M A N U A L

CAN - BUS
für Digitale Motorregler

BAMOBIL-xx
BAMOCAR-xx
DS-xx / DPC-xx

UNITEK Industrie Elektronik GmbH
Hans-Paul-Kaysser-Straße 1
71397 Leutenbach-Nellmersbach

Tel: 07195 / 92 83 - 0
contact@unitek.eu
www.unitek.eu

Ausgabe / Version

2024/ V2

Inhaltsverzeichnis

1 Sicherheit 2
1.1 Sicherheitshinweise 2
1.2 Vorschriften und Richtlinien 2
2 Allgemeines 4
2.1 Logikfunktionen 4
3 CAN-BUS Verbindungen 5
3.1 Verbindungen 5
4 Software 7
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
5 Beispiele 10
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
6 Einheiten 19
6.1 Umrechnung der Maßeinheit 19

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 dazugehörenden 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.

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, Stopp	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>.

3 CAN-BUS Verbindungen

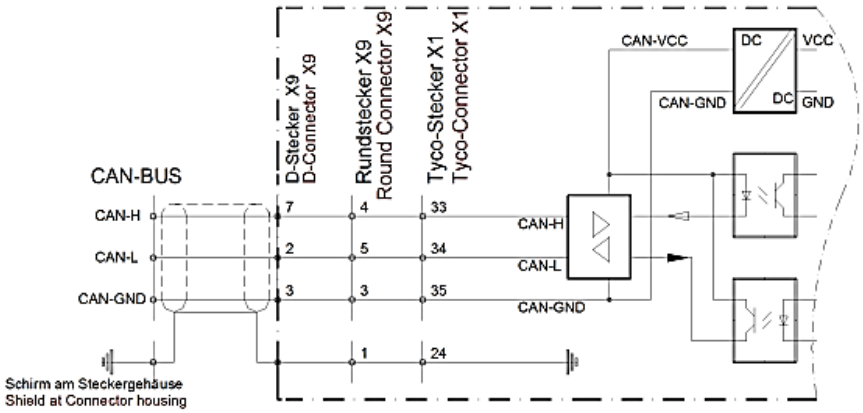
3.1 Verbindungen

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 3-1

CAN-BUS isoliert / CAN Gnd auf gemeinsames Potential bringen

CAN-BUS Verbindung mit mehreren Servo-Verstärkern DS- (Slave)
Bei abweichenden Gerätetypen die Steckerbelegung beachten

(Beispiel):
(Geräte-Manual)

Adresse xx Adresse xx Adresse xx

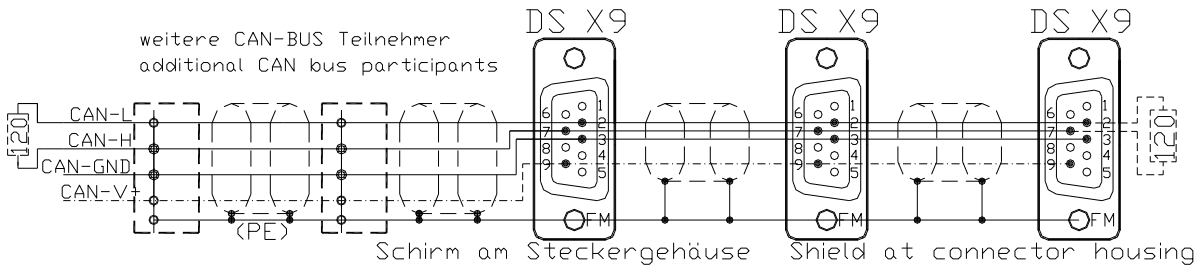


Abbildung 3-2

Terminierungswiderstand

Der Leitungs-Anschlusswiderstand (R = 120 Ω) 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

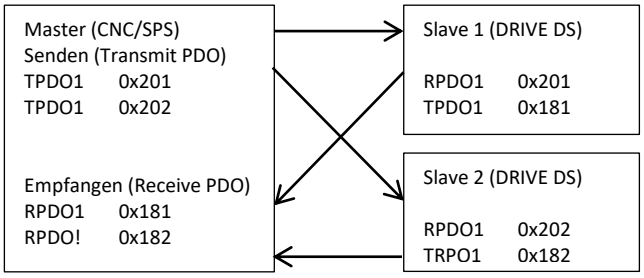
CAN Format
Das CAN-Protokoll ist ein 3 oder 5 Byte Daten-Paket beim Empfang, 4 oder 6 beim Senden.
Es können auch bis zu 8 Byte Daten-Pakete empfangen werden, jedoch wird dieses dann als 5 Byte Daten-Paket ausgewertet. Der Identifier ist 11Bit breit. Es beinhaltet den **COB-Identifier**, die **RTR-Funktion** (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
Funktion	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 0 x 181										
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
Funktion	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
1			8				1				0x181-0x1FF
	1	0	0	0	0	0	0	0	0	1	RPD01
2			0				1				0x201-0x27F

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

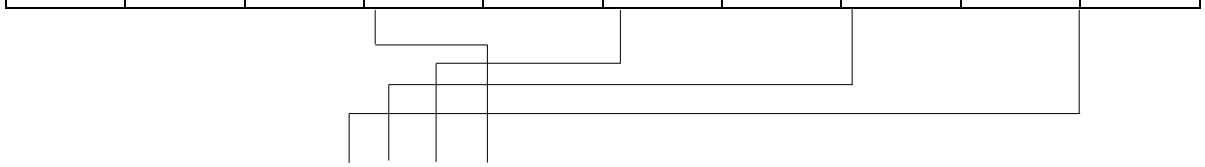
Die Datenlänge wird im Feld DLC-Bits vorgegeben (16 oder 32 Bits).
Byte 2 bis Byte 5 sind für die 32 Bit Register-Daten (4 Byte) bestimmt.
Byte 2 bis Byte 3 sind für die 16 Bit Register-Daten (2 Byte) bestimmt.

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

Byte 0		**Byte 1**		**Byte 2**		**Byte 3**		**Byte 4**	
REGID		**Daten-Bits 7 bis 0**		**Daten-Bits 15 bis 8**		**Daten-Bits 23bis 16**		**Daten-Bits 31 bis 24**	
6	E	1	0	C	A	E	1	1	1



Daten = 0x**11E1CA10** (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

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*

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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 1	0x201	3	0x68	0x10	0x02	---	---

Sende-Adresse ändern im PC-Programm NDrive



Abbildung 5-1

Eingaben:

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

Beispiel 2: Regler sperren (nicht Freigabe) **Nachricht zum Servo**

Funktion	**Hex-Wert**
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Sperre (MODE)	0x51
Wert für Sperre MODE BIT2	0x0004

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**	**3**	**0x51**	**0x04**	**0x00**	---	---

Beispiel 3: Drehzahl-Sollwert **Nachricht zum Servo**

Funktion	**Hex-Wert**
Sende-Adresse zum Servo	0x201
Datenlänge 2 Byte	DLC=3
REGID für Drehzahl-Sollwert (SPEED_SOLL)	0x31
Wert für 10 % Drehzahl Num. 3277	0x0CCD (100 % ≙ 32767)

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 3	**0x201**	**3**	**0x31**	**0xCD**	**0x0C**	---	---

Beispiel 4: Positions-Sollwert **Nachricht zum Servo**

Funktion	**Hex-Wert**
Sende-Adresse zum Servo	0x201
Datenlänge 4 Byte	DLC=5
REGID für Drehzahl-Sollwert (POS_DEST)	0x6E
Wert für Position 3000000	0x2DC6C0

Bereich	**Kopf**		**Datenfeld**				
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte4
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 4	**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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 5	**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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 6	**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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 7	**0x201**	**3**	**0x84**	**0x00**	**0x00**	---	---

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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 8	**0x201**	**3**	**0x3D**	**0x40**	**0x00**	**---**	**---**

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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 8	**0x181**	**4**	**0x40**	**0x81**	**0x01**	**0x****	

(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)

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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 9	**0x210**	**3**	**0x3D**	**0x30**	**0x64**	---	---

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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 9	**0x190**	**4**	**0x30**	**0xFF**	**0x7F**	**0x****	---

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

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 welches 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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 10	**0x201**	**3**	**0x51**	**0x10**	**0x00**	**---**	**---**

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
Funktion			REGID	b7 - 0	b15 - 8	b23 - 16	b31 - 24	b32 - 39
Beispiel 10	**0x181**	**6**	**0x40**	**0x81**	**0x31**	**0x00**	**0x00**	**0x****

Aktueller Zustand von Status (KERN_STATUS) = 0x3181:

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

<u>Beispiel 10-1: Statusmeldung nach ausgewähltem Ereignis</u> 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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 10-1	**0x201**	**3**	**0x52**	**0x12**	**0x00**	**---**	**---**

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
Funktion			REGID	b7 bis 0	b15 bis 8	b23 bis 16	b31 bis 24
Beispiel 10-1	**0x201**	**3**	**0x51**	**0x10**	**0x00**	**---**	**---**

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

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)

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 Spindelantrieb 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	
							Icon eff begrenzen auf 2 A. Icon = 110 / 5 * 2 = 44 Num. Der maximale Dauerstrom ist auf 2 A begrenzt.
Beispiel (DS205/4059							

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)