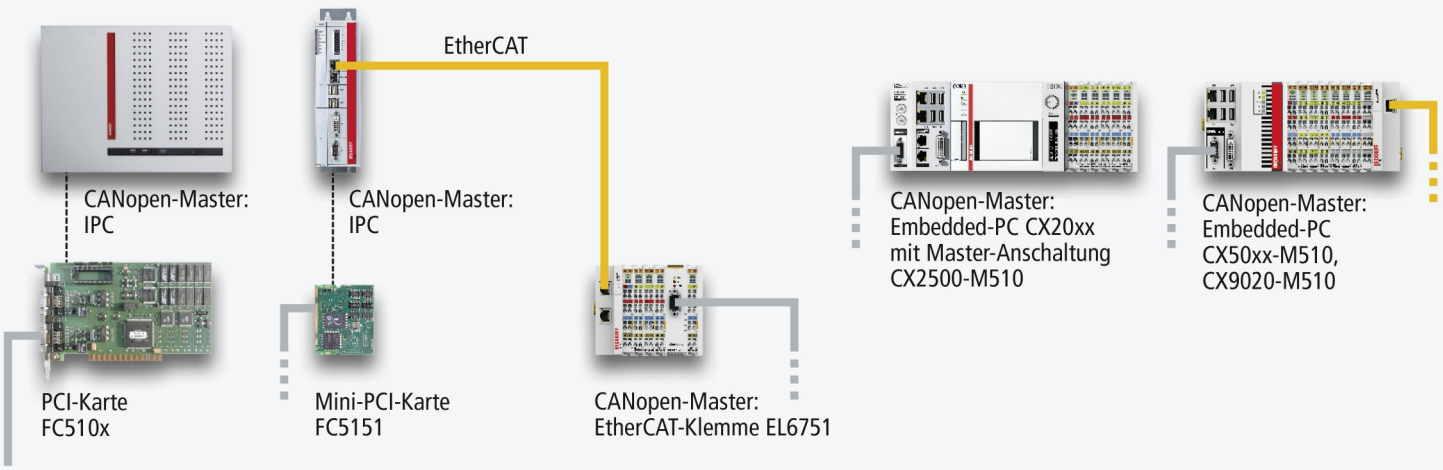
BECKHOFF New Automation Technology

Dokumentation | DE

CAN-Interface

Schnittstelle für CANopen-Master von Beckhoff



31.08.2022 | Version: 1.5

Inhaltsverzeichnis

1 Vorwort

1.1 Hinweise zur Dokumentation

Zielgruppe

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.
Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.
Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.
Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.
Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH. Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland.

Copyright

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Erklärung der Hinweise

In der vorliegenden Dokumentation werden die folgenden Hinweise verwendet.
Diese Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

⚠ GEFAHR

Akute Verletzungsgefahr!

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

⚠ WARNUNG

Verletzungsgefahr!

Wenn dieser Sicherheitshinweis nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

⚠ VORSICHT

Schädigung von Personen!

Wenn dieser Sicherheitshinweis nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt/Geräten oder Datenverlust

Wenn dieser Hinweis nicht beachtet wird, können Umweltschäden, Gerätebeschädigungen oder Datenverlust entstehen.

Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

1.3 Ausgabestände der Dokumentation

Version	Kommentar
1.5	• Kapitel *CAN-FD-Zugriff mit FC532x und CX-M530* Update
1.4	• Kapitel *CAN-FD-Zugriff mit FC532x und CX-M530* hinzugefügt
1.3	• Einleitung aktualisiert • Gestaltung der Sicherheitshinweise an IEC 82079-1 angepasst.
1.2	• Migration und Strukturanpassung
1.1	• Überarbeitung
1.0	• Erste Veröffentlichung

2 Einleitung

Fast alle CANopen-Master von Beckhoff bieten das sogenannte CAN-Interface als Schnittstelle an. Das CAN-Interface ist eine Layer-2-Implementierung der CAN-Schnittstelle. Das ermöglicht beliebige CAN-Telegramme zu empfangen wie auch zu senden. Hierbei spielt das verwendete überlagerte Protokoll keine Rolle, d.h. es können damit alle CAN-basierenden Protokolle verwendet werden wobei der Protokoll-Teil dann in der SPS realisiert werden muss.

Das CAN-Interface besteht aus einem Buffer der zyklisch abgearbeitet wird. Der Buffer kann von 11 bis 32 Datentelegramme beinhalten.

Der Transmit Buffer (Tx) enthält die Daten, die gesendet werden sollen und der Receive Buffer (Rx) die Daten, die empfangen worden sind. Je nach CAN-Master können 11 Bit- oder 29 Bit-Nachrichten empfangen bzw. gesendet werden. Der Buffer wird mit der Zykluszeit der Task bearbeitet. Bei einer Buffer-Größe von 10 können pro Task-Zyklus also maximal 10 CAN-Telegramme gesendet werden bzw. empfangen werden.

11-Bit-Identifier, auch „Base Frame Format“ genannt (CAN 2.0A)

29-Bit-Identifier, auch „Extended Frame Format“ genannt (CAN 2.0B)

CAN Interface - Unterstützte Funktionalitäten

	CAN2.0A 11 Bit ID	CAN2.0B 29 Bit ID	CAN FD	Fast CAN Queue[1]	Optimized CAN Queue[1]	Transaction Number [▶ 12]	Time Stamp[2]
EL6751 Legacy-Mapping	✓	✓	-	-	✓	✓[3]	-
EL6751 MDP-Mapping	✓	✓	-	✓	✓	✓	-
CCAT	(✓)[4]	✓	-	-	-	✓ ab FW 1.17	✓ ab FW 1.17
CX1500-M510	✓	✓	-	-	-	-	-
FC5151 FC510x	✓	✓ ab FW 2.14	-	-	-	-	-
FC532x CX-M530[5]	✓	✓	✓	-	-	✓	✓

[1]) nicht im 29 Bit Mode, nicht mit Transaction Number [▶ 12]
[2]) nur im 29 Bit Mode und mit Transaction Number [▶ 12]
[3]) nur im 29 Bit Mode
[4]) wird abgedeckt durch die 29-Bit-ID-Option
[5]) in Vorbereitung

CCAT

Was ist CCAT? Welche Beckhoff-Produkte stehen dahinter?

Das CCAT-Interface ist die aktuelle CAN-Implementierung der Firma Beckhoff und wird von den Beckhoff PCI-Express-Karten und den OnBoard-Schnittstellen der Beckhoff Embedded-PCs verwendet. Dies sind z. B. folgende Produkte und nur für die CANopen Master verfügbar:
FC512x, C20xx-M510, CX8x50, CX9x20-M510, CX51xx-M510

3 Einbinden in TwinCAT

Wenn Sie im TwinCAT einen CANopen-Master angelegt haben, können Sie anstatt von CANopen-Slaves das CAN-Interface auswählen.

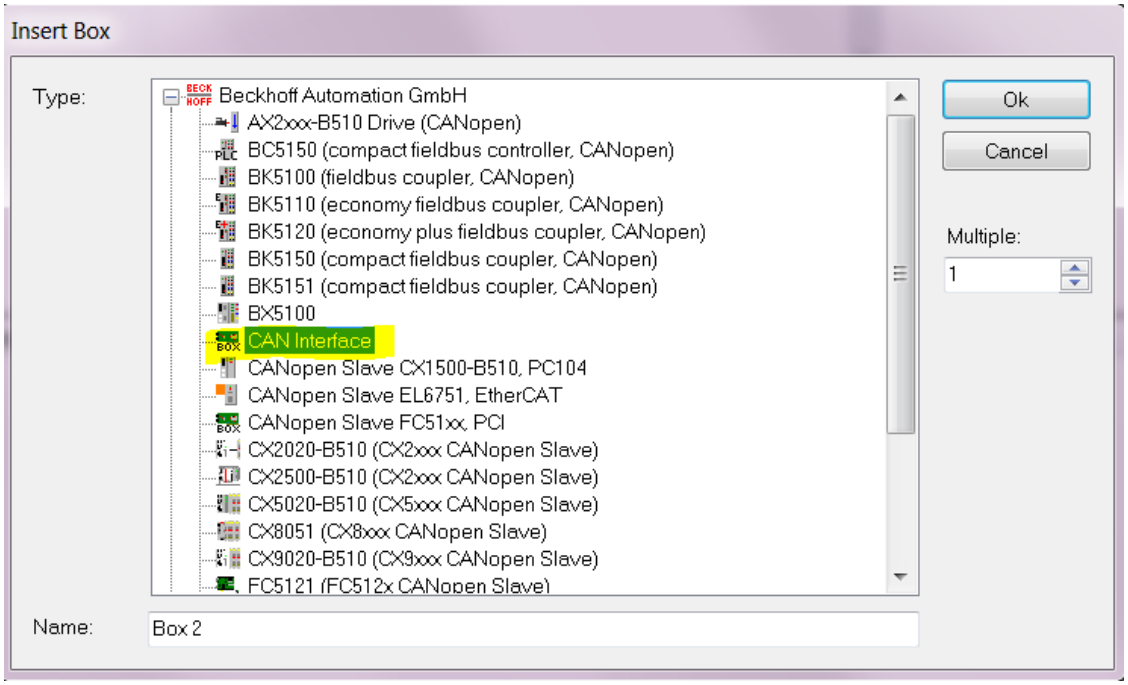


Abb. 1: Auswahl des CAN-Interfaces

Dann werden Sie aufgefordert das entsprechende Interface zu wählen, welches sie nutzen wollen, bzw. die Größe des Buffers. Achten Sie darauf wenn Sie das Interface später noch mal ändern, das es dann in der Regel so ist, dass entsprecht des Modus, das Interface neu Aufgebaut wird und damit Verknüpfungen gelöscht werden. Wählen Sie auch nur Modi aus, die Ihre Hardware auch unterstützt (siehe Tabelle *CAN Interface - Unterstützte Funktionen*).

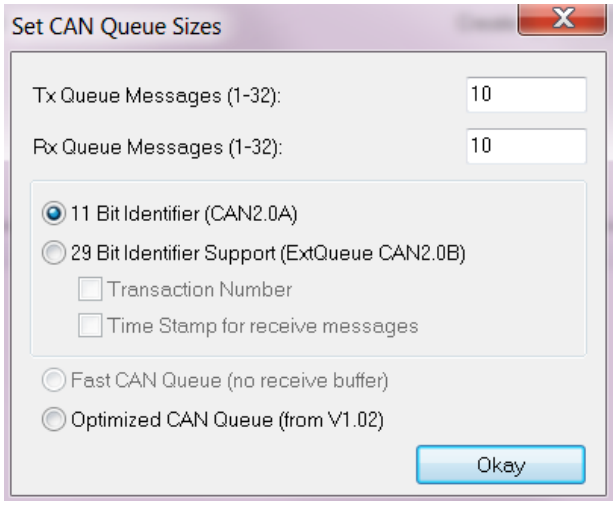


Abb. 2: Auswahl des Frame-Formats

4 Buffer-Größe in Abhängigkeit der Zykluszeit

Es gilt abzuschätzen, wie groß der Buffer bzw. wie schnell die Task-Zeit sein muss. Die folgende Tabelle hilft dies im Vorfeld fest zu legen.

Die CCAT-CAN-Master besitzen einen Speicher für 512 Messages, die EL6751 für 150 Messages (RxMessages). Sollten die Daten nicht schnell genug aus dem Speicher geholt werden, gehen diese verloren. Es gibt keinen Hinweis, daher sollte oder muss man den Worst Case vorher abschätzen, bzw. die Variable NoOfRxMessages sollte möglichst kleiner als der maximale Buffer-Wert sein. Ist dieser immer oder fast in jeden Zyklus auf dem maximalen Wert, dann ist das ein Hinweis, dass mehr Daten empfangen werden als pro Zyklus erfasst werden können. Abhilfe: Verringern Sie die Task-Zykluszeit oder erhöhen sie den Buffer der CAN-Queue.

Beispiel

Ein CAN-Telegramm mit 11 Bit Identifier und 8 Byte Nutzdaten braucht bei 500 Kbit/s ca. 260 µs. Geht man im Worst Case von 100% Buslast aus, wären es bei 1 ms maximal 3 Telegramme. D.h. ein Buffer von maximal 4 wäre in diesem Fall ausreichend. Verwendet man statt 1 ms eine Taskzeit von 5 ms sollten der Buffer 20 (5000 µs / 260 µs) oder größer sein. Hierbei ist zu berücksichtigen, dass bei dieser Betrachtung nur die Daten in einer Richtung berücksichtigt sind und die CAN Daten immer 8 Byte haben. Da in der Regel nicht von 100% Buslast auszugehen ist, kann man auch die Variable NoOfRxMessages auswerten und schauen ob diese in den meisten Fällen unter der Maximalanzahl der angelegten Buffer liegt. Ist NoOfRxMessages oft beim maximalen Wert, sollte die Task-Zeit verkleinert oder der Buffer vergrößert werden.

Worst Case

Das CAN-Interface ist aber so ausgelegt das man in der Regel die Daten immer schneller abholen kann als das diese im Buffer auflaufen.

Beispiel

1 MBaud Datenlänge 0 bedeutet 50 µs pro CAN Message. Bei 1 ms Task-Zeit wären das
1000 µ / 50 µs = 20
D.h. auch in dieser extremen Worst-Case-Betrachtung wäre ein Buffer von 20 ausreichend um alle CAN-Telegramme zu empfangen.

Tabelle für die Telegrammlaufzeiten bei 11 Bit Ident [ms][1)]

Bit rate [kBit/s]	Datenlänge in Byte								
	0	1	2	3	4	5	6	7	8
50	1.09	1.28	1.47	1.66	1.86	2.05	2.24	2.34	2.62
125	0.44	0.51	0.59	0.67	0.74	0.82	0.90	0.97	1.05
250	0.22	0.26	0.29	0.33	0.37	0.41	0.45	0.49	0.52
500	0.11	0.13	0.15	0.17	0.19	0.21	0.22	0.24	0.26
1000	0.05	0.06	0.07	0.08	0.09	0.10	0.11	0.12	0.13

[1)] Daten von CIA

5 Funktionalitäten

5.1 CAN-Queues

5.1.1 Fast-CAN-Queue (unbuffered)

Es werden keine Empfangsnachrichten mehr auf der EL6751 zwischengespeichert. Die RxQueue sollte also so groß sein, dass alle theoretisch innerhalb eines EtherCAT-Zyklus empfangbaren Nachrichten hinein passen.

Die empfangenen Nachrichten müssen nicht mehr bestätigt werden. Die EL6751 zählt den RxCounter hoch, wenn neue Nachrichten empfangen wurden.

In Senderichtung werden auch nur noch die Daten abhängig vom geänderten TxCounter und der NoOfTxMessages kopiert, so dass die Anzahl der parallelen Messages in der Queue eigentlich keinen Einfluss mehr auf die Laufzeit hat (nur noch die NoOfTxMessages).

Die EL6751 arbeitet im 3-Puffer-Betrieb (damit sie immer einen Puffer hat, in den sie empfangene CAN-Messages kopieren kann), kann es passieren, dass in den **nicht benutzten** Messages falsche oder alte Daten stehen.

Auf dem Karteireiter *CoE-Online* kann das Objektverzeichnis ausgelesen werden. Wenn der Index 0x1C32:08 auf 1 gesetzt wird/ist, wird die lokale Zykluszeit der EL6751 gemessen und in Index 0x1C32:05 abgelegt (Maximalwert). Damit kann man sehen, ob die EL6751 innerhalb des EtherCAT - Zyklus fertig wird.

Die Fast-CAN-Queue darf keine weiteren CANopen- oder CAN-Layer-2-Nodes beinhalten.

5.1.2 Optimized-CAN-Queue (buffered)

Die Empfangsnachrichten der EL6751 werden zwischengespeichert. Die EL6751 arbeitet im 1-Buffer-Betrieb.

Für beide Funktionen Fast-CAN-Queue und Optimized-CAN-Queue gilt:

Vorteile

- Höhere Verarbeitungsgeschwindigkeit
- Die Fast-CAN-Queue verzichtet auf sämtliches Beiwerk und ist somit für die schnellste Verarbeitung/ Reaktion der Daten vom Bus prädestiniert.

Nachteile

- Beide Modi unterstützen nur 11 Bit Identifier.
- Es sind keine Filter nutzbar.

5.2 Transaction Number

Anhand der *transactionNumber* kann man erkennen, bis zu welcher CAN-Nachricht bei einer CAN-Queue im letzten CAN-Zyklus gesendet wurde. Bei den einzelnen TxMessages[n] kann eine beliebige *transactionNumber* eingetragen werden (z. B. eine fortlaufende Nummer). Am Ende eines CAN-Zyklus wird die *transactionNumber* der zuletzt gesendeten TX-Message in die Inputs.RxQueue.TransactionNumber geschrieben.

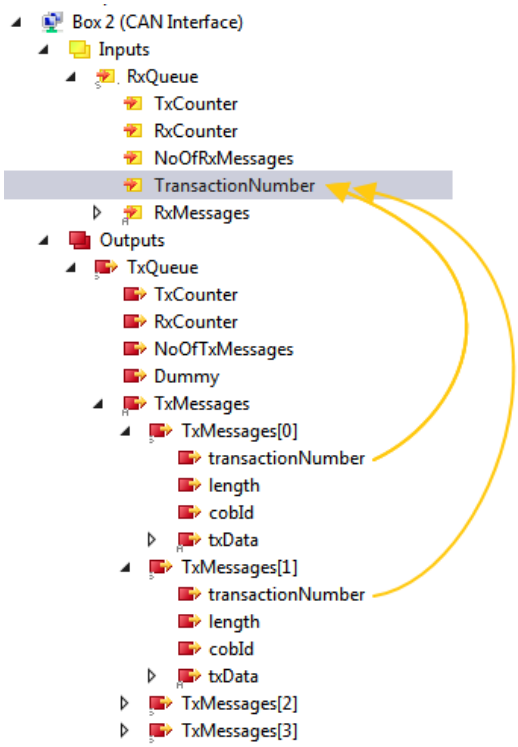


Abb. 3: Transaction Number

5.3 Time Stamp

CCAT basierende CAN-Controller (z. B. FC512x, -M510) liefern mit einen Empfangs-Zeitstempel die Zeit zu welcher der CAN-Frame eingetroffen ist (64-bit Integer Wert in Nanosekunden).

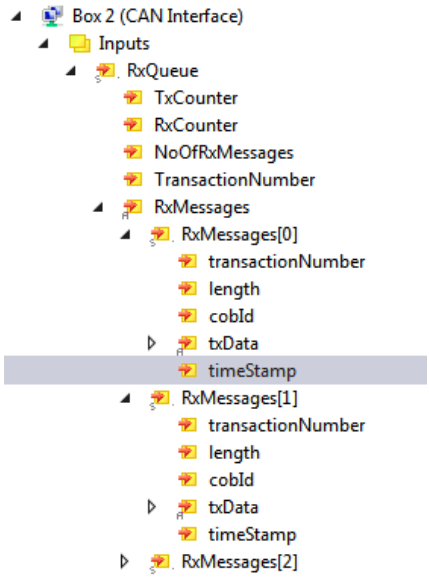


Abb. 4: Time Stamp

6 Aufbau des CAN-Interfaces

Je nach ausgewählter Option sieht das CAN-Interface unterschiedlich aus. Ein 11 Bit Identifier Interface ist anders Aufgebaut als ein 29 Bit Identifier Interface. Zusätzlich kann das Interface die Transaction Number wie auch den Time Stamp beinhalten. Bei Verwendung von Strukturen muss das Zielsystem berücksichtigt werden, welches Alignment es unterstützt. Unter TwinCAT 3 sind entsprechende Attribute nutzbar {attribute 'pack_mode':= '0'}.

Das Interface besteht aus der Kommunikation mit dem Interface und den bis zu 32 CAN-Nachrichten. Die Inputs können nur gelesen werden die Outputs beschrieben werden.

Das Interface wird wie folgt angesprochen:

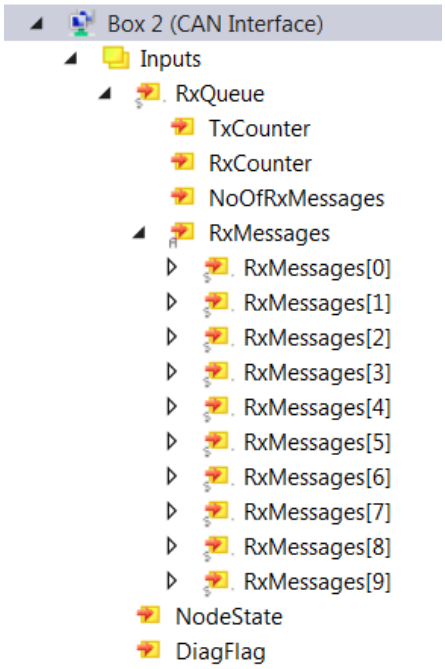


Abb. 5: CAN-Interface - Inputs

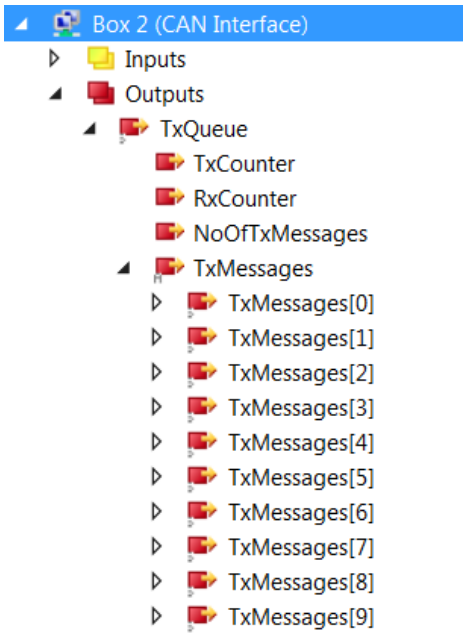


Abb. 6: CAN-Interface - Outputs

Outputs.TxCounter wird +1 gesetzt, wenn Daten gesendet werden sollen. Wie viele Nachrichten aus dem Buffer gesendet werden sollen, wird im NoOfTxMessages mitgeliefert. Der RxCounter zeigt an ob Daten neu im Buffer liegen. Wie viele Daten neu im Buffer liegen wird mit NoOfRxMessages übergeben.

Haben Sie die Daten abgeholt, so setzen Sie den Outputs.RxCounter:=Inputs.RxCounter. Damit weiß das CAN-Interface, dass es den Buffer erneut Füllen kann. Es müssen immer alle Daten ausgelesen werden, da das CAN Interface wieder alle Message Strukturen füllt wenn es notwendig ist.

Beispiel-Code zum Senden

```
if Outputs.TxCounter = Inputs.TxCounter then
      for i=0 to NumberOfMessagesToSend do
               Outputs.TxMessage[i] = MessageToSend[i];
      End_for
      Outputs.NoOfTxMessages = NumberOfMessagesToSend;
      Outputs.TxCounter := Outputs.TxCounter + 1;
end_if
```

Beispiel-Code zum Lesen

```
if Outputs.RxCounter <> Inputs.RxCounter then
       for I := 0 to (Inputs.NoOfRxMessages-1) do
            MessageReceived[i] := Inputs.RxMessage [i];
       End_for
       Outputs.RxCounter := Inputs.RxCounter;
end_if
```

Message-Struktur bei Verwendung des 11 Bit Identifiers

Die Message-Struktur bei Verwendung des 11 Bit Identifiers besteht aus der COB ID [2 Byte] und den 8 Byte Daten. Die COB ID ist wie folgt aufgebaut:

- Bit 0-3: Länge der Daten (0…8)
- Bit 4: RTR
- Bit 5-15: 11 Bit-Identifier

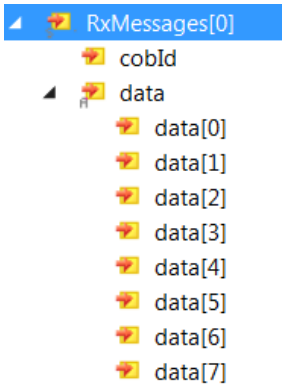


Abb. 7: Message Struktur bei Verwendung des 11 Bit Identifiers

Da beim 11 Bit Identifier COB ID, Länge und RTR Bit in einem Wort kodiert sind ist folgendes Beispiel hilfreich um die Daten aus dem Wort zu decodieren. Legen Sie sich hier eine Struktur an, um die dekodierten Daten in dieser Abzulegen.

```
IF RXCounter_Out <> RXCounter_In THEN
   FOR I := 0 TO (NoOfTxMessages-1) DO
       stCANInterfaceMessageValue[i].Lengh:=WORD_TO_BYTE(stCANInterfaceMessage[i].CobID) AND 16#0F;
       stCANInterfaceMessageValue[i].RTR:=stCANInterfaceMessage[i].CobID.4;
       stCANInterfaceMessageValue[i].CobID :=ROR(stCANInterfaceMessage[i].CobID,5) AND 16#07FF;
       stCANInterfaceMessageValue[i].Data := stCANInterfaceMessage[i].Data;
       CASE stCANInterfaceMessageValue[i].CobID OF
            16#318: COB318:=COB318+1;
            16#718: COB718:=COB718+1;
            16#1CD: COB1CD:=COB1CD+1;
            memcpy(ADR(TempValue),ADR(stCANInterfaceMessage[i].Data[6]),2);
            16#1ED: COB1ED:=COB1ED+1;
       ELSE
            COBALLOther:=COBAllOther+1;
       END_CASE
   End_for
   RXCounter_Out:=RXCounter_In;
END_IF
```

Message-Struktur bei Verwendung des 29 Bit Identifiers

Die Message-Struktur bei Verwendung des 29 Bit Identifiers besteht aus der Länge [2 Byte] der COB ID [4 Byte] und den 8 Byte Daten.

Lenght: Länge der Daten (0…8)

Die COB ID ist wie folgt aufgebaut:

- Bit 0-28: 29 Bit Identifier
- Bit 30: RTR
- Bit 31:
  - 0: 11 Bit-Identifier,
  - 1: 29 Bit Identifier

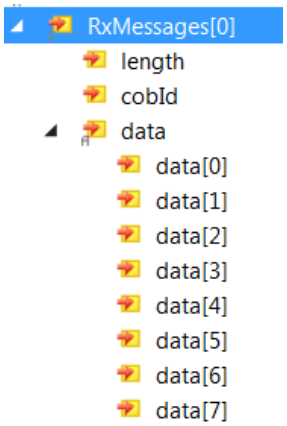


Abb. 8: Message-Struktur bei Verwendung des 29 Bit Identifiers

7 Benutzung eines Filters

Will man nicht alle Telegramme im CAN-Interface empfangen, gibt es die Möglichkeit Filter zu setzen. Damit reduziert man die Anzahl der CAN-Telegramme im CAN-Interface und lässt damit nur die Telegramme zu, die auch benötigt werden.

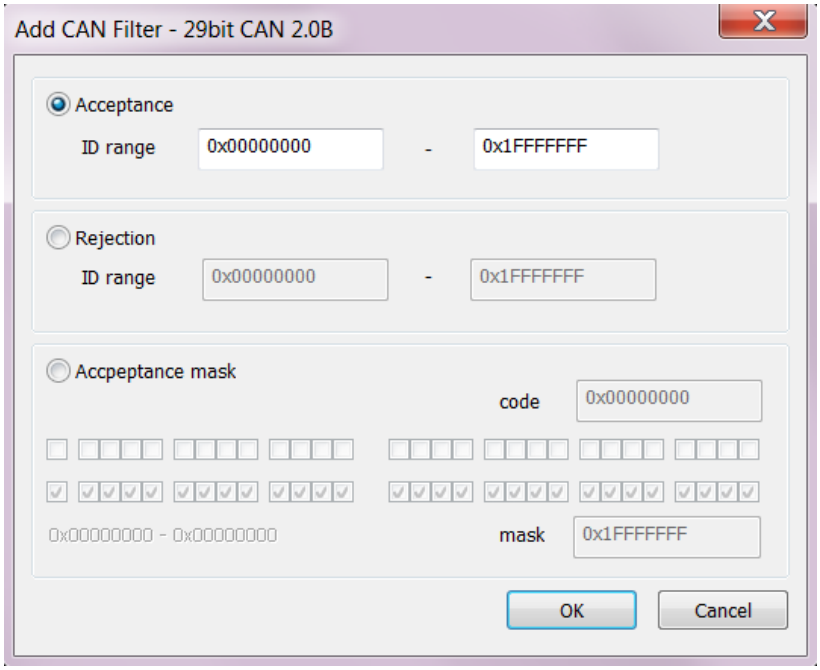


Abb. 9: CAN-Filter

Acceptance:
Hier werden die Identifier eigetragen die zum CAN-Interface weitergeleitet werden sollen.

Rejection:
Hier sind die Identifier eingetragen die nicht zum CAN-Interface weitergeleitet werden sollen.

Acceptance mask:
Hier kann auf Bit-Ebene eingestellt werden, welche Identifier zum CAN-Interface weitergeleitet werden sollen.

Beispiel anhand des 29 Bit Identifiers

In dem Beispiel werden alle Telegramme von Identifier 0x0400 … 0x0700 in das CAN-Interfaces übertragen. Dies wird mit einem "+" bei Info angezeigt.
"+" bedeutet, der Filter lässt die Daten zum CAN-Interface durch (Acceptance)
"-" bedeutet, der Filter lässt die Daten nicht zum CAN-Interface durch (Rejection)

CAN Rx	Acceptance	Rejection	Info	Comment
Filter 1		0x00000000 - 0x000003FF	-	
Filter 2	0x00000400 - 0x00000700		+	manually added (code/mask)
Filter 3		0x00000701 - 0x1FFFFFFF	-	

Abb. 10: Beispiel anhand des 29 Bit Identifiers

8 CAN-FD-Zugriff mit FC532x und CX-M530

Senden und Empfangen von FD Messages

Dieses Kapitel beschreibt die CAN-FD-Funktion des CAN-FD-Interfaces.

8.1 CAN-FD-Interface

Das CAN-Interface für die FC532x, CX2500-M530 sowie das Options-Interface für CX-M530 CAN unterstützen den Zugriff auf die CAN-FD-Funktionalität.

Die Bedienung dies CAN-Interfaces sowie die Funktionen *Transactions-Number* und *Timestamp* entsprechend der des bekannten Interfaces (siehe Kapitel Aufbau des CAN-Interfaces [▶ 13]).

Neu sind der Message-Datentyp der RX- und TX-Queue und die Baudrateneinstellung.

8.2 CAN-FD-Device und Baudrateneinstellung

Auf die CAN-FD-Funktion ist über das Device *Flexibel Data-Rate CAN (CCAT)* zuzugreifen.

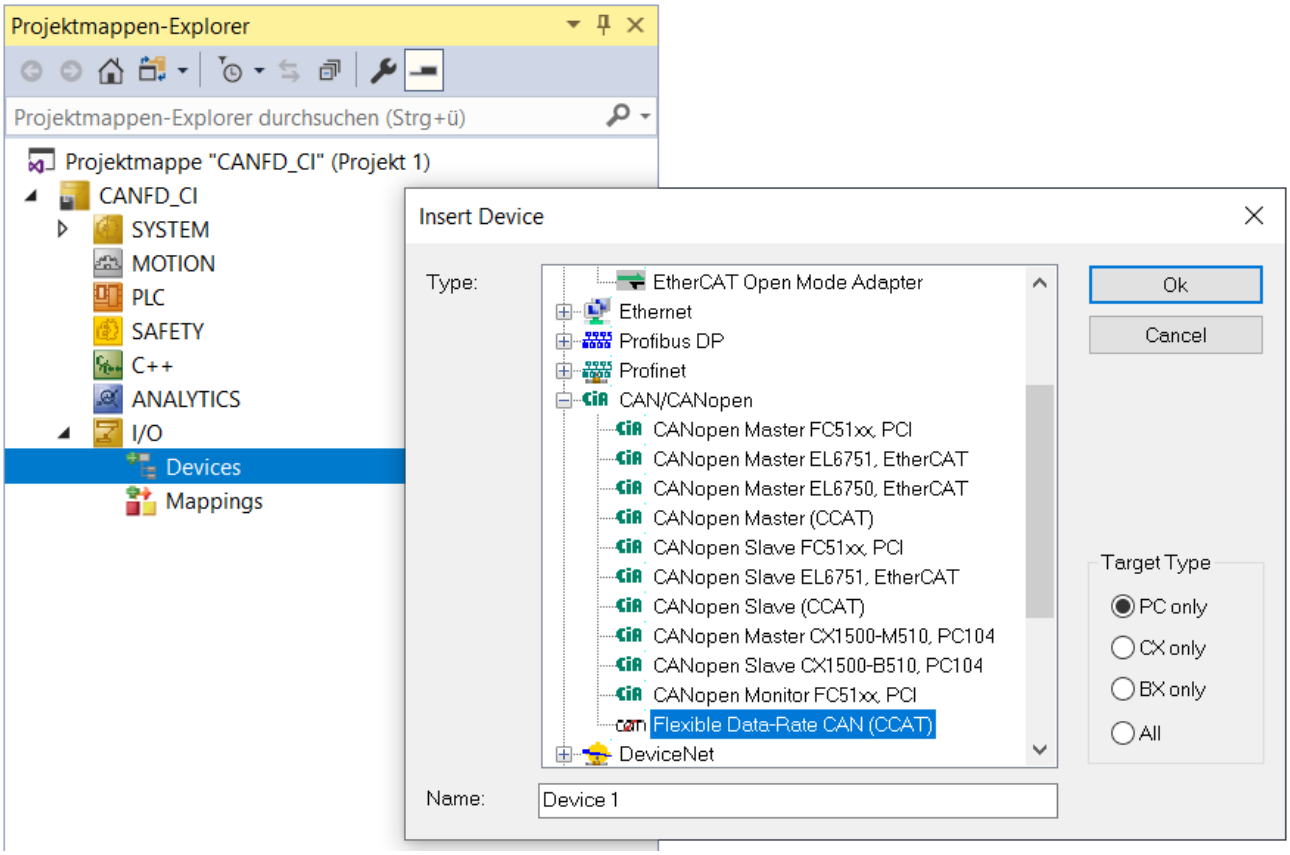


Abb. 11: Flexibel Data-Rate CAN (CCAT)

Die Baudraten bei CAN FD können für die Arbitierungs- und die Datenübertragungs-Phase unterschiedlich eingestellt werden. Ebenso ist es weiterhin möglich wie beim klassischen CAN für beide Phasen die gleiche Baudrate einzustellen.

Folgende Baudraten sind möglich:

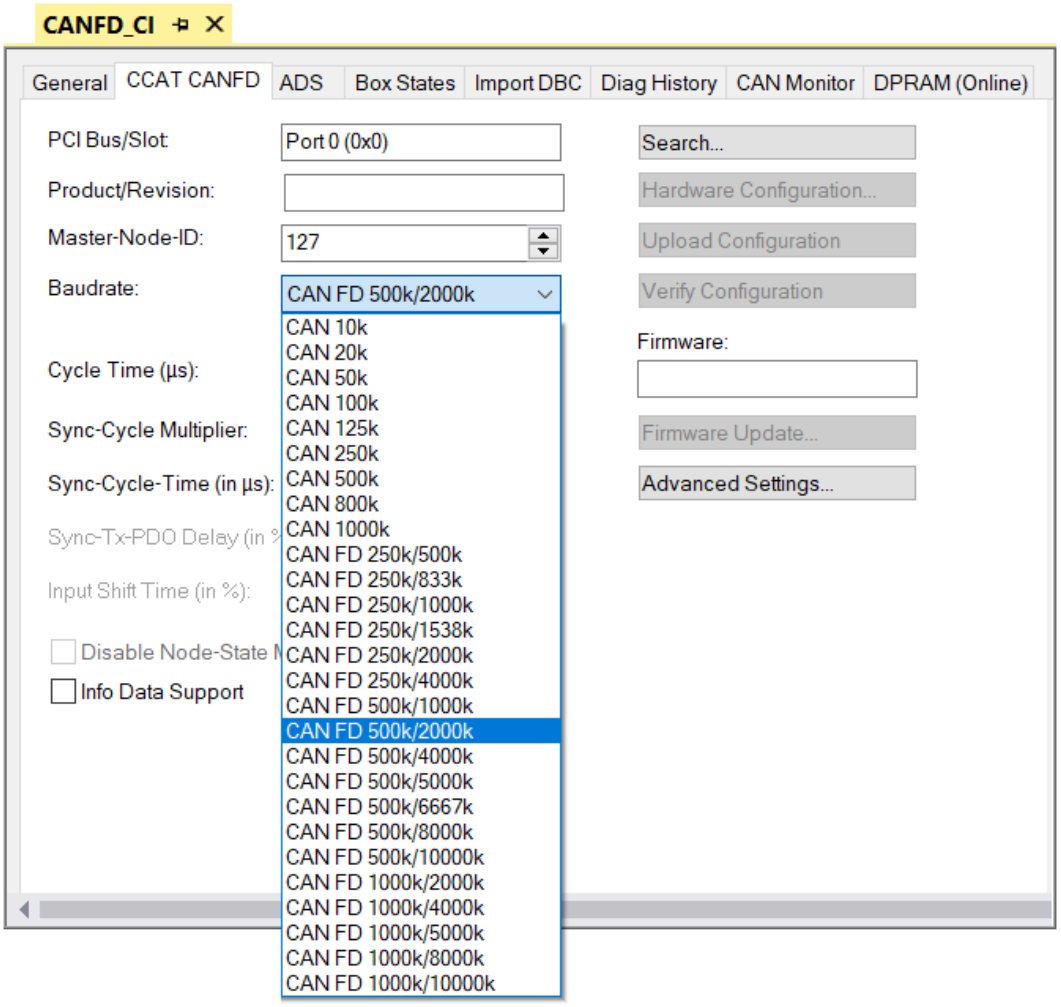


Abb. 12: Mögliche Baudraten

Die erste Zahl gibt die Baudrate für die Arbitierungsphase und die zweite für die Datenphase an.

Hinweis zur Baudrate 10 Mbit/s

Die Baudrate 10 Mbit/s ist aktuell nur unter sehr idealen Bedingungen möglich und entspricht momentan nicht einer praxisnahen Einstellung. Eine aktuell gebräuchliche Einstellung für CAN FD ist 500k/2000k.

8.3 CAN FD Message Datenstrukturen

Für die CAN-FD-Unterstützung beim CAN-Interface wurde folgende neue Datenstrukturen eingeführt.

```
TYPE CANFDTSRXQUEUE :
      STRUCT
            dataLenght : BYTE;
            EDL : BIT;
            BSR : BIT;
            ESI : BIT;
            cobId : UDINT;
            rxData : CANFDMESSAGE;
            timeStamp : ULINT;
      END_STRUCT
END_TYPE
```

```
TYPE CANFDTXQUEUE :
      STRUCT
            transactionNumber : UINT;
            dataLenght : BYTE;
            EDL : BIT;
            BSR : BIT;
            ESI : BIT;
            cobId : UDINT;
            txData : CANFDMESSAGE;
      END_STRUCT
END_TYPE
```

```
TYPE CANFDMESSAGE :
      ARRAY [0..63] OF USINT;
END_TYPE
```

Diese Strukturen stehen bei den IOs im System-Manager als auch in der PLC zur Verfügung.

8.3.1 Datenlänge

Für die Datenlänge sind bei Can FD Frames Werte bis 64 Byte möglich. Da diese Werte im klassischen CAN-Arbitierung-Header übertragen werden sind folgende Werte möglich:

0 … 8, 12, 16, 20, 24, 36, 48 und 64

Entsprechen die Längen beim Senden nicht diesen Werten, so werden sie vom Device auf den nächsthöheren Wert angepasst.

Bei einem CAN FD Frame sind die Werte 0 bis 15 für das DLC Feld gültig. Der Wert des DLC Feldes bestimmt die Anzahl der Bytes im Data Field und wird in folgender Weise interpretiert:

DLC Wert	Data Field size (Bytes)
0 .. 8	0 .. 8
9	12
10	16
11	20

DLC Wert	Data Field size (Bytes)
12	24
13	36
14	48
15	64

Abb. 13: DLC Wert vs. Data field size

Das CAN Interface macht diese Umrechnung automatisch, d.h. im CAN Interface geben Sie nur die tatsächliche Anzahl der Bytes an (erlaubt sind Werte von 0..64 Byte).
Die Umrechnung auf die DLC Werte wird vom CAN Interface dann im Hintergrund mit dem nächstgrößeren DLC Wert durchgeführt.

Beispiel: wenn Sie im CAN Interface die Datenlänge 32 eintragen, werden 36 Byte mit dem DLC Wert 13 versendet.

8.3.2 CAN FD Bits

Die gegenüber dem klassischen CAN-Interface neuen Bits in der Datenstruktur *Queue* haben folgende Bedeutungen.

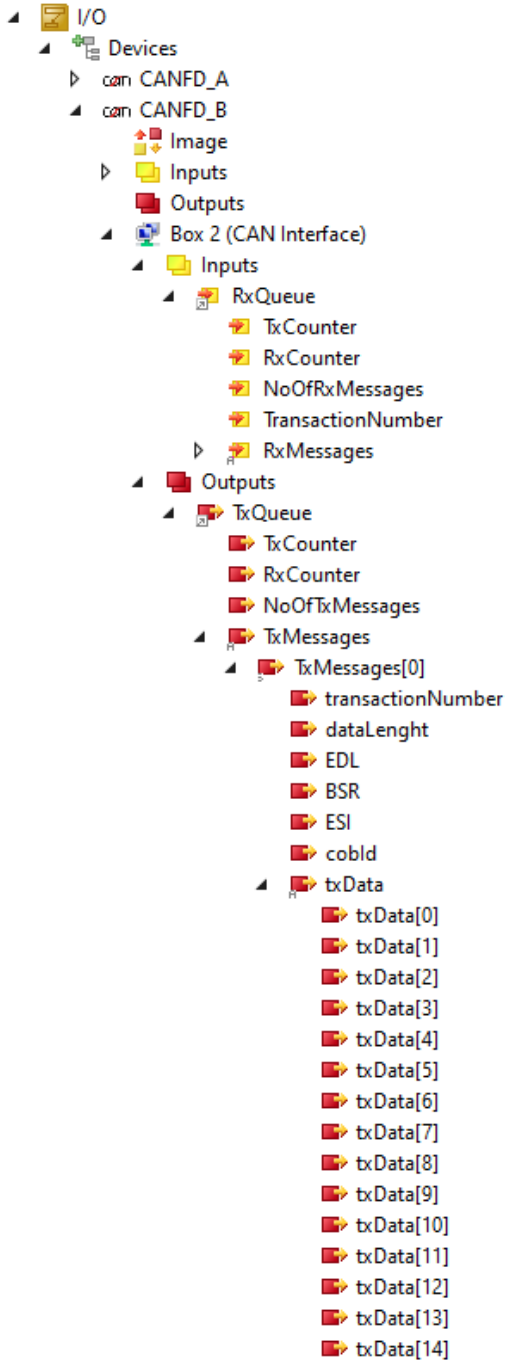


Abb. 14: Datenstruktur *Queue*

EDL
Mit dem **EDL** (Enhanced Data Length) Bit steuert man, ob ein FD oder ein classic Frame gesendet werden soll bzw. was für ein Frame empfangen wurde.

BSR
Mit dem **BSR** (Bit Rate Switched) Bit, wird angegeben, ob in der Datenphase auf die höhere Baudrate umgeschaltet werden soll bzw. wie der Frame empfangen wurde.

ESI
Das Bit **ESI** (Error State Indicator) zeigt an, ob es bei dem Frame einen Fehler gab (Rx) oder es gibt (Tx).

8.3.3 InfoData

Aus den Infodaten der FC532x / CX-M530 können folgende Daten gelesen werden:

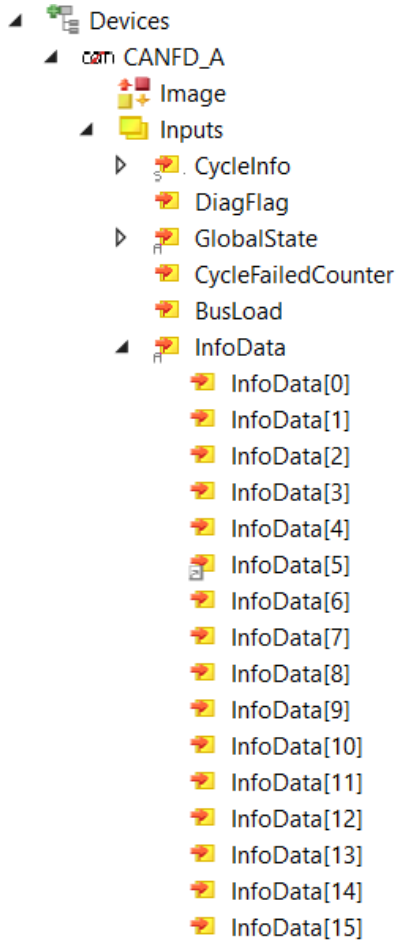


Abb. 15: InfoData

InfoData[0] – CAN Status
Bit0 – BusOff (0 no error - 1 bus off error)
Bit1 – Error Passive
Bit2 – Node Active
Bit3 – Warning Limit
Bit4 – Overload
Bit6 – Bus Idle;

InfoData[1] – Arbitration phase baudrate
Byte0 = Jump Width
Byte1 = Time A
Byte2 = Time B
Byte3 = Pre Scaler:

InfoData[2] – Data phase baudrate
Byte0 = Jump Width
Byte1 = Time A
Byte2 = Time B
Byte3 = Pre Scaler:

InfoData[4] – Send Fifo (High-Prio) UsedWords;
InfoData[5] – Send Fifo (Low-Prio) UsedWords;
InfoData[6] – Receive Fifo Counter;
InfoData[7] – AckError Counter;
InfoData[8] – BitError Counter;
InfoData[9] – CrcError Counter;
InfoData[10] – FormError Counter;

9 Anhang

9.1 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten: https://www.beckhoff.de

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline:	+49(0)5246 963 157
Fax:	+49(0)5246 963 9157
E-Mail:	support@beckhoff.com

Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline:	+49(0)5246 963 460
Fax:	+49(0)5246 963 479
E-Mail:	service@beckhoff.com

Beckhoff Firmenzentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon:	+49(0)5246 963 0
Fax:	+49(0)5246 963 198
E-Mail:	info@beckhoff.com
Internet:	https://www.beckhoff.de

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.de
www.beckhoff.de