BETRIEBSANLEITUNG
INSTRUCTION MANUAL

CAN in AUTOMATION

KEB COMBICOM

CANopen-ANSCHALTUNG
CANopen-INTERFACE

CC.F5.0D0-K002

12/2003

D **Seite 3 ..... 64**

Das in dieser Betriebsanleitung verwendete Pictogramm entspricht folgender Bedeutung:

**Achtung,
Unbedingt
beachten**

In dieser Betriebsanleitung befindet sich auf Seite 60 ein Literaturverzeichnis, in dem Nachschlagewerke aufgeführt sind, die bestimmte Normen und Aussagen in dieser Anleitung erläutern. An den entsprechenden Textstellen, befinden sich mit eckigen Klammern [ ] gekennzeichnete Ziffern.

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.

In dieser Betriebsanleitung wird die neue Version des F5-CAN-Operator beschrieben. Für die alte Version verweisen wir auf die Betriebsanleitung CC.F5.010-K001.

2. Bestell-bezeichnungen

Diese Bedienungsanleitung:	CC.F5.0D0-K002
F5-CAN-Operator mit Anzeige und Tastatur:	00.F5.060-5010
F5-CAN-Operator ohne Anzeige und Tastatur:	00.F5.060-5110
F5-CAN-Operator mit Anzeige und Tastatur (Klemmleiste):	00.F5.060-5011
F5-CAN-Operator ohne Anzeige und Tastatur (Klemmleiste):	00.F5.060-5111
Zubehör für Diagnoseschnittstelle:	
HSP5-Kabel zwischen PC und Adapter:	00.F5.0C0-0001
Adapter DSUB9 / Western:	00.F5.0C0-0002

3. F5-CAN-Operator

KEB-Antriebstechnik entwickelt, produziert und vertreibt weltweit statische Frequenzumrichter im industriellen Leistungsbereich. Die Umrichter des Typs **F5** können optional mit einer **CAN** (**C**ontroller-**A**rea-**N**etwork)-Schnittstelle ausgerüstet werden. Es handelt sich hierbei um eine intelligente Schnittstelle, die den Zugriff über CAN auf die Parameter des Frequenzumrichters steuert.

Der F5-CAN-Operator wird durch Einstecken in das FU-Gehäuse integriert und paßt in alle KEB-F5-Frequenzumrichter. Parallel zum Feldbusbetrieb ist die Bedienung über die integrierte Anzeige/Tastatur sowie eine weitere Schnittstelle zur Diagnose/Parametrierung (KEB COMBIVIS) möglich.

Zur Programmierung des KEB-F5-Umrichters über CAN, benötigt der Anwender außer dieser Bedienungsanleitung zudem noch die Betriebsanleitung der jeweiligen Frequenzumrichtersteuerung [1].

4. Hardware-beschreibung

Tastatur und Anzeige nur bei
00.F5.060-5010 und 00.F5.060-5011

SDO: (grün) SDO-Kommunikation aktiv

PDOUT: (grün)
- PDOUT-Daten werden zur FU-Steuerung geschrieben.

PDIN: (grün)
- PDIN-Daten werden von der FU-Steuerung gelesen.
- Leuchtet kurz nach einem Reset-Kommando über CAN auf.

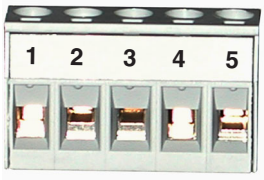
E (rot):
An ==> Umrichter betriebsbereit
blinkt ==> Umrichter in Fehler
Aus ==> keine Versorgungsspannung

X6B: Diagnoseschnittstelle zum PC (s. Kapitel 4.1)

X6C: CAN-Schnittstelle (Buchsen-Stecker)

Wahlweise erhältlich mit Klemmleiste

X6D: CAN-Schnittstelle als 5polige Klemmleiste (optional erhältlich)



Pinbelegung CANo:

Pin	Signal
1	V- (Bezugspotential externe Versorgungsspannung)*
2	CAN_L
3	Schirm
4	CAN_H
5	V+ (externe Versorgungsspannung) *

* Hier nicht angeschlossen

4.1 Diagnoseschnittstelle

Um eine Zerstörung der PC-Schnittstelle zu vermeiden, darf die Diagnoseschnittstelle nur über ein spezielles HSP5-Kabel mit Spannungsanpassung an einen PC angeschlossen werden!

An die Diagnoseschnittstelle wird über einen Adapter ein HSP5-Kabel angeschlossen (siehe Kapitel 2 Bestellbezeichnungen). Über die PC-Software KEB COMBIVIS 5 kann nun auf alle Umrichterparameter normal zugegriffen werden. Die Operator-internen Parameter können ebenfalls ausgelesen und zum Teil eingestellt oder mittels Download parametriert werden.

4.2 CAN-Schnittstelle

Die CAN-Schnittstelle besteht aus einem D-SUB-9pol.-Stift-Stecker (nach DIN41652 Teil 1). Die Belegung der CAN-Stecker gemäß [2]:

Pin	Signal	Beschreibung
1	-	Reserviert
2	**CAN_L**	**CAN-Bussignal dominant low**
3	CAN_GND	Hier nicht angeschlossen
4	-	Reserviert
5	(CAN_SHLD)	Hier nicht angeschlossen
6	(GND)	Hier nicht angeschlossen
7	**CAN_H**	**CAN-Bussignal dominant high**
8	-	Reserviert
9	(CAN_V+)	Hier nicht angeschlossen

Übertragungspegel auf CAN: Nach ISO/DIS 11898, ISO-High Speed

Übertragungsgeschwindigkeit auf CAN: Einstellbar über CAN (10, 20, 25, 50, 100, 125, 250, 500, 800, 1000 Kbit/s)

Potentialtrennung: Sichere Trennung nach VDE0160.

Busabschluß: 124 Ohm , muß extern erfolgen (zwischen Pin 2 und 7).

5. Grundlegendes zum CAN-BUS

Es soll an dieser Stelle das System des **CAN** (**C**ontroller-**A**rea-**N**etwork)-BUS vorgestellt und dabei einige Begriffe erläutert werden, die im folgenden oft Verwendung finden.

Der CAN ist ein **Multi-Master-System**, d.h. jeder Teilnehmer kann auf den BUS zugreifen und Telegramme absenden. Damit bei gleichzeitigem Zugriff zweier Teilnehmer keine ungültigen Zustände entstehen, kennt der CAN-BUS eine sogenannte Arbitrierungs- (Schlichtungs-) Phase, die den Telegrammanfang bestimmt. Bei Zugriffskonflikten erkennen alle Teilnehmer während dieser Arbitrierung, wer die niedrigste Telegrammnummer (Identifier) sendet. Dieser Teilnehmer kann dann sein Telegramm vollständig, ohne von vorne beginnen zu müssen, weitersenden. Alle anderen (sendewilligen) Teilnehmer gehen dann in den Empfangsstatus über und brechen ihr Telegramm zunächst ab. Somit ist festgelegt, dass niedrigere Telegrammnummern automatisch Vorrang haben vor höheren. Die Anzahl der Telegrammnummern ist beim CAN Version 2.0A begrenzt auf 2032 Identifier (0...2031).

Die CAN-Telegramme können maximal 8 Byte Anwender-Daten enthalten.

Wenn im Folgenden der Begriff **logischer CAN-Master** gebraucht wird, so ist damit der CAN-Teilnehmer gemeint, dem die Steuerung des Gesamt-CAN-Systems obliegt. Auch wenn es physikalisch beim CAN nur Master gibt, so wird es in den meisten Einsatzfällen doch einen oder mehrere Teilnehmer geben, die die Kontrolle haben. Der KEB-Frequenzumrichter ist in diesem Zusammenhang als Befehlsempfänger (logischer Slave) zu sehen.

6. Funktionen

Das CAN-Protokoll ist bis zur Datensicherungsschicht einheitlich standardisiert. Die Abarbeitung dieses Protokolls übernimmt vollständig ein CAN-Controller. Weiterhin hat der **C**AN **i**n **A**utomation Verein **(CiA)** einen Standard für die höhere Protokollschicht verabschiedet, der mit **C**AN **A**pplication **L**ayer **(CAL)** bezeichnet wurde. Auf diesem Standard aufbauend wurde dann im September 1995 das **„CAL-based Communication Profile“** (CiA,DS301) veröffentlicht. Dieser Standard bildet die Basis für alle **CANopen**-Geräte-Profile. In diesem Standard wird eine bestimmte Untermenge des CAL-Standards ausgewählt. Das Kommunikationsprofil definiert u.a. ein **Minimum Capability Device**. Das ist die minimal erforderliche Funktionalität, die ein CANopen-Knoten zur Verfügung stellen muß. Die vorliegende CAN-Anschaltung realisiert ein solches Minimum Capability Device.

Ein wichtiger Punkt in jedem CAN-Netzwerk ist die Vergabe der Telegrammnummern (Identifier), zumal deren Anzahl bei CAN V2.0 A auf 2032 begrenzt ist. Im CAL-Standard ist hierzu ein eigenes Verfahren definiert, das diese Vergabe dynamisch über ein eigenes Protokoll (DBT = Distributor) abwickelt. Dieses relativ aufwendige Verfahren zur Vergabe der Identifier ist für ein Minimum Capabiltity Device nicht zwingend vorgeschrieben und in der KEB-CAN-Anschaltung nicht integriert. Für diesen Fall wird im Kommunikationsprofil ein einfacheres Verfahren zur Absprache der Identifiervergabe festgelegt. Dieses Verfahren wird auch von der KEB-CAN-Anschaltung unterstützt und sieht wie folgt aus:

Jeder Frequenzumrichter erhält eine eindeutige CAN-Adresse, die **Node_Id.** Es gibt bei KEB zwei Möglichkeiten für die Quelle dieser Node_Id.

- Wenn der Wert des Parameters OP_Node_Id den Wert 255(dez) besitzt:
  ==> Node_Id = Umrichteradresse (SY.06) +1
- In allen anderen Fällen gibt der Wert des Parameters OP_Node_Id selbst den Wert für Node_Id vor:
  ==> Node_Id = OP_Node_Id

Nach Auslieferung haben alle KEB-Frequenzumrichter die Umrichteradresse = 1. Sollen mehrere KEB-Frequenzumrichter über CAN vernetzt werden, müssen diesen zunächst unterschiedliche Umrichteradressen vorgegeben werden. Dies geschieht z. B. über die Tastatur am Operator.

Jedem Frequenzumrichter werden sechs Identifier zugeordnet.

Über einen Identifier kann ein beliebiger CAN-Knoten das Lesen oder das Schreiben eines Parameterwertes anfordern **(Request-Identifier)**.

Ein weiterer Identifier ist reserviert für die zugehörige Antwort vom Frequenzumrichter **(Response-Identifier)**. Der Mechanismus von Anforderung und Antwort wird auch als **bestätigter Dienst** bezeichnet. Das CANopen-Kommunikationsprofil faßt diese Funktion unter dem Begriff **Service-Data-Object (SDO)** zusammen:

SDO(rx) = Request-Identifier	= 1536+Node_Id
SDO(tx) = Response-Identifier	= 1408+Node_Id

Beispiel: Node_Id= 30 ==> Schreib/Leseanforderungen über Identifier = 1566(dez)
==> Schreib/Lesebestätigungen über Identifier = 1438(dez)

Hinweis: Grundsätzlich reicht die Funktion des SDO vollständig aus, um den KEB F5-Frequenzumrichter über CAN zu steuern. Jeder Parameterwert im Umrichter kann hierüber verändert oder erfragt werden.

Über den 3. Identifier kann der CAN-Master dem Frequenzumrichter Daten unadressiert und unbestätigt vorgeben. In Anlehnung an die Datenrichtung vom Master zum Slave wird dieser im Folgenden als **OUT1-Identifier** bezeichnet.

Über den 4. Identifier leitet der Frequenzumrichter seinerseits neue Daten unadressiert und unbestätigt an den CAN-Master **(IN1-Identifier).**

Diese Funktionalität wird vom Kommunikationsprofil als **Process-Data-Object (PDO)** bezeichnet. Die beiden Objektteile werden mit PDO1(rx) und PDO1(tx) benannt.

PDO1(rx) = Out-Identifier = 512 + Node_Id
PDO1(tx) = IN-Identifier = 384 + Node_Id

Ab der Software-Version 1.3 ist die PDO-Funktionalität in der KEB F5-CAN-Anschaltung zweimal vorhanden. Dieses sog. 2. PDO belegt dann die Identifier fünf bis sechs:

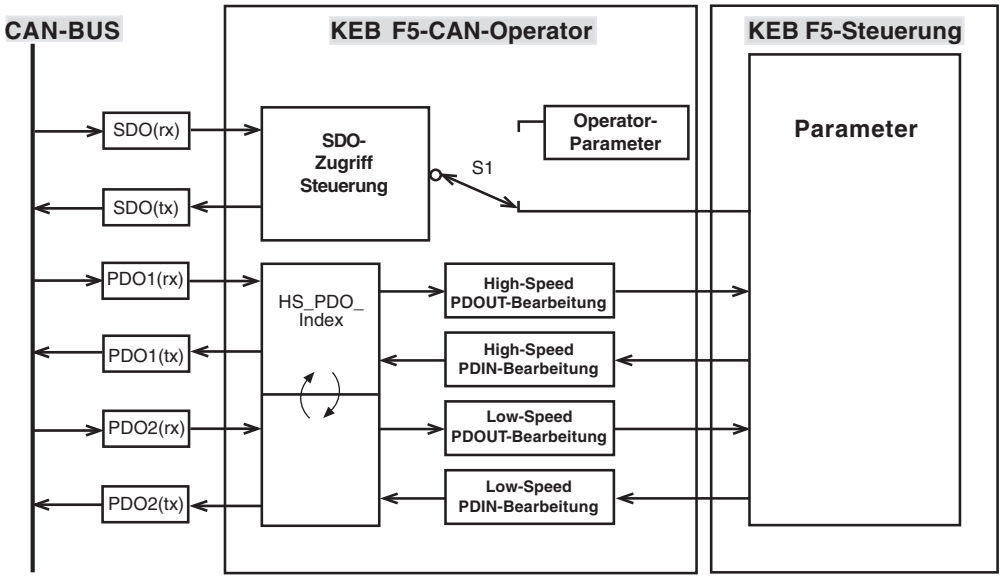
PDO2(rx) = Out-Identifier = 768 + Node_Id
PDO2(tx) = IN-Identifier = 640 + Node_Id

Über den 5. Identifier kann der CAN-Master dem Frequenzumrichter Daten unadressiert und unbestätigt vorgeben. In Anlehnung an die Datenrichtung vom Master zum Slave wird dieser im Folgenden als **OUT2-Identifier** bezeichnet.

Über den 6. Identifier leitet der Frequenzumrichter seinerseits neue Daten unadressiert und unbestätigt an den CAN-Master **(IN2-Identifier).**

Die beiden PDO's sind zwar bzgl. der Verwaltung nach Aussen identisch, unterscheiden sich aber deutlich in der Form der internen Bearbeitung. Nur eines der beiden kann wie in früheren Software-Versionen als High-Speed-PDO bearbeitet werden. Das hinzugekommene PDO ist von der Abarbeitung her gleichgestellt mit den SDO-Kommandos und wird im Folgenden als Low-Speed-PDO bezeichnet. Welches der beiden PDO's das High-Speed-PDO ist, kann eingestellt werden. Nach Auslieferung ist wie bisher das erste PDO 'High-Speed' und das zweite PDO abgeschaltet. Bestehende CAN-Applikationen müssen also nicht geändert werden.

Die CAN-Anschaltung steuert den Datenfluß vom CAN-BUS (SDO(rx), PDO1(rx) und PDO2(rx) hin zur Frequenzumrichter-Steuerung und ebenso vom Frequenzumrichter zum CAN-BUS (SDO (tx), PDO1(tx) und PDO2(tx):



Das obige Bild zeigt die Funktion der CAN-Anschaltung. Die Stellung von Schalter **S1** wird allein von der im CAN-SDO(rx)-Telegramm enthaltenen Parameter-Adresse (16 Bit Index plus 8 Bit Subindex) bestimmt. In einem bestimmten Index-Bereich liegen die sog. Konfigurationsdaten der CAN-Anschaltung. Diese Parameter bestimmen das Verhalten der CAN-Anschaltung und sind deshalb auch in dieser realisiert. Zugriffe auf Parameter im Index-Bereich 2000(hex) bis 5EFF(hex) werden als Schreib-/Leseaufträge zur Umrichter-Steuerung weitergeleitet.

6.1 Charakteristik des High-Speed-PDO

- Die Prozessdatenabbildung liegt in der Umrichter-Steuerung. Die entsprechenden Parameter liegen in der System-Parametergruppe (SY). Da die Kodierung des PD-Mapping des Umrichters nicht mit der von CANopen übereinstimmt, wird diese vom CAN-Operator entsprechend automatisch umgewandelt.
- Das Vorgeben neuer Prozessausgangsdaten vom CAN wird auf nur einen speziellen Prozessdatendienst zur Umrichter-Steuerung umgesetzt.
- Die minimale Zykluszeit für neue Prozessausgangsdaten liegt bei ca. 3ms.
- Das zyklische Lesen der Prozesseingangsdaten wird durch nur einen speziellen Prozessdaten-Lesedienst durchgeführt.
- Die minimal erreichbare Zykluszeit für das Lesen der Prozesseingangsdaten liegt bei ca. 3ms.
- Nicht alle Parameter der Umrichter-Steuerung können auf das High-Speed-PDO abgebildet werden.

6.2 Charakteristik des Low-Speed-PDO

- Prozessdatenabbildung wird allein vom CAN-Operator verwaltet.
- Das Vorgeben neuer Prozessausgangsdaten vom CAN wird auf ‘n’ einfache Dienste (wie SDO-Kommandos) zur Umrichter-Steuerung umgesetzt, wobei ‘n’ der Anzahl abgebildeter Parameter im PDO-Mapping entspricht.
- Die minimale Zykluszeit für neue Prozessausgangsdaten liegt bei ca. ‘n’ * 5 ms.
- Das zyklische Lesen der Prozesseingangsdaten wird durch ‘n’ einfache Lesedienste durchgeführt, wobei ‘n’ der Anzahl abgebildeter Parameter im PDO-Mapping entspricht.
- Die minimal erreichbare Zykluszeit für das Lesen der Prozesseingangsdaten liegt bei ca. ‘n’ * 5 ms.
- Alle Parameter der Umrichter-Steuerung können auf das Low-Speed-PDO abgebildet werden.

6.3 Prozessdatenabbildung

Die Bestimmung des Ziels für die Daten in den PDO(rx)-Telegrammen bzw. der Quelle für die Daten in den PDO(tx)-Telegrammen hält sich vollständig an die Vorschriften des CANopen-Kommunikationsprofils [12]. Hierbei definiert für jede Datenrichtung jeweils ein komplex aufgebautes Objekt (Parameter) die PDO-Abbildung (PDO-Mapping).

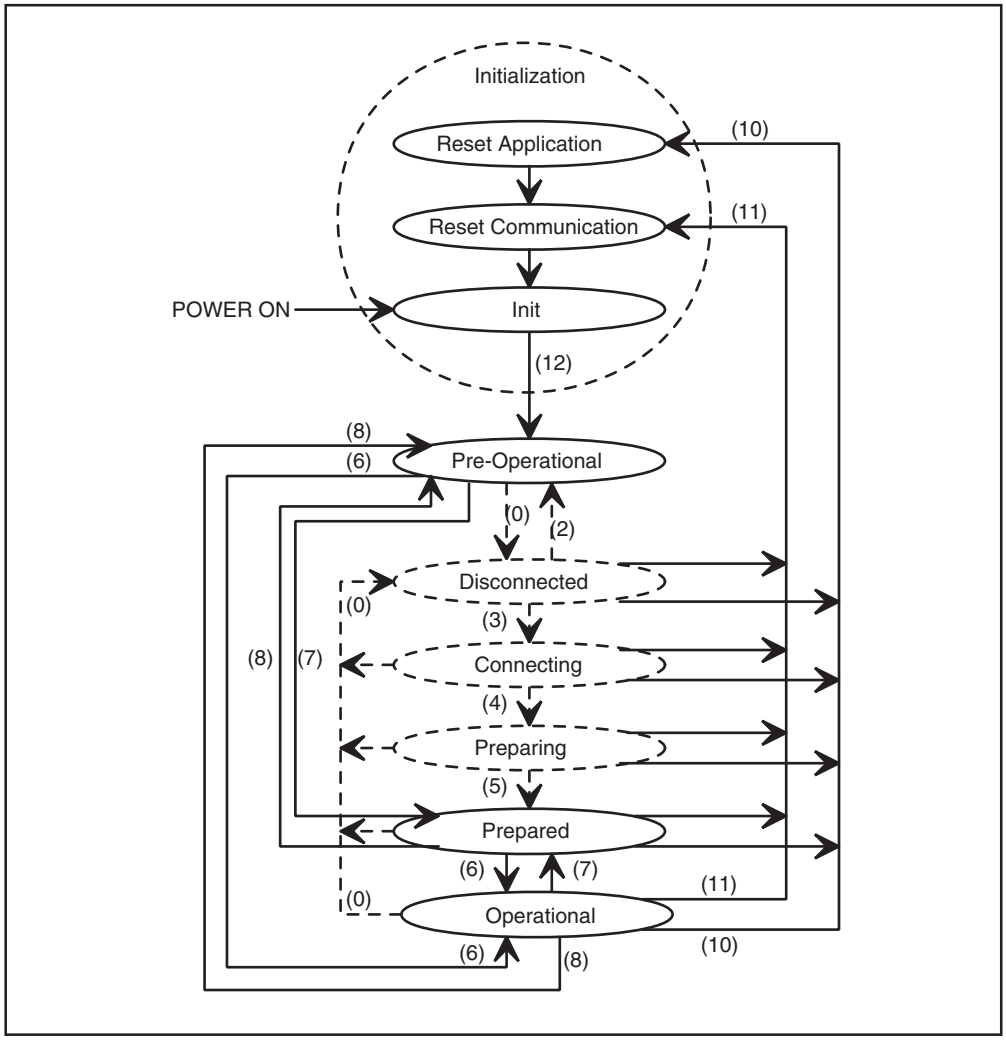
Ein weiteres Objekt je Datenrichtung bestimmt die Kommunikationsdefinitionen (PDO Communication Parameter). Siehe Parameterbeschreibungen von

- **1st receive PDO Mapping**
- **1st transmit PDO Mapping**
- **1st receive PDO Parameter**
- **1st transmit PDO Parameter**
- **2nd receive PDO Mapping**
- **2nd transmit PDO Mapping**
- **2nd receive PDO Parameter**
- **2nd transmit PDO Parameter**

in dieser Bedienungsanleitung.

6.4 CANopen Bootup-Sequenz

Die KEB-CAN-Anschaltung geht automatisch nach der Initialisierungsphase in den Status **Pre-Operational.** In diesem Status ist bereits Kommunikation über das SDO(rx) und SDO(tx) mit den Diensten Domain Download (Parameter Schreiben) und Domain Upload (Parameter Lesen) aktiviert. Lediglich die Prozessdatenkommunikation ist in diesem Status noch inaktiv. Diese wird dann durch das NMT-Kommando Start_Remote_Node() freigegeben (Bild). Das Ziel dieser Startsequenz ist der Betriebszustand **Operational.** In diesem Status ist die Kommunikation vollständig aktiviert. Adressiert werden beim NMT-Protokoll bestimmte CAN-Knoten durch die oben bereits erwähnte **Node-Id.**



Die KEB-CANopen-Anschaltung realisiert folgende in obigem Schaubild mit durchgezogener Linie gezeichnete Übergänge:

6: Start_Remote_Node()

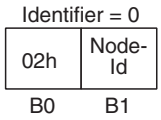
CAN-Telegramm:

Identifier = 0	
01h	Node-Id
B0	B1

Node_Id = 0 (alle NMT-Slaves sind angesprochen) oder
Node_Id = Umrichter-Adresse + 1 (nur 1 Frequenzumrichter ist angesprochen)

7: Stop_Remote_Node()

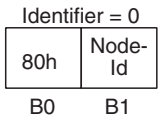
CAN-Telegramm:



Node_Id = 0 (alle NMT-Slaves sind angesprochen) oder
Node_Id = Umrichter-Adresse + 1 (nur 1 Frequenzumrichter ist angesprochen)

8: Enter_Pre-Operational_State()

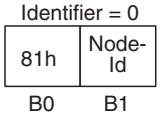
CAN-Telegramm:



Node_Id = 0 (alle NMT-Slaves sind angesprochen) oder
Node_Id = Umrichter-Adresse + 1 (nur 1 Frequenzumrichter ist angesprochen)

10: Reset_Node(): Beim Ausführen dieser Funktion wird ein Software-Reset in der KEB-CAN-Anschaltung durchgeführt.

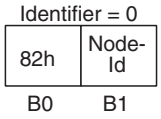
CAN-Telegramm:



Node_Id = 0 (alle NMT-Slaves sind angesprochen) oder
Node_Id = Umrichter-Adresse + 1 (nur 1 Frequenzumrichter ist angesprochen)

11: Reset_Communication(): Funktion wie bei Reset_Node().

CAN-Telegramm:



Node_Id = 0 (alle NMT-Slaves sind angesprochen) oder
Node_Id = Umrichter-Adresse + 1 (nur 1 Frequenzumrichter ist angesprochen)

12: Enter Pre-Operational automatically(): s.o.

6.5 Bootup-Message

Der KEB-F5-CAN-Operator gibt eine Bootup-Message aus, wenn nach POWER ON die Initialisierungsphase abgeschlossen ist. Dies ist ein Telegramm auf Identifier = 1792 + Node_Id mit der Datenlänge = 1 und dem Wert = 0.

6.6 Node-Guarding

In [12] ist ein Protokoll vorgesehen, mit dem ein CAN-Knoten den aktuellen Status eines beliebigen Knoten erfragen kann. Dies gehört zur Netzwerkmanagement-Funktionalität (NMT) des CAN-Knoten und wird als Node-Guarding bezeichnet. Die KEB-CANopen-Anschaltung unterstützt das Node-Guarding. Die Node-Guarding-Anforderung wird durch einen Remote-Frame auf dem Node-Guarding-Identifier abgesetzt. Die Antwort kommt dann als Datentelegramm mit 1 Byte Daten auf dem gleichen Identifier. Das Datenbyte enthält den Knoten-Status plus ein Toggelbit (MSBit), welches von Meldung zu Meldung invertiert wird. Jeder Knoten hat seinen speziellen Node-Guarding-Identifier.

Beim Minimum Capability Device ergibt sich dieser Identifier direkt aus der Node-Id:

Node-Guarding-Identifier = 1792 + Node-Id

Wert des Knoten-Status	Bedeutung
1	DISCONNECTED
2	CONNECTING
3	PREPARING
4	PREPARED
5	OPERATIONAL
127d	PRE_OPERATIONAL

6.7 Life-Guarding

Der F5-CANopen-Operator unterstützt das Life-Guarding. Es handelt sich dabei um eine Überwachung des zyklischen Node-Guarding des CAN-Masters. Aus diesem Grund sollte das Life-Guarding nur beim zyklischen Node-Guarding aktiviert werden. Das Life-Guarding arbeitet vollständig losgelöst von allen anderen Überwachungsfunktionen. Es wird aktiviert über das Produkt der beiden Parameterwerte Guard Time und Life Time Factor. Ist das Produkt = 0, so ist das Life-Guarding nicht aktiv. Sonst gibt das Produkt die Life-Guarding-Timeoutzeit an. Bei aktiviertem Life-Guarding beginnt die Node-Guarding-Überwachung, sobald der erste Node-Guard-Request empfangen wurde. Die Funktion, die bei Eintreten des Life-Guarding-Timeoutfalls ausgeführt wird, ist über zwei weitere Parameter (LifeGuardTout_Addr, LifeGuardTout_Data) einstellbar. Es handelt sich dabei zum Einen um einen Schreibzugriff auf einen beliebigen Parameter in der Umrichter-Steuerung und zudem um einen Funktionscode, der bestimmt, welche Aktion im Operator ausgeführt werden soll. Nach Auslieferung ist der CAN-Operator so eingestellt, dass bei auftretendem Life-Guarding-Timeout der Parameter SY.50(Steuerwort) in Satz0 mit dem Wert 1 geschrieben wird. Zudem schaltet der CAN-Operator in den Zustand Pre_Operational.

6.8 Emergency Objekt

Das CANopen-Kommunikationsprofil DS301 definiert einen Mechanismus, nachdem sich Knoten selbständig melden, wenn wichtige Ereignisse eingetreten sind. Diese Emergency-Message unterstützt auch der KEB-F5-CANopen-Operator. Die Funktion ist in der Standardeinstellung deaktiviert. Aktiviert wird die Emergency-Message durch Verändern des Parameters EmergencyCycle auf einen Wert ungleich 0. Dann liest der CAN-Operator in dieser Zykluszeit den Wert des Parameters Umrichter Status (RU.00) von der Umrichter-Steuerung und konvertiert diesen in den ErrorCode-Wert nach [13]. Hat sich der Wert geändert, wird eine Emergency-Message auf Identifier 128d + Node_Id abgeschickt. Das bedeutet, dass auch der Übergang vom Fehlerzustand zu normalen Betriebszuständen durch eine Emergency-Message bekannt gemacht wird. Der Inhalt des Telegramms ist vom Profil nur zum Teil fest vorgegeben. Insgesamt sieht der Inhalt der Emergency-Message beim KEB-F5-CAN wie folgt aus:

Identifier = 128 + Node_Id

B0	B1	B2	B3	B4	B5	B6	B7
Error Code LB	Code HB	Error-Register	Umrichter Status LB	Status HB	00h	00h	00h

Alle Fehler werden in dem vom Profil definierten ‚Pre-defined Error Field' gespeichert. Dieses Feld enthält beim KEB-F5-CANopen-Operator maximal fünf Einträge. Wobei der erste Eintrag immer den zeitlich zuletzt aufgetretenen Fehler enthält. Die Kodierung der Einträge sind der Beschreibung des gleichnamigen Parameters zu entnehmen.

7. Kodierung der Daten in den vier CAN-Telegramm-Typen

Über dieses Telegramm kann der logische CAN-Master den Wert eines Parameters Erfragen (Lesen) oder Verändern (Schreiben). Im Kommunikationsprofil wird ein Schreib-Dienst als **Domain Download** und ein Lese-Dienst als **Domain Upload** bezeichnet. Die KEB-CAN-Anschaltung unterstützt lediglich die Kurzform dieser beiden Dienste, so dass nur ein Telegramm für die Dienstaufforderung und ein weiteres für die Dienstbestätigung zwischen logischem CAN-Master und der KEB-CAN-Anschaltung ausgetauscht werden.

7.1 SDO(rx)-Telegramm

Die Adressierung des Parameters geschieht über den vorzeichenlosen 16-Bit-Index plus den vorzeichenlosen 8-Bit-Subindex. Die Parameter der Frequenzumrichtersteuerung liegen im Indexbereich 2000(hex) bis 5EFF(hex). Dabei ergibt sich der CAN-Index aus der Parameter-Adresse (siehe Parameterbeschreibung der eingesetzten FU-Steuerung) durch Addition mit dem Offset 2000(hex):

CAN-Index = KEB-Parameter-Adresse + 2000(hex)

Der Subindex dient als Zusatzadressierung für komplexe Parameter des Operators. Ebenso kann er bei Parametern der Frequenzumichter-Steuerung zur Satzadressierung verwendet werden. Dabei gilt:

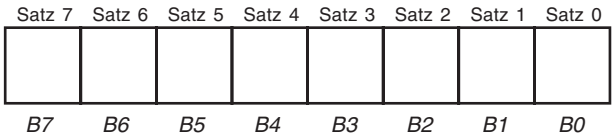
Subindex = 0

Bei satzprogrammierbaren Parametern bestimmt der Wert des Parameters FR.09 den gewählten Satz.

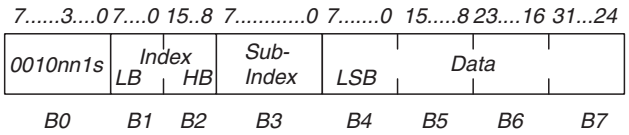
Subindex ungleich 0

Bei satzprogrammierbaren Parametern bestimmt der Subindex den gewählten Satz. Dabei ist zu beachten, dass der Satz bitkodiert ist. Dadurch ist es möglich, beim Schreiben den Wert des Parameters gleichzeitig in mehreren Sätzen zu verändern. Werden beim Lesen mehrere Sätze gleichzeitig adressiert, wird nur dann der Wert des Parameters zurückgegeben, wenn dieser in allen adressierten Sätzen gleich ist. Sind nicht alle Werte gleich, wird ein Fehler zurückgegeben.

Subindex (wenn ungleich 0):



7.1.1 Initiate Domain Download Request (Schreibanforderung des Master)



nn: Nur gültig, wenn s=1: Enthält die Anzahl Bytes des Data-Feldes, die keine Daten enthalten.

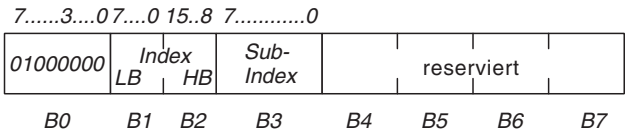
s: Wenn gleich 1, dann enthält nn die Anzahl Bytes im Data-Feld, die keine Daten enthalten.
Sonst keine Anzeige der Datenlänge in nn.

Index: 16-Bit (vorzeichenlos) Adressierung des Parameters (s.o.).

Subindex: 8-Bit (vorzeichenlos) Unteradressierung für komplexe Parameter und die direkte Satzadressierung.

Data: Zu übertragende Daten. Das LSByte wird zuerst übertragen.

7.1.2 Initiate Domain Upload Request (Leseanforderung des Master)



Index: 16-Bit (vorzeichenlos) Adressierung des Parameters (s.o.).

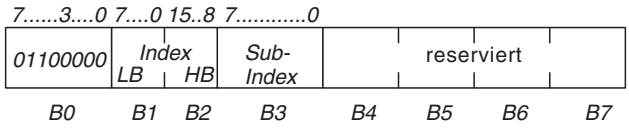
Subindex: 8-Bit (vorzeichenlos) Unteradressierung für komplexe Parameter und die direkte Satzadressierung.

7.2 SDO(tx)-Telegramm

7.2.1 Initiate Domain Download Response (Schreibbestätigung vom FU)

Diese Antwort wird von der KEB-CAN-Anschaltung gesendet, wenn der angeforderte Schreibdienst fehlerfrei ausgeführt werden konnte.

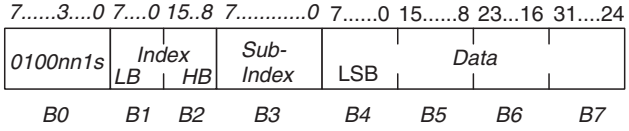
Index: s.o.
Subindex: s.o.



7.2.2 Initiate Domain Upload Response (Lesebestätigung vom FU)

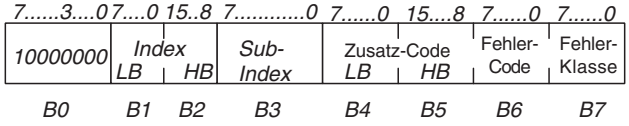
Diese Antwort wird von der KEB-CAN-Anschaltung gesendet, wenn der angeforderte Lesedienst fehlerfrei ausgeführt werden konnte.

nn: s.o.
s: s.o.
Index: s.o.
Subindex: s.o.
Data: s.o.



7.2.3 Abort Domain Transfer (Fehlerantwort vom FU)

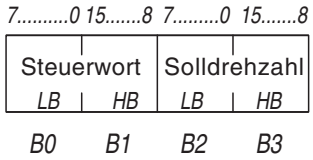
Diese Antwort sendet die KEB-CAN-Anschaltung, wenn der angeforderte Schreib- oder Lesedienst nicht ausgeführt werden konnte. In diesem Fall wird eine Fehlerbeschreibung zurückgeliefert.



Fehler-Klasse	Fehler-Code	Zusatz-Code	Bedeutung
6	1	0000h	Ungültiger Zugriff auf einen Parameter, z. B. Schreiben auf einen Read_Only-Parameter.
6	1	0010h	Ungültiges Passwort.
6	1	0011h	Operation nicht möglich.
6	4	0000h	Der adressierte Parameter existiert nicht.
6	4	0041h	Ungültige PD-Belegung.
6	6	0000h	Die interne Kommunikation zwischen Operator und FU-Steuerung ist gestört.
6	7	0010h	Datenlänge ungültig.
6	9	0011h	Ungültiger Subindex.
6	9	0012h	Sprachkennung ungültig.
6	9	0030h	Der geschriebene Wert liegt außerhalb des gültigen Wertebereichs.
8	0	0022h	Umrichter beschäftigt.

7.3 PDO1(rx)-Telegramm

Mit diesem Telegramm übergibt der logische CAN-Master dem Frequenzumrichter neue Prozessausgangsdaten. In der Standardeinstellung erwartet die KEB-CAN-Anschaltung ein Telegramm mit ≥ 4 Byte Daten mit folgendem Inhalt:



Die Länge und Belegung des PDO1(rx)-Telegramms kann durch verschiedene Operator-Parameter verändert werden. Diese Veränderung kann nur über das SDO(tx)-Telegramm erfolgen (s.o.).

Folgende Operator-Parameter haben Einfluss auf den Aufbau der Prozessausgangsdaten:

- 1st receive PDO Mapping
- 1st receive PDO Parameter

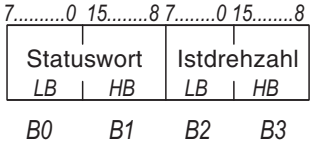
7.4 PDO1(tx)-Telegramm

Über dieses Telegramm gibt die KEB-CAN-Anschaltung dem (logischen) CAN-Master Prozesseingangsdaten bekannt.

Die Länge, Belegung und Steuerung dieses Telegramms wird durch folgende Operator-Parameter beeinflußt:

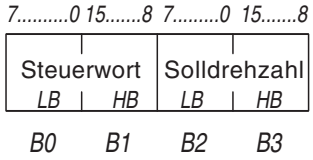
- 1st transmit PDO Mapping
- 1st transmit PDO Parameter

Die Standardeinstellung bewirkt folgenden Telegrammaufbau:



7.5 PDO2(rx)-Telegramm

Mit diesem Telegramm übergibt der logische CAN-Master dem Frequenzumrichter neue Prozessausgangsdaten. In der Standardeinstellung erwartet die KEB-CAN-Anschaltung ein Telegramm mit ≥ 4 Byte Daten mit folgendem Inhalt:



Die Länge und Belegung des PDO2(rx)-Telegramms kann durch verschiedene Operator-Parameter verändert werden. Diese Veränderung kann nur über das SDO(tx)-Telegramm erfolgen (s.o.).

Folgende Operator-Parameter haben Einfluss auf den Aufbau der Prozessausgangsdaten:

- 2nd receive PDO Mapping
- 2nd receive PDO Parameter

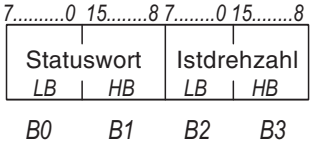
7.6 PDO2(tx)-Telegramm

Über dieses Telegramm gibt die KEB-CAN-Anschaltung dem (logischen) CAN-Master Prozesseingangsdaten bekannt.

Die Länge, Belegung und Steuerung dieses Telegramms wird durch folgende Operator-Parameter beeinflußt:

- 2nd transmit PDO Mapping
- 2nd transmit PDO Parameter

Die Standardeinstellung bewirkt folgenden Telegrammaufbau:



8. Operator-Parameter

Legende	**Parametername**	Objekt-Typ	**CAN-SDO-Index**

8.1 Von KEB definierte Parameter

Diese Parameter bestimmen die Konfiguration der KEB F5-CAN-Anschaltung und sind deshalb auch in dieser und nicht in der Frequenzumrichtersteuerung realisiert:

PD_Stored	Einfach Variable (Var)	**5FE2h**

Subindex:	0
Datenlänge:	1 Byte
Zugriff:	Read_Write
Bedeutung:	Bestimmt, ob die aktuelle Prozessdatenbelegung aus dem EEPROM gelesen wird oder mit der Standard-PD-Belegung gearbeitet wird.
Kodierung:	FFh ==> arbeitet mit der abgespeicherten PD-Belegung sonst ==> arbeitet mit der Standard-PD-Belegung.
Standard Einstellung:	FFh
Bemerkung:	Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.

OP_NodeId	Einfach Variable (Var)	**5FE3h**

Subindex:	0
Datenlänge:	1 Byte
Zugriff:	Read_Write
Bedeutung:	Ermöglicht die Vorgabe der CANopen-Knoten-Adresse im CAN-Operator, unabhängig von der Umrichter-Adresse.
Kodierung:	255: Die Knoten-Adresse wird wie bisher aus der Umrichter-Adress (ud.06) bestimmt: Node_Id = Umrichter-Adresse + 1 1...127: Die Knoten-Adresse wird im Operator gehalten und gespeichert: Node_Id = OP_NodeId
Standard Einstellung:	255
Bemerkung:	Eine Wertänderung wird sofort aktiv und nichtflüchtig gespeichert.

Watchdog Activation — Einfach Variable (Var) — 5FDAh

SDO-Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Ermöglicht die Verzögerung der Aktivierung des Feldbus-Watchdog nach POWER On bzw. einem Reset-Kommando über CAN.
Kodierung: 0: Der Feldbus-Watchdog ist sofort aktiv.
Werte ungleich Null sind bitkodiert mit der folgenden Bedeutung:
Bit0: Aktivierung des Feldbus-Watchdog nach erstem SYNC-Telegramm
Bit1: Aktivierung des Feldbus-Watchdog nach erstem Node-Guarding
Bit2: Aktivierung des Feldbus-Watchdog nach erstem Übergang in den Knoten-Status OPERATIONAL
Bit3: Aktivierung des Feldbus-Watchdog nach erstem PDOUT1-Telegramm
Bit4: Aktivierung des Feldbus-Watchdog nach erstem PDOUT2-Telegramm
Bit5: Aktivierung des Feldbus-Watchdog nach erstem SDO-Telegramm
Standard Einstellung: 0
Erlaubte PDO-Abbildung: keine Abbildung
Bemerkung: Eine Wertänderung wird sofort aktiv und nichtflüchtig gespeichert. Mehrere Ereignisse können als Feldbus-Watchdog-Aktivierung definiert werden. In diesem Fall wird der Watchdog aktiv, sobald eines der definierten Ereignisse eintritt.

B7	B6	B5	B4	B3	B2	B1	B0
		1. SDO	1. PDOUT2	1. PDOUT1	1. OPERA-TIONAL	1. Node-Guarding	1. SYNC

Watchdog Inhibit	Einfach Variable (Var)	**5FF9h**

Subindex: 0

Datenlänge: 1 Byte

Zugriff: Read_Write

Bedeutung: Bestimmt, auf welche Ereignisse der Feldbus-Watchdog getriggert wird. Der Feldbus-Watchdog dient dazu, den Frequenzumrichter in den Fehlerzustand zu bringen, wenn auf CAN keine Aktivitäten mehr stattfinden. Die eigentliche Aktivierung und Programmierung des Watchdog wird in der FU-Steuerung eingestellt. Die dazu einzustellenden Parameter sind der Anleitung der FU-Steuerung zu entnehmen.

Kodierung: Bitkodiert:

Bit 0 = 1

Beim Starten eines PDOUT-Telegramms zur FU-Steuerung wird der Watchdog zurückgesetzt. Beachten Sie, dass das Eintreten dieses Ereignisses auch von der Einstellung des Parameters 1st Receive PDO Parameter.Tx_type sowie von dem Wert des Parameters PDOUT_Wr_Mode abhängt.

Bit 1 = 1

Beim Beginn der Bearbeitung eines SDO-Auftrages wird der Watchdog zurückgesetzt.

Bit 2 = 1

Wenn der Knoten keine Übertragungsprobleme auf CAN feststellt, wird der Watchdog zurückgesetzt.

Bit 3 = 1

Der Watchdog wird bei jedem Empfang eines SYNC-Telegramms zurückgesetzt.

Bit 4 = 1

Der Watchdog wird bei jedem Empfang eines Node-Guard-Request-Telegramms zurückgesetzt.

Bit 5 = 1

Der Watchdog wird bei jedem Empfang eines SYNC-Telegramms zurückgesetzt, vorausgesetzt, mindestens einmal wurden Prozessausgangsdaten zur Frequenzumrichter-Steuerung gesendet.

Standard Einstellung: 07h

Der Watchdog wird zurückgesetzt, wenn:

- Prozessausgangsdaten zur FU-Steuerung geschrieben werden,
- ein SDO-Auftrag gestartet wird und
- keine Übertragungsprobleme auf CAN festgestellt werden.

Bemerkung: Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.

PDOUT_WrMode — Einfach Variable (Var) — 5FE4h

Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Bestimmt unter welchen Bedingungen PDOUT-Daten zur FU-Steuerung geschrieben werden. Hiermit kann die Kommunikation zwischen CAN-Operator und FU-Steuerung entlastet werden.
Kodierung:
0: Es werden immer alle PDOUT-Daten zur FU-Steuerung geschrieben, egal ob diese geändert sind oder nicht.
255: Es werden immer alle PDOUT-Daten zum Umrichter geschrieben, wenn mindestens einer der Werte geändert wurde.
Sonst: Es werden nur die geänderten Werte geschrieben.
Standard Einstellung: 255
Bemerkung: Eine Wertänderung wird sofort aktiv und nichtflüchtig gespeichert.

HS_PDO_Index — Einfach Variable (Var) — 5FE5h

Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Mit diesem Parameter wird bestimmt, welches PDO das High-Speed PDO sein soll.
Kodierung:
0 : 1.PDO ist High-Speed-PDO
1 : 2.PDO ist High-Speed-PDO
Standard Einstellung: 0
Bemerkung: Ein geändeter Wert wird nichtflüchtig gespeichert, aber erst beim nächsten Einschalten oder Reset Node-Kommando aktiv. Zudem werden bei Wertänderung alle PDO's deaktiviert (Bit 31 von PDO Parameter CobID = 1).

PDIN1_Cycle_Time — Einfach Variable (Var) — 5FE6h

Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Bedeutung: Gibt die Zykluszeit an, in der die Prozesseingangsdaten des PDO1 im Zustand OPERATIONAL von der FU-Steuerung gelesen werden.
Kodierung: 1ms
Standard Einstellung: 25 = 25ms
Bemerkung: Ein geändeter Wert wird sofort aktiv und nichtflüchtig gespeichert.

PDIN2_Cycle_Time — Einfach Variable (Var) — 5FE7h

Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Bedeutung: Gibt die Zykluszeit an, in der die Prozesseingangsdaten des PDO2 im Zustand OPERATIONAL von der FU-Steuerung gelesen werden.
Kodierung: 1ms
Standard Einstellung: 100 = 100ms
Bemerkung: Ein geändeter Wert wird sofort aktiv und nichtflüchtig gespeichert.

SAVE_CAN_Baud — Einfach Variable (Var) — 5FFEh

Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Dient zum nichtflüchtigen Abspeichern der eingestellten CAN-Übertragungsgeschwindigkeit.
Kodierung Schreiben: FFh = nichtflüchtiges Abspeichern von CAN_Baud
0 = kein Abspeichern
Kodierung Lesen: FFh = eingestellter Wert stimmt mit abgespeichertem überein
00h = eingestellter Wert ungleich abgespeicherten Wert

CAN_Baud — Einfach Variable (Var) — 5FFFh

Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Index für CAN-Übertragungsgeschwindigkeit
Kodierung:

0	= 10	Kbit/s	5	= 250	Kbit/s
1	= 20	Kbit/s	6	= 500	Kbit/s*
2	= 50	Kbit/s	7	= 1000	Kbit/s*
3	= 100	Kbit/s	8	= 800	Kbit/s*
4	= 125	Kbit/s	9	= 25	Kbit/s

Standard Einstellung: 1
Bemerkung: Ein geänderter Wert wird sofort aktiv aber nicht automatisch nichtflüchtig gespeichert. Das Bit-Timing hält sich an die Vorgaben des Arbeitskreises Physical-Layer der CiA [2]. Zum Bit-Timing siehe Anhang. Welche Übertragungsgeschwindigkeiten gefahren werden können, hängt von der Leitungslänge, der Summe der Verzögerungszeiten und dem Bit-Timing ab und muß im Einzelfall geklärt werden.

CAN_Baud2 — Einfach Variable (Var) — 5FECh

Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Index für CAN-Übertragungsgeschwindigkeit alternativ zu CAN_Baud (s.o.)
Kodierung:

0	= 10	Kbit/s	5	= 250	Kbit/s
1	= 20	Kbit/s	6	= 500	Kbit/s*
2	= 50	Kbit/s	7	= 1000	Kbit/s*
3	= 100	Kbit/s	8	= 800	Kbit/s*
4	= 125	Kbit/s	9	= 25	Kbit/s

Standard Einstellung: 1
Bemerkung: Ein geänderter Wert wird im Gegensatz zum Parameter CAN_Baud sofort nichtflüchtig gespeichert, aber erst nach einem Reset Node-Kommando oder nach dem nächsten Einschalten aktiv.

* Beachte Kapitel ‘Wichtiger Warnhinweis’

8.2 Vom Kommunikations-profil [12] definierte Parameter

device type (nach CANopen (13)) Einfach Variable (Var) 1000h

Subindex: 0
Datenlänge: 4 Byte
Zugriff: Read_Only
Bedeutung: Beschreibt den Geräte-Typ nach CANopen-Kommunikationsprofil.
Kodierung: Bisher keine Festlegungen.
Standard Einstellung: 0
Bemerkung: Dieser Parameter ist konstant und kann deshalb nur gelesen werden.

error register (nach CANopen (13)) Einfach Variable (Var) 1001h

Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Only
Bedeutung: Zeigt den Fehler-Status des CANopen-Teilnehmers an.
Kodierung: Bit0 : = 1 ==> Fehler liegt vor
Standard Einstellung: 0
Bemerkung: Dieser Parameter kann nur gelesen werden. Als Quelle für diesen Parameter dient der Umrichter-Parameter Status (Ru.00). Die Umsetzung der Ru.00-Werte in Werte des Error-Register ist der Tabelle im Anhang zu entnehmen.
Beachten Sie, dass der Wert des error register nur bei aktivierter Emergency-Bearbeitung aktualisiert wird (--> Emergency Cycle).

Manufacturer Status Register Einfach Variable (Var) 1002h

SDO-Subindex: 0
Datenlänge: 4 Byte
Zugriff: Read_Only
Bedeutung: Direkte Abbildung des Parameters Umrichter-Status (RU.00) im DS301-Parameterbereich.
Kodierung: Siehe Beschreibung des Parameters Umrichter-Status (RU.00) in der Bedienungsanleitung der Umrichter-Steuerung.
Standard Einstellung: 0
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	JA	NEIN	JA

Bemerkung: Wird intern auf den Parameter RU.00 abgebildet.

Manufacturer Device Name — Einfach Variable (Var) — 1008h

SDO-Subindex: 0
Datenlänge: 4 Byte
Zugriff: Read_Only
Bedeutung: Gibt den Wert des Parameters Umrichter_Identifikation (SY.02) der FU-Steuerung als 4-Zeichen hexadezimal-String aus.
Kodierung: Der Wert 1234h würde wie folgt im SDO-Response-Telegramm übertragen:

B4	B5	B6	B7
31h	32h	33h	34h

Standard Einstellung: je nach Umrichertyp
PDO-Abbildung: keine Abbildung

Identify Object — Strukturierte Variable (Record) — 1018h

Subindex: 0 (Number of supported entries in the record)
Datenlänge: 1 Byte
Zugriff: Read_Only
Bedeutung: Gibt die Anzahl Einträge in diesem Objekt an.
Kodierung: 1
Standard Einstellung: 2
Bemerkung: Der Wert dieses Parameters kann nur gelesen werden.

Subindex: 1 (Vendor-ID)
Datenlänge: 4 Byte
Bedeutung: Herstellerbezeichnung von der CAN in Automation Nutzergruppe vergeben.
Kodierung: Bit31 . . . Bit24: Abteilung
Bit23 . . . Bit0: Firma
Standard Einstellung: 00000014h
Bemerkung: Der Wert dieses Parameters kann nur gelesen werden.

Subindex: 2 (Product code)
Datenlänge: 4 Byte
Bedeutung: Produktbezeichnung
Kodierung: 00000004h = Typ F4
00000005h = Typ F5
Standard Einstellung: 00000005h
Bemerkung: Der Wert dieses Parameters kann nur gelesen werden.

Manufacturer Software Version — Einfach Variable (Var) — 100Ah

SDO-Subindex: 0
Datenlänge: 4 Byte
Zugriff: Read_Only
Bedeutung: Gibt den Wert des Parameters Software-Version (IN.06) der FU-Steuerung als 4-Zeichen hexadezimal-String aus.
Kodierung: Der Wert 140h/260d (= Version 2.60) würde wie folgt im SDO-Response-Telegramm übertragen:

B4	B5	B6	B7
30h	31h	30h	34h

Standard Einstellung: je nach Software-Version der Umricher-Steuerung
PDO-Abbildung: keine Abbildung

1st receive PDO Parameter — Strukturierte Variable (Record) — 1400h

Subindex: 0 (Number of supported entries in the record)
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Gibt die Anzahl Einträge an, die unter diesem Objekt angesprochen werden können.
Kodierung: 1
Standard Einstellung: 2
Bemerkung: Der Wert dieses Parameters kann nur gelesen werden.

Subindex: 1 (COB-ID)
Datenlänge: 4 Byte
Bedeutung: Gibt an, auf welchem Identifier das PDO(rx) für den Transfer der Prozessaugangsdaten gesendet wird. Zusätzlich sind noch Steuerinformationen für dieses PDO in den obersten Bits enthalten.
Kodierung:
Bit31(MSB) = 0 ==> Die Bearbeitung der Prozessausgangsdaten ist aktiviert.
Bit31(MSB) = 1 ==> Bearbeitung der Prozessausgangsdaten ausgeschaltet.
Bit30 = 0 ==> Remote Frame auf dem entspr. Identifier wird beantwortet.
Bit30 = 1 ==> Remote Frame wird nicht beantwortet.
Bit29 = 0 ==> 11-bit Identifier (CAN V2.0A)
Bit29 = 1 ==> 29-bit Identifier (CAN V2.0B), hier nicht einstellbar.
Es werden aber 29-Bit-Identifier-Telegramme empfangen und bearbeitet.
Bit28...Bit0: Identifier (Bit0 = LSB), hier für Bit28 bis Bit11=fest=0.
Standard Einstellung: 00000200h + Node_Id
Bemerkung: Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert. Beim Einschalten der Prozessdatenbearbeitung (Bit31 von "1" auf "0") wird die Einstellung des Parameters 1st Receive PDO Mapping (Index = 1600h) zur Umrichtersteuerung transferiert. Sollte die FU-Steuerung die Abbildung nicht akzeptieren, wird an dieser Stelle eine Fehlerantwort zurückgegeben und die Prozessausgangsdatenbearbeitung bleibt ausgeschaltet. Wird die PD-Abbildung vom FU akzeptiert, wird diese automatisch nichtflüchtig gespeichert und die Prozessausgangsdatenbearbeitung wie gewünscht eingeschaltet. Da die Identifiervergabe der PDOs direkt von der Node_Id abgeleitet wird, können die Bits Bit28 bis Bit0 nur gelesen werden. Beim Schreiben werden diese Bits ignoriert.

Subindex: 2 (transmission type)
Datenlänge: 1 Byte
Bedeutung: Bestimmt, wann und wie dieses Objekt auf dem CAN-Bus gesendet wird.
Kodierung: **0 ... 240:**
Bei Empfang eines SYNC-Kommandos (Identifier = 128d, Datenlänge = 0) werden die aktuellen Prozessausgangsdaten zur FU-Steuerung transferiert.
254 (asynchron, herstellerspezifisch):
Die Prozessausgangsdaten werden zur FU-Steuerung transferiert, sobald sich mindestens ein Byte geändert hat.
255 (asynchron, profilspezifisch):
Siehe asynchron, herstellerspezifisch.
Bemerkung: Ein geändeter Wert wird sofort aktiv und nichtflüchtig gespeichert.
Beachten Sie auch den Einfluss des Parameters PDOUT_WrMode.

2nd receive PDO Parameter — Strukturierte Variable (Record) — 1401h

Subindex: 0 (Number of supported entries in the record)
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Gibt die Anzahl Einträge an, die unter diesem Objekt angesprochen werden können.
Kodierung: 1
Standard Einstellung: 2
Bemerkung: Der Wert dieses Parameters kann nur gelesen werden.

Subindex: 1 (COB-ID)
Datenlänge: 4 Byte
Bedeutung: Gibt an, auf welchem Identifier das PDO(rx) für den Transfer der Prozessaugangsdaten gesendet wird. Zusätzlich sind noch Steuerinformationen für dieses PDO in den obersten Bits enthalten.
Kodierung:
Bit31(MSB) = 0 ==> Die Bearbeitung der Prozessausgangsdaten ist aktiviert.
Bit31(MSB) = 1 ==> Bearbeitung der Prozessausgangsdaten ausgeschaltet.
Bit30 = 0 ==> Remote Frame auf dem entspr. Identifier wird beantwortet.
Bit30 = 1 ==> Remote Frame wird nicht beantwortet.
Bit29 = 0 ==> 11-bit Identifier (CAN V2.0A)
Bit29 = 1 ==> 29-bit Identifier (CAN V2.0B), hier nicht einstellbar. Es werden aber 29-Bit-Identifier-Telegramme empfangen und bearbeitet.
Bit28...Bit0: Identifier (Bit0=LSB), hier für Bit28 bis Bit11=fest=0.
Standard Einstellung: 80000300h + Node_Id
Bemerkung: Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert. Beim Einschalten der Prozessdatenbearbeitung (Bit31 von "1" auf "0") wird die Einstellung des 2nd Receive PDO Mapping auf ein entsprechendes Umrichter Mapping umgesetzt. Konnte dies erfolgreich durchgeführt werden, wird das Mapping automatisch nichtflüchtig abgespeichert. Da die Identifiervergabe der PDOs direkt von der Node_Id abgeleitet wird, können die Bits Bit28 bis Bit0 nur gelesen werden. Beim Schreiben werden diese Bits ignoriert.

Subindex: 2 (transmission type)
Datenlänge: 1 Byte
Bedeutung: Bestimmt, wann und wie dieses Objekt auf dem CAN-Bus gesendet wird.
Kodierung:
0 ... 240:
Bei Empfang eines SYNC-Kommandos (Identifier = 128d, Datenlänge = 0) werden die aktuellen Prozessausgangsdaten zur FU-Steuerung transferiert.
254 (asynchron, herstellerspezifisch):
Die Prozessausgangsdaten werden zur FU-Steuerung transferiert, sobald sich mindestens ein Byte geändert hat.
255 (asynchron, profilspezifisch):
Siehe asynchron, herstellerspezifisch.
Bemerkung: Ein geändeter Wert wird sofort aktiv und nichtflüchtig gespeichert.
Beachten Sie auch den Einfluss des Parameters PDOUT_WrMode.

1st receive PDO Mapping — Strukturierte Variable (Record) — 1600h

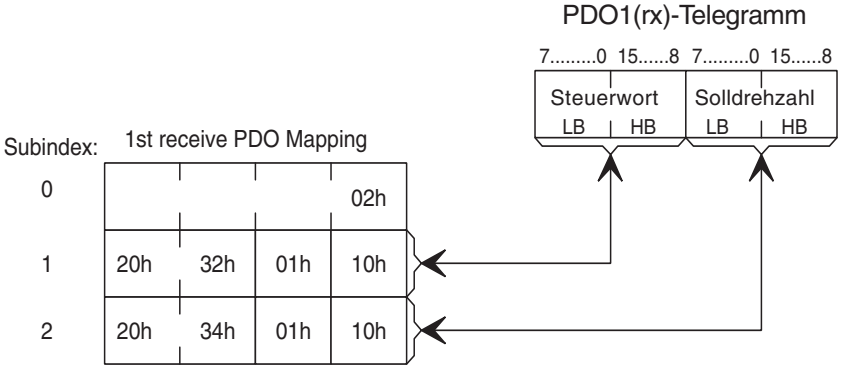
Subindex: 0 (Number of mapped objects in PDO)
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Gibt die Anzahl Einträge an, die unter diesem Objekt angesprochen werden können.
Kodierung: 1 (Maximal gültiger Wertebereich 1.....4).
Standard Einstellung: 2
Bemerkung: Ein Schreiben dieses Parameters bedingt das automatische Abschalten der Prozessausgangsdatenbearbeitung (Bit31 von Index 1400h, Subindex = 1 wird auf "1" gesetzt).

Subindex: 1 bis maximal 4 (nth object to be mapped)
Datenlänge: 4 Byte
Bedeutung: Bezeichnet eine Objektabbildung. Es wird der Index, Subindex und die Objektlänge in Bits angegeben.
Kodierung:

Index		*Sub-Index*	Objekt-Länge
HB	*LB*		
B3	*B2*	*B1*	*B0*

Standard Einstellung: s.u.
Bemerkung: Ein Schreiben dieses Parameters bedingt das automatische Abschalten der Prozessausgangsdatenbearbeitung (Bit31 von Index 1400h, Subindex = 1 wird auf "1" gesetzt).

Der Zusammenhang zwischen Prozessausgangsdatenabbildung und dem entsprechenden PDO1(rx)-Telegrammaufbau soll an der Standard-Belegung nochmals zusammengefaßt werden:



Beispiel für die Kodierung der Daten auf dem CAN-BUS:

Das erste abgebildete Objekt im Receive-PDO soll nicht das Steuerwort, sondern der Parameter mit dem Index 2302h und Subindex = 1 sein. In diesem Fall sind die 8 Bytes des Initiate-Domain-Download-Requests wie folgt zu füllen:

23h	00h	16h	01h	10h	01h	02h	23h
B0	B1	B2	B3	B4	B5	B6	B7

2nd receive PDO Mapping — Strukturierte Variable (Record) — 1601h

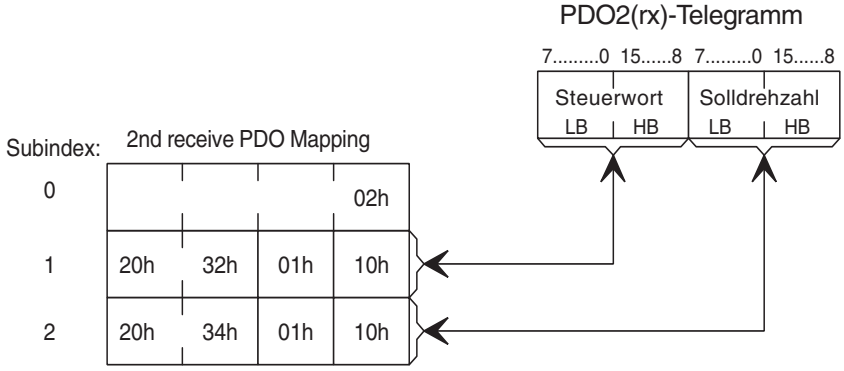
Subindex: 0 (Number of mapped objects in PDO)
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Gibt die Anzahl Einträge an, die unter diesem Objekt angesprochen werden können.
Kodierung: 1 (maximal gültiger Wertebereich 1.....4).
Standard Einstellung: 2
Bemerkung: Ein Schreiben dieses Parameters bedingt das automatische Abschalten der Prozessausgangsdatenbearbeitung (Bit31 von Index 1401h, Subindex = 1 wird auf "1" gesetzt).

Subindex: 1 bis maximal 4 (nth object to be mapped)
Datenlänge: 4 Byte
Bedeutung: Bezeichnet eine Objektabbildung. Es wird der Index, Subindex und die Objektlänge in Bits angegeben.
Kodierung:

Index		*Sub-Index*	Objekt-Länge
HB	*LB*		
B3	*B2*	*B1*	*B0*

Standard Einstellung: s.u.
Bemerkung: Ein Schreiben dieses Parameters bedingt das automatische Abschalten der Prozessausgangsdatenbearbeitung (Bit31 von Index 1401h, Subindex = 1 wird auf "1" gesetzt).

Der Zusammenhang zwischen Prozessausgangsdatenabbildung und dem entsprechenden PDO2(rx)-Telegrammaufbau soll an der Standard-Belegung nochmals zusammengefaßt werden:



1st transmit PDO Parameter — Strukturierte Variable (Record) — 1800h

Subindex:	0 (Number of supported entries in the record)
Datenlänge:	1 Byte
Zugriff:	Read_Write
Bedeutung:	Gibt die Anzahl Einträge in diesem Objekt an.
Kodierung:	1
Standard Einstellung:	3
Bemerkung:	Der Wert dieses Parameters kann nur gelesen werden.
Subindex:	1 (COB-ID)
Datenlänge:	4 Byte
Bedeutung:	Gibt an, auf welchem Identifier das PDO(tx) für den Transfer der Prozesseingangsdaten gesendet wird. Zusätzlich sind noch einige Zusatzinformationen in den obersten Bits enthalten.
Kodierung:	Bit31(MSB) = 0 ==> Die Bearbeitung der Prozesseingangsdaten ist aktiv. Bit31(MSB) = 1 ==> Bearbeitung der Prozesseingangsdaten ausgeschaltet. Bit30 = 0 ==> Remote Frame auf dem entspr. Identifier wird beantwortet. Bit30 = 1 ==> Remote Frame wird nicht beantwortet. Bit29 = 0 ==> 11-Bit-Identifier (CAN V2.0A) Bit29 = 1 ==> 29-Bit Identifier (CAN V2.0B), hier nicht einstellbar. Bit28...Bit0: Identifier (Bit0 = LSB), hier Bit28 bis Bit11=fest=0.
Standard Einstellung:	00000180h + Node_Id
Bemerkung:	Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert. Beim Einschalten der Prozessdatenbearbeitung (Bit31 von "1" auf "0") wird die Einstellung des Parameters 1st transmit PDO Mapping (Index 1A00h) zur Umrichtersteuerung transferiert. Sollte die FU-Steuerung die Abbildung nicht akzeptieren, wird an dieser Stelle eine Fehlerantwort zurückgegeben und die Prozesseingangsdatenbearbeitung bleibt ausgeschaltet. Wird die PD-Abb. vom FU akzeptiert, wird diese automatisch nichtflüchtig gespeichert und die Prozesseingangsdatenbearbeitung wie gewünscht eingeschaltet. Da die Identifiervergabe der PDOs direkt von der Node_Id abgeleitet wird, können die Bits Bit28 bis Bit0 nur gelesen werden. Beim Schreiben werden diese Bits ignoriert.
Subindex:	2 (transmission type)
Datenlänge:	1 Byte
Bedeutung:	Bestimmt, wann und wie dieses Objekt auf dem CAN-Bus gesendet wird.
Kodierung:	**0 (synchron azyklisch):** Bei jedem Empfang eines SYNC wird ein PDO(tx)-Telegramm auf CAN gesendet. **1 - 240 (synchron, zyklisch):** In diesem Einstellbereich wird über den Wert eingestellt, wie viele SYNC-Telegramme empfangen werden müssen, bevor ein PDO(tx)-Telegramm auf CAN gesendet wird. **252 (synchron-RTROnly):** Ein PDO(tx)-Telegramm wird nur nach einem Remote-Request auf dem PDO(tx)-Identifier gesendet. **253 (asynchron-RTROnly):** Ein PDO(tx)-Telegramm wird nur nach einem Remote-Request auf dem PDO(tx)-Identifier gesendet. **254 (asynchron, herstellerspezifisch):** Ein PDO(tx)-Telegramm wird gesendet, sobald sich mindestens ein Byte geändert hat. **255 (asynchron, profilspezifisch):** Ein PDO(tx)-Telegramm wird gesendet, sobald sich mindestens ein Byte geändert hat.
Bemerkung:	Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.
Subindex:	3 (inhibit time)
Datenlänge:	2 Byte
Bedeutung:	Bezeichnet den minimalen zeitlichen Abstand zwischen zwei CAN-Telegrammen auf diesem Identifier.
Kodierung:	100 µs
Standard Einstellung:	150 (= 15 ms)
Bemerkung:	Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert. Die interne Auflösung für die Inhibit-Time beträgt 1ms. Somit hat der eingestellte Wert eine Ungenauigkeit von +/- 1 ms.

2nd transmit PDO Parameter — Strukturierte Variable (Record) — 1801h

Subindex:	0 (Number of supported entries in the record)
Datenlänge:	1 Byte
Zugriff:	Read_Write
Bedeutung:	Gibt die Anzahl Einträge in diesem Objekt an.
Kodierung:	1
Standard Einstellung:	3
Bemerkung:	Der Wert dieses Parameters kann nur gelesen werden.
Subindex:	1 (COB-ID)
Datenlänge:	4 Byte
Bedeutung:	Gibt an, auf welchem Identifier das PDO(tx) für den Transfer der Prozesseingangsdaten gesendet wird. Zusätzlich sind noch einige Zusatzinformationen in den obersten Bits enthalten.
Kodierung:	Bit31(MSB) = 0 ==> Die Bearbeitung der Prozesseingangsdaten ist aktiv. Bit31(MSB) = 1 ==> Bearbeitung der Prozesseingangsdaten ausgeschaltet. Bit30 = 0 ==> Remote Frame auf dem entspr. Identifier wird beantwortet. Bit30 = 1 ==> Remote Frame wird nicht beantwortet. Bit29 = 0 ==> 11-Bit-Identifier (CAN V2.0A) Bit29 = 1 ==> 29-Bit Identifier (CAN V2.0B), hier nicht einstellbar. Bit28...Bit0: Identifier (Bit0 = LSB), hier Bit28 bis Bit11=fest=0.
Standard Einstellung:	80000280h + Node_Id
Bemerkung:	Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert. Beim Einschalten der Prozessdatenbearbeitung (Bit31 von "1" auf "0") wird die Einstellung des Parameters 2nd transmit PDO Mapping auf ein entsprechendes Umrichter Mapping umgesetzt. Konnte dies erfolgreich durchgeführt werden, wird das Mapping automatisch nichtflüchtig abgespeichert. Da die Identifiervergabe der PDOs direkt von der Node_Id abgeleitet wird, können die Bits Bit28 bis Bit0 nur gelesen werden. Beim Schreiben werden diese Bits ignoriert.
Subindex:	2 (transmission type)
Datenlänge:	1 Byte
Bedeutung:	Bestimmt, wann und wie dieses Objekt auf dem CAN-Bus gesendet wird.
Kodierung:	**0 (synchron azyklisch):** Bei jedem Empfang eines SYNC wird ein PDO(tx)-Telegramm auf CAN gesendet. **1 - 240 (synchron, zyklisch):** In diesem Einstellbereich wird über den Wert eingestellt, wie viele SYNC-Telegramme empfangen werden müssen, bevor ein PDO(tx)-Telegramm auf CAN gesendet wird. **252 (synchron-RTROnly):** Ein PDO(tx)-Telegramm wird nur nach einem Remote-Request auf dem PDO(tx)-Identifier gesendet. **253 (asynchron-RTROnly):** Ein PDO(tx)-Telegramm wird nur nach einem Remote-Request auf dem PDO(tx)-Identifier gesendet. **254 (asynchron, herstellerspezifisch):** Ein PDO(tx)-Telegramm wird gesendet, sobald sich mindestens ein Byte geändert hat. **255 (asynchron, profilspezifisch):** Ein PDO(tx)-Telegramm wird gesendet, sobald sich mindestens ein Byte geändert hat.
Bemerkung:	Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.
Subindex:	3 (inhibit time)
Datenlänge:	2 Byte
Bedeutung:	Bezeichnet den minimalen zeitlichen Abstand zwischen zwei CAN-Telegrammen auf diesem Identifier.
Kodierung:	100 µs
Standard Einstellung:	1000 (= 100 ms)
Bemerkung:	Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert. Die interne Auflösung für die Inhibit-Time beträgt 1ms. Somit hat der eingestellte Wert eine Ungenauigkeit von +/- 1 ms.

1st transmit PDO Mapping — Strukturierte Variable (Record) — 1A00h

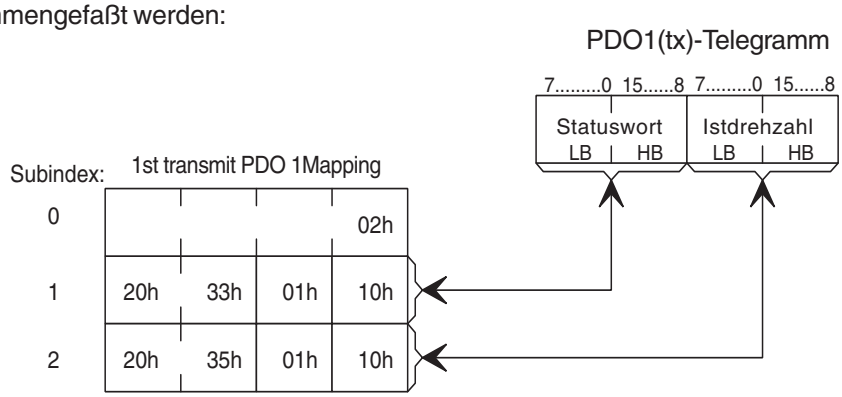
Subindex: 0 (Number of mapped objects in PDO)
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Gibt die Anzahl Einträge an, die unter diesem Objekt angesprochen werden können.
Kodierung: 1 (maximal gültiger Wertebereich 1.....4)
Standard Einstellung: 2
Bemerkung: Ein Schreiben dieses Parameters bedingt das automatische Abschalten der Prozesseingangsdatenbearbeitung (Bit31 von Index 1800h, Subindex=1 wird auf “1” gesetzt).

Subindex: 1 bis maximal 4 (nth object to be mapped)
Datenlänge: 4 Byte
Bedeutung: Bezeichnet eine Objektabbildung. Es wird der Index, Subindex und die Objektlänge in Bits angegeben.
Kodierung:

Index HB	Index LB	Sub-Index	Objekt-Länge
B3	B2	B1	B0

Standard Einstellung: s.u.
Bemerkung: Ein Schreiben dieses Parameters bedingt das automatische Abschalten der Prozesseingangsdatenbearbeitung (Bit31 von Index 1800h, Subindex=1 wird auf “1” gesetzt).

Der Zusammenhang zwischen Prozesseingangsdatenabbildung und dem entsprechenden PDO1(tx)-Telegrammaufbau soll an der Standard-Belegung nochmals zusammengefaßt werden:



2nd transmit PDO Mapping — Strukturierte Variable (Record) — 1A01h

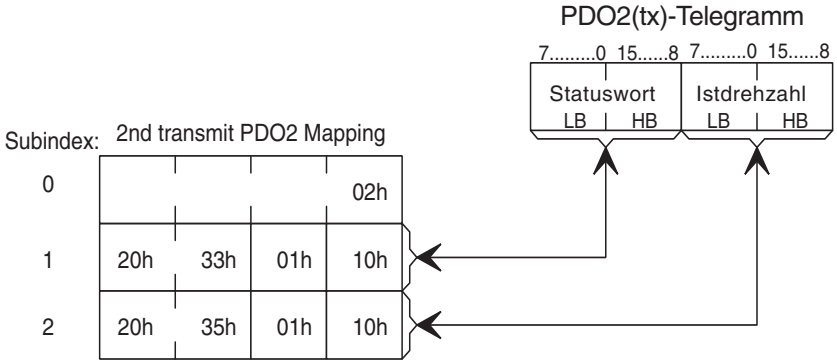
Subindex: 0 (Number of mapped objects in PDO)
Datenlänge: 1 Byte
Zugriff: Read_Write
Bedeutung: Gibt die Anzahl Einträge an, die unter diesem Objekt angesprochen werden können.
Kodierung: 1 (maximal gültiger Wertebereich 1.....4)
Standard Einstellung: 2
Bemerkung: Ein Schreiben dieses Parameters bedingt das automatische Abschalten der Prozesseingangsdatenbearbeitung (Bit31 von Index 1801h, Subindex=1 wird auf "1" gesetzt).

Subindex: 1 bis maximal 4 (nth object to be mapped)
Datenlänge: 4 Byte
Bedeutung: Bezeichnet eine Objektabbildung. Es wird der Index, Subindex und die Objektlänge in Bits angegeben.
Kodierung:

Index HB	Index LB	Sub-Index	Objekt-Länge
B3	B2	B1	B0

Standard Einstellung: s.u.
Bemerkung: Ein Schreiben dieses Parameters bedingt das automatische Abschalten der Prozesseingangsdatenbearbeitung (Bit31 von Index 1801h, Subindex=1 wird auf "1" gesetzt).

Der Zusammenhang zwischen Prozesseingangsdatenabbildung und dem entsprechenden PDO2(tx)-Telegrammaufbau soll an der Standard-Belegung nochmals zusammengefaßt werden:



8.3 Parameter für das Life-Guarding

Guard Time — Einfach Variable (Var) — 100Ch

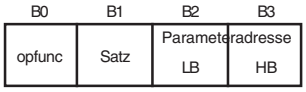
Bedeutung: Gibt zusammen mit dem Life Time Factor die Überwachungszeit für das Life-Guarding an.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung: 0 = Life-Guarding abgeschaltet, sonst: 1ms
Erlaubte PDO-Abbildung: Nicht abbildbar
Bemerkung: Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.

Life Time Factor — Einfach Variable (Var) — 100Dh

Bedeutung: Gibt zusammen mit der Guard Time die Überwachungszeit für das Life-Guarding an.
SDO-Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Kodierung: 0 = Life-Guarding abgeschaltet, sonst: 1
Erlaubte PDO-Abbildung: Nicht abbildbar
Bemerkung: Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.

LifeGuardTout.Addr — Einfach Variable (Var) — 5FDFh

Bedeutung: Bestimmt zusammen mit LifeGuardTout_Data die Funktion, die einmalig nach Auftreten des Life-Guarding Timeout ausgeführt wird.
SDO-Subindex: 0
Datenlänge: 4 Byte
Zugriff: Read_Write
Kodierung: Der Wert besteht aus der zu schreibenden Parameter-Adresse und dem Parameter-Satz sowie einem Funktionskode für den Operator. Die unten stehende Abbildung zeigt den Aufbau des Wertes, wie er im CAN-SDO-Telegramm erscheint:



opfunc = 0 --> keine Aktion im Operator
1 --> Übergang in Pre_Operational
Parameter-Adresse: Adresse des Parameters, der geschrieben wird
Satz: Satzauswahlbyte des zu schreibenden Parameters.
Erlaubte PDO-Abbildung: Nicht abbildbar
Standardeinstellung: Parameter-Adresse = 0032h(SY.50)
Parameter-Satz = 1(Satz0)
Opfunc = 1: Übergang in Pre_Operational
Bemerkung: Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.

LifeGuardTout.Data	Einfach Variable (Var)	**5FDEh**

Bedeutung:	Bestimmt den Wert des zu schreibenden Umrichter-Parameters bei auftretendem Life-Guarding-Timeout.
SDO-Subindex:	0
Datenlänge:	4 Byte
Zugriff:	Read_Write
Kodierung:	Je nach ausgewähltem Umrichter-Parameter.
Erlaubte PDO-Abbildung:	Nicht abbildbar
Standardeinstellung:	1
Bemerkung:	Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.

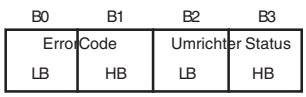
8.4 Parameter der Emergency-Bearbeitung

EmergencyCyle	Einfach Variable (Var)	**5FDDh**

Bedeutung: Dient zur Aktivierung/Deaktivierung der Emergency-Bearbeitung. Der Wert gibt zudem die Zykluszeit an, in der der Wert des Parameters Umrichter Status gelesen wird, um einen eventuellen Fehler anzuzeigen.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung: 0 = abgeschaltet, sonst: 1ms
Erlaubte PDO-Abbildung: Nicht abbildbar
Standardeinstellung: 0 (abgeschaltet)
Bemerkung: Ein geänderter Wert wird sofort aktiv und nichtflüchtig gespeichert.

Pre-defined ErrorField	Feld Variable (Array)	**1003h**

Bedeutung: Dieses Feld enthält die letzten fünf Fehlermeldungen. Dabei enthält das erste Fehlerfeld (Subindex=1) immer den zuletzt aufgetretenen Fehler. Einträge mit höherem Subindex sind dementsprechend früher aufgetreten.
SDO-Subindex: 0,1,2,3,4,5
Datenlänge: 4 Byte
Zugriff: Subindex = 0 : Read_Write, sonst: Read_Only
Kodierung: Subindex = 0 : Anzahl gefüllter Einträge
Subindex != 0 :



Erlaubte PDO-Abbildung: Nicht abbildbar
Standardeinstellung: 0 (abgeschaltet)
Bemerkung: Nur der Subindex = 0 ist schreibbar. Beim Schreiben mit dem Wert = 0 wird das komplette Feld zurückgesetzt.

8.5 Parameter für den Synchron-Modus

ComCyle	Einfach Variable (Var)	**1006h**
Bedeutung:	Dient zum Aktivieren/Deaktivieren des Synchron-Modus. Der Wert ist kodiert in µs, hat aber eine interne Auflösung von 1ms.	
SDO-Subindex:	0	
Datenlänge:	4 Byte	
Zugriff:	Read_Write	
Kodierung:	0 = OFF (normaler Modus), sonst: 1µs	
Wertebereich:	0, 1000, 2000, 3000 . . . 65000	
Bemerkung:	Der Parameter ist nur über CAN-SDO verfügbar. Der Parameter wird nicht nichtflüchtig gespeichert und hat nach jedem Neustart den Wert Null. Der CAN-Operator schaltet automatisch wieder in den normalen Modus zurück, wenn nach der vierfachen ComCycle-Zeit kein SYNC-Telegramm empfangen wird (s. auch HS_SyncToutDelay).	

HS_SyncToutDelay	Einfach Variable (Var)	**5FE0h**
Bedeutung:	Mit diesem Parameter kann die SYNC-Timeoutüberwachung um die gewünschte Anzahl von SYNCs verzögert werden.	
SDO-Subindex:	0	
Datenlänge:	2 Byte	
Zugriff:	Read_Write	
Kodierung:	0: Die SYNC-Timeoutüberwachung ist sofort nach der Umschaltung in den Synchron-Modus aktiv. Sonst: Anzahl SYNC-Telegramme, bis die Timeoutüberwachung aktiv wird.	
Wertebereich:	0 . . . 65535	
Bemerkung:	Der Parameter wird nichtflüchtig gespeichert.	

9. Zugriff auf Operator-Parameter über die Diagnose-schnittstelle

Die Operator-Parameter werden im Operator selbst verwaltet. Auf diese Parameter kann sowohl über die CAN-Schnittstelle als auch über die Diagnoseschnittstelle (per Combivis) zugegriffen werden. Dabei ist zu beachten, dass ein CAN-Parameter teilweise auf mehrere Operator-Parameter verteilt wird und somit das äußere Erscheinungsbild eines Parameters auf CAN und in Combivis leicht unterschiedlich sein kann. In diesem Kapitel werden die für den Anwender relevanten Operator-Parameter aufgelistet und der Verweis auf den entsprechenden CAN-Parameter genannt. Die komplette Beschreibung ist dann dem Kapitel Operator-Parameter zu entnehmen.

Zur Adressierung von Operator-Parametern und Parametern der Umrichter-Steuerung folgende Information: Alle Parameter eines KEB-Umrichters (FU-Parameter + Operator-Parameter) werden über die Diagnoseschnittstelle über eine 16-Bit Parameter-Adresse plus 8-Bit Satzauswahlbyte adressiert. Dabei gilt für die Parameter-Adresse, dass diese in die Parameter-Gruppenadresse (High-Byte) und eine laufende Nummer (Low-Byte) aufgeteilt ist:

KEB-Parameter-Adresse:

Jede Parameter-Gruppe kann sowohl mit Operator-Parametern als auch mit Parametern der Umrichter-Steuerung belegt sein. Zur Unterscheidung der Lokalisierung des Parameters ist die laufende „Nummer“ in zwei Bereiche aufgeteilt:

Nummer =	0 . . . 127	-->	Parameter der Umrichter-Steuerung
Nummer =	128 . . . 255	-->	Operator-Parameter

Des weiteren ist zu bemerken, dass, genau wie bei den Umrichter-Parametern, einige Operator-Parameter mehrfach vorhanden sind. Diese nutzen dann das Satzauswahlbyte in KEB gewohnter Weise. Wobei gilt, dass über Satz 0 das erste Abbild des Parameters, in Satz 1 das zweite angesprochen wird. Derzeit beschränken sich die satzprogrammierbaren Operator-Parameter auf die Prozessdaten-Parameter. Da der CAN-Operator derzeit zwei PDOs unterstützt, existieren diese Parameter in Satz 0 für das erste PDO und in Satz 1 für das zweite.

Alle satzprogrammierbaren Parameter haben unter *Bemerkung* einen entsprechenden Hinweis.Alle anderen Parameter existieren nur in Satz 0.

Beispiel:
Der Parameter PD_In_Para_CobId des ersten PDO wird über die Diagnoseschnittstelle adressiert mit Satzauswahlbyte = 01 (hex). Der gleiche Parameter des zweiten PDO liegt in Satz 1 (Satzauswahlbyte = 02 (hex)).
Bitte beachten Sie, dass die gleichzeitige Adressierung eines Operator-Parameters in mehreren Sätzen generell verboten ist.

Legende	**Parametername**	Kurzdarstellung	**Parameter-Adresse**

Operator type

OS.00 — **0180h**

Bedeutung: Beschreibt den Operator-Typ.
Datenlänge: 2 Byte
Satzprogrammierung: Nicht satzprogrammierbar
Kodierung: Die höchstwertigen beiden Dezimalstellen beschreiben den Kunden, die nächsten beiden Dezimalstellen den Typ, die nächsten beiden Dezimalstellen bezeichnen die Version aufgelöst in 0.1.
Beispiel:
Der Wert 10421 (dez) bedeutet:
Kunde = 1 (KEB)
Type = 4 (CAN)
Version = 21 (V 2.1)
Bemerkung: Nur zur Information.

Software date

OS.02 — **0182h**

Bedeutung: Gibt das Software-Datum an.
Datenlänge: 2 Byte
Kodierung: Die niedrigstwertige Dezimalstelle gibt das Jahr-2000 an.
Die nächsten beiden Dezimalstellen geben den Monat an.
Die nächsten beiden Dezimalstellen geben den Tag an.
Bemerkung: Die Darstellung in Combivis ist wie folgt: TTMM, Jahr-2000

Diag Error Count

OS.03 — **0183h**

Bedeutung: Gibt die Anzahl aufgetretener Fehler auf der Diagnoseschnittstelle an.
Datenlänge: 2 Byte
Kodierung: 1
Bemerkung: Nur zur Information.

Diag Response delay time

OS.04 — **0184h**

Bedeutung: Minimale Antwortverzugszeit für Anfragen über die Diagnoseschnittstelle.
Datenlänge: 1 Byte
Kodierung: 1 ms
Bemerkung: Dieser Parameter ist über CAN nicht verfügbar.

Diag Baudrate

OS.05 — **0185h**

Bedeutung: Gibt die Übertragungsgeschwindigkeit auf der Diagnoseschnittstelle an.
Datenlänge: 1 Byte
Kodierung:
0 : 1200 Bit/s
1 : 2400 Bit/s
2 : 4800 Bit/s
3 : 9600 Bit/s
4 : 19200 Bit/s
5 : 38400 Bit/s
Bemerkung: Der Wert des Parameters ist ein Abbild des Parameters SY.07 und ist deshalb über die Diagnoseschnittstelle nur lesbar.

HSP5 Max InvBusy retries	OS.06	0186h

Bedeutung: Gibt an, wie oft ein HSP5-Dienst zur Umrichtersteuerung wiederholt wird, wenn dieser den Dienst mit dem Fehler 'Umrichter beschäftigt' ablehnt.
Datenlänge: 1 Byte
Kodierung: 1

HSP5 Tout Count	OS.07	0187h

Bedeutung: Zählt die Zeitüberschreitungen bei der internen Kommunikation zwischen Operator und FU-Steuerung.
Datenlänge: 2 Byte
Kodierung: 1
Bemerkung: Dieser Parameter ist über CAN nicht verfügbar und existiert nur in Satz 0.

OP_Node_Id	FB.00	0280h

Datenlänge: 1 Byte
Bemerkung: Dieser Parameter ist identisch mit dem CAN-Parameter OP_Node_Id und existiert nur in Satz 0.

CAN_Baud2	FB.01	0281h

Datenlänge: 1 Byte
Bemerkung: Dieser Parameter entspricht dem CAN-Parameter CAN_Baud2 (s.o.) und existiert nur in Satz 0.

Act_CAN_Baud	FB.02	0282h

Bedeutung: Zeigt die aktuell eingestellte CAN-Bitrate an.
Datenlänge: 1 Byte
Kodierung: Siehe CAN_Baud.
Bemerkung: Dieser Parameter ist Read_Only und auf CAN nicht verfügbar und existiert nur in Satz 0.

Watchdog Activation	FB.03	0283h

Datenlänge: 1 Byte
Bemerkung: Dieser Parameter ist identisch mit dem CAN-Parameter Watchdog_Activation und existiert nur in Satz 0.

Watchdog inhibit	FB.04	0284h

Datenlänge: 1 Byte
Bemerkung: Dieser Parameter ist identisch mit dem CAN-Parameter Watchdog_Inhibit (s. o.) und existiert nur in Satz 0.

HS_PDO_Index	FB.05	0285h

Bemerkung: Dieser Parameter entspricht dem Operator-Parameter HS_PDO_Index und existiert nur in Satz 0.

DSP402_OpMode		FB.06	**0286h**
Datenlänge:	4 Byte		
Bemerkung:	Dieser Parameter entspricht dem Operator-Parameter DSP402_ModesOfOperation und existiert nur in Satz 0.		
PD_In_Para_CobID		FB.07	**0287h**
Datenlänge:	4 Byte		
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter „nth transmit PDO Parameter, Cob ID“ * und existiert in Satz 0 und Satz 1.		
PD_In_Para_TxType		FB.08	**0288h**
Datenlänge:	1 Byte		
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter „nth transmit PDO Parameter, TxType“ und existiert in Satz 0 und Satz 1.		
PD_In_Para_Inhibit		FB.09	**0289h**
Datenlänge:	2 Byte		
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter „nth transmit PDO Parameter, Inhibit Time“ und existiert in Satz 0 und Satz 1.		
PD_In_Cycle		FB.10	**028Ah**
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter PDIN_Cycle_Time und existiert in Satz 0 (PDIN1_Cycle_Time) und Satz 1 (PDIN2_Cycle_Time).		
Nr_PDIn_Objs		FB.11	**028Bh**
Datenlänge:	1 Byte		
Bemerkung:	Dieser Parameter entspricht dem niederwertigsten Byte (LSB) des CAN-Parameters nth transmit PDO Mapping, Nr Mapped Objects und existiert in Satz 0 und Satz 1.		
PD_Inx Index	(mit x = 1 . . . 4)	FB.12,15,18,21	**028Ch,028Fh,0292h,0295h**
Datenlänge:	2 Byte		
Bemerkung:	Diese Parameter entsprechen dem höchstwertigen Wort des Parameters nth transmit PDO Mapping, PDO Mapping for the nth application object to be mapped und existiert in Satz 0 und Satz 1.		
PD_Inx Set	(mit x = 1 . . . 4)	FB.13,16,19,22	**028Dh,0290h,0293h,0296h**
Datenlänge:	1 Byte		
Bemerkung:	Dieser Parameter entspricht dem dritthöchstwertigen Byte des Parameters nth transmit PDO Mapping, PDO Mapping for the nth application object to be mapped und existiert in Satz 0 und Satz 1.		
PD_Inx _BitDlen	(mit x = 1 . . . 4)	FB.14,17,20,23	**028Eh,0291h,0294h,0297h**
Bemerkung:	Dieser Parameter entspricht dem niederwertigsten Byte des Parameters nth transmit PDO Mapping, PDO Mapping for the nth application object to be mapped und existiert in Satz 0 und Satz 1.		

PDOUT_WrMode	FB.25	0299h

Bemerkung: Dieser Parameter entspricht dem CAN-Parameter „PDOUT_WrMode“ und existiert nur in Satz 0.

PD_Out_Para_CobID	FB.26	029Ah

Datenlänge: 4 Byte
Bemerkung: Dieser Parameter entspricht dem CAN-Parameter „nth Receive PDO Parameter, Cob ID“ und existiert in Satz 0 und Satz 1.

PD_Out_Para_TxType	FB.27	029Bh

Datenlänge: 1 Byte
Bemerkung: Dieser Parameter entspricht dem CAN-Parameter „nth Receive PDO Parameter, TxType“ und existiert in Satz 0 und Satz 1.

Nr_PDOut_Objs	FB.28	029Ch

Datenlänge: 1 Byte
Bemerkung: Dieser Parameter entspricht dem niederwertigsten Byte (LSB) des Parameters nth Receive PDO Mapping, PDO Mapping for the nth application object to be mapped und existiert in Satz 0 und Satz 1.

PD_Outx Index	(mit x = 1 . . . 4)	FB.29,32,35,38	029Dh,02A0h,02A3h,02A6h

Datenlänge: 2 Byte
Bemerkung: Dieser Parameter entspricht dem höchstwertigen Wort des Parameters nth Receive PDO Mapping, PDO Mapping for the nth application object to be mapped und existiert in Satz 0 und Satz 1.

PD_Outx Set	(mit x = 1 . . . 4)	FB.30,33,36,39	029Eh,02A1h,02A4h,02A7h

Datenlänge: 1 Byte
Bemerkung: Dieser Parameter entspricht dem dritthöchstwertigen Byte des Parameters nth Receive PDO Mapping, PDO Mapping for the nth application object to be mapped und existiert in Satz 0 und Satz 1.

PD_Outx_BitDlen	(mit x = 1 . . . 4)	FB.31,34,37,40	029Fh,02A2h,02A5h,02A8h

Bemerkung: Dieser Parameter entspricht dem niederwertigsten Byte des Parameters nth Receive PDO Mapping, PDO Mapping for the nth application object to be mapped und existiert in Satz 0 und Satz 1.

ProcessData Inx	(mit x = 1 . . . 4)	FB.42...45	02AAh - 02ADh

Bedeutung: x. Prozesseingangsdatenwort
Datenlänge: 2 Byte
Kodierung: je nach abgebildetem Parameter
Bemerkung: Dieser Parameter ist Read_Only und entspricht dem x. Wort des PDO (tx) - Telegramms auf CAN und existiert in Satz 0 und Satz 1.

ProcessData Outx	(mit x = 1 . . . 4)	FB.46...49	**02AEh - 02B1h**
Bedeutung:	x. Prozessausgangsdatenwort		
Datenlänge:	2 Byte		
Kodierung:	je nach abgebildetem Parameter		
Bemerkung:	Dieser Parameter ist Read_Only und entspricht dem x. Wort des PDO (rx) - Telegramms auf CAN und existiert in Satz 0 und Satz 1.		

Take Stored PD-Map	FB.50	**02B2h**
Bemerkung:	Dieser Parameter entspricht dem CAN Parameter PD_Stored (Index = 5FE2h), (s. o.) und existiert nur in Satz 0.	

Check PD Setting	FB.51	**02B3h**
Bedeutung:	Zeigt an, ob die zuletzt eingestellte PD-Belegungsänderung fehlerfrei durchgeführt wurde.	
Datenlänge:	1 Byte	
Kodierung:	0: Fehler aufgetreten in letzter PD-Belegungsänderung. 255d: Letzte PD-Belegungsänderung ist fehlerfrei durchgeführt worden.	
Bemerkung:	Dieser Parameter ist auf CAN nicht verfügbar und existiert nur in Satz 0.	

ComCycle	FB.52	**02B4h**
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter ComCycle und existiert nur in Satz 0.	

HS_SyncToutDelay	FB.53	**02B5h**
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter HS_SyncToutDelay und existiert nur in Satz 0.	

LifeGuardTout.Addr	FB.54	**02B6h**
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter LifeGuardTout.Addr und existiert nur in Satz 0.	

LifeGuardTout.Data	FB.55	**02B7h**
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter LifeGuardTout.Data und existiert nur in Satz 0.	

EmergencyCycle	FB.56	**02B8h**
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter EmergencyCycle und existiert nur in Satz 0.	

Save_VLRamps	FB.57	**02B9h**
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter Save_VLRamps und existiert nur in Satz 0.	

VL_Ramp_CalcMode	FB.58	**02BAh**
Bemerkung:	Dieser Parameter entspricht dem CAN-Parameter VL_Ramp_CalcMode und existiert nur in Satz 0.	

10. Umschaltung des transmission-type der PDOs

Der transmission-type sowohl des Parameters **1st/2nd receive PDO Parameter** als auch des **1st/2nd transmit PDO Parameter** ist veränderbar. Die gültigen Werte sind:

- Asynchron herstellerspezifisch (Wert = 254 = Standard) sowie
- Asynchron profilspezifisch (Wert = 255)
- Synchron azyklisch (Wert = 0)
- Synchron zyklisch (Wert = 1)
- Synchron zyklisch (Werte = 1, 2, . . . 240)
- Synchron RTROnly (Wert = 252)
- Asynchron RTROnly (Wert = 253)

Nach CANopen besitzen beim PDO(rx) die Werte 0 bis 240 identisches Verhalten. D. h., dass beim PDO(rx) die PDOUT-Daten bei jedem SYNC aktualisiert werden, egal welchen Wert der tx-type genau hat. Beachten Sie bitte auch den Einfluss des Parameters PDOUT_WrMode für die PDO1, PDO2(rx)-Bearbeitung.

10.1 Asynchron herstellerspezifisch (Wert = 254d/FEh) oder Asynchron profilspezifisch (Wert = 255d/FFh)

Wenn im Parameter **1st/2nd receive PDO Parameter** der transmission-type auf einen dieser Werte eingestellt ist, bedeutet dies, dass die Prozessausgangsdaten im Zustand OPERATIONAL beim Empfang eines gültigen PDO(rx)-Telegramms zur Umrichter-Steuerung transferiert werden, wenn sich mindestens 1 Byte geändert hat. Ein gültiges PDO(rx)-Telegramm ist ein Telegramm auf dem entsprechenden Identifier mit einer Datenlänge von ≥ der Datenlänge, die sich aus dem PDO(rx)-Mapping ergibt. Im Standardfall bedeutet das, dass alle Telegramme auf dem OUT-Identifier mit einer Datenlänge von ≥ 4 Byte akzeptiert werden.

Im Zustand OPERATIONAL werden zudem die Prozesseingangsdaten zyklisch von der FU-Steuerung gelesen. Wenn im Parameter **1st/2nd transmit PDO Parameter** der Wert 254d oder 255d eingestellt ist, bedeutet dies, dass ein PDO(tx)-Telegramm auf den CAN gesendet wird, wenn sich die Prozesseingangsdaten geändert haben.

10.2 Synchron azyklisch (Wert = 0) oder synchron zyklisch (Werte = 1 bis 240)

Wenn im Parameter **1st/2nd receive PDO Parameter** der transmission-type auf einen dieser Werte eingestellt ist, bedeutet dies, dass die Prozessausgangsdaten im Zustand OPERATIONAL nach Empfang eines SYNC-Telegramms zur Umrichter-Steuerung transferiert werden. Vorausgesetzt, zuvor wurde ein gültiges PDO(rx)-Telegramm empfangen.

Für den Parameter **1st/2nd transmit PDO Parameter** bedeutet der Wert transmission-type = 0/1, dass im Zustand OPERATIONAL ein PDO(tx)-Telegramm umgehend nach Empfang eines SYNC-Telegramms auf den CAN gesendet wird.

Für alle synchronen Werte des tx_type gilt, dass das SYNC-Telegramm das Senden des betreffenden PDO(tx) bzw. die Weiterleitung des PDO(rx) auslöst. Durch den genauen Wert wird dann nur noch bestimmt, wie viele SYNC-Telegramme dazu erforderlich sind. Bei den Werten 0 und 1 aktiviert jeder SYNC das entsprechende Ereignis. Bei den Werten 2 bis 240 gibt der Wert selbst die Anzahl erforderlicher SYNC-Telegramme an. Allerdings ist hierbei zu beachten, dass das Verhalten in diesem Wertebereich für alle PDO(rx)-Telegramme identisch mit dem Wert = 1 ist.

Beispiel:

PDO1(tx).tx_type = 10: Nach jeweils 10 SYNC-Telegrammen sendet der Slave eine PDO1(tx) auf CAN mit den aktuellen PDIN1-Daten.

PDO1(rx).tx_type = 10: Nach jedem SYNC-Telegramm werden die aktuellen PDOUT1-Daten weitergegeben.

10.3 Synchron/asynchron RTROnly (Werte = 252, 253)

Diese Werte sind nur gültig für Tx-PDOs. Das Senden der aktuellen PDIN-Daten über das entsprechende PDO(tx)-Telegramm auf CAN wird hierbei nur bei Empfang einen Remote-Frame-Request auf dem entsprechenden Identifier gestartet.

11. Synchron-Modus

Im Synchron-Modus wird über das SYNC-Telegramm auf CAN der interne Verarbeitungstakt des CAN-Operators und der angeschlossenen Frequenzumrichter-Steuerung vorgegeben. Kürzeste Verzögerungszeiten und vor allem extrem wenig zeitliche Abweichung in den Verzögerungszeiten sind bei dieser Betriebsart das oberste Ziel. Dieses wird erreicht bei gleichzeitiger Kompatibilität auf CAN. Allerdings sind mit dem Synchron-Modus deutliche Funktionseinschränkungen verbunden. Der generelle Betrieb bleibt aber erhalten.
Im Synchron-Modus liegt die höchste Priorität auf dem schnellst möglichen Transfer der Prozessdaten. Die Abbildung der Prozessdaten ist über die Prozessdatenabbildung einstellbar und unterliegt lediglich den Einschränkungen des schon bekannten High-Speed-PDOs im CAN-Operator.

Folgende Bedingungen gelten für den Synchron-Modus:
Der Synchron-Modus arbeitet nur im OPERATIONAL-Zustand des Knoten.
Es darf nur das High-Speed-PDO aktiv sein.
Das PDO arbeitet in beiden Richtungen Synchron.
Das PDO-Mapping in beiden Richtungen erfüllt folgende Bedingungen:
- Anzahl abgebildeter Parameter in beiden Richtungen = 2 oder 3:
- Die erste Abbildung belegt 32-Bit
- Jede weitere Abbildung belegt 16-Bit

Aktiviert wird der Synchron-Modus durch Schreiben auf den neuen Parameter ComCycle mit einem Wert ungleich Null.
Im Synchron-Modus müssen die SYNC-Telegramme auf CAN in der eingestellten Zeit (ComCycle) gesendet werden. Die maximale zeitliche Abweichung zweier aufeinander folgender SYNC-Telegramme darf ca. 80µs nicht überschreiten. Wird diese maximale Abweichung überschritten, muss im Einzelfall geklärt werden, ob die Applikation in gewünschter Weise funktioniert. Der CAN-Operator überwacht das Empfangen der SYNC-Telegramme. Wenn innerhalb der Timeoutzeit kein SYNC empfangen wird, schaltet der Operator automatisch zurück in den normalen Modus. Die Timeoutzeit entspricht dem vierfachen der erwarteten SYNC-Zykluszeit (ComCycle).

Die notwendigen Voreinstellungen sind nochmals in folgender Tabelle zusammegefaßt:

Index	Subindex	Dlen	Wert
5FE5h	0	1	0
1801h	1	4	8000XXXXh
1401h	1	4	8000XXXXh
1800h	2	1	0 oder 1
1400h	2	1	0 oder 1
1A00h	0	1	≤ 3
1A00h	1	4	XXXXXX20h
1A00h	2	4	XXXXXX10h
1A00h	3	4	XXXXXX 10h
1600h	0	1	≤ 3
1600h	1	4	XXXXXX 20h
1600h	2	4	XXXXXX 10h
1600h	3	4	XXXXXX 10h
1006h	0	4	Vielfaches von 1000(dez)

11.1 Funktionsein-schränkungen im Synchron-Modus

Im Synchron-Modus werden alle CAN-SDO-Aufträge und Aufträge von der Diagnoseschnittstelle mit in den Prozessdatentransfer eingeflochten. Aus diesem Grund sind dann nur noch CAN-SDO-Zugriffe auf Parameter in der Umrichter-Steuerung mit Subindex = 0 möglich. D.h. es können Parameter im Umrichter nur noch in dem durch den Satzzeiger (Fr.09) bestimmten Satz angesprochen werden (indirekte Satzadressierung). Beachten Sie bitte auch, dass jedes CAN-Telegramm den SYNC zeitlich verschieben kann, auch wenn das SYNC-Telegramm durch seinen niedrigen Identifier eine sehr hohe Priorität besitzt. Aus diesem Grund ist nach Möglichkeit im Synchron-Modus auf jegliche andere CAN-Kommunikation zu verzichten (Node-Guarding, SDO-Kommandos, NMT-Kommandos). Es sollten dann wirklich nur PDO(rx)-Telgramme, PDO(tx)-Telegramme und der SYNC gesendet werden.

Die Tastatur wird im Synchron-Modus nicht bearbeitet. Die Anzeige ist statisch und zeigt ‚Synch' an. Die Diagnoseschnittstelle arbeitet weiter mit ähnlichen Einschränkungen, die für die CAN-SDO-Kommunikation gilt: Es können Parameter im Umrichter nur noch über den HSP5-Dienst = 1 mit dem Satzauswahl-Byte = 1 (indirekte Satzadressierung über Fr.09) gelesen oder geschrieben werden.

12. DSP402-Unterstützung

Welche DSP402-Modi unterstützt werden, ist der Beschreibung des Parameters DSP402_ModesOfOPeration zu entnehmen.
Die CAN in Automation Nutzergruppe hat am 26.07.2002 die Version 2.0 des DSP402-Geräteprofils für Antriebe veröffentlicht. Die KEB-F5-CANopen-Anschaltung unterstützt eine Teilmenge der Funktionen und Parameter, die in dem DSP402 definiert sind. Dabei übernimmt der CAN-Operator die Umsetzung der DSP402-Parameter auf Parameter der Umrichter-Steuerung. Diese Umsetzung ist teilweise aufwendig und deshalb auch laufzeitintensiv. Aus diesem Grund ist eine Abbildung solcher Parameter, die umkodiert werden müssen, auf das High-Speed-PDO in den meisten Fällen nicht erlaubt. Die DSP402-Parameter können aber über SDO-Kommandos angesprochen werden. Ebenso sind fast alle DSP402-Parameter auf das Low-Speed-PDO abbildbar.
Einige Parameter im KEB-F5-Frequenzumrichter, die als Basis für realisierte DSP402-Parameter dienen, sind satzprogrammierbar. Da das DSP402-Profil keine Satzprogrammierung unterstützt, wurde für die DSP402-Realisierung folgende Festlegung getroffen: Alle DSP402-Profilparameter, die auf Parameter in der Frequenzumrichtersteuerung umgesetzt werden, werden im Satz0 abgelegt. Auf die Parameter in anderen Sätzen haben diese keinen Einfluss:

Das DSP402-Profil arbeitet ausschließlich in Parameter-Satz 0

12.1 Voreinstellungen für DSP402-Betrieb

Das DSP402-Profil unterstützt keine Unterscheidung der Rampen für Rechtslauf und für Linkslauf. Aus diesem Grund müssen die Rampenzeiten für Rechts- und Linkslauf gleiche Werte haben. Das bedingt die folgenden Voreinstellungen in der Umrichter-Steuerung:

Parameter	Parameter-Adresse	Parameter-Satz	Parameter-Wert
OP.29	031Dh	Satz 0	-1
OP.31	031Fh	Satz 0	-1

Zum Betrieb über das DSP402-Steuer-und Statuswort sind folgende Voreinstellungen in der Umrichter-Steuerung vorzunehmen:

Parameter	Parameter-Adresse	Parameter-Satz	Parameter-Wert
UD.01	0801h	Satz	440
OP.00	0300h	Satz 0	5
OP.01	0301h	Satz 0	6
OP.02	0302h	Satz 0	0
OP.60	033Ch	Satz 0	0
OP.61	033Dh	Satz 0	0
DI.01	0B01h	Satz 0	Bit0 = 1
DI.02	0B02h	Satz 0	Bit0 = 1
DI.09	0B09h	Satz 0	2

12.2 Hinweise zu den DSP402-Velocity Rampen

Das DSP402-Profil definiert eine Velocity-Rampe(VL-Rampe) als eine Struktur aufgebaut aus zwei Teilen:

- VL-Rampe.Dspeed: Delta-Geschwindigkeits-Wert der Rampe in rpm.
- VL-Rampe.Dtime: Delta-Zeit-Wert der Rampe in Sekunden.

Intern in der Umrichter-Steuerung wird eine Rampe durch einen festen Teil, den sogenannten Rampen-Referenzwert und einen einstellbaren Wert, die Rampenzeit festgelegt. Die DSP402-Rampenwerte werden im CAN-Operator gehalten. Bei einem Lesezugriff wird auf diese Zwischenspeicher-Werte zugegriffen. Wird einer der Werte geschrieben ist zusätzlich ein Schreibzugriff auf die entsprechende Rampenzeit in der Umrichter-Steuerung erforderlich. Die Rampenwerte nach DSP402-Kodierung werden im Operator nicht automatisch nichtflüchtig gespeichert. Dies kann der Anwender explizit anfordern über den Parameter Save_VL_Ramps.

Die Umsetzung einer VL-Rampe in eine FU-Rampenzeit ist eindeutig. Allerdings ist die ebenfalls notwendige Umsetzung einer FU-Rampenzeit in eine VL-Rampe nicht eindeutig. Aus diesem Grund musste eine Methode für diese Rückkonvertierung gefunden werden. Der KEB-F5-CANopen-Operator unterstützt verschiedene Rückkonvertierungsmodi, welche über den Parameter VL_Ramp_CalcMode ausgewählt werden können (s.u.).

Weiterhin ist zu beachten, dass bei jeder Veränderung einer der beiden VL-Rampenteile immer ein Schreibzugriff auf die relevante FU-Rampenzeit durchgeführt wird. Das bedeutet, dass bei einer Änderung beider VL-Rampenteile zunächst der eine Teil in eine FU-Rampenzeit umgesetzt wird. Diese FU-Rampenzeit entspricht zu diesem Zeitpunkt nicht der gewünschten Rampe. Erst wenn danach auch der zweite VL-Rampenteil geschrieben wird, wird die gewünschte Rampe als FU-Rampenzeit vorgegeben. Diese Problematik gilt auch für das Low-Speed-PDO. Das DSP402-Profil macht für die konsistente Vorgabe der VL-Rampen keine Vorschriften. Das eben geschilderte Problem muss demnach der Anwender lösen:

- Z.B. kann man einen der beiden VL-Rampenteile immer unverändert lassen und die Rampe nur über den anderen Teil der VL-Rampe variieren.
- Ebenso wäre ein Ansatz die Rampen niemals zu ändern während der FU Rampen fährt.

12.3 DSP402-Profil und Synchron-Modus

Es ist zwar generell möglich auch im Synchron-Modus über DSP402-Profilparameter zu arbeiten, allerdings ist dabei zu berücksichtigen, dass im Synchron-Modus keine Konvertierung/Dekonvertierung von Parameterwerten unterstützt wird. Demzufolge sind umzurechnende DSP402-Parameter im Synchron-Modus nicht auf Prozessdaten abbildbar. Die Parameter, die keine Umrechnung benötigen sind:

- VL_TargetVelocity(Index=6042h)
- VL_ControlEffort(Index=6043h)

Ebenso ist im Synchron-Modus der direkte satzadressierte SDO-Zugriff nicht erlaubt. Also würden die meisten SDO-Zugriffe auf DSP402-Parameter mit Fehler abgelehnt. Aus diesen Gründen ist es praktisch unrealistisch, im Synchron-Modus mit DSP402-Parametern zu arbeiten.

12.4 Allgemeine Parameter des DSP402-Profils

Legende | **Parametername** | Objekt-Typ | **CAN-SDO-Index**

DSP402_ErrorCode — Einfach Variable (Var) — 603Fh

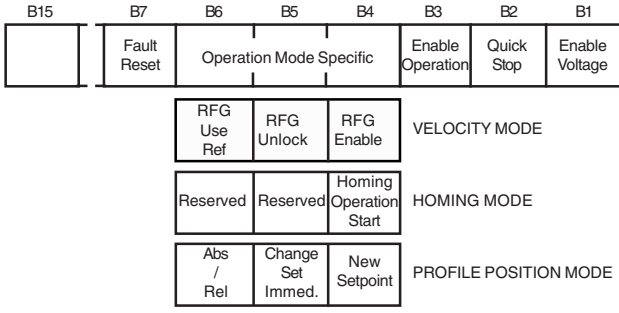
Bedeutung: Gibt den aktuellen Fehlerstatus des Gerätes an.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Only
Kodierung: Nach DSP402-Vorgabe, s. Tabelle im Anhang
PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	JA	NEIN	JA

Bemerkung: Wird intern auf den Parameter RU.00 abgebildet. Wenn der FU keinen Fehlerzustand meldet, aber der CAN-Operator einen Fehler erkannt hat, wird dieser zurückgegeben.

DSP402_Controlword — Einfach Variable (Var) — 6040h

Bedeutung: Dient zur Vorgabe von Steuerungskommandos. Der Parameter ist bitkodiert und wird in der Umrichter-Steuerung auf den Parameter SY.50 (Steuerwort) abgebildet.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung: Im folgenden Bild werden nur die unterstützten Bits aufgeführt:



Die Bits B6 bis B4 sind modeabhängig definiert. Grau hinterlegte Bits sind derzeit in der KEB-CANopen-Anschaltung nicht realisiert.

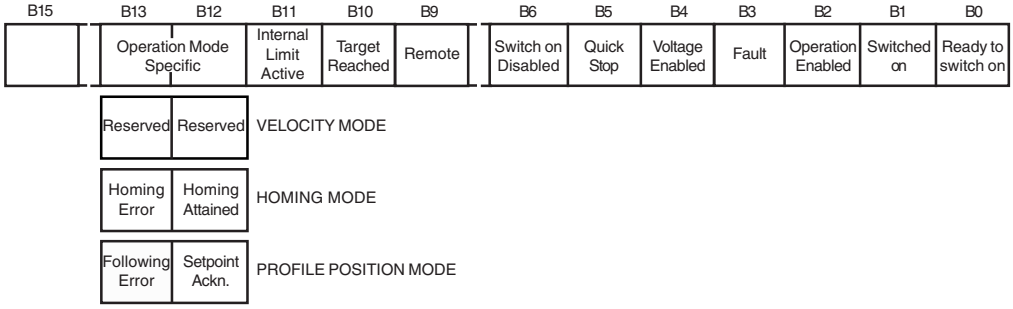
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
JA	JA	JA	JA

Bemerkung: Wird intern auf den Parameter SY.50 abgebildet.

DSP402_Statusword — Einfach Variable (Var) — 6041h

Bedeutung: Dient zur Bekanntgabe des aktuellen Zustands. Der Parameter ist bitkodiert und wird in der Umrichter-Steuerung auf den Parameter SY.51 (Statuswort) abgebildet.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Only
Kodierung: Im folgenden Bild werden nur die unterstützten Bits aufgeführt:



Die Bits B13, B12 sind modeabhängig definiert. Grau hinterlegte Bits sind derzeit in der KEB-CANopen-Anschaltung nicht realisiert.

Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	JA	NEIN	JA

Bemerkung: Wird intern auf den Parameter SY.51 abgebildet.

DSP402_ModesOfOperation — Einfach Variable (Var) — 6060h

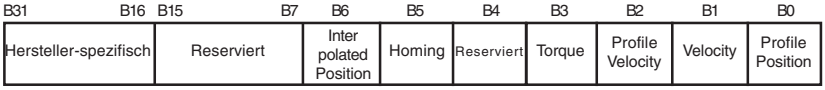
Bedeutung: Dient zur Vorgabe des gewünschten DSP402-Arbeitsmodus.
SDO-Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Kodierung:
(-1): Herstellerspezifisch
0: Reserviert
1: Profile Position Mode[*1]
2: Velocity Mode
3: Profile Velocity Mode(hier nicht möglich)
4: Torque Profile Mode(hier nicht möglich)
5: Reserviert
6: Homing Mode[*1]
7: Interpolated Position Mode(hier nicht möglich)
Standardeinstellung: (-1): Herstellerspezifisch
Erlaubte PDO-Abbildung: Nicht abbildbar
Bemerkung: Derzeit besteht kein realer Unterschied zwischen den Modi (-1) und 2.
[*1]: Diese Modi sind nur bei Steuerungstypen ud.02 = 4, 5, 6, 8, 9, 10 erlaubt.

DSP402_ModesOfOperationDisp — Einfach Variable (Var) — 6061h

Bedeutung: Gibt den aktuellen DSP402-Arbeitsmodus an.
SDO-Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Only
Kodierung: s. DSP402_ModesOfOperation
Standardeinstellung: (-1): Herstellerspezifisch
Erlaubte PDO-Abbildung: Nicht abbildbar

DSP402_SuppDriveModes — Einfach Variable (Var) — 6502h

Bedeutung: Gibt bitkodiert die unterstützten Modi der CANopen-Anschaltung an.
SDO-Subindex: 0
Datenlänge: 4 Byte
Zugriff: Read_Only
Kodierung:



Die herstellerspezifischen Bits werden hier nicht genutzt.
Erlaubte PDO-Abbildung: Nicht abbildbar

DSP402_AbortConnOptionCode — Einfach Variable (Var) — 6007h

Bedeutung: Bestimmt das Verhalten nachdem die Verbindung zum CAN abgebrochen wurde. Abbruch der Verbindung zum CAN ist hier gleich bedeutend mit den folgenden Ereignissen:
- Ansprechen der Life Guarding Timeout Überwachung
- BusOff-Zustand des CAN-Controllers

SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung:
(**-1**): Das Verhalten nach einem Verbindungsabbruch auf CAN wird durch die beiden Parameter LifeGuardTout_Addr und LifeGuardTout_Data bestimmt, s.o.
0: Keine Aktion
1: Übergang in PRE_OPERATIONAL, wenn aktueller Zustand OPERATIONAL und explizites Auslösen des Feldbus-Watchdog (E.Bus) bei der Umrichter-Steuerung (wenn diese im FU aktiviert).
2: Übergang in PRE_OPERATIONAL, wenn aktueller Zustand OPERATIONAL und Komanndo, Disable Voltage' über das DSP402_Controlword.
3: Übergang in PRE_OPERATIONAL, wenn aktueller Zustand OPERATIONAL und Komanndo, Quick Stop' über das DSP402_Controlword.
Standardeinstellung: (-1)
Erlaubte PDO-Abbildung: Nicht abbildbar
Bemerkung: Eine Wertänderung wird sofort aktiv und nichtflüchtig gespeichert.

DSP402_MotionProfileType — Einfach Variable (Var) — 6086h

Bedeutung: Gibt die Art der Realisierung von Bewegungen an.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung:
(-1): Lineare Rampen mit aktivierbaren s-Kurven
0: Lineare Rampen (hier nicht einstellbar)
1: sin^2-Rampen (hier nicht einstellbar)
2: Ruckfreie Rampen (hier nicht einstellbar)
3: Ruckbegrenzte Rampen (hier nicht einstellbar)
Standardeinstellung: (-1)
Erlaubte PDO-Abbildung: Nicht abbildbar
Bemerkung: Derzeit ist nur ein Wert wählbar.

12.5 Parameter des Velocity Mode

VL_TargetVelocity — Einfach Variable (Var) — 6042h

Bedeutung: Gibt die Sollgeschwindigkeit vor.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung: 1 rpm
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
JA	JA	JA	JA

Bemerkung: Wird intern auf den Parameter SY.52 abgebildet.

VL_VelocityDemand — Einfach Variable (Var) — 6043h

Bedeutung: Gibt den Geschwindigkeitswert am Ausgang des Rampengenerators an.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Only
Kodierung: 1 rpm
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	JA	NEIN	JA

Bemerkung: Wird intern auf den Parameter RU.02 abgebildet.

VL_ControlEffort — Einfach Variable (Var) — 6044h

Bedeutung: Gibt den Geschwindigkeitsistwert an.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Only
Kodierung: 1 rpm
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	JA	NEIN	JA

Bemerkung: Wird intern auf den Parameter SY.53 abgebildet.

VL_VelocityMinAmount — Einfach Variable (Var) — 6046h

Bedeutung: Gibt den Betrag des unteren Grenzwertes des Sollwertes an.
SDO-Subindex: 1
Datenlänge: 4 Byte
Zugriff: Read_Write
Kodierung: 1 rpm
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	NEIN	JA	JA

Bemerkung: Wird intern auf den Parameter OP.06 abgebildet.

VL_VelocityMaxAmount — Einfach Variable (Var) — 6046h

Bedeutung: Gibt den Betrag des oberen Grenzwertes des Sollwertes an.
SDO-Subindex: 2
Datenlänge: 4 Byte
Zugriff: Read_Write
Kodierung: 1 rpm
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	NEIN	JA	JA

Bemerkung: Wird intern auf den Parameter OP.10 abgebildet.

VL_VelocityAcceleration.Dspeed — Einfach Variable (Var) — 6048h

Bedeutung: Gibt zusammmen mit VL_VelocityAcceleration.Dtime die Beschleinigungsrampe vor.
SDO-Subindex: 1
Datenlänge: 4 Byte
Zugriff: Read_Write
Kodierung: 1 rpm
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	NEIN	JA	JA

Bemerkung: Wird intern auf den Parameter OP.28 abgebildet.

VL_VelocityAcceleration.Dtime — Einfach Variable (Var) — 6048h

Bedeutung: Gibt zusammmen mit VL_VelocityAcceleration.Dspeed die Beschleinigungsrampe vor.
SDO-Subindex: 2
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung: 1s
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	NEIN	JA	JA

Bemerkung: Wird intern auf den Parameter OP.28 abgebildet.

VL_VelocityDeceleration.Dspeed — Einfach Variable (Var) — 6049h

Bedeutung: Gibt zusammmen mit VL_VelocityDeceleration.Dtime die Verzögerungsrampe vor.
SDO-Subindex: 1
Datenlänge: 4 Byte
Zugriff: Read_Write
Kodierung: 1 rpm
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	NEIN	JA	JA

Bemerkung: Wird intern auf den Parameter OP.30 abgebildet.

VL_VelocityDeceleration.Dtime

Einfach Variable (Var) **6049h**

Bedeutung: Gibt zusammmen mit VL_VelocityDeceleration.Dspeed die Verzögerungsrampe vor.
SDO-Subindex: 2
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung: 1s
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	NEIN	JA	JA

Bemerkung: Wird intern auf den Parameter OP.30 abgebildet.

VL_VelocityQuickStop.Dspeed

Einfach Variable (Var) **604Ah**

Bedeutung: Gibt zusammmen mit VL_VelocityQuickStop.Dtime die Schnellhaltrampe vor.
SDO-Subindex: 1
Datenlänge: 4 Byte
Zugriff: Read_Write
Kodierung: 1 rpm
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	NEIN	JA	JA

Bemerkung: Wird intern auf den Parameter PN.60 abgebildet.

VL_VelocityQuickStop.Dtime

Einfach Variable (Var) **604Ah**

Bedeutung: Gibt zusammmen mit VL_VelocityQuickStop.Dspeed die Schnellhaltrampe vor.
SDO-Subindex: 2
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung: 1s
Erlaubte PDO-Abbildung:

High-Speed-PDO		Low-Speed-PDO	
rx	tx	rx	tx
NEIN	NEIN	JA	JA

Bemerkung: Wird intern auf den Parameter PN.60 abgebildet.

VL_QuickStopOptionCode

Einfach Variable (Var) **605Ah**

Bedeutung: Gibt das Verhalten des Schnellhalt vor.
SDO-Subindex: 0
Datenlänge: 2 Byte
Zugriff: Read_Write
Kodierung: Das DSP402-Profil bestimmt den Wertebereich für herstellerspezifische Modi in dem Bereich –32768....-1. Aus diesem Grund wird im Operator der Wert auf die interne Kodierung des Parameters PN.58 wie folgt umgerechnet:
Wert(PN.58) = Betrag(VL_QuickStopOptionCode) - 1
Die Bedeutung der einzelnen Modi sind der Applikationsanleitung der eingesetzten Umrichter-Steuerung zu entnehmen.
-1 --> PN.58 = 0
-2 --> PN.58 = 1
-3 --> PN.58 = 2 usw.
Erlaubte PDO-Abbildung: Nicht abbildbar
Bemerkung: Wird intern abgebildet auf den Parameter PN.58.

VL_PoleNr — Einfach Variable (Var) — 604Dh

Bedeutung: Gibt die Polzahl des Motors an. Dieser Wert wird aus den Parametern DR.01 und DR.05 berechnet und für alle Umrechnungen von Drehzahl(rpm) in Frequenzen(Hz) benötigt.
SDO-Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Only
Kodierung: 1
Erlaubte PDO-Abbildung: Nicht abbildbar
Bemerkung: Wird intern aus den Parametern DR.01 und DR.05 berechnet.

VL_Ramp_CalcMode — Einfach Variable (Var) — 5FDBh

Bedeutung: Bestimmt die Berechnungsweise für die Umrechnung einer KEB-Rampenzeit in eine DSP402-Velocity-Rampe.
SDO-Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Kodierung:
0: Beide Teile der VL-Rampe (Dspeed, Dtime) werden so bestimmt, dass die Werte möglichst klein werden, aber die Genauigkeit der umzusetzenden Rampenzeit erhalten bleibt.
1: Nur VL-Rampe.Dtime wird berechnet, VL-Rampe.Dspeed bleibt unverändert.
2: Als Wert für VL-Rampe.Dtime wird der Wert der KEB-Rampenzeit übernommen. VL-Rampe.Dspeed wird dementsprechend gesetzt.
Erlaubte PDO-Abbildung: Nicht abbildbar
Bemerkung: Eine Wertänderung wird sofort aktiv und nichtflüchtig gespeichert.

Save_VL_Ramps — Einfach Variable (Var) — 5FDCh

Bedeutung: Dient zum nichtflüchtigen Abspeichern der Velocity-Mode Rampen im CAN-Operator.
SDO-Subindex: 0
Datenlänge: 1 Byte
Zugriff: Read_Write
Kodierung:

B7	B6	B5	B4	B3	B2	B1	B0
		Save VL-QST. Dtime	Save VL-DEC. Dtime	Save VL-ACC. Dtime	Save VL-QST. Dspeed	Save VL-DEC. Dspeed	Save VL-ACC. Dspeed

mit VL-ACC.Dspeed: Index=6048h,Subindex=1
mit VL-DEC.Dspeed: Index=6049h,Subindex=1
mit VL-QST.Dspeed: Index=604Ah,Subindex=1
mit VL-ACC.Dtime: Index=6048h,Subindex=2
mit VL-DEC.Dtime: Index=6049h,Subindex=2
mit VL-QST.Dtime: Index=604Ah,Subindex=2
Erlaubte PDO-Abbildung: Nicht abbildbar.
Bemerkung: Beim Lesen wird immer der Wert 0 zurückgeliefert.

13. Faktoren

Das DSP402-Profil definiert viele Parameter mit sog. Anwender-Einheiten. Um diese Parameter zu realisieren, müssen Umrechnungsfaktoren vorhanden sein, die die Umrechnung in die internen Grössen vornehmen. Zu diesem Zweck spezifiziert das Profil eine eigene Gruppe von Parametern die sog. Factor Group. Die KEB-DSP402-Realisierung unterstützt keine Parameter dieser Gruppe. Es werden aber die folgenden Faktoren zur Umrechnung von Einheiten unterstützt, die bei verschiedenen DSP402-Parametern zum Einsatz kommen.

Jeder Faktor besteht aus einem vorzeichenlosen 32-Bit Zähler und einem vorzeichenlosen 32-Bit Nenner. Jeder Faktor ist als Struktur mit drei Mitgliedern definiert (wie unten beschrieben). Die genaue Berechnungsformel wird jeweils bei der Beschreibung der betroffenen Parameter aufgeführt:

- **Factor0**: Umrechnung von anwenderspezifischen Weg-Einheiten in die vom Frequenzumrichter verwendeten Weg-Einheiten.
- **Factor1**: Umrechnung von anwenderspezifischen Geschwindigkeit-Einheiten in die vom Frequenzumrichter verwendeten Geschwindigkeit-Einheiten.
- **Factor2**: Umrechnung von anwenderspezifischen Beschleunigung-Einheiten in die vom Frequenzumrichter verwendeten Beschleunigung-Einheiten.

FactorX.NrEntries	Einfach Variable (Var)	**5FC0h + X**

Bedeutung:	Anzahl Mitglieder in der Struktur Factorx
SDO-Subindex:	0
Datenlänge:	1 Byte
Zugriff:	Read_Only
Kodierung:	1
Erlaubte PDO-Abbildung:	Nicht abbildbar
Bemerkung:	Eine Wertänderung wird sofort aktiv und nichtflüchtig gespeichert.

FactorX.Numerator	Einfach Variable (Var)	**5FC0h + X**

Bedeutung:	Zählerwert von Factorx
SDO-Subindex:	1
Datenlänge:	4 Byte
Zugriff:	Read_Write
Kodierung:	1
Standardeinstellung:	1
Erlaubte PDO-Abbildung:	Nicht abbildbar
Bemerkung:	Eine Wertänderung wird sofort aktiv und nichtflüchtig gespeichert.

FactorX.Divisor	Einfach Variable (Var)	**5FC0h + X**

Bedeutung:	Nennerwert von Factorx
SDO-Subindex:	2
Datenlänge:	4 Byte
Zugriff:	Read_Write
Kodierung:	1
Standardeinstellung:	1
Erlaubte PDO-Abbildung:	Nicht abbildbar
Bemerkung:	Eine Wertänderung wird sofort aktiv und nichtflüchtig gespeichert.

13.1 Weitergehende Umrechnungen

Für einige Parameter reicht die Umrechnung über einen Faktor bestehend aus Zähler und Nenner nicht aus. Diese Umrechnungen beziehen noch Referenzwerte der FU-Steuerung für die Geschwindigkeit mit ein. Dies ist z.B. erforderlich bei der Umrechnung einer Beschleunigung(Delta Geschwindigkeit/Delta Zeit) in eine Rampenzeit. Die Referenzwerte sind dazu noch abhängig vom F5-Steuerungstyp(siehe auch Beschreibung des Parameters UD.02 in der Applikationsanleitung der Umrichter-Steuerung). Die folgende Liste von Referenzwerten ist im CAN-Operator abgelegt:

Ud.02-Wert	Geschwindigkeit-Referenzwert(VRef)	Normierung
0	100	Hz
1	200	Hz
2	400	Hz
3	reserviert	reserviert
4	1000	min^{-1}
5	2000	min^{-1}
6	4000	min^{-1}
7	reserviert	reserviert
8	1000	min^{-1}
9	2000	min^{-1}
10	4000	min^{-1}

13.2 Beispiel für die Bestimmung der Faktoren

Generell gilt für die optimale Bestimmung der Faktoren-Werte folgendes:

- Begrenzen Sie den Zähler und Nenner der Faktoren, wenn möglich, auf 16-Bit-Breite
- Der laufzeitintensivste Teil der Berechnung ist die Division durch FactorX.Divisor. Deshalb ist dieser Wert nach Möglichkeit = 1 zu setzen.

13.2.1 Factor0: Anwender-Weg-Einheiten in Inkremente

Für diese Umrechnung müssen die Werte der Parameter EC.01 (Geberstrichzahl Geber1) bzw. EC.11 (Geberstrichzahl Geber2) und EC.07 (Vielfachauswertung Geber1) bzw. EC.17 (Vielfachauswertung Geber2) bekannt sein. Hier wird von Geber1 ausgegangen.

Wenn der Factor0 bestimmt werden soll für eine Vorgabe in µm (lineare Bewegung), dann gilt folgendes:

$$F0 = \frac{EC.01 * 2^{EC.07}}{U_{Treibscheibe}} \quad \text{mit } U_{Treibscheibe} = \text{Umfang der Treibscheibe in µm}$$

--> Factor0.Numerator = $EC.01 * 2^{EC.07}$
--> Factor0.Divisor = $U_{Treibscheibe}$, mit $U_{Treibscheibe}$ = Umfang der Treibscheibe in µm

Wenn der Factor0 für die Vorgabe in 0,01 Winkelgrad (Drehbewegung) bestimmt werden soll, gilt folgendes:

$$F0 = \frac{EC.01 * 2^{EC.07}}{36000}$$

--> Factor0.Numerator = $EC.01 * 2^{EC.07}$
--> Factor0.Divisor = 36000

13.2.2 Factor1: Anwender-Geschwindigkeit-Einheiten in 0,125 rpm

Wenn der Factor1 für die Vorgabe-Auflösung 0,1 rpm bestimmt werden soll, gilt folgendes:

--> Factor1.Numerator = 8
--> Factor1.Divisor = 10

Wenn der Factor1 für die Vorgabe 1 µm/s bestimmt werden soll, gilt folgendes:

--> Factor1.Numerator = 480
--> Factor1.Divisor = $U_{Treibscheibe}$, mit $U_{Treibscheibe}$ = Umfang der Treibscheibe in µm

Wenn der Factor1 für die Vorgabe 0,01 Winkelgrad/s bestimmt werden soll, gilt folgendes:

--> Factor1.Numerator = 8
--> Factor1.Divisor = 600

13.2.3 Factor2: Anwender-Beschleunigung-Einheiten in eine KEB-Rampenzeit

Die Bestimmung des Factor2 ist etwas schwieriger. Beachten Sie, dass bei KEB die Beschleunigungs-/Verzögerungs-Parameter als Rampenzeit definiert sind. Deshalb ist die Umrechnung in diesem Fall umfangreicher. Hier soll der Faktor am Beispiel des Parameters HM_Homing_Acc gezeigt werden, der auf den Parameter PS.20 in der F5-Steuerung abgebildet wird:

Zur Vereinfachung kann man den Factor2 zunächst als ein Element schreiben. Dann sieht die Berechnungsformel wie folgt aus:

$$PS.20 = \frac{Vref}{HM\_Homing\_Acc} * Factor\ 2$$

Umgestellt nach Factor2:

$$Factor2 = \frac{PS.20 * HM\_Homing\_Acc}{Vref}$$

Angenommen:

- UD.02 = 4: F5-M, Maximaldrehzahl = 4000 min^{-1} , Vref = 1000 min^{-1}
- Der Parameter HM_Homing_Acc soll in min^{-2} aufgelöst sein

Der Wert PS.20 = 100 bedeutet eine Beschleunigung von 1000 min^{-1} pro Sekunde. Dies entspricht einem Wert von 60000 min^{-2}. Wenn man in die obige Gleichung jetzt für PS.20 den Wert 100 und für HM_Homing_Acc den Wert 60000 einsetzt, ergibt sich für Factor2:

$$Factor2 = \frac{100 * 60000}{1000} = 6000$$

--> Factor2.Numerator = 6000
--> Factor2.Divisor = 1

14. Anhang

14.1 CAN-Bit-Timing

Die KEB-CAN-Anschaltung(en) halten sich bzgl. des eingestellten Bit-Timings an die Vorgaben des CiA-Standards [2]:
Das nominale Bit-Timing sieht wie folgt aus:

Für alle einstellbaren Baudraten gilt:

- t_q: Basiszeiteinheit. Alle Segmente des Bit-Timing ergeben sich als Vielfaches dieser Zeiteinheit.
- SYNC: = 0 ==> Nur die Flanken von Rezessiv zu Dominant werden zur Synchronisation genutzt.
- SJW: = 0 ==> Synchronisations-Sprungweite = 1 * t_q
- TSEG2: = 1 ==> t_{SEG2} = 2 * t_q

Baudrate	Time-Quantum (t_q)	Sample-Point	TSEG1
10 Kbit/s	6.25 µs	14 * t_q = 87.5 µs	t_{SEG1} = 13 * t_q
20 Kbit/s	3.125 µs	14 * t_q = 43.75 µs	t_{SEG1} = 13 * t_q
25 Kbit/s	2,5 µs	14 * t_q = 35,0 µs	t_{SEG1} = 13 * t_q
50 Kbit/s	1.25 µs	14 * t_q = 17.5 µs	t_{SEG1} = 13 * t_q
100 Kbit/s	625 ns	14 * t_q = 8.75 µs	t_{SEG1} = 13 * t_q
125 Kbit/s	500 ns	14 * t_q = 7.0 µs	t_{SEG1} = 13 * t_q
250 Kbit/s	250 ns	14 * t_q = 3.5 µs	t_{SEG1} = 13 * t_q
500 Kbit/s	125 ns	13 * t_q = 1.625 µs	t_{SEG1} = 12 * t_q
800 Kbit/s	125 ns	7 * t_q = 1,25 µs	t_{SEG1} = 6 * t_q
1000 Kbit/s	125 ns	5 * t_q = 625 ns	t_{SEG1} = 4 * t_q

Die in der Tabelle grau hinterlegten Übertragungsgeschwindigkeiten sind bezüglich der möglichen Leitungslänge als besonders kritisch anzusehen. Außerdem weicht das Bit-Timing für diese Bitraten leicht von dem von [2] empfohlenen ab.

14.1.1 Wichtiger Warnhinweis

Die KEB-CAN-Anschaltung besitzt eine potentialgetrennte CAN-Schnittstelle. Durch die zusätzlichen Verzögerungsglieder (Optokoppler) im Signalverlauf verringert sich die mögliche Leitungslänge oder die mögliche Übertragungsgeschwindigkeit. Welche Leitungslänge bzw. Übertragungsgeschwindigkeit möglich ist, hängt von den Verzögerungszeiten aller im CAN-Netzwerk betriebenen Teilnehmer ab. Es obliegt dem Kunden eine Abschätzung vorzunehmen, bei welcher Bitrate welche Leitungslängen möglich sind. Die dazu notwendigen Angaben für die KEB-CAN-Anschaltung werden unten aufgeführt:

Sende-Verzögerungszeit des CAN-Treibers ..... : ≤80 ns.
Empfangs-Verzögerungszeit des CAN-Treibers ..... : ≤70 ns.
Sende-Verzögerungszeit der eingesetzten Optokoppler ..... : ≤40 ns.
Empfangs-Verzögerungszeit der eingesetzten Optokoppler ..... : ≤40 ns.

Es sollte immer die kleinste CAN-Übertragungsgeschwindigkeit gewählt werden, die von der Abarbeitung des Prozesses her erforderlich ist.

14.2 Literaturverzeichnis

[1]: Betriebsanleitung Frequenzumrichtersteuerung KEB COMBIVERT F5 mit Applikationsanleitung.

[2]: Dokument zur Vereinbarung des Arbeitskreises Physical-Layer der **C**AN **i**n **A**utomation (CiA) Nutzergruppe : CiA/DS 102-1. Herausgeber: CiA International Users and Manufacturers Group e.V., Am Weichselgarten 26, D-91058 Erlangen. Dokumente zur Vereinbarung des Arbeitskreises Higher-Layer-Protocols der CiA (Herausgeber s.o.):

[3]: CiA/WG2/DS201 : CAN in the OSI Reference Model

[4]: CiA/WG2/DS202-1 : CMS Service Specification

[5]: CiA/WG2/DS202-2 : CMS Protocol Specification

[6]: CiA/WG2/DS202-3 : CMS Encoding Rules

[7]: CiA/WG2/DS203-1 : NMT Service Specification

[8]: CiA/WG2/DS203-2 : NMT Protocol Specification

[9]: CiA/WG2/DS204-1 : DBT Service Specification

[10]: CiA/WG2/DS204-2 : DBT Protocol Specification

[11]: CiA/WG2/DS207 : Application Layer Naming Conventions

[12]: CiA/DS301 V.4.01 : Application Layer and Communication Profile vom 01.06.2000

[13]: CiA/DSP402 V.2.0 : Device Profile Drives and Motion Control

14.3 Übersicht der Operator-Parameter nach CANopen

Index	Name	Objekt-Typ	Subindex	Datenlänge in Byte	Zugriff
1000h	device type	VAR	0	4	ro
1001h	error register	VAR	0	1	ro
1002h	Manufacturer Status Register	VAR	0	4	ro
1003h	Pre-defined ErrorField	ARRAY	1 - max. 5	4	rw
1006h	ComCycle	VAR	0	4	rw
1008h	Manufacturer Device Name	VAR	0	4	ro
1018h	Identify Object	RECORD			
1400h	1st receive PDO Parameter	RECORD			
1400h	Number of supported entries	VAR	0	1	ro
1400h	COB-ID	VAR	1	4	rw
1400h	transmission type	VAR	2	1	rw
1401h	2nd receive PDO Parameter	RECORD			
1401h	Number of supported entries	VAR	0	1	ro
1401h	COB-ID	VAR	1	4	rw
1401h	transmission type	VAR	2	1	rw
1600h	1st receive PDO Mapping	RECORD			
1600h	Number of mapped objects	VAR	0	1	rw
1600h	nth object to be mapped	VAR	1 - max. 4	4	rw
1601h	2nd receive PDO Mapping	RECORD			
1601h	Number of mapped objects	VAR	0	1	rw
1601h	nth object to be mapped	VAR	1 - max. 4	4	rw
1800h	1st transmit PDO Parameter	RECORD			
1800h	Number of supported entries	VAR	0	1	ro
1800h	COB-ID	VAR	1	4	rw
1800h	transmission type	VAR	2	1	rw
1800h	Inhibit time	VAR	3	2	rw
1801h	2nd transmit PDO Parameter	RECORD			
1801h	Number of supported entries	VAR	0	1	ro
1801h	COB-ID	VAR	1	4	rw
1801h	transmission type	VAR	2	1	rw
1801h	Inhibit time	VAR	3	2	rw

Index	Name	Objekt-Typ	Subindex	Datenlänge in Byte	Zugriff
100Ah	Manufacturer Software Version	VAR	0	4	ro
100Ch	Guard Time	VAR	0	2	rw
100Dh	Life Time Factor	VAR	0	1	rw
1A00h	1st transmit PDO Mapping	RECORD			
1A00h	Number of mapped objects	VAR	0	1	rw
1A00h	nth object to be mapped	VAR	1 - max. 4	4	rw
1A01h	2nd transmit PDO Mapping	RECORD			
1A01h	Number of mapped objects	VAR	0	1	rw
1A01h	nth object to be mapped	VAR	1 - max. 4	4	rw
5FDAh	Watchdog_Activation	VAR	0	1	rw
5FDBh	VL_Ramp_CalcMode	VAR	0	1	rw
5FDCh	Save_VL_Ramps	VAR	0	1	rw
5FDDh	EmergencyCycle	VAR	0	2	rw
5FDEh	LifeGuardTout.Data	VAR	0	4	rw
5FDFh	LifeGuardTout.Addr	VAR	0	4	rw
5FE0h	HS_SyncToutDelay	VAR	0	2	rw
5FE2h	PD_Stored	VAR	0	1	rw
5FE3h	OP_NodeId	VAR	0	1	rw
5FE4h	PDOUT_WrMode	VAR	0	1	rw
5FE5h	HS_PDO_Index	VAR	0	1	rw
5FE6h	PDIN1_Cycle_Time	VAR	0	2	rw
5FE7h	PDIN2_Cycle_Time	VAR	0	2	rw
5FF9h	Watchdog_Inhibit	VAR	0	1	rw
5FECh	CAN_Baud2	VAR	0	1	rw
5FFEh	SAVE_CAN_Baud	VAR	0	1	rw
5FFFh	CAN_Baud	VAR	0	1	rw

14.4 Kompakt-Übersicht der CAN-Kommunikation

Feste Identifiervergabe:

SDO(rx)-Identifier	=	1536 + Node_Id	: SDO-Anforderung zum KEB-FU
SDO(tx)-Identifier	=	1408 + Node_Id	: SDO-Bestätigung vom KEB-FU
PDO1(rx)-Identifier	=	512 + Node_Id	: Prozessdaten zum KEB-FU
PDO1(tx)-Identifier	=	384 + Node_Id	: Prozessdaten vom KEB-FU
PDO2(rx)-Identifier	=	768 + Node_Id	: Prozessdaten zum KEB-FU
PDO2(tx)-Identifier	=	640 + Node_Id	: Prozessdaten vom KEB-FU
Node-Guarding-Identifier	=	1792 + Node_Id	
Emergency-Identifier	=	128 + Node_Id	: Emergency Message vom KEB-FU

Die wichtigsten NMT-Kommandos (Telegramme) auf Identifier = 0:

Start_Remote_Node

01h	Node-Id
B0	B1

Enter_Pre_Operational_State

80h	Node-Id
B0	B1

Reset_Node

81h	Node-Id
B0	B1

Die wichtigsten Werte des Node-State:

PRE_ OPERATIONAL = 7Fh : Kommunikation aktiv bis auf die PDO's
OPERATIONAL = 05h : Kommunikation komplett aktiv

Beispiele für SDO-Kommunkation:

Lesen des Parameters *Digitale Sollfrequenzvorgabe* (op.03)
in Satz 4 ==> Index = 2303h, Subindex = 10h

SDO(rx)-Telegramm (zum KEB-FU)

40h	03h	23h	10h	XXh	XXh	XXh	XXh
B0	B1	B2	B3	B4	B5	B6	B7

SDO(tx)-Telegramm (vom KEB-FU)

43h	03h	23h	10h	E8h	03h	00h	00h
B0	B1	B2	B3	B4	B5	B6	B7

In diesem Beispiel ist der gelesene Wert = 1000 (03E8h)

Schreiben Wert = 5 auf den Parameter *Sollwertquelle* (op.00)
in Satz 0 ==> Index = 2300h, Subindex = 01h

SDO(rx)-Telegramm (zum KEB-FU)

22h	00h	23h	01h	05h	00h	00h	00h
B0	B1	B2	B3	B4	B5	B6	B7

SDO(tx)-Telegramm (vom KEB-FU)

60h	00h	23h	01h	05h	00h	00h	00h
B0	B1	B2	B3	B4	B5	B6	B7

Beispiel für die Vorgabe von neuen Prozessdaten mit dem PDO1(rx)-Telegramm:

Vorausgesetzt ist hier die Standard-Prozessdatenbelegung.

Der Parameter *Steuerwort* (SY.50) soll den Wert = 1 erhalten,
der Parameter *Solldrehzahl* (SY.52) den Wert = 1500 (05DCh)

PDO1(rx)-Telegramm (zum KEB-FU)

01h	00h	DCh	05h

Karl E. Brinkmann GmbH
Försterweg 36-38 • **D**-32683 Barntrup
fon: +49 5263 401-0 • fax: +49 5263 401-116
net: www.keb.de • mail: info@keb.de

KEB Antriebstechnik GmbH & Co. KG
Wildbacher Str. 5 • **D**–08289 Schneeberg
fon: +49 3772 67-0 • fax: +49 3772 67-281
mail: info@keb-combidrive.de

KEB - YAMAKYU Ltd.
15–16, 2–Chome, Takanawa Minato-ku
J–Tokyo 108-0074
fon: +81 33 445-8515 • fax: +81 33 445-8215
mail: kebjt001@d4.dion.ne.jp

KEB Antriebstechnik Austria GmbH
Ritzstraße 8 • **A**-4614 Marchtrenk
fon: +43 7243 53586-0 • fax: +43 7243 53586-21
Kostelni 32/1226 • **CZ**-370 04 Ceské Budejovice
fon: +420 38 7319223 • fax: +420 38 7330697
mail: info@keb.at

KEB Antriebstechnik
Leidsevaart 126 • **NL**–2013 HD Haarlem
fon: +31 23 5320049 • fax: +31 23 5322260
mobil: +31 653964667
mail: vb.nederland@keb.de

KEB Antriebstechnik
Herenveld 2 • **B**-9500 Geraadsbergen
fon: +32 5443 7860 • fax: +32 5443 7898
mail: koen.detaeye@keb.de

KEB Portugal
Lugar de Salgueiros – Pavilhao A, Mouquim
P-4760 V. N. de Famalicao
fon: +351 252 371318 • fax: +351 252 371320
mail: keb.portugal@netc.pt

KEB China
Xianxia Road 299 • **CHN**-200051 Shanghai
fon: +86 21 62350922 • fax: +86 21 62350015
net: www.keb-cn.com • mail: info@keb-cn.com

KEB Taiwan Ltd.
1F, No.19-5, Shi Chou Rd., Tounan Town
R.O.C.-Yin-Lin Hsian / Taiwan
fon: +886 5 5964242 • fax: +886 5 5964240
mail: keb_taiwan@mail.apol.com.tw

Société Française KEB
Z.I. de la Croix St. Nicolas • 14, rue Gustave Eiffel
F-94510 LA QUEUE EN BRIE
fon: +33 1 49620101 • fax: +33 1 45767495
mail: sfkeb.4@wanadoo.fr

KEB Sverige
Box 265, Bergavägen 19
S-430 93 Hälsö
fon: +46 31 961520 • fax: +46 31 961935
mail: thomas.crona@keb.de

KEB (UK) Ltd.
6 Chieftain Buisiness Park, Morris Close
Park Farm, Wellingborough **GB**-Northants, NN8 6 XF
fon: +44 1933 402220 • fax: +44 1933 400724
net: www.keb-uk.co.uk • mail: info@keb-uk.co.uk

KEBCO Inc.
1335 Mendota Heights Road
USA-Mendota Heights, MN 55120
fon: +1 651 4546162 • fax: +1 651 4546198
net: www.kebco.com • mail: info@kebco.com

KEB Italia S.r.l.
Via Newton, 2 • **I**-20019 Settimo Milanese (Milano)
fon: +39 02 33500782 • fax: +39 02 33500790
net: www.keb.it • mail: kebitalia@keb.it

CC.F5.0D0-K002 12/2003