytes
Process subaddress of the output buffer:	
$00_{Hex}/07_{Hex}$	Invert TI-Bit

4.) Process subaddresses of the output buffer:

$01...06_{Hex}$	Enter the second 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TO-Bit

... A total of 27 bytes of data are exchanged.
For the remainder of the procedure, see Example 2 on 🗋 34.

Dynamic mode is turned off while the Mixed Data Access program is being run.

Function Description
Examples for protocol sequence

Example No. 11
Use Mixed Data Access program

For configuring with double bit header and 8-byte buffer size!

Write data carrier using Program No. 1 (data carrier type with 64 byte block size)**:**

Host:

1.) Process subaddresses of the output buffer in the order shown:

01_{Hex}	Command designator 21_{Hex}
02_{Hex}	Program number 01_{Hex}
$00_{Hex}/07_{Hex}$	CT-Bit to 1 (64 byte block size), set AV-Bit

BIS S-60_2 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

$00_{Hex}/07_{Hex}$	Set AA-Bit
$01...06_{Hex}$	Enter first 6 bytes of data
$00_{Hex}/07_{Hex}$	Set AE-Bit

3.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Copy first 6 data bytes
Process subaddress of the output buffer:	
$00_{Hex}/07_{Hex}$	Invert TI-Bit

4.) Process subaddresses of the input buffer:

$01...06_{Hex}$	Enter the second 6 data bytes
$00_{Hex}/07_{Hex}$	Invert TO-Bit

... A total of 27 bytes of data are exchanged.
For the remainder of the procedure, see Example 7 on 🗋 40.

Dynamic mode is turned off while the Mixed Data Access program is being run.

Function Description
Examples for protocol sequence

Example No. 12

Put the relevant read/write head into ground state:

Both read/write heads can be independently set to the ground state.

Host:

1.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Set GR-Bit

BIS S-60_2 Identification System:

2.) Go to ground state;
Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Reset BB-Bit

3.) Process subaddresses of the output buffer:

$00_{Hex}/07_{Hex}$	Reset GR-Bit

4.) Process subaddresses of the input buffer:

$00_{Hex}/07_{Hex}$	Set BB-Bit

Read/Write Times

Read times from Data carrier to processor in static mode (parametering: 2nd byte, bit 5 = 0, without CRC-16 data check)

For double read and compare:

Data carrier with 64 byte blocks	
No. of bytes	Read time [ms]
from 0 to 63	29
for each additional 64 bytes add	31
from 0 to 2047	= 990

Write times from processor to Data carrier in static mode (parametering: 2nd byte, bit 5 = 0, without CRC-16 data check)

Including readback and compare:

Data carrier with 64 byte blocks	
No. of bytes	Write time [ms]
from 0 to 63	31 + n * 1.5
for 64 bytes or more	y * 31 + n * 1.5

n = number of contiguous bytes to write
y = number of blocks to be processed

Example: 100 bytes from address 130 have to be written. Data carrier with 64 bytes per block. The blocks 3 and 4 will be processed since the start address 130 is in block 3 and the end address 229 in block 4.
t = 2 * 31 + 100 * 1.5 = 212 ms

☞ The indicated times apply after the Data carrier has been recognized. If the Data carrier is not yet recognized, an additional 45 ms for building the required energy field until the Data carrier is recognized must be added.

Read/Write Times

Read times from Data carrier to processor in dynamic mode (parametering: 2nd byte, bit 5 = 1, without CRC-16 data check)

Read times within the 1st block for dual read and compare:

The indicated times apply after the Data carrier has been recognized. If the Data carrier is not yet recognized, an additional 45 ms for building the required energy field until the Data carrier is recognized must be added.

Data carrier with 64 byte blocks	
No. of bytes	Read time [ms]
from 0 to 3	2
for each additional byte add	0.5
from 0 to 63	29

m = highest address to be read

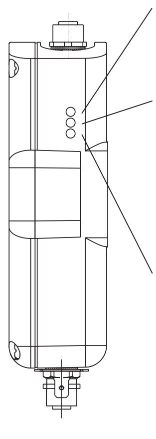
Formula: $t = (m + 1) * 0.5$ ms

Example: Read 11 bytes starting at address 9, i.e. the highest address to be read is 19. This corresponds to 10 ms.

LED Display

Function displays on BIS S-60_2

The BIS S-60_2 uses the three side-mounted LED's to indicate important conditions of the identification system.



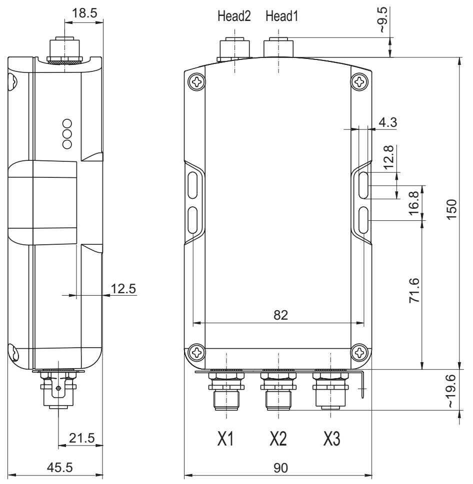
LED	Status	Meaning
Ready / Bus active	red	Supply voltage OK; no hardware error, however, bus not active.
	green	Supply voltage / hardware OK, bus active.
CT1 present / operating	green	Data carrier read/write-ready at read/write head 1.
	yellow	Read/write command at read/write head 1 in process.
	yellow flashes [f ≈ 2 Hz]	Cable break to read/write head or not connected.
	yellow flashes faster [f ≈ 4 Hz]	Communication with R/W Head 1 is faulty or R/W Head 1 is defective.
	off	No data carrier in read/write range of read/write head 1.
CT2 present / operating	green	Data carrier read/write-ready at read/write head 2.
	yellow	Read/write command at read/write head 2 in process.
	yellow flashes [f ≈ 2 Hz]	Cable break to read/write head or not connected.
	yellow flashes faster [f ≈ 4 Hz]	Communication with R/W Head 2 is faulty or R/W Head 2 is defective.
	off	No data carrier in read/write range of read/write head 2.

If all three LED's are synchronously flashing, it means a hardware error. Return the unit to the factory.

BIS S-6002
Mounting the Processor

Mounting the BIS S-6002 processor

The processor is attached using 4 M4 screws.



BIS S-6002 dimensions

BIS S-6002
Opening the Processor / Interface information

Opening the BIS S-6002 processor

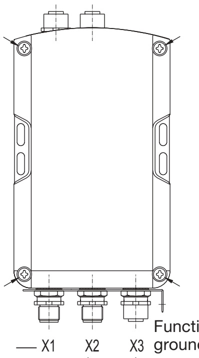
To set the PROFIBUS-DP address, activate or deactivate the internal termination resistor, or to change the EEPROM, you must open up the BIS S-6002 processor.

Remove the 4 screws on the BIS S-6002 and lift off the cover. See the following 🕮 for additional information.

BIS S-6002 interfaces

Connection for read/write head 2

Connection for read/write head 1



Be sure before opening that the unit is disconnected from power.

Mounting of the cover (4 screws), max. permissible tightening torque: 0.15 Nm

Supply voltage digital input

Function ground FE

PROFIBUS-DP Input

PROFIBUS-DP Output

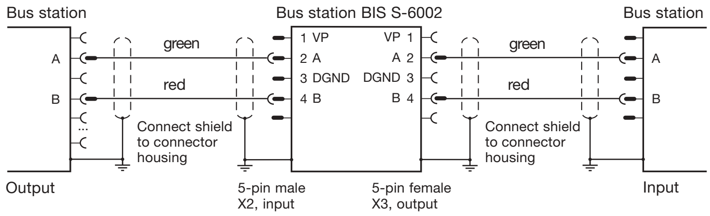
Connection locations and names

BIS S-6002
Interface Information / Wiring Diagrams

PROFIBUS-DP

Ensure that the device is turned off.

To insert BIS S-6002 processor into the serial PROFIBUS-DP, there are the terminal X2 for the PROFIBUS input and the terminal X3 for the PROFIBUS output.



BIS S-6002
Interface Information / Wiring Diagrams

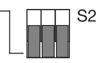
PROFIBUS-DP
Terminating resistor

In case the processor is the last bus module in the chain, then only the incoming cable is connected to X2.

The last bus module must terminate the bus with a resistor. In the case of the BIS S-6002, this can be realized in two different ways:

1. **In the device** by closing the switch S2 (factory standard is open)

   Note: Output terminal must be closed off with a screw cover in order to maintain the enclosure rating.



S2	Terminating resistor
closed	active
open	passive

2. **Outside the device** in a connector to socket X3. In this case the signal VP (pin 1) and DGND (pin 3) should be brought out in order to connect the external resistor to the potential.

   Note: In this case S2 has to be open!

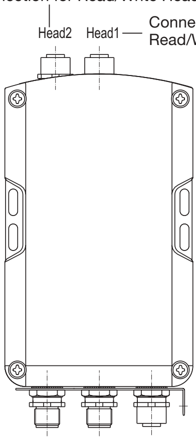
Wiring

To insert BIS S-6002 processor into the serial PROFIBUS and to connect the supply voltage and the digital input, the cables have to be connected to the terminals of the processor. The read/write heads have to be connected to the terminals of Head 1 and Head 2.

BIS S-6002
Interface Information / Wiring Diagrams

Wiring diagram for BIS S-6002 processor

Connection for Read/Write Head 2



Connection for Read/Write Head 1

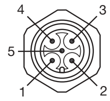
Terminal location and designation

Function ground FE

Supply voltage, digital input

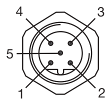
PROFIBUS-DP

X1, supply voltage, digital input

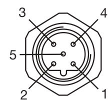


	○
1	+Vs
2	–IN
3	–Vs
4	+IN
5	n.c.

X2, PROFIBUS-input (male)



X3, PROFIBUS-output (female)



	○
1	VP
2	A
3	DGND
4	B
5	n.c.

n.c. = do not connect

The function-ground connector FE should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).

When connecting the bus leads, make sure that the shield has proper connection to connector housing.

BIS S-6002
Changing the EEPROM

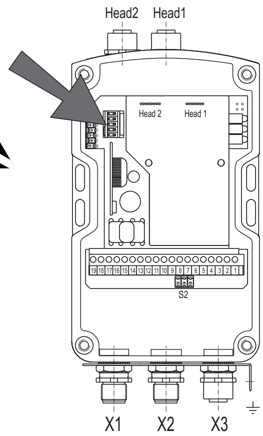
Changing the EEPROM in the BIS S-6002 processor

To replace the EEPROM, open up the processor as described on 52.

Be sure before opening that the unit is disconnected from power..

To avoid damaging the EEPROM, please observe the requirements for handling electrostatically sensitive components.

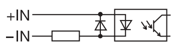
The EEPROM is replaced by unplugging and plugging back into the socket.



Location of the EEPROM

BIS S-6002
Technical Data

Dimensions, Weight	**Housing** Dimensions Weight	Plastic ca. 179 x 90 x 45,5 mm ca. 500 g
Operating Conditions	Ambient temperature	0 °C to + 60 °C
Enclosure Rating	Enclosure rating	IP 65 (when connected)
Connections	Integral connector X1 for **V_S, IN** Integral connector X2 for **PROFIBUS-DP** Input Integral connector X3 for **PROFIBUS-DP** Output	5-pin (male) 5-pin (male) 5-pin (female)
Electrical Connections	**Supply voltage V_S, input** Ripple Current draw	DC 24 V ± 10 % ≤ 10 % ≤ 600 mA
	PROFIBUS-DP slave	Terminal block, electrically isolated
	Digital Input (+IN, –IN) Control voltage active Control voltage inactive Input current at 24 V Delay time, typ.	Optocoupler isolated 4 V to 40 V 1.5 V to –40 V 11 mA 5 ms
	Read/Write Head	2 x connectors 8-pin (female) for all read/write heads BIS S-3_ _ with 8-pin connector (male)



BIS S-6002
Technical Data

Function Displays	BIS operating messages:	
	Ready / Bus active	LED red / green
	CT1 present / operating	LED green / yellow
	CT2 present / operating	LED green / yellow

The CE-Mark is your assurance that our products are in conformance with the EC-Guideline

89/336/EEC (EMC-Guideline)

and the EMC Law. Testing in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, has confirmed that Balluff products meet the EMC requirements of the Generic Standard

EN 50081-2 (Emission) and EN 50082-2 (Noise Immunity).

BIS S-6002
Ordering Information

Ordering Code

BIS S-6002-019-050-03-ST11

- Balluff Identification System
- Type S Read/Write System
- Hardware Type
  6002 = plastic housing, PROFIBUS-DP
- Software-Type
  019 = PROFIBUS-DP
- Read/Write Head, connection
  050 = with two connections for read/write heads BIS S-3_ _
- Interface
  03 = bus versions
- User Connection
  ST11 = Connector version X1, X2, X3 (2× male 5-pin, 1× female 5-pin)

BIS S-6002
Ordering Information

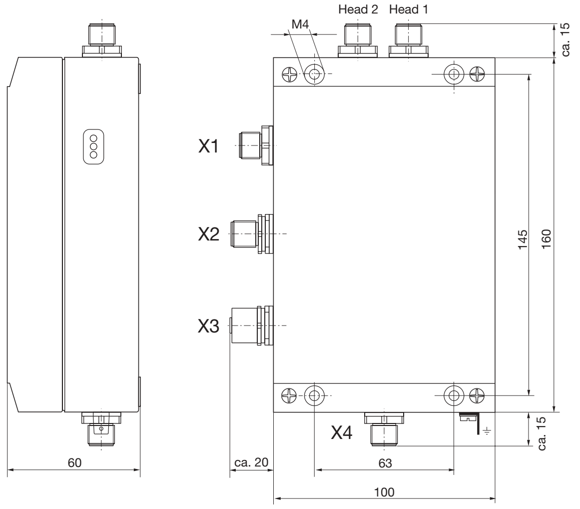
Accessory
(optional,
not included)

Type		**Ordering code**
Connector	for X1	BKS-S 79-00
	for X2	BKS-S103-00
	for X3	BKS-S105-00
Termination	for X3	BKS-S105-R01
Protective cap	for Head_, X3	BKS 12-CS-00
Connector	for Head 1, Head 2 no cable	BKS-S117-00
Connection cable	for Head 1, Head 2; 25 m one end with molded-in connector, one end for user-assembled connector, length as desired, max. 25 m	BIS-S-501-PU1-25
Connection cable	for Head 1, Head 2; 25 m one end with molded-in right-angle connector, one end for user-assembled connector, length as desired, max. 25 m	BIS-S-502-PU1-25

BIS S-6022
Mounting Processor

Mounting the BIS S-6022 processor

The processor is mounted using 4 M4 screws.



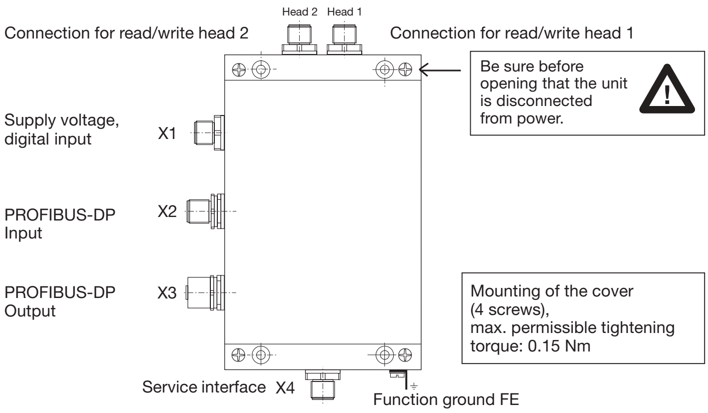
BIS S-6022
Opening the processor / Interface information

Opening the BIS S-6022 processor

To set the PROFIBUS-DP address, activate or deactivate, or to change the EEPROM, you must open up the BIS S-6022 processor.

Remove the 4 screws on the BIS S-6022 and lift off the cover. See the following 🕮 for additional information.

BIS S-6022 interfaces



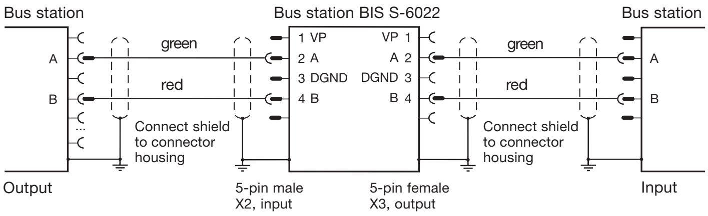
Connection locations and names

BIS S-6022
Interface Information / Wiring Diagrams

PROFIBUS-DP

Ensure that the device is turned off.

To insert BIS S-6022 processor into the serial PROFIBUS-DP, there are the terminal X2 for the PROFIBUS input and the terminal X3 for the PROFIBUS output.



BIS S-6022
Interface Information / Wiring Diagrams

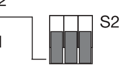
PROFIBUS-DP Terminating resistor

In case the processor is the last bus module in the chain, then only the incoming cable is connected to X2.

The last bus module must terminate the bus with a resistor. In the case of the BIS S-6002, this can be realized in two different ways:

1. **In the device** by closing the switch S2 (factory standard is open)

   Note: Output terminal must be closed off with a screw cover in order to maintain the enclosure rating.



S2	Terminating resistor
closed	active
open	passive

2. **Outside the device** in a connector to socket X3. In this case the signal VP (pin 1) and DGND (pin 3) should be brought out in order to connect the external resistor to the potential.

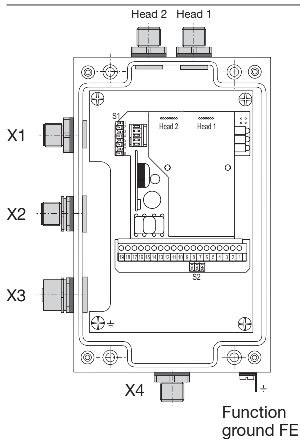
   Note: In this case S2 has to be open!

Wiring

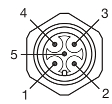
To insert BIS S-6002 processor into the serial PROFIBUS and to connect the supply voltage and the digital input, the cables have to be connected to the terminals of the processor. The read/write heads have to be connected to the terminals of Head 1 and Head 2.

BIS S-6022
Interface Information / Wiring Diagrams

Wiring diagram for BIS S-6022 processor

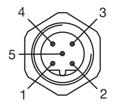


X1, supply voltage, digital input

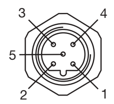


	○
1	+Vs
2	–IN
3	–Vs
4	+IN
5	n.c.

X2, PROFIBUS input (male)

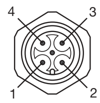


X3, PROFIBUS output (female)



	○
1	VP
2	A
3	DGND
4	B
5	n.c.

X4, Service interface



	○
1	n.c.
2	TxD
3	GND
4	RxD

n.c. = do not connect!

The function-ground connector FE should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).
When connecting the bus leads, make sure that the shield has proper connection to connector housing.

BIS S-6022
Changing the EEPROM

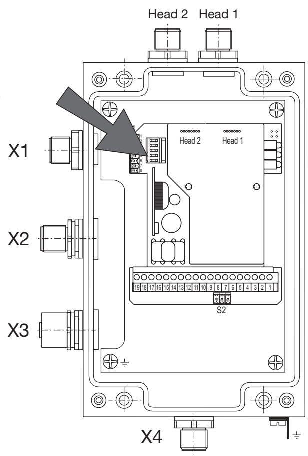
Changing the EEPROM in the BIS S-6022 processor

To change the EEPROM, open the processor as described on 62.

Be sure before opening that the unit is disconnected from power.

To avoid damaging the EEPROM, please observe the requirements for handling electrostatically sensitive components.

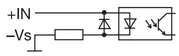
The EEPROM is replaced by unplugging and plugging back into the socket.



Location of the EEPROM

BIS S-6022
Technical Data

Dimensions, weight	**Housing** Dimensions Weight	Metal 190 x 120 x 60 mm 820 g
Operating conditions	Ambient temperature	0 °C to +60 °C
Enclosure	Protection class	IP 65 (when connected)
Connections	Integral connector X1 for **V_S, +IN** Integral connector X2 for **PROFIBUS-DP** input Integral connector X3 for **PROFIBUS-DP** output Integral connector X4 for **Service interface**	5-pin (male) 5-pin (male) 5-pin (female) 4-pin (male)
Electrical connections	**Supply voltage V_S** Ripple Current draw	DC 24 V ± 10 % ≤ 10 % ≤ 600 mA
	Digital input +IN Control voltage active Control voltage inactive Input current at 24 V Delay time, typ.	Optocoupler isolated 4 V to 40 V 1.5 V to –40 V 11 mA 5 ms
	PROFIBUS-DP, Connector X2, X3	serial interface for PROFIBUS stations
	Head 1, Head 2, Read/Write Head	via 2 x connectors 8-pin connector (female) for all read/write heads BIS S-3_ _ with 8-pin connector (male)
	Service interface X4	RS 232



BIS S-6022
Technical Data

Function displays	BIS operating messages: Ready / Bus active CT1 present / operating CT2 present / operating	 LED red / green LED green / yellow LED green / yellow

The CE-Mark is your assurance that our products are in conformance with the EC-Guideline

89/336/EEC (EMC-Guideline)

and the EMC Law. Testing in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, has confirmed that Balluff products meet the EMC requirements of the Generic Standard

EN 50081-2 (Emission) and EN 50082-2 (Noise Immunity).

BIS S-6022
Ordering Information

Ordering code

BIS S-6022-019-050-03-ST14

Balluff Identification System

Type S Read/Write System

Hardware Type
6022 = metal housing, PROFIBUS-DP

Software Type
019 = PROFIBUS-DP

Version
050 = with two connections for read/write heads BIS S-3_ _

Interface
03 = bus versions

User Connection
ST14 = Connector version X1, X2, X3, X4 (male: 2× 5-pin, 1× 4-pin, female: 1× 5-pin)

BIS S-6022
Ordering Information

Accessory (optional, not included)

Type		Ordering code
Mating connector	for X1	BKS-S 79-00
	for X2	BKS-S103-00
	for X3	BKS-S105-00
	for X4	BKS-S 10-3
Termination	for X3	BKS-S105-R01
Protective cap	for Head_, X3	BKS 12-CS-00
Protective cap	for X4	BES 12-SM-2
Connector	for Head 1, Head 2 no cable	BKS-S117-00
Connection cable	for Head 1, Head 2; 25 m one end with molded-in connector, one end for user-assembled connector, length as desired, max. 25 m	BIS-S-501-PU1-25
Connection cable	for Head 1, Head 2; 25 m one end with molded-in right-angle connector, one end for user-assembled connector, length as desired, max. 25 m	BIS-S-502-PU1-25

Appendix, ASCII Table

0	00	Ctrl @	NUL	22	16	Ctrl V	SYN	44	2C	,	65	41	A	86	56	V	107	6B	k
1	01	Ctrl A	SOH	23	17	Ctrl W	ETB	45	2D	-	66	42	B	87	57	W	108	6C	l
2	02	Ctrl B	STX	24	18	Ctrl X	CAN	46	2E	.	67	43	C	88	58	X	109	6D	m
3	03	Ctrl C	ETX	25	19	Ctrl Y	EM	47	2F	/	68	44	D	89	59	Y	110	6E	n
4	04	Ctrl D	EOT	26	1A	Ctrl Z	SUB	48	30	0	69	45	E	90	5A	Z	111	6F	o
5	05	Ctrl E	ENQ	27	1B	Ctrl [	ESC	49	31	1	70	46	F	91	5B	[	112	70	p
6	06	Ctrl F	ACK	28	1C	Ctrl \	FS	50	32	2	71	47	G	92	5C	\	113	71	q
7	07	Ctrl G	BEL	29	1D	Ctrl]	GS	51	33	3	72	48	H	93	5D	]	114	72	r
8	08	Ctrl H	BS	30	1E	Ctrl ^	RS	52	34	4	73	49	I	94	5E	^	115	73	s
9	09	Ctrl I	HT	31	1F	Ctrl _	US	53	35	5	74	4A	J	95	5F	_	116	74	t
10	0A	Ctrl J	LF	32	20		SP	54	36	6	75	4B	K	96	60	`	117	75	u
11	0B	Ctrl K	VT	33	21		!	55	37	7	76	4C	L	97	61	a	118	76	v
12	0C	Ctrl L	FF	34	22		"	56	38	8	77	4D	M	98	62	b	119	77	w
13	0D	Ctrl M	CR	35	23		#	57	39	9	78	4E	N	99	63	c	120	78	x
14	0E	Ctrl N	SO	36	24		$	58	3A	:	79	4F	O	100	64	d	121	79	y
15	0F	Ctrl O	SI	37	25		%	59	3B	;	80	50	P	101	65	e	122	7A	z
16	10	Ctrl P	DLE	38	26		&	60	3C	<	81	51	Q	102	66	f	123	7B	{
17	11	Ctrl Q	DC1	39	27		'	61	3D	=	82	52	R	103	67	g	124	7C	|
18	12	Ctrl R	DC2	40	28		(	62	3E	>	83	53	S	104	68	h	125	7D	}
19	13	Ctrl S	DC3	41	29		)	63	3F	?	84	54	T	105	69	i	126	7E	~
20	14	Ctrl T	DC4	42	2A		*	64	40	@	85	55	U	106	6A	j	127	7F	DEL
21	15	Ctrl U	NAK	43	2B		+												