

CAN 300 PRO Anwendungsbeispiel DeviceNet Slave

CAN Kommunikations-Baugruppe für S7-300
Anwendungsbeispiel für DeviceNet Slave Hantierungsbausteine

Handbuch

Ausgabe 1 / 11.03.2010



Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung dieses Handbuches, oder Teilen daraus, vorbehalten. Kein Teil des Handbuches darf ohne schriftliche Genehmigung der Systeme Helmholtz GmbH in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, oder unter Verwendung elektronischer Systeme reproduziert, verarbeitet, vervielfältigt oder verbreitet werden. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Copyright © 2010 by

Systeme Helmholtz GmbH

Hannberger Weg 2, 91091 Großenseebach

Hinweis:

Der Inhalt dieses Handbuches ist von uns auf die Übereinstimmung mit der beschriebenen Hard- und Software überprüft worden. Da dennoch Abweichungen nicht ausgeschlossen sind, können wir für die vollständige Übereinstimmung keine Gewährleistung übernehmen. Die Angaben in diesem Handbuch werden jedoch regelmäßig aktualisiert. Bitte beachten sie beim Einsatz der erworbenen Produkte jeweils die aktuellste Version des Handbuchs, die im Internet unter www.helmholtz.de einsehbar ist und auch heruntergeladen werden kann.

Unsere Kunden sind uns wichtig. Wir freuen uns über Verbesserungsvorschläge und Anregungen.

Änderungen in diesem Dokument:

Stand	Datum	Änderung
1	11.03.2010	1. Version

Inhaltsverzeichnis

1	Übersicht	6
1.1	Anwendung und Funktionsbeschreibung	6
1.1.1	Get Attribute	6
1.1.2	Set Attribute	6
1.1.3	Freigaben	7
1.2	Aufbau SPS	8
2	Projektierung der CAN Baugruppe	9
3	Programmierung in der SPS	10
3.1	Übersicht	10
3.2	FB 99 „DN-Adapter“	10
3.3	Beispiel OB 100	12
3.4	Beispiel OB 1	13
3.5	Rückgabeparameter RETVAL	14
4	Diagnose	15
4.1	Prozessabbild inder SPS	15
4.1.1	Byte 0: Baugruppenstatus	15
4.1.2	Byte 1: Fehler-Status (EFLG) des CAN-Controllers	16
4.1.3	Byte 2: FIFO-Status Bits	16
4.1.4	Byte 3/4: CAN-Controller Tx/Rx Fehlerzähler	16
4.2	CANParam Debug-Screen	17

1 Übersicht

1.1 Anwendung und Funktionsbeschreibung

Dieses Handbuch beschreibt das Anwendungsbeispiel (Hantierungsbaustein) zur Verwendung einer CAN 300 PRO als DeviceNet Slave an einer S7-300.

Es ist als Ergänzung zum Handbuch der CAN 300 PRO Baugruppe zu verwenden. Die Informationen aus diesen Handbüchern werden als bekannt vorausgesetzt.

Der hier beschriebene Hantierungsbaustein FB 99 stellen folgende DeviceNet Funktionen zur Verfügung:

- Allocate/Release
- Get/Set Attribute
- Polled Verbindung
- Cyclic Verbindung (mit und ohne Ack)

1.1.1 Get Attribute

	C/I/A
Max Instance of Class ID 0x01	1/0/2 = 1
Vendor ID	1/1/1 = 999
Device Type	1/1/2 = 12
Product Code	1/1/3 = 700
Revision	1/1/4 = 1.0
Produced Connection Size (Polled)	5/2/7 = Länge von ProducedData (ANY)
Consumed Connection Size (Polled)	5/2/8 = Länge von ConsumedData (ANY)
Produced Connection Size (Cyclic)	5/4/7 = Länge von ProducedData (ANY)
Consumed Connection Size(Cyclic)	5/4/7 = Länge von ConsumedData (ANY)

1.1.2 Set Attribute

	C/I/A
Expected Packet Rate	5/1/9
Expected Packet Rate (Polled)	5/2/9
Expected Packet Rate (Cyclic)	5/4/9
Acknowledge Handler Object	2B/1/1

Hinweis:

Die Set Attribute Aufträge werden positiv quittiert und nicht weiter verwendet.

1.1.3 Freigaben

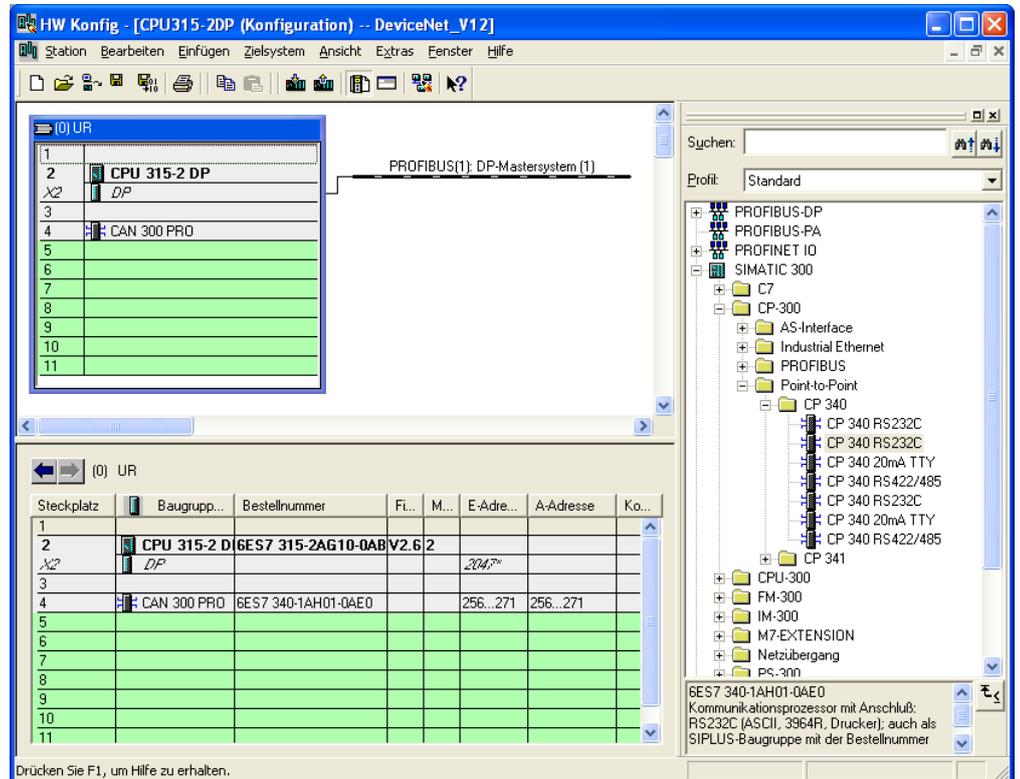
Getestet wurden die Hantierungsbausteine an einem Allen-Bradley Master (Scanner) und an einem Omron Scanner.

Desweiteren sind bereits Inbetriebnahmen mit einem Adept- und einem Kuka-Roboter durchgeführt worden.

Sollten Sie ein anderes Gerät als die oben aufgeführten als Scanner einsetzen, so können wir für die einwandfreie Funktionalität nicht garantieren, wir unterstützen Sie aber gerne bei der Erstinbetriebnahme.

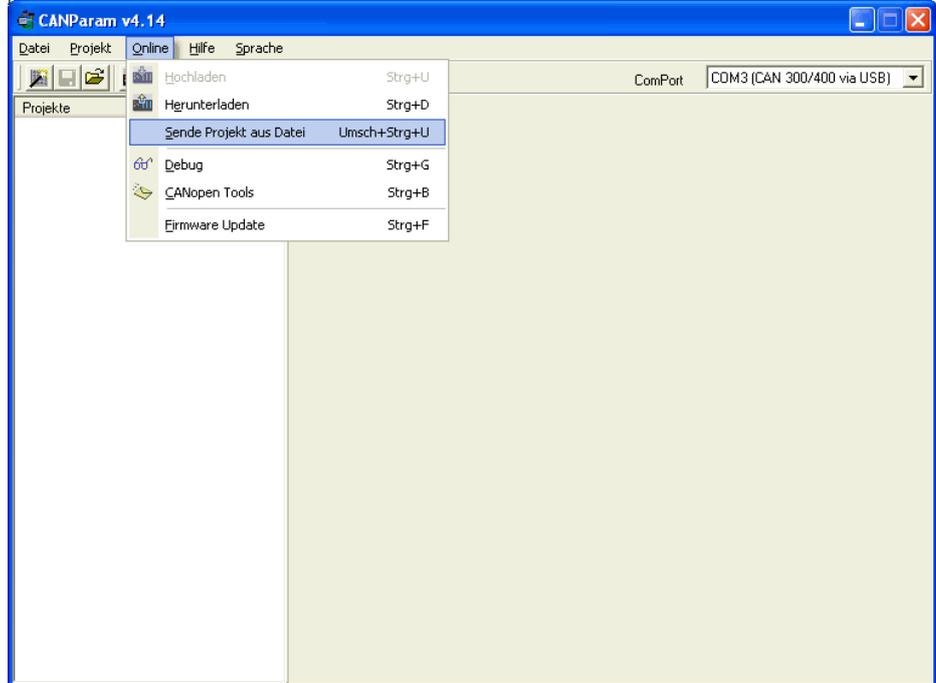
1.2 Aufbau SPS

Zur Verwendung kommt im Anwendungsbeispiel eine S7-300 CPU315-2DP sowie eine CAN 300 PRO Baugruppe.



2 Projektierung der CAN Baugruppe

Zur Verwendung des Beispielpogrammes muss das beiliegende CAN-Projekt „DeviceNet_Slave.PAR“ mit der CANParam Software (ab Version 4.1x) in die CAN 300 PRO Baugruppe übertragen werden.



Das CAN-Projekt stellt die CAN 300 PRO Baugruppe in den DeviceNet Modus.

Die Adresse des DeviceNet Slaves und die Baudrate können nun durch den DIP-Schalter eingestellt werden. Die Adresse kann nur von 0-63 eingestellt werden. Die selbe Adresse muß ebenfalls am Hantierungsbaustein eingestellt werden.

DeviceNet Slave Adresse	2^6	+ 64
	2^5	+ 32
	2^4	+ 16
	2^3	+ 8
	2^2	+ 4
	2^1	+ 2
Baudrate	2^2	+ 4
	2^1	+ 2
	2^0	+ 1



Baudraten:

0	1	2	3	4	5	6	7
10K	50K	100K	125K	250K	500K	800K	1M

3 Programmierung in der SPS

3.1 Übersicht

Das Beispiel enthält einen Hantierungsbaustein FB 99, der die gesamte DeviceNet Protokollverarbeitung durchführt.

Der FB 99 muss im Zyklus aufgerufen werden. Bei jedem Aufruf des FB werden Telegramme von der CAN 300 PRO Baugruppe abgeholt, verarbeitet und ggf. Antworten versendet.

Je nach Länge des SPS-Zyklus und nach Telegramm-Aufkommen am CAN-Bus kann der FB aber auch mehrere Telegramme hintereinander verarbeiten. Mit dem Parameter MaxComRequests kann die maximale Anzahl der zu verarbeitenden Telegramm festgelegt werden.

3.2 FB 99 „DN-Adapter“

Parameter	Richtung	Typ	Beispiel
Init	IN	BOOL	FALSE
EnableFB	IN	BOOL	TRUE
Base	IN	INT	256
MacID	IN	INT	10
ProducedData	IN	ANY	P#A 50.0 BYTE 32
ConsumedData	IN	ANY	P#E 50.0 BYTE 32
Cyclic	IN	BOOL	FALSE
CycAck	IN	BOOL	FALSE
ExpectedPacketRate	IN	TIME	T#1s
MaxComRequests	IN	INT	20
MasterRun	OUT	BOOL	M 50.0
DataExchange	OUT	BOOL	M 50.1
Allocate	OUT	BOOL	M 50.2
SendBusy	OUT	BOOL	M 50.3
Done	OUT	BOOL	M 50.4
RcvBusy	OUT	BOOL	M 50.5
NewData	OUT	BOOL	M 50.6
Error	OUT	BOOL	M 50.7
RetVal	OUT	INT	MW 52

Init	Im Anlauf (OB 100) mit TRUE aufrufen um die Grundinitialisierung durchzuführen, im Zyklus immer mit FALSE aufrufen
EnableFB	Freigabe-Flag zur Aktivierung des Bausteins
Base	Adresse der CAN 300 PRO Baugruppe
MacID	DeviceNet-Adresse des Slaves, muß zwingend mit der DIP-Switch Einstellung übereinstimmen.
ProducedData	Any-Pointer auf die Empfangsdaten
ConsumedData	Any-Pointer auf die Sendedaten
Cyclic	FALSE: Polled Betrieb / TRUE: Cyclic Betrieb
CycAck	Cyclic-Verbindung Acknowledged
ExpectedPacketRate	für Cyclic Betrieb und Timeoutüberwachung
MaxComRequests	Anzahl der verarbeiteten CAN-Telegramme pro Aufruf des FB 99

MasterRun	Master ist im Run und er sendet Daten
DataExchange	Kommunikation aktiv
Allocate	Verbindung aufgebaut (liegt nur einen Zyklus an)
SendBusy	Senden aktiv
Done	Senden abgeschlossen
RcvBusy	Empfangen aktiv
NewData	Empfangen abgeschlossen
Error	Ein Fehler ist aufgetreten (liegt nur einen Zyklus an)
RetVal	Fehlernummer, siehe Kapitel 3.5

Hinweis:

Der Hantierungsbaustein FB 99 ruft die FCs 65, 66, 67 absolut auf. Die FC-Nummern können somit nur Verändert werden, wenn der FB 99 angepasst wird.

3.3 Beispiel OB 100

Im Anlauf OB 100 muß der FB 99 einmalig mit dem Parameter Init = TRUE aufgerufen werden, um eine Grundinitialisierung durchzuführen.

Aufrufbeispiel:

```
CALL FB    99 , DB99
  Init      :=TRUE
  EnableFB  :=TRUE
  Base      :=256
  MacID     :=10
  ProducedData :=
  ConsumedData :=
  Cyclic    :=
  CycAck    :=
  ExpectedPacketRate:=
  MaxComRequests :=
  MasterRun :=
  DataExchange :=
  Allocate  :=
  SendBusy  :=
  Done      :=
  RcvBusy   :=
  NewData   :=
  Error     :=
  RetVal    :=
```

3.4 Beispiel OB 1

Aufrufbeispiel:

```
CALL FB 99 , DB99
  Init           :=FALSE
  EnableFB      :=TRUE
  Base          :=256
  MacID         :=10
  ProducedData  :=P#A 50.0 BYTE 32
  ConsumedData  :=P#E 50.0 BYTE 32
  Cyclic        :=FALSE
  CycAck        :=FALSE
  ExpectedPacketRate:=T#1S
  MaxComRequests :=20
  MasterRun     :=M50.0
  DataExchange  :=M50.1
  Allocate      :=M50.2
  SendBusy      :=M50.3
  Done          :=M50.4
  RcvBusy       :=M50.5
  NewData       :=M50.6
  Error         :=M50.7
  RetVal        :=MW52

//save allocate
  U    M    50.2
  S    M    51.2
//save error
  U    M    50.7
  S    M    51.7
//save retval
  L    MW   52
  L    0
  ==I
  SPB  M000
  L    MW   52
  T    MW   54
M000: NOP  0
```

3.5 Rückgabeparameter RETVAL

Der Rückgabeparameter RETVAL des Funktionsbausteins kann sowohl funktionspezifische Fehler enthalten oder Fehlernummern der Siemens Systemfunktionsbausteine SFC 58, SFC 59 und SFC 20.

Fehlercodes der DeviceNet-Hantierung:

- 80D1: Telegramm mit Länge > 8 empfangen
- 80D5: Scanner versucht nicht implementiertes Attribut zu lesen
- 80D6: Scanner versucht nicht implementiertes Attribut zu schreiben
- 80DA: Timeout ($3 \cdot \text{ExpectedPacketRate}$), Scanner meldet sich nicht (Poll)

4 Diagnose

4.1 Prozessabbild inder SPS

Die CAN 300 PRO Baugruppe belegt 16 Bytes im Eingangs- und Ausgangs-Prozessabbild. Der Inhalt des Ausgangs-Prozessabbildes wird nicht verwendet.

Der Inhalt des Eingangs-Prozessabbildes kann vom Anwender in der Applikation zu Informationszwecken verwendet werden:

Byte	Bedeutung
0	Baugruppenstatus allgemein, CAN Sammelfehler Anzeige
1	CAN-Controllers Status (Register des CAN-Controllers)
2	FIFO Status Bits (Send & Receive)
3	CAN-Controller: TX-Fehlerzähler
4	CAN-Controller: RX-Fehlerzähler
5	CANopen: Masterstatus
6	CANopen: Belegung SDO-Auftragsfächer
7	CANopen: Anzahl Nodes in Operational
8	Node-ID bei Verwendung des Bitfilters oder vom Master
9	<i>reserviert</i>
10	<i>reserviert</i>
11...15	<i>intern verwendet</i>

Zugriffe auf das Eingangsabbild können nur mit den Peripherie-direktzugriffsbefehlen durchgeführt werden: L PEB, L PEW

4.1.1 Byte 0: Baugruppenstatus

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CAN-Controller Sammelfehler	Baugruppe ist CAN 300 PRO						Baugruppe parametrierung und läuft

Bit 0: Die CAN 300 PRO Baugruppe hat die Projektierung verarbeitet und ist betriebsbereit.

Bit 6: Dieses Bit ist immer 1 um die CAN 300 PRO erkennen zu können.

Bit 7: Sammelfehlerbit für Fehler am CAN-Controller, genauere Auskunft über die Fehlerursache ist im Byte 1 zu erkennen.

4.1.2 Byte 1: Fehler-Status (EFLG) des CAN-Controllers

	RX1OVR	RX0OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN
	bit 7							bit 0
bit 7	RX1OVR: Receive Buffer 1 Overflow Flag - Set when a valid message is received for RXB1 and CANINTF.RX1IF = 1 - Must be reset by MCU							
bit 6	RX0OVR: Receive Buffer 0 Overflow Flag - Set when a valid message is received for RXB0 and CANINTF.RX0IF = 1 - Must be reset by MCU							
bit 5	TXBO: Bus-Off Error Flag - Bit set when TEC reaches 255 - Reset after a successful bus recovery sequence							
bit 4	TXEP: Transmit Error-Passive Flag - Set when TEC is equal to or greater than 128 - Reset when TEC is less than 128							
bit 3	RXEP: Receive Error-Passive Flag - Set when REC is equal to or greater than 128 - Reset when REC is less than 128							
bit 2	TXWAR: Transmit Error Warning Flag - Set when TEC is equal to or greater than 96 - Reset when TEC is less than 96							
bit 1	RXWAR: Receive Error Warning Flag - Set when REC is equal to or greater than 96 - Reset when REC is less than 96							
bit 0	EWARN: Error Warning Flag - Set when TEC or REC is equal to or greater than 96 (TXWAR or RXWAR = 1) - Reset when both REC and TEC are less than 96							

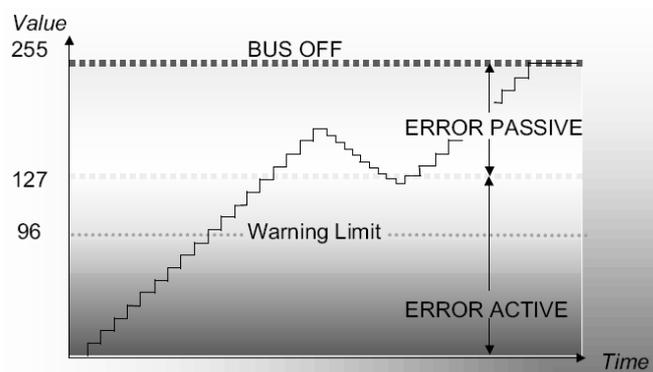
4.1.3 Byte 2: FIFO-Status Bits

Bit 7	Bit 6	Bit 5	Bit 4
Send-FIFO (high) halb voll	Send-FIFO (high oder low) Overflow	Send-FIFO (low) halb voll	Send-FIFOs (high & low) ganz leer

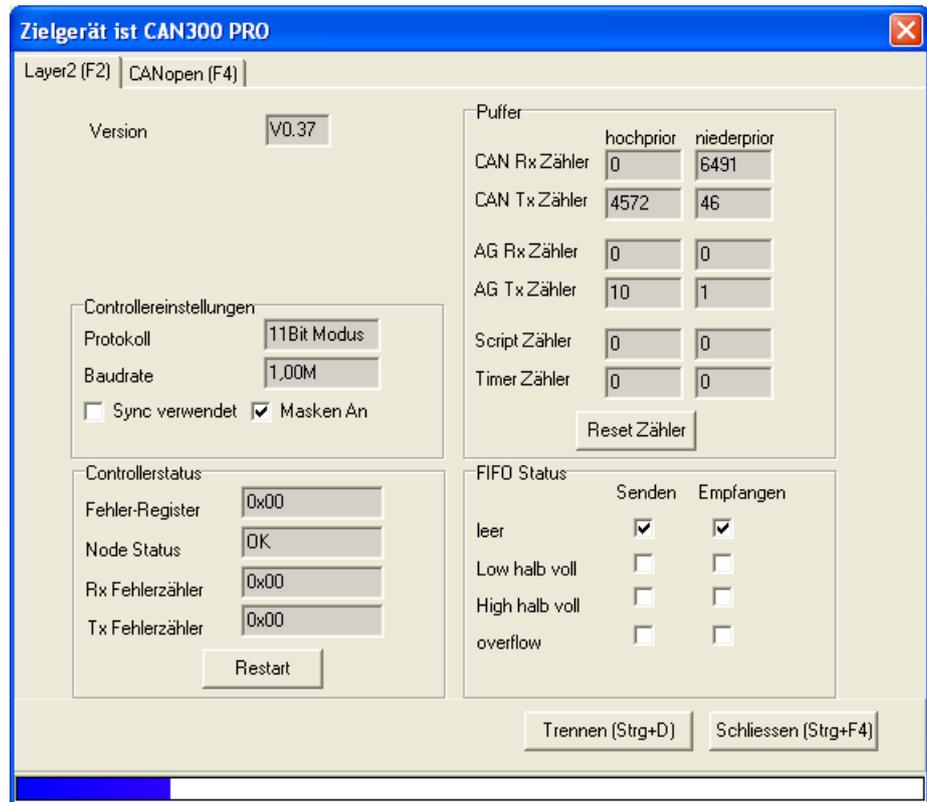
Bit 3	Bit 2	Bit 1	Bit 0
Receive-FIFO (high) halb voll	Receive -FIFO (high oder low) Overflow	Receive-FIFO (low) halb voll	Receive-FIFOs (high & low) ganz leer

4.1.4 Byte 3/4: CAN-Controller Tx/Rx Fehlerzähler

Der Fehlerzähler wird bei jedem fehlerhaft versendetem oder empfangenen CAN-Telegramm hochgezählt. Wenn ein CAN-Telegramm korrekt übertragen wurde, wird der Fehlerzähler wieder heruntergezählt. Wenn der Zähler größer als 96 ist, geht der CAN-Controller in den Modus „Warning“. Sollte der Fehlerzähler 127 übersteigen geht der CAN-Controller in „Error-Passiv“.



4.2 CANParam Debug-Screen



Mit dem Button „Verbinden“ wird der Überwachungsmodus aktiviert. Nochmaliges Drücken des Buttons unterbricht die Verbindung wieder.

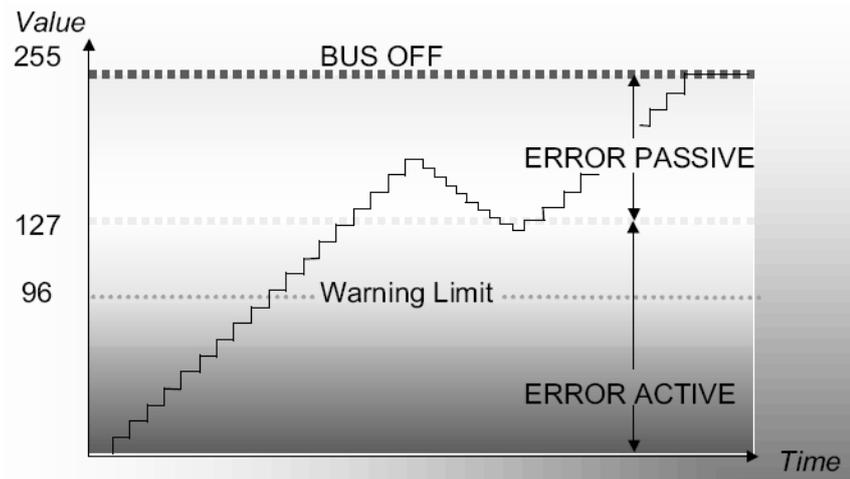
Folgende Informationen liefert der Debug-Dialog:

- Version** Versionsnummer des Betriebssystems
- Protokoll** projektiertes CAN-Protokoll (11Bit/29Bit)
- Baudrate** aktive CAN-Baudrate
- Controllerstatus** Inhalt des CAN-Status-Registers:
- Fehlerregister** Inhalt des CAN-Fehler Registers EFLG
- Node-Status** Inhalt des CAN-Status-Registers (s.o.):
"OK", "Warning", "Passiv", "Bus Off"
- Rx Fehlerzähler** Fehlerzähler CAN-Empfang
- Tx Fehlerzähler** Fehlerzähler CAN-Senden

!
Node Status sollte immer auf „OK“ stehen, damit eine störungsfreie CAN-Datenübertragung möglich ist.

!
Die Fehlerzähler müssen auf „0“ stehen, sonst ist die Datenübertragung am CAN-Bus gestört.

Hinweis: Die Sende und Empfangszähler werden vom CAN-Controller hochgezählt, wenn das Senden oder der Empfang eines Telegrammes fehlgeschlagen ist. Sobald ein Telegramm korrekt gesendet, oder empfangen wurde, wird der entsprechende Zähler wieder heruntergezählt. Diese Zähler sollten bei einem korrekt arbeitenden CAN-Bus immer auf 0 stehen!



Die Informationen über die Puffer und die FIFOs sind nur im Layer 2 Modus relevant. Im CANopen Master Modus übernimmt die Firmware die Kontrolle über die Puffer.

Puffer

Anzeige der Anzahl der empfangenen bzw. gesendeten CAN-Telegramme in den Puffern am CAN-Bus, zur SPS und intern

Hinweis: Die CAN 300 PRO Baugruppe hat Empfangs- und Sendepuffer von jeweils 400 Telegrammen (niederprior) und 20 Telegrammen (hochprior). Die Zähler zeigen an, wie viele Telegramme verarbeitet wurden.

Zwischen den Rx-/Tx-Zählerpaaren sollte nie eine grosse Differenz entstehen. Ist dies doch der Fall, so werden die CAN-Telegramme nicht schnell genug von der SPS abgeholt, oder zu schnell von der SPS gesendet.

Wenn bei einem vollen FIFO weitere Telegramme empfangen oder gesendet werden, so wird das Fehlerbit FIFO-Overflow gesetzt und das älteste Telegramm im FIFO gelöscht.

FIFO-Status Anzeige des Füllgrades der FIFOs

In der SPS kann der FIFO-Status über das Peripheriebyte 2 ausgewertet werden. Die FIFOs können mit dem Hantierungsbaustein FC 67 CANCTRL gelöscht werden.