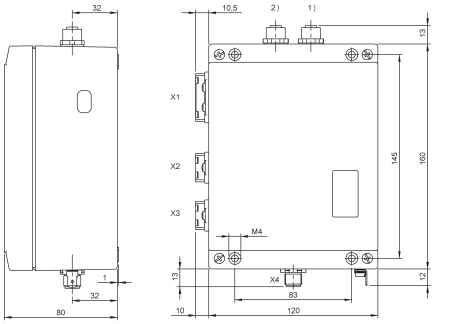
BALLUFF

sensors worldwide

BIS M-60_8 PROFINET IO

Technical Description, User's Guide



english

www.balluff.com

1 Notes to the user

1.1 About this manual

This manual describes processors in the series BIS M-60_8 identification system as well as startup instructions for immediate operation.

1.2 Structure of the manual

The manual is organized so that the sections build on each other
Section 2: Basic safety information
Section 3: Key steps for installing the Identification System
Section 4: Introduction to the material
Section 5: Technical data for the processor
Section 6: Mechanical and electrical connection
Section 7: Logging the processor on to the network
Section 8: User-defined settings for the processor
Section 9: Processor and host system interaction

1.3 Typographical conventions

The following conventions are used in this manual:

Enumerations

Enumerations are shown as a list with en-dash.
– Entry 1,
– entry 2.

Actions

Action instructions are indicated by a preceding triangle. The result of an action is indicated by an arrow.
▶ Action instruction 1.
⇒ Action result.
▶ Action instruction 2.

Syntax

Numbers
– Decimal numbers are shown without additional indicators (e.g. 123),
– Hexadecimal numbers are shown with the additional indicator hex (e.g. 00hex).

Parameters
Parameters are shown in italics (e.g. CRC_*16*).

Directory paths
References to paths in which data are stored or are to be saved to are shown in small caps (e.g. Project:\Data Types\User Defined).

Cross-references

Cross-references indicate where additional information on the topic can be found (see Technical Data“ starting page 14).

1.4 Symbols

Attention!
This symbol indicates a safety instruction that must be followed.

Note, tip
This symbol indicates general notes.

1 Notes to the user

1.5 Abbreviations

BIS	Balluff Identification System
CRC	Cyclic Redundancy Check
DIL	Dual in-line package (also Dual In-Line)
EEPROM	Electrically Erasable and Programmable Read Only Memory
EMC	Electromagnetic Compatibility
GSD	General Station Description
GSDML	General Station Description Markup Language
MAC-ID	Media Access Control Identifier
PC	Personal Computer
PNO	Profibus Nutzerorganisation e.V. (organized user group)
PLC	Programmable Logic Controller

2 Safety

2.1 Intended use

The BIS M-60_8 processor is a component of the BIS M Identification System. Within the Identification System it is used to for connecting to a host computer (PLC, PC). It may be used only for this purpose in an industrial environment corresponding to Class A of the EMC Law.
This description is valid for processors in series BIS M-60_8-…

2.2 General safety notes

Installation and startup
Installation and startup are to be performed only by trained personnel. Any damage resulting from unauthorized manipulation or improper use voids the manufacturer's guarantee and warranty.
When connecting the processor to an external controller, observe proper selection and polarity of the connection as well as the power supply (see "Installation" section on page 18).
The processor may be operated only using an approved power supply (see "Technical Data starting page 14).

Operation and testing
The operator is responsible for ensuring that local safety regulations are observed.
When defects and non-clearable faults in the Identification System occur, take the system out of service and secure it against unauthorized use.

2.3 Meaning of the warning notes

⚠ **Attention!**
The pictogram together with the expression "Attention!“ warns of a possible hazardous situation for the health of persons or of equipment damage. Disregard of these warning notes may result in injury or damage to equipment.
► Always observe the described measures for preventing this danger.

3 Getting Started

3.1 Quick start

Mechanical connection

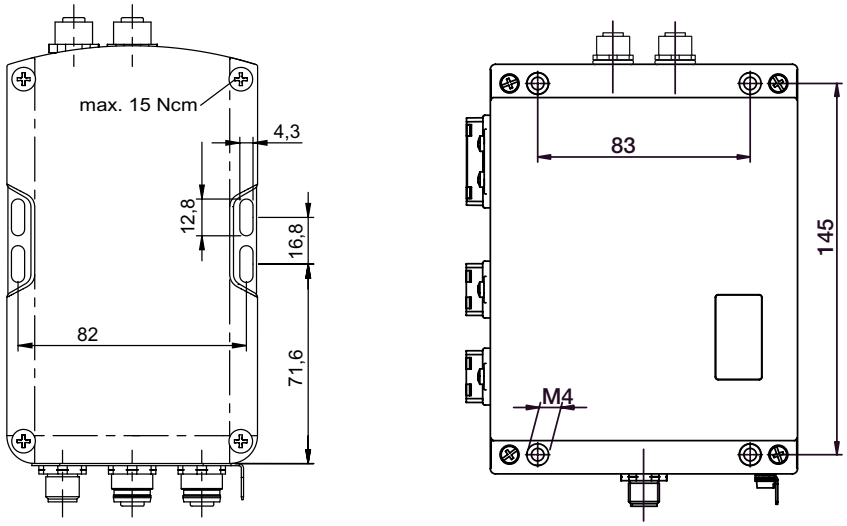


Fig. 1: Mechanical connection (dimensions in mm)

► Attach processor using 4 M4 screws.

Installation with support rail (accessory for BIS M-6008)

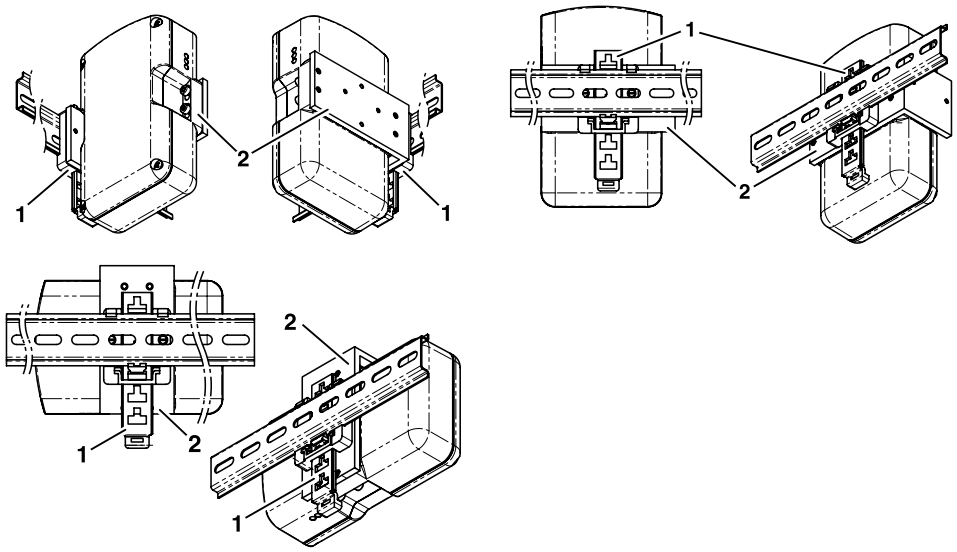


Fig. 2: Mounting using rail holder BIS Z-HW-001 (accessory)

1 Rail holder
2 Mounting brackets

3 Getting Started

Electrical connection

BIS M -6008

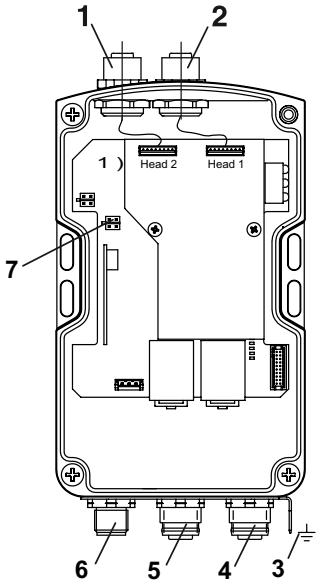
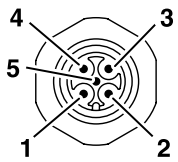


Fig. 3: Electrical connection BIS M-6008

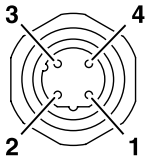
1 Head 2 – Read/write head 2
2 Head 1 – Read/write head 1
3 Function ground FE
4 X3 – PROFINET Port 2
5 X2 – PROFINET Port 1
6 X1 – Supply voltage
7 X7 – Service port

X1 – Supply voltage



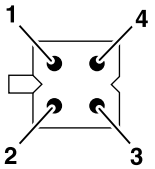
PIN	Function
1	+Vs
2	n.c.
3	-Vs
4	n.c.
5	n.c.

X2, X3 – PROFINET



PIN	Function
1	TD+
2	RD+
3	TD-
4	RD-

X7 – Service port



PIN	Function
1	TxD
2	RxD
3	GND
4	n.c.

Attention!
Make the ground connection either directly or using an RC combination to ground. When making your connection to the Ethernet, be sure that the shield is perfectly connected to the connector body.

3 Getting Started

BIS M -6028

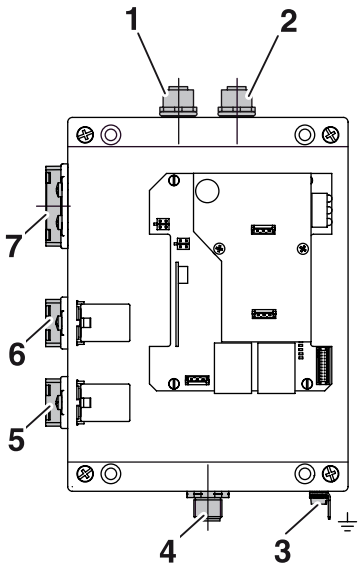
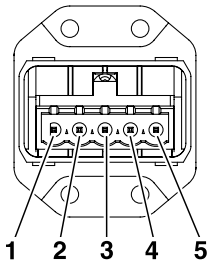


Fig. 4: Electrical connection BIS M-6028

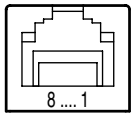
1 Head 2 – Read/write head 2
2 Head 1 – Read/write head 1
3 Function ground FE
4 X4 – Service port
5 X3 – PROFINET Port 2
6 X2 – PROFINET Port 1
7 X1 – Supply voltage

X1 – Supply voltage



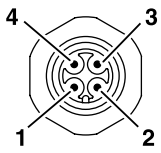
PIN	Function
1	+Vs
2	-Vs
3	n.c.
4	n.c.
5	n.c.

X2, X3 – PROFINET



PIN	Function
1	TD+
2	TD-
3	RD+
4	n.c.
5	n.c.
6	RD-
7	n.c.
8	n.c.

X4 – Service port



PIN	Function
1	n.c.
2	TxD
3	GND
4	RxD

3 Getting Started

Project administration

Project administration is accomplished using the project administration tool "SIMATIC NCM PC Manager“ or using "STEP 7“.
The following steps are required for integrating a BIS M-60_8 processor:

1. Install the GSDML file of the IO device in the hardware configuration
2. Update catalog
3. Use "Insert object" to add the IO device "BIS M-60x8_RT“ or „BIS M-60x8_IRT“
4. Insert both modules for inputs and outputs
   (e.g.: "RT 32 Byte E“ and „RT 32 Byte A“ for processor "BIS M-60x8_RT“ or „IRT 32 Byte E“ and IRT 32 Byte A“ for processor "BIS M-60x8_IRT“)

Additional project administration steps:

5. The name suffix "RT" or "IRT" tells you how the read and write data are exchanged.

Note
Both processors have a 2-port IRT switch and are therefore able to pass IRT data packets.

The object properties of these modules can be used to set the start addresses of the input and output data.

Note
The input and output data can be used to control the BIS M-60_8 as described in Section 9.

Device name

6. The object properties of the inserted object "m-60_8" can be used to assign the device name, the device number and the IP address.

The processor and the host system communicate using PROFINET protocol. This means an IP address and a unique device name are required. The device name and IP address can be saved in the IO device using "Target system > Ethernet > Edit Ethernet device“.

Note
The BIS M-60_8 processor is shipped without a device name. In the included GSDML file the device name "m-60x8" is preset.

4 Basic Knowledge

4.1 Function principle of Identification Systems

The BIS M Identification System is classified as a non-contacting system with read and write function. This makes it possible to not only transport information which is fixed programmed in the data carrier, but also to collect and pass on current information.

The main components of the BIS M Identification System are:
– Processor,
– Read/write heads,
– Data carriers.

The main areas of application are:
– In production for controlling material flow (e.g. in model-specific processes), in workpiece transport with conveying systems, for acquiring safety-relevant data,
– In warehousing for monitoring material movement,
– transporting and conveying..

4.2 Product description

Processor BIS M-6008:
– Plastic housing,
– PROFINET connections using 2 round M12 connectors, D-coded, supply voltage using M12 round connector,
– Two read/write heads can be connected,
– Read/write heads are suitable for dynamic and static operation,
– Power for the system components provided by the processor,
– Power for the data carrier provided by the read/write heads via carrier signal.

Processor BIS M-6028:
– Metal housing,
– PROFINET connections using 2 RJ45 plugs - IP65 (AIDA), supply voltage via push-pull power connector (AIDA),
– Two read/write heads can be connected,
– Read/write heads are suitable for dynamic and static operation,
– Power for the system components provided by the processor,
– Power for the data carrier provided by the read/write heads via carrier signal.

Arrangement of the read/write heads:
Which arrangement of the read/write heads makes the most sense depends essentially on the possible spatial arrangement of the components. There are no functional restrictions. Distance and relative speed depend on the characteristics of the data carriers used.

Note
If two read/write heads are connected to the BIS M-60_8 processor, both can be operated independently of each other:
One data carrier can be read at the first read/write head, while a different data carrier can be written to at the second read/write head.

4 Basic Knowledge

4.3 Control function

The processor is the link between data carrier and controlling system. It manages two-way data transfer between data carrier and read/write head and provides buffer storage.
The processor uses the read/write head to write data from the controlling system to the data carrier or reads the data from the carrier and makes it available to the controlling system.

Host systems may be the following:
– A control computer (e.g. industrial PC),
– a PLC.

Double bit header for asynchronous data transmission:
If a controller does not send the data range for updating the input/output buffer synchronous, data inconsistencies may occur when sending more than 2 bytes. Consistency of the sent data can then only be ensured by sending the control bits in the first byte and again in the last bytes of the in-/output buffer. By comparing the two bit headers it can be determined whether the data are fully updated and can be accepted.
This method affects neither the PLC cycle time nor the bus access time.
Only one byte in the data buffer for the byte of the 2nd bit header is required instead of using it for data.

4 Basic Knowledge

4.4 Data integrity

In order to ensure data integrity, data transfer between the data carrier and processor must be monitored using a check procedure.
The factory default setting in the processor is for double reading with comparison. Alternately the CRC_16 data check can be selected.
In CRC_16 data checking a checksum is written to the data carrier which enables the data to be checked for validity at any time.
Which procedure should be used depends on how the identification system is used.

Note
Mixed operation of both check procedures is not possible!

The following table provides an overview of the advantages of the respective check procedure.

CRC_16 data check	Double reading
Data integrity even during the non-active phase (data carrier outside the read/write head zone)	No user bytes are sacrificed for storing a check code.
Shorter read time – page is read once	Shorter write time – no check code is written.

4.5 Bus connection

Processor and controlling system are connected via PROFINET.
The PROFINET IO (decentralized peripheral) is tailored to communication between a controller and decentralized field devices.
PROFINET is a combination of ProfiBus DP and Ethernet in one system, whereby the IO view of ProfiBus is retained. The device model of PROFINET IO is also oriented towards the ProfiBus technology. The characteristics of the IO devices are however described by GSD files based on XML (GSDML), and project administration/system integration is accomplished analogous to ProfiBus devices.
In a PROFIBUS network IO controllers and IO devices are connected to each other using all common network topologies: Star, line, ring or tree type topologies are possible.
The BIS M-60_8 has a built-in IRT switch with 2 ports for this purpose. This means both RT and IRT can be used.

5 Technical Data

5.1 BIS M-6008

Dimensions

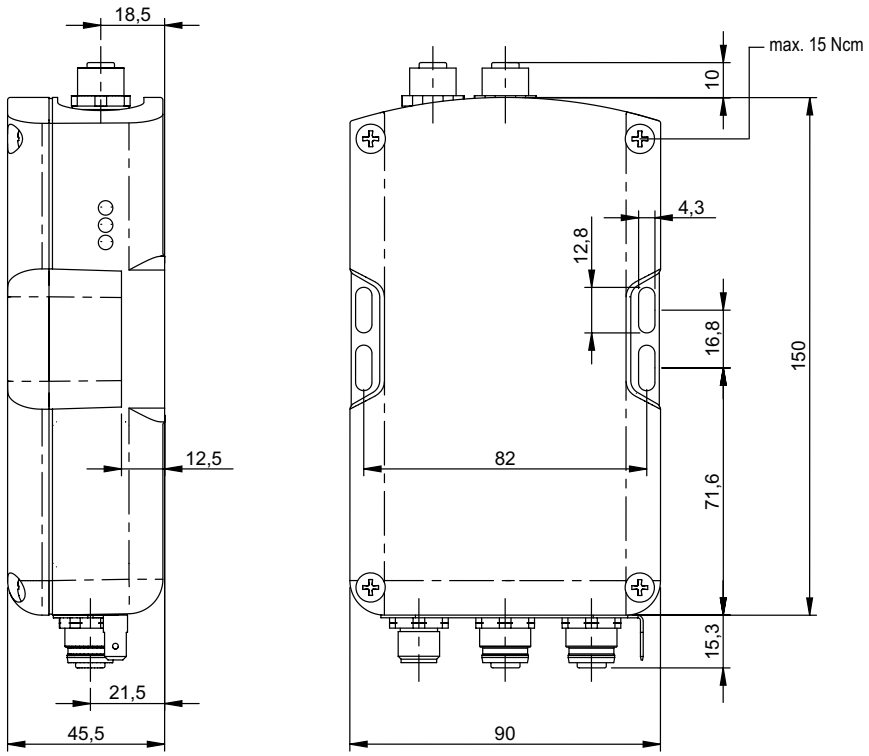


Fig. 5: Dimensions in mm

Mechanical Data

Housing material	ABS
X1 – Supply voltage	V_S 24 V DC, 5-pin male panel connector, A-coded
X2 – PROFINET Port 1	Panel socket, M12 4-pin, D-coded
X3 – PROFINET Port 2	Panel socket, M12 4-pin, D-coded
Head 1, 2 (read/write head connections)	Male panel connector, 4-pin, A-coded
Enclosure rating	IP65 (with connectors)
Weight	500 g

Electrical Data

Operating voltage VS	24 V DC ±10 %
Ripple	≤ 10 %
Current draw	≤ 400 mA
Device interface	Ethernet
Service interface	RS 232

5 Technical Data

Operating Conditions

Ambient temperature range	0 °C…+60 °C
EMC – EN 61000-4-2/3/4/5/6 – EN 55011	 – Schärfegrad 3A/2A/3A/2A/3A – Gr. 1, Kl. A
Vibration/shock	EN 60068 Part 2-6/27/29/64/32

Function Indicators

BIS operating states	Ready CT1 Present/operating CT2 Present/operating	LED green LED green/yellow LED green/yellow
Ethernet status	Port 1 Link Port 2 Link Port 1 Activity Port 2 Activity	LED green LED green LED yellow LED yellow
PROFINET status	Status Error	LED green LED red

Note
For detailed description see "Function Indicators" on page 36.

5 Technical Data

5.2 BIS M-6028

Dimensions

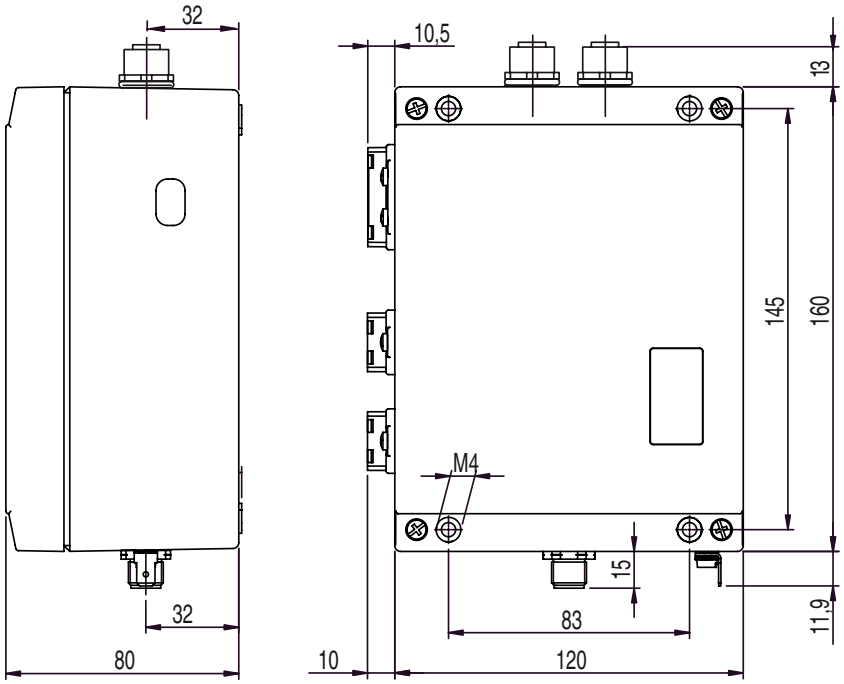


Fig. 6: Dimensions in mm

Mechanical Data

Housing material	EN AC-AlSi12 (a), DIN EN 1706
X1 – Supply voltage	V_S 24 V DC, Push-Pull Power connector
X2 – PROFINET Port 1	RJ45 plug (AIDA recommendation)
X3 – PROFINET Port 2	RJ45 plug (AIDA recommendation)
Head 1, 2 (read/write head connections)	Male panel connector, 4-pin, A-coded
Enclosure rating	IP65 (with connectors)
Weight	1080 g

Electrical Data

Operating voltage VS	24 V DC ±10 %
Ripple	≤ 10 %
Current draw	≤ 400 mA
Device interface	Ethernet
Service interface	RS 232

5 Technical Data

Operating Conditions

Ambient temperature range	0 °C…+60 °C
EMC – EN 61000-4-2/3/4/5/6 – EN 55011	 – Schärfegrad 3A/2A/3B/2A/XA – Gr. 1, Kl. A+B
Vibration/Shock	EN 60068 Part 2-6/27/29/64/32

Function Indicators

BIS operating states	Ready CT1 Present/operating CT2 Present/operating	LED green LED green/yellow LED green/yellow
Ethernet status	Port 1 Link Port 2 Link Port 1 Activity Port 2 Activity	LED green LED green LED yellow LED yellow
PROFINET status	Status Error	LED green LED red

Note
For detailed description see "Function Indicators" on page 37.

6 Installation

6.1 Processor installation

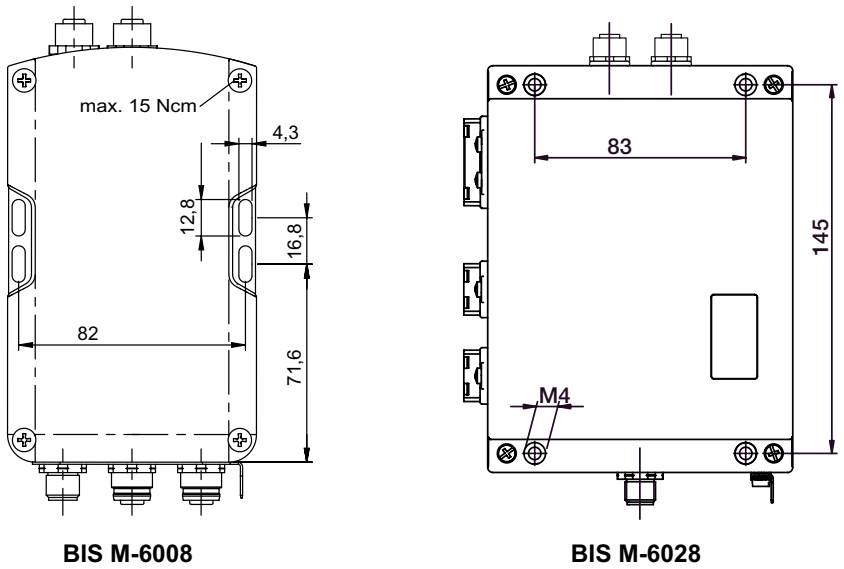


Fig. 7: Mechanical connection (dimensions in mm)

► Attach processor using 4 M4 screws.

Installation with support rail (accessory for BIS M-6008)

Installation examples using mounting bracket and rail holder BIS Z-HW-001 (accessory).

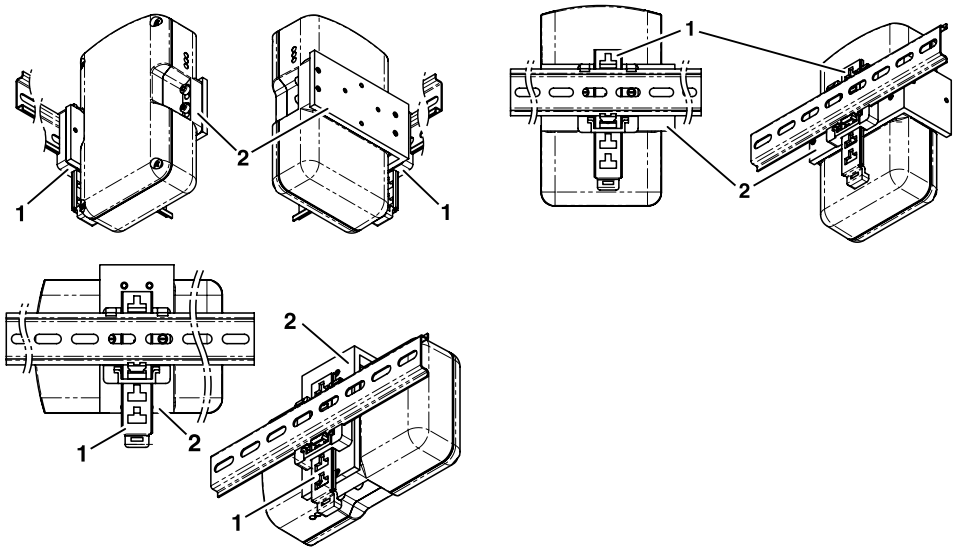


Fig. 8: Mounting using rail holder

1 Rail holder
2 Mounting brackets

6 Installation

6.2 Interface-information/ Wiring diagrams

BIS M-6008

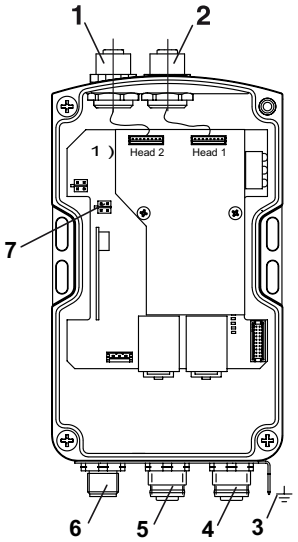
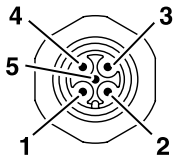


Fig. 9: BIS M-6008 connections

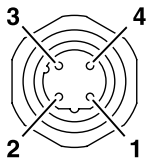
1 Head 2 – Read/write head 2
2 Head 1 – Read/write head 1
3 Function ground FE
4 X3 – PROFINET Port 2
5 X2 – PROFINET Port 1
6 X1 – Supply voltage
7 X7 – Service port

X1 – Supply voltage



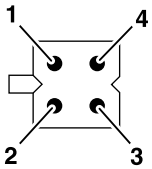
PIN	Function
1	+Vs
2	n.c.
3	-Vs
4	n.c.
5	n.c.

X2, X3 – PROFINET



PIN	Function
1	TD+
2	RD+
3	TD-
4	RD-

X7 – Service port



PIN	Function
1	TxD
2	RxD
3	GND
4	n.c.

Attention!
Make the ground connection either directly or using an RC combination to ground. When making your connection to the Ethernet, be sure that the shield is perfectly connected to the connector body.

6 Installation

BIS M-6028

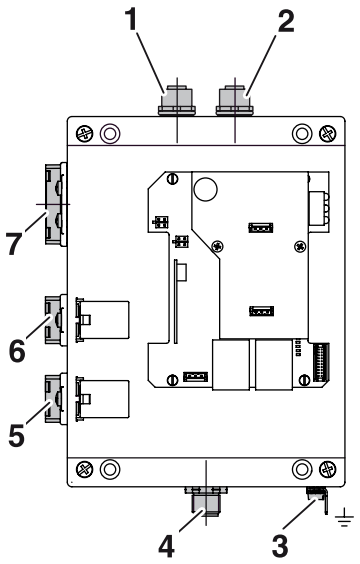
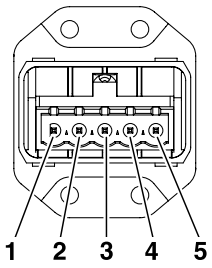


Fig. 10: BIS M-6028 connections

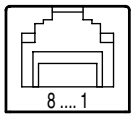
1 Head 2 – Read/write head 2
2 Head 1 – Read/write head 1
3 Function ground FE
4 X4 – Service port
5 X3 – PROFINET Port 2
6 X2 – PROFINET Port 1
7 X1 – Supply voltage

X1 – Supply voltage



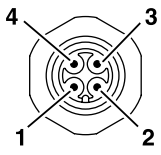
PIN	Function
1	+Vs
2	-Vs
3	n.c.
4	n.c.
5	n.c.

X2, X3 – PROFINET



PIN	Function
1	TD+
2	TD-
3	RD+
4	n.c.
5	n.c.
6	RD-
7	n.c.
8	n.c.

X4 – Service port



PIN	Function
1	n.c.
2	TxD
3	GND
4	RxD

7 Bus connection

7.1 Project administration

In project administration of fieldbus devices, a physical device is mapped as a modular system consisting of a head module and multiple data modules. The device needed for project planning are stored in GSD files (**G**eneral-**S**tation-**D**escription)
in GSDML format (**G**eneral **S**tation **D**escription **M**arkup **L**anguage).

GSD file

The GSDML file for the BIS M-60_8 is contained on the included product CD.
The data modules of an IO-Link device are represented in the project planning software by slot.
The GSDML file provides the possible data modules (inputs or outputs of various data width). For configuring an IO-Link device the appropriate data modules are assigned to a particular port.

Data modules

Input and output modules of 8 bytes, 16 bytes, 32 bytes, 64 bytes, 128 Byte and 254 bytes can be configured.

Integration in project planning software

Project administration is accomplished using the project administration tool "SIMATIC NCM PC Manager“ or using "STEP 7“.
The following steps are required for integrating a BIS M-60_8 processor:

1. Install the GSDML file of the IO device in the hardware configuration
2. Update catalog
3. Use "Insert object" to add the IO device "BIS M-60x8_RT“ or „BIS M-60x8_IRT“
4. Insert both modules for inputs and outputs
   (e.g.: "RT 32 Byte E“ and "RT 32 Byte A“ for processor "BIS M-60x8_RT“ or "IRT 32 Byte E“ and "IRT 32 Byte A“ for processor "BIS M-60x8_IRT“)

Additional project administration steps:

5. The name suffix "RT" or "IRT" tells you how the read and write data are exchanged

Note
Both processors have a 2-port IRT switch and are therefore able to pass IRT data packets.

The object properties of these modules can be used to set the start addresses of the input and output data.

Note
The input and output data can be used to control the BIS M-60_8 as described in Section 9.

7.2 Device name and IP address

6. The object properties of the inserted object "m-60_8" can be used to assign the device name, the device number and the IP address

The processor and the host system communicate using PROFINET protocol. This means an IP address and a unique device name are required. The device name and IP address can be saved in the IO device using "Target system > Ethernet > Edit Ethernet device“.

Note
The BIS M-60_8 processor is shipped without a device name. In the included GSDML file the device name "m-60x8" is preset.

8 Setting the processor parameters

8.1 Basic knowledge

Data carrier

The following data carriers may be used with the BIS M-60 _8 processor:

Mifare data carriers:

Balluff data carrier type	Manufacturer	Description	Memory capacity	Memory type
BIS M-1_ _-**01**	Philips	Mifare Classic	752 bytes	EEPROM

ISO15693 data carriers:

Balluff data carrier type	Manufacturer	Description	Memory capacity	Memory type
BIS M-1_ _-**02**	Fujitsu	MB89R118	2000 bytes	FRAM
BIS M-1_ _-**03***	Philips	SL2ICS20	112 bytes	EEPROM
BIS M-1_ _-**04***	Texas Instruments	TAG-IT Plus	256 bytes	EEPROM
BIS M-1_ _-**05***	Infineon	SRF55V02P	224 bytes	EEPROM
BIS M-1_ _-**06***	EM	EM4135	288 bytes	EEPROM
BIS M-1_ _-**07***	Infineon	SRF55V10P	992 bytes	EEPROM

* On request

Note
The data carriers contain additional memory ranges for configuration and protected data. These ranges cannot be processed using the BIS M-60_8 processor.

CRC check

The CRC check is a procedure for determining a check value for data in order to be able to recognize transmission errors. If the CRC check is activated, an error message is sent when a CRC error is detected.

Initializing
To use the CRC data check, the data carriers must be initialized. Initialization of the data carriers is accomplished in the output buffer using the command designator 12_{hex}. If the data carrier does not receive the correct CRC, the processor sets an error message in the input buffer (see example 10 on page 48).
Data carriers as shipped from the factory can be written immediately with a checksum, since all data are set to 0.

Error message
- If an error message is the result of a failed write job, the data carrier must be reinitialized to make it usable again.
- If an error message is not the result of a failed write job, the one or more memory cells in the data carrier is likely defective. The data carrier must be replaced.

8 Setting the processor parameters

Checksum
The checksum is written to the data carrier as 2 bytes of information. 2 bytes per block are sacrificed. This leaves 14 bytes per block available. The usable number of bytes can be found in the following table.

Data carrier	Balluff data carrier type	Memory capacity	
		w/o CRC checksum	with CRC checksum
Mifare	BIS M-1_ _-**01**	752 bytes	658 bytes
ISO 15693	BIS M-1_ _-**02**	2000 bytes	1750 bytes
	BIS M-1_ _-**03***	112 bytes	98 bytes
	BIS M-1_ _-**04***	256 bytes	224 bytes
	BIS M-1_ _-**05***	224 bytes	196 bytes
	BIS M-1_ _-**06***	288 bytes	252 bytes
	BIS M-1_ _-**07***	992 bytes	868 bytes

* On request

Simultaneous data transmission

Read
The processor reads the data from the data carrier directly into the input buffer. As soon as the buffer is full, the Toggle-Bit Out (TO-Bit) is inverted to indicate data ready to the controlling system. By inverting the toggle-Bit In (TI-Bit) the controlling system indicates it is ready to receive and data read in the meantime are sent to the input buffer. This is repeated until the desired data have been read by the data carrier. After the read process is finished the processor sets the Job End-Bit (AE-Bit) and sends the remaining data to the input buffer (see example 2 on page 39).

Write
The processor begins to write the data to the data carrier as soon as it has received the first data from the controlling system. Once all the data have been written to the data carrier, the AE-Bit is set.

Dynamic mode

As soon as the dynamic mode function (*Dynamic*) is activated, the processor accepts the read/write job from the controlling system and stores it, regardless of whether a data carrier is in the active zone of the read/write head or not. As soon as a data carrier enters the active zone of the read/write head, the stored job is run.

Auto Read (Standard)

When a data carrier enters the active zone of the read/write head 14 bytes starting at address 00_{hex} are automatically read into the input buffer. No additional read command is required. This allows small data amounts which are stored starting at address 00_{hex}.

If the parameter *TypeSN* (type and serial number when CT Present) is set, then instead of the user data the data carrier type and the unique serial number of the data carrier are sent.

Auto Read (CP-Extra)

If Auto-Read *CP-Extra* is enabled, the data are read starting at a specified start address as soon as a data carrier is recognized. These data are provided to the input buffer using the rising edge of the CP bit. The start address must be specified for each head using the command designator 07_{hex} . The start addresses may differ. The number of bytes read is determined by the selected size of the input buffer, which is divided over both heads when two heads are used.

Type and serial number

If this function is enabled, when CT present occurs the data carrier type and serial number (UID = unique 10) for the data carrier are output.
For data carrier type BIS M-1_ _-01 the serial number is 4 bytes in size. For all other data carrier types the serial number is 8 bytes.

8 Setting the processor parameters

8.2 Setting parameters

Parameter setting for project planning/integration is accomplished using project planning software such as "SIMATIC NCM PC Manager“ or "STEP 7". Alternately the parameter setting bytes can be sent directly using the controller.
The parameters for operating the processor are stored in the GSDML file.

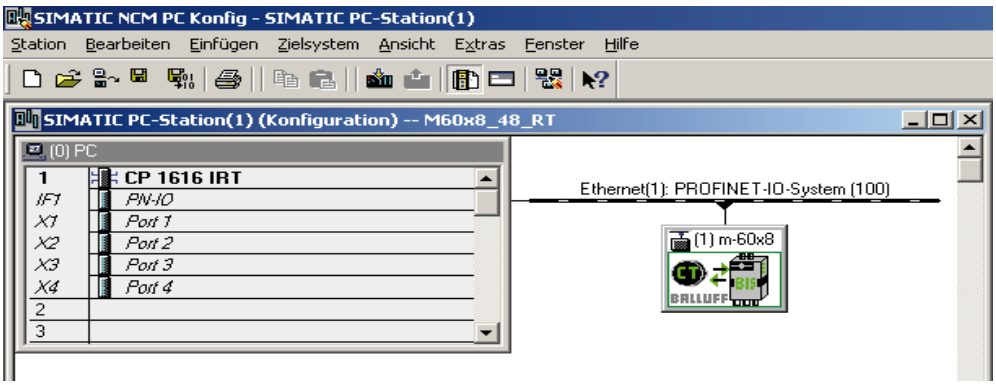
GSDML file
The GSDML file contains all the device parameters for the processor. The file can be found on the included BIS-CD.

Parameter overview
- **CRC 16:**
  If this function is activated, the correctness of the read/written data is ensured by the CRC 16 data check.
- **CP Extra for read/write head 1 or 2:**
  If the function *CP Extra* is enabled, a start address other than 0 can be selected. The start address must first be stored on the EEPROM in the processor using the command designator 07_{hex}.
- **Dynamic for read/write head 1 or 2:**
  If Dynamic is set, a read/write job can be sent even though there is no data carrier in the active range of the head. Now if a data carrier arrives at the head, the command is immediately executed (saves time).
- **DT Type:**
  This parameter is used to set which data carrier types can be recognized and processed.
- **Simultaneous data transmission for both read/write heads:**
  In simultaneous data transmission shorter read/write times can be achieved depending on the amount of data to be read or written and the type of controller.
- **Buffer Head 1:**
  This value indicates how many bytes of the entire in- and output buffer should be used for Head 1. The remainder of the in- and output buffer is then available for Head 2.

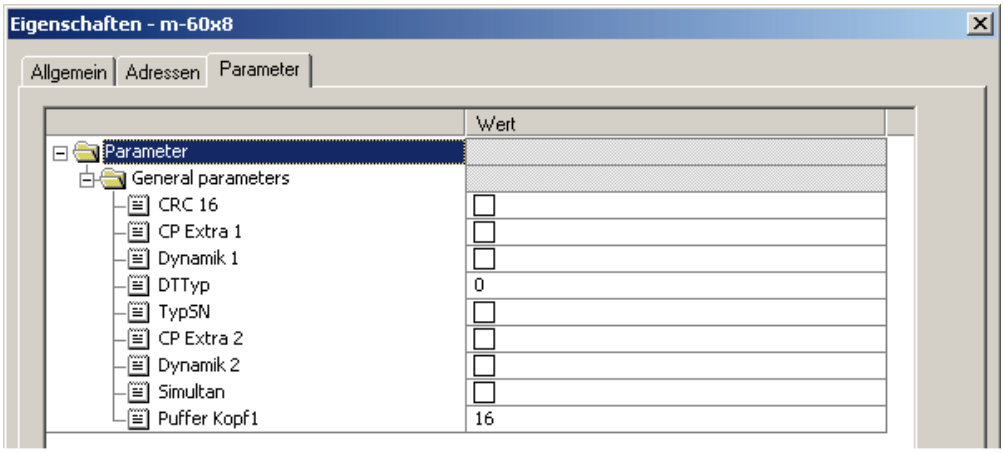
8 Setting the processor parameters

Parameter setting for project planning

The project planning software (e.g. "SIMATIC NCM PC Manager“ or "STEP 7") can be used to parameterize the processor in the hardware configuration.
For this the added IO device is selected and the object properties opened by right-clicking on "m-60_8" on Slot 0:



The "Parameters" tab is used to open the selection window for parameters:



8 Setting the processor parameters

Parameter bytes

The user parameters have a data length of 6 bytes and can be set in project planning. The presetting is stored in the GSDML file.

> ⚠ **Attention!**
> Incorrect parameter setting may result in malfunction of the processor.
> ► The 6 bytes must be sent in full and in hexadecimal format.
> ► Only the indicated bytes may be changed.

	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte
Hex[1)]	00	80	00	82	00	02
Binary	000**0**0000	100**00**000	**00000000**	10000010	**0**00**00**000	**00000010**
	Bit 5	Bit 5, Bit 4	Bits 1…8		Bit 8	Bit 5, Bit 4	Bits 1…8

Bit status: 1 = activated, 0 = deactivated
1) Factory default setting

The parameter bytes may be used to activate or deactivate the following functions:

Bit	Function
1st byte, bit 5	Activate CRC_16 checksum
2nd byte, bit 5	Activate dynamic mode on read/write head 1
2nd byte, bit 4	Activate Auto Read starting at specified address after CT Present for Head 1
3rd byte, bits 1...8	Selection fore the data carrier type to be processed. 00_{hex}: All data carrier types FE_{hex}: Mifare (all Mifare data carriers supported by Balluff) FF_{hex}: ISO 15693 (all ISO 15693 data carriers supported by Balluff)
5th byte, bit 8	Activate simultaneous data transmission for both read/write heads
5th byte, bit 5	Activate dynamic mode on read/write head 2
5th byte, bit 4	Activate Auto Read starting at specified address after CT Present for Head 2
6th byte, bits 1...8	Distribution of the input/output buffer over the two read/write heads. Bits 1…8 indicate the number of bytes used for Read/Write Head 1. The remaining bytes of the input/output buffer are assigned to Read/Write Head 2. The number must be at least 02_{hex}, with a maximum value of FE_{hex} (254 bytes).

8 Setting the processor parameters

Notes

9 Device Function

9.1 Function principle BIS M-60_8

Two buffers are needed to exchange data and commands between the processor and the host system. The buffer contents is exchanged using cyclical polling. The buffer contents depends on the cycle in which it is written (e.g. control commands at the beginning of a job).
When writing the buffer, the sent data from the preceding cycle are overwritten. Unwritten bytes are not deleted and retain their data content.

Entire buffer

The buffer size of the entire buffer corresponds to which module is plugged in (RT or IRT). The distribution over the two read/write heads is done using the parameter "*Buffer Head1*".

Example:
If an entire buffer of 16 bytes is selected and the parameter "Buffer Head1" is set to 10 bytes, then 6 bytes remain for Head 2.

2 bytes less per read/write head are available for data exchange, since the first and last bytes of the respective data buffer is used for control and for status messages.

Output buffer

The control commands for the identification system and the data to be written to the data carrier are sent through the output buffer.

Bit-No. / Subaddress	7	6	5	4	3	2	1	0
00_{hex} = bit header		TI				GR		AV
01_{hex}	Command designator					or	Data	
02_{hex}	Start address (Low Byte) or program no.					or	Data	
03_{hex}	Start address (High Byte)					or	Data	
04_{hex}	Number of bytes (Low Byte)					or	Data	
05_{hex}	Number of bytes (High Byte)					or	Data	
06_{hex}	Data							
…	Data							
Last byte	2nd bit header (as above)					or	Data	

Configuration and explanation (output buffer)

Subaddress	Bit name	Meaning	Function description
00_{hex} = bit header	TI	Toggle-Bit In	Controller is ready to receive additional data (read job).
	GR	Ground state	Identification system goes into the base state for the respective read/write head. Any pending job is canceled.
	AV	Job	A job is pending for the respective read/write head.

9 Device Function

Output buffer (cont.)

Subaddress	Meaning	Function description
01hex	Command designator 00hex 01hex 02hex 06hex 07hex 12hex 21hex 22hex	 No command present. Read data carrier. Write to data carrier Store program for "Mixed Data Access" in the EEPROM. Store start address for the "Auto Read" function in the EEPROM. Initialize CRC_16 data check Read data carrier as per a program for "Mixed Data Access“. Write to data carrier as per a program for „Mixed Data Access“.
	or data	Send the data which are being written to the data carrier.
	or program data	Send the program data which are being written to the EEPROM.
02hex	Start address (Low Byte)	Start address for reading or writing (address range from 0 to 255 is available).
	or Start address (Low Byte)	Start address for the "Auto Read" function starting at which the data carrier is read. The value is stored in the EEPROM (address range from 0 to 255 is available).
	or Program No.	Number of the program to be stored for "Mixed Data Access" together with command designator 06hex (value range 01hex to 0Ahex).
	or Program No.	Number of the program to be executed for "Mixed Data Access" together with command designator 21hex or 22hex (value range 01hex to 0Ahex).
	or Data	Send the data which are being written to the data carrier.
	or program data	Send the program data which are being written to the EEPROM.
03hex	Start address (High Byte)	Start address for reading or writing (The High Byte is also needed for the address range 256 to 1999).
	or Start address (High Byte)	Start address for the "Auto Read" function starting at which the data carrier is read. The value is stored in the EEPROM. (The High Byte is also needed for the address range 256 to 1999).
	or Data	Send the data which are being written to the data carrier.
	or program data	Send the program data which are being written to the EEPROM.

9 Device Function

Output buffer (cont.)

Subaddress	Meaning	Function description
04_{hex}	No. of bytes (Low Byte)	Numbers of bytes to read or write beginning with the start address. The number (1 to 2000) is formatted in hex (0001_{hex} to $07D0_{hex}$). It is split into low and high bytes. Possible values for the low byte are therefore 1 to 255 (01_{hex} to FF_{hex})
	or Data	Send the data which are being written to the data carrier.
	or program data	Send the program data which are being written to the EEPROM.
05_{hex}	No. of bytes (High Byte)	Numbers of bytes to read or write beginning with the start address. The number (1 to 2000) is formatted in hex (0001_{hex} to $07D0_{hex}$). It is split into low and high bytes. Possible values for the high byte are therefore 0 to 7 (00_{hex} to 07_{hex})
	or Data	Send the data which are being written to the data carrier.
	or program data	Send the program data which are being written to the EEPROM.
06_{hex}	Data	Send the data which are being written to the data carrier.
	or program data	Send the program data which are being written to the EEPROM.
...	Data	Send the data which are being written to the data carrier.
	or program data	Send the program data which are being written to the EEPROM.
Last byte	2. bit header	If 1st and 2nd bit headers agree, there are valid data.

9 Device Function

Input buffer

The input buffer is used to send the data read from the identification system, the designations and the error codes to the host system.

Bit-No. / Subaddress	7	6	5	4	3	2	1	0
00_{hex} = bit header	BB	HF	TO		AF	AE	AA	CP
01_{hex}	Error code or Data							
02_{hex}	Data							
…	Data							
Last byte	2nd bit header (as above)							

Configuration and explanation (input buffer)

Subaddress	Bit name	Meaning	Function description
00_{hex} = bit header	BB	Power	Identification system is ready.
	HF	Head error	Cable break read/write head or read/write head not connected
	TO	Toggle-Bit Out	**Read:** Additional data are being provided by the identification system. **Write:** Identification system can accept additional data.
	AF	Job error	Error in processing the job or job canceled.
	AE	Job End	Confirmation – Job finished without error
	AA	Job Start	Confirmation – Job was recognized and begun
	CP	Codetag Present	Data carrier in range of the read/write head
		Parallel to the CP-Bit the output signal "CT-Present" is available The presence of a data carrier can be processed directly as a hardware signal.	

9 Device Function

Input buffer (cont.)

Subaddress	Meaning	Function description
01hex	Error code	Error number only valid with AF bit!
	00hex	No error.
	01hex	Job cannot be executed because there is no data carrier in the range of the read/write head.
	02hex	Read error.
	03hex	Data carrier was removed from the read/write head during reading.
	04hex	Write error.
	05hex	Data carrier was removed from the read/write head during writing.
	07hex	No command designator or invalid designator for set AV-Bit or the number of bytes is 00hex.
	09hex	Cable break read/write head or read/write head not connected
	0Chex	EEPROM cannot be read or written.
	0Dhex	Communication fault with data carrier. **Note:** Check installation criteria or distance of data carrier from read/write head.
	0Ehex	CRC for the read data and CRC for the data carrier do not agree.
	0Fhex	1st and 2nd bit header are not equal The 2nd bit header must be used.
	or data	Transmission of data which were read from the data carrier.
02hex	Data	Transmission of data which were read from the data carrier.
...	Data	Transmission of data which were read from the data carrier.
Last byte	2. bit header	If 1st and 2nd bit headers agree, there are valid data.

9 Device Function

Communication

Communication between the controlling system and processor is defined by a sequence protocol. Communication between host system and processor is implemented using control bit in the output and input buffer.

Basic sequence

1. Controller sends command designator in the output buffer with AV-Bit set.
   The AV-Bit tells the processor that a job is beginning and the sent data are valid.
2. Processor accepts the job and confirms the job by setting the AA-Bit in the input buffer.
3. If additional data need to be exchanged for the job, readiness for additional data exchange is indicated by inverting the toggle bits Bit TI and TO.
4. Processor has correctly executed the job and sets the AE-Bit in the input buffer.
5. Controller has accepted all the data. The AV-Bit in the output buffer is set.
6. Processor resets all the control bits set in the input buffer during the job (AA-Bit, AE-Bit). The processor is ready for the next job.

Mixed data access

By running the read/write programs it is possible to write data to different address ranges in the data carrier or read data located in different address ranges on the data carrier. This function is referred to as "Mixed Data Access".
The read/write programs are stored in the processor's EEPROM. 10 programs with up to 25 instructions can be stored. Each program instruction contains the start address and number of bytes. The amount of data which can be sent is limited to 2 KB.

Storing programs:
The command designator 06_{hex} in the output buffer sends the program to the processor. Storing of a program is a job. All 25 instructions and two additional bytes with $FF_{hex}FF_{hex}$ as a termination designator must always be sent. This means 104 bytes per program, including command designator and program number are to be sent (see example 7 on page 44).

Note
The individual program records must be attached to each other with no gaps. They must be sent in order and terminated with 2 bytes FF_{hex} as a terminator. It is recommended that the remaining unused memory range be filled with FF_{hex}. If an address range is selected twice, the data are also sent twice.

9 Device Function

Example: Program structure

Program structure	Subaddress	Value	Value range
Command designator	01_{hex}	06_{hex}	
1st program record			
Program number	02_{hex}	01_{hex}	01_{hex} bis $0A_{hex}$
1. 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}		
2. Data record			
…			
25. 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}		
Termination character	$FF_{hex}FF_{hex}$		

Running programs:
The programs stored in the EEPROM can be used both for reading data records from the data carrier and for writing data records to a data carrier. Whether to read or write is determined by command designator 21_{hex} (read) or 22_{hex} (write) in the output buffer (see example 8 on page 45 and example 9 on page 46).

9 Device Function

Read/write times

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

Read times:

Data carrier with 16 byte blocks*	BIS M-1_ _-01	BIS M-1_ _-02
Data carrier recognition	≤ 20 ms	≤ 20 ms
Read bytes 0 to 15	≤ 20 ms	≤ 30 ms
for each additional start of 16 bytes	≤ 10 ms	≤ 15 ms

Write times:

Data carrier with 16 byte blocks*	BIS M-1_ _-01	BIS M-1_ _-02
Data carrier recognition	≤ 20 ms	≤ 20 ms
Write bytes 0 to 15	≤ 40 ms	≤ 60 ms
for each additional start of 16 bytes	≤ 30 ms	≤ 40 ms

* Values for data carriers BIS M-1_ _-03 to BIS M-1_ _-07 on request

9 Device Function

9.2 Function indicators

The operating states of the identification system, the Ethernet/IP connection and the Ethernet connection are indicated by LEDs.
On the BIS M-6008 processor, the LEDs 4 to 9 (indicators for Ethernet and PROFINET status) are not visible when the housing is closed.

Overview
Display elements
BIS M-6008

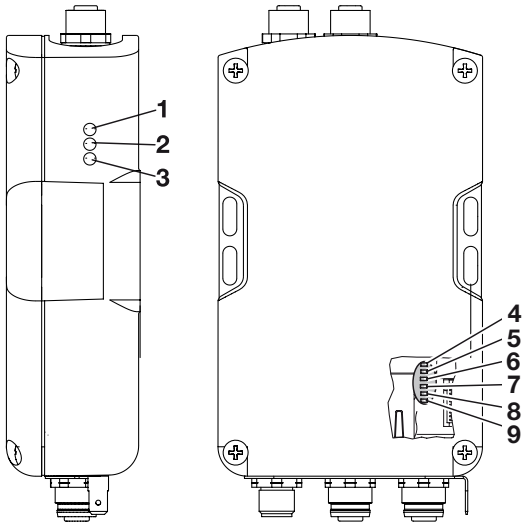


Fig. 11: Function indicators

1 Ready (green)
2 CT1 Present/Operating (green/yellow)
3 CT2 Present/Operating (green/yellow)
4 Port 1 Link (Ethernet)
5 Port 2 Link (Ethernet)
6 Port 1 Activity (Ethernet)
7 Port 2 Activity (Ethernet)
8 Status (PROFINET)
9 Error (PROFINET)

Link LED	Meaning
off	Ethernet – no connection
green	Ethernet – connection OK

Activity LED	Meaning
off	Ethernet – no activity
flashing yellow	Ethernet – RX-/TX activity

Status LED	Meaning
off	PROFINET not yet ready
green	PROFINET ready

Error LED	Meaning
off	PROFINET – connection OK
red	PROFINET – connection in Time-Out

9 Device Function

Overview
Display elements
BIS M-6028

On the BIS M-6028 processor, the LEDs 4 to 9.(for the Display of ethernet and PROFINET status) are visible integrated in the housing.

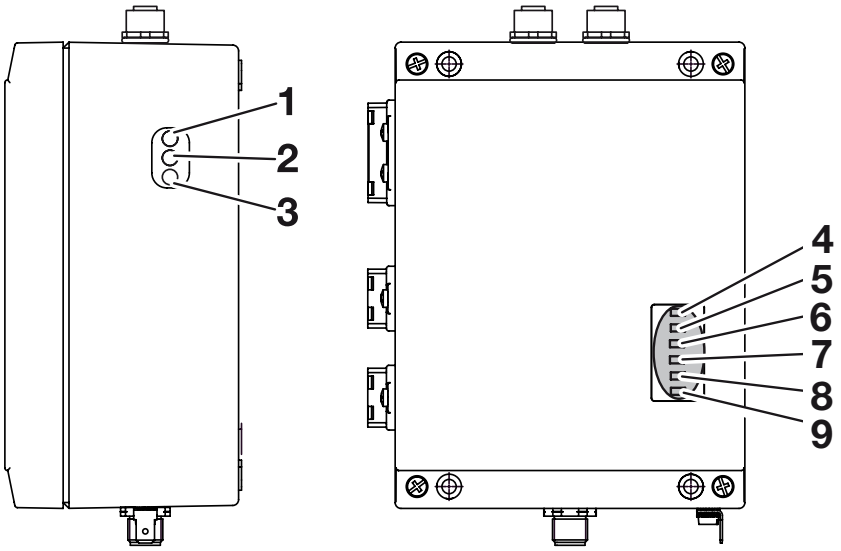


Fig. 12: Function indicators

1	Ready (green)	**4**	Port 1 Link (Ethernet)	**7**	Port 2 Activity (Ethernet)
2	CT1 Present/Operating (green/yellow)	**5**	Port 2 Link (Ethernet)	**8**	Status (PROFINET)
3	CT2 Present/Operating (green/yellow)	**6**	Port 1 Activity (Ethernet)	**9**	Error (PROFINET)

Link LED	Meaning
off	Ethernet – no connection
green	Ethernet – connection OK

Activity LED	Meaning
off	Ethernet – no activity
flashing yellow	Ethernet – RX-/TX activity

Status LED	Meaning
off	PROFINET not yet ready
green	PROFINET ready

Error LED	Meaning
off	PROFINET – connection OK
red	PROFINET – connection in Time-Out

9 Device Function

9.3 Examples

1st example

Initializing the data carrier for CRC_16 data check
This command is considered to be a write command. The start address and number of bytes must correspond to the maximum used data quantity.

In the example the complete memory area of a data carrier (752 bytes) is used. 658 bytes on the data carrier are available as data bytes, since 94 bytes are required for the CRC.

For configuration with 8-byte buffer size!

Command from controller		BIS M-60_8 response	
1. Process output buffer (note sequence):		2. Process input buffer: (note sequence):	
01_{hex}	Command designator 12_{hex}	$00_{hex}/07_{hex}$	Set AA bit, invert TO-Bit
02_{hex}	Start address 00_{hex}		
03_{hex}	Start address 00_{hex}		
04_{hex}	No. of bytes 92_{hex}		
05_{hex}	No. of bytes 02_{hex}		
$00_{hex}/07_{hex}$	Set AV-Bit		
3. Process output buffer:		4. Process output buffer:	
$01…06_{hex}$	Enter first 6 bytes	$01…06_{hex}$	Copy first 6 bytes
$00_{hex}/07_{hex}$	Invert TI bit	Process input buffer:	
		$00_{hex}/07_{hex}$	Invert TO bit
5. Process output buffer:		6. Process output buffer:	
$01…06_{hex}$	Enter second 6 bytes	$01…06_{hex}$	Copy second 6 bytes
$00_{hex}/07_{hex}$	Invert TI bit	Process input buffer:	
		$00_{hex}/07_{hex}$	Invert TO bit
Continue until the entire memory range is filled.			
7. Process output buffer:		8. Process output buffer:	
$01…06_{hex}$	Enter last bytes	$01…06_{hex}$	Copy last bytes
$00_{hex}/07_{hex}$	Invert TI bit	Process input buffer:	
		$00_{hex}/07_{hex}$	Invert TO bit
9. Process output buffer:		10. Process input buffer:	
$00_{hex}/07_{hex}$	Reset AV bit	$00_{hex}/07_{hex}$	Reset AA and AE bits

9 Device Function

2nd example **Read and write 17 bytes starting at address 10**

For configuration with 8-byte buffer size!

Command from controller		BIS M-60_8 response	

1. Process output buffer (note sequence):

01_{hex}	Command designator 01_{hex}
02_{hex}	Start address $0A_{hex}$
03_{hex}	Start address 00_{hex}
04_{hex}	No. of bytes 11_{hex}
05_{hex}	No. of bytes 00_{hex}
$00_{hex}/07_{hex}$	Set AV-Bit

2. Process input buffer: (note sequence):

$00_{hex}/07_{hex}$	Set AA bit
$01…06_{hex}$	Enter first 6 bytes
$00_{hex}/07_{hex}$	Set AE bit

3. Process input buffer:

$01…06_{hex}$	Copy first 6 bytes

Process output buffer:

$00_{hex}/07_{hex}$	Invert TI bit

4. Process input buffer:

$01…06_{hex}$	Enter second 6 bytes
$00_{hex}/07_{hex}$	Invert TO bit

5. Process input buffer:

$01…06_{hex}$	Copy second 6 bytes

Process output buffer:

$00_{hex}/07_{hex}$	Invert TI bit

6. Process input buffer:

$01…05_{hex}$	Enter last 5 bytes
$00_{hex}/07_{hex}$	Invert TO bit

7. Process input buffer:

$01…05_{hex}$	Copy last 5 bytes

Process output buffer:

$00_{hex}/07_{hex}$	Reset AV bit

8. Process input buffer:

$00_{hex}/07_{hex}$	Reset AA and AE bits

9 Device Function

3rd example **Read 17 bytes starting at address 10 with simultaneous data transmission**

For configuration with 8-byte buffer size!

During the read job and as soon as sufficient data have been read, in order to fill the input buffer of Read/Write Head 1, the data are sent to the input buffer. The AE-Bit is not set until the processor has finished the "Read" operation.
The reply "Job End" (AE-Bit) is reliably set no later than before the last data have been sent. This time point depends on the requested data amount and the time response of the controller. In the example the italic format "*Set AE-Bit*" calls your attention to this fact.

Command from controller		BIS M-60_8 response	
1. Process output buffer (note sequence):		2. Process input buffer: (note sequence):	
01_{hex}	Command designator 01_{hex}	$00_{hex}/07_{hex}$	Set AA bit
02_{hex}	Start address $0A_{hex}$	$01…06_{hex}$	Enter first 6 bytes
03_{hex}	Start address 00_{hex}	$00_{hex}/07_{hex}$	Invert TO bit
04_{hex}	No. of bytes 11_{hex}	$00_{hex}/07_{hex}$	*Set AE bit*
05_{hex}	No. of bytes 00_{hex}		
$00_{hex}/07_{hex}$	Set AV-Bit		
3. Process input buffer:		4. Process input buffer:	
$01…06_{hex}$	Copy first 6 bytes	$01…06_{hex}$	Enter second 6 bytes
Process output buffer:		$00_{hex}/07_{hex}$	Invert TO bit
$00_{hex}/07_{hex}$	Invert TI bit	$00_{hex}/07_{hex}$	*Set AE bit*
5. Process input buffer:		6. Process input buffer:	
$01…06_{hex}$	Copy second 6 bytes	$01…05_{hex}$	Enter last 5 bytes
Process output buffer:		$00_{hex}/07_{hex}$	Invert TO bit
$00_{hex}/07_{hex}$	Invert TI bit	$00_{hex}/07_{hex}$	*Set AE bit*
7. Process input buffer:		8. Process input buffer:	
$01…05_{hex}$	Copy last 5 bytes	$00_{hex}/07_{hex}$	Reset AA and AE bits
Process output buffer:			
$00_{hex}/07_{hex}$	Reset AV bit		

9 Device Function

4th example **Read 30 bytes starting at address 10 with read error**

For configuration with 8-byte buffer size!

> **Note**
> If an error occurs, the AF-Bit is set instead of the AE-Bit, together with a corresponding error number. Setting the AF-Bit cancels the operation and declares it as finished.

Command from controller	BIS M-60_8 response

1. Process output buffer (note sequence):

01_{hex}	Command designator 01_{hex}
02_{hex}	Start address $0A_{hex}$
03_{hex}	Start address 00_{hex}
04_{hex}	No. of bytes $1E_{hex}$
05_{hex}	No. of bytes 00_{hex}
$00_{hex}/07_{hex}$	Set AV-Bit

2. Process input buffer: (note sequence): **If error occurs immediately!**

$00_{hex}/07_{hex}$	Set AA bit
01_{hex}	Enter error number
$00_{hex}/07_{hex}$	Set AF bit

3. Process input buffer:

01_{hex}	Copy error number

Process output buffer:

$00_{hex}/07_{hex}$	Reset AV bit

4. Process input buffer:

$00_{hex}/07_{hex}$	Reset AA and AF bits

Device Function

5th example

Read 30 bytes starting at address 10 with read error and simultaneous data transmission

For configuration with 8-byte buffer size!

> **Note**
> If an error occurs, the AF-Bit is set instead of the AE-Bit, together with a corresponding error number. Setting the AF-Bit cancels the operation and declares it as finished.

Command from controller		BIS M-60_8 response	
1. Process output buffer (note sequence):		2. Process input buffer: (note sequence): **If error occurs immediately!**	
01_{hex}	Command designator 01_{hex}	$00_{hex}/07_{hex}$	Set AA bit
02_{hex}	Start address $0A_{hex}$	01_{hex}	Enter error number
03_{hex}	Start address 00_{hex}	$00_{hex}/07_{hex}$	Set AF bit
04_{hex}	No. of bytes $1E_{hex}$		
05_{hex}	No. of bytes 00_{hex}		
$00_{hex}/07_{hex}$	Set AV-Bit		
3. Process input buffer:		4. Process input buffer:	
01_{hex}	Copy error number	$00_{hex}/07_{hex}$	Reset AA and AF bits
Process output buffer:			
$00_{hex}/07_{hex}$	Reset AV bit		

> **Note**
> An error may also occur after data have already been sent
> (see example 6 on page 43).

9 Device Function

6th example

Read 30 bytes starting at address 10 with read error and simultaneous data transmission

For configuration with 8-byte buffer size!

Note
If an error occurs after sending of data has already begun, the AF-Bit is provided instead of the AE-Bit together with a corresponding error number. The error message AF is dominant. Which data are defective cannot be specified. Setting the AF-Bit cancels the operation and declares it as finished.

Command from controller		BIS M-60_8 response	
1. Process output buffer (note sequence):		2. Process input buffer: (note sequence):	
01_{hex}	Command designator 01_{hex}	$00_{hex}/0F_{hex}$	Set AA bit
02_{hex}	Start address $0A_{hex}$	$01…06_{hex}$	Enter first 6 bytes
03_{hex}	Start address 00_{hex}	$00_{hex}/07_{hex}$	Invert TO bit
04_{hex}	No. of bytes $1E_{hex}$		
05_{hex}	No. of bytes 00_{hex}		
$00_{hex}/07_{hex}$	Set AV-Bit		
3. Process input buffer:		4. Process input buffer: **If an error has occurred!**	
$01…06_{hex}$	Copy first 6 bytes	01_{hex}	Enter error number
Process output buffer:		$00_{hex}/07_{hex}$	Set AF bit
$00_{hex}/07_{hex}$	Invert TI bit		
5. Process input buffer:		6. Process input buffer:	
$01…0E_{hex}$	Copy error number	$00_{hex}/07_{hex}$	Reset AA and AF bits
Process output buffer:			
$00_{hex}/07_{hex}$	Reset AV bit		

9 Device Function

7th example **Write 16 bytes starting at address 20**

For configuration with 8-byte buffer size!

Command from controller		BIS M-60_8 response	
1. Process output buffer (note sequence):		2. Process input buffer: (note sequence):	
01_{hex}	Command designator 02_{hex}	$00_{hex}/07_{hex}$	Set AA bit, invert TO-Bit
02_{hex}	Start address 14_{hex}		
03_{hex}	Start address 00_{hex}		
04_{hex}	No. of bytes 10_{hex}		
05_{hex}	No. of bytes 00_{hex}		
$00_{hex}/07_{hex}$	Set AV-Bit		
3. Process output buffer:		4. Process output buffer:	
$01…06_{hex}$	Enter first 6 bytes	$01…06_{hex}$	Copy first 6 bytes
$00_{hex}/07_{hex}$	Invert TI bit	Process input buffer:	
		$00_{hex}/07_{hex}$	Invert TO bit
5. Process output buffer:		6. Process output buffer:	
$01…06_{hex}$	Enter second 6 bytes	$01…06_{hex}$	Copy second 6 bytes
$00_{hex}/07_{hex}$	Invert TI bit	Process input buffer:	
		$00_{hex}/07_{hex}$	Invert TO bit
7. Process output buffer:		8. Process output buffer:	
$01…04_{hex}$	Enter last 4 bytes	$01…04_{hex}$	Copy last 4 bytes
$00_{hex}/07_{hex}$	Invert TI bit	Process input buffer:	
		$00_{hex}/07_{hex}$	Set AE bit
9. Process output buffer:		10. Process input buffer:	
$00_{hex}/07_{hex}$	Reset AV bit	$00_{hex}/07_{hex}$	Reset AA and AE bits

9 Device Function

8th example **Programming start address 75**

For configuration with 8-byte buffer size!

Command from controller		BIS M-60_8 response	
1. Process output buffer (note sequence):		2. Process input buffer: (note sequence):	
01_{hex}	Command designator 07_{hex}	$00_{hex}/07_{hex}$	Set AA-bit and AE-bit
02_{hex}	Start address 48_{hex}		
03_{hex}	Start address 00_{hex}		
$00_{hex}/07_{hex}$	Set AV-Bit		
3. Process input buffer:		4. Process input buffer:	
$00_{hex}/07_{hex}$	Reset AV bit	$00_{hex}/07_{hex}$	Reset AA and AE bits

Note
To ensure correct data output, use the command designator 07_{hex} for each partial buffer of Head 1 and/or Head 2.
If the Auto Read function is not activated, the processor runs in standard mode and sends data starting with data carrier address 0 until the buffer is full, but to a maximum of 30 bytes.

9 Device Function

9th example **Mixed data access – Save program (three data records)**

For configuration with 8-byte buffer size!

1st data record	Start address	5	No. of bytes	7
2nd data record	Start address	75	No. of bytes	3
3rd data record	Start address	312	No. of bytes	17
Total exchanged in the operation:				27 bytes

All 104 bytes are written for the programming.

Command from controller	BIS M-60_8 response

1. Process output buffer (note sequence):

01hex	Command designator 06hex
02hex	Program number 01hex
00hex/07hex	Set AV-Bit

2. Process input buffer: (note sequence):

00hex/07hex	Set AA bit, invert TO-Bit

3. Process output buffer:

01hex	1st start address	05hex
02hex		00hex
03hex	1st no. of bytes	07hex
04hex		00hex
05hex	2nd start address	4Bhex
06hex		00hex
00hex/07hex	Invert TI bit	

4. Process input buffer:

00hex/07hex	Invert TO bit

5. Process output buffer:

01hex	2nd no. of bytes	03hex
02hex		00hex
03hex	3rd start address	38hex
04hex		00hex
05hex	3rd no. of bytes	11hex
06hex		00hex
00hex/07hex	Invert TI bit	

6. Process input buffer:

00hex/07hex	Invert TO bit

7. Process output buffer:

01hex/02hex	Terminator	FFhexFFhex
03hex … 04hex	(not used)	FFhexFFhex
05hex/06hex	(not used)	FFhexFFhex
00hex/07hex	Invert TI bit	

8. Process input buffer:

00hex/07hex	Invert TO bit

Continuation on next page

9 Device Function

9th example (cont.)

Continued from previous page

Command from controller			BIS M-60_8 response	
9. Process output buffer:			10. Process input buffer:	
$01_{hex}/02_{hex}$	Terminator	$FF_{hex}FF_{hex}$	$00_{hex}/07_{hex}$	Set AE bit
$03_{hex}/04_{hex}$	(not used)	$FF_{hex}FF_{hex}$		
$05_{hex}/06_{hex}$	(not used)	$FF_{hex}FF_{hex}$		
$00_{hex}/07_{hex}$	Invert TI bit			
11. Process output buffer:			12. Process input buffer:	
$00_{hex}/07_{hex}$	Reset AV bit		$00_{hex}/07_{hex}$	Reset AA and AE bits

Note
Fill all unused start addresses and number of bytes with FFhex!

9 Device Function

10th example **Mixed data access – Read data carrier with program no. 1**

For configuration with 8-byte buffer size!

Note
Dynamic mode is turned off while the program is running.

A total of 27 bytes are exchanged.

Command from controller		BIS M-60_8 response	
1. Process output buffer (note sequence):		2. Process input buffer: (note sequence):	
01_{hex}	Command designator 21_{hex}	$00_{hex}/07_{hex}$	Set AA bit
02_{hex}	Start address 01_{hex}	$01…06_{hex}$	Enter first 6 bytes
$00_{hex}/07_{hex}$	Set AV-Bit	$00_{hex}/0F_{hex}$	Set AE bit
3. Process input buffer:		4. Process input buffer:	
$01…06_{hex}$	Copy first 6 bytes	$01…06_{hex}$	Enter second 6 bytes
$00_{hex}/07_{hex}$	Invert TI bit	$00_{hex}/07_{hex}$	Invert TO bit

Continued until the entire data have been read.

11. Process input buffer:		12. Process input buffer:	
$01…03_{hex}$	Copy last 3 bytes	$00_{hex}/0F_{hex}$	Reset AA and AE bits
Process output buffer:			
$00_{hex}/0F_{hex}$	Reset AV bit		

9 Device Function

11th example **Mixed data access – Write to data carrier with program no. 1**

For configuration with 8-byte buffer size!

Note
Dynamic mode is turned off while the program is running.

A total of 27 bytes are exchanged.

Command from controller	BIS M-60_8 response

1. Process output buffer (note sequence):

01hex	Command designator 22hex
02hex	Program number 01hex
00hex/07hex	Set AV-Bit

2. Process input buffer: (note sequence):

00hex/07hex	Set AA bit, invert TO-Bit

3. Process output buffer:

01…06hex	Enter first 6 bytes
00hex/07hex	Invert TI bit

4. Process output buffer:

01…06hex	Copy first 6 bytes

Process input buffer:

00hex/07hex	Invert TO bit

Continued until the entire data have been written.

11. Process output buffer:

01…03hex	Enter last bytes
00hex/07hex	Invert TI bit

12. Process output buffer:

00hex/03hex	Copy last bytes

Process input buffer:

00hex/07hex	Set AE bit

13. Process output buffer:

00hex/07hex	Reset AV bit

14. Process input buffer:

00hex/07hex	Reset AA and AE bits

9 Device Function

12th example **Generate base state of Read/Write Head 1**

Both read/write heads may be placed in the base state independently of each other.

Controller **Identification System**

1. Process output buffer:

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

2. Go to base state.
Process input buffer:

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

3. Process output buffer:

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

4. Process input buffer:

$00_{hex}/07_{hex}$	Set BB bit

13th example **Turn off Read/Write head**

In normal operation both read/write heads are turned off. Setting the KA-Bit allows one or both read/write heads to be turned off.

Controller

1. Process output buffer:

$00_{hex}/07_{hex}$	Set KA-Bit

Resetting the KA-Bit turns the read/write head on again.

9 Device Function

Notes

Appendix

A.1 BIS M-6008 Ordering information

Type designation code **BIS M– 6008 –0_ _–_ _ _–06–ST_ _**

Balluff Identification System

Series M Read/Write System

Hardware type
6008 = Plastic housing

Software type
044 = PROFINET IRT
048 = PROFINET RT

Version
050 = with two connectors for external read/write heads type BIS M-3_ _

Interface
06 = Ethernet

Customer connection
ST23 = Connector types
X1 = Round connector for supply voltage (5-pin male)
X2 = Round connector for Ethernet (4-pole female, D-coded)
X3 = Round connector for Ethernet (4-pole female, D-coded)

Accessories (optional, not included)

Type		Ordering code
Connector	for X1	BKS-S 79-00
	for X2, X3	BKS-S 182-00
Cover cap	for Head 1, Head 2	Cover cap, M12 female (121 671)
	for X2, X3	BKS 12-CS-00
Adapter cable, M12 D-coded per RJ45		BIS M-526-PVC-00,5
Mounting brackets (mounting kit)	For attaching the processor to rails	BIS Z-HW-001

Appendix

A.2 BIS M-6028 Ordering information

Type designation code **BIS M – 6028 – 0_ _ – _ _ _ – 06 – ST_ _**

Balluff Identifikations-System

Series M Read/Write System

Hardware type
6028 = Metal housing

Software type
044 = PROFINET IRT
048 = PROFINET RT

Version
050 = with two connectors for external read/write heads type BIS M-3_ _

Interface
06 = Ethernet

Customer connection
ST22 = Connector types
X1 = Round connector for supply voltage (5-pin male, AIDA recommendation)
X2 = Ethernet connector (8-pole female, AIDA recommendation)
X3 = Ethernet connector (8-pole female, AIDA recommendation)
X4 = Round connector for RS232 interface (4-pin male)

Accessories (optional, not included)

Type		Ordering code
Cover cap	for Head 1, Head 2, X4 for X2, X3	Cover cap, M12 female (121 671) on request
Female	X1	5-pole female, Push-Pull Power
Male	X2, X3	8-pin male, Push-Pull RJ-45

Appendix

ASCII table

Decimal	Hex	Control Code	ASCII	Decimal	Hex	ASCII	Decimal	Hex	ASCII
0	00	Ctrl @	NUL	43	2B	+	86	56	V
1	01	Ctrl A	SOH	44	2C	,	87	57	W
2	02	Ctrl B	STX	45	2D	-	88	58	X
3	03	Ctrl C	ETX	46	2E	.	89	59	Y
4	04	Ctrl D	EOT	47	2F	/	90	5 A	Z
5	05	Ctrl E	ENQ	48	30	0	91	5B	[
6	06	Ctrl F	ACK	49	31	1	92	5C	\
7	07	Ctrl G	BEL	50	32	2	93	5D	[
8	08	Ctrl H	BS	51	33	3	94	5E	^
9	09	Ctrl I	HT	52	34	4	95	5F	_
10	0 A	Ctrl J	LF	53	35	5	96	60	`
11	0B	Ctrl K	VT	54	36	6	97	61	A
12	0C	Ctrl L	FF	55	37	7	98	62	B
13	0D	Ctrl M	CR	56	38	8	99	63	c
14	0E	Ctrl N	SO	57	39	9	100	64	d
15	0F	Ctrl O	SI	58	3 A	:	101	65	e
16	10	Ctrl P	DLE	59	3B	;	102	66	f
17	11	Ctrl Q	DC1	60	3C	<	103	67	g
18	12	Ctrl R	DC2	61	3D	=	104	68	h
19	13	Ctrl S	DC3	62	3E	>	105	69	i
20	14	Ctrl T	DC4	63	3F	?	106	6 A	j
21	15	Ctrl U	NAK	64	40	@	107	6B	k
22	16	Ctrl V	SYN	65	41	A	108	6C	L
23	17	Ctrl W	ETB	66	42	B	109	6D	m
24	18	Ctrl X	CAN	67	43	C	110	6E	n
25	19	Ctrl Y	EM	68	44	D	111	6F	o
26	1 A	Ctrl Z	SUB	69	45	E	112	70	p
27	1B	Ctrl [	ESC	70	46	F	113	71	q
28	1C	Ctrl \	FS	71	47	G	114	72	r
29	1D	Ctrl]	GS	72	48	H	115	73	s
30	1E	Ctrl ^	RS	73	49	I	116	74	t
31	1F	Ctrl _	US	74	4 A	J	117	75	u
32	20		SP	75	4B	K	118	76	V
33	21		!	76	4C	L	119	77	W
34	22		„	77	4D	M	120	78	X
35	23		#	78	4E	N	121	79	Y
36	24		$	79	4F	O	122	7 A	Z
37	25		%	80	50	P	123	7B	{
38	26		&	81	51	Q	124	7C	|
39	27		‘	82	52	R	125	7D	}
40	28		(	83	53	S	126	7E	~
41	29		)	84	54	T	127	7F	DEL
42	2 A		*	85	55	U			

Index

www.balluff.com

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Germany
Tel. +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.com
www.balluff.com

No. 866928 E . Edition 0903; Subject to modification.