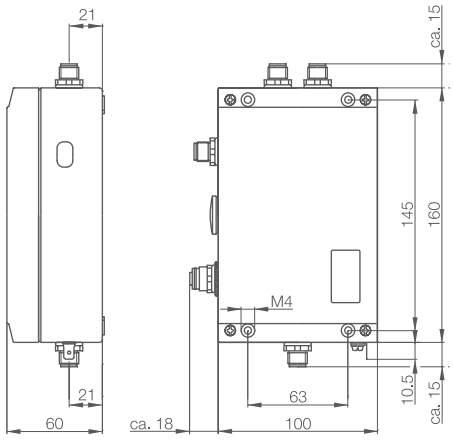
BALLUFF

sensors worldwide

BIS C-6027 Ethernet with TCP/IP-Protocol

Technical Description, User's Manual

BIS C



english

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 safety instructions 6
3 Getting Started 7
4 Basic Knowledge 9
4.1 Identification system principles of operation 9
4.2 Product description 9
4.3 Control function 9
4.4 Data integrity 9
4.5 Communication module 10
4.6 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 BIS SetIP 16
8 Parameterizing the Processor 17
8.1 Basic knowledge 17
8.2 Software COM Port Redirector 19
8.3 Parameterizing 21
9 Device Function 24
9.1 Function principle of BIS C-6027 24
9.2 Protocol sequence (examples) 25
9.3 Communication 26
9.4 Error numbers 35
9.5 Read/write times 36
9.6 Function indicators 37
9.7 Telegram examples 39
Appendix 44

1 User Notes

1.1 About this Manual

This manual describes the processor for the BIS C-6027 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.
Section 2: Basic information regarding safety.
Section 3: The main steps in installing the identification system.
Section 4: An introduction into the material.
Section 5: Technical data for the processor.
Section 6: Mechanical and electrical connections.
Section 7: Logging the processor on to the network.
Section 8: User-defined settings for the processor.
Section 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. 00hex).

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).

Control characters:
Control characters for sending are placed in arrow brackets (e.g. <ACK>).

ASCII code:
Characters to be sent in ASCII code are placed in apostrophes (e.g. 'L').

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

BIS	Balluff Identification System
CRC	Cyclic Redundancy Code
EEPROM	Electrical Erasable and Programmable ROM
EMC	Electromagnetic Compatibility
IP	Internet Protocol
MAC-ID	Media Access Control Identifier
PC	Personal Computer
PLC	Programmable Logic Controller
TCP	Transmission Control Protocol

2 Safety

2.1 Intended use

The BIS C-6027 processor is a component of the BIS C 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 C-6027-039-….

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
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 safety instructions

⚠ **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.

3 Getting Started

Mechanical connection

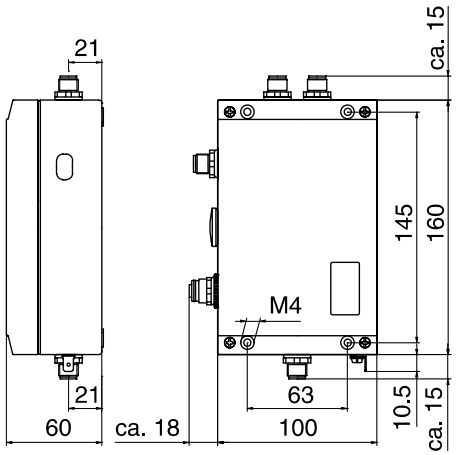


Fig. 1: Mechanical connection

► 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.

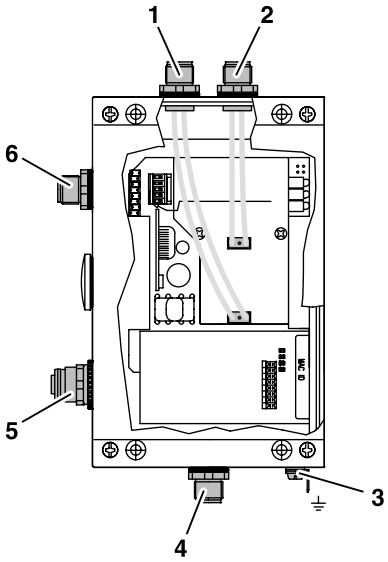
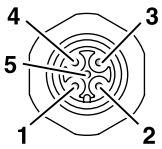


Fig. 2: Electrical connection

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

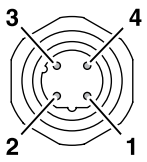
3 Getting Started

X1 - Power supply



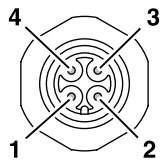
PIN	Function
1	+Vs
2	
3	-Vs
4	
5	

X3 - Ethernet



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

X4 - Service port

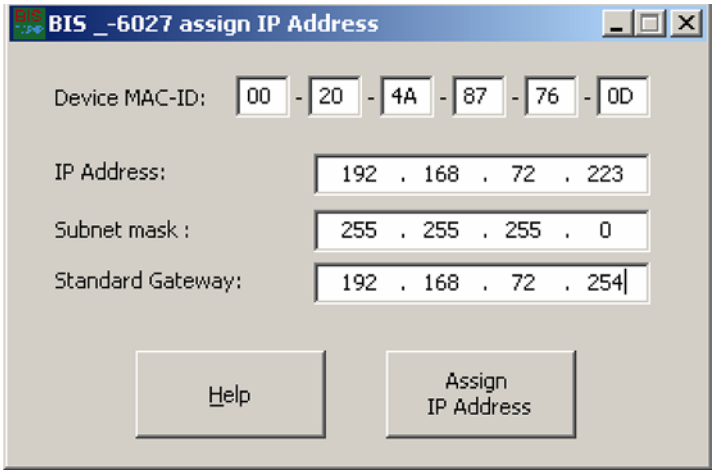


PIN	Function
1	
2	TxD
3	GND
4	RxD

BUS connection

The bus connection is established using the "BIS SetIP" program running on a Windows PC having an Ethernet connection. The "BIS SetIP" application is included on the BIS-CD supplied.

► Start "BIS SetIP".
⇒ The "BIS_-6027 assign IP Address" window is opened.



► Enter the MAC-ID for the device.

Note
The MAC-ID for the device can be found on the sticker on the housing cover.

► Assign IP address, subnet mask and gateway address.
► Confirm your setting by clicking on "Assign IP Address".

4 Basic Knowledge

4.1 Identification system principles of operation

The BIS C 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 C 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 tool coding and monitoring,
– In process equipment organization,
– In inventory systems for monitoring inventory movements,
– In transport and conveying technology,
– In waste management for quantity-dependent logging.

4.2 Product description

BIS C-6027 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.

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.

4 Basic Knowledge

Note
Mixed operation of the two check procedures is not possible!

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 Communication module

The communication module is used for implementing data exchange between the processor and the host system.

LED Indicator

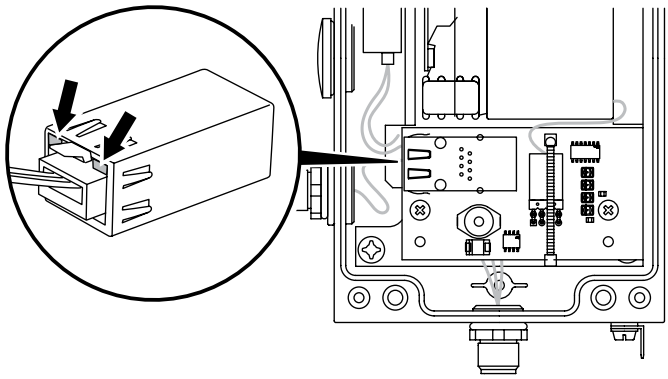


Fig. 3: LED indicator on communication module

The LED on the communication module indicates the status of the Ethernet connection.

LED 1 (10 BASE-T connection)	LED 2 (100 BASE-T connection)	Connection type
Off	Off	No connection
Off	Yellow	100 BASE-T half-duplex
Off	Flashing yellow	100 BASE-T half-duplex; activity
Off	Green	100 BASE-T full-duplex
Off	Flashing green	100 BASE-T full-duplex; activity
Yellow	Off	10 BASE-T half-duplex
Flashing yellow	Off	10 BASE-T half-duplex; activity
Green	Off	10 BASE-T full-duplex
Flashing green	Off	10 BASE-T full-duplex; activity

4 Basic Knowledge

Resetting the communication module

The communication module settings can be reset to their factory default condition.

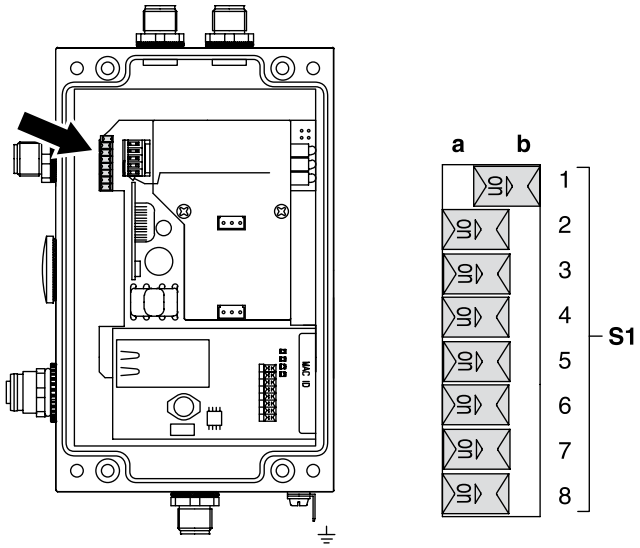


Fig. 4: Switch S1

a S1.1 OFF: Device functions in normal operating state
b S1.1 ON: Reset communication settings to factory default

Procedure:

Attention!
The Switches S1.2 … S1.8 have to be set to OFF.

- ► Turn off power supply
- ► Set S1.1 to **ON.**
  - ⇒ Communication module settings are reset.
  - ⇒ After a successful reset, the "Ready", "CT Present" and "CT Operating" LEDs flash cyclically.
- ► Turn off device.
- ► Set S1.1 to **OFF**.
- ► Turn on power supply.
  - ⇒ Settings are reset to factory default values.

4.6 Bus connection

The processor and host system communicate using the physical Ethernet network.
The device uses Internet Protocol (IP) for network communication.
Transmission Control Protocol (TCP) is used to ensure complete, errorless and properly sequenced data transmission.

5 Technical Data

Dimensions

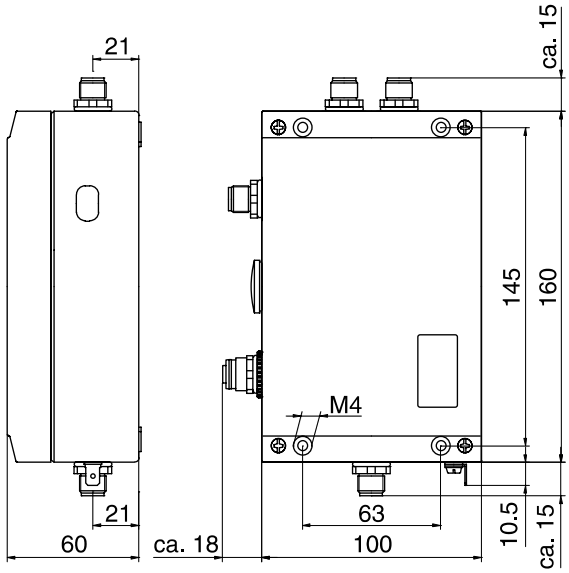


Fig. 5: Dimensions in mm

Mechanical data

Housing material	EN AC-AlSi12 (a), DIN EN 1706
X1 – Input	V_S 24 V DC - 5-pin plug
X3 – Ethernet	M12 - 4-pin female, D-coded
X4 – Service port	RS 232 - 4-pin plug
Head 1, 2 (Read/Write head connections)	4-pin plug
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	 – Severity level 4A/3A/3A/1B/3A – Gr. 1, Cl. A
Shock/Vibration	EN 60068 Part 2-6/27/29/64/32

Function indicators

BIS operating states	Ready CT Present CT Operating	Green LED Yellow LED Yellow LED
Status Ethernet TCP/IP connection	Receive Data (RxD) Transwith Data (TxD) Network Status (NS) Ready (BB)	Yellow LED Yellow LED Green LED Green LED
Physical Ethernet status (displays on communication module)	No connection Half-duplex connection Half-duplex; activity Full-duplex connection Full-duplex; activity	Off LED Yellow LED Yellow flashing LED Green LED Green flashing LED

6 Installation

6.1 Processor installation

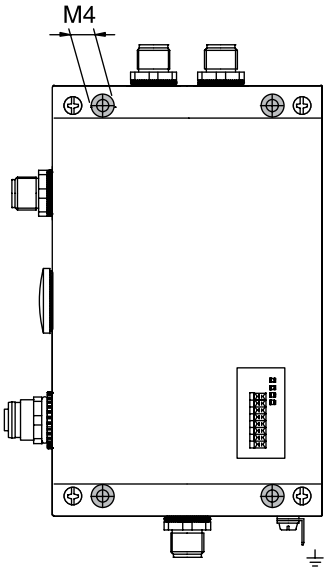


Fig. 6: Installation

► 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.

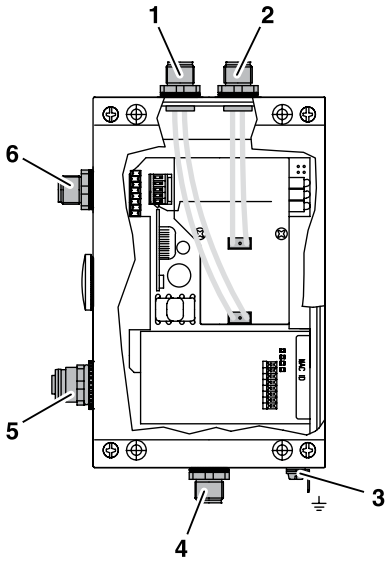
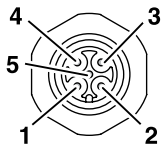


Fig. 7: Processor connections

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

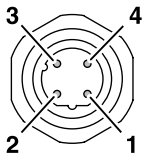
6 Installation

X1 - Power supply



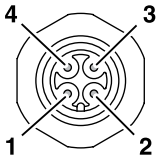
PIN	Function
1	+Vs
2	
3	-Vs
4	
5	

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.

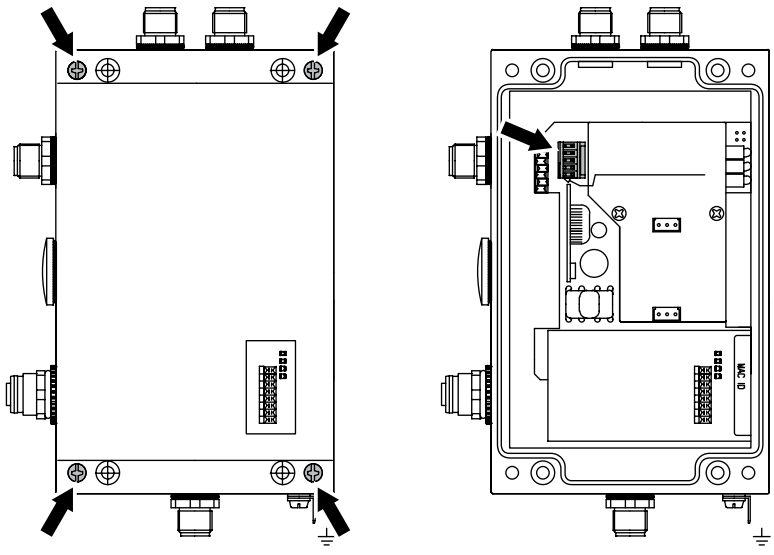


Fig. 8: Changing the EEPROM

► 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 over the Ethernet. Assigning a unique IP address associates the processor with a network.

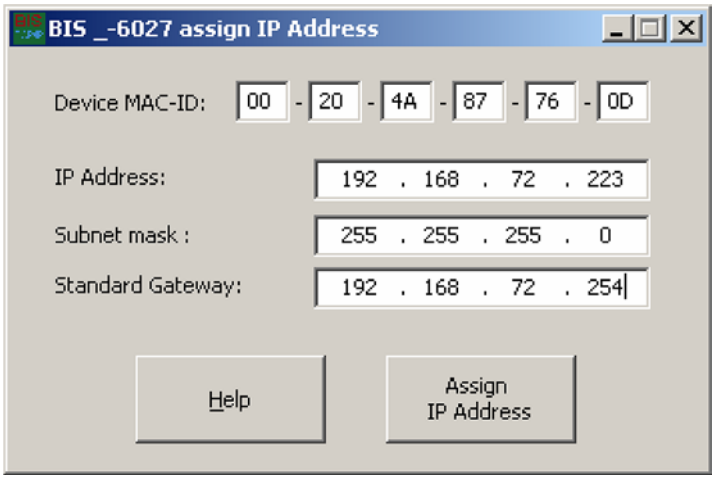
The MAC address is used as the basis for incorporating into the network. This hardware address is used only one time and uniquely identifies network devices such as the processor.

7.2 BIS SetIP

"BIS SetIP" is a software program that allows you to address the hardware for the corresponding subnet before installing it.

The "BIS SetIP" application is included on the BIS CD which comes with the processor.

► Start "BIS SetIP".
⇒ The "BIS_-6027 assign IP Address" window is opened.



► Enter the MAC address for the device.

Note
The MAC-ID for the device can be found on the sticker on the housing.

► Assign the IP address, subnet mask and gateway address.
► Confirm your settings by clicking on "Assign IP Address".

8 Parameterizing the Processor

8.1 Basic knowledge

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 using the command ID 'Z'. If the data carrier does not contain the correct CRC when reading or writing, the processor sends the error message 'CRC-Error'.
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 per block. 2 bytes per block are lost for the data transmission. This leaves 30 or 62 bytes remaining depending on the block size of the data carrier. The usable number of bytes can be determined from the following table.

Data carrier type	Usable bytes
128 bytes	120 bytes
256 bytes	240 bytes
511 bytes	450 bytes
1023 bytes	930 bytes
2047 bytes	1922 bytes
2048 bytes	1984 bytes
8192 bytes	7936 bytes

Send CT data immediately

Each time a data carrier is recognized, it is read depending on the setting. The data are output over the interface.
This setting allows you to eliminate a read command in dialog mode.
The prescribed amount of data (start addresses and number of bytes) can be set (see Configuration on page 22).

Dynamic mode

As soon as the Dynamic mode 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 executed.

8 Parameterizing the Processor

Protocol type

The factory setting is for block check BCC. The BCC is formed as an EXOR operation from the serially sent binary characters of the telegram block.

If needed the termination using BCC block check can be replaced by the ASCII character "Carriage Return" (CR).

For controllers that always require a termination character, this must always be inserted into the telegrams. The following are available:
– "Carriage Return" (CR) or
– "Line Feed with Carriage Return" (LF CR).

Examples for terminating the telegrams:

Protocol type	Telegram with command, address, no. of bytes, head-no., block size	Terminator	Acknowl-edgement	Termination identifier
Block check BCC	L 0000 0001 10	BCC	<ACK> 0	
CR	L 0000 0001 10	CR	<ACK> 0	
Termination identifier CR	L 0000 0001 10	CR	<ACK> 0	CR
Termination identifier LF CR	L 0000 0001 10	LF CR	<ACK> 0	CR

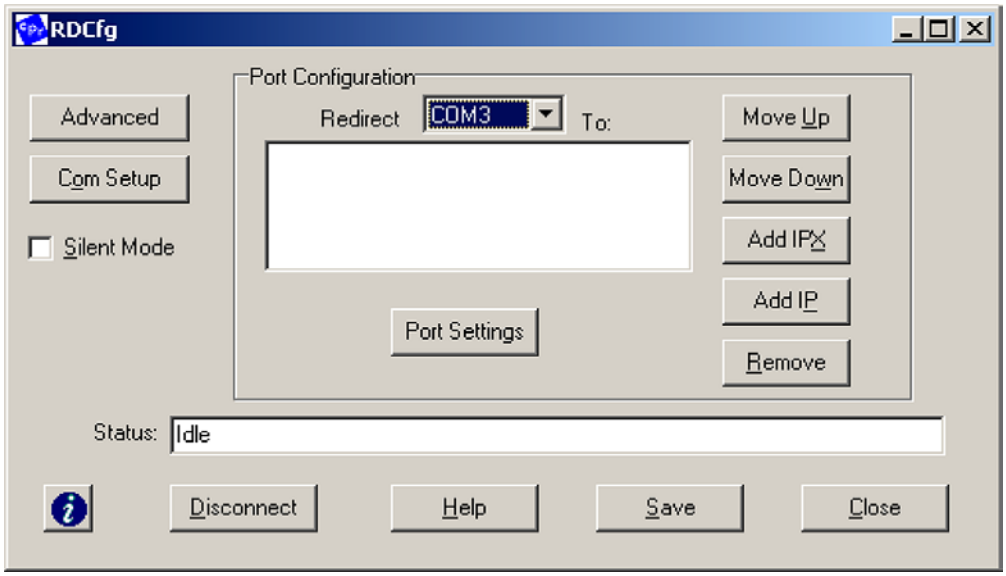
8 Parameterizing the Processor

8.2 Software COM Port Redirector

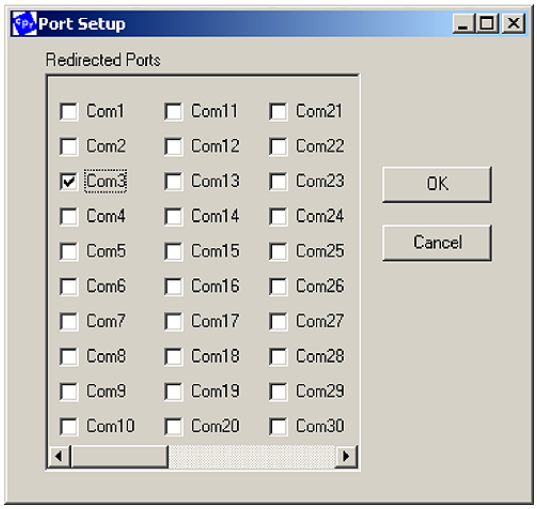
The Com Port Redirector allows a software program with COM port support to send TCP/IP sockets on the Ethernet.
The COM Port Redirector installs virtual Windows Communication (or COM) Ports for this purpose. Data which are sent through these virtual COM Ports (e.g. COM3) are passed along to the network station over the network as TCP/IP sockets.
The "Com Port Redirector" software program is included on the BIS-CD provided.

Set up virtual COM Port

► Start "Lantronix Redirector --> Configuration" software,
⇒ Configuration window "RDCfdg" opens.

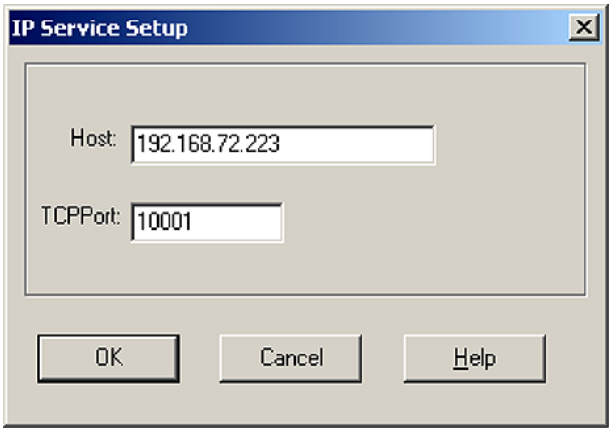


► Click on "Com Setup" button.
► Select number of COM port you want to set up as a virtual prot.
► Confirm selection by clicking on "OK".

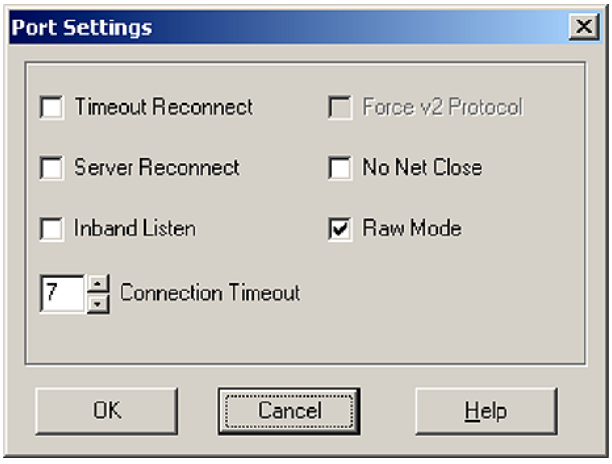


8 Parameterizing the Processor

- Click on "Add IP" button in "RCDfg" window.
- Enter IP address of the processor in the "Host" field.
- Enter 10001 in the "TCPPort" field.
- Confirm entries by clicking on "OK".



- Click on "Port Settings" button in "RCDfg" window.
- Activate "Raw Mode" option.
- Confirm selecting by clicking on "OK".



- Click on "Save" button in "RCDfg" window.
  ⇒ The settings are saved.
- Quit program and restart PC.
  ⇒ The virtual Windows Port is ready to use.

8 Parameterizing the Processor

8.3 Parameterizing

Basics

Parameterizing is done using the "Configuration Software BIS" software.
Configuration is done online. The parameters may be overwritten at any time. The configuration may be saved in a file so that it is always available.

The "Configuration Software BIS" software is included in the BIS-CD provided.

Requirements

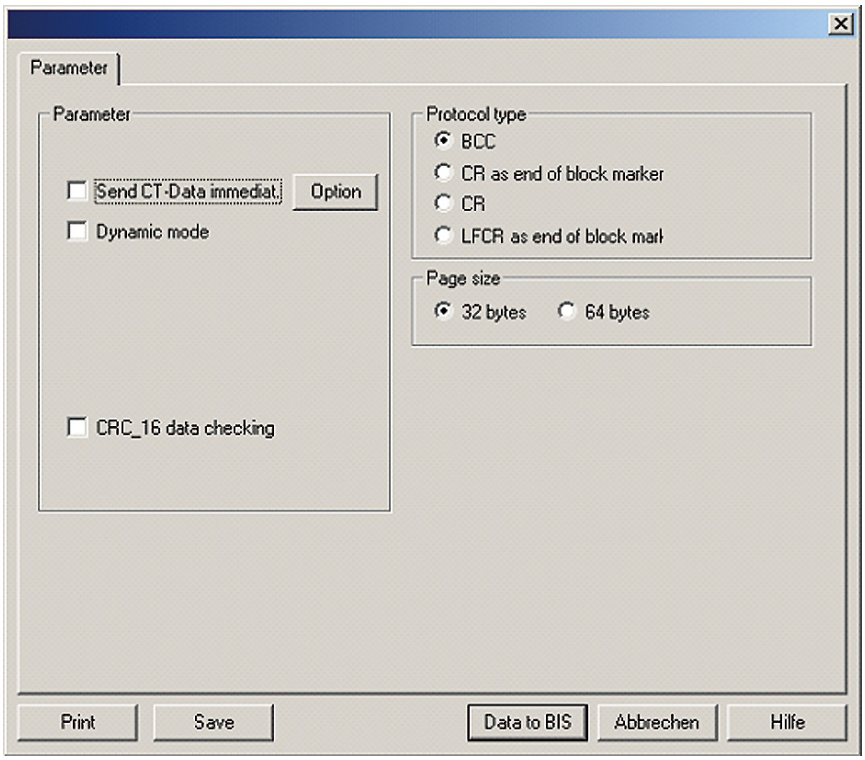
– Software "Com Port Redirector" is installed and a virtual port is set up (see Section 8.2)
– The device is online (available on the Ethernet).

Start configuration software

► Start BIS configuration software.
► Select COM Port in "Configuration --> Port" menu
► Select "**BIS C-6027-039**".

8 Parameterizing the Processor

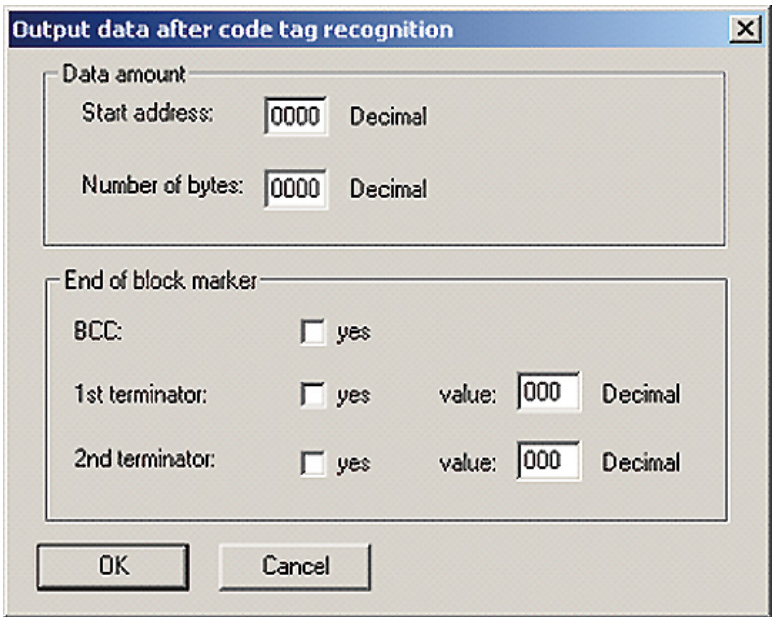
► Select "Online --> Initialize" menu.
⇒ "Parameter" dialog box opens.



Send CT Data immediately parameter

Each time a data carrier is recognized, it is read depending on the configuration. The data are output to the port.

Send CT Data immediately --> Option
Further configuration of the parameter if *Send CT Data immediately* is active.



8 Parameterizing the Processor

Data quantity
Specified amount of data to be read from a newly recognized data carrier (number of bytes beginning with the start address).

end identifier
As an option a BCC and/or 1 or 2 freely definable termination characters may be sent as a terminator.

Factory setting: *Send CT Data immediately* not active.

Dynamic mode

Dynamic-mode activated:
A read/write command is stored until a data carrier enters the working range of the read/write head.

Dynamic-mode not activated:
A read/write command is carried out only if there is a data carrier in the range of the read/write head.
If there is no data carrier in the range of the read/write head, a read/write command is rejected with the error message <NAK>'1'. The processor goes into the base state.

Factory setting: *Dynamic-mode* not active.

Parameter CRC_16 data check

CRC_16 data check activated:
The validity of the data is verified using a CRC check (see also Data Integrity section on page 9).

CRC_16 data check not activated:
The validity of the data is verified by a double read.

Factory setting: *CRC_16 data check* not active.

Page size

Page size (also: block size) describes the memory organization of the data carriers.

32 bytes	BIS C-1__-02, -03, -04, -05
64 bytes	BIS C-1__-10, -11, -30, -32
Factory setting:	*32 bytes*

Protocol type

For selecting the protocol type (protocol variant).

BCC	Blockcheck	Factory setting.
CR as end identifier	Carriage Return as end identifier	Also possible for controllers always requiring a termination character.
CR	Carriage Return	If needed, terminator using BCC can be replaced with *CR*.
LFCR as end identifier	Line Feed with Carrige Return	Also possible for controllers always requiring a termination character.

9 Device Function

9.1 Function principle of BIS C-6027

The processor and host system communicate over the physical Ethernet of BIS C-6027 network.
The device uses Internet Protocol (IP) for network communication.
Transmission Control Protocol (TCP) is used for ensuring complete, errorless and properly sequenced data transmission.

The host system and BIS C-6027 communicate via TCP/IP sockets. Communication is done in raw mode (only user data is exchanged, no configuration or status information).

Possibilities for opening a connection:

1. Socket connection to the IP address of the device, Port 10001. How the connection is established depends on the (PC operating) system used and the programming language.
2. Use of the "Com Port Redirector" software (see Section 8.2 on page 19) and a software program with access to a COM port (e.g. "Hyperterminal". For simple read/write access, the "BISCOMRW" program (included on the supplied BIS Software CD) can be used.

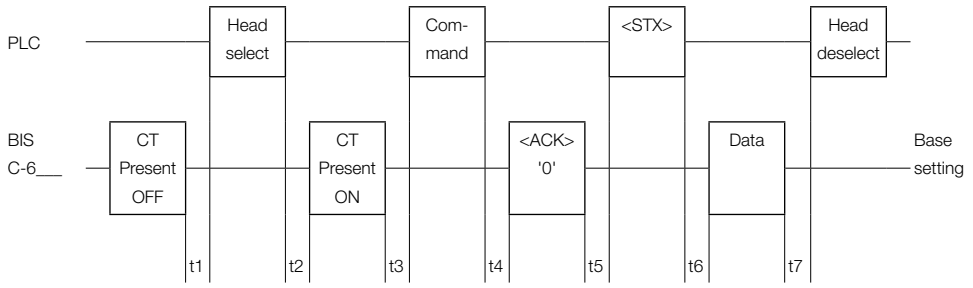
9 Device Function

9.2 Protocol sequence (examples)

Dialog mode with head select

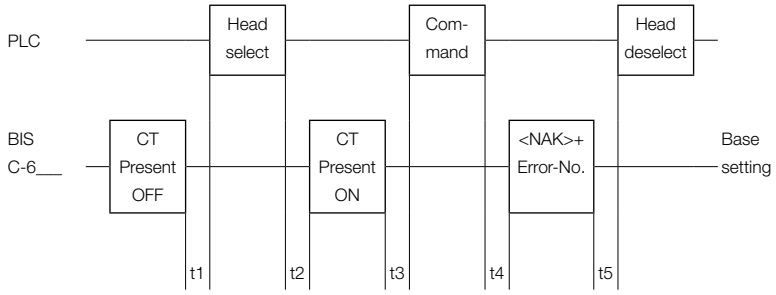
Read:

1. No error occurs:



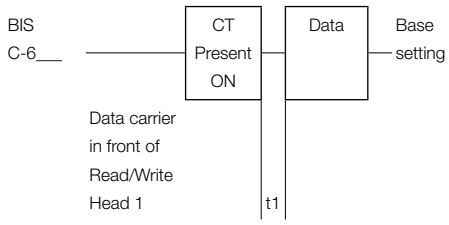
t1, t3, t7 ≥ 0	t5 ≥0 (not monitored by the processor)
t2 = **max 500 ms**	t6 System dependent
t4 depending on no. of bytes to read	

2. An error occurs:



t1, t3, t5 ≥ 0	t4 Depends on number of bytes to read and error type (recommended monitoring time 15 s)
t2 = **max 500 ms**	

Direct Read mode



t1 Depends on number of bytes to read

Prerequisite for validity of these figures:

- The processor is in the base setting.
- There is a data carrier in front oa the read/write head.

9 Device Function

9.3 Communication

The host system and processor communicate with each other using telegrams. There are specific telegrams for each task. These always begini with the command which is associated with the telegram type.

Telegram types with associated command (ASCII character)

'L' Read the data carrier and select the read/write head
'P' Write to the data carrier and select the read/write head
'C' Write a constant value to the data carrier and select the read/write head
'H' Select the read/write head and block size with the variants:
'?' Find the next data carrier – one time
'!' Find the next data carrier – continuous
'S' Check status message
'Q' Restart the processor – Quit
'Z' Initialize CRC-16 data check

Note
Continuous querying on the interface is not allowed. The wait time between two commands must be at least 300 ms.

Explanations of telegram contents

Start address and number of bytes	The start address (A3, A2, A1, A0) and the number of bytes for sending (L3, L2, L1, L0) are sent in decimal as ASCII characters. For the start address the range 0000 to 8191 and for the number of bytes 0001 to 8192 may be used. A3 ... L0 stand for one ASCII character each.
Note	The start address + number of bytes may not exceed the data carrier capacity.
Head number and block size	In the case of the commands 'L' (read with head select and block size) and 'P' (write with head select and block size) the number of the read/write head K ('1' or '2') is sent first and then the block size B ('0', '1'). B = '0' is equivalent to 64 bytes, B = '1' is equivalent to 32 bytes.
Acknowl-edgement	The acknowledgement <ACK> '0' is sent by the identification system if the serially transwithted characters were recognized as correct and there is a data carrier in the working range of a read/write head. <NAK> + 'ErrorNo.' is acknowledged if an error was detected or if there is no data carrier in the working range of the read/write head.
Start	<STX> is used to start data transmission.
Sent bytes	The data are sent code-transparent (no data conversion).

9 Device Function

Telegram for read/write data carrier with R/W head select and block size

Read from the data carrier and select the read-write head and block size, write to the data carrier and select the read-write head and block size:

	Task	Data flow	Com-mand	Start address of the first byte to send	Number of bytes to send	Head num-ber	Block size	Ter. 2)	Acknow. 3)	EI 4)	Start for sen-ding	EI 4)	Data 5)	Ter. 2)	Acknow. 3)	EI 4)
Read		To BIS 6)	'L'	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L2 L1 L0 '0 0 0 1' to '8 1 9 2'	K '1' or '2'	B '0' or '1'	BCC or 2)			<STX>	'CR or 'LF CR'				
		From BIS 7)							<ACK> '0' or <NAK> + Error no.	'CR' or 'LF CR'			D1 D2 D3...Dn	BCC or 2)		
				1)								1)				
Write		To BIS 6)	'P'	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L2 L1 L0 '0 0 0 1' to '8 1 9 2'	K '1' or '2'	B '0' or '1'	BCC or 2)			<STX>		D1 D2 D3...Dn	BCC or 2)		
		From BIS 7)							<ACK> '0' or <NAK> + Error no.	'CR' or 'LF CR'					<ACK> '0' or <NAK> + Error no.	'CR' or 'LF CR'
				1)									1)			

1) The commands Status and/or Quit are not perwithted at this point.
2) Terminator; Instead of BCC block check either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used depending on the protocol variant
3) As an acknowledgement <ACK> '0', is used if there was no error, or <NAK> + 'ErrorNo.', if an error occured.
4) End identifier; for protocol variants which always require an end identifier, one of the termination characters 'CR' or 'LF CR' must be inserted here.
5) Data from start address to start address + number of bytes
6) From host system to BIS
7) From BIS to host system

Note
Telegram examples can be found in Section 9.7 starting page 40.

9 Device Function

Telegram for writing a constant value to the data carrier with R/W head select and block size

For writing a constant value to the data carrier with read/write head select and block size. This command can be used for erasing a data carrier. This saves time in transwithting the bytes for writing.

	Task	Data flow	Com-mand	Start address of the first byte to send	Number of bytes to send	Head num-ber	Block size	Ter. 2)	Acknow. 3)	EI 4)	Start for sen-ding	EI 4)	Data 5)	Ter. 2)	Acknow. 3)	EI 4)
Write		To BIS 6)	'C'	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L2 L1 L0 '0 0 0 1' to '8 1 9 2'	K '1' or '2'	B '0' or '1'	BCC or 2)			<STX>		D	BCC or 2)		
		From BIS 7)							<ACK> '0' or <NAK> + Error no.	'CR' or 'LF CR'					<ACK> '0' or <NAK> + Error no.	'CR' or 'LF CR'
				1)									1)			

1) The commands Status and/or Quit are not perwithted at this point.
2) Terminator; Instead of BCC block check either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used depending on the protocol variant.
3) As an acknowledgement <ACK> '0', is used if there was no error, or <NAK> + 'ErrorNo.', if an error occured.
4) End identifier; for protocol variants which always require an end identifier, one of the termination characters 'CR' or 'LF CR' must be inserted here.
5) Data from start address to start address + number of bytes.
6) From host system to BIS
7) From BIS to host system

Note
Telegram examples can be found in Section 9.7 starting page 41.

9 Device Function

Telegram for selecting the read/write head

Selecting the read/write heads using the commands:
- 'H1' Select read/write head 1,
- 'H2' Select read/write head 2,
- 'HT' "Head Twin": Select both read/write heads.

When selecting **both** read/write heads note the following:
1. Only one data carrier at a time is perwithted at a read/write head.
2. The read or write time increases by approx. 40 ms depending on the amount of data to be read or written (exception: data carrier recognition).
3. The positive acknowledgement for a read or write job is
- <ACK> '1', if there is a data carrier at read/write head 1,
- <ACK> '2', if there is a data carrier at read/write head 2.

Task	Data flow	Com-mand	Head number	Termination 2)	Acknow. 3)	End identifier 4)
Select read/write head	From host to BIS	'H'	'1', '2' or 'T'	BCC or 2)		
	From BIS to host				<ACK> '0' or <NAK> + Error no.	'CR' or 'LF CR'
			1)			

1) The commands Status and/or Quit are not perwithted at this point.
2) Terminator; Instead of BCC block check either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used depending on the protocol variant .
3) As an acknowledgement <ACK> '0', is used if there was no error, or <NAK> + 'ErrorNo.', if an error occured.
4) End identifier; for protocol variants which always require an end identifier, one of the termination characters 'CR' or 'LF CR' must be inserted here.

Note
Telegram examples can be found in Section 9.7 starting page 41.

9 Device Function

Telegram for finding next data carrier (one time)

This telegram is used to find the next data carrier. A check is made to see whether there is a data carrier in front of the next following read/write head.
If the read/write head and data carrier are compatible, 'H ?' detects any data carrier, regardless of the set block size.

Telegram replies:
- **Data carrier in front of read/write head**: The telegram reply contains the number of the read/write head and the first 4 bytes from the data carrier.
- **No data carrier in front of read/write head**: The original read/write head is selected again and checked. If no data carrier is found here either, the telegram reply is 'H ? 0000 w'.

Task	Data flow	Com-mand	Iden-tifica-tion	Termination 2)	Acknow.	End identifier 3)	Reply	Head number	Data from the data carrier	Termination 2)
Find next data carrier (one time)	From host to BIS	'H'	'?'	BCC or 2)						
	From BIS to host				<ACK> '0'	'CR' or 'LF CR'	'H'	'1', '2' or '?'	D1 D2 D3 D4	BCC or see 2)
			1)							

1) The commands Status and/or Quit are not perwithted at this point.
2) Terminator; Instead of BCC block check either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used depending on the protocol variant.
3) End identifier; for protocol variants which always require an end identifier, one of the termination characters 'CR' or 'LF CR' must be inserted here.

Note
Telegram examples can be found in Section 9.7 starting page 42

9 Device Function

Telegram for finding next data carrier (continuous)

This telegram is used to find the next data carrier. A check is made to see whether there is a data carrier in front of the next following read/write head.
If the read/write head and data carrier are compatible, 'H ?' detects any data carrier, regardless of the set block size.

Telegram replies:

- **Data carrier in front of read/write head**: The telegram reply contains the number of the read/write head and the first 4 bytes from the data carrier.
- **No data carrier in front of read/write head**: The original read/write head is selected again and checked. This is repeated continuously until a data carrier is recognized.

Task	Data flow	Com-mand	Iden-tifica-tion	Termination 2)	Acknow.	End identifier 3)	Reply	Head number	Data from the data carrier	Termination 2)
Find next data carrier (continuous)	From host to BIS	'H'	'!'	BCC or 2)						
	From BIS to host				<ACK> '0'	'CR' or 'LF CR'	'H'	'1', '2' or '?'	D1 D2 D3 D4	BCC or 2)
			1)							

1) The commands Status and/or Quit are not perwithted at this point.
2) Terminator; Instead of BCC block check either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used depending on the protocol variant.
3) End identifier; for protocol variants which always require an end identifier, one of the termination characters 'CR' or 'LF CR' must be inserted.

Note
Telegram examples can be found in Section 9.7 starting page 42.

9 Device Function

Telegram for restarting the processor (Quit)

Sending the Restart (Quit) telegram breaks off any telegram currently in process. The processor is placed in the base state.

Attention!
The Restart (Quit) telegram is not allowed while the processor is waiting for a termination character (BCC, 'CR' or 'LF CR').
In this situation Quit is misinterpreted as a terminator or data character.

Note
After acknowledgement of this telegram allow a pause of **at least 1600 ms** before starting a new telegram.

Task	Data flow	Command	Termination 2)	Acknow.	Termination 2)
Restart (Quit)	From host to BIS	'Q'	BCC or 2)		
	From BIS to host			'Q'	BCC or 2)
			1)		

1) The commands Status and/or Quit are not perwithted at this point.
2) Terminator; Instead of BCC block check either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used depending on the protocol variant.

Note
Telegram examples can be found in Section 9.7 starting page 42.

9 Device Function

Telegram for querying the status message

The status telegram is used to query what kind of telegram is in process.

Attention!
The Status command is not allowed while the processor is waiting for a termination character (BCC, 'CR' or 'LF CR').

Note
A status query during a read or write access to a data carrier will increase the read/write time.
Especially when in dynamic mode this can mean that the dwell time of the data carrier in the working range of the read/write head is no longer sufficient for complete reading or writing.

A continuous status query disturbs processing of the data carrier. This can result in the data carrier not being recognized.

Task	Data flow	Command	Termination 2)	Status message	Termination 2)
Status message query	From host to BIS	'S'	BCC or 2)		
	From BIS to host			'S' '_', 'L', 'P' or 'H' 3)	BCC or see 2)
			1)		

1) The commands Status and/or Qujit are not allowed at this point.
2) Instead of BCC block check either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used depending on the protocol variant.
3) The characters between the apostrophes represent the respective characters in ASCII code.
Example: '_' = Space = ASCII character 20hex

Meaning of the status messages

Status message	Meaning
'S L'	Read data carrier with read/write head select and block size of data carrier.
'S P'	Write to data carrier with read/write head select and block size of data carrier.
'S H'	Select read/write head and block size of data carrier.
'S _'	No telegram in process.

Note
Telegram examples can be found in Section 9.7 starting page 43.

9 Device Function

Telegram fo initializing CRC_16 data check

This telegram is used to initialize a data carrier in front of a read/write head for using CRC_16 data checking.
The telegram must also be resent if a CRC error has occurred as a consequence of a failed write job.

Attention!
The sum of the start address and number of bytes is not allowed to exceed the usable data carrier capacity (see table on page 17).

Task	Data flow	Com-mand	Start address of the first byte to send	Number of bytes to send	Head num-ber	Block size	Ter. 2)	Acknow. 3)	El 4)	Start for sen-ding	Data 5)	Ter. 2)	Acknow. 3)	El 4)
CRC_16 Initialize range	To BIS 6)	'Z'	A3 A2 A1 A0 '0 0 0 0' to '7 9 3 5'	L3 L2 L1 L0 '0 0 0 1' to '7 9 3 6'	K '1' or '2'	B '0' or '1'	BCC or 2)			<STX>	D1 D2 D3...Dn	BCC or 2)		
	From BIS 7)							<ACK> '0' or <NAK> + Error no.	'CR' or 'LF CR'				<ACK> '0' or <NAK> + Error no.	'CR' or 'LF CR'
			1)								1)			

1) The commands Status and/or Quit are not perwithted at this point.
2) Terminator; Instead of BCC block check either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used depending on the protocol variant.
3) As an acknowledgement <ACK> '0', is used if there was no error, or <NAK> + 'ErrorNo.', if an error occured.
4) End identifier; for protocol variants which always require an end identifier, one of the termination characters 'CR' or 'LF CR' must be inserted here.
5) Data from start address to start address + number of bytes
6) From host system to BIS
7) From BIS to host system

9 Device Function

9.4 Error numbers

BIS C-6027 always outputs an error number. Their meanings are listed in the following table.

No.	Error	Effect	
1	No data carrier present.	Telegram broken off. Processor goes into base state.	
2	Read error.	Read telegram broken off. Processor goes into base state.	
3	Read broken off because data carrier was removed.	Processor goes into base state.	
4	Write error.	Write telegram broken off. Processor goes into base state.	⚠ Attention! When the write process is broken off, incomplete data may be written to the data carrier. 1)
5	Write broken off because data carrier was removed.	Processor goes into base state.	
6	Interface error.	Processor goes into base state (parity or stop bit error).	
7	Telegram format error.	Processor goes into base state. Possible format errors: – Command is not 'L', 'P', 'C', 'H', 'Q', 'S' or 'Z'. – Start address or number of bytes outside the permissible range.	
8	BCC error. The sent BCC is wrong.	Telegram broken off. Processor goes into the base state	
9	Cable break on selected read/write head or read/write head not connected. CT Present LED flashes.	Telegram broken off. Processor goes into base state. If both read/write heads were selected using the 'HT' command, one head may not be connected. If both read/write heads are selected, the cable break message is only sent if there is no data carrier in front of the connected, non-defective head.	
A	New command not possible because a read command is already in process.	After the error message the read command is quit internally but no longer acknowledged. Processor goes into base state.	
B	New command not possible because a write command is already in process.	After the error message the write command is quit internally but no longer acknowledged. Processor goes into base state.	
C	New command not possible because a head select command is already in process.	After the error message there is no longer a positive acknowledgement, even though the head select was carried out. Processor goes into base state.	
E	CRC error. The CRC on the data carrier is wrong 2).	Telegram broken off. Processor goes into base state.	
I	EEPROM error.	Telegram broken off. Processor goes into base state.	

1) If you are using a CRC data check, the error message E may occur at the next read command if error 4 or 5 was not cleared.

2) If you are using a CRC data check, the error message E may occur if error 4, 5 or B was reported for the preceding command.

9 Device Function

9.5 Read/write times

Note
The times indicated below commence as soon as the data carrier is recognized. Otherwise 45 ms must be added to allow for energy to be generated before the data carrier is recognized.

Read times in static mode (double reading for data integrity)

32-byte block size data carrier	
No. of bytes	**Read times [ms]**
0 to 31	110
each additional 32 bytes	120
0 to 255	950

64-byte block size data carrier	
No. of bytes	**Read times [ms]**
0 to 63	220
each additional 64 bytes	230
0 to 2047	7350

Write times in static mode (double reading for data integrity)

32-byte block size data carrier	
No. of bytes	**Write times [ms]**
0 to 31	110 + n * 10
32 bytes	Y * 120 + n * 10

64-byte block size data carrier	
No. of bytes	**Write times [ms]**
0 to 63	220 + n * 10
64 bytes	Y * 230 + n * 10

n = No. of contiguous bytes to write
y = No. of blocks to process

Example:
Write 17 bytes starting at Address 187. A data carrier with 32-byte block size is used. Blocks 5 and 6 need to be processed, since the start address 187 is located in Block 5 and the end address 203 is in Block 6.

Write time = 2 * 120 + 17 * 10 = 410 ms

Read times in dynamic mode, first block (double reading for data integrity)

32-byte block size data carrier	
No. of bytes	**Read times [ms]**
0 to 3	14
each additional byte	3.5
0 to 31	112

64-byte block size data carrier	
No. of bytes	**Read times [ms]**
0 to 3	14
each additional byte	3.5
0 to 63	224

Formula: Read time = (m+1) * 3.5
m = highest address to be read

Example:
Read 11 bytes beginning at Address 9. This means the highest address to be read is 19.

Read time = (19+1) * 3.5 = 70 ms

9 Device Function

9.6 Function indicators

Overview of indicators

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

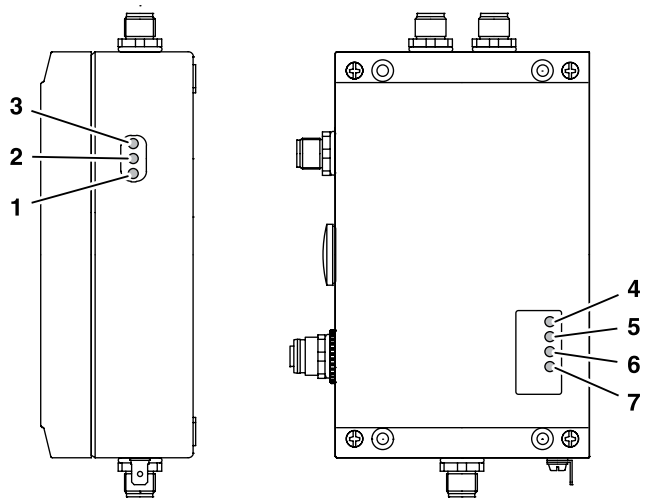


Fig. 9: Function indicators

Identification system

1 CT Operating
2 CT Present
3 Ready

Ethernet

4 Receive Data (RxD)
5 Transwith Data (TxD)
6 Network Status (NS)
7 Ready (BB)

Power-up

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

LED name	LED sequence							
Receive Data (RxD)	off							
Transwith Data (TxD)	off							
Network Status (NS)	on			off	1 x flash	off		
Ready (BB)	on	off	4 x flash			off	1 x flash	off	on

Diagnostics

Identification system

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

CT Present	
yellow	Data carrier ready to read/write at Read/Write Head
yellow flashing	Cable break on Read/Write Head or Read/Write Head not connected
off	No data carrier in the active zone of the Read/Write Head

9 Device Function

Status LED	Meaning
CT Operating	
yellow	Read/write job being processed at Read/Write Head
off	No command

Ethernet and TCP/IP connection

Status LED	Meaning
Receive Data	
off	No data transmission
yellow	Device receiving data

Transwith Data	
off	No data transmission
yellow	Device sending data

Network Status	
off	Device has no TCP/IP connection
green flashd	Device has a TCP/IP connection

Ready	
off	Network module defective. Inform service department
green	Network module is ready

9 Device Function

9.7 Telegram examples

Forming the block check BCC

The BCC is formed as an EXOR operation from the serially sent binary characters of the telegram block.

Example: Read starting at address 13, 128 Byte are to be read.

The command line without BCC is: 'L 0013 0128 20'. BCC is formed:

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

Protocol variants

If needed, the terminator using BCC block check can be replaced by Carriage Return ('CR') or Line Feed with Carriage Return ('LF CR').

The command line 'L 0013 0128 20 G' with 'G' as BCC results from the preceding example. This command line is compared here in the possible variants. The various forms of acknowledgement with and without end identifier are shown.

Command line from host system to BIS	Acknowledgement from BIS for correct reception	Acknowledgement from BIS for incorrect reception
With BCC as Termination, without end identifier 'L 0013 0128 20 G'	without end identifier <ACK> '0'	without end identifier <NAK> '1'
With 'CR' instead of BCC without end identifier 'L 0013 0128 20 CR'	without end identifier <ACK> '0'	without end identifier <NAK> '1'
Without BCC with end identifier 'CR' 'L 0013 0128 20 CR'	with end identifier 'CR' <ACK> '0 CR'	with end identifier 'CR' <NAK> '1 CR'
Without BCC with end identifier 'LF CR' 'L 0013 0128 20 LF CR'	with end identifier 'LF CR' <ACK> '0 LF CR'	with end identifier 'LF CR' <NAK> '1 LF CR'

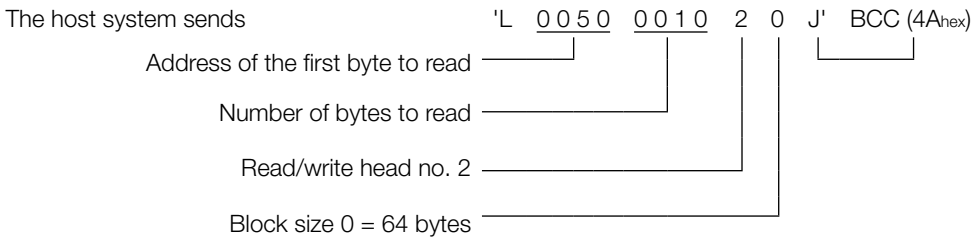
In the table <NAK> '1' (=no data carrier present) is given as an error example.

9 Device Function

Read a data carrier

Telegram example: Read data carrier with read/write head select and block size with BCC block check.

Task: Head 1 is selected. Read 10 bytes from the data carrier at read/write head 2 starting at address 50. The data carrier has a block size of 64 bytes.



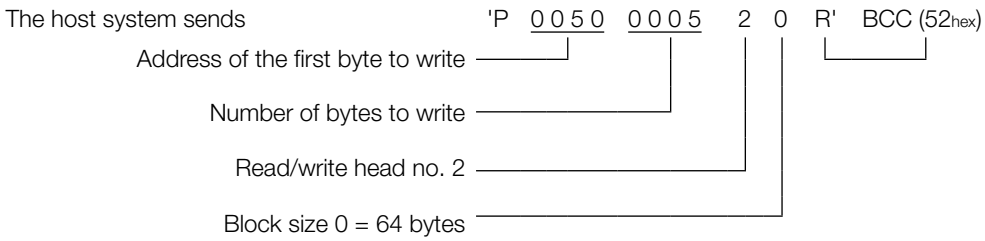
The processor acknowledges with	<ACK> '0'
The host system gives the start command	<STX>
The processor provides the data from the data carrier	'1 2 3 4 5 6 7 8 9 F' BCC (70hex)

After telegram activity has stopped Head 2 with 64 byte block size remains selected.

Write to the data carrier (1)

Telegram example: Write to the data carrier with read/write head select and block size with BCC block check.

Task: Head 1 is selected. Write 5 bytes to the data carrier at read/write head 2 starting at address 50. The data carrier in front of Head 2 has 64-byte block size.



The processor acknowledges with	<ACK>	'0'	
The host system gives the start command and the Data	<STX>	'1 2 3 4 5 3'	BCC (33hex)
The processor acknowledges with	<ACK>	'0'	

After telegram activity has stopped Head 2 with 64 byte block size remains selected.

9 Device Function

Write to the data carrier (2)

Telegram example: Write to the data carrier with read/write head select and block size with BCC block check.

Task: Head 1 is selected. Write 500 bytes to the data carrier at read/write head 2 starting at address 20. The data carrier in front of Head 2 has 64-byte block size.

The host system sends 'C 0020 0500 2 0 F' BCC (46hex)

- Address of the first byte to write
- Number of bytes to write
- Read/write head no. 2
- Block size 0 = 64 bytes

The processor acknowledges with	<ACK>	'0'				
The host system gives the start command and the Data	<STX>		'0	2'	BCC (32hex)	
The processor acknowledges with	<ACK>	'0'				

After telegram activity has stopped Head 2 with 64 byte block size remains selected.

Select the read/ write head

Telegram example: Select the read/write head with BCC block check.

Task: Switch to Head 1.

The host system sends		'H	1	y'	BCC (79hex)
The processor acknowledges with	<ACK> '0'				

9 Device Function

Find next data carrier (one time)

Telegram example: Find next data carrier (one time) with BCC block check.

Task: Head 1 is selected. There is a data carrier only in front of read/write head 2, whose first 4 bytes contain the value 9876.

The host system sends		'H ? w' BCC (77$_{hex}$)
The processor acknowledges with	<ACK> '0'	
and sends the data		'H 2 9 8 7 6 z' BCC (7Ah$_{ex}$)

Find next data carrier (continuous)

Telegram example: Find next data carrier (continuous) with BCC block check.

Task: Head 1 is selected. There is a data carrier only in front of read/write head 2, whose first 4 bytes contain the value 9876.

The host system sends		'H ! i' BCC (69$_{hex}$)
The processor acknowledges with	<ACK> '0'	
and sends the data		'H 2 9 8 7 6 z' BCC (7Ah$_{ex}$)

Restart the processor (Quit)

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

Task: Bring the BIS system to the base state.

The host system sends	'Q Q' BCC (51$_{hex}$)
The processor acknowledges with	'Q Q' BCC (51$_{hex}$)

9 Device Function

Query status message

Telegram example: Query the status message with BCC block check.

Task: Query the status in the BIS system after a **read telegram** has just been sent.

The host system sends	'S S' BCC (53hex)
The processor acknowledges with	'S L US' BCC (1Fhex)

Task: Query the status in the BIS system after a **write telegram** has just been sent.

The host system sends	'S S' BCC (53hex)
The processor acknowledges with	'S P ETX' BCC (03hex)

Task: Query the status in the BIS system after a **telegram for switching the read/write head** has just been sent.

The host system sends	'S S' BCC (53hex)
The processor acknowledges with	'S H ESC' BCC (1Fhex)

Task: Query the status of the BIS system. **No telegram** has just been sent.

The host system sends	'S S' BCC (53hex)
The processor acknowledges with	'S _' BCC (20hex)

Appendix

Ordering code

BIS C– 6027 –039–050–06–ST19

Balluff Identification system

Series C Read/Write System

Hardware Type
6027 = Metal enclosure, Ethernet

Software Type
039 = Ethernet with TCP/IP protocol

Version
050 = with two ports for external read/write heads BIS C-3_ _
(except BIS C-350 and -352)

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	for X1	BKS-S 79-00
	for X3	BKS-S 182-00
Cover cap	for Head 1, Head 2, X4	BES 12-SM-2
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 44
Areas of application 9
ASCII Table 45

B

BIS SetIP 16
Block Check BCC 18
Bus connection 11

C

Communication module 10
  Resetting 11
Components 9
COM Port Redirector 19
  Set up virtual COM Port 19
Configuration Software 21
  Dynamic mode 23
  Page size 23
  Protocol type 23
  Start 21
Connections 14
  Ethernet 15
  Power supply 15
  Service port 15
Control function 9
Conventions
  Actions 4
  Enumerations 4
  Notation 4
CRC_16 data check 10, 17
  Checksum 17
  Error message 17
  Initializing 17
CT Data 17
  Send CT Data immediately 17, 22

D

Data
  Electrical 12
  Mechanical 12
Data integrity 9
Description 9
Dimensions 12
Double read 10
Dynamic mode 17

E

EEPROM changing 15
Error numbers 35

F

Function indicators 13, 37
  Diagnostics 37
  Overview 37
  Power-up 37
Function principle 24

I

Installation 14
Intended use 6
Interface information 14
IP address 16

L

LED Indicator 10
  LED 1 10
  LED 2 10

M

MAC address 16
Main components 9

O

Operating conditions 13
Ordering code 44

P

Parameterizing 21
  Basics 21
  CRC_16 data check 23
  Requirements 21
  Start configuration software 21
Processor Connections 14
Product description 9
Protocol sequence 25
  Dialog mode with head select 25
  Direct Read mode 25
Protocol type 18, 23
  Factory settings 18

R

Read/write times 36
  In dynamic mode 36
  In static mode 36

S

Safety 6
  General notes 6
  Warning notes 6
Safety instructions
  Meaning 6
Symbols 4

T

Telegram contents 26
Telegram examples 39
  Find next data carrier (continuous) 42
  Find next data carrier (one time) 42
  Forming the block check BCC 39
  Protocol variants 39
  Query status message 43
  Read a data carrier 40
  Restart the processor (Quit) 42
  Select the read/write head 41
  Write to the data carrier 40
Telegram types 26
  Finding next data carrier (continuous) 31
  Finding next data carrier (one time) 30
  InitializingCRC_16 data check 34
  Meaning of the status messages 33
  Querying the status message 33
  Read/write data carrier 27
  Restarting the processor (Quit) 32
  Selecting the read/write head 29
  Writing a constant value to the data carrier 28
Termination characters 18

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

No. 847527 E • 02.108147 • Edition 0711 • Replaces edition 0703 • Subjects to modification.