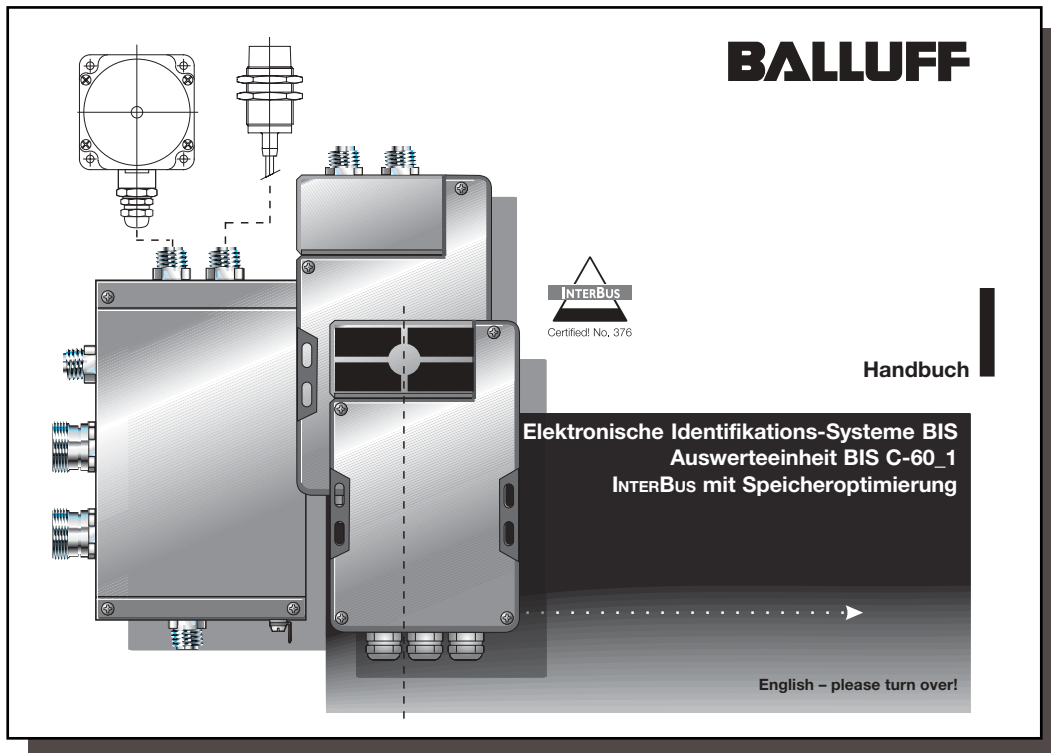


1



2

Nr. 823 188 D/E • Ausgabe 0108
Änderungen vorbehalten.

<http://www.balluff.de>

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Telefon +49 (0) 71 58/1 73-0
Telefax +49 (0) 71 58/50 10
E-Mail: balluff@balluff.de

Inhaltsverzeichnis

Sicherheitshinweise 4

Einführung Identifikations-System BIS C 5-7

Auswerteeinheit BIS C-60_1, Basiswissen für die Anwendung 8

Optimierte Speicherverwaltung im Codeträger 9-11

BUS-Anbindung: INTERBUS 12/13

Kompatibilität zur Auswerteeinheit BIS C-6_1 14

Funktionsbeschreibung:

Kommunikation mit der Auswerteeinheit 15

Ein- und Ausgangspuffer auf dem INTERBUS 16/17

Ausgangspuffer, Belegung und Erklärung 18-22

Eingangspuffer, Belegung und Erklärung 23-26

Konfiguration der Auswerteeinheit BIS C-60_1 27-29

Codeträger bearbeiten 30-33

Lesen und Schreiben 30

Lesen und Schreiben im Dynamikbetrieb 31

Lesen und Schreiben mit Programm 31

Gemischter Datenzugriff 32/33

Initialisierung überwachen 34

Speicher splitten 34

Beispiele für den Protokollablauf 35-47

Schreib-/Lesezeiten 48/49

Funktionsanzeigen 50/51

BIS C-6001 BIS C-6021

Montage Kopf / Auswerteeinheit 52 65

Öffnen der Auswerteeinheit / Schnittstelleninformation 53 66

Montage der Anschlusskabel / Montage PG-Verschraubung 54/55

Schnittstelleninformationen / Anschlusspläne 56-60 66-71

Wechseln des EEPROM 61 72

Technische Daten 62/63 73/74

Bestellinformationen 64 75

Anhang, ASCII-Tabelle 76

Sicherheitshinweise

Bestimmungsgemäßer Betrieb	Auswerteeinheiten BIS C-60_1 bilden zusammen mit den anderen Bausteinen des Systems BIS C 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 C-6001-029...03-KL2 sowie für die beiden Versionen ST8 und ST9 der Baureihe BIS C-6021-029-050-03-ST_.

INTERBUS ist ein eingetragenes Warenzeichen der Firma Phoenix Contact.

Einführung Identifikations-System BIS C

Dieses Handbuch soll den Anwender beim Einrichten des Steuerprogramms sowie bei der Installation und Inbetriebnahme der Auswerteeinheit BIS C-60_1 und weiterer Komponenten des Identifikations-Systems BIS C anleiten, so dass sich ein sofortiger, reibungsloser Betrieb anschließt.

Prinzip

Das Identifikations-System BIS C 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 Codeträger programmierte Informationen transportiert, sondern auch aktuelle Informationen gesammelt und weitergegeben werden.



Sind 2 Schreib-/Leseköpfe an die Auswerteeinheit BIS C-60_1 angeschlossen, können beide Schreib-/Leseköpfe unabhängig voneinander bearbeitet werden. D.h., an dem einen Schreib-/Lesekopf kann z.B. ein Codeträger gelesen werden, während an dem anderen Schreib-/Lesekopf auf einen anderen Codeträger 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,
- in der Werkzeugcodierung und -überwachung;
- in der Betriebsmittelorganisation;
- im Lagerbereich zur Kontrolle der Lagerbewegungen;
- im Transportwesen und in der Fördertechnik;
- in der Entsorgung zur mengenabhängigen Erfassung.

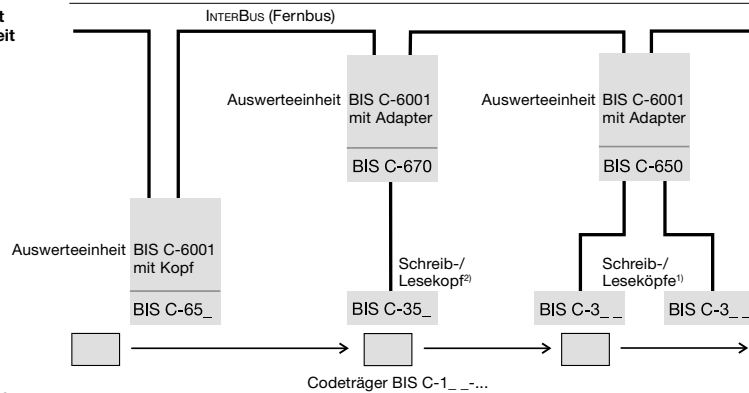
Einführung Identifikations-System BIS C

System- komponenten

Die Hauptbestandteile des Identifikationssystems BIS C sind:

- Auswerteeinheit,
- Schreib-/Leseköpfe und
- Codeträger

Anordnung mit Auswerteeinheit BIS C-6001



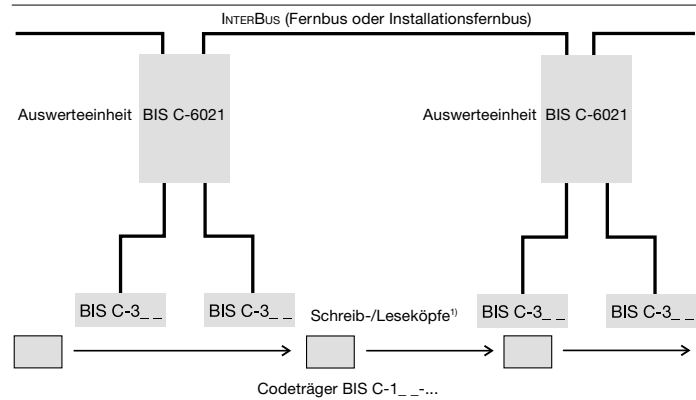
Schematische
Darstellung eines
Identifikations-Systems
(Beispiel)

¹) BIS C-3_ -Serie, ausgenommen BIS C-350 und -352

²) nur BIS C-350 oder -352

Einführung Identifikations-System BIS C

Anordnung mit
Auswerteeinheit
BIS C-6021



¹) BIS C-3_ _-Serie, ausgenommen BIS C-350 und -352

Auswerteeinheit BIS C-60_1 Basiswissen für die Anwendung

Auswahl der Systemkomponenten

Die Auswerteeinheit **BIS C-6001** besitzt ein Kunststoffgehäuse. Der Anschluss erfolgt über eine Klemmleiste, wobei die Kabel mittels PG-Verschraubung gesichert werden. An die Auswerteeinheit kann ein einzelner Schreib-/Lesekopf der Baureihe BIS C-65_ direkt montiert werden, wodurch eine kompakte Einheit entsteht. Alternativ können mittels Adapter BIS C-650 oder BIS C-670 Schreib-/Leseköpfe über Kabel an die Auswerteeinheit angeschlossen werden.

Die Auswerteeinheit **BIS C-6021** besitzt ein Metallgehäuse. Der Anschluss erfolgt über Rundsteckverbinder. Es können zwei Schreib-/Leseköpfe über Kabel angeschlossen werden.

Die Auswerteeinheiten BIS C-60_1 verfügen zusätzlich über einen digitalen Eingang. Der Eingang hat je nach Konfiguration unterschiedliche Funktionen (siehe Konfiguration).

Welche der oben beschriebenen Anordnungen bei den Schreib-/Leseköpfen sinnvoll ist, richtet sich im wesentlichen nach der möglichen räumlichen Anordnung der Bausteine. Funktionale Einschränkungen sind nicht gegeben. Alle Schreib-/Leseköpfe sind für statisches und dynamisches Lesen und Schreiben geeignet. Abstand und Relativgeschwindigkeit richten sich nach der Wahl des Codeträgers. In den jeweiligen Handbüchern zu den Schreib-/Leseköpfen der Baureihe BIS C-65_ sowie der Baureihe BIS C-3_ _ finden Sie sämtliche Kombinationen von Schreib-/Lesekopf und passenden Codeträgern.

Die Systemkomponenten werden von der Auswerteeinheit elektrisch versorgt. Der Codeträger 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 Codeträger versorgt, sobald der notwendige Abstand erreicht ist. In dieser Phase findet der Schreib-/Lesevorgang statt. Dieser kann statisch oder dynamisch erfolgen.

Auswerteeinheit BIS C-60_1
Optimierte Speicherverwaltung im Codeträger

Beschreibung

Die Version BIS C-60_1-029 arbeitet mit einem weiterentwickelten Balluff-Protokoll, das die Speicherverwaltung im Codeträger optimiert. Diese Variante ist dort notwendig, wo die erforderliche Anzahl der Schreibzyklen die für das EEPROM im Codeträger zugelassene Anzahl überschreitet. Diese liegt je nach Typ zwischen 100.000 und 1.000.000 und ist im Datenblatt eines jeden Codeträgers angegeben. Es muß ein Codeträger gewählt werden, dessen Speichergröße ein Mehrfaches der beim Schreiben tatsächlich genutzten Bytes beträgt.

Für einen gegebenen Codeträger kann man die maximal mögliche Anzahl der Schreibzyklen errechnen:

$$S = S_{CT} \cdot \frac{K_{CT} - 5}{n + 4}$$

K_{CT} = Speichergröße des Codeträgers in Byte
 S_{CT} = maximale Anzahl Schreibzyklen des Codeträgers lt. Datenblatt
 n = maximale Anzahl Byte der Schreib-/Lesebefehle

Bei dieser intelligenten Speicherverwaltung wird in einem Zähler die Anzahl der Schreibzyklen registriert. Nach Erreichen der zulässigen Zahl wird der anfangs benutzte Speicherbereich ausgeklammert und ein neuer Speicherbereich beschrieben. Dies wird von der Speicherverwaltung solange fortgesetzt, bis der letzte Speicherbereich "aufgebraucht" ist. Während der letzten 1.000 Schreibzyklen bestätigt die Auswerteeinheit alle erfolgreichen Schreibaufträge mit einer 'Vorwarnung' (56_{Hex} in Subadresse 02_{Hex} im Eingangspuffer).

Ist die maximale Anzahl der Schreibzyklen im letzten Speicherbereich erreicht, wird weiterhin auf diesen Speicherbereich geschrieben und es wird zusätzlich eine 'Endwarnung' (45_{Hex} in Subadresse 02_{Hex} im Eingangspuffer) angezeigt.

Auswerteeinheit BIS C-60_1
Optimierte Speicherverwaltung im Codeträger

Initialisieren des Codeträgers

Um einen Codeträger für diese Speicheroptimierung verwenden zu können, ist eine Initialisierung des Codeträgers durchzuführen. Das kann entweder mit dem Handyterminal BIS C-800 oder mit einem PC-Arbeitsplatz, in den die Karte BIS C-480-007-PC eingebaut ist, oder mit dem Initialisierungsbefehl des BIS C-60_1-029 erfolgen.

Die ersten 5 Byte des Speichers werden für die Kennung verwendet:

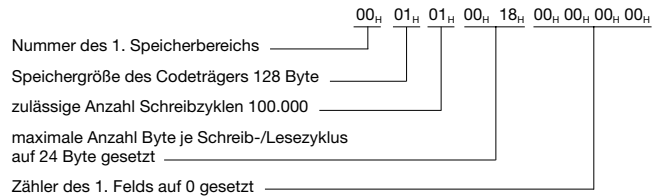
Byte Nr.	Initialisierung (hexadezimal)	Bedeutung / Funktion
0	00H	Nummer des aktuellen Speicherbereichs auf dem Codeträger
1	01H	128 Byte Speichergröße des Codeträgers eintragen
	02H	256 Byte
	03H	512 Byte
	04H	1 024 Byte
	05H	2 048 Byte
	06H	4 096 Byte
	07H	8 192 Byte
	08H	16 384 Byte
	09H	127 Byte
	0AH	255 Byte
2	0BH	511 Byte
	0CH	1 023 Byte
	0DH	2 047 Byte
	01H	100.000 maximale Anzahl Schreibzyklen des Codeträgers eintragen
	02H	200.000 (siehe Datenblatt)
	03H	300.000
3 und 4	04H	400.000
	05H	500.000
	0AH	1.000.000
	maximal 0400H	maximale Anzahl Byte je Schreib-/Lesezyklus = 1 kByte (Segmentgröße)

Auswerteeinheit BIS C-60_1 Optimierte Speicherverwaltung im Codeträger

Initialisieren des Codeträgers (Fortsetzung)

Beispiel einer Initialisierung:

Auf einem Codeträger mit 128 Byte Speichergröße sollen 24 Byte zum Schreiben/Lesen verwendet werden. Die maximal zulässige Anzahl von Schreibzyklen laut dem Codeträger-Datenblatt beträgt 100.000. Daraus resultiert die folgende Initialisierung der ersten 9 Byte des Speichers:



Eine im Schreib- oder Lesebefehl genannte Adresse unterscheidet sich nicht zwischen Codeträgern mit und ohne Initialisierung.

Das Verhältnis zwischen der Speichergröße des Codeträgers und der tatsächlich benötigten Größe des Speicherbereichs läßt 400.000 Schreibzyklen zu, da die verfügbare Speichergröße nacheinander für die Belegung von 4 Speicherbereichen genutzt werden kann. Die Vorwarnung setzt nach dem 399.000. Schreibvorgang ein.

Steuerfunktion

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

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

BUS-Anbindung: INTERBUS

INTERBUS

Die Kommunikation zwischen der Auswerteeinheit BIS C-60_1 und dem steuernden System erfolgt über den INTERBUS.

Das INTERBUS-System besteht aus drei Komponenten:

- der Anschaltbaugruppe (Einschubkarte für Industrie-PC oder SPS),
- der Busklemme als Netzwerkknoten und/oder
- den E/A-Modulen (hier die Auswerteeinheit BIS C-60_1).

Je nach Anschaltbaugruppe können maximal 63 BIS C-60_1 angeschlossen werden.

Die Auswerteeinheit BIS C-6001 wird als Fernbusteilnehmer eingesetzt. Die Auswerteeinheit BIS C-6021 kann als Fernbus- oder als Installationsbusteilnehmer eingesetzt werden.



Wichtiger Hinweis für den Einsatz mit SPS:

Es gibt Steuerungen, bei denen der Datenbereich des INTERBUS nicht synchron zur Aktualisierung des Ein-/Ausgangsabbildes übertragen wird. Werden mehr als 2 Byte Daten übertragen, muss ein Mechanismus verwendet werden, der garantiert, dass die Daten in der SPS und die Daten im BIS C immer gleich sind!

Abhilfe: 2. Bitleiste einstellen

Der Datenaustausch zwischen SPS und BIS wird über die sogenannte Bitleiste gesteuert. Dies ist immer das erste Byte des jeweiligen Schreib-/Lesekopfs im Datenpuffer. Sowohl im Eingangsbereich (Daten vom BIS an die SPS) als auch im Ausgangsbereich (Daten von der SPS an das BIS) ist diese Bitleiste vorhanden. Wird nun diese Bitleiste zusätzlich als letztes Byte übertragen, kann durch Vergleich dieser beiden Byte die Konsistenz der übertragenen Daten garantiert werden.

Mit dieser Methode wird weder der SPS-Zyklus beeinflusst noch die Bus-Zugriffszeit verändert. Es wird lediglich ein Byte im Datenpuffer für das Byte der 2. Bitleiste benötigt, anstatt es für Daten zu nutzen.

Diese Möglichkeit wird von Balluff als Einstellung empfohlen (Werkseinstellung).

BUS-Anbindung: INTERBus

Die Adresseinstellung erfolgt auf der Anschaltbaugruppe (nicht auf den E/A-Modulen, also auch nicht auf der Auswerteeinheit BIS C-60_1). Es sind zwei Arten der Adressierung möglich:

1. die logische Adressierung und
2. die physikalisch Adressierung.

Logische Adressierung

Die logische Adressierung ermöglicht die freie Adressierung eines jeden Moduls.
Vorteil: hohe Sicherheit und Flexibilität;
Nachteil: aufwendig beim Einrichten.

Für das Identifikationssystem BIS C-60_1 gilt:

E/A-Modultyp	IDENT-Nr.	IN-Adresse (Byte)	OUT-Adresse (Byte)
Auswerteeinheit BIS C-60_1	03	16	16

Physikalische Adressierung

Die physikalische Adressierung ist starr an die Konfiguration des Systems gebunden. Die Adresse eines jeden Moduls hängt von der Reihenfolge der Anschlüsse ab.
Vorteil: beim Einrichten einfach zu realisieren;
Nachteil: Änderungen der Modulanordnung im stromlosen Zustand werden bei der Initialisierung berücksichtigt, aber dem Anwender nicht mitgeteilt.

Kompatibilität zur Auswerteeinheit BIS C-6_1

Kompatibilität einstellen



Über die Stiftleiste X5 und einen Jumper kann die Kompatibilität zu den Auswerteeinheiten BIS C-6_1 hergestellt werden.

Ist die Auswerteeinheit BIS-C-60_1 kompatibel zu BIS C-601 bzw. BIS C-621 eingestellt, müssen alle Einstellungen für den Datenaustausch entsprechend den Kapiteln Parametrierung, Funktionsbeschreibung, Protokollablauf und LED-Anzeige der Betriebsanleitung für die Auswerteeinheiten BIS C-6_1 vorgenommen werden! Diese Betriebsanleitung erhalten Sie entweder auf Anforderung, oder Sie können sie im Internet unter www.balluff.de herunterladen.

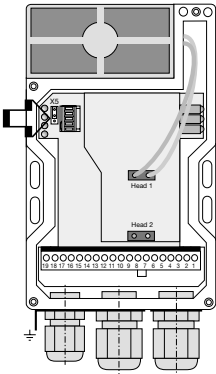
Öffnen des Deckels der Auswerteeinheit
BIS C-6001 siehe  59, BIS C-6021 siehe  72.

Jumper-Stellung Stiftleiste X5	Auswerteeinheit kompatibel mit BIS C-6_1
1-2	nein
2-3	ja



Im Bild ist keine Kompatibilität zu BIS C-6_1 eingestellt.

 In der Betriebsart "kompatibel mit BIS C-6_1" ist der Fehlercode um die Fehlernummer 11_{Hex} erweitert. Diese Fehlernummer besagt, dass ein verwendeter Befehl in dieser Betriebsart nicht funktioniert.



Stiftleiste X5
(bei geöffnetem Deckel)

Funktionsbeschreibung Kommunikation mit der Auswerteeinheit

Prinzipieller Ablauf

Die Kommunikation zwischen dem steuernden System und der Auswerteeinheit erfolgt in einem festen Protokollaufbau. 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 ein Handshake zwischen Steuerung und Auswerteeinheit realisiert.

Hieraus ergibt sich der folgende, vereinfacht dargestellte Ablauf eines Auftrags der Steuerung an die Auswerteeinheit:

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.

Funktionsbeschreibung Ein- und Ausgangspuffer auf dem INTERBUS

Eingangs- und Ausgangspuffer

Zur Übertragung der Steuerbit, Befehle und Daten zwischen der Auswerteeinheit BIS C-60_1 und dem steuernden System muss dieses zwei Felder bereitstellen. Die beiden Felder sind:

- **der Ausgangspuffer**
für die Steuerbit (Bitleiste) und Befehle der Steuerung, die **zum** BIS-Identifikations-System geschickt werden,
für die zu schreibenden Daten und
für die Konfiguration zur Einstellung der Auswerteeinheit BIS C-60_1.
- **der Eingangspuffer**
für die Steuerbit (Bitleiste) der Auswerteeinheit BIS C-60_1,
für die zu lesenden Daten,
für die Kennungen und Fehlercodes, die **vom** BIS-Identifikations-System kommen und
für das Auslesen der Konfigurationsdaten.

Die gesamte Puffergröße des BIS C-60_1 beträgt 16 Byte für den Eingangspuffer und 16 Byte für den Ausgangspuffer. Diese Gesamtpuffergröße wird in 2 Bereiche aufgeteilt:

Pufferbereich 1 für Schreib-/Lesekopf 1: 8 Byte Eingangspuffer, 8 Byte Ausgangspuffer
Pufferbereich 2 für Schreib-/Lesekopf 2: 8 Byte Eingangspuffer, 8 Byte Ausgangspuffer

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

Funktionsbeschreibung Ein- und Ausgangspuffer auf dem InterBus

Eingangs- und Ausgangspuffer (Fortsetzung)

Beispiel: Bei einer SPS soll der Pufferbereich für das BIS C-60_1 bei Eingangsbyte EB 32 und Ausgangsbyte AB 32 beginnen.

Speicherbild:

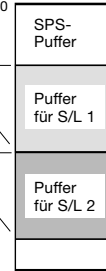
Schreib-/Lesekopf 1 (S/L1):
Eingangspuffer
Ausgangspuffer

Schreib-/Lesekopf 2 (S/L2):
Eingangspuffer
Ausgangspuffer

SPS:
von EB 32 bis EB 39
von AB 32 bis AB 39

von EB 40 bis EB 47
von AB 40 bis AB 47

EB 0 / AB 0



BIS:

Kopf S/L 1
Subadresse 0

...
Subadresse 7

Kopf S/L 2
Subadresse 0

...
Subadresse 7



Es ist zu beachten, dass diese Puffer je nach Steuerungstyp unterschiedlich abgebildet werden.

Nachfolgend wird stets die Beschreibung nach Variante 1 dargestellt!

	Variante 1	Variante 2
Subadresse	00	Subadresse 01
	01	00
	02	03
	03	02
	04	05
	05	04
	06	07
	07	06

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

Funktionsbeschreibung Ausgangspuffer, Belegung und Erklärung

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

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

Erklärungen zum Ausgangspuffer

Sub-adresse	Bit-name	Bedeutung	Funktionsbeschreibung
00 _{Hex} Bitleiste	CT	Codeträgertyp 0	Codeträgertyp auswählen: 32 Byte Blockgröße 64 Byte Blockgröße
	TI	Toggle-Bit In	beim Lesen: Steuerung ist bereit neue/weitere Daten zu übernehmen. beim Schreiben: Steuerung hat neue/weitere Daten bereitgestellt.
	GR	Grundzustand	Veranlasst das BIS-System, in den Grundzustand zu gehen.
	AV	Auftrag	Signalisiert dem Identifikations-System, dass ein Auftrag vorliegt.

(Fortsetzung siehe nächste )

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

Funktionsbeschreibung
Ausgangspuffer, Belegung und Erklärung

Erklärungen zum
Ausgangspuffer
(Fortsetzung)

Sub- adresse	Bedeutung	Funktionsbeschreibung
01_{Hex}	Befehlskennung	
00 _{Hex}		Kein Befehl vorhanden
01 _{Hex}		Codeträger lesen
02 _{Hex}		auf Codeträger schreiben
04 _{Hex}		Auswerteeinheit konfigurieren
05 _{Hex}		Konfigurationsdaten lesen
06 _{Hex}		Speichern des Programms im EEPROM für die Funktion Gemischter Datenzugriff
07 _{Hex}		Speichern der Anfangsadresse für die Funktion Auto-Lesen im EEPROM
08 _{Hex}		Codeträger initialisieren
09 _{Hex}		Überwachung der Codeträger-Initialisierung im EEPROM speichern
0A _{Hex}		Splitadresse zum Aufteilen des Codeträgers in einen Bereich mit Speicheroptimierung und einen Bereich ohne Speicher-optimierung im EEPROM speichern
21 _{Hex}		Codeträger lesen über die Funktion Gemischter Datenzugriff (entsprechend dem im EEPROM abgelegten Programm)
22 _{Hex}		Auf Codeträger schreiben über die Funktion Gemischter Datenzugriff (entsprechend dem im EEPROM abgelegten Programm)
oder	Konfiguration 1. Byte	
00 _{Hex}		Defaultwert (Werkseinstellung). Dieser Wert ändert sich je nach Konfiguration.
oder	Initialisierungsdaten	
oder	Daten	zum Schreiben auf den Codeträger
oder	Programmdaten	zum Schreiben auf das EEPROM.

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

(Fortsetzung siehe nächste )

Funktionsbeschreibung
Ausgangspuffer, Belegung und Erklärung

Erklärungen zum
Ausgangspuffer
(Fortsetzung)

Sub- adresse	Bedeutung	Funktionsbeschreibung
02_{Hex}	Anfangsadresse (Low Byte)	Startadresse, ab der vom Codeträger gelesen bzw. auf den Codeträger 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.
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	Konfiguration 2. Byte	
80 _{Hex}		Defaultwert (Werkseinstellung). Dieser Wert ändert sich je nach Konfiguration.
oder	Initialisierungsdaten	
oder	Daten	zum Schreiben auf den Codeträger
oder	Programmdaten	zum Schreiben auf das EEPROM
oder	Parameterdaten	zum Schreiben auf das EEPROM.

(Fortsetzung siehe nächste )

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

Funktionsbeschreibung
Ausgangspuffer, Belegung und Erklärung

Erklärungen zum
Ausgangspuffer
(Fortsetzung)

Sub- adresse	Bedeutung	Funktionsbeschreibung
03_{Hex}	Anfangsadresse (High Byte)	Startadresse, ab der vom Codeträger gelesen bzw. auf den Codeträger geschrieben werden soll (das High Byte wird zusätzlich für den Adressbereich von 256 bis 8.191 benötigt).
oder	Konfiguration 3. Byte 00 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert darf nicht geändert werden!
oder	Initialisierungsdaten	
oder	Daten	zum Schreiben auf den Codeträger
oder	Programmdaten	zum Schreiben auf das EEPROM.
oder	Parameterdaten	zum Schreiben auf das EEPROM.
04_{Hex}	Anzahl Byte (Low Byte)	Anzahl Byte, die ab Anfangsadresse gelesen bzw. geschrieben werden sollen (das Low Byte deckt den Umfang von 1 bis 255 Byte ab).
oder	Konfiguration 4. Byte 82 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert ändert sich je nach Konfiguration.
oder	Initialisierungsdaten	
oder	Daten	zum Schreiben auf den Codeträger
oder	Programmdaten	zum Schreiben auf das EEPROM.

(Fortsetzung siehe nächste Seite)

Bitte beachten Sie den prinzipiellen Ablauf auf den Seiten 15 und 30...34 und die Beispiele auf den Seiten 35...47.

Funktionsbeschreibung
Ausgangspuffer, Belegung und Erklärung

Erklärungen zum
Ausgangspuffer
(Fortsetzung)

Sub- adresse	Bedeutung	Funktionsbeschreibung
05_{Hex}	Anzahl Byte (High Byte)	Anzahl Byte, die ab Anfangsadresse gelesen bzw. geschrieben werden sollen (das High Byte wird zusätzlich für den Umfang von 256 bis 8.191 Byte benötigt).
oder	Konfiguration 5. Byte 00 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert ändert sich je nach Konfiguration.
oder	Initialisierungsdaten	
oder	Daten	zum Schreiben auf den Codeträger
oder	Programmdaten	zum Schreiben auf das EEPROM.
06_{Hex}	Konfiguration 6. Byte 00 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert darf nicht geändert werden!
oder	Daten	zum Schreiben auf den Codeträger
oder	Programmdaten	zum Schreiben auf das EEPROM.
07_{Hex}	2. Bitleiste	Stimmen 1. und 2. Bitleiste überein, liegen gültige Daten vor.
oder	Daten	zum Schreiben auf den Codeträger
oder	Programmdaten	zum Schreiben auf das EEPROM.

Bitte beachten Sie den prinzipiellen Ablauf auf den Seiten 15 und 30...34 und die Beispiele auf den Seiten 35...47.

Funktionsbeschreibung Eingangspuffer, Belegung und Erklärung

**Belegung des
Eingangspuffers für
einen (1) Schreib-/
Lesekopf**

Bit-Nr.	7	6	5	4	3	2	1	0	Bitname	
Subadresse										
00 _{Hex} = Bitleiste	BB	HF	TO	IN	AF	AE	AA	CP		
01 _{Hex}	Fehlercode		oder			Daten		oder		Konfig. 1. Byte
02 _{Hex}	Daten				oder		Konfig. 2. Byte			
03 _{Hex}	Daten				oder		Konfig. 3. Byte			
04 _{Hex}	Daten				oder		Konfig. 4. Byte			
05 _{Hex}	Daten				oder		Konfig. 5. Byte			
06 _{Hex}	Daten				oder		Konfig. 6. Byte			
07 _{Hex}	2. Bitleiste (wie oben)					oder		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.
		(Fortsetzung siehe nächste Seite)	

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

Funktionsbeschreibung Eingangspuffer, Belegung und Erklärung

**Erklärungen zum
Eingangspuffer
(Fortsetzung)**

Sub- adresse	Bit- name	Bedeutung	Funktionsbeschreibung
00 _{Hex} Bitleiste	(Fortsetzung) IN	Input	Wenn der Parameter Eingang IN = 1 ist, zeigt dieses Bit den Zustand des Eingangs an.
	AF	Auftrag Fehler	Der Auftrag wurde fehlerhaft bearbeitet oder abgebrochen.
	AE	Auftrag Ende	Der Auftrag wurde ohne Fehler beendet.
	AA	Auftrag Anfang	Der Auftrag wurde erkannt und begonnen.
	CP	Codetag Present	Codeträger im Schreib-/Lesebereich des angewählten Schreib-/Lesekopfs.
Sub- adresse	Bedeutung		Funktionsbeschreibung
01 _{Hex}	Fehlercode		Fehlernummer ist eingetragen, wenn Auftrag fehlerhaft bearbeitet oder abgebrochen wurde. Nur mit AF-Bit gültig!
	01 _{Hex}		Lesen oder Schreiben nicht möglich, da kein Codeträger im Schreib-/Lesebereich des Schreib-/Lesekopfs vorhanden.
	02 _{Hex}		Fehler beim Lesen.
	03 _{Hex}		Codeträger wurde während des Lesens aus dem Schreib-/Lesebereich des Schreib-/Lesekopfs entfernt.
	04 _{Hex}		Fehler beim Schreiben.
	05 _{Hex}		Codeträger 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; Anzahl Byte ist 00 _{Hex} .
		oder	(Fortsetzung siehe nächste Seite)

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

**Funktionsbeschreibung
Eingangspuffer, Belegung und Erklärung**

**Erklärungen zum
Eingangspuffer
(Fortsetzung)**

Sub- adresse	Bedeutung	Funktionsbeschreibung
01_{Hex}	Fehlercode (Fortsetzung)	
	09 _{Hex}	Kabelbruch zum angewählten Schreib-/Lesekopf oder Kopf nicht angeschlossen.
	0C _{Hex}	Das EEPROM kann nicht gelesen/beschrieben werden.
	0D _{Hex}	Gestörtes Timing bei der Kommunikation mit dem Codeträger.
	0F _{Hex}	Inhalt der 1. und 2. Bitleiste (1. und letztes Byte) des Ausgangspuffers sind ungleich (2. Bitleiste muss aktiviert sein).
	13 _{Hex}	Startadresse + Anzahl Byte > Speicherbereich in Initialisierung angegeben.
	14 _{Hex}	Ungültige max. Anzahl Schreibzyklen in Initialisierung angegeben.
	15 _{Hex}	Ungültige Speichergröße des Codeträgers in Initialisierung angegeben.
	16 _{Hex}	Max. Anzahl von 1 kByte als Segmentgröße überschritten.
	17 _{Hex}	Codeträger falsch initialisiert (Vergleich mit Vorgabedaten über Befehlskennung 09 _{Hex} bringt keine Übereinstimmung).
	18 _{Hex}	Codeträger noch nicht initialisiert. Alle Byte sind noch 00 _{Hex} .
oder	Konfiguration 1. Byte	
	00 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert ändert sich je nach Konfiguration.
oder:	Daten	Daten, die vom Codeträger gelesen werden.

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

(Fortsetzung siehe nächste )

**Funktionsbeschreibung
Eingangspuffer, Belegung und Erklärung**

**Erklärungen zum
Eingangspuffer
(Fortsetzung)**

Sub- adresse	Bedeutung	Funktionsbeschreibung
02_{Hex}	Konfiguration 2. Byte	
	80 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert darf nicht geändert werden.
oder	Daten	Daten, die vom Codeträger gelesen werden.
oder:	Warnmeldung	Speicherverwaltung (Erklärungen siehe  9) meldet: Vorwarnung (noch 1.000 Schreibzyklen erlaubt). Endwarnung.
	56 _{Hex}	
	45 _{Hex}	
03_{Hex}	Konfiguration 3. Byte	
	00 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert ändert sich je nach Konfiguration.
oder	Daten	Daten, die vom Codeträger gelesen werden.
04_{Hex}	Konfiguration 4. Byte	
	82 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert ändert sich je nach Konfiguration.
oder	Daten	Daten, die vom Codeträger gelesen werden.
05_{Hex}	Konfiguration 5. Byte	
	00 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert ändert sich je nach Konfiguration.
oder	Daten	Daten, die vom Codeträger gelesen werden.
06_{Hex}	Konfiguration 6. Byte	
	00 _{Hex}	Defaultwert (Werkseinstellung). Dieser Wert darf nicht geändert werden.
oder	Daten	Daten, die vom Codeträger gelesen werden.
07_{Hex}	2. Bitleiste	Stimmen 1. und 2. Bitleiste überein, liegen gültige Daten vor.
oder	Daten	Daten, die vom Codeträger gelesen werden.

Bitte beachten Sie den prinzipiellen Ablauf auf den  15 und 30...34 und die Beispiele auf den  35...47.

Funktionsbeschreibung
Konfiguration der Auswerteeinheit BIS C-60_1

Konfiguration, Übersicht

Folgende Funktionen können mit Hilfe der Konfiguration aktiviert / deaktiviert werden:

- **Dynamikbetrieb an Schreib-/Lesekopf 1 oder 2:**
Ist Dynamikbetrieb parametrierbar, kann ein Schreib-/Leseauftrag gesendet werden, obwohl kein Codeträger im aktiven Bereich des Kopfs vorhanden ist. Fährt ein Codeträger nun vor den Kopf, wird der Befehl sofort ausgeführt.
- **2. Bitleiste am Ende des Ein- und Ausgangspuffers:**
Die 2. Bitleiste (Werkseinstellung) verhindert, dass Daten vom Bus übernommen werden, solange dieser noch nicht vollständig aktualisiert ist.
- **Zustand des digitalen Eingangs in der Bitleiste des Eingangspuffers anzeigen:**
Ist diese Funktion aktiviert, zeigt das IN-Bit den Zustand des digitalen Eingangs der Auswerteeinheit an: IN = 0 → digitaler Eingang low; IN = 1 → digitaler Eingang high
- **Reset der Auswerteeinheit BIS C-60_1 über den digitalen Eingang:**
Ist diese Funktion aktiviert, wird ein Reset der Auswerteeinheit durchgeführt, wenn der digitale Eingang auf high gelegt wird.
- **Überwachung der Codeträgerinitialisierung:**
Ist diese Funktion aktiviert, werden die Initialisierungsdaten auf dem Codeträger mit den in der Auswerteeinheit abgespeicherten Daten verglichen. Stimmen diese nicht überein, wird ein Schreib-/Lesebefehl mit einer Fehlermeldung abgewehrt.
- **Speicher des Codeträgers splitten:**
Ist diese Funktion aktiviert, werden Daten auf dem Codeträger ab einer vorgegebenen Adresse nicht mehr speicheroptimiert bearbeitet. Die Splitadresse muss zuvor mit der Befehlskennung 0A_{hex} in der Auswerteeinheit hinterlegt werden.

Funktionsbeschreibung
Konfiguration der Auswerteeinheit BIS C-60_1

Konfiguration

Die Auswerteeinheit BIS C-60_1 wird von der Steuerung über den Ausgangspuffer konfiguriert. Die Konfigurationsdaten sind innerhalb von 6 Konfigurations-Byte angeordnet, die mit der Befehlskennung 04_{hex} an die Auswerteeinheit BIS C-60_1 gesendet werden (siehe Beispiel 11 auf 46). Mit der Befehlskennung 05_{hex} kann die momentane Gerätekonfiguration ausgelesen werden (siehe Beispiel 12 auf 47).



Zur Eingabe der Konfiguration müssen immer alle 6 Byte in Hex eingegeben werden. Es dürfen nur die genannten Bit verändert werden. Bei einer Änderung der restlichen Bit kann keine Garantie für die richtige Funktion des BIS C-60_1 übernommen werden.

Die Defaultwerte (Werkseinstellung) der 6 Byte sind:

	1. Byte	2. Byte	3. Byte	4. Byte	5. Byte	6. Byte
Hex	00	80	00	82	00	00
Binär	00000000	10000000	00000000	10000010	00000000	00000000
		Bit 1 Bit 2		Bit 7 Bit 8	Bit 2 Bit 5	

Zur Konfiguration dienen:

mit folgenden Funktionen:

- 1. Byte, Bit 2** Codeträgerinitialisierung überwachen
- 1. Byte, Bit 1** Codeträgerspeicher in einen Bereich mit Speichererweiterung und in einen Bereich ohne Speichererweiterung aufteilen
- 2. Byte, Bit 5** Dynamikbetrieb an Schreib-/Lesekopf 1 (Auswirkungen auf die Schreib-/Lesezeiten siehe 48/49)

Bitstatus: 0 = nein
1 = ja

Funktionsbeschreibung

Konfiguration der Auswerteeinheit BIS C-60_1

Konfiguration (Fortsetzung)

Bitstatus: 0 = nein
1 = ja

- | | |
|--|---|
| 4. Byte, Bit 8

4. Byte, Bit 7

4. Byte, Bit 2

5. Byte, Bit 5 | 2. Bitleiste am Ende des Eingangs- und des Ausgangspuffers anordnen

Zustand des digitalen Eingangs in der Bitleiste des Eingangspuffers anzeigen

Reset der Auswerteeinheit BIS C-60_1 über den digitalen Eingang

Dynamikbetrieb an Schreib-/Lesekopf 2
(Auswirkungen auf die Schreib-/Lesezeiten siehe 48/49) |
|--|---|

Funktionsbeschreibung

Codeträger bearbeiten, Lesen und Schreiben

Lesen und Schreiben

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

Ein Lese-/Schreibauftrag hat folgenden Ablauf (siehe Beispiele auf den 37...39):

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 Codeträgers,
 - 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 Codeträger in den Eingangspuffer,
 Schreiben: vom Ausgangspuffer auf den Codeträger.
 Größere Datenmengen werden in Blöcken übertragen (6 Byte mit doppelter Bitleiste, 7 Byte mit einfacher Bitleiste). Dazu wird mit den Toggle-Bits ein Handshake zwischen Steuerung und Auswerteeinheit BIS C-60_1 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

Codeträger bearbeiten, Lesen und Schreiben

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 C-60_1 abgelehnt, wenn sich kein Codeträger im aktiven Bereich des Schreib-/Lesekopfs befindet. Ist die Funktion Dynamikbetrieb konfiguriert, nimmt die Auswerteeinheit den Lese-/Schreibauftrag an und speichert ihn. Wird ein Codeträger erkannt, wird der gespeicherte Auftrag ausgeführt.

Vom Codeträger lesen, mit Programm Gemischter Datenzugriff

Mit der Befehlskennung 21_{Hex} können die Programmsätze, die im Programm hinterlegt sind, vom Codeträger ausgelesen werden. Der Anwender muss genau dokumentieren, welche Daten von wo und mit welcher Anzahl Byte für das gewählte Programm gelesen werden (siehe Beispiel 8 auf S. 43).

Auf Codeträger schreiben, mit Programm Gemischter Datenzugriff

Mit der Befehlskennung 22_{Hex} können die Programmsätze, die im Programm hinterlegt sind, auf dem Codeträger beschrieben werden. Der Anwender muss genau dokumentieren, welche Daten von wo und mit welcher Anzahl Byte für das gewählte Programm gelesen werden (siehe Beispiel 9 auf S. 44).

Funktionsbeschreibung

Codeträger bearbeiten, Gemischter Datenzugriff

Gemischter Datenzugriff

Im EEPROM der Auswerteeinheit BIS C-60_1 können kleine Schreib-/Leseprogramme abgespeichert werden.

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

Es können 10 Programme mit bis zu 25 Anweisungen abgespeichert werden. Jede Programm-anweisung 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 C-60_1 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
Codeträger bearbeiten, Gemischter Datenzugriff

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 7. Beispiel auf den 40...42 dargestellt.

Das Auswechseln des EEPROMS ist auf 61 für BIS C-6001 und auf 72 für BIS C-6021 beschrieben.

Funktionsbeschreibung
Initialisierung überwachen, Speicher splitten

Überwachen der Codeträger-initialisierung
(siehe 2. Beispiel auf 36)

Mit der Befehlskennung 09_{Hex} können Initialisierungsdaten in der Auswerteeinheit gespeichert werden.

Ist die Funktion Überwachung der Codeträgerinitialisierung aktiviert, werden bei einem Schreib-/Lesebefehl zunächst die Initialisierungsdaten auf dem Codeträger mit denen in der Auswerteeinheit verglichen. Stimmen diese nicht überein, wird die Fehlermeldung 17_{Hex} ausgegeben. Wurde der Codeträger noch gar nicht initialisiert, d.h. alle Initialisierungsdaten sind 0, wird die Fehlermeldung 18_{Hex} ausgegeben.

Speicher des Codeträgers splitten
(siehe 3. Beispiel auf 36)

Mit der Befehlskennung 0A_{Hex} kann in der Auswerteeinheit eine Adresse abgespeichert werden, ab der die Daten im Codeträger nicht mehr speicheroptimiert bearbeitet werden.

Bei der Aufteilung des Speichers ist der Bereich mit Speicheroptimierung so groß zu wählen, dass er für die maximal vorgesehene Anzahl von Schreibzyklen ausreicht (Berechnung siehe 9).

Die Aufteilung ist vorteilhaft, um kleine Datenmengen, die häufig geschrieben werden, im speicheroptimierten Bereich des Codeträgers zu bearbeiten. Größere Datenmengen, die selten geschrieben werden, werden im Bereich ohne Speicheroptimierung bearbeitet.

Ist die Funktion Speicher des Codeträgers splitten aktiviert, gilt die Aufteilung des Speichers für jeden Codeträger, der in den Schreib-/Lesebereich der Auswerteeinheit gelangt.

Funktionsbeschreibung

Beispiele für den Protokollablauf

1. Beispiel

Bei Konfiguration
mit doppelter
Bitleiste!

Initialisieren des Codeträgers für die Speicheroptimierung (Codeträger mit 32 Byte Blockgröße)

Steuerung:

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

01 _{Hex}	Befehlskennung 08 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit auf 0 (32 Byte Blockgröße), AV-Bit setzen

- 3.) Subadressen des Ausgangspuffers bearbeiten:

01...05 _{Hex}	Die 5 Initialisierungsbyte eintragen
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

- 5.) Subadressen des Ausgangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen
--------------------------------------	-------------------

Identifikations-System BIS C-60_1:

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

00 _{Hex} /07 _{Hex}	AA-Bit setzen, TO-Bit invertieren
--------------------------------------	-----------------------------------

- 4.) Subadressen des Eingangspuffers bearbeiten:

01...05 _{Hex}	Die 5 Initialisierungsbyte kopieren
00 _{Hex} /07 _{Hex}	AE-Bit setzen

- 6.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit und AE-Bit rücksetzen
--------------------------------------	------------------------------

Funktionsbeschreibung

Beispiele für den Protokollablauf

2. Beispiel

Bei Konfiguration
mit doppelter
Bitleiste!

Initialisierungsdaten des Codeträgers zur Überwachung in die Auswerteeinheit speichern

Steuerung:

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

01 _{Hex}	Befehlskennung 09 _{Hex}
02...05 _{Hex}	2. bis 5. Initialisierungsbyte eintragen
00 _{Hex} /07 _{Hex}	AV-Bit setzen

- 3.) Subadressen des Ausgangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen
--------------------------------------	-------------------

Identifikations-System BIS C-60_1:

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

00 _{Hex} /07 _{Hex}	AA-Bit und TO-Bit setzen
02...05 _{Hex}	2. bis 5. Initialisierungsbyte kopieren
00 _{Hex} /07 _{Hex}	AE-Bit setzen

- 4.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit und AE-Bit rücksetzen
--------------------------------------	------------------------------

3. Beispiel

Bei Konfiguration
mit doppelter
Bitleiste!

Splitadresse in die Auswerteeinheit speichern

Steuerung:

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

01 _{Hex}	Befehlskennung 0A _{Hex}
02...03 _{Hex}	Splitadresse (Low-/Highbyte) eintragen
00 _{Hex} /07 _{Hex}	AV-Bit setzen

- 3.) Subadressen des Ausgangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen
--------------------------------------	-------------------

Identifikations-System BIS C-60_1:

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

00 _{Hex} /07 _{Hex}	AA-Bit und TO-Bit setzen
02...03 _{Hex}	Splitadresse (Low-/Highbyte) kopieren
00 _{Hex} /07 _{Hex}	AE-Bit setzen

- 4.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit und AE-Bit rücksetzen
--------------------------------------	------------------------------

Funktionsbeschreibung

Beispiele für den Protokollablauf

4. Beispiel

Bei Konfiguration
mit doppelter
Bitleiste!

Lesen von 17 Byte ab Codeträgeradresse 10 (Codeträgertyp mit 32 Byte Blockgröße):

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} /07 _{Hex}	CT-Bit auf 0 (32 Byte Blockgröße), AV-Bit setzen

- 3.) Subadressen des Eingangspuffers bearbeiten:

01...06 _{Hex}	Die ersten 6 Byte Daten kopieren
Subadresse des Ausgangspuffers bearbeiten:	
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

- 5.) Subadressen des Eingangspuffers bearbeiten:

01...06 _{Hex}	Die zweiten 6 Byte Daten kopieren
Subadresse des Ausgangspuffers bearbeiten:	
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

- 7.) Subadressen des Eingangspuffers bearbeiten:

01...05 _{Hex}	Die restlichen 5 Byte Daten kopieren
Subadresse des Ausgangspuffers bearbeiten:	
00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen

Identifikations-System BIS C-60_1:

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

00 _{Hex} /07 _{Hex}	AA-Bit setzen
01...06 _{Hex}	Die ersten 6 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	AE-Bit setzen

- 4.) Subadressen des Eingangspuffers bearbeiten:

01...06 _{Hex}	Die zweiten 6 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	TO-Bit invertieren

- 6.) Subadressen des Eingangspuffers bearbeiten:

01...05 _{Hex}	Die restlichen 5 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	TO-Bit invertieren

- 8.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit und AE-Bit rücksetzen
--------------------------------------	------------------------------

Funktionsbeschreibung

Beispiele für den Protokollablauf

5. Beispiel

Bei Konfiguration
mit doppelter
Bitleiste!

Lesen von 30 Byte ab Codeträgeradresse 10 mit Lesefehler (Codeträgertyp mit 64 Byte Blockgröße):

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} /07 _{Hex}	CT-Bit auf 1 (64 Byte Blockgröße), AV-Bit setzen

- 3.) Subadressen des Ausgangspuffers bearbeiten:

01 _{Hex}	Fehlernummer kopieren
Subadresse des Ausgangspuffers bearbeiten:	
00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen

Identifikations-System BIS C-60_1:

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

Wenn Fehler sofort eintritt:

00 _{Hex} /07 _{Hex}	AA-Bit setzen
01 _{Hex}	Fehlernummer eintragen
00 _{Hex} /07 _{Hex}	AF-Bit setzen

- 4.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit und AF-Bit rücksetzen
--------------------------------------	------------------------------

Funktionsbeschreibung
Beispiele für den Protokollablauf

6. Beispiel

**Bei Konfiguration
mit doppelter
Bitleiste!**

Schreiben von 16 Byte ab Codeträgeradresse 20 (Codeträgertyp mit 32 Byte Blockgröße):

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} /07 _{Hex}	CT-Bit auf 0 (32 Byte Blockgröße), AV-Bit setzen

3.) Subadressen des Ausgangspuffers bearbeiten:

01...06 _{Hex}	Die ersten 6 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

5.) Subadressen des Ausgangspuffers bearbeiten:

01...06 _{Hex}	Die zweiten 6 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

7.) Subadressen des Ausgangspuffers bearbeiten:

01...04 _{Hex}	Die restlichen 4 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

9.) Subadressen des Ausgangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen
--------------------------------------	-------------------

Identifikations-System BIS C-60_1:

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

00 _{Hex} /07 _{Hex}	AA-Bit setzen, TO-Bit invertieren
--------------------------------------	-----------------------------------

4.) Subadressen des Eingangspuffers bearbeiten:

01...06 _{Hex}	Die ersten 6 Byte Daten kopieren
------------------------	----------------------------------

Subadresse des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	TO-Bit invertieren
--------------------------------------	--------------------

6.) Subadressen des Eingangspuffers bearbeiten:

01...06 _{Hex}	Die zweiten 6 Byte Daten kopieren
------------------------	-----------------------------------

Subadresse des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	TO-Bit invertieren
--------------------------------------	--------------------

8.) Subadressen des Eingangspuffers bearbeiten:

01...04 _{Hex}	Die restlichen 4 Byte Daten kopieren
------------------------	--------------------------------------

Subadresse des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AE-Bit setzen
--------------------------------------	---------------

10.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit und AE-Bit rücksetzen
--------------------------------------	------------------------------

Funktionsbeschreibung
Beispiele für den Protokollablauf

7. Beispiel
Gemischter
Datenzugriff

**Bei Konfiguration
mit doppelter
Bitleiste!**

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	312	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} /07 _{Hex}	CT-Bit auf 0 oder 1 (je nach Blockgröße), 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}
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

Steuerung:

2.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit setzen, TO-Bit invertieren
--------------------------------------	-----------------------------------

4.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	TO-Bit invertieren
--------------------------------------	--------------------

Fortsetzung siehe nächste 

Funktionsbeschreibung
Beispiele für den Protokollablauf

7. Beispiel
Gemischter
Datenzugriff
(Fortsetzung)

Bei Konfiguration
mit doppelter
Bitleiste!

5.) Subadressen des Ausgangspuffers bearbeiten:

01 _{Hex}	2. Anzahl Byte	(Low Byte) 03 _{Hex}
02 _{Hex}		(High Byte) 00 _{Hex}
03 _{Hex}	3. Anfangsadresse	(Low Byte) 38 _{Hex}
04 _{Hex}		(High Byte) 01 _{Hex}
05 _{Hex}	3. Anzahl Byte	(Low Byte) 11 _{Hex}
06 _{Hex}		(High Byte) 00 _{Hex}
00 _{Hex} /07 _{Hex}	TI-Bit invertieren	

7.) Subadressen des Ausgangspuffers bearbeiten:

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

6.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	TO-Bit invertieren
--------------------------------------	--------------------

8.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	TO-Bit invertieren
--------------------------------------	--------------------

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

Funktionsbeschreibung
Beispiele für den Protokollablauf

7. Beispiel
Gemischter
Datenzugriff
(Fortsetzung)

Bei Konfiguration
mit doppelter
Bitleiste!

35.) 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}	(nicht verwendet)	FF _{Hex} /FF _{Hex}
00 _{Hex} /07 _{Hex}	TI-Bit invertieren	

37.) Subadressen des Ausgangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen
--------------------------------------	-------------------

36.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AE-Bit setzen
--------------------------------------	---------------

38.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{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
Beispiele für den Protokollablauf

8. Beispiel

**Bei Konfiguration
mit doppelter
Bitleiste!**

Lesen des Codeträgers mit Programm Nr. 1 (Codeträgertyp mit 32 Byte Blockgröße):

Steuerung:

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

01 _{Hex}	Befehlskennung 21 _{Hex}
02 _{Hex}	Programmnummer 01 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit auf 0 (32 Byte Blockgröße), AV-Bit setzen

3.) Subadressen des Eingangspuffers bearbeiten:

01...06 _{Hex}	Die ersten 6 Byte Daten kopieren
Subadresse des Ausgangspuffers bearbeiten:	
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

Identifikations-System BIS C-60_1:

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

00 _{Hex} /07 _{Hex}	AA-Bit setzen
01...06 _{Hex}	Die ersten 6 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	AE-Bit setzen

4.) Subadressen des Eingangspuffers bearbeiten:

01...06 _{Hex}	Die zweiten 6 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	TO-Bit invertieren

... Insgesamt werden 27 Byte Daten ausgetauscht.
(Für den weiteren Verlauf siehe Beispiel 4 auf □ 37).



Während das Programm Gemischter Datenzugriff abgearbeitet wird, ist der Dynamikbetrieb ausgeschaltet.

Funktionsbeschreibung
Beispiele für den Protokollablauf

9. Beispiel

**Bei Konfiguration
mit doppelter
Bitleiste!**

Schreiben des Codeträgers mit Programm Nr. 1 (Codeträgertyp mit 32 Byte Blockgröße):

Steuerung:

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

01 _{Hex}	Befehlskennung 22 _{Hex}
02 _{Hex}	Programmnummer 01 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit auf 0 (32 Byte Blockgröße), AV-Bit setzen

3.) Subadressen des Ausgangspuffers bearbeiten:

01...06 _{Hex}	Die ersten 6 Byte Daten eintragen
00 _{Hex} /07 _{Hex}	TI-Bit invertieren

Identifikations-System BIS C-60_1:
2.) Subadressen des Eingangspuffers in der Reihenfolge der Darstellung bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit setzen, TO-Bit invertieren
--------------------------------------	-----------------------------------

4.) Subadressen des Ausgangspuffers bearbeiten:

01...06 _{Hex}	Die ersten 6 Byte Daten kopieren
Subadresse des Eingangspuffers bearbeiten:	
00 _{Hex} /07 _{Hex}	TO-Bit invertieren

... Insgesamt werden 27 Byte Daten ausgetauscht.
Die Bearbeitung der Puffer entspricht dem Beispiel 6 auf □ 39.



Während das Programm Gemischter Datenzugriff abgearbeitet wird, ist der Dynamikbetrieb ausgeschaltet.

Funktionsbeschreibung

Beispiele für den Protokollablauf

10. Beispiel

Bei Konfiguration
mit doppelter
Bitleiste!

Schreib-/Lesekopf des Identifikations-System BIS C-60_1 in den Grundzustand bringen:

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

Steuerung:

1.) Subadressen des Ausgangspuffers bearbeiten:

00 _{hex} /07 _{hex}	GR-Bit setzen
--------------------------------------	---------------

3.) Subadressen des Ausgangspuffers bearbeiten:

00 _{hex} /07 _{hex}	GR-Bit rücksetzen
--------------------------------------	-------------------

Identifikations-System BIS C-60_1:

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

00 _{hex} /07 _{hex}	BB-Bit rücksetzen
--------------------------------------	-------------------

4.) Subadressen des Eingangspuffers bearbeiten:

00 _{hex} /07 _{hex}	BB-Bit setzen
--------------------------------------	---------------

Funktionsbeschreibung

Beispiele für den Protokollablauf

11. Beispiel

Bei Konfiguration
mit doppelter
Bitleiste!

Konfigurationsdaten programmieren:

Das Programmieren von Konfigurationsdaten ist in beiden Puffern – für Kopf 1 und Kopf 2 – möglich.

Steuerung:

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

01 _{hex}	Befehlskennung 04 _{hex}
00 _{hex} /07 _{hex}	AV-Bit setzen

3.) Subadressen des Ausgangspuffers bearbeiten:

01...06 _{hex}	Die 6 Konfigurations-Byte eintragen
00 _{hex} /07 _{hex}	TI-Bit invertieren

5.) Subadressen des Ausgangspuffers bearbeiten:

00 _{hex} /07 _{hex}	AV-Bit rücksetzen
--------------------------------------	-------------------

Identifikations-System BIS C-60_1:

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

00 _{hex} /07 _{hex}	AA-Bit setzen, TO-Bit invertieren
--------------------------------------	-----------------------------------

4.) Subadressen des Ausgangspuffers bearbeiten:

01...06 _{hex}	Die 6 Konfigurations-Byte übernehmen
Subadresse des Eingangspuffers bearbeiten:	

00 _{hex} /07 _{hex}	AE-Bit setzen
--------------------------------------	---------------

6.) Subadressen des Eingangspuffers bearbeiten:

00 _{hex} /07 _{hex}	AA-Bit und AE-Bit rücksetzen
--------------------------------------	------------------------------

Funktionsbeschreibung Beispiele für den Protokollablauf

12. Beispiel

Bei Konfiguration
mit doppelter
Bitleiste!

Programmierte Konfigurationsdaten auslesen:

Steuerung:

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

01 _{Hex}	Befehlskennung 05 _{Hex}
00 _{Hex} /07 _{Hex}	AV-Bit setzen

3.) Subadressen des Eingangspuffers bearbeiten:

01...06 _{Hex}	Die 6 Konfigurations-Bytes kopieren
------------------------	-------------------------------------

Subadresse des Ausgangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen
--------------------------------------	-------------------

Identifikations-System BIS C-60_1:

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

00 _{Hex} /07 _{Hex}	AA-Bit setzen
01...06 _{Hex}	Die 6 Konfigurations-Bytes eintragen
00 _{Hex} /07 _{Hex}	AE-Bit setzen

4.) Subadressen des Ausgangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	AA-Bit und AE-Bit rücksetzen
--------------------------------------	------------------------------

Schreib-/Lesezeiten

**Lesezeiten
vom Codeträger
zur Auswerteeinheit
im statischen
Betrieb**
(Konfiguration
2. Byte, Bit 5 = 0)

Für zweimaliges Lesen und Vergleichen:

Codeträger mit 32 Byte je Block	
Anzahl Byte	Lesezeit [ms]
von 0 bis 31	110
für jeweils weitere angebrochene 32 Byte addieren Sie weitere	120
von 0 bis 255	= 950

Codeträger mit 64 Byte je Block	
Anzahl Byte	Lesezeit [ms]
von 0 bis 63	220
für jeweils weitere angebrochene 64 Byte addieren Sie weitere	230
von 0 bis 2047	= 7350

**Schreibzeiten
von der
Auswerteeinheit
zum Codeträger
im statischen
Betrieb**
(Konfiguration
2. Byte, Bit 5 = 0)

Inclusive Rücklesen und Vergleichen:

Codeträger mit 32 Byte je Block	
Anzahl Byte	Schreibzeit [ms]
von 0 bis 31	110 + n * 10
≥ 32	y * 120 + n * 10

Codeträger mit 64 Byte je Block	
Anzahl Byte	Schreibzeit [ms]
von 0 bis 63	220 + n * 10
≥ 64	y * 230 + n * 10

n = Anzahl der zusammenhängend zu schreibenden Bytes
y = Anzahl der zu bearbeitenden Blöcke

Beispiel: Es sollen 17 Byte ab Adresse 187 geschrieben werden. Codeträger = 32 Byte je Block. Bearbeitet werden Block 5 und 6, da Anfangsadresse 187 in Block 5 und Endadresse 203 in Block 6 ist.

$$t = 2 * 120 + 17 * 10 = 410$$

Die angegebenen Zeiten gelten, nachdem der Codeträger erkannt wurde. Andernfalls müssen für den Energieaufbau bis zum Erkennen des Codeträgers 45 ms hinzugerechnet werden.

Schreib-/Lesezeiten

Lesezeiten vom Codeträger zur Auswerteeinheit im dynamischen Betrieb
(Konfiguration 2. Byte, Bit 5 = 1)

Lesezeiten innerhalb des 1. Blocks für zweimaliges Lesen und Vergleichen:

Die angegebenen Zeiten gelten, nachdem der Codeträger erkannt wurde. Ist der Codeträger noch nicht erkannt, müssen für den Energieaufbau bis zum Erkennen des Codeträgers 45 ms hinzuge-rechnet werden.

Codeträger mit 32 Byte je Block		Codeträger mit 64 Byte je Block	
Anzahl Byte	Lesezeit [ms]	Anzahl Byte	Lesezeit [ms]
von 0 bis 3	= 14	von 0 bis 3	= 14
für jedes weitere Byte	3,5	für jedes weitere Byte	3,5
von 0 bis 31	= 112	von 0 bis 63	= 224

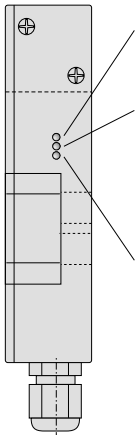
m = größte zu lesende Adresse

Formel: $t = (m + 1) \cdot 3,5 \text{ ms}$

Beispiel: Es sollen 11 Byte ab Adresse 9 gelesen werden. D.h. die größte zu lesende Adresse ist 19. Dies ergibt 70 ms.

Funktionsanzeigen

Betriebszustand BIS C-60_1



Über die drei seitlichen LED meldet die Auswerteeinheit BIS C-60_1 die wichtigsten Betriebs-zustände des Identifikations-Systems.

Betriebszustand	LED	Bedeutung
Ready / Bus active	rot	Betriebsspannung in Ordnung; kein Hardwarefehler, aber Bus nicht aktiv.
	grün	Betriebsspannung in Ordnung; kein Hardwarefehler, Bus aktiv.
CT1 Present / Operating	grün	Codeträger schreib-/lesebereit am Schreib-/Lesekopf 1.
	gelb	Schreib-/Leseauftrag wird bearbeitet am Schreib-/Lesekopf 1.
	gelb blinkt	Kabelbruch zum Schreib-/Lesekopf 1 bzw. nicht angeschlossen.
	aus	Kein Codeträger im Schreib-/Lesebereich am Schreib-/Lesekopf 1.
CT2 Present / Operating	grün	Codeträger schreib-/lesebereit am Schreib-/Lesekopf 2.
	gelb	Schreib-/Leseauftrag wird bearbeitet am Schreib-/Lesekopf 2.
	gelb blinkt	Kabelbruch zum Schreib-/Lesekopf 2 bzw. nicht angeschlossen.
	aus	Kein Codeträger im Schreib-/Lesebereich am Schreib-/Lesekopf 2.

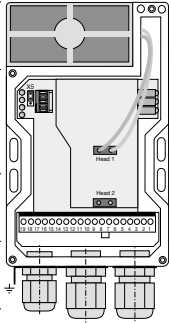
Wenn alle drei LEDs synchron blinken, liegt ein Hardwarefehler vor. Das Gerät muss zur Reparatur.

Funktionsanzeigen

Betriebszustand
INTERBUS

Über 4 Diagnose-LED auf der Grundplatte in Innern der Auswerteeinheit meldet diese die wichtigsten Zustände auf dem INTERBUS.

LED	Zustand	INTERBUS-Meldung
UL	grün / aus	Reset Protokollchip wird / wird nicht mit Spannung versorgt.
RC / CC Remotebus Check Cable Check	grün / aus	Communication Ready Eine Kommunikation zum IBS- Master ist / ist nicht möglich, die Applikation hat den Datenaustausch aber noch nicht gestartet.
BA Bus Active	grün / aus	Communication active oder auch Run Es werden vom Master aktiv / keine Nutzdaten ausgetauscht.
RD Remotebus Disable	gelb / aus	Remotebus Disable Die weiterführende Busschnittstelle ist / ist nicht abgeschaltet.



BIS C-6001
Montage Kopf / Auswerteeinheit

Anordnung des
Schreib-/Lesekopfes
bzw. des Adapters

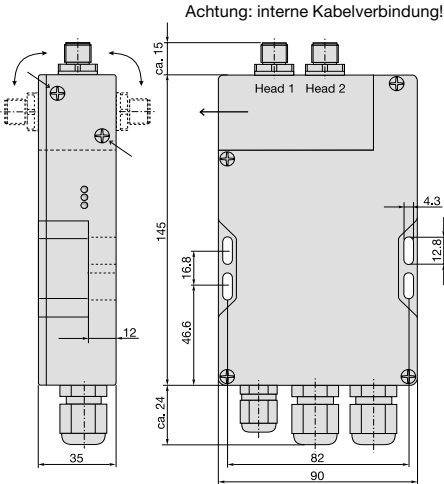
Je nach Ausführung ist die Auswerteeinheit mit einem Schreib-/Lesekopf oder dem Adapter für abgesetzte Schreib-/Leseköpfe ausgestattet. Sowohl der Schreib-/Lesekopf als auch der Adapter können vom Anwender durch Umsetzen um +90° oder -90° in die gewünschte Lage gebracht werden (siehe Bild). Sorgen Sie dafür, dass das Gerät spannungsfrei geschaltet ist. Öffnen Sie die beiden Schrauben (im Bild durch Pfeile gekennzeichnet). Ziehen Sie den Kopf bzw. den Adapter vorsichtig nach der Seite heraus (Pfeilrichtung, rechtes Bild).

Achtung:
interne Kabelverbindung!

Montieren Sie den Kopf/
Adapter in der gewünschten
Lage und schrauben Sie ihn
wieder an.

Montage der
Auswerteeinheit
BIS C-6001

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



BIS C-6001 Öffnen der Auswerteeinheit

Öffnen der Auswerteeinheit BIS C-6001

Um die folgenden Aktionen ausführen zu können, ist die Auswerteeinheit BIS C-6001 zu öffnen:

- Modus für die Kompatibilität einstellen/ändern,
- EEPROM wechseln,
- elektrische Verbindungen (Stromversorgung, Ein-/Ausgang, INTERBUS-Anschlüsse) herstellen.

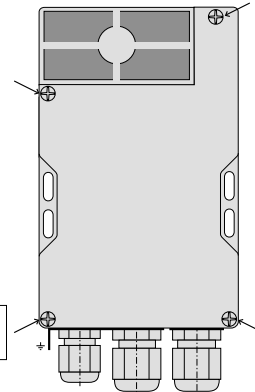


Sorgen Sie vor dem Öffnen dafür, dass das Gerät spannungsfrei geschaltet ist.

Öffnen Sie die 4 Schrauben am BIS C-6001 und entfernen Sie den Deckel.

Führen Sie die gewünschte Aktion aus. Um die elektrischen Verbindungen herzustellen, führen Sie die Kabel durch die Klemmverschraubungen. Weitere Einzelheiten zur Verdrahtung siehe folgende [Tabelle](#).

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



Öffnen der
Auswerteeinheit

BALLUFF (D) 53

BIS C-6001 Montage der Anschlusskabel

Anschlüsse an der Auswerteeinheit BIS C-6001 herstellen

Um die Betriebsspannung, den digitalen Eingang und die INTERBUS-Verbindungen herzustellen, ist die Auswerteeinheit BIS C-6001 zu öffnen (siehe [Tabelle 53](#)).

Sorgen Sie dafür, dass das Gerät spannungsfrei geschaltet ist.

Öffnen Sie die 4 Schrauben am BIS C-6001 und entfernen Sie den Deckel.

Führen Sie die beiden INTERBUS-Kabel durch die Klemmverschraubungen PG 11 (siehe [Tabelle 55](#)). Weitere Einzelheiten zur Verdrahtung siehe folgende [Tabelle](#).

Führen Sie das Kabel für die Betriebsspannung und für den digitalen Eingang durch die Klemmverschraubung PG 9.

Schließen Sie den Deckel der Auswerteeinheit.

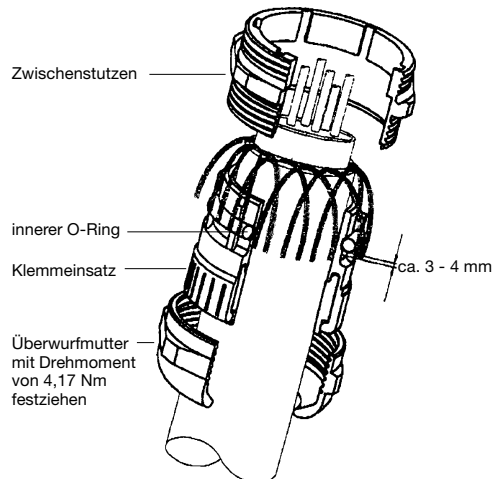
Wenn die Auswerteeinheit mit einem Adapter ausgestattet ist:

- Mit BIS C-650: An den Anschlüssen Head 1 und Head 2 schließen Sie die Schreib-/Leseköpfe an.
- Mit BIS C-670: Am Anschluss Head 1 schließen Sie den Schreib-/Lesekopf an.

BIS C-6001 Montage PG-Verschraubung

Montage der Verschraubungen PG-11 an der Auswerteeinheit BIS C-6001

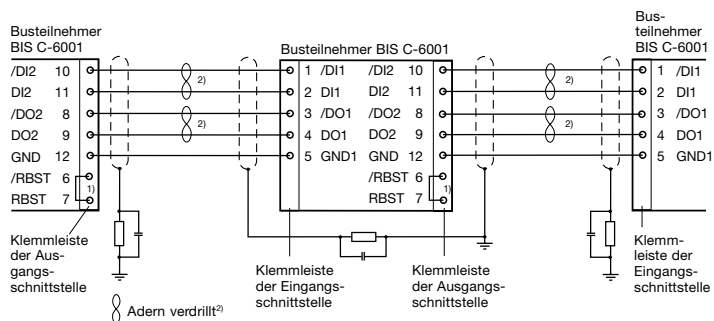
Beim Anschluss der Bus-Leitungen ist darauf zu achten, dass der Schirm eine einwandfreie Verbindung zum PG-Gehäuse hat.



BIS C-6001 Schnittstelleninformationen / Anschlusspläne

Fernbus, Kabel und Schnittstellen am INTERBUS

Um die Auswerteeinheiten BIS C-6001 in den seriellen INTERBUS einzuschleifen, befinden sich auf der Klemmleiste die Anschlüsse 1...5 für die Eingangsschnittstelle und die Anschlüsse 8...12 für die Ausgangsschnittstelle des INTERBUS. Die folgende Zeichnung gibt die Verkabelung wieder, wenn die Auswerteeinheiten BIS C-6001 miteinander verbunden werden sollen.



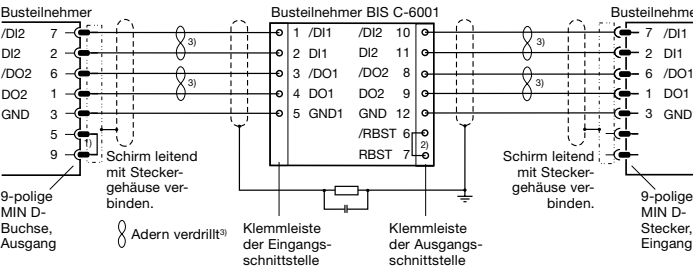
¹⁾ Die Brücke verbleibt im BIS C-6001, wenn ein weiterer Busteilnehmer folgt. Sie ist zu entfernen, wenn kein weiterer Busteilnehmer folgt.

²⁾ Die differentiellen Signale DO und /DO sowie DI und /DI müssen paarweise verdreht sein. Empfohlener Kabeltyp: LiCY 3x2x0,25 mm²; maximale Kabelkapazität: 120 pF/m

BIS C-6001
Schnittstelleninformationen / Anschlusspläne

Fernbus, Kabel und Schnittstellen am INTERBUS

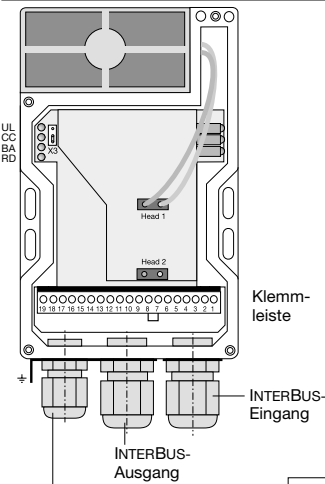
Um die Auswerteeinheiten BIS C-6001 in den seriellen INTERBUS einzuschleifen, befinden sich auf der Klemmleiste die Anschlüsse 1...5 für die Eingangsschnittstelle und die Anschlüsse 8...12 für die Ausgangsschnittstelle des INTERBUS. Die folgende Zeichnung gibt die Verkabelung wieder, wenn die Schnittstelle des BIS C-6001 mit einem 9-poligen Anschluss verbunden werden soll (z.B. mit BIS C-6021).



- 1) Die Brücke ist im Stecker zu verdrahten, wenn ein weiterer Busteilnehmer folgt. Sie ist zu entfernen, wenn kein weiterer Busteilnehmer folgt.
- 2) Die Brücke verbleibt im BIS C-6001, wenn ein weiterer Busteilnehmer folgt. Sie ist zu entfernen, wenn kein weiterer Busteilnehmer folgt.
- 3) Die differentiellen Signale DO und /DO sowie DI und /DI müssen paarweise verdreht sein. Empfohlener Kabeltyp: LiYCY 3x2x0,25 mm²; maximale Kabelkapazität: 120 pF/m

BIS C-6001
Schnittstelleninformationen / Anschlusspläne

Anschlussplan für Auswerteeinheit BIS C-6001 mit integriertem Schreib-/Lesekopf



Belegung der Klemmleiste

5	4	3	2	1		
GND	DO1	/DO1	DI1	/DI1		
InterBus Input						
12	11	10	9	8	7	6
GND	DI2	/DI2	DO2	/DO2	RBST	/RBST
InterBus Output						
19	18	17	16	15	14	13
+ VS	- VS	TxD	RxD	GND	+IN	-IN
Power		RS 232			Digital Input	

Lage und Bezeichnung der Anschlüsse

Stromversorgung und digitaler Eingang

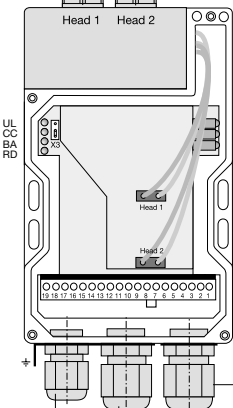
Beim Anschluss der Bus-Leitungen ist darauf zu achten, dass der Schirm eine einwandfreie Verbindung zum PG-Gehäuse hat. Bitte Einbauanleitung auf 55 beachten.

BIS C-6001
Schnittstelleninformationen / Anschlusspläne

Anschlussplan für
Auswerteeinheiten
BIS C-6001 mit
Adapter BIS C-650

Anschluss für Schreib-/Lesekopf 1

Anschluss für Schreib-/Lesekopf 2



Lage und Bezeichnung
der Anschlüsse

Stromversorgung und digit. Eingang

Belegung der
Klemmleiste

5	4	3	2	1
GND	DO1	/DO1	DI1	/DI1
INTERBUS Input				

12	11	10	9	8	7	6
GND	DI2	/DI2	DO2	/DO2	RBST	/RBST
INTERBUS Output						

19	18	17	16	15	14	13
+ VS	- VS	TxD	RxD	GND	+IN	-IN
Power		RS 232		Digital Input		

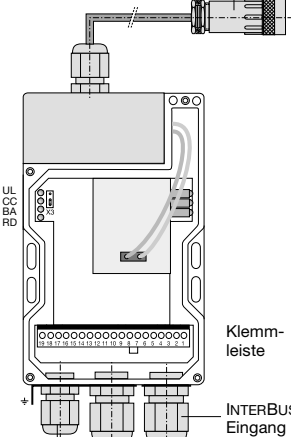
Beim Anschluss der Bus-Leitungen ist darauf zu achten,
dass der Schirm eine einwandfreie Verbindung zum PG-
Gehäuse hat. Bitte Einbauanleitung auf 55 beachten.

BIS C-6001
Schnittstelleninformationen / Anschlusspläne

Anschlussplan für
Auswerteeinheiten
BIS C-6001 mit
Adapter BIS C-670

Anschluss für Schreib-/Lesekopf, 8-polig

Belegung der
Klemmleiste



Lage und Bezeichnung
der Anschlüsse

Stromversorgung und digitaler Eingang

5	4	3	2	1
GND	DO1	/DO1	DI1	/DI1
INTERBUS Input				

12	11	10	9	8	7	6
GND	DI2	/DI2	DO2	/DO2	RBST	/RBST
INTERBUS Output						

19	18	17	16	15	14	13
+ VS	- VS	TxD	RxD	GND	+IN	-IN
Power		RS 232		Digital Input		

Beim Anschluss der Bus-Leitungen ist
darauf zu achten, dass der Schirm eine
einwandfreie Verbindung zum PG-Gehäuse
hat. Bitte Einbauanleitung auf 55 beachten.

BIS C-6001
Wechseln des EEPROM

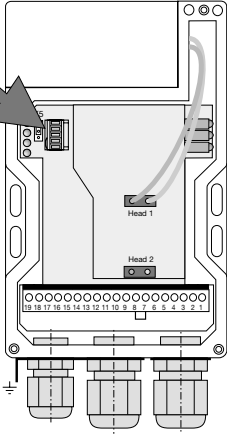
**EEPROM in der
Auswerteeinheit
BIS C-6001
wechseln**



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.

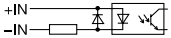
Der Wechsel des EEPROM geschieht durch Aus- und Einstecken.



Lage des EEPROM

BIS C-6001
Technische Daten

Abmessungen, Gewicht	Gehäuse	Kunststoff PS
	Abmessungen mit Schreib-/Lesekopf BIS C-652 Abmessungen mit Adapter BIS C-650 Gewicht	ca. 169 x 90 x 35 mm ca. 184 x 90 x 35 mm ca. 400 g
Betriebs- bedingungen	Umgebungstemperatur	0 °C bis +60 °C
Anschlussart	Klemmleiste	19-polig
	Kabeleinführung für Betriebsspannung für INTERBUS, Ein-/Ausgang Kabeldurchmesser	1 x Klemmkorb PG 9 (Metall) 2 x Klemmkorb PG 11 (Metall) 4 bis 8 mm bei PG 9 5 bis 10 mm bei PG 11
	Leitergrößen	0,14 bis 1 mm ²
	mit Adernhülsen	0,25 bis 0,34 mm ²
Schutzart	Schutzart	IP 65 (in angeschlossenem Zustand)
Elektrische Anschlüsse	Betriebsspannung V_s, Eingang	DC 24 V ± 20 %
	Restwelligkeit Stromaufnahme	≤ 10 % ≤ 400 mA
	Busschnittstelle	INTERBUS Fernbusteilnehmer, Ident-Nr.: 03, 16 Byte IN, 16 Byte OUT
	Digitaler Eingang (+IN, -IN)	über Optokoppler galvanisch 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



BIS C-6001
Technische Daten

Elektrische Anschlüsse (Fortsetzung)	Service-Schnittstelle Schreib-/Lesekopf* alternativ bei montiertem Adapter BIS C-650*		RS 232 integriert, BIS C-65_ und folgende; 2 x Einbaustecker 4-polig (Stift) für alle Schreib-/Leseköpfe BIS C-3_ _ mit 4-poligem Stecker (Buchse), nicht BIS C-350 und BIS C-352 1 Einbaustecker 8-polig (Stift) für die Schreib-/Leseköpfe BIS C-350 / BIS C-352
	alternativ bei montiertem Adapter BIS C-670*		
Funktionsanzeigen	BIS-Betriebszustände (LED im Gehäuse)	LED rot / grün LED grün / gelb LED grün / gelb	Ready / Bus active CT1 Present / Operating CT2 Present / Operating
	Zustand INTERBUS (LED seitlich am Gehäuse)	LED grün LED grün LED grün LED gelb	Reset Cable Check Bus active Remotebus Disable



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 55011 (Emission), EN 61000-4-2/3/4/5/6 (Störfestigkeit) erfüllen.

* um $\pm 90^\circ$ umsetzbar

BIS C-6001
Bestellinformationen

Typenschlüssel

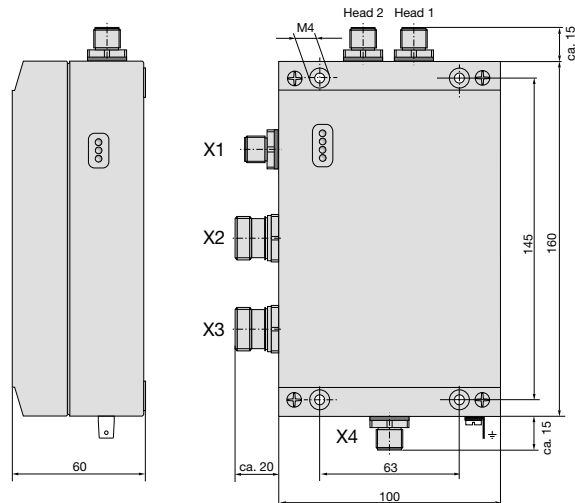
BIS C-6001-029- _ _ -03-KL2

Balluff Identifikations-System	_____
Baureihe C Schreib-/Lesesystem	_____
Hardware-Typ	_____
6001 = INTERBUS (Fernbus)	_____
Software-Typ	_____
029 = INTERBUS mit Speicheroptimierung	_____
Schreib-/Lesekopf	_____
000 = ohne Schreib-/Lesekopf	_____
651 = Schreib-/Lesekopf Typ 651 (mit Rundantenne stirnseitig)	_____
652 = Schreib-/Lesekopf Typ 652 (mit Rundantenne frontseitig)	_____
653 = Schreib-/Lesekopf Typ 653 (mit Stabantenne)	_____
650 = Adapter mit zwei Anschlüssen für externe Schreib-/Leseköpfe BIS C-3_ _ (ausgenommen BIS C-350 und BIS C-352)	_____
670 = Adapter mit einem abgesetztem Anschlussstecker für einen externen Schreib-/Lesekopf BIS C-350 oder BIS C-352	_____
Schnittstelle	_____
Bus-Varianten	_____
Kundenanschluss	_____
KL2 Klemmenanschluss über 1 x PG 9 und 2 x PG 11	_____

BIS C-6021
Montage Auswerteeinheit

Montage der Auswerteeinheit BIS C-6021

Die Auswerteeinheit wird mit 4 Schrauben M4 befestigt.



BALLUFF (D) 65

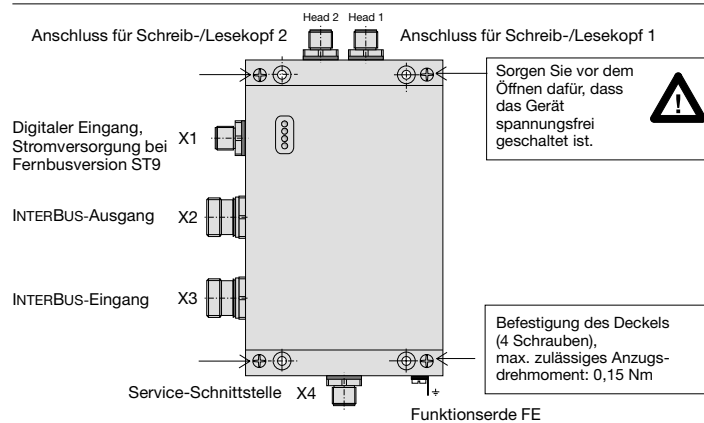
BIS C-6021
Öffnen der Auswerteeinheit / Schnittstelleninformation

**Öffnen der
Auswerteeinheit
BIS C-6021**

Um den Modus für die Kompatibilität einzustellen oder das EEPROM zu wechseln, ist die Auswerteeinheit zu öffnen.

Öffnen Sie die 4 Schrauben am BIS C-6021 und entfernen Sie den Deckel. Weitere Einzelheiten siehe folgende .

Anschlussplan für Auswerteeinheit BIS C-6021



*Lage und Bezeichnung
der Anschlüsse*

BIS C-6021 Schnittstelleninformationen

Anschluss am Fernbus oder Installationsfernbus

Um die Verbindungen für den INTERBUS, die Betriebsspannung und den digitalen Eingang herzustellen, sind die konfektionierten Kabel an der Auswerteeinheit anzuschließen. Weitere Einzelheiten zur Verdrahtung siehe folgende Tabelle.
An den Anschlüssen Head 1 und Head 2 schließen Sie die Schreib-/Leseköpfe an.

Die Auswerteeinheit BIS C-6021...ST8 ist für den Einsatz am Installationsfernbus vorgesehen, der die Versorgungsspannung auf dem Bus bereitstellt. Die Auswerteeinheit BIS C-6021...ST9 ist für den Einsatz am Fernbus vorgesehen, hier wird die Versorgungsspannung für die Auswerteeinheit an X1 angeschlossen.

Schließen Sie das ankommende INTERBUS-Kabel an den INTERBUS-Eingang X3 an. Schließen Sie das abgehende INTERBUS-Kabel an den INTERBUS-Ausgang X2 an.

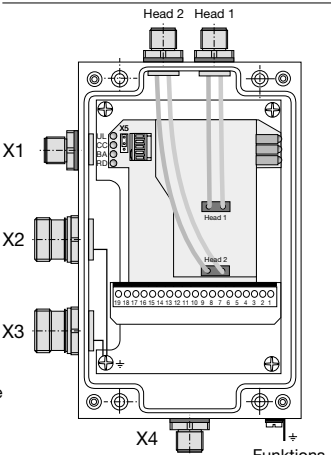
Wenn dieser Fernbusteilnehmer als letzter am Bus angeschlossen ist, muss der INTERBUS-Ausgang X2 mit einer Verschlusskappe verschraubt werden, um die Schutzart zu gewährleisten.



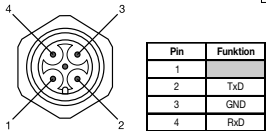
Bitte beachten Sie die Belastbarkeit des INTERBUS-Kabels und überprüfen Sie im Betrieb, ob die Betriebsspannung an der Auswerteeinheit eingehalten wird (Werte siehe Technische Daten).

BIS C-6021...ST8 Anschlussplan (Installationsfernbus)

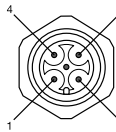
Anschlussplan für Auswerteeinheit BIS C-6021...ST8 am Installationsfernbus



X4, Service-Schnittstelle

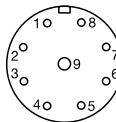


X1, Digitaler Eingang



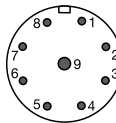
Pin	Funktion
1	nicht angeschlossen
3	nicht angeschlossen
2	-IN
4	+IN

X2, INTERBUS-Ausgang



Pin	Funktion
1	DO2
2	/DO2
3	DI2
4	/DI2
5	GND
9	/RBST
6	FE
7	+24 V
8	0 V

X3, INTERBUS-Eingang

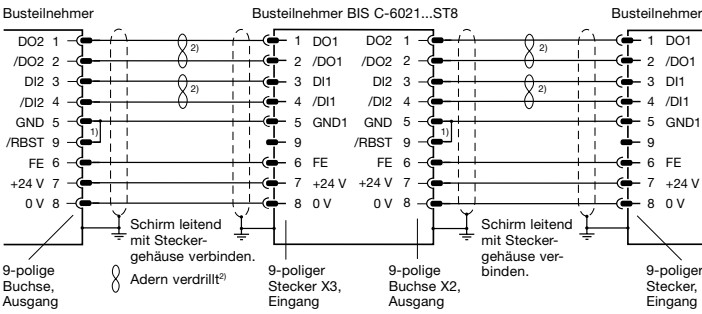


Pin	Funktion
1	DO1
2	/DO1
3	DI1
4	/DI1
5	GND1
6	FE
7	+24 V
8	0 V

BIS C-6021...ST8 Anschlussplan (Installationsfernbus)

Anschlussplan für Auswerteeinheit BIS C-6021...ST8 am Installationsfernbus (Fortsetzung)

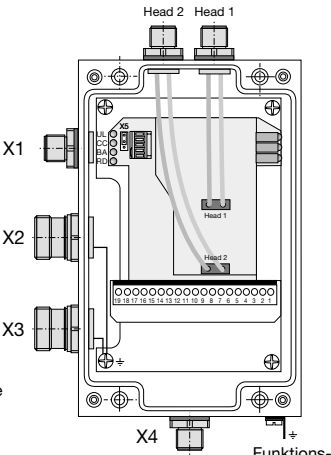
Um die Auswerteeinheiten BIS C-6021 in den seriellen INTERBUS einzuschleifen, befinden sich am Gehäuse der Anschluss X2 als INTERBUS-Ausgang und der Anschluss X3 als INTERBUS-Eingang. Beim Installationsfernbus wird die Versorgungsspannung für die Busteilnehmer auf dem Bus mitgeführt.



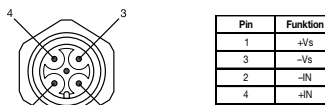
- ¹⁾ Die Brücke ist im Stecker zu verdrahten, wenn ein weiterer Busteilnehmer folgt. Sie ist zu entfernen, wenn kein weiterer Busteilnehmer folgt.
²⁾ Die differentiellen Signale DO und /DO sowie DI und /DI müssen paarweise verdreht sein.
Empfohlener Kabeltyp: LiYCY 3x2x0,25 mm²; maximale Kabelkapazität: 120 pF/m

BIS C-6021...ST9 Anschlussplan (Fernbus)

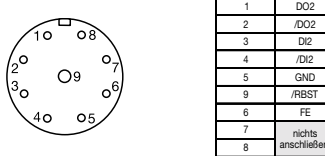
Anschlussplan für Auswerteeinheit BIS C-6021...ST9 am Fernbus



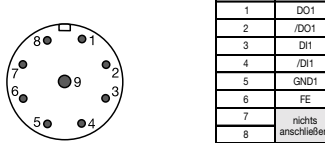
X1, Stromversorgung und digitaler Eingang



X2, INTERBUS-Ausgang



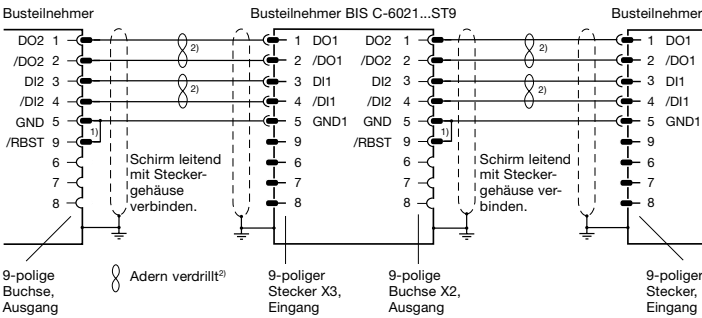
X3, INTERBUS-Eingang



BIS C-6021...ST9 Anschlussplan (Fernbus)

Anschlussplan für Auswerteeinheit BIS C-6021...ST9 am Fernbus (Fortsetzung)

Um die Auswerteeinheiten BIS C-6021 in den seriellen INTERBUS einzuschleifen, befinden sich am Gehäuse der Anschluss X2 als INTERBUS-Ausgang und der Anschluss X3 als INTERBUS-Eingang. Beim Fernbus wird keine Versorgungsspannung für die Busteilnehmer auf dem Bus mitgeführt.



- ¹⁾ Die Brücke ist im Stecker zu verdrahten, wenn ein weiterer Busteilnehmer folgt. Sie ist zu entfernen, wenn kein weiterer Busteilnehmer folgt.
- ²⁾ Die differentiellen Signale DO und /DO sowie DI und /DI müssen paarweise verdreht sein. Empfohlener Kabeltyp: LiYCY 3x2x0,25 mm²; maximale Kabelkapazität: 120 pF/m

BIS C-6021 Wechseln des EEPROM

EEPROM in der Auswerteeinheit BIS C-6021 wechseln

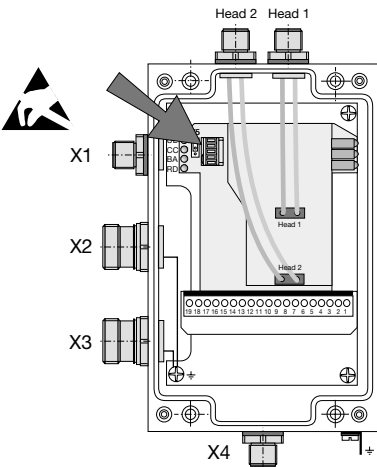
Um das EEPROM zu wechseln, ist die Auswerteeinheit entsprechen der Angaben auf 66 zu öffnen.



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.

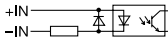


Lage des EEPROM

BIS C-6021
Technische Daten

Abmessungen, Gewicht	Gehäuse	Metall
	Abmessungen	190 x 120 x 60 mm
	Gewicht	820 g
Betriebsbedingungen	Umgebungstemperatur	0 °C bis +60 °C
Anschlussart	Einbaustecker X1	5-polig (Stift)
	Einbaustecker Head 1, Head 2	4-polig (Stift)
	Rundsteckverbinder X2	9-polig (Buchse)
	Rundsteckverbinder X3	9-polig (Stift)
	Einbaustecker X4	4-polig (Stift)
Schutzart	Schutzart	IP 65 (in angeschlossenem Zustand)
Elektrische Anschlüsse	Betriebsspannung V_s	DC 24 V $\pm$ 20 %
	Restwelligkeit	$\leq$ 10 %
	Stromaufnahme	$\leq$ 400 mA
	Anschluss der Betriebsspannung V_s bei Installationsfernbus bei Fernbus	über INTERBUS Eingang X3, Ausgang X2 über Stecker X1
	INTERBUS Ausgang X2, Eingang X3	serielle Schnittstelle für Fernbusteilnehmer, Ident-Nr.: 03, 16 Byte IN, 16 Byte OUT (bei Modus BIS C-621: 8 Byte IN, 8 Byte OUT)
	Head 1, Head 2, Schreib-/Lesekopf	über 2 x Einbaustecker 4-polig (Stift) für alle Schreib-/Leseköpfe BIS C-3_ _ mit 4-poligem Stecker (Buchse), nicht BIS C-350 und BIS C-352

BIS C-6021
Technische Daten

Elektrische Anschlüsse (Fortsetzung)	Digitaler Eingang X1 (+IN, -IN)	über Optokoppler galvanisch getrennt 4 V bis 40 V 1,5 V bis -40 V 11 mA 5 ms	
	Steuerspannung aktiv Steuerspannung inaktiv Eingangsstrom bei 24 V Verzögerungszeit typisch		
	Service-Schnittstelle X4	RS 232	
Funktionsanzeigen	BIS-Betriebszustände (LED im Gehäuse)	LED rot / grün LED grün / gelb LED grün / gelb	Ready / Bus active CT1 Present / Operating CT2 Present / Operating
	Zustand INTERBUS (LED seitlich am Gehäuse)	LED grün LED grün LED grün LED gelb	Reset Cable Check Bus active Remotebus Disable



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 55011 (Emission), EN 61000-4-2/3/4/5/6 (Störfestigkeit) erfüllen.

BIS C-6021 Bestellinformationen

Typenschlüssel

BIS C-6021-029-050-03-ST

Balluff Identifikations-System
Baureihe C Schreib-/Lesesystem
Hardware-Typ
6021 = Metallgehäuse, INTERBUS
Software-Typ
029 = INTERBUS mit Speicheroptimierung
Adapter
050 = mit zwei Anschlüssen für externe Schreib-/Leseköpfe BIS C-3_ _
(ausgenommen BIS C-350 und -352)
Schnittstelle
03 = BUS-Varianten
Kundenanschluss
ST8 = Steckervariante Installationsfernbus
ST9 = Steckervariante Fernbus
(2 Rundsteckverbinder für Stromversorgung/digitaler Eingang und Service-Schnittstelle,
2 Rundsteckverbinder für INTERBUS)

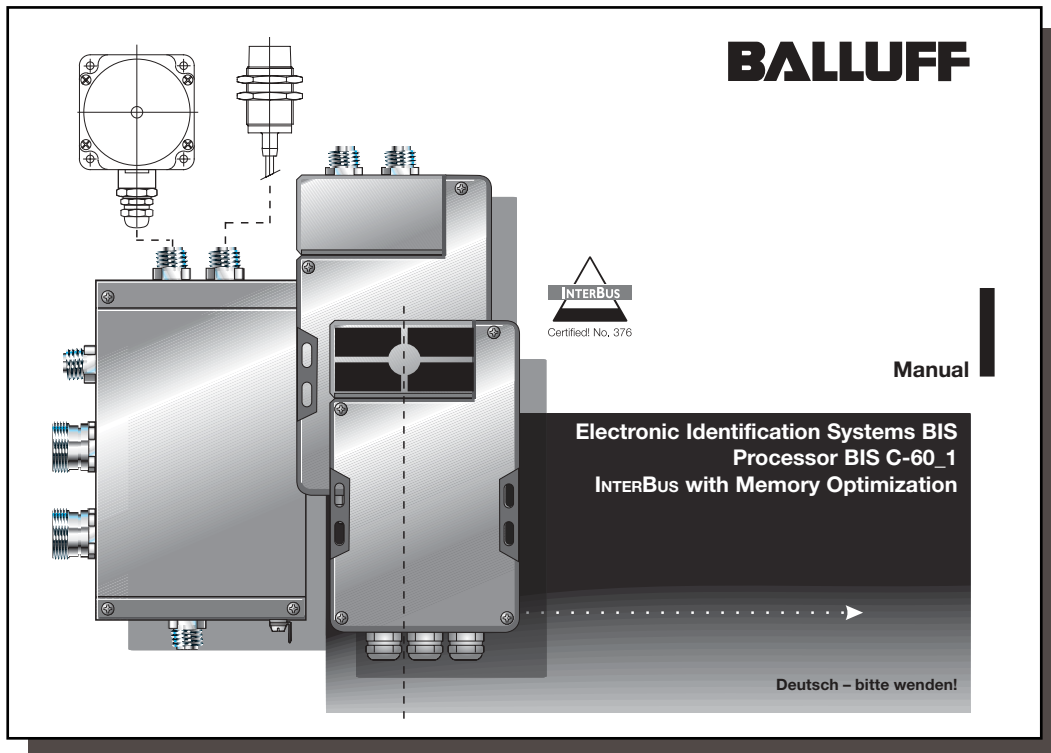
Zubehör (optional, nicht im Lieferumfang)

Artikel	Bestellbezeichnung
Gegenstecker zu X1	BKS-S79-00
zu X2	BKS-S83-00
zu X3	BKS-S84-00
Gegenstecker zu X4	BKS-S 10-3
Verschlusskappe für X1, Head 1, Head 2, X4	BES 12-SM-2
Verschlusskappe für X2	115 475

Anhang, ASCII-Tabelle

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

1



2

No. 823 188 D/E • Edition 0108
Subject to modification.

<http://www.balluff.de>

Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Germany
Phone +49 (0) 71 58/1 73-0
Fax +49 (0) 71 58/50 10
E-Mail: balluff@balluff.de

Contents

Safety Considerations	4
Introduction BIS C-60_1 Identification System	5-7
Application BIS-C60_1 Processor	8
Basic knowledge for application	8
Optimized code tag memory management	9-11
BUS interface: InterBus	12/13
Compatibility with BIS C-6_1 processor	14
Function Description:	15
Communication with the processor	15
In- and output buffer on INTERBUS	16/17
Output buffer, configuration and explanation	18-22
Input buffer, configuration and explanation	23-26
Configuring the BIS C-60_1 processor	27-29
Processing code tags	30-33
Reading and writing	30
Reading and writing in dynamic mode	31
Reading and writing with program	31
Mixed Data Access	32/33
Monitoring code tag initialization	34
Splitting code tag memory	34
Examples for protocol sequence	35-47
Read/Write Times	48/49
LED Display	50/51
	BIS C-6001 BIS C-6021
Mounting Head / Processor	52
Opening the processor / Interface information	53
Installing the connection cables / Mounting the PG connection	54/55
Interface information / Wiring diagrams	56-60
Changing the EEPROM	61
Technical Data	62/63
Ordering information	64
Appendix, ASCII Table	75
	76

Safety Considerations

Approved operation	Series BIS C-60_1 processors along with the other BIS C system components comprise an identification system and may only be used for this purpose in an industrial environment in conformity with Class A of the EMC Law.
Installation and Operation	Installation and operation should be carried out by technically trained personnel only. Unauthorized access and improper use will lead to loss of warranty and liability claims. When installing the processor, consult the section on wiring diagrams carefully. Special caution must be used when wiring the processor to external controllers, particularly with respect to selection and polarity of the signals and power supply. Only approved power supplies may be used with the processor. See the section on Technical Data for details.
Use and Checking	The relevant safety procedures must be followed when using the Identification System. In particular, steps must be taken to ensure that no danger to persons or equipment can arise should a fault occur in the Identification System. This includes maintaining the published ambient operating conditions and regular checking of the functionality of the Identification System with all its associated components.
Fault Conditions	As soon as there is evidence that the Identification System is not functioning properly, it should be taken out of service and protected against unauthorized use.
Scope	This description is valid for series BIS C-6001-029...03-KL2 processors and both the ST8 and ST9 versions of series BIS C-6021-029-050-03-ST_.

INTERBUS is a registered trademark of the Phoenix Corporation.

Introduction BIS C-60_1 Identification System

This manual is designed to assist the user in setting up the control program and installing and starting up the components of the BIS C-60_1 Identification System, and to assure rapid, trouble-free operation.

Principles

The BIS C-60_1 Identification System belongs in the category of

non-contact systems for reading and writing.

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



If 2 read/write heads are connected to a BIS C-60_1 processor, both heads can be operated independently of each other. This means for example that you can read a code tag from one head while writing to another code tag 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 tool coding and monitoring;**
- **in equipment organization;**
- **in storage systems for monitoring inventory movement;**
- **in transporting and conveying systems;**
- **in waste management for quantity-based fee assessment.**

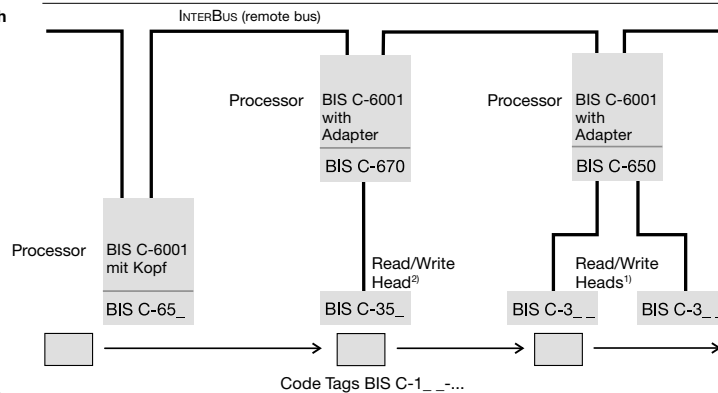
Introduction BIS C Identification System

System Components

The main components of the BIS C Identification System are

- **Processor,**
- **Read/Write Heads and**
- **Code Tags.**

Configuration with BIS C-6001 processor



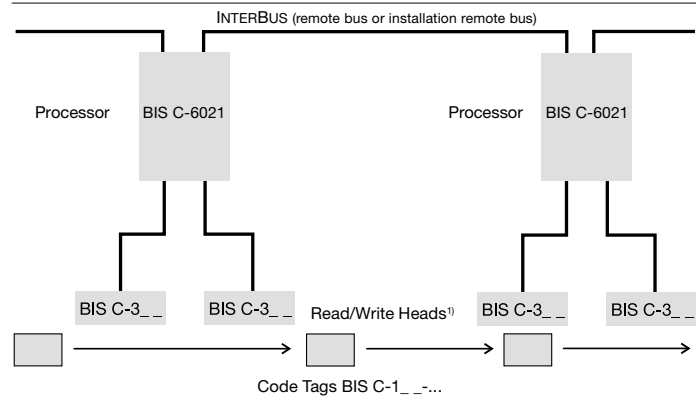
Schematic
representation of an
Identification System
(example)

¹⁾ BIS C-3_ _ series, except BIS C-350 and -352

²⁾ BIS C-350 or -352 only

Introduction BIS C Identification System

Configuration with
BIS C-6021
processor



Schematic
representation of an
Identification System
(example)

¹⁾ BIS C-3_ _ series, except BIS C-350 and -352

Application BIS-C60_1 Processor

Selecting System Components

The **BIS C-6001** processor has a plastic housing. Connections are made through a terminal strip, with the cables secured by PG fittings. A single read/write head from BIS C-65_ _ series can be directly mounted to the processor, which creates a compact unit. If the BIS C-650 adapter is attached instead of the BIS C-65_ _ read/write head, two read/write heads may be cable connected. If the BIS C-670 adapter is attached, one read/write head may be cable connected.

The **BIS C-6021** processor has a metal housing. Connection is made through round connectors. Two read/write heads can be cable connected to the BIS C-6021 processor.

Series BIS C-60_1 processors have in addition a digital input. The input has various functions depending on the configuration (see Configuration).

Whether the compact version of the processor with integrated read/write head makes sense or whether the external solution is preferred depends primarily on the spatial arrangement of the components. There are no functional limitations. All read/write heads are suitable for both static and dynamic reading. Distance and relative velocity are based on which code tag is selected. Additional information on the read/write heads in series BIS C-65_ _ and series BIS C-3_ _ including all the possible code tag/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 code tag represents an 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.

BIS C-60_1 Processor
Optimized code tag memory management

Protocol

The BIS C-60_1-029 version works with a further developed Balluff protocol, which optimizes memory handling in the code tag. This version would be required whenever the required number of write cycles exceeds the number permitted for the EEPROM in the code tag. Depending on type, this will lie between 100,000 and 1,000,000 cycles and is indicated on the data sheet for each respective code tag. You must select a code tag whose memory size is a multiple of the number of bytes actually used for writing.

For any given code tag the maximum number of write cycles permitted can be calculated as follows:

$$W = W_{CT} \cdot \frac{K_{CT} - 5}{n + 4}$$

K_{CT} = Memory size of code tag in bytes
 W_{CT} = maximum number of write cycles for code tag per data sheet
 n = maximum number of bytes for read/write commands

Using this intelligent method of memory handling, the number of write cycles is stored in a counter. When the permitted number is reached, the previously used memory range is blocked and a new memory range is used. This process is continued until the last memory range is used up. During the last 1,000 write cycles the processor acknowledges all successful write operations with a 'pre-warning' (56Hex in subaddress 02Hex of the input buffer).

When the maximum number of write cycles for the last memory range is reached, writing to this memory range continues but an additional 'final warning' is indicated (45Hex in subaddress 02Hex of the input buffer).

BIS C-60_1 Processor
Optimized code tag memory management

Initializing the code tag

In order to use a code tag for memory optimization, the code tag must first be initialized. This can be done either with the Handy Terminal BIS C-800, or from a PC station with a built-in BIS C-480-007-PC card, or using the initializing command of BIS C-60_1-029.

The first 5 bytes of the memory are used for the designator:

Byte No.	Initialization (hexadecimal)	Meaning / Function
0	00H	Number of currently used memory range on code tag
1	01H	128 Byte Enter memory range of code tag
	02H	256 Byte
	03H	512 Byte
	04H	1 024 Byte
	05H	2 048 Byte
	06H	4 096 Byte
	07H	8 192 Byte
	08H	16 384 Byte
	09H	127 Byte
	0AH	255 Byte
	0BH	511 Byte
2	0CH	1 023 Byte
	0DH	2 047 Byte
	01H	100,000 Enter maximum number of write cycles of the code tag
	02H	200,000 (see data sheet)
	03H	300,000
	04H	400,000
3 and 4	05H	500,000
	0AH	1,000,000
	maximum 0400H	maximum number of bytes per read/write cycle = 1 kByte (segment size)

BIS C-60_1 Processor Optimized code tag memory management

Initializing the code tag (continued)

An address named in a telegram does not distinguish between code tags with and without initialization.

Example of an initialization:

On a code tag with 128 byte memory size, 24 bytes shall be used for reading and writing. The maximum permissible number of write cycles according to the code tag data sheet is 100,000. The result is the following initialization of the first 9 bytes of memory:

Number of the 1st memory range 00_H 01_H 01_H 00_H 18_H 00_H 00_H 00_H 00_H
 Memory size of code tag 128 bytes |
 Permissible no. of write cycles 100,000 |
 Maximum no. of bytes per read/write cycle set to 24 bytes |
 Counter of 1st field set to 0 |

The ratio of memory size to memory requirement thus permits 400,000 write cycles, since the available memory size can be divided into 4 memory ranges of 100,000 bytes each. The pre-warning is given after the 399,000th write cycle.

Control Function

The processor writes data from the host system to the code tag 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)

BUS interface: INTERBUS

INTERBUS

Communication between the BIS C-60_1 processor and the host system is via INTERBUS.

The INTERBUS system consists of three components:

- the wiring module (rack card for industrial PC or PLC),
- Bus terminal as network node and/or
- the I/O modules (here the BIS C-60_1 processor).

Depending on the wiring module up to a maximum of 63 BIS C-60_1 processors can be connected.

The BIS C-6001 processor is used as a remote bus station. The BIS C-6021 can be used as a remote bus or installation bus station.



Important hints for use with PLC:

In some control systems the INTERBUS data area is not synchronously transmitted with the updating of the input/output content. If more than 2 bytes of data are sent, a mechanism must be used which guarantees that the data in the PLC and the data in the BIS C are always identical!

2nd alternative: Set 2nd bit header

Data exchange between PLC and BIS is controlled by the so-called bit header. This is always the first byte of the respective read/write head in the data buffer. This bit header exists both in the input range (data from BIS to the PLC) and in the output range (data from the PLC to the BIS). If this bit header is also sent as the last byte, a comparison of these two bytes can be used to guarantee the consistency of the transmitted data.

In this method the PLC cycle is unaffected nor is the bus access time changed. All that is required is that a byte in the data buffer be used for the 2nd bit header instead of for user data.

This 2nd alternative is the Balluff recommended setting (factory default).

BUS interface: INTERBUS

Address setting is done on the module (not on the I/O modules), i.e. not on the BIS C-6_1 processor). There are two types of addressing possible:

1. logical addressing, and
2. physical addressing.

Logical Addressing

Logical addressing permits free addressing of each module.

Advantage: high security and flexibility;
Disadvantage: more difficult at setup.

For the BIS C-60_1 Identification System use:

I/O Module Type	IDENT-No.	IN-Address (Byte)	OUT-Address (Byte)
Processor BIS C-60_1	03	16	16

Physical Addressing

Physical addressing is rigidly fixed to the system configuration. The address of each module depends on its location in the system.

Advantage: easy to configure at setup;
Disadvantage: changes in module location when power was off are recognized upon initialization, but are not made known to the user.

Compatibility with BIS C-6_1 processor

Setting compatibility

Compatibility with the BIS C-6_1 processors is established using terminal X5 and a jumper.



If the BIS C-60_1 processor is set to be compatible with the BIS C-601 or BIS C-621, all settings for data exchange must be made as described in the sections on parametering, function description, protocol sequence and LED display in the user's manual for the BIS C-6_1 processor! This user's manual can be mailed on request, or you may download it from the Internet at www.balluff.de.

To open the cover of the BIS C-6001 processor, see  59, and for BIS C-6021 see  72.

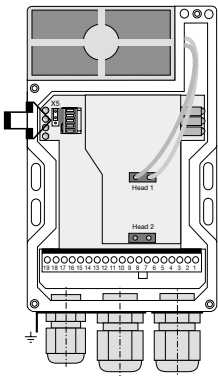
Jumper setting Terminal X5	Processor compatible with BIS C-6_1
1-2	no
2-3	yes



In the illustration compatibility with the BIS C-6_1 is not set.



In "Compatible with BIS C-6_1" mode, the error code also has the error number 11_{Hex} added. This error number indicates that a command used in this mode does not work.



Terminal X5
(with cover removed)

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 control to 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 representation of the sequence of a job sent from the control to the processor:

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 (AE 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.

Function description

In- and output buffer on INTERBUS

Input and Output Buffers

To transmit the control bit, commands and data between the BIS C-60_1 processor and the host system, the latter must provide two fields. These two fields are:

- **the output buffer**
For the control bit (bit header) and controller commands sent to the BIS identification system,
For the data to be written, and
For configuring the BIS C-60_1 processor
- **the input buffer**
For the control bit (bit header) of the BIS C-60_1 processor,
For the data to be read,
For the ID's and error codes coming from the BIS identification system, and
For reading out the configuration data.

The total buffer size of the BIS C-60_1 is 16 bytes for the input buffer and 16 bytes for the output buffer. This total buffer size is divided into 2 sectors:

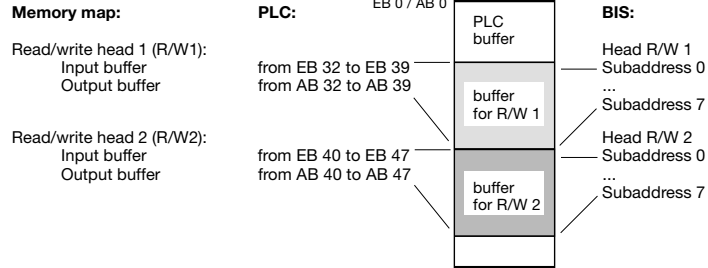
Buffer sector 1 for Read/Write Head 1: 8 bytes input buffer, 8 bytes output buffer
Buffer sector 2 for Read/Write Head 2: 8 bytes input buffer, 8 bytes output buffer

Please note the basic procedure on [p. 15](#) and [p. 30...34](#) and the examples on [p. 35...47](#).

Function description In- and output buffer on INTERBUS

Input and Output Buffers (continued)

Example: Using a PLC, the buffer sector for the BIS C-60_1 will start at input byte EB32 and output byte AB 32.



Please note the basic procedure on 15 and 30...34 and the examples on pages 35...47.

Note that these buffers can be in two different sequences depending on the type of control.

The following description is based on sequence 1!

	Sequence 1	Sequence 2
Subaddress	00	01
	01	00
	02	03
	03	02
	04	05
	05	04
	06	07
	07	06

Function Description Output buffer, configuration and explanation

Configuration of the output buffer for one (1) read/write head

Subaddress	Bit No.	7	6	5	4	3	2	1	0	Bit Name
00Hex = Bit Header		CT	TI				GR		AV	
01Hex		Command Designator				Data	or	Config. 1st byte		
02Hex		Start Address (Low Byte)				Data	or	Config. 2nd Byte		
03Hex		Start Address (High Byte)				Data	or	Config. 3rd Byte		
04Hex		No. of Bytes (Low Byte)				Data	or	Config. 4th Byte		
05Hex		No. of Bytes (High Byte)				Data	or	Config. 5th Byte		
06Hex						Data	or	Config. 6th Byte		
07Hex		2nd Bit Header (as above)				Data				

Description of Output Buffer

Sub-address	Bit Name	Meaning	Function Description
00Hex Bit Header	CT	Code tag type	Select code tag type: for code tag type: 0 32 Byte block size BIS C-1_-02, -03, -04, -05 1 64 Byte block size BIS C-1_-10, -11, -30
	TI	Toggle-Bit In	for reading: Controller is ready to accept new/additional data. for writing: Controller has prepared new/additional data.
	GR	Ground state	Causes the BIS system to go to ground state.
	AV	Command	Signals the ID system that a job is waiting.

(continued next 18)

Please note the basic procedure on 15 and 30...34 and the examples on pages 35...47.

Function Description
Output buffer, configuration and explanation

**Description of
Output Buffer**
(continued)

Sub- address	Meaning	Function Description
01_{Hex}	Command designator	
00 _{Hex}		No command present
01 _{Hex}		Read code tag
02 _{Hex}		Write to code tag
04 _{Hex}		Configure processor
05 _{Hex}		Read configuration data
06 _{Hex}		Store program in the EEPROM for the Mixed Data Access function
07 _{Hex}		Store the start address for the Auto-Read function in the EEPROM
08 _{Hex}		Initialize code tag
09 _{Hex}		Save monitoring of code tag initialization in EEPROM
0A _{Hex}		Save split address for dividing the code tag into an area having memory optimization in EEPROM
21 _{Hex}		Read code tag using Mixed Data Access function (corresponding to the program stored in the EEPROM)
22 _{Hex}		Write to code tag using the Mixed Data Access function (corresponding to the program stored in the EEPROM)
or	Configuration 1st byte	
00 _{Hex}		Default value (factory setting). Changes depending on the configuration.
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.

(continued next 19)

Please note the basic procedure on 15 and 30...34 and the examples on pages 35...47.

Function Description
Output buffer, configuration and explanation

**Description of
Output Buffer**
(continued)

Sub- address	Meaning	Function Description
02_{Hex}	Start address (Low Byte)	Address at which reading from or writing to the code tag 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.
or:	Program No.	Number of the program stored in the EEPROM for read or write operations in conjunction with command ID 22 _{Hex} or 22 _{Hex} for the Mixed Data Access function.
or	Configuration 2nd byte	
80 _{Hex}		Default value (factory setting) Changes depending on the configuration.
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
or:	Parameterizing data	for writing to the EEPROM.

(continued next 20)

Please note the basic procedure on 15 and 30...34 and the examples on pages 35...47.

Function Description
Output buffer, configuration and explanation

**Description of
Output Buffer**
(continued)

Sub- address	Meaning	Function Description
03_{Hex}	Start address (High Byte)	Address for reading from or writing to the code tag. (The High Byte is additionally used for the address range from 256 to 8,191)
or:	Configuration 3rd byte 00 _{Hex}	Default value (factory setting) This value must not be changed!
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
or:	Parameterizing 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:	Configuration 4th byte 82 _{Hex}	Default value (factory setting) Changes depending on the configuration.
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.

(continued next )

Please note the
basic procedure on
 15 and 30...34
and the examples on
pages  35...47.

Function Description
Output buffer, configuration and explanation

**Description of
Output Buffer**
(continued)

Sub- address	Meaning	Function Description
05_{Hex}	No. of bytes (High Byte)	Number of bytes to read or write beginning with the start address (the High Byte is additionally used for the range between 256 and 8,191 bytes).
or:	Configuration 5th byte 00 _{Hex}	Default value (factory setting) Changes depending on the configuration.
or:	Initialization data	
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
06_{Hex}	Configuration 6th byte 00 _{Hex}	Default value (factory setting) This value must not be changed!
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.
07_{Hex}	2nd Bit header	The data are valid if the 1st and 2nd bit header are identical.
or:	Data	for writing to the code tag
or:	Program data	for writing to the EEPROM.

Please note the
basic procedure on
 15 and 30...34
and the examples on
pages  35...47.

Function Description

Input buffer, configuration and explanation

Configuration of the input buffer for one (1) Read/Write head

Bit No.	7	6	5	4	3	2	1	0	Bit name	
Subaddress										
00 _{Hex} = Bit Header	BB	HF	TO	IN	AF	AE	AA	CP		
01 _{Hex}	Error code		or			Data	or			Config. 1st byte
02 _{Hex}			Data					or		Config. 2nd Byte
03 _{Hex}			Data					or		Config. 3rd Byte
04 _{Hex}			Data					or		Config. 4th Byte
05 _{Hex}			Data					or		Config. 5th Byte
06 _{Hex}			Data					or		Config. 6th Byte
07 _{Hex}	2nd Bit Header (as above)							or	Data	

Description of Input Buffer

Sub-address	Bit Name	Meaning	Function Description
00 _{Hex}	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	for read: BIS has new/additional data ready. for write: BIS is ready to accept new/additional data.
	(continued on next 23)		

Please note the basic procedure on 15 and 30...34 and the examples on pages 35...47.

Function Description

Input buffer, configuration and explanation

Description of Input Buffer (continued)

Sub-address	Bit Name	Meaning	Function Description
00 _{Hex}	(continued)		
Bit Header	IN	Input	If the parameter "Input IN" 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	Code tag present within the active zone of the read/write head.

Sub-address	Meaning	Function Description
01 _{Hex}	Error code	Error number is entered if command was incorrectly processed or aborted. Only valid with AF bit!
00 _{Hex}		No error.
01 _{Hex}		Reading or writing not possible because no code tag is present in the active zone of a read/write head.
02 _{Hex}		Read error.
03 _{Hex}		Code tag was removed from the active zone of the read/write head while it was being read.
04 _{Hex}		Write error.
05 _{Hex}		Code tag was removed from the active zone of the read/write head while it was being written.
07 _{Hex}		AV bit is set but the command designator is missing or invalid.
	or:	Number of bytes is 00 _{Hex} .
	(continued on next 24)	

Please note the basic procedure on 15 and 30...34 and the examples on pages 35...47.

Function Description
Input buffer, configuration and explanation

**Description of
Input Buffer**
(continued)

Sub- address	Meaning	Function Description
01_{Hex}	Error code (continued)	
	09 _{Hex}	Cable break to select read/write head, or head not connected.
	0C _{Hex}	The EEPROM cannot be read/programmed.
	0D _{Hex}	Faulty communication with the code tag.
	0F _{Hex}	Contents of the 1st and 2nd bit header (1st and last bytes) of the output buffers are not identical (2nd bit header must be served).
	13 _{Hex}	Start address + number of bytes > memory range entered in the initialization.
	14 _{Hex}	invalid max. number of write cycles entered in the initialization.
	15 _{Hex}	invalid memory size of the code tag entered in the initialization.
	16 _{Hex}	Max. 1 kB segment size exceeded.
	17 _{Hex}	Code tag incorrectly initialized (comparison with nominal data with command identifier 09 _{Hex} does not bring agreement).
	18 _{Hex}	Code tag not initialized. All bytes are still 00 _{Hex} .
or:	Configuration 1st byte	
	00 _{Hex}	Default value (factory setting). Changes depending on the configuration.
or:	Data	Data which was read from the code tag.

Please note the basic procedure on 15 and 30...34 and the examples on pages 35...47.

(continued on next 26)

Function Description
Input buffer, configuration and explanation

**Description of
Input Buffer**
(continued)

Sub- address	Meaning	Function Description
02_{Hex}	Configuration 2nd byte	
	80 _{Hex}	Default value (factory setting). Do not change!
or:	Data	Data which was read from the code tag,
or:	Warning	Memory management (details see page 9) signals:
	56 _{Hex}	Pre-warning (1,000 write cycles left).
	45 _{Hex}	End warning.
03_{Hex}	Configuration 3rd byte	
	00 _{Hex}	Default value (factory setting). Changes depending on the configuration.
or:	Data	Data which was read from the code tag.
04_{Hex}	Configuration 4th byte	
	82 _{Hex}	Default value (factory setting). Changes depending on the configuration.
or:	Data	Data which was read from the code tag.
05_{Hex}	Configuration 5th byte	
	00 _{Hex}	Default value (factory setting). Changes depending on the configuration.
or:	Data	Data which was read from the code tag.
06_{Hex}	Configuration 6th byte	
	00 _{Hex}	Default value (factory setting). Do not change!
or:	Data	Data which was read from the code tag.
07_{Hex}	2nd Bit header	The data are valid if the 1st and 2nd bit headers are in agreement.
or:	Data	Data which was read from the code tag.

Please note the basic procedure on 15 and 30...34 and the examples on pages 35...47.

Function Description
Configuring the BIS C-60_1 processor

Configuration,
Overview

The following functions can be activated / deactivated through the configuration:

- **Dynamic operation on Read/Write Head 1 or 2:**
If dynamic operation is parametered, a read/write job can be sent even though there is no code tag in the active zone of the head. As soon as a code tag passes by the head, the command is immediately carried out.
- **2nd bit header at end of in- and output buffer:**
The 2nd bit header (factory setting) prevents data from being accepted by the bus as long as it is not fully updated.
- **Display state of the digital input in the bit header of the input buffer:**
If this function is activated, the IN-bit displays the state of the digital input of the processor: IN = 0 → digital input low; IN = 1 → digital input high
- **Reset BIS C-60_1 processor through the digital input:**
If this function is activated, the processor is reset when the digital input is set to high.
- **Monitor code tag initialization:**
If this function is activated, the initialization data on the code tag are compared with the initialization data stored in the processor. If these do not agree, a read/write error is rejected with an error message.
- **Split code tag memory:**
If this function is activated, data on the code tag beyond a certain specified address are no longer optimized. The split address must have first been stored on the processor using command identifier 0A_{Hex}.

Function Description
Configuring the BIS C-60_1 processor

Configuration

The BIS C-60_1 processor is configured by the controller using the output buffer. The configuration data are arranged within 6 configuration bytes that are sent to the BIS C-60_1 processor using the command identifier 04_{Hex} (see Example 11 on 46). Command identifier 05_{Hex} is used to read out the current device configuration (see Example 12 on 47).



To input the configuration, all 6 bytes must be entered in _{Hex}. Only the named bits are permitted to be changed. If any of the other bits are changed, there is no assurance that the BIS C-6_1 will function properly.

The default values of the 6 bytes are (factory setting):

	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte
Hex	00	80	00	82	00	00
Binary	00000000	10000000	00000000	10000010	00000000	00000000
		bit 1 bit 2		bit 7 bit 8	bit 5	

These are used for
configuration:

- Having the following functions:
- 1st byte, bit 2,** Monitor code tag initialization
 - 1st byte, bit 1** Divide code tag memory into one area with memory expansion and one area without memory expansion.
 - 2nd byte, bit 5,** Dynamic mode on read/write head 1 (for effects on read/write times, see 48/49)
 - 4th byte, bit 8,** Arrange a 2nd bit header at the end of the input and output buffers.

Bit state: 0 = no
1 = yes

Function Description
Configuring the BIS C-60_1 processor

Configuration (continued)	4th byte, bit 7	Display state of the digital input in the bit header of the input buffers
	4th byte, bit 2	Reset the BIS C-60_1 processor through the digital input
	5th byte, bit 5	Dynamic mode on read/write head 2 (for effects on read/write times, see 48/49)

Bit state: 0 = no
1 = yes

Function Description
Processing code tags, reading and writing

Reading and writing	To carry out a read or write job, the code tag must be located in the active zone of the read/write head. A read/write job has the following sequence (see examples on 37...39): 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_{Hex}/03_{Hex}, – the number of bytes for reading or writing to subaddress 04_{Hex}/05_{Hex}, – the CT bit according to the code tag 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 code tag to input buffer, write = from output buffer to code tag. Larger data quantities are sent in blocks: block size of 6 bytes with 2nd bit header, block size of 7 bytes without 2nd bit header. The toggle bits in the two bit headers are used as a kind of handshaking between the host and the BIS C-60_1 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.
----------------------------	--

Function Description
Processing code tags, reading and writing

Reading and writing in dynamic mode	In normal operation a read/write job is rejected by the processor by setting the AF bit and an error number if there is no code tag 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 code tag is recognized, the stored job is carried out.
Read from code tag, 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 code tag. 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 8 on 43)
Write to code tag, with program Mixed Data Access	The command identifier 22 _{Hex} can be used to write the program records stored in the program to the code tag. 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 9 on 44)

Function Description
Processing code tags, Mixed Data Access

Mixed Data Access	Small read/write programs can be stored in the BIS C-60_1 processor's EEPROM. The Mixed Data Access function is useful when the required information is stored on the code tag at various addresses. This function makes it possible to read out this "mixed", i.e. non-contiguously stored data from the code tag 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 06_{Hex} is used to send the read/write program to the BIS C-60_1 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 contiguous. They must be sent one after the other and be terminated 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
Processing code tags, Mixed Data Access

Mixed Data Access
(cont.)

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:			
...			
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 7th example on [p. 40...42](#).

Replacing the EEPROM is described on [p. 61](#) for BIS C-6001 and on [p. 72](#) for BIS C-6021.

Function Description
Monitoring initialization, splitting memory

Monitoring code tag initialization
(see 2nd example on [p. 36](#))

Command identifier 09_{Hex} allows initialization data to be stored in the processor.

If the Monitor Code Tag Initialization function is activated, a read/write command results first in the initialization data on the code tag being compared with the data in the processor. If these do not agree, error message 17_{Hex} is output. If the code tag has not even been initialized yet, i.e. all initialization data are 0, error message 18_{Hex} is output.

Splitting code tag memory
(see 3rd example on [p. 36](#))

Command identifier 0A_{Hex} allows an address to be stored in the processor starting at which the code tag data are no longer memory-optimized.

When dividing the memory, make the sector with memory optimization large enough so that it is sufficient for the maximum intended number of write cycles (see [p. 9](#) for calculation).

Division brings an advantage in processing small quantities of data that are frequently programmed (written) in the sector not having memory optimization.

If the split function is activated, this memory division applies to every code tag entering the read/write zone of the processor.

Function Description
Examples for protocol sequence

Example No. 1

For configuring with double bit header!

Initializing the code tag for memory optimization
(code tag type with 32 byte block size)

Host:

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

01 _{Hex}	Command designator 08 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit to 0 (32 Byte block size), set AV-Bit

3.) Process subaddresses of output buffer:

01...06 _{Hex}	Enter first 6 bytes of data
00 _{Hex} /07 _{Hex}	Invert TI-Bit

5.) Process subaddresses of output buffer:

00 _{Hex} /07 _{Hex}	Reset AV-Bit
--------------------------------------	--------------

BIS C-60_1 Identification System:

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

00 _{Hex} /07 _{Hex}	Set AA-Bit, invert TO-Bit
--------------------------------------	---------------------------

4.) Process subaddresses of input buffer:

01...05 _{Hex}	Copy the 5 initialization bytes
00 _{Hex} /07 _{Hex}	Set AE-Bit

6.) Process subaddresses of input buffer:

00 _{Hex} /07 _{Hex}	Reset AA-Bit and AE-Bit
--------------------------------------	-------------------------

Function Description
Examples for protocol sequence

Example No. 2

For configuring with double bit header!

Store code tag initialization data for monitoring in the processor

Host:

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

01 _{Hex}	Command identifier 09 _{Hex}
02...05 _{Hex}	Enter 2nd through 5th initialization bytes
00 _{Hex} /07 _{Hex}	Set AV-bit

3.) Process subaddresses of output buffer:

00 _{Hex} /07 _{Hex}	Reset AV-Bit
--------------------------------------	--------------

BIS C-60_1 Identification System:

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

00 _{Hex} /07 _{Hex}	Set AA-bit and TO-bit
02...05 _{Hex}	Copy 2nd through 5th initialization bytes
00 _{Hex} /07 _{Hex}	Set AE-bit

4.) Process subaddresses of input buffer:

00 _{Hex} /07 _{Hex}	Reset AA-Bit and AE-Bit
--------------------------------------	-------------------------

Example No. 3

For configuring with double bit header!

Splitadresse in die Auswerteeinheit speichern

Host:

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

01 _{Hex}	Command identifier 0A _{Hex}
02...03 _{Hex}	Enter split address (Low/High byte)
00 _{Hex} /07 _{Hex}	Set AV-bit

3.) Process subaddresses of output buffer:

00 _{Hex} /07 _{Hex}	Reset AV-Bit
--------------------------------------	--------------

BIS C-60_1 Identification System:

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

00 _{Hex} /07 _{Hex}	Set AA-bit and TO-bit
02...03 _{Hex}	Copy split address (Low/High byte)
00 _{Hex} /07 _{Hex}	Set AE-bit

4.) Process subaddresses of input buffer:

00 _{Hex} /07 _{Hex}	Reset AA-Bit and AE-Bit
--------------------------------------	-------------------------

Function Description

Examples for protocol sequence

Example No. 4

For configuring with
double bit header!

Read 17 bytes starting at code tag address 10 (code tag type with 32 byte block size):

Host:

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

01 _{Hex}	Command designator 01 _{Hex}
02 _{Hex}	Start address Low Byte 0A _{Hex}
03 _{Hex}	Start address High Byte 00 _{Hex}
04 _{Hex}	No. of bytes Low Byte 11 _{Hex}
05 _{Hex}	No. of bytes High Byte 00 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit to 0 (32 Byte block size), set AV-Bit

3.) Process subaddresses of input buffer:

01...06 _{Hex}	Copy first 6 data bytes
------------------------	-------------------------

Process subaddress of the output buffer:

00 _{Hex} /07 _{Hex}	Invert TI-Bit
--------------------------------------	---------------

5.) Process subaddresses of input buffer:

01...06 _{Hex}	Copy second 6 data bytes
------------------------	--------------------------

Process subaddress of the output buffer:

00 _{Hex} /07 _{Hex}	Invert TI-Bit
--------------------------------------	---------------

7.) Process subaddresses of input buffer:

01...05 _{Hex}	Copy the remaining 5 data bytes
------------------------	---------------------------------

Process subaddress of the output buffer:

00 _{Hex} /07 _{Hex}	Reset AV-Bit
--------------------------------------	--------------

BIS C-60_1 Identification System:

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

00 _{Hex} /07 _{Hex}	Set AA-Bit
01...06 _{Hex}	Enter first 6 bytes of data
00 _{Hex} /07 _{Hex}	Set AE-Bit

4.) Process subaddresses of input buffer:

01...06 _{Hex}	Enter the second 6 data bytes
00 _{Hex} /07 _{Hex}	Invert TO-Bit

6.) Process subaddresses of input buffer:

01...05 _{Hex}	Enter the remaining 5 data bytes
00 _{Hex} /07 _{Hex}	Invert TO-Bit

8.) Process subaddresses of input buffer:

00 _{Hex} /07 _{Hex}	Reset AA-Bit and AE-Bit
--------------------------------------	-------------------------

Function Description

Examples for protocol sequence

Example No. 5

For configuring with
double bit header!

Read 30 bytes starting at code tag address 10 with read error
(code tag type with 64 byte block size):

Host:

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

01 _{Hex}	Command designator 01 _{Hex}
02 _{Hex}	Start address Low Byte 0A _{Hex}
03 _{Hex}	Start address High Byte 00 _{Hex}
04 _{Hex}	No. of bytes Low Byte 1E _{Hex}
05 _{Hex}	No. of bytes High Byte 00 _{Hex}
00 _{Hex} /07 _{Hex}	Set CT-Bit to 1 (64 Byte block size), set AV-Bit

3.) Process subaddress of the input buffer:

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

Process subaddress of the output buffer:

00 _{Hex} /07 _{Hex}	Reset AV-Bit
--------------------------------------	--------------

BIS C-60_1 Identification System:

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

If an error occurs right away:

00 _{Hex} /07 _{Hex}	Set AA-Bit
01 _{Hex}	Enter error number
00 _{Hex} /07 _{Hex}	Set AF-Bit

4.) Process subaddresses of the input buffer:

00 _{Hex} /07 _{Hex}	Reset AA-Bit and AF-Bit
--------------------------------------	-------------------------

Function Description
Examples for protocol sequence

Example No. 6
For configuring with double bit header!

Write 16 bytes starting at code tag address 20 (code tag type with 32 byte block size):

Host:

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

01 _{Hex}	Command designator 02 _{Hex}
02 _{Hex} /03 _{Hex}	Start address 14 _{Hex} / 00 _{Hex}
04 _{Hex} /05 _{Hex}	No. of bytes 10 _{Hex} / 00 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit to 0 (32 Byte block size), set AV-Bit

3.) Process subaddresses of the output buffer:

01...06 _{Hex}	Enter the first 6 data bytes
00 _{Hex} /07 _{Hex}	Invert TI-Bit

5.) Process subaddresses of the output buffer:

01...06 _{Hex}	Enter the second 6 data bytes
00 _{Hex} /07 _{Hex}	Invert TI-Bit

7.) Process subaddresses of the output buffer:

01...04 _{Hex}	Enter the remaining 4 data bytes
00 _{Hex} /07 _{Hex}	Invert TI-Bit

9.) Process subaddresses of the output buffer:

00 _{Hex} /07 _{Hex}	Reset AV-Bit
--------------------------------------	--------------

BIS C-60_1 Identification System:

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

00 _{Hex} /07 _{Hex}	Set AA-Bit, invert TO-Bit
--------------------------------------	---------------------------

4.) Process subaddresses of the output buffer:

01...06 _{Hex}	Copy the first 6 data bytes
------------------------	-----------------------------

Process subaddress of the input buffer:

00 _{Hex} /07 _{Hex}	Invert TO-Bit
--------------------------------------	---------------

6.) Process subaddresses of the output buffer:

01...06 _{Hex}	Copy the second 6 data bytes
------------------------	------------------------------

Process subaddress of the input buffer:

00 _{Hex} /07 _{Hex}	Invert TO-Bit
--------------------------------------	---------------

8.) Process subaddresses of the output buffer:

01...04 _{Hex}	Copy the remaining 4 data bytes
------------------------	---------------------------------

Process subaddress of the input buffer:

00 _{Hex} /07 _{Hex}	Set AE-Bit
--------------------------------------	------------

10.) Process subaddresses of the input buffer:

00 _{Hex} /07 _{Hex}	Reset AA-Bit and AE-Bit
--------------------------------------	-------------------------

Function Description
Examples for protocol sequence

Example No. 7
Store Mixed Data Access program

For configuring with double bit header!

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	312	Number of bytes	17

Total number of bytes exchanged in the operation: 27 bytes

All 104 bytes are written for the programming.

Host:

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

01 _{Hex}	Command designator 06 _{Hex}
02 _{Hex}	Program number 01 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit to 0 or 1 (depending on block size), set AV-Bit

3.) Subadressen des Ausgangspuffers bearbeiten:

01 _{Hex}	1st start address (Low Byte) 05 _{Hex}
02 _{Hex}	(High Byte) 00 _{Hex}
03 _{Hex}	1st number of bytes (Low Byte) 07 _{Hex}
04 _{Hex}	(High Byte) 00 _{Hex}
05 _{Hex}	2nd start address (Low Byte) 4B _{Hex}
06 _{Hex}	(High Byte) 00 _{Hex}
00 _{Hex} /07 _{Hex}	Invert TI-Bit

Host:

2.) Process subaddresses of the input buffer:

00 _{Hex} /07 _{Hex}	Set AA-Bit, invert TO-Bit
--------------------------------------	---------------------------

4.) Subadressen des Eingangspuffers bearbeiten:

00 _{Hex} /07 _{Hex}	Invert TO-Bit
--------------------------------------	---------------

Continued on next □.

Function Description
Examples for protocol sequence

Example No. 7
Store Mixed Data
Access program
(continued)

**For configuring with
double bit header!**

5.) Process subaddresses of the output buffer:

01 _{Hex}	2nd number of bytes	(Low Byte) 03 _{Hex}
02 _{Hex}		(High Byte) 00 _{Hex}
03 _{Hex}	3rd start address	(Low Byte) 38 _{Hex}
04 _{Hex}		(High Byte) 01 _{Hex}
05 _{Hex}	3rd number of bytes	(Low Byte) 11 _{Hex}
06 _{Hex}		(High Byte) 00 _{Hex}
00 _{Hex} /07 _{Hex}	Invert TI-Bit	

7.) Process subaddresses of the output buffer:

01 _{Hex} /02 _{Hex}	Terminator	FF _{Hex} /FF _{Hex}
03 _{Hex} /04 _{Hex}	(not used)	FF _{Hex} /FF _{Hex}
05 _{Hex} /06 _{Hex}	(not used)	FF _{Hex} /FF _{Hex}
00 _{Hex} /07 _{Hex}	Invert TI-Bit	

Fill all unused start addresses and number of bytes with FF_{Hex}!

6.) Process subaddresses of the input buffer:

00 _{Hex} /07 _{Hex}	Invert TO-Bit
--------------------------------------	---------------

8.) Process subaddresses of the input buffer:

00 _{Hex} /07 _{Hex}	Invert TO-Bit
--------------------------------------	---------------

Continued on next .

Function Description
Examples for protocol sequence

Example No. 7
Store Mixed Data
Access program
(continued)

**For configuring with
double bit header!**

35.) Process subaddresses of the output buffer:

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

37.) Process subaddresses of the output buffer:

00 _{Hex} /07 _{Hex}	AV-Bit rücksetzen
--------------------------------------	-------------------

36.) Process subaddresses of the input buffer:

00 _{Hex} /07 _{Hex}	AE-Bit setzen
--------------------------------------	---------------

38.) Process subaddresses of the input buffer:

00 _{Hex} /07 _{Hex}	AA-Bit und AE-Bit rücksetzen
--------------------------------------	------------------------------



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

Examples for protocol sequence

Example No. 8 Use Mixed Data Access program

For configuring with
double bit header!

Read code tag using Program No. 1 (code tag type with 32 byte block size):

Host:

BIS C-60_1 Identification System:

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

01 _{Hex}	Command designator 21 _{Hex}
02 _{Hex}	Program number 01 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit to 0 (32 byte block size), set AV-Bit

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

00 _{Hex} /07 _{Hex}	Set AA-Bit
01...06 _{Hex}	Enter first 6 bytes of data
00 _{Hex} /07 _{Hex}	Set AE-Bit

3.) Process subaddresses of the input buffer:

01...06 _{Hex}	Copy first 6 data bytes
Process subaddress of the output buffer:	
00 _{Hex} /07 _{Hex}	Invert TI-Bit

4.) Process subaddresses of the output buffer:

01...06 _{Hex}	Enter the second 6 data bytes
00 _{Hex} /07 _{Hex}	Invert TO-Bit

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

For the remainder of the procedure, see Example 4 on □ 37.



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

Function Description

Examples for protocol sequence

Example No. 9 Use Mixed Data Access program

For configuring with
double bit header!

Write code tag using Program No. 1 (code tag type with 32 byte block size):

Host:

BIS C-60_1 Identification System:

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

01 _{Hex}	Command designator 21 _{Hex}
02 _{Hex}	Program number 01 _{Hex}
00 _{Hex} /07 _{Hex}	CT-Bit to 0 (32 byte block size), set AV-Bit

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

00 _{Hex} /07 _{Hex}	Set AA-Bit, invert TO-Bit
--------------------------------------	---------------------------

3.) Process subaddresses of the output buffer:

01...06 _{Hex}	Enter the first 6 data bytes
00 _{Hex} /07 _{Hex}	Invert TI-Bit

4.) Process subaddresses of the output buffer:

01...06 _{Hex}	Copy the first 6 data bytes
Process subaddress of the input buffer:	
00 _{Hex} /07 _{Hex}	Invert TO-Bit

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

For the remainder of the procedure, see Example 6 on □ 39.



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

Function Description

Examples for protocol sequence

Example No. 10

For configuring with double bit header!

Put the relevant read/write head into ground state:

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

Host:

1.) Process subaddresses of the output buffer:

00Hex/07Hex	Set GR-Bit
-------------	------------

3.) Process subaddresses of the output buffer:

00Hex/07Hex	Reset GR-Bit
-------------	--------------

BIS C-60_1 Identification System:

2.) Go to ground state;
Process subaddresses of the input buffer:

00Hex/07Hex	Reset BB-Bit
-------------	--------------

4.) Process subaddresses of the input buffer:

00Hex/07Hex	Set BB-Bit
-------------	------------

Function Description

Examples for protocol sequence

Example No. 11

For configuring with double bit header!

Program configuration data:

Configuration data can be programmed in both buffers - for Head 1 and Head 2 - as desired.

Host:

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

01Hex	Command designator 04Hex
00Hex/07Hex	Set AV-Bit

3.) Process subaddresses of the output buffer:

01...06Hex	Enter the 6 configuration bytes
00Hex/07Hex	Invert TI-Bit

5.) Process subaddresses of the output buffer:

00Hex/07Hex	Reset AV-Bit
-------------	--------------

BIS C-60_1 Identification System:

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

00Hex/07Hex	Set AA-Bit, invert TO-Bit
-------------	---------------------------

4.) Process subaddresses of the output buffer:

01...06Hex	Enter the 6 configuration bytes
Process subaddresses of the input buffer	
00Hex/07Hex	Set AE-Bit

6.) Process subaddresses of the input buffer:

00Hex/07Hex	Reset AA-Bit and AE-Bit
-------------	-------------------------

Function Description

Examples for protocol sequence

Example No. 12

For configuring with double bit header!

Read-out programmed configuration data:

Host:

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

01 _{Hex}	Command designator 05 _{Hex}
00 _{Hex} /07 _{Hex}	Set AV-Bit

3.) Process subaddresses of the input buffer::

01...06 _{Hex}	Copy the 6 configuration bytes
Process subaddress of the output buffer:	
00 _{Hex} /07 _{Hex}	Reset AV-Bit

BIS C-60_1 Identification System:

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

00 _{Hex}	Set AA-Bit
01...06 _{Hex}	Enter the 6 configuration bytes
00 _{Hex}	Set AE-Bit

4.) Process subaddresses of the output buffer:

00 _{Hex} /07 _{Hex}	Reset AA-Bit and AE-Bit
--------------------------------------	-------------------------

Read/Write Times

Read times from code tag to processor in static mode
(parametering: 2nd byte, bit 5 = 0)

For double read and compare:

Code tag with 32 byte blocks	
No. of bytes	Read time [ms]
from 0 to 31	110
for each additional 32 bytes add	120
from 0 to 255	= 950

Code tag with 64 byte blocks	
No. of bytes	Read time [ms]
from 0 to 63	220
for each additional 64 bytes add	230
from 0 to 2047	= 7350

Write times from processor to code tag in static mode
(parametering: 2nd byte, bit 5 = 0)

Including readback and compare:

Code tag with 32 byte blocks	
No. of bytes	Write time [ms]
from 0 to 31	110 + n * 10
for 32 bytes or more	y * 120 + n * 10

Code tag with 64 byte blocks	
No. of bytes	Write time [ms]
from 0 to 63	220 + n * 10
for 64 bytes or more	y * 230 + n * 10

n = number of contiguous bytes to write
y = number of blocks to be processed

Example: 17 bytes from address 187 have to be written. Code tag with 32 bytes per block. The blocks 5 and 6 will be processed since the start address 187 is in block 5 and the end address 203 in block 6. $t = 2 * 120 + 17 * 10 = 410$ ms



The indicated times apply after the code tag has been recognized. If the code tag is not yet recognized, an additional 45 ms for building the required energy field until the code tag is recognized must be added.

Read/Write Times

Read times from code tag to processor in dynamic mode (parametering: 2nd byte, bit 5 = 1)

Read times within the 1st block for dual read and compare:

The indicated times apply after the code tag has been recognized. If the code tag is not yet recognized, an additional 45 ms for building the required energy field until the code tag is recognized must be added.

Code tag with 32 byte blocks		Code tag with 64 byte blocks	
No. of bytes	Read time [ms]	No. of bytes	Read time [ms]
from 0 to 3	14	from 0 to 3	14
for each additional byte add	3.5	for each additional byte add	3.5
from 0 to 31	112	from 0 to 63	224

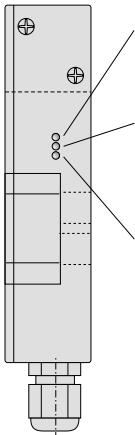
m = highest address to be read

Formula: $t = (m + 1) \cdot 3.5 \text{ ms}$

Example: Read 11 bytes starting at address 9, i.e. the highest address to be read is 19. This corresponds to 70 ms.

LED Display

Function displays on BIS C-60_1



The BIS C-60_1 uses the three side-mounted LED's to indicate important conditions of the identification system.

Status	LED	Meaning
Ready / Bus active	red	Supply voltage OK; no hardware error, however, bus not active.
	green	Supply voltage / hardware OK, bus active.
CT1 Present / operating	green	Code tag read/write-ready at read/write head 1. Read/write command at read/write head 1 in process.
	yellow	Cable break to read/write head or not connected.
	yellow flashes off	No code tag in read/write range of read/write head 1.
CT2 Present / operating	green	Code tag read/write-ready at read/write head 2. Read/write command at read/write head 2 in process.
	yellow	Cable break to read/write head or not connected.
	yellow flashes off	No code tag in read/write range of read/write head 2.

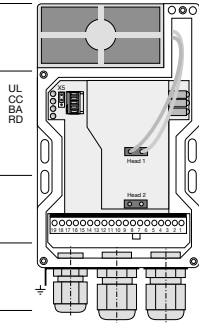
If all three LED's are synchronously flashing, it means a hardware error. Return the unit to the factory.

Function displays

Operating state
INTERBUS

4 diagnostic LEDs on the motherboard inside the processor report the key operating states on the INTERBUS.

LED	State	INTERBUS message
UL	green / off	Reset Protocol chip is / is not supplied with power.
RC / CC Remotebus Check Cable Check	green / off	Communication Ready Communication with the IBS- Master is / is not possible, the application has not however yet started to exchange data.
BA Bus Active	green / off	Communication active or also Run Master is / is not exchanging user data.
RD Remotebus Disable	yellow / off	Remotebus Disable The extended bus interface is / is not turned off.



BIS C-6001
Mounting Head / Processor

Mounting the
read/write head or
adapter

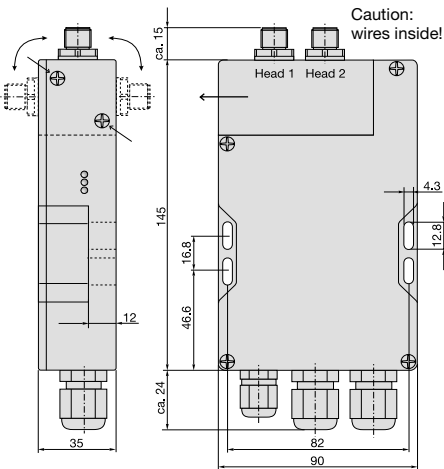
Depending on model, the processor is equipped with a read/write head or the adapter for offset read/write heads. Both the read/write head and the adapter can be rotated by the user by + or -90 deg. to the desired position (see drawing). Be sure that power is off first. Loosen both screws (indicated with arrows). Carefully pull the head or adapter out towards the side (direction of arrow, right drawing).

Caution: wires inside!

Reattach at the desired orientation and screw tight again.

Mounting the
BIS C-6001
processor

The processor is attached using 4 lateral mounting holes.



BIS C-6001 Opening the Processor

Opening the Processor BIS C-6001

The BIS C-6001 processor must be opened to perform the following steps:

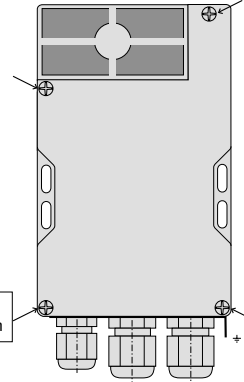
- Set/change compatibility mode
- Replace EEPROM
- Make electrical connections (supply voltage, in-/output, INTERBUS connections).



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

Remove the 4 screws on the BIS C-6002 and lift off the cover.

Perform the desired action. To make the electrical connections, push the cables through the fittings. For additional wiring details, see the following [□ 53](#).



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

Opening the processor

BIS C-6001 Installing the connection cables

Make connections on the BIS C-6001 processor

The BIS C-6001 processor must be opened in order to make the connections for the supply voltage, the digital input and the INTERBUS connections (see [□ 53](#)).

First be sure that the unit is disconnected from power.

Remove the 4 screws on the BIS C-6001 and lift off the cover.

Guide the two INTERBUS cables through the PG 11 fittings (see [□ 55](#)). For additional information on wiring, see the following [□ 53](#).

Push the cable for supply voltage and for the digital input through the PG 9 fitting.

Close up the processor.

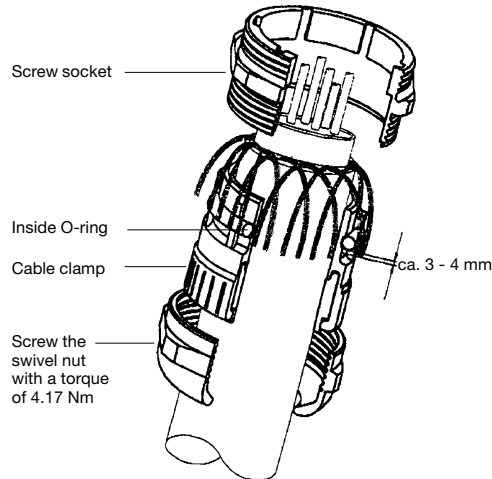
If the processor is equipped with an adapter:

- BIS C-650: Connect the read/write heads to terminals Head 1 and Head 2.
- BIS C-670: Connect the read/write head to terminal Head 1.

BIS C-6001 Mounting the PG Connection

Mounting the PG Connection on the processor BIS C-6001

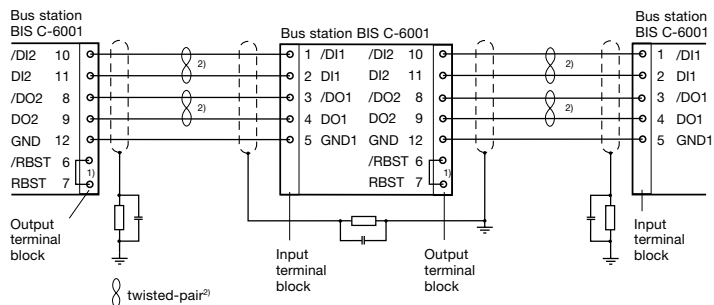
After connecting the (field) bus leads to the terminal block, make sure that the shield has proper connection to the PG housing.



BIS C-6001 Interface Information / Wiring Diagrams

Remote bus cable and interfaces for INTERBUS

To insert the BIS C-6001 processors into the serial INTERBUS, the terminal strip provides terminals 1...5 for the input interface and terminals 8...12 for the output interface of the INTERBUS. The following drawing shows the wiring when the BIS C-6001 processors need to be connected together.



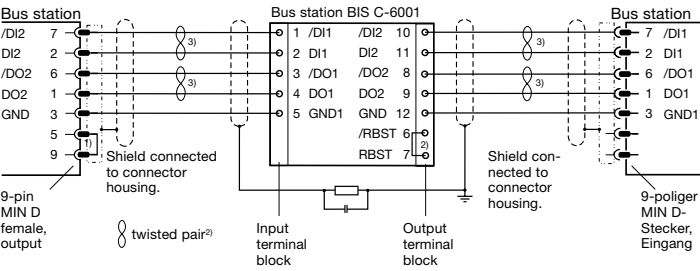
¹⁾ Leave the jumper in the BIS C-6001 if an additional station is to follow. Remove it if no additional station follows.

²⁾ The differential signals DO and /DO as well as DI and /DI must be twisted-pair. Recommended cable: LiYCY 3x2x0.25 mm² (AWG 24); maximum cable capacitance: 120 pF/m

BIS C-6001 Interface Information / Wiring Diagrams

Remote bus cable and interfaces for INTERBUS (continued)

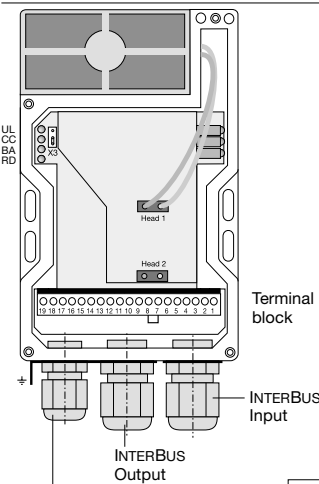
To insert the BIS C-6001 processors into the serial INTERBUS, the terminal strip provides terminals 1...5 for the input interface and terminals 8...12 for the output interface of the INTERBUS. The following drawing shows the wiring when the BIS C-6001 interface needs to be connected using a 9-pin terminal (e.g. to a BIS C-6021).



- 1) Connect the jumper in the connector if another station is to follow. Remove it if no additional station follows.
- 2) The jumper remains in the BIS C-601 if an another station is to follow. Remove it if no additional station follows.
- 3) The differential signals DO and /DO as well as DI and /DI must be twisted-pair. Recommended cable: LiYCY 3x2x0.25 mm² (AWG 24); maximum cable capacitance: 120 pF/m

BIS C-6001 Interface Information / Wiring Diagrams

Wiring diagram for BIS C-6001 processor with integrated read/write head



Terminal block assignments				
5	4	3	2	1
GND	DO1	/DO1	DI1	/DI1
INTERBUS Input				

12	11	10	9	8	7	6
GND	DI2	/DI2	DO2	/DO2	RBST	/RBST
INTERBUS Output						

19	18	17	16	15	14	13
+ VS	- VS	TxD	RxD	GND	+IN	-IN
Power		RS 232		Digital Input		

Terminal location and designation

Power supply and digital input

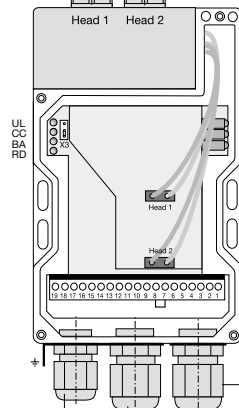
When connecting the bus leads, make sure that the shield has proper connection to the PG housing. Please note the assembling instructions on 55.

BIS C-6001 Interface Information / Wiring Diagrams

Wiring diagram for
BIS C-6001
processor with
BIS C-650 adapter

Connection for Read/Write Head 1

Connection for Read/Write Head 2



Terminal block
assignments

5	4	3	2	1
GND	DO1	/DO1	DI1	/DI1
INTERBUS Input				

12	11	10	9	8	7	6
GND	DI2	/DI2	DO2	/DO2	RBST	/RBST
INTERBUS Output						

19	18	17	16	15	14	13
+VS	-VS	TxD	RxD	GND	+IN	-IN
Power		RS 232		Digital Input		

Terminal location
and designation

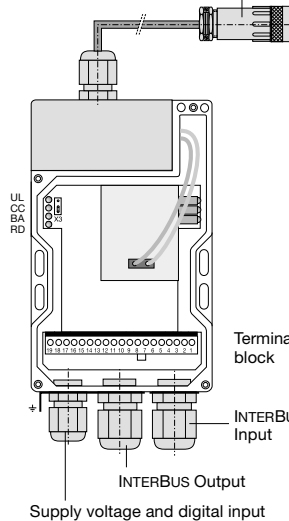
INTERBUS
Input
Power supply and digital input

When connecting the bus leads, make sure that the shield has proper connection to the PG housing. Please note the assembling instructions on □ 55.

BIS C-6001 Interface Information / Wiring Diagrams

Wiring plan for
BIS C-6001 using
BIS C-670 adapter

Connection for read/write head, 8-pin



Terminal block
assignments

5	4	3	2	1
GND	DO1	/DO1	DI1	/DI1
INTERBUS Input				

12	11	10	9	8	7	6
GND	DI2	/DI2	DO2	/DO2	RBST	/RBST
INTERBUS Output						

19	18	17	16	15	14	13
+VS	-VS	TxD	RxD	GND	+IN	-IN
Power		RS 232		Digital Input		

Connection location
and name

INTERBUS
Input
Supply voltage and digital input

When connecting the bus leads, make sure that the shield has proper connection to the PG housing. Please note the assembling instructions on □ 55.

BIS C-6001
Changing the EEPROM

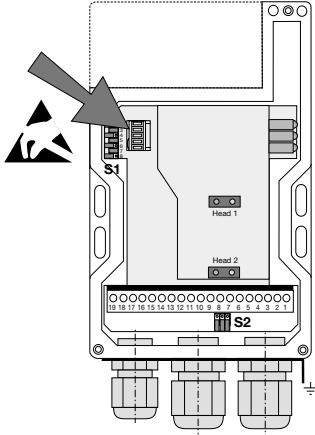
**Changing the
EEPROM in the
BIS C-6001
processor**

To replace the EEPROM, open up the processor as described on 53.



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 C-6001
Technical Data

Dimensions, Weight	Housing	Plastic PS
	Dimensions with read/write head BIS C-652	169 x 90 x 35 mm
Operating Conditions	Dimensions with adapter BIS C-650	184 x 90 x 35 mm
	Weight	400 g
Connections	Ambient Temperature	0 °C to +60 °C
	Terminal Block	19-pin
Enclosure Rating	Cable Entry for supply voltage	1 x PG 9 fitting (metal)
	Cable Entry for INTERBUS, in-/output	2 x PG 11 fittings (metal)
Electrical Connections	Cable Diameter	4 to 8 mm for PG 9 5 to 10 mm for PG 11
	Wire gauge	0.14 to 1 mm ² 0.25 to 0.34 mm ²
Enclosure Rating	with end crimps	
	Enclosure Rating	IP 65 (when connected)
Electrical Connections	Supply Voltage V_s, input	DC 24 V ± 20 %
	Ripple	≤ 10 %
Electrical Connections	Current Draw	≤ 400 mA
	INTERBUS, In- / and Outputs	serial interface for remote bus station, Ident-No. 03, 16 bytes IN, 16 bytes OUT
Electrical Connections	Digital Input (+IN, -IN)	Terminal block, Optocoupler isolated
	Control voltage active	4 V to 40 V
Electrical Connections	Control voltage inactive	1.5 V to -40 V
	Input current at 24 V	11 mA
Electrical Connections	Delay time, typ.	5 ms

BIS C-6001
Technical Data

Electrical Connections (continued)	Service interface		RS 232
	Read/Write Head* option for mounted adapter BIS C-650*		integrated, BIS C-65_ and following; 2 x connectors 4-pin (male) for all read/write heads BIS C-3_ _ with 4-pin connector (female), not BIS C-350 and BIS C-352 1 x connector 4-pin (male) for read/write heads BIS C-350 / BIS C-352
	option for mounted adapter BIS C-670*		
Function Displays	BIS operating states (LED in housing)	LED red / green LED green / yellow LED green / yellow	Ready / Bus active CT1 Present / Operating CT2 Present / Operating
	INTERBUS state (LED on side of housing)	LED green LED green LED green LED yellow	Reset Cable Check Bus active Remotebus Disable



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 50081-2 (Emission) and EN 50082-2 (Noise Immunity).

* rotatable by 90 degrees

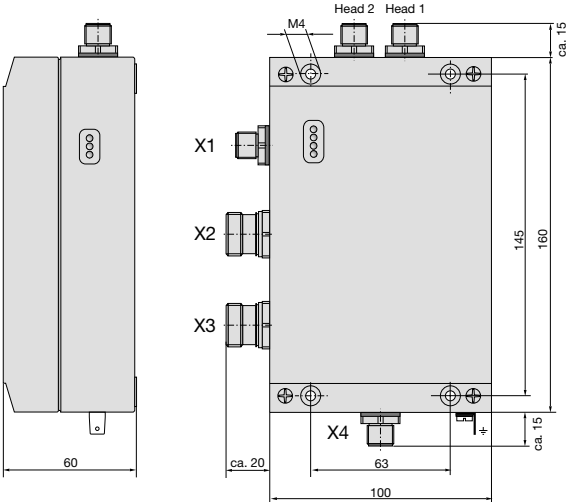
BIS C-601
Ordering Information

Ordering Code	BIS C-6001-029- -03-KL2
Balluff Identification System	
Type C Read/Write System	
Hardware Type	
6001 = INTERBUS (remote bus)	
Software Type	
029 = INTERBUS	
Read/Write Head	
000 = no read/write head	
651 = with read/write head Type 651 (with circular antenna on top)	
652 = with read/write head Type 652 (with circular antenna on front)	
653 = with read/write head Type 653 (with rod antenna)	
650 = with two connections for external read/write heads BIS C-3_ _ (except BIS C-350 and -352)	
670 = with offset connection for an external read/write head BIS C-350 or BIS C-352	
Interface	
BUS versions	
User Connection	
KL2 Clamp connection via 1 x PG 9 and 2 x PG 11	

BIS C-6021 Mounting Processor

Mounting the BIS C-6021 processor

The processor is mounted using 4 M4 screws.



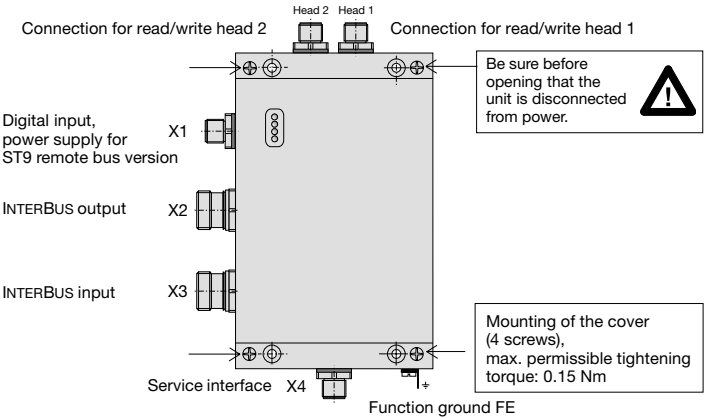
BIS C-6021 Opening the processor / Interface information

Opening the BIS C-6021 processor

To convert the processor for the power supply via the installation remote bus instead of supplying it via X1, the internal connections have to be changed.

Ensure that the device is turned off. Remove the 4 screws on the BIS C-6021 and lift off the cover. Conversion see the following [1].

Connection diagram for BIS C-6021 processor



Connection locations
and names

BIS C-6021 Interface information / Wiring Diagrams

To make the connections for the InterBus, the supply voltage and the digital input, connect the pre-assembled cable to the processor. For additional wiring information, see the following [1].
Connect the read/write heads to the terminals for Head 1 and Head 2.

Connecting on the remote bus or installation remote bus

The BIS C-6021...ST8 processor is intended for use on the installation remote bus, which provides the supply voltage over the bus. The BIS C-6021...ST9 processor is intended for use on the remote bus, whereby the supply voltage for the processor is brought in through X1.

Connect the incoming INTERBUS cable to the INTERBUS input X3. Connect the outgoing INTERBUS cable to the INTERBUS output X2.

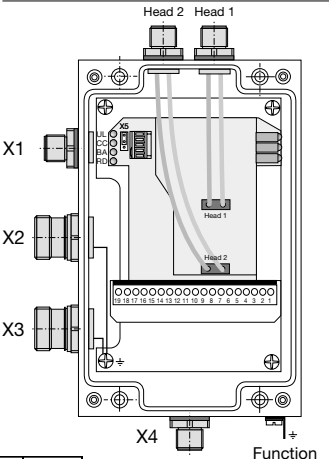
If this remote bus station is the last one on the bus, the INTERBUS output X2 must be closed off with a threaded cap to maintain the enclosure rating.



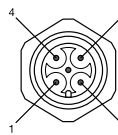
Please note the load capacity of the INTERBUS cable and verify during operation that the supply voltage is maintained at the processor (see Technical Data for specifications).

BIS C-6021...ST8 Interface information / Wiring Diagrams (installation remote bus)

Wiring diagram for BIS C-6021...ST8 processor on installation remote bus

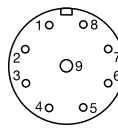


X1, supply voltage and digital input



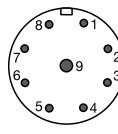
Pin	Function
1	not connected
3	not connected
2	-IN
4	+IN

X2, INTERBUS output



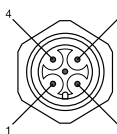
Pin	Function
1	DO2
2	/DO2
3	DI2
4	/DI2
5	GND
9	/RBST
6	FE
7	+24 V
8	0 V

X3, INTERBUS input



Pin	Function
1	DO1
2	/DO1
3	DI1
4	/DI1
5	GND1
6	FE
7	+24 V
8	0 V

X4, Service interface

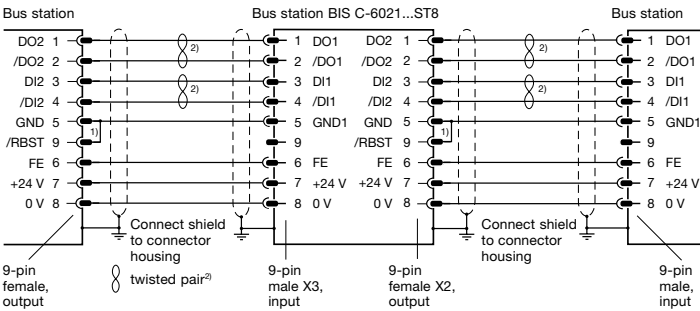


Pin	Function
1	
2	TxD
3	GND
4	RxD

BIS C-6021...ST8
Interface information / Wiring Diagrams (installation remote bus)

**Wiring diagram for
BIS C-6021...ST8
processor on
installation remote
bus**
(continued)

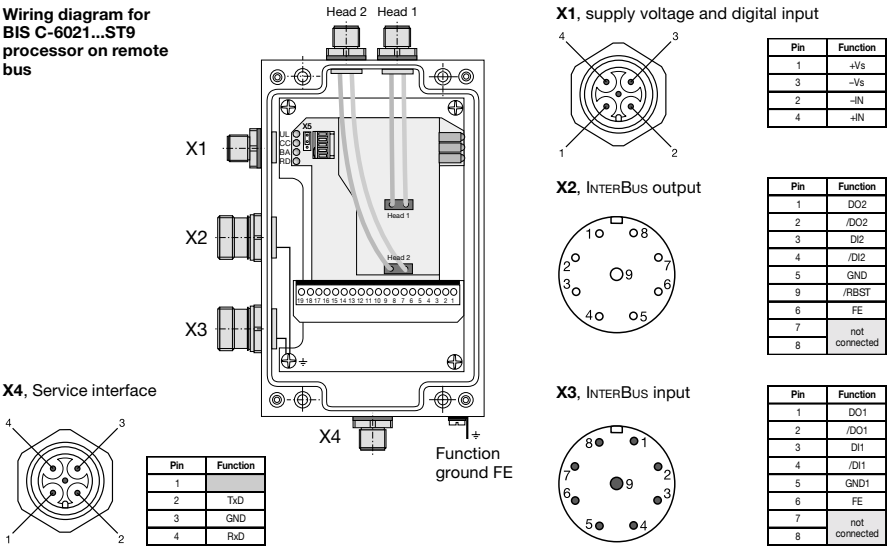
To integrate BIS C-621 processor into the serial INTERBUS, two terminals are provided on the housing, X2 as INTERBUS output and X3 as INTERBUS input. For installation remote bus, the stations are supplied through the bus.



- ¹⁾ Connect the jumper in the connector if another station is to follow. Remove it if no additional station follows.
- ²⁾ The differential signals DO and /DO as well as DI and /DI must be twisted-pair.
Recommended cable: LiYCY 3x2x0.25 mm² (AWG 24); maximum cable capacitance: 120 pF/m

BIS C-6021...ST9
Interface information / Wiring Diagrams (remote bus)

**Wiring diagram for
BIS C-6021...ST9
processor on remote
bus**

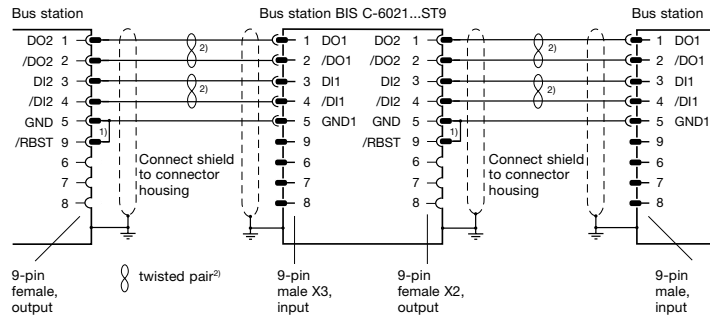


BIS C-6021...ST9

Interface information / Wiring Diagrams (remote bus)

Wiring diagram for BIS C-6021...ST9 processor on remote bus (continued)

To integrate BIS C-621 processor into the serial INTERBUS, two terminals are provided on the housing, X2 as INTERBUS output and X3 as INTERBUS input. For remote bus, the stations are not supplied through the bus.



¹⁾ Connect the jumper in the connector if another station is to follow. Remove it if no additional station follows.

²⁾ The differential signals DO and /DO as well as DI and /DI must be twisted-pair.
Recommended cable: LiYCY 3x2x0.25 mm² (AWG 24); maximum cable capacitance: 120 pF/m

BIS C-6021

Changing the EEPROM

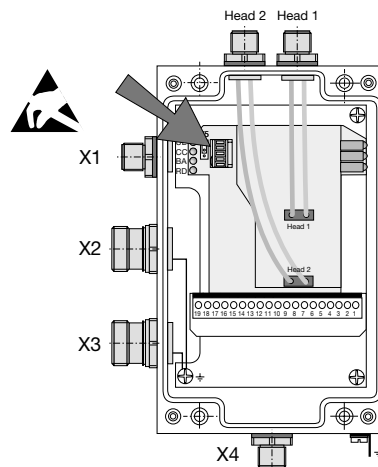
Changing the EEPROM in the BIS C-6021 processor



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 C-6021
Technical Data

Dimensions, weight	Housing	Metal
	Dimensions	190 x 120 x 60 mm
	Weight	820 g
Operating conditions	Ambient temperature	0 °C to +60 °C
Connection type	Integral connector X1	5-pin (male)
	Integral connector Head 1, Head 2	4-pin (male)
	Round connector for X2	9-pin (female)
	Round connector for X3	9-pin (male)
	Integral connector X4	4-pin (male)
Enclosure	Protection class	IP 65 (when connected)
Electrical connections	Supply voltage V_s	DC 24 V $\pm$ 20 %
	Ripple	$\leq$ 10 %
	Current draw	$\leq$ 400 mA
	Connections for supply voltage V_s with installation remote bus with remote bus	at INTERBUS input X3, output X2 at input X1
	Output X2, input X3, INTERBUS	serial interface for remote bus station, Ident-No. 03, 16 bytes IN, 16 bytes OUT (bei Modus BIS C-621: 8 Byte IN, 8 Byte OUT)
	Head 1, Head 2, Read/Write Head	via integrated adapter with 2 x connectors for all read/write heads BIS C-3_ _ with 4-pin connector (female), excluding BIS C-350 and BIS C-352

BIS C-6021
Technical Data

Electrical Connections (continued)	Digital input X1 (+IN, -IN)	galvanically isolated (optocoupler)
	Control voltage active	4 V to 40 V
	Control voltage inactive	1.5 V to -40 V
	Input current at 24 V	11 mA
	Delay time, typ	5 ms
	Service interface X4	RS 232
Function Displays	BIS operating states (LED in housing)	LED red / green LED green / yellow LED green / yellow Ready / Bus active CT1 Present / Operating CT2 Present / Operating
	INTERBUS state (LED on side of housing)	LED green LED green LED green LED yellow Reset Cable Check Bus active Remotebus Disable



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 50081-2 (Emission) and EN 50082-2 (Noise Immunity).

BIS C-6021 **Ordering Information**

Ordering code

BIS C-6021-029-050-03-ST

Balluff Identification System _____
 Type C Read/Write System _____
 Hardware Type _____
 6021 = metal housing, INTERBUS
 (remote bus or installation remote bus)
 Software Type _____
 029 = INTERBUS with memory optimization
 Adapter _____
 050 = with two connections for external read/write heads BIS C-3_ _
 (except BIS C-350 and -352)
 Interface _____
 03 = BUS versions
 User Connection _____
 ST8 = Connector version (installation remote bus)
 ST9 = Connector version (remote bus)
 (2 round connector for power supply/digital input and service interface, 2 round connectors
 for INTERBUS)

Accessory (optional, not included)

Type	Ordering code
Mating connector for X1	BKS-S79-00
Mating connector for X2	BKS-S83-00
Mating connector for X3	BKS-S84-00
Mating connector for X4	BKS-S 10-3
Protective cap for X1, Head 1, Head 2, X4	BES 12-SM-2
Protective cap for X2	115 475

Appendix, ASCII Table

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