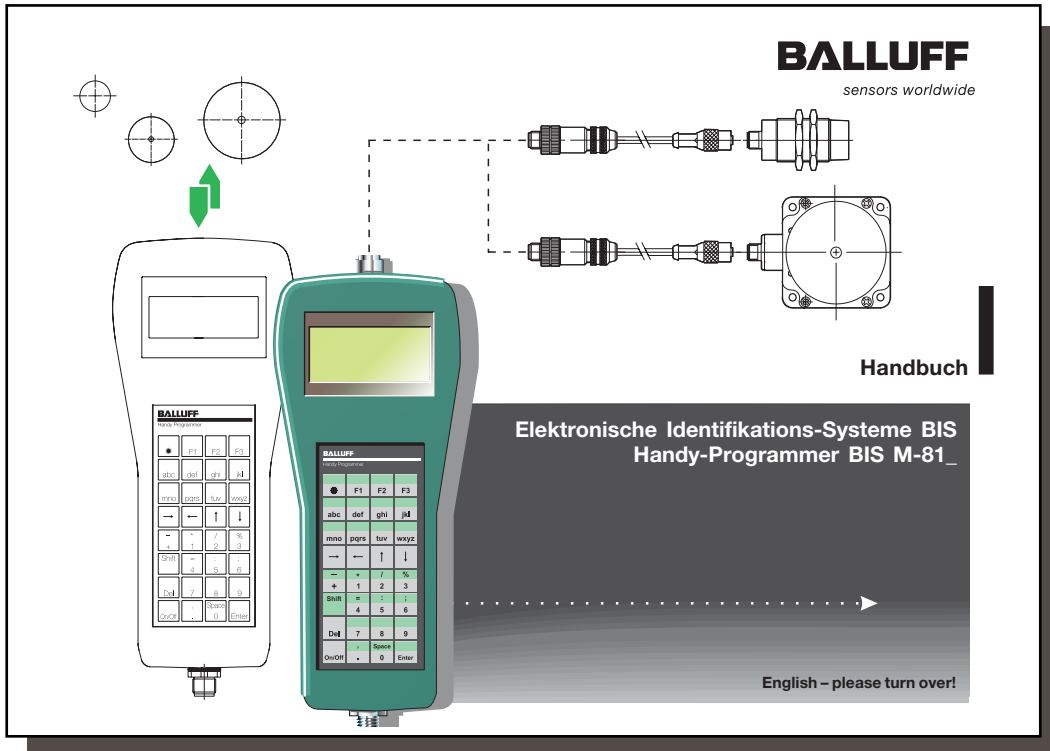


1



2

Nr. 854 610 D/E • Ausgabe 0703
Änderungen vorbehalten.

Schreibweise:

Zu sendende Steuerzeichen sind in spitze Klammern gesetzt.

Im ASCII-Code zu übertragende Zeichen sind in Hochkommata gesetzt.

Beispiel: <STX> '1 2 3 4 5 6' BCC

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

www.balluff.com

Inhaltsverzeichnis

Sicherheitshinweise	4
Einführung Identifikations-System BIS M	5/6
Basiswissen für die Anwendung	7-11
Inbetriebnahme des Geräts	12/13
Konfiguration	14-17
Funktionsbeschreibung BIS M-81_	
Datenträger bearbeiten	18-25
Arbeiten mit Dateien	26/27
Spezielle Datenträgerfunktionen	28
Funktionsbeschreibung BIS M-81_ / PC	
Lesen / Schreiben über die serielle Schnittstelle	29-33
Telegrammaufbau und Beispiele	34-39
Fehlernummern	40
Schreib-/Lesezeiten	41
Arbeiten mit Dateien über die serielle Schnittstelle	42
Telegrammaufbau und Beispiele	43-48
Fehlernummern	49
Schnittstelleninformationen	
BIS M-810	50
BIS M-811	51
BIS M-812	52
Akku laden	53
Technische Daten	54/55
Bestellinformationen	
Typenschlüssel	56
Zubehör	57-60
Anhang, ASCII-Tabelle	61

Sicherheitshinweise

Bestimmungs- gemäßer Betrieb	Der Handy-Programmer BIS M-81_ bildet zusammen mit den anderen Bausteinen des Systems BIS M ein mobiles Identifikations-System und darf nur für diese Aufgabe im industriellen Bereich entsprechend Klasse A des EMV-Gesetzes eingesetzt werden.
Installation und Betrieb	Installation und Betrieb sind nur durch geschultes Fachpersonal zulässig. Unbefugte Eingriffe und unsachgemäße Verwendung führen zum Verlust von Garantie- und Haftungsansprüchen. Bei der Installation des Handy-Programmiers ist das Kapitel Inbetriebnahme des Geräts genau zu beachten.
Einsatz und Prüfung	Für den Einsatz des Identifikations-Systems sind die einschlägigen Sicherheitsvorschriften zu beachten. Insbesondere müssen Maßnahmen getroffen werden, dass bei einem Defekt des Identifikations-Systems keine Gefahren für Personen und Sachen entstehen können. Hierzu gehören die Einhaltung der zulässigen Umgebungsbedingungen und die regelmäßige Überprüfung der Funktionsfähigkeit des Identifikations-Systems mit allen damit verbundenen Komponenten.
Funktionsstörungen	Wenn Anzeichen erkennbar sind, dass das Identifikations-System nicht ordnungsgemäß arbeitet, ist es außer Betrieb zu nehmen und gegen unbefugte Benutzung zu sichern.
Gültigkeit	Diese Beschreibung gilt für Handy-Programmer der Baureihe BIS M-810-0-003... und BIS M-812-0-003... (ab Software-Stand V1.2, Hardware-Stand V2.0) sowie BIS M-811-0-003... (ab Software-Stand V1.2, Hardware-Stand V1.1).

5

Einführung Identifikations-System BIS M

Dieses Handbuch soll den Anwender bei der Inbetriebnahme und Bedienung des Handy-Programmers BIS M-81_ und weiterer Komponenten des Identifikations-Systems BIS M anleiten, so dass ein sofortiger, reibungsloser Betrieb erreicht wird.

Prinzip

Das Identifikations-System BIS M gehört zur Kategorie der

berührungslos arbeitenden Systeme, die sowohl lesen als auch schreiben können.

Diese Doppelfunktion ermöglicht Einsätze, bei denen nicht nur fest in den Datenträger programmierte Informationen transportiert, sondern auch aktuelle Informationen gesammelt und weitergegeben werden.

Einsatzgebiete

Einige der wesentlichen Einsatzgebiete finden sich

- **in der Produktion zur Steuerung des Materialflusses**
(z.B. bei variantenspezifischen Prozessen),
beim Werkstücktransport mit Förderanlagen,
zur Datengewinnung für die Qualitätssicherung,
zur Erfassung sicherheitsrelevanter Daten;
- **in der Werkzeugcodierung und -überwachung;**
- **in der Betriebsmittelorganisation;**
- **im Lagerbereich zur Kontrolle der Lagerbewegungen und -bestände;**
- **im Transportwesen und in der Fördertechnik;**
- **in der Entsorgung zur mengenabhängigen Erfassung.**

Mit dem mobilen Identifikations-System auf der Basis des Handy-Programmers BIS M-81_ können Datenträger einer Anlage z.B. zu Servicezwecken überprüft, korrigiert oder initialisiert werden. Die an der Anlage gelesenen Daten können bei Bedarf im Handy-Programmer gespeichert und zu einem späteren Zeitpunkt an einen PC übertragen und analysiert werden.

deutsch BALLUFF 5

6

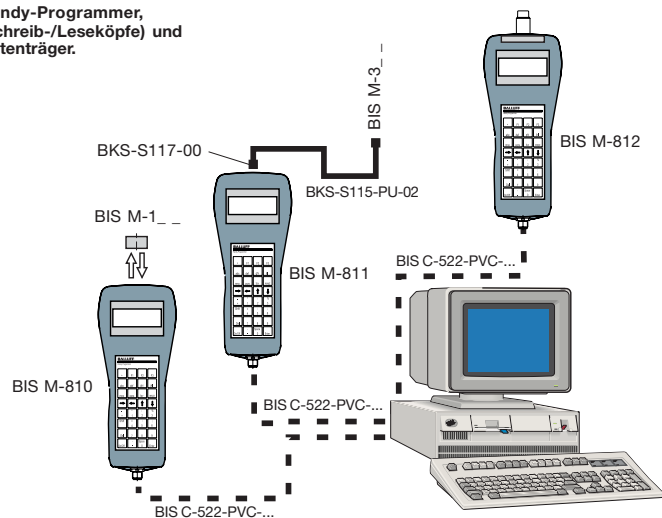
Einführung Identifikations-System BIS M

System- komponenten

Die Hauptbestandteile des Identifikations-Systems BIS M sind

- Handy-Programmer,
- (Schreib-/Leseköpfe) und
- Datenträger.

Anordnung mit Handy-Programmer



Schematische
Darstellung eines
Identifikations-Systems
(Beispiel)

6 BALLUFF deutsch

BIS M-81_ Basiswissen für die Anwendung

Bedienung, Tastatur und Display

Die Bedienung des Handy-Programmers erfolgt menügesteuert. Die Eingaben werden über eine Folientastatur mit 32 Tasten vorgenommen. Einige Tasten sind mehrfach belegt. Zwei Arten der Mehrfachbelegung sind zu unterscheiden:

- 2 Funktionen Umschaltung mit der Taste **[SHIF]**
- 3 Buchstaben 1× drücken = 1. Buchstabe, 2× = 2. Buchstabe, 3× = 3. Buchstabe
- Umschaltung auf Großbuchstaben mit der Taste **[SHIF]**

Das LCD-Display, mit dem alle Daten angezeigt werden, hat 4 Zeilen mit je 20 Zeichen.

Sprachen Deutsch/Englisch

Im Hauptmenü kann durch gleichzeitiges Drücken von Shift+F1 eine andere Sprache gewählt werden. Die gewählte Sprache bleibt jedoch nur bis zum nächsten Ausschalten des Handy-Programmers erhalten. Um sie dauerhaft anzuwählen, muss sie im Konfigurationsmenü ausgewählt werden.

Menü-/Bedienkonzept

Die wesentlichen Funktionen des Geräts werden im Hauptmenü 1 angezeigt. Jedes Untermenü ist durch eine Ziffer gekennzeichnet. Durch Drücken der entsprechenden Zifferntaste wird das betreffende Untermenü aufgerufen und das zugehörige Fenster angezeigt. In einigen Fällen wird die Funktion direkt ausgeführt, in anderen Fällen muss eine weitere Menüauswahl durchlaufen werden.

Zur raschen Orientierung ist im Handbuch stets die Taste/Tastenfolge angegeben, mit der die jeweilige Funktion / das jeweilige Menü – stets ausgehend vom Hauptmenü 1 – direkt erreicht wird.

```

--- Hauptmenü ---
1 Lesen      4 Datei
2 Schreiben  5 Konfig
3 Editieren  6 Weiter
    
```

Hauptmenü 1

```

-- Hauptmenü 2 --
1 Init
2 Serial Number
6 <--
    
```

Hauptmenü 2

BIS M-81_ Basiswissen für die Anwendung

Batteriebetrieb

Die Spannungsversorgung des Gerätes erfolgt über einen Akku. Beim Schreiben oder Lesen beträgt der Stromverbrauch 275 mA. Findet kein Schreib-/Lesevorgang statt, hat das Gerät einen Stromverbrauch von 65 mA.

Die Systemkomponenten werden vom Handy-Programmer elektrisch versorgt. Der Datenträger stellt eine eigenständige Einheit dar, benötigt also keine leitungsgebundene Stromzuführung. Er bekommt seine Energie vom Schreib-/Lesekopf. Dieser sendet ständig ein Trägersignal aus, das den Datenträger versorgt, sobald der notwendige Abstand erreicht ist. In dieser Phase findet der Schreib-/Lesevorgang statt. Dieser kann statisch oder dynamisch erfolgen.

Auswahl der Schreib-/Leseköpfe beim BIS M-811

Für den Handy-Programmer sind verschiedene Schreib-/Leseköpfe *) erhältlich. Diese werden über ein Kabel an den Handy-Programmer angeschlossen. Die Auswahl des Schreib-/Lesekopfs *) hängt von dem verwendeten Datenträgertyp ab. Über das Adapterkabel BKS-S115-PU-02 *) kann ein Schreib-/Lesekopf BIS M-3_ _ (ab Hardware-Stand V2.0) aus dem BIS M-Lieferprogramm angeschlossen werden.

Dialogmodus

Über den Schreib-/Lesekopf schreibt der Handy-Programmer Daten auf den Datenträger oder liest sie vom Datenträger. Um zu schreibende Daten vorzubereiten oder gelesene Daten zu verändern, besitzt der Handy-Programmer eine Editor-Funktion. Die Daten können im ASCII-, HEX-, BCD- oder BINÄR-Format gelesen oder auf den Datenträger geschrieben werden.

Das Darstellungsformat der angezeigten Daten kann mit den Tasten **[abc]** und **[ghi]** umgeschaltet werden:

Taste	Darstellungsformat	Taste	Darstellungsformat
[abc]	->ASCII -> BCD -> BINÄR	[ghi]	HEX

*) siehe Zubehörliste auf 157

BIS M-81_ Basiswissen für die Anwendung

Datensicherheit der gelesenen und geschriebenen Daten

Bei der Übertragung der Daten zwischen Schreib-/Lesekopf und Datenträger bedarf es eines Verfahrens, welches erkennen kann, ob die Daten richtig gelesen bzw. richtig geschrieben worden sind.

Bei der Auslieferung ist der Handy-Programmer auf das bei Balluff gebräuchliche Verfahren des doppelten Einlesens mit anschließendem Vergleich eingestellt. Neben diesem Verfahren steht ein zweites als Alternative zur Verfügung: die CRC_16-Datenprüfung. Hier wird ein Prüfcode auf den Datenträger geschrieben, der jederzeit und überall das Kontrollieren der Daten auf Gültigkeit erlaubt.

Vorteile mit CRC_16 Check	Vorteile mit doppeltem Lesen
Datensicherheit auch während der nicht aktiven Phase (Datenträger außerhalb des Schreib-/Lesebereichs).	Beim Datenträger gehen keine Nutzbyte zur Speicherung eines Prüfcodes verloren.
Kürzere Lesezeiten, da jede Seite nur einmal gelesen wird.	Kürzere Schreibzeiten, da kein CRC geschrieben werden muss.

Da beide Varianten je nach Anwendung vorteilhaft sind, kann die Methode der Datensicherheit vom Kunden eingestellt werden (siehe Konfiguration ¶ 15 "Datenträger auswählen").

Ein Mischbetrieb der beiden Prüfverfahren ist nicht möglich!

BIS M-81_ Basiswissen für die Anwendung

Datensicherheit der gelesenen und geschriebenen Daten

Um die Methode mit dem CRC-Check verwenden zu können, müssen die Datenträger initialisiert werden. Entweder man benutzt Datenträger mit dem Datensatz bei Werksauslieferung (alle Daten sind 0) oder man muss den Datenträger initialisieren. Dazu stehen zwei Möglichkeiten zur Verfügung:

- Über die Funktion "Init" im Hauptmenü 2 den Datenträger initialisieren (siehe ¶ 28).
- Über den speziellen Initialisierungsbefehl 'Z' den Datenträger beschreiben (siehe ¶ 39).

Ist die CRC_16-Datenprüfung aktiviert, wird bei Erkennen eines CRC-Fehlers eine spezielle Fehlermeldung ausgegeben.

Wenn die Fehlermeldung keine Folge aus einem missglückten Schreibauftrag ist, kann davon ausgegangen werden, dass eine oder mehrere Speicherzellen auf dem Datenträger defekt sind. Der betreffende Datenträger ist auszutauschen.

Ist der CRC-Fehler jedoch eine Folge aus einem missglückten Schreibauftrag, muss der Datenträger neu initialisiert werden, um ihn wieder verwenden zu können.

CRC_16

Die Prüfsumme wird je CRC-Block (entspricht 16 Byte) auf den Datenträger als 2 Byte grosse Information geschrieben. Es gehen 2 Byte pro CRC-Block verloren, d.h. der CRC-Block enthält nur noch 14 Byte Nutzdaten. Dies bedeutet, dass sich die konkret nutzbare Anzahl Byte verringert (siehe ¶ 11 "Unterstützte Datenträger und Speicherkapazität").

Werkseitig ist die CRC_16-Datenprüfung nicht eingestellt.

Gelesene Daten abspeichern

Aus einem Datenträger gelesene Daten, die sich im Arbeitsspeicher des Handy-Programmers befinden, können unter einem Dateinamen in einem EEPROM abgelegt werden.

BIS M-81_ Basiswissen für die Anwendung

Kommunikation zwischen Handy-Programmer und PC über die RS 232-Schnittstelle

Der Handy-Programmer verfügt über eine serielle RS 232-Schnittstelle, über die Schreib-/Leseaufträge ausgeführt und Dateien zwischen einem PC und dem Handy-Programmer übertragen werden können. Hierzu wird das Kabel BIS C-522 *) benötigt.

Die Kommunikation auf der seriellen Schnittstelle erfolgt in einem festgelegten Protokoll.

Der Datenverkehr zwischen dem Handy-Programmer und dem Laptop/PC geschieht über festgelegte Telegramme. Folgende Möglichkeiten bestehen:

- Lese-/Schreibaufträge vom PC über den Handy-Programmer abwickeln, dabei übernimmt der Handy-Programmer die Funktion einer BIS-Auswerteeinheit. Details siehe ab ¶ 29.
- Dateiverwaltung und Datenaustausch gespeicherter Daten zwischen PC und Handy-Programmer. Details siehe ab ¶ 42.

Unterstützte Datenträger und Speicherkapazität

Mifare

Balluff Datenträgertyp	Hersteller	Bezeichnung	Speicherkapazität	Nutzbare Byte bei CRC	Speichertyp
BIS M-1__-01	Philips	Mifare Classic	752 Byte	658 Byte	EEPROM



Die ISO15693 Datenträger können bei den Geräten BIS M-810 und BIS M-812 ab dem Hardware-Stand V2.0 bearbeitet werden. Zusätzlich muss beim BIS M-811 der Hardware-Stand der extern angeschlossenen Schreib-/Leseköpfe BIS M-3__ V2.0 oder höher sein!

ISO15693

Balluff Datenträgertyp	Hersteller	Bezeichnung	Speicherkapazität	Nutzbare Byte bei CRC	Speichertyp
BIS M-1__-02	Fujitsu	MB89R118	2000 Byte	1750 Byte	FRAM
BIS M-1__-03 ¹	Philips	SL2/CS20	112 Byte	98 Byte	EEPROM
BIS M-1__-04 ¹	Texas Inst.	TAG-IT Plus	256 Byte	224 Byte	EEPROM
BIS M-1__-05 ¹	Infineon	SRF55V02P	224 Byte	196 Byte	EEPROM
BIS M-1__-06 ¹	EM	EM4135	288 Byte	252 Byte	EEPROM
BIS M-1__-07 ¹	Infineon	SRF55V10P	992 Byte	868 Byte	EEPROM

*) siehe Zubehörliste auf ¶ 55

¹ auf Anfrage

BIS M-81_ Inbetriebnahme des Geräts

Schreib-/Lesekopf einstecken beim BIS M-811

Die Handy-Programmer BIS M-810 und BIS M-812 haben eine integrierte Antenne und benötigen keinen externen Schreib-/Lesekopf.

Bei dem Handy-Programmer BIS M-811 wird an der Stirnseite des Gerätes über ein Lesekopfkabel ein Standard Schreib-/Lesekopf BIS M-3__ betrieben. Wird kein Schreib-/Lesekopf BIS M-3__ angeschlossen und das Gerät eingeschaltet, erscheint folgender Warnhinweis:

```
!!!! FEHLER !!!!
-----
S/L-Kopf Fehler
```

Fehlermeldung wenn kein S/L-Kopf angeschlossen wurde

Handy-Programmer einschalten

Der Handy-Programmer wird eingeschaltet, indem die Taste **ON** gedrückt wird.

Das Gerät meldet sich mit der Anzeige von Gerätetyp und Software-Version. Gleichzeitig wird ein Akkutest durchgeführt.

```
BALLUFF BIS M-81_
Handy Programmer
Software Vx.x
```

Startdisplay mit Software-Version

Die Anzeigedauer kann durch drücken der Taste **Enter** verkürzt werden.

BIS M-81_ Inbetriebnahme des Geräts

Handy-Programmer einschalten (Fortsetzung)

```
!!!!!!!!!!!!!!!!!!!!!!
  Akkuspannung
  zu niedrig!
  !!!!!!!!!!!!!!!!!!!!!!!
```

Warnmeldung Akkuspannung

Falls die Spannung unter eine bestimmte Nennkapazität abgesunken ist, wird eine Warnmeldung ausgegeben. Das Gerät kann trotz dieser Warnung noch eine kurze Zeit benutzt werden. Der Akku sollte jedoch sobald wie möglich aufgeladen werden (siehe ¶ 53).

Nach Drücken der Taste **Enter** zeigt das Gerät mit folgender Anzeige seine Betriebsbereitschaft an.

Nach einem bestandenen Akkustest erscheint diese Anzeige direkt.

```
--- Hauptmenü ---
1 Lesen      4 Datei
2 Schreiben  5 Konfig
3 Editieren  6 Weiter
```

Hauptmenü 1

BIS M-81_ Konfiguration

Falls nicht mit der Werkseinstellung gearbeitet werden soll, ist zu Beginn die Konfiguration des Handy-Programmers durchzuführen.

Konfiguration 1 **5**

Über Menüpunkt 5 im Hauptmenü 1 gelangt man in das Konfigurationsmenü 1:

```
-- Konfiguration --
1 Adressen   4 V24
2 Format
3 Datenträger 6 -->
```

Konfigurationsmenü 1

Adressen konfigurieren **5 1**

Die Adressbereiche für Lesen und Schreiben der Datenträger und für die Einsprungadresse beim Editieren können voreingestellt werden.

Die Auswahlziffer des jeweiligen Modus sowie die gewünschten Werte müssen eingegeben und mit **Enter** bestätigt werden. Werte, die übernommen werden sollen, können direkt mit **Enter** bestätigt werden. Auf diese Weise werden die gezeigten Menüs nacheinander durchlaufen.

```
Eingabemodus LESEN
-----
0 -> Start/Endadr
1 -> Start/Anzahl 0
```

Eingabemodus zum Lesen wählen

Eingabemodus LESEN:

Startadresse / Endadresse bzw.
Startadresse / Anzahl Byte für Lesen

Bereich LESEN

```
Bereich LESEN
-----
von Byte Nr.: 0
bis Byte Nr.: 255
```

Eingabemenü 0 für Modus
Start-/Endadresse zum Lesen

```
Bereich LESEN
-----
von Byte Nr.: 0
Anzahl Byte: 0256
```

Eingabemenü 1 für Modus Start-
adresse/Anzahl Byte zum Lesen

BIS M-81_ Konfiguration

Adressen konfigurieren
 [5] [1]
 (Fortsetzung)

Eingabemodus SCHREIBEN

- Startadresse / Endadresse bzw.
- Startadresse / Anzahl Byte für Schreiben

```
Eingabemodus SCHR.
-----
0 -> Start/Endadr
1 -> Start/Anzahl 0
```

Eingabemodus zum Schreiben wählen

Bereich SCHREIBEN

```
Bereich SCHREIBEN
-----
von Byte Nr.: 0
bis Byte Nr.: 255
```

Eingabemenü 0 für Modus Start-/Endadresse zum Schreiben

```
Bereich SCHREIBEN
-----
von Byte Nr.: 0
Anzahl Byte: 0256
```

Eingabemenü 1 für Modus Startadresse/Anzahl Byte zum Schreiben

Bereich EDITIEREN

Einsprungadresse und Größe beim EDITIEREN

Hier sind anzugeben:

- die größte zu editierende Adresse sowie
- die Adresse (Cursorposition), die beim Editieren zuerst angezeigt werden soll.

```
Bereich EDITIEREN
-----
Endadresse: 0751
Cursorposition: 0000
```

Eingabemenü für Endadresse beim Editieren

Nach der letzten bestätigten Eingabe kehrt das Programm ins Konfigurationsmenü 1 zurück.

BIS M-81_ Konfiguration

Format wählen
 [5] [2]

Das Datenformat, in welchem Daten im Editor eingegeben werden, kann zwischen vier Formaten gewählt werden:

- hexadezimal (HEX)
- binär codiert dezimal (BCD)
- ASCII
- BINÄR (BIN)

Bei der Auswahl des Datenformats BCD kann zusätzlich noch eine Umcodierung für die Sonderzeichen eingegeben werden. Wird BCD eingestellt und auf dem Datenträger befinden sich Daten mit den Ziffern A_{Hex} bis F_{Hex}, werden diese Ziffern als Sonderzeichen interpretiert. Mit dieser Funktion können BCD-Werte mit Vorzeichen und Dezimalpunkt ausgegeben werden. Folgende Sonderzeichen sind voreingestellt:

'A' = * 'B' = + 'C' = , 'D' = - 'E' = . 'F' = /

Datenträger auswählen
 [5] [3]

```
- Datenträger Daten -
DTTyp: 0 All Types
CRC: N
```

In dieser Maske kann der Datenträgertyp, der bearbeitet werden soll, eingestellt werden.

0 = ALL Types: Alle von Balluff unterstützte Datenträger können bearbeitet werden.

1 = Mifare: Alle von Balluff unterstützte Mifare-Datenträger können bearbeitet werden. (Siehe ¶ 11 "Unterstützte Datenträger und Speicherkapazität")

2 = ISO15693: Alle von Balluff unterstützte Datenträger der ISO15693 können bearbeitet werden. (Siehe ¶ 11 "Unterstützte Datenträger und Speicherkapazität")



Die ISO15693 Datenträger können bei den Geräten BIS M-810 und BIS M-812 ab dem Hardware-Stand V2.0 bearbeitet werden. Zusätzlich muss beim BIS M-811 der Hardware-Stand der extern angeschlossenen Schreib-/Leseköpfe BIS M-3_ _ V2.0 oder höher sein!

Auch wird in dieser Maske eingestellt, ob mit oder ohne CRC-Prüfung gearbeitet werden soll.

- CRC: Prüfung J (Ja), N (Nein)

BIS M-81_ Konfiguration

Schnittstelle parametrieren

5 4

Einstellung der seriellen RS 232-Schnittstelle zur Kommunikation mit einem PC oder Laptop:

Folgende Parameter können eingestellt werden:

- Baudrate: 9600, 19200, 38400 oder 57600 Baud
- Stopbit: 1 oder 2
- Datenbit: 8 (kann nicht geändert werden)
- Parität: E (even), O (odd) oder N (none)
- BCC / CR: B (BCC), C (CR)

Die gewünschten Werte müssen eingegeben und mit **Enter** bestätigt werden. Werte, die übernommen werden sollen, können direkt mit **Enter** bestätigt werden.

Konfiguration 2 Language

5 6 1

Über Menüpunkt 6 kommt man in das Konfigurationsmenü 2:

```
-- Konfiguration --
1 Language
2 <--
```

Hier kann die Sprache des Menüs gewählt werden.

Menüpunkt 2 führt über die Abfrage, ob die Konfiguration gespeichert werden soll, wieder zurück ins Hauptmenü 1.

Konfigurationsmenü 2

Funktionsbeschreibung BIS M-81_ Datenträger bearbeiten

Datenträger lesen

1

Durch Auswahl des Menüpunkts 1 "Lesen" im Hauptmenü 1 erreicht man das Menü zum Lesen des Datenträgers. Es wird die Eingabe der Startadresse und – abhängig vom konfigurierten Eingabemodus – der Endadresse oder der Anzahl Byte verlangt:

```
Datenträger lesen
-----
von Byte Nr.:
bis Byte Nr.:
```

oder

```
Datenträger lesen
-----
von Byte Nr.:
Anzahl Byte :
```

Eingabemenü 1 zum Lesen vom Datenträger

Eingabemenü 2 zum Lesen vom Datenträger

Die Daten der zuletzt erfolgten Eingabe bzw. die in der Konfiguration angegebenen Daten für Startadresse und Endadresse bzw. Anzahl Byte werden als Vorgabewerte angezeigt. Mit den Tasten **↓** und **↑** kann zwischen den beiden Eingabefeldern gewechselt werden. Mit der Taste **Enter** werden die Daten übernommen.

Werden die eingegebenen Daten im Feld "bis Byte Nr.:" bzw. im Feld "Anzahl Byte ::" mit der Taste **Enter** bestätigt, wird der Lesevorgang gestartet. Solange noch kein Datenträger im Lesebereich erkannt wird, blinkt ein Balken in der Mitte der 2. Zeile. Befindet sich ein Datenträger im Lesebereich des Schreib-/Lesekopfs, hört das Blinken des Balkens auf und die Daten werden aus dem Datenträger gelesen.

```
Datenträger lesen
-----
von Byte Nr.:
Anzahl Byte :
```

Maske während dem Lesen vom Datenträger



Während der Zeit, in der die Daten vom Datenträger gelesen werden, darf der Datenträger nicht aus dem Schreib-/Lesebereich entfernt werden!

Funktionsbeschreibung BIS M-81_ Datenträger bearbeiten

Datenträger lesen



(Fortsetzung)

Soll der Lesevorgang abgebrochen werden, ist die Taste zu drücken.

Nach einem erfolgreichen Lesevorgang verschwindet der Balken und die Daten werden auf dem Display angezeigt.

Die Daten werden abhängig vom eingestellten Format als HEX-Werte, ASCII-Zeichen, BCD-Ziffern oder BINÄR dargestellt. Mit den Tasten und kann zwischen den Darstellungsformaten umgeschaltet werden:

Taste	Darstellungsformat	Taste	Darstellungsformat
	>ASCII -> BCD -> BIN		HEX

```
Anzeige 0010 - 0049
-----
 ABCDEFGHIJKLMNOPQRST
 UVWXYZ123456789abcde
```

Anzeige der Daten im
ASCII-Format

```
Anzeige 0100 - 0119
-----
 01234567890123456789
 01234567890123456789
```

Anzeige der Daten im
BCD-Format

```
Anzeige 0150 - 0163
-----
 00 01 02 03 04 05 06
 07 08 09 0A 0B 0C 0D
```

Anzeige der Daten im
HEX-Format

Nach Betätigen der Taste befindet man sich wieder im Hauptmenü 1.

Funktionsbeschreibung BIS M-81_ Datenträger bearbeiten

Fehler beim Lesen

Tritt während des Lesens ein Fehler auf, wird eine Fehlermeldung auf dem Display ausgegeben.

Fehlermeldung "Datenträger entfernt"

```
!!!! FEHLER !!!!
-----
 Fehler beim Lesen
 Datenträger entfernt
```

Fehlermeldung 1 beim Lesen

Fehlerursachen:

- Datenträger zu früh aus dem Lesebereich entfernt
- Abstand Datenträger - Schreib-/Lesekopf zu groß

Nach Bestätigen der Fehlermeldung mit der Taste befindet man sich wieder im Hauptmenü 1.

Fehlermeldung "Maximale Endadresse überschritten"

```
!!!! FEHLER !!!!
-----
 Maximale Endadresse
 überschritten
```

Fehlermeldung 2 beim Lesen

Fehlerursachen:

- Endadresse ist größer als die konfigurierte Kapazität des Datenträgers

Nach Bestätigen der Fehlermeldung mit der Taste befindet man sich wieder in der Eingabemaske für das Lesen.

Funktionsbeschreibung BIS M-81_ Datenträger bearbeiten

Fehler beim Lesen
(Fortsetzung)

Fehlermeldung "???" Abstand ???"

```
Datenträger lesen
??? Abstand ???
von Byte Nr.: 0
Anzahl Byte : 0600
```

Fehlerursachen:

- Abstand Datenträger – Schreib-/Lesekopf zu groß
- Endadresse (= Startadresse + Anzahl Byte) liegt außerhalb der Kapazität des Datenträgers
- Datenträger defekt

Fehlermeldung 3 beim Lesen

Der Handy-Programmer wiederholt den Lesevorgang noch 2 Mal. Schlägt auch der 3. Lesevorgang fehl, wird mit dieser Fehlermeldung abgebrochen:

```
!!!! FEHLER !!!!
-----
Fehler beim Lesen
Abstand / sonstig.
```

Fehlermeldung 4 beim Lesen

Funktionsbeschreibung BIS M-81_ Datenträger bearbeiten

Auf Datenträger schreiben

2

Wird Menüpunkt 2 "Schreiben" im Hauptmenü ausgewählt, erscheint die Eingabemaske zum Schreiben auf den Datenträger. Es wird die Eingabe der Startadresse und – abhängig vom konfigurierten Eingabemodus – der Endadresse oder der Anzahl Byte verlangt.

```
Datenträg. schreiben
-----
von Byte Nr.:
bis Byte Nr.:
```

oder

```
Datenträg. schreiben
-----
von Byte Nr.:
Anzahl Byte :
```

Eingabemenü 1 zum Schreiben auf den Datenträger

Eingabemenü 2 zum Schreiben auf den Datenträger

Die Daten der zuletzt erfolgten Eingabe bzw. die in der Standard-Konfiguration angegebenen Daten für Startadresse und Endadresse bzw. Anzahl Byte werden als Vorgabewerte benutzt. Mit den Tasten **[↑]** und **[↓]** kann zwischen den beiden Eingabefeldern gewechselt werden. Mit der Taste **[Enter]** werden die Daten übernommen.

Werden die eingegebenen Daten im Feld "bis Byte Nr.:" bzw. im Feld "Anzahl Byte ::" mit der Taste **[Enter]** bestätigt, wird der Schreibvorgang gestartet. Solange noch kein Datenträger im Schreibbereich erkannt wird, blinkt ein Balken in der Mitte der 2. Zeile. Befindet sich ein Datenträger im Schreibbereich des Schreib-/Lesekopfs, hört das Blinken des Balkens auf und die Daten werden auf den Datenträger geschrieben.

```
Datenträg. schreiben
-----
von Byte Nr.:
Anzahl Byte :
```

Maske während dem Schreiben auf den Datenträger



Während der Zeit, in der die Daten vom Datenträger gelesen werden, darf der Datenträger nicht aus dem Schreib-/Lesebereich entfernt werden!

Funktionsbeschreibung BIS M-81_ Datenträger bearbeiten

Fehler beim Schreiben

Tritt während des Schreibens ein Fehler auf, wird eine Fehlermeldung auf dem Display ausgegeben.


Fehlermeldung "Datenträger entfernt"

```
!!!! FEHLER !!!!
-----
Fehler b. Schreiben
Datenträger entfernt
```

Fehlermeldung 1 beim Schreiben

Fehlerursachen:

- Datenträger zu früh aus dem Schreibbereich entfernt
- Abstand Datenträger – Schreib-/Lesekopf zu groß

Nach Bestätigen der Fehlermeldung mit der Taste  befindet man sich wieder im Hauptmenü 1.


Fehlermeldung "Maximale Endadresse überschritten"

```
!!!! FEHLER !!!!
-----
Maximale Endadresse
überschritten
```

Fehlermeldung 2 beim Schreiben

Fehlerursachen:

- Endadresse ist größer als die konfigurierte Kapazität des Datenträgers

Nach Bestätigen der Fehlermeldung mit der Taste  befindet man sich wieder in der Eingabemaske für das Schreiben.

Funktionsbeschreibung BIS M-81_ Datenträger bearbeiten

Fehler beim Schreiben (Fortsetzung)

Fehlermeldung "??? Abstand ???"

```
Datenträg. schreiben
??? Abstand ???
von Byte Nr.: 0
Anzahl Byte : 0600
```

Fehlermeldung 3 beim Schreiben

Fehlerursachen:

- Abstand Datenträger – Schreib-/Lesekopf zu groß
- Endadresse (= Startadresse + Anzahl Byte) liegt außerhalb der Kapazität des Datenträgers
- Datenträger defekt

Der Handy-Programmer wiederholt den Schreibvorgang noch 2 Mal. Schlägt auch der 3. Schreibvorgang fehl, wird mit dieser Fehlermeldung abgebrochen:

```
!!!! FEHLER !!!!
-----
Fehler b. Schreiben
Abstand / sonstig.
```

Fehlermeldung 4 beim Schreiben

Funktionsbeschreibung BIS M-81_ Datenträger bearbeiten

Daten editieren

3

Über den Menüpunkt 3 "Editieren" im Hauptmenü 1 wird der Editor aufgerufen.

In der folgenden Anzeige wird die Einsprung-Adresse in den Editor eingegeben (0-0751).

```

Edit Cursor Position
-----
von Byte Nr.: 0000
    
```

Die Daten werden abhängig vom eingestellten Format als HEX-Werte, ASCII-Zeichen, BCD-Ziffern oder Binär dargestellt. Das Zeichen, welches vom Cursor unterstrichen ist, kann editiert werden.

```

Edit Byte Nr.: 0049
-----
_ABCDEFGHIJKLMNOPQRST
UVWXYZ123456789abcde
    
```

Editieren der Daten im ASCII-Format

```

Edit Byte Nr.: 0119
-----
_Q1234567890123456789
01234567890123456789
    
```

Editieren der Daten im BCD-Format

```

Edit Byte Nr.: 0163
-----
_Q0 01 02 03 04 05 06
07 08 09 0A 0B 0C 0D
    
```

Editieren der Daten im HEX-Format

```

Edit Byte Nr.: 0003
-----
_Q0110001 00110010
00110011 00110100
    
```

Editieren der Daten im BINÄR-Format

Mit den Tasten **←** und **→** erreicht man das benachbarte Zeichen. Ist HEX eingestellt, wird mit den Tasten **←** und **→** jeweils ein Halbbyte weitergeschaltet.

Mit den Tasten **↑** und **↓** springt man eine Zeile höher oder tiefer. Wird dabei der Darstellungsbereich des Display überschritten, wird automatisch eine Zeile nach oben oder unten gescrollt.

Nach Betätigen der Taste **Enter** befindet man sich wieder im Hauptmenü 1.

Funktionsbeschreibung BIS M-81_ Arbeiten mit Dateien

Dateimenü

4

Der Handy-Programmer BIS M-81_ verfügt über einen nicht-flüchtigen EEPROM-Speicher. In diesem EEPROM können von einem Datenträger gelesene Daten unter einem Dateinamen gespeichert werden. Der Dateiname ist auf 8 Zeichen begrenzt. Es können 15 Dateien mit einem Datenumfang von maximal 752 Byte je Datei gespeichert werden.

```

--- Dateimenü ---
1 Laden      4 Lösche
2 Speichern  5 Verzei
3 Datenüber  6 <--
    
```

Dateimenü

Dateiformat

Datenträgerdateien haben den folgenden Aufbau:

Startadresse der Daten auf dem Datenträger 4 Byte ASCII
 Anzahl Byte 4 Byte ASCII
 Datenträgerdaten datentransparent; Datenumfang wie in Anzahl Byte angegeben

Beispiel: Datenträgerdatei ab Adresse 75 mit 16 Byte Daten im ASCII-Format
 Daten: ABCDEFGHIJKLMNOP
 Inhalt der Datei: 00750016ABCDEFGHIJKLMNOP

Datei laden

4 1

Über den Menüpunkt 1 "Laden" im Dateimenü kann eine der im EEPROM-Speicher gespeicherten Dateien in den Arbeitsspeicher des Handy-Programmers geladen werden.

Die Daten dieser Datei können dann über den Menüpunkt 3 "Editieren" aus dem Hauptmenü 1 weiterverarbeitet werden.

```

Bitte Namen eingeben
-
    
```

Datei aus EEPROM-Speicher laden

Nach dem Laden der Daten werden im Display die Startadresse und die Endadresse oder Anzahl Byte angezeigt. Mit der Taste **Enter** wird die Anzeige beendet.

Die Daten dieser Datei können dann über den Menüpunkt 3 "Editieren" aus dem Hauptmenü 1 weiterverarbeitet werden.

```

geladene Daten
-----
von Byte Nr.: 0
Anzahl Byte: 256
    
```

Funktionsbeschreibung BIS M-81_ Arbeiten mit Dateien

Datei speichern

4 2

Über den Menüpunkt 2 "Speichern" im Dateimenü können diejenigen Daten, die zuletzt von einem Datenträger gelesen wurden und sich noch im Arbeitsspeicher des Handy-Programmers befinden, als Datei im EEPROM-Speicher abgespeichert werden.

```
Bitte Namen eingeben
-
```

Datei im EEPROM-Speicher abspeichern

Datenaustausch über die serielle Schnittstelle

4 3

Über den Menüpunkt 3 "Datenüber" (für Datenübertragung) können Dateien über die serielle Schnittstelle von einem PC / Laptop in den Handy-Programmer übertragen werden und umgekehrt. Weitere Einzelheiten siehe ¶ 42ff.

Datei(en) löschen

4 4

Über den Menüpunkt 4 "Lösch" im Dateimenü können gespeicherte Dateien gelöscht werden.

Mit dem Menüpunkt 1 wird eine einzelne Datei gelöscht. Es öffnet sich eine Eingabemaske, in welcher der Dateiname einzugeben ist.

Mit dem Menüpunkt 2 werden alle gespeicherten Dateien gemeinsam gelöscht.

```
--- Löschenmenü ---
1 Einen Eintrag
2 Alle Einträge
6 <---
```

Löschenmenü

Verzeichnis anzeigen

4 5

Über den Menüpunkt 5 "Verzei" (für Verzeichnis) kann die Liste der gespeicherten Dateien angezeigt werden.

Mit der Taste **Enter** wird die Anzeige beendet.

```
Daten1   Daten5
Daten2   Daten6
Daten3
Daten4
```

Verzeichnis

Funktionsbeschreibung BIS M-81_ Spezielle Datenträgerfunktionen

Hauptmenü 2

6

Durch Auswahl des Menüpunkts 6 "Weiter" im Hauptmenü 1 erreicht man das Hauptmenü 2.

```
-- Hauptmenü 2 --
1 Init
2 Serial Number
6 <--
```

Hauptmenü 2

Init-Menü: Datenträger initialisieren

1

```
---- Initmenü ----
1 CRC 6 <--
```

Init-Menü

Nach Auswahl des Menüpunkts 1 "Init" im Hauptmenü 2 gelangt man zum Init-Menü.

CRC

Mit dem Menüpunkt 1 "CRC" wird der Datenträger mit 00_{Hex} beschrieben und dadurch für den Betrieb mit CRC-Datenprüfung vorbereitet.

Serial Number lesen

2

```
Serial 11 22 33 44
Number 55 66 77 88

DTTyp: BIS M-1..-XX
```

Über den Menüpunkt 2 "Serial Number" wird die 4 Byte (Mifare) oder 8 Byte (ISO15693) lange Seriennummer des Datenträgers ausgelesen und im HEX-Format angezeigt. Sie ist nur lesbar. Weiterhin wird der Datenträgertyp angezeigt. XX steht für Balluffdatenträgertyp. ?? wird für Typen angezeigt, die Balluff nicht unterstützt.

Mit **Enter** wird wieder in das Hauptmenü 2 verzweigt.

Funktionsbeschreibung BIS M-81_ / PC Lesen / Schreiben über die serielle Schnittstelle

Neben der tastaturgesteuerten Bearbeitung der Datenträger besteht auch die Möglichkeit, Lese- und Schreibaufträge von einem PC oder einem Laptop über die serielle Schnittstelle des Handy-Programmers auszuführen. Der Handy-Programmer arbeitet dann wie eine BIS-Auswerteeinheit.

Die Verbindung zwischen PC und Handy-Programmer wird durch das Kabel BIS C-522 *) hergestellt. Die Kommunikation zwischen dem Handy-Programmer und dem Laptop / PC erfolgt in einem festgelegten Protokoll (siehe ¶ 30).

Details zu den einzelnen Telegrammen siehe ¶ 31ff.

Datenübertragung aktivieren

4 3

Im Hautmenü 1 wird der Menüpunkt 4 "Datei" gewählt und im darauffolgenden Dateimenü der Menüpunkt 3 "Datenüber" (für Datenübertragung) aktiviert, um diese Funktion zu starten.

```

--- Dateimenü ---
1 Laden      4 Lösche
2 Speichern  5 Verzei
3 Datenüber  6 <--
    
```

Dateimenü

```

- Datenübertragung -
                                6 <--
    
```

Datenübertragung

Datensicherheit auf der RS 232-Schnittstelle

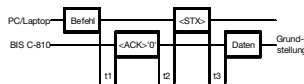
Alle Befehle zum Schreiben und Lesen (L, Z, C und P) sowie der Quit-Befehl (Q) können je nach Konfiguration entweder mit dem Blockcheck BCC oder mit Carriage Return 'CR' abgeschlossen werden. Der Handy-Programmer schließt dann ebenfalls alle übertragenen Zeichen mit der gewählten Abschlussart ab.

*) siehe Zubehörliste auf ¶ 57

Funktionsbeschreibung BIS M-81_ / PC Lesen / Schreiben über die serielle Schnittstelle

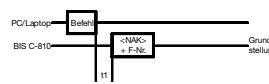
Protokollablauf

Lesen: a) Es tritt kein Fehler auf:



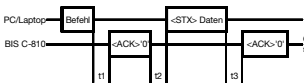
t1 je nach Anzahl zu lesender Byte (siehe ¶ 43)
t2 ≥ 0 (wird vom Handy-Programmer nicht überwacht)
t3 = max. 50 ms

b) Es tritt ein Fehler auf:



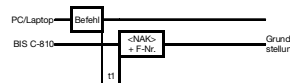
t1 je nach Anzahl zu lesender Byte (siehe ¶ 43) und Fehlerart (empfohlene Überwachungszeit: 15 s)

Schreiben: a) Es tritt kein Fehler auf:



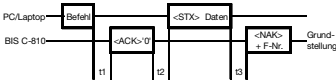
t1 = max. 50 ms
t2 ≥ 0 (wird vom Handy-Programmer nicht überwacht)
t3 je nach Anzahl zu schreibender Byte (siehe ¶ 43)

b) Es tritt ein Fehler im Befehl auf:



t1 = max. 50 ms

c) Es tritt ein Fehler beim Schreiben auf:



t1 = max. 50 ms t2 ≥ 0 (wird vom Handy-Programmer nicht überwacht)
t3 je nach Anzahl zu schreibender Byte (siehe ¶ 43) und Fehlerart (empfohlene Überwachungszeit: 30 s)

Voraussetzung für die Gültigkeit der Darstellungen:
- Der Handy-Programmer muss sich in Grundstellung befinden.
- Vor dem Schreib-/Lesekopf befindet sich ein Datenträger.

Funktionsbeschreibung BIS M-81_ / PC Lesen / Schreiben über die serielle Schnittstelle

Telegrammarten mit zugehörigem Befehl (ASCII-Zeichen)

Für die einzelnen Aufgaben im Handy-Programmer BIS M-81_ existieren spezifische Telegramme. Sie beginnen stets mit dem Befehl, welcher der Telegrammart zugeordnet ist:

- 'L' Lesen des Datenträgers
- 'P' Schreiben auf den Datenträger
- 'C' Schreiben eines konstanten Wertes auf den Datenträger
- 'Q' Neustart des Handy-Programmiers (Quit)
- 'Z' Datenträger für CRC_16-Datenprüfung initialisieren

Bitte beachten Sie:

- Eine Dauerabfrage auf der Schnittstelle ist nicht zulässig!
- Die Mindestwartezeit zwischen zwei Befehlen beträgt 300 ms!

Funktionsbeschreibung BIS M-81_ / PC Lesen / Schreiben über die serielle Schnittstelle

Erklärung einiger Telegramminhalte

Startadresse und Anzahl Byte Die Startadresse (A3, A2, A1, A0) und die Anzahl der zu übertragenden Byte (L3, L2, L1, L0) werden dezimal als ASCII-Zeichen übertragen. Für die Startadresse kann der Bereich 0000 bis 1999 und für die Anzahl Byte 0001 bis 2000 verwendet werden. A3 ... L0 stehen für je ein ASCII-Zeichen.

Bitte beachten Sie: Startadresse + Anzahl Byte dürfen die Datenträgerkapazität nicht überschreiten.

Quittung Die Quittung <ACK> '0' wird vom Identifikations-System gesendet, wenn die seriell übertragenen Zeichen als richtig erkannt wurden und sich ein Datenträger im Arbeitsbereich eines Schreib-/Lesekopfs befindet.
Mit <NAK> + 'Fehler-Nr.' wird quittiert, wenn ein Fehler erkannt wurde oder wenn sich kein Datenträger im Arbeitsbereich des Schreib-/Lesekopfs befindet.

Start Mit <STX> wird die Datenübertragung gestartet.

Übertragene Die Daten werden datentransparent (ohne Datenwandlung) übertragen.

Funktionsbeschreibung BIS M-81_ / PC Lesen / Schreiben über die serielle Schnittstelle

Bildung des Blockcheck BCC

Byte

Der Blockcheck BCC wird als EXOR-Verknüpfung aus den seriell übertragenen Binärzeichen des Telegrammblocks gebildet. Beispiel: Lesen ab Adresse 13, 128 Byte sind zu lesen.

Die Befehlszeile ohne BCC lautet: 'L 0013 0128 10'

BCC wird gebildet:

'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

ergibt als Blockcheck: BCC = 0100 0100 = 'D'

Variante bei Abschluss mit 'CR' anstatt BCC

Bei Bedarf kann der Abschluss mittels Blockcheck BCC durch ein spezielles ASCII-Zeichen ersetzt werden. Dies ist:
- Carriage Return 'CR'

Beispiel: Lesen ab Adresse 13, 128 Byte sind zu lesen

Die Befehlszeile mit 'CR' als Abschluss lautet: 'L 0013 0128 10 CR'

Telegrammaufbau und Beispiele Lesen / Schreiben über die serielle Schnittstelle

Lesen / Schreiben auf den Datenträger

Task	Datenfluss	Befehl	Startadresse des ersten zu übertragenden Byte	Anzahl der zu übertragenden Byte	Abschluss	Quittung 2)	Start zur Übertragung	Daten (von Startadresse bis Startadresse + Anzahl Bytes)	Abschluss	Quittung 2)
Lesen	vom steuernden System zum BIS	'L'	A3 A2 A1 A0 '0' '0' '0' '0' bis Speicherkapazität - 1	L3 L2 L1 L0 '0' '0' '0' '1' bis Speicherkapazität	BCC oder '1' '0' 'CR'		<STX>			
	vom BIS zum steuernden System					<ACK>'0' oder <NAK> + Fehler-Nr.		D1 D2 D3 ... Dn	BCC oder 'CR'	
1)										
Schreiben	vom steuernden System zum BIS	'P'	A3 A2 A1 A0 '0' '0' '0' '0' bis Speicherkapazität - 1	L3 L2 L1 L0 '0' '0' '0' '1' bis Speicherkapazität	BCC oder '1' '0' 'CR'		<STX>	D1 D2 D3 ... Dn	BCC oder 'CR'	
	vom BIS zum steuernden System					<ACK>'0' oder <NAK> + Fehler-Nr.				<ACK>'0' oder <NAK> + Fehler-Nr.
1)										

1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.

2) Als Quittung kommt <ACK> '0', wenn kein Fehler aufgetreten ist, oder <NAK> + 'Fehler-Nr.', wenn ein Fehler aufgetreten ist.

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
Angaben in Hochkommata stellen das / die jeweilige/n Zeichen im ASCII-Code dar.

Telegrammaufbau und Beispiele Lesen / Schreiben über die serielle Schnittstelle

Telegrammbeispiel zu ¶34:
Lesen vom Datenträger mit Blockcheck (BCC)

Es sollen 10 Byte ab Adresse 50 vom Datenträger gelesen werden.
Das Steuersystem sendet 'L 0 0 5 0 0 0 1 0 1 0' BCC (49 Hex)
Adresse des ersten zu lesenden Byte
Anzahl der zu lesenden Byte
Fest im Protokoll
Fest im Protokoll
Der Handy-Programmer quittiert mit <ACK> '0'
Das Steuersystem gibt den Startbefehl <STX>
Der Handy-Programmer liefert die Daten vom Datenträger '1 2 3 4 5 6 7 8 9 A F' BCC (70 Hex)

Telegrammbeispiel zu ¶34:
Schreiben auf den Datenträger mit Blockcheck (BCC)

Es sollen 5 Byte ab Adresse 500 auf den Datenträger geschrieben werden.
Das Steuersystem sendet 'P 0 5 0 0 0 0 0 5 1 0 Q' BCC (51 Hex)
Adresse des ersten zu schreibenden Byte
Anzahl der zu schreibenden Byte
Fest im Protokoll
Fest im Protokoll
Der Handy-Programmer quittiert mit <ACK> '0'
Das Steuersystem gibt den Startbefehl und die Daten <STX> '1 2 3 4 5 3' BCC (33 Hex)
Der Handy-Programmer quittiert mit <ACK> '0'

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Telegrammaufbau und Beispiele Lesen / Schreiben über die serielle Schnittstelle

Schreiben eines konstanten Wertes auf den Datenträger

Dieser Befehl kann zum Löschen eines Datenträgers verwendet werden. Man spart die Zeit zur Übertragung der zu schreibenden Byte.

Task	Datenfluss	Be- fehl	Startadresse des ersten zu übertragen- den Byte	Anzahl der zu über- tragenden Byte	Abschluss	Quittung 2)	Start zur Über- tragung	Daten (von Startadresse bis Startadresse + Anzahl Byte)	Abschluss	Quittung 2)
Schreiben	vom steuernden System zum BIS	'C'	A3 A2 A1 A0 '0 0 0 0' bis Speicherka- pazität -1	L3 L2 L1 L0 '0 0 0 1' bis Speicherka- pazität	'1' '0' 'CR'	BCC oder 'CR'	<STX>	D	BCC oder 'CR'	
	vom BIS zum steuernden System					<ACK> '0' oder <NAK> + F-Nr.				<ACK> '0' oder <NAK> + F-Nr.
			1)						1)	

1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.
2) Als Quittung kommt <ACK> '0', wenn kein Fehler aufgetreten ist, oder <NAK> + 'Fehler-Nr.', wenn ein Fehler aufgetreten ist.

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Telegrammaufbau und Beispiele Lesen / Schreiben über die serielle Schnittstelle

Telegrammbeispiel zu 36:
Schreiben auf den Datenträger mit Blockcheck (BCC)

Es sollen 500 Byte ab Adresse 20 auf den Datenträger mit dem ASCII Datenwert 0 (30 Hex) geschrieben werden.

Das Steuersystem sendet

	'C	0	0	2	0	0	5	0	0	1	0	E'	BCC (46 Hex)
Adresse des ersten zu schreibenden Byte	_____											_____	
Anzahl der zu schreibenden Byte	_____											_____	
Fest im Protokoll	_____											_____	
Fest im Protokoll	_____											_____	

Der Handy-Programmer quittiert mit <ACK> '0'
 Das Steuersystem gibt den Startbefehl und die Daten <STX> '0 2' BCC (32 Hex)
 Der Handy-Programmer quittiert mit <ACK> '0'

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
 Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Telegrammaufbau und Beispiele Lesen / Schreiben über die serielle Schnittstelle

Neustart des Handy-Programmers (Quit)

Durch das Absenden des Telegramms Neustart wird ein in Arbeit befindliches Telegramm abgebrochen und der Handy-Programmer in den Grundzustand gebracht.

Nach der Quittierung dieses Telegramms sind mindestens 1600 ms Pause vorzusehen, bevor ein neues Telegramm gestartet wird.



Wichtig! Der Befehl Quit ist nicht zugelassen, während der Handy-Programmer auf ein Abschlusszeichen wartet (BCC oder 'CR'). In dieser Situation würde Quit als Abschluss- oder Nutzzeichen fehlinterpretiert.

Task	Datenfluss	Befehl	Abschluss	Quittung	Abschluss
Neustart (Quit)	vom steuernden System zum BIS	'Q'	BCC oder 'CR'		
	vom BIS zum steuernden System			'Q'	BCC oder 'CR'
1)					

1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.

Telegrammbeispiel mit Blockcheck (BCC):

Der Handy-Programmer soll in den Grundzustand gebracht werden.

Das Steuersystem sendet 'Q Q' BCC (51 Hex)
 Der Handy-Programmer quittiert mit 'Q Q' BCC (51 Hex)

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
 Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Telegrammaufbau und Beispiele Lesen / Schreiben über die serielle Schnittstelle

Datenträger für die CRC_16-Datenprüfung initialisieren

Mit diesem Telegramm wird ein Datenträger, der sich vor dem aktiven Schreib-/Lesekopf befindet, für die Verwendung bei CRC_16-Datenprüfung initialisiert. Dieses Telegramm muss auch dann erneut gesendet werden, wenn ein CRC-Fehler als Folge aus einem missglückten Schreibauftrag aufgetreten ist, d.h. der Datenträger muss neu initialisiert werden, um ihn wieder verwenden zu können.

Bitte beachten Sie die Tabelle auf 10! Die angegebene Anzahl nutzbarer Byte darf nicht überschritten werden. D.h. die Summe aus Startadresse plus Anzahl Byte darf die nutzbare Datenträger-Kapazität nicht überschreiten!

Task	Datenfluss	Befehl	Startadresse des ersten zu übertragenden Byte	Anzahl der zu übertragenden Byte	Abschluss	Quittung 2)	Start zur Übertragung	Daten (von Startadresse bis Startadresse + Anzahl Byte)	Abschluss	Quittung 2)
CRC_16 Bereich initialisieren	vom steuernden System zum BIS	'Z'	A3 A2 A1 A0 '0 0 0 0' bis nutzbare Byte bei CRC -1	L3 L2 L1 L0 '0 0 0 1' bis nutzbare Byte bei CRC	BCC oder '1' '0' 'CR'		<STX>	D1 D2 D3 ... Dn	BCC oder 'CR'	
	vom BIS zum steuernden System					<ACK>'0' oder <NAK> + Fehler-Nr.				<ACK>'0' oder <NAK> + Fehler-Nr.
				1)					1)	

- 1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.
2) Als Quittung kommt <ACK> '0', wenn kein Fehler aufgetreten ist, oder <NAK> + 'Fehler-Nr.', wenn ein Fehler aufgetreten ist.

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Lesen / Schreiben über die serielle Schnittstelle Fehlernummern

Fehlernummern

BIS M-81_ gibt immer eine Fehlernummer aus. Deren Bedeutung zeigt nachfolgende Tabelle.

Nr.	Fehlerbeschreibung	Auswirkung
2	Fehler beim Lesen	Lesetelegrammabbruch, Handy-Programmer geht in den Grundzustand.
3	Lesen abgebrochen, da der Datenträger entfernt wurde.	Handy-Programmer geht in den Grundzustand.
4	Fehler beim Schreiben	Schreibtelegrammabbruch, Handy-Programmer geht in den Grundzustand.
5	Schreiben abgebrochen, da der Datenträger entfernt wurde.	Handy-Programmer geht in den Grundzustand.
6	Fehler auf der Schnittstelle	Handy-Programmer geht in den Grundzustand. (Paritäts- oder Stopbitfehler)
7	Telegramm-Formatfehler	Handy-Programmer geht in den Grundzustand. Mögliche Formatfehler: - Befehl ist kein 'L', 'P', 'Q', 'C' oder 'Z'. - Startadresse oder Anzahl Byte außerhalb des zugelassenen Bereichs
8	BCC-Fehler, der übertragene BCC ist falsch	Telegrammabbruch, Handy-Programmer geht in den Grundzustand
E	CRC-Fehler	Der CRC der gelesenen Daten stimmt nicht mit dem CRC auf dem Datenträger überein.

ACHTUNG: Durch den abgebrochenen Schreibvorgang könnten neue Daten auf den Datenträger geschrieben worden sein, die unvollständig sein können!

Lesen / Schreiben über die serielle Schnittstelle Schreib-/Lesezeiten

Lesezeiten

Datenträger mit je 16 Byte/Block	BIS M-1_-01	BIS M-1_-02
Zeit zur Datenträgererkennung/serial ID	≤ 20 ms	≤ 30 ms
Lesen von Byte 0 bis 15	≤ 20 ms	≤ 30 ms
für jeweils weitere angebrochene 16 Byte addieren Sie weitere	≤ 10 ms	≤ 15 ms

Schreibzeiten

Datenträger mit je 16 Byte/Block	BIS M-1_-01	BIS M-1_-02
Zeit zur Datenträgererkennung/serial ID	≤ 20 ms	≤ 30 ms
Schreiben von Byte 0 bis 15	≤ 40 ms	≤ 65 ms
für jeweils weitere angebrochene 16 Byte addieren Sie weitere	≤ 30 ms	≤ 45 ms



Schwankungen im ms-Bereich sind möglich.
Elektrische Störeinflüsse können die Schreib-/Lesezeit erhöhen.

Funktionsbeschreibung BIS M-81_ / PC Arbeiten mit Dateien über die serielle Schnittstelle

Datenaustausch über die serielle Schnittstelle

4 | 3

Über den Menüpunkt 3 "Datenüber" (für Datenübertragung) können Dateien über die serielle Schnittstelle von einem PC / Laptop in den Handy-Programmer übertragen werden und umgekehrt. Es besteht die Möglichkeit der Dateiverwaltung. Dazu ist die Software BISHandy.exe auf dem PC / Laptop zu installieren, indem die Datei setup.exe ausgeführt wird. Beide Dateien befinden sich auf der mitgelieferten CD-ROM.



Im Hautmenü 1 wird der Menüpunkt 4 "Datei" gewählt und im darauffolgenden Menü der Menüpunkt 3 "Datenüber", um die Funktion "Datenübertragung" zu aktivieren.

Datensicherheit auf der RS 232-Schnittstelle

Jeder der Befehle (D, G, T, K und E) wird immer mit dem Blockcheck BCC abgeschlossen. Der Handy-Programmer schließt ebenfalls alle übertragenen Zeichen mit dem BCC ab.

Telegrammarten mit zugehörigem Befehl (ASCII-Zeichen)

Für die einzelnen Aufgaben zur Datenverwaltung im Handy-Programmer BIS M-81_ existieren spezifische Telegramme. Sie beginnen stets mit dem Befehl, welcher der Telegrammart zugeordnet ist:

- 'T' Datei an den Handy-Programmer senden
- 'G' Datei aus dem Handy-Programmer laden
- 'D' Dateiverzeichnis aus dem Handy-Programmer laden
- 'E' Eine bestimmte Datei im Handy-Programmer löschen
- 'K' Alle Dateien im Handy-Programmer löschen

Bitte beachten Sie:

- Eine Dauerabfrage auf der Schnittstelle ist nicht zulässig!
- Die Mindestwartezeit zwischen zwei Befehlen beträgt 300 ms!
- Bei diesen Telegrammen wird ausschließlich Blockcheck BCC für die Datensicherheit verwendet (unabhängig von der Konfigurationseinstellung).

Arbeiten mit Dateien über die serielle Schnittstelle Telegrammaufbau und Beispiele

Datei vom PC an den Handy-Programmer senden

Dieser Befehl wird dazu verwendet eine in einem PC / Laptop gespeicherte Datei an den Handy-Programmer zu senden.

Task	Datenfluss	Be- fehl	Dateiname 3)	Startadresse auf dem Datenträger	Anzahl Byte auf dem Datenträger	Abschluss	Quittung 2)	Start zur Über- tragung	Daten (von Startadresse bis Startadresse + Anzahl Byte)	Abschluss	Quittung 2)
Datei an den Handy- Programmer senden	vom steuernden System zum BIS	'T'	'D7 ... D0' 'MUSTER_ _'	A3 A2 A1 A0 '0 0 0 0' bis Speicherka- pazität -1	L3 L2 L1 L0 '0 0 0 1' bis Speicherka- pazität	BCC		<STX>	D1 D2 D3 ... Dn	BCC	
	vom BIS zum steuernden System						<ACK>'0' oder <NAK> + Fehler-Nr.				<ACK>'0' oder <NAK> + Fehler-Nr.
				1)							
								1)			

- 1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.
- 2) Als Quittung kommt <ACK> '0', wenn kein Fehler aufgetreten ist, oder <NAK> + 'Fehler-Nr.', wenn ein Fehler aufgetreten ist.
- 3) Hat der Dateiname weniger als 8 Zeichen, muss er mit Leerzeichen (20_{hex}) auf 8 Zeichen aufgefüllt werden.

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Arbeiten mit Dateien über die serielle Schnittstelle Telegrammaufbau und Beispiele

Datei vom Handy-Programmer in den PC / Laptop auslesen

Dieser Befehl wird dazu verwendet eine im Handy-Programmer gespeicherte Datei auszulesen.

Task	Datenfluss	Be- fehl	Dateiname 3)	Abschluss	Quittung 2)	Start zur Über- tragung	Startadresse auf dem Datenträger	Anzahl Byte auf dem Datenträger	Daten (von Startadresse bis Startadresse + Anzahl Byte)	Abschluss	
Datei vom Handy- Programmer auslesen	vom steuernden System zum BIS	'G'	'D7 ... D0' 'MUSTER_ _'	BCC		<STX>					
	vom BIS zum steuernden System				<ACK>'0' oder <NAK> + Fehler-Nr.		A3 A2 A1 A0 '0 0 0 0' bis Speicherka- pazität -1	L3 L2 L1 L0 '0 0 0 1' bis Speicherka- pazität	D1 D2 D3 ... Dn	BCC	
				1)							
								1)			

- 1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.
- 2) Als Quittung kommt <ACK> '0', wenn kein Fehler aufgetreten ist, oder <NAK> + 'Fehler-Nr.', wenn ein Fehler aufgetreten ist.
- 3) Hat der Dateiname weniger als 8 Zeichen, muss er mit Leerzeichen (20_{hex}) auf 8 Zeichen aufgefüllt werden.

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Arbeiten mit Dateien über die serielle Schnittstelle Telegrammaufbau und Beispiele

Telegrammbeispiel zu 43:
Datei an den Handy-Programmer senden mit Blockcheck (BCC)

Die Datei MUSTER1 soll an den Handy-Programmer gesendet werden. Auf den Datenträger soll ab Adresse 27 die Anzahl von 5 Byte geschrieben werden.

Das Steuersystem sendet 'T MUSTER1_ 0 0 2 7 0 0 0 5 M' BCC (4D Hex)

Dateiname (auf 8 Byte aufgefüllt *)
 Adresse des ersten Byte auf dem Datenträger
 Anzahl Byte

Der Handy-Programmer quittiert mit <ACK> '0'
 Das Steuersystem gibt den Startbefehl und die Daten <STX> '12345 3' BCC (33 Hex)
 Der Handy-Programmer quittiert mit <ACK> '0'

Telegrammbeispiel zu 44:
Datei aus dem Handy-Programmer auslesen mit Blockcheck (BCC)

Die Datei MUSTER1 soll aus dem Handy-Programmer ausgelesen werden. Auf dem Datenträger soll ab Adresse 27 die Anzahl von 5 Byte gelesen werden.

Das Steuersystem sendet 'G MUSTER1_ ^' BCC (5E Hex)

Dateiname (auf 8 Byte aufgefüllt *)

Der Handy-Programmer quittiert mit <ACK> '0'
 Das Steuersystem sendet <STX>
 Der Handy-Programmer sendet '0027000512345 1' BCC (33 Hex)

*) Auch ein Leerzeichen „_“, das zum Auffüllen benutzt wird, muss im BCC berücksichtigt werden.

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
 Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Arbeiten mit Dateien über die serielle Schnittstelle Telegrammaufbau und Beispiele

Dateiverzeichnis vom Handy-Programmer in den PC / Laptop auslesen

Dieser Befehl wird dazu verwendet, um das im Handy-Programmer gespeicherte Dateiverzeichnis auszulesen.

Task	Datenfluss	Befehl	Abschluss	Quittung 2)	Start zur Übertragung	Daten	Abschluss
Verzeichnis aus dem Handy-Programmer auslesen	vom steuernden System zum BIS	'D'	BCC		<STX>		
	vom BIS zum steuernden System			<ACK>'0' oder <NAK> + Fehler-Nr.		D1 D2 D3 ... Dn BCC	
				1)			1)

- 1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.
- 2) Als Quittung kommt <ACK> '0', wenn kein Fehler aufgetreten ist, oder <NAK> + 'Fehler-Nr.', wenn ein Fehler aufgetreten ist.

Telegrammbeispiel:
Dateiverzeichnis aus dem Handy-Programmer auslesen mit Blockcheck (BCC)

Das Dateiverzeichnis soll aus dem Handy-Programmer ausgelesen werden. Auf dem Handy-Programmer sind 2 Dateien gespeichert.

Das Steuersystem sendet 'D D' BCC (44 Hex)

Der Handy-Programmer quittiert mit <ACK> '0'
 Das Steuersystem sendet <STX>
 Der Handy-Programmer sendet '0002MUSTER1_00270005 MUSTER2_00350012 EOT'
 BCC (04 Hex)

*) Auch ein Leerzeichen „_“, das zum Auffüllen benutzt wird, muss im BCC berücksichtigt werden.

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
 Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Arbeiten mit Dateien über die serielle Schnittstelle Telegrammaufbau und Beispiele

Eine Datei im Handy-Programmer löschen

Dieser Befehl wird dazu verwendet eine im Handy-Programmer gespeicherte Datei zu löschen.

Task	Datenfluss	Befehl	Dateiname 3)	Abschluss	Quittung 2)
Eine Datei im Handy-Programmer löschen	vom steuernden System zum BIS	'E'	'D7 ... D0' 'MUSTER_ _'	BCC	
	vom BIS zum steuernden System				<ACK>'0' oder <NAK> + Fehler-Nr.
1)					

- 1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.
- 2) Als Quittung kommt <ACK> '0', wenn kein Fehler aufgetreten ist, oder <NAK> + 'Fehler-Nr.', wenn ein Fehler aufgetreten ist.
- 3) Hat der Dateiname weniger als 8 Zeichen, muss er mit Leerzeichen (20^{hex}) auf 8 Zeichen aufgefüllt werden.

Telegrammbeispiel:
Datei im Handy-Programmer löschen mit Blockcheck (BCC)

Die Datei MUSTER1 soll aus dem Verzeichnis im Handy-Programmer gelöscht werden.

Das Steuersystem sendet 'E MUSTER1_ _ ' BCC (5C Hex)

Dateiname (auf 8 Byte aufgefüllt) *)

Der Handy-Programmer quittiert mit <ACK> '0'

*) Auch ein Leerzeichen „_“, das zum Auffüllen benutzt wird, muss im BCC berücksichtigt werden.

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Arbeiten mit Dateien über die serielle Schnittstelle Telegrammaufbau und Beispiele

Alle Dateien im Handy-Programmer löschen

Dieser Befehl wird dazu verwendet eine im Handy-Programmer gespeicherte Datei zu löschen.

Task	Datenfluss	Befehl	Abschluss	Quittung 2)
Alle Dateien im Handy-Programmer löschen	vom steuernden System zum BIS	'K'	BCC	
	vom BIS zum steuernden System			<ACK>'0' oder <NAK> + Fehler-Nr.
1)				

- 1) Der Befehl Quit ist an dieser Stelle nicht zugelassen.
- 2) Als Quittung kommt <ACK> '0', wenn kein Fehler aufgetreten ist, oder <NAK> + 'Fehler-Nr.', wenn ein Fehler aufgetreten ist.
- 3) Hat der Dateiname weniger als 8 Zeichen, muss er mit Leerzeichen (20^{hex}) auf 8 Zeichen aufgefüllt werden.

Telegrammbeispiel:
Alle Dateien im Handy-Programmer löschen mit Blockcheck (BCC)

Alle Dateien sollen aus dem Handy-Programmer gelöscht werden.

Das Steuersystem sendet 'K K' BCC (4B Hex)

Der Handy-Programmer quittiert mit <ACK> '0'

Angaben in spitzen Klammern stellen ein Steuerzeichen dar.
Angaben in Hochkommata stellen das/die jeweilige/n Zeichen im ASCII-Code dar.

Arbeiten mit Dateien über die serielle Schnittstelle Fehlernummern

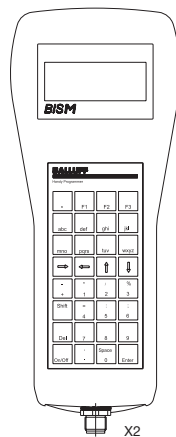
Fehlernummern

BIS M-81_ gibt immer eine Fehlernummer aus. Deren Bedeutung zeigt nachfolgende Tabelle.

Fehler-Nr.	Fehlerbeschreibung	Fehlerursache
0 = 30 Hex	Kein Fehler	Nur in Verbindung mit <ACK>
Q = 51 Hex	Timeout	Zeichenverzugszeit von 1 s überschritten Schnittstellenproblem
R = 52 Hex	Eintrag-Fehler	Angegebene Datei nicht vorhanden
S = 53 Hex T = 54 Hex	Memory-Fehler	Speicherende des Handy-Programmers erreicht
V = 56 Hex	Schnittstellenfehler	Fehlerhafte Einstellung bei den Schnittstellenparametern (Baudraten-, Parity-, Datenbit- oder Stoppbitfehler)
W = 57 Hex	Protokoll-Formatfehler	Protokollformat nicht eingehalten
X = 58 Hex	BCC ist falsch	Falscher BCC erhalten

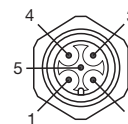
BIS M-810 Schnittstelleninformationen

BIS M-810
Integrierte Antenne



**Serielle Schnittstelle
RS 232
Schreib-/Lesekopf-
anschluss**

X2, serielle Schnittstelle RS 232

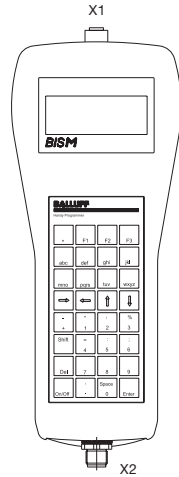


Pin	Funktion
1	n.c.
2	TxD
3	GND
4	RxD
5	n.c.

n.c. = nichts anschließen

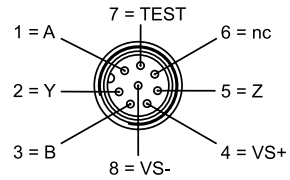
BIS M-811 Schnittstelleninformationen

BIS M-811
Einbaustecker zum Anschluss von Standard BIS M-3_ _ Schreib-/Leseköpfen (ab Hardware-Stand V2.0)

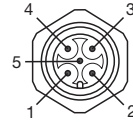


Serielle Schnittstelle RS 232
Schreib-/Lesekopfanschluss

X1, Anschluss für externen Schreib-/Lesekopf für BIS M-811



X2, serielle Schnittstelle RS 232

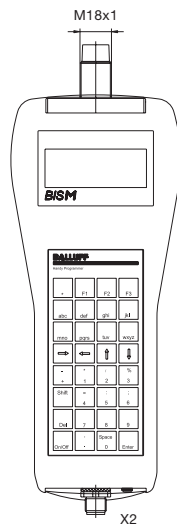


Pin	Funktion
1	n.c.
2	TxD
3	GND
4	RxD
5	n.c.

n.c. = nichts anschließen

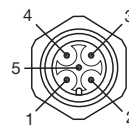
BIS M-812 Schnittstelleninformationen

BIS M-812
Integrierte Antenne mit M18 Gehäuse



Serielle Schnittstelle RS 232
Schreib-/Lesekopfanschluss

X2, serielle Schnittstelle RS 232



Pin	Funktion
1	n.c.
2	TxD
3	GND
4	RxD
5	n.c.

n.c. = nichts anschließen

BIS M-81_ Akku laden

Akku laden

Es bestehen zwei Möglichkeiten den Akku des Handy-Programmers zu laden:

Im eingebauten Zustand: Der Klinkenstecker 2,5 mm des Ladegeräts BIS C-701/A *) wird an der Buchse X2 an der Unterseite des Handy-Programmers eingesteckt.

Im ausgebauten Zustand: Hierzu wird der Akku ausgebaut (siehe unten) und in die Ladeschale BIS C-702-A *) eingesetzt.

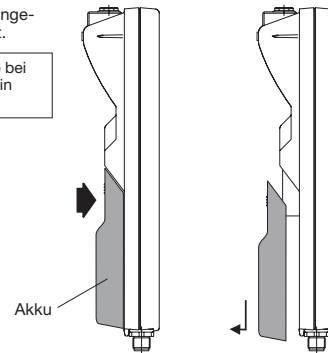
Akku aus-/einbauen

Um den Akku aus dem Handy-Programmer auszubauen, wird die Entriegelung hineingedrückt und der Akku ein kurzes Stück nach unten geschoben, bis die Führungsnasen den Akku freigeben und er abgenommen werden kann (siehe Abbildung).

Nach dem Ladevorgang wird der Akku wieder eingesetzt und nach oben geschoben, bis er einrastet.



Geben Sie verbrauchte Batterien oder Akkus bitte bei einer Sammelstelle ab. Werfen Sie diese **niemals** in Gewässer oder in den Hausmüll.



*) siehe Zubehörliste auf 157

Akku ausbauen

deutsch BALLUFF 53

BIS M-81_ Technische Daten

Gehäuse	Gehäusematerial	Kunststoff ABS	
	Abmessungen	ca. 97 x 232 x 55 mm	
	Gewicht mit Akku	ca. 420 g	
	Tastatur	32 Tasten, 4 x 8 mm, alphanumerisch und Sonderzeichen	
	Anzeige	LCD-Display, 80 Zeichen / 4 Zeilen	
Anschlüsse	Serielle Schnittstelle RS 232	Einbaustecker 5-polig	
	Lesekopfanschluss	BIS M-810 BIS M-811 BIS M-812	interne Antenne Einbaustecker 8-polig interne Antenne im M18 Gehäuse
	Ladebuchse	Klinkenbuchse 2,5 mm geeignet für BIS C-701-A	
Schutzart	Schutzart	IP 40 (in angeschlossenem Zustand)	
	Temperaturbereich	Umgebungstemperatur	0 °C bis +50 °C

BIS M-81_ Technische Daten

Elektrische Daten	Spannungs- und Stromversorgung	2,4 V Akkupack NiMH
	Kapazität	1650 mAh
	max. Ladespannung	5 – 10 V, intern begrenzt
	Ladestrom	ca. 400 mA
Stromverbrauch	Ladezeit	ca. 4 h
	Stromverbrauch, normaler Betrieb	ca. 65 mA
	Stromverbrauch, Lesen/Schreiben	ca. 275 mA



Mit dem CE-Zeichen bestätigen wir, dass unsere Produkte den Anforderungen der EG-Richtlinie 89/336/EWG (EMV-Richtlinie)

und des EMV-Gesetzes entsprechen. In unserem EMV-Labor, das von der DATech für Prüfungen der elektromagnetischen Verträglichkeit akkreditiert ist, wurde der Nachweis erbracht, dass die Balluff-Produkte die EMV-Anforderungen der Fachgrundnorm EN 61000-6-4 (Emission), EN 61000-6-2 (Störfestigkeit) erfüllen.

BIS M-81_ Bestellinformationen

Typenschlüssel

BIS M-81_-0-003

- Balluff Identifikations-System
- Baureihe M
- Hardware-Typ
- 810 = interne Antenne
mit Tastatur und Display, zum Anschluss an PC oder Laptop mit 9-poligem SUB-D-Stecker (seriell)
- 811 = Einbaustecker 8-polig, zum Anschluss von Standard Schreib-/Leseköpfen BIS M-3_ _ (ab Hardware-Stand V2.0) über Anschlusskabel, mit Tastatur . . .
- 812 = interne Antenne im M18 Gehäuse
mit Tastatur und Display, zum Anschluss an PC oder Laptop mit 9-poligem SUB-D-Stecker (seriell)
- Schnittstelle
- 0 = RS 232 (V.24 Schnittstelle)
- Software-Typ
- 003 = Balluff Software für BIS M-81_-...

BIS M-81_ Zubehör

Zubehör
(im Lieferumfang
enthalten)

Die Handlesegeräte werden mit einer Schutztasche ausgeliefert.

Zubehör
(optional, nicht im
Lieferumfang)

Artikel	Bestellbezeichnung
Ladegerät (siehe auch ¶ 58)	BIS C-701-A
Ladeschale (siehe auch ¶ 59)	BIS C-702-A
Schnittstellen-Kabel für den Anschluss des Handy-Programmers BIS M-81_ an eine 9-polige SUB-D COM-Schnittstelle an einem Laptop oder PC	2 m BIS C-522-PVC-02 3 m BIS C-522-PVC-03
Zusätzlicher Akku (siehe auch ¶ 60)	125997
Für BIS M-811	
Anschlussstecker für Schreib-/Leseköpfe BIS M-3_ (ohne Kabel)	BKS-S117-00
Anschlusskabel für Schreib-/Leseköpfe BIS M-3_ (max. 2 m) *)	BKS-S115-PU-02
Leseköpfe BIS M-3_ (siehe Katalog) ab Hardware-Stand V2.0	

*) ein Stecker angespritzt
anderes Ende offen (für BKS-S117-00)

deutsch BALLUFF 57

BIS M-81_ Zubehör

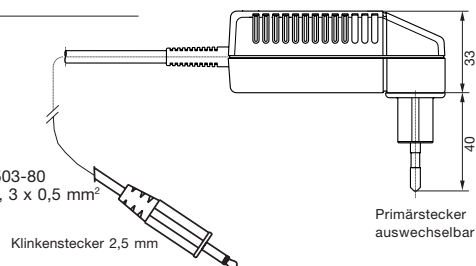
Zubehör
(optional, nicht im
Lieferumfang)

Ladegerät BIS C-701-A
bestehend aus Lade-/Netzteil mit Klinkenstecker 2,5 mm

Technische Daten

Merkmal	Wert / Einheit / Bedingung
$U_{\text{Primär}}$	100 - 240 V _{AC}
$f_{\text{Primär}}$	47 - 63 Hz
$U_{\text{Sek.}}$	6 V _{DC} ±5%
$I_{\text{Sek.}}$	2,1 A
T_A	-40 ... +70 °C
Primärstecker auswechselbar	EURO, UK, USA (im Lieferumfang enthalten)
Zulassung	CE, UL

Material: PPE-V1-125°C
Farbe: schwarz
Gehäuse: MPP10
Ausgangsleitung: 11.7892.503-80
Länge: 2000 mm, 3 x 0,5 mm²



BIS M-81_ Zubehör

Zubehör
(optional, nicht im
Lieferumfang)

Technische Daten

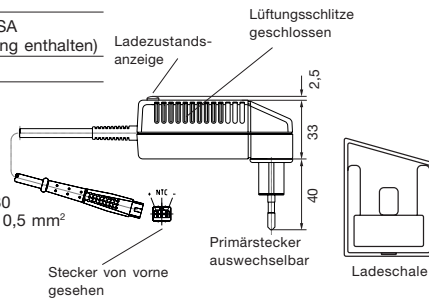
Ladeschale BIS C-702-A

bestehend aus Lade-/Netzteil mit Ladezustandsanzeige und Ladeschale

Merkmal	Wert / Einheit / Bedingung
$U_{\text{Primär}}$	100 - 240 V _{AC}
$F_{\text{Primär}}$	50 - 60 Hz
$U_{\text{Sek.}}$	10 V _{DC} ±5%
$I_{\text{Sek.}}$	800 mA
T_A	-40 ... +70 °C
Primärstecker auswechselbar	EURO, UK, USA (im Lieferumfang enthalten)
Zulassung	CE, UL

Ladezustandsanzeige:
LED rot Akku geladen
LED rot blinkend Akku lädt

Material: PPE-V1-125°C
Farbe: schwarz
Gehäuse: MPP10
Ausgangsleitung: 11.7892.503-80
Länge: 2000 mm, 3 x 0,5 mm²



BIS M-81_ Zubehör

Zubehör
(optional, nicht im
Lieferumfang)

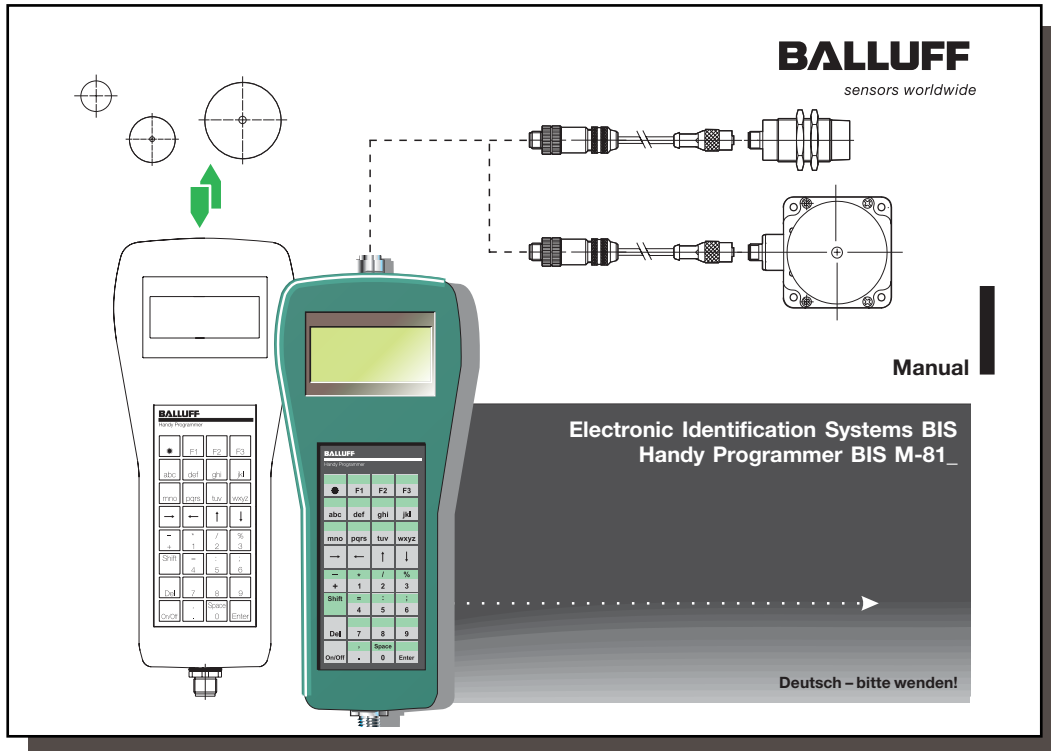
Zusätzlicher Akku

Spannungs- und Stromversorgung 2,4 V Akkupack NiMH
Kapazität 1650 mAh

Anhang, ASCII-Tabelle

Deci- mal	Hex	Control Code	ASCII	Deci- mal	Hex	Control Code	ASCII	Deci- mal	Hex	ASCII	Deci- mal	Hex	ASCII	Deci- mal	Hex	ASCII	Deci- mal	Hex	ASCII
0	00	Ctrl @	NUL	22	16	Ctrl V	SYN	44	2C	,	65	41	A	86	56	V	107	6B	k
1	01	Ctrl A	SOH	23	17	Ctrl W	ETB	45	2D	-	66	42	B	87	57	W	108	6C	l
2	02	Ctrl B	STX	24	18	Ctrl X	CAN	46	2E	.	67	43	C	88	58	X	109	6D	m
3	03	Ctrl C	ETX	25	19	Ctrl Y	EM	47	2F	/	68	44	D	89	59	Y	110	6E	n
4	04	Ctrl D	EOT	26	1A	Ctrl Z	SUB	48	30	0	69	45	E	90	5A	Z	111	6F	o
5	05	Ctrl E	ENQ	27	1B	Ctrl [ESC	49	31	1	70	46	F	91	5B	[112	70	p
6	06	Ctrl F	ACK	28	1C	Ctrl \	FS	50	32	2	71	47	G	92	5C	\	113	71	q
7	07	Ctrl G	BEL	29	1D	Ctrl]	GS	51	33	3	72	48	H	93	5D]	114	72	r
8	08	Ctrl H	BS	30	1E	Ctrl ^	RS	52	34	4	73	49	I	94	5E	^	115	73	s
9	09	Ctrl I	HT	31	1F	Ctrl _	US	53	35	5	74	4A	J	95	5F	_	116	74	t
10	0A	Ctrl J	LF	32	20		SP	54	36	6	75	4B	K	96	60	`	117	75	u
11	0B	Ctrl K	VT	33	21		!	55	37	7	76	4C	L	97	61	a	118	76	v
12	0C	Ctrl L	FF	34	22		*	56	38	8	77	4D	M	98	62	b	119	77	w
13	0D	Ctrl M	CR	35	23		#	57	39	9	78	4E	N	99	63	c	120	78	x
14	0E	Ctrl N	SO	36	24		\$	58	3A	:	79	4F	O	100	64	d	121	79	y
15	0F	Ctrl O	SI	37	25		%	59	3B	;	80	50	P	101	65	e	122	7A	z
16	10	Ctrl P	DLE	38	26		&	60	3C	<	81	51	Q	102	66	f	123	7B	{
17	11	Ctrl Q	DC1	39	27		'	61	3D	=	82	52	R	103	67	g	124	7C	
18	12	Ctrl R	DC2	40	28		(62	3E	>	83	53	S	104	68	h	125	7D	}
19	13	Ctrl S	DC3	41	29)	63	3F	?	84	54	T	105	69	i	126	7E	~
20	14	Ctrl T	DC4	42	2A		*	64	40	@	85	55	U	106	6A	j	127	7F	DEL
21	15	Ctrl U	NAK	43	2B		+												

1



2

No. 854 610 D/E • Edition 0703
Subject to modification.

Writing convention:

Control characters to be transmitted are in angle brackets.

Characters to be transmitted in ASCII code are enclosed in apostrophes.

Example: <STX> '1 2 3 4 5 6' BCC

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

Content

Safety Advisory	4
Introduction BIS M Identification System	5/6
Basic Knowledge for the Application	7-11
Startup	12/13
Configuration	14-17
Function Description BIS M-81_	
Processing Data Carriers	18-25
Working with Files	26/27
Special Data Carrier Functions	28
Function Description BIS M-81_ / PC	
Reading / Writing over the Serial Interface	29-33
Telegram Structure and Examples	34-39
Error Numbers	40
Read/write Times	41
Working with Files over the Serial Interface	42
Telegram Structure and Examples	43-48
Error Numbers	49
Interface Information	
BIS M-810	50
BIS M-811	51
BIS M-812	52
Battery Charging	53
Technical Data	54/55
Ordering Information	
Ordering Code	56
Accessories	57-60
Appendix, ASCII Table	61

Safety advisory

Approved operation	Series BIS M-81_ Handy Programmer along with the other BIS M system components comprise an identification system and may only be used for this purpose in an industrial environment in conformity with Class A of the EMC Law.
Installation and operation	Installation and operation should be carried out only by trained personnel. Unauthorized work and improper use will void the warranty and liability. When installing the Handy Programmer, follow the chapter "Start-up" closely.
Use and testing	Prevailing safety regulations must be adhered to when using the Identification System. In particular, steps must be taken to ensure that a failure of or defect in the identification system does not result in hazards to persons or equipment. This includes maintaining the specified ambient conditions and regular testing for functionality of the Identification System including all its associated components.
Function faults	Should there ever be indications that the identification system is not working properly, it should be taken out of commission and secured from unauthorized use.
Scope	This manual applies to Handy Programmer in the series BIS M-810-0-003... and BIS M-812-0-003... (Software version V1.2, Hardware version V2.0 and higher) as well as BIS M-811-0-003... (Software version V1.2, Hardware version V1.1 and higher).

Introduction BIS M Identification System

This manual is designed to assist the user of the Handy Programmer BIS M-81_ in setting up the control program and installing and starting up the components of the BIS M Identification System, and to assure rapid, trouble-free operation.

Principles

The BIS M Identification System belongs in the category of **non-contact systems for reading and writing.**

This dual function permits applications for not only transporting information in fixed-programmed data carriers, but also for gathering and passing along up-to-date information as well.

Applications

Some of the notable areas of application include

- **for controlling material flow in production processes**
(e.g. in model-specific processes),
for workpiece conveying in transfer lines,
in data gathering for quality assurance ,
for gathering safety-related data;
- **in tool coding and monitoring;**
- **in equipment organization;**
- **in storage systems for monitoring inventory movement;**
- **in transporting and conveying systems;**
- **in waste management for quantity-based fee assessment.**

Using the portable Identification System with the BIS M-81_ Handy Programmer allows data carrier service functions such as checking, correcting or initializing. The data read on the equipment can if needed be stored in the Handy Programmer and transmitted to and then analyzed later on a PC.

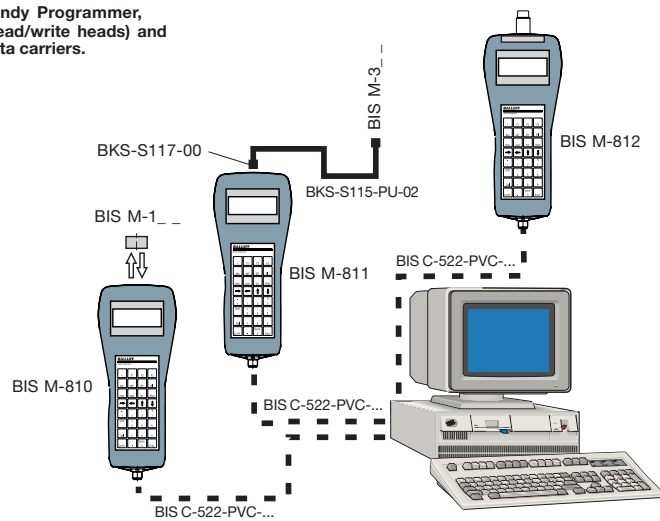
Introduction BIS M Identification System

System components

The main components of the BIS M Identification System are

- **Handy Programmer,**
- **(Read/write heads) and**
- **Data carriers.**

Configuration with Handy Programmer





Schematic
representation of an
identification system
(example)

BIS M-81_

Basic knowledge for the application

Operation, keyboard and display

The Handy Programmer features menu-driven operation. Entries are made on a membrane keypad with 32 keys. Some keys have multiple functions. There are two types of multiple function:

- 2 functions Switch using the  key
- 3 letters press 1x = 1st letter, 2x = 2nd letter, 3x = 3rd letter
- Switching to capital letters using the  key

The LCD display where all data are displayed has 4 lines with 20 characters each.

Languages German/English

From the main menu you can press Shift+F1 to change the language. The selected language is only retained however until the next time the Handy Programmer is turned off. To make the selection permanent, you must select it in the configuration menu.

Menu/Controls

The main functions of the unit are shown in Main Menu 1. Each sub-menu is indicated by a number. Pressing the appropriate number key opens the corresponding menu and the associated window. In some cases the function is carried out immediately, in other cases you must first make some menu selections.

For quick orientation, the key / key sequence is always indicated in the manual for directly reaching the respective function or menu, always starting from Main Menu 1.

```

--- Main menu ---
1 READ      4 File
2 WRITE     5 Config
3 Edit      6 More
    
```

Main menu 1



```

-- Main Menu 2 --
1 Init
2 Serial Number
6 <--
    
```

Main menu 2

BIS M-81_

Basic knowledge for the application

Battery operation

The device is powered by a rechargeable battery. The current draw when reading or writing is 275 mA. When there is no read/write procedure in process, the unit has a current draw of 65 mA.

The system components are powered by the Handy Programmer. The data carrier represents an independent unit, and does not require any cable-supplied power but rather gets its energy from the read/write head. The latter sends a continuous carrier signal which powers the data carrier as soon as the required distance is reached. The read/write procedure takes place during this phase, and may be static or dynamic.



Selecting read/write heads on the BIS M-811

Various read/write heads *) are available for the Handy Programmer. These are connected to the Handy Programmer using a jack plug. Which read/write head *) model is used depends on the data carrier type used. The BKS-S115-PU-02 *) adapter cable can be used to connect a series BIS M-3_ _ read/write head (Hardware version V2.0 and higher) .

Dialog mode

The Handy Programmer writes to or reads from the data carrier through the read/write head. To prepare data for writing or change data that have been read, the Handy Programmer features an editor function. The data can be read from or written to the data carrier in ASCII, HEX, BCD or BINARY format.

Switch formats for the displayed data by pressing the  and  keys:

Key	Format	Key	Format
	>ASCII > BCD > BINARY		HEX

*) See Accessories on 157

BIS M-81_ Basic knowledge for the application

Data security for written and read data

When sending data between the read/write head and the data carrier a procedure is required for recognizing whether the data were correctly read or written.

The processor is supplied with standard Balluff procedure of double reading and comparing. In addition to this procedure a second alternative is available: CRC_16 data checking. Here a test code is written to the data carrier, allowing data to be checked for validity at any time or location.

Advantages of CRC_16	Advantages of double reading
Data checking even during the non-active phase (data carrier outside read/write head zone).	No bytes on the data carrier need to be reserved for storing a check code.
Shorter read times since each page is read only once.	Shorter write times since no CRC needs to be written.

Since both variations have their advantages depending on the application, the user is free to select which method of data checking he wishes to use (see Configuration on ¶ 15 "Carrier data").

It is not permitted to operate the system using both check procedures!

BIS M-81_ Basic knowledge for the application

Data security for written and read data

To be able to use the CRC check method, the data carriers must be initialized. You either use data carriers with the factory default data set (all data are 0), or you must initialize the data carrier. There are two ways of doing this:

- Initializing the data carrier using the "Init" function in Main Menu 2 (see ¶ 28).
- Writing to the data carrier using the special initialization command 'Z' (see ¶ 39).

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

If the error message is not caused by a failed read request, it may be assumed that one or more memory cells on the data carrier is defective. That data carrier must then be replaced.

If the CRC error is however due to a failed write request, you must reinitialize the data carrier in order to continue using it.

CRC_16

The checksum is written to the data carrier for each CRC block (corresponds to 16 bytes) in 2-byte units of information. 2 bytes are lost for each CRC block, i.e. the CRC block contains only 14 bytes of user information. This means that the actual usable number of bytes is reduced (see ¶ 11 "Supported data carriers and memory capacity").

CRC_16 is factory deactivated.

Saving read data

Data which have been read from a data carrier which are now stored in the Handy Programmer can be saved in an EEPROM under a file name of your choice.

BIS M-81_ Basic knowledge for the application

Communication between Handy Programmer and PC over the RS 232 interface

The Handy Programmer has its own serial RS 232 port through which read/write jobs can be executed and files exchanged between a PC and the Handy Programmer. For this you will need a BIS C-522 cable *).

Communication on the serial interface uses a specific protocol.

Data transmission between the Handy Programmer and a laptop or PC takes place using specific telegrams. The following possibilities are offered:

- Using the Handy Programmer to handle read/write jobs from the PC, with the Handy Programmer assuming the function of a BIS Processor. Details see starting ¶ 29.
- File management and data exchange of saved files between the PC and Handy Programmer. Details see starting ¶ 42.

Supported data carriers and memory capacity

Mifare

Balluff data carrier type	Manufacture	Name	Memory capacity	Usable bytes using CRC	Memory type
BIS M-1_ _01	Philips	Mifare Classic	752 Byte	658 Byte	EEPROM



The ISO15693 data carriers can be processed by the BIS M-810 and BIS M-812 Handy-Programmers having hardware version V2.0 or higher. In addition, when using BIS M-811 the hardware version for the externally connected read/write heads BIS M-3_ _ must be V2.0 or higher!

ISO15693

Balluff data carrier type	Manufacture	Name	Memory capacity	Usable bytes using CRC	Memory type
BIS M-1_ _02	Fujitsu	MB89R118	2000 Byte	1750 Byte	FRAM
BIS M-1_ _03 ¹⁾	Philips	SL2ICS20	112 Byte	98 Byte	EEPROM
BIS M-1_ _04 ¹⁾	Texas Inst.	TAG-IT Plus	256 Byte	224 Byte	EEPROM
BIS M-1_ _05 ¹⁾	Infineon	SRF55V02P	224 Byte	196 Byte	EEPROM
BIS M-1_ _06 ¹⁾	EM	EM4135	288 Byte	252 Byte	EEPROM
BIS M-1_ _07 ¹⁾	Infineon	SRF55V10P	992 Byte	868 Byte	EEPROM

¹⁾ See Accessories on ¶ 55

¹⁾ on request

english

BALLUFF | 11

BIS M-81_ Start-up

The BIS M-810 and BIS M-812 Handy Programmer have an integrated antenna and do not require an external read/write head.

Plugging in the read/write head in the BIS M-811

With the BIS M-811 Handy Programmer a standard BIS M-3_ _ read/write head is operated from the front of the device through a read/write cable. If no BIS M-3_ _ read/write head is connected and the device is turned on, the following warning message appears:

```
!!!!  ERROR  !!!!
-----
      R/W head error
```

Error message if no R/W head is connected

Turning on the Handy Programmer

Turn on the Handy Programmer by pressing the button.

The unit comes on while displaying the unit type and software version. A battery test is performed at the same time.

You can shorten the time the display remains on by pressing the key.

```
BALLUFF BIS M-81_
Handy Programmer
Software Vx.x
```

Start display with software version

BIS M-81_ Start-up

Turning on the Handy Programmer (cont.)

```
!!!!!!!!!!!!!!!!!!!!
Battery low!
!!!!!!!!!!!!!!!!!!!!
```

If the voltage has dropped below a certain nominal capacity, a warning message is issued. The unit can still be operated for a short time in spite of this warning. The battery should however be recharged as soon as possible (see ¶ 53).

Battery warning message

After pressing the **Enter** key the unit indicates its battery state in the following display. Following a passed battery test this display will appear automatically.

```
--- Main menu ---
1 READ      4 File
2 WRITE     5 Config
3 Edit      6 More
```

Main menu 1

BIS M-81_ Configuration

If you do not wish to use the factory setting, start by configuring the Handy Programmer.

Configuration 1
5

Menu item 5 in Main Menu 1 takes you to Configuration Menu 1:

```
-- Configuration --
1 Addresses 4 Serial
2 Format
3 Carrier   6 More
```

Configuration menu 1

Configuring addresses
5 1

The address ranges for reading and writing of data carriers and for the jump-to address for editing can be preset.

The selection number for the respective mode as well as the desired values must be entered and confirmed by pressing the **Enter** key. Values you want to copy can be confirmed directly by pressing **Enter**. In this way you can proceed through the displayed menus one after the other.

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

Select input mode for reading

Input mode READ:

Start address / End address or
Start address / Number of bytes for reading

READ range

```
READ range
-----
at byte no. : 0
to byte no. : 255
```

Input menu 0 for mode
Start/End address for reading

```
READ range
-----
at byte no. : 0
no. of bytes: 0256
```

Input menu 1 for mode Start
address/No. of bytes for reading

BIS M-81_ Configuration

Configuring addresses

[5] [1]

(cont.)

Input mode WRITE

- Start address / End address or
- Start address / No. of bytes for writing

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

Select input mode for writing

WRITE range

```
WRITE range
-----
at byte no. : 0
to byte no. : 255
```

Input menu 0 for mode
Start/End address for writing

```
WRITE range
-----
at byte no. : 0
no. of bytes: 0256
```

Input menu 1 for mode Start
address/No. of bytes for writing

EDIT range

Jump-to address and size for EDIT

Here you must specify:

- the largest address to be edited, and
- the address (cursor position) to be displayed first when editing.

```
Edit range
-----
End address: 0751
Cursor position:0000
```

Input menu for End address
when editing

After the last confirmed entry, the program returns to Main Menu 1.

BIS M-81_ Configuration

Select format

[5] [2]

Four data formats are available for entering data in the Editor:

- hexadecimal (HEX)
- binary coded decimal (BCD)
- ASCII
- BINARY (BIN)

If you choose BCD as the data format, you can also enter a code conversion for the special characters. When BCD is set and data with the numbers A_{hex} bis F_{hex} are stored on the data carrier, these numbers are interpreted as special characters. This function can be used to output BCD values with sign and decimal point. The following special characters are preset:

'A' = * 'B' = + 'C' = , 'D' = - 'E' = . 'F' = /

Select data carrier

[5] [3]

```
- Carrier Data -
DTTyp: 0 All Types
CRC: N
```

In this screen you can set the data carrier type you want to process.

0 = ALL Types: All data carriers supported by Balluff can be processed.

1 = Mifare: All Mifare data carriers supported by Balluff can be processed. (See [1] 11 "Supported data carriers and memory capacity")

2 = ISO15693: All ISO15693 data carriers supported by Balluff can be processed. (See [1] 11 "Supported data carriers and memory capacity")



The ISO15693 data carriers can be processed by the BIS M-810 and BIS M-812 Handy-Programmers having hardware version V2.0 or higher. In addition, when using BIS M-811 the hardware version for the externally connected read/write heads BIS M-3_ must be V2.0 or higher!

This screen is also used to set whether CRC data checking is enabled or not.

- CRC: Check Y (Yes), N (No)

BIS M-81_ Configuration

Configure port

5 4

Configuring the serial RS 232 port for communicating with a PC or laptop:

The following parameters can be set:

- Baud rate: 9600, 19200, 38400 or 57600 Baud
- Stop bits: 1 or 2
- Data bits: 8 (cannot be changed)
- Parity: E (even), O (odd) or N (none)
- BCC / CR: B (BCC), C (CR)

The desired values must be entered and confirmed by pressing **Enter**. Copied values can be confirmed directly by pressing **Enter**.

Configuration 2 Language

5 6 1

Menu item 6 takes you to Configuration Menu 2:

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

Here you can select the menu language.

Menu item 2 returns you to Main Menu 1 following a prompt asking whether you want to save the configuration.

Configuration menu 2

Function Description BIS M-81_ Processing data carriers

Read data carrier

1

Selecting "READ" in Main Menu 1 takes you to the menu for reading the data carrier. You are asked to enter the start address and – depending on which input mode is configured – the end address or the number of bytes:

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

or

```
READ data carrier
-----
at byte no. :
no. of bytes:
```

Entry menu 1 for reading a data carrier

Entry menu 2 for reading a data carrier

The data for the last entry made or for the data specified in the configuration for start address and end address with number of bytes are displayed as the default values. Press **↓** and **↑** to toggle between the two entry fields. Press **Enter** to accept the data.

If you press the **Enter** key to confirm the entered data in the "up to byte no.:" or "number of bytes:" field, the read procedure will be started. As long as there is no data carrier in the read range, a bar will flash in the middle of the 2nd line. As soon as a data carrier is recognized, the flashing will stop and the data will be read from the data carrier.

```
READ data carrier
-----
at byte no. :
no. of bytes:
```

Screen shown while reading a data carrier



The data carrier must not be removed from the active zone of the read/write head while data are being read from the data carrier!



To cancel the read, press the **Enter** key.


Function Description BIS M-81_ Processing data carriers

Read data carrier

 (cont.)

After a successful read, the bar disappears and the data are shown in the display.

Depending on which format you have selected, the data will be displayed as hex values, ASCII characters, BCD numbers or BINARY. Use the  and  keys to toggle between the various formats:

Key	Format	Key	Format
	>ASCII -> BCD -> BIN		HEX

```
Display 0010 - 0049
-----
_ABCDEFGHIJKLMNOPQRST
UVWXYZ123456789abcde
```


Data displayed in ASCII format

```
Display 0100 - 0119
-----
01234567890123456789
01234567890123456789
```

Data displayed in BCD format

```
Display 0150 - 0163
-----
00 01 02 03 04 05 06
07 08 09 0A 0B 0C 0D
```

Data displayed in HEX format

Pressing  returns you to Main Menu 1.

Function Description BIS M-81_ Processing data carriers

Read errors

If an error occurs while reading, an error message is shown in the display.


Error message "**Data carrier removed**"

```
!!!! Error !!!!
-----
READ error!
data carrier removed
```

Error message 1 for reading

Causes:

- Data carrier was removed from the active zone too soon
- Data carrier - read/write head distance too great

After acknowledging the error message by pressing  you will be returned to Main Menu 1.


Error message "**Maximum end address exceeded**"

```
!!!! Error !!!!
-----
Maximum end address
exceeded
```

Error message 2 for reading

Causes:

- End address is greater than the configured capacity of the data carrier

After acknowledging the error message by pressing  you will be returned to the entry screen for reading.

Function Description BIS M-81_ Processing data carriers

Read errors
(cont.)

Error message "?? Distance ???"

```

READ data carrier
?? Distance ??
at byte no. : 0
no. of byte : 0600
    
```

Error message 3 for reading

Causes:

- Data carrier – read/write head distance too great
- End address (= Start address + no. of bytes) is greater than the capacity of the data carrier
- Wrong block size configured for this data carrier
- Data carrier defective

The Handy Programmer repeats the read operation another 2 times. If the 3rd attempt also fails, the operation is cancelled with this error message:

```

!!!! Error !!!!
-----
READ error
Distance / others
    
```

Error message 4 for reading

Function Description BIS M-81_ Processing data carriers

Writing to data carriers

2

If "WRITE" is selected in the Main Menu, the entry screen for writing to the data carrier is displayed. You are prompted for the start address and – depending on the input mode configured – the end address or number of bytes.

```

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

Entry menu 1 for writing to the data carrier

or

```

WRITE to carrier
-----
at byte no. :
no. of bytes:
    
```

Entry menu 2 for writing to the data carrier

The data for the last entry resp. the data for start address and end address or number of bytes are used as default values. Press **[J]** and **[I]** to toggle between the two entry fields. Press **[Enter]** to accept the data.

Pressing the **[Enter]** key to confirm the data entered in the "to byte no.:" or "number of bytes" field starts the write operation. As long as there is no data carrier in the read range, a bar will flash in the middle of the 2nd line. As soon as a data carrier is recognized, the flashing will stop and the data will be read from the data carrier.

```

WRITE to carrier
-----
at byte no. :
no. of bytes:
    
```

Screen shown while writing to the data carrier



The data carrier must not be removed from the active zone of the read/write head while data are being read from the data carrier!

Function Description BIS M-81_ Processing data carriers

Write errors

If an error occurs while writing, an error message is shown in the display.

Error message "Data carrier removed"

```

!!!! Error !!!!
-----
WRITE error!
data carrier removed
    
```

Causes:

- Data carrier was removed from the active zone too soon
- Data carrier – read/write head distance too great

After acknowledging the error message by pressing **Enter** you will be returned to Main Menu 1.

Error message 1 for writing

Error message "Maximum end address exceeded"

```

!!!! Error !!!!
-----
Maximum end address
exceeded
    
```

Causes:

- End address is greater than the configured capacity of the data carrier

After acknowledging the error message by pressing **Enter** you will be returned to the entry screen for writing.

Error message 2 for writing

Function Description BIS M-81_ Processing data carriers

**Write errors
(cont.)**

Error message "?? Distance ???"

```

WRITE to carrier
?? Distance ???
at byte no. : 0
no. of byte : 0600
    
```

Causes:

- Data carrier – read/write head distance too great
- End address (= Start address + no. of bytes) is greater than the capacity of the data carrier
- Wrong block size configured for this data carrier
- Data carrier defective

The Handy Programmer repeats the write operation another 2 times. If the 3rd attempt also fails, the operation is cancelled with this error message:

```

!!!! Error !!!!
-----
WRITE error
Distance / other
    
```

Error message 4 for writing

Function Description BIS M-81_

Processing data carriers

Editing data

3

Menu point 3 "Edit" in Main Menu 1 opens the Editor.
In the next display you enter the start address into the Editor (0-0751).

```

Edit Cursor Position
-----
at byte no. : 0000
    
```

The data are displayed as HEX values, ASCII characters, BCD numbers or BINARY depending on the format selected. The character highlighted by the cursor can be edited.

```

Edit Byte No.: 0049
-----
ABCDEFGHIJKLMNQRST
UVWXYZ123456789abcde
    
```

Editing data in ASCII format

```

Edit Byte No.: 0119
-----
01234567890123456789
01234567890123456789
    
```

Editing data in BCD format

```

Edit Byte No.: 0163
-----
00 01 02 03 04 05 06
07 08 09 0A 0B 0C 0D
    
```

Editing data in HEX format

```

Edit Byte No.: 0003
-----
00110001 00110010
00110011 00110100
    
```

Editing data in BINARY format

Press and to go to the adjacent character. If HEX is set, pressing and advances by a half-byte.

Pressing and jumps one line higher or lower. If the display range is exceeded, the system automatically scrolls up or down by one line.

Press to return to Main Menu 1.

Function Description BIS M-81_

Working with files

File menu

4

The BIS M-81_ Handy Programmer contains a non-volatile EEPROM memory. This EEPROM can be used to store data read from a data carrier under a file name. The file name is limited to 8 characters. Up to 15 files having a maximum of 752 bytes per file can be stored.

```

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

File menu

File format

Data carrier files have the following structure:

Start address of the data on the data carrier 4 bytes ASCII
No. of bytes 4 bytes ASCII
Data carrier data Data-transparent; data amount as specified in number of bytes

Example: Data carrier file starting at Address 75 with 16 bytes of data in ASCII format
Data: ABCDEFGHIJKLMNQP
File contents: 00750016ABCDEFGHIJKLMNQP

Load file

4 1

Menu point 1 "Load" in the file menu can be used to load a file stored in the EEPROM into RAM of the Handy Programmer.

The data in this file can then be further processed by selecting menu item 3 "Edit" in Main Menu 1.

```

Please enter name:
-
    
```

Load file from EEPROM

After loading the data, the display shows the start address and the end address or number of bytes. Pressing the key quits the display.

The data from this file can then be further processed selecting menu item 3 "Edit" in Main Menu 1.

```

Loaded data
-----
at byte No.: 0
Number of bytes: 256
    
```

Function Description BIS M-81_ Working with files

Save file

4 2

Use menu item 2 "Save" to save the data that were last read from a data carrier and are still contained in RAM of the Handy Programmer as a file in the EEPROM memory.

```
Please enter name:
_
```

Save file in EEPROM

Exchanging data over the serial interface

4 3

Using menu item 3 "Trans." (for data transmission) you can transmit files back and forth between a PC or laptop to the Handy Programmer over the serial interface. For additional information see ¶ 42ff.

Delete file(s)

4 4

Menu point 4 "Delete" allows you to delete saved files.
Menu point 1 deletes a single file. An input screen opens where you enter the file name.
Menu point 2 deletes all saved files at one time.

```
--- Delete Menu ---
1 = One entry
2 = All entries
6 <--
```

Delete menu

Show directory

4 5

Menu point 5 "Directory" displays a lists of the saved files
Press **Enter** to exit the display.

```
File1   File5
File2   File6
File3
File4
```

Directory

Function Description BIS M-81_ Special data carrier functions

Main Menu 2

6

Selecting menu point 6 "More" in Main Menu 1 takes you to Main Menu 2.

```
-- Main Menu 2 --
1 Init
2 Serial Number
6 <--
```

Main menu 2

Init menu: Initialize data carrier

1

```
---- Initmenu ----
1 CRC 6 <--
```

Selecting menu point 1 "Init" in Main Menu 2 takes you to the Init menu.

Init menu

CRC

Menu point 1 "CRC" writes 00_{Hex} to the data carrier, which prepares for operation with a CRC data check.

Read Serial Number

2

```
Serial  11 22 33 44
Number  55 66 77 88

DTTyp: BIS M-1..-XX
```

Menu item 2 "Serial Number" is used to read out the 4-byte (Mifare) or 8-byte (ISO15693) long serial number from the data carrier and display it in HEX format. This is read-only. In addition the data carrier type is displayed. XX stands for Balluff data carrier type . ?? is displayed for types which Balluff does not support.

Use **Enter** to return to Main Menu 2.

Function Description BIS M-81_ / PC Reading / Writing over the serial interface

In addition to key-controlled processing of the data carriers, it is also possible to perform read and write operations from a PC or laptop over the serial interface of the Handy Programmer. The Handy Programmer then functions just like a BIS Processor.

The PC and Handy Programmer are connected using a BIS C-522 *) cable. Communication between the Handy Programmer and the laptop / PC uses a specific protocol (see ¶ 30). For details on the individual telegrams see ¶ 31ff.

Activate data transmission

4 3

Menu point 4 "File" is selected in Main Menu 1 and menu point 3 "Trans." activated in the following file menu to start this function.

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

File menu

Data transmission	
6 <--	

Data transmission

Data integrity on the RS 232 interface

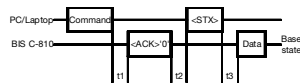
All commands for reading and writing (L, Z, C and P) as well as the Acknowledge command (Q) can, depending on the configuration, be terminated either with a BCC block check or with Carriage Return 'CR'. The Handy Programmer then also terminates all transmitted characters using the selected termination type.

*) See Accessories on ¶ 55

Function Description BIS M-81_ / PC Reading / Writing over the serial interface

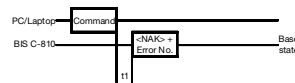
Protocol sequence

Read: a) If no error:



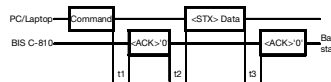
t1 depending on no. of bytes to read (see ¶ 43)
t2 ≥ 0 (is not monitored by the processor)
t3 = max. 50 ms

b) With error:



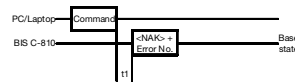
t1 depending on no. of bytes to read (see ¶ 43) and error type (recommended monitor time: 15 s)

Write: a) If no error:



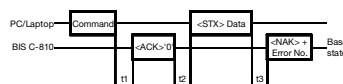
t1 = max. 50 ms
t2 ≥ 0 (is not monitored by the processor)
t3 depending on no. of bytes to write (see ¶ 43)

b) With error in command:



t1 = max. 50 ms

c) With error in writing:



t1 = max. 50 ms t2 ≥ 0 (is not monitored by the laptop adapter)
t3 depending on no. of bytes to write (see ¶ 43) and error type (recommended monitor time: 30 s)

The examples are valid only if:
- The Handy Programmer is in the base state.
- A data carrier is present in front of a read/write head.

Function Description BIS M-81_ / PC Reading / Writing over the serial interface

Telegram types with associated command (ASCII)

There are specific telegrams for the individual operations in the BIS M-81_ Handy Programmer. They always begin with the command which corresponds to the telegram type:

- 'L' Read data carrier
- 'P' Write to data carrier
- 'C' Write a constant value to the data carrier
- 'Q' Restart Handy Programmer (Quit)
- 'Z' Initialize data carrier for CRC_16 data check

Please note:

- Continuous querying on the interface is not permitted!
- The minimum wait time between two commands is 300 ms!

Function Description BIS M-81_ / PC Reading / Writing over the serial interface

Explanation of selected telegram contents

Start address and no. of bytes The start address (A3, A2, A1, A0) and the number of bytes to send (L3, L2, L1, L0) are transmitted as ASCII characters. For the start address a range of 0000 to 1999 and for the number of bytes 0001 to 2000 can be used. A3 ... L0 stand for 1 ASCII character each.

Please note: Start address + number of bytes may not exceed the data carrier capacity.

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

Start <STX> starts data transmission.

Bytes sent The data are transmitted code-transparent (not converted).

Function Description BIS M-81_ / PC Reading / Writing over the serial interface

BCC Block Check

The BCC block check is formed as an EXOR of the serially transmitted binary characters of the telegram block. Example: Read 128 bytes starting at address 13.
The command line without BCC is: 'L 0013 0128 10'

The BCC is formed:

'L'	=	0100 1100	EXOR
0	=	0011 0000	EXOR
0	=	0011 0000	EXOR
1	=	0011 0001	EXOR
3	=	0011 0011	EXOR
0	=	0011 0000	EXOR
1	=	0011 0001	EXOR
2	=	0011 0010	EXOR
8	=	0011 1000	EXOR
1	=	0011 0001	EXOR
0'	=	0011 0000	EXOR

Block check result: BCC = 0100 0100 = 'D'

Variant for terminating with 'CR' instead of BCC

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

- Carriage Return 'CR'

Example: Read 128 bytes starting at address 13.

The command line with 'CR' as a terminator is then: 'L 0013 0128 10 CR'

Telegram Structure and Examples Reading / Writing over the serial interface

Read from data carrier with block size select
Write to data carrier with block size select

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

1) The Quit command is not permitted at this point.
2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

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

Telegram Structure and Examples Reading / Writing over the serial interface

Telegram example for 134:
Read from data carrier with BCC block check

You want to read 10 bytes from the data carrier starting at address 50.

The host sends

```
'L 0 0 5 0 0 0 1 0 1 0' BCC (49 Hex)
```

Address of first byte to read _____
 Number of bytes to read _____
 Fixed _____
 Fixed _____

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

Telegram example for 134:
Write to data carrier with BCC block check

You want to write 5 bytes to the data carrier starting at address 500.

The host sends

```
'P 0 5 0 0 0 0 0 5 1 0 Q' BCC (51 Hex)
```

Address of first byte to write _____
 Number of bytes to write _____
 Fixed _____
 Fixed _____

The Handy Programmer acknowledges with <ACK> '0'
 The host gives the start command and data <STX> '1 2 3 4 5 3' BCC (33 Hex)
 The Handy Programmer acknowledges with <ACK> '0'

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

Telegram Structure and Examples Reading / Writing over the serial interface

Writing a constant value in the data carrier

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

Task	Data Flow	Command	Start address of first byte to be sent	Number of bytes to be sent	End	Acknowledge 2)	Start transmission	Data (from start address to start address + no. of bytes)	End	Acknowledge 2)
Write	from host system to BIS	'C'	A3 A2 A1 A0 '0 0 0 0' to '0 7 5 1'	L3 L2 L1 L0 '0 0 0 1' to '0 7 5 2'	'1' '0'	BCC or 'CR'	<STX>	D	BCC or 'CR'	
	from BIS to host system					<ACK> '0' or <NAK> + Error-No.				<ACK> '0' or <NAK> + Error-No.

1) The Quit command is not permitted at this point.
 2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

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

Telegram Structure and Examples Reading / Writing over the serial interface

Telegram example for 136:
Writing a constant value in the data carrier with BCC block check

Write ASCII data value 0 (30Hex) to 500 bytes of the data carrier starting a Address 20.

The host sends

Address of first byte to write 'C 0 0 2 0 0 5 0 0 1 0 E' BCC (46 Hex)

Number of bytes to write _____

Fixed _____

Fixed _____

The Handy Programmer acknowledges with <ACK> '0'

The host gives the start command and data <STX> '0 2' BCC (32 Hex)

The Handy Programmer acknowledges with <ACK> '0'

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

Telegram Structure and Examples Reading / Writing over the serial interface

Restart the Handy Programmer (Quit)

Sending the Restart command causes a telegram in process to be aborted and puts the Handy Programmer in the ground state.
After this telegram is acknowledged, a minimum of 1600 ms pause should be allowed before starting a new telegram.



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

Task	Data Flow	Command	Terminator	Acknowledge	Terminator
Restart	from host system to BIS	'Q'	BCC or 'CR'		
(Quit)	from BIS to host system			'Q'	BCC or 'CR'
1)					

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

Telegram example with BCC block check:

Put the Handy Programmer into the ground state.

The host sends 'Q Q' BCC (51 Hex)

The Handy Programmer acknowledges with 'Q Q' BCC (51 Hex)

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

Telegram Structure and Examples Reading / Writing over the serial interface

Initialize a data carrier for CRC_16 data checking

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

Please note the table on 10! The indicated number of usable bytes must not be exceeded, i.e., the sum of start address plus number of bytes must not exceed the data carrier memory capacity!

Task	Data Flow	Com- mand	Start address of first byte to be sent	Number of bytes to be sent	End	Acknow- ledge 2)	Start trans- mission	Data (from start address to start address + no. of bytes)	End	Acknow- ledge 2)
Initialize CRC_16 range	from host system to BIS	'Z'	A3 A2 A1 A0 '0 0 0 0' to usable bytes using CRC -1	L3 L2 L1 L0 '0 0 0 1' to usable bytes using CRC	'1' '0'	BCC or 'CR'	<STX>	D1 D2 D3 ... Dn	BCC or 'CR'	
	from BIS to host system					<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) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

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

Reading / Writing over the serial interface Error Numbers

Error numbers

The BIS M-81_ always outputs an error number. The meaning of these error numbers is indicated in the following table.

No.	Error Description	Effect
2	Read error	Read telegram aborted, Handy Programmer goes into ground state.
3	Read aborted, since the data carrier was removed	Handy Programmer goes into ground state.
4	Write error	Write telegram aborted, Handy Programmer goes into ground state.
5	Writing aborted, since the data carrier was removed	Handy Programmer goes into ground state.
6	Interface error	Handy Programmer goes into ground state. (parity or stop bit error)
7	Telegram format error	Handy Programmer goes into ground state. Possible format errors: - Command is not 'L', 'P', 'Q', 'C' or 'Z' Start address or number of bytes exceed permissible range
8	BCC error, the transmitted BCC is wrong	Telegram is aborted, Handy Programmer goes into ground state.
E	CRC error	The CRC for the read data does not agree with the CRC on the data carrier.

CAUTION:
An aborted write could cause new data to be written to the data carrier which may be incomplete!

Reading / Writing over the serial interface Read/Write Times

Read times

Data carrier with each 16 bytes/block	BIS M-1_-01	BIS M-1_-02
Time for data carrier recognition/serial ID	≤ 20 ms	≤ 30 ms
Read bytes 0 to 15	≤ 20 ms	≤ 30 ms
For each additional 16 bytes add another	≤ 10 ms	≤ 15 ms

Write times

Data carrier with each 16 bytes/block	BIS M-1_-01	BIS M-1_-02
Time for data carrier recognition/serial ID	≤ 20 ms	≤ 30 ms
Write bytes 0 to 15	≤ 40 ms	≤ 65 ms
For each additional 16 bytes add another	≤ 30 ms	≤ 45 ms



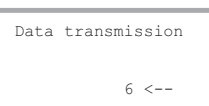
Fluctuations in the ms range are possible.
Electrical noise may increase the read/write time.

Function Description BIS M-81_ / PC Working with files over the serial interface

Data transfer on the serial interface

4 | 3

Menu point 3 "Trans." can be used to transmit data over the serial interface from a PC or laptop to the Handy Programmer and back. It is also possible to administer the file. For this the BISHandy.exe program must be installed on the PC or laptop by running the file setup.exe. Both files are included on the supplied CD-ROM.



Data transmission

Select menu point 4 "File" in the Main Menu 1 and in the following menu select 3 "Trans." to activate the "Data transmission" function.

Data integrity on the serial interface

Each of the commands (D, G, T, K and E) is always terminated with a BCC block check. The Handy Programmer likewise terminates all transmitted characters with the BCC.

Telegram types with associated command (ASCII characters)

Specific telegrams exist for the individual data administration tasks in the Handy Programmer BIS M-81_. They always begin with the command which corresponds to the telegram type:

- 'T' Transmit file to the Handy Programmer
- 'G' Get file from the Handy Programmer
- 'D' Get file directory from the Handy Programmer
- 'E' Delete a single file in the Handy Programmer
- 'K' Delete all files in the Handy Programmer

Please note:
 - Continuous querying on the interface is not permitted!
 - The minimum wait time between two commands is 300 ms!
 - With these telegrams only the BCC block check is used for data integrity (regardless of the configuration).

Working with files over the serial interface Telegram Structure and Examples

Send file from the PC to the Handy Programmer

This command is used to send a file stored in a PC or laptop to the Handy Programmer.

Task	Data Flow	Com- mand	File name 3)	Start address on the data carrier	No. of bytes on the data carrier	End	Acknow- ledge 2)	Start trans- mission	Data (from start address to start address + no. of bytes)	End	Acknow- ledge 2)
Send file to the Handy Programmer	from host system to BIS	'T'	'D7 ... D0' 'MUSTER_...'	A3 A2 A1 A0 '0 0 0 0' to memory capacity -1	L3 L2 L1 L0 '0 0 0 1' to memory capacity	BCC		<STX>	D1 D2 D3 ... Dn	BCC	
	from BIS to host system						<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) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
- 3) If the file name is shorter than 8 characters, it must be filled out to 8 characters using spaces (20_{hex}).

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

Working with files over the serial interface Telegram Structure and Examples

Read file from the Handy Programmer to the PC or laptop

This command is used to read out a file stored in the Handy Programmer.

Task	Data Flow	Com- mand	File name 3)	End	Acknow- ledge 2)	Start trans- mission	Start address on the data carrier	No. of bytes on the data carrier	Data (from start address to start address + no. of bytes)	End	
Read file from the Handy Programmer	from host system to BIS	'G'	'D7 ... D0' 'MUSTER_...'	BCC		<STX>					
	from BIS to host system				<ACK>'0' or <NAK> + Error-No.		A3 A2 A1 A0 '0 0 0 0' to memory capacity -1	L3 L2 L1 L0 '0 0 0 1' to memory capacity	D1 D2 D3 ... Dn	BCC	
				1)						1)	

- 1) The Quit command is not permitted at this point.
- 2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
- 3) If the file name is shorter than 8 characters, it must be filled out to 8 characters using spaces (20_{hex}).

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

Working with files over the serial interface Telegram Structure and Examples

Telegram example for 43:
Send file to the Handy Programmer with BCC block check

Send file MUSTER1 to the Handy Programmer. Write 5 bytes to the data carrier starting with Address 27.

The host sends 'T MUSTER1_ 0 0 2 7 0 0 0 5 M' BCC (4D Hex)

File name (filled out to 8 bytes *)
Address of the first byte on the data carrier
Number of bytes

The Handy Programmer acknowledges with <ACK> '0'
The host gives the start command and data <STX> '12345 3' BCC (33 Hex)
The Handy Programmer acknowledges with <ACK> '0'

Telegram example for 44:
Read file from the Handy Programmer with BCC block check

Read file MUSTER1 from the Handy Programmer. Read 5 bytes on the data carrier starting with Address 27.

The host sends 'G MUSTER1_ ^' BCC (5E Hex)

File name (filled out to 8 bytes *)

The Handy Programmer acknowledges with <ACK> '0'
The host sends <STX>
The Handy Programmer sends '0027000512345 1' BCC (33 Hex)

*) A space _ used for filling out the file name must also be accounted for in the BCC.

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

Working with files over the serial interface Telegram Structure and Examples

Read file directory from Handy Programmer to PC / Laptop

This command is used to read the file directory stored in the Handy Programmer.

Task	Data Flow	Command	End	Acknowledge 2)	Start transmission	Data	End
Read directory from Handy Programmer	from host system to BIS	'D'	BCC		<STX>		
	from BIS 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) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.

Telegram example:
Read file directory from Handy Programmer with BCC block check

Read the file directory from the Handy Programmer. Two files are stored in the Handy Programmer.

The host sends 'D D' BCC (44 Hex)

The Handy Programmer acknowledges with <ACK> '0'
The host sends <STX>
The Handy Programmer sends '0002MUSTER1_00270005 MUSTER2_00350012 EOT' BCC (04 Hex)

*) A space _ used for filling out the file name must also be accounted for in the BCC.

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

Working with files over the serial interface Telegram Structure and Examples

Delete a file in the Handy Programmer

This command is used to delete a file stored in the Handy Programmer.

Task	Data Flow	Command	Dateiname 3)	End	Acknowledge 2)
Delete a file in the Handy Programmer	from host system to BIS	'E'	'D7 ... D0' 'MUSTER_ _'	BCC	
	from BIS to host system				<ACK>'0' or <NAK> + Error-No.
1)					

- 1) The Quit command is not permitted at this point.
- 2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
- 3) If the file name is shorter than 8 characters, it must be filled out to 8 characters using spaces (20_{hex}).

Telegram example:
Delete file in Handy Programmer
with BCC block check

Delete file MUSTER1 from the directory in the Handy Programmer.
The host sends 'E MUSTER1_ ' BCC (5C Hex)
File name (filled out to 8 bytes) *)
The Handy Programmer acknowledges with <ACK> '0'

*) A space ' ', used for filling out the file name must also be accounted for in the BCC.

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

Working with files over the serial interface Telegram Structure and Examples

Delete all files in the Handy Programmer

This command is used to delete all files in the Handy Programmer.

Task	Data Flow	Command	End	Acknowledge 2)
Delete all files in Handy Programmer	from host system to BIS	'K'	BCC	
	from BIS to host system			<ACK>'0' or <NAK> + Error-No.
1)				

- 1) The Quit command is not permitted at this point.
- 2) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
- 3) If the file name is shorter than 8 characters, it must be filled out with spaces (20_{hex}).

Telegram example:
Delete all files in the Handy Programmer
with BCC block check

Delete all files in the Handy Programmer
The host sends 'K K' BCC (4B Hex)
The Handy Programmer acknowledges with <ACK> '0'

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

Working with files over the serial interface Error Numbers

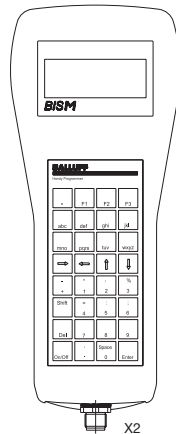
Error numbers

The BIS M-81_ always outputs an error number. The meaning of these error numbers is indicated in the following table.

Error No.	Error description	Cause
0 = 30 Hex	No error	Only in conjunction with <ACK>
Q = 51 Hex	Timeout	Character delay time of 1 s exceeded. Interface problem
R = 52 Hex	Entry error	Specified file not found
S = 53 Hex T = 54 Hex	Memory error	End of Handy Programmer memory reached
V = 56 Hex	Interface error	Incorrect port settings (baud rate, parity, data bit or stop bit error)
W = 57 Hex	Protocol format error	Protocol format not maintained
X = 58 Hex	BCC is incorrect	Incorrect BCC returned

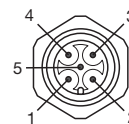
BIS M-81_ Interface Information

BIS M-810
Integrated antenna



Serial interface RS 232
Read/Write head connector

X2, Serial interface RS 232

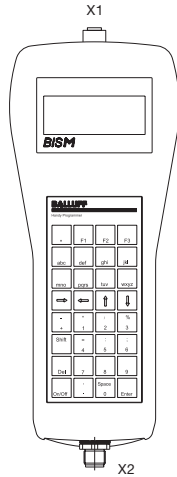


Pin	Function
1	n.c.
2	TxD
3	GND
4	RxD
5	n.c.

n.c. = do not connect

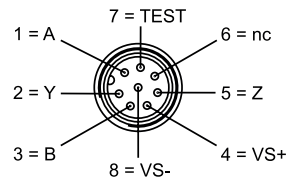
BIS M-811 Interface Information

BIS M-811
Integrated plug for
connecting standard
BIS M-3_ _
read/write heads
(Hardware version
V2.0 and higher)

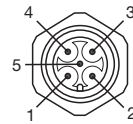


**Serial interface
RS 232
Read/Write head
connector**

**X1, Connection for external read/write
head for BIS M-811**



X2, Serial interface RS 232

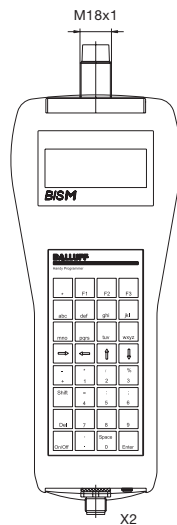


Pin	Function
1	n.c.
2	TxD
3	GND
4	RxD
5	n.c.

n.c. = do not connect

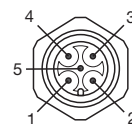
BIS M-812 Interface Information

BIS M-812
Integrated antenna
with M18 housing



**Serial interface
RS 232
Read/Write head
connector**

X2, Serial interface RS 232



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

n.c. = do not connect

BIS M-81_ Battery charging

Battery charging

There are two ways to charge the battery in the Handy Programmer:

Without removing: Insert the 2.5 mm jack on the BIS C-701/A *) charger into the charging socket X2 on the bottom of the Handy Programmer.

Removed: For this the battery is removed (see below) and inserted into the BIS C-702-A*) charging holder.

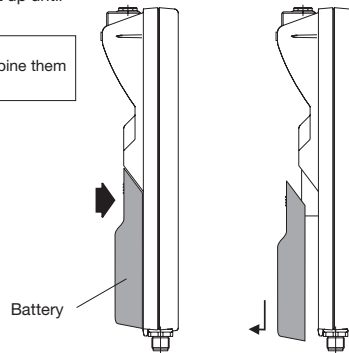
Replace battery

To remove the battery from the Handy Programmer, press the latch in and slide the battery down slightly until the guides release the battery and it can be removed.

After charging, reinsert the battery and slide it up until it latches.



Please properly dispose of used batteries. **Never** throw them into bodies of water or combine them with normal refuse.



*) See Accessories on 157

Removing the battery

BIS M-81_ Technical Data

Housing	Housing material	ABS
	Dimensions	approx. 97 × 232 × 55 mm
	Weight incl. battery	approx. 420 g
	Keypad	32 keys, 4 × 8 mm, alphanumeric and special characters
Connections	Display	LCD-Display, 80 characters / 4 lines
	RS 232 serial port	5-pin round connector
	Read/Write head	BIS M-810 internal antenna BIS M-811 8-pin round connector BIS M-812 internal antenna in M18 housing
	Charging socket	2.5 mm jack designed for BIS C-701-A
Enclosure rating	Enclosure rating	IP 40 (when connected)
Temperature range	Ambient temperature	0 °C to +50 °C

BIS M-81_ Technical Data

Electrical data	Power supply	2.4 V NiMH rechargeable battery
	Capacity	1650 mAh
	max. charge voltage	5 – 10 V, internally limited
	Charging current	approx. 400 mA
	Charging time	approx. 4 h
Current draw	Current draw, normal operation	approx. 65 mA
	Current draw, reading/writing	approx. 275 mA

CE With the CE Mark we affirm that our products are in accordance with the requirements of the EU (European Union) Guideline 89/336/EEC (EMC Guideline) and the EMC Law. It has been verified in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, that Balluff products meet the EMC requirements of the Harmonized Standard EN 61000-6-4 (Emission), EN 61000-6-2 (Noise Immunity)

BIS M-81_ Ordering Information

Ordering code	BIS M-81_-0-003
Balluff Identification System	_____
Series M	_____
Hardware type	_____
810 = internal antenna with keypad and display, for connecting to PC or laptop with 9-pin SUB-D connector (serial)	_____
811 = 8-pin connector for connecting standard read/write heads BIS M-3_ _ (Hardware version V2.0 and higher)over cable, with keypad . . .	_____
812 = internal antenna in M18 housing with keypad and display, for connecting to PC or laptop with 9-pin SUB_D plug (serial)	_____
Interface	_____
0 = RS 232 (V.24 Interface)	_____
Software type	_____
003 = Balluff software for BIS M-81_-...	_____

BIS M-81_ Accessories

Accessories
(included)

The hand-held readers include a carrying case.

Accessories
(optional,
not included)

Article		Ordering code
Charger (see ¶ 58)		BIS C-701-A
Charge holder (see ¶ 59)		BIS C-702-A
Interface cable for connecting the BIS M-81_ Handy Programmer to a 9-pin SUB-D COM port on a laptop or PC	2 m 3 m	BIS C-522-PVC-02 BIS C-522-PVC-03
Additional battery (see ¶ 60)		125997
For BIS M-811		
Round connector for read/write heads BIS M-3_ _ (without cable)		BKS-S117-00
Connector cable for read/write heads BIS M-3_ _ (max. 2 m *)		BKS-S115-PU-02
Read heads BIS M-3_ _ (see catalog) Hardware version V2.0 and higher		

*) One molded-on connector
Other end open (for BKS-S117-00)

BIS M-81_ Accessories

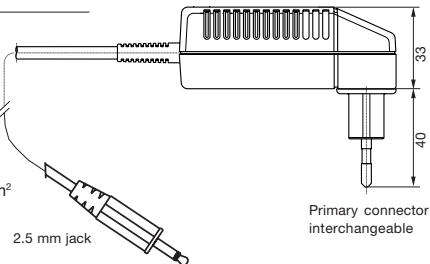
Accessories
(optional,
not included)

Charger BIS C-701-A
Consisting of a charger with 2.5 mm jack

Technical Data

Feature	Value / Units / Condition
$U_{Primary}$	100 - 240 V _{AC}
$F_{Primary}$	47 - 63 Hz
$U_{Sec.}$	6 V _{DC} ±5%
$I_{Sec.}$	2.1 A
T_A	-40 ... +70 °C
Primary connector interchangeable	EURO, UK, USA (included)
Approvals	CE, UL

Material: PPE-V1-125°C
Color: black
Housing: MPP10
Output cable: 11.7892.503-80
Length: 2000 mm, 3 x 0.5 mm²



BIS M-81_ Accessories

Accessories
(optional,
not included)

Technical Data

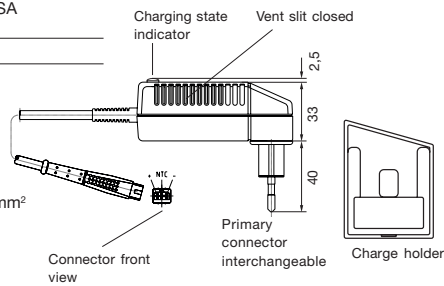
Charge holder BIS C-702-A

Consisting of a charger with charging status indicator and base

Feature	Value / Units / Condition
$U_{Primary}$	100 - 240 V _{AC}
$F_{Primary}$	50 - 60 Hz
$U_{Sec.}$	10 V _{DC} ±5%
$I_{Sec.}$	800 mA
T_A	-40 ... +70 °C
Primary connector interchangeable	EURO, UK, USA (included)
Approvals	CE, UL

Charging state indicator:
LED red Battery charged
LED red flashing Charging

Material: PPE-V1-125°C
Color: black
Housing: MPP10
Output cable: 11.7892.503-80
Length: 2000 mm, 3 x 0.5 mm²



BIS M-81_ Accessories

Accessories
(optional,
not included)

Additional battery

Power supply pack 2.4 V NiMH battery
Capacity 1650 mAh

Appendix, ASCII Table

Deci- mal	Hex	Control Code	ASCII	Deci- mal	Hex	Control Code	ASCII	Deci- mal	Hex	ASCII	Deci- mal	Hex	ASCII	Deci- mal	Hex	ASCII	Deci- mal	Hex	ASCII
0	00	Ctrl @	NUL	22	16	Ctrl V	SYN	44	2C	,	65	41	A	86	56	V	107	6B	k
1	01	Ctrl A	SOH	23	17	Ctrl W	ETB	45	2D	-	66	42	B	87	57	W	108	6C	l
2	02	Ctrl B	STX	24	18	Ctrl X	CAN	46	2E	.	67	43	C	88	58	X	109	6D	m
3	03	Ctrl C	ETX	25	19	Ctrl Y	EM	47	2F	/	68	44	D	89	59	Y	110	6E	n
4	04	Ctrl D	EOT	26	1A	Ctrl Z	SUB	48	30	0	69	45	E	90	5A	Z	111	6F	o
5	05	Ctrl E	ENQ	27	1B	Ctrl [ESC	49	31	1	70	46	F	91	5B	[112	70	p
6	06	Ctrl F	ACK	28	1C	Ctrl \	FS	50	32	2	71	47	G	92	5C	\	113	71	q
7	07	Ctrl G	BEL	29	1D	Ctrl]	GS	51	33	3	72	48	H	93	5D]	114	72	r
8	08	Ctrl H	BS	30	1E	Ctrl ^	RS	52	34	4	73	49	I	94	5E	^	115	73	s
9	09	Ctrl I	HT	31	1F	Ctrl _	US	53	35	5	74	4A	J	95	5F	_	116	74	t
10	0A	Ctrl J	LF	32	20		SP	54	36	6	75	4B	K	96	60	`	117	75	u
11	0B	Ctrl K	VT	33	21		!	55	37	7	76	4C	L	97	61	a	118	76	v
12	0C	Ctrl L	FF	34	22		*	56	38	8	77	4D	M	98	62	b	119	77	w
13	0D	Ctrl M	CR	35	23		#	57	39	9	78	4E	N	99	63	c	120	78	x
14	0E	Ctrl N	SO	36	24		\$	58	3A	:	79	4F	O	100	64	d	121	79	y
15	0F	Ctrl O	SI	37	25		%	59	3B	;	80	50	P	101	65	e	122	7A	z
16	10	Ctrl P	DLE	38	26		&	60	3C	<	81	51	Q	102	66	f	123	7B	{
17	11	Ctrl Q	DC1	39	27		'	61	3D	=	82	52	R	103	67	g	124	7C	
18	12	Ctrl R	DC2	40	28		(62	3E	>	83	53	S	104	68	h	125	7D	}
19	13	Ctrl S	DC3	41	29)	63	3F	?	84	54	T	105	69	i	126	7E	~
20	14	Ctrl T	DC4	42	2A		*	64	40	@	85	55	U	106	6A	j	127	7F	DEL
21	15	Ctrl U	NAK	43	2B		+												