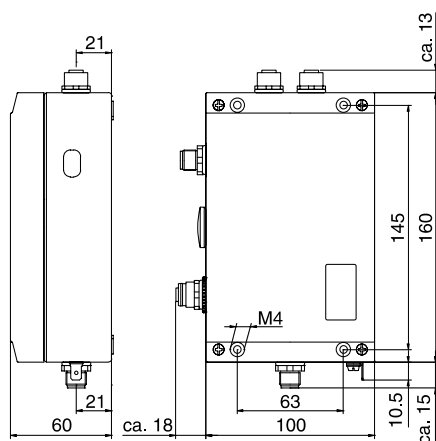


BIS M-6026 EtherNet/IP

Technical Description, User's Manual



www.balluff.com

1	User Notes	4
1.1	About this Manual	4
1.2	Manual layout	4
1.3	Conventions	4
1.4	Symbols	4
1.5	Abbreviations	5
2	Safety	6
2.1	Intended use	6
2.2	General notes on device safety	6
2.3	Meaning of the warning notes	6
3	Getting Started	7
4	Basic Knowledge	10
4.1	Identification system principles of operation	10
4.2	Product description	10
4.3	Control function	10
4.4	Data integrity	10
4.5	Bus connection	11
5	Technical Data	12
6	Installation	14
6.1	Processor installation	14
6.2	Interface information/Connection diagrams	14
6.3	Changing the EEPROM	15
7	Bus Connection	16
7.1	IP address	16
7.2	AnyBus IPconfig	16
8	Parameterizing the Processor	18
8.1	Basic knowledge	18
8.2	Parameterizing	20
9	Device Function	26
9.1	Function principle BIS M-6026	26
9.2	Function indicators	33
9.3	Examples	35
	Appendix	46

1 User Notes

1.1 About this Manual

This manual describes the processor for the BIS M-6026 identification system and guides you through startup for immediate operation.

1.2 Manual layout

The manual is designed so that each section builds on the previous sections.

chapter 2: Basic information regarding safety.

chapter 3: The main steps in installing the identification system.

chapter 4: An introduction into the material.

chapter 5: Technical data for the processor.

chapter 6: Mechanical and electrical connections.

chapter 7: Logging the processor on to the network.

chapter 8: User-defined settings for the processor.

chapter 9: How the processor and host system work.

1.3 Conventions

The following conventions are used in this manual.

Enumerations

Enumerations are represented as a list with bullet points.

- 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.
⇒ Result of action.
- ▶ Action instruction 2.

Notation

Numbers:

- Decimal numbers are represented without additional description (e.g. 123),
- hexadecimal numbers are represented by appending the abbreviation hex (e.g. 00_{hex}).

Parameters:

Parameters are written in italics, e.g. (*CRC_16*).

Directory paths:

Paths in which data are or will be saved/stored are represented in small caps (e.g. PROJECT:\DATA TYPES\USERDEFINED).

Cross-references

Cross-references indicate where additional information on the topic can be found (see [Technical Data starting page 12](#)).

1.4 Symbols



Attention!

This symbol indicates a safety advisory which must be observed.



Note, tip

This symbol indicates general notes.

1 User Notes

1.5 Abbreviations

ARP	Address Resolution Protocol
BIS	Balluff Identification System
CIP	Common Industrial Protocol
CRC	Cyclic Redundancy Code
DHCP	Dynamic Host Configuration Protocol
EDS	Electronic Data Sheet
EEPROM	Electrical Erasable and Programmable ROM
EMV	Electromagnetic Compatibility
MAC-ID	Media Access Control Identifier
ODVA	Open DeviceNet Vendor Association
PC	Personal Computer
RPI	Requested Packed Interval
PLC	Programmable Logic Controller

2 Safety

2.1 Intended use

The BIS M-6026 processor is a component of the BIS M identification system. Within the identification system it is used for linking to a host computer (PLC, PC).
It is intended only for use only in this way and in an industrial environment complying with Class A of the EMC Law.
This description applies to processors in series BIS M-6026-034-....

2.2 General notes on device safety

Installation and startup

Installation and startup are to be carried out only by trained specialists. The manufacturer revokes the right to any warranty or liability claims resulting from unauthorized modifications or improper use. When connecting the processor to an external controller, be sure to observe proper polarity for all connections including the power supply (see section "Installation" on page 14). The processor must be operated only using approved power supplies (see section "Technical Data" on page 12).

Operation and testing

It is the responsibility of the operator to ensure that the locally applicable safety regulations are maintained.

In case of defects and faults in the identification system which cannot be remedied, take it out of operation and protect against unauthorized use.

2.3 Meaning of the warning notes



Attention!

The pictogram used with the word "Attention" warns of a possibly hazardous situation for the health of persons or equipment damage.

Disregarding these warnings may result in personal injury or equipment damage.
► Always observe the instructions given for avoiding this hazard.

Getting Started

Mechanical connection

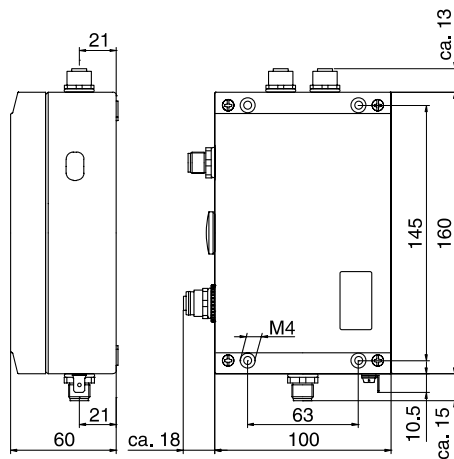


Fig. 1: Mechanical connection

- Attach processor using (4) M4 screws.

Electrical connection



Note

Route the ground wire to ground either directly or through an RC combination, depending on the system.
When connecting to the Ethernet, be sure that the connector shield is perfectly connected to the connector body.
Do not alter the factory setting for the DIL switches.

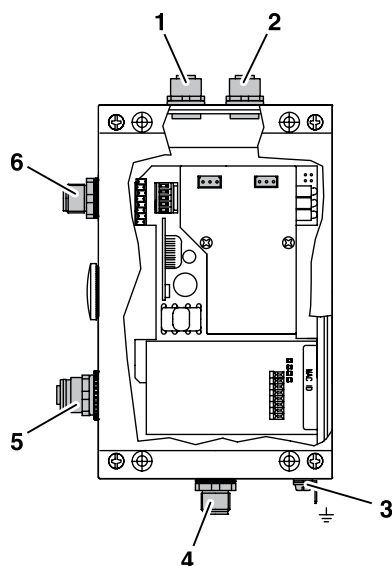


Fig. 2: Electrical connection

- | | |
|------------------------------|---------------------|
| 1 Head 2 - Read/write head 2 | 4 X4 - Service port |
| 2 Head 1 - Read/write head 1 | 5 X3 - Ethernet |
| 3 Function ground FE | 6 X1 - Power supply |

3 Getting Started

X1 - Power supply

PIN	Function
1	+Vs
2	
3	-Vs
4	
5	

X3 - Ethernet

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

X4 - Service port

PIN	Function
1	
2	TxD
3	GND
4	RxD

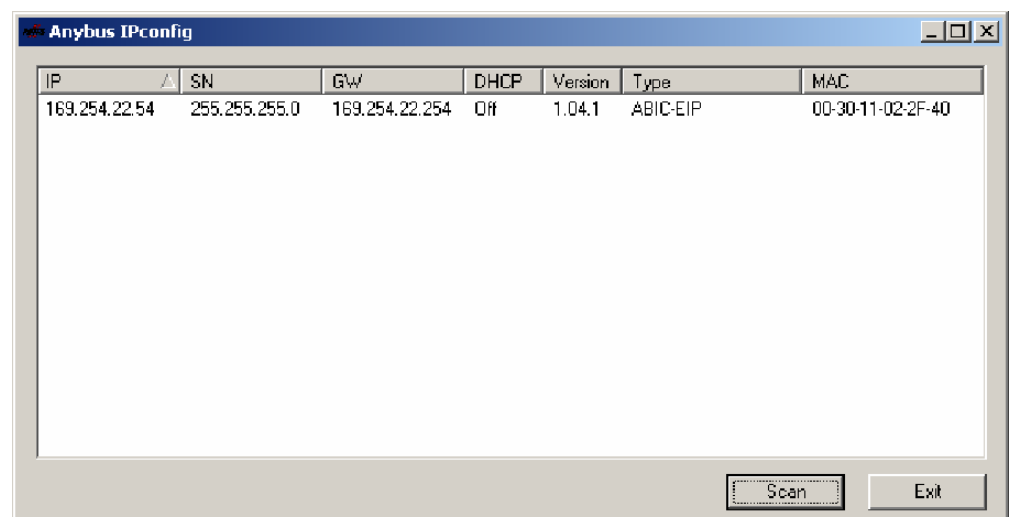
Bus connection

There are two ways of making the bus connection:

- Setting the IP address, subnet mask, gateway address and selecting DHCP using the application "Anybus IPconfig" on a Windows PC having an Ethernet network card.
- Using a DHCP server or BOOTP software.

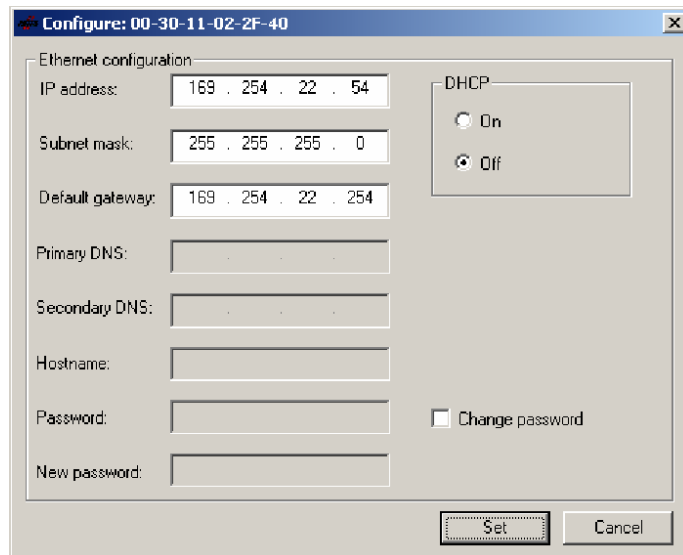
Use the "Anybus IPconfig" program to make the bus connection. The "Anybus IPconfig" application is included on the BIS SD which comes with the processor.

- ▶ Start "Anybus IPconfig".
 - ⇒ The subnet is scanned for a connected BIS _6026. The result of the scan is displayed in the "Anybus IPconfig" window.



- ▶ Select the device from the scan list and double-click on it.
 - ⇒ The "Configure" window is opened.

Getting Started



The screenshot shows a Windows-style dialog box titled "Configure: 00-30-11-02-2F-40". It contains an "Ethernet configuration" section with the following fields and controls:

- IP address:** A text box containing "169 . 254 . 22 . 54".
- Subnet mask:** A text box containing "255 . 255 . 255 . 0".
- Default gateway:** A text box containing "169 . 254 . 22 . 254".
- Primary DNS:** An empty text box.
- Secondary DNS:** An empty text box.
- Hostname:** An empty text box.
- Password:** An empty text box.
- New password:** An empty text box.
- DHCP:** A group box containing two radio buttons: "On" (unselected) and "Off" (selected).
- Change password:** A checkbox that is currently unchecked.

At the bottom right of the dialog are two buttons: "Set" and "Cancel".

- ▶ Assign the IP address, subnet mask and gateway address.
- ▶ Turn DHCP on/off.
- ▶ Confirm your settings by clicking on Set.

4 Basic Knowledge

4.1 Identification system principles of operation

The BIS M identification system belongs to the category of non-contact systems having a read and write function. This enables you to not only read data contained in the data carriers, but also to write new data to them at any point in the process.

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. for variant-specific processes, workpiece transport using conveying systems, for collecting safety-related data),
- in inventory systems for monitoring inventory movements,
- in transport and conveying technology.

4.2 Product description

BIS M-6026 processor:

- Metal enclosure,
- round connectors for making plug connections,
- capacity for two read/write heads,
- read/write heads are suitable for both dynamic and static operation,
- processor provides power for system components,
- Carrier signal from the read/write heads provides power for the data carrier.

4.3 Control function

The processor represents the link between the data carrier and the host control system. It manages two-way data transfer between the data carrier and read/write head and provides a buffer storage function. The processor writes data from the host signal to the data carrier through the read/write head, or reads data from the data carrier and makes the data available to the host system.

Host systems may be:

- A control computer (e.g. industrial PC),
- a PLC.

Dual bit-header:

In order to ensure consistency of the transmitted data, the control bits in the first and last byte (bit-header) of the data buffer for each read/write head are sent and compared. If the two bit-headers are identical, then the data were fully updated and may be accepted. This means that the data for each read/write head are only valid if both bit-headers are identical. The host system must therefore also compare the bits in the bit-headers.

4.4 Data integrity

To ensure data integrity the data transfer between data carrier and processor must be monitored using a check procedure. The factory default setting in the processor is for double read with compare. A CRC_16 check may however be selected as an alternative. In CRC_16 checking a check code is written to the data carrier, which enables checking the data for validity at any time. Which procedure is used depends on how you are using the identification system.



Note

Mixed operation of the two check procedures is not possible!

4 Basic Knowledge

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

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

4.5 Bus connection

The processor and host system communicate using EtherNet/IP protocol.

EtherNet/IP is an industrial network standard. The IP in EtherNet/IP stands for "Industrial Protocol". EtherNet/IP uses the open communications protocol "Common Industrial Protocol" (CIP) on the application layer (as per the ISO/OSI reference model). EtherNet/IP is supported by the network organization "Open DeviceNet Vendor Association".

Use of a switch in full-duplex mode is necessary for collision-free data exchange.

5 Technical Data

Dimensions

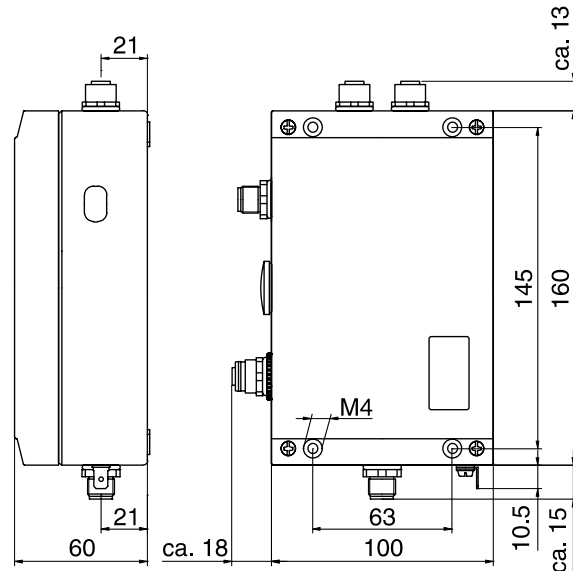


Fig. 3: Dimensions in mm

Mechanical data

Housing material	EN AC-AISi12 (a), DIN EN 1706
X1 – Input	V _s 24 V DC – 5-pin terminal
X3 – Ethernet	M12 – 4-pin socket, D-coded
X4 – Service port	RS 232 – 4-pin terminal
Head 1, 2 (Write/read head connections)	8-pin socket
Enclosure rating	IP65 (with plugs connected)
Weight	950 g

Electrical data

Operating voltage V _s	24 V DC ±10 %
Ripple	≤ 10 %
Current consumption	≤ 400 mA
Device interface	Ethernet
Service port	RS 232

5 Technical Data

Operating conditions

Ambient temperature	0 °C ... 60 °C
EMV	– EN 61000-4-2/3/4/5/6 – EN 55011
Shock/Vibration	EN 60068 Part 2-6/27/29/64/32

Function indicators

BIS operating states	Ready CT1 Present/Operating CT2 Present/Operating	Green LED Green/yellow LED Green/yellow LED
Status EtherNet/IP	Data Rate (DR) Module Status (MS) Network Status (NS) Link/Activity (L/A)	LED LED LED LED

6 Installation

6.1 Processor installation

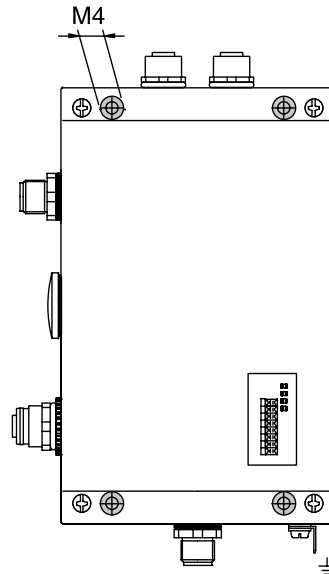


Fig. 4: Installation

- Attach processor using (4) M4 screws.

6.2 Interface information/ Connection diagrams



Note

Route the ground wire to ground either directly or through an RC combination, depending on the system.
When connecting to the Ethernet, be sure that the connector shield is perfectly connected to the connector body.
Do not alter the factory setting for the DIP switches.

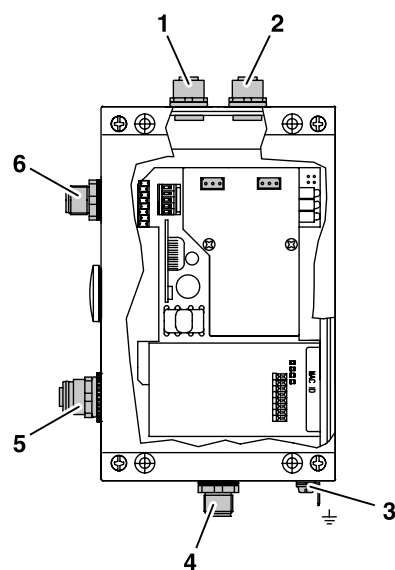


Fig. 5: Processor connections

- | | |
|------------------------------|---------------------|
| 1 Head 2 - Read/write head 2 | 4 X4 - Service port |
| 2 Head 1 - Read/write head 1 | 5 X3 - Ethernet |
| 3 Function ground FE | 6 X1 - Power supply |

6 Installation

X1 - Power supply

PIN	Function
1	+Vs
2	
3	-Vs
4	
5	

X3 - Ethernet

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

X4 - Service port

PIN	Function
1	
2	TxD
3	GND
4	RxD

6.3 Changing the EEPROM



Attention!

Components may be damaged by electrostatic discharge.

- Be sure to turn off power to the device before opening it.

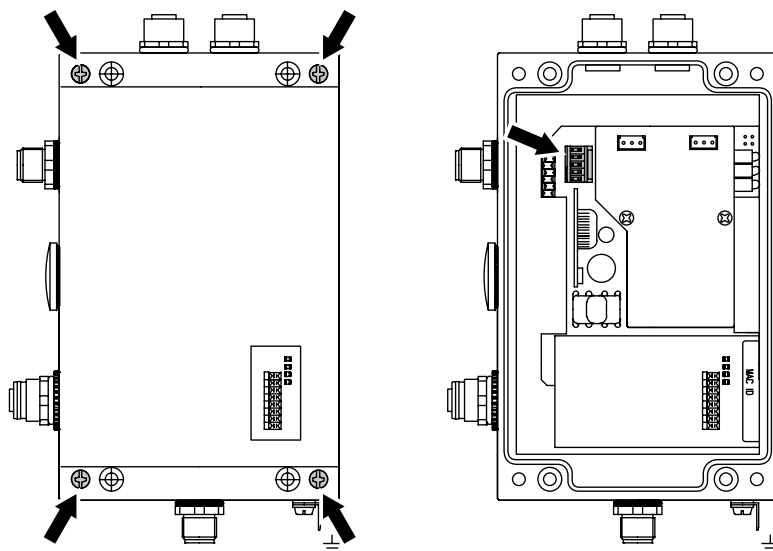


Fig. 6: Changing the EEPROM

- Remove 4 screws from housing cover and remove cover.
- Pull EEPROM from socket.
- Insert new EEPROM into socket.
- Replace cover and tighten 4 screws.

7 Bus Connection

7.1 IP address

The processor and host system communicate using EtherNet/IP protocol. Assigning a unique IP address associates the processor with a network.

The processor can be incorporated into a network in various ways (DHCP, ARP). The MAC address is used as the basis for incorporating into the network. This hardware address is used only one time and uniquely identifies network devices such as the processor.

DHCP

Dynamic Host Configuration Protocol (DHCP) allows a server to be used for dynamic assigning of an IP address. The hardware can be inserted into the network without having to perform any additional configuration. Only automatic obtaining (MAC address) of the IP address needs to be set.

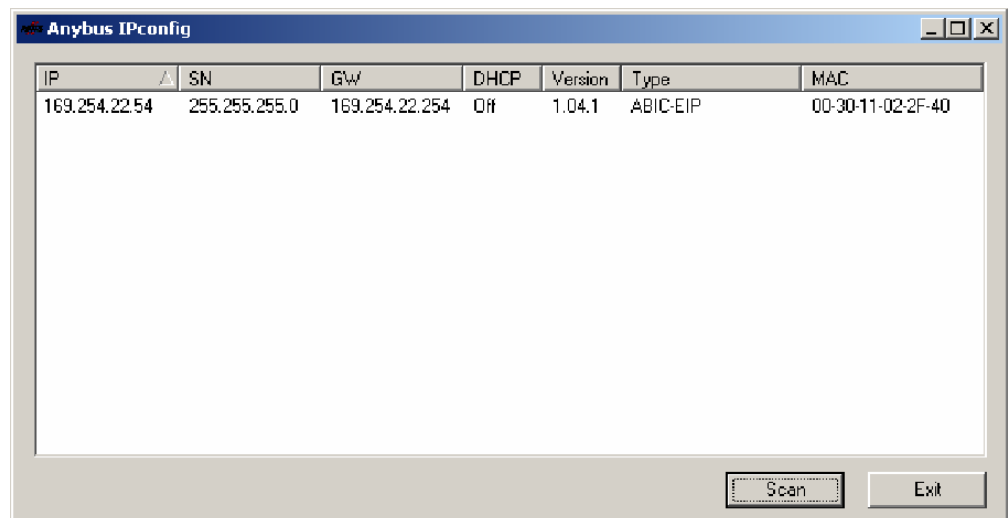
7.2 AnyBus IPconfig

"AnyBus IPconfig" is software which allows the hardware to be addressed for the corresponding subnet prior to installation. In addition, assigning of the IP address through a DHCP server or BOOTP program can be activated (DHCP on) or deactivated (DHCP off).

The "Anybus IPconfig" application is included on the BIS SD which comes with the processor.

Start "Anybus IPconfig".

The subnet is scanned for a connected BIS _-6026. The result of the scan is displayed in the "Anybus IPconfig" window.



- Select the device from the scan list and double-click on it.
⇒ The "Configure" window is opened.

7 Bus Connection

The screenshot shows a Windows-style dialog box titled "Configure: 00-30-11-02-2F-40". It contains an "Ethernet configuration" section with the following fields and controls:

- IP address:** 169 . 254 . 22 . 54
- Subnet mask:** 255 . 255 . 255 . 0
- Default gateway:** 169 . 254 . 22 . 254
- Primary DNS:** (empty field)
- Secondary DNS:** (empty field)
- Hostname:** (empty field)
- Password:** (empty field)
- New password:** (empty field)
- DHCP:** Radio buttons for "On" and "Off". "Off" is selected.
- Change password:** A checkbox that is currently unchecked.
- Buttons:** "Set" and "Cancel" buttons at the bottom right.

- ▶ Assign the IP address, subnet mask and gateway address.
- ▶ Turn DHCP on/off.
- ▶ Confirm your settings by clicking on Set.

8 Parameterizing the Processor

8.1 Basic knowledge

Data carrier type For the BIS M-6026 processor the following data carrier types are available.

Mifare:

Balluff data carrier type	Manufacturer	Name	Memory capacity	Memory type
BIS M-1_-01	Philips	Mifare Classic	752 Byte	EEPROM

ISO15693:

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

* On request



Note

The data carriers contains additional memory ranges for configuration and protected data. These ranges cannot be processed using the BIS M-6026 processor.

CRC check

The CRC check is a procedure for determining a test value for data so as to detect errors in transferring data. If CRC check is activated, an error message is output when a CRC error is detected.

Initializing

To be able to use the CRC check, the data carriers must be initialized. The data carriers are initialized in the output buffer using the command 12_{hex}. If the data carrier does not contain the correct CRC, then the processor sets an error message in the input buffer (see [Example 10 on page 44](#)).

As shipped from the factory, data carriers may be immediately written a checksum, since all the data are set to 0.

Error message

- If an error message is the result of a failed write job, then the data carrier must be reinitialized before it can be used again.
- If an error message is not the result of a failed write job, then one or more of the memory cells in the data carrier are defective. This means the data carrier must be replaced.

Checksum

The checksum is written to the data carrier as a 2-byte information. 2 bytes per block are lost. This leaves 14 bytes remaining per block. The usable number of bytes can be determined from the following table.

8 Parameterizing the Processor

Data carrier	Balluff data carrier type	Memory capacity	Usable bytes for CRC_16
Mifare	BIS M-1__-01	752 Byte	658 Byte
ISO15693	BIS M-1__-02	2000 Byte	1750 Byte
	BIS M-1__-03	112 Byte	98 Byte
	BIS M-1__-04	256 Byte	224 Byte
	BIS M-1__-05	224 Byte	196 Byte
	BIS M-1__-06	288 Byte	252 Byte
	BIS M-1__-07	992 Byte	868 Byte

Simultaneous data transmission

Reading

The processor reads the data from the data carrier directly into the input buffer. As soon as the buffer is filled, the Toggle-Bit Out (TO-Bit) is inverted to indicate data ready to the host system. The system inverts the Toggle-Bit In (TI-Bit) to indicate that it is ready to receive, and data read in the meantime are transmitted to the input buffer. This repeats itself until the desired data have been read from the data carrier. After the read procedure is finished, the processor sets the Job End bit (AE-bit) and transmits the remaining data to the input buffer ([see example 2 on page 36](#)).

Writing

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

Dynamic mode

As soon as the Dynamic function is activated, the processor accepts the read/write job from the host system and stores it regardless of whether there is a data carrier in the active zone of the read/write head. When a data carrier enters the active zone of the read/write head, the stored job is carried out.

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} to be read faster.

If the parameter *TypSN* (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 *Extra*

If Auto-Read *Extra* is activated, then the 14 bytes starting at a specified address are read from the data carrier to the input buffer and then the Codetag Present bit (CP bit) is set. The start address is specified using the parameter *Extra_Adr*.

Type and serial number

If this function is activated, with CT Present the data carrier type and serial number (UID = unique ID) of the data carrier are issued. In a BIS M-1__-01 data carrier the serial number takes up 4 bytes. In all other data carrier types the serial number is 8 bytes.

8 Parameterizing the Processor

8.2 Parameterizing

There are two different ways to set parameters. Parameterizing from an application program or using the EDS file.

Basics

The parameters for operating the processor are stored in the BIS Sonfig Object (class 64hex). Explicit messages are used to access the parameters.

Parameterizing from an application program

One widely used application for EtherNet/IP device parameterizing is the Windows program RSLogix 5000 written for the Logix 5000 controller of Rockwell Automation. An example for device programming is included on the BIS-CD. For additional information [see section "Example for parameterizing with application program" on page 23](#).

EDS file

The EDS file contains all the device parameters for the processor. The file is included on the BIS-CD.

Parameters

CRC_16 class: 64hex
 instance: 01hex
 attribute: 01hex

Factory setting: Disable (= 0)

Data validity is checked using double read.

Other settings: Enable (= 1)

Data validity is ensured using CRC check.

Simultaneous class: 64hex
 instance: 01hex
 attribute: 02hex

Factory setting: Disable (= 0)

Read/write jobs and data transmission are run in sequence.

Other settings: Enable (= 1)

Read/write jobs and data transmission are run simultaneously.

Dynamic1 class: 64hex
 instance: 01hex
 attribute: 03hex

Factory setting: Disable (= 0)

Read/write head 1 is in static mode. Read/write command from the controller is carried out only if there is a data carrier in the active zone of Read/Write Head 1.

Other settings: Enable (= 1)

Read/Write Head 1 is in dynamic mode.

8 Parameterizing the Processor

Dynamic2 class: 64_{hex}
 instance: 01_{hex}
 attribute: 04_{hex}

Factory setting: Disable (= 0)

Read/write head 2 is in static mode. Read/write command from the controller is carried out only if there is a data carrier in the active zone of Read/Write Head 2.

Other settings: Enable (= 1)

Read/Write Head 2 is in dynamic mode.

Extra1 class: 64_{hex}
 instance: 01_{hex}
 attribute: 05_{hex}

Factory setting: Disable (= 0)

CT Present data if there is a data carrier in the active zone of Read/Write Head 1.

Other settings: Enable (= 1)

The Auto-Read function is active.

Extra2 class: 64_{hex}
 instance: 01_{hex}
 attribute: 06_{hex}

Factory setting: Disable (= 0)

CT Present data if there is a data carrier in the active zone of Read/Write Head 2.

Other settings: Enable (= 1)

The Auto-Read function is active.

Extra_Adr1 class: 64_{hex}
 instance: 01_{hex}
 attribute: 07_{hex}

Factory setting: 0

Other settings: 1...1999

Specifies the start address (Auto-Read) beginning at which the data carrier is read when a data carrier enters the active zone of Read/Write Head 1.

8 Parameterizing the Processor

Extra_Adr2 class: 64_{hex}
 instance: 01_{hex}
 attribute: 08_{hex}

Factory setting: 0

Other settings: 1...1999

Specifies the start address (Auto-Read) beginning at which the data carrier is read when a data carrier enters the active zone of Read/Write Head 2.

TypSN class: 64_{hex}
 instance: 01_{hex}
 attribute: 09_{hex}

Factory setting: Disable (= 0)

At CT Present the data carrier type and the serial number of the data carrier are output.

Other settings: Enable (= 1)

DTTyp class: 64_{hex}
 instance: 01_{hex}
 attribute: 0A_{hex}

Factory setting: All data carrier types (= 0)

Depending on the selection either all or only certain data carriers can be processed.

Other settings: Mifare (= FE_{hex})
 ISO15693 (= FF_{hex})

Mifare: All Mifare data carriers **supported by Balluff** are processed.

ISO15693: All ISO15693 data carriers **supported by Balluff** are processed.

8 Parameterizing the Processor

Example of parameterizing using the application program

This example shows how the example project included on the BIS-CD can be used with the RSLogix 5000 software for a user project.

Note the following procedure:

1. Add the BIS M-6026 to a user project.
2. Import the example project into a new project.
3. Copy user-defined data type from the example project to the user project.
4. Create a sub-routine in the user project.
5. Set invoking of the sub-routine in the main program of the user program.

To run the example the files stored on the BIS-CD must be copied to a local directory.

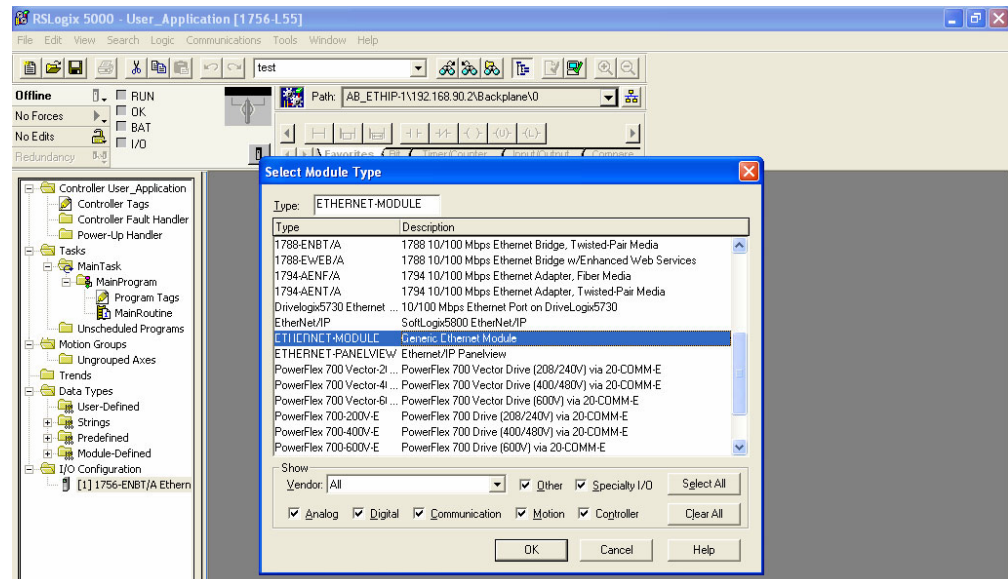


Note

Information about the software, installation, creating projects and working in projects can be found in the manual for the RSLogix 5000 manual.

1. Add the processor

- Open user project.
- Under I/O-CONFIGURATION\1] 1756-ENBT/A ETHERNET/IP create a new module.
(type: Generic Ethernet Module).



8 Parameterizing the Processor

- Set module properties:

Name:	e.g. BIS_M
Communication format:	Data SINT
IP address:	e.g. 192.168.90.3

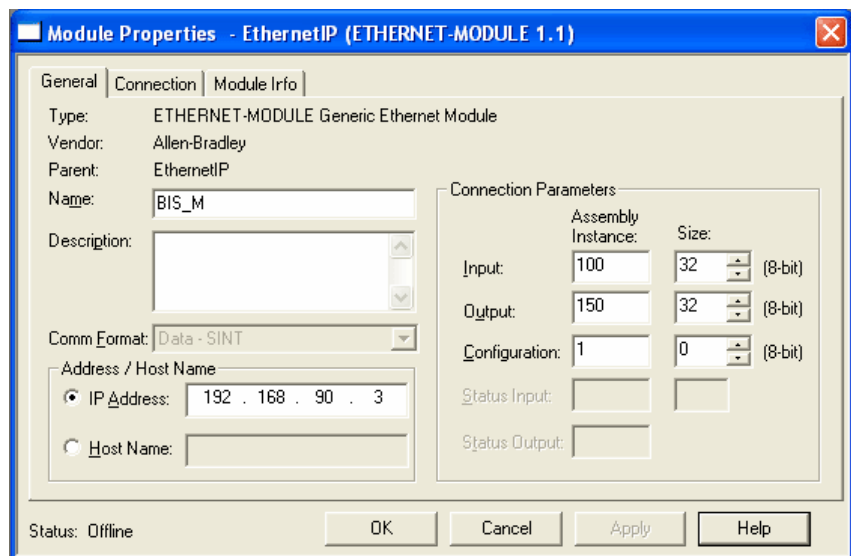
- Set the connection parameters as follows:

Connection parameters		Instance	Size
Input:		100	32 bytes
Output:		150	32 byte
Configuration:		1	0



Note

"Configuration" is not supported. Therefore the values are set to 1 and 0.



- Save settings by clicking on "OK" and confirm the remaining dialog fields until the module has been successfully created.



Note

When confirming the dialog fields, be sure that the Requested Packed Interval (RPI) ≥ 10 ms is set.

2. Import example project



Note

Only one project per window can be opened in RSLogix 5000.

- Open a new project.
- Import the example project "Example_Project_Param_BIS_M.L5K" from the local directory to the project (File\Open).
- Save example project in *.ACD format (File\Save as) - File name is freely selectable.

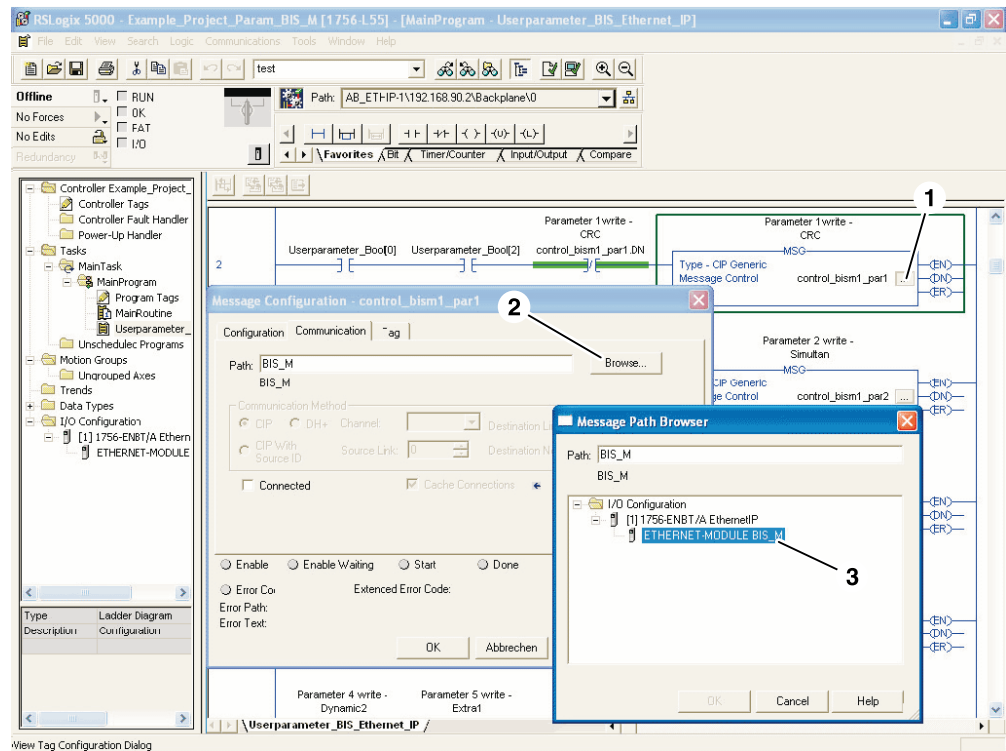
8 Parameterizing the Processor

3. Copy user-defined data type

- Under DATA TYPES\USER DEFINED in the example project copy "BIS_M_Userparameters".
- Under DATA TYPES\USER DEFINED in the example project paste "BIS_M_Userparameters".

4. Create a sub-routine in the user project

- Under TASKS\MAINTASK\MAINPROGRAM in the user project create a new routing with the name "Userparameters_BIS_Ethernet_IP".
- Double-click to open the new routine.
- Right-click on "Import flow path" from the context menu.
- Import the file "Example_Project_Rung_BIS_M.L5X" from the local directory to the user project.
- Reconfigure the communication paths for all messages – see screenshot for sequence.



5. Set invoking of the sub-routine

- Under TASKS\MAINTASK\MAINPROGRAM select MainRoutine.
- In the MainRoutine set "Userparameter_Bool (0)" to high.
⇒ Sub-routine is activated.

9 Device Function

9.1 Function principle BIS M-6026

Two buffers are required to exchange data and commands between the processor and the host system. Cyclical polling is used for exchanging the buffer contents. The buffer content depends on the cycle in which it is written (e.g. control commands at the start of the job). When writing the buffer, the transmitted data from the previous cycle are overwritten. Unwritten bytes are not deleted and retain their data content.

The buffer size of the overall buffer is 32 bytes. 16 bytes are available for each read/write head.

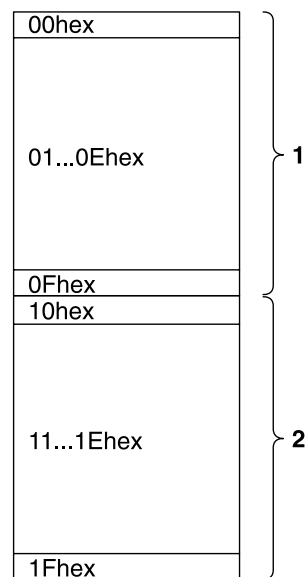


Fig. 7: Total buffer Read/Write Heads 1 and 2

1 Read/Write Head 1

2 Read/Write Head 2

Only 14 bytes per read/write head are available for data exchange, since the first and last bytes in the respective data buffer are used for control and for status messages.

Output buffer

The output buffer is used to transmit the identification system commands and the data to be written to the data carrier.

Bit no. Subaddress	7	6	5	4	3	2	1	0
00 _{hex} = Bit header		TI	KA			GR		AV
01 _{hex}	Command					or Data		
02 _{hex}	Stared address (Low Byte) or program no.					or Data		
03 _{hex}	Start address (High Byte)					or Data		
04 _{hex}	No. of bytes (Low Byte)					or Data		
05 _{hex}	No. of bytes (High Byte)					or Data		
06 _{hex}	Data							
...	Data							
0F _{hex} = Bit header		TI	KA			GR		AV

9 Device Function

Allocation and explanation

Subaddress	Bit name	Meaning	Function description
00 _{hex} /0F _{hex}	TI	Toggle-Bit In	Controller is ready to receive additional data (read job).
	KA	Head deselect	Turns the read/write head off.
	GR	Base state	Identification goes into base state for the respective read/write head. Any pending job is cancelled.
	AV	Job	A job is pending for the respective read/write head.

Subaddress	Meaning	Function description
01 _{hex}	Command	
	00 _{hex}	No command.
	01 _{hex}	Read data carrier.
	02 _{hex}	Write to data carrier.
	06 _{hex}	Save program for "Mixed data access" in EEPROM.
	12 _{hex}	Initialize CRC_16 data check.
	21 _{hex}	Read data carrier as per a program for "Mixed data access".
	22 _{hex}	Write to data carrier as per a program for "Mixed data access".
	or Data	Transmitting data written to the data carrier.
	or Program data	Transmitting program data written to the EEPROM.

02 _{hex}	Start address (Low Byte)	Address starting at which reading or writing should commence (address range from 0 to 255 is covered).
	or Program No.	Program No. to be stored for "Mixed Data Access" in conjunction with command 06 _{hex} (value range 01 _{hex} to 0A _{hex}).
	or Program No.	Program No. to be run for "Mixed Data Access" in conjunction with command 21 _{hex} or 22 _{hex} (value range 01 _{hex} to 0A _{hex}).
	or Data	Transmitting data written to the data carrier.
	or Program data	Transmitting program data written to the EEPROM.

03 _{hex}	Start address (High Byte)	Address starting at which reading or writing should commence (address range from 256 to 1999 is covered).
	or Data	Transmitting data written to the data carrier.
	or Program data	Transmitting program data written to the EEPROM.

9 Device Function

Subaddress	Meaning	Function description
04 _{hex}	No. of bytes (Low Byte)	No. of bytes (1 to 255 bytes) to read or write beginning at the start address (Low Byte).
	or Data	Transmit the data which are written to the data carrier.
	or Program data	Transmitting program data written to the EEPROM.
05 _{hex}	No. of bytes (High Byte)	No. of bytes (256 to 2000 bytes) to read or write commencing with the start address (High Byte).
	or Data	Transmitting data written to the data carrier.
	or Program data	Transmitting program data written to the EEPROM.
06 _{hex}	Data	Transmitting the data written to the data carrier.
	or Program data	Transmit the program data which are written to the EEPROM.
...	Data	Transmitting the data written to the data carrier.
	or Program data	Transmit the program data which are written to the EEPROM.

Input buffer

The input buffer is used to transmit the data read by the identification system, the identifiers and 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							
0F _{hex} = Bit header	BB	HF	TO		AF	AE	AA	CP

9 Device Function

Allocation and explanation

Subaddress	Bit name	Meaning	Function description
00 _{hex} /0F _{hex}	BB	Ready	Identification system is ready.
	HF	Head error	Cable break on read/write head or no read/write head connected.
	TO	Toggle-Bit Out	Read procedure: Identification system has additional data ready. Write procedure: Identification system can accept additional data.
	AF	Job error	Error in processing the job, or job cancelled.
	AE	Job end	Confirmation – Job ended without error.
	AA	Job start	Confirmation – Job was recognized and started.
	CP	Codetag Present	There is a data carrier in the active zone of the read/write head.

Subaddress	Meaning	Function description
01 _{hex}	Error code	Error number valid only with AF-bit!
	01 _{hex}	Job cannot be carried out because there is no data carrier in the active zone of the read/write head.
	02 _{hex}	Read error.
	03 _{hex}	Data carrier was removed from the active zone of the read/write head during reading.
	04 _{hex}	Write error.
	05 _{hex}	Data carrier was removed from the active zone of the read/write head during writing.
	06 _{hex}	Memory access error.
	07 _{hex}	Invalid or no command for set AV-bit or number of bytes is 00 _{hex} .
	09 _{hex}	Cable break on read/write head or no read/write head connected.
	0C _{hex}	EEPROM cannot be read or programmed.
	0D _{hex}	Communication fault with the read/write head.
	0E _{hex}	CRC for read data and CRC for data carrier do not agree.
	0F _{hex}	1 st and 2 nd bit header are not identical. The 2 nd bit header must be operated.
	20 _{hex}	Addressing of the read/write job is outside the memory range of the data carrier.
	21 _{hex}	This function is not possible for this data carrier.
	or Data	Transmit data which were read from the data carrier.
02 _{hex}	Data	Transmit data which were read from the data carrier.
...	Data	Transmit data which were read from the data carrier.

9 Device Function

Communication

Communication between the host system and the processor is defined by a sequence protocol. A control bit in the output and input buffer is used to implement communication between the host system and the processor.

Basic sequence

1. Controller sends command in output buffer to processor with set AV bit. The AV bit tells the processor that a job is beginning and the transmitted data are valid.
2. Processor takes 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 exchange is indicated by inverting the Toggle-Bits TI and TO.
4. The processor has correctly executed the job and set the AE bit in the input buffer.
5. The controller has received all data. The AV bit in the output buffer is reset.
6. The processor resets all the control bits (AA, AE) in the input buffer which were set during the job. The processor is ready for the next job.

Mixed data access

By carrying out read/write programs it is possible to write data to various address ranges on the data carrier or read data which are contained in various 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 information about the start address and number of bytes. The maximum allowable amount of data that can be transmitted is 2 kB.

Saving programs:

Command 06_{hex} in the output buffer sends the program to the processor. Saving a program is considered a job. All 25 instructions and two additional bytes with FF_{hex}FF_{hex} as an end delimiter must always be transmitted. This means that 104 bytes per program, including the command and program number, are transmitted ([see example 7 on page 41](#)).

9 Device Function

Program structure example:

Program structure	Subaddress	Value	Value range
Command	01 _{hex}	06 _{hex}	
1. Program set			
Program number	02 _{hex}	01 _{hex}	01 _{hex} to 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}		
End delimiter	FF _{hex} FF _{hex}		

Running programs:

The programs stores in the EEPROM may be used for reading data records from data carriers as well as for writing data records to a data carrier. The command 21_{hex} (read) or 22_{hex} (write) in the output buffer is used to specify reading or writing (see [example 8 on page 42](#) and [example 9 on page 43](#)).

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 BIS M-1__-02 to BIS M-1__-07 data carriers on request.

9 Device Function

9.2 Function indicators

Overview of indicators

The operating states of the identification system, the Ethernet connection and the EtherNet/IP connection are indicated by means of LED's.

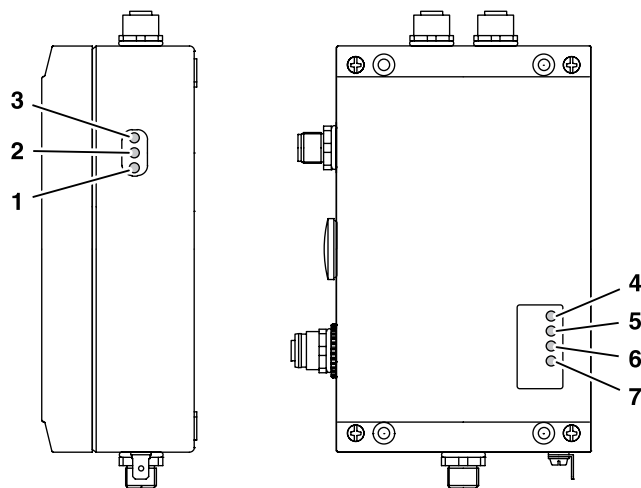


Fig. 8: Function indicators

Identification system

- 1 CT2 Present/Operating
- 2 CT1 Present/Operating
- 3 Ready

Ethernet and EtherNet/IP

- 4 Data Rate (DR)
- 5 Module Status (MS)
- 6 Network Status (NS)
- 7 Link/Activity (L/A)

Power-up

During power-up all LED's for the Ethernet and EtherNet/IP connection are tested as described in the following table.

LED name	LED sequence			
Data Rate (DR)	off			green red
Module Status (MS)	green	red	green	
Network Status (NS)	off		green red	off
Link/Activity (L/A)	off		green red	off

Diagnostics

Identification system

Status LED	Meaning
Ready	
green	Operating voltage present; no hardware error

CT1 Present/Operating	
green	Data carrier ready to read/write at Read/Write Head 1
yellow	Read/write job being processed at Read/Write Head 1
yellow flashing	Cable break on Read/Write Head 1 or Read/Write Head 1 not connected
off	No data carrier in the active zone of the Read/Write Head 1

9 Device Function

Status LED	Meaning
CT2 Present/Operating	
green	Data carrier ready to read/write at Read/Write Head 2
yellow	Read/write job being processed at Read/Write Head 2
yellow flashing	Cable break on Read/Write Head 2 or Read/Write Head 2 not connected
off	No data carrier in the active zone of the Read/Write Head 2

Ethernet- and EtherNet/IP connection

Status LED	Meaning
Data Rate	
off	Transmission rate 10 Mbit
green	Transmission rate 100 Mbit
red	-

Module Status	
off	No power to module
green	Device ready
green flashing	Module configuration missing or incorrect
red	Non-clearable error
red flashing	Clearable error

Network Status	
off	No voltage or no IP address
green	Device has at least one EtherNet/IP connection
green flashing	Device has no EtherNet/IP connection
red	An IP address is duplicated
red flashing	One or more EtherNet/IP connections has timed out

Link/Activity	
off	No power
green	Device is connected to Ethernet
green flashing	RX/TX activity
red	-

9 Device Function

9.3 Examples

1. Read 30 bytes on Head 1, start address 10

Controller

1. Process output buffer
(note sequence):

01 _{hex}	Command 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} / 0F _{hex}	Set AV bit

3. Process input buffer:

01...0E _{hex}	Copy first 14 bytes
------------------------	---------------------

Process output buffer:

00 _{hex} / 0F _{hex}	Invert TI-Bit
---------------------------------------	---------------

5. Process input buffer:

01...0E _{hex}	Copy second 14 bytes
------------------------	----------------------

Process output buffer:

00 _{hex} / 0F _{hex}	Invert TI-Bit
---------------------------------------	---------------

7. Process input buffer:

01...02 _{hex}	Copy last bytes
------------------------	-----------------

Process output buffer:

00 _{hex} / 0F _{hex}	Reset AV-Bit
---------------------------------------	--------------

Identification system

2. Process input buffer
(note sequence):

00 _{hex} / 0F _{hex}	Set AA bit
01...0E _{hex}	Enter first 14 bytes
00 _{hex} / 0F _{hex}	Set AE-Bit

4. Process input buffer:

01...0E _{hex}	Enter second 14 bytes
00 _{hex} / 0F _{hex}	Invert TO-Bit

6. Process input buffer:

01...02 _{hex}	Enter last bytes
00 _{hex} / 0F _{hex}	Invert TO-Bit

8. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA and AE-Bit
---------------------------------------	---------------------

9 Device Function

2. Read 30 bytes on Head 1, start address 10, simultaneous data transmission

While the read job is being carried out and as soon as sufficient data have been read for filling the input buffer of Read/Write Head 1, these data are sent to the input buffer. The AE bit is not set until the "Read" operation has been finished by the processor.

The reply "Job End" (AE-Bit) is reliably set no later than when the last data are sent. The actual time depends on the requested amount of data and the time response of the controller. In the example the italicized "*Set AE-Bit*" draws your attention to this fact.

Controller

Identification system

1. Process output buffer
(note sequence):

01 _{hex}	Command 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} / 0F _{hex}	Set AV bit

2. Process input buffer
(note sequence):

00 _{hex} / 0F _{hex}	Set AA bit
01...0E _{hex}	Enter first 14 bytes
00 _{hex} / 0F _{hex}	Invert TO-Bit
00 _{hex} / 0F _{hex}	<i>Set AE-Bit</i>

3. Process input buffer:

01...0E _{hex}	Copy first 14 bytes
Process output buffer:	
00 _{hex} / 0F _{hex}	Invert TI-Bit

4. Process input buffer:

01...0E _{hex}	Enter second 14 bytes
00 _{hex} / 0F _{hex}	Invert TO-Bit
00 _{hex} / 0F _{hex}	<i>Set AE-Bit</i>

5. Process input buffer:

01...0E _{hex}	Copy second 14 bytes
Process output buffer:	
00 _{hex} / 0F _{hex}	Invert TI-Bit

6. Process input buffer:

01...02 _{hex}	Enter last bytes
00 _{hex} / 0F _{hex}	Invert TO-Bit
00 _{hex} / 0F _{hex}	<i>Set AE-Bit</i>

7. Process input buffer:

01...02 _{hex}	Copy last bytes
Process output buffer:	
00 _{hex} / 0F _{hex}	Reset AV-Bit

8. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA and AE-Bit
---------------------------------------	---------------------

9 Device Function

3. Read 30 bytes on Head 1, start address 10, with read error



Note

If an error occurs, the AF-Bit with corresponding error number is provided instead of the AE-Bit. Setting the AF-Bit cancels the job and declares it as ended.

Controller

1. Process output buffer
(note sequence):

01 _{hex}	Command 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} / 0F _{hex}	Set AV bit

3. Process input buffer:

01 _{hex}	Copy error number
-------------------	-------------------

Process output buffer:

00 _{hex} / 0F _{hex}	Reset AV-Bit
---------------------------------------	--------------

Identification system

2. Process input buffer
(note sequence):

If error occurs immediately!

00 _{hex} / 0F _{hex}	Set AA bit
01 _{hex}	Enter error number
00 _{hex} / 0F _{hex}	Set AF-Bit

4. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA and AF-Bit
---------------------------------------	---------------------

9 Device Function

4. Read 30 bytes on Head 1, start address 10, simultaneous data transmission, with read error

Controller

1. Process output buffer
(note sequence):

01 _{hex}	Command 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} / 0F _{hex}	Set AV bit

3. Process input buffer:

01 _{hex}	Copy error number
-------------------	-------------------

Process output buffer:

00 _{hex} / 0F _{hex}	Reset AV-Bit
---------------------------------------	--------------

Identification system

2. Process input buffer
(note sequence):

If error occurs immediately!

00 _{hex} / 0F _{hex}	Set AA bit
01 _{hex}	Enter error number
00 _{hex} / 0F _{hex}	Set AF-Bit

4. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA and AF-Bit
---------------------------------------	---------------------

9 Device Function

5. Read 30 bytes on Head 1, simultaneous data transmission, start address 10, with read error



Note

If an error occurs after data have begun to be sent, the AF-Bit with corresponding error number is provided instead of the AE-Bit. The error message AF is dominant. Which data are faulty cannot be specified. Setting the AF-Bit cancels the job and declares it as ended.

Controller

Identification system

1. Process output buffer
(note sequence):

01 _{hex}	Command 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} / 0F _{hex}	Set AV bit

2. Process input buffer
(note sequence):

00 _{hex} / 0F _{hex}	Set AA bit
01...0E _{hex}	Enter first 14 bytes
00 _{hex} / 0F _{hex}	Invert TO-Bit

3. Process input buffer:

01...0E _{hex}	Copy first 14 bytes
Process output buffer:	
00 _{hex} / 0F _{hex}	Invert TI-Bit

4. Process input buffer:
If error has occurred!

01 _{hex}	Enter error number
00 _{hex} / 0F _{hex}	Set AF-Bit

5. Process input buffer:

01...0E _{hex}	Copy error number
Process output buffer:	
00 _{hex} / 0F _{hex}	Reset AV-Bit

6. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA and AF-Bit
---------------------------------------	---------------------

9 Device Function

6. Write 30 bytes on Head 1, start address 20

Controller

Identification system

1. Process output buffer
(note sequence):

01 _{hex}	Command 02 _{hex}
02 _{hex}	Start address 14 _{hex}
03 _{hex}	Start address 00 _{hex}
04 _{hex}	No. of bytes 1E _{hex}
05 _{hex}	No. of bytes 00 _{hex}
00 _{hex} / 0F _{hex}	Set AV bit

2. Process input buffer
(note sequence):

00 _{hex} / 0F _{hex}	Set AA bit, invert TO-Bit
---------------------------------------	---------------------------

3. Process output buffer:

01...0E _{hex}	Enter first 14 bytes
00 _{hex} / 0F _{hex}	Invert TI-Bit

4. Process output buffer:

01...0E _{hex}	Copy first 14 bytes
------------------------	---------------------

Process input buffer:

00 _{hex} / 0F _{hex}	Invert TO-Bit
---------------------------------------	---------------

5. Process output buffer:

01...0E _{hex}	Enter second 14 bytes
00 _{hex} / 0F _{hex}	Invert TI-Bit

6. Process output buffer:

01...0E _{hex}	Copy second 14 bytes
------------------------	----------------------

Process input buffer:

00 _{hex} / 0F _{hex}	Invert TO-Bit
---------------------------------------	---------------

7. Process output buffer:

01...02 _{hex}	Enter last 2 bytes
00 _{hex} / 0F _{hex}	Invert TI-Bit

8. Process output buffer:

01...02 _{hex}	Copy last 2 bytes
------------------------	-------------------

Process input buffer:

00 _{hex} / 0F _{hex}	Set AE-Bit
---------------------------------------	------------

9. Process output buffer:

00 _{hex} / 0F _{hex}	Reset AV-Bit
---------------------------------------	--------------

10. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA and AE-Bit
---------------------------------------	---------------------

9 Device Function

7. Mixed data access – Save program (3 data records)

1 st data record	Start address	5	No. of bytes	7
2 nd data record	Start address	75	No. of bytes	3
3 rd data record	Start address	112	No. of bytes	17
Total number of bytes exchanged in the operation:				27 bytes

All 104 bytes are written for this programming.

Controller

Identification system

1. Process output buffer
(note sequence):

01 _{hex}	Command 06 _{hex}
02 _{hex}	Program number 01 _{hex}
00 _{hex} / 0F _{hex}	Set AV bit

2. Process input buffer
(note sequence):

00 _{hex} / 0F _{hex}	Set AA bit, invert TO-Bit
---------------------------------------	---------------------------

3. Process output buffer:

01 _{hex}	1 st start address	05 _{hex}
02 _{hex}		00 _{hex}
03 _{hex}	1 st no. of bytes	07 _{hex}
04 _{hex}		00 _{hex}
05 _{hex}	2 nd start address	4B _{hex}
06 _{hex}		00 _{hex}
07 _{hex}	2 nd no. of bytes	03 _{hex}
08 _{hex}		00 _{hex}
09 _{hex}	3 rd start address	70 _{hex}
0A _{hex}		00 _{hex}
0B _{hex}	3 rd no. of bytes	11 _{hex}
0C _{hex}		00 _{hex}
0D _{hex} / 0E _{hex}	End delimiter	FF _{hex} FF _{hex}
00 _{hex} / 0F _{hex}	Invert TI-Bit	

4. Process input buffer:

00 _{hex} / 0F _{hex}	Invert TO-Bit
---------------------------------------	---------------

5. Process output buffer:

01 _{hex} ...0E _{hex}	(not used) FF _{hex} FF _{hex}
00 _{hex} / 0F _{hex}	Invert TI-Bit

6. Process input buffer:

00 _{hex} / 0F _{hex}	Invert TO-Bit
---------------------------------------	---------------

17. Process output buffer:

01 _{hex} ...0E _{hex}	(not used) FF _{hex} FF _{hex}
00 _{hex} / 0F _{hex}	Invert TI-Bit

18. Process input buffer:

00 _{hex} / 0F _{hex}	Set AE-Bit
---------------------------------------	------------

19. Process output buffer:

00 _{hex} / 0F _{hex}	Reset AV-Bit
---------------------------------------	--------------

20. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA and AE-Bit
---------------------------------------	---------------------

9 Device Function

8. Mixed data access – Reading the data carrier with Program No. 1



Note

Dynamic mode is turned off while the program is running.

A total of 27 bytes are exchanged.

Controller

Identification system

1. Process output buffer
(note sequence):

01 _{hex}	Command 21 _{hex}
02 _{hex}	Start address 01 _{hex}
00 _{hex} / 0F _{hex}	Set AV bit

2. Process input buffer
(note sequence):

00 _{hex} / 0F _{hex}	Set AA bit
01...0E _{hex}	Enter first 14 bytes
00 _{hex} / 0F _{hex}	Set AE-Bit

3. Process input buffer:

01...0E _{hex}	Copy first 14 bytes
------------------------	---------------------

Process output buffer:

00 _{hex} / 0F _{hex}	Invert TI-Bit
---------------------------------------	---------------

4. Process input buffer:

01...0D _{hex}	Enter last bytes
------------------------	------------------

Process output buffer:

00 _{hex} / 0F _{hex}	Invert TO-Bit
---------------------------------------	---------------

7. Process input buffer:

01...0D _{hex}	Copy last bytes
------------------------	-----------------

Process output buffer:

00 _{hex} / 0F _{hex}	Reset AV-Bit
---------------------------------------	--------------

8. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA and AE-Bit
---------------------------------------	---------------------

9 Device Function

9. Mixed data access – Writing the data carrier with Program No. 1



Note

Dynamic mode is turned off while the program is running.

A total of 27 bytes are exchanged.

Controller

Identification system

1. Process output buffer
(note sequence):

01 _{hex}	Command 22 _{hex}
02 _{hex}	Program number 01 _{hex}
00 _{hex} / 0F _{hex}	Set AV-Bit

2. Process input buffer
(note sequence):

00 _{hex} / 0F _{hex}	Set AA-Bit Invert TO-Bit
---------------------------------------	-----------------------------

3. Process output buffer:

01...0E _{hex}	Enter first 14 bytes
00 _{hex} / 0F _{hex}	Invert TI-Bit

4. Process output buffer:

01...0E _{hex}	Enter first 14 bytes
Process input buffer:	
00 _{hex} / 0F _{hex}	Invert TO-Bit

5. Process output buffer:

01...0D _{hex}	Enter last bytes
00 _{hex} / 0F _{hex}	Invert TI-Bit

6. Process output buffer:

00 _{hex} / 0D _{hex}	Copy last bytes
Process input buffer:	
00 _{hex} / 0F _{hex}	Set AE-Bit

7. Process output buffer:

00 _{hex} / 0F _{hex}	Reset AV-Bit
---------------------------------------	--------------

8. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA-Bit Reset AE-Bit
---------------------------------------	------------------------------

9 Device Function

10. Initialize data carrier for CRC

CRC initializing is handled like a write command. The start address and number of bytes must correspond to the maximum used data quantity.
In the example the complete memory range of a 752-byte data carrier is used. 658 bytes of the data carrier are available as user bytes, since 94 bytes are required for the CRC.

Controller

Identification system

1. Process output buffer
(note sequence):

01 _{hex}	Command 12 _{hex}
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} / 0F _{hex}	Set AV bit

2. Process input buffer
(note sequence):

00 _{hex} / 0F _{hex}	Set AA bit, invert TO-Bit
---------------------------------------	---------------------------

3. Process output buffer:

01...0E _{hex}	Enter first 14 bytes
00 _{hex} / 0F _{hex}	Invert TI-Bit

4. Process output buffer:

01...0E _{hex}	Copy first 14 bytes
------------------------	---------------------

Process input buffer:

00 _{hex} / 0F _{hex}	Set AE-Bit
---------------------------------------	------------

5. Process output buffer:

01...0E _{hex}	Enter second 14 bytes
00 _{hex} / 0F _{hex}	Invert TI-Bit

6. Process output buffer:

01...0E _{hex}	Copy second 14 bytes
------------------------	----------------------

Process input buffer:

00 _{hex} / 0F _{hex}	Invert TO-Bit
---------------------------------------	---------------

95. Process output buffer:

01...08 _{hex}	Enter last bytes
00 _{hex} / 0F _{hex}	Invert TI-Bit

96. Process output buffer:

01...08 _{hex}	Copy last bytes
------------------------	-----------------

Process input buffer:

00 _{hex} / 0F _{hex}	Set AE-Bit
---------------------------------------	------------

97. Process output buffer:

00 _{hex} / 0F _{hex}	Reset AV-Bit
---------------------------------------	--------------

98. Process input buffer:

00 _{hex} / 0F _{hex}	Reset AA-Bit Reset AE-Bit
---------------------------------------	------------------------------

9 Device Function

11. Set Read/Write Head 1 to base state

Both read/write heads in the identification system can be independently set to the base state.

Controller

Identification system

1. Process output buffer:

00 _{hex} / 0F _{hex}	Set GR-Bit
---------------------------------------	------------

2. Go to base state.
Process input buffer:

00 _{hex} / 0F _{hex}	Reset BB-Bit
---------------------------------------	--------------

3. Process output buffer:

00 _{hex} / 0F _{hex}	Reset GR-Bit
---------------------------------------	--------------

4. Process input buffer:

00 _{hex} / 0F _{hex}	Set BB-Bit
---------------------------------------	------------

12. Deselecting the Read/Write Head

In normal operation both read-write heads are selected. Setting the KA bit allows one or both read-write heads to be deselected (turned off).

Controller

1. Process output buffer:

00 _{hex} / 0F _{hex}	Set KA-Bit
---------------------------------------	------------

Resetting the KA bit selects the read/write head again.



Note

Selecting a read/write head may take up to a second. Deselecting it requires much less time.

Appendix

Ordering code

BIS M - 6026 - 034 - 050 - 06 - ST19

Balluff Identification system _____
 Series M Read/Write System _____
 Hardware Type _____
 6026 = Metal enclosure, EtherNet/IP
 Software Type _____
 034 = EtherNet/IP
 Version _____
 050 = with two ports for external read/write heads BIS M-3_ _
 Interface _____
 06 = Ethernet
 Customer connection _____
 ST19 = Plug variant
 X1 = Round connector for supply voltage (5-pin male)
 X3 = Round connector for Ethernet (4-pole female)
 X4 = Round connector for RS 232 interface (4-pin male)

**Accessories
(optional, not
included in scope
of delivery)**

Type

Ordering code

Connector no cable:	for Head 1, Head 2	BKS-S117-00
Connection cable: one end with a straight, molded-in connector (female), one end for user-assembled connector, length as desired.	for Head 1, Head 2; 5 m for Head 1, Head 2; 10 m for Head 1, Head 2; 25 m for Head 1, Head 2; 50 m	BIS S-501-PU1-05 BIS S-501-PU1-10 BIS S-501-PU1-25 BIS S-501-PU1-50
Connection cable: one end with a right-angle format, molded-in connector (female), one end for user-assembled connector, length as desired.	for Head 1, Head 2; 5 m for Head 1, Head 2; 10 m for Head 1, Head 2; 25 m	BIS S-502-PU1-05 BIS S-502-PU1-10 BIS S-502-PU1-25
Connector	for X1 for X3	BKS-S 79-00 BKS-S 182-00
Cover cap	for X4 for Head 1, Head 2	BES 12-SM-2 Cover cap M12 female (121 671)
Adapter cable M12 D coded to RJ45		BIS C-526-PVC-00,5

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	5A	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	0A	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	3A	:	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	6A	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	1A	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	4A	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	7A	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	2A		*	85	55	U			

Index

A

Abbreviations 5
Accessories 46
AnyBus IPconfig 16
Areas of application 10
ASCII Table 47
Auto-Read
 Extra 19
 Standard 19

B

Bus connection 11

C

Communication 30
 Basic sequence 30
Connection diagrams 14
Connections
 Ethernet 15
 Power supply 15
 Service port 15
Control function 10
Conventions Enumerations 4
 Actions 4
 Notation 4
CRC check 18
 Checksum 18
 Error message 18
 Initializing 18

D

Data
 Electrical 12
 Mechanical 12
Data carrier types 18
 BIS L-10_-01/L 18
Data integrity 10
Description 10
DHCP 16
Dimensions 12
Dual bit-header 10
Dynamic mode 19

E

EEPROM, changing 15
Electrical data 12
EtherNet/IP 11

F

Function indicators 13, 33
 Diagnostics 33
 Overview 33
 Power-up 33
Function principle 26

I

Indicators 13
Input buffer 28
 Allocation and explanation 29
Installation 14
Intended use 6
Interface 14

M

Mechanical data 12
Mixed data access 30
 Program structure example 31
 Running programs 31
 Saving programs 30

O

Operating conditions 13
Ordering code 46
Output buffer 26
 Allocation and explanation 27
 Command 27

P

Parameterizing 20
 Application program 20
 Basics 20
 EDS file 20
 Examples 23
Parameters 20
 CRC_16 20
 DTTyp 22
 Dynamic1 20
 Dynamic2 21
 Extra_Adr1 21
 Extra_Adr2 22
 Extra1 21
 Extra2 21
 Simultaneous 20
 TypSN 22
Principles of operation 10
Processor installation 14
Product description 10

R

Read/write times 32
Read times 32

S

Safety 6
 General notes 6
 Warning notes 6
Simultaneous data transmission 19
 Reading 19
 Writing 19
Symbols 4

W

Write times 32

 **www.balluff.com**

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**