

BTL5-D11...

deutsch Betriebsanleitung
Konfiguration
und CAN-Ankopplung

Balluff GmbH
Schurwaldstraße 9
73765 Neuhausen a.d.F.
Deutschland
Telefon +49 7158 173-0
Telefax +49 7158 5010
Servicehotline +49 7158 173-370
DeviceNet@balluff.de
www.balluff.com

DeviceNet™

Eigenschaften

Der Micropulse Wegaufnehmer BTL5-D...bietet Ihnen die Möglichkeiten der BTL-Konfiguration in einem DeviceNet System:

- Betriebsparameter des BTL
- Auflösung von Position und Geschwindigkeit
- Einheiten
- Skalierung
- Nocken / Schaltpunkte mit Polarität und Hysterese
- Grenzwerte
- Arbeitsbereiche

DeviceNet – Kommunikationsparameter:

- Anzahl Ausgabebytes
- Erwartete Übertragungsrate
- Explicit Messaging
- Polled I/O
- BitStrobe I/O
- COS / Cyclic I/O

Busparameter:

- Einstellen der Übertragungsgeschwindigkeit (Baudrate)
- Einstellen der MAC ID

Definitionen

Baudrate	Geschwindigkeit der Datenübertragung auf dem CAN-Bus
CAN	Controller Area Network
MAC ID	Knotennummer zur gerätespezifischen Identifikation (Media Access Control Identifier)
CIP	Common Industrial Protocol
COS	Change of State
ODVA	Open DeviceNet Vendor Association

Inhaltsverzeichnis

1	Systembeschreibung des BTL5-D11... am CAN-Bus	3
1.1	CAN-Technologie	3
1.2	DeviceNet-Technologie	3
1.3	Datenübertragung	3
1.4	Objektmodell	3
1.5	Knotenidentifikation	4
1.6	Betriebsparameter	4
1.7	Systemstart	6
1.8	Abschätzen der Übertragungsdauer	6
1.9	Kabellängen	7
2	Übertragung der Daten in einer I/O-Nachricht	8
2.1	Konfiguration der I/O-Nachrichten	8
2.2	Datenformat für Position und Geschwindigkeit	9
3	Fehlermeldungen und Warnungen	9
3.1	Fehlermeldungen über DeviceNet	9
3.2	Geräteinterne Alarme und Warnungen	10
4	Kommunikation über DeviceNet	11
4.1	Schritt 1: Aufbau der Verbindung	11
4.2	Schritt 2a: Abfragen eines Attributs	11
4.3	Schritt 2b: Setzen eines Attributs	11
4.4	Schritt 2c: Aufruf eines Services	12
4.5	Schritt 3: Lösen der Verbindung	12
5	Parametrierung der DeviceNet Schnittstelle	13
5.1	Auslesen der Ident-Daten	13
5.2	Vergabe der Baudrate	13
5.3	Vergabe der MAC ID	13
5.4	Speicherübernahme	14
6	Konfiguration des Knotens	14
6.1	Betriebsart (Art der Nachrichtenübertragung)	14
6.2	Beispiel für einen Kommunikationsauf- und abbau	14
6.3	Anzahl Positionsgeber	17
6.4	Positionsdaten	18
6.5	Geschwindigkeitseinstellungen	22
7	Weitere Attribute	23
7.1	Position Sensor Objekt	23
7.2	Balluff Configuration Objekt	24
8	Inbetriebnahme	25
8.1	Projektierung unter RsNetworx	25
9	Klassenverzeichnis (Objektverzeichnis)	28
9.1	Identy Klasse (ID = 1)	29
9.2	DeviceNet Klasse (ID = 3)	30
9.3	Assembly Klasse (ID = 4)	30
9.4	Connection Objekt	31
9.5	Position Sensor Objekt	32
9.6	Acknowledge Handler Objekt	34
9.7	Balluff Configuration Objekt	34
10	Weitere Einstellungen	36

1 Systembeschreibung des BTL5-D11... am CAN-Bus

1.1 CAN-Technologie

DeviceNet ist ein offenes System, das auf der Basis von CAN aufsetzt. CAN wurde vor einigen Jahren von der Firma R. Bosch für die Datenübertragung in Kraftfahrzeugen entwickelt. Seitdem sind Millionen von CAN-Chips im Einsatz. Nachteilig für einen Einsatz in der Automatisierungstechnik ist, dass CAN keine Definitionen für die Applikationsschicht enthält. CAN definiert nur die physikalische und die Datensicherungsschicht.

1.2 DeviceNet-Technologie

Mit DeviceNet ist eine einheitliche Applikationsschicht festgelegt, mit der das CAN-Protokoll für Industrieanwendungen nutzbar wird. Die ODVA unterstützt Hersteller und Anwender des Systems DeviceNet als unabhängiger Verein. Die ODVA stellt sicher, dass alle Geräte, die der Spezifikation entsprechen, herstellernerneutral zusammen in einem System arbeiten.

1.3 Datenübertragung

Die Datenübertragung in DeviceNet erfolgt über Nachrichtentelegramme. Grundsätzlich lassen sich die Telegramme schematisch in CAN ID und 8 Folgebytes aufteilen:

CAN ID	Message Header	Message Body
11 Bit	1 Byte	7 Byte

1.3.1 Definition der CAN ID

DeviceNet basiert auf dem Standard-CAN Protokoll und verwendet einen 11 Bit (2048 Nachrichten unterscheidbar) Nachrichten-identifizier. Zur Kennzeichnung eines Gerätes oder Knotens in einem DeviceNet Netzwerk reichen 6 Bit aus, da ein Netzwerk auf 64 Teilnehmer begrenzt ist. Diese wird als MAC ID bezeichnet. Der CAN Identifier setzt sich aus der Kennung der Message Group, der Message ID innerhalb dieser Gruppe und der MAC ID des Gerätes zusammen.

Bei dem BTL5-D11... handelt es sich um einen Group 2 Only Slave. In der unten stehenden Tabelle sind die wichtigsten CAN-IDs für bestimmte Kommunikationsarten dargestellt:

10	9	8	7	6	5	4	3	2	1	0	Verwendung	Bereich [hex]
0	Group 1 Message ID				Source MAC ID						GROUP 1 Message	0x000-0x3ff
0	1	1	0	1	Source MAC ID						Slave's I/O Change of State or Cyclic Message	
0	1	1	1	1	Source MAC ID						Slave's I/O Poll Response or Change of State/ Cyclic Acknowledge Message	
1	0	MAC ID						Group 2 Message ID			GROUP 2 Message	0x400-0x5ff
1	0	Destination MAC ID						0 1 0			Master's Change of State or Cyclic Acknowledge Message	
1	0	Source MAC ID						0 1 1			Slave's Explicit/Unconnected Response Messages	
1	0	Destination MAC ID						1 0 0			Master's Explicit Request Message	
1	0	Destination MAC ID						1 0 1			Master's I/O Poll Command/Change of State/Cyclic Message	
1	0	Destination MAC ID						1 1 0			Group 2 Only Unconnected Explicit Request Message (reserved)	
1	0	Destination MAC ID						1 1 1			Duplicate MAC ID Check Messages	

Tabelle 1-1: Definition der CAN ID

1.4 Objektmodell

DeviceNet beschreibt alle Daten und Funktionen eines Gerätes anhand eines Objektmodells. Mit Hilfe dieser objektorientierten Beschreibung kann ein Gerät mit einzelnen Objekten vollständig definiert werden. Ein Objekt ist bestimmt durch die Zusammenfassung von zugehörigen Attributen (z.B. Prozessdaten), seine nach außen bereitgestellten Funktionen (Lese- oder Schreibzugriff auf ein einzelnes Attribut) sowie durch sein definiertes Verhalten.

DeviceNet unterscheidet zwischen drei Arten von Objekten:

1. Kommunikationsobjekte
Definieren die über DeviceNet ausgetauschten Nachrichten und werden als Connection Objects bezeichnet. (DeviceNet Object, Message Router Object, Connection Object, Acknowledge Handler Object)
2. Systemobjekte
Definieren allgemeine DeviceNet-spezifische Daten und Funktionen. (Identity Object, Parameter Object)
3. Applikationsspezifische Objekte
Definieren gerätespezifische Daten und Funktionen. (Application Object, Assembly Object).

1 Systembeschreibung des BTL5-D11... am CAN-Bus (Fortsetzung)

1.5 Knotenidentifikation

Der Micropulse Wegaufnehmer BTL5-D11... wird am CAN-Bus betrieben und stellt einen Knoten (MAC ID) des Bussystems dar. Er funktioniert in CAN-Netzwerken für DeviceNet und Class 2.

Die eindeutige Identifizierung der BTL-Messsysteme erfolgt über das Identity Objekt, diese Informationen sind fest im Gerät gespeichert, z.B.:

Vendor ID: 0x2B BALLUFF
 Gerätetyp: 0x09 Resolver Device
 Produktcode: 0x01 Profil
 0x02 Stab
 Seriennummer: 957F9
 Produktname: BTL5D

1.6 Betriebsparameter

Für die optimale Industriesteuerung werden zwei verschiedene Nachrichtentypen zur Verfügung gestellt: I/O-Nachrichten (Implicit Messaging) und explizite Nachrichten (Explicit Messaging). Damit ist ein universeller Einsatz des BTL mit DeviceNet gewährleistet.

Mit I/O-Nachrichten werden I/O-Daten in Echtzeit ausgetauscht. Die BTL-Daten Position, Geschwindigkeit und Nockenstatus von allen vier Positionsgebern können über die I/O-Nachricht übertragen werden.

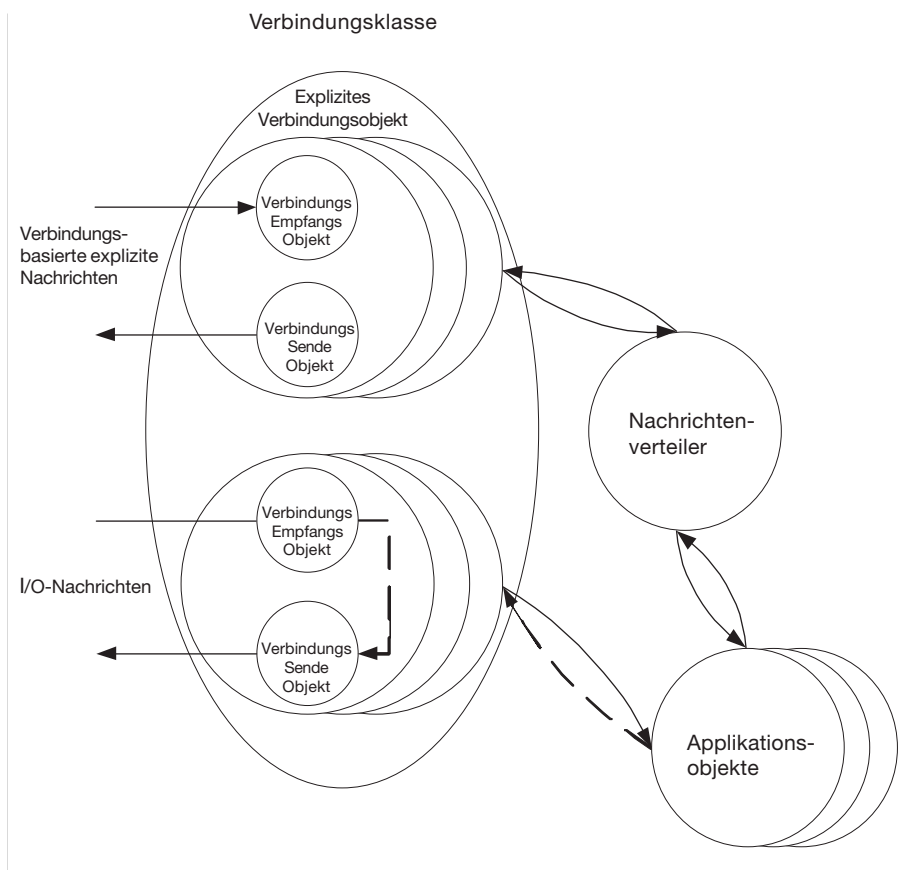


Bild 1-1: Objektmodell des BTL5-D11...

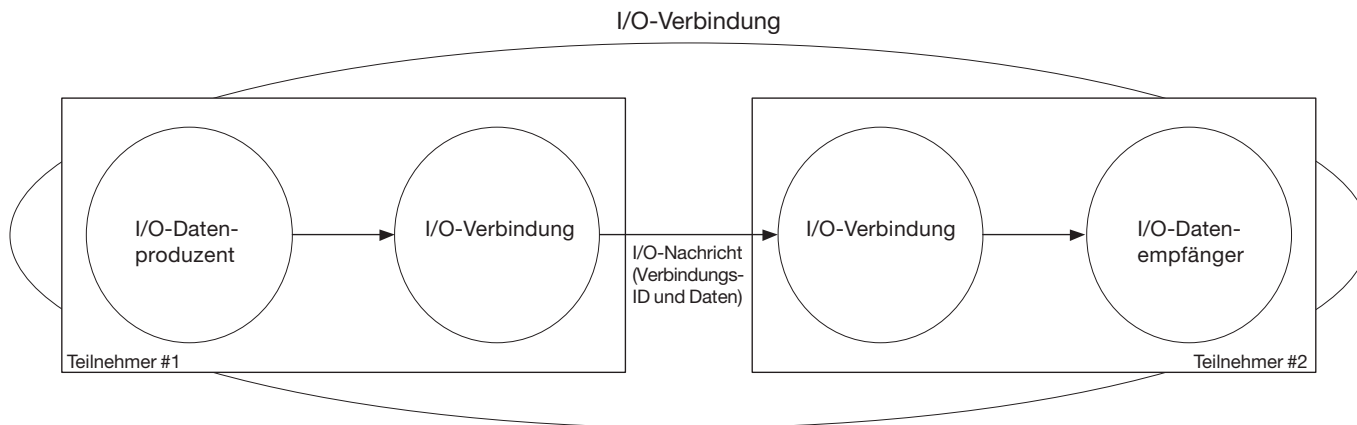


Bild 1-2: I/O-Nachricht

1 Systembeschreibung des BTL5-D11... am CAN-Bus (Fortsetzung)

Mittels expliziter Nachrichten werden Daten zur Konfiguration eines Gerätes, zum Beispiel die Parameter für die Konfiguration des Wegaufnehmers, ausgetauscht. CIP stellt dem Anwender vier wesentliche Funktionalitäten zur Verfügung:

- Einheitliche Steuerungsdienste
- Einheitliche Kommunikationsdienste
- Einheitliche Verteilung von Nachrichten
- Gemeinsame Wissensbasis

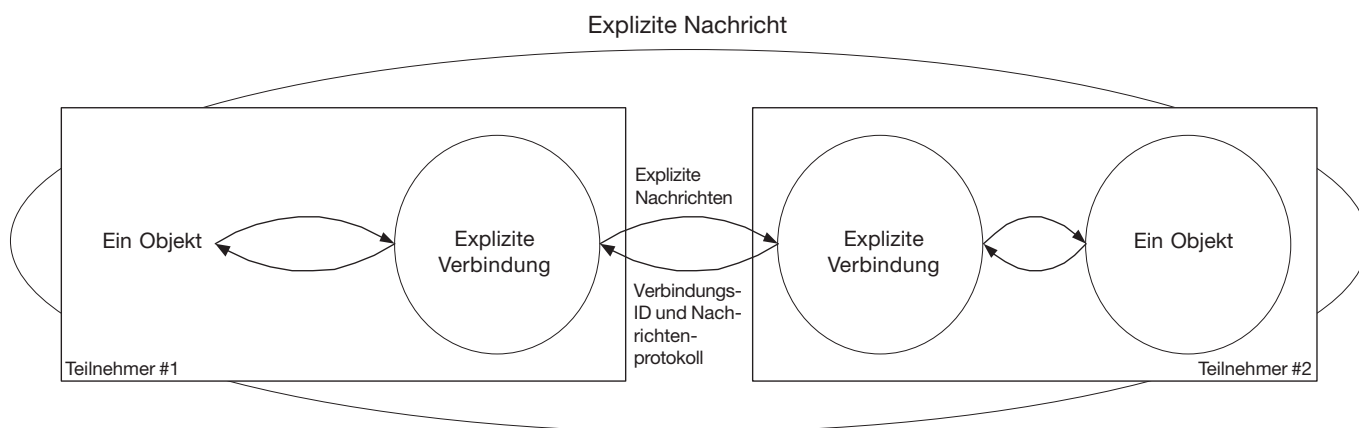


Bild 1-3: explizite Nachricht

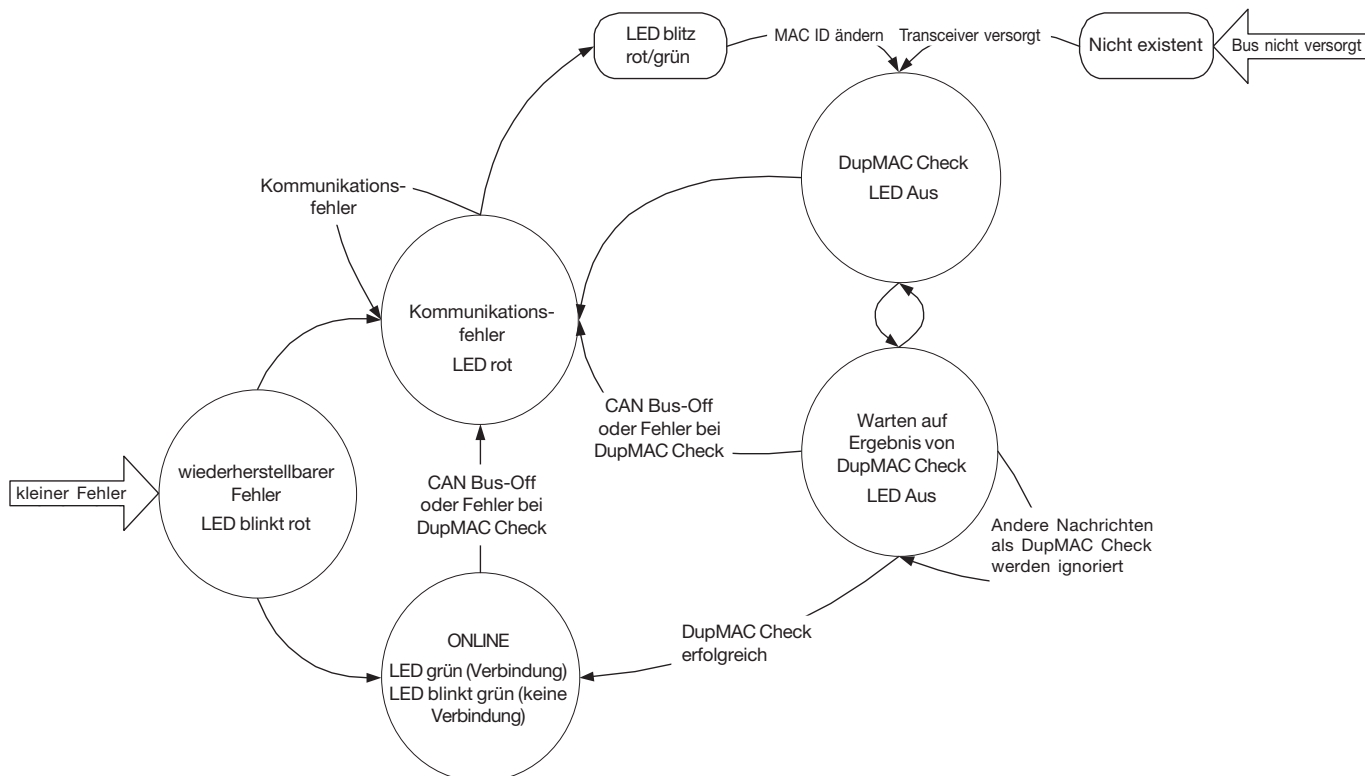


Bild 1-4: Zustandsdiagramm

1 Systembeschreibung des BTL5-D11... am CAN-Bus (Fortsetzung)

Die Micropulse Wegaufnehmer vom Typ BTL5-D11... werden mit dem CIP 2.1 PositionSensor Object ausgeliefert. Die einstellbaren Parameter des DeviceNet Communication Profile und des PositionSensor Objektes sind über explizite Nachricht "Explicit Message" (Set Attribute) einstellbar.

Klasse 1: Identity Objekt (nur lesbar) ➔ Kapitel 9.1

- Hersteller
- Produktkennung
- Produktname
- Seriennummer
- Status

Klasse 3: DeviceNet Objekt ➔ Kapitel 9.2

- Busparameter
- MAC ID
- Baudrate

Klasse 4: Assembly Objekt ➔ Kapitel 9.3

- Kopie der I/O Daten zum Zugriff mittels Explicit Messaging

Klasse 5: Connection Objekt ➔ Kapitel 9.4

- Anzahl Ausgabebytes
- Erwartete Übertragungsrate
- Explicit Messaging
- Polled I/O
- BitStrobe I/O
- COS / Cyclic

Klasse 35: Position Sensor Objekt ➔ Kapitel 9.5

- Betriebsparameter des BTL
- Auflösung von Position und Geschwindigkeit
- Einheiten
- Skalierung
- Nocken / Schaltpunkte mit Polarität und Hysterese
- Grenzwerte
- Arbeitsbereiche

Klasse 43: Acknowledge Handler Objekt ➔ Kapitel 9.6

- Verwaltet Nachrichtenbestätigungen

Klasse 100: Balluff Configuration Objekt ➔ Kapitel 9.7

- Konfiguration der I/O-Nachricht

- Spezielle Konfigurationsparameter

1.7 Systemstart

Nach dem Systemstart (bzw. einem Reset) wird das BTL entsprechend dem Zustandsdiagramm gestartet ➔ Bild 1-4.

1.7.1 DupMAC Check

Nach dem Einschalten sendet jeder Knoten eine Duplicate MAC ID Check Request Nachricht. Diese Nachricht sendet der Knoten praktisch an seine eigene Adresse. Die Daten der Nachricht enthalten die ID des Herstellers und die Seriennummer des Knotens. Die LED bleibt aus.

1.7.2 Warten auf Ergebnis von DupMAC Check

Nach dem Senden der Duplicate MAC ID Check Request Nachricht wartet der Knoten maximal eine Sekunde auf das Eintreffen einer Duplicate MAC ID Check Response Nachricht. Wenn die vom Knoten verwendete Adresse tatsächlich nur einmal im Netzwerk vorkommt, dann bleibt die DupMAC-Nachricht ohne Antwort (Response). Nach Ablauf der Sekunde steht damit fest, dass die Adresse des Knotens nur einmal im Netzwerk vorhanden ist und der Knoten wechselt in den Zustand Online.

Sollte später noch eine Response empfangen werden, dann wechselt das Gerät sofort in die Betriebsart Kommunikationsfehler und die LED leuchtet statisch rot.

Existiert ein anderer Knoten mit der gleichen MAC ID, dann wird dieser durch die Duplicate MAC ID Check Request Nachricht adressiert und sendet eine Duplicate MAC ID Check Response Nachricht. Damit steht für den Sender der Duplicate MAC ID Check Request Nachricht fest, dass seine Adresse nicht mehr frei ist und er wechselt in den Zustand Kommunikationsfehler.

1.7.3 Online

Im Zustand ONLINE ist das Gerät betriebsbereit. Wird das Gerät adressiert oder sendet es Nachrichten, dann leuchtet die LED statisch grün. Wurde das Gerät innerhalb der letzten 10 Sekunden nicht adressiert oder hat es in diesem Zeitraum selbst keine Nachrichten gesandt, dann schließt das Gerät die Verbindungen und die LED blinkt grün.

1.7.4 Kommunikationsfehler

Ist der CAN-Bus gestört, fehlt die Spannungsversorgung auf dem Bus oder wurde durch einen Duplicate MAC ID Check festgestellt, dass die eigene MAC ID bereits belegt ist, dann wechselt der Knoten in diesen Zustand. Die rote LED leuchtet statisch rot.

Das Gerät lässt sich weiter durch eine Communicated Faulted Message Request adressieren. In diesem Zustand kann man das Gerät anhand seiner Herstellerkennung und Seriennummer identifizieren. Auch kann man ihm eine neue MAC ID zuweisen.

1.7.5 Wiederherstellbarer Fehler

Bei kleineren Fehlern, wenn zum Beispiel eine Duplicate MAC ID Check Request Nachricht mit eigener Adresse empfangen wurde, blinkt die LED rot. Das Gerät ist im Offline Modus, ist aber grundsätzlich noch kommunikationsbereit.

1.8 Abschätzen der Übertragungsdauer

Da es Abhängigkeiten zwischen der CAN-Bus-Leitungslänge und der Baudrate sowie der Wahl des Datenintervalls gibt, ist die Abschätzung der Übertragungsdauer bei kontinuierlicher Übertragung für die Einstellungen der Baudrate und des Datenintervalls von Bedeutung.

Eine CAN I/O-Nachricht im Auslieferungszustand besteht aus folgenden Daten:

- 35 Bit für die Übertragung im CAN Protokoll
- 4 Byte Position à 8 Bit
- 1 Byte für Nockenstatus mit 8 Bit

1 Systembeschreibung des BTL5-D11... am CAN-Bus (Fortsetzung)

Eine Nachricht im Standard-Format mit 8 Datenbytes benötigt maximal 130 Bits. Dabei geht man von einer maximalen Anzahl von 19 Stuff-Bits und 3 Zwischenraumbits aus.

Baudrate	Übertragungs- dauer	mögliche Daten- intervalle
[kBaud]	t / ms	t / ms
500	0,212	$\geq 0,250$
250	0,424	$\geq 0,500$
125	0,848	$\geq 1,000$

Sollen mehr als 8 Byte übertragen werden, dann werden die Nachrichten automatisch fragmentiert. Da dann pro CAN-Nachricht jeweils 1 Byte für die Fragmentierungskennung entfällt, verlängert sich die Übertragungsdauer dann überproportional.

Die Übertragungsdauer in Abhängigkeit von der Baudrate kann nebenstehendem Diagramm entnommen werden. Die Stufen entstehen, wenn jeweils ein weiteres Fragment nötig wird ➡ Bild 1-5.

1.9 Kabellängen

In Abhängigkeit von der gewählten Baudrate sind unterschiedliche maximale Kabellängen möglich. Die aufgeführten Kabellängen sind als Summe von Hauptstrang und Stichleitungen zu betrachten.

Baudrate	Kabellänge
125 kBaud	500 m
250 kBaud	250 m
500 kBaud	100 m

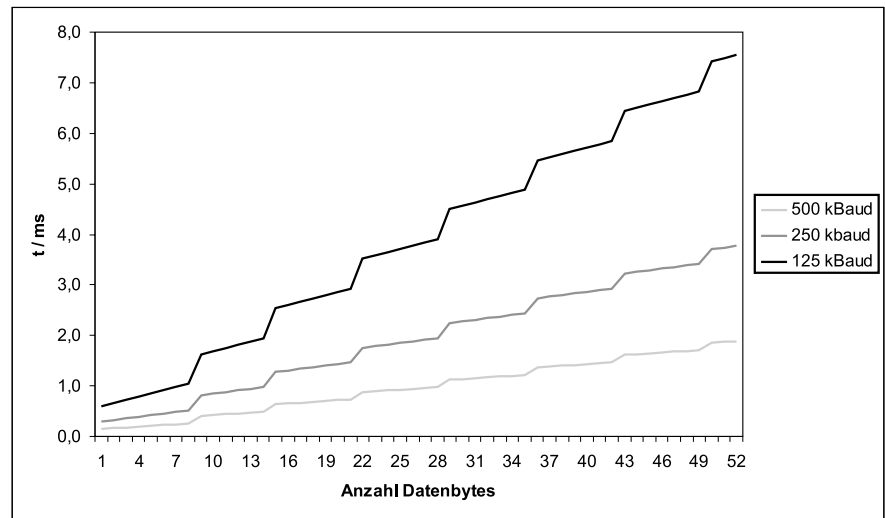


Bild 1-5: Diagramm - Übertragungsdauer

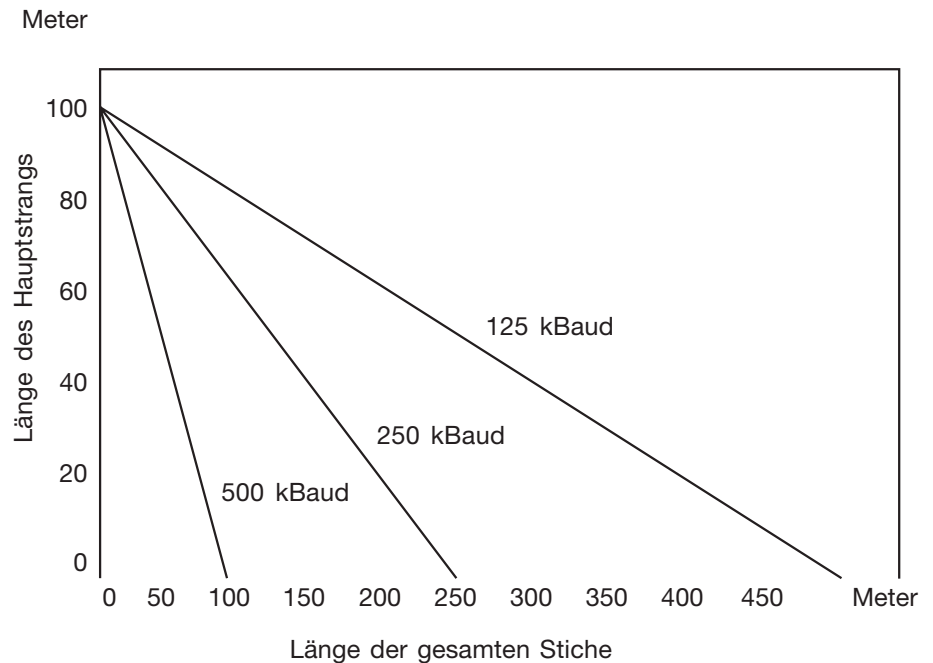


Bild 1-6: Kabellängen

$$L_{\text{stich}} + 5 \times L_{\text{haupt}} = 500 \text{ m} \quad \text{bei 125 kBaud}$$

$$L_{\text{stich}} + 2,5 \times L_{\text{haupt}} = 250 \text{ m} \quad \text{bei 250 kBaud}$$

$$L_{\text{stich}} + L_{\text{haupt}} = 100 \text{ m} \quad \text{bei 500 kBaud}$$

L_{stich} ist die Länge des Stichkabels und L_{haupt} ist die Länge des Hauptkabels.

2 Übertragung der Daten in einer I/O-Nachricht

I/O-Nachrichten können gepolt werden, zyklisch oder im Falle einer Änderung der Daten automatisch gesendet oder zyklisch abgefragt werden. Der Prozess-Istwert wird entsprechend dem Telegramm-Schema übertragen ➡ Tabelle 2-1:

Dieses Beispiel entspricht dem voreingestellten Telegramm, in dem der aktuelle Positionswert des ersten Positionsgebers in den Bytes 0 bis 3 und sein Nockenstatus (CAM) im Byte 4 übertragen wird. Mit Hilfe der in Abschnitt 9.7 "Balluff Configuration Objekt" beschriebenen Optionsattribute können für alle vier möglichen Positionsgeber wahlweise Positionsdaten, Geschwindigkeit, Nockenstatus, Warnungen oder Alarime in beliebiger Kombination in die I/O-Nachricht kopiert werden.

Dabei sollte beachtet werden, dass maximal 8 Bytes (z.B. zwei Positionswerte) in einer CAN-Nachricht übertragen werden können, bevor automatisch fragmentierte Nachrichten gesendet werden, die auf mehrere Telegramme aufgeteilt werden.

2.1 Konfiguration der I/O-Nachrichten

In die I/O-Nachricht können für alle vier Positionsgeber die Attribute des Position Sensor Objects

- Position Position Sensor Objekt Attribut 10: Position Value Signed
- Geschwindigkeit Position Sensor Objekt Attribut 24: Velocity Value
- Nockenschalter Position Sensor Objekt Attribut 35: CAM Channel State Register
- Warnungen Position Sensor Objekt Attribut 47: Warnings
- Alarime Position Sensor Objekt Attribut 44: Alarms

aktiviert werden. Hierzu gibt es im Abschnitt 9.7 "Balluff Configuration Objekt" die Attribute

- Attribut 6: Assembly Position Active: Aktiviert Position Value Signed
- Attribut 7: Assembly Velocity Active: Aktiviert Velocity Value

CAN ID	Positionswert				CAM
11 Bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
500	2 ⁷ to 2 ⁰	2 ¹⁵ to 2 ⁸	2 ²³ to 2 ¹⁶	2 ³¹ to 2 ²⁴	2 ⁷ to 2 ⁰

Tabelle 2-1: Telegramm-Schema

Aufbau von Attribut 6 ... 10 im Balluff Configuration Object																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Positionsggeber	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabelle 2-2: Objektkonfiguration

- Attribut 8: Assembly CAM Activ: Aktiviert CAM Channel State Register
- Attribut 9: Assembly Warnings Active: Aktiviert Warnings
- Attribut 10: Assembly Alarms Active: Aktiviert Alarms

In jedem dieser Attribute ist jeweils ein Bit einem Positionsgeber zugeordnet. Wird dieses Bit gesetzt, dann wird der entsprechende Wert in die I/O-Nachricht übertragen.

Wird beispielsweise Bit 2 im Attribut "Assembly Position Active" gesetzt, dann wird der Positionswert von Positionsgeber 3 in die I/O-Nachricht übertragen.

Beispiel: Sie möchten im I/O Positionswert von Positionsgeber 2, Nockenstatus von Positionsgeber 3 und Warnungen von Positionsgeber 1 aktivieren. Dann setzen Sie die Attribute wie folgt:

Position2: Attribut 6: Assembly Position Active Bit 1 -> binär 0000 0000 0000 0010 = 0x02 (Hex)

Nocken3: Attribut 8: Assembly CAM Active Bit 2 -> binär 0000 0000 0000 0100 = 0x04 (hex)

Warnungen1: Attribut 9: Assembly Warnings Active Bit 0 -> binär 0000 0000 0000 0001 = 0x01 (hex)

2.1.1 Übertragungsreihenfolge

Für die Reihenfolge der Daten in der I/O-Nachricht gibt es zwei Möglichkeiten. Im Auslieferungszustand ist die Anordnung voreingestellt ➡ Tabelle 2-3. Diese Reihenfolge ist zu bevorzugen, da sie die schnellere Variante ist.

Die Übertragungsreihenfolge ist durch die links stehende Ordnungszahl definiert. Unabhängig von der Reihenfolge ihrer Aktivierung werden die Werte in der Reihenfolge ihrer Anordnung übertragen.

Für unser Beispiel aus dem vorherigen Abschnitt bedeutet dies:

Position2 hat die Ordnungsnummer 6, Nocken3 hat die Ordnungsnummer 13 und Warnungen1 die Ordnungsnummer 4.

Also ist die Anordnung Warnungen1 an erster Stelle, dann Position2 und zuletzt Nocken3

➡ Tabelle 2-4:

Assembly Options Bit 0 = 0 (Default)									
1	Position1	2	Geschwindigkeit1	3	Nocken1	4	Warnungen1	5	Alarime1
6	Position2	7	Geschwindigkeit2	8	Nocken2	9	Warnungen2	10	Alarime2
11	Position3	12	Geschwindigkeit3	13	Nocken3	14	Warnungen3	15	Alarime3
16	Position4	17	Geschwindigkeit4	18	Nocken4	19	Warnungen4	20	Alarime4

Tabelle 2-3: Übertragungsreihenfolge Default

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Warnungen1	Position2			Nocken3	

Tabelle 2-4: Übertragung - Beispiel 1

2 Übertragung der Daten in einer I/O-Nachricht (Fortsetzung)

Eine alternative Übertragungsreihenfolge kann aktiviert werden, wenn das Bit 0 im Attribut 5: Assembly Options im Balluff Configuration Objekt gesetzt wird (➔ Kapitel 9.7). Dann gilt die Anordnung ➔ Tabelle 2-5:

Hier sind die Ordnungszahlen anders angeordnet. In dem Fall würde in unserem Beispiel Position2 die Ordnungsnummer 2, Nocken3 die Ordnungsnummer 11 und Warnungen1 die Ordnungsnummer 13 bekommen.

Eine Änderung der Konfiguration der I/O-Nachrichten wirkt sich nicht auf eine bereits bestehende I/O-Verbindung aus und muss vor dem Aufbau der I/O-Verbindung erfolgen.

2.2 Datenformat für Position und Geschwindigkeit

Die aktuellen Daten des Wegaufnehmers (Position, Geschwindigkeit) werden als Datentyp DINT bereitgestellt. Die Rahmenbedin-

Assembly Options Bit 0 = 1									
1	Position1	5	Geschwindigkeit1	9	Nocken1	13	Warnungen1	17	Alarmer1
2	Position2	6	Geschwindigkeit2	10	Nocken2	14	Warnungen2	18	Alarmer2
3	Position3	7	Geschwindigkeit3	11	Nocken3	15	Warnungen3	19	Alarmer3
4	Position4	8	Geschwindigkeit4	12	Nocken4	16	Warnungen4	20	Alarmer4

Tabelle 2-5: Übertragungsreihenfolge

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Position2				Nocken3	Warnungen1

Tabelle 2-6: Übertragung - Beispiel 2

gungen (Einheiten, Schrittweite, Offset, ...) werden in der zugehörigen Instanz der Position Sensor Klasse eingestellt, die in Kapitel 9.5 "Position Sensor Objekt" beschrieben ist.

Die Codierung erfolgt entsprechend dem Standard nach ICC 754-1985 und kann somit von den meisten Prozessoren und Compilern ohne weitere Umrechnung direkt verarbeitet werden.

3 Fehlermeldungen und Warnungen

3.1 Fehlermeldungen über DeviceNet

Nach dem Aufruf eines Dienstes (Service) wird unter Umständen eine Error Response Nachricht zurückgesendet.

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3
ID	Source	Response	Error Code	Add. Code

Tabelle 3-1: Error Response Nachricht

Diese enthält einen generellen Fehlercode und einen zusätzlichen Fehlercode, der mitunter weitere Informationen über den aufgetretenen Fehler enthält. Nachfolgend werden die generellen Fehlercodes, die beim BTL auftreten können, aufgelistet.

3.1.1 Code 0: Service Success

Der Service wurde von dem spezifizierten Objekt erfolgreich ausgeführt.

3.1.2 Code 2: Ressource Unavailable

Es stehen keine Kapazitäten bereit, um eine Anfrage zu bearbeiten. Dies ist zum Beispiel der Fall, wenn eine Change of State-Verbindung besteht und zusätzlich eine Poll-Verbindung etabliert werden soll, auch wenn eine Verbindung gelöst werden soll und ein anderer Allocation Choice eingestellt ist, als beim Aufbau der Verbindung.

3.1.3 Code 8: Service Not Supported

Ein Dienst steht für das entsprechende Objekt nicht zur Verfügung. Zum Beispiel steht der Reset Service nur bei bestimmten Objekten zur Verfügung. Wenn im Balluff Konfigurations-Objekt das Attribut Autosave gesetzt ist, dann steht der Save-Service im Balluff Configuration Objekt und im Position Sensor Objekt nicht zur Verfügung und man erhält diesen Fehlercode.

3.1.4 Code 9: Invalid Attribute Value

Es wurde versucht, ein Attribut auf einen ungültigen Wert zu setzen. Viele Attribute sind in ihrem Wertebereich beschränkt, zum Beispiel darf eine Adresse nur zwischen 0 und 63 liegen.

3.1.5 Code 11: Already in requested Mode / State

Dieser Fehler kann in zwei Fällen auftreten:

Eine I/O-Verbindung kann nicht erneut zugewiesen werden, wenn sie zuvor bereits zugewiesen war, auch wenn sie sich im Timeout-Zustand befindet. Lösen Sie zuerst die Verbindung auf.

Es wurde versucht, eine Verbindung aufzulösen, die bereits aufgelöst wurde oder noch nicht besteht. Mitunter erscheint diese Fehlermeldung, wenn eine Verbindung bereits vom BTL in den Timeout-Zustand gesetzt wurde,

3 Fehlermeldungen und Warnungen (Fortsetzung)

bevor der Master sie auflösen möchte. Auch erscheint die Fehlermeldung nach einem ausgeführten Reset-Dienst, da aus der Sicht des Masters die Verbindung noch besteht, die der Slave durch seinen Reset "vergessen" hat. In dem Fall kann diese Fehlermeldung ignoriert werden.

3.1.6 Code 12: Object State Conflict

Ein Objekt kann den geforderten Dienst nicht in seinem aktuellen Modus oder Zustand ausführen. Beim Position Sensor Objekt erscheint dieser Fehler, wenn das Attribut Preset Value gesetzt wird, obwohl zur Zeit ein Positionsfehler besteht und kein gültiger Positionswert für diese Funktion ermittelt werden kann.

3.1.7 Code 14: Attribute not Setable

Ein Attribut kann nicht gesetzt werden. Attribute, die nur mit R und nicht mit R/W bezeichnet sind, können nur gelesen werden.

3.1.8 Code 19: Not Enough Data

Die Anzahl der Daten zum Setzen eines Attributes ist nicht ausreichend. Zum Beispiel wurden nur zwei Datenbytes zum Setzen eines vier Byte Attributes gesendet.

3.1.9 Code 20: Attribute Not Supported

Ein Attribut existiert nicht. Ursache ist meist eine falsch eingegebene Attributnummer. Einige Attribute stehen mitunter nur unter bestimmten Bedingungen zur Verfügung.

3.1.10 Code 22: Object does not exist:

Es wurde versucht, auf ein Objekt zuzugreifen, dass in diesem Gerät nicht existiert. Ursache ist meist eine falsch eingegebene Klassennummer.

3.1.11 Code 23: Fragmentation Segment Error

Bei der Übertragung einer Fragmented Message wurde ein Segment nicht oder nicht richtig übertragen.

3.1.12 Code 25: Store Operation

Eine Speicheroperation in einem nichtflüchtigen Speicherbereich (EEPROM) ist fehlgeschlagen.

3.1.13 Code 32: Invalid Parameter

Es wurde versucht, einen ungültigen Parameter zu setzen, der außerhalb des zulässigen Bereichs liegt.

3.1.14 Klassenspezifische Fehlercodes

Die folgenden Fehlercodes treten nur in Verbindung mit bestimmten Klassen auf. Sie werden als zusätzlicher Code in Verbindung mit dem Hauptfehlercode ausgegeben:

3.2 Geräteinterne Alarmer und Warnungen

Im Position Sensor Object gibt es die Attribute 44 (Alarms), 46 (Alarm Flag), 47 (Warnings) und 49 (Warning Flag) in denen geräteinterne Warnungen und Alarmer ausgegeben werden. Diese Attribute können mit Explicit Messaging abgefragt werden. Die Bedingungen, die zu diesen Alarmen und Warnungen führen, können vom Anwender beeinflusst werden (➔ Kapitel 9.5 Position Sensor Objekt).

Die Alarmer (Attribut 44) und Warnungen (Attribut 47) des Position Sensor Objects können zusätzlich mit Hilfe von Attribut 9 (Assembly Warnings Active) und Attribut 10 (Assembly Alarms Active) in das Assembly kopiert werden.

Code		Beschreibung
1	Allocation Conflict	Es wurde versucht, den Predefined Master Slave Connection Set von einer MAC ID aufzurufen, die nicht mit der gespeicherten Master MAC ID übereinstimmt.
2	Invalid Allocation Choice	Dieser Code tritt in Verbindung mit Code Ressource Unavailable, Invalid Attribute Value und Already in Mode / State auf und zeigt an, dass die gewünschte Zuordnung nicht möglich ist.
3	Invalid Unconnected Request	Eine Group 2 Unconnected Request Nachricht ist ungültig.
4	Object Specific Error	In Verbindung mit Ressource Unavailable. Es wurde versucht, zusätzlich zu einer COS / Cyclic-Verbindung noch eine Poll-Verbindung zu etablieren:

Tabelle 3-2: Klassenspezifische Fehlercodes

4 Kommunikation über DeviceNet

Für die meisten Anwendungsfälle läuft die Kommunikation immer nach dem gleichen Schema ab. Zuerst muss eine Verbindung zwischen Master und dem Slave etabliert werden. Dann können über diese Verbindung Daten ausgetauscht werden. Dabei können mehrere Attribute gelesen oder gesetzt werden. Schließlich beendet man die Verbindung wieder, wenn sie nicht mehr benötigt wird. Nach 10 Sekunden wird die Verbindung automatisch wegen Zeitüberschreitung beendet. In den folgenden Beispielen hat der Master die MAC ID 0 und das BTL die MAC ID 63. Die Bytes, die sich nicht in Abhängigkeit von den MAC IDs oder den abgefragten Attributen ändern, sind grau hinterlegt.

4.1 Schritt 1: Aufbau der Verbindung

Eine Verbindung wird aufgebaut, indem man das Master Connection Set zuweist:

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FE	0x00	0x4B	0x03	0x01	0x01	0x00
	Source	Service	Klassen ID	Instanz ID	Explicit	Allocator MacID

Tabelle 4-1: Master Connection Set

Die CAN ID 0x5FE ist binär 10 111111 110. Die ersten zwei Binärziffern sind 10 für die Kennung der Nachrichtengruppe 2. Die folgenden 6 Bits sind die Adresse des Slaves (BTL). Die letzten 3 Bits enthalten die Nachrichten ID. 0x06 (binär 110) ist die Kennung eines Unconnected Explicit Requests.

Das Byte 0 enthält die Source MAC ID, hier 0x00. Byte 1 ist 0x4B und kennzeichnet den Service "Allocate M/S Connection Set". Byte 2 spezifiziert die Klassen ID. 0x03 ist die ID der im Abschnitt 9.2 beschriebenen DeviceNet Klasse. Die Instanz ID in Byte 3 zeigt auf Instanz 1 und in Byte 4

CAN ID	Byte 0	Byte 1	Byte 2
0x5FB	0x00	0xCB	0x00
	Source	Response	Error Code

Tabelle 4-2: Response Nachricht

ist das Bit für eine Explizite Verbindung gesetzt. Byte 5 enthält schließlich die MAC ID, zu der diese Verbindung zugeordnet ist. Hier wird wieder die Master MAC ID eingesetzt.

4.2 Schritt 2a: Abfragen eines Attributs

Die CAN ID 0x5FC setzt sich genau so, wie bereits zuvor beschrieben, zusammen. Jedoch sind die letzten drei Bits jetzt 100 und kennzeichnen eine "Masters Explicit Request Message". Das Byte 1 steht für den Service "Get Attribute Single". Die Bytes 2 bis 4 enthalten Klassen (Objekt) ID, Instanz ID und Attribut ID. Hier wird Klasse 3 (Device Net), Instanz 1 und Attribut 1 (MAC ID) abgefragt.

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x5FC	0x00	0x0E	0x03	0x01	0x01
	Source	Service	Klassen ID	Instanz ID	Attribut ID

Tabelle 4-3: Masters Explicit Request Message

Man erhält als Response den gewünschten Wert:

CAN ID	Byte 0	Byte 1	Byte 2
0x5FB	0x00	0x8E	0x3F
	Source	Response	Wert

Tabelle 4-4: Response Wert

4.3 Schritt 2b: Setzen eines Attributs

Das Byte 1 steht für den Service "Set Attribute Single". Die Bytes 2 bis 4 enthalten Klassen (Objekt) ID, Instanz ID und Attribut ID. Hier wird Klasse 35 (Position Sensor Object), Instanz 1 (Positionsgeber 1), Attribut 12 (Direction Counting Toggle) auf den in Byte 5 übergebenen Wert 1 gesetzt.

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FC	0x00	0x10	0x23	0x01	0x0C	0x01
	Source	Service	Klassen ID	Instanz ID	Attribut ID	Wert

Tabelle 4-5: Set Attribute Single

Man erhält als Response die Ausführung bestätigt oder eine Error Response:

CAN ID	Byte 0	Byte 1
0x5FB	0x00	0x90
	Source	Response

Tabelle 4-6: Response (Attribut setzen)

4 Kommunikation über DeviceNet (Fortsetzung)

4.4 Schritt 2c: Aufruf eines Services

Das Byte 1 steht für den Service. Hier wird der Service "Reset" aufgerufen. Die Bytes 2 und 3 enthalten Klassen (Objekt) ID und Instanz. Da der Reset Service immer auf eine ganze Instanz wirksam ist, entfällt die Angabe der Attribut ID. Hier wird Klasse 100 (Balluff Configuration Objekt), Instanz 1 zurückgesetzt.

Bei einigen Objekten hat der Reset Befehl noch ein zusätzliches Argument. Ist dieses 0, dann wird nur ein Reset durchgeführt, als wenn die Betriebsspannung aus- und

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FC	0x00	0x05	0x64	0x01	0x01	0x01
	Source	Service	Klassen ID	Instanz ID	Attribut ID	Argument

Tabelle 4-7: Reset Service

wieder eingeschaltet wird. Ist das Argument 1, dann wird die Instanz in ihren Auslieferungszustand zurückgesetzt und anschließend noch ein Reset durchgeführt.

Man erhält als Response die Ausführung bestätigt oder eine Error Response:

CAN ID	Byte 0	Byte 1
0x5FB	0x00	0x85
	Source	Response

Tabelle 4-8: Response (Serviceaufruf)

4.5 Schritt 3: Lösen der Verbindung

Diese Sequenz ähnelt der, mit der anfangs die Verbindung hergestellt wurde. Lediglich ist jetzt der Service Code 0x4C für "Release M/S Connection Set". Byte 5 wird zum Schließen der Verbindung nicht benötigt.

5FE 00 4C 03 01 01

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x5FE	0x00	0x4C	0x03	0x01	0x01
	Source	Service	Klassen ID	Instanz ID	Explicit

Tabelle 4-9: Release M/S Connection Set

5 Parametrierung der DeviceNet Schnittstelle

5.1 Auslesen der Ident-Daten

Um die Ident-Daten zu ermitteln, braucht man das Gerät nur einzuschalten, denn es sendet dann automatisch einen Duplicate MAC ID Check, der aus zwei identischen Nachrichten besteht.

Die ID enthält in den ersten zwei Binärziffern 10 als Kennung einer Group 2 Nachricht. Die nächsten 6 Bits stellen die MAC ID des BTL dar (hier dezimal 63). Die letzten drei Bits kennzeichnen die Nachricht als Duplicate MAC ID Request.

5.2 Vergabe der Baudrate

Die Baudrate ist in Klasse 3 (DeviceNet), Instanz 1, Attribut 2 als USINT abgelegt. Sie kann mit folgender Nachrichtensequenz geändert werden:

Sendung des Masters:

0x5FE 0x00 0x4B 0x03 0x01 0x01 0x00

➔ Kapitel 4.1: Schritt 1: Aufbau der Verbindung

Antwort des BTL:

0x5FB 0x00 0xCB 0x00

➔ Kapitel 4.3: Schritt 2b: Setzen eines Attributs. Der übergebene Wert entspricht der in DeviceNet üblichen Konvention:

5.3 Vergabe der MAC ID

Die MAC ID ist in Klasse 3 (DeviceNet), Instanz 1, Attribut 1 als USINT abgelegt. Sie kann mit folgender Nachrichtensequenz geändert werden:

Sendung des Masters:

0x5FE 0x00 0x4B 0x03 0x01 0x01 0x00

➔ Abschnitt 4.1: Schritt 1: Aufbau der Verbindung

Antwort des BTL:

0x5FB 0x00 0xCB 0x00

➔ Kapitel 4.3: Schritt 2b: Setzen eines Attributs. Der übergebene Wert für die MAC ID muss zwischen 0 und 63 liegen.

Eine Nachricht hat dabei folgendes Format:

0x5FF	0x00	0x2B	0x00	0xF9	0x57	0x09	0x00
		Lo	Hi	Lo			Hi
ID	Phys.	Hersteller ID		Seriennummer			

Tabelle 5-1: Duplicate MAC ID Request Nachricht

Die Hersteller ID ist immer 0x00 2B und kennzeichnet Balluff GmbH.

Die letzten vier Bytes enthalten die Seriennummer dieses Geräts.

Diese Attribute können auch mit Hilfe des in Kapitel 9.1 Identity Klasse (ID = 1) beschriebenen Identity Objects abgefragt werden.

Sendung des Masters

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FC	0x00	0x10	0x03	0x01	0x02	Baud
	Source	Service	Klassen ID	Instanz ID	Attribut ID	Wert

Tabelle 5-2: Vergabe Baudrate

Baud	Bedeutung
0	125 kBaud
1	250 kBaud
2	500 kBaud

Response des BTL:

0x5FB 0x00 0x90

Master:

0x5FE 0x00 0x4C 0x03 0x01 0x01

➔ Kapitel 4.5 Schritt 3: Lösen der Verbindung

BTL:

0x5FB 0x00 0xCC

Diese Änderung betrifft nur die intern gespeicherte Baudrate, die nur dann verwendet und geändert werden kann, wenn sich DIP-Schalter 8 und 9 in Stellung "On" befinden. Die Änderung wird erst nach dem nächsten Reset wirksam.

Sendung des Masters

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FC	0x00	0x10	0x03	0x01	0x01	MacID
	Source	Service	Klassen ID	Instanz ID	Attribut ID	Wert

Tabelle 5-3: Vergabe MAC ID

Response des BTL:

0x5FB 0x00 0x90

Wenn DIP-Schalter 7 in Stellung "On" ist, dann ist die intern gespeicherte MAC ID aktiv. In dem Fall führt das BTL nach Änderung der MAC ID sofort einen Reset durch und sendet einen 'Duplicate MAC ID Check' mit seiner neuen MAC ID.

Master:

0x5FE 0x00 0x4C 0x03 0x01 0x01

➔ Kapitel 4.5 Schritt 3: Lösen der Verbindung

BTL:

0x5FB 0x00 0xCC

Diese Änderung betrifft nur die intern gespeicherte MAC ID, die nur geändert werden kann, wenn sich DIP-Schalter 7 in Stellung "On" befindet.

5 Parametrierung der DeviceNet Schnittstelle (Fortsetzung)

5.4 Speicherübernahme

Attribute, die im Klassenverzeichnis (➔ Kapitel 9) mit NV gekennzeichnet sind, werden im nichtflüchtigen Speicher (EEPROM) automatisch gespeichert, wenn sie geändert werden.

Bei Attributen des im Kapitel 9.5 beschriebenen Position Sensor Objekt kann die automatische Speicherung abgeschaltet werden.

Hierzu gibt es das im Balluff Configuration Objekt "Attribut 4: Autosave". Ist Autosave deaktiviert, dann erfolgt eine Speicherung der geänderten Daten des Position Sensor Objektes nur nach Aufruf des Save Services. In der normalen Anwendung, in der das BTL vor seiner Verwendung einmalig parametrierung wird, sollte Autosave aktiviert bleiben.

Lediglich in Spezialfällen, in denen während des Betriebs fortwährende Änderungen der mit NV gekennzeichneten Attribute erfolgen, empfiehlt es sich, Autosave zu deaktivieren, da dann die Zeit zum Speichern der Daten entfällt.

6 Konfiguration des Knotens

6.1 Betriebsart (Art der Nachrichtenübertragung)

Je nachdem, welche Bedingungen für den Einsatz des Wegaufnehmers gelten, können die Daten des BTL mittels Cyclic I/O kontinuierlich, Change of State bei Änderung oder auf Abruf übertragen werden.

Bei den folgenden Beispielen wird immer von einer Master MAC ID 0x00, sowie einer Slave MAC ID 0x63 ausgegangen.

6.1.1 Bit Strobe

Der Master sendet mit einem Bit Strobe Befehl ein allgemeines Synchronisationssignal zu allen Slaves. Die Bit Strobe Command Nachricht enthält einen String von 64 Bits (8 Byte). Das übertragene Datenbit hat keine Bedeutung für das BTL, der Bit Strobe Befehl selbst veranlasst aber die Ausgabe der Messdaten und eine Triggerung der Messung.

Das BTL kann seine Messung so starten, dass rechtzeitig beim nächsten erwarteten Bit Strobe Befehl aktuelle Daten vorliegen. Dazu muss jedoch der Bit Strobe Befehl mit einer möglichst regelmäßigen Taktrate kommen und das Attribut Sync Period auf die Zeit zwischen zwei Bit Strobe Befehlen gesetzt werden (➔ Kapitel 7.2.3 "Attribut 16: Sync Period").

6.1.2 Polled

Beim Polled Mode handelt es sich um eine klassische Master-Slave-Kommunikation. Der Master kann mittels der Poll Command Message den aktuellen Positions-Istwert des Absolutwertgebers abfragen. Das BTL sendet anschließend den Prozess-Istwert in Form der definierten I/O-Nachricht mittels einer Poll-Response-Message an den Master.

6.1.3 Cyclic

Das BTL sendet den aktuellen Prozesswert in Abhängigkeit eines programmierbaren Timers. Hierdurch kann eine Reduzierung der Buslast bewirkt werden, da sich der Teilnehmer im Netz nur nach einem bestimmten Zeitintervall ohne Aufforderung durch den Master meldet.

6.1.4 Change of State (COS)

Das BTL sendet den Prozesswert, wenn sich dieser um mehr als den in Attribut 20 (COS/Delta) des Position Sensor Objects eingestellten Betrag ändert. Wird Attribut 20 auf Null gesetzt, dann wird jede Änderung gesendet.

Diese Betriebsart kann mit Cyclic kombiniert werden, wenn man zusätzlich den Timer setzt. Dann sendet das BTL seine Daten zyklisch und zusätzlich bei Änderungen sofort.

6.2 Beispiel für einen Kommunikationsauf- und abbau

Der Datenaustausch mit einem DeviceNet-Gerät darf nur über eine bestehende Verbindung stattfinden. Daher muss zuerst einmal dieser Kanal geöffnet werden. Hierzu sendet man ein Telegramm mit 6 Datenbytes.

6 Konfiguration des Knotens (Fortsetzung)**Allocate Cyclic**

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4B]						
2	Klassen ID [0x03 DeviceNet]							
3	Instanz ID [01 Explizite Nachricht]							
4	Allocation Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explizit
5	[00]		Allocator MAC ID					

Byte 2 kennzeichnet den Service (Befehl)

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4B	0x03	0x01	0x61	0x00		

Expected_Packet_Rate der explizit Message auf 0 setzen:

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FC	0x00	0x10	0x05	0x01	0x09	0x00	0x00	

Expected_Packet_Rate der Cyclic Verbindung setzen:

Das BTL soll alle 100 ms einen Wert senden (Byte 5 und 6).

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FC	0x00	0x10	0x05	0x04	0x09	0x64	0x00	

Release Cyclic

Beendet die Cyclic-Übertragung, wenn sie nicht mehr benötigt wird. Das BTL stellt die zyklische Sendung ein.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4C]						
2	Klassen ID [0x03 DeviceNet]							
3	Instanz ID [01 Explizite Nachricht]							
4	Release Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explizit

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4C	0x03	0x01	0x60			

6 Konfiguration des Knotens (Fortsetzung)

Release explicit Messaging

Beendet die explizite Übertragung.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4C]						
2	Klassen ID [0x03 DeviceNet]							
3	Instanz ID [01 Explizite Nachricht]							
4	Release Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explizit

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4C	0x03	0x01	0x01			

6.2.1 COS - Change of State Mode

Das BTL sendet ohne Aufforderung durch den Host, wenn sich der aktuelle Prozess-Istwert geändert hat, mindestens aber nach der eingestellten Expected-Packet-Rate, womit eine Reduzierung der Buslast sichergestellt ist. Es sollte das Attribut 20 (COS/Delta (der jeweiligen Instanz im Positions Sensor Objekt eingestellt werden.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4B]						
2	Klassen ID [0x03 DeviceNet]							
3	Instanz ID [01 Explizite Nachricht]							
4	Allocation Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explizit
5	[00]		Allocator MAC ID					

Allocate COS

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4B	0x03	0x01	0x51	0x00		

Expected_Packet_Rate der explizit Message auf 0 setzen:

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FC	0x00	0x10	0x05	0x01	0x09	0x00	0x00	

Expected_Packet_Rate der COS setzen:

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FC	0x00	0x10	0x05	0x04	0x09	0xE8	0x03	

Wenn man nur eine Nachricht erhalten möchte, wenn sich der Wert um das angegebene Delta ändert, muss man diesen auf 0 setzen. Wenn man auch bei keiner Änderung aber alle 1000 ms eine Positionsnachricht erhalten möchte, muss man diesen Wert auf 1000 (0x3E8) setzen (Byte 5 und 6).

6 Konfiguration des Knotens (Fortsetzung)

Release COS

Beendet eine nicht mehr nötige COS Verbindung. Das BTL stellt die Sendung der Daten ein.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4C]						
2	Klassen ID [0x03 DeviceNet]							
3	Instanz ID [01 Explizite Nachricht]							
4	Release Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explizit

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4C	0x03	0x01	0x50			

Release explicit Messaging

Beendet die explizite Verbindung.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4C]						
2	Klassen ID [0x03 DeviceNet]							
3	Instanz ID [01 Explizite Nachricht]							
4	Release Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explizit

Beispiel Nachricht

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4C	0x03	0x01	0x01			

6.3 Anzahl Positionsgeber

Es können bis zu vier Positionsgeber verwendet werden. Jedem Positionsgeber ist eine Instanz des Position Sensor Objekt zugeordnet. Im Auslieferungszustand ist das BTL fest auf einen Positionsgeber eingestellt.

Mit Attribut 3 im Balluff Configuration Objekt kann die Anzahl der Positionsgeber eingestellt werden. Werden weniger Positionsgeber erkannt, als hier konfiguriert werden, dann werden alle Positionswerte auf den Fehlerwert (0x7FFFFFFF) gesetzt und die zugehörigen Alarmer aktiviert.

Im Flexible Magnet Mode kann die Anzahl der Positionsgeber variiert werden. Hierzu wird Attribut 3 des Balluff Configuration Objekts auf 0 gesetzt. Die Instanzen sind dabei

den Positionsgebern in der Reihenfolge zugeordnet, in der sie - vom Stecker aus gesehen - erkannt werden.

6.3.1 Beschreibung FMM-Mode

Hat man zum Beispiel vier Positionsgeber (vom Stecker aus A, B, C

und D), dann wird Positionsgeber A der Instanz 1, B der Instanz 2, usw. zugeordnet.

Entfernt man jetzt Positionsgeber C, dann wird zunächst einmal für alle Instanzen ein Fehler erkannt und auch gemeldet. Die Zeitdauer, für die der Fehler angezeigt wird, wird

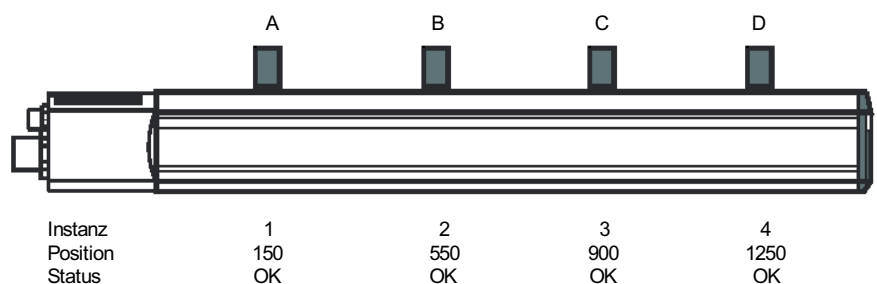


Bild 6-1: BTL mit 4 Positionsgeber

6 Konfiguration des Knotens (Fortsetzung)

mit Attribut 15 (Flexible Magnet Mode Delay) in der Klasse 100 (Balluff Configuration Objekt) in Millisekunden eingestellt. Während dieser Zeit wird als Positionswert der Fehlerwert ausgegeben. Die zugehörigen Fehlerflags sind gesetzt.

Nach Ablauf der eingestellten Zeit werden die Positionsgeber den Instanzen wieder in der Reihenfolge zugeordnet, in der sie erkannt werden. In unserem Beispiel werden nur noch drei Positionsgeber erkannt. Diese werden den Instanzen 1 ... 3 zugeordnet. Der fehlende 4. Positionsgeber wird der Instanz 4 zugeordnet, die weiterhin auf dem Fehlerwert stehen bleibt.

Wenn man in den verschiedenen Instanzen mit verschiedenen Einstellungen (z.B. Position Format, Offsets oder CAMs) arbeitet, dann muss man die mögliche Verschiebung der Magneten im Flexible Magnet Mode berücksichtigen. Dies ist besonders im Assembly zu beachten.

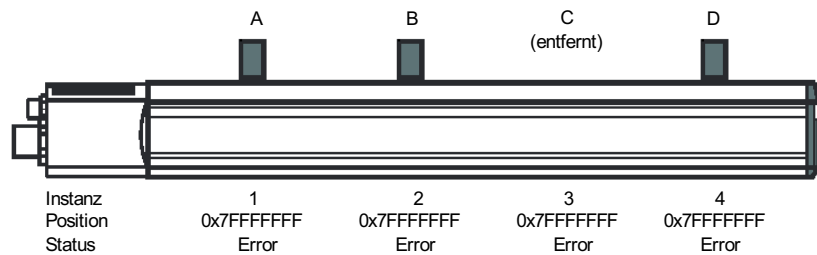


Bild 6-2: 1 Positionsgeber entfernt

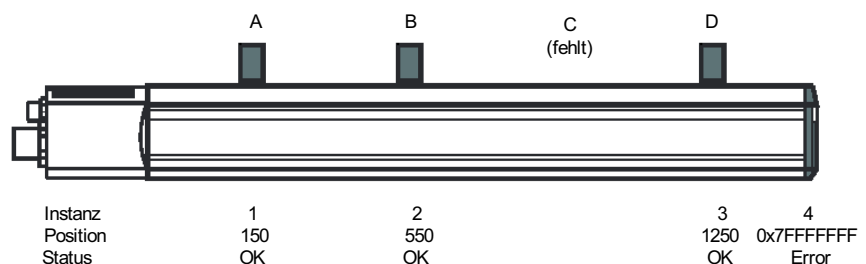


Bild 6-3: neue Zuordnung Positionsgeber

6.4 Positionsdaten

Die Positionsdaten werden von den folgenden Attributen im Position Sensor Objekt beeinflusst.

6.4.1 Attribut 10: Position Value Signed

Der Positionswert wird als 32-Bit Wert ausgegeben. Er wird beeinflusst durch die Attribute 12, 14, 15, 18, 19 und 51. Die interne Auflösung des Positionssensors (Counts) liegt bei 5 µm, obwohl als Ausgabeformat auch nm eingestellt werden kann. Beachten Sie bitte, dass der Wertebereich bei 4 Byte in nm nur von -2,14 m bis 2,14 m reicht.

Dieses Attribut ist auch in die I/O-Nachricht übertragbar (→ Kapitel 2.1 Konfiguration der I/O-Nachrichten).

6.4.2 Attribut 11: Position Sensor Type

Dieses Attribut liefert den Wert 8 für 'Absolute Linear Encoder' zurück.

6.4.3 Attribut 12: Direction Counting Toggle

Im Auslieferungszustand befindet sich der Nullpunkt der Position an der Steckerseite. Zum Ende des BTL werden die Positionen aufsteigend gezählt. Wird Attribut 12 auf TRUE gesetzt, dann wird die Zählrichtung umgekehrt. Der Positionswert, der zuvor dem Anfang des Messbereichs zugeordnet war, ist nach Setzen dieses Attributs um die Nennlänge verschoben am Ende des BTL.

6.4.4 Attribut 14: Scaling Function Control

Dieses Attribut ist im Auslieferungszustand auf TRUE (On) gesetzt. Dies bedeutet, dass eine Skalierung der Positionswerte entsprechend der Attribute 15, 16 und 51 erfolgt.

Wird dieses Attribut auf FALSE (Off) gesetzt, arbeiten alle Funktionen des Position Sensor Objektes weiter mit ihren skalierten Werten in den gesetzten Parametern. Lediglich der Positionswert wird im

Rohformat (Counts) ausgegeben. Diese Funktion dient der Fehlersuche und der Kalibrierung.

6.4.5 Attribut 15: Position Format

Die Attribute 10 (Position Value Signed), 19 (Preset Value), 22 (Position Low Limit), 23 (Position High Limit), 38 (CAM Low Limit), 39 (CAM High Limit), 40 (CAM Hysteresis) und 51 (Offset Value) werden von diesem Attribut beeinflusst. Die Ein- und Ausgabe dieser Attribute ist immer in dem aktuell gewählten Position Format.

Die oben genannten Attribute werden mit einer Genauigkeit von 5 µm ermittelt und bei der Ein- bzw. Ausgabe in das gewählte Positionsformat umgerechnet. Daher werden die eingegebenen Positionen bei einer Änderung des Positionsformats nicht beeinflusst, auch nicht die Genauigkeit, mit der sie gesetzt werden.

Nach einer Veränderung des Attributs Position Format sollte Attribut 18 (Position Measuring Increment) geprüft werden.

6 Konfiguration des Knotens (Fortsetzung)

6.4.5.1 Tabelle gültiger Formate für Positionswerte

Einheit	Sym- bol	Engunit (Hex)	Basisein- heit
Zentimeter	cm	0x2202	10 ⁻² m
Millimeter	mm	0x2203	10 ⁻³ m
Mikrometer (Micron)*	μm	0x2204	10 ⁻⁶ m
Nanometer**	nm	0x2205	10 ⁻⁹ m
Inch	in	0x2207	2,54*10 ⁻² m
Foot	ft	0x2208	3,048*10 ⁻¹ m
Point (Computer)		0x220F	1/72 in
Point (Drucker)		0x2210	3,54598 * 10 ⁻⁴ m
Counts (5μm)		0x1001	5*10 ⁻⁶ m

* Bitte beachten Sie, dass die Einheiten Mikrometer und Nanometer kleiner als die minimale Messauflösung des BTL sind. Daher ist hier die kleinste Schrittweite 5 μm, bzw. 5000 nm.

** Da der Positionswert nur als 32 Bit Wert dargestellt wird, reicht der Wertebereich in nm nur bis 2,14 m. Daher ist diese Einstellung bei längeren BTL nicht sinnvoll und kann zu einem Überlauf der Positionsdaten führen.

6.4.6 Attribut 18: Position Measuring Increment

Setzt die kleinste Schrittweite, in der Positionsänderungen erfasst werden. Dieser Parameter hängt von Attribut 15 (Position Format) ab. Daher sollte Attribut 15 zuerst gesetzt werden.

6.4.7 Attribut 19: Preset Value

Über dieses Attribut kann ein gewünschter Positionswert dem aktuellen Positionswert zugewiesen werden. Wird das Attribut 19 gesetzt, passiert das Folgende:

1. Attribut 19 wird auf den eingegebenen Wert gesetzt,
2. Offset Value (Attribut 51) wird auf die Differenz zwischen Preset Value und aktuellem Positionswert gesetzt.
3. Während des weiteren Betriebs wird der ausgegebene Positionswert (Attribut 10) wie folgt berechnet:

Ausgegebener Positionswert = (interner Positionswert) + Offset Value (Attribut 51)

Beachten Sie, dass der eingegebene Wert sich auf die Einstellung von Attribut 15 (Position Format) bezieht. Daher ist Attribut 15 zuerst zu setzen.

Der Wert, der sich in Preset Value (Attribut 19) befindet, ist, nachdem er gesetzt wurde, für den weiteren Betrieb bedeutungslos. Das Attribut Offset Value (Attribut 51) bleibt konstant und bestimmt ab jetzt den ausgegebenen Positionswert.

Attribut 51 kann im Auslieferungszustand nicht direkt gesetzt werden, sondern nur über den Set Attribute Zugriff auf Attribut 19 (siehe hierzu auch ➔ Kapitel 7.2.2 Attribut 14: Direct Offset Enable)

Das interne Speicherformat erlaubt nur einen maximalen Wertebereich von -21 Meter bis + 21 Meter. Beachten Sie, dass Sie Preset Value nicht so setzen, dass diese Limits überschritten werden.

6.4.8 Attribut 20: COS Delta

Wenn der Positionswert (Attribut 10) sich um den hier gesetzten Wert verändert (oder mehr), dann wird eine COS I/O-Nachricht generiert. Wird dieses Attribut auf 0 gesetzt, dann ist diese Schwelle deaktiviert und jede Positionsänderung wird eine COS-Nachricht generieren.

Attribut 20 wird als reiner Zahlenwert gespeichert. Ist Attribut 15 (Position Format) auf Millimeter gesetzt und man setzt Attribut COS Delta auf 30, dann wird bei einer Positionsänderung von 30 Millimetern eine COS-Nachricht generiert. Stellt man Attribut 15 nun auf Zentimeter um, dann wird eine COS-Nachricht erst nach einer Änderung des Positionswertes von 30 cm gesendet.

Dieses Attribut hat nur Auswirkung auf die Auswertung des zugehörigen Positionswertes. Veränderungen anderer Werte im Input Assembly führen weiterhin zu der sofortigen Generierung einer COS-Nachricht.

6.4.9 Attribut 21 bis 23: Position State Register und Position Limits

Der benutzerdefinierte Arbeitsbereich wird durch die Attribute 22 (Position Low Limit - Unterer Grenzwert) und 23 (Position High Limit - oberer Grenzwert) definiert. Die Einstellung beider Attribute hängt von der momentan verwendeten Einheit ab (Attribut 15: Position Format). Daher sollte Attribut 15 vor diesen Attributen gesetzt werden.

Ist die aktuelle Position größer oder gleich Attribut 22 und kleiner oder gleich Attribut 23, dann sind alle Bits im Position State Register (Attribut 21) gelöscht.

Wenn die aktuelle Position kleiner als Attribut 22 ist, dann wird Bit 0 (Out of Range) und Bit 2 (Range Underflow) gesetzt. Ist die aktuelle Position größer als Attribut 23, dann werden Bit 0 (Out of Range) und Bit 1 (Range Overflow) gesetzt.

Attribut 21: Position State Register								
Bit	7	6	5	4	3	2	1	0
Position	-	-	-	-	-	Range Under-flow	Range Over-flow	Out of Range

Tabelle 6-1: Attribut 21

6 Konfiguration des Knotens (Fortsetzung)

6.4.10 Arbeitsbereiche

Jedem Positionsgeber ist eine Instanz des Position Sensor Object zugeordnet, die unabhängig von den anderen Instanzen parametrierbar sein kann.

6.4.11 Nocken/Schaltpunkte

Jede Instanz stellt vier voneinander unabhängige Nockenschalter (CAMs) zur Verfügung. Die Nockenschalter werden durch die Attribute

- Untere Schaltschwelle Attribut 38: CAM Low Limit
- Obere Schaltschwelle Attribut 39: CAM High Limit
- Hysterese Attribut 40: CAM Hysteresis

in ihrer Position bestimmt. Voreingestellt stehen alle diese Attribute auf 0. Ein Nockenschalter wird aktiv, wenn der Positionswert größer als der eingestellte untere Grenzwert CAM Low Limit und gleichzeitig kleiner als der eingestellte obere Grenzwert CAM High Limit ist.

Man kann die Nockenschalter auch als Schwellenschalter verwenden, wenn man einen der Grenzwerte außerhalb des Messbereichs positioniert.

Die folgenden Attribute des Position Sensor Objects beeinflussen das Verhalten der Nocken:

6.4.12 Attribut 34: Number of CAM Channels

Das BTL hat pro Instanz (Positionsgeber) vier frei programmierbare Nockenschalter. Daher liefert dieses Attribut den Wert 4 zurück.

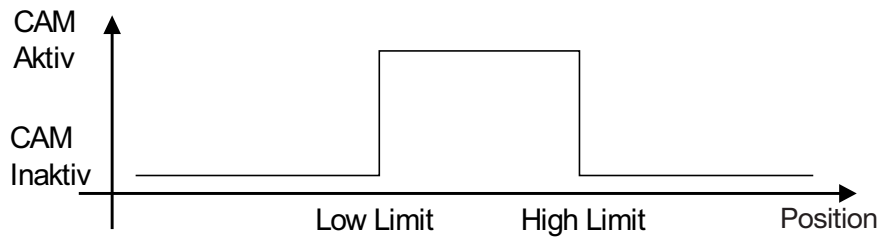


Bild 6-4: Nockenschalter aktivieren

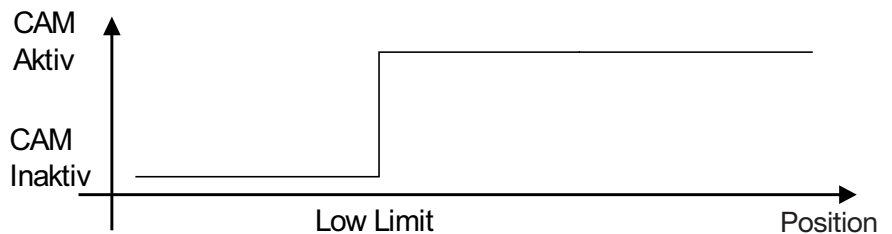


Bild 6-5: Nockenschalter - untere Schaltschwelle

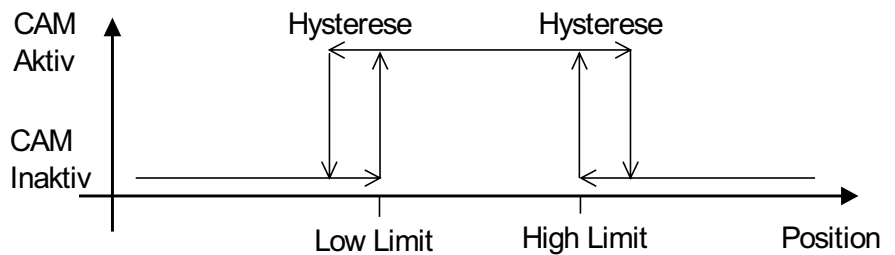


Bild 6-6: Nockenschalter - Hysterese

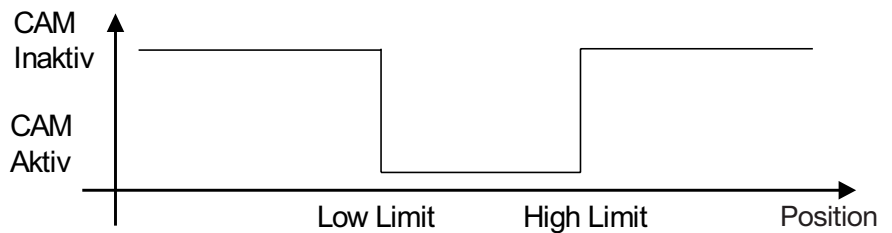


Bild 6-7: Nockenschalter umschalten

6 Konfiguration des Knotens (Fortsetzung)

6.4.13 Attribut 35: CAM Channel State Register

Jedem der vier Nockenschalter ist ein Bit in diesem Register zugeordnet. Dieses Attribut ist auch in die I/O-Nachricht übertragbar (➔ Kapitel 2.1 Konfiguration der I/O-Nachrichten).

CAM Channel State Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Nicht verwendet	Nicht verwendet	Nicht verwendet	Nicht verwendet	CAM4	CAM3	CAM2	CAM1
0	0	0	0	Zustand	Zustand	Zustand	Zustand

Tabelle 6-2: Attribut 35

6.4.14 Attribut 36: CAM Channel Polarity Register

Jedem der vier Nockenschalter ist ein Bit in diesem Register zugeordnet. Ist dieses Bit gelöscht (FALSE), dann wird bei Aktivierung des Nockens das zugehörige Bit im Attribut 35 CAM Channel State Register auf TRUE (1) gesetzt.

Ist ein Bit im CAM Channel Polarity Register gesetzt (TRUE), dann wird das zugehörige Bit im Attribut 35 CAM Channel State Register bei Aktivierung des zugehörigen Nockens gelöscht, nimmt also den Zustand FALSE (0) ein.

CAM Channel Polarity Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Nicht verwendet	Nicht verwendet	Nicht verwendet	Nicht verwendet	CAM4	CAM3	CAM2	CAM1
0	0	0	0	Polarität invers	Polarität invers	Polarität invers	Polarität invers

Tabelle 6-3: Attribut 36

6.4.15 Attribut 37: CAM Channel Enable Register

Analog zu den Attributen 35 und 36 ist hier jedes Bit einem Nockenschalter zugeordnet. Ist dieses Bit gelöscht (FALSE), dann ist der zugehörige Nockenschalter deaktiviert. Das zugehörige Bit im CAM Channel State Register bleibt gelöscht (FALSE).

Wenn das Bit gesetzt ist, dann ist der zugehörige Nockenschalter aktiv und verhält sich so, wie in den Registern 35 und 36 spezifiziert.

CAM Channel Enable Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Nicht verwendet	Nicht verwendet	Nicht verwendet	Nicht verwendet	CAM4	CAM3	CAM2	CAM1
0	0	0	0	Aktiv	Aktiv	Aktiv	Aktiv

Tabelle 6-4: Attribut 37

6.4.16 Attribut 38: CAM Low Limit

Jede Instanz des Position Sensor Objects (die jeweils einem Positionsgeber zugeordnet ist) hat vier mögliche Nockenschalter (CAMs). Jedem Nockenschalter ist jeweils ein unterer Grenzwert (Low Limit) und ein oberer Grenzwert (High Limit) zugeordnet.

Attribut 38 speichert die unteren Grenzwerte für alle vier Nockenschalter einer Instanz. Die vier Nockenschalter werden als Array übergeben, d.h. vier Positionen mit je vier Byte. Zusammen hat dieses Attribut also 16 Byte.

CAM Low/High Limit																
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Channel	LSB			MSB	LSB			MSB	LSB			MSB	LSB			MSB
	CAM1				CAM2				CAM3				CAM4			

Tabelle 6-5: Attribut 38/39

6.4.17 Attribut 39: CAM High Limit

Dieses Attribut speichert die oberen Grenzwerte der vier Nockenschalter einer Instanz. Details und Aufbau siehe Attribut 38: CAM Low Limit.

6 Konfiguration des Knotens (Fortsetzung)

6.4.18 Attribut 40: CAM Hysteresis

Die Hysterese wird zum CAM High Limit addiert und vom CAM Low Limit subtrahiert, wenn der Zustand der Nockenschalter berechnet wird. Die Grafik ➔ Bild 6-6 veranschaulicht die Details.

Dieses Attribut ist ein Array von UINT. Für jeden CAM-Kanal enthält es zwei Byte. Damit enthält dieses Attribut 8 Byte.

Attribut 40: CAM Hysteresis								
Byte	0	1	2	3	4	5	6	7
	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB
Channel	CAM1		CAM2		CAM3		CAM4	

Tabelle 6-6: Attribut 40

6.5 Geschwindigkeitseinstellungen

Der Geschwindigkeit kann in verschiedenen Einheiten, die im Attribut 25 (Velocity Format) gewählt werden, ausgegeben werden. Die Einstellungen diverser anderer Attribute beziehen sich auf die gewählte Einheit.

Daher wird empfohlen, zuerst die gewünschte Einheit einzustellen und anschließend die davon abhängigen Attribute zu setzen. Werden die Attribute automatisch, z.B. mittels RSNetWorx gesetzt, dann sollte anschließend ein Upload durchgeführt und die Einstellungen überprüft werden.

6.5.1 Auflösung

Die Geschwindigkeit wird mit einer Auflösung von 0,1 mm/s erfasst und mit einer Genauigkeit von 32 Bit verarbeitet. Die nach CIP voreingestellte Einheit für die Geschwindigkeit ist die Standardeinheit für die Position (Counts = 5 µm) pro Sekunde, also 5 µm/s.

Diese Einheit liegt jedoch unter der Messauflösung der Geschwindigkeit und hat daher eine minimale Schrittweite von 20.

Die Auflösung, in der die Geschwindigkeitsdaten ausgegeben werden, kann über die folgenden Attribute des Position Sensor Objects eingestellt werden.

6.5.1.2 Attribut 24: Velocity Value

Aktueller Wert der Geschwindigkeit abhängig von den Attributen 25 und 26.

Dieses Attribut ist auch in die I/O-Nachricht übertragbar (➔ Kapitel 2.1 Konfiguration der I/O-Nachrichten).

6.5.1.3 Attribut 25: Velocity Format

In diesem Attribut befindet sich die Einheit der Geschwindigkeit. Diese Einheit beeinflusst die Attribute 26 (Velocity Resolution), 27 (Minimum Velocity Setpoint) und 28 (Maximum Velocity Setpoint). Deshalb sollte dieses Attribut vor den Attributen 26, 27 und 28 gesetzt werden.

Einheit	Sym- bol	Engunit (Hex)	Basisein- heit
Schritte pro Sekunde*	CPS	0x1F04	5 µm/s
Meter pro Sekunde	m/s	0x2B00	1 m/s
Zentimeter pro Sekunde	cm/s	0x2B01	10 ⁻² m/s
Kilometer pro Stunde	km/h	0x2B02	(1000 / 3600) m/s
Meilen pro Stunde	mi/h	0x2B04	0,44704 m/s
Knoten	kt	0x2B05	(1852 / 3600) m/s
Fuß pro Sekunde	ft/s	0x2B06	3,048*10 ⁻¹ m/s
Inch pro Sekunde	in/s	0x2B07	2,54*10 ⁻² m/s

* Bitte beachten Sie, dass die interne Messauflösung für die Geschwindigkeit 100 µm/s ist. Damit liegt die Einheit Schritte pro Sekunde unterhalb dieser Auflösung und die kleinste Schrittweite ist 100 µm/s.

6.5.1.4 Attribut 26: Velocity Resolution

Gibt die kleinste Schrittweite im gewählten Geschwindigkeitsformat an, in der der Geschwindigkeitswert ausgegeben wird.

6.5.1.5 Attribut 27: Minimum Velocity Setpoint

In diesem Attribut wird ein unterer Grenzwert für die Geschwindigkeit gesetzt. Ist die Geschwindigkeit gleich oder kleiner als dieser Wert, dann wird das Minimum Velocity Flag in Attribut 47 (Warnings) gesetzt.

6.5.1.6 Attribut 28: Maximum Velocity Setpoint

Analog zu Attribut 27 wird hier der obere Grenzwert für die Geschwindigkeit gesetzt. Ist die Geschwindigkeit gleich oder größer als dieser Wert, dann wird das Maximum Velocity Flag in Attribut 47 (Warnings) gesetzt.

6.5.2 Schaltpunkte

Die Attribute 27 (Minimum Velocity Setpoint) und 28 (Maximum Velocity Setpoint) sind zwei Bits im Attribut 47 (Warnings) zugeordnet. Erreicht oder unterschreitet die Geschwindigkeit den Minimum Velocity Setpoint, dann wird Bit 6 (Minimum Velocity Flag) im Attribut Warnings gesetzt. Erreicht oder überschreitet die Geschwindigkeit den Maximum Velocity Setpoint, dann wird das Maximum Velocity Flag (Bit 7) im Attribut Warnings gesetzt. Die Bits werden gelöscht, wenn die Geschwindigkeit wieder zwischen Attribut 27 und 28 liegt.

7 Weitere Attribute

7.1 Position Sensor Objekt

7.1.1 Attribut 41: Operating Status

Dieses Attribut enthält Informationen über verschiedene Zustände des Encoders.

Bit 0 zeigt die Zählrichtung an. Ist dieses Bit 0 (FALSE), dann ist die Zählrichtung positiv, d.h. die Positionswerte steigen in Richtung vom Stecker zum Ende des Messbereichs. Ist dieses Bit gesetzt (TRUE), dann ist die Zählrichtung umgekehrt.

Bit 1 zeigt an, ob die Skalierung aktiv ist. Ist dieses Bit gelöscht, dann ist die Skalierung deaktiviert. Das BTL gibt in dem Fall Rohwerte aus. Ist dieses Bit gesetzt, dann ist die Skalierung aktiviert.

7.1.2 Attribut 44: Alarms

Das Attribut Alarms zeigt an, ob eine Fehlfunktion auftritt, die die Bestimmung des Positionswertes stört. Der Alarm bleibt aktiv, bis der Fehler beseitigt ist und das Gerät wieder einen gültigen Positionswert liefert.

Das Flag Position Error wird gesetzt, wenn eine Positionsbestimmung nicht möglich ist. Dies ist beispielsweise der Fall, wenn sich der zugehörige Positionsgeber nicht mehr im Messbereich befindet. Im Flexible Magnet Mode werden alle Positionen für eine definierte Zeit ungültig, wenn sich die Anzahl der Positionsgeber ändert.

Das Flag Temperature Error ist nur aktiv, wenn ein Temperatursensor bestückt ist (nur bei A/B/Z-Stab). Es zeigt an, dass das BTL außerhalb seines zulässigen Temperaturbereiches betrieben wird.

Dieses Attribut ist auch in die I/O-Nachricht übertragbar (➔ Kapitel 2.1 Konfiguration der I/O-Nachrichten).

7.1.3 Attribut 45: Supported Alarms

Für jedes in Attribut 44 unterstützte Bit ist in diesem Attribut ein Bit gesetzt. Der Wert ist also immer konstant 0x1001.

Attribut 41: Operating Status																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Scaling	Direction

Tabelle 7-1: Attribut 41

Attribut 44: Alarms																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alarms				Tem- pera- ture												Position Error

Tabelle 7-2: Attribut 44

Attribut 45: Supported Alarms																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alarms				Tem- pera- ture												Posi- tion
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Tabelle 7-3: Attribut 45

Attribut 47: Warnings																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Warn-ings						Position Limits			Maximum Velocity	Minimum Velocity						

Tabelle 7-4: Attribut 47

7.1.4 Attribut 46: Alarm Flag

Wenn eines der Flags in Attribut 44 gesetzt wird, dann wird Attribut 46 auf 1 (TRUE) gesetzt. Sind alle Flags in Attribut 44 gelöscht, dann hat Attribut 46 den Wert 0 (FALSE).

7.1.5 Attribut 47: Warnings

Das Attribut Warnings zeigt an, dass die Toleranz verschiedener interner Parameter überschritten wurde. Im Gegensatz zum Attribut 44 Alarms werden hier ungültige Positionswerte nicht berücksichtigt. Die Warnungen werden gelöscht, wenn die internen Parameter wieder in ihren Grenzen liegen.

Das Flag Minimum Velocity wird gesetzt, wenn die aktuelle Geschwindigkeit kleiner als der in Attribut 27 Minimum Velocity Setpoint gesetzte Wert ist. Analog dazu zeigt das Maximum Velocity Flag an, dass aktuelle Geschwindigkeit größer als der in Attribut 28 gesetzte Wert ist.

Das Flag Position Limits wird gesetzt, wenn die aktuelle Position kleiner als der in Attribut 22 Position Low Limit oder größer als der in Attribut 23 Position High Limit spezifizierte Wert ist.

Dieses Attribut ist auch in die I/O-Nachricht übertragbar (➔ Kapitel 2.1 Konfiguration der I/O-Nachrichten).

7.1.6 Attribut 48: Supported Warnings

Für jedes in Attribut 47 unterstützte Bit ist in diesem Attribut ein Bit gesetzt. Der Wert ist also immer konstant 0x4C0.

7.1.7 Attribut 49: Warning Flag

Wenn eines der Flags in Attribut 47 gesetzt wird, dann wird Attribut 49 auf 1 (TRUE) gesetzt. Sind alle Flags in Attribut 47 gelöscht, dann hat Attribut 49 den Wert 0 (FALSE).

7 Weitere Attribute (Fortsetzung)**7.1.8 Reset Service - Position Sensor Objekt**

Um das Position Sensor Objekt wieder zurück auf seine Vorgabewerte zu setzen, ist ein Reset-Service implementiert. Hierzu ist der Service mit dem Argument 1 aufzurufen. Details sind im ➡ Kapitel 4.4 Schritt 2c: Aufruf eines Services beschrieben.

Es wird nur die jeweils angegebene Instanz zurückgesetzt. Die restlichen Instanzen bleiben unverändert. Möchte man alle Instanzen zurücksetzen, dann muss der Reset-Service für jede Instanz aufgerufen werden.

7.2 Balluff Configuration Objekt**7.2.1 Attribut 11, 12 und 13: Temperatur (nur bei A/B/Z-Stab)**

In diesen Attributen wird die aktuelle Temperatur (Attribut 11), die bisher tiefste Betriebstemperatur (Attribut 13) und die bisher höchste Betriebstemperatur (Attribut 12) gespeichert. Die Temperaturwerte werden alle 6 Minuten aktualisiert, wenn der Temperatursensor bestückt ist.

7.2.2 Attribut 14: Direct Offset Enable

Laut CIP Spezifikation kann Attribut 51 des Position Sensor Objektes (Offset Value) nicht direkt gesetzt werden. Er wird vielmehr berechnet, wenn Attribut 19 (Preset Value) gesetzt wird. Dies setzt voraus, dass der Positionsgeber zum Setzen von Preset Value an einen definierten Punkt gesetzt wird.

Dieses Attribut Direct Offset Enable ist im Auslieferungszustand auf False. Dies bedeutet, dass sich das Position Sensor Objekt wie oben beschrieben verhält. Bei einem Schreibzugriff auf Attribut 51 Offset Value wird eine Fehlermeldung zurückgegeben.

Attribute 48: Supported Warnings																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Warnings						Position Limits			Maximum Velocity	Minimum Velocity						
Default	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0

Tabelle 7-5: Attribut 48

Setzt man Direct Offset Enable auf True, dann ist ein Schreibzugriff auf Attribut 51 möglich, es kann also direkt der Offset eingegeben werden.

7.2.3 Attribut 16: Sync Period

Wird das Gerät mittels Bit-Strobe synchronisiert, dann sendet es seine Daten nach Erhalt jedes Bit-Strobe-Kommandos. Dabei versucht der Sensor seinen Messungsstart so zu synchronisieren, dass rechtzeitig zum erwarteten nächsten Bit-Strobe-Kommando ein neuer Messwert ermittelt wird. Dazu muss dieses Attribut auf den erwarteten Zeitabstand zwischen den einzelnen Bit-Strobe-Kommandos gesetzt werden.

7.2.4 Reset Service - Balluff Configuration Objekt

Um alle Attribute des Balluff Configuration Objekts wieder zurück auf seine Vorgabewerte zu setzen, ist ein Reset-Service implementiert. Hierzu ist der Service mit dem Argument 1 aufzurufen. Details sind im ➡ Kapitel 4.4 Schritt 2c: Aufruf eines Services beschrieben.


8 Inbetriebnahme

8.1 Projektierung unter RsNetworkx

8.1.1 EDS-Datei

Das EDS-File, welches im Internet unter www.balluff.de zu finden ist, enthält Informationen über geräte-spezifische Parameter sowie mögliche Betriebsarten des BTL. Hiermit steht ein elektronisches Datenblatt zur Verfügung, das zur Projektierung unter z.B. RsNetworkx verwendet werden kann.

Bevor das BTL an den Bus angeschlossen werden kann, muss das EDS-File mittels des EDS Wizards installiert werden. Um den EDS Wizard zu starten, klickt man mit dem Cursor in der Menüleiste auf 'Tools/EDS Wizard'. Es erscheint ein Fenster ➔ Bild 8-1.

Um ein EDS-File zu installieren, muss 'Register an EDS file(s)' und danach  angeklickt werden. Im nächsten Schritt muss 'Register a directory of EDS files' angeklickt und mittels Browse der Pfad des EDS-Files angegeben werden ➔ Bild 8-2.

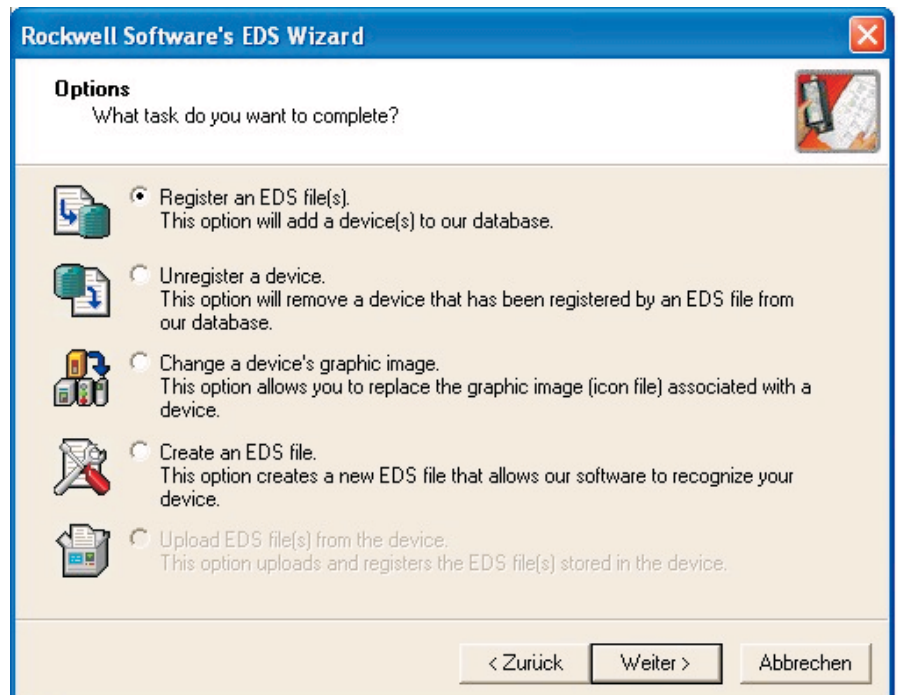


Bild 8-1: Startmenü des EDS - Wizard

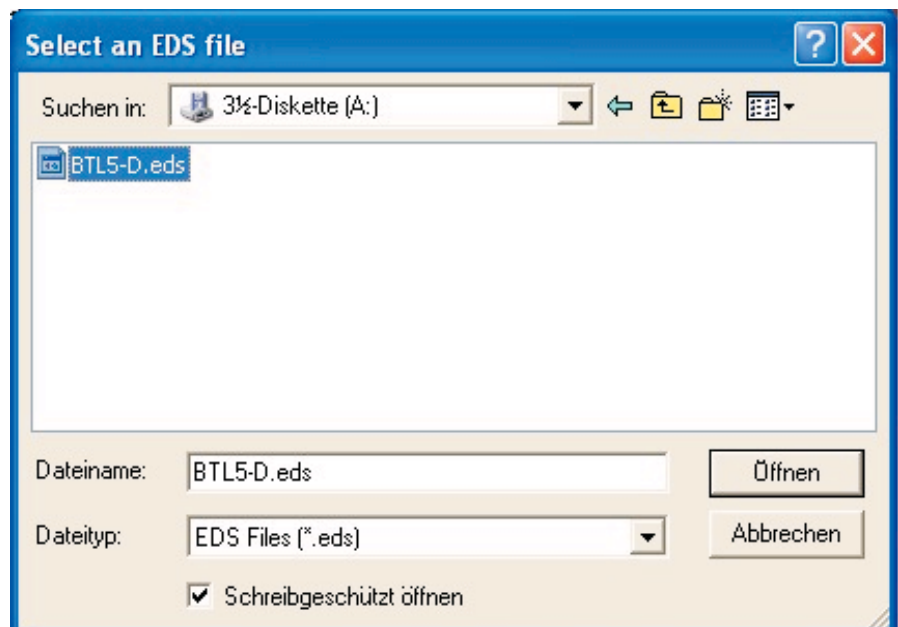


Bild 8-2: Auswahl der EDS- Datei

8 Inbetriebnahme (Fortsetzung)

Der Wizard findet alle EDS-Files, die in diesem Pfad abgelegt sind und führt nach dem Öffnen einen Test durch, ob Fehler im EDS-File enthalten sind ➔ Bild 8-3.

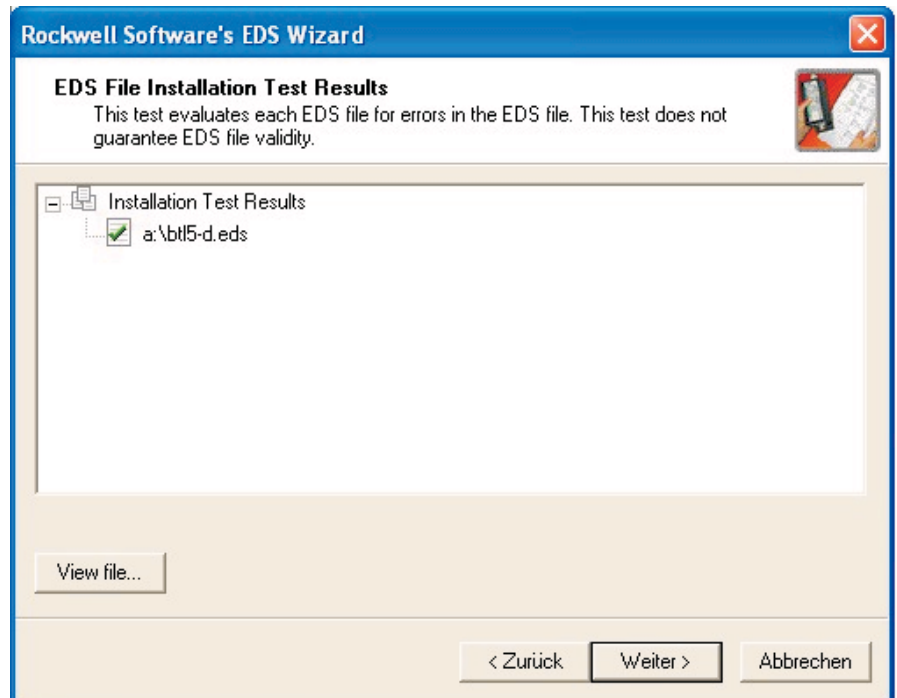




Bild 8-3: Testergebnis der EDS-Datei

Wird nun der Button  betätigt, wird man zum 'Change Graphic Image' Fenster (➔ Bild 8-4: Auswahl der Grafik) weitergeleitet, hier können den verwendeten Knoten Bilder zugewiesen werden. Mit  kann die Installation fortgeführt und fertig gestellt werden.

Anschließend erscheint eine Meldung, die die erfolgreiche Installation der EDS-Datei bestätigt.

Wenn die Steuerung jetzt das Gerät im Netzwerk findet, dann wird es richtig erkannt und dargestellt.

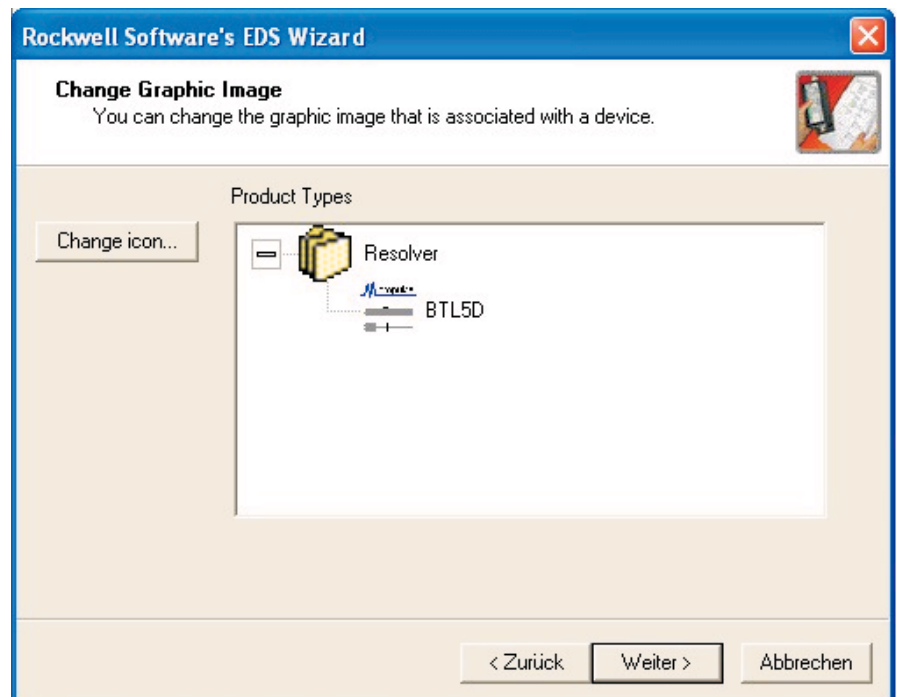


Bild 8-4: Auswahl der Grafik


8 Inbetriebnahme (Fortsetzung)


Um das BTL zu parametrieren, klickt man im Online-Mode das Symbol des BTL an und öffnet das Konfigurations-Fenster über Device/Properties in der Menüleiste oder über das Menü, das nach Betätigen der rechten Maustaste erscheint.

Über **Parameters** erfolgt dann ein Upload der Parameter.

Nach dem erfolgreichen Upload der Parameter können diese konfiguriert werden ➔ Bild 8-7.


Ein Download der konfigurierten Attribute kann mit dem gelben Pfeil, der nach unten zeigt und rechts oben im

Fenster zu finden ist , durchgeführt werden - ein Upload mit dem links neben dem Download Button stehenden Pfeil, der nach oben zeigt

. Geänderte Attribute werden nach Betätigen des Buttons

Übernehmen direkt in das BTL geladen.

Die Attribute werden nach einem Download automatisch im EEPROM gespeichert. Damit wird das BTL nach einem Abschalten der Versorgung wieder mit diesen Daten gestartet. Soll keine permanente Speicherung erfolgen, dann ist das Attribut 4 im Balluff Configuration Object (ID 504) zuvor auf FALSE zu setzen und dieser Wert zu übernehmen. Dann wird das BTL mit den zuletzt gespeicherten Einstellungen gestartet.

Den aktuellen Wert eines Attributes kann man permanent auslesen, indem man den Button  betätigt. Wählt man ‚Single‘ aus, dann wird das jeweils mit der Maus angeklickte Attribut zyklisch abgefragt. Wählt man ‚All‘ aus, dann werden alle angezeigten Attribute der Reihe nach aktualisiert. Ein Pfeil am linken Rand der Liste markiert den jeweils aktualisierten Wert.

An der ID kann man die Instanz und die Attributnummer erkennen. Die IDs 101 ... 151 beziehen sich auf Instanz 1 und sind dem ersten Positionsgeber zugeordnet. Analog sind 201 ... 251 der Instanz 2 für den Positionsgeber 2, 301 ... 351 Instanz 3 für Positionsgeber 3 und 401 ... 451 Instanz 4 für Positionsgeber 4 zugeordnet.

Die IDs 501 ... 517 sind dem Balluff Configuration Object zugeordnet. Dessen Attribute gelten global.

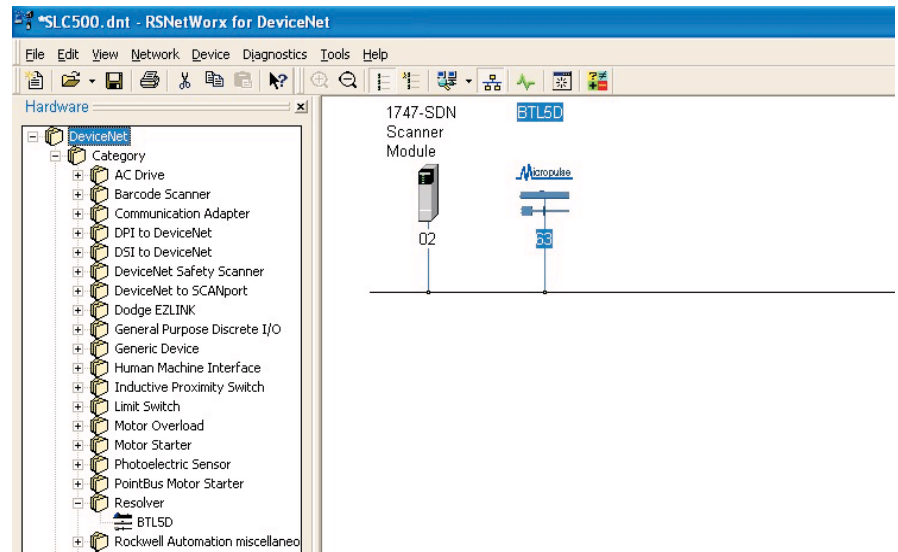


Bild 8-5: Darstellung des BTL im Netzwerk

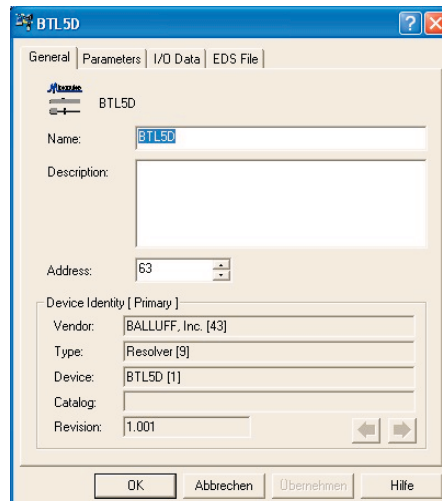


Bild 8-6: Allgemeine Angaben zum BTL

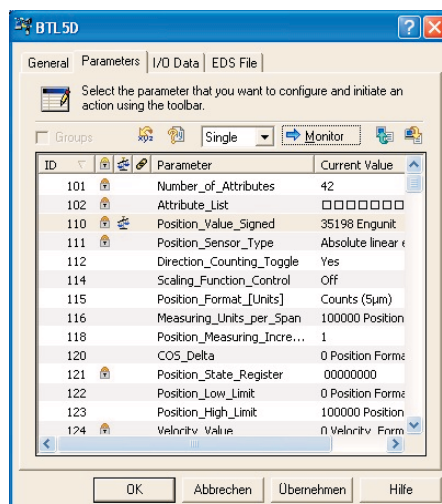


Bild 8-7: Parameter

9 Klassenverzeichnis (Objektverzeichnis)

Das Klassenverzeichnis des Wegaufnehmers ist in sechs Bereiche aufgeteilt, die nachfolgend tabellarisch dargestellt sind.

Die Daten (Position, Geschwindigkeit) ebenso wie die Konfigurationsparameter werden in der Klasse des Position Sensor Objekt abgelegt.

Verwendete Abkürzungen in den folgenden Abschnitten:

R/W Read/Write: Attribut kann gelesen und geschrieben werden

R Read only: Attribut kann nur gelesen werden

ROM Read Only Memory: Attribut ist konstant und kann nicht verändert werden

NV Non Volatile: Attribut wird im nichtflüchtigen Speicher gesichert

V Volatile: Attribut wird nicht gespeichert.

	Bezeichnung	Anzahl Bytes	Minimalwert	Maximalwert
BOOL	Boolean	1	0 (FALSE)	1 (TRUE)
SINT	Short Integer	1	-128	127
INT	Integer	2	-32768	32767
DINT	Double Integer	4	-2 ³¹	2 ³¹ -1
USINT	Unsigned Short Integer	1	0	255
UINT	Unsigned Integer	2	0	65535
UDINT	Unsigned Double Integer	4	0	2 ³² -1
BYTE	bit string - 8 bits	1		
WORD	bit string - 16-bits	2		
DWORD	bit string - 32-bits	4		
ENGUNITS	Engineering Units	2		Siehe nachfolgende Tabelle

Tabelle 9-1: Datentypen

1F04 counts per second	cps	7940	0x1F04
2B00 meter per second	m/s	11008	0x2B00
2B01 centimeter per second	cm/s	11009	0x2B01
2B02 kilometer per hour	km/h	11010	0x2B02
2B04 mile per hour	mi/h	11012	0x2B04
2B05 knot (nautical mile per hour)	kt	11013	0x2B05
2B06 foot per second	ft/s	11014	0x2B06
2B07 inch per second	in/s	11015	0x2B07

Tabelle 9-2: Einheiten (Engineering Units) für die Geschwindigkeit

1001 counts		4097	0x1001
2202 centimeter	cm	8706	0x2202
2203 millimeter	mm	8707	0x2203
2204 micron (micrometer)	μ	8708	0x2204
2207 inch	in	8711	0x2207
2208 foot	ft	8712	0x2208
220F point (computer)	(1/72) in	8719	0x220F
2210 point (printer's)	3.514598 * 10 ⁻⁴ m	8720	0x2210

Tabelle 9-3: Einheiten (Engineering Units) für die Position

9 Klassenverzeichnis (Objektverzeichnis, Fortsetzung)**9.1 Identity Klasse (ID = 1)**

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Default-wert	Beschreibung
1	1	1	Vendor ID	R	ROM	UINT	0x2B	Identifikation des Herstellers 0x2B: BALLUFF
1	1	2	Device Type	R	ROM	UINT	0x09	Identifikation des Produkttypes 0x09: Resolver Device
1	1	3	Product Code	R	ROM	UINT		Identifikation des Produkts 0x01: Profil –P 0x02: Stab –A 0x03: Stab –B 0x04: Stab –Z
1	1	4	Revision	R	ROM			Nummer der Softwareversion
1	1		Major Revision		ROM	USINT		Vorkomma
1	1		Minor Revision		ROM	USINT		Nachkomma
1	1	5	Status	R	NV	WORD	0	Gerätezustand Bit gesetzt Bit 2: konfiguriert; jeder andere Zustand wie "out-of-box" Bit 8: geringer wiederherstellbarer Fehler; Gerät geht nicht in einen der Fehlerzustände Bit 9: geringer nicht wiederherstellbarer Fehler; Gerät geht nicht in einen der Fehlerzustände Bit 10: wiederherstellbarer Fehler; Gerät wechselt in den Zustand "Major Recoverable Fault" Bit 12: nicht wiederherstellbarer Fehler; Gerät wechselt in den Zustand "Major Unrecoverable Fault"
1	1	6	Serial Number	R	ROM	UDINT		Seriennummer
1	1	7	Product Name	R	ROM	Short_String	BTL5D	Produktname
1	1	10	Heartbeat Interval	R/W	V	USINT	0	Zeit zwischen zwei Heartbeat-Nachrichten in Sekunden. 0 schaltet die Heartbeat-Nachricht ab.

Tabelle 9-4: Identity Klasse

9 Klassenverzeichnis (Objektverzeichnis, Fortsetzung)**9.2 DeviceNet Klasse (ID = 3)**

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Default-wert	Beschreibung
3	n	1	MAC ID	R/W	NV	USINT	0x3F	Eindeutige Adresse des Gerätes im Netzwerk
3	n	2	Baud Rate	R/W	NV	USINT	0x00	Baudrate 0x00: 125 kBaud 0x01: 250 kBaud 0x02: 500 kBaud
3	n	3	BOI (Bus-Off Interrupt)	R		BOOL	FALSE	FALSE: Der CAN Chip bleibt im Reset TRUE: Nach dem Reset Kommunikation fortsetzen
3	n	4	Bus-Off Counter	R/W		USINT	0	Zählt, wie oft das Gerät in den Bus-off Zustand wechselt. Startet nach Einschalten mit 0. Mit einem Schreibzugriff wird der Zähler zurückgesetzt.
3	n	5	Allocation Information	R		STRUCT		Enthält Informationen, welche der Predefined Master/Slave Verbindungen aktiv sind und die Master MacID.
			Allocation Choice Byte				0	Bit 7: Reserved Bit 6: Acknowledge Suppression Bit 5: Cyclic Bit 4: Change of State Bit 3: Multicast Polling Bit 2: Bit Strobed Bit 1: Polled Bit 0: Explicit Message
			Master's MAC ID				0	Die MAC ID, die z.B. einen Bit Strobe Request auslösen darf.

Tabelle 9-5: DeviceNet Klasse

9.3 Assembly Klasse (ID = 4)

Im Assembly Objekt sind die I/O-Daten nochmals abrufbar. Anzahl und Inhalt der Daten können wie im Kapitel 2.1 "Konfiguration der I/O-Nachrichten" beschrieben, konfiguriert werden.

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Default-wert	Beschreibung
4	100	3	Data	R		ARRAY von BYTE		Position und Nocken des 1. Positionsgebers

Tabelle 9-6: Assembly Klasse

9 Klassenverzeichnis (Objektverzeichnis, Fortsetzung)

9.4 Connection Objekt

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Default-wert	Beschreibung
5	X	1	State	R		USINT		Status des Objektes 00: Nicht existent 01: Wird konfiguriert 02: Warte auf Verbindungs-ID 03: Verbindung etabliert 04: Zeitüberschreitung
5	X	2	Instance Type	R		USINT		Type der Instanz: 00: Explicit Message Verbindung 01: I/O-Verbindung
5	X	3	Transport Class Trigger	R		BYTE		Bestimmt das Verhalten der Verbindung Bit 7: Richtung 0: Client; 1: Server Bit 6 ... 4: Produktion 0: Cyclic 1: Change-Of-State 2: Application Object Bit 3 ... 0: Transportklasse 0: Klasse 0 1: Reserviert / nicht verwendet 2: Klasse 2 3: Klasse 3
5	X	4	Produced Connection ID	R		UINT		Wird in das CAN Identifier Feld kopiert, wenn über die Verbindung gesendet wird.
5	X	5	Consumed Connection ID	R		UINT		CAN Identifier Wert, der die über eine Verbindung empfangene Nachricht bezeichnet.
5	X	6	Initial Comm Characteristics	R		BYTE		Definiert die Nachrichtengruppe(n) über die die produzierten und konsumierten Daten bezogen auf diese Verbindung auftreten.
5	X	7	Produced Connection Size	R/W		UINT		Maximale Anzahl von Bytes, die über diese Verbindung gesendet werden.
5	X	8	Consumed Connection Size	R		UINT		Maximale Anzahl von Bytes, die über diese Verbindung empfangen werden.
5	X	9	Expected Packet Rate	R/W		UINT		Abstand zwischen zyklischen Sendungen.
5	X	12	Watchdog Timeout Action	R		USINT		Definiert das Verhalten auf Inactivity- oder Watchdog Zeitüberschreitungen.
5	X	13	Produced Conn. Path Length	R		UINT		Anzahl der Bytes im Produced Connection Path Attribut
5	X	14	Produced Connection Path	R		PACKED		Spezifiziert das Applikationsobjekt dessen Daten in dieses Connection Object exportiert werden.
5	X	15	Consumed Conn. Path Length	R		UINT		Anzahl der Bytes im Consumed Connection Path Attribut
5	X	16	Consumed Connection Path	R		PACKED		Spezifiziert das Applikationsobjekt, dessen Daten aus diesem Connection Object importiert werden.
5	X	17	Production Inhibit Time	R/W		UINT		Definiert die minimale Zeit zur Datenproduktion.

Tabelle 9-7: Connection Objekt

X = 1: explizite Nachricht
X = 2: Polled I/O
X = 3: Bit Strobed I/O

X = 4: COS/Cyclic
X = 5: Multicast Poll I/O

9 Klassenverzeichnis (Objektverzeichnis, Fortsetzung)

9.5 Position Sensor Objekt

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Default-wert	Beschreibung
35	1..4	1	Number of Attributes	R	ROM	USINT		Anzahl der in diesem Objekt unterstützen Attribute.
35	1..4	2	Attribute List	R	ROM	STRING		Array of USINT: Liste der in diesem Objekt unterstützen Attribute.
35	1..4	10	Position Value Signed	R	V	DINT		Aktuelle Position unter Berücksichtigung der Attribute 14, 15, 16 und 17.
35	1..4	11	Position Sensor Type	R	ROM	UINT	8	Gerätetyp (Absolute Linear Encoder)
35	1..4	12	Direction Counting Toggle	R/W	NV	BOOL	FALSE	Definiert die positive Zählrichtung des Positionswerts.
35	1..4	14	Scaling Function Control	R/W	NV	BOOL	TRUE	True (default): Daten werden gemäß der anderen Attribute umgewandelt. False: Positionsdaten werden im Rohformat ausgegeben.
35	1..4	15	Position Format	R/W	NV	ENGUNIT		Einheit der Positionswerte
35	1..4	16	Measuring units per span	R/W	NV	UDINT		Anzahl der Schritte über die Nennlänge. Gleich oder kleiner als Attribut 42.
35	1..4	18	Position Measuring Increment	R/W	NV	UDINT		Spezifiziert die kleinste Schrittweite für Attribut 10.
35	1..4	19	Preset Value	R/W	NV	DINT		Der ausgegebene Positionswert wird auf den vorgegebenen Wert gesetzt.
35	1..4	20	COS / delta	R/W	NV	UDINT		Schrittweite, bei deren Überschreitung eine Change-Of-State-Nachricht gesendet wird.
35	1..4	21	Position State Register	R	V	BYTE		Überwachung der Positionsgrenzen (Attribut 22 und 23).
35	1..4	22	Position Low Limit	R/W	NV	DINT		Untere Positionsgrenze. Wenn Attribut 10 kleiner als dieser Wert ist, dann wird Bit 0 in Attribut 21 (Position State Register) gesetzt.
35	1..4	23	Position High Limit	R/W	NV	DINT		Obere Positionsgrenze. Ist Attribut 10 größer als dieser Wert, dann wird Bit 1 im Position State Register (Attribut 21) gesetzt.
35	1..4	24	Velocity Value	R	V	DINT		Aktuelle Geschwindigkeit. Format wird in den Attributen 25 und 26 definiert.
35	1..4	25	Velocity Format	R/W	NV	ENGUNIT	0x1F04	Geschwindigkeitsformat. Default: 0x1F04 = counts (Schritte) pro Sekunde
35	1..4	26	Velocity Resolution	R/W	NV	UDINT		Kleinste Schrittweite von Attribut 24. Die Einheit hängt von Attribut 25 ab.
35	1..4	27	Minimum Velocity Setpoint	R/W	NV	DINT		Wert für die untere Geschwindigkeitsschwelle. Beeinflusst das Bit Min. Velocity Flag in Attribut 47.
35	1..4	28	Maximum Velocity Setpoint	R/W	NV	DINT		Wert für die obere Geschwindigkeitsschwelle. Beeinflusst das Bit Max. Velocity Flag in Attribut 47.

Tabelle 9-8: Position Sensor Objekt

9 Klassenverzeichnis (Objektverzeichnis, Fortsetzung)

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Default-wert	Beschreibung
35	1..4	34	Number of CAM Channels	R	NV	USINT	4	Anzahl der unabhängigen Nockenschalter
35	1..4	35	CAM Channel State Register	R	NV	ARRAY		Array von BOOL: Enthält den Zustand der Nockenschalter (Bit 0 = CAM_1, Bit1 = CAM: _2,...)
35	1..4	36	CAM Channel Polarity Register	R/W	NV	ARRAY		Array von BOOL: Enthält die Polarität der Nockenschalter. Bit 0 = CAM_1 Polarity, Bit1 = CAM_2,...
35	1..4	37	CAN Channel Enable register	R/W	NV	ARRAY		Array von BOOL: Aktiviert die einzelnen Nockenschalter Bit 0 = CAM_1, Bit1 = CAM_2,...
35	1..4	38	CAM Low Limit	R/W	NV	ARRAY		Array von DINT: Unterere Schaltpunkte für die Nockenschalter.
35	1..4	39	CAM High Limit	R/W	NV	ARRAY		Array von DINT: Obere Schaltpunkte für die Nockenschalter.
35	1..4	40	CAM Hysteresis	R/W	NV	ARRAY		Array von UINT: Dieser Wert wird als Hysterese zum Attribut CAM High Limit addiert und vom CAM Low Limit subtrahiert.
35	1..4	41	Operating Status	R	V	BYTE		Status des Wegaufnehmers
35	1..4	42	Physical Resolution Span	R	NV	UDINT		Anzahl der physikalisch möglichen Schritte über die Nennlänge.
35	1..4	44	Alarms	R	V	WORD		Zeigt auftretende Fehler an, die zu einem ungültigen Positionswert führen und ein Eingreifen des Benutzers erfordern.
35	1..4	45	Supported Alarms	R	ROM	WORD		Ein Bit für jedes unterstützte Bit in Attribut 44.
35	1..4	46	Alarm Flag	R	V	BOOL		Zeigt an, dass ein Alarm in Attribut 44 gesetzt ist. 0 = OK; 1 = Alarm Error
35	1..4	47	Warnings	R	V	WORD		Werden interne Grenzwerte überschritten, dann werden die entsprechenden Bits gesetzt.
35	1..4	48	Supported Warnings	R	ROM	WORD		Je ein Bit für jedes unterstützte Bit in Attribut 47.
35	1..4	49	Warning Flag	R	V	BOOL		Zeigt an, dass eine Warnung in Attribut 47 gesetzt ist. 0 = OK; 1 = Warnung
35	1..4	50	Operating Time	R	NV	UDINT		Speichert die Betriebszeit des BTL in Schritten von 0,1 Stunden.
35	1..4	51	Offset Value	R	NV	DINT		Der Offsetwert wird durch Setzen des Attributs 19 Preset Value berechnet.

Tabelle 9-8: Position Sensor Objekt (Fortsetzung)

9 Klassenverzeichnis (Objektverzeichnis, Fortsetzung)**9.6 Acknowledge Handler Objekt**

Das Acknowledge Handler Object verwaltet die empfangenen Nachrichtenbestätigungen. Dieses Ob-

jekt kommuniziert mit den Nachrichten generierenden Objekten in dem BTL und verwaltet Nachrichtenbestätigungen, Zeitüberschreitungen und maximale Wiederholungsversuche.

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Defaultwert	Beschreibung
43	1	1	Acknowledge Timer	R/W		BOOL	16	Zeitdauer in ms, die auf eine Bestätigung gewartet wird, bevor ein erneuter Versuch stattfindet.
43	1	2	Retry Limit	R		BOOL	1	Anzahl der maximalen Acknowledge Zeitüberschreitungen.
43	1	3	COS Producing Connection Instance	R		BOOL	4	ID der Verbindungsinstanz, die den Pfad der des produzierenden I/O Anwendungsobjekts enthält und die über Ereignisse des Acknowledge Handlers informiert wird.

Tabelle 9-9: Acknowledge Handler Objekt

9.7 Balluff Configuration Objekt

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Defaultwert	Beschreibung
100	1	1	Number of Attributes	R	ROM	USINT		Anzahl der in diesem Objekt verfügbaren Attribute.
100	1	2	Attribute List	R	ROM	STRING		Array of USINT: Liste der in diesem Objekt verfügbaren Attribute.
100	1	3	Number of Magnets	R/W	NV	USINT	1	Anzahl der Positionsgeber (1..4; 0 = Flexible Magnet Mode)
100	1	4	Autosave	R/W	NV	BOOL	TRUE	TRUE: Autosave ist aktiv (default) FALSE: Autosave ist deaktiviert
100	1	5	Assembly Options	R/W	NV	USINT	0	Bitfeld: Anordnung der Ausgabeparameter im I/O
100	1	6	Assembly Position Activ	R/W	NV	UINT	1	Bitfeld: Je ein Bit für die Übernahme des Positionswertes in die I/O-Nachricht.
100	1	7	Assembly Velocity Active	R/W	NV	UINT	1	Bitfeld: Je ein Bit für die Übernahme der Geschwindigkeit in die I/O-Nachricht.
100	1	8	Assembly CAM Active	R/W	NV	UINT	0	Bitfeld: Je ein Bit für die Übernahme des Nockenstatus in die I/O-Nachricht.
100	1	9	Assembly Warnings Active	R/W	NV	UINT	0	Bitfeld: Je ein Bit für die Übernahme der Warnungen in die I/O-Nachricht.
100	1	10	Assembly Alarms Active	R/W	NV	UINT	0	Bitfeld: Je ein Bit für die Übernahme der Alarmer in die I/O Nachricht.
100	1	11	Temperature	R	V	SINT	-	Aktuelle Temperatur °C
100	1	12	Temperature Maximum	R	NV	SINT	-	Maximale Betriebstemperatur in °C
100	1	13	Temperature Minimum	R	NV	SINT	-	Minimale Betriebstemperatur in °C

Tabelle 9-10: Balluff Configuration Objekt

9 Klassenverzeichnis (Objektverzeichnis, Fortsetzung)

Kl.	Inst.	Attr.	Name	Zugriff	Speicher	Type	Default-wert	Beschreibung
100	1	14	Direct Offset Enable	R/W	NV	BOOL	FALSE	TRUE: Ermöglicht direktes Setzen von Attribut 51 (Position Offset) im Position Sensor Object. FALSE: Offset kann nur über Preset Value gesetzt werden (default).
100	1	15	Flexible Magnet Mode Delay	R/W	NV	UINT	0	Alarmzeit bei Änderungen im Flexible Magnet Mode.
100	1	16	Sync Period	R/W	NV	UDINT	0	Erwartete Periodendauer der BitStrobe Nachrichten in ms. 0 = Deaktiviert.

Tabelle 9-11: Balluff Configuration Objekt (Fortsetzung)

Attribut 6: Assembly Positions Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Position	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Tabelle 9-12: Attribut 6

Attribut 7: Assembly Velocity Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Geschwindigkeit	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabelle 9-13: Attribut 7

Attribut 8: Assembly CAM Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAM	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Tabelle 9-14: Attribut 8

Attribut 9: Assembly Warnings Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Warnings	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabelle 9-15: Attribut 9

Attribut 10: Assembly Alarms Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alarms																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabelle 9-16: Attribut 10

10 Weitere Einstellungen

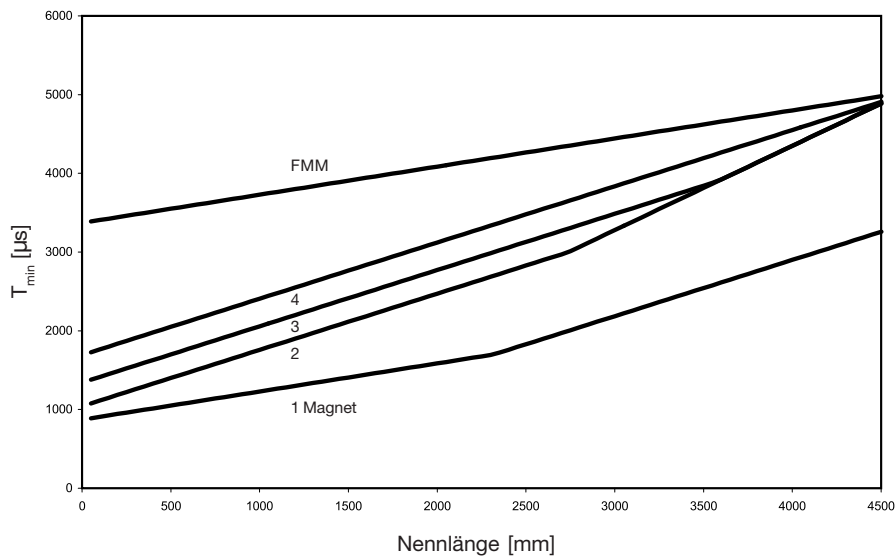


Bild 10-1: Minimale Zeit für Sync-Periode

Formel für 1 Positionsgeber:

Länge ≤ 2350 mm

$$t_{min} = (\text{Nennl.} + 60 \text{ mm}) / 2800 \text{ m/s} + \text{CALC}$$

Länge > 2350 mm

$$t_{min} = (\text{Nennl.} + 60 \text{ mm}) / 2800 \text{ m/s} * 2$$

Formel für 2 Positionsgeber:

Länge ≤ 2750 mm

$$t_{min} = (\text{Nennl.} + 60 \text{ mm}) / 2800 \text{ m/s} * 2 + \text{CALC}$$

Länge > 2750 mm

$$t_{min} = (\text{Nennl.} + 60 \text{ mm}) / 2800 \text{ m/s} * 3$$

Formel für 3 Positionsgeber:

Länge ≤ 3600 mm

$$t_{min} = (\text{Nennl.} + 60 \text{ mm}) / 2800 \text{ m/s} * 2 + \text{CALC}$$

Länge > 3600 mm

$$t_{min} = (\text{Nennl.} + 60 \text{ mm}) / 2800 \text{ m/s} * 3$$

Formel für 4 Positionsgeber:

$$t_{min} = (\text{Nennl.} + 60 \text{ mm}) / 2800 \text{ m/s} * 2 + \text{CALC}$$

Formel für FMM:

$$t_{min} = (\text{Nennl.} + 60 \text{ mm}) / 2800 \text{ m/s} + 1700 \mu\text{s} + \text{CALC}$$

CALC = 850 μs (1 Positionsgeber)

CALC = 1.000 μs (2 Positionsgeber)

CALC = 1.300 μs (3 Positionsgeber)

CALC = 1.650 μs (4 Positionsgeber)

CALC = 1.650 μs (FMM-Mode)

BTL5-D11...

english

User's Guide Configuration and CAN Interface

Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Germany
Phone +49 7158 173-0
Fax +49 7158 5010
Servicehotline +49 7158 173-370
DeviceNet@balluff.de
www.balluff.com

DeviceNet™

Characteristics

The BTL5-D... allows you to configure the BTL in a DeviceNet system:

- BTL operating parameters
- Position and velocity resolution
- Units
- Scaling
- Cams/switchpoints with polarity and hysteresis
- Limits
- Working ranges

DeviceNet – communication parameters:

- Number of output bytes
- Expected transmission rate
- Explicit Messaging
- Polled I/O
- BitStrobe I/O
- COS / Cyclic I/O

Bus parameters:

- Setting the transmission speed (baud rate)
- Setting the MAC ID

Definitions

baud rate Data transmission speed on the CAN-Bus

CAN Controller Area Network

MAC ID Node number for the device-specific identification (Media Access Control Identifier)

CIP Common Industrial Protocol

COS Change of State

ODVA Open DeviceNet Vendor Association

Contents

1	System description of the BTL5-D11... on the CAN-Bus	3
1.1	CAN technology	3
1.2	DeviceNet technology	3
1.3	Data transmission	3
1.4	Object model	3
1.5	Node identification	4
1.6	Operating parameters	4
1.7	System start	6
1.8	Estimating the transmission duration	6
1.9	Cable lengths	7
2	Data transmission in an I/O message	8
2.1	Configuration of the I/O messages	8
2.2	Data format for position and velocity	9
3	Error messages and warnings	9
3.1	Error messages over DeviceNet	9
3.2	Device-internal alarms and warnings	10
4	Communication over DeviceNet	11
4.1	Step 1: Opening the connection	11
4.2	Step 2a: Querying an attribute	11
4.3	Step 2b: Setting an attribute	11
4.4	Step 2c: Invoking a service	12
4.5	Step 3: Closing the connection	12
5	Parameterizing the DeviceNet interface	13
5.1	Reading the ID data	13
5.2	Assigning the baud rate	13
5.3	Assigning the MAC ID	13
5.4	Saving	14
6	Node configuration	14
6.1	Mode type (messaging type)	14
6.2	Example for opening and closing communication	14
6.3	Number of magnets	17
6.4	Position data	18
6.5	Velocity settings	22
7	Additional attributes	23
7.1	Position Sensor Object	23
7.2	Balluff Configuration Object	24
8	Startup	25
8.1	Setup under RsNetworx	25
9	Class directory (Object directory)	28
9.1	Identity Class (ID = 1)	29
9.2	DeviceNet Class (ID = 3)	30
9.3	Assembly Class (ID = 4)	30
9.4	Connection Object	31
9.5	Position Sensor Object	32
9.6	Acknowledge Handler Object	34
9.7	Balluff Configuration Object	34
10	Additional settings	36

1 System description of the BTL5-D11... on the CAN-Bus

1.1 CAN technology

DeviceNet is an open system built on the base of CAN. CAN was developed some years ago by the R. Bosch company for data transmission in motor vehicles. Since then millions of CAN chips have come into use. A drawback to using CAN in automation technology is that CAN includes no definitions for the application layer. CAN defines only the physical and the data security level.

1.2 DeviceNet technology

DeviceNet specifies a uniform application layer that makes the CAN protocol useful for industrial applications. The ODVA supports manufacturers and users of the DeviceNet system as an independent association. The ODVA ensures that all devices based on the specification work together in a system without regard to the manufacturer.

1.3 Data transmission

Data transmission in DeviceNet takes place using message telegrams. The telegrams can be divided basically into CAN ID and 8 message bytes:

CAN ID	Message Header	Message Body
11 Bits	1 Byte	7 Bytes

1.3.1 Definition of the CAN ID

DeviceNet is based on the standard CAN protocol and uses an 11-bit (2048 discrete messages) message identifier. 6 bits are sufficient to identify a device or node in a DeviceNet network, since a network is limited to 64 devices. This is called the MAC ID. The CAN Identifier is made up of the Message Group, the Message ID within this group, and the MAC ID of the device.

The BTL5-D11... is considered a Group 2 Only Slave. The most important CAN IDs for particular types of communication are shown in the table below:

10	9	8	7	6	5	4	3	2	1	0	Use	Range [hex]
0	Group 1 Message ID				Source MAC ID						GROUP 1 Message	0x000-0x3ff
0	1	1	0	1	Source MAC ID						Slave's I/O Change of State or Cyclic Message	
0	1	1	1	1	Source MAC ID						Slave's I/O Poll Response or Change of State/ Cyclic Acknowledge Message	
1	0	MAC ID						Group 2 Message ID			GROUP 2 Message	0x400-0x5ff
1	0	Destination MAC ID						0 1 0			Master's Change of State or Cyclic Acknowledge Message	
1	0	Source MAC ID						0 1 1			Slave's Explicit/Unconnected Response Messages	
1	0	Destination MAC ID						1 0 0			Master's Explicit Request Message	
1	0	Destination MAC ID						1 0 1			Master's I/O Poll Command/Change of State/Cyclic Message	
1	0	Destination MAC ID						1 1 0			Group 2 Only Unconnected Explicit Request Message (reserved)	
1	0	Destination MAC ID						1 1 1			Duplicate MAC ID Check Messages	

Table 1-1: Definition of the CAN ID

1.4 Object model

DeviceNet describes all data and functions of a device based on an object model. With the help of this object-based description a device can be fully defined using individual objects. An object is determined by the combination of associated attributes (e.g. process data), its functions made available to the outside (read or write access to an individual attribute), as well as by its defined behavior.

DeviceNet distinguishes among three object types:

1. Communication objects

Define the messages sent over DeviceNet and referred to as Connection Objects. (DeviceNet Object, Message Router Object, Connection Object, Acknowledge Handler Object)

2. System objects

Define general DeviceNet-specific data and functions. (Identity Object, Parameter Object)

3. Application-specific objects

Define device-specific data and functions. (Application Object, Assembly Object).

1 System description of the BTL5-D11... on the CAN-Bus (cont.)

1.5 Node identification

The BTL5-D11... Micropulse transducer is operated on the CAN bus and represents a node (MAC ID) in the bus system. It functions in CAN networks for DeviceNet and Class 2.

The unique identification of the BTL displacement feedback system is accomplished using the Identity Object; this information is permanently stored in the device, e.g.:

Vendor ID: 0x2B BALLUFF
 Device type: 0x09 Resolver Device
 Product code: 0x01 Profile
 0x02 Rod
 Serial number: 957F9
 Product name: BTL5D

1.6 Operating parameters

To ensure optimum industrial usability, two different message types are provided: I/O messages (Implicit Messaging) and Explicit Messages. This means the BTL is universally compatible with DeviceNet

I/O messages are used to exchange I/O data in real-time. The BTL data position, velocity and CAM status for all four magnets can be sent using the I/O message

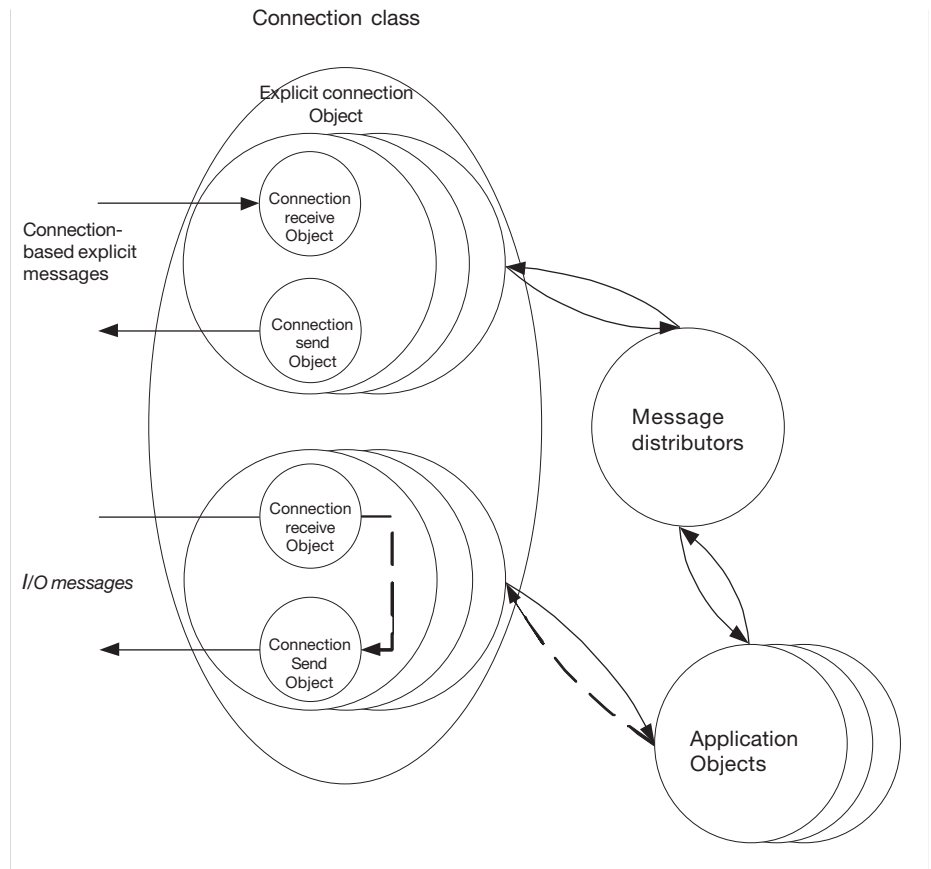


Fig. 1-1: Object model of the BTL5-D11...

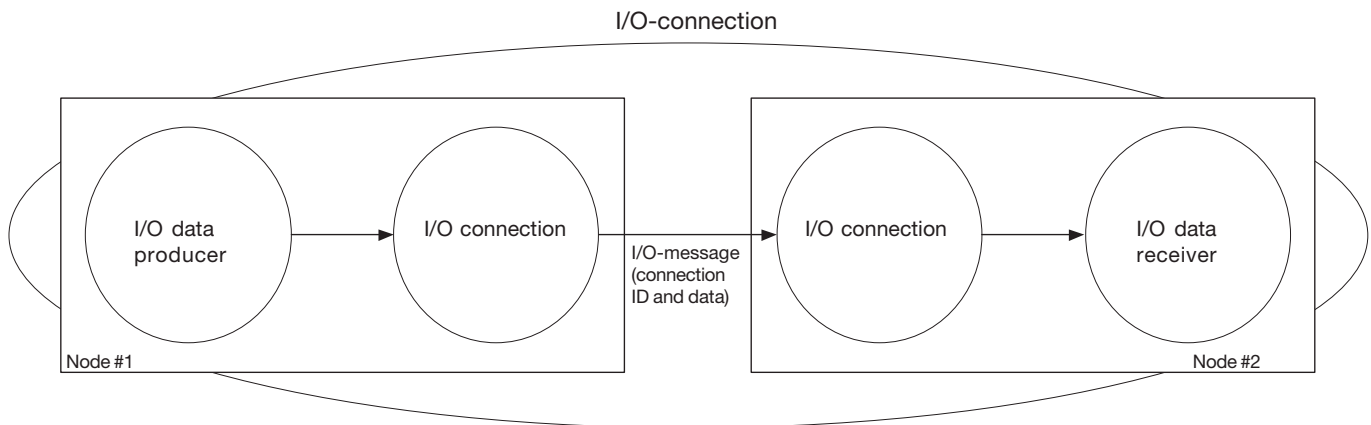


Fig. 1-2: I/O message

1 System description of the BTL5-D11... on the CAN-Bus (cont.)

Explicit messages are used to exchange data for configuring a device, for example the parameters for configuring the transducer. CIP provides the user with four essential functionalities:

- Uniform control services
- Uniform communication services
- Uniform distribution of messages
- Common knowledge basis

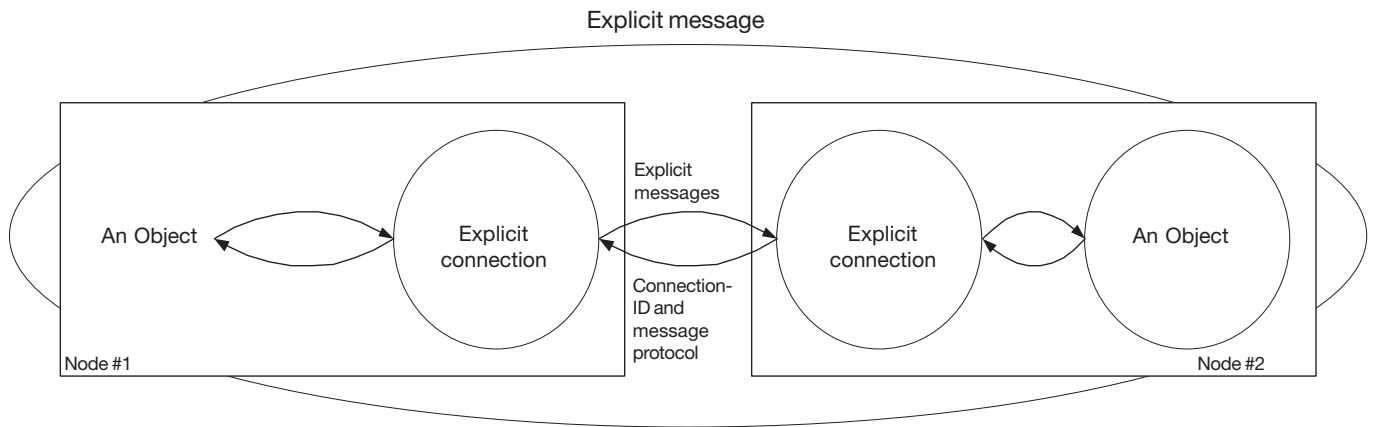


Fig. 1-3: Explicit message

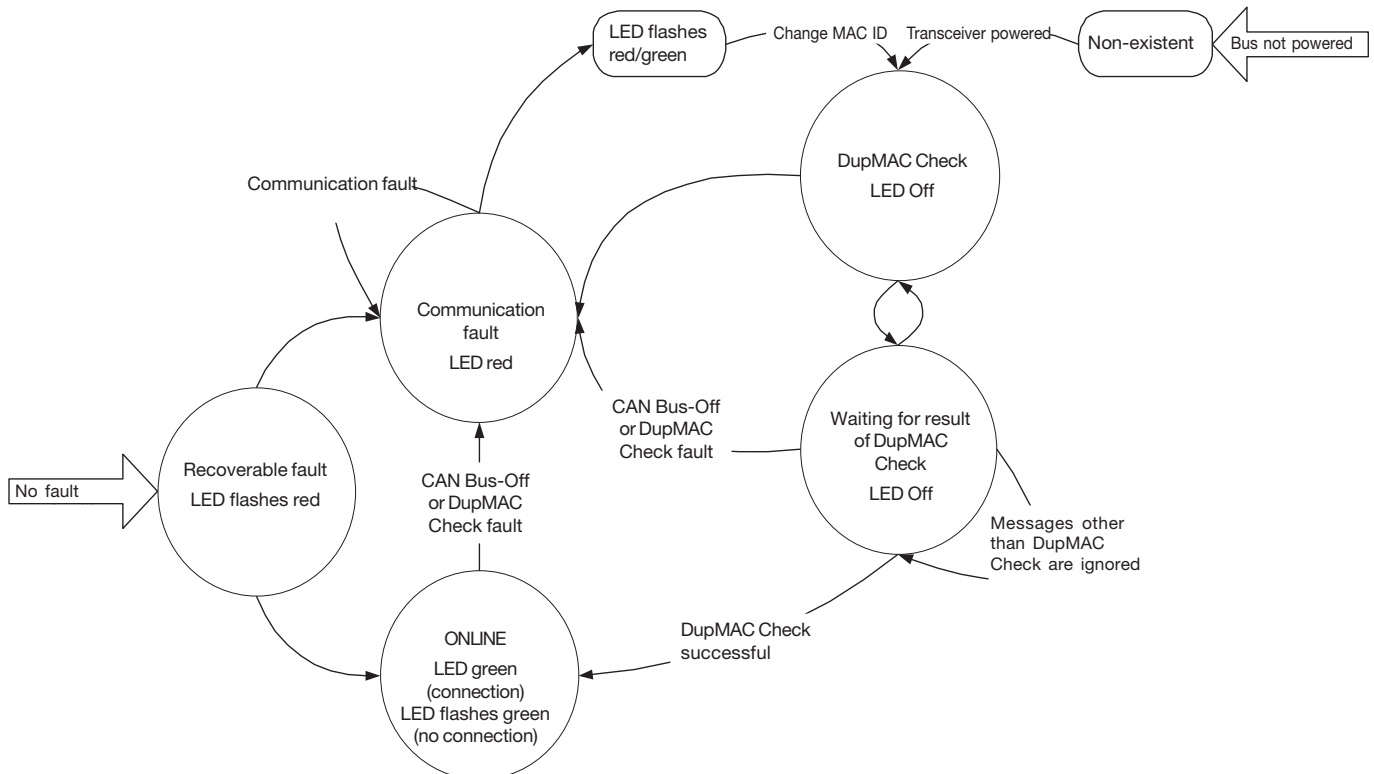


Fig. 1-4: State diagram

1 Systembeschreibung des BTL5-D11... am CAN-Bus (cont.)

The type BTL5-D11... transducers are shipped with the CIP 2.1 PositionSensor Object. The configurable parameters of the DeviceNet Communication Profile and the PositionSensor Object can be set using the explicit message "Explicit Message" (Set Attribute).

Class 1: Identity Object (read-only)
➔ Section 9.1

- Vendor
- Product ID
- Product name
- Serial number
- Status

Class 3: DeviceNet Object
➔ Section 9.2

- Bus parameters
- MAC ID
- baud rate

Class 4: Assembly Object
➔ Section 9.3

- Copy of the I/O data for accessing using Explicit Messaging

Class 5: Connection Object
➔ Section 9.4

- Number of output bytes
- Expected transmission rate
- Explicit Messaging
- Polled I/O
- BitStrobe I/O
- COS / Cyclic

Class 35: Position Sensor Object
➔ Section 9.5

- Operating parameters of the BTL
- Position and velocity resolution
- Units
- Scaling
- CAMs / switchpoints with polarity and hysteresis
- Limits
- Working ranges

Class 43: Acknowledge Handler Object ➔ Section 9.6

- Manages message confirmations

Class 100: Balluff Configuration Object ➔ Section 9.7

- I/O message configuration

- Special configuration parameters

1.7 System start

After a system start (or after a reset) the BTL starts up as indicated in the state diagram ➔ Fig. 1-4.

1.7.1 DupMAC Check

After power-on each node sends a duplicate MAC ID Check Request message. The node sends this practically to its own address. The message data contain the vendor ID and the serial number of the node. The LED stays off.

1.7.2 Waiting for a result from DupMAC Check

After the duplicate MAC ID Check Request message is sent the node waits for maximum one second for a duplicate MAC ID Check Response message to arrive. If the address used by the node in fact exists only once in the network, then the DupMAC message does not get a response. After the second has expired it is known that the node address exists only once in the network and the node changes to the online state.

If a response is received later, then the device immediately switches to the communication fault mode and the LED stays on red.

If there is another node with the same MAC ID, then it is addressed by the duplicate MAC ID Check Request message and sends a duplicate MAC ID Check Response message. This tells the sender that its address is no longer available, and it changes over to the communication fault state.

1.7.3 Online

In the ONLINE state the device is ready. If the device is addressed or sends messages, then the LED is constant green. If the device was not addressed within the last 10 seconds or has not itself sent any messages during this time, the device closes the connection and the LED flashes green.

1.7.4 Communication faults

If the CAN bus is faulted, if the bus is getting no power or if a duplicate MAC ID Check determined that the device's own MAC ID is already in use, the node switches to this state. The red LED is constantly red.

The device can also be addressed by a Communicated Faulted Message Request. In this state one can identify the device based on its vendor ID and serial number. You can also assign it a new MAC ID.

1.7.5 Recoverable faults

In the case of minor faults, such as when a duplicate MAC ID Check Request message was received with its own address, the LED flashes red. The device is on Offline mode, but is basically still ready for communication.

1.8 Estimating the transmission duration

Since there are dependencies between the CAN bus cable length and the baud rate as well as the selection of the data interval, it can be important to estimate the transmission duration for continuous transmission when it comes to setting the baud rate and data interval.

A CAN I/O message contains by default the following data:

- 35 bits for the transmission in the CAN protocol
- 4 bytes for position à 8 bits
- 1 byte for CAM status with 8 bits

1 System description of the BTL5-D11... on the CAN-Bus (cont.)

A message in standard format with 8 data bytes requires a maximum of 130 bits. This assumes a maximum number of 19 Stuff bits and 3 Inter Frame Space bits.

Baud rate	Transmission duration	Possible data intervals
[kBaud]	t / ms	t / ms
500	0.212	≥ 0.250
250	0.424	≥ 0.500
125	0.848	≥ 1.000

If more than 8 bytes need to be sent, the messages are automatically fragmented. Since 1 byte for the fragmentation ID then is dropped for each CAN message, the transmission duration then increases disproportionately.

The transmission duration as a function of baud rate can be taken from the adjacent diagram. The steps arise whenever an additional fragment is required ➡ Fig. 1-5.

1.9 Cable lengths

Depending on the selected baud rate, various maximum cable lengths are possible. The cable lengths shown should be considered a total of the trunk cable and the stubs.

Baud rate	Cable length
125 kBaud	500 m
250 kBaud	250 m
500 kBaud	100 m

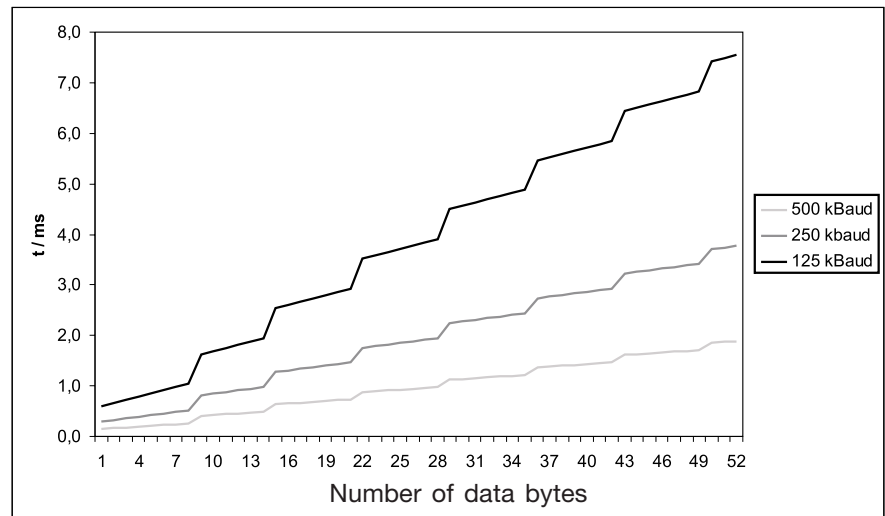


Fig. 1-5: Diagram - Transmission duration

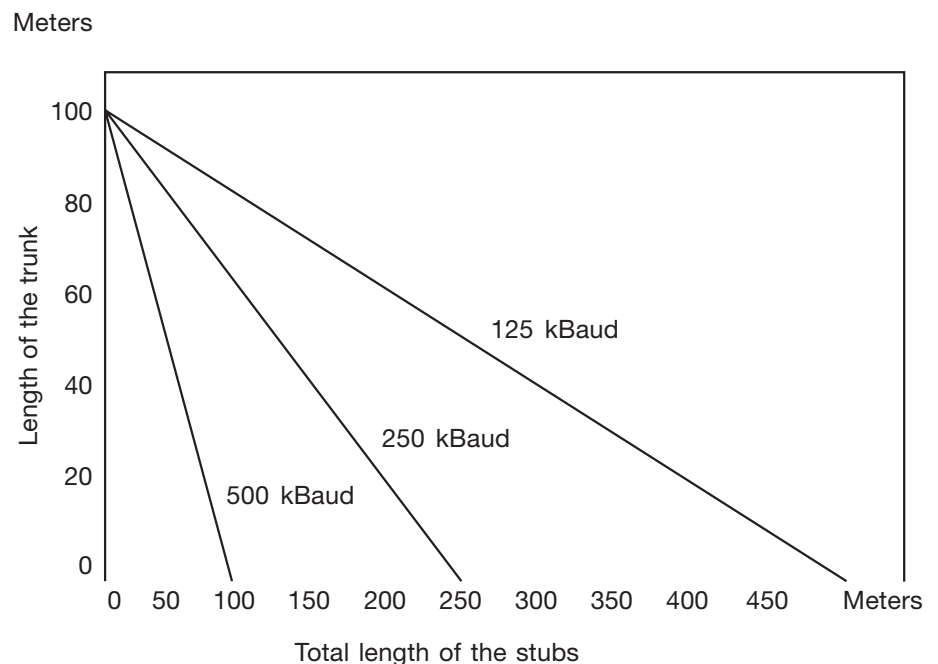


Fig. 1-6: Cable lengths

$$L_{\text{stub}} + 5 \times L_{\text{trunk}} = 500 \text{ m} \quad \text{at 125 kBaud}$$

$$L_{\text{stub}} + 2.5 \times L_{\text{trunk}} = 250 \text{ m} \quad \text{at 250 kBaud}$$

$$L_{\text{stub}} + L_{\text{trunk}} = 100 \text{ m} \quad \text{at 500 kBaud}$$

L_{stub} is the length of the stub and L_{trunk} the length of the trunk (main) cable.

2 Data transmission in an I/O message

I/O messages can be polled, cyclically or if the data are changed automatically sent, or cyclically queried. The process actual value is send according to the telegram scheme ➡ Table 2-1:

This example corresponds to the preset telegram, in which the current position value of the first magnet is sent to bytes 0 to 3 and its CAM status (CAM) to byte 4. By using the option attributes described in section 9.7 "Balluff Configuration Object" position data, velocity, CAM status, warnings or alarms can be copied for all four possible magnets and in any desired combination to the I/O message.

Note that a maximum of 8 bytes (e.g. two position values) can be sent in one CAN message before fragmented messages are automatically sent, which are divided over multiple telegrams.

2.1 Configuration of the I/O messages

The following attributes of the Position Sensor Object can be activated into the I/O message for all four magnets:

- Position Position Sensor Object Attribute 10: Position Value Signed
- Velocity Position Sensor Object Attribute 24: Velocity Value
- Channel Position Sensor Object Attribute 35: CAM Channel State Register
- Warnings Position Sensor Object Attribute 47: Warnings
- Alarms Position Sensor Object Attribute 44: Alarms

Section 9.7 "Balluff Configuration Object" describes the attributes

- Attribute 6: Assembly Position Active: Activates Position Value Signed
- Attribute 7: Assembly Velocity Active: Activates Velocity Value
- Attribute 8: Assembly CAM Activ: Activates CAM Channel State Register
- Attribute 9: Assembly Warnings Active: Activates Warnings

CAN ID	Position value				CAM
11 Bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
500	2 ⁷ to 2 ⁰	2 ¹⁵ to 2 ⁸	2 ²³ to 2 ¹⁶	2 ³¹ to 2 ²⁴	2 ⁷ to 2 ⁰

Table 2-1: Telegram scheme

Structure of Attribute 6 ... 10 in the Balluff Configuration Object																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Magnet	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2-2: Object configuration

- Attribute 10: Assembly Alarms Active: Activates Alarms

In each of these attributes there is one bit assigned to a magnet. When this bit is set, the corresponding value is sent to the I/O message.

If for example Bit 2 in the attribute "Assembly Position Active" is set, the position value of Magnet 3 is sent to the I/O message.

Example: You want to activate in the I/O the position value of Magnet 2, CAM status of Magnet 3 and warnings for Magnet 1. You then set the attributes as follows:

Position2: Attribute 6: Assembly Position Active Bit 1 -> binary 0000 0000 0000 0010 = 0x02 (Hex)

CAM2: Attribute 8: Assembly CAM Active Bit 2 -> binary 0000 0000 0000 0100 = 0x04 (hex)

Warnings1: Attribute 9: Assembly Warnings Active Bit 0 -> binary 0000 0000 0000 0001 = 0x01 (hex)

2.1.1 Transmission sequence

There are two ways to sequence the data in the I/O message. The default setting is for a preset arrangement as shown in ➡ Table 2-3. This sequence is preferred, since it is the faster method.

The transmission sequence is defined by the order number at left. The values are sent in the order of their arrangement regardless of the sequence of their activation.

For our example in the previous section this means:

Position2 has order number 6, CAM3 has order number 13 and Warnings1 order number 4.

In other words, Warnings1 comes first, then Position2 and finally CAM3 ➡ Table 2-4:

Assembly Options Bit 0 = 0 (Default)									
1	Position1	2	Velocity1	3	CAM1	4	Warnings1	5	Alarms1
6	Position2	7	Velocity2	8	CAM2	9	Warnings2	10	Alarms2
11	Position3	12	Velocity3	13	CAM3	14	Warnings3	15	Alarms3
16	Position4	17	Velocity4	18	CAM4	19	Warnings4	20	Alarms4

Table 2-3: Default transmission sequence

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Warnings1	Position2			CAM3	

Table 2-4: Transmission - Example 1

2 Data transmission in an I/O message (cont.)

An alternate transmission sequence can be activated if Bit 0 in Attribute 5: Assembly Options in the Balluff Configuration Object is set (➡ Section 9.7). Then the arrangement shown in ➡ Table 2-5 applies:

Here the order numbers are arranged differently. In this case and in our example Position2 would receive Order Number 2, CAM3 order number 11 and Warnings1 order number 13.

A change in the configuration of the I/O messages has no effect on an already existing I/O connection and must be done before opening the I/O connection.

2.2 Data format for position and velocity

The current data for the transducer (position, velocity) is provided as data type DINT. The general conditions (units, increment, offset, ...) are set in the associated instance of the Position Sensor

Assembly Options Bit 0 = 1									
1	Position1	5	Velocity1	9	CAM1	13	Warnings1	17	Alarms1
2	Position2	6	Velocity2	10	CAM2	14	Warnings2	18	Alarms2
3	Position3	7	Velocity3	11	CAM3	15	Warnings3	19	Alarms3
4	Position4	8	Velocity4	12	CAM4	16	Warnings4	20	Alarms4

Table 2-5: Transmission sequence

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Position2				CAM3	Warnings1

Table 2-6: Transmission- Example 2

class, which is described in Section 9.5 "Position Sensor Object".

Coding is in accordance with ICC 754-1985 and can therefore be directly processed by most processors and compilers without any additional conversion.

3 Error messages and warnings

3.1 Error messages over DeviceNet

After invoking a service, sometimes an Error Response message is sent

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3
ID	Source	Res- ponse	Error Code	Add. Code

Table 3-1: Error Response message

This contains a general error code and an additional error code which sometimes includes further information about the error which occurred. In the following the general error codes which can occur when using the BTL are listed.

3.1.1 Code 0: Service Success

The service was successfully carried out by the specified object.

3.1.2 Code 2: Resource Unavailable

There are no resources available for processing a request. This can happen for example when there is a Change-of-State connection and a poll connection also needs to be established. This happens even if a connection is supposed to be closed and another Allocation Choice is set than when opening the connection.

3.1.3 Code 8: Service Not Supported

There is no service available for the corresponding object. For example, the Reset service is only available for certain objects. If the Autosave attribute is set in the Balluff Configuration Object, then the Save service is not available in the Balluff Configuration Object and in the Position Sensor Object and this error code will be generated.

3.1.4 Code 9: Invalid Attribute Value

An attempt was made to set an attribute to an invalid value. Many attributes have a limited value range, for example an address may lie only between 0 and 63.

3.1.5 Code 11: Already in requested Mode / State

This error can occur in two cases:

An I/O connection cannot be allocated again if it has already been allocated, even if it is in Timeout state. First close the connection.

An attempt was made to close a connection which was already closed or which does not yet exist. Sometimes this error message appears if the BTL already set a connection to Timeout state before the Master could resolve it. The error message also appears after a Reset service has been

3 Error messages and warnings (cont.)

performed, since as far as the Master is concerned the connection is still intact even though the Slave "forgot" it due to its reset. In this case the error message can be ignored.

3.1.6 Code 12: Object State Conflict

An object can not perform the requested service in its current mode or state. For the Position Sensor object this happens when the Preset Value attribute is set although there is presently a position error and no valid position value can be obtained for this function.

3.1.7 Code 14: Attribute not Setable

An attribute can not be set. Attributes marked only for R and not for R/W can only be read.

3.1.8 Code 19: Not Enough Data

The number of data for setting an attribute is insufficient. For example only two data bytes were sent for setting a four-byte attribute.

3.1.9 Code 20: Attribute Not Supported

An attribute does not exist. The reason is usually an incorrectly entered attribute number. Some attributes are sometimes only available under certain conditions.

3.1.10 Code 22: Object does not exist:

An attempt was made to access an object which does not exist in this device. The reason is usually an incorrectly entered class number.

3.1.11 Code 23: Fragmentation Segment Error

In sending a Fragmented Message a segment was not sent or incorrectly sent.

3.1.12 Code 25: Store Operation

A store operation failed in a non-volatile (EEPROM) memory.

3.1.13 Code 32: Invalid Parameter

An attempt was made to set an invalid parameter which lies outside the permissible range.

3.1.14 Class specific error codes

The following error codes occur only in conjunction with certain classes. They are output as an additional code together with the main error code:

3.2 Device-internal alarms and warnings

The Position Sensor Object contains Attribute 44 (Alarms), 46 (Alarm Flag), 47 (Warnings) and 49 (Warning Flag) in which device-internal warnings and alarms are output. These attributes can be polled with Explicit Messaging. The conditions which result in these alarms and warnings can be influenced by the user (► Section 9.5 Position Sensor Object).

The alarms (Attribute 44) and warnings (Attribute 47) of the Position Sensor Object can also be copied to the assembly using Attribute 9 (Assembly Warnings Active) and Attribute 10 (Assembly Alarms Active).

Code		Description
1	Allocation Conflict	An attempt was made to invoke the Predefined Master Slave Connection from a MAC ID which does not agree with the stored MAC ID.
2	Invalid Allocation Choice	This code occurs in conjunction with Code Resource Unavailable, Invalid Attribute Value and Already in Mode / State and indicates that the desired allocation is not possible.
3	Invalid Unconnected Request	A Group 2 Unconnected Request message is invalid.
4	Object Specific Error	In conjunction with Resource Unavailable. An attempt was made to establish a poll connection in addition to a COS / Cyclic connection.

Table 3-2: Class-specific error codes

4 Communication over DeviceNet

For most applications the communication always runs according to the same scheme. First a connection between Master and Slave needs to be established. Then this connection can be used to exchange data. Multiple attributes can be read or set. Finally, the connection is ended when it is no longer needed. After 10 seconds the connection is automatically ended due to a timeout. In the following examples the Master has MAC ID 0 and the BTL MAC ID 63. The bytes which do not change depending on the MAC IDs or the queried attributes are highlighted.

4.1 Step 1: Opening the connection

A connection is opened by allocating the Master Connection Set:

The CAN ID 0x5FE is binary 10 111111 110. The first two binary digits are 10 for message group 2.

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FE	0x00	0x4B	0x03	0x01	0x01	0x00
	Source	Service	Class ID	Instance ID	Explicit	Allocator MacID

Table 4-1: Master Connection Set

The following 6 bits are the address of the Slave (BTL). The last 3 bits contain the Message ID. 0x06 (binary 110) is the identifier for an Unconnected Explicit Request.

Byte 0 contains the Source MAC ID, here 0x00. Byte 1 is 0x4B and indicates the Service "Allocate M/S Connection Set". Byte 2 specifies the Class ID. 0x03 is the ID of the DeviceNet Class described in Section 9.2. The instance ID in byte 3 points to Instance 1 and is set in Byte 4 for an Explicit Connection. Byte 5 contains the MAC ID to which this connection is allocated. Here again the Master MAC ID is used.

CAN ID	Byte 0	Byte 1	Byte 2
0x5FB	0x00	0xCB	0x00
	Source	Response	Error Code

Table 4-2: Response message

4.2 Step 2a: Querying an attribute

The CAN ID 0x5FC is constructed exactly as already described above. But here the last three bits are now 100 and indicate a "Masters Explicit Request Message". Byte 1 is for the service "Get Attribute Single". Bytes 2 and 4 contain Class (Object) ID, Instance ID and Attribute ID. Here Class 3 (Device Net), Instance 1 and Attribute 1 (MAC ID) are queried.

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x5FC	0x00	0x0E	0x03	0x01	0x01
	Source	Service	Class ID	Instance ID	Attribute ID

Table 4-3: Masters Explicit Request Message

The response is the desired value:

CAN ID	Byte 0	Byte 1	Byte 2
0x5FB	0x00	0x8E	0x3F
	Source	Response	Value

Table 4-4: Response value

4.3 Step 2b: Setting an attribute

Byte 1 stands for the service "Set Attribute Single". Bytes 2 to 4 contain the Class (Object) ID, Instance ID and Attribute ID. Here Class 35 (Position Sensor Object), Instance 1 (Magnet 1), Attribute 12 (Direction Counting Toggle) is set to the value 1 given to byte 5.

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FC	0x00	0x10	0x23	0x01	0x0C	0x01
	Source	Service	Class ID	Instance ID	Attribute ID	Value

Table 4-5: Set Attribute Single

As a response either the action is confirmed or an Error Response is returned:

CAN ID	Byte 0	Byte 1
0x5FB	0x00	0x90
	Source	Response

Table 4-6: Response (Set Attribute)

4 Communication over DeviceNet (cont.)

4.4 Step 2c: Invoking a service

Byte 1 stands for the service. Here the "Reset" service is invoked. Bytes 2 and 3 contain Class (Object) ID and Instance. Since the Reset service always affects an entire Instance, the Attribute ID is not needed. Here Class 100 (Balluff Configuration Object), Instance 1 is reset.

For some objects the Reset command also has an additional argument. If this is 0, then only a Reset is carried out, as if the supply voltage were turned off and on. If the argument is 1, then the

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FC	0x00	0x05	0x64	0x01	0x01	0x01
	Source	Service	Class ID	Instance ID	Attribute ID	Argument

Table 4-7: Reset Service

Instance is reset to its default value and then a Reset is carried out.

As a response either the action is confirmed or an Error Response is returned:

CAN ID	Byte 0	Byte 1
0x5FB	0x00	0x85
	Source	Response

Table 4-8: Response (service invoke)

4.5 Step 3: Closing the connection

This sequence resembles that which was used at the beginning to establish the connection. Except now the Service Code 0x4C for "Release M/S Connection Set". Byte 5 is no longer needed for closing the connection.

5FE 00 4C 03 01 01

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0x5FE	0x00	0x4C	0x03	0x01	0x01
	Source	Service	Class ID	Instance ID	Explicit

Table 4-9: Release M/S Connection Set

5 Parameterizing the DeviceNet interface

5.1 Reading the ID data

To get the ID data, you only need to turn on the device, since it then automatically sends a Duplicate MAC ID Check, which consists of two identical messages.

The ID contains in the first two binary digits 10 to indicate a Group 2 message. The next 6 bits represent the MAC ID of the BTL (here decimal 63). The last three bits indicate that the message is a Duplicate MAC ID Request.

A message then has the following format:

0x5FF	0x00	0x2B	0x00	0xF9	0x57	0x09	0x00
		Lo	Hi	Lo			Hi
ID	Phys.	Manufacturer ID		Serial number			

Table 5-1: Duplicate MAC ID Request message

The Vendor ID is always 0x00 2B and indicates Balluff GmbH.

The last four bits contain the serial number of this BTL.

These attributes can also be queried using the Identity Object described in Section 9.1 Identity Class (ID = 1).

5.2 Assigning the baud rate

The baud rate is stored in Class 3 (DeviceNet), Instance 1, Attribute 2 as USINT. It can be changed using the following message sequence:

Message from Master:

0x5FE 0x00 0x4B 0x03 0x01 0x01 0x00

➡ Section 4.1: Step 1: Opening the connection

Response from BTL:

0x5FB 0x00 0xCB 0x00

➡ Section 4.3: Step 2b: Setting an attribute. The value sent is in accordance with the usual DeviceNet convention:

Message from Master

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FC	0x00	0x10	0x03	0x01	0x02	Baud
	Source	Service	Class ID	Instance ID	Attribute ID	Value

Table 5-2: Assigning the baud rate

Baud	Meaning
0	125 kBaud
1	250 kBaud
2	500 kBaud

Response from BTL:

0x5FB 0x00 0x90

Master:

0x5FE 0x00 0x4C 0x03 0x01 0x01

➡ Section 4.5 Step 3: Closing the connection

BTL:

0x5FB 0x00 0xCC

This change only affects the internally stored baud rate, which can only be used and modified if DIP switch 8 and 9 are in the "On" position. The change only takes effect after the next Reset.

5.3 Assigning the MAC ID

The MAC ID is stored in Class 3 (DeviceNet), Instance 1, Attribute 1 as USINT. It can be changed using the following message sequence:

Message from Master:

0x5FE 0x00 0x4B 0x03 0x01 0x01 0x00

➡ Section 4.1: Step 1: Opening the connection

Response from BTL:

0x5FB 0x00 0xCB 0x00

➡ Section 4.3: Step 2b: Setting an attribute. The value sent for the MAC ID must lie between 0 and 63.

Message from Master

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x5FC	0x00	0x10	0x03	0x01	0x01	MacID
	Source	Service	Class ID	Instance ID	Attribute ID	Value

Table 5-3: Assigning the MAC ID

Response from BTL:

0x5FB 0x00 0x90

If DIP switch 7 is in the "On" position, then the internally stored MAC ID is used. In this case the BTL immediately performs a Reset after the MAC ID is changed and sends a Duplicate MAC ID Check' with its new MAC ID.

Master:

0x5FE 0x00 0x4C 0x03 0x01 0x01

➡ Section 4.5 Step 3: Closing the connection

BTL:

0x5FB 0x00 0xCC

This change only affects the internally stored MAC ID, which can only be changed if DIP switch 7 is in the "On" position.

5 Parameterizing the DeviceNet interface (cont.)

5.4 Saving

Attributes indicated with NV in the class directory (➔ Section 9) are automatically saved to non-volatile memory (EEPROM) when they are modified.

For attributes of the Position Sensor Object described in Section 9.5 Balluff Configuration Object, automatic saving can be turned off. To do this use the parameter

"Attribute 4: Autosave" in the Balluff Configuration Object. If Autosave is deactivated, then the changed data of the Position Sensor Object is saved only after invoking the Save service. In a normal application, where the BTL is parameterized only once before it is used, Autosave should remain activated.

Only in special cases in which there are continuous changes during operation of the attributes indicated with NV should you deactivate Autosave, since then the time for saving the data is eliminated.

6 Node configuration

6.1 Mode type (messaging type)

Depending on which conditions are in effect for using the transducer, the BTL data can be sent using Cyclic I/O continuous, Change of State for when there is a state change, or on demand.

In the following examples we always assume a Master MAC ID 0x00, as well as a Slave MAC ID 0x63.

6.1.1 Bit Strobe

The Master uses a Bit Strobe command to send a general synchronization signal to all Slaves. The Bit Strobe Command Message contains a string of 64 bits (8 bytes). The sent data bit has no meaning for the BTL, but the Bit Strobe command itself causes the measured data to be output and triggers the measurement.

The BTL can start its measurement so that current data are present in time at the next Bit Strobe command. But to do this the Bit Strobe command must come at as regular a cycle rate as possible and the Sync Period attribute must be set to the time between two Bit Strobe commands (➔ Section 7.2.3 "Attribute 16: Sync Period").

6.1.2 Polled

Polled Mode is a classic Master-Slave communication. The Master can use the Poll Command Message to poll the current actual position value of the transducer. The BTL then sends the process actual value to the Master in the form of the defined I/O message using a Poll Response Message.

6.1.3 Cyclic

The BTL sends the current process value depending on a programmed timer. This can reduce the bus load, since the slave in the network only reports to the master after a certain time interval without being prompted.

6.1.4 Change of State (COS)

The BTL sends the process value if it changes by more than the amount set in Attribute 20 (COS/Delta) of the Position Sensor Object. If Attribute 20 is set to zero, then every change is sent.

This mode can be combined with Cyclic if you also set the timer. Then the BTL sends its data cyclically as well as immediately when there are changes.

6.2 Example for opening and closing communication

Data exchange with a DeviceNet device can take place only over an existing connection. Therefore you must first open this channel. To do this, send a telegram with 6 data bytes.

6 Node configuration (cont.)

Allocate Cyclic

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4B]						
2	Class ID [0x03 DeviceNet]							
3	Instance ID [01 Explicit Message]							
4	Allocation Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explicit
5	[00]		Allocator MAC ID					

Byte 2 indicates the service (command)

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4B	0x03	0x01	0x61	0x00		

Set Expected_Packet_Rate of the explicit message to 0:

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FC	0x00	0x10	0x05	0x01	0x09	0x00	0x00	

Set Expected_Packet_Rate of the cyclic connection

The BTL should send a value every 100 ms (Bytes 5 and 6)

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FC	0x00	0x10	0x05	0x04	0x09	0x64	0x00	

Release Cyclic

Ends Cyclic transmission if it is no longer needed. The BTL stops sending cyclically.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4C]						
2	Class ID [0x03 DeviceNet]							
3	Instance ID [01 Explicit Message]							
4	Release Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explicit

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4C	0x03	0x01	0x60			

6 Configuration of the node (cont.)

Release explicit Messaging

Ends explicit transmission.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4C]						
2	Class ID [0x03 DeviceNet]							
3	Instance ID [01 Explicit Message]							
4	Release Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explicit

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4C	0x03	0x01	0x01			

6.2.1 COS - Change of State Mode

The BTL sends without any request from the host when the actual process value has changed, but at no less than the set Expected Packet Rate, which ensures a reduction in the bus load. Attribute 20 (COS/Delta) of the respective instance in the Position Sensor Object should be set.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4B]						
2	Class ID [0x03 DeviceNet]							
3	Instance ID [01 Explicit Message]							
4	Allocation Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explicit
5	[00]		Allocator MAC ID					

Allocate COS

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4B	0x03	0x01	0x51	0x00		

Set Expected_Packet_Rate of the explicit message to 0:

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FC	0x00	0x10	0x05	0x01	0x09	0x00	0x00	

Set Expected_Packet_Rate of the COS:

To receive a message only when the value changes by the specified delta, set this to 0. If you also want to receive a position message every 1000 ms when there is no change, set this value to 1000 (0x3E8) (bytes 5 and 6).

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FC	0x00	0x10	0x05	0x04	0x09	0xE8	0x03	

6 Configuration of the node (cont.)

Release COS

Ends a no longer needed COS connection. The BTL stops sending data.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4C]						
2	Class ID [0x03 DeviceNet]							
3	Instance ID [01 Explicit Message]							
4	Release Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explicit

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4C	0x03	0x01	0x50			

Release explicit messaging

Ends the explicit connection.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Frag[0]	XID	MacID					
1	R/R[0]	Service [4C]						
2	Class ID [0x03 DeviceNet]							
3	Instance ID [01 Explicit Message]							
4	Release Choice							
		AckSp	Cyclic	COS	MCP	BitStrobe	Poll	Explicit

Sample message

CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x5FE	0x00	0x4C	0x03	0x01	0x01			

6.3 Number of magnets

Up to four magnets can be used. Each magnet has an instance of the Position Sensor Object allocated to it. The default configuration for the BTL is for one magnet.

Attribute 3 in the Balluff Configuration Object can be used to set the number of magnets. If fewer magnets are detected than configured here, then all position values are set to the error value (0x7FFFFFFF) and the associated alarms are activated.

In Flexible Magnet Mode the number of magnets can be varied. Do this by setting Attribute 3 of the Balluff Configuration Object to 0. The instances are then

allocated to the magnets in the order they are detected - as seen from the connector end.

6.3.1 Description of FMM-Mode

If for example you have four magnets (A, B, C and D as seen from the connector end), then Magnet A is allocated to Instance 1, B to Instance 2, etc.

Now if you remove Magnet C, first an error is detected for all instances and this error is reported. How long the error is displayed is set using Attribute 15 (Flexible Magnet Mode Delay) in Class 100 (Balluff Configuration Object) in milliseconds.

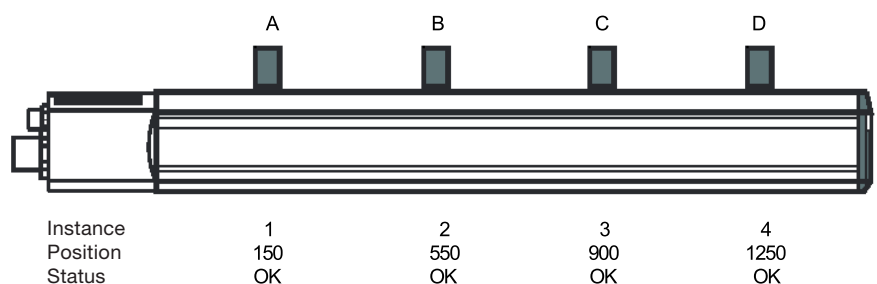


Fig. 6-1: BTL with 4 magnets

6 Configuration of the node (cont.)

During this time the error value is output as the position value. The associated error flags are set.

After the set time has elapsed, the magnets are again allocated in the order in which they are detected. In our example only three magnets are still detected. These are allocated to Instances 1...3. The missing 4th magnet is allocated to Instance 4, which retains the error value.

If in the various Instances you are working with various settings (e.g. Position Format, Offsets or CAMs), then you must take the possible shifting of the magnets into account in Flexible Magnet Mode.

6.4 Position data

The position data are affected by the following attributes in the Position Sensor Object.

6.4.1 Attribute 10: Position Value Signed

The position value is sent as a 32-bit value. It is influenced by Attributes 12, 14, 15, 18, 19 and 51. The internal resolution of the position sensor (counts) is 5 µm, although nm can also be set as an output format. Please note that the value range for 4 bytes in nm only ranges from -2.14 m to 2.14 m.

This attribute can also be transmitted to the I/O Message (➔ Section 2.1 Configuration of the I/O message).

6.4.2 Attribute 11: Position Sensor Type

This attribute returns the value 8 for 'Absolute Linear Encoder'.

6.4.3 Attribute 12: Direction Counting Toggle

The default setting is for the zero point of the position at the connector end. The positions are counted up as the magnet moves away from the connector. If Attribute 12 is set to TRUE, then the count direction is reversed. The position value which was previously assigned to the beginning of the measuring range is shifted by an amount equal to

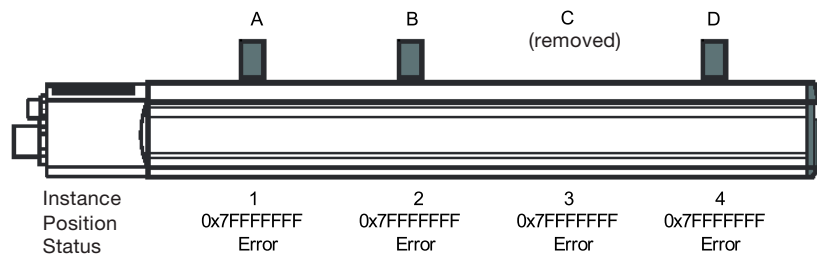


Fig. 6-2: 1 magnet removed

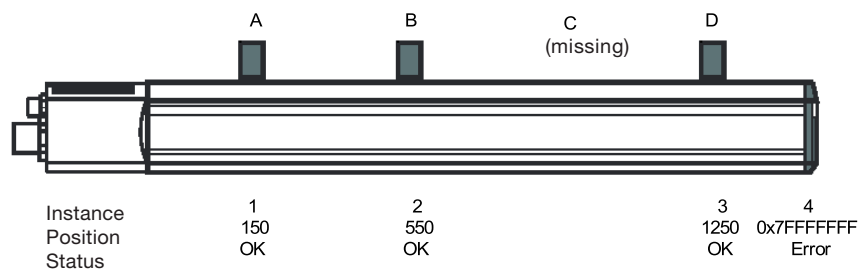


Fig. 6-3: Magnets with new allocation

the nominal stroke length to the end of the BTL after setting this attribute.

6.4.4 Attribute 14: Scaling Function Control

This attribute is set to TRUE (On) by default. This means that the position values are scaled according to Attributes 15, 16 and 51.

If this attribute is set to FALSE (Off), all functions of the Position Sensor Object continue to work with their scaled values in the set parameters. Only the position value is output in raw format (counts). This function is used for troubleshooting and calibration.

6.4.5 Attribute 15: Position Format

Attributes 10 (Position Value Signed), 19 (Preset Value), 22 (Position Low Limit), 23 (Position High Limit), 38 (CAM Low Limit), 39 (CAM High Limit), 40 (CAM Hysteresis) and 51 (Offset Value) are influenced by this attribute. The in-

and output of these attributes is always in the currently selected Position Format.

The above attributes are obtained with an accuracy of 5 µm and converted into the selected position format when they are input and output. This means the entered positions are not affected when there is a change in the Position Format, nor is the accuracy with which they are set.

After a change of the Position Format attribute, you should check Attribute 18 (Position Measuring Increment).

6 Configuration of the node (cont.)

6.4.5.1 Table of valid formats for position values

Unit	Sym- bol	Engunit (Hex)	Base unit
Centimeters	cm	0x2202	10 ⁻² m
Millimeters	mm	0x2203	10 ⁻³ m
Micrometers (Microns)*	μm	0x2204	10 ⁻⁶ m
Nanometers* **	nm	0x2205	10 ⁻⁹ m
Inches	in	0x2207	2,54*10 ⁻² m
Feet	ft	0x2208	3,048*10 ⁻¹ m
Points (Computer)		0x220F	1/72 in
Points (Printer)		0x2210	3.54598 * 10 ⁻⁴ m
Counts (5μm)		0x1001	5*10 ⁻⁶ m

* Please note that the units micrometers and nanometers are less than the minimum resolution of the BTL. Therefore the smallest increment is 5 μm, or 5000 nm.

** Since the position value can only be represented as a 32-bit value, the value range in nm is only up to 2.14 m. This setting is therefore impractical for longer BTLs and can result in an overflow of the position data.

6.4.6 Attribute 18: Position Measuring Increment

Sets the smallest increment of detectable position changes. This parameter depends on Attribute 15 (Position Format). Therefore you should set Attribute 15 first.

6.4.7 Attribute 19: Preset Value

This attribute can be used to allocate a desired position value to the actual position value. If Attribute 19 is set, the following will happen:

1. Attribute 19 is set to the entered value,
2. Offset Value (Attribute 51) is set to the difference between Preset Value and the actual position value.
3. After this the output position value (Attribute 10) is calculated as follows:

Output position value = (internal position value) + Offset Value (Attribute 51)

Note that the entered value depends on the setting for Attribute 15 (Position Format). Therefore you should set Attribute 15 first.

The value which is in Preset Value (Attribute 19) is, after it has been set, without meaning for further operation. The Offset Value attribute (Attribute 15) remains constant and from now on determines the output position value.

Attribute 51 can not be directly set as shipped from the factory, but rather only by using the Set Attribute access to Attribute 19 (see also ➡ Section 7.2.2 Attribute 14: Direct Offset Enable).

The internal memory format only allows a maximum value range of from -21 meters to +21 meters. Be sure to set the Preset Value so that these limits are not exceeded.

6.4.8 Attribute 20: COS Delta

If the position value (Attribute 10) changes by the value set here (or more), then a COS I/O message is generated. If this attribute is set to 0, this threshold is deactivated and every position change causes a COS message to be generated.

Attribute 20 is stored as a pure numeric value. If Attribute 15 (Position Format) is set to millimeters, and you set Attribute COS Delta to 30, then a position change of 30 millimeters generates a COS message. If you now set Attribute 15 to centimeters, then a COS message is sent only after a change in the position value of 30 cm.

This attribute only affects the processing of the associated position value. Changes in other values in the Input Assembly continue to result in immediate generation of a COS message.

6.4.9 Attributes 21 to 23: Position State Register and Position Limits

The user-defined working range is defined by Attributes 22 (Position Low Limit) and 23 (Position High Limit). The setting of both attributes depends on the currently used unit (Attribute 15: Position Format). Therefore you should set Attribute 15 before these attributes.

If the actual position is greater than or equal to Attribute 22 and less than or equal to Attribute 23, then all bits in the Position State Register (Attribute 21) are deleted.

If the actual position is less than Attribute 22, then Bit 0 (Out of Range) and Bit 2 (Range Underflow) is set. If the actual position is greater than Attribute 23, then Bit 0 (Out of Range) and Bit 1 (Range Overflow) is set.

Attribute 21: Position State Register								
Bit	7	6	5	4	3	2	1	0
Position	-	-	-	-	-	Range Underflow	Range Overflow	Out of Range

Table 6-1: Attribute 21

6 Configuration of the node (cont.)

6.4.10 Working ranges

Each magnet has an instance of the Position Sensor Object allocated to it which can be parameterized independently of the other instances.

6.4.11 CAMs/Switchpoints

Each instance provides four mutually independent CAMs. The CAM channels are determined in their position by the attributes

- Low Limit Attribute 38: CAM Low Limit
- High Limit Attribute 39: CAM High Limit
- Hysteresis Attribute 40: CAM Hysteresis

The default setting for all these attributes is 0. A CAM channel becomes active when the position value is greater than the set CAM Low Limit and at the same time less than the set CAM High Limit.

You can also use the CAM channels as threshold switches by positioning one of the limits outside the measuring range.

The following attributes of the Position Sensor Object influence the behavior of the CAMs:

6.4.12 Attribute 34: Number of CAM Channels

The BTL has four freely programmable CAM channels per instance (transducer). Therefore this attribute returns the value 4.

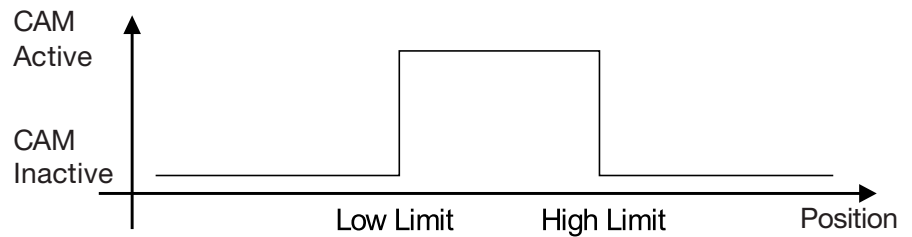


Fig. 6-4: Activate CAMs

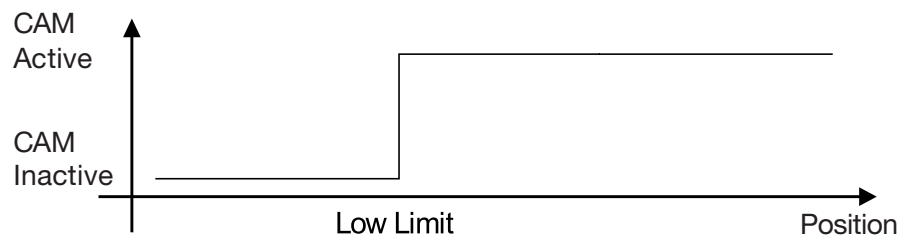


Fig. 6-5: CAMs- Low Limit

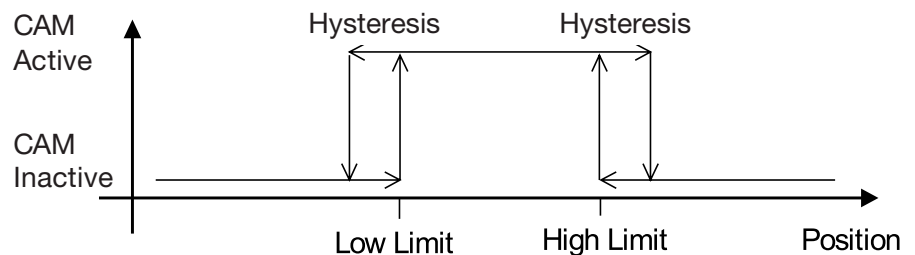


Fig. 6-6: CAMs - Hysteresis

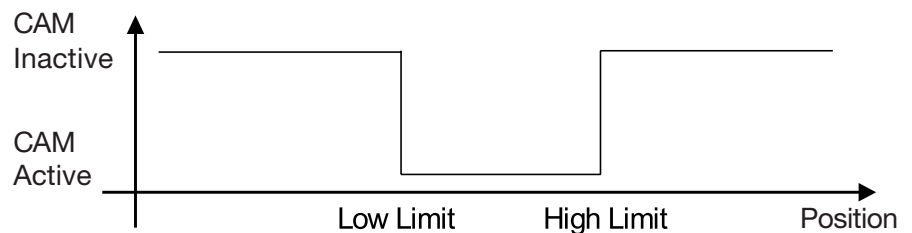


Fig. 6-7: Toggle CAMs

6 Configuration of the node (cont.)

6.4.13 Attribute 35: CAM Channel State Register

Each of the four CAM channels has a bit allocated to it in this register. This attribute can also be sent to the I/O message (➔ Section 2.1 Configuration of I/O messages).

CAM Channel State Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not used	Not used	Not used	Not used	CAM4	CAM3	CAM2	CAM1
0	0	0	0	State	State	State	State

Table 6-2: Attribute 35

6.4.14 Attribute 36: CAM Channel Polarity Register

Each of the four CAM channels has a bit allocated to it in this register. If this bit is cancelled (FALSE), then when the CAM is activated the associated bit in Attribute 35 CAM Channel State Register is set to TRUE (1).

If a bit is set in the CAM Channel Polarity Register (TRUE), then the associated bit in Attribute 35 CAM Channel State Register is cancelled when the associated CAM is activated, in other words it takes on the FALSE (0) state.

CAM Channel Polarity Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not used	Not used	Not used	Not used	CAM4	CAM3	CAM2	CAM1
0	0	0	0	Polarity inverse	Polarity inverse	Polarity inverse	Polarity inverse

Table 6-3: Attribute 36

6.4.15 Attribute 37: CAM Channel Enable Register

Analogous to Attributes 35 and 36, here each bit is allocated to a CAM channel. If this bit is cancelled (FALSE), then the associated CAM is deactivated. The associated bit in the CAM Channel State Register remains cancelled (FALSE).

If the bit is set, then the associated CAM is active and behaves as specified in Registers 35 and 36.

CAM Channel Enable Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not used	Not used	Not used	Not used	CAM4	CAM3	CAM2	CAM1
0	0	0	0	Active	Active	Active	Active

Table 6-4: Attribute 37

6.4.16 Attribute 38: CAM Low Limit

Every instance of the Position Sensor Object (each of which is allocated to a transducer) has four possible CAMs. Each CAM has a Low Limit and a High Limit assigned to it.

Attribute 38 stores the Low Limits for all four CAMs of an instance. The four CAMs are passed as an array, i.e. four positions with four bytes each. In total this attribute thus has 16 bytes.

CAM Low/High Limit																
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	LSB			MSB	LSB			MSB	LSB			MSB	LSB			MSB
Channel	CAM1				CAM2				CAM3				CAM4			

Table 6-5: Attribute 38/39

6.4.17 Attribute 39: CAM High Limit

This attribute stores the High Limits for the four CAMs of an instance. For details and structure see Attribute 38: CAM Low Limit.

6 Configuration of the node (cont.)

6.4.18 Attribute 40: CAM Hysteresis

The hysteresis is added to the CAM High Limit and subtracted from the CAM Low Limit when the state of the CAMs is calculated. The graphic ➡ Fig. 6-6 shows this in detail.

This attribute is an array of UNIT. It contains two bytes for each CAM channel. This attribute therefore contains 8 bytes.

Attribute 40: CAM Hysteresis								
Byte	0	1	2	3	4	5	6	7
	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB
Channel	CAM1		CAM2		CAM3		CAM4	

Table 6-6: Attribute 40

6.5 Velocity settings

The velocity can be output in various units as selected in Attribute 25 (Velocity Format). The settings for various other attributes are referenced to the selected unit.

Therefore you should first set the desired unit and then the attributes which depend on it. If the attributes are set automatically, by using RSNetWorx for example, then you should then perform an upload and check the settings.

6.5.1 Resolution

The velocity is captured at a resolution of 0.1 mm/s and with an accuracy of 32 bits. The unit preset according to the CIP for velocity is the standard unit for the position (Counts = 5 µm) per second, i.e. 5 µm/s.

This unit however is less than the measuring resolution for the velocity and thus has a minimum increment of 20.

The resolution at which the velocity data are output can be set using the following attributes of the Position Sensor Object.

6.5.1.2 Attribute 24: Velocity Value

Actual velocity value depending on Attributes 25 and 26.

This attribute can also be sent to the I/O message (➡ Section 2.1 Configuration of the I/O messages).

6.5.1.3 Attribute 25: Velocity Format

This attribute contains the velocity unit. This unit affects Attributes 26 (Velocity Resolution), 27 (Minimum Velocity Setpoint) and 28 (Maximum Velocity Setpoint). Therefore you should set this attribute before Attributes 26, 27 and 28.

Unit	Sym- bol	Engunit (Hex)	Base unit
Steps per second*	CPS	0x1F04	5 µm/s
Meters per second	m/s	0x2B00	1 m/s
Centimeters per second	cm/s	0x2B01	10 ⁻² m/s
Kilometers per hour	km/h	0x2B02	(1000 / 3600) m/s
Miles per hour	mi/h	0x2B04	0.44704 m/s
Knots	kt	0x2B05	(1852 / 3600) m/s
Feet per second	ft/s	0x2B06	3.048*10 ⁻¹ m/s
Inches per second	in/s	0x2B07	2.54*10 ⁻² m/s

* Please note that the internal measuring resolution for velocity is 100 µm/s ist. This means the unit steps per second is less than this resolution and the smallest increment is 100 µm/s.

6.5.1.4 Attribute 26: Velocity Resolution

Specifies the smallest increment in the selected velocity format in which the velocity value is output.

6.5.1.5 Attribute 27: Minimum Velocity Setpoint

In this attribute you set a Low Limit for the velocity. If the velocity is equal or less than this value, then the Minimum Velocity Flag in Attribute 47 (Warnings) is set.

6.5.1.6 Attribute 28: Maximum Velocity Setpoint

Analogous to Attribute 27, here the High Limit for velocity is set. If the velocity is greater than or equal than this value, then the Maximum Velocity Flag in Attribute 47 (Warnings) is set.

6.5.2 Setpoints

Attributes 27 (Minimum Velocity Setpoint) and 28 (Maximum Velocity Setpoint) are associated with two bits in Attribute 47 (Warnings). If the velocity reaches or falls below the Minimum Velocity Setpoint, then Bit 6 (Minimum Velocity Flag) in the Warnings attribute is set. If the velocity reaches or exceeds the Maximum Velocity Setpoint, then the Maximum Velocity Flag (Bit 7) in the Warnings attribute is set. The bits are cancelled when the velocity again lies between Attribute 27 and 28.

7 Additional attributes

7.1 Position Sensor Object

7.1.1 Attribute 41: Operating Status

This attribute contains information about various states of the encoder.

Bit 0 indicates the count direction. If this bit is 0 (FALSE), then the count direction is positive, i.e. the position values increase as the magnet moves away from the connector end. If this bit is set (TRUE), then the count direction is the reverse.

Bit 1 indicates whether scaling is active. If this bit is cancelled, then scaling is deactivated. In this case the BTL outputs raw values. If this bit is set, then scaling is activated.

7.1.2 Attribute 44: Alarms

The Alarms attribute indicates whether a fault has occurred which affects determination of the position value. The alarm remains active until the error has been eliminated and the device is again providing a valid position value.

The Flag Position Error is set when a position determination is not possible. This can happen for example if the associated magnet is no longer within the measuring range. In Flexible Magnet Mode all positions are invalid for a defined time if the number of magnets changes.

The Flag Temperature Error is only active if a temperature sensor is fitted (A/B/Z rod-style only). It indicates that the BTL is being operated outside its permissible temperature range.

This attribute can also be sent to the I/O message (➔ Section 2.1 Configuration of the I/O messages).

7.1.3 Attribute 45: Supported Alarms

A bit is set in this attribute for each bit supported in Attribute 40. The value is therefore always a constant 0x1001.

Attribute 41: Operating Status																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Status	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Scaling	Direction

Table 7-1: Attribute 41

Attribute 44: Alarms																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alarms				Tem- pera- ture												Position Error

Table 7-2: Attribute 44

Attribute 45: Supported Alarms																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alarms				Temperature												Position
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Table 7-3: Attribute 45

Attribute 47: Warnings																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Warn-ings						Position Limits			Maximum Velocity	Minimum Velocity						

Table 7-4: Attribute 47

7.1.4 Attribute 46: Alarm Flag

If one of the flags in Attribute 44 is set, then Attribute 46 is set to 1 (TRUE). If all flags in Attribute 44 are cancelled, then Attribute 46 has the value 0 (FALSE).

7.1.5 Attribute 47: Warnings

The Warnings attribute indicates that the tolerance of various internal parameters was exceeded. In contrast to the Alarms attribute 44, here invalid position values are not taken into account. The warnings are cancelled when the internal parameters resume their proper range.

The Minimum Velocity flag is set when the actual velocity is less than the Minimum Velocity Set-point value set in Attribute 27. Likewise, the Maximum Velocity flag indicates that the actual velocity is greater than the value set in Attribute 28.

The Position Limits flag is set when the actual position is less than the Position Low Limit set in Attribute 22 or greater than the Position High Limit set in Attribute 23.

This attribute can also be sent to the I/O message (➔ Section 2.1 Configuration of the I/O messages).

7.1.6 Attribute 48: Supported Warnings

A bit is set in this attribute for each bit supported in Attribute 47. The value is therefore always a constant 0x4C0.

7.1.7 Attribute 49: Warning Flag

When one of the flags in Attribute 47 is set, then Attribute 49 is set to 1 (TRUE). If all flags in Attribute 47 are cancelled, then Attribute 49 has the value 0 (FALSE).

7 Additional attributes (cont.)

7.1.8 Reset Service - Position Sensor Object

To reset the Position Sensor Object back to its default settings, a Reset Service is implemented. The service is invoked using Argument 1. Details can be found in ➡ Section 4.4 Step 2c: Invoking a service.

Only the specified instance is reset. The rest of the instances remain unchanged. To reset all instances, the Reset Service must be invoked for each individual instance.

7.2 Balluff Configuration Object

7.2.1 Attributes 11, 12 and 13: Temperature (A/B/Z rod-style only)

These attributes are used to store the actual temperature (Attribute 11), the previous lowest operating temperature (Attribute 13) and the previous highest operating temperature (Attribute 12). The temperature values are updated every 6 minutes if the temperature sensor is fitted.

7.2.2 Attribute 14: Direct Offset Enable

In accordance with CIP specification Attribute 51 of the Position Sensor Object (Offset Value) cannot be set directly. Rather it is calculated if Attribute 19 is set. This presumes that the magnet for setting the Preset Value is placed at a defined point.

This attribute Direct Offset Enable has a default value of False. This means that the Position Sensor Object behaves as describe above. A write access to Attribute 51 causes an error message to be returned.

If you set Direct Offset Enable to True, then it is possible to make write access to Attribute 51, i.e., you can then enter the offset directly.

Attribute 48: Supported Warnings																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Warnings						Position Limits			Maximum Velocity	Minimum Velocity						
Default	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0

Table 7-5: Attribute 48

7.2.3 Attribute 16: Sync Period

If the transducer is synchronized using Bit Strobe, then it sends its data after receiving each Bit Strobe command. The sensor attempts to synchronize the start of its measurement so that a new measurement value is gotten in time for the expected next Bit Strobe command. This means that this attribute must be set to the expected time interval between the individual Bit Strobe commands.

7.2.4 Reset Service - Balluff Configuration Object

To reset all attributes of the Balluff Configuration Object to their default settings, a Reset Service is implemented. The service is invoked using Argument 1. For details, see ➡ Section 4.4 Step 2c: Invoking a service.


8 Startup

8.1 Setup using RSNetWorx

8.1.1 EDS file

The EDS file, which can be found in the Internet at www.balluff.de, contains information about device-specific parameters as well as possible operating modes for the BTL. An electronic data sheet is available which can be used for setup using RSNetWorx for example.

Before the BTL can be connected to the bus, the EDS file must be installed using the EDS Wizard. To start the Wizard, click in the menu bar on 'Tools/EDS Wizard'. The window shown in ➡ Fig. 8-1 appears.

To install an EDS file, you must first click on 'Register an EDS file(s)' and then . In the next step click on 'Register a directory of EDS files' and use the browser to specify the path of the EDS file ➡ Fig. 8-2.

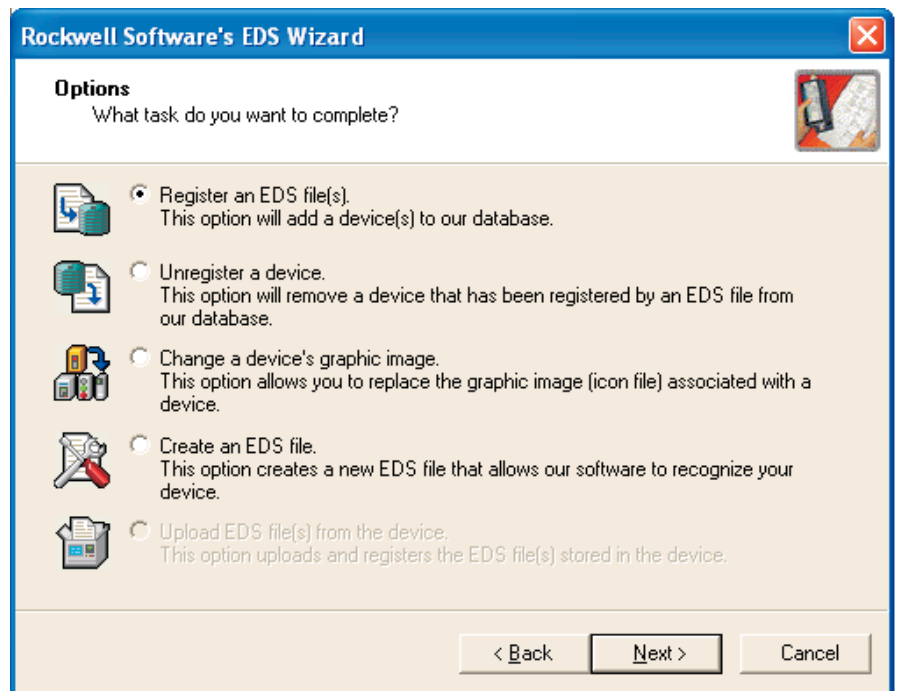


Fig. 8-1: Start menu of the EDS - Wizard

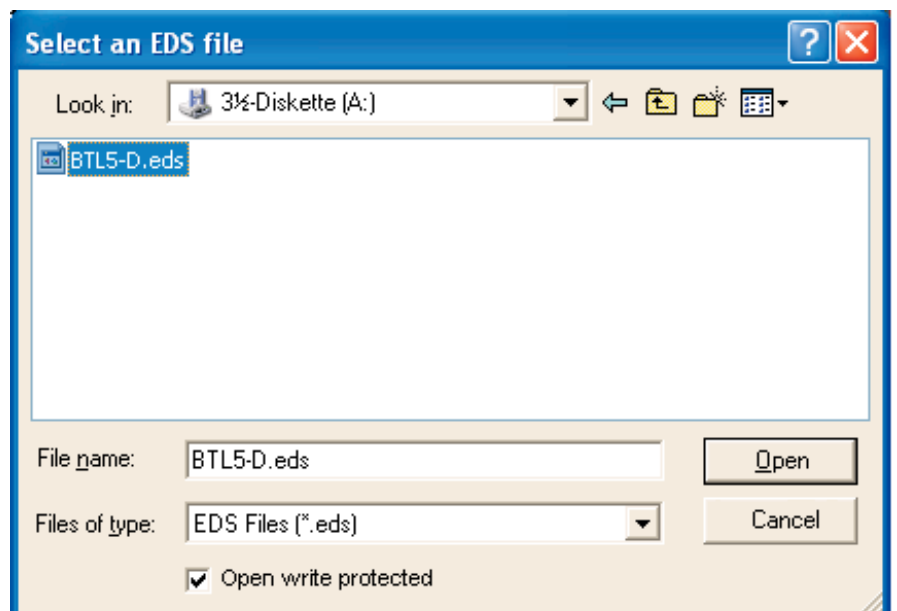


Fig. 8-2: Selecting the EDS file

8 Startup (cont.)

The Wizard finds all EDS files which are stored in this path and after opening carries out a test to see whether the EDS file contains any errors ➔ Fig. 8-3.

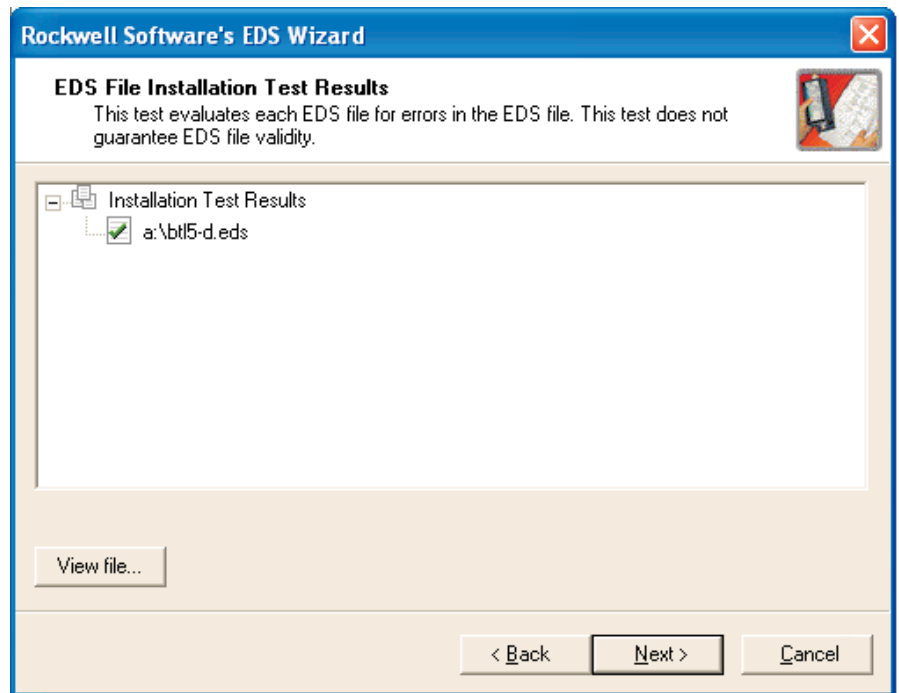




Fig. 8-3: Test result of the EDS file

Now if you click on the  button you are taken to the window 'Change Graphic Image' (➔ Fig. 8-4: Selecting the graphic image) where you can associate graphics with the nodes used. Click on  to continue the installation and finish it.

Then a message appears confirming successful installation of the EDS file.

Now if the controller finds the device in the network, it will be correctly recognized and represented.

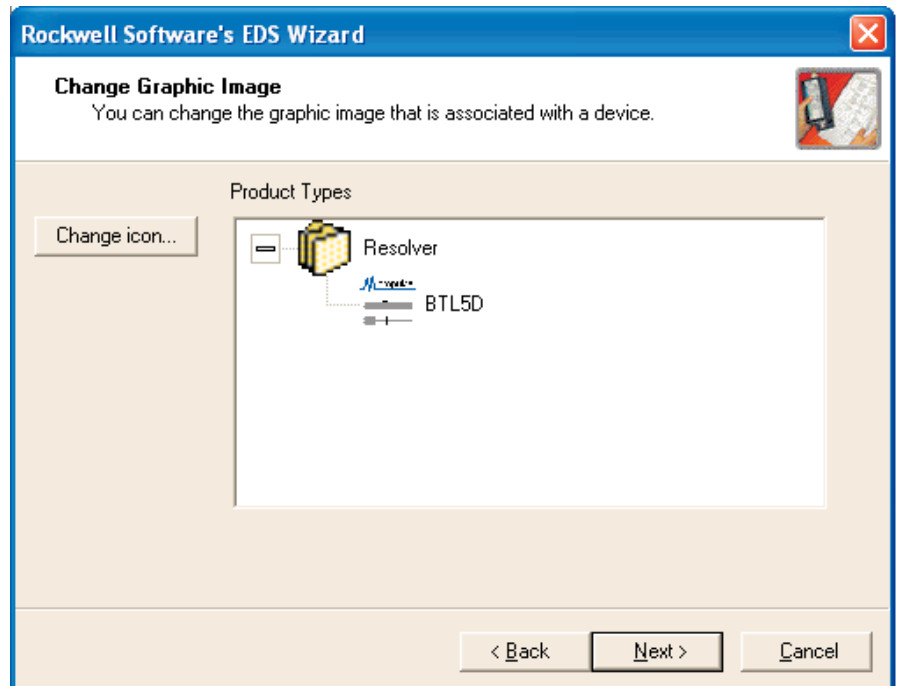




Fig. 8-4: Selecting the graphic

8 Startup (cont.)

To parameterize the BTL, click on the BTL symbol in online mode and open the Configuration window from Device/Properties or in the menu bar using the menu that appears after right-clicking. Then use **Parameters** to upload the parameters.

After successfully uploading the parameters you can configure them ➡ Fig. 8-7.

To download the configured attributes, use the yellow arrow pointing down in the upper right of the window  - and to upload use the arrow pointing up to the left of the Download button .

Changed attributes are loaded directly into the BTL after clicking on the **Apply** button.

The attributes are automatically saved in the EEPROM after a download. The BTL is then started up with these data after power to the BTL has been turned off. If you do not want the data permanently stored, first set Attribute 4 in the Balluff Configuration Object (ID 504) and apply this value. Then the BTL will be restarted with the last saved settings.

You can continuously read out the current value of an attribute by clicking on the **Monitor** button. If you select 'Single', then the attribute you have clicked on will be cyclically polled. If you select 'All', then all shown attributes are updated in order. An arrow at the left side of the list indicates the respective updated value.

The instance and attribute number can be seen from the ID. IDs 101...151 refer to Instance 1 and are associated with the first magnet. Likewise 201 ... 251 refer to Instance 2 for Magnet 2, 301 ... 351 Instance 3 for Magnet 3 and 401 ... 451 Instance 4 for Magnet 4.

The IDs 501 ... 517 are allocated to the Balluff Configuration Object. Its attributes are global.

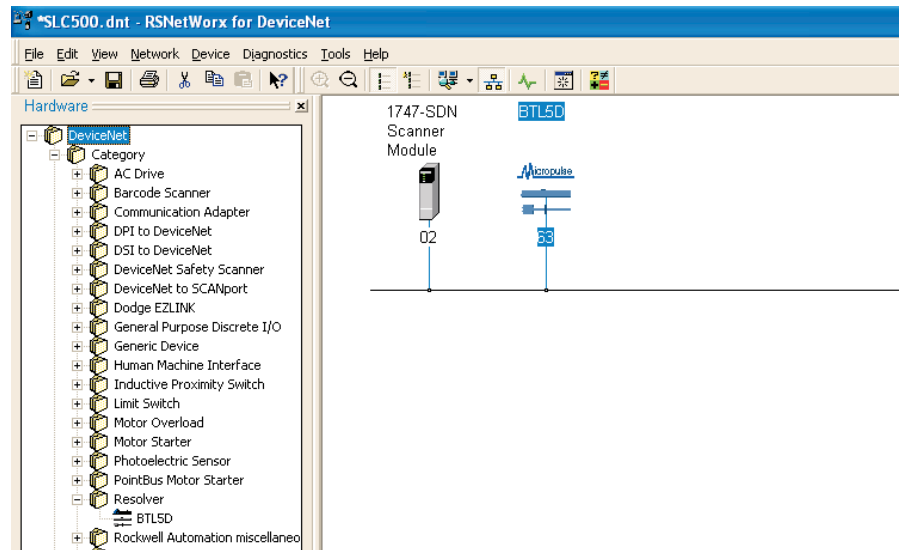


Fig. 8-5: Representation of the BTL in the network

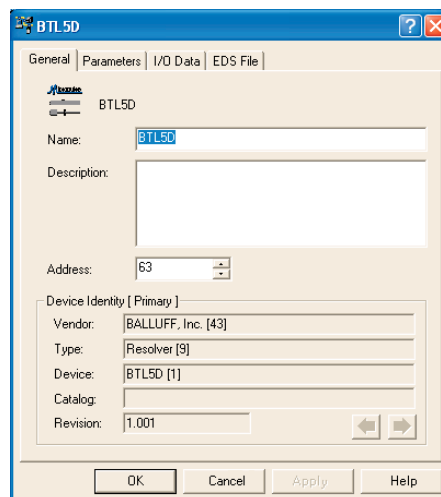


Fig. 8-6: General BTL information

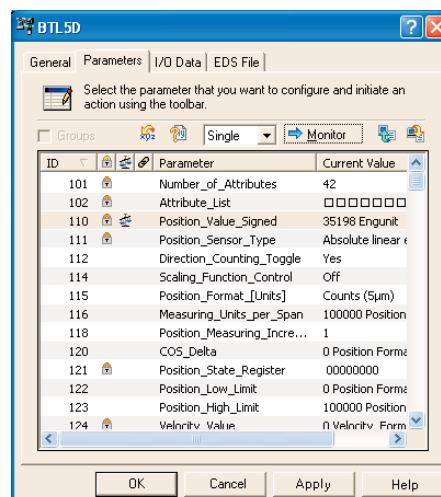


Fig. 8-7: Parameters

9 Class directory (Object directory)

The class directory for the transducer is divided into six areas which are shown here in table form.

The data (position, velocity) as well as the configuration parameters are stored in the class of the Position Sensor Object.

The abbreviations below are used in the following sections:

R/W Read/Write: Attribute is read and write accessible

R Read only: Attribute can only be read

ROM Read Only Memory: Attribute is constant and cannot be changed

NV Non Volatile: Attribute is stored in non-volatile memory

V Volatile: Attribute is not stored.

	Name	Number of bytes	Minimum value	Maximum value
BOOL	Boolean	1	0 (FALSE)	1 (TRUE)
SINT	Short Integer	1	-128	127
INT	Integer	2	-32768	32767
DINT	Double Integer	4	-2 ³¹	2 ³¹ -1
USINT	Unsigned Short Integer	1	0	255
UINT	Unsigned Integer	2	0	65535
UDINT	Unsigned Double Integer	4	0	2 ³² -1
BYTE	bit string - 8 bits	1		
WORD	bit string - 16-bits	2		
DWORD	bit string - 32-bits	4		
ENGUNITS	Engineering Units	2		See following table

Table 9-1: Data types

1F04 counts per second	cps	7940	0x1F04
2B00 meter per second	m/s	11008	0x2B00
2B01 centimeter per second	cm/s	11009	0x2B01
2B02 kilometer per hour	km/h	11010	0x2B02
2B04 mile per hour	mi/h	11012	0x2B04
2B05 knot (nautical mile per hour)	kt	11013	0x2B05
2B06 foot per second	ft/s	11014	0x2B06
2B07 inch per second	in/s	11015	0x2B07

Table 9-2: Engineering units for velocity

1001 counts		4097	0x1001
2202 centimeter	cm	8706	0x2202
2203 millimeter	mm	8707	0x2203
2204 micron (micrometer)	μ	8708	0x2204
2207 inch	in	8711	0x2207
2208 foot	ft	8712	0x2208
220F point (computer)	(1/72) in	8719	0x220F
2210 point (printers)	3.514598 * 10 ⁻⁴ m	8720	0x2210

Table 9-3: Engineering units for position

9 Class directory (Object directory, cont.)**9.1 Identity Class (ID = 1)**

Cl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
1	1	1	Vendor ID	R	ROM	UINT	0x2B	Manufacturer's ID 0x2B: BALLUFF
1	1	2	Device Type	R	ROM	UINT	0x09	Product type ID 0x09: Resolver Device
1	1	3	Product Code	R	ROM	UINT		Product identification 0x01: Profile –P 0x02: Rod –A 0x03: Rod –B 0x04: Rod –Z
1	1	4	Revision	R	ROM			Software revision number
1	1		Major Revision		ROM	USINT		Major revision
1	1		Minor Revision		ROM	USINT		Minor revision
1	1	5	Status	R	NV	WORD	0	Device status bit set Bit 2: configured; every other status such as "out-of-box" Bit 8: minor recoverable fault; device does not go into fault state Bit 9: minor non-recoverable fault; device does not go into one of the fault states Bit 10: recoverable fault; device changes to "Major Recoverable Fault" state Bit 12: non-recoverable fault; device changes to "Major Unrecoverable Fault" state
1	1	6	Serial Number	R	ROM	UDINT		Serial number
1	1	7	Product Name	R	ROM	Short_String	BTL5D	Product name
1	1	10	Heartbeat Interval	R/W	V	USINT	0	Time between two Heartbeat messages in seconds. 0 turns the Heartbeat message off.

Table 9-4: Identity Class

9 Class directory (Object directory, cont.)**9.2 DeviceNet Class (ID = 3)**

Cl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
3	n	1	MAC ID	R/W	NV	USINT	0x3F	Unique device address in the network
3	n	2	Baud Rate	R/W	NV	USINT	0x00	Baud rate 0x00: 125 kBaud 0x01: 250 kBaud 0x02: 500 kBaud
3	n	3	BOI (Bus-Off Interrupt)	R		BOOL	FALSE	FALSE: The CAN Chip remains in Reset TRUE: Resume communication after reset
3	n	4	Bus-Off Counter	R/W		USINT	0	Counts how often the device changes to Bus-off state. Starts at 0 after power-on. Counter is reset by a write access.
3	n	5	Allocation Information	R		STRUCT		Contains information about which of the predefined Master/Slave connections are active and the Master MAC ID.
			Allocation Choice Byte				0	Bit 7: Reserved Bit 6: Acknowledge Suppression Bit 5: Cyclic Bit 4: Change of State Bit 3: Multicast Polling Bit 2: Bit Strobed Bit 1: Polled Bit 0: Explicit Message
			Master's MAC ID				0	The MAC ID, which for example may trigger a Bit Strobe.

Table 9-5: DeviceNet Class

9.3 Assembly Class (ID = 4)

The I/O data can be called again in the Assembly Object. The number and content of the data can be configured as described in Section 2.1 "Configuration of the I/O messages".

Cl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
4	100	3	Data	R		ARRAY of BYTE		Position and CAM of the 1st magnet

Table 9-6: Assembly Class

9 Class directory (Object directory, cont.)

9.4 Connection Object

Cl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
5	X	1	State	R		USINT		Object status 00: Non-existent 01: Being configured 02: Wait for connection ID 03: Connection established 04: Time exceeded
5	X	2	Instance Type	R		USINT		Instance type: 00: Explicite Message connection 01: I/O connection
5	X	3	Transport Class Trigger	R		BYTE		Determines connection behavior Bit 7: Direction 0: Client; 1: Server Bit 6 ... 4: Production 0: Cyclic 1: Change-Of-State 2: Application Object Bit 3 ... 0: Transport class 0: Class 0 1: Reserved / not used 2: Class 2 3: Class 3
5	X	4	Produced Connection ID	R		UINT		Copied into the CAN Identifier field if sending over the connection.
5	X	5	Consumed Connection ID	R		UINT		CAN Identifier value which identifies the message received over a connection.
5	X	6	Initial Comm Characteristics	R		BYTE		Defines the message group(s) through which the produced and consumed data referenced to this connection appear.
5	X	7	Produced Connection Size	R/W		UINT		Maximum number of bytes which can be sent over this connection.
5	X	8	Consumed Connection Size	R		UINT		Maximum number of bytes which can be received over this connection.
5	X	9	Expected Packet Rate	R/W		UINT		Interval between cyclic messages.
5	X	12	Watchdog Timeout Action	R		USINT		Defines the response to inactivity or watchdog timeouts.
5	X	13	Produced Conn. Path Length	R		UINT		Number of bytes in the Product Connection Path attribute.
5	X	14	Produced Connection Path	R		PACKED		Specifies the application object whose data are exported to this Connection Object.
5	X	15	Consumed Conn. Path Length	R		UINT		Number of bytes in the Consumed Connection Path attribute
5	X	16	Consumed Connection Path	R		PACKED		Specifies the application object whose data are imported from this connection object.
5	X	17	Production Inhibit Time	R/W		UINT		Defines the minimum time for data production.

Table 9-7: Connection Object

X = 1: Explicit message
X = 2: Polled I/O
X = 3: Bit Strobed I/O

X = 4: COS/Cyclic
X = 5: Multicast Poll I/O

9 Class directory (Object directory, cont.)

9.5 Position Sensor Object

Cl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
35	1..4	1	Number of Attributes	R	ROM	USINT		Number of attributes supported in this object.
35	1..4	2	Attribute List	R	ROM	STRING		Array of USINT: List of the attributes supported in this object.
35	1..4	10	Position Value Signed	R	V	DINT		Actual position taking into account attributes 14,15,16 and 17.
35	1..4	11	Position Sensor Type	R	ROM	UINT	8	Device type (Absolute Linear Encoder)
35	1..4	12	Direction Counting Toggle	R/W	NV	BOOL	FALSE	Defines the positive count direction of the position value.
35	1..4	14	Scaling Function Control	R/W	NV	BOOL	TRUE	True (default): Data are converted according to the other attributes. False: Position data are output in raw format.
35	1..4	15	Position Format	R/W	NV	ENGUNIT		Unit for the position values
35	1..4	16	Measuring units per span	R/W	NV	UDINT		Number of steps over the nominal length. Equal to or less than attribute 42.
35	1..4	18	Position Measuring Increment	R/W	NV	UDINT		Specifies the smallest step size for attribute 10.
35	1..4	19	Preset Value	R/W	NV	DINT		The output position value is set to the specified value.
35	1..4	20	COS / delta	R/W	NV	UDINT		Increment which, when exceeded, causes a Change-of-State message.
35	1..4	21	Position State Register	R	V	BYTE		Monitoring of the position limits (Attributes 22 and 23).
35	1..4	22	Position Low Limit	R/W	NV	DINT		Lower position limit. If attribute 10 is less than this value, then Bit 0 in attribute 21 (Position State Register) is set.
35	1..4	23	Positon High Limit	R/W	NV	DINT		Upper position limit. If attribute 10 is greater than this value, then Bit 1 in attribute 21 (Position State Register) is set.
35	1..4	24	Velocity Value	R	V	DINT		Actual velocity. Format is defined in attributes 25 and 26.
35	1..4	25	Velocity Format	R/W	NV	ENGUNIT	0x1F04	Velocity format. Default: 0x1F04 = counts (steps) per second
35	1..4	26	Velocity Resolution	R/W	NV	UDINT		Smallest increment of attribute 24. The unit depends on attribute 25.
35	1..4	27	Minimum Velocity Setpoint	R/W	NV	DINT		Value for the lower velocity threshold. Affects the Bit Min. Velocity Flag in attribute 47.
35	1..4	28	Maxmum Velocity Setpoint	R/W	NV	DINT		Value for the upper velocity threshold. Affects the Bit Max. Velocity Flag in attribute 47.

Table 9-8: Position Sensor Object

BTL5-D11.. Configuration and CAN coupling – CIP Resolver Device

9 Class directory (Object directory, cont.)

Cl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
35	1..4	34	Number of CAM Channels	R	NV	USINT	4	Number of independent CAM channels
35	1..4	35	CAM Channel State Register	R	NV	ARRAY		Array of BOOL: Contains the state of the CAM channels (Bit 0 = CAM_1, Bit1 = CAM: _2,...)
35	1..4	36	CAM Channel Polarity Register	R/W	NV	ARRAY		Array of BOOL: Contains the polarity of the CAM channels (Bit 0 = CAM_1 Polarity, Bit1 = CAM_2,...)
35	1..4	37	CAM Channel Enable register	R/W	NV	ARRAY		Array of BOOL: Activates the individual CAM channels (Bit 0 = CAM_1, Bit1 = CAM_2,...)
35	1..4	38	CAM Low Limit	R/W	NV	ARRAY		Array of DINT: Low switchpoints for the CAM channels.
35	1..4	39	CAM High Limit	R/W	NV	ARRAY		Array of DINT: High switchpoints for the CAM channels.
35	1..4	40	CAM Hysteresis	R/W	NV	ARRAY		Array of UINT: This value is added as hysteresis to the attribute CAM High Limit and subtracted from CAM Low Limit.
35	1..4	41	Operating Status	R	V	BYTE		Status of the transducer
35	1..4	42	Physical Resolution Span	R	NV	UDINT		Number of physically possible increments over the span.
35	1..4	44	Alarms	R	V	WORD		Indicates faults which have occurred which result in an invalid position value and require user intervention.
35	1..4	45	Supported Alarms	R	ROM	WORD		One bit for each supported bit in attribute 44.
35	1..4	46	Alarm Flag	R	V	BOOL		Indicates than an alarm is set in attribute 44. 0 = OK; 1 = Alarm Error
35	1..4	47	Warnings	R	V	WORD		If internal limits are exceeded, then the corresponding bits are set
35	1..4	48	Supported Warnings	R	ROM	WORD		One bit each for every supported bit in attribute 47.
35	1..4	49	Warning Flag	R	V	BOOL		Indicates that a warning is set in attribute 47. 0 = OK; 1 = Warning
35	1..4	50	Operating Time	R	NV	UDINT		Stores the operating time of the BTL in increments of 0.1 seconds.
35	1..4	51	Offset Value	R	NV	DINT		The offset value is calculated by setting the attribute 19 Preset Value.

Table 9-8: Position Sensor Object (cont.)

9 Class directory (Object directory, cont.)

9.6 Acknowledge Handler Object

The Acknowledge Handler Object manages the received message acknowledgements. This object

communicates with the objects generating messages in the BTL and manages message acknowledgements, exceeded times and maximum repeat attempts.

Kl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
43	1	1	Acknowledge Timer	R/W		BOOL	16	Time in ms waited for a confirmation before another attempt takes place.
43	1	2	Retry Limit	R		BOOL	1	Maximum number of times Acknowledge time can be exceeded.
43	1	3	COS Producing Connection Instance	R		BOOL	4	ID of the connection instance which contains the path of the producing I/O application object and which provides information about Acknowledge Handler events.

Table 9-9: Acknowledge Handler Object

9.7 Balluff Configuration Object

Cl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
100	1	1	Number of Attributes	R	ROM	USINT		Number of attributes available in this object.
100	1	2	Attribute List	R	ROM	STRING		Array of USINT: List of the attributes available in this object.
100	1	3	Number of Magnets	R/W	NV	USINT	1	Number of magnets (1..4; 0 = Flexible Magnet Mode)
100	1	4	Autosave	R/W	NV	BOOL	TRUE	TRUE: Autosave is active (default) FALSE: Autosave is deactivated
100	1	5	Assembly Options	R/W	NV	USINT	0	Bit field: Arrangement of the output parameters in the I/O
100	1	6	Assembly Position Active	R/W	NV	UINT	1	Bit field: One bit each for copying the position value to the I/O message.
100	1	7	Assembly Velocity Active	R/W	NV	UINT	1	Bit field: One bit each for copying the velocity to the I/O message.
100	1	8	Assembly CAM Active	R/W	NV	UINT	0	Bit field: One bit each for copying the CAM status to the I/O message.
100	1	9	Assembly Warnings Active	R/W	NV	UINT	0	Bit field: One bit each for copying the warning to the I/O message.
100	1	10	Assembly Alarms Active	R/W	NV	UINT	0	Bit field: One bit each for copying the alarms to the I/O message.
100	1	11	Temperature	R	V	SINT	-	Actual temperature °C
100	1	12	Temperature Maximum	R	NV	SINT	-	Maximum operating temperature in °C
100	1	13	Temperature Minimum	R	NV	SINT	-	Minimum operating temperature in °C

Table 9-10: Balluff Configuration Object

BTL5-D11.. Configuration and CAN coupling – CIP Resolver Device

9 Class directory (Object directory, cont.)

Cl.	Inst.	Attr.	Name	Access	Memory	Type	Default value	Description
100	1	14	Direct Offset Enable	R/W	NV	BOOL	FALSE	TRUE: Enables direct setting of Attribute 51 (Position Offset) in the Position Sensor Object. FALSE: Offset can only be set using Preset (default).
100	1	15	Flexible Magnet Mode Delay	R/W	NV	UINT	0	Alarm time for changes in Flexible Magnet Mode.
100	1	16	Sync Period	R/W	NV	UDINT	0	Experiend period duration of the BitStrobe messages in ms 0 = Deactivated.

Table 9-11: Balluff Configuration Object (cont.)

Attribute 6: Assembly Positions Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Position	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 9-12: Attribute 6

Attribute 7: Assembly Velocity Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 9-13: Attribute 7

Attribute 8: Assembly CAM Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAM	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 9-14: Attribute 8

Attribute 9: Assembly Warnings Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Warnings	-	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 9-15: Attribute 9

Attribute 10: Assembly Alarms Active																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alarms																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 9-16: Attribute 10

10 Additional settings

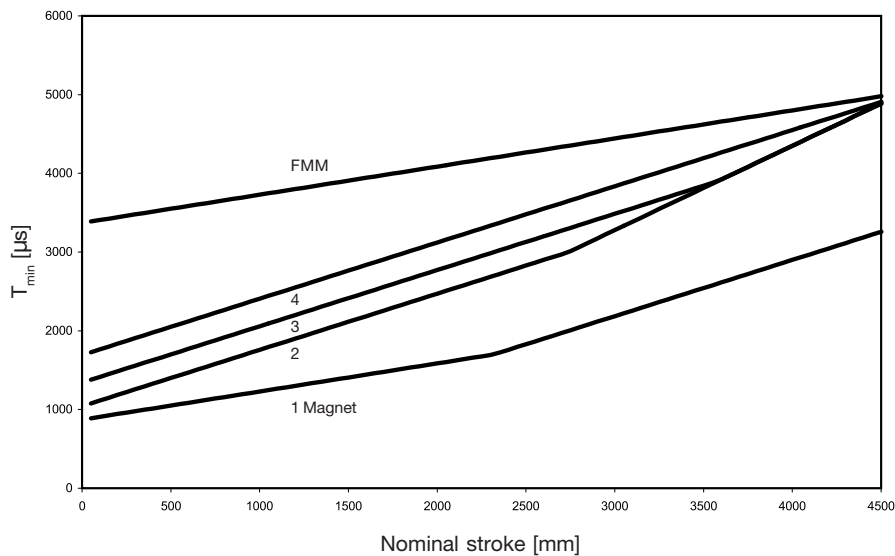


Fig. 10-1: Minimum time for Sync periods

Formula for 1 magnet:

Length ≤ 2350 mm

$$t_{min} = (\text{nom. stroke} + 60 \text{ mm}) / 2800 \text{ m/s} + \text{CALC}$$

Length > 2350 mm

$$t_{min} = (\text{nom. stroke} + 60 \text{ mm}) / 2800 \text{ m/s} * 2$$

Formula for 2 magnets:

Length ≤ 2750 mm

$$t_{min} = (\text{nom. stroke} + 60 \text{ mm}) / 2800 \text{ m/s} * 2 + \text{CALC}$$

Length > 2750 mm

$$t_{min} = (\text{nom. stroke} + 60 \text{ mm}) / 2800 \text{ m/s} * 3$$

Formula for 3 magnets:

Length ≤ 3600 mm

$$t_{min} = (\text{nom. stroke} + 60 \text{ mm}) / 2800 \text{ m/s} * 2 + \text{CALC}$$

Length > 3600 mm

$$t_{min} = (\text{nom. stroke} + 60 \text{ mm}) / 2800 \text{ m/s} * 3$$

Formula for 4 magnets:

$$t_{min} = (\text{nom. stroke} + 60 \text{ mm}) / 2800 \text{ m/s} * 2 + \text{CALC}$$

Formula for FMM:

$$t_{min} = (\text{nom. stroke} + 60 \text{ mm}) / 2800 \text{ m/s} + 1700 \mu\text{s} + \text{CALC}$$

CALC = 850 µs (1 magnet)

CALC = 1,000 µs (2 magnets)

CALC = 1,300 µs (3 magnets)

CALC = 1,650 µs (4 magnets)

CALC = 1,650 µs (FMM-Mode)