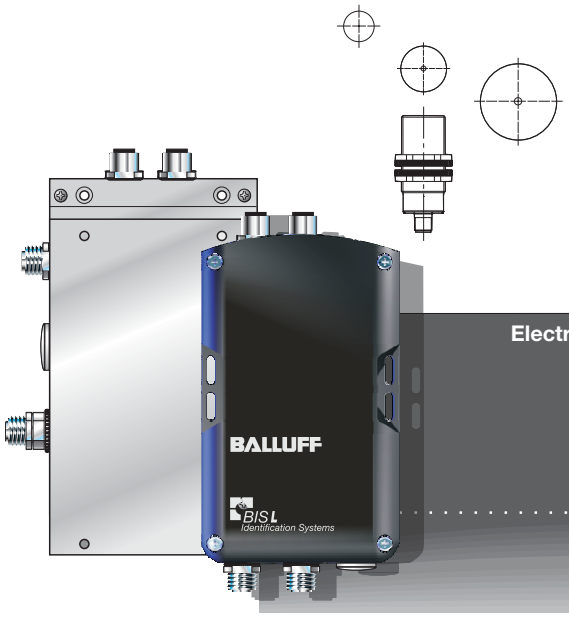
1

BALLUFF



Manual

Electronic Identification Systems BIS
Processor BIS L-60_0
RS232

Deutsch – bitte wenden!

2

No. 829 572 D/E • Edition 0608
Subject to modification.
Replaces edition 0311.

Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Germany
Phone +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de

www.balluff.com

Contents

Safety Considerations

Approved Operation

Series BIS L-60_0 processors along with the other BIS L 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 L-6000-007-050-00-ST15 and BIS L-6020-007-050-00-ST15.

Introduction
BIS L 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 L Identification System, and to assure rapid, trouble-free operation.

Principles

The BIS L 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. The BIS L identification system also allows the use of read-only data carriers.

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 storage systems for monitoring inventory movement;**
- **in transporting and conveying systems.**

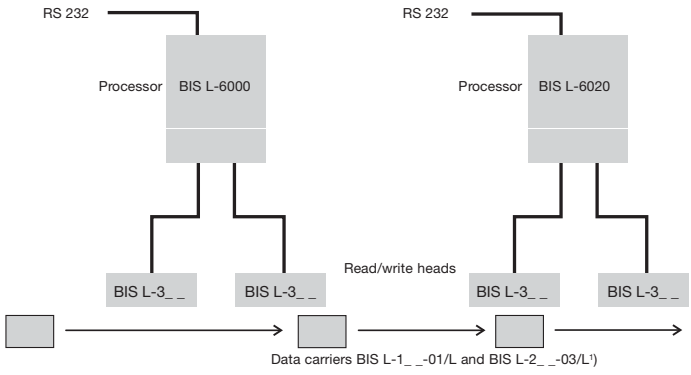
Introduction
BIS L Identification Systems

System Components

The main components of the BIS L Identification Systems are:

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

Configuration with BIS L-6000 and BIS L-6020 processor



Schematic representation of an Identification System (example)

¹) Mixed operation of type BIS L-1_ _-01/L and BIS L-2_ _-03/L together is possible

BIS L-60_0 Processor
Basic knowledge for application

Selecting System Components

The **BIS L-6000** processor has a plastic housing.

The **BIS L-6020** processor has a metal housing.

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

Series BIS L-60_0 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 L-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 L-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 data carrier 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 with CRC_16

For applications requiring high security against bad data, CRC_16 checking can be used. Here a check code is written to the data carrier which allows the data to be checked for integrity at any time or location.

Advantages to CRC_16: Very high data integrity, even during the non-active phase (data carrier outside the active zone of the Read/Write head)

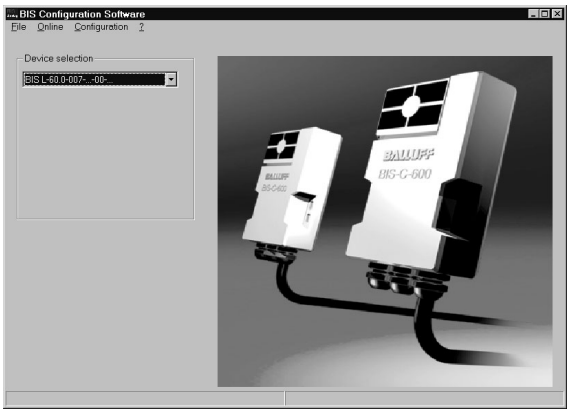
Disadvantages to CRC_16: Longer read/write times, some user data space is taken up on the data carrier.

Use of CRC_16 can be parameterized by the user (see ↰15).

Configuration

Before programming, the processor configuration must be carried out, in case the factory settings will not be used.

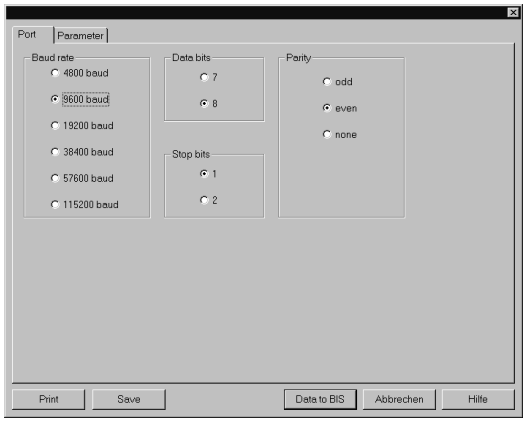
Configuration is done using a computer and the Balluff software *Configuration software BIS*, and it is stored in the processor. It may be overwritten at any time. The configuration can be stored in a file, making it accessible when required.



Configuration

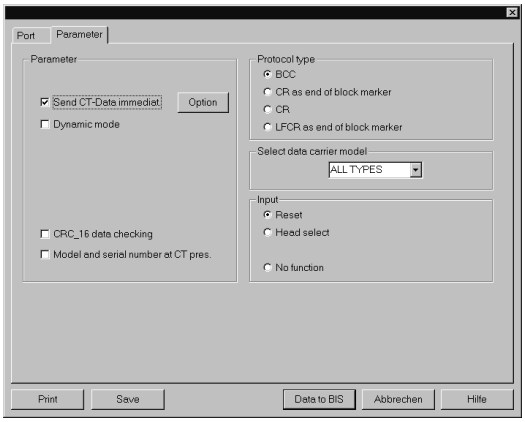
Interface BIS L-60_0

The first screen shows the parameters baud rate, number of data and stop bits, and parity type for the serial interface selected. The graphic shows the factory settings. The other settings are carried out in the corresponding masks which are illustrated in the following ⇈.



Configuration

Parameters BIS L-60_0



Configuration

Protocol Type

Operation with blockcheck BCC is factory set. For host devices which require a terminator, the additional use of Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' is made available. The following page contains examples of the various possibilities.

Examples for terminating telegrams:

Protocol Variants	Telegram with command, Address and no. of bytes	End	Acknow-ledge	Terminator
with blockcheck BCC	'R 0000 0001'	BCC	<ACK> '0'	
with Carriage Return	'R 0000 0001'	'CR'	<ACK> '0'	
with Terminator Carriage Return	'R 0000 0001'	'CR'	<ACK> '0'	'CR'
with Terminator Carriage return and Line feed	'R 0000 0001'	'LF CR'	<ACK> '0'	'LF CR'

Configuration

Parameters

– **Immediately send CT data**
Each time another data carrier is detected, it is read according to the configuration and the data are output. This setting eliminates the read command in dialog mode.
– **Dynamic Mode**
This function switches off the error-message "No data carrier present", i.e.:
-> In dynamic mode, a read or write telegram is stored until a data carrier enters the working range of the corresponding read/write head.
-> Without dynamic mode, a read or write telegram is acknowledged with an error message (<NAK> '1') if there is no data carrier present in front of a read/write head; the processor goes into the ground state.

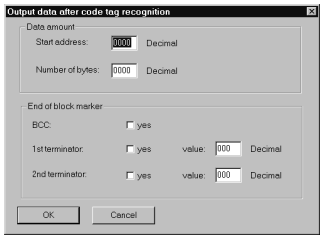
Configuration

Parameters
(continued)

Read and send data carrier data without direct command:
The specified data amount (number of bytes beginning at start address) is read from the newly detected data carrier (refer also to configuring data carrier type on 17 and 18).

After reading, the data are automatically output.

If desired, a BCC and/or 1 or 2 freely definable terminators may be sent also.



Configuration

Parameters
(continued)

– CRC_16 initialization

To be able to use the CRC_16 check, the data carrier must first be initialized with the command identifier Z (see ↱32). The CRC_16 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_16 checksum. Data carriers as shipped from the factory (all data are 0) can immediately be written with CRC-checked data.

If CRC_16 data checking is activated, a special error message is output to the interface whenever a CRC_16 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.

Configuration

CRC_16 and Codetag Present

If CRC_16 was parameterized and a data carrier is recognized whose CRC_16 checksum is incorrect, the read data are not output. The CT present LED comes on and the digital output is set - the data carrier can be processed using the initialization command (**Z**).

CRC_16 and memory capacity

The checksum is written to the data carrier as a 2-byte datum for each CRC block (corresponds to 16 bytes). 2 bytes are used (lost) for each CRC block, i.e., the CRC block contains only 14 bytes of user data. This means that the actual usable number of bytes is reduced:

Data carrier model		Memory capacity	Usable bytes with CRC_16
BIS L-1_ _-01/L	=	192 bytes	168 bytes
BIS L-2_ _-03/L	=	5 bytes	CRC_16 is not supported

Configuration

Data carrier type

Select the data carrier type to process:

- All data carrier types
- BIS L-1_ _-01/L
- BIS L-2_ _-03/L

Data carrier BIS L-1_ _-01/L

Model BIS L-1_ _-01/L data carriers have a memory capacity of 192 bytes of user data. These data can be read or programmed. These data carriers also have a unique, 4-byte serial number, which is read-only.

The data carrier also contains additional memory ranges for configuration and protected data. These areas cannot be processed using the BIS L-60_0 processor.

Model BIS L-1_ _-01/L data carriers are supplied with FF_{Hex} 37_{Hex} configuration. Only data carriers having this configuration are processed.

CT present

At CT present the first user data are read from the data carrier and output on the interface. If the function “Output type and serial number at CT present” is parameterized, then 01_{Hex} followed by the 4-byte unique serial number is output.

Functions

The full command set of the BIS L-60_0 processor can be used with model BIS L-10_01/L data carriers.

Configuration

Data carrier BIS L-2_ _-03/L

Model BIS L-2_ _-03/L data carriers have a unique serial number consisting of 5 bytes. These are read-only and are considered like user data.

CT present

At CT present the 5 bytes of the serial number are read from the data carrier and output on the interface. If the function “Output type and serial number at CT present” is parameterized, then 03_{Hex} followed by the 5-byte unique serial number is output.

Functions

With model BIS L-2_ _-03/L data carriers, all data are read and output as soon as CT present occurs. No other BIS L-60_0 processor commands are usable.

Input

The function of the digital control input of the BIS L-60_0 can be selected.
The factory setting is "Reset".

– **Reset**
  If Reset is selected, a High signal on this input causes a reset of the BIS C-600 processor. Commands already started will be cancelled.
– **Head Select**
  If Head Select is selected, this input is used to select read/write heads.
  Input Low: Head 1 selected.
  Input High: Head 2 selected.
– **Not used**
  The input has no function.

Programming Information

The preceding sections describe basic telegram sequence, and configuration and wiring of the interfaces. What now follows is information about the proper construction of the telegrams themselves.

Specific telegrams exist in the BIS L Identification System for particular tasks. They always begin with the command which is associated with the telegram type.

Telegram types with their associated commands (ASCII characters)

'L'	Read the data carrier with read/write select
'P'	Write to the data carrier with read/write select
'C'	Write a constant value to the data carrier with read/write select
'R'	Read the data carrier
'W'	Write to the data carrier
'H'	Select the read/write head and block size with the variants
'S'	Find the next data carrier (one time)
'Q'	Restart the processor (acknowledge)
'Z'	Initialize CRC-16 data check
'U'	Read data carrier ID and output with status byte.

Please note:
– Continuous querying on the interface is not permitted!
– The minimum wait time between two commands is 300 ms!

Programming Information

Telegram Contents

Start address and no. of bytes	The start address (A3, A2, A1, A0) and the number of bytes to send (L3, L2, L1, L0) are sent in decimal as ASCII characters. For the start address, the range 0000 to 0191 can be used, and for the number of bytes 0001 to 0192. A3 ... L0 represent one ASCII character each. **Please note:** Start address + number of bytes may not exceed the data carrier capacity.
Head number	The commands 'L' (Read with head select) and 'P' (Write with head select) include the number of the read/write head K ('1' or '2').
Acknowledge	The acknowledgement <ACK> '0' is sent by the Identification System if the serially transmitted characters were recognized as correct and there is a data carrier in the active zone of a read/write head. In the 'R' command, the <ACK> '0' is only sent if the data is ready for transmission. <NAK> + Error No.' is sent if an error was recognized or if there is no data carrier in the active zone of a read/write head.
Start	<STX> starts the data transmission.
Transmitted Bytes	The data are transmitted code transparent (no data conversion).

Programming Information

BCC Block Check

The BCC block check is formed as an EXOR of the serially transmitted binary characters of the telegram block. Example: Read 128 bytes starting at address 13.
The command line without BCC is: 'L 0013 0128 20'. The BCC is formed:

	'L	=	0100 1100 EXOR
	0	=	0011 0000 EXOR
	0	=	0011 0000 EXOR
	1	=	0011 0001 EXOR
	3	=	0011 0011 EXOR
	0	=	0011 0000 EXOR
	1	=	0011 0001 EXOR
	2	=	0011 0010 EXOR
	8	=	0011 1000 EXOR
	2	=	0011 0010 EXOR
	0'	=	0011 0000 EXOR
Block check result:	BCC	=	0100 0111 = 'G'

Variants for finish with BCC, Terminator

If necessary the finish using block check BCC can be replaced with a special ASCII character. This is:

– Carriage Return 'CR'

For hosts which always require a terminator character, this must always be included in the telegrams. Available are:

– Carriage Return 'CR' or
– Line Feed with Carriage Return 'LF CR'.

The various protocol variants are represented on the following page.
See also: Configuration starting on ⌐9.

Programming Information

Description of Various Protocol Variants

Reference is now made to the command string 'L 0013 0128 20 G' with 'G' as BCC (see preceding page). This command string is here shown in its possible variants; also shown are the various forms of acknowledgement with and without terminator:

Command line from host system to BIS	Acknowledge from BIS for correct reception	Acknowledge from BIS for incorrect reception
with BCC but no terminator 'L 0013 0128 20 G'	No terminator <ACK> '0'	No terminator <NAK> '1'
with 'CR' instead of BCC, no terminator 'L 0013 0128 20 CR'	No terminator <ACK> '0'	No terminator <NAK> '1'
no BCC, with terminator 'CR' 'L 0013 0128 20 CR'	with terminator 'CR' <ACK> '0 CR'	with terminator 'CR' <NAK> '1 CR'
no BCC, with terminator 'LF CR' 'L 0013 0128 20 LF CR'	with terminator 'LF CR' <ACK> '0 LF CR'	with terminator 'LF CR' <NAK> '1 LF CR'

For <NAK> with error number a '1' was used here (no data carrier present) as an error example.

The respective positions for the additional terminator are shown in the tables in italics.

Programming Information

Read from data carrier with head select
Write to data carrier with head select

Task	Data Flow	Com-mand	Start address of first byte to be sent	Number of bytes to be sent	Head No.		End 2)	Acknow-ledge 3)	*Termi-nator 4)*	Start trans-mission	*Termi-nator 4)*	Data (from start address to start address + no. of bytes)	End 2)	Acknow-ledge 3)	*Termi-nator 4)*
Read	from host system to BIS	'L'	A3 A2 A1 A0 '0 0 0 0' to '0 1 9 1'	L3 L2 L1 L0 '0 0 0 1' to '0 1 9 2'	K '1' or '2'	'0'	BCC or see 2)			<STX>	*'CR' or 'LF CR'*				
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*			D1 D2 D3 ... Dn	BCC or see 2)		
			1)								1)				
Write	from host system to BIS	'P'	A3 A2 A1 A0 '0 0 0 0' to '0 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '0 1 9 2'	K '1' or '2'	'0'	BCC or see 2)			<STX>		D1 D2 D3 ... Dn	BCC or see 2)		
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*
				1)								1)			

1) The command 'Quit' is not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + Error No. if an error occurs.
4) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here.

Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Telegram example for ↰23:
Read from data carrier with read/write head select
with block check (BCC)

-> Head 1 is selected. Read 10 bytes starting at address 50 of the data carrier at read/write Head 2.

The host sends 'L 0050 0010 20J' BCC ($4A_{Hex}$)

Address of first byte to read
Number of bytes to read
Read/write Head No. 2

The BIS processor acknowledges with <ACK> '0'
The host system gives the start command <STX>
The BIS processor provides the data from the data carrier 1 2 3 4 5 6 7 8 9 0 '1' BCC (31_{Hex})
After the telegram sequence, Head 2 remains selected.

Telegram example for ↰23:
Write to data carrier with read/write head select
with block check (BCC)

-> Head 1 is selected. Write 5 bytes starting at address 100 of the data carrier at read/write Head 2.

The host sends 'P 0100 0005 20V' BCC (56_{Hex})

Address of first byte to write
Number of bytes to write
Read/write Head No. 2

The BIS processor acknowledges with <ACK> '0'
The host system gives the start command and data <STX> 1 2 3 4 5 '3' BCC (33_{Hex})
The processor acknowledges with <ACK> '0'
After the telegram sequence, Head 2 remains selected.

Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Writing a constant value in the data carrier with read/write select and block size

This command can be used to erase a data carrier data. One saves the time for the transmission of the write byte.

Task	Data Flow	Com-mand	Start address of first byte to be sent	Number of bytes to be sent	Head No.		End 2)	Acknow-ledge 3)	*Termi-nator 4)*	Start trans-mission	*Termi-nator 4)*	Data (from start address to start address + no. of bytes)	End 2)	Acknow-ledge 3)	*Termi-nator 4)*
Write	from host system to BIS	'C'	A3 A2 A1 A0 '0 0 0 0' to '0 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '0 1 9 2'	K '1' or '2'	'0'	BCC or see 2)			<STX>		D	BCC or see 2)		
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)									1)			

1) The command 'Quit' is not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
4) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Telegram example for ↰25:
Write to data carrier with read/write head select
with block check (BCC)

-> Head 1 is selected. Write 100 bytes of ASCII data value 0 (30_{HEX}) starting at address 20 of the data carrier at read/write Head 2.

The host sends 'C 0020 0100 20B' BCC (42_{HEX})

Address of first byte to write
Number of bytes to write
Read/write Head No. 2

The BIS processor acknowledges with <ACK> '0'
The host system gives the start command and data <STX> '0 2' BCC (32_{HEX})
The processor acknowledges with <ACK> '0'

After the telegram sequence, Head 2 remains selected.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Read from Data carrier, Write to Data carrier

Task	Data Flow	Com-mand	Start address of first byte to send	Number of bytes to send	End 2)	Acknow-ledge 3)	*Termi-nator 4)*	Start trans-mission	*Termi-nator 4)*	Data (from start address to start address + no. of bytes)	End 2)	Acknow-ledge 3)	*Termi-nator 4)*
Read	from host system to BIS	**'R'**	A3 A2 A1 A0 '0 0 0 0' to '0 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '0 1 9 2'	BCC or see 2)			<STX>	*'CR' or 'LF CR'*				
	from BIS to host system					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*			D1 D2 D3 ... Dn	BCC or see 2)		
			1)										
Write	from host system to BIS	**'W'**	A3 A2 A1 A0 '0 0 0 0' to '0 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '0 1 9 2'	BCC or see 2)			<STX>		D1 D2 D3 ... Dn	BCC or see 2)		
	from BIS to host system					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)							1)			

1) The command 'Quit' is not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + Error No. if an error occurs.
4) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here.

Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Telegram example for ↰27:
Read from Data carrier
with block check (BCC)

Read from Data carrier: -> Read 10 bytes starting at address 50.

The host sends	'R 0050 0010 V' BCC (56Hex)
Address of first byte to read	0050
Number of bytes to read	0010
The BIS processor acknowledges with	<ACK> '0'
The host gives the start command	<STX>
The BIS processor provides the data from the data carrier	1 2 3 4 5 6 7 8 9 0 'SOH' BCC (01Hex)

Telegram example for ↰27:
Write to Data carrier
with block check (BCC)

Write to Data carrier: -> Write 5 bytes starting at address 100.

The host system sends	'W 0100 0005 S' BCC (53Hex)
The BIS processor acknowledges with	<ACK> '0'
The host sends the data	<STX> 1 2 3 4 5 '3' BCC (33Hex)
The BIS processor acknowledges with	<ACK> '0'

The 'R' and 'W' commands represent a subtype of the 'L' and 'P' commands.

Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Selecting a Read/Write Head

The 'H1' command selects Read/Write Head 1, 'H2' Read/Write Head 2.

Task	Data Flow	Command	Head number	End 2)	Acknowledge 3)	*Terminator 4)*
Select Read/Write Head	from host system to BIS	**'H'**	'1' or '2'	BCC or see 2)		
	from BIS to host system				<ACK>'0' resp. <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)			

1) The command 'Quit' is not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or Line Feed with Carriage Return may be used.
3) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + Error No. if an error occurs.
4) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here.

Telegram example:
Selecting a Read/ Write Head
with block check (BCC)

-> Switch to Head 1.

The host sends 'H 1 y' BCC (79_{Hex})
The BIS processor acknowledges with <ACK> '0'

Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Find Next Data carrier (one time)

The following telegram is used to find the next data carrier. The next following read/write head is selected and checked to see if a data carrier is in front of this read/write head. If yes, the telegram reply contains the associated number of the read/write head, the data carrier type (01_{Hex} = BIS L-1_ _-01/L; 03_{Hex} = BIS L-2_ _-03/L) and the data carrier ID. If no tag is found, the original read/write head is reselected and checked. If no data carrier is found here, then the telegram reply is: 'HS 000000←'.

Task	Data Flow	Command	Designator	End 2)	Acknowledge	*Terminator 3)*	Reply	Head number	Data carrier type	Data carrier ID	End 2)
Find next data carrier (contin.)	from host system to BIS	**'H'**	'S'	BCC or see 2)							
	from BIS to host system				<ACK>'0'	*'CR' or 'LF CR'*	'H'	'1', '2' or 'S'	01_{Hex} 03_{Hex}	D1 D2 D3 D4 00Hex D1 D2 D3 D4 D5	BCC or see 2)
			1)								

1) The command 'Quit' is not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here.

Telegram example:
Find Next Data carrier (one time)
with block check (BCC)

-> Head 1 is selected. Only read/write head 2 has a data carrier in front of it, whose data carrier ID is 9876.

The host sends 'H S ←' BCC ($1B_{Hex}$)
The BIS processor acknowledges with <ACK> '0 H 2' 01_{Hex} '9 8 7 6' 00_{Hex} '{' BCC ($7B_{Hex}$)

Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Restart the Processor (Quit)

Sending the Restart command causes a telegram in process to be aborted and puts the processor in the ground state. After this telegram is acknowledged, an approx. 500 ms pause should be allowed before starting a new telegram.

Important! The Quit command is not permitted while the processor is waiting for a terminator (BCC, 'CR' or 'LF CR'). In this situation, the Quit would be incorrectly interpreted as a terminator or datum.

Task	Data Flow	Command	End 2)	Acknowledge	End 2)
Restart (Quit)	from host system to BIS	**'Q'**	BCC or see 2)		
	from BIS to host system			'Q'	BCC or see 2)
			1)		

1) The command 'Quit' is not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.

Telegram example:
Restart the Processor (Quit)
with block check (BCC)

Put the BIS system into the ground state.

The host sends 'Q Q' BCC (51_{Hex})

The BIS processor acknowledges with 'Q Q' BCC (51_{Hex})

Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Initialize CRC_16 data check

This telegram initializes a data carrier located at the active read/write head for use of CRC_16 data checking. This telegram must also be send again if a CRC error results from a failed write operation, i.e., the data carrier must be reinitialized in order to use it again.

Please note the table on ↱ 16! The indicated number of usable bytes may not be exceeded, i.e., the sum of start address plus number of bytes must not exceed the data carrier memory capacity!

Task	Data Flow	Command	Start address of first byte to be sent	Number of bytes to be sent	Head No.		End 2)	Acknowledge 3)	*Terminator 4)*	Start transmission	Data (from start address to start address + no. of bytes)	End 2)	Acknowledge 3)	*Terminator 4)*
Initialize CRC_16 range	from host system to BIS	**'Z'**	A3 A2 A1 A0 '0 0 0 0' to '0 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '0 1 9 2'	K '1' or '2'	'0'	BCC or see 2)			<STX>	D1 D2 D3 .... Dn	BCC or see 2)		
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*				<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)								1)			

1) The command 'Quit' is not permitted at this point.
2) Instead of BCC block check, depending on the protocol variant either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used.
3) <ACK> '0" is sent as an acknowledgement if there was no error, or <NAK> + 'Error-No.' if there was an error.
4) For protocol variants which always need a terminator, either 'CR' or 'LF CR' must be inserted here.

The characters between the apostrophes represent the respective ASCII character(s). '_' = Space = ASCII 20_{HEX}.

Programming Information

Query status byte, data carrier type, data carrier ID

With the telegram the status byte (cable break and Tag Present), data carrier type and data carrier ID of data carriers in front of both read/write heads are read and sent. In contrast to the standard command, here the reply is not an <ACK> or <NAK>, but rather a fixed data telegram.

Task	Data Flow	Command	End 2)	Status message	End 2)
Check Status Message	From host system to BIS	**'U'**	BCC or see 2)		
	From BIS to host system			S1 Type1 ID1 S2 Type2 ID2	BCC or see 2)
			1)		

1) The Command 'Quit' is not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.

S1 = Status byte ('1' no data carrier; '9' cable break) of read/write Head 1
Typ1 = Data carrier type (01Hex = BIS L-1_ _-01/L; 03Hex = BIS L-2_ _-03/L) of the data carrier in front of Read/Write Head 1
ID1 = Data carrier ID of the data carrier in front of Read/Write Head 1
S2 = Status byte of Read/Write Head 2
Typ2 = Data carrier type of data carrier in front of Read/Write Head 2
ID2 = Data carrier ID of the data carrier in front of Read/Write Head 2

Telegram example:
Query status byte, data carrier type and data carrier ID

The host sends 'U U' BCC (55Hex)

The BIS processor acknowledges with '0♥123450♥02468 O' BCC (09Hex)

Values inside apostrophes represent the respective character(s) in ASCII code.

Error Numbers

Error Numbers

The BIS L-60_0 always outputs an error number. The meaning of these error numbers is indicated in the following table.

No.	Error Description	Effect	
1	No data carrier present	Telegram aborted, processor goes into ground state.	
2	Read error	Read telegram aborted, processor goes into ground state.	
3	Read aborted, since the data carrier was removed	Processor goes into ground state.	
4	Write error	Write telegram aborted, processor goes into ground state.	**CAUTION:** An aborted write could cause new data to be written to the data carrier which may be incomplete! *)
5	Writing aborted,since the data carrier was removed	Processor goes into ground state.	
6	Interface error	Processor goes into ground state. (parity or stop bit error)	
7	Telegram format error	Processor goes into ground state. Possible format errors: - Command is not 'L', 'P', 'C', 'R', 'W', 'H', 'Q', 'Z' or 'U'. - Start address or number of bytes exceed permissible range	

*) **Note:** If a CRC data check is used, error message "E" could result if error 4 or 5 was not cleared.

35

Error Numbers

Error Numbers
(continued)

No.	Error Description	Effect
8	BCC error, the transmitted BCC is wrong	Telegram is aborted, processor goes into ground state.
9	Cable break from read/write head or cable not connected, LED Codetag Present flashes.	Telegram is aborted, processor goes into ground state.
D	CT error	Bad CT signal, processor goes into ground state.
E	CRC error: the CRC on the data carrier is wrong. *)	Telegram aborted, processor goes into ground state.
F	Address error, address is outside the memory range of the data carrier.	Telegram aborted, processor goes into ground state.
G	Data carrier error, invoking this function is not supported by the data carrier.	Telegram aborted, processor goes into ground state.

*) **Note:** If a CRC data check is used, error message “E” could result if in the preceding command error 4, 5 or B was reported.

36

Read/Write Times

Read times

Data carrier BIS L-1_ _ with 4 bytes per block	
Time for data carrier recognition	~ 370 ms
Read bytes 0 to 3	~ 180 ms
For each additional 4 bytes add another	~ 90 ms

Data carrier BIS L-2_ _

Recognize data carrier + read data carrier
≈ 270 ms

Write times

Data carrier BIS L-1_ _ with 4 bytes per block	
Time for data carrier recognition	~ 370 ms
Write bytes 0 to 3	~ 305 ms
For each additional 4 bytes add another	~ 215 ms

Data carrier BIS L-2_ _

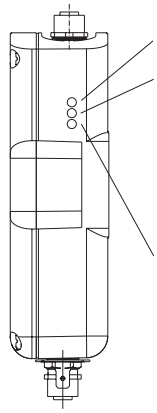
Writing not possible

All data are typical values. Deviations are possible depending on the application and combination of read/write head and data carrier!
The data apply to static operation, no CRC_16 data checking.

37

LED Display

Function displays on BIS L-60_0



The BIS L-60_0 uses the three side-mounted LED's to indicate important conditions of the identification system.

Status	LED	Meaning
Ready	green	Supply voltage / hardware OK.
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 1 or not connected.
	yellow flashes faster [f ≈ 4 Hz]	Communication with read/write head 1 is faulty or read/write 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 2 or not connected.
	yellow flashes faster [f ≈ 4 Hz]	Communication with read/write head 2 is faulty or read/write head 2 is defective.
	off	No data carrier in read/write range of read/write head 2.

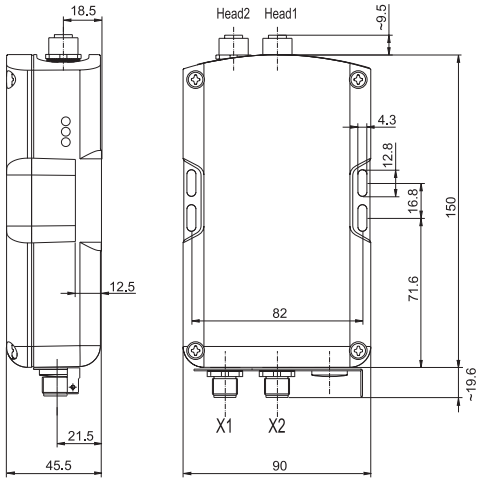
If all three LED's are synchronously flashing, it means a hardware error. Return the unit to the factory.

38

BIS L-6000
Mounting the Processor

Mounting the BIS L-6000 processor

The processor is attached using 4 M4 screws.



BIS L-6000 dimensions

BIS L-6000
Opening the Processor / Interface Information

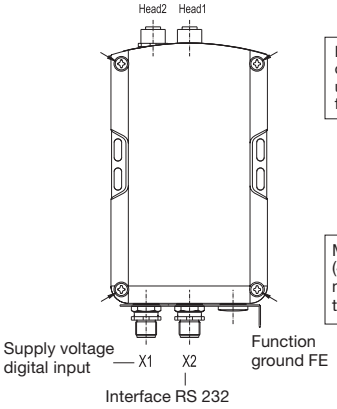
Opening the BIS L-6000 processor

To replace the EEPROM, open up the BIS L-6000 processor.

Remove the 4 screws on the BIS L-6000 and lift off the cover. See the following for additional information.

BIS L-6000 interfaces

Connection for read/write head 2 — Connection for read/write head 1



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

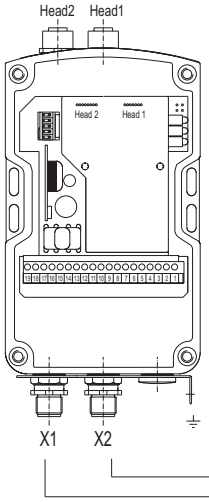
Mounting of the cover (4 screws), max. permissible tightening torque: 0.15 Nm

Terminal location and designation

BIS L-6000
Interface Information / Wiring Diagrams

Wiring diagram for BIS L-6000 processor

Processor open



Wiring the terminal block

19	18	17	16	15	14
+VS	–VS	⏚	TxD	RxD	GND
POWER			RS232		

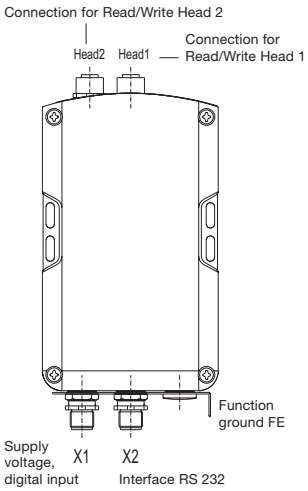
13	12	11	10	9	8
+IN	–IN	NC	NC	NC	NC
INPUT					

7	6	5	4	3	2	1
NC	NC	NC	NC	NC	NC	NC

NC = do not connect

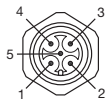
BIS L-6000
Interface Information / Wiring Diagrams

Wiring diagram for BIS L-6000 processor



Terminal location and designation

X1, supply voltage, digital input



Pin	Function
1	+Vs
2	–IN
3	–Vs
4	+IN
5	n.c.

X2, Interface RS 232

Pin	Function
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 L-6000
Changing the EEPROM

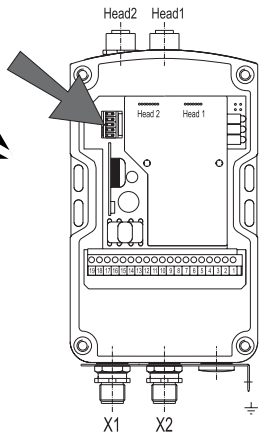
Changing the EEPROM in the BIS L-6000 processor

To replace the EEPROM, open up the processor as described on 39.

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.

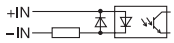
The EEPROM is replaced by unplugging and plugging back into the socket.



Location of the EEPROM

BIS L-6000
Technical Data

Dimensions, Weight	**Housing**	Plastic
	Dimensions	approx. 179 x 90 x 45,5 mm
	Weight	approx. 330 g
Operating Conditions	Ambient temperature	0 °C to + 60 °C
Connections	Terminal block	19-pin
	Cable fittings	3 x PG 9
	Cable diameter	4 to 8 mm
	Conductor size	0.14 to 1 mm²
	with crimp terminals	0.25 to 0.34 mm²
Enclosure Rating	Enclosure rating	IP 65
Connections	Integral connector X1 for **V_S, IN**	5-pin (male)
	Integral connector X2 for **Interface RS 232**	4-pin (male)
Electrical Connections	**Supply voltage V_S, input**	DC 24 V ± 20 %
	Ripple	≤ 10 %
	Current draw	≤ 400 mA
	Interface RS 232	RS 232
	Digital Input (+IN, –IN)	Optocoupler isolated
	Control voltage active	4 V to 40 V
	Control voltage inactive	1.5 V to –40 V
	Input current at 24 V	11 mA
	Delay time, typ.	5 ms



BIS L-6000
Technical Data

Electrical Connections (continued)	**Read/Write Head**	2 x connectors 8-pin (female) for all read/write heads BIS L-3_ _ with 8-pin connector (male)
Function displays	BIS operating messages:	
	Ready	LED 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 61000-6-4 (Emission) and EN 61000-6-2 (Noise Immunity).

BIS L-6000
Ordering Information

Ordering Code

BIS L-6000-007-050-00-ST15

Balluff Identification System

Type L Read/Write System

Hardware-Typ
6000 = plastic housing

Software-Typ
007 = Balluff protocol

Read/Write Head, Connection
050 = with two connections for external read/write heads BIS L-3_ _

Interface
00 = RS 232

User Connection
ST15 = Connector version X1, X2 (male: 1 × 5-pin, 1 × 4-pin)

BIS L-6000
Ordering Information

Accessory
(optional,
not included)

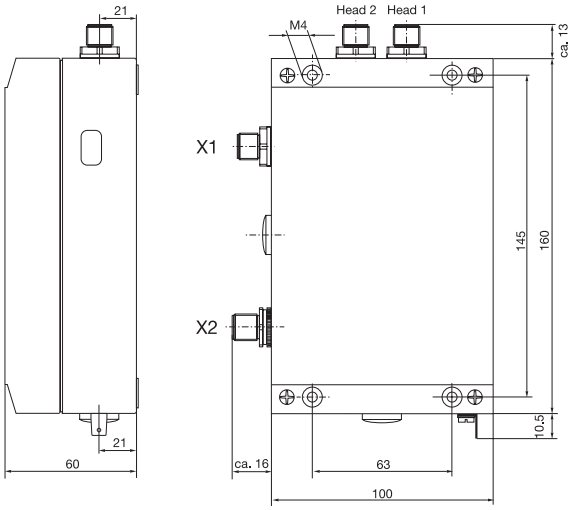
Type		Ordering code
Connector	for X1	BKS-S 79-00
Interface cable	for X2	BIS-C-522-PVC-02
Connector	for Head 1, Head 2 no cable	BKS-S117-00
Connection cable	for Head 1, Head 2; 5 m	BIS-L-500-PU-05
Connection cable	for Head 1, Head 2; 10 m	BIS-L-500-PU-10
Connection cable	for Head 1, Head 2; 25 m one end with molded-in connector, one end for user-assembled connector, length as desired	BIS-L-501-PU-25

47

BIS L-6020
Mounting the Processor

Mounting the BIS L-6020 processor

The processor is mounted using 4 M4 screws.



48

BIS L-6020
Opening the Processor / Interface Information

Opening the BIS L-6020 processor

To replace the EEPROM, open up the BIS L-6020 processor.

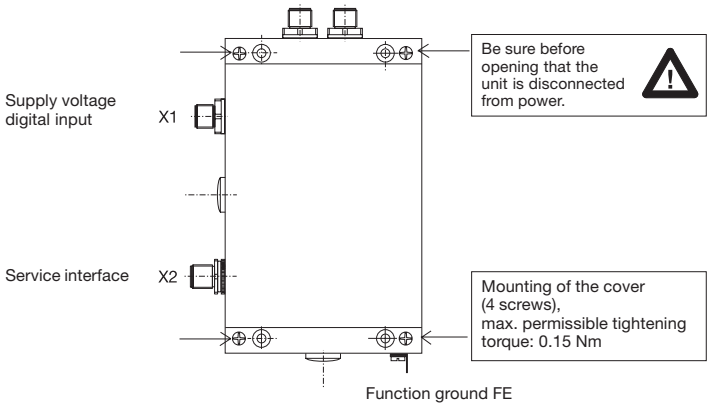
Remove the 4 screws on the BIS L-6020 and lift off the cover. See the following for additional information.

BIS L-6020 interfaces

Connection for read/write head 2

Head 2 Head 1

Connection for read/write head 1

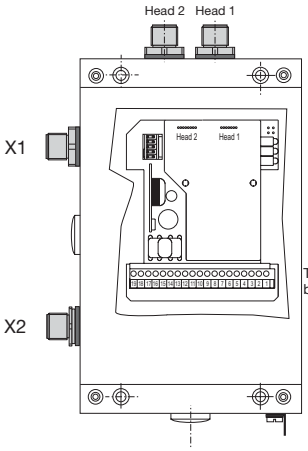


Connection locations and names

BIS L-6020
Interface Information / Wiring Diagrams

Wiring diagram for BIS L-6020 processor

Processor open



X1 Power supply, Input
X2 Interface RS 232

Wiring the terminal block

19	18	17	16	15	14
+VS	–VS	⏚	TxD	RxD	GND
POWER			RS232		

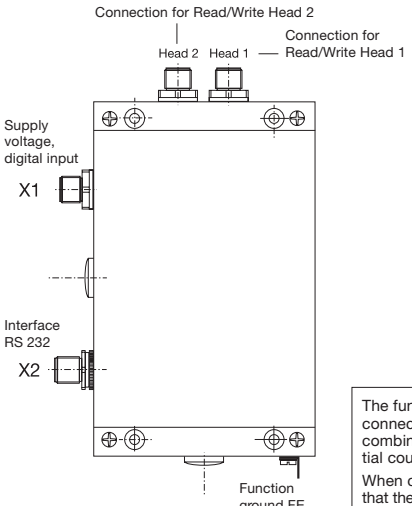
13	12	11	10	9	8
+IN	–IN	NC	NC	NC	NC
INPUT					

7	6	5	4	3	2	1
NC	NC	NC	NC	NC	NC	NC

NC = do not connect

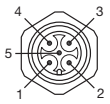
BIS L-6020
Interface Information / Wiring Diagrams

Wiring diagram for BIS L-6020 processor



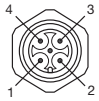
Terminal location and designation

X1, supply voltage, digital input



Pin	Function
1	+Vs
2	–IN
3	–Vs
4	+IN
5	n.c.

X2, Interface RS 232



Pin	Function
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 L-6020
Changing the EEPROM

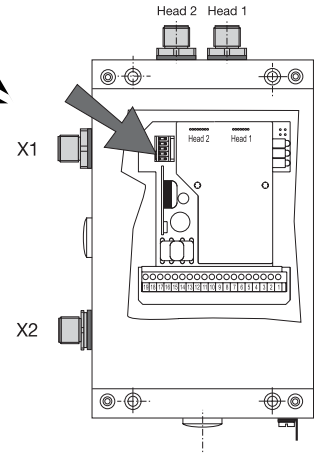
Changing the EEPROM in the BIS L-6020 processor

To replace the EEPROM, open up the processor as described on ⍐ 48.

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.

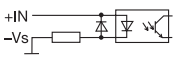
The EEPROM is replaced by unplugging and plugging back into the socket.



Location of the EEPROM

BIS L-6022
Technical Data

Dimensions, Weight	**Housing**	Metal
	Dimensions	approx. 184 x 120 x 60 mm
	Weight	approx. 820 g
Operating Conditions	Ambient temperature	0 °C to +60 °C
Connections	Terminal block	19-pin
	Cable fittings	3 x PG 9
	Cable diameter	4 to 8 mm
	Conductor size	0.14 to 1 mm²
	with crimp terminals	0.25 to 0.34 mm²
Enclosure	Protection class	IP 65
Connections	Integral connector X1 for **V_S, IN**	5-pin (male)
	Integral connector X2 for **Interface RS 232**	4-pin (male)
Electrical Connections	**Supply voltage V_S**	DC 24 V ± 20 %
	Ripple	≤ 10 %
	Current draw	≤ 400 mA
	Interface	RS 232
	Digital Input (+IN, –IN)	Optocoupler isolated
	Control voltage active	4 V to 40 V
	Control voltage inactive	1.5 V to –40 V
	Input current at 24 V	11 mA
	Delay time, typ.	5 ms



BIS L-6020
Technical Data

Electrical Connections (continued)	**Read/Write Head**	via 2 x connectors 8-pin connector (female) for all read/write heads BIS L-3_ _ with 8-pin connector (male)

Function displays	BIS operating messages:	
	Ready	LED green
	CT1 Present / operating	LED green / yellow
	CT2 Present / operating	LED green / yellow

CE *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 61000-6-4 (Emission) and EN 61000-6-2 (Noise Immunity).

BIS L-6020
Ordering Information

Ordering code

BIS L-6020-007-050-00-ST15

Balluff Identification System

Type L Read/Write System

Hardware-Type
6020 = metal housing

Software-Type
007 = Balluff protocol

Read/Write Head, Connection
050 = with two connectors for read/write heads BIS L-3_ _

Interface
00 = RS 232

User Connection
ST15 = Connector version X1, X2 (male: 1 × 5-pin, 1 × 4-pin)

BIS L-6020
Ordering Information

Accessory
(optional,
not included)

Type		Ordering code
Connector	for X1	BKS-S 79-00
Interface cable	for X2	BIS-C-522-PVC-02
Connector	for Head 1, Head 2 no cable	BKS-S117-00
Connection cable	for Head 1, Head 2; 5 m	BIS-L-500-PU-05
Connection cable	for Head 1, Head 2; 10 m	BIS-L-500-PU-10
Connection cable	for Head 1, Head 2; 25 m one end with molded-in connector, one end for user-assembled connector, length as desired	BIS-L-501-PU-25

Appendix, ASCII Table

Decimal	Hex	Control Code	ASCII
0	00	Ctrl @	NUL
1	01	Ctrl A	SOH
2	02	Ctrl B	STX
3	03	Ctrl C	ETX
4	04	Ctrl D	EOT
5	05	Ctrl E	ENQ
6	06	Ctrl F	ACK
7	07	Ctrl G	BEL
8	08	Ctrl H	BS
9	09	Ctrl I	HT
10	0A	Ctrl J	LF
11	0B	Ctrl K	VT
12	0C	Ctrl L	FF
13	0D	Ctrl M	CR
14	0E	Ctrl N	SO
15	0F	Ctrl O	SI
16	10	Ctrl P	DLE
17	11	Ctrl Q	DC1
18	12	Ctrl R	DC2
19	13	Ctrl S	DC3
20	14	Ctrl T	DC4
21	15	Ctrl U	NAK

Decimal	Hex	Control Code	ASCII
22	16	Ctrl V	SYN
23	17	Ctrl W	ETB
24	18	Ctrl X	CAN
25	19	Ctrl Y	EM
26	1A	Ctrl Z	SUB
27	1B	Ctrl [	ESC
28	1C	Ctrl \	FS
29	1D	Ctrl]	GS
30	1E	Ctrl ^	RS
31	1F	Ctrl _	US
32	20		SP
33	21		!
34	22		"
35	23		#
36	24		$
37	25		%
38	26		&
39	27		'
40	28		(
41	29		)
42	2A		*
43	2B		+

Decimal	Hex	ASCII
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@

Decimal	Hex	ASCII
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U

Decimal	Hex	ASCII
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j

Decimal	Hex	ASCII
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	DEL