

# MANUAL

## CAN - BUS

### für Servoverstärker

DS-xx / DPC-xx

BAMOCAR-xx

BAMOBIL-xx



Industrie Elektronik  
G m b H

Hans-Paul-Kaysser-Straße 1  
71397 Leutenbach-Nellmersbach

Tel: 07195 / 92 83 - 0

[contact@unitek.eu](mailto:contact@unitek.eu)

[www.unitek.eu](http://www.unitek.eu)

**Ausgabe / Version**

**2022/ V1**

## Inhaltsverzeichnis

<b>1 Sicherheit</b> .....	<b>2</b>
1.1 Sicherheitshinweise.....	2
1.2 Vorschriften und Richtlinien.....	2
<b>2 Allgemeines</b> .....	<b>4</b>
2.1 Logikfunktionen.....	4
<b>3 CAN-BUS Verbindungen</b> .....	<b>5</b>
3.1 Verbindungen.....	5
<b>4 Software</b> .....	<b>7</b>
4.1 Formatbeschreibung .....	7
4.2 Kopf-Feld .....	8
4.3 COB-ID-Bits (CAN OBJECT ID) .....	8
4.4 RTR-Bit (REMOTE TRANSMISSION REQUEST).....	8
4.5 DLC-Bits (DATA LENGTH CODE).....	8
4.6 Datenfeld.....	9
4.7 REGID.....	9
4.8 Daten .....	9
<b>5 Beispiele</b> .....	<b>10</b>
5.1 CAN-BUS-Daten empfangen.....	10
5.2 CAN-DATEN senden vom DSxx und BAXX-Servo auf den CAN-BUS.....	10
5.3 Senden vom Master auf den CAN-BUS zum DS-Servo .....	11
5.4 Senden vom DS-Servo auf den CAN-BUS .....	14
<b>6 Einheiten</b> .....	<b>20</b>
6.1 Umrechnung der Maßeinheit.....	20

## 1 Sicherheit

### 1.1 Sicherheitshinweise

#### **Achtung:**

**Die Inbetriebnahme-Anleitung ist nur in Verbindung mit dem Hardware-Manual DS und der Software-Beschreibung NDrive zu verwenden!!**



Dieses Manual muss vor der Installation oder Inbetriebnahme sorgfältig durch Fachpersonal gelesen und verstanden werden. Bei Unklarheiten ist der Hersteller oder Händler zu kontaktieren.

### 1.2 Vorschriften und Richtlinien

Die Geräte und die dazugehörigen Komponenten sind nach den örtlichen gesetzlichen und technischen Vorschriften zu montieren und anzuschließen.

EU-Richtlinie	2004/108/EG, 2006/95/EG, 2006/42/EG EN 60204-1, EN292, EN50178, EN60439-1, EN61800-3, ECE-R100 ISO 6469, ISO 26262, ISO 16750, ISO 20653, ISO12100
IEC/UL:	IEC 61508, IEC364, IEC664, UL508C, UL840
VDE Vorschrift/TÜV-Vorschrift:	VDE100, VDE110, VDE160
Vorschrift der Berufsgenossenschaft:	VGB4

#### **Der Anwender muss sicherstellen:**

- dass nach einem Ausfall des Gerätes
- bei Fehlbedienung
- bei Ausfall der Regel- und Steuereinheit usw.

der Antrieb in einen sicheren Betriebszustand geführt wird.

Maschinen, Anlagen und Fahrzeuge sind außerdem mit geräteunabhängigen Überwachungs- und Sicherheitseinrichtungen zu versehen.

Nicht geerdete Systeme (z.B. Fahrzeuge) müssen mit unabhängigen Isolationswächtern gesichert werden.

Es darf keine Gefahr für Menschen und Sachen entstehen!!!



## Montagearbeiten

- nur im spannungslosen Zustand
- nur von geschultem Elektro-Fachpersonal

## Installationsarbeiten

- nur im spannungslosen Zustand
- nur von geschultem Elektro-Fachpersonal
- Sicherheitsvorschriften beachten

## Einstell- und Programmierarbeiten

- nur von Fachpersonal mit Kenntnissen in elektronischen Antrieben und Software
- Programmierhinweise beachten
- Sicherheitsvorschriften beachten
- bei begrenztem Fahrweg müssen die Wegüberwachungen aktiv sein.

## CE

Bei Einbau in Maschinen, Anlagen und Fahrzeugen ist die Aufnahme des bestimmungsgemäßen Betriebs des Gerätes so lange untersagt, bis festgestellt wurde, daß die Maschine, die Anlage oder das Fahrzeug den Bestimmungen der EG-Maschinenrichtlinie 2006/42/EG, der EMV-Richtlinie 2004/108/EG und ECE-R100 entspricht.

Die EG-Richtlinie 2004/108/EG mit den EMV-Normen EN61000-2 und EN61000-4 wird unter den vorgegebenen Installations- und Prüfbedingungen (siehe Kapitel CE-Hinweise) eingehalten.

Eine Herstellererklärung kann angefordert werden.

Die Einhaltung der durch die EMV-Gesetzgebung geforderten Grenzwerte liegt in der Verantwortung des Herstellers des Fahrzeugs, der Anlage oder Maschine.

## QS

Die Geräte sind über ihre Seriennummer mit den Prüfdaten beim Hersteller für 5 Jahre archiviert. Die Prüfprotokolle können angefordert werden.

# Allgemeines

## 2 Allgemeines

### 2.1 Logikfunktionen

Das serielle Datenbussystem CAN (Controller Area Network) wurde ursprünglich für die Automobilbranche entwickelt. Inzwischen hat der CAN-BUS ein breites Anwendungsfeld im Anlagen- und Maschinenbau. CAN ist als ISO11898 international standardisiert. CAN genügt den besonders hohen Sicherheitsanforderungen von hochverfügbaren Maschinen und medizinischen Geräten. Hohe Übertragungsraten und günstige Anschlusskosten sprechen für den CAN-BUS.

Bei der CAN-Datenübertragung werden keine Stationen adressiert, sondern der Inhalt einer Nachricht wird durch einen netzweiten eindeutigen Identifier gekennzeichnet. Der Identifier legt auch die Priorität der Nachricht fest.

Durch die inhaltbezogene Adressierung wird eine hohe System- und Konfigurations-Flexibilität erreicht. Es lassen sich sehr einfach weitere Geräte dem Netz hinzufügen.

Das CAN-BUS Interface ist in allen digitalen UNITEK-Geräten als **Slave** installiert.

Es ist zum Anschluss an einen CAN-BUS Master bestimmt.

Das Interface ist optoentkoppelt.

Die primäre Versorgung erfolgt über einen DC/DC-Wandler.

#### Der UNITEK-CAN-BUS kann folgende Funktionen übertragen:

Beispiele vom Master (CNC/SPS) zum Slave (DRIVE-DS) / (Empfangen, Receive):

Logikfunktionen	Sollwerte	Parameter	
Freigabe	Momenten-Sollwert	Regelparameter	
Referenzfahrt	Drehzahl-Sollwert	Einstellungen	
Start, Stop	Position-Sollwert		
	Stromgrenzen		

Beispiele vom Slave (DRIVE-DS) zum Master (CNC/SPS) / (Senden, Transmit):

Logikfunktionen	Istwerte	Parameter	Meldungen
RUN	Momenten-Istwert	Regelparameter	Status-Meldungen
ENABLE	Drehzahl-Istwert	Einstellungen	Fehler-Meldungen
POS	Position-Istwert		
Endschalter			

Die Adressen (REGID) stehen in der Parameter-Übersicht (siehe NDrive Manual)

z.B. Drehzahl-Sollwert (SPEED\_CMD) = REGID 0x31 <Wert in Hex>.

# CAN-BUS Verbindungen

## 3 CAN-BUS Verbindungen

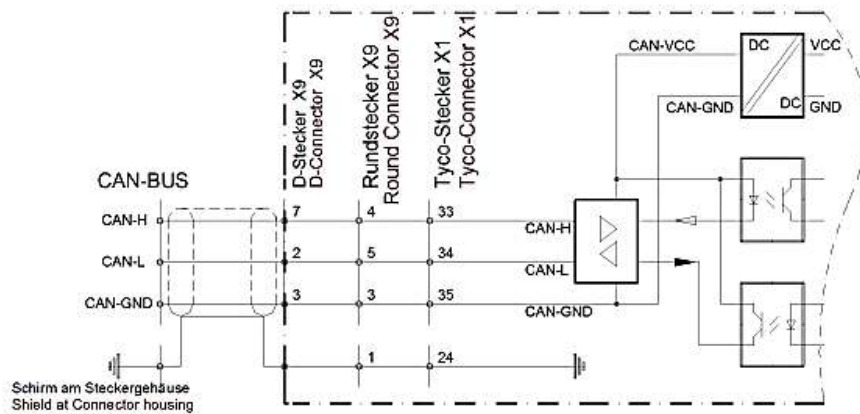
### 3.1 Verbindungen

Der CAN-BUS ist die digitale Verbindung zur CNC-Steuerung (CAN-Master).  
 Programmierung und Bedienung mittels Bedienfeld über CAN-BUS.  
 Interface nach ISO 11898-2.

Verbindungshardware:

Wellenwiderstand	120 Ω
Leiterwiderstand (Schleife)	160 Ω/km
Betriebskapazität (800 Hz)	<60 nF/km

### Eingangsbeschaltung



Steckerbelegung:  
 siehe Geräte-Manual  
 Kabelfarben (empfohlen)  
 LiYCY 4x0.25 + Schirm  
 CAN-GND      weiss  
 CAN-H        grün  
 CAN-L        gelb

(Achtung: Farben können unterschiedlich sein)



Abbildung 4-1

**CAN-BUS isoliert / CAN Gnd auf gemeinsames Potential bringen**

### CAN-BUS Verbindung mit mehreren Servo-Verstärkern DS- (Slave)

Bei abweichenden Gerätetypen die Steckerbelegung beachten

Adresse xx      Adresse xx



(Beispiel):

(Geräte-Manual)

Adresse xx

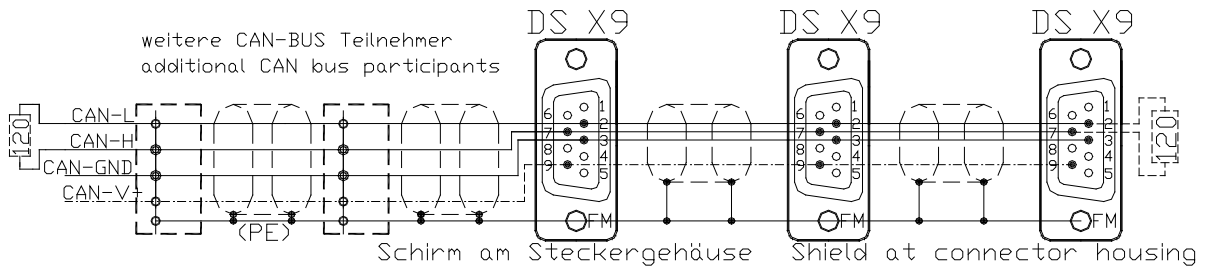


Abbildung 4-2

## Terminierungswiderstand

Der Leitungs-Anschlusswiderstand ( $R = 120 \Omega$ ) muss am ersten und letzten BUS-Teilnehmer zwischen CAN-H und CAN-L angebracht werden

## Spannungsversorgung

Die Spannungsversorgung vom CAN-BUS erfolgt intern über einen DC/DC-Wandler.

## Einstellung CAN-BUS

Die Sende-Adressen für Empfang und Senden und die Übertragungsrate werden beim PC-Programm NDrive im Parameterfeld CAN-Setup eingegeben.

Adresse	Kurzzeichen	Grundwert (default)	Bereich
Empfangs-Adresse (Slave)	RPD01	0x201	0x201 bis 0x27F
Sende-Adresse (Slave)	TPD01	0x181	0x181 bis 0x1FF

Übertragungsrate NBT	Einstellwert BTR	Leitungslänge max.	
1000 kBaud	0x4002	20 m	
500 kBaud	0x4025	70 m	
625 kBaud	0x4014	70 m	
250 kBaud	0x405c	100 m	
100 kBaud	0x4425	500 m	

## 4 Software

### 4.1 Formatbeschreibung

Das Software-Format ist für die optimale Kommunikation mit den CNC-Maschinensteuerungen und CAN-Modulen der Firma Labod electronic ausgelegt.

Dieses Format entspricht nicht CANopen.

Die Übertragungsrate (Baud rate) ist programmierbar.

Der UNITEK Standard beträgt 500 kB/s (Labod 615 kB/s).

Die Geräte UNITEK DSxx und BAXx können in ein CANopen Netzwerk (TPDO1, RPDO1) als Slave eingefügt werden.

#### Zahlenformat

Parameterwert und Parameternummer im Little-Endian-Format (Intel-Format)

Bit7 bis 0 / Bit15 bis 8 / Bit23 bis 16 / Bit31 bis 24

#### CAN Format

Das CAN-Protokoll ist ein 3 oder 5 Byte Daten-Paket beim Empfang, 4 oder 6 beim Senden.

Es können auch bis zu 8 Byte Daten-Pakete empfangen werden, jedoch wird dieses dann als 5 Byte Daten-Paket ausgewertet. Der Identifier ist 11Bit breit. Es beinhaltet den **COB-Identifier**, die **RTR-Funktion** (Remote Transmission Request) und die **DLC-Information** (Data Length Code).

Das Byte 0 vom Datenfeld ist für die REGID-Index (Parameternummer).

Das zweite bis fünfte Byte (Byte 1 bis Byte 4) beinhaltet die Daten der REGID-Index (Parameterwert).

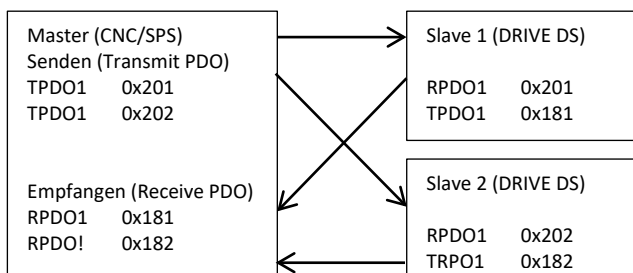
Bereich	Kopf			Datenfeld				
	COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>	11 Bit	0	Länge	REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24

#### Master-Slave Verbindung

Um die Konfiguration zu vereinfachen, wurde in CANopen ein Predefined-Master/Slave-Connection-Set spezifiziert. Für Netzwerke mit einem Master und bis zu 127 Slaves, bietet Teilnehmern diese Zuordnung von COB-Identifier eine einfache Lösung für ein CANopen Netzwerk. Die Informationsverteilung erfolgt nur vom Master aus. Eine direkte Kommunikation von Slave zu Slave ist nicht möglich.

COB - Identifier										
Service							Node-ID			
10						7			6	0
Beispiel 0x181										
0	0	1	1	0	0	0	0	0	0	1
1			8				1			

Bevorzugte Objekte (Slave) sind  
 TPDO1 (0x201 bis 0x27F) und  
 RPDO1 (0x181 bis 0x1FF)  
 Die Objekte TPDO2..4 und RPDO2..4  
 können auch verwendet werden.



Verbindung von Master zu Slave



## 4.2 Kopf-Feld

Bereich	Kopf			Datenfeld				
	COB-ID	RTR	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>	11 Bit	0	Länge	REGID	B7 bis 0	B15 bis 8	B23 bis 16	B31 bis 24

## 4.3 COB-ID-Bits (CAN OBJECT ID)

Der Vorgabewert (default) ist bei CANopen für TPDO1=0x181 und für RPDO1=0x201.

COB - Identifier											Objekt	
Service				Node - ID								
0	0	1	1	0	0	0	0	0	0	1	TPDO1 0x181-0x1FF	
1				8				1				
	1	0	0	0	0	0	0	0	0	1	RPDO1 0x201-0x27F	
2				0				1				

Die Adresse kann geändert werden indem eine direkte Sende-Adresse im Servo-Verstärker (DSxx, BAxx) für Empfangen (CAN-ID-Rx 0x68) und für Senden Feld CAN-Setup in NDrive eingegeben wird. Die Adressen von Tx-ID und Rx-ID können auch direkt über CAN geändert werden (siehe Bsp. 1)

## 4.4 RTR-Bit (REMOTE TRANSMISSION REQUEST)

Der Wert für RTR ist immer auf 0 gesetzt / RTR wird nicht verwendet.

## 4.5 DLC-Bits (DATA LENGTH CODE)

Mit den DLC-Bits wird die Größe des Datenfeldes bestimmt.

Empfangen: Wert 0x03 entspricht REGID plus 2 Byte (16 Bit)  
 Wert 0x05 entspricht REGID plus 4 Byte (32 Bit)

Senden: Wert 0x04 entspricht REGID plus 2 Byte plus Dummy (16 Bit)  
 Wert 0x06 entspricht REGID plus 4 Byte plus Dummy (32 Bit)

## 4.6 Datenfeld

Die Länge des Datenfelds für im Servo empfangene Nachrichten beträgt 3 oder 5 Byte.  
 Die oberen Datenbytes werden beim Empfang registriert, aber nicht berücksichtigt.  
 Die Nachricht für das Senden vom Servo auf den CAN-BUS ist 4 oder 6 Byte breit.

## 4.7 REGID

Das erste Byte ist für die REGID-Index (Parameternummer) vorgesehen.  
 Es können bis zu 254 Register bestimmt werden.  
 Die wichtigsten Parameter-Indexe stehen in der REGID-Liste (siehe Manual NDrive).

## 4.8 Daten

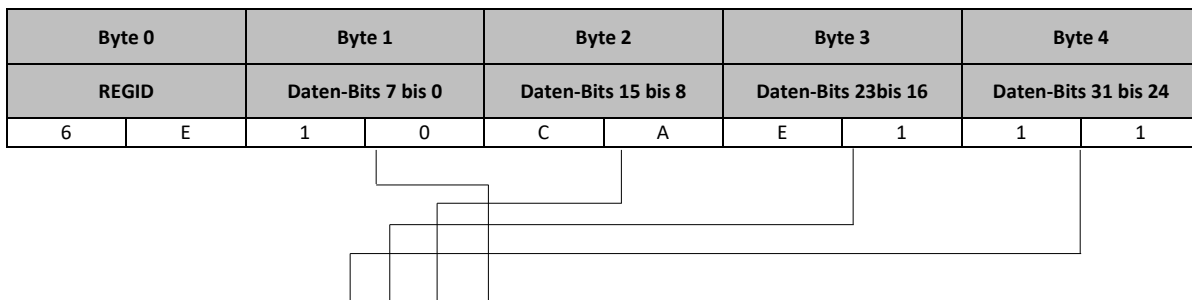
Die Datenlänge wird im Feld DLC-Bits vorgegeben (16 oder 32 Bits).  
 Byte 2 bis Byte 5 sind für die 32 Bit Register-Daten (4 Byte) bestimmt.  
 Byte 2 bis Byte 3 sind für die 16 Bit Register-Daten (2 Byte) bestimmt.

### Beispiel für Datenfeld

Positions-Sollwert für Num 300010000

Funktion	Hex-Wert
Sende-Adresse für Empfangen	0x201
Datenlänge 4 Byte	DLC=5
REGID für Positions-Sollwert (POS_SOLL)	0x6E
Datenlänge 4 Byte	DLC=5
Daten für Positions-Sollwert Num 300010000	0x11E1CA10

### Dateneingabe



Daten = **0x11E1CA10** (entspricht der Num. Position 300010000)

Die Eingabe ist im Little-Endian-Format (Intel-Format)

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 2	0x201	5	0x6E	0x10	0xCA	0xE1	0x11

# Beispiele

## 5 Beispiele

### 5.1 CAN-BUS-Daten empfangen

Sende-Adresse am DS-Servo	COB-ID	(default = 0x201)
Datenformat	DLC	(3, 4, 5)
Parameter	Byte 0	(REGID – siehe Parameterliste)
Parameter-Inhalt	Byte 1 bis Byte 4	

#### Beispiele:

Sende-Adresse ändern über CAN	siehe Beispiel 1
Regler sperren (nicht Freigabe)	siehe Beispiel 2
Drehzahl-Sollwert	siehe Beispiel 3
Positions-Sollwert	siehe Beispiel 4
Drehmoment-Sollwert	siehe Beispiel 5
Parameterwert	siehe Beispiel 6
EEPROM schreiben	siehe Beispiel 7

### 5.2 CAN-DATEN senden vom DSxx und BAXX-Servo auf den CAN-BUS

Generell für die Aufforderung zum Senden vom DS-Servo gilt:

Datenfeld:	Byte 0 =	0x3D	Parameter Sendeaufforderung
(DLC = 3)	Byte 1 =	REGID Value	Inhalt dieser REGID
	Byte 2 =	0x??	Zeitintervall

1. Einmal senden: (siehe Beispiel 8)

Datenfeld:	Byte 0 =	0x3D	Parameter Sendeaufforderung
(DLC = 3)	Byte 1 =	0xA8	Inhalt dieser REGID
	Byte 2 =	0x00	Einmaliges senden

2. Zyklisch senden: (siehe Beispiel 9)

Datenfeld:	Byte 0 =	0x3D	Parameter Sendeaufforderung
(DLC = 3)	Byte 1 =	0xA8	Inhalt dieser REGID
	Byte 2 =	0x0A	alle 10 ms senden (0 bis 254 ms)

**Hinweis:** Byte 2 = 0xFF Zyklisch senden stoppen

3. Statusmeldung nach Ereignis anfordern: (siehe Beispiel 10)

Datenfeld:	Byte 0 =	0x51	REGID für Daten nach Ereignis
(DLC = 3)	Byte 1 =	0x10	Aktivierung über Bit 4
	Byte 2 =	0x00	Don't care

# Beispiele

## 5.3 Senden vom Master auf den CAN-BUS zum DS-Servo

### Beispiel 1: Sende-Adresse ändern über CAN

Die Adresse für Empfangen (Slave) in einen neuen DSxx, BAXx-Servo ist 0x201 (default).  
Diese Adresse soll geändert werden in 0x210.  
Der REGID-Index für die Empfangs-ID zur Konfiguration dieser Adresse ist 0x68 (CAN-ID-RX).

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für CAN-Rx Adresse	0x68
Wert für neue CAN-Rx Adresse	0x0210

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 1</b>	0x201	3	0x68	0x10	0x02	---	---

### Sende-Adresse ändern im PC-Programm NDrive



#### Eingaben:

NBT	Übertragungsrate (kBaud)
Rx-ID	Empfangs-Adresse im DS (default 0x201)
Tx-ID	Sende-Adresse vom DS (default 0x181)
T-Out	Zeitfehler-Überwachung

Abbildung 6-1

# Beispiele

## Beispiel 2: Regler sperren (nicht Freigabe)

## Nachricht zum Servo

### Funktion

Sende-Adresse zum Servo  
 Datenlänge 2 Byte  
 REGID für Sperre (MODE)  
 Wert für Sperre MODE BIT2

### Hex-Wert

0x201  
 DLC=3  
 0x51  
 0x0004

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 2</b>	0x201	3	0x51	0x04	0x00	---	---

## Beispiel 3: Drehzahl-Sollwert

## Nachricht zum Servo

### Funktion

Sende-Adresse zum Servo  
 Datenlänge 2 Byte  
 REGID für Drehzahl-Sollwert (SPEED\_SOLL)  
 Wert für 10 % Drehzahl Num. 3277

### Hex-Wert

0x201  
 DLC=3  
 0x31  
 0x0CCD (100 %  $\triangleq$  32767)

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 3</b>	0x201	3	0x31	0xCD	0x0C	---	---

## Beispiel 4: Positions-Sollwert

## Nachricht zum Servo

### Funktion

Sende-Adresse zum Servo  
 Datenlänge 4 Byte  
 REGID für Drehzahl-Sollwert (POS\_DEST)  
 Wert für Position 3000000

### Hex-Wert

0x201  
 DLC=5  
 0x6E  
 0x2DC6C0

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 4</b>	0x201	5	0x6E	0xC0	0xC6	0x2D	0x00

## Beispiel 5: Drehmoment-Sollwert

## Nachricht zum Servo

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Drehzahl-Sollwert (TORQUE-CMD)	0x90
Wert für 50 % Drehmoment Num 16380	0x3FFC

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 5</b>	0x201	3	0x90	0xFC	0x3F	---	---

## Beispiel 6: Einstell-Parameter

## Nachricht zum Servo

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Parameter Beschleunigung (ACC-Ramp)	0x35
Wert für 1000 ms Beschleunigung	0x03E8

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 6</b>	0x201	3	0x35	0xCD	0x0C	---	---

## Beispiel 7: EEPROM schreiben

## Nachricht zum Servo

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für EEPROM schreiben	0x84
EEPROM-Ebene 0 (EEPROM-Ebene1 = 0X0001)	0x0000

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 7</b>	0x201	3	0x84	0x00	0x00	---	---

# Beispiele

## 5.4 Senden vom DS-Servo auf den CAN-BUS

Alle Beispiele haben die default Sende-Adressen (Rx-ID=0x201 empfangen, Tx-ID=0x181 senden)

### Beispiel 8: Status-Meldung

### Einmaliges senden vom Servo

Um die Information einer bestimmten REGID zu erhalten, muss eine Sendeaufforderung an den Servo gesendet werden. In diesem Beispiel wird ein einmaliges Senden der REGID-Information angefordert.

#### Nachricht zum Servo für eine Sendeaufforderung:

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Daten aus dem Servo lesen und an CAN senden (READ)	0x3D
REGID für Status (KERN_STATUS)	0x40
Zeitintervall (Einmal senden)	0x00

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 8</b>	<b>0x201</b>	<b>3</b>	<b>0x3D</b>	<b>0x40</b>	<b>0x00</b>	---	---

#### Zurückgesendete Information vom Servo:

Funktion	Hex-Wert
Sende-Adresse vom Servo	0x181
Datenlänge 2 Byte	DLC=4
REGID für Status (KERN_STATUS)	0x40
Wert von KERN_STATUS (0x40)	0x0181

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 8</b>	<b>0x181</b>	<b>4</b>	<b>0x40</b>	<b>0x81</b>	<b>0x01</b>	<b>0x**</b>	

(Datenbereich Byte 1 bis Byte 4 im Little Endian Format)

Aktueller Zustand von Status = 0x0181:

Bit0	Antrieb freigegeben	(Ena)
Bit7	Positionsregelung	(P-N)
Bit8	Drehzahlregelung	(N-I)

# Beispiele

## Beispiel 9: Drehzahl-Istwert

## Zyklisches senden vom Servo

Für die zyklische Rücksendung wird das Register REGID\_READ mit einer Wiederholzeit geladen. In das Byte 2 wird eine Zykluszeit in ms für die Sendewiederholung im HEX-Format eingegeben (1 – 254 ms).

### Nachricht zum Servo für eine Sendeaufforderung:

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Daten aus dem Servo lesen und an CAN senden (READ)	0x3D
REGID für Drehzahl-Istwert (SPEED_IST)	0x30
Für Wiederholzeit 100 ms ist die Eingabe in Byte 2	0x64

### Hinweis:

Zum Stoppen des permanenten senden in Byte 2 0xFF

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 9</b>	<b>0x210</b>	<b>3</b>	<b>0x3D</b>	<b>0x30</b>	<b>0x64</b>	---	---

### Im Zeitintervall von 100 ms vom Servo zurückgesandte Information

Funktion	Hex-Wert
Sende-Adresse vom Servo	0x201
Datenlänge 2 Byte	DLC=4
REGID für Drehzahl-Istwert (SPEED_IST)	0x30
Wert von Drehzahl-Istwert 100 % (Num 32767)	0x7FFF

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 9</b>	<b>0x190</b>	<b>4</b>	<b>0x30</b>	<b>0xFF</b>	<b>0x7F</b>	<b>0x**</b>	---

### Hinweis:

Es können maximal 8 Zustandswerte konfiguriert werden, die zyklisch ihren Status senden.



# Beispiele

## Beispiel 10: Statusmeldung nach Ereignis      senden vom Servo (0x51 – BIT4)

### Aktivierung:

Durch setzen von Bit 4 in der REGID Adresse 0x51, wird das automatisierte Senden anhand einer bestimmten Konfiguration aktiviert. Das Signal das automatisiert versendet wird, ist der Gerätestatus (REGID 0x40). Das Versenden erfolgt bei einer Änderung vom Gerätestatus abhängig einer Konfigurationsmaske (Bit-Maske (REGID 0x52)).

### Konfiguration:

Die Konfiguration erfolgt über die Bit-Maske (REGID 0x52). Die Bit-Maske hat einen voreingestellten Wert von 0x0030.

D.h. bei einer Änderung von Status-Bit 12 (Cal) oder Status-Bit 13 (Tol) wird die komplette Status-Meldung (KERN\_STATUS Bit 0 bis Bit 15) auf den CAN-BUS gesendet.

### Sendeaufforderung zum Servo:

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Daten nach Ereignis (Ereignis-Trigger)	0x51
REGID für MODE BIT 4	0x10

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 10</b>	<b>0x201</b>	<b>3</b>	<b>0x51</b>	<b>0x10</b>	<b>0x00</b>	---	---

### Zurückgesendete Information vom Servo:

Als Beispiel wird bei einer Positionsfahrt die gewünschte Position erreicht und somit Bit 13 im Gerätestatus gesetzt (Tol). Somit wird das automatische Senden vom Gerätestatus (REGID 0x40) getriggert.

Funktion	Hex-Wert
Sende-Adresse vom Servo	0x181
Datenlänge 4 Byte	DLC=6
REGID für Status (KERN_STATUS)	0x40
Wert von KERN_STATUS (0x40)	0x0181

Bereich	Kopf		Datenfeld					
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
<b>Funktion</b>			REGID	b7 - 0	b15 - 8	b23 - 16	b31 - 24	b32 - 39
<b>Beispiel 10</b>	<b>0x181</b>	<b>6</b>	<b>0x40</b>	<b>0x81</b>	<b>0x31</b>	<b>0x00</b>	<b>0x00</b>	<b>0x**</b>

Aktueller Zustand von Status (KERN\_STATUS) = 0x3181:

Bit0	Antrieb freigegeben	(Ena)
Bit7	Positionsregelung	(P-N)
Bit8	Drehzahlregelung	(N-I)
Bit12	Kalibriert	(Cal)

# Beispiele

## Beispiel 10-1: Statusmeldung nach ausgewähltem Ereignis    senden von Servo

Mit der Konfigurationsmaske (REGID 0x52) wird der Ereignis-Trigger auf die zugeordneten Status-Bits geändert.

Zum Beispiel: Konfig.-Maske (0x52) = 0x20 entspricht Dauerstrom (Icns)  
Konfig.-Maske (0x52) = 0x12 entspricht Endschalter + und – (Lim+, Lim-)

### Trigger-Ereignis mit Konfig.-Maske (0x52) bestimmen.

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Konfig.-Maske	0x52
REGID für Status_Trigger-Auswahl (z. B. Endschalter)	0x12

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 10-1</b>	<b>0x201</b>	<b>3</b>	<b>0x52</b>	<b>0x12</b>	<b>0x00</b>	---	---

### Status nach ausgewähltem Status-Ereignis senden:

Eingestellter Wert für die Konfig.-Maske (0x52) ist 0x0012.  
Beim Belegen eines Endschalters (+ oder -) wird die komplette Status-Meldung (4 Byte) gesendet.

Funktion	Hex-Wert
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Daten nach Ereignis (Ereignis-Trigger)	0x51
REGID für MODE BIT 4	0x10

Bereich	Kopf		Datenfeld				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
<b>Funktion</b>			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
<b>Beispiel 10-1</b>	<b>0x201</b>	<b>3</b>	<b>0x51</b>	<b>0x10</b>	<b>0x00</b>	---	---

# Beispiele

v

Zurückgesendete Information vom Servo:

Funktion	Hex-Wert
Sende-Adresse vom Servo	0x181
Datenlänge 4 Byte	DLC=6
REGID für Status (KERN_STATUS)	0x40
Wert von KERN_STATUS (0x40)	0x0181

Bereich	Kopf		Datenfeld					Byte 5
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	
<b>Funktion</b>			REGID	b7 - 0	b15 - 8	b23 - 16	b31 - 24	b32 - 39
<b>Bsp. 10-1</b>	<b>0x181</b>	<b>6</b>	<b>0x40</b>	<b>0x85</b>	<b>0x31</b>	<b>0x00</b>	<b>0x00</b>	<b>0x**</b>

Aktueller Zustand von Status (KERN\_STATUS) = 0x3185

Bit 0	Antrieb freigegeben	(Ena)
Bit 2 oder Bit 3	Endschalter belegt	(Lim+ oder Lim-)
Bit 7	Positionsregelung	(P-N)
Bit 8	Drehzahlregelung	(N-I)
Bit 12	Kalibriert	(Cal)

## Beispiel 11: Routine für einfache Drehzahl-Regelung

Fahren mit verschiedenen Drehzahlen und Stop (Rx-ID = 0x201; Tx-ID=0x181).

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Kommentar (Sicht Master)
201	3	3D	E2	00				Senden Sendeanforderung BTB
181	4	E2	01	00	00			Empfangen BTB 0xE2
201	3	51	04	00				Senden Sperre
201	3	3D	E8	00				Senden Sendeanforderung Freigabe (Hardware)
181	4	E8	01	00	00			Empfangen Freigabe 0xE8
201	3	51	00	00				Senden Nicht Sperre (Freigabe)
201	3	35	F4	01				Senden ACC-Rampe (500 ms = 0x01FE4)
201	3	ED	E8	03				Senden DEC-Rampe (1000 ms =0x03E8)
201	3	31	D4	03				Senden Drehzahl-Sollwert 0x31 (30 % = 0x03D4)
201	3	3D	30	64				Senden Sendeanforderung Drehzahl-Istwert (alle 100ms)
181	4	30	xx	xx	xx			Empfangen Drehzahl Istwert 0x30 (Wert xxx alle 100 ms)
201	3	31	A4	7F				Senden Drehzahl-Sollwert 0x31 (100 % = 0x7FA4)
201	3	31	00	00				Senden Drehzahl Null
201	3	51	04	00				Senden Sperre

## Beispiel 12: Routine für einfache Positions-Regelung

Referenzfahrt und fahren auf eine Position und zurück auf Null-Position

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Kommentar (Sicht vom Master)
201	3	3D	E2	00				Senden Sendeanforderung BTB
181	4	E2	01	00	00			Empfangen BTB 0xE2
201	3	51	04	00				Senden Sperre
201	3	31	00	00				Senden Sendeanforderung Freigabe (Hardware)
201	3	3D	E8	00				Empfangen Freigabe 0xE8
181	4	E8	01	00	00			Senden Nicht Sperre (Freigabe)
201	3	51	00	00				Senden Start Referenzfahrt
201	3	78	01	00				Senden Wert für Konfigurationsmaske
201	3	52	30	00				Senden Drehzahl-Sollwert 0x31 (30 % = 0x03D4)
201	3	51	10	00				Senden Sendeanforderung Statusmeldung nach Ereignis
181	4	40	xx	xx	xx			Empfangen Statusmeldung (Wert xxxx)
201	5	6E	C0	C6	2D	00		Senden Positionsziel 3000000 Num
201	3	3D	F4	00				Senden Sendeanforderung in Toleranz
181	4	F4	01	00	00			Empfangen in Toleranz
201	5	6E	00	00	00	00		Senden Positionsziel Null
181	3	F4	01	00	00			Empfangen in Toleranz
201	3	51	04	00				Senden Sperre

# Einheiten

## 6 Einheiten

### 6.1 Umrechnung der Maßeinheit

Für Position, Drehzahl, Strom und Sollwert:

Die Messwerte werden im Gerät nicht umgerechnet.

Es werden die numerischen Werte (Num) angezeigt und verarbeitet.

Diese Werte sind bei der Datenübertragung (CAN-BUS, RS232) sowie bei der Track- und Oszilloskopanzeige zu beachten.

#### Position

Bereich Pos-Istwert	Resolver	Inkrementalgeber
Impulse/rpm Maximalwert +/-2147483647 (31Bit-1)	65536	65536
Auflösung (kleinster Wert)	16 (65536/4096 (12Bit))	65536/(Ink x4)
Beispiel Spindeltrieb Steigung 5 mm/rpm	Fahrweg 1000 mm = 200 rpm 200 Upm = 13107200 Auflösung 65536/4096 = 16	Inkrementalgeber 2048 Imp/rpm Fahrweg 1000mm = 200 rpm 200 rpm = 1638400 Auflösung 65536/8192 = 8

#### Drehzahl

Bereich Speed-Istwert	Kalibrierung Drehzahl (Nmax)	Begrenzung
Maximalwert +/- 32767 (15Bit-1)	N max-Wert im Parameterfeld Motor und Speed = 32767	Begrenzung im Parameterfeld Speed im Limit
Beispiel	N max = 2000 Die Drehzahl von 2000 rpm entspricht 32767	Drehzahl begrenzen auf 1500 rpm Limit = $32767/2000 * 1500 = 24575$ Num Die maximale Drehzahl ist auf 1500 rpm begrenzt

#### Strom

Bereich Strom-Ist	I 100%	Kalibrierung Nennstrom I-device			Spitzenstrom DC blockiert		Begrenzung
Maximalwert +/- 9Bit	mV	Num	Aeff	A=	Num	A=	
DS 205/405	550	110	5	7	160	10	Begrenzung im Parameterfeld. Motor und Current. Der kleinere Wert ist wirksam.
DS 412	800	160	12	17	230	24	
DS 420	700	140	20	28	200	40	
Beispiel (DS205/4059)							I <sub>con eff</sub> begrenzen auf 2 A. I <sub>con</sub> = $110 / 5 * 2 = 44$ Num. Der maximale Dauerstrom ist auf 2 A begrenzt.

#### Sollwerte

Bereich Pos-Soll	Bereich Speed-Soll	Bereich Strom-Soll		
Maximalwert +/- 31Bit	Maximalwert +/- 15Bit	Maximalwert +/- 9Bit		
+/- 2147483647 Num	+/- 32767 Num	DS205/405	nenn:110	max:160
		DS 412	nenn:160	max:230
		DS 420	nenn: 140	max:200

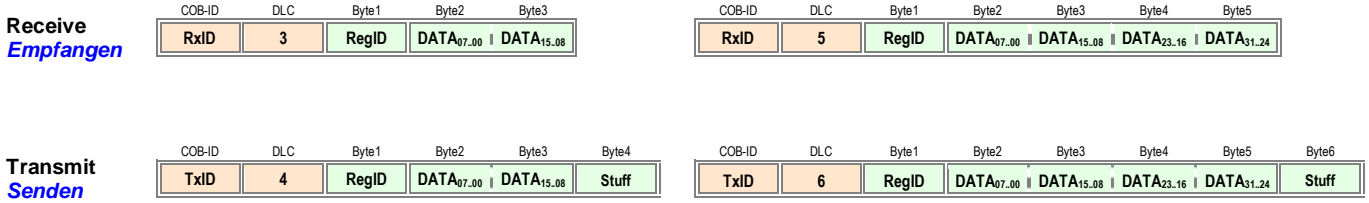
**Achtung: Analog-Sollwert (AIN1, AIN2) 10 V entspricht 29490 (90 % der max. Drehzahl)**

**Valid from  
Gültig ab**

**FW 378**



**A short description of the CAN-Bus interface  
Ein kurze Erklärung des CAN-Bus Interfaces**



- As standard drive CAN-Bus command messages are 3 bytes long (16-bit data) or 5 bytes long (32-bit data).  
*Standartmässig sind die Regler CAN-Bus Befehl-Telegramme 3 Byte lang (16-Bit Daten) oder 5 Byte lang (32-Bit Daten).*
  - “Remote Transmit Requests” (RTR) will be ignored.  
*“Remote Transfer Requests” (RTR) werden ignoriert.*
  - If a 3 byte message (16-bit data) is received and 32-bit data expected, the value will be zero / sign extended as required.  
*Wenn ein 3 Byte Telegramm(16-Bit Daten) ankommt und 32-Bit Daten erwartet wird, wird der Wert nach Bedarf null- / vorzeichen-erweitert.*
  - If a 5 byte message (32-bit data) is received and 16 bit data expected, the upper data will be thrown away.  
*Wenn ein 5 Byte Telegramm(32-Bit Daten) ankommt und 32-Bit Daten erwartet wird, werden die oberen Daten wegwerfen*
- As standard drive CAN-Bus reply messages are 4 bytes long (16-bit data) or 6 bytes long (32-bit data).  
*Standartmässig sind die Regler CAN-Bus Antwort-Telegramme 4 Byte lang (16-bit Daten) oder 6 Byte lang (32-bit Daten).*
- To get the drive to send all replies as 6 byte messages (32-bit data) a bit in RegID 0xDC has to be manually modified.  
*Daß der Regler alle Antworten als 6-Byte Telegramme schicken, muß ein Bit in RegID 0xDC manuell modifiziert werden.*
  - In NDrive open “Manual Read/Write” in the Diagnostic window  
*In NDrive “Manual Read/Write” in der Diagnose-Fenster aufmachen.*
  - Read / Lesen ID register 0xDC value 0x00nn
  - change upper byte to 01 (00 = standard configuration)  
*ändere obere Byte zum 01 (00 = Standardkonfiruration)*
  - Write / Schreiben ID register 0xDC value 0x01nn
- Don't forget to save using “Write 0” in the Settings window.  
*Vergesse nicht mit “Schreibe 0” in den Einstellungen-Fenster zum sichern.*



Valid from  
Gültig ab

FW 378



5. Commands with 16-bit formats (as examples)  
Befehle mit 16-Bit Formate (als Beispiele)

Command Befehl	(PC → Drive) (PC → Regler)	Range (16-Bit) Bereich	Units Einheiten
<b>SPEED_COMMAND</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 RxD 3 31 0xNN 0xNN RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	+100% +32767 0x7FFF	±32767
		+50% +16384 0x4000	→ ±100%
<b>STOP</b> ≡ <b>SPEED_COMMAND = 0</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 RxD 3 31 0x00 0x00 RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	0 % 0 0x0000	
		-50% -16384 0xC000	
		-100% -32767 0x8001	
<b>FUNC_REF_START</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 RxD 3 78 0xNN 0xNN RegID Data <sub>07..00</sub> Data <sub>15..08</sub>		
<b>TORQUE_COMMAND</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 RxD 3 90 0xNN 0xNN RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	+150% +32767 0x7FFF	±32767
		+100% +21845 0x5555	
		0 % 0 0x0000	→ ±150%
		-100% -21845 0xAAAA	
		-150% -32767 0x8001	

6. Commands with 32-bit formats (as examples)  
Befehle mit 32-Bit Formate (als Beispiele)

Command Befehl	(PC → Drive) (PC → Regler)	Range (32-Bit) Bereich	Units Einheiten
<b>POS_DEST</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 Byte4 Byte5 RxD 5 6E 0xNN 0xNN 0xNN 0xNN RegID Data <sub>07..00</sub> Data <sub>15..08</sub> Data <sub>23..16</sub> Data <sub>31..24</sub>	+2147483647 0x7FFF'FFFF	±65536
		+1073741824 0x4000'0000	≡ ±rev
<b>POS_PRESET</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 Byte4 Byte5 RxD 5 7E 0xNN 0xNN 0xNN 0xNN RegID Data <sub>07..00</sub> Data <sub>15..08</sub> Data <sub>23..16</sub> Data <sub>31..24</sub>	+1048576 0x0010'0008	
		+65536 0x0001'0000	
<b>VAR1</b> (send on requirement noch Bedarf schicken)		+32767 0x0000'7FFF	
		+16384 0x0000'4000	
		0 0x0000'0000	
		-16384 0xFFFF'C000	
<b>VAR2</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 Byte4 Byte5 RxD 5 D1 0xNN 0xNN 0xNN 0xNN RegID Data <sub>07..00</sub> Data <sub>15..08</sub> Data <sub>23..16</sub> Data <sub>31..24</sub>	-32767 0xFFFF'8001	
		-65536 0xFFFF'0000	
<b>VAR3</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 Byte4 Byte5 RxD 5 D2 0xNN 0xNN 0xNN 0xNN RegID Data <sub>07..00</sub> Data <sub>15..08</sub> Data <sub>23..16</sub> Data <sub>31..24</sub>	-1048576 0xFFFF'0000	
		-1073741824 0xC000'0000	
<b>VAR4</b> (send on requirement noch Bedarf schicken)	COB-ID DLC Byte1 Byte2 Byte3 Byte4 Byte5 RxD 5 D3 0xNN 0xNN 0xNN 0xNN RegID Data <sub>07..00</sub> Data <sub>15..08</sub> Data <sub>23..16</sub> Data <sub>31..24</sub>	-2147483647 0x8000'0001	

7. Commands for an immediate reply request (as examples)  
Sofortiger Antwortanforderungsbefehl (als Beispiele)

	Request Anforderung	(PC → Drive) (PC → Regler)	→	Reply Antwort	(Drive → PC) (Regler → PC)	Range Bereich	Units Einheiten
<b>SPEED_RPM_MAX_INT</b> (request once) <i>einmal anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0xCE 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 TxID 4 0xCE 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	16-bit	rpm U/min
<b>CURRENT_DEVICE</b> (request once) <i>einmal anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0xC6 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 TxID 4 0xC6 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	16-bit	dA
<b>CURRENT_200PC</b> (request once) <i>einmal anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0xD9 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 TxID 4 0xD9 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	16-bit	ADC units
→							
<b>SPEED_ACTUAL</b> (request on requirement) <i>noch Bedarf anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0x30 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 TxID 4 0x30 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	16-bit	±32767 ≡ ±100%
<b>CURRENT_ACTUAL</b> (request on requirement) <i>noch Bedarf anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0x20 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 TxID 4 0x20 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	16-bit	ADC units
<b>STATUS</b> (request on requirement) <i>noch Bedarf anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0x40 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 Byte5 Byte6 TxID 6 0x40 0xNN 0xNN 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub> Data <sub>23..16</sub> Data <sub>31..24</sub>	32-bit	Bit-Map
			→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 TxID 4 0x40 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	16-bit	Bit-Map
<b>LOGICMAP_ERRORS</b> (request on requirement) <i>noch Bedarf anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0x8F 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 Byte5 Byte6 TxID 6 0x8F 0xNN 0xNN 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub> Data <sub>23..16</sub> Data <sub>31..24</sub>	32-bit	Bit-Map
			→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 TxID 4 0x8F 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	16-bit	Bit-Map
<b>LOGICMAP_IO</b> (request on requirement) <i>noch Bedarf anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0xD8 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 TxID 4 0xD8 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub>	16-bit	Bit-Map
<b>POS_ACTUAL</b> (request on requirement) <i>noch Bedarf anfordern</i>	COB-ID DLC Byte1 Byte2 Byte3 RxID 3 0x3D 0x6E 0x00	Read RegID Time	→	COB-ID DLC Byte1 Byte2 Byte3 Byte4 Byte5 Byte6 TxID 6 0x6E 0xNN 0xNN 0xNN 0xNN Stuff	RegID Data <sub>07..00</sub> Data <sub>15..08</sub> Data <sub>23..16</sub> Data <sub>31..24</sub>	32-bit	±65536 ≡ ±rev
→							
(units conversions Einheiten-Umstellung)	$\overset{[rpm]}{N_{act}} = 100 \cdot \left( \frac{RegID[0x30][units]}{SpeedActual} / 32767 \right)$		$\overset{[rpm]}{N_{act}} = SpeedRpmMaxInt \cdot \left( \frac{RegID[0x30][units]}{SpeedActual} / 32767 \right)$				
	$\overset{[A]}{I_{act}} = 200 \cdot \left( \frac{RegID[0x20][units]}{CurrentActual} \cdot \frac{RegID[0xD9][units]}{Current200pc} \right)$		$\overset{[A]}{I_{act}} = \frac{2}{10} \cdot \frac{RegID[0xC6][dA]}{CurrentDevice} \cdot \left( \frac{RegID[0x20][units]}{CurrentActual} \cdot \frac{RegID[0xD9][units]}{Current200pc} \right)$				

8. Up to 8 time-triggered reply requests can be activated  
Bis 8 zeitgesteuerte Antwortanforderungen können aktiviert werden

- The format is as above, with the "Time" entry setup as follows:  
Der Format ist wie oben, mit dem „Time“ Feld folgendes definiert:
- |      |           |      |                      |           |      |                 |              |
|------|-----------|------|----------------------|-----------|------|-----------------|--------------|
| 0x00 | immediate | 0xFF | suspend transmission | otherwise | 0xNN | timer setup     | (1 – 254 ms) |
| Time | sofort    | Time | Senden suspendiert   | sonst     | Time | Zeit einstellen | (1 – 254 ms) |
- Entries with suspended transmissions can be overwritten by newer requests.  
Eingaben mit suspendierten Senden können bei neueren Anforderungen überschrieben werden.



RegNr	Typ	Hex value	Decimal	Label	(intern name)	Description
0x00	(UK)	0x0000	0	(rsv)	(rsv)	(reserved)
0x01	(UK)	0x0000	0	Usr-Opt	(USER_SPEC_OPT)	(Deif) Options
0x02	(RO)	0x0000	0	SC-info	(USER_SPEC_STA)	(Deif) Safety-State
0x03	(SP)	0x0000	0	Cmd-Spec	(USER_SPEC_DEM)	(Deif) Cmd-Specials
0x04	(SP)	0x0000	0	(Key)	(USER_KEY)	?? (User Key)
0x05	(RW)	0x05dc	1500	F nom	(MOTOR_NOM_F)	Nominal motor frequency (FU)
0x06	(RW)	0x0000	0	V nom	(MOTOR_NOM_V)	Motor nominal voltage (FU)
0x07	(RW)	0x00000000	0	T dc	(UF_TDC)	Time DC-pre-mag. (FU)
0x08	(RW)	0x00000000	0	V dc	(UF_UDC)	DC voltages (FU)
0x09	(RW)	0x00000064	100	F dc	(UF_SPEZIAL)	??
0x0a	(RW)	0x00000000	0	U min	(UF_UMIN)	Minimum voltage (FU)
0x0b	(RW)	0x00000000	0	F min	(UF_FMIN)	Minimum frequency (FU)
0x0c	(RW)	0x00000000	0	V corner	(UF_UECK)	Voltage für max. frequency (FU)
0x0d	(RW)	0x00000000	0	F corner	(UF_FECK)	Frequency for max. voltage (FU)
0x0e	(RW)	0x0000	0	Cos Phi	(UF_POWF)	Power factor (FU)
0x0f	(RW)	0x0064	100	(...)	(UF_EXTRA)	(...)
0x10	(SP)	0x0000	0	Chan	(CAPTURE_CHAN)	Oscilloscope trigger channel
0x11	(RO)	0xcb5e37b4	3411949492	Ctrl	(CONTROL_STATU)	Control-Status
0x12	(SP)	0x7ae8	31464	Trig. Level	(CAPTURE_TRIGL)	Oscilloscope trigger level
0x13	(SP)	0x0001	1	Trig. Edge	(CAPTURE_TRIGE)	Oscilloscope trigger function
0x14	(SP)	0x9134	37172	Trig. Sce	(CAPTURE_TRIGS)	Oscilloscope trigger source
0x15	(SP)	0x0001	1	Source	(CAPTURE_SOURC)	Oscilloscope source
0x16	(SP)	0x0001	1	Skip	(CAPTURE_SKIP)	Oscilloscope skip
0x17	(FN)	0x0000	0	Read Cmd	(CAPTURE_READ)	Oscilloscope read
0x18	(FN)	0xface	64206	Run Cmd	(CAPTURE_RUN)	Oscilloscope Run
0x19	(RW)	0x0000	0	PWM freq.	(PWM-FREQ)	Frequency PWM-stage
0x1a	(SP)	0x0000	0	Look-up	(LOOKUP_TEMP)	lookup field (temperary)
0x1b	(RO)	0x01d8	472	FW	(FW-VERSION)	Firmware
0x1c	(RW)	0x000a	10	Kp	(I_KP)	Proportional amplification current
0x1d	(RW)	0x03e8	1000	Ti	(I_KI)	Integral action time current control
0x1e	(RW)	0x0000	0	Cutoff (dig.)	(DIG_CUTOFF)	Cutoff-digital-cmd
0x1f	(RO)	0x07f0	2032	??	(I3_ISTOFFSET)	Offset actual current 3
0x20	(RO)	0x0002	2	I actual	(I_IST)	current actual value
0x21	(SP)	0x0000	0	Id set (dig.)	(I_SOLLOFFSET)	D-current setpoint
0x22	(RO)	0x0000	0	I cmd (ramp)	(I_REF)	current set point numeric
0x23	(RO)	0x0000	0	Id ref	(ID_REF)	D-Current reference
0x24	(RO)	0x013f	319	I max inuse	(I_MAXPLUS)	I max inuse
0x25	(RW)	0x03e9	1001	Ramp	(I_DELTAMAXPLU)	Icmd ramp
0x26	(RO)	0x0000	0	I cmd	(I_SOLL)	command current
0x27	(RO)	0xffff	-1	Iq actual	(IQ_ACTUAL)	Q-current actual
0x28	(RO)	0xfffd	-3	Id actual	(ID_ACTUAL)	D-current actual
0x29	(RO)	0x0000	0	Vq	(VQ)	Q-Outputvoltage
0x2a	(RO)	0x0000	0	Vd	(VD)	D-Outputvoltage
0x2b	(RW)	0x0050	80	TiM	(I_ERRSUMMAX)	Max. integration sample count
0x2c	(RW)	0x000a	10	Kp	(SPEED_KP)	Proportional gain speed
0x2d	(RW)	0x0064	100	Ti	(SPEED_KI)	Integration time speed
0x2e	(RW)	0x0000	0	Td	(SPEED_KD)	D_ speed
0x2f	(RW)	0x10000000	268435456	Ain1 offset/scale	(ANALOG_OFFSET)	Offset/scale Ain1
0x30	(RO)	0x0000	0	N actual	(SPEED_ACTUAL)	Speed actual value
0x31	(RW)	0x0000	0	N set (dig.)	(SPEED_CMD)	Digital Speed Set Point
0x32	(RO)	0x0000	0	N cmd (ramp)	(SPEED_REF)	Command speed after ramp
0x33	(RW)	0x0000	0	N error	(SPEED_ERR)	Speed error
0x34	(RW)	0x7fff	32767	N-Lim	(SPEED_LIMIT)	Speed limit
0x35	(RW)	0x00010064	65636	Accel.	(SPEED_DELTAMA)	Speed/torque acceleration ramp
0x36	(RW)	0x0000	0	Command	(COMMAND_SOURC)	Selection command speed
0x37	(RO)	0x0002	2	Loop	(SPEED_COUNTMA)	current to speed loop factor
0x38	(RO)	0x0000	0	Iq error	(IQ_ERR)	Current Iq error
0x39	(RO)	0x0000	0	Id error	(ID_ERR)	Current Id error
0x3a	(RW)	0xface	64206	?? (...)	(0x3a (...))	?? (...)
0x3b	(RW)	0x0050	80	TiM	(SPEED_ERRSUMM)	Max. integration sample count
0x3c	(RW)	0x7fff	32767	I-red-N	(I_RD_N)	Current derating speed
0x3d	(FN)	0x0618	1560	Read	(READ)	Function
0x3e	(RW)	0x8000	-32768	N-Lim-	(SPEED_CLIP_NE)	Speed limit negative
0x3f	(RW)	0x7fff	32767	N-Lim+	(SPEED_CLIP_PO)	Speed limit positive

RegNr	Typ	Hex value	Decimal	Label	(intern name)	Description
0x40	(RO)	0x00000380	896	Status map	(STATUS )	Status
0x41	(RO)	0x0000	0	incr_delta	(INCR_DELTA )	??
0x42	(RO)	0x86b9	-31047	MotorPos mech	(MPOS_ACTUAL_M)	Motor actual angular position me
0x43	(RO)	0x6a13	27155	MotorPos elec	(MPOS_ACTUAL_E)	Motor actual angular position e
0x44	(RW)	0xfdb0	-592	FB-Offset	(MPOS_ISTOFFSE)	phase angle offset Feedback
0x45	(RO)	0x00000000	0	I2t & Regen. Energy	(IT_RG_MONITOR)	monitor i2t & regen circuit
0x46	(RW)	0x7fff	32767	I lim dig	(I_LIMIT )	Current limit with a digital sw
0x47	(RW)	0xface	64206	...	(... )	...
0x48	(RO)	0x013f	319	I lim inuse	(I_LIM_INUSE )	actual current limit
0x49	(RO)	0x0000	0	T-motor	(T_MOTOR )	motor temperature
0x4a	(RO)	0x0000	0	T-igbt	(T_IGBT )	power stage temperature
0x4b	(RO)	0x0000	0	T-air	(T_AIR )	air temperature
0x4c	(RW)	0x0000	0	I-red-TE	(I_RD_TE )	Current derate Temp.
0x4d	(RW)	0x0035	53	I max	(MOTOR_I_MAX )	max. motor current
0x4e	(RW)	0x0035	53	I nom	(MOTOR_I_DAUER)	Motor continuous current
0x4f	(RW)	0x0006	6	M-Pole	(MOTOR_POLE )	Motor pole count
0x50	(RW)	0x0000	0	Cutoff	(AIN1_CUTOFF )	cutoff window Ain1
0x51	(SP)	0x0000	0	Mode	(MODE )	Mode State
0x52	(SP)	0x0000f811	63505	Status mask	(STATUS_MASK )	Status mask
0x53	(RW)	0x0000	0	Cutoff	(AIN2_CUTOFF )	cutoff window Ain2
0x54	(RO)	0xffff	-1	I1 actual	(I1_IST )	Current actual value I1
0x55	(RO)	0x0004	4	I2 actual	(I2_IST )	Current actual value I2
0x56	(RO)	0x0002	2	I3 actual	(I3_IST )	Current actual value I3
0x57	(RO)	0x0000	0	I lim inuse rmp	(I_LIM_INUSE_R)	??
0x58	(RW)	0x0000	0	I-red-TD	(I_RD-TD )	??
0x59	(RW)	0x0bb8	3000	N nom	(MOTOR_RPMMAX )	Rated motor speed
0x5a	(RW)	0x00000808	2056	Device Options	(KERN_OPTIONS )	Device settings (options)
0x5b	(RW)	0x0000	0	Kacc	(SPEED_KS )	Acceleration amplification
0x5c	(RO)	0x86b9	34489	Rotor	(ROTOR )	Rotor signals
0x5d	(RO)	0x0000	0	N cmd (int)	(SPEED_CMD_INT)	Command speed internal
0x5e	(RW)	0x0002	2	Filter	(SPEED_FILTER_)	Filter speed actual value
0x5f	(RO)	0x0000	0	I act (filt)	(I_IST_FILT )	Filtered actual current
0x60	(RW)	0x0000	0	Filter	(AINx_FILT )	
0x61	(RO)	0x0000	0	I t	(IT_MONITOR )	I t monitor
0x62	(RW)	0x075bcd15	123456789	S-Nr.	(DEVICE_SERIAL)	Device Serial number Servo
0x63	(RO)	0x0000	0	fpga Status	(POWER_BOARD_S)	FPGA Status
0x64	(RW)	0x00e6	230	Mains	(DEVICE_MAINS )	Mains supply voltage
0x65	(RW)	0x00500019	5242905	Regen-P, Regen-R	(DEVICE_EXT_RE)	Regenerative Resistor power rat
0x66	(RO)	0xface	64206	Vdc-Bat	(DC_BUS )	Battery voltage
0x67	(RW)	0x00011313	70419	Type	(DEVICE_AUTO_I)	Device type
0x68	(RW)	0x0201	513	Rx ID	(CAN_ID_RX )	CAN-Bus drive rx address
0x69	(RW)	0x0181	385	Tx ID	(CAN_ID_TX )	CAN-Bus drive tx address
0x6a	(RW)	0x000f	15	Kp	(POS_KP )	position controller proportional
0x6b	(RW)	0x01f5	501	Ti	(POS_KI )	integral action time (Integral p
0x6c	(RW)	0x0000	0	Td	(POS_KD )	advancing-time (Differezial-part
0x6d	(RO)	0x000086b9	34489	Pos actual	(POS_ACTUAL )	actuael position numeric
0x6e	(SP)	0x00000000	0	Pos dest	(POS_DEST )	position-destination
0x6f	(RO)	0x00000000	0	Pos actual 2	(RegName_0x6f)	Pos actual 2
0x70	(RO)	0x00000000	0	Pos error	(POS_ERR )	position error
0x71	(RW)	0x0033	51	TiM	(POS_ERRSUMMAX)	Max.integration sample count, p
0x72	(RW)	0x00000000	0	Off. Ref.	(POS_REF_OFFSE)	reference zero offset
0x73	(RW)	0x4025	16421	NBT	(CAN_BTR )	CAN-BUS transmission rate
0x74	(RO)	0x91be	-28226	Zero-Capture	(POS_ZEROCAPTU)	Pos Zero Capture
0x75	(RW)	0x0000	0	Reso edge	(POS_REFRESOED)	Reso pos. at Rsw
0x76	(RW)	0x0078	120	Speed 1	(SPEED_CALIB_F)	Reference speed (fast)
0x77	(RW)	0x0078	120	Speed 2	(SPEED_CALIB_S)	Reference speed (slow)
0x78	(FN)	0x444d	17485	Start park cycle	(FUN_REF_START)	Start park cycle
0x79	(RW)	0x0064	100	Tol-wind	(POS_WINDOW )	Tolerance window for position
0x7a	(SP)	0xfd944f98	4254355352	Preset	(POS_PRESET )	Preset value
0x7b	(RO)	0x00000000	0	Off. Var	(POS_VAR_OFFSE)	user zero offset
0x7c	(RW)	0x00000000	0	ND-Scale	(NDRIVE_SCALE)	Display-conversion factor-posit
0x7d	(RW)	0x00000000	0	ND-Offset	(NDRIVE_OFFSET)	Verschiebefaktor Pos-Anzeige
0x7e	(RW)	0x00000000	0	Factor-ext	(ENCODER_2_SCA)	Scale 2nd encoder
0x7f	(RW)	0x00000000	0	??	(OFFSET_SLACK )	??

RegNr	Typ	Hex value	Decimal	Label	(intern name)	Description
0x80	(RW)	0x86b9	34489	??	(POS_DIFF_SLAC)	??
0x81	(UK)	0xface	64206	...	(...)	...
0x82	(RO)	0xface	64206		(DEVICE_SERIAL)	Device serial number ext.
0x83	(FN)	0x444d	17485	??	(FUN_PARAREAD)	??
0x84	(FN)	0x444d	17485	??	(FUN_PARAWRITE)	??
0x85	(FN)	0x0000	0	Auto-Fn	(FUN_SPEZIAL)	Auto-Functions
0x86	(UK)	0xface	-1330	??	(READ_INFO)	??
0x87	(RW)	0xface	64206	...	(...)	...
0x88	(RW)	0x00000000	0	Rx ID 2	(CAN_ID_RX_2)	CAN-Bus drive rx 2 address
0x89	(RW)	0x00000000	0	Tx ID 2	(CAN_ID_TX_2)	CAN-Bus drive tx 2 address
0x8a	(RO)	0x0000	0	V out	(VOUT)	Output-voltage usage
0x8b	(RW)	0x0000	0	V red	(VRED)	Start point field reduction
0x8c	(RW)	0x0000	0	V kp	(VKP)	Proportional amplification field
0x8d	(RW)	0x0000	0	V-Ti	(VTI)	Time constant integral part fie
0x8e	(FN)	0x444d	17485	??	(FUN_ERRCANCEL)	Clear error list
0x8f	(RO)	0x00000020	32	Warning-Error map	(ERR_BITMAP1)	Description of 0x8f
0x90	(SP)	0x0000	0	M set (dig.)	(TORQUE_SETPOI)	Digital Torque Set Point
0x91	(RO)	0x000086b9	34489	Pos cmd	(POS_COMMAND)	Command position
0x92	(RO)	0x0000	0	??	(CAN_ERROR_BUS)	CAN-BUS Bus-Off count
0x93	(RO)	0x0000	0	??	(CAN_ERRWRITET)	CAN-BUS ??
0x94	(RO)	0x0000	0	fpga 1st error	(POWER_BOARD_E)	FPGA 1st Error
0x95	(RO)	0x0000	0	??	(CAN_COUNTREAD)	CAN-BUS ??
0x96	(RO)	0x0000	0	??	(CAN_COUNTWRIT)	CAN-BUS ??
0x97	(RO)	0x0000	0	??	(CAN_COUNTREJ)	CAN-BUS
0x98	(RO)	0xface	-1330	O-Block	(LOG_O_BLOCK)	O-Block
0x99	(RO)	0x02b5	693	Info Intr	(INFO_INTERRUP)	Info - Interrupt time
0x9a	(RO)	0x0000	0	(dbg) temp	(TEMP)	(dbg) Temp
0x9b	(RO)	0xface	64206	in Block	(LOG_I_BLOCK)	I-Block
0x9c	(UK)	0xface	-1330	Pt100-1	(T-PT-1)	Temp. Sensor Pt100-1
0x9d	(UK)	0xface	-1330	Pt100-2	(T-PT-2)	Temp. Sensor Pt100-2
0x9e	(UK)	0xface	-1330	Pt100-3	(T-PT-3)	Temp. Sensor Pt100-3
0x9f	(UK)	0xface	-1330	Pt100-4	(T-PT-4)	Temp. Sensor Pt100-4
0xa0	(RO)	0x0000	0	M out	(TORQUE_OUT)	Digital Torque Intern
0xa1	(RO)	0x0000	0	Ballast counter	(BALLAST_COUNT)	Ballast counter
0xa2	(RW)	0x15e0	5600	I-red-TM	(I_RD_TM)	??
0xa3	(RW)	0x1b58	7000	M-Temp	(MOTOR_TEMP_ER)	Motor-Temperatur Abschaltpunkt
0xa4	(RW)	0x3001	12289	Label 0xa4	(MOTOR_OPTION)	Description of 0xa4
0xa5	(RW)	0x00000064	100	DC-Bus min, DC-Bus max	(DEVICE_DCBUS_)	Description of 0xa5
0xa6	(RW)	0x0400	1024	FB-Incr (Mot)	(MOTOR_GEBER_I)	Increments per Rpm
0xa7	(RW)	0x0002	2	FB-Pole	(MOTOR_GEBER_P)	Resolver pole
0xa8	(RO)	0x0000	0	N act (filt)	(SPEED_ACTUAL_)	Actual speed value (filtered)
0xa9	(RO)	0x07ef	2031	I3 adc	(I1_ADC)	Current sensor M1
0xaa	(RO)	0x07ee	2030	I2 adc	(I2_ADC)	Current sensor M3
0xab	(RO)	0xfde8	65000	Logic freq.	(LOGIC_HZ)	Forerground frequency
0xac	(RO)	0x0618	1560	pwm1 (5/6)	(PWM1)	pulse widths modulation Ph1
0xad	(RO)	0x0618	1560	pwm2 (3/4)	(PWM2)	pulse widths modulation Ph2
0xae	(RO)	0x0618	1560	pwm3 (1/2)	(PWM3)	pulse widths modulation Ph3
0xaf	(RO)	0x007d	125	T-intr	(TIMER_DELTA)	Intr. time
0xb0	(RW)	0x444d	17485	??	(FUN_SERIALBOO)	??
0xb1	(RW)	0x0000	0	L sigma-q	(MOTOR_INDUCTA)	Stator Leakage inductance
0xb2	(RW)	0x0000	0	Id nom	(ID_NOM)	nominal magnetising current
0xb3	(RW)	0x007b	123	L magnet.	(MOTOR_MAGN_L)	Motor magnetising inductance
0xb4	(RW)	0x0000	0	R rotor	(MOTOR_ROTOR_R)	rotor resistance
0xb5	(RW)	0x0000	0	Id min	(ID_MIN)	minimum magnetising current
0xb6	(RW)	0x07d0	2000	TC rotor	(MOTOR_TR)	time constant rotor
0xb7	(SP)	0x9133	37171	(dbg) ptr1	(TEMP1_PTR)	(dbg) ptr1
0xb8	(UK)	0x0000	0	(dbg) *ptr1	(TEMP1_PTR_IND)	(dbg) *ptr1
0xb9	(SP)	0x902b	36907	(dbg) ptr2	(TEMP2_PTR)	(dbg) ptr2
0xba	(UK)	0x0002	2	(dbg) *ptr2	(TEMP2_PTR_IND)	(dbg) *ptr2
0xbb	(RW)	0x0000	0	L sigma-d	(MOTOR_INDUCTA)	leakage inductance ph-ph
0xbc	(RW)	0x007b	123	R stator	(MOTOR_STATOR_)	stator resistance ph-ph
0xbd	(RW)	0x0000	0	TC stator	(MOTOR_SPECS_I)	time constant stator
0xbe	(RW)	0x8005	32773	Label 0xbe	(LOGIC_DEFINE_)	Description of 0xbe
0xbf	(RW)	0x8004	32772	Label 0xbf	(LOGIC_DEFINE_)	Description of 0xbf

RegNr	Typ	Hex value	Decimal	Label	(intern name)	Description
0xc0	(RW)	0x800c	32780	Label 0xc0	(LOGIC_DEFINE_)	Description of 0xc0
0xc1	(RW)	0x800c	32780	Label 0xc1	(LOGIC_DEFINE_)	Description of 0xc1
0xc2	(RW)	0x0000	0	Label 0xc2	(LOGIC_DEFINE_)	Description of 0xc2
0xc3	(RW)	0x0000	0	Label 0xc3	(LOGIC_DEFINE_)	Description of 0xc3
0xc4	(RW)	0x20a3	8355	I max pk	(DEVICE_I_MAX_)	Limit for peak current (Servo)
0xc5	(RW)	0x3a3d	14909	I con eff	(DEVICE_I_CNT_)	Limit for continius current (Servo)
0xc6	(RW)	0x0032	50	I device	(DEVICE_I_)	Type current, protected
0xc7	(RW)	0x000a	10	R-Lim	(SPEED_DELTAMA)	Emergency stops time ramp, limit
0xc8	(RW)	0x0e10	3600	Nmax-100%	(SPEED_RPMMAX_)	Maximum rotation speed in turns
0xc9	(RW)	0x0000	0	xKp2	(I_KP2)	proportional amplification posit
0xca	(RW)	0x0000	0	Ti	(POSI_KI)	integral action time (Integral p
0xcb	(RW)	0x0000	0	Kf	(I_KF)	...
0xcc	(RO)	0xc953	-13997	0xcc	(POSI_ERR)	0xcc
0xcd	(RW)	0x0000	0	TiM	(POSI_ERRSUMMA)	Limit integral storeroom peak v
0xce	(RO)	0x0e10	3600	Label 0xce	(SPEED_RPMMAX_)	Description of 0xce
0xcf	(RW)	0x0000	0	Label 0xcf	(POSI_KY)	Description of 0xcf
0xd0	(SP)	0x0000	0	T-Out	(CAN_TIMEOUT)	CAN timeout
0xd1	(RW)	0x0000003e	62	Var1	(VAR1)	Comparison variable-1
0xd2	(RW)	0x00002710	10000	Var2	(VAR2)	Comparison variable-2
0xd3	(RW)	0x00000000	0	Var3	(VAR3)	Comparison variable-3
0xd4	(RW)	0x00000000	0	Var4	(VAR4)	Comparison variable-4
0xd5	(RO)	0xffe0ffe0	-2031648	Ain1	(AIN1)	Analog Ain1 in/scaled
0xd6	(RO)	0x00580058	5767256	Ain2	(AIN2)	Analog Ain2 in/scaled
0xd7	(RW)	0x10000000	268435456	Offset 2	(AIN2_OFFSET)	analog input 2 offset compensat
0xd8	(RO)	0x0020	32	Label 0xd8	(LOGIC_READ_BI)	Description of 0xd8
0xd9	(RO)	0x0349	841	Label 0xd9	(KERN_I_200PC)	Description of 0xd9
0xda	(RW)	0x0000	0		(LOGIC_DEFINE_)	
0xdb	(RW)	0x0000	0		(LOGIC_DEFINE_)	
0xdc	(RW)	0x0030	48	??	(DEFINE_DAC)	??
0xdd	(UK)	0xface	64206	...	(...)	...
0xde	(RO)	0x0000	0	out Dout3	(O_DOUT3)	Digital output 3
0xdf	(RO)	0x0000	0	out Dout4	(O_DOUT4)	Digital output 4
0xe0	(RO)	0x0000	0	out Dout1	(O_DOU1)	Digital output 1
0xe1	(RO)	0x0000	0	out Dout2	(O_DOU2)	Digital output 2
0xe2	(RO)	0x0000	0	out Rdy (BTB)	(O_BTBT)	Device ready
0xe3	(RO)	0x0000	0	O Go	(O_GO)	Internal run
0xe4	(RO)	0x0000	0	(in) Limit1	(I_END1)	Digital input END1
0xe5	(RO)	0x0000	0	(in) Limit2	(I_END2)	Digital input END2
0xe6	(RO)	0x0000	0	(in) Din1	(I_DIN1)	Digital input DIN1
0xe7	(RO)	0x0000	0	(in) Din2	(I_DIN2)	Digital input DIN2
0xe8	(RO)	0x0000	0	(in) Run (Frg)	(I_FRG)	Digital input RUN
0xe9	(RO)	0x0000	0	I Fault	(I_FAULT)	internal error message of the po
0xea	(RO)	0x0000	0	I Regen	(I_BALLAST)	message regen circuit
0xeb	(RO)	0x0001	1	Vdc-Bus	(DC_BUS)	DC-Bus voltage
0xec	(RO)	0x0000	0	I LossOfSignal	(I_LOS)	Resolver fault. Incorrect or mi
0xed	(RW)	0x00010064	65636	Decel.	(SPEED_DELTAMA)	Speed/torque deceleration ramp t
0xee	(RW)	0x0226	550	I 100% (Stromsensor)	(IIST_100PC)	Current sensor justage (protecte
0xef	(RO)	0x0001	1	Label 0xef	(O_NOFAULT)	Description of 0xef
0xf0	(RW)	0x0005	5	T-peak	(TIME_IPEAK)	Timing for peak current
0xf1	(RW)	0x00fa	250	Brake delay	(USER_T_BRAKE)	Brake delay time
0xf2	(RO)	0x0001	1	O Brake	(VO_BRAKE)	Brake on
0xf3	(RO)	0x0000	0	O Icns	(VO_ICNS)	message continuous current
0xf4	(RO)	0x0000	0	O Toler	(VO_TOLER)	message position in tolerance
0xf5	(RO)	0x0001	1	O Less N0	(VO_Less_N0)	message speed <1%
0xf6	(RO)	0x0000	0	Power	(POWER)	Power
0xf7	(RO)	0x0000	0	Work	(WORK)	Work
0xf8	(RW)	0x0000444d	17485	Axis	(ASCII_USER)	Axis label
0xf9	(FN)	0x444d	17485	??	(ASCII_WR_EEP)	??
0xfa	(FN)	0x444d	17485	??	(ASCII_RD_EEP)	??
0xfb	(RO)	0xffd4	-44	Ain1 calc	(AIN1_CALC)	Ain1 calc
0xfc	(RO)	0x0054	84	Ain2 calc	(AIN2_CALC)	Ain2 calc
0xfd	(UK)	0xface	64206	...	(...)	...
0xfe	(UK)	0xface	64206	...	(...)	...
0xff	(UK)	0xface	-1330	rsv	(rsv)	reserved