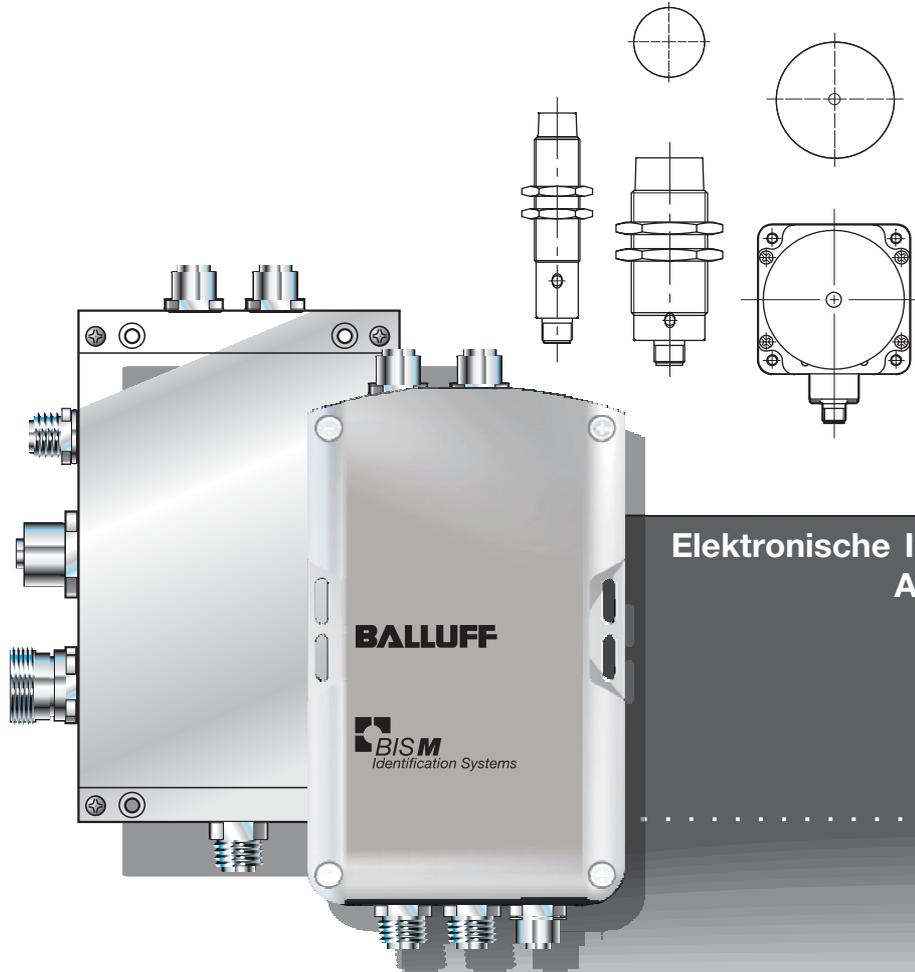


BALLUFF

sensors worldwide



Handbuch

Elektronische Identifikations-Systeme BISM
Auswerteeinheit BIS M-60_3
DeviceNet

English – please turn over!

Nr. 833 667 D/E • Ausgabe 1401
Änderungen vorbehalten.
Ersetzt Ausgabe 0703.

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 | 5-7 | |
| Auswerteeinheit BIS M-60_3, Basiswissen für die Anwendung | 8/9 | |
| BUS-Anbindung DeviceNet | 10-12 | |
| Funktionsbeschreibung: Parametrierung der Auswerteeinheit BIS M-60_3 | 13-27 | |
| Betriebsarten (Mode 1, Mode 2) | 28 | |
| Kommunikation mit der Auswerteeinheit | 29/30 | |
| Ein- und Ausgangspuffer | 31-34 | |
| Mode 1 | | |
| Ausgangspuffer, Belegung und Erklärung | 35-38 | |
| Eingangspuffer, Belegung und Erklärung | 39-41 | |
| Datenträger-Typen | 42 | |
| Datenträger bearbeiten | 43-49 | |
| Beispiele für den Protokollablauf | 50-62 | |
| Mode 2 | | |
| Ausgangspuffer, Belegung und Erklärung | 63/64 | |
| Eingangspuffer, Belegung und Erklärung | 65-67 | |
| Datenträger bearbeiten | 68-70 | |
| Beispiele für den Protokollablauf | 71-74 | |
| Schreib-/Lesezeiten | 75 | |
| Funktionsanzeigen | 76 | |
| | BIS M-6003 BIS M-6023 | |
| Montage Kopf / Auswerteeinheit | 77 | 85 |
| Öffnen der Auswerteeinheit | 78 | 86 |
| Schnittstelleninformationen / Anschlusspläne | 79 | 87 |
| Wechseln des EEPROM | 80 | 88 |
| Technische Daten | 81/82 | 89/90 |
| Bestellinformationen: Typschlüssel, Zubehör | 83/84 | 91/92 |
| Anhang, ASCII-Tabelle | | 93 |

Sicherheitshinweise

Bestimmungsgemäßer Betrieb

Auswerteeinheiten BIS M-60_3 bilden zusammen mit den anderen Bausteinen des Systems BIS M das Identifikations-System und dürfen 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 der Auswerteeinheit sind die Kapitel mit den Anschlussplänen genau zu beachten. Besondere Sorgfalt erfordert der Anschluss der Auswerteeinheit an externe Steuerungen, speziell bezüglich Auswahl und Polung der Verbindungen und der Stromversorgung.

Für die Stromversorgung der Auswerteeinheit dürfen nur zugelassene Stromversorgungen benutzt werden. Einzelheiten enthält das Kapitel Technische Daten.

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 Auswerteeinheiten der Baureihe BIS M-6003-025-050-03-ST12 und BIS M-6023-025-050-03-ST13.

RS NetWorx ist ein eingetragenes Warenzeichen der Firma Rockwell Automation.
Windows ist ein eingetragenes Warenzeichen der Microsoft Corporation.

Einführung

Identifikations-System BIS M

Dieses Handbuch soll den Anwender beim Einrichten des Steuerprogramms und der Installation und Inbetriebnahme der Komponenten des Identifikations-Systems BIS M anleiten, so dass sich ein sofortiger, reibungsloser Betrieb anschließt.

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.



Sind 2 Schreib-/Leseköpfe an die Auswerteeinheit BIS M-60_3 angeschlossen, können beide Schreib-/Leseköpfe unabhängig voneinander bearbeitet werden. D.h., am einen Schreib-/Lesekopf kann ein Datenträger gelesen werden, während am anderen Schreib-/Lesekopf auf einen anderen Datenträger geschrieben wird.

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,
- **im Lagerbereich zur Kontrolle der Lagerbewegungen;**
- **im Transportwesen und in der Fördertechnik.**

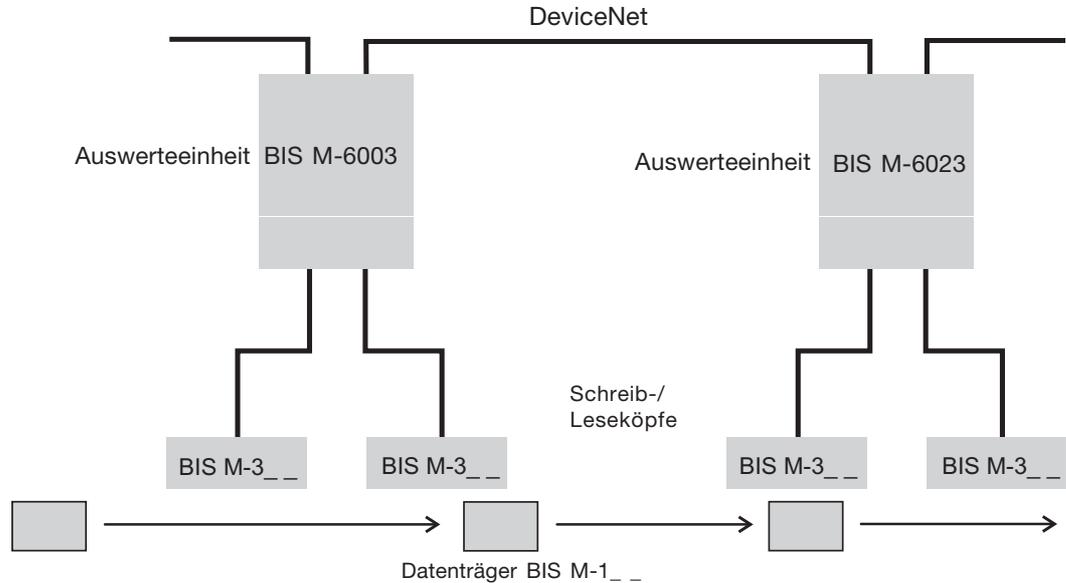
Einführung Identifikations-System BIS M

System- komponenten

Die Hauptbestandteile des Identifikationssystems BIS M sind:

- **Auswerteeinheit,**
- **Schreib-/Leseköpfe und**
- **Datenträger**

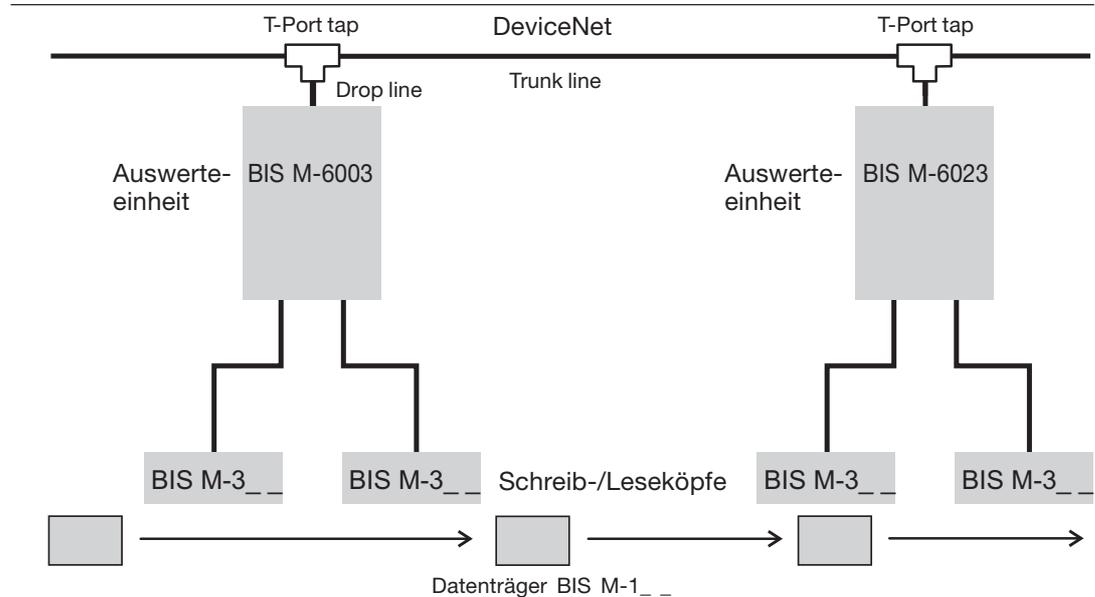
**Anordnung mit
Auswerteeinheit
BIS M-6003 und
BIS M-6023**
(Verbindung mit
Trunk lines)



*Schematische
Darstellung eines
Identifikations-Systems
(Beispiel)*

Einführung Identifikations-System BIS M

**Anordnung mit
Auswerteeinheit
BIS M-6003 und
BIS M-6023**
(Verbindung mit T-
Port taps und Drop
lines ¹⁾)



*Schematische
Darstellung eines
Identifikations-Systems
(Beispiel)*

¹⁾ Bei Anschluss der Auswerteeinheit BIS M 60_3 per Drop line, sollte der DeviceNet Ausgang mit einer Verschlusskappe (siehe Zubehör □ 84/92) verschlossen werden.

Auswerteeinheit BIS M-60_3

Basiswissen für die Anwendung

Auswahl der Systemkomponenten

Die Auswerteeinheit **BIS M-6003** besitzt ein Kunststoffgehäuse.

Die Auswerteeinheit **BIS M-6023** besitzt ein Metallgehäuse.

Der Anschluss erfolgt über Rundsteckverbinder. Es können zwei Schreib-/Leseköpfe über Kabel angeschlossen werden.

Die Auswerteeinheiten BIS M-60_3 verfügen zusätzlich über einen digitalen Eingang. Der Eingang hat je nach Konfiguration unterschiedliche Funktionen (siehe Parametrierung).

Die Schreib-/Leseabstände richten sich nach der Wahl des Datenträgers. In den jeweiligen Handbüchern zu den Schreib-/Leseköpfen der Baureihe BIS M-3_ _ finden Sie sämtliche Kombinationen von Schreib-/Lesekopf und passenden Datenträgern.

Die Systemkomponenten werden von der Auswerteeinheit 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.

Auswerteeinheit BIS M-60_3

Basiswissen für die Anwendung

Steuerfunktion

Über den Schreib-/Lesekopf schreibt die Auswerteeinheit Daten vom steuernden System auf den Datenträger oder liest sie vom Datenträger und stellt sie dem steuernden System zur Verfügung. Steuernde Systeme können sein:

- ein Steuerrechner (z.B. Industrie-PC) oder
 - eine speicherprogrammierbare Steuerung (SPS)
-

Datensicherheit mit CRC_16

Für Applikationen, die hohe Sicherheit gegen falsche Daten erfordern, kann das CRC_16 Verfahren eingesetzt werden. Hier wird ein Prüfcode auf den Datenträger geschrieben, der jederzeit und überall das Kontrollieren der Daten auf Gültigkeit erlaubt.

Vorteile mit CRC_16: Sehr hohe Datensicherheit, auch während der nicht aktiven Phase (Datenträger außerhalb des S/L-Kopfes)

Nachteile mit CRC_16: Längere Schreib-/ Lesezeiten, es gehen Nutzbyte auf dem Datenträger verloren.

Die Verwendung des CRC_16 kann vom Anwender parametrisiert werden. (siehe □ 20)

BUS-Anbindung DeviceNet

DeviceNet

Die Kommunikation zwischen der Auswerteeinheit BIS M-60_3 und dem steuernden System erfolgt über das DeviceNet.

Das System DeviceNet besteht aus den Komponenten:

- die **Anschaltbaugruppe (DeviceNet Master) als Komponente einer Steuerung und**
- **den Busknoten/Nodes (hier die Auswerteeinheit BIS M-60_3)**

Es können maximal 63 Knoten an eine Anschaltbaugruppe angehängt werden.

Die Anschaltbaugruppe wandelt die Daten um: vom steuernden System in das serielle Protokoll des DeviceNet-Netzwerks und umgekehrt. Sie besitzt dazu die DeviceNet-Schnittstelle zur Verbindung mit den dezentralen DeviceNet-Knoten. Informationen zum Einrichten der Anschaltbaugruppe entnehmen Sie bitte dem zugehörigen Handbuch.

Verbindungsaufbau

Der Aufbau einer Verbindung zwischen der Anschaltbaugruppe und der Auswerteeinheit BIS M-60_3 erfolgt mit Hilfe des *predifined master/slave connection set*. Die DeviceNet-Funktionalität des BIS M-60_3 entspricht einem *group 2 only server*.

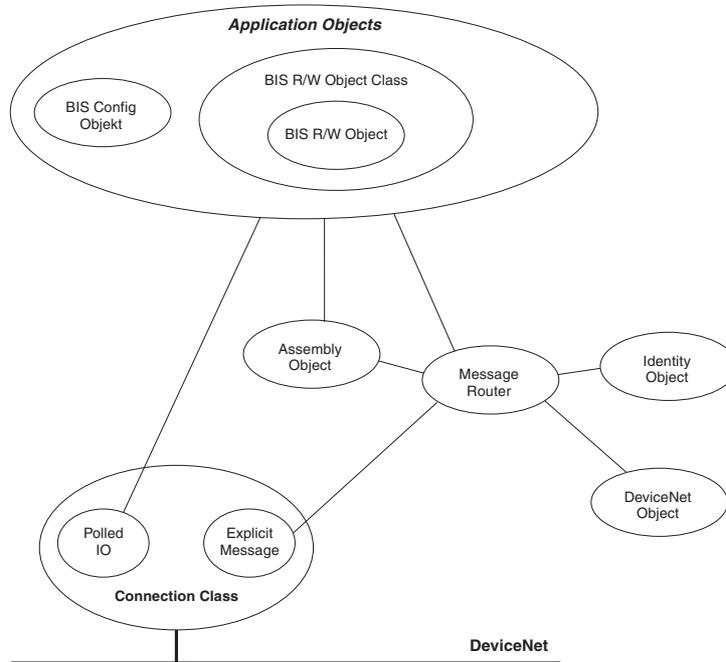
Objektmodell

DeviceNet beschreibt alle Parameter und Funktionen eines Gerätes anhand eines Objektmodells. Der Zugriff vom Netzwerk auf die einzelnen Objekte erfolgt ausschließlich über Verbindungsobjekte (*connection objects*). Das Objektmodell der Auswerteeinheit BIS M-60_3 ist auf □ 11 dargestellt.

BUS-Anbindung DeviceNet

Objektemodell (Fortsetzung)

Im Bild ist das Objektemodell der Auswertereinheit BIS M-60_3 dargestellt. Hierbei spiegelt das „BIS Config Objekt“ die Konfigurationseigenschaften des Gerätes wieder, das „BIS S/L Objekt“ die zwei Schreib-/Leseköpfe.

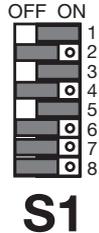


BUS-Anbindung DeviceNet

MAC-ID einstellen

Die MAC-ID der Auswerteeinheit BIS M-60_3 kann mit dem Schiebeschalter S1.1 - 1.6 eingestellt werden. Der Schiebeschalter S1.1 - 1.6 ist binär codiert. Die Einstellung der Stationsadresse erfolgt nach dem in der Tabelle gezeigten Schema:

| MAC-ID | Schiebeschalter S1 | | | | | |
|--------|--------------------|-------|-------|-------|-------|-------|
| | 6 | 5 | 4 | 3 | 2 | 1 |
| | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| 0 | OFF | OFF | OFF | OFF | OFF | OFF |
| 1 | OFF | OFF | OFF | OFF | OFF | ON |
| 2 | OFF | OFF | OFF | OFF | ON | OFF |
| 3 | OFF | OFF | OFF | OFF | ON | ON |
| 4 | OFF | OFF | OFF | ON | OFF | OFF |
| ... | | | | | | |
| 42 | ON | OFF | ON | OFF | ON | OFF |
| ... | | | | | | |
| 63 | ON | ON | ON | ON | ON | ON |



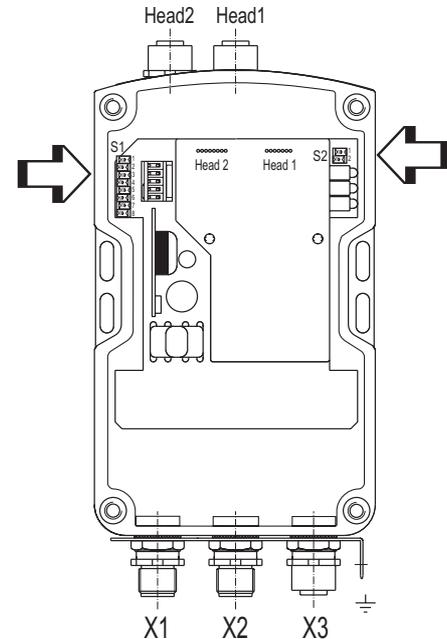
Schiebeschalter S1
(bei geöffnetem Deckel)

ACHTUNG: Die Position S1.7 und S1.8 muss immer auf OFF eingestellt sein.

Baudrate einstellen

Die Baudrate der Auswerteeinheit BIS M-60_3 kann mit dem Schiebeschalter S2 eingestellt werden. Folgende Einstellungen sind möglich:

| Baudrate | Schiebeschalter S2 | |
|-----------|--------------------|-------|
| | 2 | 1 |
| | 2^1 | 2^0 |
| 125 kBaud | OFF | OFF |
| 250 kBaud | OFF | ON |
| 500 kBaud | ON | OFF |
| ungültig | ON | ON |



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

Die Parameter zum Betrieb der Auswerteeinheit BIS M-60_3 sind im BIS Config Objekt (class 64Hex) und im BIS S/L Objekt (class 65Hex) abgelegt. Der Zugriff auf die Parameter erfolgt über *explicit messages*.

Parametrierung mit RS NetWorx

Ein verbreitetes Tool zur DeviceNet-Geräteparametrierung ist die Windows-Software RS NetWorx der Firma Rockwell Automation. Die Geräteparametrierung wird daher beispielhaft an Hand dieser Software erklärt. Informationen zur Software RS NetWorx und zur Installation entnehmen Sie bitte dem Handbuch von RS NetWorx. Der Zugriffspfad (*class* → *instance* → *attribute*) auf jeden Parameter wird ebenfalls angegeben, so dass auch der direkte Zugriff auf jeden Parameter per *explicit message* möglich ist.

Die Parametrierung der Auswerteeinheit BIS M-60_3 erfolgt innerhalb der Projektierung des gesamten DeviceNet-BUS-Systems. Dieses besteht in der Regel aus einem DeviceNet-Master und mehreren DeviceNet-Nodes. Informationen zur Erstellung eines Projekts entnehmen Sie bitte dem Handbuch der Software RS NetWorx.

EDS-file installieren

Jeder Lieferung der Auswerteeinheit BIS M-60_3 liegt eine Diskette mit der EDS-Datei (EDS = *electronic data sheet*) bei. Die EDS-Datei enthält sämtliche Geräteparameter der Auswerteeinheit BIS M-60_3 und wird in der EDS-Library von RS NetWorx installiert.

Starten Sie hierzu RS NetWorx. Wählen Sie unter Menü Tools → EDS Wizard und folgen Sie den Installationsanweisungen.

Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

Parametrierung online / offline

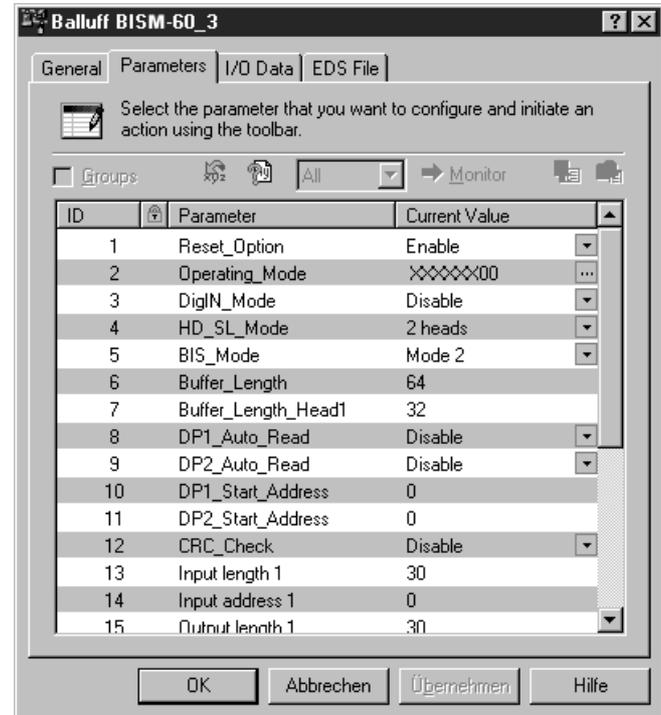
Die Parametrierung des Systems kann online oder offline vorgenommen werden. Beim Offline-Projektieren werden die gewünschten Geräte aus der Hardwareliste per Drag&Drop in das Netzwerk-Fenster übernommen und dann konfiguriert. Beim Online-Projektieren wird über Menü *Network* → *Online* das Netzwerk durchsucht und die gefundenen Geräte dargestellt. Die Parametrierung erfolgt in beiden Fällen durch Doppelklick auf das zu parametrierende Gerät, hier BIS M-60_3. Dadurch erscheint das Fenster zur Einstellung des Geräts. Durch Klicken auf Parameter erscheint das Fenster für die Geräteparameter.

Werkseinstellungen

Die unter *Current Value* gezeigten Einstellungen und Werte geben bei Beginn der Parametrierung die Werkseinstellung wieder.

Wenn Sie einen Parameter eingestellt haben, wird die Taste [Übernehmen] aktiv, und Sie können die Einstellung damit übernehmen.

Die Taste [OK] beendet die Parametrierung.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

Parameter 1 Reset-Option

Reset_Option class: 64Hex
instance: 01Hex
attribute: 64Hex

Werkseinstellung: *Enable* (= 1)
In dieser Einstellung kann die Auswerteeinheit von der Steuerung mit einem High-Signal auf dem digitalen Eingang rückgesetzt werden.

andere Einstellungen: *Disable* (= 0)
Rücksetzen durch die Steuerung nicht möglich.

Parameter 2 Dynamikbetrieb Kopf 1 / Kopf 2

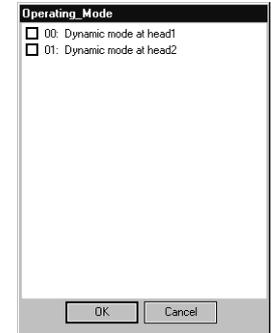
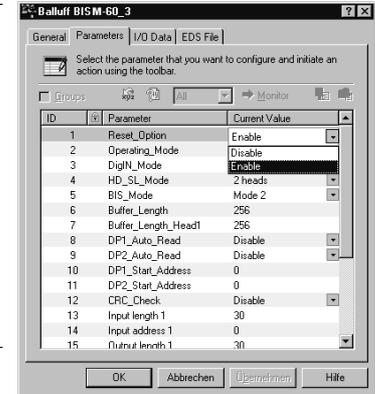
Operating_Mode class: 64Hex
instance: 01Hex
attribute: 65Hex

Werkseinstellung: *no dynamic mode* (= 0)
In dieser Einstellung kann für jeden Kopf einzeln Dynamikbetrieb aktiviert/deaktiviert werden.

andere Einstellungen: *dynamic head1* (= 1)
dynamic head2 (= 2)
dynamic head1 und 2 (= 3)

Ist die Einstellung *dynamic mode* für einen Kopf nicht aktiviert, wird ein Lesebefehl der Steuerung nur bearbeitet, wenn sich ein Datenträger vor dem Schreib-/Lesekopf befindet. Andernfalls wird der Befehl mit dem Fehlercode 1 abgelehnt.

Ist die Einstellung *dynamic mode* für einen Kopf aktiviert, wird der Schreib-/Lesebefehl zwischengespeichert und erst dann ausgeführt, wenn sich ein Datenträger vor dem Schreib-/Lesekopf befindet (siehe □ 45).



Funktionsbeschreibung

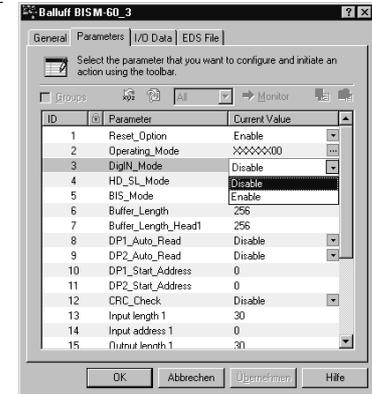
Parametrierung der Auswerteeinheit BIS M-60_3

Parameter 3 Zustand des Digitalen Eingangs

DigIN_Mode class: 64Hex
instance: 01Hex
attribute: 66Hex

Werkseinstellung: *Disable* (= 0)
In dieser Einstellung wird der Zustand des digitalen Eingangs nicht in der Bitleiste des Eingangspuffers angezeigt.

andere Einstellungen: *Enable* (= 1)
Der Zustand des digitalen Eingangs wird in der Bitleiste des Eingangspuffers angezeigt.

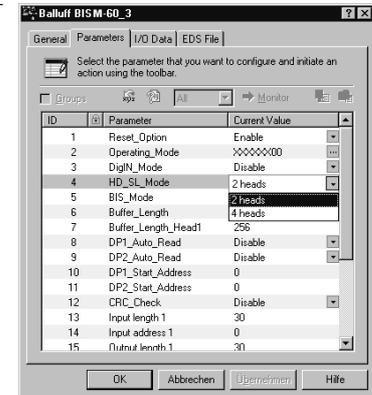


Parameter 4 Anwahl Schreib-/ Lesekopf

HD_SL_Mode class: 64Hex
instance: 01Hex
attribute: 67Hex

Werkseinstellung: *2 heads* (= 0)
Wählen Sie diese Einstellung, wenn die Auswerteeinheit BIS M-60_3 mit 2 Köpfen betrieben wird.

andere Einstellungen: *reserved* (= 1)
Diese Einstellung ist reserviert für spätere Anwendungen.



Funktionsbeschreibung

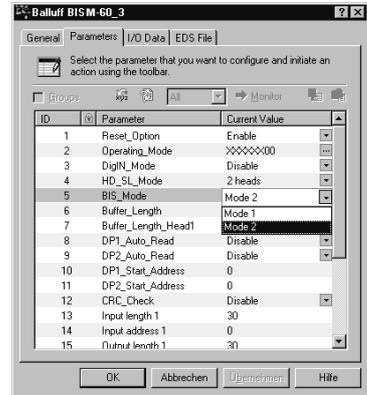
Parametrierung der Auswerteeinheit BIS M-60_3

Parameter 5 Betriebsart BIS

BIS_Mode class: 64Hex
instance: 01Hex
attribute: 68Hex

Werkseinstellung: *Mode 1* (= 0)
Standard-Mode: Gerät arbeitet mit dem Balluff-BUS-Protokoll. Lese- und Schreibaufträge werden von der Steuerung über die I/O-polling-Daten koordiniert. Nähere Informationen zum Mode 1 finden Sie ab □ 28 und ab □ 35.

andere Einstellungen: *Mode 2* (= 1)
Automatik-Mode: Lese- und Schreibaufträge werden automatisch mit den in der Parametrierung festgelegten Einstellungen ausgeführt. Ein Auftrag wird lediglich durch Setzen von Steuerbits in der Bitleiste der I/O-polling-Daten durch die Steuerung angestoßen. Nähere Informationen zum Mode 2 finden Sie ab □ 28 und ab □ 64.



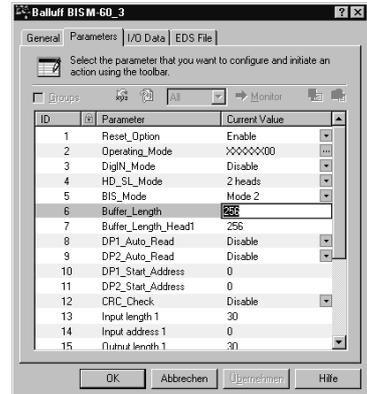
Parameter 6 Gesamte Puffergröße

Buffer_Length class: 64Hex
instance: 01Hex
attribute: 69Hex

Werkseinstellung: 64 *)
andere Einstellungen: 6...256

In dieser Einstellung wird die Größe des Eingangspuffers und des Ausgangspuffers festgelegt. Die gesamte Puffergröße wird beim I/O-polling zwischen der Steuerung und der Auswerteeinheit BIS M-60_3 übertragen.

Die Einstellung ist in BIS-Mode 1 und in BIS-Mode 2 gültig.



*) Bitte beachten Sie bei der Einstellung dieses Parameters besonders die Erklärung der Ein-/Ausgangspuffer ab □ 31.

Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

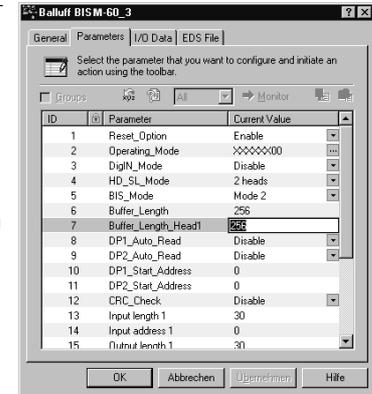
Parameter 7 Puffergröße für Kopf 1

Buffer_Length_Head1 class: 64Hex
instance: 01Hex
attribute: 6AHex

Werkseinstellung: 32 *)
andere Einstellungen: 0...256 (*maximal Buffer_Length*)

Mit diesem Parameter wird der Teil des Gesamtpuffers festgelegt, der für die Bitleiste und zu lesenden/schreibenden Daten von Kopf 1 genutzt werden soll. Der restliche Teil des Gesamtpuffers wird für die Bitleiste und die zu lesenden/schreibenden Daten von Kopf 2 genutzt. Der Parameter *Buffer_Length_Head1* ist nur im BIS-Mode 1 gültig.

*) Bitte beachten Sie bei der Einstellung dieses Parameters besonders die Erklärung der Ein-/Ausgangspuffer ab 31.



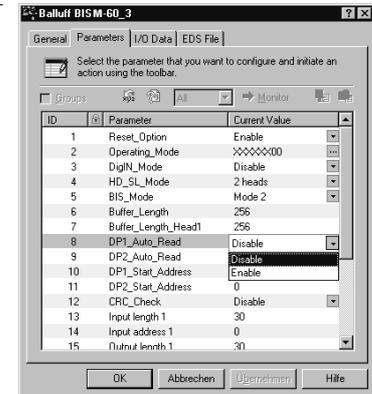
Parameter 8 Autolesen bei CT Present Kopf 1

DP1_Auto_Read class: 64Hex
instance: 01Hex
attribute: 6BHex

Werkseinstellung: *Disable* (= 0)
CT Present-Daten, wenn der Datenträger in den Schreib-/Lesebereich von Kopf 1 kommt.

andere Einstellungen: *Enable* (= 1)
Ist die Funktion Autolesen für Kopf 1 aktiviert, werden Daten ab einer einstellbaren Anfangsadresse gelesen, sobald ein Datenträger vor Kopf 1 erkannt wird.

Näheres zur Funktion Autolesen siehe 31.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

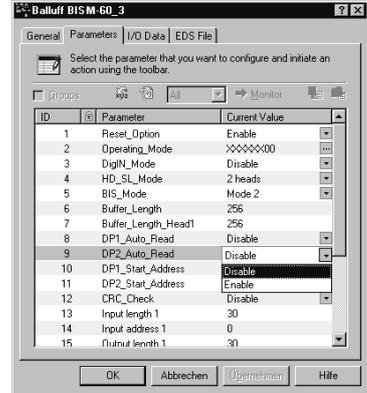
Parameter 9 Autolesen bei CT Present Kopf 2

DP2_Auto_Read class: 64Hex
instance: 01Hex
attribute: 6CHex

Werkseinstellung: *Disable* (= 0)
CT Present-Daten, wenn der Datenträger in den Schreib-/
Lesebereich von Kopf 2 kommt.

andere Einstellungen: *Enable* (= 1)
Ist die Funktion Autolesen für Kopf 2 aktiviert, werden Daten
ab einer einstellbaren Anfangsadresse gelesen, sobald ein
Datenträger vor Kopf 2 erkannt wird.

Näheres zur Funktion Autolesen siehe □ 44.



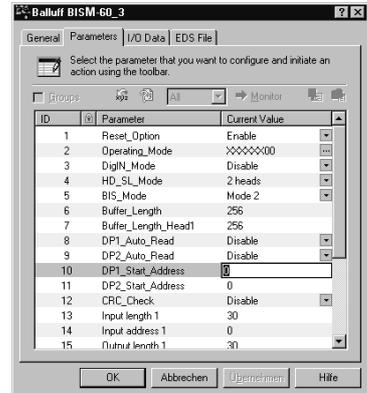
Parameter 10 Anfangsadresse für Autolesen Kopf 1

DP1_Start_Address class: 64Hex
instance: 01Hex
attribute: 6DHex

Werkseinstellung: 0
andere Einstellungen: 1...1999

Mit diesem Parameter wird die Startadresse eingestellt, ab
der ein Datenträger gelesen wird, wenn ein Datenträger vor
Kopf 1 erkannt wird.

Näheres zur Funktion Autolesen siehe □ 44.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

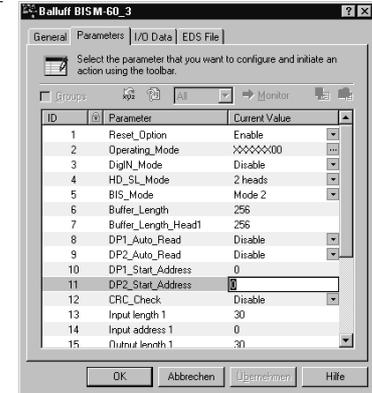
Parameter 11 Anfangsadresse für Autolesen Kopf 2

DP2_Start_Address class: 64Hex
instance: 01Hex
attribute: 6EHex

Werkseinstellung: 0
andere Einstellungen: 1...1999

Mit diesem Parameter wird die Startadresse eingestellt, ab der ein Datenträger gelesen wird, wenn ein Datenträger vor Kopf 2 erkannt wird.

Näheres zur Funktion Autolesen siehe □ 44.

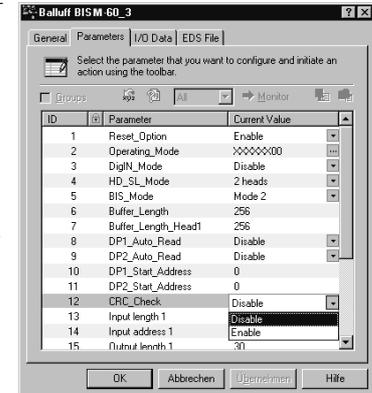


Parameter 12 CRC

CRC_Check class: 64Hex
instance: 01Hex
attribute: 6FHex

Werkseinstellung: *Disable* (= 0)
Kein *CRC_Check*.

andere Einstellungen: *Enable* (= 1)
Gültigkeit der gelesenen Daten wird durch Berechnung des *CRC_16* über die gelesenen Daten und Vergleich mit dem auf dem Datenträger gespeicherten *CRC_16* sichergestellt.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

Parameter 13
Mode 2:
Anzahl Byte
Lesen Kopf 1

Input length 1 class: 65Hex
instance: 01Hex
attribute: 65Hex

Werkseinstellung: 31 Byte
andere Einstellungen: 2 Byte ...
(*Buffer_Length – Input length 2 – 2*) Byte

Mit dieser Einstellung wird festgelegt, wie viele Byte vom Datenträger im BIS-Mode 2 am Kopf 1 gelesen werden sollen.



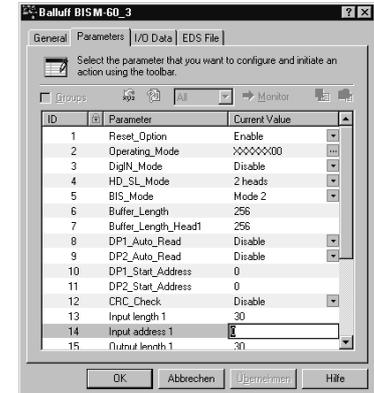
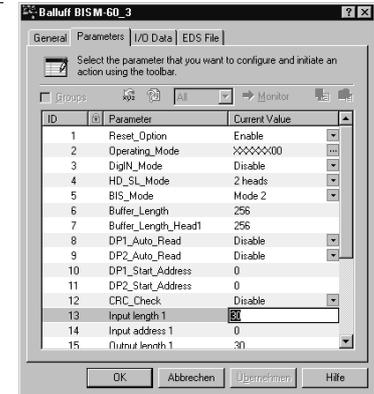
Bitte beachten Sie bei der Einstellung dieses Parameters besonders die Erklärung der Ein-/Ausgangspuffer ab 31.

Parameter 14
Mode 2:
Anfangsadresse
Lesen Kopf 1

Input address 1 class: 65Hex
instance: 01Hex
attribute: 66Hex

Werkseinstellung: *Anfangsadresse 0*
andere Einstellungen: 1 ... 1999

Mit dieser Einstellung wird festgelegt, ab welcher Anfangsadresse der Datenträger im BIS-Mode 2 am Kopf 1 gelesen werden soll.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

Parameter 15
Mode 2:
Anzahl Byte
Schreiben Kopf 1

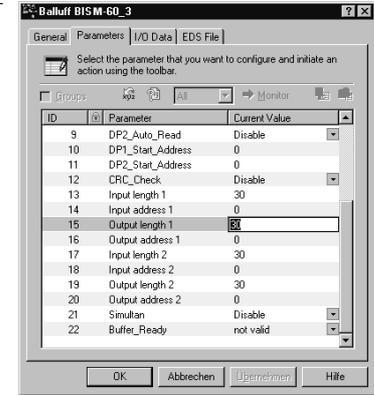
Output length 1 class: 65Hex
instance: 01Hex
attribute: 68Hex

Werkseinstellung: 31 Byte
andere Einstellungen: 2 Byte ...
(*Buffer_Length* – *Output length 2* – 2) Byte

Mit dieser Einstellung wird festgelegt, wie viele Byte vom Datenträger im BIS-Mode 2 am Kopf 1 geschrieben werden sollen.



Bitte beachten Sie bei der Einstellung dieses Parameters besonders die Erklärung der Ein-/Ausgangspuffer ab 31.

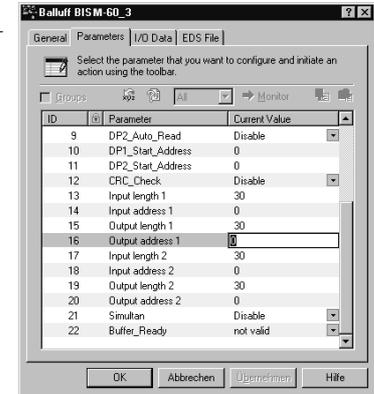


Parameter 16
Mode 2:
Anfangsadresse
Schreiben Kopf 1

Output address 1 class: 65Hex
instance: 01Hex
attribute: 69Hex

Werkseinstellung: Anfangsadresse 0
andere Einstellungen: 1...1999

Mit dieser Einstellung wird festgelegt, ab welcher Anfangsadresse der Datenträger im BIS-Mode 2 am Kopf 1 beschrieben werden soll.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

Parameter 17
Mode 2:
Anzahl Byte
Lesen Kopf 2

Input length 2 class: 65Hex
instance: 01Hex
attribute: 6BHex

Werkseinstellung: 31 Byte
andere Einstellungen: 2 Byte ...
(*Buffer_Length* – *Input length 1* – 2) Byte

Mit dieser Einstellung wird festgelegt, wie viele Byte vom Datenträger im BIS-Mode 2 am Kopf 2 gelesen werden sollen.



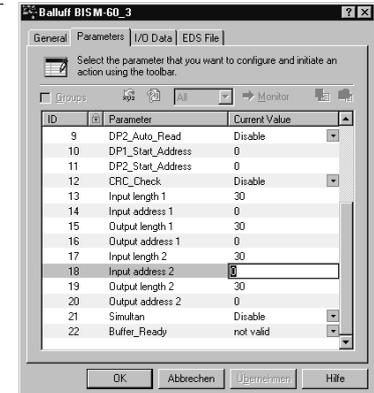
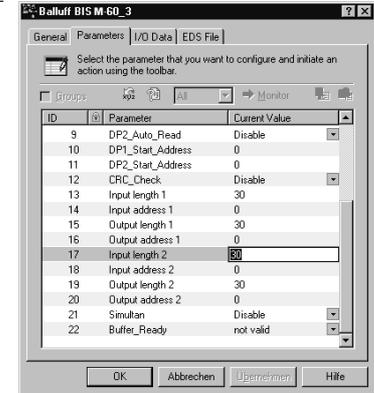
Bitte beachten Sie bei der Einstellung dieses Parameters besonders die Erklärung der Ein-/Ausgangspuffer ab 31.

Parameter 18
Mode 2:
Anfangsadresse
Lesen Kopf 2

Input address 2 class: 65Hex
instance: 01Hex
attribute: 6CHex

Werkseinstellung: Anfangsadresse 0
andere Einstellungen: 1 ... 1999

Mit dieser Einstellung wird festgelegt, ab welcher Anfangsadresse der Datenträger im BIS-Mode 2 am Kopf 2 gelesen werden soll.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

Parameter 19
Mode 2:
Anzahl Byte
Schreiben Kopf 2

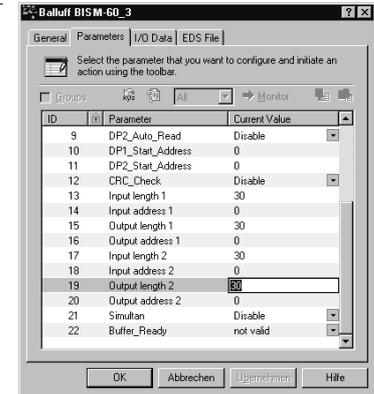
Output length 2 class: 65Hex
instance: 01Hex
attribute: 6EHex

Werkseinstellung: 31 Byte
andere Einstellungen: 2 Byte ...
(*Buffer_Length* – *Output length 1* – 2) Byte

Mit dieser Einstellung wird festgelegt, wie viele Byte vom Datenträger im BIS-Mode 2 am Kopf 2 geschrieben werden sollen.



Bitte beachten Sie bei der Einstellung dieses Parameters besonders die Erklärung der Ein-/Ausgangspuffer ab 31.

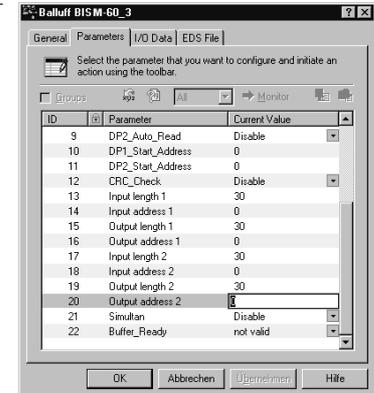


Parameter 20
Mode 2:
Anfangsadresse
Schreiben Kopf 2

Output address 2 class: 65Hex
instance: 01Hex
attribute: 6FHex

Werkseinstellung: Anfangsadresse 0
andere Einstellungen: 1...1999

Mit dieser Einstellung wird festgelegt, ab welcher Anfangsadresse der Datenträger im BIS-Mode 2 am Kopf 2 beschrieben werden soll.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

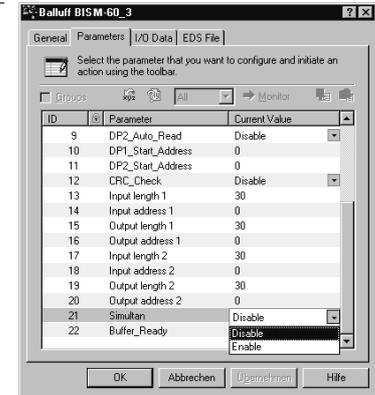
Parameter 21 Simultane Daten- Übertragung

Simultan class: 64Hex
instance: 01Hex
attribute: 70Hex

Werkseinstellung: *Disable* (= 0)
Schreib-/Leseaufträge und Datenübertragung auf dem DeviceNet erfolgen nacheinander.

andere Einstellungen: *Enable* (= 1)
Schreib-/Leseaufträge und Datenübertragung auf dem DeviceNet erfolgen simultan zueinander.

Siehe Beschreibung der Funktion Simultane Datenübertragung auf [S. 45](#).



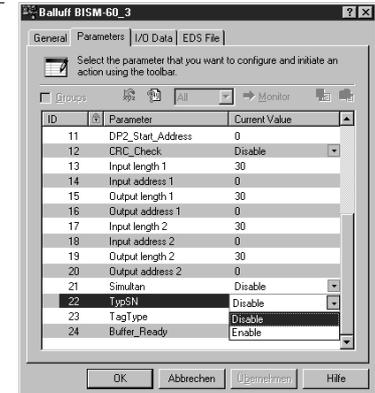
Parameter 22 Typ und Seriennummer ausgeben

TypSN class: 64Hex
instance: 01Hex
attribute: 71Hex

Werkseinstellung: *Disable* (= 0)
Bei CT present werden die ersten Lesedaten des Datenträgers auf dem DeviceNet übertragen.

andere Einstellungen: *Enable* (= 1)
Bei CT present wird der Datenträgertyp und die Seriennummer des Datenträgers auf dem DeviceNet übertragen.

Siehe Beschreibung der Funktion Typ und Seriennummer ausgeben auf [S. 42](#).



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

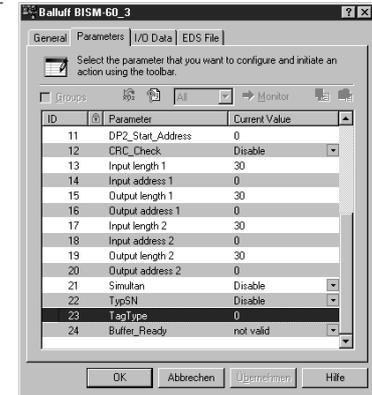
Parameter 23 Datenträgertyp

TagType class: 64Hex
instance: 01Hex
attribute: 72Hex

Werkseinstellung: *Alle Datenträgertypen* (= 0)
In dieser Einstellung werden alle Datenträgertypen von der Auswerteeinheit BIS M-60_3 erkannt und bearbeitet.

andere Einstellungen: *Mifare* (= FEHex)
Alle von Balluff unterstützten Mifare Datenträger.

ISO15693 (= FFHex)
Alle von Balluff unterstützten Datenträger der ISO15693.



Funktionsbeschreibung

Parametrierung der Auswerteeinheit BIS M-60_3

Parameter 24 Puffer-Einstellung übernehmen

Buffer_Ready class: 64Hex
instance: 01Hex
attribute: 73Hex

Werkseinstellung: *Not ready* (= 0)
Mit diesem Parameter wird der Auswerteeinheit BIS M-60_3 signalisiert, dass die Einstellungen für die Puffergrößen jetzt übernommen werden sollen.

andere Einstellungen: *Ready* (= 1)



Bitte beachten Sie, dass die Einstellungen der Parameter *Buffer_Length*, *Buffer_Length_Head1*, *Input length 1*, *Output length 1*, *Input length 2*, *Output length 2* von der Auswerteeinheit BIS M-60_3 erst übernommen werden, nachdem der Parameter *Buffer_Ready* mit *Ready* (=1) gesendet wurde.

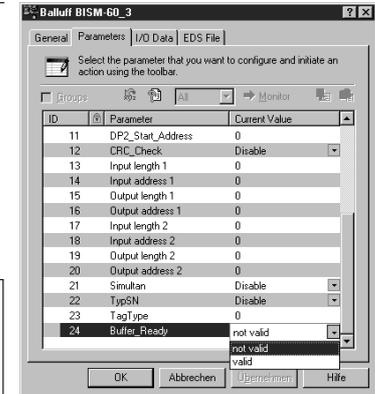
Erklärungen zur Puffergröße siehe ab [] 31.



Wird der Parameter *Buffer_Ready* mit *Ready* (=1) gesendet, dann überprüft die Auswerteeinheit BIS M-60_3, ob die Einstellung gültig ist. Ist dies der Fall, werden die neuen Einstellungen übernommen. Andernfalls wird die Einstellung des Parameters mit „Invalid attribute value“ abgewehrt.



Nach der Prüfung der Puffereinstellungen wird der Wert von *Buffer_Ready* von der Auswerteeinheit BIS M-60_3 automatisch auf *Not ready* (= 0) zurückgesetzt.



Funktionsbeschreibung

Betriebsarten (Mode 1, Mode 2)

Auswahl der Betriebsart

Die Auswahl der BIS-Betriebsart wird über den Parameter 5 *BIS_Mode* vorgenommen. Folgende Betriebsarten stehen zur Verfügung:

Mode 1

Im Mode 1 läuft das Starten eines Schreib-/Leseauftrags und der Datenaustausch nach dem standardisierten Balluff-Protokoll für BUS-Systeme ab. Ein Schreib-/Leseauftrag muss mit Befehlskennung, Anfangsadresse, ab der gelesen/geschrieben werden soll, und Anzahl zu lesenden/schreibenden Byte gestartet werden. Die Steuerung muss auch den Datenaustausch mit der Auswerteeinheit koordinieren, d.h. auf dem I/O-Pufferbereich des jeweiligen Kopfes die Gültigkeit von Lesedaten überwachen und von Schreibdaten anzeigen.

Die BIS-Betriebsart Mode 1 ist dann geeignet,

- wenn große Datenmengen auf dem Datenträger gelesen oder beschrieben werden sollen, oder
- wenn ständig wechselnde Bereiche auf dem Datenträger gelesen oder beschrieben werden sollen.

Mode 2

Mode 2 ist ein vereinfachtes Protokoll zur Umsetzung von Lese-/Schreibaufträgen. Das Starten eines Auftrages wird der Auswerteeinheit lediglich durch Setzen eines Bit in der Bitleiste signalisiert. Bei Schreibaufträgen werden sofort die Schreibdaten mit übergeben. Der Auftrag wird von der Auswerteeinheit automatisch mit den vorher parametrisierten Werten für Startadresse und Anzahl Byte durchgeführt. Die Steuerung muss nach Bearbeitung des Auftrags durch die Auswerteeinheit nur noch die Quittierung prüfen und bei Leseaufträgen die Lesedaten übernehmen.

Die BIS-Betriebsart Mode 2 ist dann geeignet,

- wenn kleinere Datenmengen auf dem Datenträger gelesen oder beschrieben werden sollen, **und**
- wenn immer der gleiche Bereich auf dem Datenträger gelesen oder beschrieben werden soll.

Funktionsbeschreibung

Kommunikation mit der Auswerteeinheit

Prinzipieller Ablauf

Die Kommunikation zwischen dem steuernden System und der Auswerteeinheit erfolgt in einem festen Protokollablauf. Die Gültigkeit von Daten von der Steuerung an die Auswerteeinheit oder umgekehrt von der Auswerteeinheit an die Steuerung wird durch Steuer-Bit angezeigt. Mit Hilfe dieser Bit wird eine Handshake zwischen Steuerung und Auswerteeinheit realisiert.

Hieraus ergeben sich folgende, vereinfacht dargestellte Abläufe eines Auftrags der Steuerung an die Auswerteeinheit für Mode 1 und Mode 2:

Prinzipieller Ablauf für Mode 1

1. Die Steuerung sendet an die Auswerteeinheit eine Befehlskennung zusammen mit den zugehörigen Befehlparametern und setzt ein Bit (AV-Bit). Dieses Bit signalisiert der Auswerteeinheit, dass die übergebenen Daten gültig sind und der Auftrag jetzt beginnt.
2. Die Auswerteeinheit übernimmt den Auftrag und setzt ein Bit (AA-Bit), das dies der Steuerung signalisiert.
3. Ist für die Durchführung des Auftrags ein weiterer Datenaustausch zwischen Steuerung und Auswerteeinheit notwendig, so benutzen diese jeweils ein Bit (TI-Bit und TO-Bit), mit dem signalisiert wird, dass die Steuerung / Auswerteeinheit jetzt für den weiteren Datenaustausch bereit ist bzw. erhaltenen Daten übernommen hat.
4. Hat die Auswerteeinheit den Auftrag korrekt ausgeführt, setzt sie ein Bit (AE-Bit).
5. Hat die Steuerung alle wichtigen Daten übernommen, signalisiert sie dies der Auswerteeinheit durch Rücksetzen des am Beginn gesetzten Bit (AV-Bit).
6. Die Auswerteeinheit setzt nun ebenfalls alle während des Ablaufs gesetzten Steuerbit zurück (AA-Bit, AE-Bit) und ist bereit für den nächsten Auftrag.

Mode 1: Bitte beachten Sie auch die
☐☐ 35...49 und die
Beispiele auf den
☐☐ 50...62.

Funktionsbeschreibung

Kommunikation mit der Auswerteeinheit

Prinzipieller Ablauf für Mode 2

1. Die Steuerung sendet an die Auswerteeinheit die Bitleiste mit dem RW-Bit und dem AV-Bit. Das RW-Bit signalisiert der Auswerteeinheit, ob ein Lese- oder ein Schreibauftrag ausgeführt werden soll. Das AV-Bit signalisiert der Auswerteeinheit, dass ein neuer Auftrag vorliegt. Wird ein Schreibauftrag ausgeführt, so schickt die Steuerung sofort die Schreibdaten mit an die Auswerteeinheit.
2. Die Auswerteeinheit übernimmt den Auftrag und setzt das AA-Bit. Sie übernimmt die Werte für die Anfangsadresse und die Anzahl der zu lesenden/schreibenden Byte aus der Geräte-Parametrierung. Hat die Auswerteeinheit den Auftrag korrekt ausgeführt, setzt sie das AE-Bit. Wurde ein Leseauftrag durchgeführt, übergibt sie die Lesedaten an die Steuerung.
3. Die Steuerung übernimmt bei einem Leseauftrag die Lesedaten und setzt das AV-Bit auf 0.
4. Die Auswerteeinheit setzt das AA-Bit und das AE-Bit auf 0 – sie ist nun für den nächsten Auftrag bereit.

Mode 2: Bitte beachten Sie auch die
☐ 63...70 und die
Beispiele auf den
☐ 71...74.

Funktionsbeschreibung

Ein- und Ausgangspuffer

Ausgangs- und Eingangspuffer

Zur Übertragung von Befehlen und Daten zwischen der Auswerteeinheit BIS M-60_3 und dem steuernden System muss dieses zwei Felder bereitstellen. Die beiden Felder sind:

- **der Ausgangspuffer**
für die Steuerbefehle, die **zu der** Auswerteeinheit BIS M-60_3 geschickt werden und für die zu schreibenden Daten.
- **der Eingangspuffer**
für die zu lesenden Daten und für die Kennungen und Fehlercodes, die **von der** Auswerteeinheit BIS M-60_3 kommen.

Der Austausch dieser Datenbereiche zwischen Steuerung und Auswerteeinheit BIS M-60_3 erfolgt durch zyklisches Polling.



Die Polling-I/O-Verbindung muss von der Steuerung erfolgreich eingerichtet worden sein.

Die möglichen Einstellwerte sind in der EDS-Datei hinterlegt.

Die Puffergröße kann zwischen 6 und 256 Byte gewählt werden. Die parametrisierte Gesamtpuffergröße gilt für beide BIS-Mode (Mode 1 und Mode 2).

Die Aufteilung des Gesamtpuffers auf die beiden Schreib-/Leseköpfe hängt vom ausgewählten BIS-Mode ab.



Bitte beachten Sie beim Aufteilen des Gesamtpuffers auf die 2 Köpfe besonders die Erklärungen auf  32...34.

Funktionsbeschreibung

Ein- und Ausgangspuffer

Aufteilung der Puffer im Mode 1

Im **Mode 1** wird der gesamte Eingangs- und Ausgangspuffer in 2 Bereiche aufgeteilt. Die Bereiche des Eingangspuffer und des Ausgangspuffers je Kopf sind gleich groß (siehe □ 33).

1. Pufferbereich Kopf 1: In diesem Bereich stehen Befehlskennungen und Daten für den Schreib-/Lesekopf 1. Die Größe für den Pufferbereich Kopf 1 wird durch den Parameter 11 *Buffer_Length_Head1* festgelegt.
2. Pufferbereich Kopf 2: In diesem Bereich stehen Befehlskennungen und Daten für den Schreib-/Lesekopf 2. Die Größe für den Pufferbereich Kopf 2 ergibt sich aus dem verbleibenden Bereich Pufferbereich Kopf 2 = (Gesamtpuffer – Pufferbereich Kopf 1).

Aufteilung der Puffer im Mode 2

Im **Mode 2** wird der gesamte Eingangspuffer in 2 Bereiche aufgeteilt und der gesamte Ausgangspuffer in 2 Bereiche aufgeteilt. Die Bereiche des Eingangspuffer und des Ausgangspuffers je Kopf sind unterschiedlich groß (siehe □ 33).

- 1.1 Eingangspuffer-Bereich Kopf 1: In diesem Bereich stehen Befehlskennungen und Lese-daten vom Schreib-/Lesekopf 1. Die Größe für den Eingangspuffer-Bereich Kopf 1 wird durch den Parameter 13 *Input length 1* (Lesedaten) + 1 Byte (Bitleiste) festgelegt.
- 1.2 Eingangspuffer-Bereich Kopf 2: In diesem Bereich stehen Befehlskennungen und Lese-daten vom Schreib-/Lesekopf 2. Die Größe für den Eingangspuffer-Bereich Kopf 1 wird durch den Parameter 17 *Input length 2* (Lesedaten) + 1 Byte (Bitleiste) festgelegt.
- 2.1 Ausgangspuffer-Bereich Kopf 1: In diesem Bereich stehen Befehlskennungen und Schreibdaten vom Schreib-/Lesekopf 1. Die Größe für den Ausgangspuffer-Bereich Kopf 1 wird durch den Parameter 15 *Output length 1* (Schreibdaten) + 1 Byte (Bitleiste) festgelegt.
- 2.2 Ausgangspuffer-Bereich Kopf 2: In diesem Bereich stehen Befehlskennungen und Schreibdaten vom Schreib-/Lesekopf 2. Die Größe für den Ausgangspuffer-Bereich Kopf 2 wird durch den Parameter 19 *Output length 2* (Schreibdaten) + 1 Byte (Bitleiste) festgelegt.

Funktionsbeschreibung

Ein- und Ausgangspuffer

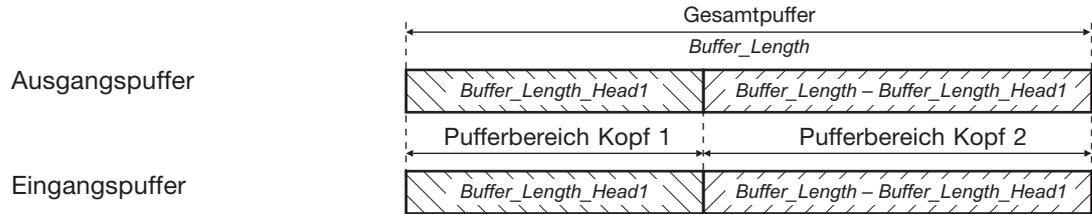
Mode 1 und Mode 2 Gegenüberstellung

Beim Einstellen der Puffergrößen müssen folgende Regeln eingehalten werden:

Es müssen immer alle Regeln für die Parametrierung der Puffer eingehalten werden, auch wenn der betreffende Mode nicht benutzt wird!

Mode 1

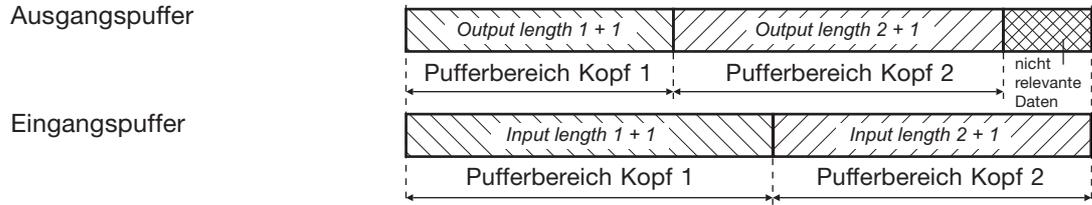
Die Puffergröße für Kopf 1 darf maximal dem Gesamtpuffer entsprechen:
 $\rightarrow Buffer_Length_Head1 \leq Buffer_Length$



Mode 2

Die Ausgangspuffer für Kopf 1 und Kopf 2 dürfen maximal dem Gesamtpuffer – 2 entsprechen:
 $\rightarrow Output\ length\ 1 + Output\ length\ 2 \leq Buffer_Length - 2$

Die Eingangspuffer für Kopf 1 und Kopf 2 dürfen maximal dem Gesamtpuffer – 2 entsprechen:
 $\rightarrow Input\ length\ 1 + Input\ length\ 2 \leq Buffer_Length - 2$



Byte 0

Funktionsbeschreibung

Ein- und Ausgangspuffer

Assembly object Statusabfrage

Um den momentanen Status der beiden Köpfe der Auswerteeinheit abzufragen, kann auf das *assembly object* (class 0x04, instance 0x01, attribute 0x03) zugegriffen werden. Der Zugriff auf das *assembly object* durch die Steuerung erfolgt durch *explicit message*.



Die *explicit-message*-Verbindung muss von der Steuerung erfolgreich eingerichtet worden sein.

Der Aufruf *GetAttributeSingle* liefert 4 Byte:

| Byte 3 | Byte 2 | Byte 1 | Byte 0 |
|--------------------------|--|--------------------------|-----------------------------|
| Kopf 2 | | Kopf 1 | |
| Eingangspuffer Byte 1 | Eingangspuffer Bitleiste (00 _{Hex}) | Eingangspuffer Byte 1 | Eingangspuffer Bitleiste |



Nähere Einzelheiten zum Eingangspuffer finden Sie auf [S. 39ff.](#)

Funktionsbeschreibung

Mode 1: Ausgangspuffer, Belegung und Erklärung

Belegung des Ausgangspuffers für einen Schreib-/Lesekopf (Darstellung gilt für Kopf 1)

| Subadresse \ Bit-Nr. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bitname |
|-------------------------------|---|----|----|---|---|------|---|----|---------|
| 00 _{Hex} = Bitleiste | | TI | KA | | | GR | | AV | |
| 01 _{Hex} | Befehlskennung | | | | | oder | | | Daten |
| 02 _{Hex} | Anfangsadresse (Low Byte) oder Programm-Nr. | | | | | oder | | | Daten |
| 03 _{Hex} | Anfangsadresse (High Byte) | | | | | oder | | | Daten |
| 04 _{Hex} | Anzahl Byte (Low Byte) | | | | | oder | | | Daten |
| 05 _{Hex} | Anzahl Byte (High Byte) | | | | | oder | | | Daten |
| 06 _{Hex} | Daten | | | | | | | | |
| ... | Daten | | | | | | | | |

Erklärungen zum Ausgangspuffer

| Sub-adresse | Bit-name | Bedeutung | Funktionsbeschreibung |
|--------------------------------|----------|---------------|--|
| 00 _{Hex} Bitleiste | TI | Toggle-Bit In | Zeigt während eines Leseauftrags an, dass die Steuerung für weitere Daten bereit ist. |
| | KA | Kopffunktion | Schreib-/Lesekopf An- oder Abschaltung bei Bedarf. Aktiv = 0 Schreib-/Lesekopf ist angeschaltet. Inaktiv = 1 Schreib-/Lesekopf ist abgeschaltet. |
| | GR | Grundzustand | Veranlasst das BIS-System, in den Grundzustand für den jeweiligen Schreib-/Lesekopf zu gehen. Ein evtl. anstehender Auftrag wird abgebrochen. |
| | AV | Auftrag | Signalisiert dem Identifikations-System, dass ein Auftrag für den jeweiligen Schreib-/Lesekopf vorliegt. |

Bitte beachten Sie den prinzipiellen Ablauf auf [S. 29](#) und die Beispiele auf den [S. 50...62](#).

(Fortsetzung siehe nächste [S.](#))

Funktionsbeschreibung

Mode 1: Ausgangspuffer, Belegung und Erklärung

Erklärungen zum Ausgangspuffer (Fortsetzung)

| Sub- adresse | Bedeutung | Funktionsbeschreibung |
|--------------------------|-------------------|--|
| 01 _{Hex} | Befehlskennung | |
| | 00 _{Hex} | Kein Befehl vorhanden |
| | 01 _{Hex} | Datenträger lesen |
| | 02 _{Hex} | auf Datenträger schreiben |
| | 06 _{Hex} | Speichern des Programms im EEPROM für die Funktion Gemischter Datenzugriff |
| | 12 _{Hex} | Initialisieren der CRC_16-Datenprüfung |
| | 21 _{Hex} | Lesen bei der Funktion Gemischter Datenzugriff (entsprechend dem im EEPROM abgelegten Programm) |
| | 22 _{Hex} | Schreiben bei der Funktion Gemischter Datenzugriff (entsprechend dem im EEPROM abgelegten Programm) |
| oder | Daten | zum Schreiben auf den Datenträger |
| oder | Programmdaten | zum Schreiben auf das EEPROM. |

(Fortsetzung siehe nächste )

Bitte beachten Sie
den prinzipiellen
Ablauf auf  29 und
die Beispiele auf
den  50...62.

Funktionsbeschreibung

Mode 1: Ausgangspuffer, Belegung und Erklärung

Erklärungen zum Ausgangspuffer (Fortsetzung)

| Sub-adresse | Bedeutung | Funktionsbeschreibung |
|-------------------------|----------------------------------|---|
| 02_{Hex} | Anfangsadresse (Low Byte) | Adresse, ab der vom Datenträger gelesen bzw. auf den Datenträger geschrieben werden soll (das Low Byte deckt den Adressbereich von 0 bis 255 ab). |
| | oder Programm-Nr. | Nr. des im EEPROM abzulegenden Programms in Verbindung mit Befehlskennung 06 _{Hex} für die Funktion Gemischter Datenzugriff (Werte zwischen 01 _{Hex} und 0A _{Hex} erlaubt!). |
| | oder Programm-Nr. | Nr. des im EEPROM abgelegten Programms für Lese- oder Schreiboperationen in Verbindung mit Befehlskennung 21 _{Hex} oder 22 _{Hex} für die Funktion Gemischter Datenzugriff. |
| | oder Daten oder Programmdaten | zum Schreiben auf den Datenträger zum Schreiben auf das EEPROM. |
| 03_{Hex} | Anfangsadresse (High Byte) | Adresse, ab der vom Datenträger gelesen bzw. auf den Datenträger geschrieben werden soll (das High Byte deckt den Adressbereich von 256 bis 1999 ab). |
| | oder Daten | zum Schreiben auf den Datenträger |
| | oder Programmdaten | zum Schreiben auf das EEPROM. |
| 04_{Hex} | Anzahl Byte (Low Byte) | Anzahl Byte, die ab der Anfangsadresse gelesen bzw. geschrieben werden sollen (das Low Byte deckt den Umfang von 1 bis 255 Byte ab). |
| | oder Daten | zum Schreiben auf den Datenträger |
| | oder Programmdaten | zum Schreiben auf das EEPROM. |

(Fortsetzung siehe nächste □)

Bitte beachten Sie den prinzipiellen Ablauf auf □ 29 und die Beispiele auf den □□ 50...62.

Funktionsbeschreibung

Mode 1: Ausgangspuffer, Belegung und Erklärung

Erklärungen zum Ausgangspuffer (Fortsetzung)

| Sub- adresse | Bedeutung | Funktionsbeschreibung |
|--------------------------|----------------------------|--|
| 05 _{Hex} | Anzahl Byte (High Byte) | Anzahl Byte, die ab der Anfangsadresse gelesen bzw. geschrieben werden sollen (das High Byte deckt den Adressbereich von 256 bis 1999 ab). |
| oder | Daten | zum Schreiben auf den Datenträger |
| oder | Programmdaten | zum Schreiben auf das EEPROM. |
| 06 _{Hex} | Daten | zum Schreiben auf den Datenträger |
| oder | Programmdaten | zum Schreiben auf das EEPROM. |
| ... | Daten | zum Schreiben auf den Datenträger |
| oder | Programmdaten | zum Schreiben auf das EEPROM. |

Bitte beachten Sie
den prinzipiellen
Ablauf auf [S. 29](#) und
die Beispiele auf
den [S. 50...62](#).

Funktionsbeschreibung

Mode 1: Eingangspuffer, Belegung und Erklärung

Belegung des Eingangspuffers für einen Schreib-/Lesekopf (Darstellung gilt für Kopf 1)

| Subadresse \ Bit-Nr. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bitname |
|-------------------------------|------------|----|----|----|-------|----|-------|----|---------|
| 00 _{Hex} = Bitleiste | BB | HF | TO | IN | AF | AE | AA | CP | |
| 01 _{Hex} | Fehlercode | | | | oder | | Daten | | |
| 02 _{Hex} | | | | | Daten | | | | |
| 03 _{Hex} | | | | | Daten | | | | |
| 04 _{Hex} | | | | | Daten | | | | |
| 05 _{Hex} | | | | | Daten | | | | |
| 06 _{Hex} | | | | | Daten | | | | |
| ... | | | | | Daten | | | | |

Erklärungen zum Eingangspuffer

| Sub-adresse | Bit-name | Bedeutung | Funktionsbeschreibung |
|--------------------------------|----------|----------------|---|
| 00 _{Hex} Bitleiste | BB | betriebsbereit | Das BIS-Identifikations-System befindet sich in betriebsbereitem Zustand. |
| | HF | Head Fehler | Kabelbruch zum Schreib-/Lesekopf oder kein Schreib-/Lesekopf angeschlossen. |
| | TO | Toggle-Bit Out | beim Lesen: BIS hat neue/weitere Daten bereitgestellt. beim Schreiben: BIS ist bereit, neue/weitere Daten zu übernehmen. |
| Bit | IN | Input | Wenn der Parameter <i>DigIN_Mode</i> = 1 ist, zeigt dieses den Zustand des Eingangs an. |

(Fortsetzung siehe nächste Seite)

Bitte beachten Sie den prinzipiellen Ablauf auf Seite 29 und die Beispiele auf den Seiten 50...62.

Funktionsbeschreibung

Mode 1: Eingangspuffer, Belegung und Erklärung

Erklärungen zum Eingangspuffer (Fortsetzung)

| Sub-adresse | Bit-name | Bedeutung | Funktionsbeschreibung |
|--------------------------|----------|-----------------|---|
| 00 _{Hex} | | (Fortsetzung) | |
| Bitleiste | AF | Auftrag Fehler | Der Auftrag wurde fehlerhaft bearbeitet/abgebrochen. |
| | AE | Auftrag Ende | Der Auftrag wurde ohne Fehler beendet. |
| | AA | Auftrag Anfang | Der Auftrag wurde erkannt und begonnen. |
| | CP | Codetag Present | Datenträger im Schreib-/Lesebereich des Schreib-/Lesekopfs. |

| Sub-adresse | Bedeutung | Funktionsbeschreibung |
|--------------------------|-------------------|---|
| 01 _{Hex} | Fehlercode | Fehlernummer ist eingetragen, wenn Auftrag fehlerhaft bearbeitet oder abgebrochen wurde. Nur mit AF-Bit gültig! |
| | 00 _{Hex} | Kein Fehler. |
| | 01 _{Hex} | Lesen oder Schreiben nicht möglich, da kein Datenträger im Schreib-/Lesebereich des Schreib-/Lesekopfs vorhanden. |
| | 02 _{Hex} | Fehler beim Lesen. |
| | 03 _{Hex} | Datenträger wurde während des Lesens aus dem Schreib-/Lesebereich des Schreib-/Lesekopfs entfernt. |
| | 04 _{Hex} | Fehler beim Schreiben. |
| | 05 _{Hex} | Datenträger wurde während des Schreibens aus dem Schreib-/Lesebereich des Schreib-/Lesekopfs entfernt. |
| | | (Fortsetzung siehe nächste Seite) |

Bitte beachten Sie den prinzipiellen Ablauf auf Seite 29 und die Beispiele auf den Seiten 50...62.

Funktionsbeschreibung

Mode 1: Eingangspuffer, Belegung und Erklärung

Erklärungen zum Eingangspuffer (Fortsetzung)

| Sub-adresse | Bedeutung | Funktionsbeschreibung |
|--------------------------|--|--|
| 01 _{Hex} | Fehlercode | (Fortsetzung) |
| | 07 _{Hex} | AV-Bit ist gesetzt, aber die Befehlskennung fehlt oder ist ungültig. |
| | oder: | Anzahl Byte ist 00 _{Hex} . |
| | 09 _{Hex} | Kabelbruch zum angewählten Schreib-/Lesekopf oder Kopf nicht angeschlossen. |
| | 0C _{Hex} | Das EEPROM kann nicht gelesen/beschrieben werden. |
| | 0D _{Hex} | Gestörte Kommunikation mit dem Datenträger. |
| | 0E _{Hex} | Der CRC der gelesenen Daten stimmt nicht mit dem CRC auf dem Datenträger überein! |
| | 20 _{Hex} | Adressierung des Schreib-/Leseauftrags liegt außerhalb des Speicherbereiches des Datenträgers. |
| 21 _{Hex} | Aufruf einer Funktion, die bei dem Datenträger nicht möglich ist, der sich vor dem Schreib-/Lesekopf befindet. | |
| oder: | Daten | Daten, die vom Datenträger gelesen wurden. |
| 02 _{Hex} | Daten | Daten, die vom Datenträger gelesen wurden. |
| ... | Daten | Daten, die vom Datenträger gelesen wurden. |

Bitte beachten Sie den prinzipiellen Ablauf auf [S. 29](#) und die Beispiele auf den [S. 50...62](#).

Datenträger-Typen

Datenträger-Typ

Für die Auswerteeinheit BIS M-60_3 stehen folgende Datenträger zur Verfügung.

Mifare

| Balluff Datenträgertyp | Hersteller | Bezeichnung | Speicherkapazität | Speichertyp |
|------------------------|------------|----------------|-------------------|-------------|
| BIS M-1_ _-01 | Philips | Mifare Classic | 752 Byte | EEPROM |

ISO15693

| Balluff Datenträgertyp | Hersteller | Bezeichnung | Speicherkapazität | Speichertyp |
|----------------------------|-------------|-------------|-------------------|-------------|
| BIS M-1_ _-02 | Fujitsu | MB89R118 | 2000 Byte | FRAM |
| BIS M-1_ _-03 ¹ | Philips | SL2ICS20 | 112 Byte | EEPROM |
| BIS M-1_ _-04 ¹ | Texas Inst. | TAG-IT Plus | 256 Byte | EEPROM |
| BIS M-1_ _-05 ¹ | Infineon | SRF55V02P | 224 Byte | EEPROM |
| BIS M-1_ _-06 ¹ | EM | EM4135 | 288 Byte | EEPROM |
| BIS M-1_ _-07 ¹ | Infineon | SRF55V10P | 992 Byte | EEPROM |

Auf dem Datenträger befinden sich zusätzliche Speicherbereiche zur Konfiguration und geschützte Daten. Diese Bereiche lassen sich mit der Auswerteeinheit BIS M-60_3 nicht bearbeiten.

CT present

Bei CT present werden die ersten Nutzdaten vom Datenträger ausgelesen und auf den Eingangspuffer des DeviceNet gelegt (siehe □ 44). Ist die Funktion „Typ und Seriennummer bei CT present ausgeben“ parametrierbar, so wird der Datenträgertyp im Byte 1 des Eingangspuffers und anschließend die Seriennummer ausgegeben.



Bitte beachten Sie für die Parametrierung die □ 13ff.

¹ auf Anfrage

Funktionsbeschreibung

Mode 1: Datenträger bearbeiten

Lesen und Schreiben

Für die Durchführung eines Lese- oder Schreibauftrags muss sich ein Datenträger im aktiven Bereich des Schreib-/Lesekopfs befinden.

Ein Lese-/Schreibauftrag hat folgenden Ablauf (siehe Beispiele auf den  50ff):

1. Die Steuerung gibt auf den Ausgangspuffer:
 - die Befehlskennung an Subadresse 01_{Hex},
 - die Anfangsadresse, ab der gelesen/geschrieben werden soll, an Subadresse 02_{Hex}/03_{Hex},
 - die Anzahl Byte, die gelesen/geschrieben werden sollen, an Subadresse 04_{Hex}/05_{Hex},
 - das CT-Bit in der Bitleiste je nach Blockgröße des Datenträgers,
 - das AV-Bit in der Bitleiste auf high.
2. Die Auswerteeinheit:
 - übernimmt den Auftrag (AA-Bit in der Bitleiste des Eingangspuffers auf high),
 - beginnt, die Daten zu transportieren;
Lesen: vom Datenträger in den Eingangspuffer,
Schreiben: vom Ausgangspuffer auf den Datenträger.
Größere Datenmengen werden in Blöcken übertragen.
Dazu wird mit den Toggle-Bits ein Handshake zwischen Steuerung und Auswerteeinheit BIS M-60_3 ausgeführt.
3. Die Auswerteeinheit hat den Auftrag korrekt bearbeitet (AE-Bit in der Bitleiste des Eingangspuffers). Ist bei der Bearbeitung des Auftrags ein Fehler entstanden, wird eine Fehlernummer in die Subadresse 01_{Hex} des Eingangspuffers geschrieben und das AF-Bit in der Bitleiste des Eingangspuffers gesetzt.

Funktionsbeschreibung

Mode 1: Datenträger bearbeiten

Codetag Present (CP-Bit)

Kommt der Datenträger in den aktiven Bereich des Schreib-/Lesekopfs, signalisiert dies die Auswerteeinheit durch das Setzen des CP-Bit (Codetag Present).



Um das Lesen kleiner Datenmengen zu beschleunigen, stellt das Identifikations-System beim Erkennen eines Datenträgers sofort die ersten Byte des Datenträgers im Eingangspuffer des jeweiligen Schreib-/Lesekopfs zur Verfügung. Die Anzahl Byte, die übertragen wird, entspricht der eingestellten Puffergröße – 1 Byte (2 Byte bei 2. Bitleiste).

Ist der Parameter „Typ und Seriennummer ausgeben“ eingestellt, werden – anstelle der Nutzdaten – der Datenträger-Typ und die einmalige Seriennummer des Datenträgers ausgegeben.



Ist diese Funktion aktiviert, so werden keine Lesedaten bei CT present ausgegeben.

Die Daten sind nur nach der steigenden Flanke des CP-Bit in der Bitleiste des Eingangspuffers gültig. Sie bleiben gültig bis zur fallenden Flanke des CP-Bit, oder bis die Steuerung einen anderen Auftrag erteilt.

Anfangsadresse bei Auto-Lesen

Ist die Funktion Auto-Lesen aktiviert, werden die Daten ab einer festgelegten Anfangsadresse gelesen, sobald ein Datenträger erkannt wird. Mit der steigenden Flanke des CP-Bits werden diese Daten im Eingangspuffer bereitgestellt. Die Anfangsadresse wird durch die Parameter DP1_Start_Address (10) und DP2_Start_Address (11) für jeden Kopf festgelegt. Die Anfangsadressen können unterschiedlich sein. Die Anzahl der gelesenen Byte wird von der gewählten Größe des Eingangspuffers bestimmt, der auf beide Köpfe aufgeteilt ist.

Funktionsbeschreibung

Mode 1: Datenträger bearbeiten

Lesen und Schreiben im Dynamikbetrieb

Im normalen Betrieb wird ein Lese-/Schreibauftrag mit dem Setzen des AF-Bit und einer Fehlernummer von der Auswerteeinheit BIS M-60_3 abgelehnt, wenn sich kein Datenträger im aktiven Bereich des Schreib-/Lesekopfs befindet. Ist die Funktion Dynamikbetrieb konfiguriert, nimmt die Auswerteeinheit den Lese-/Schreibauftrag an und speichert ihn. Wird ein Datenträger erkannt, wird der gespeicherte Auftrag ausgeführt.

Lesen und Schreiben mit simultaner Datenübertragung

Lesen ohne simultane Datenübertragung: Bei einem Leseauftrag liest die Auswerteeinheit nach Erhalt der Anfangsadresse und der gewünschten Anzahl Byte zunächst alle gewünschten Daten vom Datenträger aus und setzt dann das AE-Bit. Danach werden die vom Datenträger gelesenen Daten in den Eingangspuffer geschrieben. Bei größeren Datenmengen erfolgt dies blockweise, gesteuert durch das Handshake mit den Toggle-Bits wie auf [43](#) beschrieben.

Lesen mit simultaner Datenübertragung: Bei einem Leseauftrag beginnt die Auswerteeinheit mit der Übertragung der Daten in den Eingangspuffer, sobald die erste, der Pufferlänge des Kopfes entsprechende Anzahl Byte (-1 Byte für die Bitleiste) vom Datenträger gelesen wurde, und zeigt dies durch Invertieren des TO-Bit an. Sobald die Steuerung das TI-Bit invertiert, überträgt die Auswerteeinheit die inzwischen gelesenen Daten zum Eingangspuffer. Dies wiederholt sich, bis die Auswerteeinheit die gewünschte Anzahl Daten vom Datenträger ausgelesen hat. Nun setzt die Auswerteeinheit das AE-Bit und gibt die restlichen Daten auf dem Eingangspuffer aus.

Schreiben ohne simultane Datenübertragung: Bei einem Schreibauftrag wartet die Auswerteeinheit, bis sie alle zu schreibenden Daten von der Steuerung erhalten hat. Erst danach werden die Daten auf den Datenträger geschrieben, wie auf [43](#) beschrieben.

Schreiben mit simultaner Datenübertragung: Bei einem Schreibauftrag beginnt die Auswerteeinheit mit dem Schreiben der Daten auf den Datenträger, sobald sie die ersten zu schreibenden Daten aus dem Ausgangspuffer von der Steuerung erhalten hat. Sind alle Daten auf den Datenträger geschrieben, wird das AE-Bit gesetzt.

Funktionsbeschreibung

Mode 1: Datenträger bearbeiten

Gemischter Datenzugriff

Im EEPROM der Auswerteeinheit BIS M-60_3 können kleine Schreib-/Leseprogramme abgespeichert werden.

Die Funktion Gemischter Datenzugriff ist sinnvoll, wenn die benötigten Informationen auf dem Datenträger an unterschiedlichen Adressen vorliegen. Diese Funktion erlaubt es, diese "gemischten", d.h. nicht zusammenhängend gespeicherten Daten vom Datenträger in einem Vorgang und mit nur einem Befehl auszulesen.

Es können 10 Programme mit bis zu 25 Anweisungen abgespeichert werden. Jede Programmanweisung beinhaltet eine Information Anfangsadresse und eine Information Anzahl Byte. Der Umfang der auszulesenden Daten darf maximal 2 kByte betragen.

Programm abspeichern:

Mit der Befehlskennung 06_{Hex} wird das Schreib-/Leseprogramm an die Auswerteeinheit BIS M-60_3 übergeben. Pro Befehl wird ein Programm abgespeichert. Es müssen immer alle 25 Programmsätze plus zusätzlich 2 Byte mit FF_{Hex}FF_{Hex} als Endekennung übergeben werden. Insgesamt sind somit **104 Byte** Informationen je Programm zu übertragen (einschließlich Befehlskennung und Programmnummer).



Die einzelnen Programmsätze müssen lückenlos aneinander anschließen. Sie müssen nacheinander übergeben und mit 2 Byte FF_{Hex}FF_{Hex} als Endekennung abgeschlossen werden. Es wird empfohlen, den verbleibenden, ungenutzten Speicherbereich mit FF_{Hex}FF_{Hex} zu füllen.

Bei doppelter Auswahl eines Adressbereichs werden die Daten entsprechend zweimal ausgegeben.

Funktionsbeschreibung

Mode 1: Datenträger bearbeiten

Gemischter Datenzugriff (Fortsetzung)

Folgende Darstellung soll den Aufbau eines Programms verdeutlichen:

| Programmaufbau | Subadresse | Wert | Wertebereich |
|--------------------------|-------------------|-------------------|---|
| Befehlskennung | 01 _{Hex} | 06 _{Hex} | |
| 1. Programmsatz | | | |
| Programmnummer | 02 _{Hex} | 01 _{Hex} | 01 _{Hex} bis 0A _{Hex} |
| 1. Datensatz: | | | |
| Anfangsadresse Low Byte | 03 _{Hex} | | |
| Anfangsadresse High Byte | 04 _{Hex} | | |
| Anzahl Byte Low Byte | 05 _{Hex} | | |
| Anzahl Byte High Byte | 06 _{Hex} | | |
| 2. Datensatz: | | | |
| ... | | | |
| 25. Datensatz: | | | |
| Anfangsadresse Low Byte | 03 _{Hex} | | |
| Anfangsadresse High Byte | 04 _{Hex} | | |
| Anzahl Byte Low Byte | 05 _{Hex} | | |
| Anzahl Byte High Byte | 06 _{Hex} | | |
| Endekennung | FF _{Hex} | FF _{Hex} | |

Um ein zweites Programm zu speichern wird der oben dargestellte Vorgang wiederholt.

Der Vorgang, wie diese Einstellungen in das EEPROM zu schreiben sind, wird im 6. Beispiel auf den □□ 57...59 dargestellt.

Das Auswechseln des EEPROM ist auf □ 80 für BIS M-6003 und auf □ 88 für BIS M-6023 beschrieben.

Funktionsbeschreibung

Mode 1: Datenträger bearbeiten

Vom Datenträger lesen, mit Programm Gemischter Datenzugriff

Mit der Befehlskennung 21_{Hex} können die Programmsätze, die im Programm hinterlegt sind, vom Datenträger ausgelesen werden. Der Anwender muss genau dokumentieren, welche Daten von wo und mit welcher Anzahl Byte für das gewählte Programm gelesen werden (siehe Beispiel 7 auf □ 60).

Auf Datenträger schreiben, mit Programm Gemischter Datenzugriff

Mit der Befehlskennung 22_{Hex} können die Programmsätze, die im Programm hinterlegt sind, auf den Datenträger geschrieben werden. Der Anwender muss genau dokumentieren, welche Daten von wo und mit welcher Anzahl Byte für das gewählte Programm geschrieben werden (siehe Beispiel 8 auf □ 61).

CRC_16- Initialisierung

Um das CRC_16-Verfahren verwenden zu können, müssen die Datenträger zunächst mit der Befehlskennung 12_{Hex} initialisiert werden (siehe □ 50). Die CRC_16-Initialisierung wird wie ein normaler Schreibauftrag verwendet. Dieser wird mit einer Fehlermeldung abgelehnt, wenn die Auswerteeinheit erkennt, dass der Datenträger nicht die richtigen CRC_16-Prüfsumme enthält. Datenträger ab Werksauslieferung (alle Daten sind 0) können sofort mit CRC-geprüften Daten beschrieben werden.

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.

Funktionsbeschreibung

Mode 1: Datenträger bearbeiten

CRC_16 und Codetag Present

Wurde CRC_16 parametrierung und es wird ein Datenträger erkannt, dessen CRC_16-Prüfsumme fehlerhaft ist, so werden die Lesedaten nicht ausgegeben und das CP-Bit in der Eingangs-Bitleiste wird nicht gesetzt. Die LED CT present wird eingeschaltet und der digitale Ausgang wird gesetzt – der Datenträger kann mit dem Initialisierungsbefehl (12_{Hex}) bearbeitet werden.

CRC_16 und Speicherkapazität

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:

| Datenträger | Balluff Datenträgertyp | Speicherkapazität | Nutzbare Byte bei CRC |
|-----------------|----------------------------|-------------------|-----------------------|
| Mifare | BIS M-1_-_-01 | 752 Byte | 658 Byte |
| ISO15693 | BIS M-1_-_-02 | 2000 Byte | 1750 Byte |
| | BIS M-1_-_-03 ¹ | 112 Byte | 98 Byte |
| | BIS M-1_-_-04 ¹ | 256 Byte | 224 Byte |
| | BIS M-1_-_-05 ¹ | 224 Byte | 196 Byte |
| | BIS M-1_-_-06 ¹ | 288 Byte | 252 Byte |
| | BIS M-1_-_-07 ¹ | 992 Byte | 868 Byte |

¹ auf Anfrage

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

1. Beispiel

Bei Konfiguration mit 8 Byte Puffergröße!

Initialisieren des Datenträgers für die CRC_16-Datenprüfung

Dieser Befehl entspricht im Ablauf einem Schreibbefehl. Anfangsadresse und Anzahl Byte müssen der maximal verwendeten Datenmenge entsprechen.

Im Beispiel soll der komplette Speicherbereich eines Datenträgers mit 752 Byte verwendet werden (BIS M-10 _-01/L). Da 2 Byte je Block für den CRC_16 verwendet werden, sind lediglich 658 Byte des Datenträgers für die Nutzbyte verfügbar.

Somit: Anfangsadresse = 0, Anzahl Byte = 658.

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|----------------------------------|
| 01 _{Hex} | Befehlskennung 12 _{Hex} |
| 02 _{Hex} | Anfangsadresse 00 _{Hex} |
| 03 _{Hex} | Anfangsadresse 00 _{Hex} |
| 04 _{Hex} | Anzahl Byte 92 _{Hex} |
| 05 _{Hex} | Anzahl Byte 02 _{Hex} |
| 00 _{Hex} | AV-Bit setzen |

- 3.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|------------------------|-----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten eintragen |
| 00 _{Hex} | TI-Bit invertieren |

- 5.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|------------------------|------------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten eintragen |
| 00 _{Hex} | TI-Bit invertieren |

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|-----------------------------------|
| 00 _{Hex} | AA-Bit setzen, TO-Bit invertieren |
|-------------------|-----------------------------------|

- 4.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|--|----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten kopieren |
| Subadresse des Eingangspuffers bearbeiten: | |
| 00 _{Hex} | TO-Bit invertieren |

- 6.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|--|-----------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten kopieren |
| Subadresse des Eingangspuffers bearbeiten: | |
| 00 _{Hex} | TO-Bit invertieren |

... Solange fortsetzen, bis der gesamte Speicherbereich geschrieben ist. Siehe nächste □.

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

1. Beispiel (Fortsetzung)

**Bei Konfiguration
mit 8 Byte
Puffergröße!**

Steuerung:

7.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|--|----------------------------------|
| 01 _{Hex} ...07 _{Hex} | Die letzten Byte Daten eintragen |
| 00 _{Hex} | TI-Bit invertieren |

9.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|-------------------|-------------------|
| 00 _{Hex} | AV-Bit rücksetzen |
|-------------------|-------------------|

Identifikations-System BIS M-60_3:

8.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|--|---------------------------------|
| 01 _{Hex} ...07 _{Hex} | Die letzten Byte Daten kopieren |
|--|---------------------------------|

Subadresse des Eingangspuffers bearbeiten:

| | |
|-------------------|---------------|
| 00 _{Hex} | AE-Bit setzen |
|-------------------|---------------|

10.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------------|
| 00 _{Hex} | AA-Bit und AE-Bit rücksetzen |
|-------------------|------------------------------|

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

2. Beispiel

Bei Parametrierung mit 8 Byte Puffergröße!

Lesen von 17 Byte ab Datenträgeradresse 10 (Datenträgertyp BIS M-10 _-01/L):

Steuerung:

1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|--|
| 01 _{Hex} | Befehlskennung 01 _{Hex} |
| 02 _{Hex} | Anfangsadresse Low Byte 0A _{Hex} |
| 03 _{Hex} | Anfangsadresse High Byte 00 _{Hex} |
| 04 _{Hex} | Anzahl Byte Low Byte 11 _{Hex} |
| 05 _{Hex} | Anzahl Byte High Byte 00 _{Hex} |
| 00 _{Hex} | AV-Bit setzen |

3.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten kopieren |
|------------------------|----------------------------------|

Subadresse des Ausgangspuffers bearbeiten:

| | |
|-------------------|--------------------|
| 00 _{Hex} | TI-Bit invertieren |
|-------------------|--------------------|

5.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|-----------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten kopieren |
|------------------------|-----------------------------------|

Subadresse des Ausgangspuffers bearbeiten:

| | |
|-------------------|--------------------|
| 00 _{Hex} | TI-Bit invertieren |
|-------------------|--------------------|

7.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|--------------------------------------|
| 01...03 _{Hex} | Die restlichen 3 Byte Daten kopieren |
|------------------------|--------------------------------------|

Subadresse des Ausgangspuffers bearbeiten:

| | |
|-------------------|-------------------|
| 00 _{Hex} | AV-Bit rücksetzen |
|-------------------|-------------------|

Identifikations-System BIS M-60_3:

2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|------------------------|-----------------------------------|
| 00 _{Hex} | AA-Bit setzen |
| 01...07 _{Hex} | Die ersten 7 Byte Daten eintragen |
| 00 _{Hex} | AE-Bit setzen |

4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|------------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten eintragen |
| 00 _{Hex} | TO-Bit invertieren |

6.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|---------------------------------------|
| 01...03 _{Hex} | Die restlichen 3 Byte Daten eintragen |
| 00 _{Hex} | TO-Bit invertieren |

8.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------------|
| 00 _{Hex} | AA-Bit und AE-Bit rücksetzen |
|-------------------|------------------------------|

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

**3. Beispiel
(wie 2. Beispiel,
jedoch mit
simultaner
Datenübertragung)**

**Bei Parametrierung
mit 8 Byte
Puffergröße!**

Lesen von 17 Byte ab Datenträgeradresse 10 mit simultaner Datenübertragung
(Datenträgertyp BIS M-10 _-01/L):

Während der Leseauftrag ausgeführt wird und sobald der Eingangspuffer gefüllt ist, werden die ersten Daten gesendet. Das AE-Bit wird erst gesetzt, wenn die Operation "Lesen" von der Auswerteeinheit beendet ist.

Die Rückmeldung "Auftrag Ende" = AE-Bit wird spätestens vor der Zusendung der letzten Daten sicher gesetzt. Der Zeitpunkt ist von der angeforderten Datenmenge, der Eingangspuffergröße und dem Zeitverhalten der Steuerung abhängig. Darauf wird in der nachfolgenden Darstellung durch die kursive Schreibweise *AE-Bit setzen* aufmerksam gemacht.

Steuerung:

1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|--|
| 01 _{Hex} | Befehlskennung 01 _{Hex} |
| 02 _{Hex} | Anfangsadresse Low Byte 0A _{Hex} |
| 03 _{Hex} | Anfangsadresse High Byte 00 _{Hex} |
| 04 _{Hex} | Anzahl Byte Low Byte 11 _{Hex} |
| 05 _{Hex} | Anzahl Byte High Byte 00 _{Hex} |
| 00 _{Hex} | AV-Bit setzen |

3.) Subadressen des Eingangspuffers bearbeiten:

| | |
|--|----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten kopieren |
| Subadresse des Ausgangspuffers bearbeiten: | |
| 00 _{Hex} | TI-Bit invertieren |

Identifikations-System BIS M-60_3:

2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|--|-----------------------------------|
| 00 _{Hex} | AA-Bit setzen |
| 01 _{Hex} ...07 _{Hex} | Die ersten 7 Byte Daten eintragen |
| 00 _{Hex} | TO-Bit invertieren |
| 00 _{Hex} | <i>AE-Bit setzen</i> |

4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|------------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten eintragen |
| 00 _{Hex} | TO-Bit invertieren |
| 00 _{Hex} | <i>AE-Bit setzen</i> |

Fortsetzung siehe nächste .

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

3. Beispiel (Fortsetzung)

(wie 2. Beispiel,
jedoch mit simultaner
Datenübertragung)

Bei Parametrierung mit 8 Byte Puffergröße!

Steuerung:

5.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|-----------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten kopieren |
|------------------------|-----------------------------------|

Subadresse des Ausgangspuffers bearbeiten:

| | |
|-------------------|--------------------|
| 00 _{Hex} | TI-Bit invertieren |
|-------------------|--------------------|

7.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|--------------------------------------|
| 01...03 _{Hex} | Die restlichen 3 Byte Daten kopieren |
|------------------------|--------------------------------------|

Subadresse des Ausgangspuffers bearbeiten:

| | |
|-------------------|-------------------|
| 00 _{Hex} | AV-Bit rücksetzen |
|-------------------|-------------------|

Identifikations-System BIS M-60_3:

6.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|---------------------------------------|
| 01...03 _{Hex} | Die restlichen 3 Byte Daten eintragen |
|------------------------|---------------------------------------|

| | |
|-------------------|--------------------|
| 00 _{Hex} | TO-Bit invertieren |
|-------------------|--------------------|

| | |
|-------------------|---------------|
| 00 _{Hex} | AE-Bit setzen |
|-------------------|---------------|

8.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------------|
| 00 _{Hex} | AA-Bit und AE-Bit rücksetzen |
|-------------------|------------------------------|

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

4. Beispiel

Bei Parametrierung
mit 8 Byte
Puffergröße!

Lesen von 30 Byte ab Datenträgeradresse 10 mit Lesefehler
(Datenträgertyp BIS M-10 _-01/L):

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|--|
| 01 _{Hex} | Befehlskennung 01 _{Hex} |
| 02 _{Hex} | Anfangsadresse Low Byte 0A _{Hex} |
| 03 _{Hex} | Anfangsadresse High Byte 00 _{Hex} |
| 04 _{Hex} | Anzahl Byte Low Byte 1E _{Hex} |
| 05 _{Hex} | Anzahl Byte High Byte 00 _{Hex} |
| 00 _{Hex} | AV-Bit setzen |

- 3.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|-----------------------|
| 01 _{Hex} | Fehlernummer kopieren |
|-------------------|-----------------------|

Subadresse des Ausgangspuffers bearbeiten:

| | |
|-------------------|-------------------|
| 00 _{Hex} | AV-Bit rücksetzen |
|-------------------|-------------------|

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

Wenn Fehler sofort eintritt:

| | |
|-------------------|------------------------|
| 00 _{Hex} | AA-Bit setzen |
| 01 _{Hex} | Fehlernummer eintragen |
| 00 _{Hex} | AF-Bit setzen |

- 4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------------|
| 00 _{Hex} | AA-Bit und AF-Bit rücksetzen |
|-------------------|------------------------------|

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

5. Beispiel

Bei Parametrierung
mit 8 Byte
Puffergröße!

Schreiben von 16 Byte ab Datenträgeradresse 20 (Datenträgertyp BIS M-10 _-01/L):

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|----------------------|--|
| 01 _{Hex} | Befehlskennung 02 _{Hex} |
| 02/03 _{Hex} | Anfangsadresse 14 _{Hex} / 00 _{Hex} |
| 04/05 _{Hex} | Anzahl Byte 10 _{Hex} / 00 _{Hex} |
| 00 _{Hex} | AV-Bit setzen |

- 3.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|------------------------|-----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten eintragen |
| 00 _{Hex} | TI-Bit invertieren |

- 5.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|------------------------|------------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten eintragen |
| 00 _{Hex} | TI-Bit invertieren |

- 7.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|------------------------|---------------------------------------|
| 01...02 _{Hex} | Die restlichen 2 Byte Daten eintragen |
| 00 _{Hex} | TI-Bit invertieren |

- 9.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|-------------------|-------------------|
| 00 _{Hex} | AV-Bit rücksetzen |
|-------------------|-------------------|

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|-----------------------------------|
| 00 _{Hex} | AA-Bit setzen, TO-Bit invertieren |
|-------------------|-----------------------------------|

- 4.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|--|----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten kopieren |
| Subadresse des Eingangspuffers bearbeiten: | |
| 00 _{Hex} | TO-Bit invertieren |

- 6.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|--|-----------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten kopieren |
| Subadresse des Eingangspuffers bearbeiten: | |
| 00 _{Hex} | TO-Bit invertieren |

- 8.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|--|--------------------------------------|
| 01...02 _{Hex} | Die restlichen 2 Byte Daten kopieren |
| Subadresse des Eingangspuffers bearbeiten: | |
| 00 _{Hex} | AE-Bit setzen |

- 10.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------------|
| 00 _{Hex} | AA-Bit und AE-Bit rücksetzen |
|-------------------|------------------------------|

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

6. Beispiel Programm Gemischter Datenzugriff abspeichern

Bei Parametrierung mit 8 Byte Puffergröße!

Abspeichern eines Programms für das Auslesen von 3 Datensätzen:

| | | | | |
|--------------|----------------|-----|-------------|----|
| 1. Datensatz | Anfangsadresse | 5 | Anzahl Byte | 7 |
| 2. Datensatz | Anfangsadresse | 75 | Anzahl Byte | 3 |
| 3. Datensatz | Anfangsadresse | 112 | Anzahl Byte | 17 |

Insgesamt werden bei der Operation ausgetauscht: 27 Byte

Für die Programmierung werden alle 104 Byte geschrieben.

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|----------------------------------|
| 01 _{Hex} | Befehlskennung 06 _{Hex} |
| 02 _{Hex} | Programmnummer 01 _{Hex} |
| 00 _{Hex} | AV-Bit setzen |

- 3.) Subadressen des Ausgangspuffers bearbeiten:

| | | |
|-------------------|-------------------|-------------------------------|
| 01 _{Hex} | 1. Anfangsadresse | (Low Byte) 05 _{Hex} |
| 02 _{Hex} | | (High Byte) 00 _{Hex} |
| 03 _{Hex} | 1. Anzahl Byte | (Low Byte) 07 _{Hex} |
| 04 _{Hex} | | (High Byte) 00 _{Hex} |
| 05 _{Hex} | 2. Anfangsadresse | (Low Byte) 4B _{Hex} |
| 06 _{Hex} | | (High Byte) 00 _{Hex} |
| 07 _{Hex} | 2. Anzahl Byte | (Low Byte) 03 _{Hex} |
| 00 _{Hex} | | TI-Bit invertieren |

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|-----------------------------------|
| 00 _{Hex} | AA-Bit setzen, TO-Bit invertieren |
|-------------------|-----------------------------------|

- 4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|--------------------|
| 00 _{Hex} | TO-Bit invertieren |
|-------------------|--------------------|

Fortsetzung siehe nächste 

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

**6. Beispiel
Programm
Gemischter
Datenzugriff
abspeichern
(Fortsetzung)**

**Bei Parametrierung
mit 8 Byte
Puffergröße!**

Steuerung:

5.) Subadressen des Ausgangspuffers bearbeiten:

| | | |
|-------------------|--------------------|-------------------------------|
| 01 _{Hex} | 2. Anzahl Byte | (High Byte) 00 _{Hex} |
| 02 _{Hex} | 3. Anfangsadresse | (Low Byte) 70 _{Hex} |
| 03 _{Hex} | | (High Byte) 00 _{Hex} |
| 04 _{Hex} | 3. Anzahl Byte | (Low Byte) 11 _{Hex} |
| 05 _{Hex} | | (High Byte) 00 _{Hex} |
| 06 _{Hex} | Endekennung | FF _{Hex} |
| 07 _{Hex} | | FF _{Hex} |
| 00 _{Hex} | TI-Bit invertieren | |

7.) Subadressen des Ausgangspuffers bearbeiten:

| | | |
|---|--------------------|---|
| 01 _{Hex} /02 _{Hex} | (nicht verwendet) | FF _{Hex} /FF _{Hex} |
| 03 _{Hex} /04 _{Hex} | (nicht verwendet) | FF _{Hex} /FF _{Hex} |
| 05 _{Hex} /06 _{Hex} /07 _{Hex} | (nicht verwendet) | FF _{Hex} /FF _{Hex} /FF _{Hex} |
| 00 _{Hex} | TI-Bit invertieren | |

Identifikations-System BIS M-60_3:

6.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|--------------------|
| 00 _{Hex} | TO-Bit invertieren |
|-------------------|--------------------|

8.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|--------------------|
| 00 _{Hex} | TO-Bit invertieren |
|-------------------|--------------------|

Alle nicht verwendeten Anfangsadressen und Anzahl Byte mit FF_{Hex} füllen! Fortsetzung siehe nächste .

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

6. Beispiel Programm Gemischter Datenzugriff abspeichern (Fortsetzung)

Bei Parametrierung mit 8 Byte Puffergröße!

Steuerung:

9.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|---|---|
| 01 _{Hex} /02 _{Hex} | (nicht verwendet) FF _{Hex} /FF _{Hex} |
| 03 _{Hex} /04 _{Hex} | (nicht verwendet) FF _{Hex} /FF _{Hex} |
| 05 _{Hex} /06 _{Hex} /07 _{Hex} | (nicht verwendet) FF _{Hex} /FF _{Hex} /FF _{Hex} |
| 00 _{Hex} | TI-Bit invertieren |

11.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|-------------------|-------------------|
| 00 _{Hex} | AV-Bit rücksetzen |
|-------------------|-------------------|

Identifikations-System BIS M-60_3:

10.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|---------------|
| 00 _{Hex} | AE-Bit setzen |
|-------------------|---------------|

12.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------------|
| 00 _{Hex} | AA-Bit und AE-Bit rücksetzen |
|-------------------|------------------------------|



Wir empfehlen sorgfältig zu dokumentieren, welche Parameter für Anfangsadressen und Anzahl Byte verwendet werden, um die gewünschten Datensätze zu schreiben/zu lesen.

Die Daten werden genau in der im Programm festgelegten Reihenfolge aneinandergereiht.

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

7. Beispiel Programm Gemischter Datenzugriff anwenden

Bei Parametrierung
mit 8 Byte
Puffergröße!

Lesen des Datenträgers mit Programm Nr. 1 (Datenträgertyp BIS M-10 _-01/L): Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|----------------------------------|
| 01 _{Hex} | Befehlskennung 21 _{Hex} |
| 02 _{Hex} | Programmnummer 01 _{Hex} |
| 00 _{Hex} | AV-Bit setzen |

- 3.) Subadressen des Eingangspuffers bearbeiten:

| | |
|--|----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten kopieren |
| Subadresse des Ausgangspuffers bearbeiten: | |
| 00 _{Hex} | TI-Bit invertieren |

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|------------------------|-----------------------------------|
| 00 _{Hex} | AA-Bit setzen |
| 01...07 _{Hex} | Die ersten 7 Byte Daten eintragen |
| 00 _{Hex} | AE-Bit setzen |

- 4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|------------------------|------------------------------------|
| 01...07 _{Hex} | Die zweiten 7 Byte Daten eintragen |
| 00 _{Hex} | TO-Bit invertieren |

... Insgesamt werden 27 Byte Daten ausgetauscht.
(Für den weiteren Verlauf siehe Beispiel 2 auf [S. 52](#)).



Während der Gemischte Datenzugriff abgearbeitet wird, ist der Dynamikbetrieb ausgeschaltet.

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

Während der Gemischte Datenzugriff abgearbeitet wird, ist der Dynamikbetrieb ausgeschaltet

8. Beispiel Programm Gemischter Datenzugriff anwenden

Bei Parametrierung mit 8 Byte Puffergröße!

Schreiben des Datenträgers mit Programm Nr. 1 (Datenträgertyp BIS M-10 _-01/L):

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|----------------------------------|
| 01 _{Hex} | Befehlskennung 22 _{Hex} |
| 02 _{Hex} | Programmnummer 01 _{Hex} |
| 00 _{Hex} | AV-Bit setzen |

- 3.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|------------------------|-----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten eintragen |
| 00 _{Hex} | T1-Bit invertieren |

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|-----------------------------------|
| 00 _{Hex} | AA-Bit setzen, TO-Bit invertieren |
|-------------------|-----------------------------------|

- 4.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|------------------------|----------------------------------|
| 01...07 _{Hex} | Die ersten 7 Byte Daten kopieren |
|------------------------|----------------------------------|

Subadresse des Eingangspuffers bearbeiten:

| | |
|-------------------|--------------------|
| 00 _{Hex} | TO-Bit invertieren |
|-------------------|--------------------|

... Insgesamt werden 27 Byte Daten ausgetauscht.

(Die weitere Bearbeitung der Puffer entspricht dem Beispiel 5 auf [S. 56](#)).



Während der Gemischte Datenzugriff abgearbeitet wird, ist der Dynamikbetrieb ausgeschaltet.

Funktionsbeschreibung

Mode 1: Beispiele für den Protokollablauf

9. Beispiel

Grundzustand des jeweiligen Schreib-/Lesekopfs erzeugen:

Beide Schreib-/Leseköpfe des Identifikations-Systems können unabhängig voneinander in den Grundzustand gebracht werden.

Steuerung:

1.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|-------------------|---------------|
| 00 _{Hex} | GR-Bit setzen |
|-------------------|---------------|

3.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|-------------------|-------------------|
| 00 _{Hex} | GR-Bit rücksetzen |
|-------------------|-------------------|

Identifikations-System BIS M-60_3:

2.) In den Grundzustand gehen;
Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|-------------------|
| 00 _{Hex} | BB-Bit rücksetzen |
|-------------------|-------------------|

4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|---------------|
| 00 _{Hex} | BB-Bit setzen |
|-------------------|---------------|

10. Beispiel

Schreib-/Lesekopf Abschaltung:

Im Normalbetrieb sind beide Köpfe aktiv. Bei ungünstiger Montage kann es allerdings zu einer gegenseitigen Beeinflussung der Köpfe kommen. In diesem Fall sollte der nicht benutzte Kopf abgeschaltet werden, um eine Interferenz zu vermeiden.

Steuerung:

1.) Subadresse des Ausgangspuffers bearbeiten:

| | |
|--------------------------------------|---------------|
| 00 _{Hex} /07 _{Hex} | KA-Bit setzen |
|--------------------------------------|---------------|

Durch Zurücksetzen des KA-Bits wird der Schreib-/Lesekopf wieder angeschaltet. Das Anschalten eines Kopfes kann bis zu einer Sekunde dauern, das Abschalten geht hingegen sehr viel schneller.

Funktionsbeschreibung

Mode 2: Ausgangspuffer, Belegung und Erklärung

Belegung des Ausgangspuffers für einen Schreib-/Lesekopf (Darstellung gilt für Kopf 1)

| Bit-Nr. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|-------|---|----|---|---|----|----|----|---------|
| Subadresse | | | | | | | | | |
| 00 _{Hex} = Bitleiste | | | KA | | | GR | RW | AV | Bitname |
| 01 _{Hex} | Daten | | | | | | | | |
| 02 _{Hex} | Daten | | | | | | | | |
| 03 _{Hex} | Daten | | | | | | | | |
| 04 _{Hex} | Daten | | | | | | | | |
| 05 _{Hex} | Daten | | | | | | | | |
| 06 _{Hex} | Daten | | | | | | | | |
| ... | Daten | | | | | | | | |
| letztes Byte: <i>Output length 1</i> | Daten | | | | | | | | |

Erklärungen zum Ausgangspuffer

| Sub-adresse | Bit-name | Bedeutung | Funktionsbeschreibung |
|--------------------------------|----------|----------------------|--|
| 00 _{Hex} Bitleiste | KA | Kopffunktion | Schreib-/Lesekopf An- oder Abschaltung bei Bedarf. Aktiv = 0 Schreib-/Lesekopf ist angeschaltet. Inaktiv = 1 Schreib-/Lesekopf ist abgeschaltet. |
| | GR | Grundzustand | Veranlasst das BIS-System, in den Grundzustand für den jeweiligen Schreib-/Lesekopf zu gehen. Ein evtl. anstehender Auftrag wird abgebrochen. |
| | RW | Schreib-/Leseauftrag | Signalisiert dem BIS-System, ob ein Schreib- oder ein Leseauftrag durchgeführt werden soll. 0 Leseauftrag durchführen 1 Schreibauftrag durchführen |

(Fortsetzung siehe nächste Seite)

Bitte beachten Sie den prinzipiellen Ablauf auf Seite 29 und die Beispiele auf den Seiten 71...74.

Funktionsbeschreibung

Mode 2: Ausgangspuffer, Belegung und Erklärung

Erklärungen zum Ausgangspuffer (Fortsetzung)

| Sub-adresse | Bit-name | Bedeutung | Funktionsbeschreibung |
|--------------------------------|------------------------|-----------------------------------|--|
| 00 _{Hex} Bitleiste | AV | Auftrag | Signalisiert dem Identifikations-System, dass ein Auftrag für den jeweiligen Schreib-/Lesekopf vorliegt. |
| Sub-adresse | Bedeutung | Funktionsbeschreibung | |
| 01 _{Hex} | Daten | zum Schreiben auf den Datenträger | |
| ... | Daten | zum Schreiben auf den Datenträger | |
| Letztes Byte: | | | |
| | <i>Output length 1</i> | bei Kopf 1 | |
| | <i>Output length 2</i> | bei Kopf 2 | |
| | Daten | zum Schreiben auf den Datenträger | |

Bitte beachten Sie den prinzipiellen Ablauf auf [S. 29](#) und die Beispiele auf den [S. 71...74](#).

Funktionsbeschreibung

Mode 2: Eingangspuffer, Belegung und Erklärung

Belegung des Eingangspuffers für einen Schreib-/Lesekopf (Darstellung gilt für Kopf 1)

| Subadresse \ Bit-Nr. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Bitname |
|--|------------|----|---|-------|----|-------|----|----|---------|
| 00 _{Hex} = Bitleiste | BB | HF | | IN | AF | AE | AA | CP | |
| 01 _{Hex} | Fehlercode | | | oder | | Daten | | | |
| 02 _{Hex} | | | | Daten | | | | | |
| 03 _{Hex} | | | | Daten | | | | | |
| 04 _{Hex} | | | | Daten | | | | | |
| 05 _{Hex} | | | | Daten | | | | | |
| 06 _{Hex} | | | | Daten | | | | | |
| ... | | | | Daten | | | | | |
| letztes Byte: <i>Input length 1</i> | | | | Daten | | | | | |

Erklärungen zum Eingangspuffer

| Sub-adresse | Bit-name | Bedeutung | Funktionsbeschreibung |
|-----------------------------------|----------|----------------|---|
| 00 _{Hex} Bitleiste | BB | betriebsbereit | Das BIS-Identifikations-System befindet sich in betriebsbereitem Zustand. |
| | HF | Head Fehler | Kabelbruch zum Schreib-/Lesekopf oder kein Schreib-/Lesekopf angeschlossen. |
| (Fortsetzung siehe nächste Seite) | | | |

Bitte beachten Sie den prinzipiellen Ablauf auf Seite 29 und die Beispiele auf den Seiten 71...74.

Funktionsbeschreibung

Mode 2: Eingangspuffer, Belegung und Erklärung

Erklärungen zum Eingangspuffer (Fortsetzung)

| Sub-adresse | Bit-name | Bedeutung | Funktionsbeschreibung |
|--------------------------|----------|-----------------|---|
| 00 _{Hex} | | (Fortsetzung) | |
| Bitleiste Bit | IN | Input | Wenn der Parameter <i>DigIN_Mode</i> = 1 ist, zeigt dieses den Zustand des Eingangs an. |
| | AF | Auftrag Fehler | Der Auftrag wurde fehlerhaft bearbeitet/abgebrochen. |
| | AE | Auftrag Ende | Der Auftrag wurde ohne Fehler beendet. |
| | AA | Auftrag Anfang | Der Auftrag wurde erkannt und begonnen. |
| | CP | Codetag Present | Datenträger im Schreib-/Lesebereich des Schreib-/Lesekopfs. |

| Sub-adresse | Bedeutung | Funktionsbeschreibung |
|--------------------------|--------------------------|---|
| 01 _{Hex} | Fehlercode | Fehlernummer ist eingetragen, wenn Auftrag fehlerhaft bearbeitet oder abgebrochen wurde. Nur mit AF-Bit gültig! |
| | 00 _{Hex} | Kein Fehler. |
| | 01 _{Hex} | Lesen oder Schreiben nicht möglich, da kein Datenträger im Schreib-/Lesebereich des Schreib-/Lesekopfs vorhanden. |
| | 02 _{Hex} | Fehler beim Lesen. |
| | 03 _{Hex} | Datenträger wurde während des Lesens aus dem Schreib-/Lesebereich des Schreib-/Lesekopfs entfernt. |
| | 04 _{Hex} | Fehler beim Schreiben. |
| | | (Fortsetzung siehe nächste Seite) |

Bitte beachten Sie den prinzipiellen Ablauf auf Seite 29 und die Beispiele auf den Seiten 71...74.

Funktionsbeschreibung

Mode 2: Eingangspuffer, Belegung und Erklärung

Erklärungen zum Eingangspuffer (Fortsetzung)

| Sub-adresse | Bedeutung | Funktionsbeschreibung |
|-------------------------|-----------------------|--|
| 01_{Hex} | Fehlercode | (Fortsetzung) |
| | 05 _{Hex} | Datenträger wurde während des Schreibens aus dem Schreib-/Lesebereich des Schreib-/Lesekopfs entfernt. |
| | 07 _{Hex} | AV-Bit ist gesetzt, aber die Befehlskennung fehlt oder ist ungültig. |
| | oder: | Anzahl Byte ist 00 _{Hex} . |
| | 09 _{Hex} | Kabelbruch zum angewählten Schreib-/Lesekopf oder Kopf nicht angeschlossen. |
| | 0D _{Hex} | Gestörte Kommunikation mit dem Datenträger. |
| | 20 _{Hex} | Adressierung des Schreib-/Leseauftrags liegt außerhalb des Speicherbereichs des Datenträgers. |
| | 21 _{Hex} | Aufruf einer Funktion, die bei dem Datenträger nicht möglich ist, der sich vor dem Schreib-/Lesekopf befindet. |
| | oder: Daten | Daten, die vom Datenträger gelesen wurden. |
| 02_{Hex} | Daten | Daten, die vom Datenträger gelesen wurden. |
| ... | Daten | Daten, die vom Datenträger gelesen wurden. |
| Letztes Byte: | | |
| | <i>Input length 1</i> | bei Kopf 1 |
| | <i>Input length 2</i> | bei Kopf 2 |
| | Daten | Daten, die vom Datenträger gelesen wurden. |

Bitte beachten Sie den prinzipiellen Ablauf auf [S. 29](#) und die Beispiele auf den [S. 71...74](#).

Funktionsbeschreibung

Mode 2: Datenträger bearbeiten

Lesen und Schreiben

Für die Durchführung eines Lese- oder Schreibauftrags muss sich ein Datenträger im aktiven Bereich des Schreib-/Lesekopfs befinden.

Ein Lese-/Schreibauftrag hat folgenden Ablauf (siehe Beispiele auf den  71ff):

1. Die Steuerung gibt auf den Ausgangspuffer:
 - das RW-Bit zur Auswahl, ob ein Lese- (RW = 0) oder Schreibbefehl (RW-Bit = 1) ausgeführt werden soll,
 - bei einem Schreibauftrag die Schreibdaten,
 - das AV-Bit auf high, um der Auswerteeinheit einen neuen Auftrag zu signalisieren.
2. Die Auswerteeinheit:
 - übernimmt den Auftrag und bei einem Schreibauftrag die Daten aus dem Ausgangspuffer,
 - setzt das AA-Bit (Auftrag wurde angenommen) im Eingangspuffer,
 - führt den Auftrag mit den parametrisierten Werten für Anfangsadresse und Anzahl Byte auf dem Datenträger aus,
 - gibt bei einem korrekt ausgeführten Auftrag des AE-Bit (Auftrag korrekt beendet) und bei einem fehlerhaft ausgeführten Auftrag das AF-Bit (Auftrag fehlerhaft beendet) auf den Eingangspuffer,
 - sendet bei einem Leseauftrag die Lesedaten.

Funktionsbeschreibung

Mode 2: Datenträger bearbeiten

Codetag Present (CP-Bit)

Kommt der Datenträger in den aktiven Bereich des Schreib-/Lesekopfs, signalisiert dies die Auswerteeinheit durch das Setzen des CP-Bit (Codetag Present).



Um das Lesen kleiner Datenmengen zu beschleunigen, stellt das Identifikations-System beim Erkennen eines Datenträgers sofort die ersten Byte des Datenträgers im Eingangspuffer des jeweiligen Schreib-/Lesekopfs zur Verfügung. Die Anzahl Byte, die übertragen wird, entspricht dem Parameter *Input length*.

Ist der Parameter „Typ und Seriennummer ausgeben“ eingestellt, werden – anstelle der Nutzdaten – der Datenträger-Typ und die einmalige Seriennummer des Datenträgers ausgegeben.



Ist diese Funktion aktiviert, so werden keine Lesedaten bei CT present ausgegeben.

Die Daten sind nur nach der steigenden Flanke des CP-Bit in der Bitleiste des Eingangspuffers gültig. Sie bleiben gültig bis zur fallenden Flanke des CP-Bit, oder bis die Steuerung einen anderen Auftrag erteilt.

Anfangsadresse bei Auto-Lesen

Ist die Funktion Auto-Lesen aktiviert, werden die Daten ab einer festgelegten Anfangsadresse gelesen, sobald ein Datenträger erkannt wird. Mit der steigenden Flanke des CP-Bits werden diese Daten im Eingangspuffer bereitgestellt. Die Anfangsadresse wird durch die Parameter *DP1_Start_Address* (10) und *DP2_Start_Address* (11) für jeden Kopf festgelegt. Die Anfangsadressen können unterschiedlich sein. Die Anzahl der gelesenen Byte wird von der gewählten Größe des Parameters *Input length 1* (bei Kopf 1) bzw. *Input length 2* (bei Kopf 2) festgelegt.

Funktionsbeschreibung

Mode 2: Datenträger bearbeiten

Lesen und Schreiben im Dynamikbetrieb

Im normalen Betrieb wird ein Lese-/Schreibauftrag mit dem Setzen des AF-Bit und einer Fehlernummer von der Auswerteeinheit BIS M-60_3 abgelehnt, wenn sich kein Datenträger im aktiven Bereich des Schreib-/Lesekopfs befindet. Ist die Funktion Dynamikbetrieb konfiguriert, nimmt die Auswerteeinheit den Lese-/Schreibauftrag an und speichert ihn. Wird ein Datenträger erkannt, wird der gespeicherte Auftrag ausgeführt.

CRC-Initialisierung

Die CRC-Initialisierung ist im Mode 2 nicht möglich.

Funktionsbeschreibung

Mode 2: Beispiele für den Protokollablauf

1. Beispiel

Leseauftrag an Kopf 1 mit Parameter *Input length 1 = 12 Byte, Output length 1 = 8 Byte* (Datenträgertyp BIS M-10 _-01/L):

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|------------------------|
| 00 _{Hex} | RW-Bit = 0, AV-Bit = 1 |
|-------------------|------------------------|

- 3.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-----------------------|----------------|
| 01 _{Hex} | Daten kopieren |
| ... | Daten kopieren |
| <i>Input length 1</i> | Daten kopieren |

Subadresse des Ausgangspuffers bearbeiten:

| | |
|-------------------|------------|
| 00 _{Hex} | AV-Bit = 0 |
|-------------------|------------|

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-----------------------|------------------------|
| 00 _{Hex} | AA-Bit = 1, AE-Bit = 1 |
| 01 _{Hex} | Daten eintragen |
| ... | Daten eintragen |
| <i>Input length 1</i> | Daten eintragen |

- 4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------|
| 00 _{Hex} | AA-Bit = 0, AE-Bit = 0 |
|-------------------|------------------------|

Funktionsbeschreibung

Mode 2: Beispiele für den Protokollablauf

2. Beispiel

Leseauftrag an Kopf 1 mit Parameter *Input length 1* = 12 Byte, *Output length 1* = 8 Byte mit Lesefehler (Datenträgertyp BIS M-10 _-01/L):

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-------------------|------------------------|
| 00 _{Hex} | RW-Bit = 0, AV-Bit = 1 |
|-------------------|------------------------|

- 3.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|-------------------|------------|
| 00 _{Hex} | AV-Bit = 0 |
|-------------------|------------|

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|-----------------------|------------------------|
| 00 _{Hex} | AA-Bit = 1, AF-Bit = 1 |
| 01 _{Hex} | ungültig |
| ... | ungültig |
| <i>Input length 1</i> | ungültig |

- 4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------|
| 00 _{Hex} | AA-Bit = 0, AF-Bit = 0 |
|-------------------|------------------------|

Funktionsbeschreibung

Mode 2: Beispiele für den Protokollablauf

3. Beispiel

Schreibauftrag an Kopf 1 mit Parameter *Input length 1* = 12 Byte, *Output length 1* = 8 Byte (Datenträgertyp BIS M-10 _-01/L):

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|------------------------|------------------------|
| 01 _{Hex} | Daten eintragen |
| ... | Daten eintragen |
| <i>Output length 1</i> | Daten eintragen |
| 00 _{Hex} | RW-Bit = 1, AV-Bit = 1 |

- 3.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|-------------------|------------|
| 00 _{Hex} | AV-Bit = 0 |
|-------------------|------------|

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|--|----------------|
| 00 _{Hex} | AA-Bit = 1 |
| Subadresse des Ausgangspuffers bearbeiten: | |
| 01 _{Hex} | Daten kopieren |
| ... | Daten kopieren |
| <i>Output length 1</i> | Daten kopieren |
| Subadresse des Eingangspuffers bearbeiten: | |
| 00 _{Hex} | AE-Bit = 1 |

- 4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------|
| 00 _{Hex} | AA-Bit = 0, AE-Bit = 0 |
|-------------------|------------------------|

Funktionsbeschreibung

Mode 2: Beispiele für den Protokollablauf

4. Beispiel

Schreibauftrag an Kopf 1 mit Parameter *Input length 1* = 12 Byte, *Output length 1* = 8 Byte mit Schreibfehler (Datenträgertyp BIS M-10 _-01/L):

Steuerung:

- 1.) Subadressen des Ausgangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|------------------------|------------------------|
| 01 _{Hex} | Daten eintragen |
| ... | Daten eintragen |
| <i>Output length 1</i> | Daten eintragen |
| 00 _{Hex} | RW-Bit = 1, AV-Bit = 1 |

- 3.) Subadressen des Ausgangspuffers bearbeiten:

| | |
|-------------------|------------|
| 00 _{Hex} | AV-Bit = 0 |
|-------------------|------------|

Identifikations-System BIS M-60_3:

- 2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

| | |
|--|----------------|
| 00 _{Hex} | AA-Bit = 1 |
| Subadresse des Ausgangspuffers bearbeiten: | |
| 01 _{Hex} | Daten kopieren |
| ... | Daten kopieren |
| <i>Output length 1</i> | Daten kopieren |

Subadresse des Eingangspuffers bearbeiten:

| | |
|-------------------|------------|
| 00 _{Hex} | AF-Bit = 1 |
|-------------------|------------|

- 4.) Subadressen des Eingangspuffers bearbeiten:

| | |
|-------------------|------------------------|
| 00 _{Hex} | AA-Bit = 0, AF-Bit = 0 |
|-------------------|------------------------|

5. Beispiel

Schreib-/Lesekopf Abschaltung:

Im Normalbetrieb sind beide Köpfe aktiv. Bei ungünstiger Montage kann es allerdings zu einer gegenseitigen Beeinflussung der Köpfe kommen. In diesem Fall sollte der nicht benutzte Kopf abgeschaltet werden, um eine Interferenz zu vermeiden.

Steuerung:

- 1.) Subadresse des Ausgangspuffers bearbeiten:

| | |
|--------------------------------------|---------------|
| 00 _{Hex} /07 _{Hex} | KA-Bit setzen |
|--------------------------------------|---------------|

Durch Zurücksetzen des KA-Bits wird der Schreib-/Lesekopf wieder angeschaltet. Das Anschalten eines Kopfes kann bis zu einer Sekunde dauern, das Abschalten geht hingegen sehr viel schneller.

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

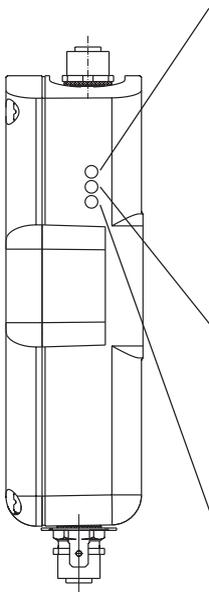
| Datenträger mit je 16 Byte/Block | BIS M-1__-01 | BIS M-1__-02 |
|--|--------------|--------------|
| Zeit zur Datenträgererkennung/serial ID | ≤ 20 ms | ≤ 30 ms |
| Schreiben von Byte 0 bis 15 | ≤ 40 ms | ≤ 65 ms |
| für jeweils weitere angebrochene 16 Byte addieren Sie weitere | ≤ 30 ms | ≤ 45 ms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



Alle Angaben sind typische Werte. Abweichungen sind je nach Anwendung und Kombination von Schreib-/Lesekopf und Datenträger möglich!
Die Angaben gelten für den statischen Betrieb, keine CRC_16-Datenprüfung.

Funktionsanzeigen

Funktionsanzeigen am BIS M-60_3



Über die drei seitlichen LED meldet die Auswerteeinheit BIS M-60_3 die wichtigsten Betriebszustände des Identifikations-Systems.

| Betriebszustand | LED | Bedeutung |
|----------------------------|-------------|---|
| MOD / NET STATUS | aus | Gerät ist nicht betriebsbereit – Gerät hat den Dup_MAC-ID Test noch nicht durchgeführt – Gerät wird nicht mit Spannung versorgt |
| | grün blinkt | Gerät arbeitet im normalen Modus, die Verbindung zum Master ist nicht eingerichtet |
| | grün | Gerät arbeitet im normalen Modus, die Verbindung zum Master ist eingerichtet |
| | rot blinkt | behebbarer Fehler und/oder eine oder mehrere I/O-Connection befinden sich im Time-Out-Status |
| CT1 Present / operating | rot | nicht behebbarer Fehler. Gerät kann Kommunikation am BUS nicht herstellen |
| | grün | Datenträger schreib-/lesebereit am Schreib-/Lesekopf 1. |
| | gelb | Lese-/Schreibauftrag am Schreib-/Lesekopf 1 wird bearbeitet. |
| | gelb blinkt | Kabelbruch zum Schreib-/Lesekopf 1 bzw. nicht angeschlossen. |
| CT2 Present / operating | aus | Kein Datenträger im Schreib-/Lesebereich von Schreib-/Lesekopf 1. |
| | grün | Datenträger schreib-/lesebereit am Schreib-/Lesekopf 2. |
| | gelb | Lese-/Schreibauftrag am Schreib-/Lesekopf 2 wird bearbeitet. |
| | gelb blinkt | Kabelbruch zum Schreib-/Lesekopf 2 bzw. nicht angeschlossen. |
| | aus | Kein Datenträger im Schreib-/Lesebereich von Schreib-/Lesekopf 2. |

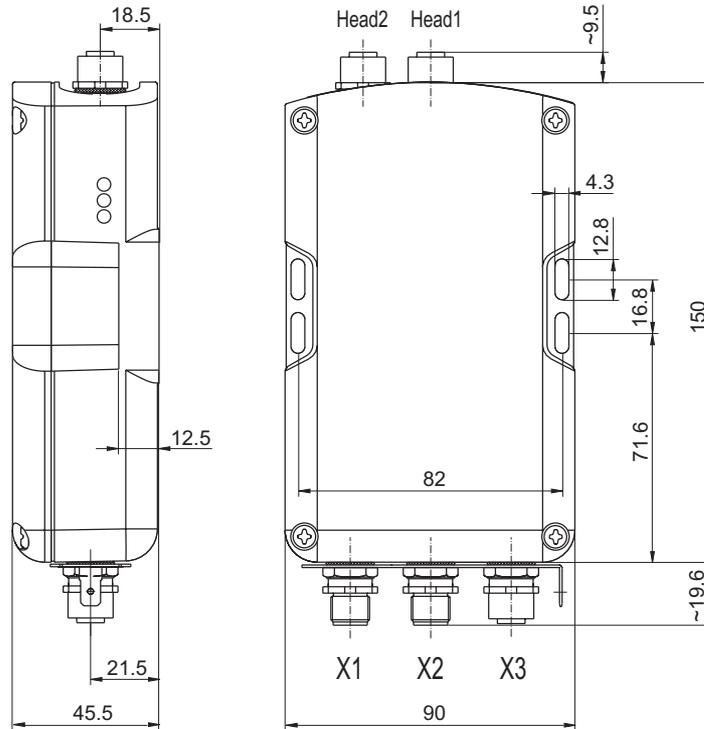
Wenn alle drei LED synchron blinken, liegt ein Hardwarefehler vor. Das Gerät muss zur Reparatur.

BIS M-6003

Montage der Auswerteeinheit

Montage der Auswerteeinheit BIS M-6003

Die Auswerteeinheit wird an den 4 seitlichen Langlöchern mit Schrauben M4 befestigt.



Maße BIS M-6003

BIS M-6003

Öffnen der Auswerteeinheit

Öffnen der Auswerteeinheit BIS M-6003

Um die folgenden Aktionen ausführen zu können, ist die Auswerteeinheit BIS M-6003 zu öffnen:

- DeviceNet MAC-ID einstellen,
- Baudrate einstellen,
- EEPROM wechseln.

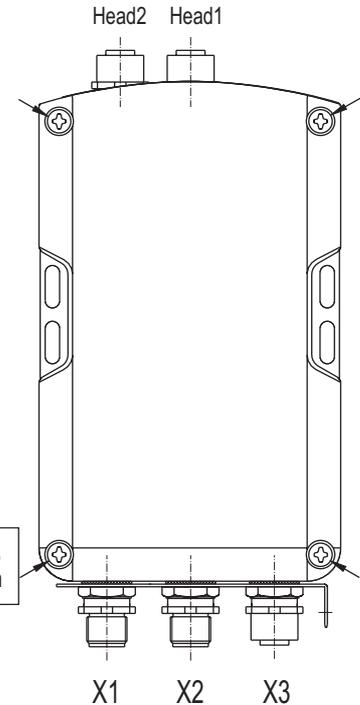


Sorgen Sie vor dem Öffnen dafür, dass das Gerät spannungsfrei geschaltet ist.

Öffnen Sie die 4 Schrauben am BIS M-6003 und entfernen Sie den Deckel.

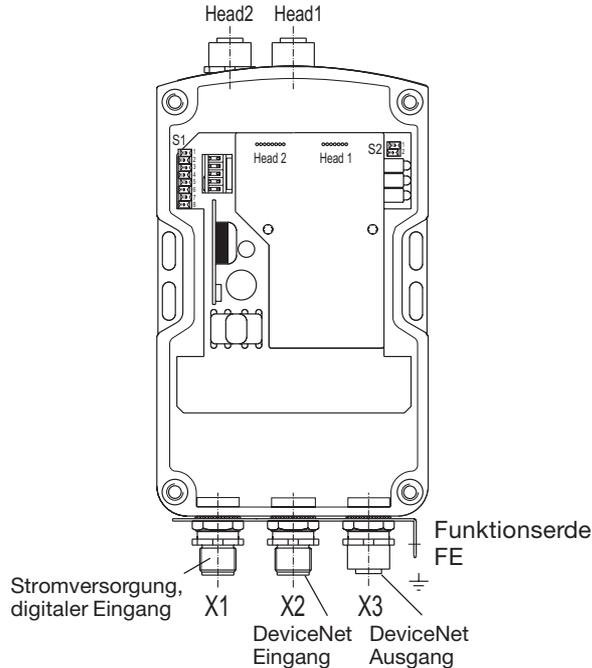
Führen Sie die gewünschte Aktion aus.

Befestigung des Deckels (4 Schrauben),
max. zulässiges Anzugsdrehmoment: 0,15 Nm



Öffnen der Auswerteeinheit

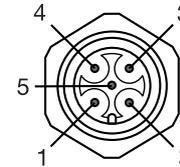
Schnittstellen für Auswertereinheit BIS M-6003



Der Erdanschluss ist je nach Anlage (Potenzialausgleich) direkt oder über eine RC-Kombination an Erde zu legen.

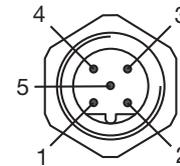
Beim Anschluss der Bus-Leitungen ist darauf zu achten, dass der Schirm eine einwandfreie Verbindung zum Steckergehäuse hat.

X1, Stromversorgung, digitaler Eingang



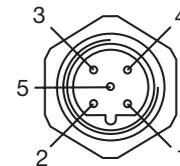
| Pin | Funktion |
|-----|----------|
| 1 | +Vs |
| 2 | -IN |
| 3 | -Vs |
| 4 | +IN |
| 5 | |

X2, DeviceNet-Eingang (Stecker)



| Pin | Funktion |
|-----|----------|
| 1 | Drain |
| 2 | V+ |
| 3 | V- |
| 4 | CAN_H |
| 5 | CAN_L |

X3, DeviceNet-Ausgang (Buchse)



| Pin | Funktion |
|-----|----------|
| 1 | Drain |
| 2 | V+ |
| 3 | V- |
| 4 | CAN_H |
| 5 | CAN_L |

BIS M-6003 Wechseln des EEPROM

EEPROM in der Auswerteeinheit BIS M-6003 wechseln

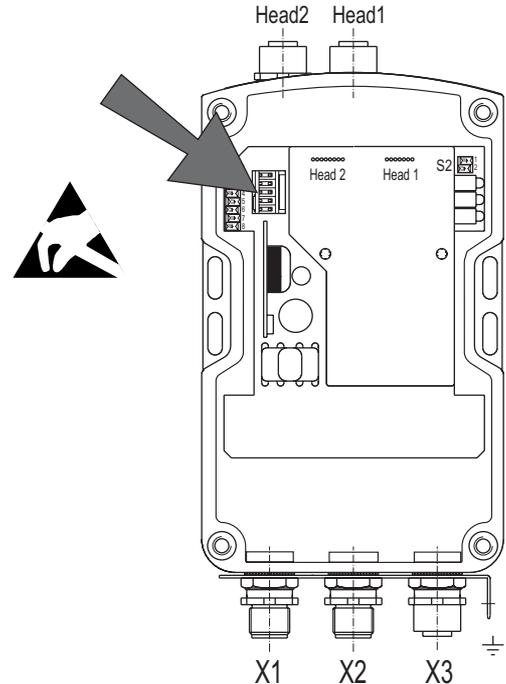


Sorgen Sie vor dem Öffnen dafür, dass das Gerät spannungsfrei geschaltet ist.

Um das EEPROM beim Wechseln nicht zu beschädigen, beachten Sie bitte die Regeln für den Umgang mit elektrostatisch gefährdeten Bauelementen.

Der Wechsel des EEPROM geschieht durch Aus- und Einstecken.

Um das EEPROM zu wechseln, ist die Auswerteeinheit entsprechend den Angaben auf  78 zu öffnen.

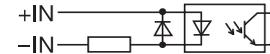


Lage des EEPROM

BIS M-6003

Technische Daten

| | | |
|-------------------------------|--|--|
| Abmessungen, Gewicht | Gehäuse Abmessungen Gewicht | Kunststoff ABS ca. 179 x 90 x 45,5 mm ca. 500 g |
| Betriebsbedingungen | Umgebungstemperatur | 0 °C bis + 60 °C |
| Schutzart | Schutzart | IP 65 (in angeschlossenem Zustand) |
| Anschlussart | Einbaustecker X1 für V_s, IN Einbaustecker X2 für DeviceNet Eingang Einbaubuchse X3 für DeviceNet Ausgang | 5-polig (Stift) 5-polig (Stift) 5-polig (Buchse) |
| Elektrische Anschlüsse | Betriebsspannung V_s, Eingang Restwelligkeit Stromaufnahme Anschluss der Betriebsspannung V _s Digitaler Eingang (+IN, -IN) Steuerspannung aktiv Steuerspannung inaktiv Eingangsstrom bei 24 V Verzögerungszeit typisch DeviceNet Eingang X2, Ausgang X3 Schreib-/Lesekopf | DC 24 V ± 20 % ≤ 10 % ≤ 600 mA über Eingang X1 über Optokoppler getrennt 4 V bis 40 V 1,5 V bis -40 V 11 mA 5 ms serielle Schnittstelle für DeviceNet-Teilnehmer 2 x Einbaustecker 8-polig (Buchse) für alle Schreib-/Leseköpfe BIS M-3_ _ mit 8-poligem Stecker (Stift) |



BIS M-6003

Technische Daten

Funktionsanzeigen

DeviceNet-Betriebszustände:
MOD / NET STATUS LED grün / rot

BIS-Betriebszustände:
CT1 Present / operating LED grün / gelb
CT2 Present / operating LED grün / gelb



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-6003

Bestellinformationen

Typenschlüssel

BIS M-6003-025-050-03-ST12

Balluff Identifikations-System

Baureihe M Schreib-/Lesesystem

Hardware-Typ

6003 = Kunststoffgehäuse, DeviceNet

Software-Typ

025 = DeviceNet

Schreib-/Lesekopf

050 = mit 2 Steckverbindern für Schreib-/Leseköpfe BIS M-3_ _

Schnittstelle

03 = BUS-Varianten

Kundenanschluss

ST12 = Steckervariante

BIS M-6003

Bestellinformationen

**Zubehör
(optional, nicht im
Lieferumfang)**

Das Zubehör zum BIS M-6_ _ _-... finden Sie im Balluff **Industrial Identification** Katalog. Den Katalog können Sie im Internet unter „www.balluff.de“ herunterladen.

BIS M-6023

Öffnen der Auswerteeinheit

Öffnen der Auswerteeinheit BIS M-6023

Um die folgenden Aktionen ausführen zu können, ist die Auswerteeinheit BIS M-6023 zu öffnen:

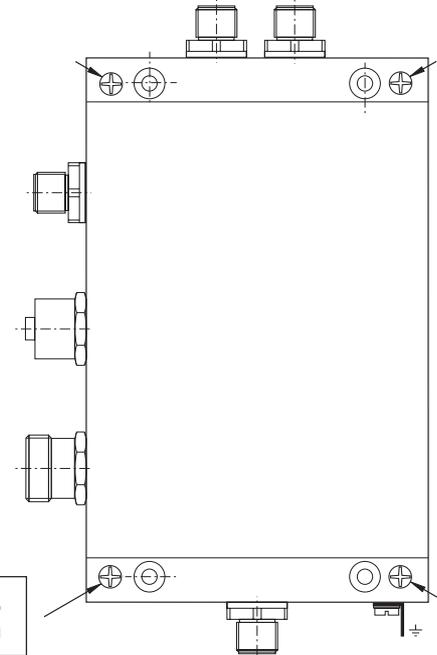
- DeviceNet MAC-ID einstellen,
- Baudrate einstellen,
- EEPROM wechseln.



Sorgen Sie vor dem Öffnen dafür, dass das Gerät spannungsfrei geschaltet ist.

Öffnen Sie die 4 Schrauben am BIS M-6023 und entfernen Sie den Deckel.

Führen Sie die gewünschte Aktion aus.



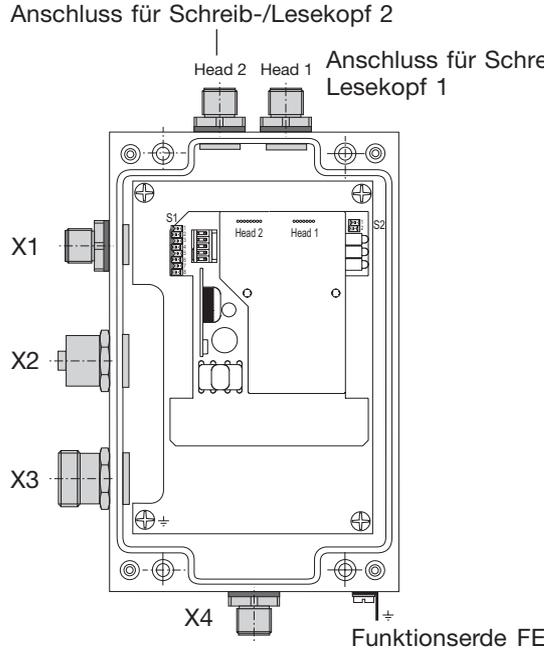
Befestigung des Deckels (4 Schrauben),
max. zulässiges Anzugsdrehmoment: 0,15 Nm

Öffnen der
Auswerteeinheit

BIS M-6023

Schnittstelleninformationen / Anschlusspläne

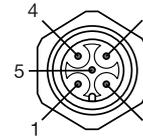
Schnittstellen für Auswerteeinheit BIS M-6023



Der Erdanschluss ist je nach Anlage (Potenzialausgleich) direkt oder über eine RC-Kombination an Erde zu legen.

Beim Anschluss der Bus-Leitungen ist darauf zu achten, dass der Schirm eine einwandfreie Verbindung zum Steckergehäuse hat.

X1, Stromversorgung und digitaler Eingang



| Pin | Funktion |
|-----|----------|
| 1 | +Vs |
| 2 | -IN |
| 3 | -Vs |
| 4 | +IN |
| 5 | n.c. |

n.c. = nichts anschließen

X2, DeviceNet-Ausgang



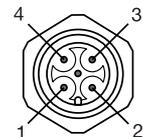
| Pin | Funktion |
|-----|----------|
| 1 | Drain |
| 2 | +VS1 |
| 3 | -VS1 |
| 4 | CAN_H |
| 5 | CAN_L |

X3, DeviceNet-Eingang



| Pin | Funktion |
|-----|----------|
| 1 | Drain |
| 2 | +VS1 |
| 3 | -VS1 |
| 4 | CAN_H |
| 5 | CAN_L |

X4, Service-Schnittstelle



| Pin | Funktion |
|-----|----------|
| 1 | |
| 2 | TxD |
| 3 | GND |
| 4 | RxD |

BIS M-6023 Wechseln des EEPROM

EEPROM in der Auswerteeinheit BIS M-6023 wechseln

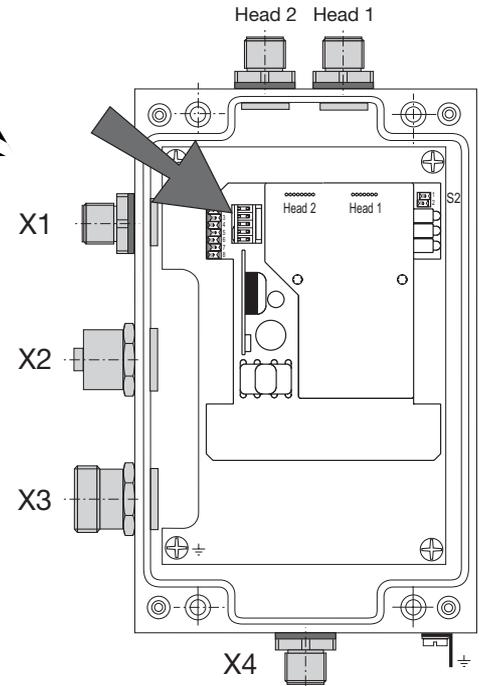


Sorgen Sie vor dem Öffnen dafür, dass das Gerät spannungsfrei geschaltet ist.

Um das EEPROM beim Wechseln nicht zu beschädigen, beachten Sie bitte die Regeln für den Umgang mit elektrostatisch gefährdeten Bauelementen.

Der Wechsel des EEPROM geschieht durch Aus- und Einstecken.

Um das EEPROM zu wechseln, ist die Auswerteeinheit zu öffnen.

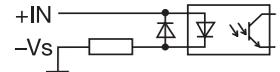


Lage des EEPROM

BIS M-6023

Technische Daten

| | | |
|---------------------------------|---|--|
| Abmessungen, Gewicht | Gehäuse | Metall |
| | Abmessungen | 190 x 120 x 60 mm |
| | Gewicht | 820 g |
| Betriebsbedingungen | Umgebungstemperatur | 0 °C bis +60 °C |
| Schutzart | Schutzart | IP 65 (in angeschlossenem Zustand) |
| Anschlussart | Einbaustecker X1 für V_s, +IN | 5-polig (Stift) |
| | Einbaustecker X2 für DeviceNet Ausgang | 5-polig (Buchse) |
| | Einbaubuchse X3 für DeviceNet Eingang | 5-polig (Stift) |
| | Einbaustecker X4 für Service-Schnittstelle | 4-polig (Stift) |
| Elektrische Anschlüsse | Betriebsspannung V_s | DC 24 V ± 20 % |
| | Restwelligkeit | ≤ 10 % |
| | Stromaufnahme | ≤ 600 mA |
| | Digitaler Eingang +IN | über Optokoppler getrennt |
| | Steuerspannung aktiv | 4 V bis 40 V |
| | Steuerspannung inaktiv | 1,5 V bis -40 V |
| | Eingangsstrom bei 24 V | 11 mA |
| | Verzögerungszeit typisch | 5 ms |
| | DeviceNet, Ausgang X2, Eingang X3 | serielle Schnittstelle für DeviceNet-Teilnehmer |
| | Head 1, Head 2, Schreib-/Lesekopf | über 2 x Einbaustecker 8-polig (Buchse) für alle Schreib-/Leseköpfe BIS M-3_ _ mit 8-poligem Stecker (Stift) |
| Service-Schnittstelle X4 | RS 232 | |



BIS M-6023

Technische Daten

Funktionsanzeigen

DeviceNet-Betriebszustände:

MOD / NET STATUS

LED grün / rot

BIS-Betriebszustände:

CT1 Present / operating

LED grün / gelb

CT2 Present / operating

LED grün / gelb



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-6023

Bestellinformationen

Typenschlüssel

BIS M-6023-025-050-03-ST13

Balluff Identifikations-System

Baureihe M Schreib-/Lesesystem

Hardware-Typ

6023 = Metallgehäuse, DeviceNet

Software-Typ

025 = DeviceNet

Schreib-/Lesekopf

050 = mit zwei Steckverbindern für Schreib-/Leseköpfe BIS M-3_ _

Schnittstelle

03 = BUS-Varianten

Kundenanschluss

ST13 = Steckervariante

(1 Rundsteckverbinder für Stromversorgung, 2 Rundsteckverbinder für DeviceNet, 1 Rundstecker für RS 232 Schnittstelle)

BIS M-6023

Bestellinformationen

Zubehör
(optional, nicht im
Lieferumfang)

Das Zubehör zum BIS M-6_ _ _-... finden Sie im Balluff **Industrial Identification** Katalog.
Den Katalog können Sie im Internet unter „www.balluff.de“ herunterladen.

Anhang, ASCII-Tabelle

| Decimal | Hex | Control Code | ASCII | Decimal | Hex | Control Code | ASCII | Decimal | 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 | | + | | | | | | | | | | | | |

Appendix, ASCII-Table

| | | | | |
|--------------------------|----|----|-------|-----|
| Deci- Hex mal Code ASCII | 0 | 00 | Ctl @ | NUL |
| Deci- Hex mal Code ASCII | 1 | 01 | Ctl A | SOH |
| Deci- Hex mal Code ASCII | 2 | 02 | Ctl B | STX |
| Deci- Hex mal Code ASCII | 3 | 03 | Ctl C | ETX |
| Deci- Hex mal Code ASCII | 4 | 04 | Ctl D | EOT |
| Deci- Hex mal Code ASCII | 5 | 05 | Ctl E | ENQ |
| Deci- Hex mal Code ASCII | 6 | 06 | Ctl F | ACK |
| Deci- Hex mal Code ASCII | 7 | 07 | Ctl G | BEL |
| Deci- Hex mal Code ASCII | 8 | 08 | Ctl H | BS |
| Deci- Hex mal Code ASCII | 9 | 09 | Ctl I | HT |
| Deci- Hex mal Code ASCII | 10 | 0A | Ctl J | LF |
| Deci- Hex mal Code ASCII | 11 | 0B | Ctl K | VT |
| Deci- Hex mal Code ASCII | 12 | 0C | Ctl L | FF |
| Deci- Hex mal Code ASCII | 13 | 0D | Ctl M | CR |
| Deci- Hex mal Code ASCII | 14 | 0E | Ctl N | SO |
| Deci- Hex mal Code ASCII | 15 | 0F | Ctl O | SI |
| Deci- Hex mal Code ASCII | 16 | 10 | Ctl P | DLE |
| Deci- Hex mal Code ASCII | 17 | 11 | Ctl Q | DC1 |
| Deci- Hex mal Code ASCII | 18 | 12 | Ctl R | DC2 |
| Deci- Hex mal Code ASCII | 19 | 13 | Ctl S | DC3 |
| Deci- Hex mal Code ASCII | 20 | 14 | Ctl T | DC4 |
| Deci- Hex mal Code ASCII | 21 | 15 | Ctl U | NAK |

| | | | | |
|--------------------------|----|----|-------|-----|
| Deci- Hex mal Code ASCII | 22 | 16 | Ctl V | SYN |
| Deci- Hex mal Code ASCII | 23 | 17 | Ctl W | ETB |
| Deci- Hex mal Code ASCII | 24 | 18 | Ctl X | CAN |
| Deci- Hex mal Code ASCII | 25 | 19 | Ctl Y | EM |
| Deci- Hex mal Code ASCII | 26 | 1A | Ctl Z | SUB |
| Deci- Hex mal Code ASCII | 27 | 1B | Ctl [| ESC |
| Deci- Hex mal Code ASCII | 28 | 1C | Ctl \ | FS |
| Deci- Hex mal Code ASCII | 29 | 1D | Ctl] | GS |
| Deci- Hex mal Code ASCII | 30 | 1E | Ctl ^ | RS |
| Deci- Hex mal Code ASCII | 31 | 1F | Ctl _ | US |
| Deci- Hex mal Code ASCII | 32 | 20 | | SP |
| Deci- Hex mal Code ASCII | 33 | 21 | i | |
| Deci- Hex mal Code ASCII | 34 | 22 | " | |
| Deci- Hex mal Code ASCII | 35 | 23 | # | |
| Deci- Hex mal Code ASCII | 36 | 24 | \$ | |
| Deci- Hex mal Code ASCII | 37 | 25 | % | |
| Deci- Hex mal Code ASCII | 38 | 26 | & | |
| Deci- Hex mal Code ASCII | 39 | 27 | ' | |
| Deci- Hex mal Code ASCII | 40 | 28 | (| |
| Deci- Hex mal Code ASCII | 41 | 29 |) | |
| Deci- Hex mal Code ASCII | 42 | 2A | * | |
| Deci- Hex mal Code ASCII | 43 | 2B | + | |

| | | | | |
|--------------------------|----|----|---|--|
| Deci- Hex mal Code ASCII | 44 | 2C | , | |
| Deci- Hex mal Code ASCII | 45 | 2D | - | |
| Deci- Hex mal Code ASCII | 46 | 2E | . | |
| Deci- Hex mal Code ASCII | 47 | 2F | / | |
| Deci- Hex mal Code ASCII | 48 | 30 | 0 | |
| Deci- Hex mal Code ASCII | 49 | 31 | 1 | |
| Deci- Hex mal Code ASCII | 50 | 32 | 2 | |
| Deci- Hex mal Code ASCII | 51 | 33 | 3 | |
| Deci- Hex mal Code ASCII | 52 | 34 | 4 | |
| Deci- Hex mal Code ASCII | 53 | 35 | 5 | |
| Deci- Hex mal Code ASCII | 54 | 36 | 6 | |
| Deci- Hex mal Code ASCII | 55 | 37 | 7 | |
| Deci- Hex mal Code ASCII | 56 | 38 | 8 | |
| Deci- Hex mal Code ASCII | 57 | 39 | 9 | |
| Deci- Hex mal Code ASCII | 58 | 3A | : | |
| Deci- Hex mal Code ASCII | 59 | 3B | ; | |
| Deci- Hex mal Code ASCII | 60 | 3C | < | |
| Deci- Hex mal Code ASCII | 61 | 3D | = | |
| Deci- Hex mal Code ASCII | 62 | 3E | > | |
| Deci- Hex mal Code ASCII | 63 | 3F | ? | |
| Deci- Hex mal Code ASCII | 64 | 40 | @ | |

| | | | | |
|--------------------------|----|----|---|--|
| Deci- Hex mal Code ASCII | 65 | 41 | A | |
| Deci- Hex mal Code ASCII | 66 | 42 | B | |
| Deci- Hex mal Code ASCII | 67 | 43 | C | |
| Deci- Hex mal Code ASCII | 68 | 44 | D | |
| Deci- Hex mal Code ASCII | 69 | 45 | E | |
| Deci- Hex mal Code ASCII | 70 | 46 | F | |
| Deci- Hex mal Code ASCII | 71 | 47 | G | |
| Deci- Hex mal Code ASCII | 72 | 48 | H | |
| Deci- Hex mal Code ASCII | 73 | 49 | I | |
| Deci- Hex mal Code ASCII | 74 | 4A | J | |
| Deci- Hex mal Code ASCII | 75 | 4B | K | |
| Deci- Hex mal Code ASCII | 76 | 4C | L | |
| Deci- Hex mal Code ASCII | 77 | 4D | M | |
| Deci- Hex mal Code ASCII | 78 | 4E | N | |
| Deci- Hex mal Code ASCII | 79 | 4F | O | |
| Deci- Hex mal Code ASCII | 80 | 50 | P | |
| Deci- Hex mal Code ASCII | 81 | 51 | Q | |
| Deci- Hex mal Code ASCII | 82 | 52 | R | |
| Deci- Hex mal Code ASCII | 83 | 53 | S | |
| Deci- Hex mal Code ASCII | 84 | 54 | T | |
| Deci- Hex mal Code ASCII | 85 | 55 | U | |

| | | | | |
|--------------------------|-----|----|---|--|
| Deci- Hex mal Code ASCII | 86 | 56 | V | |
| Deci- Hex mal Code ASCII | 87 | 57 | W | |
| Deci- Hex mal Code ASCII | 88 | 58 | X | |
| Deci- Hex mal Code ASCII | 89 | 59 | Y | |
| Deci- Hex mal Code ASCII | 90 | 5A | Z | |
| Deci- Hex mal Code ASCII | 91 | 5B | [| |
| Deci- Hex mal Code ASCII | 92 | 5C | \ | |
| Deci- Hex mal Code ASCII | 93 | 5D |] | |
| Deci- Hex mal Code ASCII | 94 | 5E | ^ | |
| Deci- Hex mal Code ASCII | 95 | 5F | _ | |
| Deci- Hex mal Code ASCII | 96 | 60 | ` | |
| Deci- Hex mal Code ASCII | 97 | 61 | a | |
| Deci- Hex mal Code ASCII | 98 | 62 | b | |
| Deci- Hex mal Code ASCII | 99 | 63 | c | |
| Deci- Hex mal Code ASCII | 100 | 64 | d | |
| Deci- Hex mal Code ASCII | 101 | 65 | e | |
| Deci- Hex mal Code ASCII | 102 | 66 | f | |
| Deci- Hex mal Code ASCII | 103 | 67 | g | |
| Deci- Hex mal Code ASCII | 104 | 68 | h | |
| Deci- Hex mal Code ASCII | 105 | 69 | i | |
| Deci- Hex mal Code ASCII | 106 | 6A | j | |

| | | | | |
|--------------------------|-----|----|-----|--|
| Deci- Hex mal Code ASCII | 107 | 6B | k | |
| Deci- Hex mal Code ASCII | 108 | 6C | l | |
| Deci- Hex mal Code ASCII | 109 | 6D | m | |
| Deci- Hex mal Code ASCII | 110 | 6E | n | |
| Deci- Hex mal Code ASCII | 111 | 6F | o | |
| Deci- Hex mal Code ASCII | 112 | 70 | p | |
| Deci- Hex mal Code ASCII | 113 | 71 | q | |
| Deci- Hex mal Code ASCII | 114 | 72 | r | |
| Deci- Hex mal Code ASCII | 115 | 73 | s | |
| Deci- Hex mal Code ASCII | 116 | 74 | t | |
| Deci- Hex mal Code ASCII | 117 | 75 | u | |
| Deci- Hex mal Code ASCII | 118 | 76 | v | |
| Deci- Hex mal Code ASCII | 119 | 77 | w | |
| Deci- Hex mal Code ASCII | 120 | 78 | x | |
| Deci- Hex mal Code ASCII | 121 | 79 | y | |
| Deci- Hex mal Code ASCII | 122 | 7A | z | |
| Deci- Hex mal Code ASCII | 123 | 7B | { | |
| Deci- Hex mal Code ASCII | 124 | 7C | | |
| Deci- Hex mal Code ASCII | 125 | 7D | } | |
| Deci- Hex mal Code ASCII | 126 | 7E | ~ | |
| Deci- Hex mal Code ASCII | 127 | 7F | DEL | |

BIS M-6023 **Ordering Information**

Accessories for the BIS M-6_ _ _-... can be found in the Balluff Industrial Identification catalog. The catalog can be downloaded on the Internet at "www.balluff.de".

Accessory
(optional,
not included)

BIS M-6023 Ordering Information

Ordering Code



DeviceNet messages:
MOD / NET STATUS

BIS status messages:
CT1 Present / operating
CT2 Present / operating

LED green / red

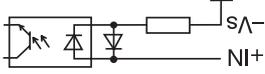
LED green / yellow
LED green / yellow

The CE-Mark is your assurance that our products are in conformance with the EC-Guideline  89/336/EEC (EMC-Guideline)

and the EMC Law. Testing in our EMC Laboratory, which is accredited by the DAtech for Testing of Electromagnetic Compatibility, has confirmed that Balluff products meet the EMC requirements of the Generic Standard EN 61000-6-4 (Emission) and EN 61000-6-2 (Noise Immunity).

BIS M-6023 Technical Data

| Dimensions, Weight | Housing | Operating Conditions | Enclosure Rating | Connections | Electrical Connections |
|---|---|---|--|--|--|
| Weight 820 g Dimensions 190 x 120 x 60 mm Housing Metal | Operating temperature 0 °C to +60 °C Protection class IP 65 (when connected) | Connections Integral connector X1 for V_S, IN 5-pin (male) Integral connector X2 for Devicenet Output 5-pin (female) Integral connector X3 for Devicenet Input 5-pin (male) Integral connector X4 for Service Interface 4-pin (male) | Supply voltage V_S DC 24 V ± 20 % Ripple ≤ 10 % Current draw ≤ 600 mA Optocoupler isolated 4 V to 40 V Control voltage inactive 1.5 V to -40 V 11 mA 5 ms | Digital input +IN Control voltage active Control voltage inactive Input current at 24 V Delay time, typ. Devicenet output X2, input X3 Head 1, head 2, Read/Write head Service interface X4 | Service interface X4 serial interface for Devicenet stations via 2 x connectors 8-pin (female) for all read/write heads BIS M-3 with 8-pin connector (male) RS 232 |



BIS M-6023 Changing the EEPROM

To replace the EEPROM, open up the processor.

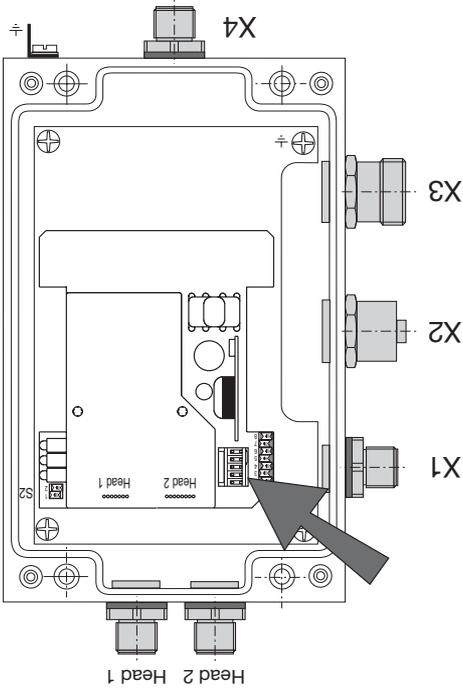
Changing the EEPROM in the BIS M-6023 processor



Be sure before opening that the unit is disconnected from power.
To avoid damaging the EEPROM, please observe the requirements for handling electrostatically sensitive components.



The EEPROM is replaced by unplugging and plugging back into the socket.

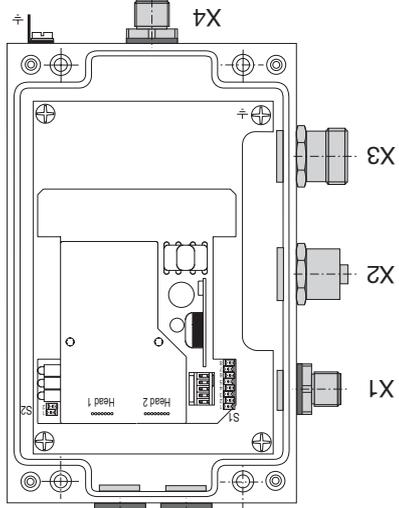


Location of the EEPROM

BIS M-6023 Interface Information / Wiring Diagrams

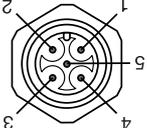
Wiring for the
BIS M-6023
processor

Connection for read/write head 2
Head 2 Head 1
Connection for read/write head 1

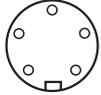


The ground connector should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).
When connecting the bus leads, make sure that the shield has proper connection to connector housing.

X1, supply voltage,
digital input



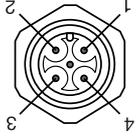
X2, DeviceNet output



X3, DeviceNet input



X4, Service interface



| Pin | Function |
|-----|----------|
| 1 | +Vs |
| 2 | -IN |
| 3 | -Vs |
| 4 | +IN |
| 5 | n.c. |

n.c. = do
not connect

| Pin | Function |
|-----|----------|
| 1 | Drain |
| 2 | +VS1 |
| 3 | -VS1 |
| 4 | CAN_H |
| 5 | CAN_L |

| Pin | Function |
|-----|----------|
| 1 | Drain |
| 2 | +VS1 |
| 3 | -VS1 |
| 4 | CAN_H |
| 5 | CAN_L |

| Pin | Function |
|-----|----------|
| 1 | |
| 2 | TXD |
| 3 | GND |
| 4 | RXD |

english

BALLUFF

BIS M-6023 Opening the Processor

The BIS M-6023 processor must be opened to perform the following steps:

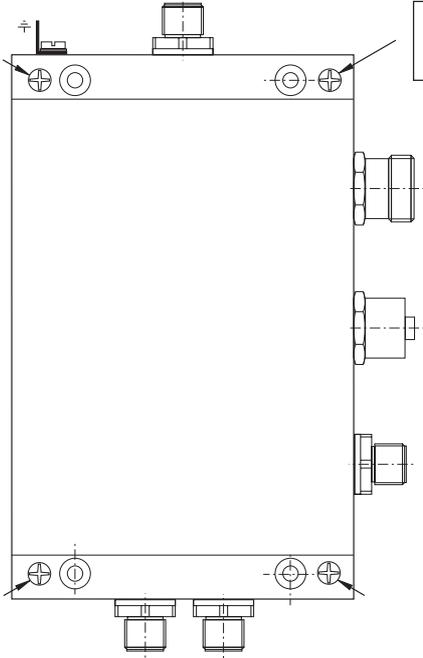
- Set DeviceNet MAC-ID,
- Set baud rate,
- Change EEPROM.



Be sure that the unit is disconnected from power before opening.

Remove the 4 screws on the BIS M-6023 and lift off the cover.
Perform the desired action.

Mounting of the cover (4 screws),
max. permissible tightening torque: 0.15 Nm

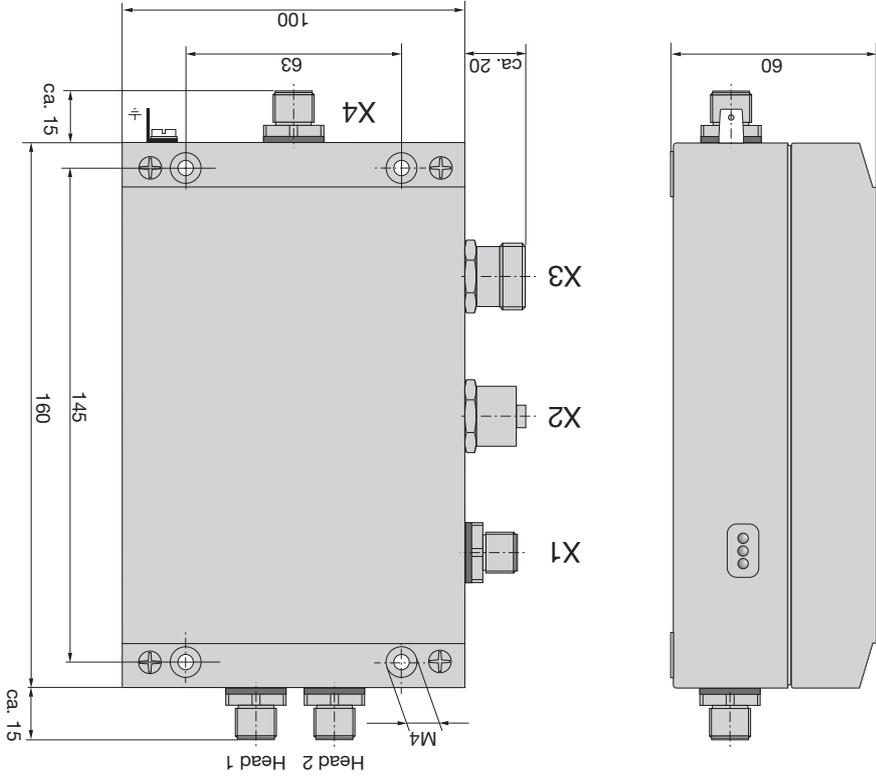


Opening the Processor BIS M-6023

Opening the processor

BIS M-6023 Mounting Processor

Mounting the
BIS M-6023
Processor



The processor is mounted using 4 M4 screws.

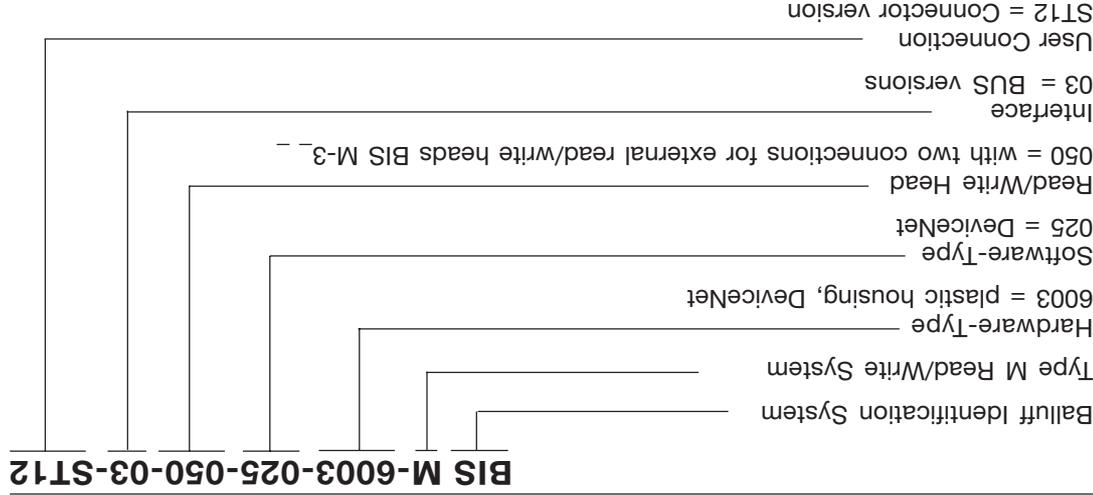
BIS M-6003 Ordering Information

Accessories for the BIS M-6_ _ _-... can be found in the Balluff Industrial Identification catalog. The catalog can be downloaded on the Internet at "www.balluff.de".

Accessory
(optional,
not included)

BIS M-6003 Ordering Information

Ordering Code



DeviceNet messages:
MOD / NET STATUS

LED green / red

BIS status messages:

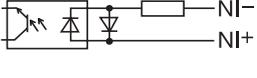
CT1 Present / operating LED green / yellow
CT2 Present / operating LED green / yellow

 The CE-Mark is your assurance that our products are in conformance with the EC-Guideline 89/336/EEC (EMC-Guideline)

and the EMC Law. Testing in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, has confirmed that Balluff products meet the EMC requirements of the Generic Standard EN 61000-6-4 (Emission) and EN 61000-6-2 (Noise Immunity).

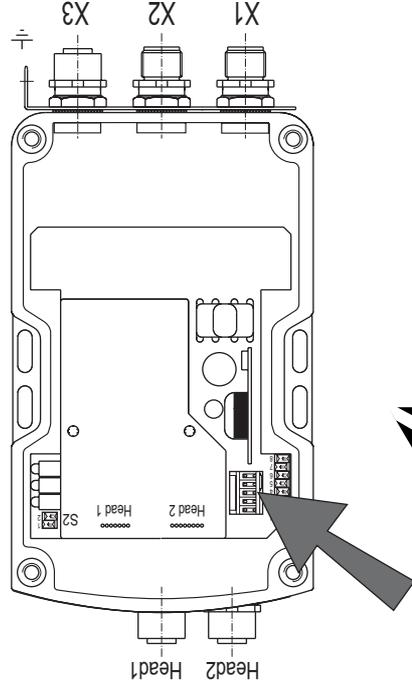
BIS M-6003 Technical Data

| | | |
|-----------------------------------|---|---|
| Dimensions, Weight | Housing | Plastic |
| | Dimensions | ca. 179 x 90 x 45,5 mm |
| | Weight | ca. 500 g |
| Operating Conditions | Ambient temperature | 0 °C to + 60 °C |
| | Enclosure Rating | IP 65 (when connected) |
| Connections | Integral connector X1 for V_s, IN | 5-pin (male) 5-pin (male) 5-pin (female) |
| | Supply voltage V_s, input | DC 24 V ± 20 % Ripple ≤ 10 % Current draw ≤ 600 mA at input X1 |
| Electrical Connections | Digital Input (+IN, -IN) | Optocoupler isolated +IN — 4 V to 40 V -IN — 1,5 V to -40 V |
| | Connections for supply voltage V_s | Control voltage active Control voltage inactive Input current at 24 V Delay time, typ. |
| Electrical Connections | DeviceNet input X2, output X3 | 2 x connectors 8-pin (female) for all read/write heads BIS M-3_ |
| | Read/Write Head | serial interface for DeviceNet stations with 8-pin connector (male) |



BIS M-6003 Changing the EEPROM

To replace the EEPROM, open up the processor as described on □ 78.



Be sure before opening that the unit is disconnected from power.
To avoid damaging the EEPROM, please observe the requirements for handling electrostatically sensitive components.

The EEPROM is replaced by unplugging and plugging back into the socket.

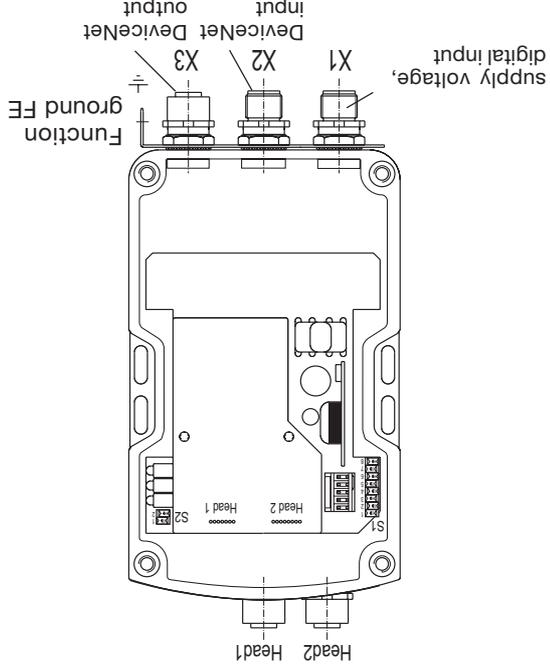
Changing the
EEPROM in the
BIS M-6003
processor



Location of the
EEPROM

BIS M-6003 Interface Information / Wiring Diagrams

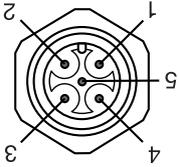
Wiring diagram for BIS M-6003 processor



The ground connector should be connected to earth directly or through a RC combination depending on the system (potential counterpoise).
When connecting the bus leads, make sure that the shield has proper connection to connector housing.

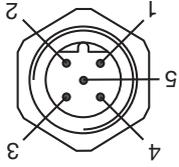
X1, supply voltage, digital input

| Pin | Function |
|-----|----------|
| 1 | +Vs |
| 2 | -IN |
| 3 | -Vs |
| 4 | +IN |
| 5 | |



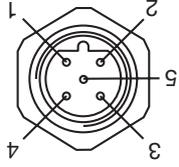
X2, DeviceNet input (male)

| Pin | Function |
|-----|----------|
| 1 | Drain |
| 2 | V+ |
| 3 | V- |
| 4 | CAN H |
| 5 | CAN L |



X3, DeviceNet output (female)

| Pin | Function |
|-----|----------|
| 1 | Drain |
| 2 | V+ |
| 3 | V- |
| 4 | CAN H |
| 5 | CAN L |



english

BALLUFF

BIS M-6003 Opening the Processor

Opening the Processor BIS M-6003

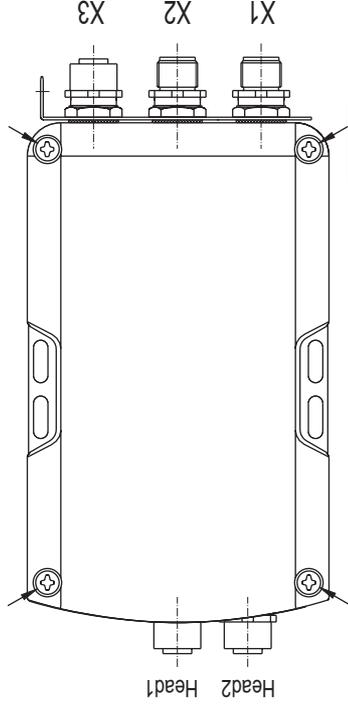


- Set DeviceNet MAC-ID,
- Set baud rate,
- Change EEPROM.

Be sure that the unit is disconnected from power before opening.

Remove the 4 screws on the BIS M-6003 and lift off the cover.
Perform the desired action.

Mounting of the cover (4 screws),
max. permissible tightening torque: 0.15 Nm

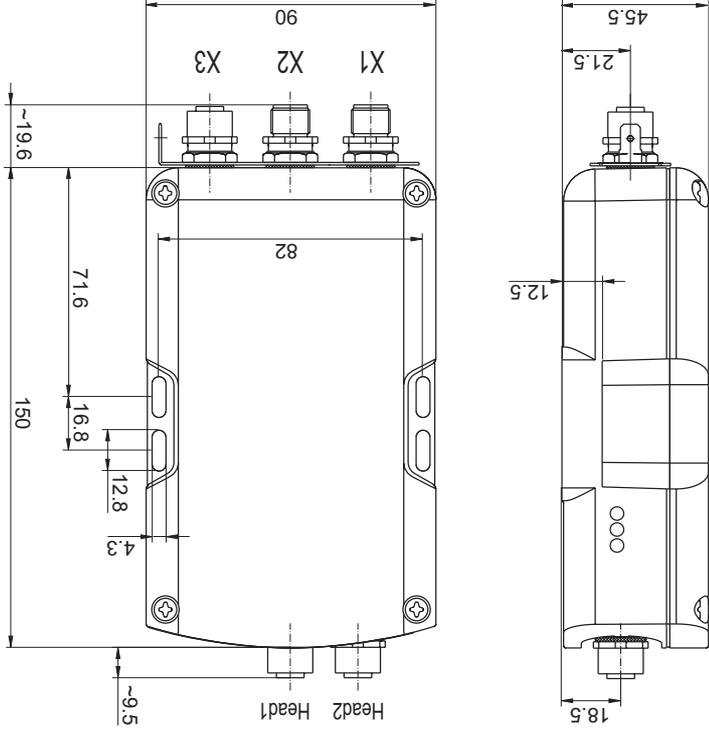


The BIS M-6003 processor must be opened to perform the following steps:

BIS M-6003 Mounting the Processor

Mounting the
BIS M-6003
processor

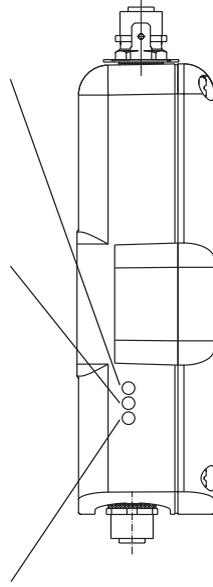
*BIS M-6003
dimensions*



The processor is attached using 4 M4 screws.

LED Display

Function displays on BIS M-60_3



The BIS M-60_3 uses the three side-mounted LED's to indicate important conditions of the identification system.

| Status | LED | Meaning |
|------------------|-----|---------------------|
| MOD / NET STATUS | off | Device is not ready |

| | | |
|---------------|-------|--|
| green flashes | green | - Device does not have power |
| | | - Device is operating in normal mode, the connection is not opened on the Master |
| red flashes | red | Device is operating in normal mode, the connection is opened on the Master |
| | | Correctable error and/or one or more I/O connections is in Time-Out status |
| red | red | Non-correctable error. Device cannot open communication on the bus. |

| | | |
|-------------------------|-------|---|
| CT1 Present / operating | green | Data carrier read/write-ready at read/write head 1. |
| | | Read/write command at read/write head 1 in process. |
| yellow flashes | off | Cable break to read/write head or not connected. |
| | | No data carrier in read/write range of read/write head 1. |

| | | |
|-------------------------|-------|---|
| CT2 Present / operating | green | Data carrier read/write-ready at read/write head 2. |
| | | Read/write command at read/write head 2 in process. |
| yellow flashes | off | Cable break to read/write head or not connected. |
| | | No data carrier in read/write range of read/write head 2. |

If all three LED's are synchronously flashing, it means a hardware error. Return the unit to the factory.

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 |



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

Function Description

Mode 2: Examples for protocol sequence

Example No. 4

Write job on Head 1 with parameter *Input length 1* = 12 Byte, *Output length 1* = 8 Byte with write error (data carrier type BIS M-10⁻01/L):

Host:

1.) Process subaddresses of the output buffer in

| | | |
|------------------------|------------|------------------------|
| 01Hex | Enter data | |
| ... | Enter data | |
| <i>Output length 1</i> | Enter data | |
| 00Hex | | RW-Bit = 1, AV-Bit = 1 |

the order shown:

BIS M-60⁻3 Identification System:

2.) Process subaddresses of the input buffer in the order shown:

| | |
|-------|------------|
| 00Hex | AA-Bit = 1 |
|-------|------------|

Process subaddress of the output buffer:

| | |
|------------------------|-----------|
| 01Hex | Copy data |
| ... | Copy data |
| <i>Output length 1</i> | Copy data |

Process subaddress of the input buffer:

| | |
|-------|------------|
| 00Hex | AF-Bit = 1 |
|-------|------------|

4.) Process subaddresses of the input buffer:

| | |
|-------|------------------------|
| 00Hex | AA-Bit = 0, AF-Bit = 0 |
|-------|------------------------|

Example No. 5

Read/write head deactivation:

In normal operation both heads are active. If the installation is less than ideal, there may be mutual interference between the heads. In this case the unused head should be turned off to prevent interference.

Host:

1.) Process subaddresses of the output buffer:

| | |
|-------------|------------|
| 00Hex/07Hex | Set KA-Bit |
|-------------|------------|

Resetting the KA bit turns the read/write head back on. It may take up to a second off takes much less time.

Function Description Mode 2: Examples for protocol sequence

Example No. 3

Write job on Head 1 with parameter *Input length 1* = 12 Byte, *Output length 1* = 8 Byte
(data carrier type BIS M-10_{-01/L}):

Host:

1.) Process subaddresses of the output buffer in the order shown:

| | | |
|------------------------|------------|------------------------|
| 01 _{Hex} | Enter data | 00 _{Hex} |
| ... | Enter data | |
| <i>Output length 1</i> | Enter data | 00 _{Hex} |
| | | RW-Bit = 1, AV-Bit = 1 |

2.) Process subaddresses of the input buffer in the order shown:

| | | |
|------------------------|------------|--|
| 00 _{Hex} | AA-Bit = 1 | Process subaddress of the output buffer: |
| 01 _{Hex} | Copy data | |
| ... | Copy data | |
| <i>Output length 1</i> | Copy data | Process subaddress of the input buffer: |
| | | AE-Bit = 1 |

3.) Process subaddresses of the output buffer:

| | |
|-------------------|------------|
| 00 _{Hex} | AV-Bit = 0 |
|-------------------|------------|

4.) Process subaddresses of the input buffer:

| | |
|-------------------|------------------------|
| 00 _{Hex} | AA-Bit = 0, AE-Bit = 0 |
|-------------------|------------------------|

BIS M-60₋₃ Identification System:

Function Description Mode 2: Examples for protocol sequence

Example No. 2

Read job on Head 1 with parameter *Input length 1* = 12 Byte, *Output length 1* = 8 Byte with read error (data carrier type BIS M-10_{-01/L}):

Host:

1.) Process subaddresses of the output buffer in the order shown:

| | |
|-------------------|------------------------|
| 00 ^{Hex} | RW-Bit = 0, AV-Bit = 1 |
|-------------------|------------------------|

2.) Process subaddresses of the input buffer in the order shown:

| | |
|-----------------------|------------------------|
| 00 ^{Hex} | AA-Bit = 1, AF-Bit = 1 |
| 01 ^{Hex} | invalid |
| ... | invalid |
| <i>Input length 1</i> | invalid |

3.) Process subaddresses of the output buffer:

| | |
|-------------------|------------|
| 00 ^{Hex} | AV-Bit = 0 |
|-------------------|------------|

4.) Process subaddresses of the input buffer:

| | |
|-------------------|------------------------|
| 00 ^{Hex} | AA-Bit = 0, AF-Bit = 0 |
|-------------------|------------------------|

BIS M-60_3 Identification System:

Function Description Mode 2: Examples for protocol sequence

Example No. 1

Read job on Head 1 with Parameter *Input length 1 = 12 Byte, Output length 1 = 8 Byte* (data carrier type BIS M-10_{-01/L}):

Host:

1.) Process subaddresses of the output buffer in the order shown:

| | |
|-------------------|------------------------|
| 00 _{Hex} | RW-Bit = 0, AV-Bit = 1 |
|-------------------|------------------------|

2.) Process subaddresses of the input buffer in the order shown:

| | | |
|-----------------------|------------------------|------------|
| 00 _{Hex} | AA-Bit = 1, AE-Bit = 1 | Enter data |
| 01 _{Hex} | Enter data | Enter data |
| ... | Enter data | Enter data |
| <i>Input length 1</i> | Enter data | |

3.) Process subaddresses of the input buffer:

| | |
|-----------------------|-----------|
| 01 _{Hex} | Copy data |
| ... | Copy data |
| <i>Input length 1</i> | Copy data |

Process subaddress of the output buffer:

| | |
|-------------------|------------|
| 00 _{Hex} | AV-Bit = 0 |
|-------------------|------------|

4.) Process subaddresses of the input buffer:

| | |
|-------------------|------------------------|
| 00 _{Hex} | AA-Bit = 0, AE-Bit = 0 |
|-------------------|------------------------|

BIS M-60₃ Identification System:

Function Description

Mode 2: Processing data carriers

Reading and writing in dynamic mode

In normal operation a read/write job is rejected by the BIS M-60_3 processor by setting the AF bit and an error number if there is no data carrier in the active zone of the read/write head. If dynamic mode is configured, the processor accepts the read/write job and stores it. When a data carrier is recognized, the stored job is carried out.

CRC initialization

CRC initialization is not possible in Mode 2.

Function Description Mode 2: Processing data carriers

Codetag Present (CP-Bit)



As soon as the data carrier enters the active one of the read/write head, the processor indicates this by setting the CP bit (Codetag Present).

To accelerate the reading of small amounts of data, the ID system makes the first bytes of the data carrier available in the input buffer of the respective read/write head as soon as the tag is detected. The number of bytes sent corresponds to the parameter Input length.

If the parameter "Output data-carrier model and serial number" is set, the data-carrier model and unique serial number are output instead of the read data.

If this function is activated, no read data are output at CT Present.



The data are only valid after the rising edge of the CP bit in the bit header of the input buffer. They remain valid until the falling edge of the CP bit, or until the controller issues a new job.

Start address for Auto-Read

If the Auto-Read function is activated, the data are read starting with a specified start address as soon as the data carrier is recognized. The rising edge of the CP bit is used to provide these data in the input buffer. The start address is determined for each head by the parameters DP1_Start_Address (10) and DP2_Start_Address (11). The start addresses may be different. The number of bytes read is determined by the selected size of the parameter Input length 1 (for Head 1) or Input length 2 (for Head 2).

To carry out a read or write job, the data carrier must be located in the active zone of the read/write head.

A read/write job has the following sequence (see examples on [p. 71ff](#)):

1. The host sends to the output buffer:
 - the RW-Bit for selecting whether to carry out a read (RW = 0) or a write (RW bit = 1) command,
 - the write data, if a write job,
 - sets the AV-Bit to high, to tell the processor there is a new job.
2. The processor:
 - accepts the job and for a write job copies the data from the output buffer, sets the AA-Bit (job was accepted) in the input buffer,
 - carries out the job using the parametered values for start address and number of bytes on the data carrier,
 - sends the AE bit (job correctly finished) or the AF bit (job finished with error) to the input buffer,
 - sends the data in the case of a read job.

Function Description Mode 2: Input buffer, configuration and explanation

| Sub- address | Meaning | Function Description |
|--------------------------|--|--|
| 01 ^{Hex} | Error code (continued) | |
| 07 ^{Hex} | AV bit is set but the command designator is missing or invalid. | |
| 09 ^{Hex} | Cable break to select read/write head, or head not connected. | |
| 0D ^{Hex} | Faulty communication with the data carrier. | |
| 20 ^{Hex} | Addressing of the read/write job lies outside the memory range of the data carrier. | |
| 21 ^{Hex} | Function invoked which is not possible for the data carrier currently in front of the read/write head. | |
| or: | | Data which was read from the data carrier. |
| 02 ^{Hex} | Data | Data which was read from the data carrier. |
| ... | Data | Data which was read from the data carrier. |
| | Last Byte: | |
| | <i>Input length 1</i> for Head 1 | <i>Input length 1</i> for Head 1 |
| | <i>Input length 2</i> for Head 2 | <i>Input length 2</i> for Head 2 |
| | Data | Data which was read from the data carrier. |

Please note the basic procedure on page 29 and the examples on pages 71...74.

**Description of
Input Buffer**
(continued)

Function Description Mode 2: Input buffer, configuration and explanation

| Sub- address | Bit Meaning | Function Description |
|-----------------|-------------------|--|
| 00Hex | IN (continued) | If the parameter "DigIN_Mode" is 1, this bit indicates the state of the Input. |
| AF | Command Error | The command was incorrectly processed or aborted. |
| AE | Command end | The command was finished without error. |
| AA | Command start | The command was recognized and started. |
| CP | Codetag Present | Data carrier present within the active zone of the read/write head. |

| | | |
|-------|--------------|---|
| 01Hex | Error code | Error number is entered if command was incorrectly processed or aborted. Only valid with AF bit! |
| 00Hex | No error. | Reading or writing not possible because no data carrier is present in the active zone of a read/write head. |
| 02Hex | Read error. | Data carrier was removed from the active zone of the read/write head while it was being read. |
| 03Hex | Write error. | Data carrier was removed from the active zone of the read/write head while it was being written. |
| 04Hex | Write error. | Data carrier was removed from the active zone of the read/write head while it was being written. |
| 05Hex | Write error. | Data carrier was removed from the active zone of the read/write head while it was being written. |

(continued on next page)

Description of
Input Buffer
(continued)

Please note the
basic procedure on
pages 29 and the
examples on pages
71...74.

Function Description Mode 2: Input buffer, configuration and explanation

Configuration of the
Input Buffer for One
Read/Write Head
(Example shown for
Head 1)

Description of
Input Buffer

Please note the
basic procedure on
pages 29 and the
examples on pages
71...74.

| Bit-No. | Subaddress | 00Hex = Bit header | BB | HF | IN | AF | AE | AA | CP | Bit name | |
|---------|------------|----------------------------------|---------------|----|----|----|----|----|----|----------|--|
| 7 | | | | | | | | | | | |
| 6 | | | | | | | | | | | |
| 5 | | | | | | | | | | | |
| 4 | | | | | | | | | | | |
| 3 | | | | | | | | | | | |
| 2 | | | | | | | | | | | |
| 1 | | | | | | | | | | | |
| 0 | | | | | | | | | | | |
| | | 01Hex | Error code or | | | | | | | Data | |
| | | 02Hex | Data | | | | | | | | |
| | | 03Hex | Data | | | | | | | | |
| | | 04Hex | Data | | | | | | | | |
| | | 05Hex | Data | | | | | | | | |
| | | 06Hex | Data | | | | | | | | |
| | | ... | Data | | | | | | | | |
| | | Last Byte: <i>Input length 1</i> | | | | | | | | | |

| Sub- address | Bit Name | Meaning | Function Description |
|-----------------|-------------|------------|--|
| 00Hex | BB | Ready | The BIS Identification System is in the Ready state. |
| | HF | Head Error | Cable break from read/write head or no read/write head connected. |

(continued on next page)

Function Description Mode 2: Output buffer, configuration and explanation

| Sub- address | Bit Meaning | Function Description |
|--------------------------|------------------------------|---|
| 00 _{Hex} | Bit Header AV (continued) | Signals the identification system that a command for the respective read/write head is present. |

| Sub- address | Meaning | Function Description |
|--------------------------|------------------------|---------------------------------|
| 01 _{Hex} | Data | for writing to the data carrier |
| ... | Data | for writing to the data carrier |
| Last Byte: | | |
| | <i>Output length 1</i> | for Head 1 |
| | <i>Output length 2</i> | for Head 2 |
| | Data | for writing to the data carrier |

Description of
Output Buffer
(continued)

Please note the
basic procedure on
pages 29 and the
examples on pages
71...74.

Function Description Mode 2: Output buffer, configuration and explanation

Configuration of the
Output Buffer for
One Read/Write
Head
(Example shown for
Head 1)

| Bit-No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|------|---|----|---|---|----|----|----|
| Subaddress | | | | | | | | |
| 00hex = Bit header | | | KA | | | GR | RW | AV |
| 01hex | Data | | | | | | | |
| 02hex | Data | | | | | | | |
| 03hex | Data | | | | | | | |
| 04hex | Data | | | | | | | |
| 05hex | Data | | | | | | | |
| 06hex | Data | | | | | | | |
| ... | Data | | | | | | | |
| Last Byte: | Data | | | | | | | |
| Output length 1 | Data | | | | | | | |

Bit name

Description of
Output Buffer

| Sub- address | Bit Name | Meaning | Function Description |
|-----------------|-------------|---------------------|--|
| 00Hex | KA | Head function | Turn read/write head on/off as needed. Active = 0 Read/write head is on. Inactive = 1 Read/write head is off. |
| | GR | Ground state | Causes the BIS system to go to the ground state for the respective read/write head. Any pending command is cancelled. |
| | RW | Read-/ write job | Signals to the BIS system whether a read or write job should be carried out. Carry out read job Carry out write job |
| | | | (continued next <input type="checkbox"/>) |

Please note the
basic procedure on
pages 29 and the
examples on pages
71...74.

Example No. 9

Put the relevant read/write head into ground state:

Both read/write heads can be independently set to the ground state.

Host: BIS M-60_3 Identification System:

1.) Process subaddresses of the output buffer:

| | |
|-------|------------|
| 00Hex | Set GR-Bit |
|-------|------------|

2.) Go to ground state;

Process subaddresses of the input buffer:

| | |
|-------|--------------|
| 00Hex | Reset BB-Bit |
|-------|--------------|

3.) Process subaddresses of the output buffer:

| | |
|-------|--------------|
| 00Hex | Reset GR-Bit |
|-------|--------------|

4.) Process subaddresses of the input buffer:

| | |
|-------|------------|
| 00Hex | Set BB-Bit |
|-------|------------|

Example No. 10

Read/write head deactivation:

In normal operation both heads are active. If the installation is less than ideal, there may be mutual interference between the heads. In this case the unused head should be turned off to prevent interference.

Host:

1.) Process subaddresses of the output buffer:

| | |
|-------------|------------|
| 00Hex/07Hex | Set KA-Bit |
|-------------|------------|

Resetting the KA bit turns the read/write head back on. It may take up to a second to reactivate the head, whereas turning it off takes much less time.

Function Description

Mode 1: Examples for protocol sequence

Write data carrier using Program No. 1 (data carrier type BIS M-10⁻-01/L):

BIS M-60⁻3 Identification System:

2) Process subaddresses of the input buffer in the order shown:

| | |
|-------------------|---------------------------|
| 00 ^{Hex} | Set AA-Bit, invert TO-Bit |
|-------------------|---------------------------|

4) Process subaddresses of the output buffer:

| | |
|------------------------|-------------------------|
| 01...07 ^{Hex} | Copy first 7 data bytes |
|------------------------|-------------------------|

Process subaddresses of the input buffer:

| | |
|-------------------|---------------|
| 00 ^{Hex} | Invert TO-Bit |
|-------------------|---------------|

1) Process subaddresses of the output buffer in the order shown:

| | |
|-------------------|--------------------------------------|
| 01 ^{Hex} | Command designator 22 ^{Hex} |
| 02 ^{Hex} | Program number 01 ^{Hex} |
| 00 ^{Hex} | Set AV-Bit |

3) Process subaddresses of the output buffer:

| | |
|------------------------|-----------------------------|
| 01...07 ^{Hex} | Enter first 7 bytes to data |
| 00 ^{Hex} | Invert TI-Bit |

... A total of 27 bytes of data are exchanged.

For the remainder of the procedure, see Example 5 on p. 56.

Dynamic mode is turned off while the Mixed Data Access program is being run.



Example No. 8
Use Mixed Data
Access program
For parametering
with 8-byte buffer
size!

Function Description

Mode 1: Examples for protocol sequence

Read data carrier using Program No. 1 (data carrier type BIS M-10_{-01/L}):

BIS M-60_3 Identification System:

2) Process subaddresses of the input buffer in the order shown:

| | |
|------------|-----------------------------|
| 00Hex | Set AA-Bit |
| 01...07Hex | Enter first 7 bytes of data |
| 00Hex | Set AE-Bit |

4) Process subaddresses of the output buffer:

| | |
|------------|-------------------------------|
| 01...07Hex | Enter the second 7 data bytes |
| 00Hex | Invert TO-Bit |

1) Process subaddresses of the output buffer in the order shown:

| | |
|-------|--------------------------|
| 01Hex | Command designator 21Hex |
| 02Hex | Program number 01Hex |
| 00Hex | Set AV-Bit |

3) Process subaddresses of the input buffer:

| | |
|------------|-------------------------|
| 01...07Hex | Copy first 7 data bytes |
| 00Hex | Invert TI-Bit |

... A total of 27 bytes of data are exchanged.
For the remainder of the procedure, see Example 2 on □ 52.



Dynamic mode is turned off while the Mixed Data Access program is being run.

Example No. 7
Use Mixed Data Access program
For parametering with 8-byte buffer size!

Function Description

Mode 1: Examples for protocol sequence

Example No. 6
Store Mixed Data
Access program
 (continued)

For parametering
with 8-byte buffer
size!



Host:

9.) Process subaddresses of the output buffer:

| | | |
|-------------------|---------------|-------------------|
| 01Hex/02Hex | (not used) | FFHex/FFHex |
| 03Hex/04Hex | (not used) | FFHex/FFHex |
| 05Hex/06Hex/07Hex | (not used) | FFHex/FFHex/FFHex |
| 00Hex | Invert T1-Bit | |

11.) Process subaddresses of the output buffer:

| | |
|-------|--------------|
| 00Hex | Reset AV-Bit |
|-------|--------------|

BIS M-60_3 Identification System:

10.) Process subaddresses of the input buffer:

| | |
|-------|------------|
| 00Hex | Set AE-Bit |
|-------|------------|

12.) Process subaddresses of the input buffer:

| | |
|-------|-------------------------|
| 00Hex | Reset AA-Bit and AE-Bit |
|-------|-------------------------|

We recommend that you carefully document which parameters are used for start addresses and number of bytes for writing/reading the desired data records.

The data are sequenced in the exact order specified in the program.

Function Description

Mode 1: Examples for protocol sequence

Host:

BIS M-60_3 Identification System:

Example No. 6
Store Mixed Data
Access program
(continued)

For parametering
with 8-byte buffer
size!

5.) Process subaddresses of the output buffer:

| | | |
|-------|---|---------------------|
| 01hex | 2. No. of bytes (High Byte) 00hex | 00hex |
| 02hex | 3. Start address (Low Byte) 70hex (High Byte) 00hex | 07hex |
| 03hex | 3. Start address (Low Byte) 70hex (High Byte) 00hex | 06hex |
| 04hex | 3. No. of bytes (Low Byte) 11hex (High Byte) 00hex | Invert TI-Bit |
| 05hex | 3. No. of bytes (Low Byte) 11hex (High Byte) 00hex | Terminator FFhex |
| 06hex | Terminator FFhex | 00hex |

7.) Process subaddresses of the output buffer:

| | | |
|-------------------|---------------------------------|---------------|
| 01hex/02hex | (not used) FFhex/FFhex | 00hex |
| 03hex/04hex | (not used) FFhex/FFhex | Invert TI-Bit |
| 05hex/06hex/07hex | (not used) FFhex/FFhex/FFhex | |

6.) Process subaddresses of the input buffer:

| | |
|-------|---------------|
| 00hex | Invert TO-Bit |
|-------|---------------|

8.) Process subaddresses of the input buffer:

| | |
|-------|---------------|
| 00hex | Invert TO-Bit |
|-------|---------------|

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

Continued on next

Function Description Mode 1: Examples for protocol sequence

Example No. 6
Store Mixed Data
Access program
For parametering
with 8-byte buffer
size!

Storing a program for reading out 3 data records:

| | | | | |
|-----------------|---------------|-----|-----------------|----|
| 1st data record | Start address | 5 | Number of bytes | 7 |
| 2nd data record | Start address | 75 | Number of bytes | 3 |
| 3rd data record | Start address | 112 | Number of bytes | 17 |

Total number of bytes exchanged in the operation:

27 bytes

All 104 bytes are written for the programming.

Host: BIS M-60_3 Identification System:

1.) Process subaddresses of the output buffer in the order shown:

| | |
|-------|--------------------------|
| 01Hex | Command designator 0bHex |
| 02Hex | Program number 01Hex |
| 00Hex | Set AV-Bit |

3.) Process subaddresses of the output buffer:

| | |
|-------|-----------------------------------|
| 01Hex | 1. Start address (Low Byte) 05Hex |
| 02Hex | (High Byte) 00Hex |
| 03Hex | 1. No. of bytes (Low Byte) 07Hex |
| 04Hex | (High Byte) 00Hex |
| 05Hex | 2. Start address (Low Byte) 4BHex |
| 06Hex | (High Byte) 00Hex |
| 07Hex | 2. No. of bytes (Low Byte) 03Hex |
| 00Hex | Invert T1-Bit |

4.) Process subaddresses of the input buffer:

| | |
|-------|-----------------------------------|
| 01Hex | 1. Start address (Low Byte) 05Hex |
| 02Hex | (High Byte) 00Hex |
| 03Hex | 1. No. of bytes (Low Byte) 07Hex |
| 04Hex | (High Byte) 00Hex |
| 05Hex | 2. Start address (Low Byte) 4BHex |
| 06Hex | (High Byte) 00Hex |
| 07Hex | 2. No. of bytes (Low Byte) 03Hex |
| 00Hex | Invert T1-Bit |

Continued on next

Function Description

Mode 1: Examples for protocol sequence

Example No. 5

For parametering with 8-byte buffer size!

english

BALLUFF 56

Host:

1) Process subaddresses of the output buffer in the order shown:

| | |
|----------|-----------------------------|
| 01Hex | Command designator 02Hex |
| 02/03Hex | Start address 14Hex / 00Hex |
| 04/05Hex | No. of bytes 10Hex / 00Hex |
| 00Hex | Set AV-Bit |

3) Process subaddresses of the output buffer:

| | |
|-----------|-----------------------------|
| 01..07Hex | Enter first 7 bytes to data |
| 00Hex | Invert TI-Bit |

5) Process subaddresses of the output buffer:

| | |
|-----------|------------------------------|
| 01..07Hex | Enter second 7 bytes of data |
| 00Hex | Invert TI-Bit |

7) Process subaddresses of the output buffer:

| | |
|-----------|-------------------------------------|
| 01..02Hex | Enter the remaining 2 bytes of data |
| 00Hex | Invert TI-Bit |

9) Process subaddresses of the output buffer:

| | |
|-------|--------------|
| 00Hex | Reset AV-Bit |
|-------|--------------|

Write 16 bytes starting at data carrier address 20 (data carrier type BIS M-10₋01/L):

BIS M-60_3 Identification System:

2) Process subaddresses of the input buffer in the order shown:

| | |
|-------|---------------------------|
| 00Hex | Set AA-Bit, invert TO-Bit |
|-------|---------------------------|

4) Process subaddresses of the output buffer:

| | |
|-----------|-------------------------|
| 01..07Hex | Copy first 7 data bytes |
| 00Hex | Invert TO-Bit |

6) Process subaddresses of the output buffer:

| | |
|-----------|--------------------------|
| 01..07Hex | Copy second 7 data bytes |
| 00Hex | Invert TO-Bit |

8) Process subaddresses of the output buffer:

| | |
|-----------|-----------------------------|
| 01..02Hex | Copy remaining 2 data bytes |
| 00Hex | Set AE-Bit |

10) Process subaddresses of the input buffer:

| | |
|-------|-------------------------|
| 00Hex | Reset AA-Bit and AE-Bit |
|-------|-------------------------|

Function Description

Mode 1: Examples for protocol sequence

Example No. 4

For parametering
with 8-byte buffer
size!

Read 30 bytes starting at data carrier address 10 with read error
(data carrier type BIS M-10_{-01/L}):

Host:

1.) Process subaddresses of the output buffer in
the order shown:

| | |
|-------|-------------------------------|
| 01Hex | Command designator 01Hex |
| 02Hex | Start address Low Byte 0AHex |
| 03Hex | Start address High Byte 00Hex |
| 04Hex | No. of bytes Low Byte 1EHex |
| 05Hex | No. of bytes High Byte 00Hex |
| 00Hex | Set AV-Bit |

3.) Process subaddress of the input buffer:

| | |
|-------|-------------------|
| 01Hex | Copy error number |
| 00Hex | Reset AV-Bit |

Process subaddress of the output buffer:

| | |
|-------|-------------------------|
| 00Hex | Reset AA-Bit and AE-Bit |
|-------|-------------------------|

4.) Process subaddresses of the input buffer:

| | |
|-------|--------------------|
| 00Hex | Set AA-Bit |
| 01Hex | Enter error number |
| 00Hex | Set AF-Bit |

If an error occurs right away:

2.) Process subaddresses of the input buffer in the
order shown:

BIS M-60₋₃ Identification System:

Function Description

Mode 1: Examples for protocol sequence

BIS M-60_3 Identification System:

6.) Process subaddresses of the input buffer:

| | |
|------------|----------------------------------|
| 01...03Hex | Enter the remaining 3 data bytes |
| 00Hex | Invert TO-Bit |
| 00Hex | Set AE-Bit |

8.) Process subaddresses of the input buffer:

| | |
|-------|-------------------------|
| 00Hex | Reset AA-Bit and AE-Bit |
|-------|-------------------------|

Host:

5.) Process subaddresses of the input buffer:

| | |
|------------|--|
| 01...07Hex | Copy second 7 data bytes |
| 00Hex | Process subaddress of the output buffer: |
| 00Hex | Invert TI-Bit |

7.) Process subaddresses of the input buffer:

| | |
|------------|--|
| 01...03Hex | Copy the remaining 3 data bytes |
| 00Hex | Process subaddress of the output buffer: |

Reset AV-Bit

Example No. 3

(continued)

like 2nd example but with simultaneous data transmission

For parametering with 8-byte buffer size!

Function Description

Mode 1: Examples for protocol sequence

Example No. 3
like 2nd example but
with simultaneous
data transmission
For parametering
with 8-byte buffer
size!

Read 17 bytes starting at data carrier address 10, with simultaneous data transmission (data carrier type BIS M-10⁻-01/L):

While the read job is being carried out and as soon as the input buffer is filled, the first data are sent. The AE bit is not set until the "Read" operation is completed by the processor.

The reply "Job End" = AE bit is reliably set no later than before the last data are sent. The exact time depends on the requested data amount, the input buffer size and the timing of the controller. This is indicated in the following by the note *Set AE-Bit* (in italics).

Host: BIS M-60₃ Identification System:

1.) Process subaddresses of the output buffer in the order shown:

| | |
|-------|-------------------------------|
| 01Hex | Command designator 01Hex |
| 02Hex | Start address Low Byte 0AHex |
| 03Hex | Start address High Byte 00Hex |
| 04Hex | No. of bytes Low Byte 1Hex |
| 05Hex | No. of bytes High Byte 00Hex |
| 00Hex | Set AV-Bit |

3.) Process subaddresses of the input buffer:

| | |
|------------|-------------------------|
| 01...07Hex | Copy first 7 data bytes |
| 00Hex | Invert T1-Bit |

2.) Process subaddresses of the input buffer in the order shown:

| | |
|---------------|-----------------------------|
| 00Hex | Set AA-Bit |
| 01Hex...07Hex | Enter first 7 bytes of data |
| 00Hex | Invert TO-Bit |
| 00Hex | Set AE-Bit |

4.) Process subaddresses of the input buffer:

| | |
|------------|------------------------------|
| 01...07Hex | Enter second 7 bytes of data |
| 00Hex | Invert TO-Bit |
| 00Hex | Set AE-Bit |

Continued on next

Function Description

Mode 1: Examples for protocol sequence

Read 17 bytes starting at data carrier address 10 (data carrier type BIS M-10_{-01/L}):

Host:

Example No. 2
For parametering
with 8-byte buffer
size!

1.) Process subaddresses of the output buffer in the order shown:

| | |
|-------|-------------------------------|
| 01hex | Command designator 01hex |
| 02hex | Start address Low Byte 0Ahex |
| 03hex | Start address High Byte 00hex |
| 04hex | No. of bytes Low Byte 11hex |
| 05hex | No. of bytes High Byte 00hex |
| 00hex | Set AV-Bit |

3.) Process subaddresses of the input buffer:

| | |
|------------|-------------------------|
| 01...07hex | Copy first 7 data bytes |
| 00hex | Invert TI-Bit |

Process subaddress of the output buffer:

| | |
|------------|--------------------------|
| 01...07hex | Copy second 7 data bytes |
|------------|--------------------------|

5.) Process subaddresses of the input buffer:

| | |
|-------|---------------|
| 00hex | Invert TI-Bit |
|-------|---------------|

Process subaddress of the output buffer:

| | |
|------------|---------------------------------|
| 01...03hex | Copy the remaining 3 data bytes |
|------------|---------------------------------|

7.) Process subaddresses of the input buffer:

| | |
|-------|--------------|
| 00hex | Reset AV-Bit |
|-------|--------------|

Process subaddress of the output buffer:

2.) Process subaddresses of the input buffer in the order shown:

| | |
|------------|-----------------------------|
| 00hex | Set AA-Bit |
| 01...07hex | Enter first 7 bytes of data |
| 00hex | Set AE-Bit |

BIS M-60₋₃ Identification System:

8.) Process subaddresses of the input buffer:

| | |
|-------|-------------------------|
| 00hex | Reset AA-Bit and AE-Bit |
|-------|-------------------------|

6.) Process subaddresses of the input buffer:

| | |
|------------|----------------------------------|
| 01...03hex | Enter the remaining 3 data bytes |
| 00hex | Invert TO-Bit |

Function Description Mode 1: Examples for protocol sequence

Example No. 1

(continued)

For parametering
with 8-byte buffer
size!

Host:

7.) Process subaddresses of the output buffer:

| | |
|---------------|---------------------------|
| 01Hex...07Hex | Enter the remaining datum |
| 00Hex | Invert T1-Bit |

9.) Process subaddresses of the output buffer:

| | |
|-------|--------------|
| 00Hex | Reset AV-Bit |
|-------|--------------|

BIS M-60_3 Identification System:

8.) Process subaddresses of the output buffer:

| | |
|---------------|--------------------------|
| 01Hex...07Hex | Copy the remaining datum |
|---------------|--------------------------|

Process subadress of the input buffer:

| | |
|-------|------------|
| 00Hex | Set AE-Bit |
|-------|------------|

10.) Process subaddresses of the input buffer:

| | |
|-------|-------------------------|
| 00Hex | Reset AA-Bit and AE-Bit |
|-------|-------------------------|

Function Description

Mode 1: Examples for protocol sequence

Example No. 1

For parametering
with 8-byte buffer
size!

Initializing the data carrier for the CRC₁₆ data checking

The processing of this command is similar to a write command. Start address and number in this example have to correspond to the maximum number of data to be used. In this example the complete memory range of a data carrier with 752 bytes shall be used (BIS M-10_{-01/L}). Because 2 bytes are used for the CRC₁₆ only 658 bytes can be used as data bytes, hence: start address = 0, number of bytes = 658.

BIS M-60_3 Identification System:

2) Process subaddresses of the input buffer in the order shown:

| | |
|-------------------|---------------------------|
| 00 _{Hex} | Set AA-Bit, invert TO-Bit |
|-------------------|---------------------------|

4) Process subaddresses of the output buffer:

| | |
|------------------------|-------------------------|
| 01...07 _{Hex} | Copy first 7 data bytes |
| 00 _{Hex} | Invert TO-Bit |

Process subaddresses of the input buffer:

6) Process subaddresses of the output buffer:

| | |
|------------------------|--------------------------|
| 01...07 _{Hex} | Copy second 7 data bytes |
| 00 _{Hex} | Invert TO-Bit |

Process subaddresses of the input buffer:

Host:

1) Process subaddresses of the output buffer in the order shown:

| | |
|-------------------|--------------------------------------|
| 01 _{Hex} | Command designator 12 _{Hex} |
| 02 _{Hex} | Start address 00 _{Hex} |
| 03 _{Hex} | Start address 00 _{Hex} |
| 04 _{Hex} | No. of bytes 92 _{Hex} |
| 05 _{Hex} | No. of bytes 02 _{Hex} |
| 00 _{Hex} | Set AV-Bit |

3) Process subaddresses of the output buffer:

| | |
|------------------------|-----------------------------|
| 01...07 _{Hex} | Enter first 7 bytes of data |
| 00 _{Hex} | Invert TI-Bit |

5) Process subaddresses of the output buffer:

| | |
|------------------------|------------------------------|
| 01...07 _{Hex} | Enter second 7 bytes of date |
| 00 _{Hex} | Invert TI-Bit |

... To be continued until the complete memory range is written. See next □.

Function Description Mode 1: Processing data carriers

**CRC_16 and
Codetag Present**

**CRC_16 and
memory capacity**

If CRC_16 was parameterized and a data carrier is recognized whose CRC_16 checksum is incorrect, the read data are not output and the CP bit in the input bit header is not set. The CT present LED comes on and the digital output is set - the data carrier can be processed using the initialization command (12_{Hex}).

The checksum is written to the data carrier as a 2-byte datum for each CRC block (corresponds to 16 bytes). 2 bytes are used (lost) for each CRC block, i.e., the CRC block contains only 14 bytes of user data. This means that the actual usable number of bytes is reduced:

| Data carrier | Balluff data carrier model | Memory capacity | Usable bytes using CRC |
|--------------|----------------------------|-----------------|------------------------|
| Mifare | BIS M-1_-01 | 752 bytes | 658 bytes |
| ISO15693 | BIS M-1_-02 | 2000 bytes | 1750 bytes |
| BIS M-1_-03 | 112 bytes | 98 bytes | 224 bytes |
| BIS M-1_-04 | 256 bytes | 196 bytes | 252 bytes |
| BIS M-1_-05 | 224 bytes | 196 bytes | 252 bytes |
| BIS M-1_-06 | 288 bytes | 252 bytes | 252 bytes |
| BIS M-1_-07 | 992 bytes | 868 bytes | 868 bytes |

¹ on request

Function Description Mode 1: Processing data carriers

Read from data carrier, with program Mixed Data Access

The command identifier 21_{hex} can be used to read out the program records stored in the program from the data carrier. The user must document exactly which data are to be read from where and with what number of bytes for the respective program (see example 7 on □ 60)

Write to data carrier, with program Mixed Data Access

The command identifier 22_{hex} can be used to write the program records stored in the program to the data carrier. The user must document exactly which data are to be written from where and with what number of bytes for the respective program (see example 8 on □ 61)

CRC_16 Initialization

To be able to use the CRC_16 check, the data carrier must first be initialized with the command identifier 12_{hex} (see □ 50). The CRC_16 initialization is used like a normal write job. The latter is rejected (with an error message) if the processor recognizes that the data carrier does not contain the correct CRC_16 checksum. Data carriers as shipped from the factory (all data are 0) can immediately be written with CRC-checked data. If CRC_16 data checking is activated, a special error message is output to the interface whenever a CRC_16 error is detected.

If the error message is not caused by a failed write 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.

Function Description

Mode 1: Processing data carriers

Mixed Data Access (continued)

The following shows the structure of a program:

| Program structure | Subaddress | Value | Range |
|---------------------------|-------------------------------------|-------------------|--|
| Command designator | 01 _{hex} | 06 _{hex} | |
| 1. Program record | | | |
| Program number | 02 _{hex} | 01 _{hex} | 01 _{hex} to 0A _{hex} |
| 1st data record: | | | |
| Start address Low Byte | 03 _{hex} | | |
| Start address High Byte | 04 _{hex} | | |
| Number of bytes Low Byte | 05 _{hex} | | |
| Number of bytes High Byte | 06 _{hex} | | |
| 2nd data record: | | | |
| Start address Low Byte | 03 _{hex} | | |
| Start address High Byte | 04 _{hex} | | |
| Number of bytes Low Byte | 05 _{hex} | | |
| Number of bytes High Byte | 06 _{hex} | | |
| ... | | | |
| 25th data record: | | | |
| Start address Low Byte | 03 _{hex} | | |
| Start address High Byte | 04 _{hex} | | |
| Number of bytes Low Byte | 05 _{hex} | | |
| Number of bytes High Byte | 06 _{hex} | | |
| Terminator | FF _{hex} FF _{hex} | | |

To store a second program, repeat this process.

The procedure for writing these settings to the EEPROM is described in the 6th example on [page 57](#)...

Replacing the EEPROM is described on [page 80](#) for BIS M-6003 and on [page 88](#) for BIS M-6023.

Function Description Mode 1: Processing data carriers

Mixed Data Access

Small read/write programs can be stored in the BIS M-60_3 processor's EEPROM. The Mixed Data Access function is useful when the required information is stored on the data carrier at various addresses. This function makes it possible to read out this "mixed", i.e. non-continuously stored data from the data carrier in a single procedure and using just one command.

Up to 10 programs with up to 25 instructions can be stored. Each program instruction contains a "start address" and a "number of bytes" specification. The amount of data for reading may not exceed 2 kB.

Storing a program:

The command identifier 0b_{hex} is used to send the read/write program to the BIS M-60_3 processor. One program per command can be stored. All 25 program records plus an additional 2 bytes with FF_{hex}FF_{hex} as a terminator must always be sent. This means a total of **104 bytes** of information per program must be sent (including the command identifier and program number).



The individual program records must all be continuous. They must be sent one after the other and be terminated with 2 bytes with FF_{hex}FF_{hex} as a terminator. It is recommended that the remaining, unused memory sector be filled with FF_{hex}FF_{hex}.

If an address range is selected twice, the data will also be output twice.

Function Description Mode 1: Processing data carriers

Reading and writing in dynamic mode

In normal operation a read/write job is rejected by the BIS M-60_3 processor by setting the AF bit and an error number if there is no data carrier in the active zone of the read/write head. If dynamic mode is configured, the processor accepts the read/write job and stores it. When a data carrier is recognized, the stored job is carried out.

Reading and writing with simultaneous data transmission

Reading without simultaneous data transmission: In the case of a read job the processor first reads all requested data from the data carrier after receiving the start address and the desired number of bytes, and then sets the AE bit. Then the data read from the data carrier are written to the input buffer. In the case of larger data amounts this is done in blocks, controlled by the handshake with the toggle bits as described on □ 43. **Reading with simultaneous data transmission:** In the case of a read job the processor begins to send data to the input buffer as soon as the first number of bytes (corresponding to the buffer length of the head (-1 byte for the bit header)) has been read from the data carrier and indicates this by inverting the TO bit. As soon as the controller inverts the TI bit, the processor sends the data, which have in the meantime been read, to the input buffer. This is repeated until the processor has read out all the desired data from the data carrier. Now the processor sets the AE bit and outputs the remaining data on the input buffer.

Writing without simultaneous data transmission: In the case of a write job the processor waits until it has received all the data that need to be written from the controller. Only then are the data written to the data carrier as described on □ 43. **Writing with simultaneous data transmission:** In the case of a write job the processor begins to write the data to the data carrier as soon as it has received the first data to be written from the controller's output buffer. Once all the data have been written to the data carrier, the AE bit is set.

Function Description Mode 1: Processing data carriers

Codetag Present (CP-Bit)



As soon as the data carrier enters the active one of the read/write head, the processor indicates this by setting the CP bit (Codetag Present).

To accelerate the reading of small amounts of data, the ID system makes the first bytes of the data carrier available in the input buffer of the respective read/write head as soon as the tag is detected. The number of bytes sent corresponds to the configured buffer size – 1 byte (2 bytes for 2nd bit header).

If the parameter "Output data-carrier model and serial number" is set, the data-carrier model and unique serial number are output instead of the read data.

If this function is activated, no read data are output at CT Present.



The data are only valid after the rising edge of the CP bit in the bit header of the input buffer. They remain valid until the falling edge of the CP bit, or until the controller issues a new job.

Start address for Auto-Read

If the Auto-Read function is activated, the data are read starting with a specified start address as soon as the data carrier is recognized. The rising edge of the CP bit is used to provide these data in the input buffer. The start address is determined for each head by the parameters DP1_Start_Address (10) and DP2_Start_Address (11). The start addresses may be different. The number of bytes read is determined by the selected size of the input buffer which is divided between the two heads.

Function Description

Mode 1: Processing data carriers

Reading and writing

To carry out a read or write job, the data carrier must be located in the active zone of the read/write head.

A read/write job has the following sequence (see examples on [p. 50ff](#)):

1. The host sends to the output buffer:
 - the command designator to subaddress 01_{Hex} ,
 - the start address for reading or writing to subaddress $02_{\text{Hex}}/03_{\text{Hex}}$,
 - the number of bytes for reading or writing to subaddress $04_{\text{Hex}}/05_{\text{Hex}}$,
 - the CT bit in the bit header according to the data carrier type (block size),
 - and sets the AV bit in the bit header to high.

2. The processor:

- takes the request (AA in the bit header of the input buffer to high),
 - begins to transport the data;
 - read = from data carrier to input buffer,
 - write = from output buffer to data carrier.
- Larger data quantities are sent in blocks.
The toggle bits in the two bit headers are used as a kind of handshaking between the host and the BIS M-60_3 processor.

3. The processor has processed the command correctly (AE bit in the bit header of the input buffer). If an error occurred during execution of the command, an error number will be written to subaddress 01_{Hex} of the input buffer and the AF bit in the bit header of the input buffer will be set.

Data-carrier models BIS M-10

Data carrier model

For the processor BIS M-60_3 there are following data carriers available.

| Balluff data carrier model | Manufacture | Name | Memory capacity | Memory type |
|----------------------------|-------------|----------------|-----------------|-------------|
| BIS M-1_-01 | Phillips | Mifare Classic | 752 bytes | EEPROM |

ISO15693

| Balluff data carrier model | Manufacture | Name | Memory capacity | Memory type |
|----------------------------|-------------|-------------|-----------------|-------------|
| BIS M-1_-02 | Fujitsu | MB89R18 | 2000 bytes | FRAM |
| BIS M-1_-03 | Phillips | SL2ICS20 | 112 bytes | EEPROM |
| BIS M-1_-04 | Texas Inst. | TAG-IT Plus | 256 bytes | EEPROM |
| BIS M-1_-05 | Infinion | SRF5V02P | 224 bytes | EEPROM |
| BIS M-1_-06 | EM | EM4135 | 288 bytes | EEPROM |
| BIS M-1_-07 | Infinion | SRF5V10P | 992 bytes | EEPROM |

The data carrier also contains additional memory ranges for configuration and protected data. These areas cannot be processed using the BIS M-60_3 processor.

At CT Present the first user data are read from the data carrier and stored in the DeviceNet input buffer (see □ 44). If the “Output data-carrier model and serial number at CT present” function is enabled, data carrier model is output in byte 1 of the input buffer and then the serial number.



Please refer to □ 13ff.

CT present

Function Description Mode 1: Input buffer, configuration and explanation

| Sub- address | Meaning | Function Description |
|-----------------|---|--|
| 01Hex | Error code | (continued) |
| 07Hex | AV bit is set but the command designator is missing or invalid. or: Number of bytes is 00Hex. | |
| 09Hex | Cable break to select read/write head, or head not connected. | |
| 0CHex | The EEPROM cannot be read/programmed. | |
| 0DHex | Faulty communication with the data carrier. | |
| 0EHex | The CRC of the read data does not coincide with the CRC of the data carrier. | |
| 20Hex | Addressing of the read/write job is outside the memory range of the data carrier. | |
| 21Hex | Function invoked which is not possible for the data carrier currently in front of the read/write head. | |
| or: | Data | Data which was read from the data carrier. |
| 02Hex | Data | Data which was read from the data carrier. |
| ... | Data | Data which was read from the data carrier. |

Please note the
basic procedure on
29 and the
examples on pages
50...62.

Description of
Input Buffer
(continued)

Function Description Mode 1: Input buffer, configuration and explanation

| Sub- address | Bit Meaning | Function Description |
|-------------------|------------------------------|---|
| 00 _{Hex} | (continued) AF Bit Header | The command was incorrectly processed or aborted. |
| AE | Command end | The command was finished without error. |
| AA | Command start | The command was recognized and started. |
| CP | Codetag Present | Data carrier present within the active zone of the read/write head. |

| | | |
|-------------------|--|---|
| 01 _{Hex} | Error code | Error number is entered if command was incorrectly processed or aborted. Only valid with AF bit! |
| 00 _{Hex} | No error. | Reading or writing not possible because no data carrier is present in the active zone of a read/write head. |
| 02 _{Hex} | Read error. | Data carrier was removed from the active zone of the read/write head while it was being read. |
| 03 _{Hex} | Data carrier was removed from the active zone of the read/write head while it was being read. | |
| 04 _{Hex} | Write error. | Data carrier was removed from the active zone of the read/write head while it was being written. |
| 05 _{Hex} | Data carrier was removed from the active zone of the read/write head while it was being written. | (continued on next <input type="checkbox"/>) |

Description of
Input Buffer
(continued)

Please note the
basic procedure on
 29 and the
examples on pages
 50...62.

Function Description Mode 1: Input buffer, configuration and explanation

Configuration of the
input buffer for one
read/write head
(Example shown for
Head 1)

Description of
Input Buffer

Please note the
basic procedure on
pages 29 and the
examples on pages
50...62.

| Bit No. | Subaddress | | | | | | | |
|--------------------|--------------------------|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 00Hex = Bit Header | BB | HF | TO | IN | AF | AE | AA | CP |
| 01Hex | Error Code or Data | | | | | | | |
| 02Hex | Data | | | | | | | |
| 03Hex | Data | | | | | | | |
| 04Hex | Data | | | | | | | |
| 05Hex | Data | | | | | | | |
| 06Hex | Data | | | | | | | |
| ... | Data | | | | | | | |

Sub-
Bit
Meaning
Function Description

| | | | |
|--------------------------|----------------|------------|--|
| 00Hex | BB | Ready | The BIS Identification System is in the Ready state. |
| Bit Header | HF | Head Error | Cable break from read/write head or no read/write head connected. |
| TO | Toggle-Bit Out | Input | for read: BIS has new/additional data ready. for write: BIS is ready to accept new/additional data. If the parameter "DIGN_Mode" is 1, this bit indicates the state of the input. |
| (continued on next page) | | | |

Function Description Mode 1: Output buffer, configuration and explanation

| Sub- address | Meaning | Function Description |
|--------------------------|--|--|
| 05 _{Hex} | No. of bytes (High Byte) Data or: Program data | Number of bytes to read or write beginning with the start address (the High Byte includes the address range from 256 to 1999). for writing to the data carrier |
| 06 _{Hex} | Data or: Program data | for writing to the data carrier |
| ... | Data or: Program data | for writing to the data carrier |

Description of
Output Buffer
(continued)

Please note the
basic procedure on
29 and the
examples on pages
50...62.

Function Description Mode 1: Output buffer, configuration and explanation

| Sub-address | Meaning | Function Description |
|-------------------------|---------------------------|--|
| 02_{Hex} | Start address (Low Byte) | Address at which reading from or writing to the data carrier begins. (The Low Byte includes the address range from 0 to 255). |
| or: | Program No. | Number of the program to be stored in the EEPROM in conjunction with command ID 06 _{Hex} for Mixed Data Access function (values between 01 _{Hex} and 0A _{Hex} are allowed). |
| or: | Program No. | Number of the program stored in the EEPROM for read or write operations in conjunction with command ID 21 _{Hex} or 22 _{Hex} for the Mixed Data Access function. |
| or: | Data | for writing to the data carrier |
| or: | Program data | for writing to the EEPROM. |
| 03_{Hex} | Start address (High Byte) | Address for reading from or writing to the data carrier (the High Byte includes the address range from 256 to 1999). |
| or: | Data | for writing to the data carrier |
| or: | Program data | for writing to the EEPROM. |
| 04_{Hex} | No. of bytes (Low Byte) | Number of bytes to read or write beginning with the start address (the Low Byte includes from 1 to 255 bytes). |
| or: | Data | for writing to the data carrier |
| or: | Program data | for writing to the EEPROM. |

(continued next )

Please note the basic procedure on  29 and the examples on pages  50...62.

Description of Output Buffer (continued)

Function Description Mode 1: Output buffer, configuration and explanation

| Sub- address | Meaning | Function Description |
|-----------------|--|---|
| 01Hex | Command designator | |
| 00Hex | No command present | |
| 01Hex | Read data carrier | |
| 02Hex | Write to data carrier | |
| 06Hex | Store program in the EEPROM for the Mixed Data Access function | |
| 12Hex | Initialize the CRC-16 data check | |
| 21Hex | Read for Mixed Data Access function | (corresponding to the program stored in the EEPROM) |
| 22Hex | Write for Mixed Data Access function | (corresponding to the program stored in the EEPROM) |
| or: | Data | for writing to the data carrier |
| or: | Program data | for writing to the EEPROM. |

(continued next)

Description of
Output Buffer
(continued)

Please note the
basic procedure on
 29 and the
 50...62.
examples on pages

Function Description Mode 1: Output buffer, configuration and explanation

Configuration of the
Output Buffer for
One Read/Write
Head (Example
shown for Head 1)

| Bit No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------------|---|----|----|---|---|----|---|----|
| Subaddress | | | | | | | | |
| 00Hex = Bit Header | | TI | KA | | | GR | | AV |
| 01Hex | Command Designator | | | | | | | |
| 02Hex | Start Address (Low Byte) or Program No. | | | | | | | |
| 03Hex | Start Address (High Byte) | | | | | | | |
| 04Hex | No. of Bytes (Low Byte) | | | | | | | |
| 05Hex | No. of Bytes (High Byte) | | | | | | | |
| 06Hex | Data | | | | | | | |
| ... | Data | | | | | | | |

| Sub-address | Bit Name | Meaning | Function Description |
|-------------|---------------|--|--|
| 00Hex | TI | Toggle-Bit In | Shows during a read action that the controller is ready for additional data. |
| 00Hex | Bit Header | | |
| KA | Head function | Turn read/write head on/off as needed. Active = 0 Read/write head is on. Inactive = 1 Read/write head is off. | |
| GR | Ground state | Causes the BIS system to go to the ground state for the respective read/write head. Any pending command is cancelled. | |
| AV | Command | Signals the identification system that a command for the respective read/write head is present. | |

(continued next ▢)

Please note the
basic procedure on
pages 29 and the
examples on pages
50...62.

Description of
Output Buffer

Function Description

Input and Output Buffers

To query the current status of both heads on the processor, the *assembly object* (class *assembly object*) can be accessed. The controller accesses the *assembly object* by using *explicit message*.



The *explicit-message* connection must have been successfully opened by the controller.

The call *GetAttributeSingle* returns 4 bytes:

| | | | |
|---------------------------------|--------------|--------------|--------------|
| Byte 3 | Byte 2 | Byte 1 | Byte 0 |
| Head 2 | | Head 1 | |
| Input buffer | Input buffer | Input buffer | Input buffer |
| Bit header (00 _{hex}) | Bit header | Bit header | Bit header |



For additional information on the input buffer, see □ 39ff.

Function Description

Input and Output Buffers

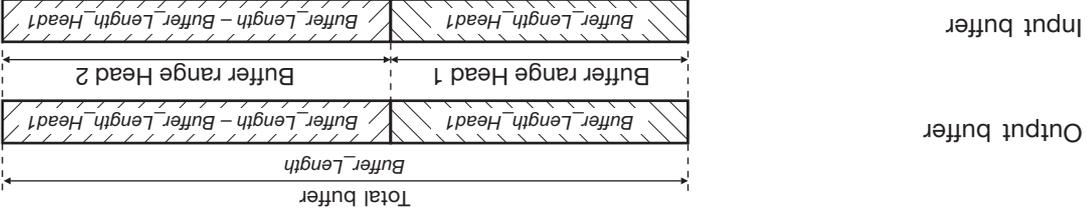
Mode 1 and Mode 2 comparison

Observe the following rules when setting the buffer lengths:

All the rules for parametering the buffers must be followed, even if the corresponding mode is not used!

Mode 1

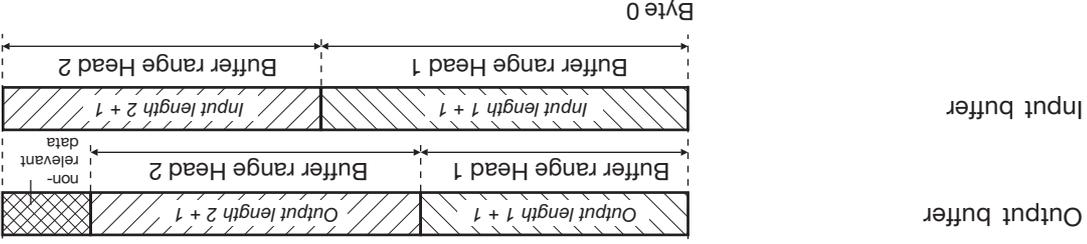
The buffer length for Head 1 must not exceed the total buffer length:
 $\rightarrow \text{Buffer_Length_Head1} \leq \text{Buffer_Length}$



Mode 2

The output buffer for Head 1 and Head 2 must not exceed the total buffer - 2:
 $\rightarrow \text{Output_length 1} + \text{Output_length 2} \leq \text{Buffer_Length} - 2$

The input buffer for Head 1 and Head 2 must not exceed the total buffer - 2:
 $\rightarrow \text{Input_length 1} + \text{Input_length 2} \leq \text{Buffer_Length} - 2$



Function Description Input and Output Buffers

Distributing the buffer in Mode 1

In **Mode 1** the entire input and output buffer is divided into 2 areas. The areas of the input buffer and output buffer are the same size for each head (see □ 33).

1. Buffer area Head 1: This area contains command identifiers and data for Read/Write Head 1. The size of the buffer area for Head 1 is specified by Parameter 11 *Buffer_Length_Head1*.

2. Buffer area Head 2: This area contains command identifiers and data for Read/Write Head 2. The size of this area is derived from the remaining area of the buffer area $\text{Head 2} = (\text{total buffer} - \text{buffer area Head 1})$.

Distributing the buffer in Mode 2

In **Mode 2** the entire input buffer is divided into 2 areas the entire output buffer into 2 areas. The areas of the input buffer and output buffer are different for each head (see □ 33).

1.1 Input buffer area Head 1: This area contains command identifiers and read data from Read/Write Head 1. The size of the input buffer area Head 1 is determined by Parameter 13 *Input_Length 1* (read data) + 1 byte (bit header).

1.2 Input buffer area Head 2: This area contains command identifiers and read data from Read/Write Head 2. The size of the input buffer area Head 2 is determined by Parameter 17 *Input_Length 2* (read data) + 1 byte (bit header).

2.1 Output buffer area Head 1: This area contains command identifiers and write data from Read/Write Head 1. The size of the output buffer area Head 1 is determined by Parameter 15 *Output_Length 1* (write data) + 1 byte (bit header).

2.2 Output buffer area Head 2: This area contains command identifiers and write data from Read/Write Head 2. The size of the output buffer area Head 2 is determined by Parameter 19 *Output_Length 2* (write data) + 1 byte (bit header).

Function Description Input and Output Buffers

Input and Output Buffers

In order to transmit commands and data between the BIS M-60_3 and the host system, the latter must prepare two fields. These two fields are:

- **the output buffer** for the control commands which are sent **to** the BIS M-60_3 and for the data to be written.

- **the input buffer**

for the data to be read and for the designators and error codes which come **from** the BIS M-60_3.

These data ranges are exchanged between the controller and the BIS M-60_3 processor by means of cyclical polling.

The polling I/O connection must have been successfully opened by the controller.



The possible setting values are stored in the EDS file.

The buffer size can be selected between 6 and 256 bytes. The parametered total buffer size applies to both BIS modes (Mode 1 and Mode 2).

The distribution of the total buffer over the two read/write heads depends on which BIS mode was selected.

When distributing the total buffer over the 2 heads, note in particular the explanations on 32...34.



Function Description Communication with the processor

Basic Procedure for Mode 2

1. The controller sends the processor the bit header with the RW bit and the AV bit. The RW bit tells the processor whether to carry out a read or write job. The AV bit tells the processor that there is a new job waiting. If a write job is carried out, the controller immediately sends the write data to the processor.
2. The processor accepts the job and sets the AA bit. It accepts the values for the start address and number of bytes to read or write from the device parametering. If the processor carried out the job correctly, it sets the AE bit. If a read job was carried out, it sends the read data to the controller.
3. In the case of a read job the controller takes the read data and sets the AV bit to 0.
4. The processor sets the AA bit and the AE bit to 0 - now it is ready for the next job.

Mode 1: Please see also 63...70 and 71...74.

Function Description Communication with the processor

Basic Procedure

Communication between the host system and the processor takes place using a fixed protocol sequence. Data integrity from the processor and vice-versa is indicated by a control bit. This bit is used to implement a handshake between the control and the processor.
Following is a simplified sequence for job sent by the controller to the processor for Mode 1 and Mode 2:

Basic Procedure for Mode 1

1. The control sends a command designator to the processor together with the associated command parameters and sets a bit (AV bit). This bit indicates to the processor that the transmitted data are valid and that the job is now beginning.
2. The processor takes the job and sets a bit (AA bit), which indicates this to the control.
3. If an additional exchange of data between the control and the processor is required to carry out the job, each uses a bit (TI bit and TO bit) to indicate that the control / processor is now ready for additional data exchange or has accepted the received data.
4. Once the processor has carried out the job correctly, it sets a bit (AF bit).
5. Once the control has accepted all the important data, it indicates this to the processor by resetting the bit that was set at the beginning (AV bit).
6. The processor now in turn sets all the control bits that were set during the sequence (AA bit, AE bit) and is ready for the next job.

Mode 1: Please see also  35...49 and  50...62. the examples on

Function Description

Operating Modes (Mode 1, Mode 2)

Mode Selection

The BIS mode is selected using parameter 5 *BIS_Mode*.
The following modes are available:

Mode 1

In Mode 1 a read/write job and dataexchange take place according to the standardized Balluff protocol for bus systems. A read/write job must be started with a command identifier, start address, and the number of bytes to read or write. The controller must also coordinate data exchange with the processor, i.e. monitor the validity of read data and display the validity of write data on the I/O buffer area of the respective r/w head. BIS Mode 1 is recommended,

– when large amounts of data need to be read or written on the data carrier
or
– when different areas on the data carrier are always read or written to.

Mode 2

Mode 2 is a simplified protocol for implementing read/write jobs. The processor indicates starting of a job simply by setting a bit in the bit header. For write jobs the write data are immediately sent. The job is automatically carried out by the processor using the previously parametered values for start address and number of bytes. The controller must only check the acknowledgement after processing the job and accept the read data in the case of read jobs. BIS Mode 2 is recommended,

– when smaller amounts of data need to be read or written on the data carrier,
and
– when the same area on the data carrier is always read or written to.

Function Description Parameterizing the BIS M-60_3 processor

Parameter 24

Apply buffer setting

Buffer_Ready

class: 64Hex
instance: 01Hex
attribute: 73Hex

Factory setting: *Not ready* (= 0)

With this parameter the BIS M-60_3 Processor signals that the settings for the buffer sizes should now be applied.

Other settings: *Ready* (= 1)

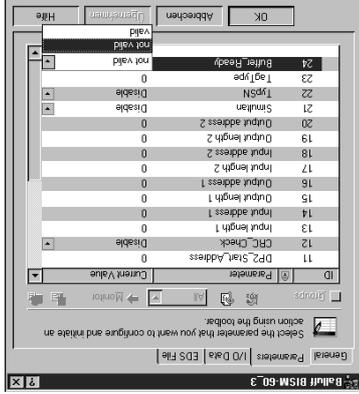
Please note that the settings for the parameters *Buffer_Length*, *Buffer_Length_Head1*, *Input_Length1*, *Output_Length1*, *Buffer_Length_Head2*, *Output_Length2* are not applied by the BIS M-60_3 Processor until the parameter *Buffer_Ready* with *Ready* (=1) has been sent.

For explanations of buffer sizes, see ¶ 31 ff.



When the parameter *Buffer_Ready* (=1) is sent, the BIS M-60_3 Processor checks whether the setting is valid. If yes, the new settings are applied. Otherwise the parameter setting is rejected with "invalid attribute value".

After checking the buffer settings, the value from *Buffer_Ready* is automatically reset by the BIS M-60_3 Processor to *Not ready* (= 0).



Function Description Parameterizing the BIS M-60_3 processor

Parameter 23 Data carrier type

TagType

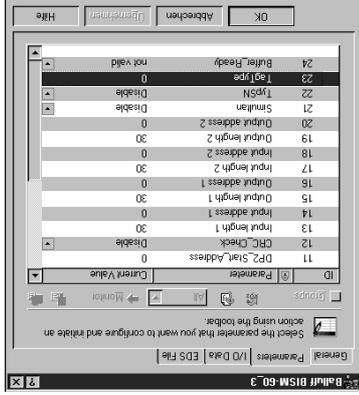
class: 64hex
instance: 01hex
attribute: 72hex

Factory setting: All data carrier types (= 0)

In this setting all data carrier types are detected and processed by the BIS M-60_3 processor.

Other settings:
Mifare (= FHex)
All Mifare data carriers supported by Balluff.

ISO15693 (= FHex)
All ISO15693 data carriers supported by Balluff.



Function Description Parameterizing the BIS M-60_3 processor

Parameter 21 Simultaneous data transmission

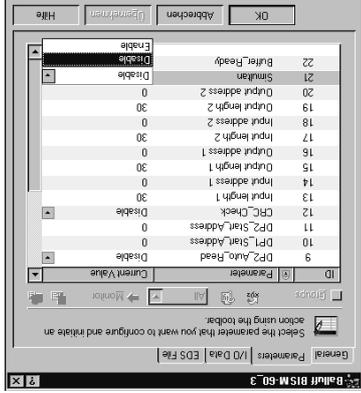
Simultan

class: 64Hex
instance: 01Hex
attribute: 70Hex

Factory setting: *Disable* (= 0)
Read/write jobs and data transmission on DeviceNet take place in succession.

Other settings: *Enable* (= 1)
Read/write jobs and data transmission on DeviceNet take place simultaneously.

See description of the Simultaneous data transmission function on □ 45.



Parameter 22 Output type and Serial number

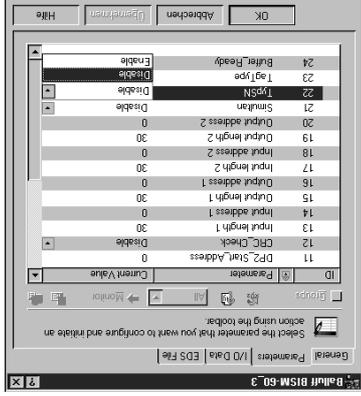
TypSN

class: 64Hex
instance: 01Hex
attribute: 71Hex

Factory setting: *Disable* (= 0)
At CT present the first read data from the data carrier are sent over the DeviceNet.

Other settings: *Enable* (= 1)
At CT present the data carrier type and the serial number of the data carrier are sent over the DeviceNet.

See description of the function "Output type and serial number" on □ 42.



Function Description

Parameterizing the BIS M-60_3 processor



Parameter 19
Mode 2:
No. of bytes for
writing Head 2

Output length 2

class: 65hex
 instance: 01hex
 attribute: 6Ehex

Factory setting:
 31 bytes

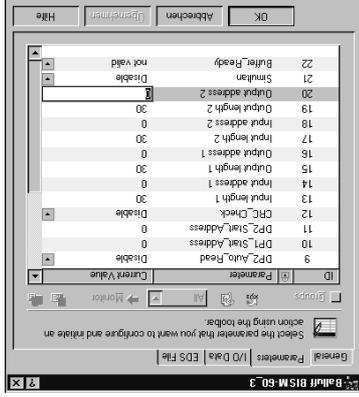
Other settings:
 2 bytes

(*Buffer_Length* – *Output length 1* – 2) bytes

This setting specifies how many bytes should be written to the data carrier in BIS Mode 2 on Head 2.



Please note when setting this parameter the explanation of the in- and output buffers starting on 31.



Parameter 20
Mode 2:
Start address for
writing Head 2

Output address 2

class: 65hex
 instance: 01hex
 attribute: 6Fhex

Factory setting:
 Start address 0

Other settings:
 1...1999

This setting specifies at which start address the data carrier on Head 2 should be written to in BIS Mode 2.

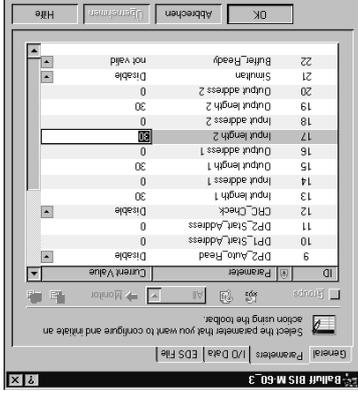
Function Description Parameterizing the BIS M-60_3 processor

Parameter 17
Mode 2:
No. of bytes to read
Head 2

Input length 2
class: 65Hex
instance: 01Hex
attribute: 6BHex
Factory setting:
31 byte
2 byte
Other settings:
 $(Buffer_Length - Input_length 1 - 2) \text{ byte}$
This setting specifies how many bytes should be read from
the data carrier in BIS Mode 2 on Head 2.

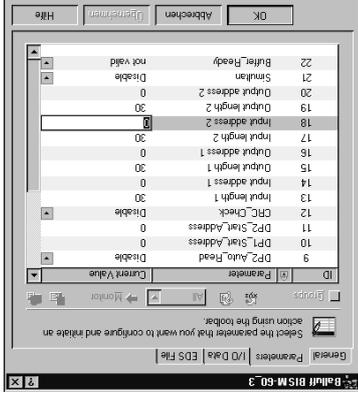


Please note when setting this parameter the explanation of
the in- and output buffers starting on 31.

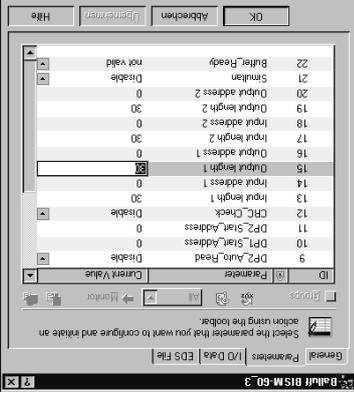


Parameter 18
Mode 2:
Start address for
Reading, Head 2

Input address 2
class: 65Hex
instance: 01Hex
attribute: 6CHex
Factory setting:
Start address 0
1 ... 1999
Other settings:
This setting specifies the starting address for reading data
carriers in BIS Mode 2 on Head 2.



Function Description Parameterizing the BIS M-60_3 processor



Parameter 15
Mode 2:
No. of bytes for
writing Head 1

Output length 1

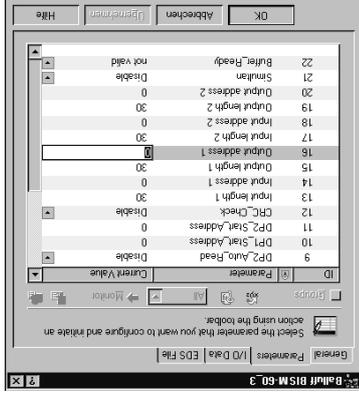
class: 65Hex
 instance: 01Hex
 attribute: 68Hex

Factory setting:
 31 bytes
 Other settings:
 2 bytes

This setting specifies how many bytes should be written to the data carrier in BIS Mode 2 on Head 1.



Please note when setting this parameter the explanation of the in- and output buffers starting on 31.



Parameter 16
Mode 2:
Start address for
writing Head 1

Output address 1

class: 65Hex
 instance: 01Hex
 attribute: 69Hex

Factory setting:
 Start address 0
 Other settings:
 1...1999

This setting specifies at which start address the data carrier on Head 1 should be written to in BIS Mode 2.

Function Description Parametering the BIS M-60_3 processor

Parameter 13
Mode 2:
No. of bytes to
read Head 1



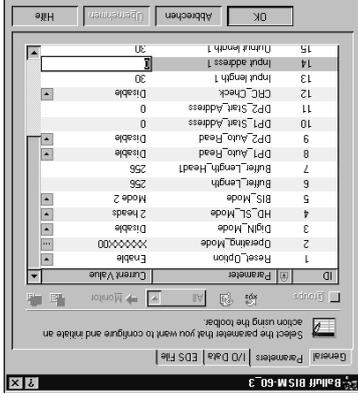
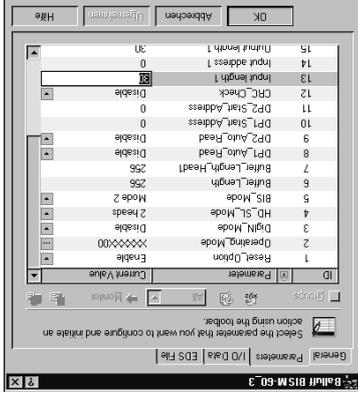
Input length 1
class: 65Hex
instance: 01Hex
attribute: 65Hex
Factory setting:
31 Byte
2 Byte
Other settings:
(Buffer_Length - Input length 2 - 2) Byte
This setting specifies how many bytes should be read from
the data carrier in BIS Mode 2 on Head 1.

Please note when setting this parameter the explanation of
the in- and output buffers starting on 31.

Parameter 14
Mode 2:
Start address for
reading, Head 1

Input address 1
class: 65Hex
instance: 01Hex
attribute: 66Hex
Factory setting:
Start address 0
Other settings:
1 ... 1999

This setting specifies the starting address for reading data
carriers in BIS Mode 2 on Head 1.



Function Description

Parameterizing the BIS M-60_3 processor

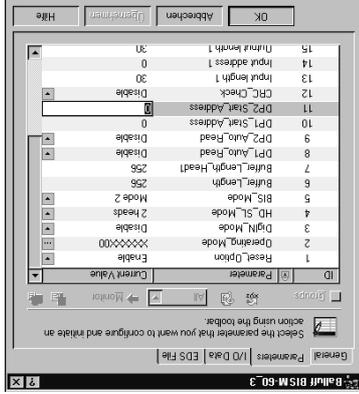
Parameter 11 Start Address for Auto Read Head 2

DP2_Start_Address

class: 64hex
instance: 01hex
attribute: 6Ehex
0
1...1999

Other settings:
This parameter sets the start address from which a data carrier is read when a data carrier is detected in front of Head 2.

For more information on the Auto Read function, see [□ 44](#).



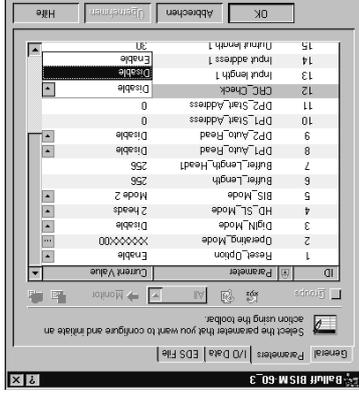
Parameter 12 CRC

CRC_Check

class: 64hex
instance: 01hex
attribute: 6Fhex
Disable (= 0)
Enable (= 1)

Other settings:

Validity of the read data is ensured by calculating the CRC_16 on the read data and comparing with the CRC_16 stored on the data carrier.



Function Description Parameterizing the BIS M-60_3 processor

Parameter 9 Auto Read for CT Present Head 2

DP2_Auto_Read

class: 64Hex
instance: 01Hex
attribute: 6CHex
Factory setting:
Disable (= 0)

CT Present data, when the data carrier enters the read/write zone of Head 2.

Other settings:
Enable (= 1)

If the Auto Read function is enabled, data are read starting from a specified start address as soon as a data carrier is detected in front of Head 2.

For more information on the Auto Read function, see □ 44.

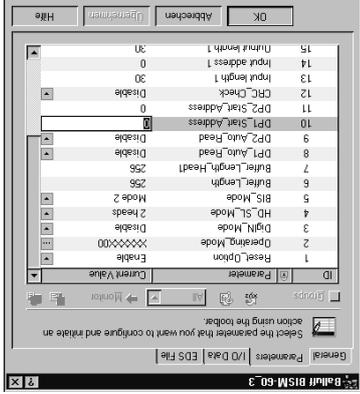
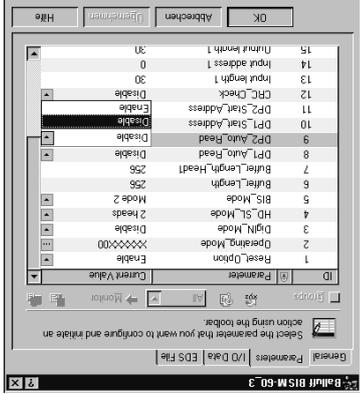
Parameter 10 Start Address for Auto Read Head 1

DP1_Start_Address

class: 64Hex
instance: 01Hex
attribute: 6DHex
Factory setting:
0
Other settings:
1...1999

This parameter sets the start address from which a data carrier is read when a data carrier is detected in front of Head 1.

For more information on the Auto Read function, see □ 44.



Function Description Parameterizing the BIS M-60_3 processor

Parameter 7 Buffer size for Head 1

*) Please take special note of the in-/output buffers starting on 31 when setting this parameter.

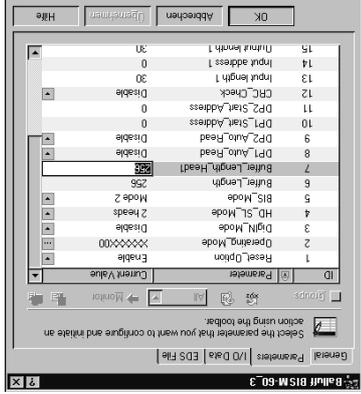
Buffer_Length_Head1

class: 64hex
instance: 01hex
attribute: 6Ahex

32 *)

Factory setting:
Other settings: 0...256 (maximal Buffer_Length)

This parameter is used to specify which part of the total buffer should be used for the bit header and the Head 1 data to be read and written. The remaining part of the total buffer is used for the bit header and the Head 2 data to be read and written. The parameter *Buffer_Length_Head1* is applicable only in BIS Mode 1.



DP1_Auto_Read

class: 64hex
instance: 01hex
attribute: 6Bhex

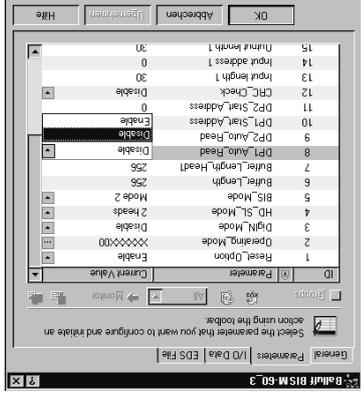
Disable (= 0)

Factory setting:
CT Present data, when the data carrier enters the read/write zone of Head 1.

Other settings: Enable (= 1)

If the Auto Read function is enabled, data are read starting from a specified start address as soon as a data carrier is detected in front of Head 1.

For more information on the Auto Read function, see 44.



Parameter 8 Auto Read for CT Present Head 1

Function Description Parameterizing the BIS M-60_3 processor

Parameter 5 BIS_Mode

BIS_Mode

class: 64Hex
instance: 01Hex
attribute: 68Hex

Mode 1 (= 0)

Factory setting: The device operates using Balluff bus protocol. Read and write jobs are coordinated by the controller using the I/O polling data. For detailed information on Mode 1, see 28 and 35 ff.

Mode 2 (= 1)

Other settings: Read and write jobs are automatically carried out using the setting specified in the parametering. A job is only triggered by the controller by setting control bits in the bit header of the I/O polling data. For detailed information on Mode 2, see 28 and 64 ff.

Parameter 6 Buffer_Length

Buffer_Length

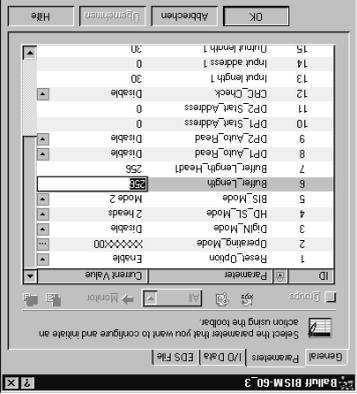
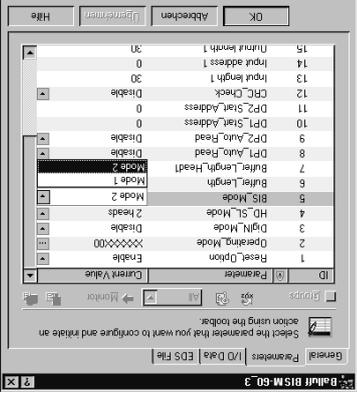
class: 64Hex
instance: 01Hex
attribute: 69Hex

Factory settings: 64 *)

Other settings: 6...256

In this setting the length of the input and output buffers are specified. The entire buffer size is sent between the controller and the BIS M-60_3 Processor in I/O polling. The setting is valid in BIS Mode 1 and BIS Mode 2.

*) Please take special note of the explanation of the in-/output buffers starting on 31 when setting this parameter.



Function Description

Parametering the BIS M-60_3 processor

Parameter 3 State of the digital input

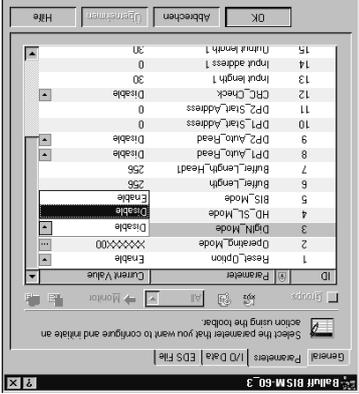
DigIn_Mode

class: 64hex
instance: 01hex
attribute: 66hex
Factory setting: *Disable* (= 0)

In this setting the state of the digital input is not shown in the bit header of the input buffer.

Other settings: *Enable* (= 1)

The state of the digital input is shown in the bit header of the input buffer.



Parameter 4 Read/write head selection

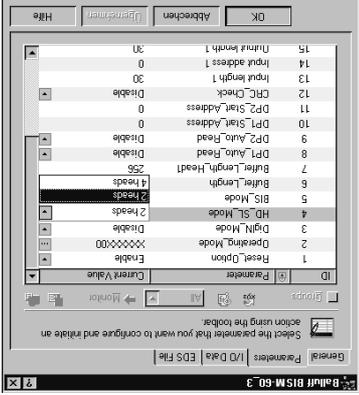
HD_SL_Mode

class: 64hex
instance: 01hex
attribute: 67hex
Factory setting: *2 heads* (= 0)

Select this setting if you want to operate the BIS M-60_3 Processor with 2 heads.

Other settings: *reserved* (= 1)

The setting is reserved for later applications.



Function Description Parametering the BIS M-60_3 processor

Parameter 1 Reset-Option

Reset_Option

class: 64Hex
instance: 01Hex
attribute: 64Hex

Factory setting: *Enable* (= 1)

In this setting the processor can be reset by the controller using a High signal on the digital input.

Other settings: *Disable* (= 0)

Reset by the controller not possible.

Parameter 2 Dynamic mode Head 1 / Head 2

Operating_Mode

class: 64Hex
instance: 01Hex
attribute: 65Hex

Factory setting: *no dynamic mode* (= 0)

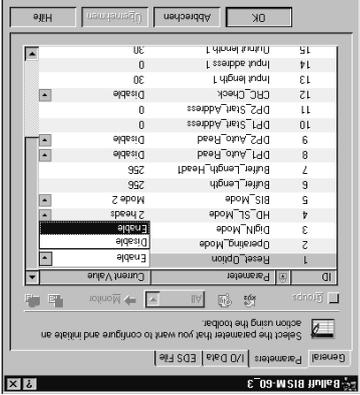
This setting allows dynamic mode to be enabled/disabled individually for each head.

Other settings:

dynamic head1 (= 1)
dynamic head2 (= 2)
dynamic head1 und 2 (= 3)

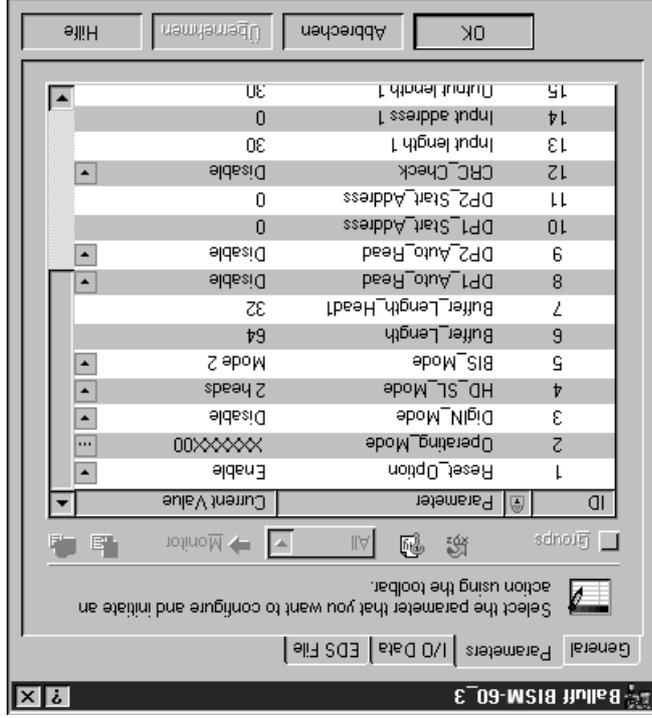
If the setting *dynamic mode* is not enabled for a head, a read command from the controller is only carried out if there is a data carrier in front of the read/write head. Otherwise the command is rejected with error code 1.

If the setting *dynamic mode* is enabled for a head, the read/write command is buffered and only carried out when there is a data carrier in front of the read/write head (see □ 45).



Function Description Parameterizing the BIS M-60_3 processor

The system can be parameterized online or offline. In offline parameterizing, the desired devices are drag&dropped from the hardware list to the network window and then configured. For online projecting, the *Network* → *Online* menu is searched and the found



devices displayed. In both cases the parameterizing is done by double-clicking on the device to be parameterized, here the BIS M-60_3. This opens the window for configuring the device. Clicking on Parameters opens the window for the device parameters.

The settings and values shown under *Current Value* indicate the factory settings when beginning the parameterizing. Once you have set a parameter, the [Apply] button is active and you can use it to apply the settings. The [OK] button ends the parametering.

Parameterizing
online / offline

Factory settings

Function Description Parameterizing the BIS M-60_3 processor

The parameters for operating the BIS M-60_3 Processor are stored in the BIS Config Object (class 64_{hex}) and in the BIS R/W Object (class 65_{hex}). The parameters are accessed using *explicit messages*.

Parameterizing with RS Network

A widely used tool for DeviceNet parameterizing is the Windows software RS Network from Rockwell Automation. The device parametering is therefore explained by example using this software. For information on the RS Network software and installation, please refer to the RS Network manual. The access path (*class* → *instance* → *attribute*) to each parameter is also indicated, so that direct access to each parameter is possible via *explicit message*.

The BIS M-60_3 Processor is parameterized with the projecting of the overall DeviceNet bus system. This generally consists of a DeviceNet Master and several DeviceNet nodes. For information on creating a project, please refer to the RS Network manual.

Installing EDS-file

Each BIS M-60_3 Processor is shipped with a diskette containing the EDS file (EDS = *electronic data sheet*). The EDS file contains all the device parameters for the BIS M-60_3 Processor and is installed in the EDS library of RS Network. To do this, start RS Network. Under the Tools menu select → EDS Wizard and follow the installation instructions.

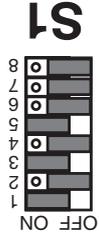
BUS interface DeviceNet

Setting the MAC-ID

The MAC-ID of the BIS M-60_3 Processor can be set using the slide switch S1.1 - 1.6. The slide switch S1.1 - 1.6 is binary coded. The station address is set as shown in the table:

| MAC-ID | 6 | 5 | 4 | 3 | 2 | 1 | 2 ₀ | 2 ₁ | 2 ₂ | 2 ₃ | 2 ₄ | 2 ₅ | 2 ₆ | 2 ₇ | 2 ₈ | 2 ₉ | ... | 42 | ... | 63 |
|-----------------|---|---|---|---|---|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----|-----|-----|-----|
| Slide switch S1 | 1 | 2 | 3 | 4 | 5 | 6 | OFF | ... | OFF | OFF | OFF |

CAUTION: Position S1.7 and S1.8 must always be OFF

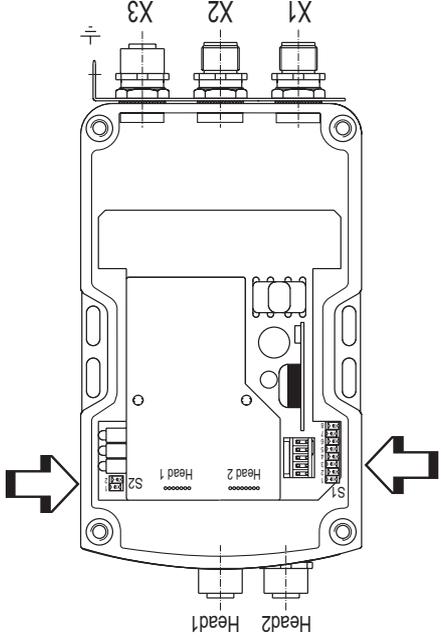


Slide switch S1
(cover opened)

Setting the baud rate

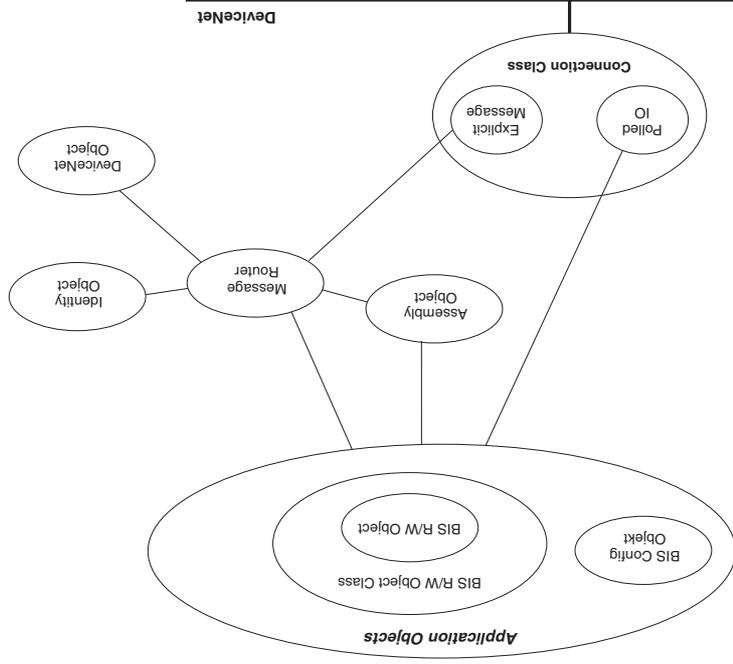
The baud rate of the BIS M-60_3 Processor can be set using slide switch S2. The following settings are possible:

| Baud rate | Slide switch S2 | 2 ₀ | 2 ₁ |
|-----------|-----------------|----------------|----------------|
| 125 kBaud | OFF | OFF | OFF |
| 250 kBaud | ON | OFF | OFF |
| 500 kBaud | ON | ON | OFF |
| invalid | ON | ON | ON |



Object model (continued)

The figure shows the object model of the BIS M-60_3 Processor. The "BIS Config Object" reflects the configuration properties of the device, and the "BIS R/W Object" the two read/write heads.



BUS interface DeviceNet

DeviceNet

Communication between the BIS M-60_3 processor and the host system is via DeviceNet. The DeviceNet system consists of the components:

- the Master (DeviceNet Master) as a controller component, and
- the bus nodes (here the BIS M-60_3 Processor)

A maximum of 63 nodes can be connected to a Master.

The Master converts the data: from the controlling system into the serial protocol of the DeviceNet network and the reverse. For this it possesses the DeviceNet port for connecting with the decentralized DeviceNet nodes. Information for configuring the Master can be found in the corresponding manual.

Opening the connection

The connection between the Master and the BIS M-60_3 Processor is made using the *predefined master/slave connection set*. The DeviceNet functionality of the BIS M-60_3 corresponds to a *group 2 only server*.

Object model

DeviceNet describes the parameters and functions of a device using an object model. Access by the network to the individual objects is accomplished exclusively using connection objects. The object model of the BIS M-60_3 Processor is shown in □ 11.

Control Function

The processor writes data from the host system to the data carrier or reads data from the tag through the read/write head and prepares it for the host system. Host systems may include:

- a host computer (e.g. industrial PC) or
- a programmable logic controller (PLC)

**Data checking with
CRC_16**

For applications requiring high security against bad data, CRC_16 checking can be used. Here a check code is written to the data carrier which allows the data to be checked for integrity at any time or location.

Advantages to CRC_16: Very high data integrity, even during the non-active phase (data carrier outside the active zone of the r/w head)

Disadvantages to CRC_16: Longer read/write times, some user data space is taken up on the data carrier.

Use of CRC_16 can be parameterized by the user. (see □ 20)

BIS M-60_3 Processor Basic knowledge for application

Selecting System Components

The **BIS M-6003** processor has a plastic housing.
The **BIS M-6023** processor has a metal housing.

Connection is made through round connectors. Two read/write heads can be cable connected.

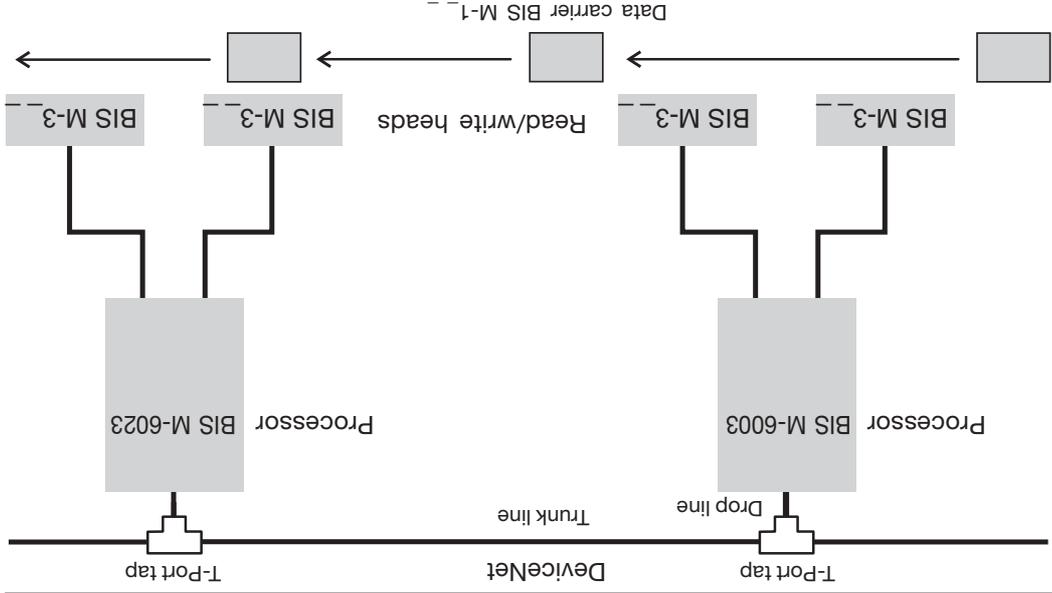
Series BIS M-60_3 processors have in addition a digital input. The input has various functions depending on the configuration (see Parametering).

The read/write distances depend on which data carriers are used. Additional information on the read/write heads in series BIS M-3_ _ including all the possible data carrier/read-write head combinations can be found in the manuals for the respective read/write heads.

The system components are electrically supplied by the processor. The data carrier represents a free-standing unit and needs no line-carried power. It receives its energy from the read/write head. The latter constantly sends out a carrier signal which supplies the code head as soon as the required distance between the two is reached. The read/write operation takes place during this phase. Reading and writing may be dynamic or static.

Introduction BIS M Identification Systems

**Configuration with
BIS M-6003 and
BIS M-6023
processor**
(connection with
T-Port taps and
Drop lines¹⁾)



*Schematic
representation of an
identification system
(example)*

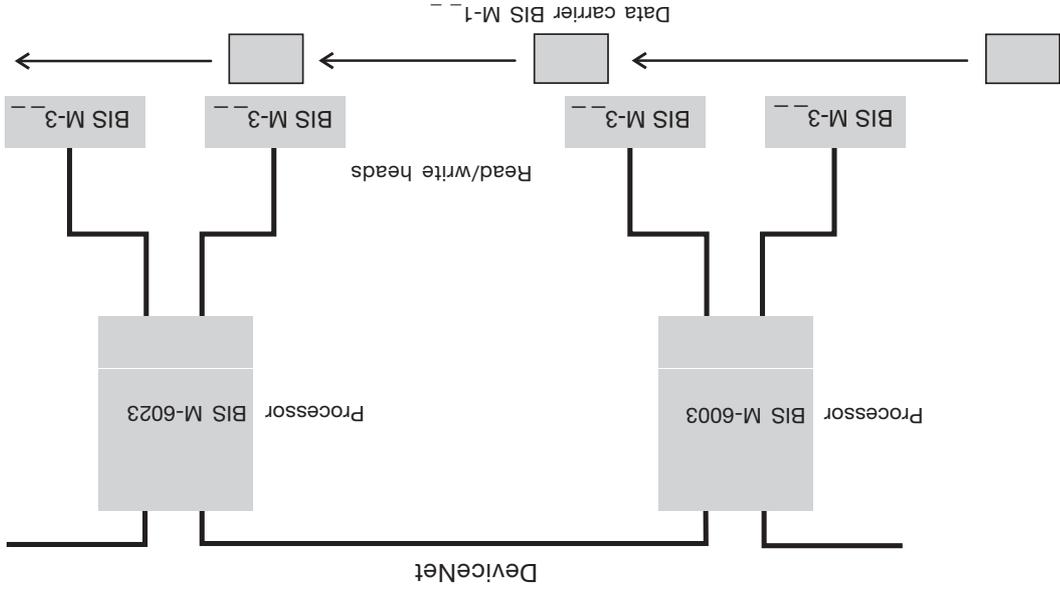
¹⁾ When connecting the BIS M-60_3 Processor per drop line, the DeviceNet output should be closed off with a cap (see Accessories □ 84/92).

Introduction BIS M Identification Systems

System Components

- The main components of the BIS M Identification Systems are:
- Processor,
 - Read/Write Heads and
 - Data Carriers

Configuration with
BIS M-6003 and
BIS M-6023
processor
(connection with
Trunk lines)



Schematic
representation of an
Identification System
(example)

Introduction BIS M Identification Systems

This manual is designed to assist the user 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.

The BIS M Identification Systems 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. The BIS M identification system allows the use of read-only data carriers.

If 2 read/write heads are connected to a BIS M-60_3 processor, both heads can be operated independently of each other. This means for example that you can read a data carrier from one head while writing to another data carrier at the other head.

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 storage systems for monitoring inventory movement;**
- **in transporting and conveying systems.**



Principles

Safety Considerations

| Approved Operation | Installation and Operation | Use and Checking | Fault Conditions | Scope |
|--|---|---|---|---|
| <p>Series BIS M-60_3 processors 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.</p> | <p>Installation and operation should be carried out by trained personnel only. Unauthorized work and improper use will void the warranty and liability.</p> <p>When installing the processor, follow the chapters containing the wiring diagrams closely. Special care is required when connecting the processor to external controllers, in particular with respect to selection and polarity of the signals and power supply.</p> <p>Only approved power supplies may be used for powering the processor. See chapter 'Technical Data' for details.</p> | <p>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.</p> <p>This includes maintaining the specified ambient conditions and regular testing for functionality of the identification system including all its associated components.</p> | <p>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.</p> | <p>This manual applies to processors in the series BIS M-6003-025-050-03-ST12 and BIS M-6023-025-050-03-ST13.</p> |
| <p>RS NetWorx is a registered trademark of the Rockwell Automation Corporation.</p> <p>Windows is a registered trademark of the Microsoft Corporation.</p> | <p>english</p> <p>BALLUFF</p> | <p>4</p> | | |

Contents

| | |
|-------|---|
| 4 | Safety Considerations |
| 5-7 | Introduction, BIS M Identification Systems |
| 8/9 | BIS M-60_3 Processor, Basic knowledge for application |
| 10-12 | BUS Interface DeviceNet |
| 13-27 | Function Description: Parameterizing the BIS M-60_3 processor |
| 28 | Operating Modes (Mode 1, Mode 2) |
| 29/30 | Communication with the processor |
| 31-34 | Input and Output Buffers |
| 35-38 | Mode 1 Output buffer, configuration and explanation |
| 39-41 | Input buffer, configuration and explanation |
| 42 | Data-carrier models |
| 43-49 | Processing data carriers |
| 50-62 | Examples for protocol sequence |
| 63-64 | Mode 2 Output buffer, configuration and explanation |
| 65-67 | Input buffer, configuration and explanation |
| 68-70 | Processing data carriers |
| 71-74 | Examples for protocol sequence |
| 75 | Read/Write Times |
| 76 | LED Display |
| 77 | Mounting Head / Processor |
| 78 | Opening the Processor |
| 79 | Interface Information / Wiring Diagrams |
| 80 | Changing the EEPROM |
| 81/82 | Technical Data |
| 83/84 | Ordering Information: Ordering Code, Accessory |
| 85 | Appendix, ASCII Table |

english

BALLUFF

No. 833 667 D/E • Edition 1401
Subject to modification.
Replaces edition 0703.

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

Manual

Electronic Identification Systems BIS
Processor BIS M-60_3
DeviceNet

Deutsch – bitte wenden!

