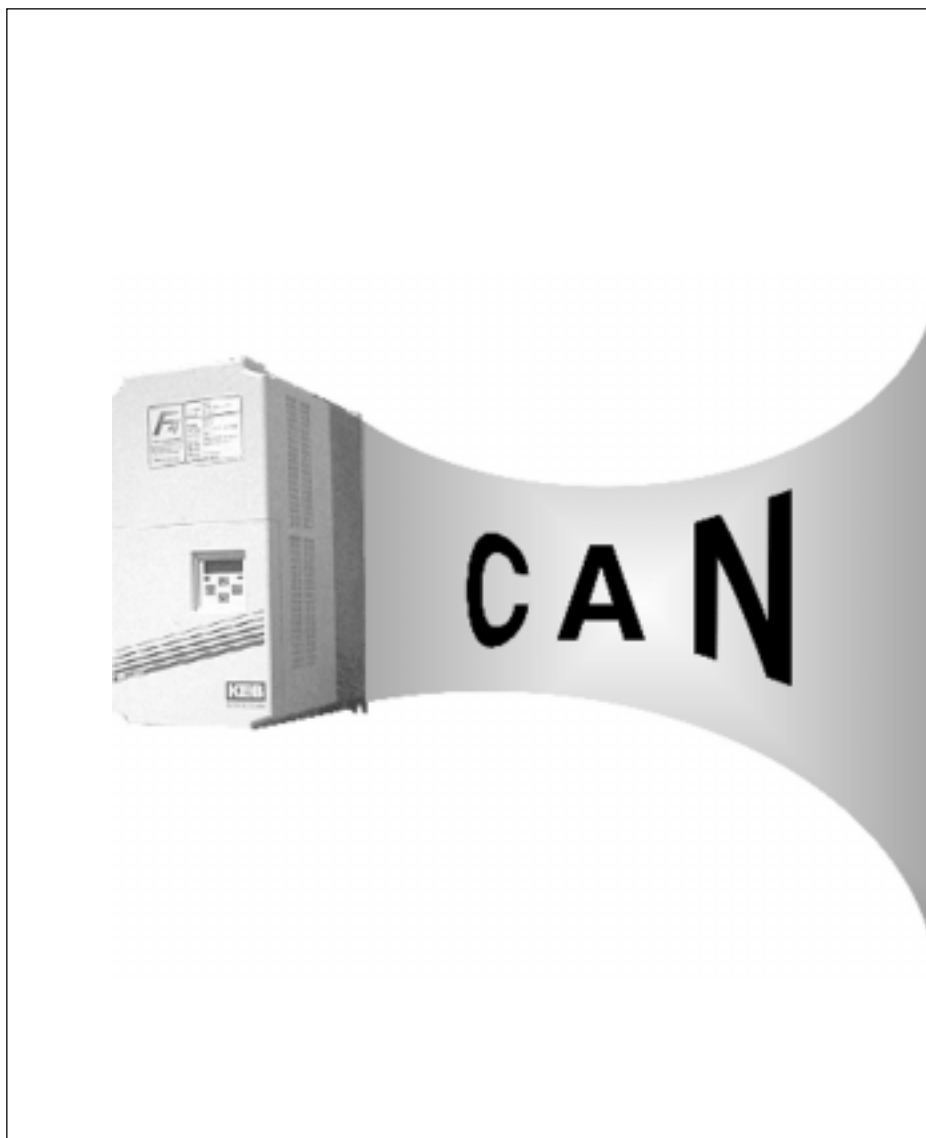


BETRIEBSANLEITUNG

INSTRUCTION MANUAL



KEB COMBICOM

**CAN-Anschaltung
CAN-Interface**



Seite.....D 3 bis D 16



Page.....E 3 to E 16

1.	Allgemeines	D4
1.1.	CAN-Interface	D4
2.	Die Hardware	D4
2.1	Zusammenfassung der technischen Daten	D5
3.	Die Software	D5
3.1.	Die Kommunikationsobjekte	D5
3.1.1.	Das READ_WRITE-Objekt	D5
3.1.2.	Das OUT-Objekt	D7
3.1.3.	Das IN-Objekt	D7
3.2.	Konfigurationsmöglichkeiten	D8
3.2.1.	Konfigurationsmöglichkeiten zu Programm-Status (1)	D8
3.2.2.	Konfigurationsmöglichkeiten zu den Programm-Stati (2) und (3)	D8
3.2.3.	Konfigurationsmöglichkeiten zu den Programm-Stati (4) und (5)	D8
3.3.	Die Identifier der Kommunikationsobjekte	D9
4.	Anhang	D10
4.1.	Literaturhinweise	D10
4.2.	Bit-Timing auf CAN	D10
4.3.	Zusammenfassung zu den höheren Protokollschichten	D11
4.3.1.	Identifier der Kommunikationsobjekte in Abhängigkeit der Teilnehmer-Adresse	D11
4.3.2.	Definition der Spezial-Parameter	D12
4.3.3.	Beschreibung der Kommunikationsobjekte in CAN-Message-Specification (CMS)	D14
4.3.4.	Kodierung der DRIVECOM-Parameter 'Steuer-/Statuswort' und 'Soll/Istdrehzahl'	D15

1. ALLGEMEINES

Die vorliegenden Unterlagen sowie die angegebene Hard- und Software sind Entwicklungen der Karl E. Brinkmann GmbH. Irrtum vorbehalten. Die Karl E. Brinkmann GmbH hat diese Unterlagen, die Hard- und Software nach bestem Wissen erstellt, übernimmt aber nicht die Gewähr dafür, daß die Spezifikationen den vom Anwender angestrebten Nutzen erbringen. Die Karl E. Brinkmann GmbH behält sich das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern oder Dritte davon in Kenntnis zu setzen.

1.1. CAN-Interface

KEB-Antriebstechnik entwickelt, produziert und vertreibt weltweit statische Frequenzumrichter im industriellen Leistungsbereich. Die Umrichter des Typs F1/ F4 können optional mit einer CAN-Schnittstelle ausgerüstet werden.

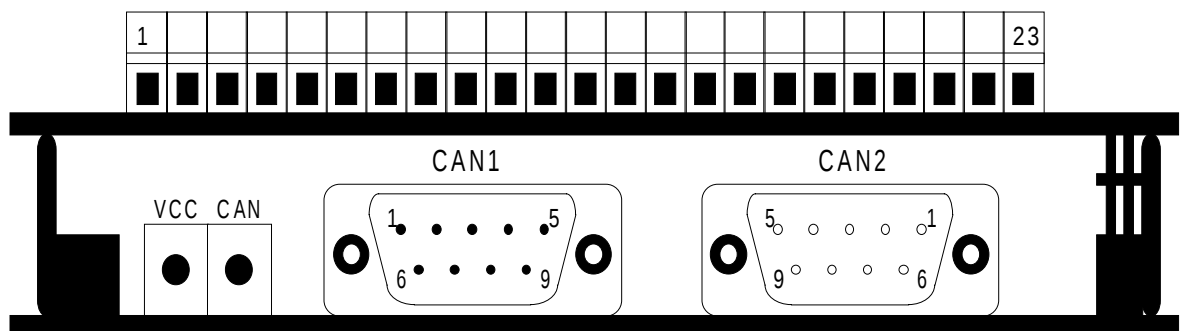
2. Die Hardware

Die CAN-Anschaltbaugruppe wird unter die Steuerkarte des KEB-Frequenzumrichters gebaut. Als Verbindung dient eine 26-polige Stiftleiste (F4) bzw. eine VG48-Leiste (F1). Nach Aussen bietet die CAN-Anschaltbaugruppe je einen SUBD-9-Stecker und eine SUBD-9-Buchse für die CAN Zu- und Weiterführung. Diese beiden Stecker sind in ihrer Anschlußbelegung identisch. So kann auch sehr leicht der CAN Busabschluß an der Buchse durchgeführt werden. U. a. Bilder zeigen jeweils die untergebaute CAN-Anschaltbaugruppe in COMBIVERT F1 sowie F4.

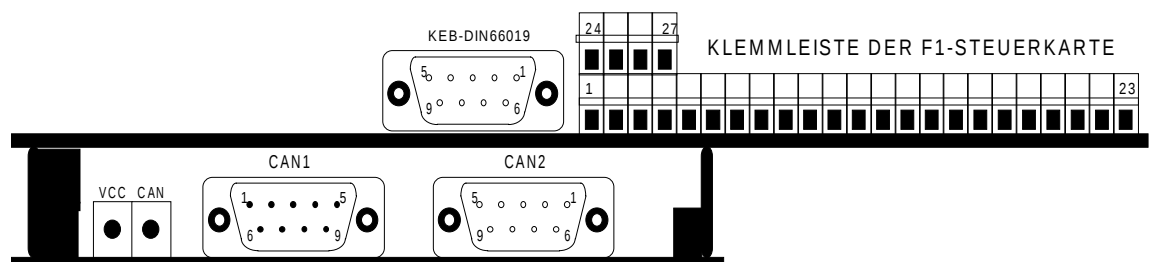
Auf der CAN-Anschaltbaugruppe befinden sich

- eine grüne LED zur Anzeige einer stabilen Versorgungsspannung (VCC).
- eine gelbe LED zur Signalisierung von Aktivitäten auf dem CAN-Bus. Sie wird bei Empfang eines gültigen Anforderungstelegramm gesetzt und beim Absenden des zugehörigen Antwort-Telegrammes wieder gelöscht. Beachte! Prozessdatenkommunikation kann über diese LED nicht kontrolliert werden, sondern nur die Kommunikation über den bestätigten Dienst.

KLEMMLEISTE DER F4-STEUERKARTE



PLATINE DER CAN-ANSCHALTUNG



2.1 Zusammenfassung der technischen Daten

CAN-Konnektor 1 : SUB-MIN-9polig-MALE, DIN 41652 Teil 1
 : Belegung nach CiA CAN Physical-Layer for Industrial Applications Version 2.0:

Pin1	: Reserviert (hier nicht angeschlossen)
Pin2	: CAN_L (dominant low)
Pin3	: CAN_GND (hier nicht angeschlossen)
Pin4	: Reserviert (hier nicht angeschlossen)
Pin5	: CAN_SHLD (hier nicht angeschlossen)
Pin6	: GND (hier nicht angeschlossen)
Pin7	: CAN_H (dominant high)
Pin8	: Reserviert (hier nicht angeschlossen)
Pin9	: CAN_V+ (hier nicht angeschlossen)

CAN-Konnektor 2 : SUB-MIN-9polig-MALE, DIN 41652 Teil 1
 : Belegung : s.o.

ACHTUNG ! Entgegen der o.g. Vorschrift des Arbeitskreises Physical-Layer der CiA sind die nicht angeschlossenen Pins von CAN-Konnektor 1 und 2 untereinander NICHT verbunden !
Das Durchschleifen von optionalen Signalen oder der Versorgungsspannung ist hier deshalb nicht möglich.

Pegel auf CAN : nach ISO/DIS11898 (PCA82C250)
 Baudrate : Einstellbar über CAN (10,20,50,100,125,250,500(1) Kbit/s).
 Potentialtrennung : 2.5 KV zwischen CAN-Controller und CAN-Treiber (1) .

3. Die Software

Das CAN-Protokoll ist bis zur Schicht 2 definiert. Die Abarbeitung dieses Protokolls übernimmt vollständig ein IC. Die Definitionen für eine einheitliche höhere Protokollschicht sind abgeschlossen. Die KEB-CAN-Schnittstelle hält sich an diese Vereinbarungen der CAN in Automation Users-Group (CiA).

3.1. Die Kommunikationsobjekte

3.1.1. Das READ_WRITE-Objekt

Eine grundlegende Eigenschaft des CAN-Protokolls ist es, daß die Nachrichten, die über den BUS gesendet werden, objektadressiert und nicht teilnehmer-adressiert sind. Da es in der Antriebstechnik jedoch notwendig ist, eine Stations-Adressierung durchzuführen, wurde dies in der vorliegenden Software durch eine feste, direkt mit der Teilnehmer-Adresse des FU verbundene Identifier-Vergabe realisiert.

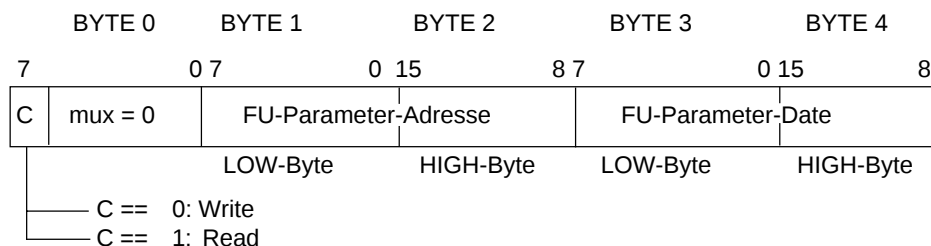
Jedem Frequenzumrichter werden vier Nachrichten (Identifier) zugewiesen.

Über dieses Objekt können beim FU bestätigte Read/Write-Anforderungen abgesetzt werden. Das bedingt, daß dieses Objekt auf zwei Identifier abgebildet wird. Einen für den Request (logischer Master --> FU) und einen zweiten für die zugehörige Response (FU -> logischer Master). Die den Dienst-Primitiven zugeordneten Identifier werden im Folgenden als Request-Identifier und Response-Identifier bezeichnet.

An dieser Stelle wird nur die Codierung des Datenbereichs für die beiden Identifier erläutert. Eine Zusammenfassung der Beschreibungsdaten zu diesem Objekt findet sich im Anhang. Generell gilt, daß in dem Request dem FU ein Kommando-Code (C) übergeben wird sowie eine FU-Parameter-Adresse und beim Schreiben zusätzlich die zu schreibenden Daten. Die Beschreibung der FU-Parameter, ihrer Adressen und Datenbereiche sind bis auf wenige Ausnahmen mit dem PRINT-Programm aus dem COMBIVIS-Programmpaket zu erstellen.

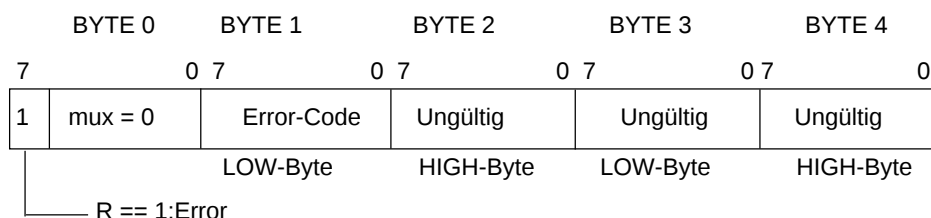
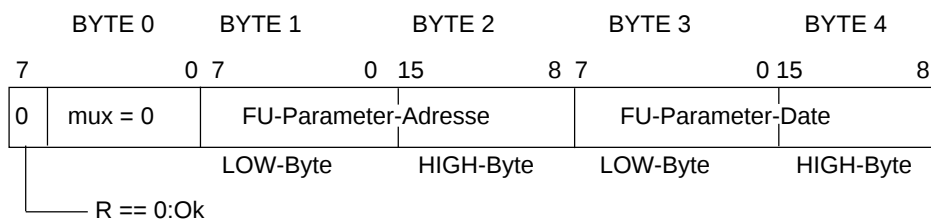
Die eben genannten Ausnahmen sind Spezial-Parameter, die nicht im FU sondern auf der CAN-Bus-Anschaltung lokalisiert sind. Die Beschreibung dieser Parameter findet sich im Anhang.

Codierung des Datenblocks im Request-Identifizier (CMS-kompatibel)



C : Command-Specifier(0=Schreiben).
 mux : Multiplexer (Unteradresse) ist konstant Null bei basic-Variablen.
 FU-Parameter-Adresse : Adresse des zu lesenden/schreibenden Parameters beim FU. Datentyp ist UNSIGNED16.
Beachte! Das LOW-Byte liegt auf der niedrigeren Adresse im Datenbereich.
 FU-Parameter-Date : Date des zu lesenden/schreibenden Parameters beim FU. Datentyp ist INTEGER16.
Beachte! LOW-Byte liegt auf der niedrigeren Adresse im Datenbereich.

Codierung des Datenblocks im Response-Identifizier (CMS-kompatibel)



R : Result (0 = OK).
 mux : Multiplexer ist konstant Null bei basic-Variablen.
 FU-Parameter-Adresse : Adresse des gelesenen/geschriebenen Parameters beim FU. Datentyp ist UNSIGNED16.
Beachte! Das LOW-Byte liegt auf der niedrigeren Adresse im Datenbereich.
 FU-Parameter-Date : Date des gelesenen/geschriebenen Parameters beim FU. Datentyp ist INTEGER16.
Beachte! Das LOW-Byte liegt auf der niedrigeren Adresse im Datenbereich.
 Error-Code : Im Falle R==1 wird im BYTE 1 des Datenbereichs der Grund für die Nichtausführung des Dienstes angegeben:
 Error-Code : Bedeutung

- 2: Angesprochene Parameter-Adresse ist nicht vorhanden oder Passwort-geschützt.
- 3: Bei Schreiben. FU-Parameter-Date hat ungültigen Wert.
- 4: Bei Schreiben. Parameter ist Schreibgeschützt.
- 5: Umrichter-Ausfall.
- 6: Umrichter beschäftigt.

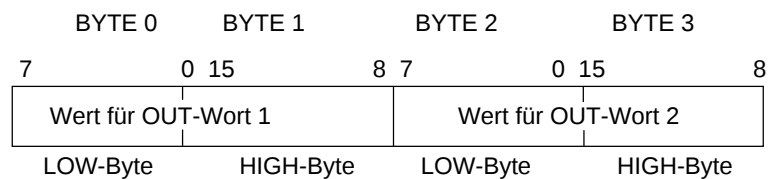


Keine parallelen Dienste !!!
 Parallele Dienste sind hier nicht zulässig. D.h., daß nach einem Request der Client (Dienstanforderer) zunächst die Response des Servers (Dienstbringer) abwarten muß, bevor er einen neuen Request aussendet.

3.1.2 Das OUT-Objekt

Über dieses Kommunikationsobjekt werden dem Umrichter neue Vorgaben für die beiden OUT-Worte übermittelt. OUT-Wort bedeutet, daß die Daten vom logischen CAN-Bus-Master zum FU gerichtet sind. Dieser Identifier besitzt 4 Byte (2 Worte) Daten. Wort 1 wird auf eine feste Parameter-Adresse (OUT1-Adresse) im FU abgebildet, ebenso Wort 2 (OUT2-Adresse). Der Reset-Zustand für OUT1-Adresse = 0106 (hex) = Steuerwort gemäß DRIVECOM-Profil für die Antriebstechnik. (s. Anhang). Der Reset-Zustand für OUT2-Adresse = 0108 (hex) = Solldrehzahl gemäß DRIVECOM-Profil für die Antriebstechnik. (s. Anhang). Die OUT-Wort-Behandlung kann über den Parameter 'Outword_Activation' (Parameter-Adresse 0087 (hex)) an- und abgeschaltet werden. Bei abgeschalteter OUT-Wort-Behandlung nimmt der Frequenzumrichter keine neuen Werte für die beiden OUT-Worte mehr an (siehe auch Definition der Spezialparameter).

Codierung des Datenblocks im OUT-Identifier (CMS-kompatibel)



Der Datentyp für OUT-Wort 1,2 ist INTEGER16.

!Beachte! Das LOW-Byte liegt auf der niedrigeren Adresse im Datenbereich. Wenn einer dieser Werte über den CAN-BUS geändert wird, schreibt die CAN-Anschaltung diesen Wert zum Umrichter:

```

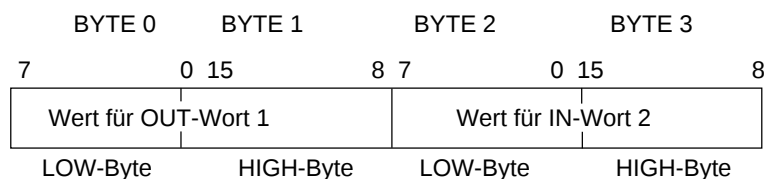
Parameter-
Adresse , Date
WRITE(OUT1-Adresse,OUT-Wort 1) bzw.

WRITE(OUT2-Adresse,OUT-Wort 2).
```

3.1.3 Das IN-Objekt

Hierbei handelt es sich um ein Kommunikationsobjekt, das wiederum auf zwei FU-Parameter abgebildet wird. Die Bezeichnung 'IN' verrät, daß hier die Datenrichtung vom Frequenzumrichter zum logischen CAN-BUS-Master weist. D.h. die CAN-Anschaltung liest zyklisch zwei Parameter vom FU (IN1-Adresse, IN2-Adresse). Bei Änderung einer der Werte wird dieser im IN-Identifier eingetragen und ein DATA-Frame mit dem entsprechenden Identifier auf den CAN-BUS gesendet, wenn der AUTO_IN_MESSAGE-Modus eingeschaltet ist (Parameter AUTO_IN_MESSAGE = ON). Ist dieser Modus abgeschaltet, muß der logische CAN-Bus-Master einen Remote-Frame auf dem IN-Identifier absetzen, um die aktuellen Werte zu erhalten (siehe auch Spezial-Parameter-Beschreibung im Anhang). Der Reset-Zustand für IN1-Adresse = 0107 (hex) = Statuswort gemäß DRIVECOM-Profil für die Antriebstechnik. (s. Anhang). Der Reset-Zustand für IN2-Adresse = 0109 (hex) = Drehzahl gemäß DRIVECOM-Profil für die Antriebstechnik. (s. Anhang). Die IN-Wort-Behandlung kann über den Parameter 'Inword_Activation' (Parameter-Adresse 0086 (hex)) an- und abgeschaltet werden. D.h. bei abgeschalteter Inword_Activation werden die beiden IN-Worte nicht zyklisch von der Frequenzumrichter-Steuerung gelesen und somit auch nicht zum CAN-Bus hin aktualisiert.

Codierung des Datenblocks im IN-Identifier (CMS-kompatibel)



Der Datentyp für IN-Wort 1,2 ist INTEGER16.

!Beachte! Das LOW-Byte liegt auf der niedrigeren Adresse im Datenbereich.

3.2. Konfigurationsmöglichkeiten

In der vorliegenden Software sind einige z.T. schon erwähnte Konfigurationen durchführbar, die es ermöglichen, den Ablauf der zyklischen Bearbeitung auf den jeweiligen Einsatzfall optimal abzustimmen. In diesem Kapitel sollen diese Einstellmöglichkeiten nochmals zusammengefaßt dargestellt werden. Der generelle Ablauf des Hauptprogramms im CAN-Interface ist folgender:

Endlos-Schleife

- (1) Bei vorliegendem Request (Lesen od. Schreiben eines Parameters) führe diesen aus.
- (2) Wenn sich der Wert des OUT-Wortes 1 geändert hat, schreibe diesen zur Umrichter-Steuerung.
- (3) Wenn sich der Wert des OUT-Wortes 2 geändert hat, schreibe diesen zur Umrichter-Steuerung.
- (4) Lies IN-Wort 1 von der Umrichter-Steuerung und aktualisiere den Wert falls verändert zum CAN-Bus.
- (5) Lies IN-Wort 2 von der Umrichter-Steuerung und aktualisiere den Wert falls verändert zum CAN-Bus.

3.2.1. Konfigurationsmöglichkeiten zu Programm-Status (1)

Die Abarbeitung von allgemeinen Requests, die über den Request-Identifizier empfangen wurden, ist NICHT abschaltbar. Es ist jedoch möglich der Request-Behandlung Priorität gegenüber der IN-Wort-Aktualisierung zu geben. Dies geschieht durch Setzen des Spezial-Parameters 'Request_Priority' (Parameter-Adresse 008A (hex)) auf 'ON' (FFFF (hex)) (siehe auch Definition der Spezialparameter). In diesem Fall wird vor Programm-Status (4) und (5) bei vorliegendem Request zunächst die Ausführung dieses Requests eingeschoben.

3.2.2. Konfigurationsmöglichkeiten zu den Programm-Status (2) und (3)

Die OUT-Wort-Bearbeitung ist generell an-/abschaltbar über den Spezial-Parameter 'Outword_Activation' (Parameter-Adresse 0087 (hex)). Bei Einstellung 'OFF' (siehe auch Definition der Spezialparameter) werden die Punkte (2) und (3) übersprungen. Das CAN-Interface empfängt dann auch keine Telegramme mehr auf dem OUT-Identifizier. Es ist auch hier möglich der OUT-Wort-Bearbeitung Vorrang zu erteilen gegenüber der IN-Wort-Aktualisierung. Dies geschieht durch Setzen des Spezial-Parameters 'Outword_Priority' (Adresse 0089 (hex)) auf 'ON' (FFFF (hex)). In diesem Fall wird bei veränderten Werten für OUT-Wort 1 oder 2 und bei eingeschalteter 'Outword_Activation' zunächst der geänderte Wert zur Umrichter-Steuerung geschrieben.

3.2.3 Konfigurationsmöglichkeiten zu den Programm-Status (4) und (5)

Die IN-Wort-Bearbeitung ist generell an-/abschaltbar über den Spezialparameter 'Inword_Activation' (Parameter-Adresse 0086 (hex)). Bei Einstellung 'OFF' (siehe auch Definition der Spezialparameter) werden die Punkte (4) und (5) übersprungen. Weiterhin kann bei eingeschalteter IN-Wort-Bearbeitung (Inword_Activation = ON) die Ausführung dieser durch einen vorliegenden Request (wenn 'Request_Priority' = ON) sowie durch die Aktualisierung veränderter OUT-Worte (wenn 'Outword_Priority' = 'ON') zeitlich verzögert werden.

Zusätzlich kann konfiguriert werden, ob das CAN-Interface bei eingeschalteter IN-Wort-Bearbeitung und veränderten Werten für eines der IN-Worte dies durch eine Message (mit dem IN-Identifizier) auf dem CAN-Bus meldet, oder dies nur in dem entsprechenden Message-Buffer aktualisiert. Dies geschieht durch Einstellung des Spezial-Parameters 'Auto_In_Message' (Parameter-Adresse 0085 (hex)) auf 'ON' (FFFF (hex)) für das zuerst genannte Verhalten (siehe auch Definition der Spezialparameter). Bei 'Auto_In_Message' = 'OFF' können bei eingeschalteter IN-Wort-Bearbeitung in jedem Fall durch Aussenden eines Remote-Frames auf dem IN-Identifizier die aktuellen Werte der beiden IN-Worte abgefragt werden.

3.3. Die Identifizierung der Kommunikationsobjekte

Wie oben schon erwähnt, werden die o.g. Kommunikationsobjekte für jeden Frequenzumrichter auf festen Identifiern abgelegt. Dabei bestimmen sich die Identifizierung direkt aus der Umrichter-Adresse des Frequenzumrichters (siehe dazu Parameterliste des Frequenzumrichters).

Der Zusammenhang zwischen dem Inhalt dieses Parameters und den Identifiern für die vier CAN-Nachrichten des FU ist folgender:

REQUEST_ID	=	(Umrichter-Adresse * 4) + 1.
RESPONSE_ID	=	(Umrichter-Adresse * 4) + 2.
OUT_ID	=	(Umrichter-Adresse * 4) + 3.
IN_ID	=	(Umrichter-Adresse * 4) + 4.

Beispiel:

Ein Frequenzumrichter mit der Umrichter-Adresse $t_adr = 54$ (dez) kann über folgende Identifizierung angesprochen werden:

REQUEST_ID	=	$(54 * 4) + 1 = 217$ (dez).
RESPONSE_ID	=	$(54 * 4) + 2 = 218$ (dez).
OUT_ID	=	$(54 * 4) + 3 = 219$ (dez).
IN_ID	=	$(54 * 4) + 4 = 220$ (dez).

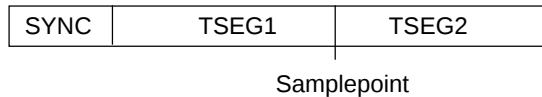
4. Anhang

4.1. Literaturhinweise

- DRIVECOM-Profil-Papier v. 16.12.1991
Herausgeber: DRIVECOM-Nutzergruppe e.V.
Postfach 1102, 32825 Blomberg, Art.-Nr: DRI 21
- Dokument zur Vereinbarung des Arbeitskreises Physical-Layer der CiA (CAN in Automation User-Group):
- CiA/DS 102-1
Herausgeber: CiA International Users and Manufacturers Group e. V.
Simon-Schöffel-Str. 21 , 90427 Nürnberg
- Dokumente zur Vereinbarung des Arbeitskreises Higher-Layer Protocols der CiA (Herausgeber s. o.):
 - CiA/WG2/DS201 : CAN in the OSI Reference Model.
 - CiA/WG2/DS202-1 : CMS Service Specification.
 - CiA/WG2/DS202-2 : CMS Protocol Specification.
 - CiA/WG2/DS202-3 : CMS Encoding Rules.
 - CiA/WG2/DS203-1 : NMT Service Specification.
 - CiA/WG2/DS203-2 : NMT Protocol Specification.
 - CiA/WG2/DS204-1 : DBT Service Specification.
 - CiA/WG2/DS204-2 : DBT Protocol Specification.
 - CiA/WG2/DS207 : Application Layer Naming Conventions.

4.2 Bit-Timing auf CAN

Das nominale Bit-Timing sieht wie folgt aus:



Für alle einstellbaren Baudraten gilt:

- t_q : Zeiteinheit aus denen die einzelnen Segmente aufgebaut sind.
- SYSC = 0 ==> Nur die Flanken von Rezessiv zu Dominant werden zur Synchronisation genutzt
- SJW = 0 ==> Synchronisations-Sprungweite = $1t_q$
- TSEG2 = 1 ==> $t_{SEG2} = 2 * t_q$
- TSEG1 = 12 ==> $t_{SEG1} = 13 * t_q$

==> Länge eines Bit = $16 * t_q$, Sample-Point = $14 * t_q$

Baudrate	Time-Quantum (t_q)	Sample-Point
10 KBit/s	6,25 μ s	$14 * t_q = 87,5 \mu$ s
20 KBit/s	3,125 μ s	$14 * t_q = 43,75 \mu$ s
50 KBit/s	1,25 μ s	$14 * t_q = 17,5 \mu$ s
100 KBit/s	625 ns	$14 * t_q = 8,75 \mu$ s
125 KBit/s	500 ns	$14 * t_q = 7,0 \mu$ s
250 KBit/s	250 ns	$14 * t_q = 3,5 \mu$ s
500 KBit/s ¹⁾	125 ns	$14 * t_q = 1,75 \mu$ s

¹⁾ Der Arbeitskreis Physical-Layer der CiA empfiehlt bei Einsatz von Optokopplern zur Potentialtrennung der CAN-Anschaltung als Obergrenze für CAN-Baudrate 250 kBit/s einzustellen. Da in der KEB-CAN-Anschaltung Optokoppler eingesetzt werden, sollte die Empfehlung befolgt werden, obwohl höhere Werte möglich sind.

4.3. Zusammenfassung zu den höheren Protokollschichten

4.3.1 Identifier der Kommunikationsobjekte in Abhängigkeit der Teilnehmer-Adresse

Name	Identifier	Daten-Länge (Byte)	Aufbau des Datenfeldes
REQUEST_ID	$(t_adr * 4) + 1$	5	01234 Com. PARA-ADR PARA-DATE LB HB LB HB
RESPONSE_ID	$(t_adr * 4) + 2$	5	01234 Res. PARA-ADR PARA-DATE LB HB LB HB
OUT_ID	$(t_adr * 4) + 3$	4	0123 OUT-Wort1 OUT-Wort2 LB HB LB HB
IN_ID	$(t_adr * 4) + 4$	4	0123 OUT-Wort1 OUT-Wort2 LB HB LB HB

4.3.2 Definition der Spezial-Parameter

Parameter-Adresse	Bedeutung	Codierung	Default-Wert																
0080h	Bestimmt die Baudrate auf dem CAN-Bus. Ein veränderter Wert wird sofort aktiv aber nicht sofort nichtflüchtig gespeichert. Das nichtflüchtige Speichern einer CAN-Baudrate erfolgt explizit über den Parameter "Save_CAN_Baud" (s.u.)	<table><tr><td>Wert</td><td>Baudrate</td></tr><tr><td>0</td><td>10KBit/s</td></tr><tr><td>1</td><td>20KBit/s</td></tr><tr><td>2</td><td>50KBit/s</td></tr><tr><td>3</td><td>100KBit/s</td></tr><tr><td>4</td><td>125KBit/s</td></tr><tr><td>5</td><td>250KBit/s</td></tr><tr><td>6</td><td>500KBit/s</td></tr></table>	Wert	Baudrate	0	10KBit/s	1	20KBit/s	2	50KBit/s	3	100KBit/s	4	125KBit/s	5	250KBit/s	6	500KBit/s	1
Wert	Baudrate																		
0	10KBit/s																		
1	20KBit/s																		
2	50KBit/s																		
3	100KBit/s																		
4	125KBit/s																		
5	250KBit/s																		
6	500KBit/s																		
0081h	'OUT1-ADRESSE': Parameter-Adresse auf die das OUT-Wort 1 abgebildet wird. Beachte!!! Spezialparameter können nicht auf In-oder OUT-Worte abgebildet werden! Wert wird nichtflüchtig abgespeichert und sofort aktiv.	(siehe Parameterliste der FU-Steuerkarte).	0106h																
0082h	'OUT2-ADRESSE': Parameter-Adresse auf die das OUT-Wort 2 abgebildet wird. Beachte!!! Spezialparameter können nicht auf IN-oder OUT-Worte abgebildet werden! Wert wird nichtflüchtig abgespeichert und sofort aktiv.	(siehe Parameterliste der FU-Steuerkarte).	0108h																
0083h	'IN1-Adresse': Parameter-Adresse auf die das IN-Wort 1 abgebildet wird. Beachte!!! Spezialparameter können nicht auf IN-oder OUT-Worte abgebildet werden! Wert wird nichtflüchtig abgespeichert und sofort aktiv.	(siehe Parameterliste der FU-Steuerkarte).	0107h																
0084h	'IN2-Adresse': Parameter-Adresse auf die das IN-Wort abgebildet wird. Beachte!!! Spezialparameter können nicht auf IN-oder OUT-Worte abgebildet werden! Wert wird nichtflüchtig abgespeichert und sofort aktiv.	(siehe Parameterliste der FU-Steuerkarte).	0109h																

Parameter-Adresse	Bedeutung	Codierung	Default-Wert
0085h	<i>'Auto_In_Message':</i> Bestimmt, ob bei Änderung eines der IN-Worte automatisch eine Message auf den CAN Bus gesendet wird (ON) oder nicht (OFF). Bei Einstellung OFF muß der logische Master die aktuellen Daten durch einen Remote-Frame auf dem IN-Identifizier anfordern.	0: OFF FFFFh: ON	OFF
0086h	<i>'Inword_Activation':</i> Aktivierung der IN-Wort-Behandlung: Wert wird nichtflüchtig abgespeichert und sofort aktiv.	0: OFF FFFFh: ON	OFF
0087h	<i>'Outword_Activation':</i> Aktivierung der OUT-Wort-Behandlung. Wert wird nichtflüchtig abgespeichert und sofort aktiv.	0: OFF FFFFh: ON	OFF
0088h	<i>'EEPROM_Check'</i> Information über das Ergebnis des EEPROM-Tests. Dieser Parameter kann nur gelesen werden. Ist der Wert=0, so kann keine der Konfigurationen nichtflüchtig abgespeichert werden. In diesem Fall läuft das CAN-Interface mit der Default-Einstellung.	0: EEPROM fehlerhaft FFFFh: EEPROM fehlerfrei	
0089h	<i>'Outword_Priority' :</i> Bei Einstellung 'ON' und 'Outword-Activation' = 'ON' hat die Aktualisierung eines geänderten OUT-Wortes zur Umrücker-Steuerung Vorrang vor der zyklischen IN-Wort-Bearbeitung. Wert wird nichtflüchtig abgespeichert und sofort aktiv.	0: OFF FFFFh: ON	OFF
008Ah	<i>'Request_Priority':</i> Bei Einstellung 'ON' und vorliegendem Request vom CAN-Bus hat die Bearbeitung dieses Requests Vorrang vor der zyklischen IN-Wort-Bearbeitung. Wert wird nichtflüchtig abgespeichert und sofort aktiv.	0: OFF FFFFh: ON	OFF
008Bh	<i>'Save_CAN_Baud'</i> Ein Schreibzugriff auf diesen Parameter mit dem Wert 'FFFFh' bewirkt das nichtflüchtige Abspeichern der CAN-Baudrate (s. P-Adr. 0080h)	0: OFF FFFFh: ON	OFF

4.3.3. Beschreibung" der Kommunikationsobjekte in CAN-Message-Specification (CMS)

CMS-Beschreibung des Objekts: KEB_Object

- Objekt-type : VARIABLE
- Name : '#KEB_Object'
- Data-Type : STRUCTURE OF UNSIGNED16,INTEGER16
- Error-Type : UNSIGNED8
- Class : BASIC
- User-Type : SERVER
- Access-Type : READ_WRITE
- Priority : In Abhängigkeit von der Slave-Adresse beginnend bei Priority = 0.
- Inhibit-Time : z. Zt. noch keine Festlegungen

CMS-Beschreibung des Objekts: KEB_Out

- Objekt-type : VARIABLE
- Name : '#KEB_Out'
- Data-Type : ARRAY[2] OF INTEGER16
- Class : BASIC
- User-Type : SERVER
- Access-Type : WRITE-ONLY
- Priority : In Abhängigkeit von der Slave-Adresse beginnend bei Priority = 0.
- Inhibit-Time : z. Zt. noch keine Festlegungen.

CMS-Beschreibung des Objekts: KEB_In

- Objekt-type : EVENT
- Name : '#KEB_In'
- Data-Type : ARRAY[2] OF INTEGER16
- Class : STORED
- User-Type : SERVER
- Priority : In Abhängigkeit von der Slave-Adresse beginnend bei Priority = 0.
- Inhibit-Time : z. Zt. noch keine Festlegungen

4.3.4 Kodierung der DRIVECOM-Parameter 'Steuer-/Statuswort' und 'Soll/Istdrehzahl'

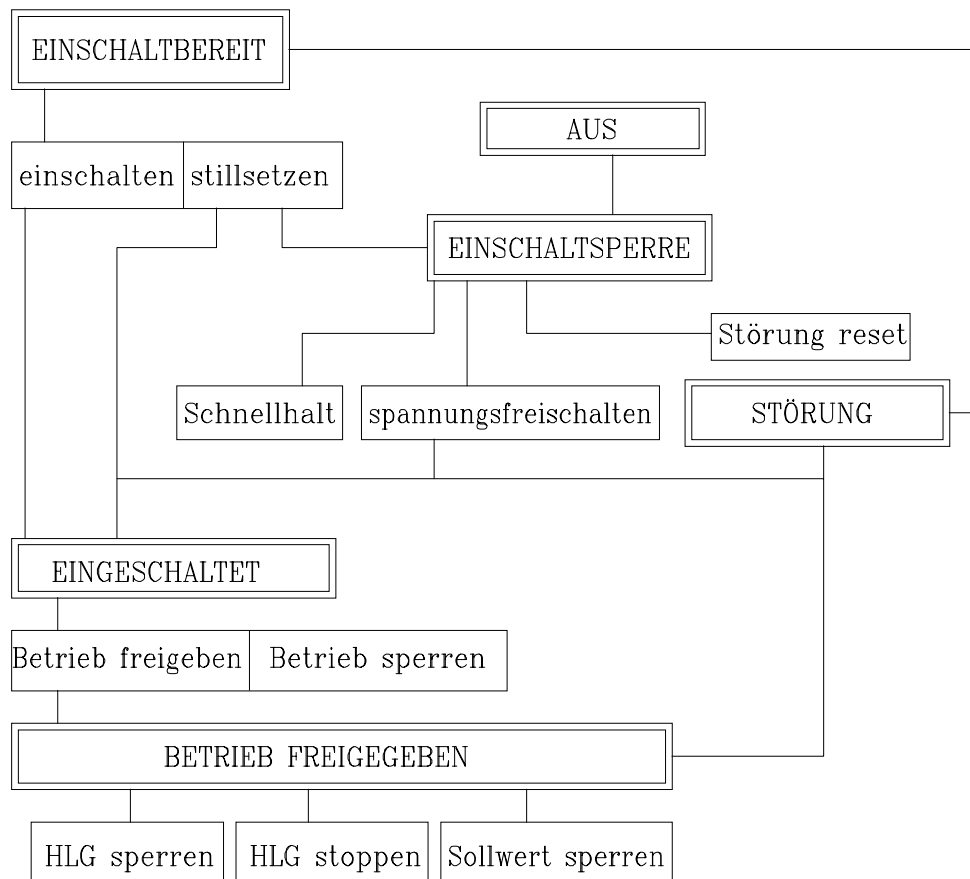
<i>Name</i>	:	Steuerwort																																																
<i>Bedeutung</i>	:	Steuerwort-Steuert die Betriebszustände des Umrichters. (siehe auch Zustandsdiagramm Status/Steuerwort)																																																
<i>Parameter-Adresse</i>	:	0106 (hex)																																																
<i>Kodierung</i>	:	<table> <tr> <td>Bit 0</td><td>1=Einschalten</td><td>(0=Stillsetzen)</td></tr> <tr> <td>Bit 1</td><td>1=Spannung nicht freischalten</td><td></td></tr> <tr> <td>Bit 2</td><td>1=Kein Schnellhalt</td><td></td></tr> <tr> <td>Bit 3</td><td>1=Betrieb freigeben</td><td>(0=Betrieb sperren)</td></tr> <tr> <td>Bit 4</td><td>1=HLG nicht Sperren</td><td></td></tr> <tr> <td>Bit 5</td><td>1=HLG nicht Stoppen</td><td></td></tr> <tr> <td>Bit 6</td><td>1=Sollwert nicht sperren</td><td></td></tr> <tr> <td>Bit 7</td><td>0->1=Störung Reset</td><td></td></tr> <tr> <td>Bit 8</td><td>z.Zt keine Funktion</td><td></td></tr> <tr> <td>Bit 9</td><td>z.Zt keine Funktion</td><td></td></tr> <tr> <td>Bit 10</td><td>z.Zt keine Funktion</td><td></td></tr> <tr> <td>Bit 11</td><td>z.Zt keine Funktion</td><td></td></tr> <tr> <td>Bit 12</td><td>z.Zt keine Funktion</td><td></td></tr> <tr> <td>Bit 13</td><td>z.Zt keine Funktion</td><td></td></tr> <tr> <td>Bit 14</td><td>z.Zt keine Funktion</td><td></td></tr> <tr> <td>Bit 15</td><td>z.Zt keine Funktion</td><td></td></tr> </table>	Bit 0	1=Einschalten	(0=Stillsetzen)	Bit 1	1=Spannung nicht freischalten		Bit 2	1=Kein Schnellhalt		Bit 3	1=Betrieb freigeben	(0=Betrieb sperren)	Bit 4	1=HLG nicht Sperren		Bit 5	1=HLG nicht Stoppen		Bit 6	1=Sollwert nicht sperren		Bit 7	0->1=Störung Reset		Bit 8	z.Zt keine Funktion		Bit 9	z.Zt keine Funktion		Bit 10	z.Zt keine Funktion		Bit 11	z.Zt keine Funktion		Bit 12	z.Zt keine Funktion		Bit 13	z.Zt keine Funktion		Bit 14	z.Zt keine Funktion		Bit 15	z.Zt keine Funktion	
Bit 0	1=Einschalten	(0=Stillsetzen)																																																
Bit 1	1=Spannung nicht freischalten																																																	
Bit 2	1=Kein Schnellhalt																																																	
Bit 3	1=Betrieb freigeben	(0=Betrieb sperren)																																																
Bit 4	1=HLG nicht Sperren																																																	
Bit 5	1=HLG nicht Stoppen																																																	
Bit 6	1=Sollwert nicht sperren																																																	
Bit 7	0->1=Störung Reset																																																	
Bit 8	z.Zt keine Funktion																																																	
Bit 9	z.Zt keine Funktion																																																	
Bit 10	z.Zt keine Funktion																																																	
Bit 11	z.Zt keine Funktion																																																	
Bit 12	z.Zt keine Funktion																																																	
Bit 13	z.Zt keine Funktion																																																	
Bit 14	z.Zt keine Funktion																																																	
Bit 15	z.Zt keine Funktion																																																	

<i>Name</i>	:	Statuswort																																																
<i>Bedeutung</i>	:	Statuswort-Gibt den Status des Umrichters zurück. (siehe auch Zustandsdiagramm Status/Steuerwort)																																																
<i>Parameter-Adresse</i>	:	0107 (hex)																																																
<i>Kodierung</i>	:	<table> <tr> <td>Bit 0</td><td>1=Einschaltbereit</td><td></td></tr> <tr> <td>Bit 1</td><td>1=Eingeschaltet</td><td></td></tr> <tr> <td>Bit 2</td><td>1=Betrieb freigegeben</td><td></td></tr> <tr> <td>Bit 3</td><td>1=Störung</td><td></td></tr> <tr> <td>Bit 4</td><td>1=Keine Spannungsfreischaltung</td><td></td></tr> <tr> <td>Bit 5</td><td>1=Kein Schnellhalt</td><td></td></tr> <tr> <td>Bit 6</td><td>1=Einschaltsperr</td><td></td></tr> <tr> <td>Bit 7</td><td>1=Warnung</td><td></td></tr> <tr> <td>Bit 8</td><td>z.Zt keine Bedeutung</td><td></td></tr> <tr> <td>Bit 9</td><td>1=Keine Lokalbedienung aktiv</td><td></td></tr> <tr> <td>Bit 10</td><td>1=Sollwert erreicht</td><td></td></tr> <tr> <td>Bit 11</td><td>1=Begrenzung aktiv</td><td></td></tr> <tr> <td>Bit 12</td><td>z.Zt keine Bedeutung</td><td></td></tr> <tr> <td>Bit 13</td><td>z.Zt keine Bedeutung</td><td></td></tr> <tr> <td>Bit 14</td><td>z.Zt keine Bedeutung</td><td></td></tr> <tr> <td>Bit 15</td><td>Handshake für Prozeßausgangsdaten</td><td></td></tr> </table>	Bit 0	1=Einschaltbereit		Bit 1	1=Eingeschaltet		Bit 2	1=Betrieb freigegeben		Bit 3	1=Störung		Bit 4	1=Keine Spannungsfreischaltung		Bit 5	1=Kein Schnellhalt		Bit 6	1=Einschaltsperr		Bit 7	1=Warnung		Bit 8	z.Zt keine Bedeutung		Bit 9	1=Keine Lokalbedienung aktiv		Bit 10	1=Sollwert erreicht		Bit 11	1=Begrenzung aktiv		Bit 12	z.Zt keine Bedeutung		Bit 13	z.Zt keine Bedeutung		Bit 14	z.Zt keine Bedeutung		Bit 15	Handshake für Prozeßausgangsdaten	
Bit 0	1=Einschaltbereit																																																	
Bit 1	1=Eingeschaltet																																																	
Bit 2	1=Betrieb freigegeben																																																	
Bit 3	1=Störung																																																	
Bit 4	1=Keine Spannungsfreischaltung																																																	
Bit 5	1=Kein Schnellhalt																																																	
Bit 6	1=Einschaltsperr																																																	
Bit 7	1=Warnung																																																	
Bit 8	z.Zt keine Bedeutung																																																	
Bit 9	1=Keine Lokalbedienung aktiv																																																	
Bit 10	1=Sollwert erreicht																																																	
Bit 11	1=Begrenzung aktiv																																																	
Bit 12	z.Zt keine Bedeutung																																																	
Bit 13	z.Zt keine Bedeutung																																																	
Bit 14	z.Zt keine Bedeutung																																																	
Bit 15	Handshake für Prozeßausgangsdaten																																																	

<i>Name</i>	:	Solldrehzahl
<i>Parameter-Adresse</i>	:	0108 (hex)
<i>Kodierung</i>	:	Ist normiert in 1/min.

<i>Name</i>	:	(Ist-)Drehzahl
<i>Parameter-Adresse</i>	:	0109 (hex)
<i>Kodierung</i>	:	Ist normiert in 1/min.

Zustandsdiagramm für Steuer/Statuswort



HLG: Hochlaufgeber oder Rampengenerator oder Steilheitsbegrenzer

1. General	E 4
1.1. CAN-Interface	E 4
2. The Hardware	E 5
3. The Software	E 5
3.1. The Communication Object	E 5
3.1.1. The READ_WRITE-Object	E 5
3.1.2. The OUT-Object	E 8
3.1.3. The IN-Object	E 7
3.2. Configuration Possibilities	E 8
3.2.1. Configuration Possibilities for Program Status (1)	E 8
3.2.2. Configuration Possibilities for the Program-Status (2) and (3)	E 8
3.2.3. Configuration Possibilities for the Program-Status (4) and (5)	E 8
3.3. The Identifier of Communication Objects	E 9
4. Appendix	E 10
4.1. Literary Information	E 10
4.2. Summary of the Physical Protocol Layer	E 10
4.3. Summary of the Higher Protocol Layers	E 11
4.3.1. Identifier of the Communication Objects Dependent on the Node Address	E 11
4.3.2. Definition of the Special-Parameter	E 12
4.3.3. Description of the Communication Objects in CAN-Message-Specification (CMS)	E 14
4.3.4. Coding of the DRIVECOM-Parameter 'Control-/Status Word' and 'Set/Real Speed'	E 15

1. General

The documentation as well as the hardware and software are developments of the Karl E. Brinkmann GmbH. Errors and omissions excepted! The Karl E. Brinkmann GmbH has prepared documentation, hardware and software to the best of their knowledge but no guarantee is given that the specifications will bring the user the efficiency aimed at. The Karl E. Brinkmann GmbH reserves the right to change the specifications without obligation. All rights reserved!

1.1. CAN-Interface

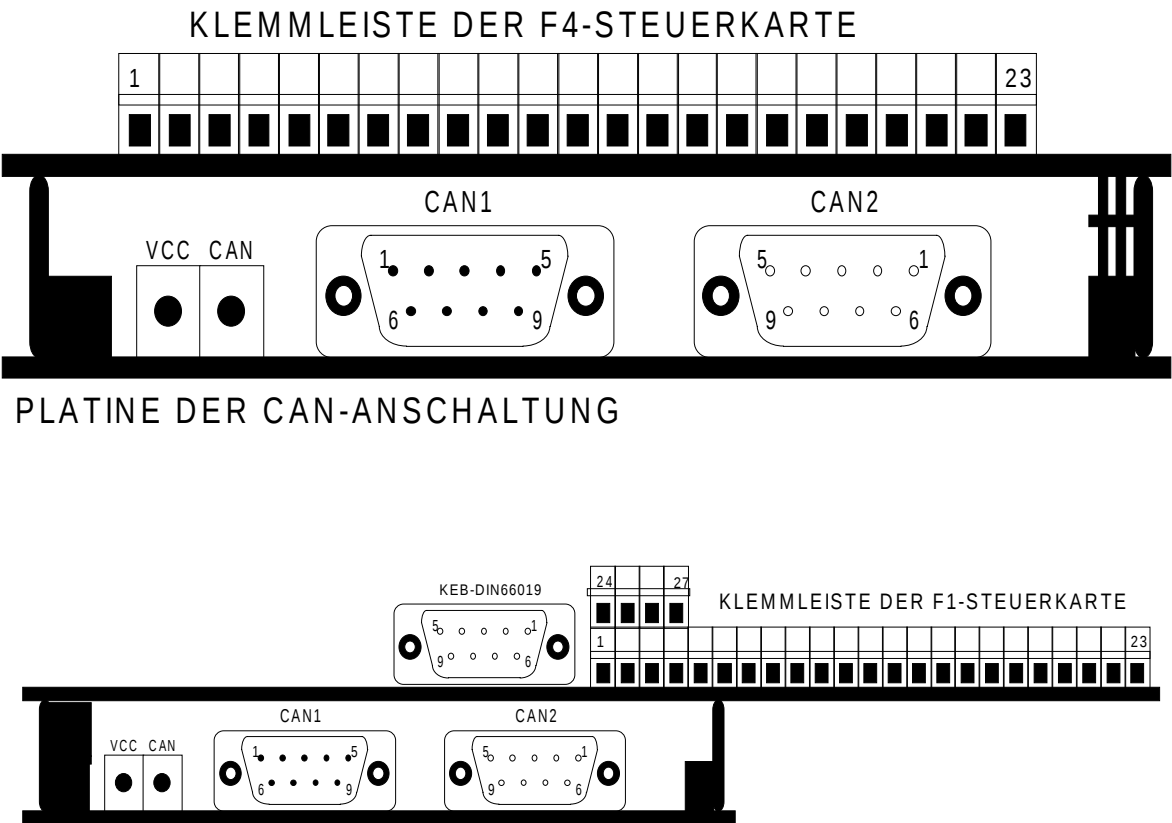
KEB-Antriebstechnik develops, produces and distributes static frequency inverters worldwide in the industrial power ranges. The inverter of type F4 / F1 can be optimally equipped with a CAN-interface.

2. The Hardware

The CAN-interface is below the control card of the KEB-frequency inverter. For connection a 26-pole plug connector (F4) or VG48-strip (F1) is used. The CAN-interface has one SUBD-9-plug and one SUBD-9-socket for CAN feeding and continuing the CAN-Bus. Both connectors have identical pin assignment. This makes it simple for the CAN Bus-termination to be fed through the socket. The figure below shows the CAN-interfaces in COMBIVERT F1 and F4.

The following LEDs are found on the CAN-interface assembly

- a green LED to display a stable supply voltage (VCC)
- a yellow LED to signal activities on the CAN-Bus. When a valid request telegram is received it is switched-on and switched-off when the respective answer-telegram is sent. Note! Process data communication cannot be controlled by this LED, but only the communication via the confirmed service.



2.1 Summary of the technical data

CAN-Connector 1 : SUB-MIN-9pole-MALE, DIN 41652 Part 1
 : Assignment in accordance with CiA CAN Physical-Layer for Industrial Applications Version 2.0:
 Pin1 : Reserved (not connected here)
 Pin2 : **CAN_L** (dominant low)
 Pin3 : CAN_GND (not connected here)
 Pin4 : Reserved (not connected here)
 Pin5 : CAN_SHLD (not connected here)
 Pin6 : GND (not connected here)
 Pin7 : **CAN_H** (dominant high)
 Pin8 : Reserved (not connected here)
 Pin9 : CAN_V+ (not connected here)

CAN-Connector 2 : SUB-MIN-9-pole-MALE, DIN 41652 Part 1
 : Assignment : see above

ATTENTION !

Contrary to the standards mentioned above of the working committee Physical-Layer of the CiA, the pins that are not connected in CAN-connector 1 + 2 are NOT connected to each another!

The looping-in of optional signals or of supply voltage is therefore not possible here.

Level on CAN : in accordance with ISO/DIS11898 (PCA82C250)
 Baud rate : set via CAN (10,20,50,100,125,250,500(1) Kbit/s).
 Isolation : 2.5 KV between CAN-controller and CAN-driver (1) .

3. The Software

The CAN-Protocol is defined up to layer 2. The realising of this protocol is completely taken over by an IC (in this case FULL-CAN-Controller Intel 82526 or SIEMENS SAE81C91). The definitions for a uniform higher protocol layer are completed. The KEB-CAN abides by the agreements of CAN in Automation Users-Group (CiA).

A fundamental property of CAN-protocol is, that the messages, which are sent through BUS, are object addressed and not node addressed. In power transmission, station-addressing is necessary and is done in this software by a fixed identifier distribution depending on the Bus address of the frequency inverter.

The bus address corresponds to the parameter 'inverter address display' (Parameter r.18) with the parameter-address 0212 (hex) on the control card. Each frequency inverter is assigned 4 messages (Identifier).

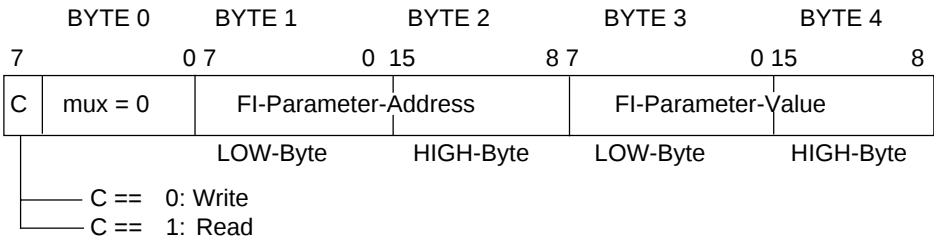
3.1. The Communication Object

3.1.1. The READ_WRITE-Object

In using this object confirmed Read/Write requests can be sent to the the frequency inverter. This requires, that the READ/WRITE object be represented on two identifiers. One for the request (logical master * frequency inverter) and a second for the corresponding response (frequency inverter * logical master). The identifier allocated to the service primitives are characterized as Request-Identifier and Response-Identifier.

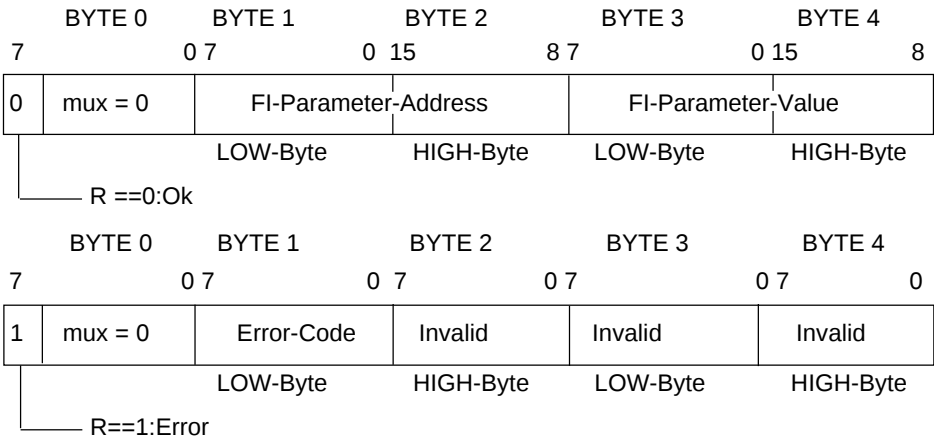
Only the coding of the data range for both identifiers will be described at this point. A summary of the READ/WRITE object is found in the appendix. In general, it can be said that there are 3 elements in the telegram: (1) commando-code (C), (2) frequency inverter parameter address, (3) the written data (this data is only present when a write request is made). With a few exceptions, the description of the frequency inverter parameters, their addresses and data ranges can be produced with the PRINT program from the COMBIVIS program package.

Coding of the Data Block in the Request-Identifier (CMS-compatible)



C : Command-Specifier (0=write).
mux : Multiplexor is constantly zero at basic variables.
FI-Parameter-Address : Address of the parameter to be read/written from/to the frequency inverter.
Data type is UNSIGNED16.
Note! The LOW-Byte lies on the lower address in the data range.
FI-Parameter-Date : Data of the parameter to be read/written from/to the frequency inverter.
Data type is INTEGER16.
Note! LOW-Byte lies on the lower address in the data range.

Coding of the Data Block in the Response-Identifier (CMS-compatible)



R : Result (0 = OK).
mux : Multiplexor is constantly zero at basic variables.
FI-Parameter-Address : Address of the read/written parameter from/to the frequency inverter.
Data type is UNSIGNED16.
Note! The LOW-Byte lies on the lower address in the data range.
FI-Parameter-Date : Data of the read/written parameter from/to the frequency inverter.
Data type is INTEGER16.
Note! The LOW-Byte lies on the lower address in the data range.
Error-Code : If R==1, then in BYTE 1 of the data range the reason the service was not completed is given:
Error-Code : Meaning
2: Desired parameter address is not available or is password protected.
3: In case of writing. Frequeency inverter parameter date has invalid values.
4: In case of writing. Parameter is write protected.
5: Inverter malfunction.
6: Inverter busy.

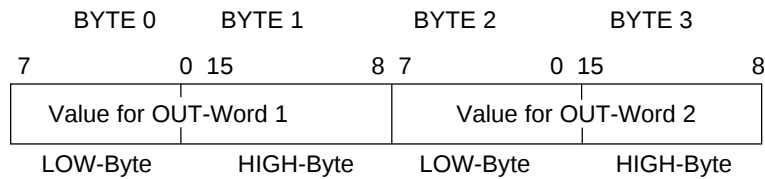


No parallel services !!! Parallel services are not allowed. This means that after a request of a client you must first wait for the response of the server before a new request is sent.

3.1.2 The OUT-Object

New data for both OUT words is transferred to the inverter by the OUT-object. The OUT-word means, that the values are directed from the logical CAN-Bus-Master to the frequency inverter. This identifier occupies 4 bytes (2 words) data. Word 1 is displayed on a fixed parameter address (OUT1-Address) in the frequency inverter, just as word 2 (OUT2-Address). The reset condition for OUT1-Address = 0106 (hex) = control word, according to DRIVECOM-profile for the power transmission (see appendix). The reset condition for OUT2-Address = 0108 (hex) = set speed, in accordance with DRIVECOM-Profile for power transmission (see appendix). The OUT-word processing can be switched-on/off by the parameter 'Outword_Activation'(parameter address 0087 (hex)). At switched-off OUT-word-processing the frequency inverter took on no new values for both OUT words (see definition of the special parameter).

Coding of the Data Block in the OUT-Identifier (CMS-compatible)



The data type for OUT-word 1, 2 is INTEGER16.

!Note! The LOW-Byte lies on the lower address in the data range. When one of these values is changed by the CAN-BUS, then the CAN-activation writes this value to the inverter:

```

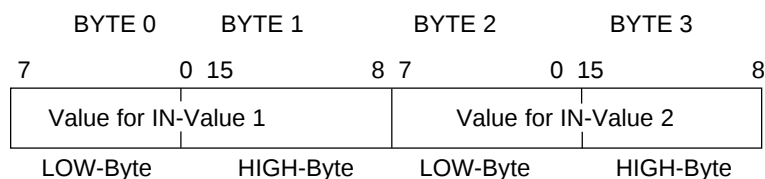
Parameter-
Address , Date
WRITE(OUT1-Address,OUT-word 1) and/or.

WRITE(OUT2-Address,OUT-word 2).
```

3.1.3 The IN-Object

This is a communication object, which is displayed on two frequency inverter parameters. The term 'IN' reveals, that the data course is divided from the frequency inverter to the logical Master. This means the CAN-switch-on, cyclically reads two parameters from the frequency inverter (IN1-Address, IN2-Address). When changing a value this is entered in the IN-identifier and a DATA-frame with the corresponding identifier is sent onto the CAN-BUS, when the AUTO_IN_MESSAGE-Mode is switched-on (Parameter AUTO_IN_MESSAGE = ON). If this mode is switched off, then the logical CAN-Bus-Master must deposit a remote-frame onto the IN-identifier, in order to maintain the actual value (see also the special parameter description in the appendix). The reset condition for IN1-Address = 0107 (hex) = status word in accordance with DRIVECOM-profile for power transmission (see appendix). The reset condition for IN2-Address = 0109 (hex) = speed in accordance with DRIVECOM-profile for the power transmission (see appendix). The IN-word processing can be switched on/off by the parameter 'Inword_Activation' (Parameter-Address 0086 (hex)). This means that when Inword_Activation is switched off both IN-Words are not cyclically read from the frequency inverter control and thus cannot be actualized to CAN-Bus.

Coding of the Data Blocks in the IN-Identifier (CMS-compatible)



The data type for IN-word 1,2 is INTEGER16.

!Note! The LOW-Byte lies on the lower address in the data range.

3.2. Configuration possibilities

In this software there are a few configurations which can be realized. They make it possible for the course of the cyclical processing to optimally coordinate onto the respective application. In this chapter these adjustment possibilities are once again summarized. The general course of the main program in CAN-interface is as follows:

Endless loop

- (1) With an existing request (reading or writing of a parameter) carry these out.
- (2) When the value of the OUT-word 1 has changed, write these to inverter-control.
- (3) When the value of the OUT-word 2 has changed, write these to inverter-control.
- (4) Read IN-word 1 from the inverter-control and actualize the value in case it is changed to CAN-Bus.
- (5) Read IN-word 2 from the inverter-control and actualize the value in case it is changed to CAN-Bus.

3.2.1. Configuration Possibilities for Program-Status (1)

The processing of general requests, which are received by the request identifier, CANNOT be switched off. It is possible to give the request-processing priority over the IN-word actualization. To do this set the special parameter 'Request_Priority' (Parameter-Address 008A (hex)) onto 'ON' (FFFF (hex)) (also see definition of the special parameter). In this case, depending on the existing request, the next completion of the request is inserted before program-status (4) and (5).

3.2.2. Configuration Possibilities for Program-Status (2) and (3)

The OUT-word-processing can generally be switched on/off by the special-parameter 'Outword_Activation' (Parameter-Address 0087 (hex)). At adjustment 'OFF' (also see definition of the special parameter) points (2) and (3) are skipped. The CAN-Interface cannot receive anymore telegrams onto the OUT-identifier. It is also possible to give the OUT-word processing priority over the IN-word actualization. This is done by setting the special parameters 'Outword_Priority' (Address 0089 (hex)) onto 'ON' (FFFF(hex)). In this case with a changed value, for OUT-word 1 or 2 and at a switched on 'Outword_Activation', the changed value is written to the inverter-control.

3.2.3 Configuration Possibilities for Program-Status (4) and (5)

The IN-word processing can generally be switched on/off by the special parameter 'Inword Activation' (Parameter-Address 0086(hex)). At the adjustment 'OFF' (also see definition of special parameter) points (4) and (5) are skipped over. Furthermore at a switched on IN-word processing (Inword activation = ON) its completion can be temporarily delayed by an existing request (when 'Request_Priority' = ON) as well as by the actualization of changed OUT-words (when 'Outward_Priority' = 'ON'). Additionally it can be configured, in case of switched on IN-word processing and changed values of one of the IN-words, whether the CAN-interface shall send a message with the IN-identifier over the BUS or not. This is configured by switching the special parameter 'Auto_In_Message' to ON. In case of 'Auto_In_Message' = OFF the CAN interface only actualizes the new value(s) to the message buffer of the CAN-Controller. In order to get the new values over the Bus, the logical Master has to send a Remote-Frame onto the IN-identifier. At the 'Auto_In_Message' = 'OFF' the actual values of both IN-words can be interrogated at switched on IN-word processing sending a remote-frame onto the IN-Identifier.

3.3. The Identifier of Communication Objects

As already stated, the above mentioned communication objects will be filed onto fixed identifiers. At the same time the identifiers are determined directly from the subscriber address of the frequency inverter ('inverter address display', parameter r.18, bus address 0212 (hex), see parameter list of the frequency inverter).

The connection between the content of these parameters and the identifiers for the 4 CAN-messages of the frequency inverter is as follows:

REQUEST_ID	=	(Node address * 4) + 1.
RESPONSE_ID	=	(Node address * 4) + 2.
OUT_ID	=	(Node address * 4) + 3.
IN_ID	=	(Node address * 4) + 4.

Example:

A frequency inverter with the subscriber address t_adr = 54 (dec) can be addressed by the following identifiers:

REQUEST_ID	=	(54 * 4) + 1 = 217 (dec).
RESPONSE_ID	=	(54 * 4) + 2 = 218 (dec).
OUT_ID	=	(54 * 4) + 3 = 219 (dec).
IN_ID	=	(54 * 4) + 4 = 220 (dec).

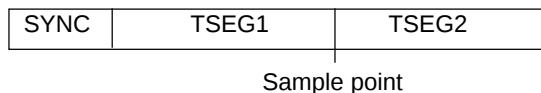
4. Appendix

4.1. Literary Information

- DRIVECOM-Profile-Paper v. 16.12.1991
Publisher: DRIVECOM-User Group e.V.
PO Box 1102, 32825 Blomberg, Art.-No: DRI 21
- Document for the Agreement of the Work Group Physical-Layer of the CiA (CAN in Automation User-Group):
- CiA/DS 102-1
Publisher: CiA International Users and Manufacturers Group e. V.
Simon-Schöffel-Str. 21 , 90427 Nürnberg
- Documents for the Agreement of the Work Group of Higher-Layer Protocols of the CiA (Publishers s. o.):
 - CiA/WG2/DS201 : CAN in the OSI Reference Model.
 - CiA/WG2/DS202-1 : CMS Service Specification.
 - CiA/WG2/DS202-2 : CMS Protocol Specification.
 - CiA/WG2/DS202-3 : CMS Encoding Rules.
 - CiA/WG2/DS203-1 : NMT Service Specification.
 - CiA/WG2/DS203-2 : NMT Protocol Specification.
 - CiA/WG2/DS204-1 : DBT Service Specification.
 - CiA/WG2/DS204-2 : DBT Protocol Specification.
 - CiA/WG2/DS207 : Application Layer Naming Conventions.

4.2 Bit-Timing auf CAN

The nominal Bit-Timing is as follows:



For all adjustable baud rates the following is valid:

- t_q : The unit of time of which each segment is constructed.
- SYSC = 0 ==> Only the edges of recessive to dominant are used for synchronization.
- SJW = 0 ==> Synchronization-jump displacement = $1t_q$
- TSEG2 = 1 ==> $t_{SEG2} = 2 * t_q$
- TSEG1 = 12 ==> $t_{SEG1} = 13 * t_q$

==> Length of a bit = $16 * t_q$, sample point = $14 * t_q$

Baud rate	Time-Quantum (t_q)	Sample-Point
10 KBit/s	6,25 μ s	$14 * t_q = 87,5 \mu$ s
20 KBit/s	3,125 μ s	$14 * t_q = 43,75 \mu$ s
50 KBit/s	1,25 μ s	$14 * t_q = 17,5 \mu$ s
100 KBit/s	625 ns	$14 * t_q = 8,75 \mu$ s
125 KBit/s	500 ns	$14 * t_q = 7,0 \mu$ s
250 KBit/s	250 ns	$14 * t_q = 3,5 \mu$ s
500 KBit/s ¹⁾	125 ns	$14 * t_q = 1,75 \mu$ s

¹⁾ When using optocouplers for isolation, the working committee Physical-Layer of CiA recommends CAN-interfacing to set the upper limit for CAN- baud rate 250 kBit/s. Since optocouplers are used in the KEB-CAN interfacing this suggestion should be followed, although higher values are possible.

4.3. Summary of the Higher Protocol Layers

4.3.1 Identifier of the Communication Objects Dependent on the Node Address

Name	Identifier	Data Length (Byte)	Structure of the Data Field
REQUEST_ID	$(t_adr * 4) + 1$	5	01234 Com. PARA-ADR PARA-Values LB HB LB HB
RESPONSE_ID	$(t_adr * 4) + 2$	5	01234 Res. PARA-ADR PARA-Values LB HB LB HB
OUT_ID	$(t_adr * 4) + 3$	4	0123 OUT-Word1 OUT-Word2 LB HB LB HB
IN_ID	$(t_adr * 4) + 4$	4	0123 OUT-Word1 OUT-Word2 LB HB LB HB

4.3.2 Definition of the Special-Parameter

Parameter-Address	Meaning	Coding	Default-Value																
0080h	Determines the baud rate on the CAN-Bus. A value changed is immediately active but not immediately stored non-volatiley. The non-volatile storage of a CAN-baud rate explicitly occurs with the parameter "Save_CAN_Baud" (s.u.)	<table><tr><td>Value</td><td>Baudrate</td></tr><tr><td>0</td><td>10KBit/s</td></tr><tr><td>1</td><td>20KBit/s</td></tr><tr><td>2</td><td>50KBit/s</td></tr><tr><td>3</td><td>100KBit/s</td></tr><tr><td>4</td><td>125KBit/s</td></tr><tr><td>5</td><td>250KBit/s</td></tr><tr><td>6</td><td>500KBit/s</td></tr></table>	Value	Baudrate	0	10KBit/s	1	20KBit/s	2	50KBit/s	3	100KBit/s	4	125KBit/s	5	250KBit/s	6	500KBit/s	1
Value	Baudrate																		
0	10KBit/s																		
1	20KBit/s																		
2	50KBit/s																		
3	100KBit/s																		
4	125KBit/s																		
5	250KBit/s																		
6	500KBit/s																		
0081h	'OUT1-ADDRESS': Parameter-Address on which the OUT-Word 1 is represented. Note!!! Special parameters cannot be represented on IN-or OUT-Words! Value is non-volatiley stored and immediately active.	(see parameter list of the frequency inverter control card).	0106h																
0082h	'OUT2-ADDRESS': Parameter-Address on which OUT-Word 2 is represented. Note!!! Special parameters cannot be represented on IN-or OUT-words! Value is non-volatiley stored and immediately active.	(see parameter list of the frequency inverter control-card).	0108h																
0083h	'IN1-Address': Parameter-Address on which IN-Word 1 is represented. Note!!! Special parameters cannot be represented on IN-or OUT-words! Value is stored non-volatiley and is immediately active.	(see parameter list of the frequency inverter control card).	0107h																
0084h	'IN2-Adresse': Parameter-Address on which IN-Word 2 is represented. Note!!! Special parameters cannot be represented on IN-or OUT-words! Value is stored non-volatiley and is immediately active.	(see parameter list of the frequency inverter control card)	0109h																

Parameter-Address	Meaning	Coding	Default Value
0085h	<i>'Auto_In_Message'</i> : Determines, whether during a change of one of the IN-words, a message is automatically sent onto the CAN Bus (ON) or not (OFF). At the adjustment of OFF the logical master must demand a remote-frame onto the IN-identifier.	0: OFF FFFFh: ON	OFF
0086h	<i>'Inword_Activation'</i> : Activation of the IN-word processing: Value is stored non-volatiley and is immediately active.	0: OFF FFFFh: ON	OFF
0087h	<i>'Outword_Activation'</i> : Activation of the OUT-word processing. Value is stored non-volatiley and is immediately active.	0: OFF FFFFh: ON	OFF
0088h	<i>'EEPROM_Check'</i> Information about the result of the EEPROM tests. This parameter can only be read. If the value = 0, then none of the configurations can be non-volatiley stored. In this case the CAN interface runs with the default adjustment.	0: EEPROM faulty FFFFh: EEPROM error-free	
0089h	<i>'Outword_Priority'</i> : At adjustment 'ON' and 'Outword-Activation' = 'ON' the actualization of a changed OUT-word to the inverter-control has priority before the cyclical IN-word processing. The value is non-volatiley stored and immediately active.	0: OFF FFFFh: ON	OFF
008Ah	<i>'Request_Priority'</i> : At adjustment 'ON' and the existing request from CAN-Bus the processing of this request has priority before the cyclical IN-word processing. Value is non-volatiley stored and is immediately active.	0: OFF FFFFh: ON	OFF

4.3.3. Description of the Communication Objects in CAN-Message-Specification

CMS-Description of the Object: KEB_Object

- Object-type : VARIABLE
- Name : '#KEB_Object'
- Data-Type : STRUCTURE OF UNSIGNED16,INTEGER16
- Error-Type : UNSIGNED8
- Class : BASIC
- User-Type : SERVER
- Access-Type : READ_WRITE
- Priority : Dependent on the slave-address beginning with priority = 0.
- Inhibit-Time : Not defined at this point.

CMS-Description of the Object: KEB_Out

- Object-type : VARIABLE
- Name : '#KEB_Out'
- Data-Type : ARRAY[2] OF INTEGER16
- Class : BASIC
- User-Type : SERVER
- Access-Type : WRITE-ONLY
- Priority : Dependent on the slave-address beginning with priority = 0.
- Inhibit-Time : Not defined at this point.

CMS-Description of the Object: KEB_In

- Object-type : EVENT
- Name : '#KEB_In'
- Data-Type : ARRAY[2] OF INTEGER16
- Class : STORED
- User-Type : SERVER
- Priority : Dependent on the slave-address beginning with priority = 0.
- Inhibit-Time : Not defined at this point.

4.3.4 Coding of the DRIVECOM-Parameter 'Control-/Status-word' and 'Set/Real Speed'

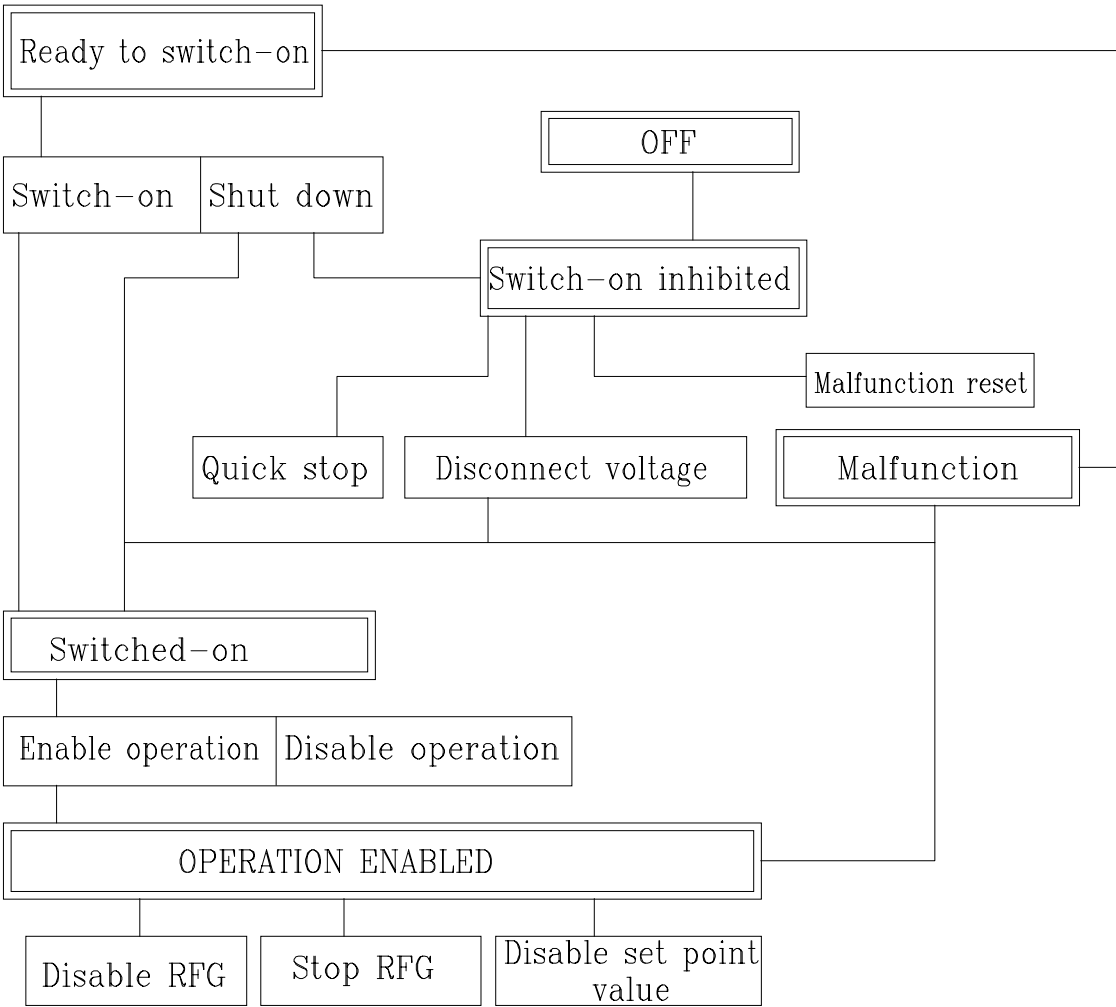
Name	:	Control word
Function	:	Regulates the operating conditions of the inverter. (see also constitution diagram status/control word)
Parameter-Address	:	0106 (hex)
Coding	:	Bit 0 1=Switch on (0=slow down)
		Bit 1 1=do not disconnect the voltage
		Bit 2 1=no quick stop
		Bit 3 1=operation enabled (0=operation blocked)
		Bit 4 1=do not disable HLG
		Bit 5 1=do not disable HLG
		Bit 6 1=Do not disable set-point value
		Bit 7 0->1=Reset malfunction
		Bit 8 no function at this point
		Bit 9 no function at this point
		Bit 10 no function at this point
		Bit 11 no function at this point
		Bit 12 no function at this point
		Bit 13 no function at this point
		Bit 14 no function at this point
		Bit 15 no function at this point

<i>Name</i>	:	Status word
<i>Function</i>	:	Returns the status to the inverter. (also see the constitution diagram status/control word) parameter address: 0107 (hex)
<i>Coding</i>	:	Bit 0 1=Ready to switch-on Bit 1 1=Switched on Bit 2 1=Operation enabled Bit 3 1=Malfuction Bit 4 1=No voltage disconnection Bit 5 1=No quick stop Bit 6 1=Starting disabled Bit 7 1=Warning Bit 8 not defined at this point Bit 9 1=no local operation active Bit 10 1=set value reached Bit 11 1=limitation active Bit 12 not defined at this point Bit 13 not defined at this point Bit 14 not defined at this point Bit 15 Handshake Out-process data

<i>Name</i>	:	Set reference
<i>Parameter-Address</i>	:	0108 (hex)
<i>Coding</i>	:	Is normalized in rpm.

<i>Name</i>	:	(Actual)speed
<i>Parameter-Address</i>	:	0109 (hex)
<i>Coding</i>	:	Is normalized in rpm

Constitution Diagram for Control/Status Word



RFG: Ramp frequency generator



Karl E. Brinkmann GmbH
Postfach 11 09 • Försterweg 36 - 38
D - 32677 Barntrup • Telefon 0 52 63 / 4 01-0
Teletex 5 263 811 keb • Telefax 4 01 - 116