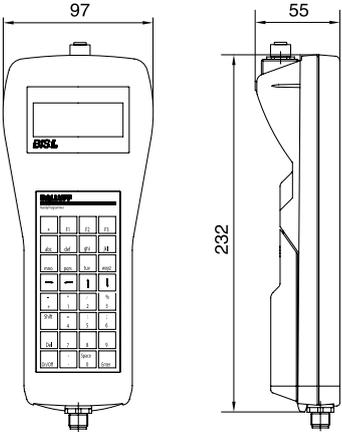


BIS L-81_ Handy Programmer

Technical Description, User's Guide



www.balluff.com

1	Notes to the user	4
1.1	About this manual	4
1.2	Structure of the manual	4
1.3	Conventions	4
1.4	Symbols	5
1.5	Abbreviations	5
2	Safety	6
2.1	Intended use	6
2.2	General safety notes	6
2.3	Meaning of the Warning notes	6
3	Getting Started	7
4	Basic Knowledge	9
4.1	Principle of operation of identification systems	9
4.2	Product description	9
4.3	Control function	9
4.4	Data integrity	9
4.5	Communication over the RS 232 interface	10
4.6	Files	10
5	Technical Data	11
6	Commissioning	13
6.1	Interface information	13
6.2	Battery operation	14
6.4	Turning on the Handy Programmer	15
7	Operation	16
7.1	Keypad and display	16
7.2	Operating concept	16
7.3	Format conversions	16
7.4	Overview of operating menu	17
8	Configuration	18
8.1	Basic knowledge	18
8.2	Configuration menu	19
8.3	Configuration menu 2	22
9	Function of the device	23
9.1	Principle of operationBIS L-81_	23
9.2	Processing data carriers	24
9.3	Working with files	27
9.4	Error messages	29
10	Working over the serial interface	30
10.1	Connecting the Handy Programmer to a PC/Laptop	30
10.2	Protocol sequence	30
10.3	Communication	32
10.4	Error numbers	38
10.5	Read/write times	39
10.6	Telegram examples	40
	Appendix	45

1 Notes to the user

- 1.1 About this manual** This manual describes startup and operation of the Handy Programmer BIS L-81_.
- 1.2 Structure of the manual** The manual is designed so that the sections build on one another.
Section 2: Basic safety information.
Section 3: The main steps for startup.
Section 4: An introduction to the material
Section 5: Technical data for the Handy Programmer.
Section 6: Steps for starting up the Handy Programmer.
Section 7: Notes on operating the Handy Programmer.
Section 8: User-defined settings for the Handy Programmer.
Section 9: Processing data carriers and the use of files.
Section 10: Using the Handy Programmer over the serial interface.
- 1.3 Conventions** The following means of representation are used in this manual.
- Enumerations** Enumerations are represented as a list with bullet points.
– Entry 1,
– Entry 2.
- Actions** Instructions for actions are indicated by a preceding triangle. The result of an action is indicated by an arrow.
▶ Action instruction 1.
⇒ Action result.
▶ Action instruction 2.
- Syntax**
- Numbers:**
– Decimal numbers are represented without additional description (e.g. 123),
– Hexadecimal numbers are indicated with the additional description _{hex} (e.g. 00_{hex}).
- Parameters:**
Parameters are represented in italics (e.g. *CRC_16*).
- Directory paths:**
Descriptions of paths in which data are stored or to be saved are represented in small caps (e.g. PROJECT:\DATA TYPES\USER DEFINED).
- Control characters:**
Control characters are set in arrow brackets (e.g. <ACK>).
- ASCII-Code:**
Characters to be sent in ASCII code are set in apostrophes (e.g. 'L').
- Cross-references** Cross-references indicate where additional information on the topic can be found (see "Technical Data" starting page 11).
- Keypad entries** The key sequence used to open a function or menu from the keypad of the Handy Programmer is represented by key symbols (e.g. **3** **1**) means: First press key 3 and then key 1).

1 Notes to the user

1.4 Symbols

**Attention!**

This symbol indicates a safety instruction which must be observed.

**Note, Tip**

This symbol indicates general notes.

1.5 Abbreviations

BCC	Block Check Character
BIS	Balluff Identification System
CRC	Cyclic Redundancy Code
EEPROM	Electrical Erasable and Programmable ROM
EMC	Electromagnetic Compatibility
PC	Personal Computer
PLC	Programmable Logic Controller

2 Safety

2.1 Intended use

The Handy Programmer BIS L-81_ is a component of the Identification System BIS L. Within the identification system it is used for checking, correcting or initializing data carriers.

The Handy Programmer may be used only for this task in an industrial environment conforming with Class A of the EMC Law.

This description applies to Handy Programmers in the product series

- BIS L-810-0-003...
- BIS L-811-0-003...
- BIS L-812-0-003...

2.2 General safety notes

Installation and Startup

Installation and startup may be carried out only by trained specialists. Damage caused by unauthorized intervention or improper use is not covered under the guarantee and warranty provisions of the manufacturer.

Follow the instructions in this guide exactly when installing and starting the system up (see "Startup" starting page 13).

The Handy Programmer may be operated only with a power supply approved for the device (see "Technical Data" starting page 11).

Operation and Checking

The operator is responsible for observing the locally applicable safety regulations.

In case of defects and non-clearable faults in the identification system, take it out of service and secure against unauthorized use.

2.3 Meaning of the Warning notes



Attention!

The pictogram associated with the word "Attention" warns of a possible hazardous situation affecting the health of persons or resulting in equipment damage. Ignoring these warning notes may result in injury or equipment damage.

- ▶ Always observe the described measures for avoiding the hazard.
-

Getting Started

Interface information

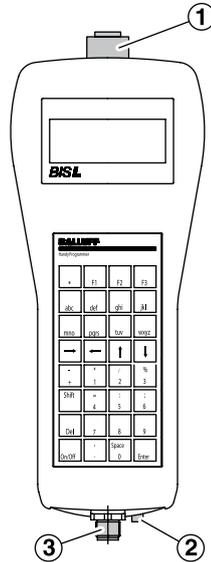


Fig. 1: Handy Programmer connections (example BIS L-811-...)

- 1 X1 - Read head terminal (only BIS L-811-...)
- 2 Charging socket
- 3 X2 - Serial port RS 232

X1 - 8-pin socket (**only BIS L-811-...**)

PIN	Function
1	A
2	Y
3	B
4	+Vs
5	Z
6	-Vs
7	TEST
8	-Vs

X2 - 5-pin terminal

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

Turning on the Handy Programmer

Prerequisites:

- Supply voltage is present (battery inserted and charged).
- For BIS L-811: Read/write head BIS L-3_ _ connected to X1.

Turning on:

- ▶ Press **On** key.
 - ⇒ The display shows the start screen (device model and software version).
 - ⇒ At the same time a battery check is performed.
 - ⇒ If the battery test is successful, the main menu appears on the display.



Note

If the start screen stays on the screen after powering up, the battery voltage is too low. Replace battery.

3 Getting Started

Operating menu overview

Menu structure

Key combination

Function

Main menu

— Read	[1]	Read data carrier
— Write	[2]	Write to data carrier
— Edit	[3]	Edit data
— File	[4]	
— Load	[4][1]	Load file
— Save	[4][2]	Save file in EEPROM
— Data transmission	[4][3]	Data exchange over the serial interface
— Delete	[4][4]	Delete files
— One entry	[4][4][1]	
— All entries	[4][4][2]	
— Directory	[4][5]	Show directory

— Configuration 1

— Addresses		Address ranges for reading and writing data carriers and for specifying the insertion point address for editing.
— Read	[5][1]	
— Start/End address	[5][1][0]	
— Start/Number	[5][1][1]	
— Write	Read + [ENTER]	
— Start/End address	[0]	
— Start/Number	[1]	
— Edit	Write + [ENTER]	
— End address	[0]	
— Cursor position	[1]	
— Data format	[5][2]	Data format preset (HEX, BCD, ASCII, BIN)
— Carrier data	[5][3]	Data carrier information (data carrier capacity, CRC)
— Serial interface	[5][4]	Parameterize interface

— Configuration 2

— Language	[5][6][1]	Language preset (German, English)
------------	-----------	-----------------------------------

Main menu 2

— Initmenu		
— CRC	[6][1][1]	Prepare data carrier for operation with CRC
— Read only tag	[6][1][2]	Convert data carrier (only BIS L-10_-05/L)
— with CRC	[6][1][2][1]	
— without CRC	[6][1][2][2]	
— Serial Number	[6][2]	Read serial number

4 Basic knowledge

4.1 Principle of operation of identification systems

The BIS L portable identification system is categorized as a non-contacting system with read and write function. This allows it to not only transport fixed programmed information to the data carrier, but also to collect and send current information.

The main components of the BIS L portable identification system are:

- Handy Programmer,
- Read/write heads,
- Data carriers.

Key areas of application are:

- In manufacturing for controlling material flow (e.g. in variant-specific processes, in workpiece transport with conveying lines, for acquiring safety-related data),
- In tool coding and monitoring,
- In process equipment organization,
- In warehousing for monitoring stock movement and levels,
- In transport and conveying technology,
- In waste disposal for quantity-based data collection.

4.2 Product description

Handy Programmer BIS L-81_:

- Plastic enclosure,
- Reading/writing using integrated antenna (BIS L-810, BIS L-812) or by connecting a read/write head (BIS L-811),
- System components powered by 2.4 V rechargeable NiMH battery pack,
- Editor function for editing the data to be read or written,
- Data carrier conversion possible (for BIS L-10_-05/L),
- File storage.

4.3 Control function

The Handy Programmer is the link between data carrier and host system. The Handy Programmer can be used to check data carriers, correct them or initialize them. The data read on the system can be stored in the Handy Programmer and later sent to the host system.

Host systems may include:

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

4.4 Data integrity

To ensure data integrity, a check procedure must be used to monitor data transfer between data carrier and Handy Programmer.

The default setting for the Handy Programmer is double reading followed by a comparison.

Alternately a CRC_16 data check may be used.

Here a check code is written to the data carrier, which allows data to be checked for validity at any time.

Which procedure is used depends on how the identification system is used.



Note

Mixing of both check procedures is not possible!

4 Basic knowledge

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

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

4.5 Communication over the RS 232 interface

The Handy Programmer and the host system (PC/PLC) communicate over an RS 232 interface. The data are sent using fixed telegrams.

The following possibilities exist:

- The Handy Programmer is used to process read/write jobs from the PC. The Handy Programmer assumes the function of a BIS processor.
- The Handy Programmer manages files and exchanges stored data with the PC.

4.6 Files

The BIS L-81_Handy Programmer has a non-volatile EEPROM memory. This memory can be used to store data read from a data carrier under a file name.

- The file name is limited to 8 characters.
- 15 files having a maximum size of 192 bytes per file can be stored.

File format

Start address of the data on the data carrier Number of bytes Data carrier data

4 bytes	4 bytes	max. 184 bytes
---------	---------	----------------

Example: Data carrier file starting at address 75 with 16 bytes of data in ASCII format.
Data: ABCDEFGHIJKLMNOP

Contents of data carrier file: 00750016ABCDEFGHIJKLMN

5 Technical Data

Dimensions

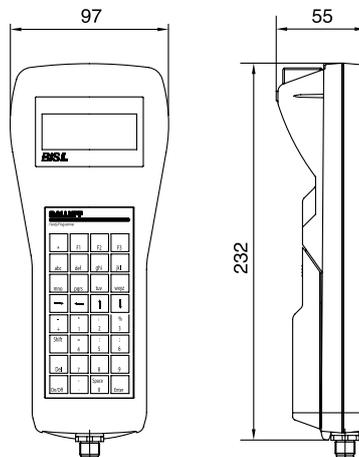


Fig. 2: Dimensions BIS L-810 (in mm)

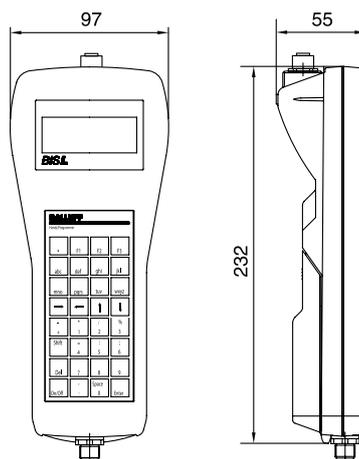


Fig. 3: Dimensions BIS L-811 (in mm)

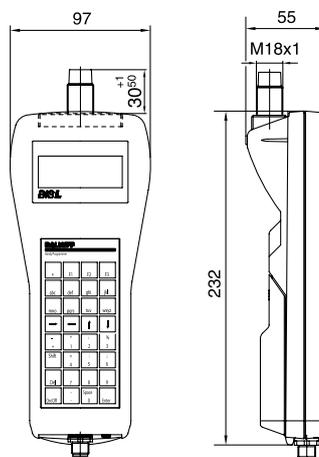


Fig. 4: Dimensions BIS L-812 (in mm)

5 Technical Data

Housing	Housing material	Plastic ABS
	Weight (incl. battery)	approx. 950 g
	Keypad	32 keys, 4 x 8 mm, alphanumeric and special characters
	Display	LCD-Display, 80 characters / 4 lines
	Enclosure rating	IP40 (when closed)

Connections	X2 - Serial port RS 232	RS 232 - 5-pin terminal	
	Read head terminal	BIS L-810	Internal antenna
		BIS L-811	8-pin socket (X1)
		BIS L-812	Remote antenna M18
Charging socket	2.5 mm jack, suitable for BIS C-701-A		

Electrical data	Operating voltage V_s		24 V DC $\pm 10\%$	
	Current consumption	normal operation	approx. 65 mA	
		Read/write	approx. 275 mA	
	Voltage and current supply	Type	2.4 V rechargeable battery pack NiMH	
		Capacity	1650 mAh	
		max. charging voltage	5-10V, internally limited	
		Charging current	approx. 400 mA	
Charging time		approx. 4 h		

Operating conditions	Ambient temperature range	0 °C ... 50 °C
	EMC	
	<ul style="list-style-type: none"> - EN 61000-4-2/3/4/5/6 - EN 55011 	<ul style="list-style-type: none"> - Severity level 4A/3A/4A/1A/3A - Gr. 1, Cl. A
Shock/vibration	EN 60068 Part 2-6/27/29/64/32	

6 Commissioning

6.1 Interface information

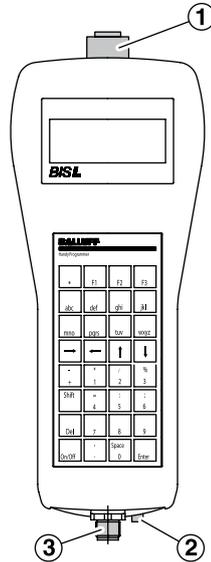


Fig. 5: Handy Programmer connections (example BIS L-811-...)

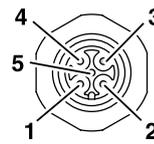
- 1 X1 - Read head terminal (only BIS L-811-...)
- 2 Charging socket
- 3 X2 - Serial port RS 232

X1 - 8-pin socket (only BIS L-811-...)



PIN	Function
1	A
2	Y
3	B
4	+Vs
5	Z
6	-Vs
7	TEST
8	-Vs

X2 - 5-pin terminal



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

1) n.c.: do not connect

6 Commissioning

6.2 Battery operation The system components of the Handy Programmer are powered by a 2.4V NiMH rechargeable battery pack.

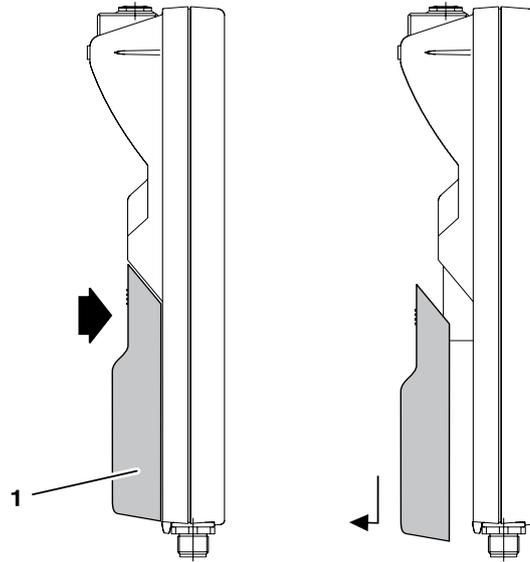


Fig. 6: Removing the battery

1 Battery

Remove battery

- ▶ Press latch.
- ▶ Slide battery down a short distance until the guide releases the battery.

Install battery

- ▶ Insert battery in the guide and slide up until it latches.

Charge battery



Attention!

Components can be damaged by using the wrong charging current.

- ▶ Use only chargers recommended by Balluff for charging the battery.

Battery being charged	Charger
Installed	BIS C-701/A
Removed	BIS C-702/A

6 Commissioning

6.4 Turning on the Handy Programmer

Prerequisites:

- Supply voltage is present (battery inserted and charged).
- For BIS L-811: Read/write head BIS L-3_ _ connected to X1.

Turning on:

- ▶ Press **On** key.
 - ⇒ The display shows the start screen (device model and software version).
 - ⇒ At the same time a battery check is performed.
 - ⇒ If the battery test is successful, the main menu appears on the display.

**Note**

If the start screen stays on the screen after powering up, the battery voltage is too low. Replace battery.

Possible error messages when powering up:

Message	Problem
R/W head error.	No BIS L-3_ _ read/write head connected.
Battery voltage too low.	Battery voltage has dropped below a nominal value. The device can only be used for a short time longer. Charge the battery as soon as possible!

7 Operation

7.1 Keypad and display

The Handy Programmer is operated from a membrane keypad with 32 keys.

Multiple key assignments for keypad

The letter keys and some functions keys have multiple functions.

Key type	Configuration	Switch
Function keys	2 different functions	SHIFT
Letter keys	3 letters capital and small	Press 1x: 1st letter Press 2x: 2nd letter Press 3x: 3rd letter SHIFT : Capital letters

Navigation

The LCD has 4 lines with 20 characters each. Use the keys **↑** **↓** to navigate between the lines and the keys **→** **←** to navigate between the characters.

The device functions are menu-driven.

7.2 Operating concept

The main functions are shown in the main menu. Each sub-menu is indicated by a number. Pressing the corresponding number key on the keypad opens the sub-menu in question.



Note

For faster orientation the key sequence for reaching a menu or a function is always indicated in the Manual.

An overview of the menus structure can be found on [page 17](#).

7.3 Format conversions

Data which have been read or which are to be written can be displayed in various formats.

Data

- ▶ Select the display format by pressing the key **A** or **G**.

Key	Display format	Remark
A	ASCII --> BCD --> BINARY--> ASCII	Sequential invoking of the data formats by repeated pressing.
G	Hex	Toggling between the currently set format and hexademical format.

Language

From the main menu you can switch between the languages German and English.

- ▶ For language selection hold down the **SHIFT** + **F1** keys at the same time.



Note

The format conversions are only temporary until the device is turned off. The default setting is made in the configuration menu (see "Configuration" starting [page 18](#)).

7 Operation

7.4 Overview of operating menu

Menu structure	Key combination	Function
Main menu		
— Read	[1]	Read data carrier
— Write	[2]	Write to data carrier
— Edit	[3]	Edit data
— File	[4]	
— Load	[4][1]	Load file
— Save	[4][2]	Save file in EEPROM
— Data transmission	[4][3]	Data exchange over the serial interface
— Delete	[4][4]	Delete files
— One entry	[4][4][1]	
— All entries	[4][4][2]	
— Directory	[4][5]	Show directory
— Configuration 1	[5]	
— Addresses		Address ranges for reading and writing data carriers and for specifying the insertion point address for editing.
— Read	[5][1]	
— Start/End address	[5][1][0]	
— Start/Number	[5][1][1]	
— Write	Read + [ENTER]	
— Start/End address	[0]	
— Start/Number	[1]	
— Edit	Write + [ENTER]	
— End address	[0]	
— Cursor position	[1]	
— Data format	[5][2]	Data format preset (HEX, BCD, ASCII, BIN)
— Carrier data	[5][3]	Data carrier information (data carrier capacity, CRC)
— Serial interface	[5][4]	Parameterize interface
— Configuration 2	[5][6]	
— Language	[5][6][1]	Language preset (German, English)
Main menu 2		
— Initmenu		
— CRC	[6][1][1]	Prepare data carrier for operation with CRC
— Read only tag	[6][1][2]	Convert data carrier (only BIS L-10_-05/L)
— with CRC	[6][1][2][1]	
— without CRC	[6][1][2][2]	
— Serial Number	[6][2]	Read serial number

8 Configuration

8.1 Basic knowledge

Three types of data carriers are available for use with the BIS L-81_. Depending on your selection either all or only one particular data carrier can be processed.

Data carrier types

Data carrier	Data
BIS L-10_-01/L	192 bytes of user data (read/write) + 4 bytes of fixed serial number (read-only).
BIS L-10_-05/L	192 bytes of user data (read/write) + 4 bytes of fixed serial number (read-only). Or after data carrier conversion: 5 bytes of fixed serial number (read-only), corresponding to the user data.
BIS L-20_-03/L	5 bytes of fixed serial number (read-only), corresponding to the user data.

Notes on data carriers

- Type BIS L-10_-01/L data carriers are shipped configured with FF_{hex}37_{hex}. Only data carriers having this configuration are processed.
- The BIS L-10_-01/L carrier contains additional memory ranges for configuration and protected data. These areas cannot be processed using the BIS L-81_ Handy Programmer.
- When using BIS L-10_-05/L data carriers the data distribution can be changed only once by the Handy Programmer (data carrier conversion).

CRC Check

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

Initializing

To be able to use the CRC check, the data carriers must be initialized. There are two ways of doing this:

- Initializing using the "Init" function in main menu 2 (see page 26).
- Initializing data carriers using the command code 'Z' (see page 35).

If the data carrier does not contain the correct CRC when reading or writing, the processor sends the error message 'CRC-Error'.

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

Error message

- If an error message is the result of a failed write attempt, 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 attempt, then one or more of the memory cells in the data carrier is likely defective. The data carrier must be replaced.

Checksum

The checksum is written to the data carrier as 2 bytes of information. 2 bytes per block are taken up by this information. This leaves 14 bytes per block available. The usable number of bytes can be looked up in the following table.

Data carrier	Memory capacity	Usable bytes for CRC_16
BIS L-10_-01/L	192 Byte	168 Byte
BIS L-10_-05/L	192 bytes or 5 Byte	168 bytes or 3 Byte
BIS L-20_-03/L	5 Byte	CRC_16 is not supported

8 Configuration

8.2 Configuration menu
5

If you do not wish to use the factory setting, the settings must be adjusted in the configuration menu.

- ▶ From the main menu press the 5 key.
 ⇒ Changing to Configuration Menu 1.

Configure addresses
5 1

The address ranges for reading and writing data carriers and for the insertion address for editing can be preset.

i Note
 Applies only to BIS L-10_-01/L and BIS L-10_-05/L (not converted).

Input mode	Address range		
	Read	Write	Edit
Start address (starting at byte number): End address (up to byte number):	0000 0191	0000 0191	
Start address (starting at byte number): Number of bytes	0000 0192	0000 0192	
End address: Insertion address:			0191 0000

Input mode and address range for reading

- ▶ From the main menu press the 5 and 1 keys in sequence.
 ⇒ Changing to the submenu "Input mode: READ".
- ▶ Select input mode (Press 0 or 1).
- ▶ Confirm entry by pressing ENTER.
 ⇒ Changes to input screen "READ range".
- ▶ Enter value for the start address and confirm with ENTER.
- ▶ Enter value for the end address or number of bytes to read and confirm with ENTER.
 ⇒ Changing to submenu "Input mode: WRITE".

```

Input mode: READ
-----
0 -> Start/End addr.
1 -> Start/Number _
```

```

READ range
-----
at byte no.:
to byte no.:
```

8 Configuration

Input mode and address range for reading

- ▶ Select input mode (Press **[0]** or **[1]**).
- ▶ Confirm entry by pressing **[ENTER]**.
⇒ Changes to input screen "WRITE range".

```
Input mode: WRITE
-----
0 -> Start/End addr.
1 -> Start/Number _
```

- ▶ Enter value for the start address and confirm with **[ENTER]**.
- ▶ Enter value for the end address or number of bytes to read and confirm with **[ENTER]**.
⇒ Changes to input screen "Edit range".

```
WRITE range
-----
at byte no.:
to byte no.:
```

End address and cursor position for editing

- ▶ Enter value for the end address (highest address to be edited) and confirm with **[ENTER]**.
- ▶ Enter value for the cursor position (= insertion address) and confirm with **[ENTER]**.
⇒ Changes to Configuration Menu 1.

```
Edit range
-----
End address:
Cursor position:
```

Select format

[5] [2]

The data format can be preset. 4 formats are available:

- HEX: Hexadecimal
- BCD: Binary Coded Decimal
- BIN: Binary
- ASCII.

- ▶ From the main menu press the **[5]** and **[2]** keys in sequence.
⇒ Changes to submenu "Data format".
- ▶ Select data format using the keys **[1]**, **[2]**, **[3]** or **[4]**.
- ▶ Confirm entry by pressing **[ENTER]**.
⇒ Changes to Configuration Menu 1.

```
--- Data format ---
1 HEX      3 ASCII
2 BCD
Current format:
```

Special characters in BCD format

When BCD format is selected the submenu "Special characters" appears in which the characters A_{hex} to F_{hex} can be converted to represent special characters.

Factory default BCD special characters:

```
Special characters
'A' : *   'D' : -
'B' : +   'E' : .
'C' : ,   'F' : /
```

8 Configuration

Data carrier specifications

5 3

The settings for capacity of the data carrier and for data transmission with/without CRC are stored in the Handy Programmer.



Note

Applies only to BIS L-10_-01/L and BIS L-10_-05/L (not converted).

- ▶ From the main menu press the 5 and 3 keys in sequence.
⇒ Changes to submenu "Carrier data".
- ▶ Enter data carrier capacity and confirm with ENTER .
- ▶ Set CRC (Y = Yes; N = No). Confirm entry by pressing ENTER .
⇒ Changes to Configuration Menu 1.

```

-- Carrier Data --
Capacity:
CRC:
    
```

Parameterize interface (V.24)

5 4

Setting the serial interface (RS 232) for communication with an external computer.

Parameter	Possible values	Default
Transmission rate	9600; 19200; 38400; 57600 baud	9600 baud
Stop bits	1, 2	1
Data bits	8 (cannot be changed)	8
Parity	E (even); O (odd); N (none)	e
BCC/CR	B (BCC); C(CR)	B

- ▶ From the main menu press the 5 and 4 keys in sequence.
⇒ Changes to submenu "Serial port".
- ▶ Enter the parameter values and confirm with ENTER .
⇒ After confirming the parameter BCC/CR, changes to Configuration menu 1.

```

--- Serial port ---
Baud:      Data  :
Stop:      Parity:
           BCC/CR:
    
```

8 Configuration

8.3 Configuration menu 2

[5] [6]

Set language

[5] [6] [1]

The menu language and configuration are stored in Configuration menu 2.

- ▶ From the main menu press the **[5]**, **[6]** and **[1]** keys in sequence.
⇒ Changes to submenu "Sprache - Language".

```
-- Configuration --  
1 Language  
2 <--
```

- ▶ Select language (**[1]** or **[2]**) key.
⇒ Changes to Configuration Menu 2.

```
Sprache - Language  
1 Deutsch  
2 Englisch
```

Save configuration

[5] [6] [2]

When exiting the configuration menu, you are prompted whether you want to save the configuration settings.

- ▶ "Y" to save the configuration settings and confirm with **[ENTER]** .
⇒ Changes to Main menu.

9 Function of the device

9.1 Principle of operation BIS L-81_

The Handy Programmer keypad is used to enter data commands (Read, Write, Edit) and configure data exchange with the data carrier.

The location of the read/write heads for data exchange depends on the model:

Type	Data exchange via
BIS L-810-...	Integrated antenna, in housing
BIS L-811-...	External read/write head using 8-pin socket
BIS L-812-...	Integrated antenna, remote

Individual data, files or the complete file directory can be exchanged.

Using as a processor

In addition to keypad-driven processing of the data carriers, it is also possible to carry out read and write operations from a PC/laptop through the serial interface of the Handy Programmer. The Handy Programmer then works like a BIS processor.



Note

Information about how the Handy Programmer works on the serial interface can be found in [Section 10 starting page 30](#).

Setup with the Handy Programmer

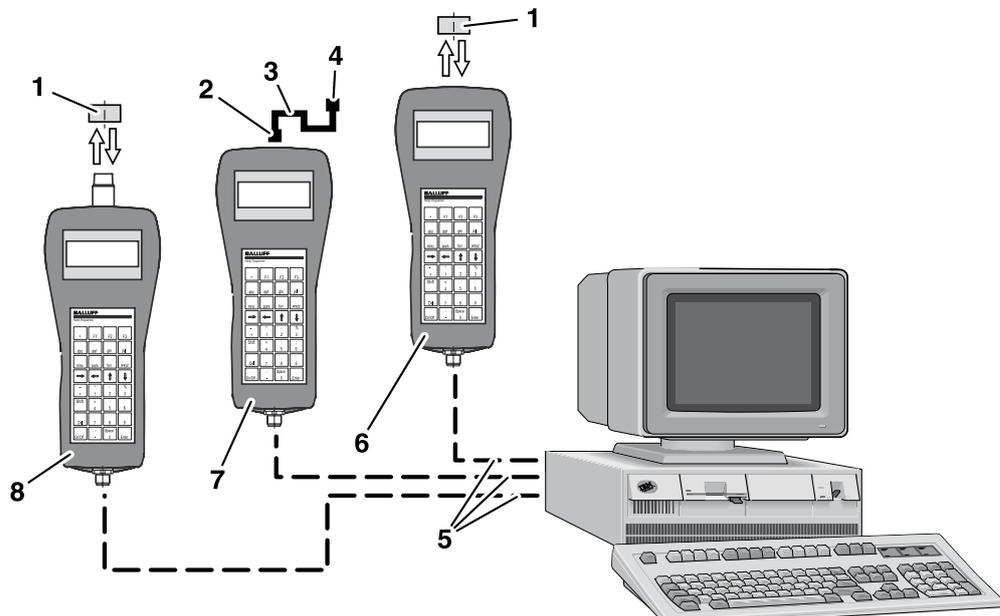


Fig. 7: Setup of BIS L-810, BIS L-811 and BIS L-812

- 1 Data carrier BIS L-1_ _
- 2 Plug for read/write head BIS L-3_ _
- 3 Connecting cable for read/write head BIS L-3_ _
- 4 Read/write head BIS L-3_ _
- 5 Interface cable BIS C-522-PVC-02/03
- 6 BIS L-810 Handy Programmer
- 7 BIS L-811 Handy Programmer
- 8 BIS L-812 Handy Programmer

9 Function of the device

9.2 Processing data carriers

i Note
Applies only to BIS L-10_-01/L and BIS L-10_-05/L (not converted).

Read data carrier
1

- ▶ From the main menu press the **1** key.
⇒ Changes to the submenu "Read data carrier".

```

READ data carrier
-----
at byte no.:
to byte no.:
    
```

i Note
The appearance of the menu depends on the input mode (see "Configure addresses" starting page 19).

Displayed default values:

- The values preset in the configuration when the device is first powered up,
- The last entered values as long as the device has not been turned off.

- ▶ Confirm default values in sequence with **ENTER**.
- or:
- ▶ Changing default values. For editing use the keys **↑** and **↓** to switch between the input fields. Apply changes by pressing **ENTER**.
⇒ Read operation is started (flashing black bar).

```

READ data carrier
-----
at byte no.:
to byte no.:
    
```

i Note
The data carrier must remain within the read zone while being read.
For possible error messages see "Error messages" on page 29.

After successful reading the data are displayed in the preset data format (for setting the data format see "Operation" on page 16).

```

Display 0000 - 0015
-----
Balluff GmbH
    
```

i Note
For multi-page displays navigate using the **↑** and **↓** keys.

- ▶ Press **ENTER**.
⇒ Changes to Main menu.

Write to data carrier
2

- ▶ From the main menu press the **2** key.
⇒ Changes to the submenu "Write data carrier".

```

Write data carrier
-----
at byte no.:
to byte no.:
    
```

i Note
The appearance of the menu depends on the input mode (see "Configure addresses" starting page 19).

Displayed default values:

- The values preset in the configuration when the device is first powered up,
- The last entered values as long as the device has not been turned off.

9 Function of the device

- ▶ Confirm default values in sequence with **ENTER** .
or:
- ▶ Changing default values. For editing use the keys **↑** and **↓** to switch between the input fields. Apply changes by pressing **ENTER** .
⇒ Write operation is started (flashing black bar).

```
WRITE data carrier
-----
at byte no.:
to byte no.:
```

i Note
The data carrier must remain within the read zone while being written to.
For possible error messages see "Error messages" on page 29.

After successful writing the display reverts to the main menu.

Edit data **3**

- ▶ From the main menu press the **3** key.
⇒ Changes to the menu for specifying the insertion address (cursor position).
- ▶ Enter the address starting at which you want to edit. Confirm entry by pressing **ENTER** .
⇒ Changes to Edit menu.
- ▶ Edit insertion address (value range 0000-0191).

```
edit cursor position
-----
at byte no.
```

The data are shown in the preset data format see "Operation" on page 16).

```
Edit Byte No.
-----
```

Navigation for editing:

- **→**: Go to the next character (HEX: next half-byte).
- **←**: Go to the previous character (HEX_ previous half-byte).
- **↑**: One line higher.
- **↓**: One line lower.
- ▶ Exit editing: Press **ENTER** .
⇒ Changes to Main menu.

Initialize data carrier **6 1 1**

The data carrier is programmed with 00_{hex} and thereby prepared for operation with CRC.

- ▶ From the main menu press the **6** key.
⇒ Changes to Main menu.2
- ▶ From the main menu 2 press the **1** key.
⇒ Changes to the Init menu.
- ▶ From the Init menu press the **1** key.
⇒ Changes to the menu "Write data carrier".
- ▶ Press **ENTER** .
⇒ The data carrier is programmed with 00_{hex} .
⇒ Changes to the Init menu.

```
---- Initmenu ----
1 CRC
2 read only tag
6 <--
```

Convert data carrier

6 1 2

Type BIS L-10_-05/L data carriers can be converted so that their properties match those of the BIS L-20_-... data carrier type.



Attention!

The data carrier can only be converted once. The data carrier conversion cannot be reversed.

Type BIS L-10_-05/L data carriers can be initialized for data transmission with or without checksum. Depending on the selection, transmission of the serial number can be done in one of the following two ways:

- BIS L-10_-05/L with CRC: 3 bytes,
- BIS L-10_-05/L without CRC: 5 bytes.

- ▶ From the main menu press the **6** key.
⇒ Changes to Main menu.
- ▶ From the main menu 2 press the **1** key.
⇒ Changes to the Init menu.
- ▶ From the Init menu press the **2** key.
⇒ Changes to the menu "read only tag".
- ▶ From the "read only tag" menu select the option with or without CRC (**1** or **2** key).
⇒ Changes to the submenu "read only tag" for entering the serial number.
- ▶ In the submenu "read only tag" enter the 3 bytes (with CRC) or 5 bytes (without CRC) of the serial number.
- ▶ Press **ENTER**.
⇒ The Handy Programmer sends the data for data carrier conversion (flashing black bar on the display).
- ▶ Bring type BIS L-10_-05/L into the active zone of the read/write head.
⇒ The data carrier is converted.
⇒ The serial number of the converted data carrier is displayed.
- ▶ Press **ENTER**.
⇒ Changes to the Init menu.

```

---- Initmenu ----
1 CRC
2 read only tag
6 <--
    
```

```

--- read only tag ---
1 with CRC
2 without CRC
6 <--
    
```

```

--- read only tag ---
00 00 00 00 00 hex
Del <--
    
```



Note

An already converted data carrier cannot be converted again. In this case the Handy Programmer sends an error message (see "Error messages" on page 29.)

9 Function of the device

Read serial number

[6] [2]

Reading out the serial number:

- BIS L-10_-01/L: 4 bytes
- BIS L-10_-05/L converted, with CRC: 3 bytes
- BIS L-10_-05/L converted, without CRC: 5 bytes
- BIS L-20_-03/L: 5 bytes

- ▶ From the main menu press the [6] key.
⇒ Changes to Main menu.
- ▶ From the main menu 2 press the [2] key.
⇒ Changes to the menu "READ Serial number".
- ▶ Press [ENTER].
⇒ The serial number is read and then displayed in HEX format.
- ▶ Press [ENTER].
⇒ Changes to Main menu.

```

READ Serial Number
-----
    
```

9.3 Working with files

The BIS L-81_ Handy Programmer has a non-volatile EEPROM memory. This memory can be used to store data read from a data carrier under a file name.

Open file menu

[4]

- ▶ From the Main menu press the [4] key.
⇒ Changes to File menu.

Load file

[4] [1]

Loading a file stored in EEPROM memory to the working memory of the Handy Programmer.

- ▶ From the File menu press the [1] key.
- ▶ Enter the name of the file to be loaded and press the [ENTER] key.
⇒ The start address and end address or number of bytes of the loaded file are displayed.
- ▶ Press [ENTER].
⇒ Changes to File menu.

```

--- File menu ---
1 Load      4 Delete
2 Save      5 Directory
3 Trans.    6 End
    
```

Save file

[4] [2]

Storing data located in the working memory of the Handy Programmer to a file in the EEPROM.

- ▶ From the File menu press the [2] key.
- ▶ Enter the name of the file to be saved and press the [ENTER] key.
⇒ Changes to File menu.

```

Please enter name:
    
```

9 Function of the device

Data exchange

4 3

Exchanging data between Handy Programmer and laptop/PC over the serial interface.



Note

For details on data exchange over the serial interface see "[Working over the serial interface](#)" starting page 30.

Delete files

4 4

- ▶ From the File menu press the 4 key.
- ▶ In the Deletion menu select the type of deletion.

1 Deleting an entry.

- ▶ Enter file name and confirm with ENTER .
⇒ The file is deleted.

2 Deleting all entries.

- ⇒ All entries are immediately deleted (no prompt).
- ⇒ Changes to File menu.

```
--- Deletion menu ---  
1 = One entry  
2 = All entries  
6 <---
```

Show directory

4 5

Display the files stored in the EEPROM.

- ▶ From the File menu press the 5 key.
⇒ The list of stored files is displayed.
- ▶ Press ENTER .
⇒ Changes to File menu.

9 Function of the device

9.4 Error messages Possible error messages on the device display and their meanings are listed in the following table.

Error message	Possible cause	Remedy
Read error Function cancelled	– Data carrier removed too soon from the read/write zone.	▶ Acknowledge error message by pressing ENTER . ⇒ Changes to Main menu.
Write error Function cancelled	– Distance between data carrier and read/write head too great.	▶ Reduce distance. ▶ Read data carrier again.
Read data carrier ???Distance???	Data carrier cannot be read or written.	▶ Reduce distance. ▶ Check configuration. ▶ Replace data carrier. ▶ Read data carrier again.
Write data carrier ???Distance???	– Distance between data carrier and read/write head too great. – End address or start address + number of bytes lies outside the data carrier capacity. – Data carrier defective.	
Maximum end address exceeded	– End address is greater than the configured capacity of the data carrier.	▶ Acknowledge error message by pressing ENTER . ⇒ Changes to Main menu.
Read error Wrong carrier	– Data carrier configuration in the Handy Programmer does not agree with that of the data carrier.	▶ Change configuration. ▶ Exchange data carrier.
Write error Wrong carrier		
Read error Distance/others	– Read/write operation failed the 3rd time.	▶ Acknowledge error message by pressing ENTER . ⇒ Changes to Main menu. ▶ Restart read/write operation.
Write error Distance/others		

10 Working over the serial interface

10.1 Connecting the Handy Programmer to a PC/Laptop

The serial port can be used to connect the Handy Programmer to a PC/laptop and be operated like a BIS processor.

The following functions can be carried out over the serial interface:

- Read and write operations from a PC or laptop
- Sending files between the Handy Programmer and PC/laptop.

Preparation

- ▶ Use the BIS C-522 cable to connect the Handy Programmer to the PC/laptop see "Accessories" on page 45).
- ▶ To work with files, install the BISHandy.exe program on the PC/laptop.



Note

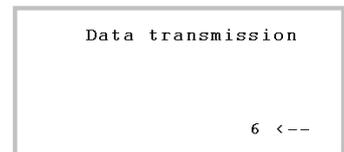
You can find the BISHandy.exe program on the supplied BIS-CD.

Data transmission

[4] [3]

- ▶ From the main menu press the [4] key.
⇒ Changes to File menu.
- ▶ From the File menu press the [3] key.
⇒ Data transmission is activated.

To end data transmission and return to the File menu:
Press [6] .

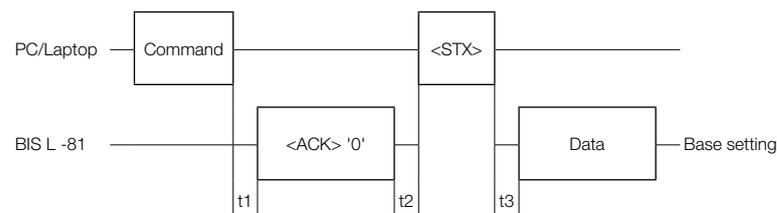


10.2 Protocol sequence

Read

The Handy Programmer and PC/laptop communicate with each other using a fixed protocol.

1. No error occurs:

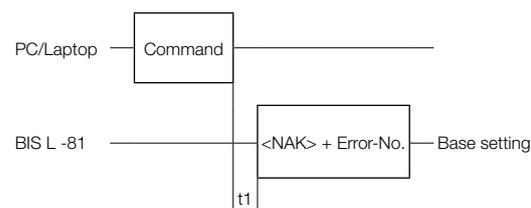


t1 Depends on the number of bytes to be read (see "Read times" on page 39)

t2 ≥ 0 (is not monitored by the Handy Programmer)

t3 = max. 50 ms

2. An error occurs:

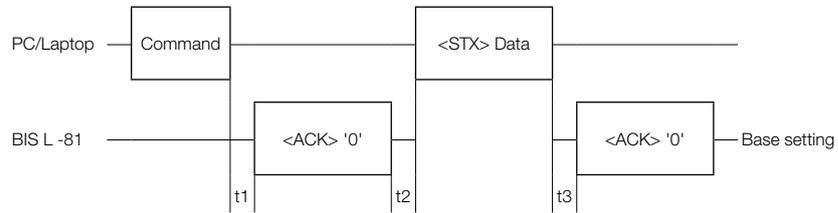


t1 Depends on the number of bytes to be read (see "Read times" on page 39) and error type (recommended monitoring time 15 s)

10 Working over the serial interface

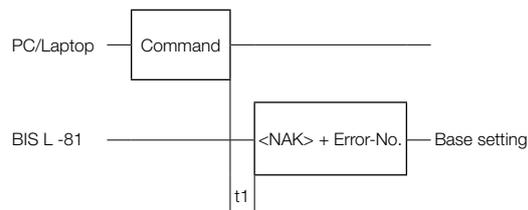
Write

1. No error occurs:



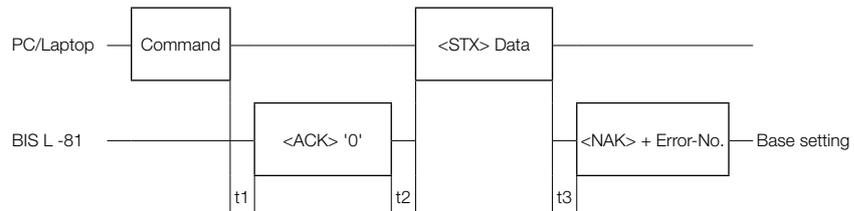
t1 = max. 50 ms
 t2 ≥ 0 (is not monitored by the Handy Programmer)
 t3 Depends on the number of bytes to be written (see "Write times" on page 39)

2. There is an error in the command:



t1 = max. 50 ms

3. There is a write error:



t1 = max. 50 ms
 t2 ≥ 0 (is not monitored by the Handy Programmer)
 t3 Depends on the number of bytes to be written (see "Write times" on page 39) and error type (recommended monitoring time 30 s)

i Prerequisites for validity of the displays:

- The Handy Programmer is in the base setting.
- There is a data carrier in front of the read/write head.

10 Working over the serial interface

10.3 Communication

The host system and Handy Programmer communicate with each other via telegrams. There are specific telegrams for the individual operations. These always start with the command which is associated with the telegram type.

Telegram types with associated command (ASCII characters)

'L'	Read the data carrier
'P'	Write to the data carrier
'C'	Write a constant value to the data carrier
'Q'	Restart the Handy Programmer - Quit
'Z'	Initialize data carrier for CRC_16 data check
'T'	Send file to the Handy Programmer
'G'	Load file from the Handy Programmer
'D'	Load file directory from the Handy Programmer
'E'	Delete a particular file in the Handy Programmer
'K'	Delete all files in the Handy Programmer



Note

Continuous querying of the interface is not allowed. The wait time between two commands must be at least 300 ms.
With telegram types 'T', 'G', 'D', 'E' and 'K' only BCC is used for data integrity (not depending on the configuration setting).

Explanation of telegram contents

Start address and number of bytes The start address (A3, A2, A1, A0) and the number of bytes to be sent (L3, L2, L1, L0) are transmitted as ASCII characters. The range between 0000 and 191 can be used for the start address and 0001 to 192 for the number of bytes. A3 ... L0 stand for one ASCII character each.



Note

The sum of the start address and the number of bytes may not exceed the usable data carrier capacity.

The acknowledgement <ACK> '0' is sent by the identification system when the serially transmitted characters were recognized as correct and there is a data carrier within the active zone of a read/write head.
<NAK> + 'Error-No.' is acknowledged if an error was recognized or if there is no data carrier within the active zone of a read/write head.

Start <STX> starts data transmission.

Sent bytes The data are sent code-transparent (no data conversion).

Telegram terminator

The factory default setting is Blockcheck BCC.
The commands for reading and writing can also be terminated with Carriage Return 'CR' depending on the configuration. The file commands must always be terminated with Blockcheck BCC.

Command	Terminator
'L', 'P', 'C', 'Q', 'Z'	BCC or 'CR'
'T', 'G', 'D', 'E', 'K'	always BCC

The Handy Programmer likewise terminates all sent telegrams using the selection Terminator type.

10 Working over the serial interface

Read/Write Data Carrier telegram

Reading from the data carrier, writing to the data carrier.

Task	Data flow	Command	Start address of the first byte to be sent	Number of bytes to be sent	Terminator	Acknowledgement 2)	Start of sending	Data 3)	Terminator	Acknowledgement 2)
Read	To HP 4)	'L'	A3 A2 A1 A0 '0 0 0 0' BIS '0 1 9 1'	L3 L2 L1 L0 '0 0 0 1' BIS '0 1 9 2'	'1' '0' or CR		<STX>			
	From HP 5)					<ACK> '0' or <NAK> + Error No.		D1 D2 D3...Dn BCC or 'CR'		
			1)							
Write	To HP 4)	'P'	A3 A2 A1 A0 '0 0 0 0' BIS '0 1 9 1'	L3 L2 L1 L0 '0 0 0 1' BIS '0 1 9 2'	'1' '0' or CR		<STX>	D1 D2 D3...Dn BCC or 'CR'		
	From HP 5)					<ACK> '0' or <NAK> + Error No.				<ACK> '0' or <NAK> + Error No.
			1)					1)		

- 1) The Quit command is not permitted at this point.
- 2) The acknowledgement is '0' if no error was detected, or <NAK> + 'Error No.', if an error occurred.
- 3) Data from start address to start address + number of bytes.
- 4) From host system to Handy Programmer.
- 5) From Handy Programmer to host system.



Note

Examples of telegrams can be found in [Section 10.6 starting page 40](#).

10 Working over the serial interface

Telegram: Write a constant value to the data carrier

Write a constant value to the data carrier.

This command can be used to delete a data carrier. The time for sending the bytes to be written is saved.

Task	Data flow	Command	Start address of the first byte to be sent	Number of bytes to be sent	Terminator	Acknowledgement 2)	Start of sending	Data 3)	Terminator	Acknowledgement 2)
Write	To HP 4)	'C'	A3 A2 A1 A0 '0 0 0 0' BIS '0 1 9 1'	L3 L2 L1 L0 '0 0 0 1' '1' '0' BIS '0 1 9 2'	BCC or 'CR'		<STX>	d	BCC or 'CR'	
	From HP 5)					<ACK> '0' or <NAK> + Error No.				<ACK> '0' or <NAK> + Error No.
				1)					1)	

- 1) The Quit command is not permitted at this point.
- 2) The acknowledgement is <ACK> '0' if no error was detected, or <NAK> + 'Error No.', if an error occurred.
- 3) Data from start address to start address + number of bytes.
- 4) From host system to Handy Programmer.
- 5) From Handy Programmer to host system.

Telegram: Restart the Processor (Quit)

Sending the Restart command (Quit) cancels any telegram currently in progress. The Handy Programmer goes into its base state.



Attention!

The Restart command (Quit) is not permitted while the Handy Programmer is waiting for a Terminator character (BCC or 'CR'). In this situation Quit is misinterpreted as a terminator or information character.



Note

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

Task	Data flow	Command	Terminator	Acknowledgement 2)	Terminator 2)
Restart (Quit)	From host system to Handy Programmer	'Q'	BCC or 'CR'		
	From Handy Programmer to host system			'Q'	BCC or 'CR'
					1)

- 1) The Quit command is not permitted at this point.



Note

Examples of telegrams can be found in [Section 10.6 starting page 40](#).

10 Working over the serial interface

**Telegram:
Initialize
CRC_16 data
checking**

This telegram initializes a data carrier located in front of the active read/write head for using CRC_16 data checking.
The telegram must then be resent if a CRC error results from a failed write operation.



Attention!

The sum of the start address and number of bytes may not exceed the usable data carrier capacity (see "CRC Check" on page 18).

Task	Data flow	Command	Start address of the first byte to be sent	Number of bytes to be sent	Terminator	Acknowledgement 2)	Start of sending	Data 3)	Terminator	Acknowledgement 2)
Initialize CRC_16 range	To HP 4)	'Z'	A3 A2 A1 A0 '0 0 0 0' BIS '0 1 9 1'	L3 L2 L1 L0 '0 0 0 1' BIS '0 1 9 2'	BCC or '1' '0' 'CR'		<STX>	D1 D2 D3...Dn	BCC or 'CR'	
	From HP 5)					<ACK> '0' or <NAK> + Error No.				<ACK> '0' or <NAK> + Error No.
				1)					1)	

- 1) The Quit command is not permitted at this point.
- 2) The acknowledgement is <ACK> '0' if no error was detected, or <NAK> + 'Error No.', if an error occurred.
- 3) Data from start address to start address + number of bytes.
- 4) From host system to Handy Programmer.
- 5) From Handy Programmer to host system.

10 Working over the serial interface

**Telegram:
Import file**

Send file from PC/laptop to the Handy Programmer.

Task	Data flow	Com-mand	File name 3)	Start address on the data carrier	Number of bytes on the data carrier	Terminator	Acknowledgement 2)	Start of sending	Data 4)	Terminator	Acknowledgement 2)
Read To HP file in 5)		'T'	'D7 . . . D0' A3 A2 A1 A0 'SAMPLE_ _' '0 0 0 0' BIS BIS '0 1 9 1' '0 1 9 2'	L3 L2 L1 L0	BCC			<STX>	D1 D2 D3...Dn BCC		
	From HP 6)						<ACK> '0' or <NAK> + Error No.				<ACK> '0' or <NAK> + Error No.
						1)				1)	

- 1) The Quit command is not permitted at this point.
- 2) The acknowledgement is <ACK> '0' if no error was detected, or <NAK> + 'Error No.', if an error occurred.
- 3) If the file name is less than 8 characters in length, spaces (20hex) must be added.
- 4) Data from start address to start address + number of bytes.
- 5) From host system to Handy Programmer.
- 6) From Handy Programmer to host system.

**Telegram:
Read out file**

Read out file from Handy Programmer to PC/laptop.

Task	Data flow	Com-mand	File name 3)	Terminator		Start of sending	Start address on the data carrier	Number of bytes on the data carrier	Data 4)	Terminator	
Read out file To HP 5)		'G'	'D7 . . . D0' BCC 'SAMPLE_ _'			<STX>					
	From HP 6)					<ACK> '0' or <NAK> + Error No.	A3 A2 A1 A0 '0 0 0 0' BIS BIS '0 1 9 1'	L3 L2 L1 L0 '0 0 0 1' BIS '0 1 9 2'	D1 D2 D3...Dn BCC		
				1)				1)			

- 1) The Quit command is not permitted at this point.
- 2) The acknowledgement is <ACK> '0' if no error was detected, or <NAK> + 'Error No.', if an error occurred.
- 3) If the file name is less than 8 characters in length, spaces (20hex) must be added.
- 4) Data from start address to start address-number of bytes.
- 5) From host system to Handy Programmer.
- 6) From Handy Programmer to host system.



Note

Examples of telegrams can be found in Section 10.6 starting page 40.

10 Working over the serial interface

Telegram: Read out file directory

Read out file directory Handy Programmer to PC/laptop.

Task	Data flow	Command	Terminator	Acknowledgement 2)	Start of sending	Data	Terminator
Read out file directory	From host system to Handy Programmer	'D'	BCC		<STX>		
	From Handy Programmer to host system			<ACK> '0' or <NAK> + Error No.		D1 D2 D3...Dn BCC	
			1)				1)

- 1) The Quit command is not permitted at this point.
- 2) The acknowledgement is <ACK> '0' if no error was detected, or <NAK> + 'Error No.', if an error occurred.

Telegram: Delete file

Delete a particular file in the Handy Programmer

Task	Data flow	Command	File name 3)	Terminator	Acknowledgement 2)
Delete a particular file	From host system to Handy Programmer	'E'	'D7 . . . D0' 'SAMPLE_ _'	BCC	
	From Handy Programmer to host system				<ACK> '0' or <NAK> + Error No.
				1)	

- 1) The Quit command is not permitted at this point.
- 2) The acknowledgement is <ACK> '0' if no error was detected, or <NAK> + 'Error No.', if an error occurred.
- 3) If the file name is less than 8 characters in length, spaces (20hex) must be added.

Telegram: Delete all files

Delete all files stored in the Handy Programmer.

Task	Data flow	Command	Terminator	Acknowledgement 2)
Delete all files	From host system to Handy Programmer	'K'	BCC	
	From Handy Programmer to host system			<ACK> '0' or <NAK> + Error No.
			1)	

- 1) The Quit command is not permitted at this point
- 2) The acknowledgement is <ACK> '0' if no error was detected, or <NAK> + 'Error No.', if an error occurred .



Note

Examples of telegrams can be found in [Section 10.6 starting page 40.](#)

10 Working over the serial interface

10.4 Error numbers

BIS L-81_ always outputs an error number. Their meaning is given in the following table.

No.	Error	Effect ¹⁾
0	No error.	Only in conjunction with <ACK>.
1	No data carrier present.	Handy Programmer goes into the base state.
2	Read error.	Handy Programmer goes into the base state.
3	Read cancelled because the data carrier was removed.	Handy Programmer goes into the base state.
4	Write error.	Handy Programmer goes into the base state.
5	Write cancelled because the data carrier was removed.	Handy Programmer goes into the base state.
		 Attention! When a write operation is cancelled, incomplete data may have been written to the data carrier. ²⁾
6	Interface error.	Handy Programmer goes into the base state (Parity or stop bit error).
7	Telegram format error.	Handy Programmer goes into the base state. Possible format errors: – Command is not 'L', 'P', 'C', 'Q' or 'Z'. – Start address or number of bytes lies outside the permissible range.
8	BCC error.	The BCC sent for read/write operations is wrong. Handy Programmer goes into the base state.
e	CRC error. The CRC on the data carrier is wrong. ³⁾	Handy Programmer goes into the base state.
i	EEPROM error.	Processor goes into the base state.
q	Timeout.	Character delay time of 1 s exceeded. Interface problem.
r	Entry error.	Specified file does not exist.
s t	Memory error.	Handy Programmer memory end reached.
V	Interface error.	Improper setting for interface parameters (baud rate, parity, data or stop bit error).
W	Protocol format error.	Protocol format not observed.
X	BCC error.	The BCC sent for file operations is incorrect. Telegram cancelled. Handy Programmer goes into the base state.

- 1) An error always results in cancelling of the telegram.
- 2) If using CRC data check, the error message E may occur at the next read command if error 4 or 5 was not remedied.
- 3) If using CRC data check, the error message E may occur if error 4 or 5 was reported for the preceding command.

10 Working over the serial interface

10.5 Read/write times



Note

The indicated times are valid beginning when the data carrier is recognized. Otherwise you must add 45 ms for power to be built up until the data carrier is recognized.

Read times in static mode (data integrity with double read, no CRC_16 data check):

Data carrier BIS L-1_ _ with 4-byte blocks		Data carrier BIS L-2_ _	
Data carrier recognition	~ 370 ms	Data carrier recognition + Read data carrier.	~ 270 ms
Read bytes 0 to 3	~ 180 ms		
for each additional 4 bytes started	+ ~ 90 ms		

Write times in static mode (data integrity with double read, no CRC_16 data check):

Data carrier BIS L-1_ _ with 4-byte blocks		Data carrier BIS L-2_ _	
Data carrier recognition	~ 370 ms	Writing not possible.	
Write bytes 0 to 3	~ 305 ms		
for each additional 4 bytes started	+ ~ 215 ms		

Read times in dynamic mode (data integrity with double read, no CRC_16 data check):

Data carrier BIS L-1_ _ with 4-byte blocks		Data carrier BIS L-2_ _	
Data carrier recognition	~ 370 ms	Data carrier recognition + Read data carrier.	~ 270 ms
Write bytes 0 to 3	~ 180 ms		
for each additional 4 bytes started	+ ~ 90 ms		

10 Working over the serial interface

10.6 Telegram examples

Forming the BCC block check

The BCC is formed out as an EXOR operation on the serially transmitted binary characters of the telegram block.

Example: Read starting at address 13, read 128 bytes.

The command line without BCC is: 'L 0013 0128 10'. 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
1    =   0011 0001 EXOR
0'   =   0011 0000 EXOR
    
```

Resulting block check: BCC = 0100 0100 = 'D'

Protocol variants

If necessary the terminator using BCC can be replaced with Carriage Return ('CR').

In the preceding example the command line 'L 0013 0128 10 D' contains 'D' as BCC. This command line is compared here with the possible variations. The various forms of acknowledgement with and without terminator are shown.

Command line from host system to BIS	Acknowledgement from BIS for correct reception	Acknowledgement from BIS for incorrect reception
With BCC as terminator, 'L 0013 0128 10 D'	<ACK> '0'	<NAK> '1'
With 'CR' instead of BCC 'L 0013 0128 10 CR'	<ACK> '0'	<NAK> '1'

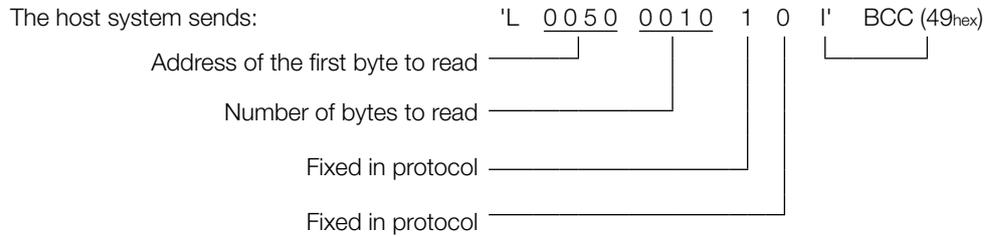
In the table the error example is <NAK> '1' (= no data carrier present).

10 Working over the serial interface

Read from the data carrier

Telegram example: Read from data carrier with BCC.

Operation: Read 10 bytes from the data carrier starting at address 50.



The Handy Programmer acknowledges with: <ACK> '0'

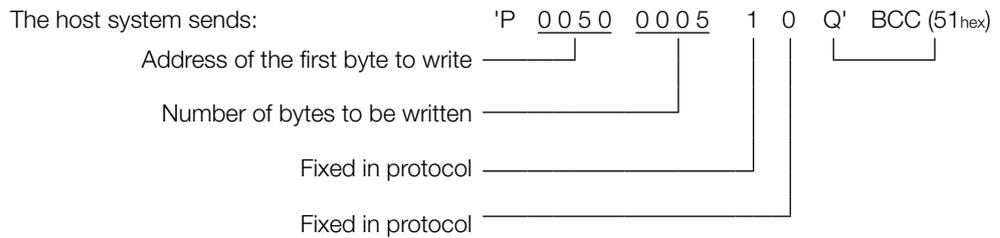
The host system gives the start command: <STX>



Write to the data carrier (1)

Telegram example: Write to the data carrier with BCC.

Operation: Write 5 bytes to the data carrier starting at address 50.



The processor acknowledges with: <ACK> '0'

The host system gives the start command and the data:

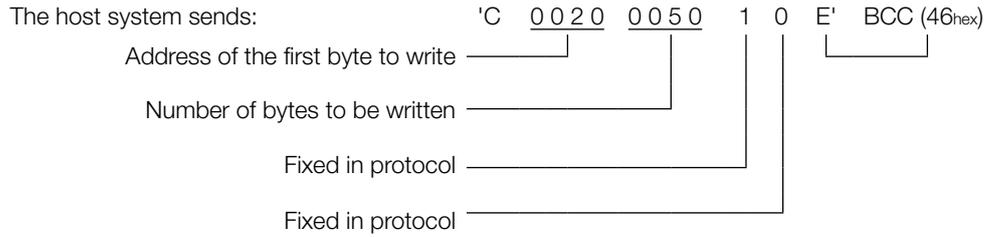


10 Working over the serial interface

Write to the data carrier (2)

Telegram example: Write a constant value to the data carrier with BCC.

Operation: Write 50 bytes to the data carrier starting at address 20 with the ASCII data value 0 (30_{hex}).



The processor acknowledges with: <ACK> '0'

The host system gives the start command and the data:

<STX> '0 2' BCC (32_{hex})

The processor acknowledges with: <ACK> '0'

Restart the Handy Programmer (Quit)

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

Operation: Place the BIS system in the base state.

The host system sends: 'Q Q' BCC (51_{hex})

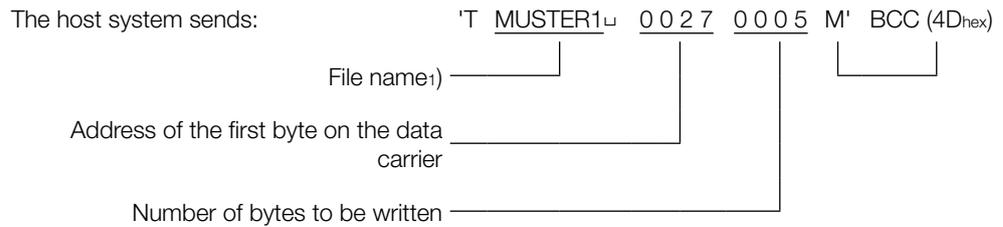
The Handy Programmer acknowledges with: 'Q Q' BCC (51_{hex})

10 Working over the serial interface

Read file in

Telegram example: Send file to the Handy Programmer with BCC.

Operation: Send the file MUSTER1 to the Handy Programmer. Write 5 bytes to the data carrier starting at address 27.

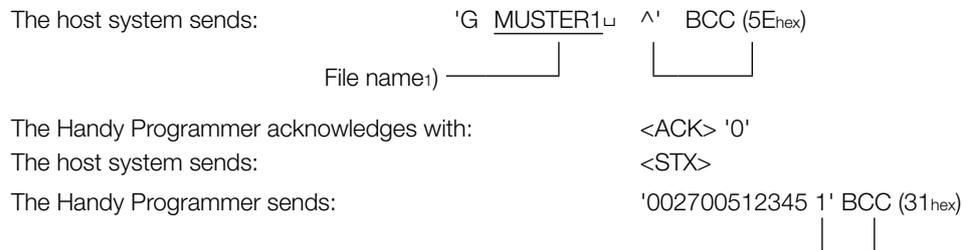


The Handy Programmer acknowledges with: <ACK> '0'
 The host system gives the start command and the data: <STX> '12345 3' BCC (32_{hex})
 The Handy Programmer acknowledges with: <ACK> '0'

Read out file

Telegram example: Read file from the Handy Programmer with BCC.

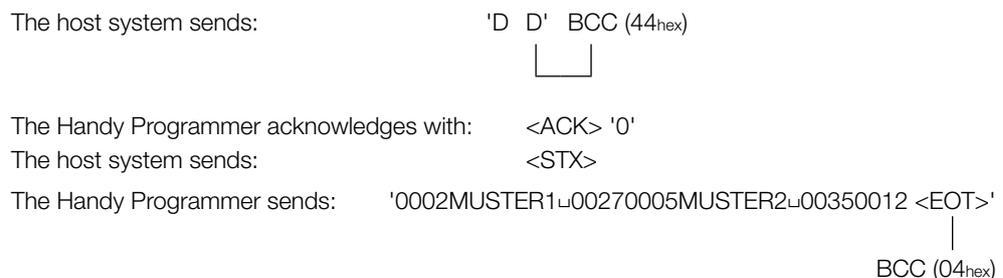
Operation: Read the file MUSTER1 from the Handy Programmer. Read 5 bytes from the data carrier starting at address 27.



Read out file directory

Telegram example: Read file directory from the Handy Programmer with blockcheck BCC.

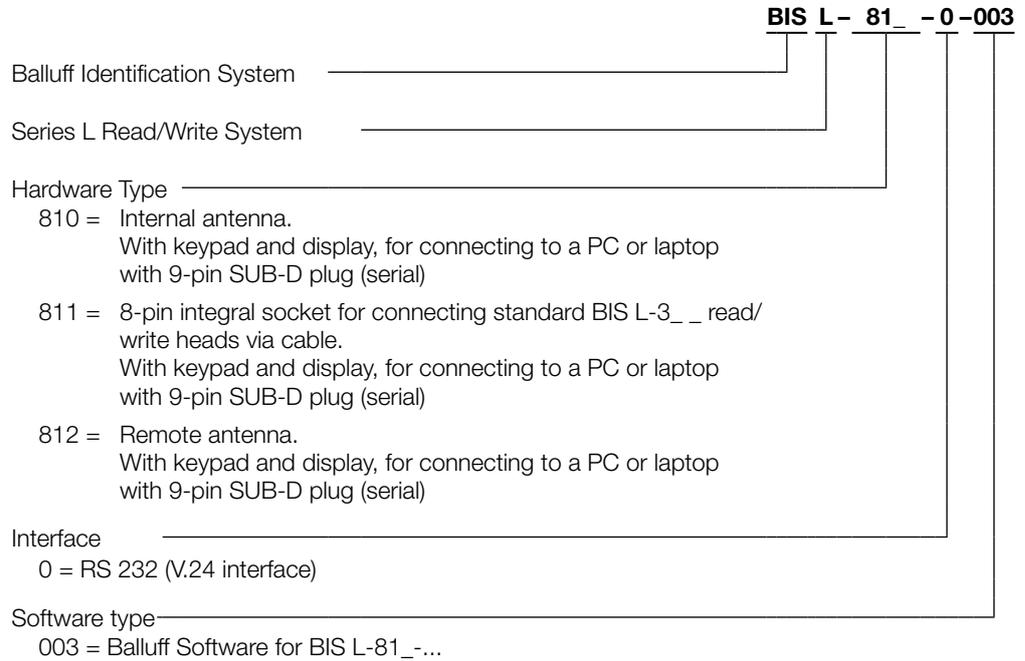
Operation: Read the file directory from the Handy Programmer. 2 files are stored in the Handy Programmer.



1) Always fill file name with 8 bytes. Even a space character (20_{hex}) used for filling must be accounted for in the BCC.

Appendix

Ordering code



Accessories (scope of delivery)

Type	Ordering Code
Carrying Pouch	149525

Accessories (optional, not included in scope of delivery)

Type	Ordering Code
Charger	BIS C-701-A
Charging stand	BIS C-702/A
Additional battery	2.4V rechargeable NiMH; 125997 1650 mAh
Connection cable: for 9-pin SUB_D COM port	2m 3m BIS C-522-PVC-02 BIS C-522-PVC-03
Connection cable: One molded male connector, other end pigtailed, length can be trimmed as needed	for BIS L-811-...; 2m BKS-S115-PU-02
Connector: without cable	for BIS L-811-... BKS-S117-00
Read heads BIS L-3_ _	for BIS L-811-... see catalog

Appendix

**Accessory
Charger
BIS C-701-A**

Features

- Consists of charger/AC adapter with 2.5 mm plug,
- primary plug EURO, UK, US interchangeable (included),
- Approvals: CE, UL.

Technical Data

Primary voltage $U_{Primary}$		100 - 240 V _{AC}
Secondary voltage U_{Sec}		6 V _{DC} ± 5 %
Input frequency $F_{Primary}$		47 - 63 Hz
Secondary current I_{Sec}		2.1 A
Ambient temperature T_A		-40 °C ... +70 °C
Material		PPE-V1-125 °C
Color		black
Housing		MPP10
Output cable	Ordering code	11.7892.503-80
	Length, diameter	2000 mm, 3 x 0.5 mm ²
	Connector	2.5 mm plug

**Accessory
Charging Stand
BIS C-702-A**

Features

- Consists of charger/AC adapter with charge state indicator and base,
- primary plug EURO, UK, US interchangeable (included),
- Approvals: CE, UL.

Charge state indicator

- LED red - battery charged
- LED red flashing - battery charging

Technical Data

Primary voltage $U_{Primary}$		100 - 240 V _{AC}
Secondary voltage U_{Sec}		6 V _{DC} ± 5 %
Input frequency $F_{Primary}$		50 - 60 Hz
Secondary current I_{Sec}		800 mA
Ambient temperature T_A		-40 °C ... +70 °C
Material		PPE-V1-125 °C
Color		black
Housing		MPP10
Output cable	Ordering code	11.7892.503-80
	Length, diameter	2000 mm, 3 x 0.5 mm ²

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 45
Address range
 Read 19
 Write 20
Areas of application 9
ASCII Table 47

B

Battery
 install 14
 Load 14
 remove 14
Blockcheck BCC 32

C

Charger 46
Charging stand 46
Check procedure 9
Checksum 18
Connecting 30
Connections 12, 13
Control function 9
Conventions 4
Convertible data carriers 26
CRC Check 18
Cursorposition
 Edit 20

D

Data
 Edit 25
Data carrier
 Configuration 21
 Converting 26
 Initializing 18, 25
 Processing 24
 Read 24
 Types 18, 23
 Write 24
Data format 20
 Capacity 21
Data integrity 9
Data transmission 30
Dimensions 11
Display format 16

E

Electrical data 12
End address
 Edit 20
 Read 19
 Write 20
Error message 18
Error messages
 Display 29
Error numbers 38

F

File menu
 Opening 27
Files 10, 27
 Delete 28
 File format 10
 File name 10

H

Handy Programmer
 Product description 9
Housing 12

I

Input mode 19
 Read 19
 Write 20
Insertion address
 Edit 20
Intended Use 6
Interface
 parameterizing 21
Interface information 13

K

Key assignments 16
Keypad
 Navigation 16

L

Language selection 16

M

Main components 9
Meaning
 Warning notes 6

O

Operating conditions 12
Ordering code 45

P

Principle of operation 23
Product description 9
Protocol sequence
 Read 30
 Write 31

R

Read times 39
RS 232 interface 10, 13

S

Safety
 Commissioning 6
 Installation 6
 Operation 6
Save configuration 22
Serial interface
 parameterizing 21
Serial number
 Read 27
Set language 22
Setup 23
Special characters 20
Start address
 Read 19
 Write 20
Symbols
 Representation 5

T

Telegram
 Data checking 35
 File directory 37
 Files 36
 Read data carrier 33
 Restart 34
 Write to data carrier 34
Telegram examples 40
Telegram terminator 32
Telegram types 32
Turn on the unit 15

W

Write times 39

 **www.balluff.com**

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