

Handbuch

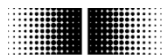
Stellantrieb *FlexiDrive* MSBA mit CANopen

Software Revision (Firmware Version) 2.00.xx



Inhalt

	Seite
1	Einleitung 4
1.1	Produktzuordnung 4
1.2	Funktionsübersicht..... 4
1.3	Sicherheits- und Betriebshinweise 5
2	Kommunikation über CANopen 6
2.1	Allgemeine Informationen 6
2.2	Einzelheiten über CANopen 7
2.2.1	Kommunikationsprofil 7
2.2.2	Meldungsaufbau 8
2.2.3	Servicedaten-Kommunikation (SDO) 8
2.2.4	Receive Prozessdaten-Kommunikation 10
2.2.5	Transmit Prozessdaten-Kommunikation..... 10
2.2.6	Emergency-Dienst 14
2.2.7	Netzwerkmanagement und NMT-Zustandsdiagramm..... 14
2.2.8	Node Guarding und Life Guarding..... 16
2.2.9	Heartbeat Protokoll 17
2.2.10	Layer Setting Services (LSS) 17
2.3	Zustandsmaschine für Positioning / Homing 21
2.4	Betrieb des Antriebs 25
2.4.1	Antriebsstatus auslesen 25
2.4.2	Aktuelle Position auslesen..... 25
2.4.3	Zielposition anfahren über SDO (mit Fahrauftrag) 25
2.4.4	Fahrauftrag stoppen 26
2.4.5	Ziel anfahren über R_PDO (mit Fahrauftrag) 26
2.5	Befehlssynchronisation zwischen Antrieb und Steuerung..... 27
3	Objektverzeichnis 29
4	Inbetriebnahme 44
4.1	Node-ID, Baudrate und Abschlusswiderstand einstellen 44
4.2	Objekte schreiben und lesen 46
4.3	Antriebsparameter konfigurieren 47
4.3.1	Notation, Dimension und Getriebe 47
4.3.2	Drehrichtung (Polarity)..... 50
4.3.3	Strom und Geschwindigkeit..... 50
4.3.4	I ² t-Begrenzung 50
4.3.5	Strombegrenzung 51
4.3.6	Positioning Timeout 52
4.3.7	Spielausgleich..... 52
4.3.8	Einstellungen speichern 52
4.4	Antriebsposition referenzieren (Homing)..... 52
4.4.1	Position oder Referenzposition direkt setzen 54
4.4.2	Referenzieren auf Referenzschalter..... 55
4.4.3	Referenzieren auf Anschlag 56
4.4.4	Referenzieren mit Einmessmethode 56
4.4.5	Zurücksetzen der Statusbits HomingOK und CalibrationOK..... 57
4.5	Positionen der Software-Endschalter setzen 57
4.6	Fahraufträge ausführen 58
4.7	Reglerabschaltung nach Positionierung..... 59
4.8	Automatische Deblockierung..... 60



5	Fehlerdiagnose	62
5.1	Feldbus-Kommunikation	62
5.2	LED Statusanzeige	62
5.3	Fehlerdiagnose am Antrieb	62
5.4	Automatische Fehler-Quittierung	64
6	EMV-gerechte Verdrahtung	65
7	Technische Daten	68
7.1	Elektrische und mechanische Daten	68
7.2	Anschlussbelegung	68
7.3	Blockschaltbild und Schalteingänge	68
7.4	Abmessungen	68
7.5	Befehlssequenzen als Beispiel	69

Haftungsausschluss

Diese Schrift wurde mit grosser Sorgfalt zusammengestellt. Fehler lassen sich jedoch nicht immer vollständig ausschliessen. Baumer übernimmt daher keine Garantien irgendwelcher Art für die in dieser Schrift zusammengestellten Informationen. In keinem Fall haftet Baumer oder der Autor für irgendwelche direkten oder indirekten Schäden, die aus der Anwendung dieser Informationen folgen.

Wir freuen uns jederzeit über Anregungen, die der Verbesserung dieses Handbuchs dienen können.

Created by:
Baumer IVO GmbH & Co. KG
Villingen-Schwenningen, Germany

1 Einleitung

1.1 Produktzuordnung

Tabelle 1: Produktzuordnung.

Produktfamilie	Produkt	Produktcode	EDS Datei	Schnittstelle
<i>FlexiDrive</i>	MSBA	102h	MSBA_xx.C.eds	CANopen

1.2 Funktionsübersicht

- Intelligenter Positioniermotor als dezentraler Teilnehmer am CAN-Bus
- Absolut-Multiturn Positionserfassung (3 Bit Singleturn, 29 Bit Multiturn)
- 2Q-Regler (Generatorbetrieb möglich)
- Getriebekonstruktionen mit Planeten-, Stirnrad oder Schneckengetriebe
- 2 Digitaleingänge für Jogging, Hardware-Endschalter oder Referenzierung
 - ansteuerbar über Standardsensor
 - verschiedene Funktionen parametrierbar (Objekt 2111h-0Dh)
- Betriebsarten (Objekt 6060h):
 - Positionieren und Referenzieren
- Sonstige Funktionen
 - Software-Endschalter zur Begrenzung des Fahrwegs (Objekt 607Dh)
 - Hardware-Endschalter (Objekt 2111h-0Dh)
 - Zielposition setzen (Objekt 607Ah)
 - Halteregele
 - Stillstandsüberwachung (Objekt 6041h)
 - Spielausgleich gegen Getriebe- oder Spindelspiel (Objekt 2111h-01h)
 - Drehmoment- / Strombegrenzung (Objekt 6073h)
 - Linear- oder Rundachse (Objekt 608Ah)
 - Drehrichtung umkehrbar (Objekt 607Eh)
 - Stand-Alone-Betrieb / Tippbetrieb ohne CAN-Bus und Parametrierung (Objekt 2111h-13h)
 - Komfortable Referenzierungsfunktionen (Objekt 6098h)
 - Wert direkt setzen (Objekt 2111h-11h)
 - Fahren auf Anschlag
 - Einmessmethode
 - Status- und Fehlerdiagnose über CAN-Bus (Objekt 1003h und 603Fh)
 - Temperaturüberwachung mit Warnmeldung und Abschaltung im Fehlerfall (Objekt 2114h)
 - Über- und Unterspannungsüberwachung
 - Einschaltstrombegrenzung
 - Bootloader
 - LED Statusanzeige
 - Deblockierfunktion zuschaltbar
 - I²t-Begrenzung
 - Einstellbares Reglerverhalten nach Zielerreichung
- Kommunikation über CANopen
 - SDO-Kommunikation (Parameter lesen / schreiben)
 - PDO-Kommunikation (Ansteuerung, Senden von Status und Position)
 - Galvanische Abtrennung der Feldbus-Elektronik für optimale Störungsunempfindlichkeit
- Einfache Inbetriebnahme
 - Anpassung weniger Parameter
 - Feldbusanschluss durchschleifbar
 - Parametrier- und Inbetriebnahmetool CANmaster (siehe CD-ROM Art. Nr. 10147362)
- Einstellschalter für Knotennummer, Baudrate und Abschlusswiderstand. Erreichbar durch abnehmbaren Abschlussdeckel.
- Nichtflüchtiger Speicher (EEPROM) für Default- und Anwenderdaten (Objekt 2300h).

1.3 Sicherheits- und Betriebshinweise

Zusätzliche Informationen

- Das Handbuch ist eine Ergänzung zu bereits vorhandenen Dokumentationen (Kataloge, Produktinformationen und Montageanleitungen).
- Die Anleitung muss unbedingt vor Inbetriebnahme gelesen werden.

Bestimmungsgemässer Gebrauch

- Der Antrieb ist ein Präzisionsgerät. Er dient ausschliesslich zur automatischen Verstellung von mechanischen Anschlägen und Spindeleinrichtungen.

Inbetriebnahme

- Einbau und Montage des Antriebs darf ausschliesslich durch eine Fachkraft erfolgen.
- Betriebsanleitung des Maschinenherstellers beachten.

Sicherheitshinweise

- Vor Inbetriebnahme der Anlage alle elektrischen Verbindungen überprüfen.
- Wenn Montage, elektrischer Anschluss oder sonstige Arbeiten am Antrieb und an der Anlage nicht fachgerecht ausgeführt werden, kann es zu Fehlfunktion oder Ausfall des Antriebs führen.
- Eine Gefährdung von Personen, eine Beschädigung der Anlage und eine Beschädigung von Betriebseinrichtungen durch den Ausfall oder Fehlfunktion des Antriebs muss durch geeignete Sicherheitsmassnahmen ausgeschlossen werden.
- Antrieb darf nicht ausserhalb der Grenzwerte betrieben werden (siehe weitere Dokumentationen).

Bei Nichtbeachtung kann es zu Fehlfunktionen, Sach- und Personenschäden kommen!

Transport und Lagerung

- Transport und Lagerung ausschliesslich in Originalverpackung.
- Antrieb nicht fallen lassen oder grösseren Erschütterungen aussetzen.

Montage

- Schläge oder Schocks auf Gehäuse und Welle vermeiden.
- Gehäuse nicht verspannen.
- Antrieb nicht öffnen oder mechanisch verändern.

Welle, Kugellager oder elektronische Teile können beschädigt werden. Die sichere Funktion ist dann nicht mehr gewährleistet.

Elektrische Inbetriebnahme

- Antrieb nicht elektrisch verändern.
- Keine Verdrahtungsarbeiten unter Spannung vornehmen.
- Der elektrische Anschluss darf unter Spannung nicht aufgesteckt oder abgenommen werden.
- Die gesamte Anlage EMV gerecht installieren. Einbaumgebung und Verkabelung beeinflussen die EMV des Antriebs. Antrieb und Zuleitungen räumlich getrennt oder in grossem Abstand zu Leitungen mit hohem Störpegel (Frequenzumrichter, Schütze usw.) verlegen.
- Bei Verbrauchern mit hohen Störpegeln separate Spannungsversorgung für den Antrieb bereitstellen.
- Antriebsgehäuse und die Anschlusskabel vollständig schirmen.
- Antrieb an Schutzerde (PE) anschliessen. Geschirmte Kabel verwenden. Schirmgeflecht muss mit dem Stecker verbunden sein. Anzustreben ist ein beidseitiger Anschluss an Schutzerde (PE), Gehäuse über den mechanischen Anbau, Kabelschirm über die nachfolgenden angeschlossenen Geräte. Bei Problemen mit Erdschleifen mindestens eine einseitige Erdung.

Bei Nichtbeachtung kann es zu Fehlfunktionen, Sach- und Personenschäden kommen!

2 Kommunikation über CANopen

2.1 Allgemeine Informationen

Der CAN-Bus (CAN: Controller Area Network) wurde ursprünglich von Bosch und Intel für die schnelle und kostengünstige Datenübertragung im Kraftfahrzeug entwickelt. Heute ist der CAN-Bus einer der am weitesten verbreiteten Feldbusse in der industriellen Automatisierung. Über ihn kommunizieren Geräte, Aktoren und Sensoren verschiedener Hersteller miteinander. Normen für den CAN-Bus werden durch die Nutzer-Vereinigung „CAN in Automation“ (CiA) festgelegt.

CAN-Bus-Eigenschaften

- Datenrate von 1 MBit/s bei einer Netzausdehnung bis 25 m
- Beidseitig abgeschlossenes Netzwerk
- Busmedium ist ein Twisted-Pair-Kabel
- Echtzeitfähigkeit: Definierte maximale Wartezeit für Nachrichten hoher Priorität.
- Bis zu 127 Teilnehmer an einem Bus (physikalisch treiberbedingt maximal 32 Teilnehmer)
- Sicherstellung netzweiter Datenkonsistenz. Gestörte Nachrichten werden für alle Netzknoten als fehlerhaft bekannt gemacht.
- Nachrichtenorientierte Kommunikation
- Jede Nachricht wird mit einer Nachrichtenkennung (Identifizier) gekennzeichnet. Alle Netzknoten prüfen anhand des Identifiers, ob die Nachricht für sie relevant ist.
- Broadcasting: Alle Netzknoten erhalten gleichzeitig jede Nachricht. Daher ist eine Synchronisation möglich.
- Multi-Master-Fähigkeit: Jeder Teilnehmer im Feldbus kann selbständig Daten senden und empfangen, ohne dabei auf eine Priorität der Master angewiesen zu sein. Jeder kann seine Nachricht beginnen, wenn der Bus nicht belegt ist. Bei einem gleichzeitigen Senden von Nachrichten setzt sich der Teilnehmer mit der höchsten Priorität durch.
- Priorisierung von Nachrichten: Der Identifizier setzt die Priorität der Nachricht fest. Dadurch können wichtige Nachrichten schnell über den Bus übertragen werden.
- Restfehlerwahrscheinlichkeit: Sicherungsverfahren im Netzwerk reduzieren die Wahrscheinlichkeit einer unentdeckten, fehlerhaften Datenübertragung auf unter 10^{-11} . Praktisch kann von einer 100% sicheren Übertragung ausgegangen werden.
- Funktionsüberwachung: Lokalisation fehlerhafter oder ausgefallener Stationen. Das CAN-Protokoll beinhaltet eine Funktionsüberwachung von Netzknoten. Netzknoten, die fehlerhaft sind, werden in ihrer Funktion eingeschränkt oder ganz vom Netzwerk abgekoppelt.
- Datenübertragung mit kurzer Fehlererholzeit: Durch mehrere Fehlererkennungsmechanismen werden verfälschte Nachrichten mit grosser Wahrscheinlichkeit erkannt. Wird ein Fehler erkannt, so wird die Nachrichtensendung automatisch wiederholt.

Im CAN-Bus sind mehrere Netzwerkteilnehmer über ein Buskabel miteinander verbunden. Jeder Netzwerkteilnehmer kann Nachrichten senden und empfangen. Die Daten zwischen den Netzwerk-Teilnehmern werden seriell übertragen.

Beispiele für CAN-Bus-Geräte als Netzwerkteilnehmer sind:

- Automatisierungsgeräte wie SPS oder PC
- Ein- / Ausgangsmodule
- Antriebssteuerungen
- Analysegeräte, z. B. ein CAN-Monitor
- Bedien- und Eingabegeräte
- Sensoren und Aktoren



2.2 Einzelheiten über CANopen

Unter technischer Leitung des Steinbeis Transferzentrum für Automatisierung wurde auf der Basis der Schicht 7 Spezifikation CAL (CAN Application Layer) das CANopen-Profil entwickelt. Im Vergleich zu CAL sind in CANopen nur die für diesen Einsatz geeigneten Funktionen enthalten. CANopen stellt somit eine für die Anwendung optimierte Teilmenge von CAL dar und ermöglicht dadurch einen vereinfachten Systemaufbau und den Einsatz vereinfachter Geräte. CANopen ist optimiert für den schnellen Datenaustausch in Echtzeitsystemen. Die CAN in Automation (CiA) ist zuständig für die geltenden Normen der entsprechenden Profile.

CANopen ermöglicht:

- einfachen Zugriff auf alle Geräte- und Kommunikationsparameter
- Synchronisation von mehreren Geräten
- automatische Konfiguration des Netzwerkes
- zyklischen und ereignisgesteuerten Prozessdatenverkehr

CANopen besteht aus vier Kommunikationsobjekten COB (communication objects) mit unterschiedlichen Charakteristiken:

- Prozessdaten-Objekte PDO (Process Data Object) zur Echtzeitübertragung von Prozessdaten
- Servicedaten-Objekte SDO (Service Data Object) für die Übermittlung von Parametern und Programmen
- Network Management (NMT, Lifeguarding) zur Initialisierung und Netzwerksteuerung
- Vordefinierte Objekte für die Synchronisation und Fehleranzeige

Alle CANopen Geräte- und Kommunikationsparameter sind in einem Objektverzeichnis gegliedert. Ein CANopen Objekt umfasst Objekt-Nummer mit Subindices, Objekt-Format sowie Struktur und Inhalt der Parameter. Gemäss CiA ist dieses Objektverzeichnis in drei verschiedene Teile unterteilt:

- Kommunikationsprofil
- Geräteprofil
- Herstellerspezifischen Teil

Für Einzelheiten zu den Objekten dieses Antriebs, siehe Kapitel Objektverzeichnis.

2.2.1 Kommunikationsprofil

Die CANopen Kommunikation zwischen Netzwerkteilnehmern und Master erfolgt über Objekte. Diese Objekte werden mit einem 16 Bit-Index adressiert. Im CANopen Kommunikationsprofil DS 301 sind die verschiedenen Kommunikationsobjekte (COB) standardisiert:

- Prozessdaten-Objekte PDO (Process Data Object) zur Echtzeitübertragung von Prozessdaten
- Servicedaten-Objekte SDO (Service Data Object) für den Schreib- und Lesezugriff auf das Objektverzeichnis
- Objekte zur Synchronisation und Fehleranzeige von CAN-Teilnehmern:
 - SYNC-Objekt (Synchronisation Object) zur Synchronisation von Netzwerkteilnehmern
 - EMCY-Objekt (Emergency Object) zur Fehleranzeige eines Gerätes oder seiner Peripherie
- Network Management (Network Management NMT, Lifeguarding) zur Initialisierung und Netzwerksteuerung
- Layer Setting Services (LSS) zur Konfiguration über Seriennummer oder Revisionsnummer

2.2.2 Meldungsaufbau

Der erste Teil einer Meldung ist die so genannte COB-ID (Identifier).

Aufbau der 11-Bit COB-ID:

Funktionscode	Node-ID
4 Bit Function code	7 Bit Node-ID

Der Funktionscode gibt Aufschluss über die Art der Meldung und die Priorität. Je niedriger der Funktionscode, desto höher die Priorität der Meldung.

Broadcast-Meldungen:

Funktionscode	COB-ID
NMT	0
SYNC	80h

Peer-to-Peer-Meldungen:

Funktionscode	COB-ID
EMERGENCY	80h + Node-ID
T_PDO1 (tx) ¹	180h + Node-ID
R_PDO1 (rx) ¹	200h + Node-ID
T_PDO2 (tx) ¹	280h + Node-ID
R_PDO2 (rx) ¹	300h + Node-ID
SDO (tx) ¹	580h + Node-ID
SDO (rx) ¹	600h + Node-ID
Nodeguard, Heartbeat	700h + Node-ID
LSS (tx) ¹	7E4h
LSS (rx) ¹	7E5h

¹(tx) und (rx) aus der Sicht des Antriebs.

Für Einstellung der Node-ID (Knotennummer des Busteilnehmers) siehe Kapitel Inbetriebnahme.

Ein CAN-Telegramm besteht aus der COB-ID und bis zu 8 Datenbyte. Die genauen Telegramme werden im Folgenden ausführlich erläutert.

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
xxx	x	xx	xx	xx	xx	xx	xx	xx	xx

Der DLC (Data Length Code) bezeichnet die Länge des Telegramms.

2.2.3 Servicedaten-Kommunikation (SDO)

Die Servicedaten-Objekte (SDO) entsprechen dem Protokoll DSP-402. Über Index und Subindex kann auf ein Objekt zugegriffen werden. Die Daten können angefordert oder ins Objekt geschrieben werden.

Ein SDO-Telegramm setzt sich zusammen aus einem Kommando-Byte, zwei Objekt-Bytes, einem Subindex-Byte sowie bis zu vier Datenbytes und ist folgendermassen aufgebaut:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
--------	-----	----------	----------	----------	----------	--------	--------	--------	--------

Die COB-ID eines SDO setzt sich folgendermassen zusammen:

- Kommunikation Master zu Antrieb: 600h + Node-ID
- Kommunikation Antrieb zu Master: 580h + Node-ID

Tabelle 2: Beschreibung von SDO-Kommandos. Das Kommando-Byte legt fest, ob Daten gelesen oder gesetzt werden und um wie viele Datenbyte es sich handelt.

SDO Kommando	Beschreibung	Datenlänge	Beschreibung
22h	Download Request	Max. 4 Byte	Parameter an Antrieb senden
23h	Download Request	4 Byte	Parameter an Antrieb senden
2Bh	Download Request	2 Byte	Parameter an Antrieb senden
2Fh	Download Request	1 Byte	Parameter an Antrieb senden
...			
60h	Download Response	-	Bestätigung der Übernahme an Master
40h	Upload Request	-	Parameter vom Antrieb anfordern
...			
42h	Upload Response	Max. 4 Byte	Parameter an Master senden mit max. 4 Byte
43h	Upload Response	4 Byte	Parameter an Master senden
4Bh	Upload Response	2 Byte	Parameter an Master senden
4Fh	Upload Response	1 Byte	Parameter an Master senden
...			
80h	Abort Message	-	Antrieb meldet Abort Code an Master

Eine SDO Abort Message zeigt einen Fehler in der CAN-Kommunikation an. Das zugehörige SDO Kommando-Byte ist 80h. Objekt und Subindex sind die des gewünschten Objektes. Byte 4...7 ergeben zusammen den SDO Abort Code (Byte 7 = MSB).

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
580h+Node-ID	8	80h	Objekt L	Objekt H	Subindex	ErrByte 0	ErrByte 1	ErrByte 2	ErrByte 3

Für Beschreibung der SDO Abort Codes, siehe Kapitel 5 Fehlerdiagnose.

Beispiele für eine SDO-Kommunikation

Anfrage eines Wertes vom Master beim Antrieb. Eine häufige verwendete Anfrage ist beispielsweise die Statusanfrage (siehe Objekt 6041h):

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	40h	41h	60h	0	x	x	x	x

Antwort des Antriebs auf die Anfrage eines Wertes. Der Status ist 4 Byte lang (siehe Objekt 6041h):

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	43h	41h	60h	0	a	b	c	d

Schreiben eines Werts vom Master zum Antrieb. Mit dem Kontrollwort wird der Antrieb gesteuert (siehe Objekt 6040h):

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	22h	40h	60h	0	a	b	x	x

Antwort des Antriebs auf das Schreiben des Werts:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

2.2.4 Receive Prozessdaten-Kommunikation

Receive Prozessdatenobjekte (R_PDO) dienen dem Echtzeit-Datenaustausch von Prozessdaten zum Antrieb. Übermittelt wird beispielsweise ein neues Ziel oder das Kontrollwort. R_PDO können synchron oder asynchron vom Antrieb empfangen werden. Der Antrieb unterstützt das statische R_PDO Mapping mit R_PDO1 (Controlword) und R_PDO2 (Controlword und Target position).

Synchrone Kommunikation

Im synchronen Betrieb werden die Prozessdaten mittels R_PDO vom Antrieb zeitgleich mit dem Eintreffen des SYNC-Telegramms übernommen. Dazu muss im Objekt 1400-2h bzw. 1401-2h ein Wert zwischen 01h und F0h eingeschrieben werden. Mit Wert = 3 wird das R_PDO auf jedes dritte SYNC-Telegramm empfangen. SYNC-Telegramm:

COB-ID	DLC
80h	0

Asynchrone Kommunikation

Sollen die R_PDO sofort nach Empfang übernommen werden, muss ins Objekt 1400-2h bzw. 1401-2h der Wert FEh geschrieben werden. Beispiele für Einstellungen des Objektes R_PDO1 type

1400h-02h	Beschreibung
FEh	R_PDO werden sofort vom Antrieb übernommen
01h	R_PDO werden erst nach dem nächsten SYNC vom Antrieb übernommen.

R_PDO1 (Controlword)

Telegrammaufbau:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
200h+Node-ID	2	xx	xx	-	-	-	-	-	-

COB-ID : 200h + Node-ID

DLC : Länge 2 Datenbyte

Byte 0..1 : neues Kontrollwort (Objekt 6040h Controlword)

R_PDO2 (Controlword und Target position)

Telegrammaufbau:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
300h+Node-ID	6	xx	xx	xx	xx	xx	xx	-	-

COB-ID : 300h + Node-ID

DLC : Länge 6 Datenbyte

Byte 0..1 : neues Kontrollwort (Objekt 6040h Controlword)

Byte 5..2 : neues Ziel (Objekt Target 607Ah position)

2.2.5 Transmit Prozessdaten-Kommunikation

Transmit Prozessdatenobjekte (T_PDO) dienen dem Echtzeit-Datenaustausch von Prozessdaten vom Antrieb zum Master, beispielsweise die aktuelle Position des Antriebs oder das Statuswort. T_PDO können synchron oder zyklisch (asynchron) gesendet werden.

Synchrone Kommunikation

Im synchronen Betrieb werden die T_PDO vom Master über das SYNC-Telegramm angefordert. Um die Prozessdaten synchron zu senden, muss im Objekt 1800-2h bzw. 1801-2h ein Wert zwischen 1 und F0h eingeschrieben werden. Mit Wert = 3 wird das T_PDO auf jedes dritte SYNC-Telegramm gesendet. Im Objekt 2800h bzw. 2801h ist definiert, ob nur bei Änderung gesendet wird.

SYNC-Telegramm:

COB-ID	DLC
80h	0

Asynchrone Kommunikation

Sollen T_PDO zyklisch gesendet werden, muss ins Objekt 1800-02h bzw. 1801-02h der Wert FEh geschrieben werden. Im Objekt 1800-05h bzw. 1801-05h steht die Zykluszeit in Millisekunden. Die kleinstmögliche Zykluszeit beträgt 1ms. Wird der Wert 0ms gespeichert, werden keine T_PDO gesendet. Die Funktion ist dann ausgeschaltet. Im Weiteren kann im Objekt 2800h bzw. 2801h bestimmt werden, ob das Senden des T_PDO nur auf Änderung erfolgen soll: Beträgt der Wert 0, läuft das zyklische Senden wie oben beschrieben. Beträgt der Wert 1, wird zyklisch geprüft, ob eine Änderung der zu sendenden Werte vorliegt. Falls ja, werden die aktuellen Daten gesendet. Wenn keine Änderung vorliegt, wird nicht gesendet. Beträgt der Wert 4, wird bei Änderung das T_PDO viermal gesendet im vordefinierten Zyklus. Alternativ kann in Objekt 1800-02h bzw. 1801-02h der Kommunikationstyp FFh eingestellt werden. Hierbei wird das T_PDO umgehend nach einer Änderung der Daten gesendet. Zusätzlich kann im Objekt 1800-05h bzw. 1801-05h eine zyklische Übertragung konfiguriert werden, die auch dann stattfindet, wenn sich die Daten nicht ändern. Um zu verhindern, dass bei vielen unmittelbar aufeinanderfolgenden Änderungen der Bus blockiert wird, empfiehlt sich die Konfiguration einer Sendepause in Objekt 1800-03h bzw. 1801-03h. Diese wird nach dem Senden des T_PDO mindestens gewartet, bis es erneut gesendet wird.

Tabelle 3: Beispiele für Einstellungen der T_PDO Kommunikation.

1800h bzw. 1801			2800h bzw. 2801h	Beschreibung
Subindex 02h	Subindex 03h	Subindex 05h		
FEh	xxx	10 ms	0h	Zyklisches Senden alle 10ms
FEh	xxx	10 ms	1h	Alle 10ms wird das T_PDO ein Mal gesendet, falls eine Änderung vorliegt.
FEh	xxx	10 ms	2h	Alle 10ms wird das T_PDO doppelt gesendet, falls eine Änderung vorliegt.
FEh	xxx	0 ms	x	T_PDO senden ausgeschaltet
FFh	5 ms (32h)	0 ms (0h)	xxx	T_PDO wird bei Änderung der Daten umgehend gesendet. Nach der Kommunikation wird das erneute Senden des T_PDO für 5 ms gesperrt.
FFh	5 m (32h)	10 ms (0Ah)	xxx	T_PDO wird bei Änderung der Daten umgehend gesendet. Wenn keine Änderung vorliegt, alle 10 ms. Nach der Kommunikation wird das erneute Senden des T_PDO für 5 ms gesperrt.
03h	xxx	xxx	00h	Bei jedem dritten SYNC-Telegramm senden
03h	xxx	xxx	2Bh	Auf jedes dritte SYNC-Telegramm, aber insgesamt nur 43 mal (=2Bh).

T_PDO1 (Statuswort)

Telegrammaufbau:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
180h+Node-ID	x	Xx	xx	xx	-	xx	xx	xx	xx

COB-ID : 180h + Node-ID

DLC : Länge 2 (bzw. 3 oder 8) Datenbyte (einstellbar im Objekt 2110h Version control)

Byte 0...1 : Statuswort

Byte 2 : Statuswort-Zusatz (falls in Objekt 2110h Version control eingestellt)

Byte 3 : enthält nie Information

Byte 4...7 : Anzeige der digitalen Hardware-Schalteingänge (falls in Objekt 2110h Version control eingestellt)



Tabelle 4: Statuswort in Byte 0...1 des T_PDO1-Telegramms (Objekt 6041h).

Bit	Beschreibung
0	Ready to switch on
1	SwitchOn enabled
2	Operation enabled
3	Error active
4	Voltage enabled
5	QuickStop active
6	SwitchOn disabled
7	Warning active
8	CalibrationOK
9	Not used (permanently 1)
10	Target position reached
11	Internal software limit switch active
12	Drive moving
13	Not used (permanently 0)
14	HomingOK
15	External hardware limit switch active

Tabelle 5: Statuswort-Zusatz in Byte 2 des T_PDO1-Telegramms für Anzeige des internen Antriebszustands. Statuswort-Zusatz ist standardmässig aktiviert, die Länge des T_PDO1-Telegramms ist dann 3 Byte. Deaktivieren von Statuswort-Zusatz erfolgt in Versionskontrolle Objekt 2110h (Bit 1).

Bit	Byte2	Beschreibung
SELF_TEST	00	Selbsttest nach Initialisierung
READY	01	Nach erfolgreichem Selbsttest -> Ready
POSITIONING_READY	02	SwitchOn, EnableVoltage, QuickStop, EnableOperation und Positioning -> PositioningReady
HOMING_READY	03	SwitchOn, EnableVoltage, QuickStop, EnableOperation und Homing -> HomingReady
POSITIONING_MOVING	04	Einschalten des Antriebs (Bit4 des Kontrollworts) -> PositioningMove
POSITIONING_FIRSTTARGET	05	Bei indirektem Anfahren der Zielposition, um Spindelspiel aufzuheben
POSITIONING_ENDTARGET	06	Zielposition setzen
HOMING_MOVING1	07	
HOMING_MOVING2	08	
HOMING_SET	09	
HOMING_STEP1	10 (0Ah)	Manuell fahren
HOMING_STEP2	11 (0Bh)	Manuell fahren
HOMING_STEP3	12 (0Ch)	Manuell fahren
HOMING_STEP4	13 (0Dh)	Manuell fahren
QUICK_STOP_ACTIVE_1	15 (0Fh)	Motor stoppen; Bit [0:3] des Kontrollworts bewirkt QuickStop, → Zielposition verwerfen
QUICK_STOP_ACTIVE_2	16 (10h)	Position lesen
QUICK_STOP_ACTIVE_3	17 (11h)	Zielposition = aktuelle Position; Zielposition verwerfen
HALT_ACTIVE	18 (12h)	Bit [8] des Kontrollworts bewirkt Zwischenhalt, Zielposition wird beibehalten
ERROR_DIAGNOSTIC	19 (13h)	Aus jedem Zustand wird im Fehlerfall auf ErrorDiagnostic gewechselt
COMU	20 (14h)	Eigener Zustand für interne Kommunikation
WAIT_TIME_UNBLOCKING	29 (1Dh)	Zustand zwischen Blockierung und Freifahren
MOVE_UNBLOCKING	30 (1Eh)	Fahre um den Weg s (Free referencing distance, Objekt 2111-1Eh)

Tabelle 6: Statuswort-Zusatz in Byte 4...7 des T_PDO1-Telegramms für Anzeige der digitalen Hardware-Schalteingänge. Aktivieren erfolgt in Objekt 2110h Version control (Bit 12). Die Gesamtlänge des T_PDO1 Telegramms beträgt dann 8 Byte. Byte 3 enthält nie Informationen.

Bit	Beschreibung
0..15	Nicht definiert
16	Monitorbit für Digital input 1
17	Monitorbit für Digital input 2
18..31	Nicht definiert

T_PDO2 (Position)

Telegrammaufbau:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
280h+Node-ID	x	Xx	xx	xx	xx	xx	xx	xx	xx

COB-ID : 280h + Node-ID

DLC : Länge 4 oder 8 Datenbyte (aktivierbar in Objekt 2110h Version control)

Byte 0...3 : Aktuelle Position Objekt 6064h Position actual value in benutzerdefinierter Einheit (z.B. Mikrometer)

Byte 4.. 7 : enthält Strom (Objekt 6078h Current actual value) oder Geschwindigkeit (Objekt 606Ch Velocity actual value), wenn in Objekt 2110h Version control Bit 13 bzw. 14 aktiviert ist.

2.2.6 Emergency-Dienst

Interne Fehler im Antrieb oder Busfehler lösen eine Emergency-Meldung aus (EMCY-Meldung):

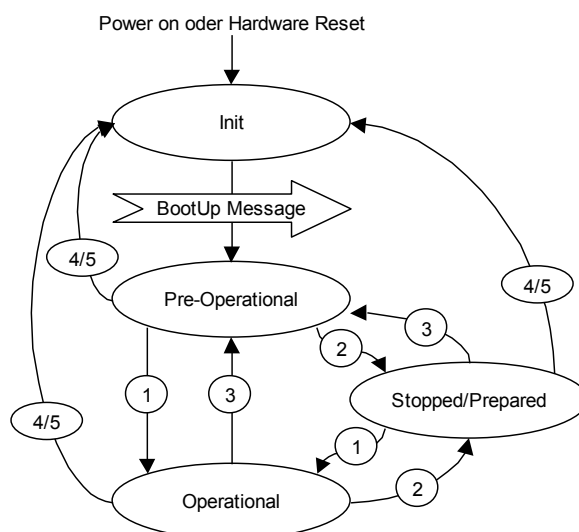
COB-ID	DLC	Byte0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80h+Node-ID	8	Error Code	Error Register 1001h	Error Code Zusatz Information			Not used	Not used	Not used

Für Beschreibung der Error Register und Error Code Definitionen, siehe Kapitel 5 Fehlerdiagnose.

2.2.7 Netzwerkmanagement und NMT-Zustandsdiagramm

Nachdem der Antrieb initialisiert wurde, befindet er sich im NMT-Zustand Pre-Operational (siehe Figur 1). In diesem Zustand können SDO gelesen und geschrieben werden. Um die PDO-Kommunikation zu starten, muss das NMT-Kommando Start Remote Node gesendet werden. Dann befindet sich der Antrieb im Operational. Nun werden die gewünschten PDO gesendet. Zusätzlich können SDO gelesen und geschrieben werden.

Nachdem der Antrieb mit dem NMT-Kommando Stop Remote Node gestoppt wird, befindet sich der Antrieb im NMT-Zustand Stopped/Prepared. Es können keine SDO- oder PDO-Parameter mehr gelesen oder gesetzt werden. In diesem Zustand ist nur noch NMT-Kommunikation, LSS, Heartbeat oder Node Guarding möglich. Durch das NMT-Kommando Reset Remote Node wird der Antrieb erneut initialisiert und befindet sich danach wieder im Zustand Pre-Operational.



Figur 1: NMT-Zustandsdiagramm.

Mit den Diensten des Netzwerkmanagement (NMT) können die Busteilnehmer initialisiert, gestartet und gestoppt werden. Zusätzlich existieren NMT-Dienste zur Verbindungsüberwachung.

Beschreibung der NMT-Kommandos

Alle NMT-Kommandos werden als unbestätigte Objekte übertragen und sind folgendermassen aufgebaut:

Byte 0	Byte 1	Byte 2
COB-ID = 0	Kommando-Byte	Node-ID

Die COB-ID für NMT-Kommandos ist immer Null. NMT-Kommandos haben damit maximale Priorität. Die Node-ID des gewünschten Teilnehmers wird in Byte 2 des NMT-Kommandos übertragen. Mit Node-ID = 0 werden alle Teilnehmer angesprochen (Broadcast Meldung).

Tabelle 7: NMT-Kommando-Byte.

Kommando-Byte	Beschreibung	Sprung im NMT Zustandsdiagramm (siehe Figur 1)
01h	Start Remote Node	1
02h	Stop Remote Node	2
80h	Enter Pre-Operational	3
81h, 82h	Reset Remote Node	4, 5

NMT-Zustandswechsel

Init

Nach dem Initialisieren meldet sich der Antrieb mit einer BootUp Meldung auf dem CAN-Bus. Danach wechselt der Antrieb automatisch in den Zustand Pre-Operational.

Die COB-ID der BootUp Meldung setzt sich zusammen aus 700h und der Node-ID:

COB-ID	Byte 0
700h + Node-ID	00h

Start Remote Node (1)

Mit dem Startbefehl wird der Antrieb in den Zustand Operational gebracht.

COB-ID	Kommando Byte	Node-ID
0	1h	0..127

Stop Remote Node (2)

Mit dem Stopbefehl wird der Antrieb in den Zustand Stopped/Prepared gebracht.

COB-ID	Kommando Byte	Node-ID
0	2h	0..127

Enter Pre-Operational (3)

Wechsle in den Zustand Pre-Operational.

COB-ID	Kommando Byte	Node-ID
0	80h	0..127

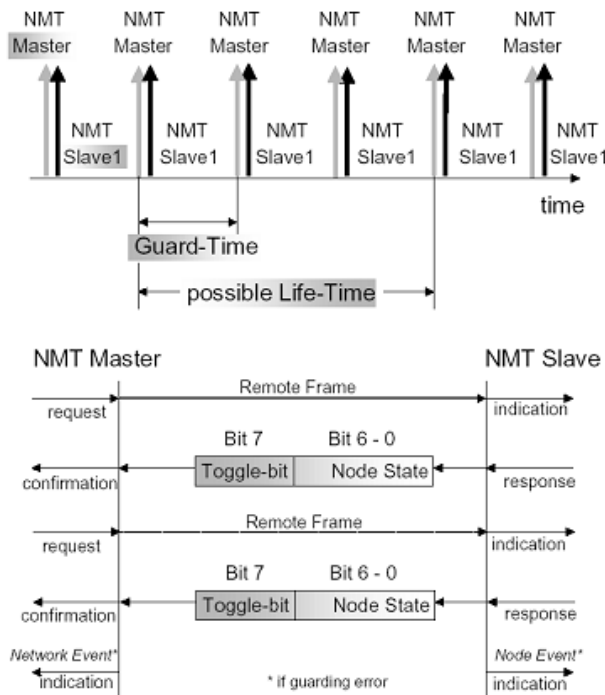
Reset Remote Node (4) oder Reset Kommunikation (5)

Mit dem Resetbefehl wird der Antrieb neu initialisiert.

COB-ID	Kommando Byte	Node-ID
0	81h	0..127

2.2.8 Node Guarding und Life Guarding

Das Nodeguarding Protokoll dient zur Überwachung von NMT-Slaves durch den NMT-Master. Node Guarding wird eingeschaltet durch Setzen von Bit5 im Objekt 2110h, Version control. Der NMT-Slave erscheint dann in einer Datenbank des NMT-Masters mit den NMT-Zuständen der NMT-Slaves. Mit dem Nodeguarding Protokoll wird überprüft, ob alle NMT-Slaves korrekt arbeiten oder ob sich beispielsweise ein NMT-Slave vom Bus zurückgezogen hat. Gleichzeitig kann jeder NMT-Slave überwachen, ob der NMT-Master noch aktiv ist.



Figur 2: Node Guarding und Life Guarding.

Der NMT-Master startet den Überwachungsdienst durch zyklisches Senden von Remote Frames an den NMT-Slave. Das Zeitintervall dazwischen ist die so genannte Guard Time (siehe Figur 2 und Objekt 100Ch). Sie kann unterschiedlich sein für jeden NMT-Slave.

Die Antwort des NMT-Slaves beinhaltet seinen jeweiligen NMT-Zustand. Auf diese Weise kann der NMT-Master überprüfen, ob sich der NMT-Slave im richtigen NMT-Zustand befindet und gegebenenfalls auf Fehler reagieren.

Die so genannte Knoten-Lebensdauer (Node life time) eines NMT-Slaves ist gegeben durch die Guard Time multipliziert mit dem Lifetime factor (Objekt 100Dh). Sie kann ebenfalls unterschiedlich sein für jeden NMT-Slave. Erhält ein NMT-Slave nicht innerhalb der Knoten-Lebensdauer einen Remote Frame des NMT-Masters, so wird ein Life Guarding Event ausgelöst. Das Verhalten im Fehlerfall wird definiert im Objekt Error behavior - Communication error (Objekt 1029h, Subindex 01h).

Gemäss CAN-Nutzerorganisation sollte zur Reduktion der Buslast anstatt Node Guarding das aktuellere Überwachungsprotokoll Heartbeat verwendet werden.

Beispiel eines Node Guarding Protokolls:

COB-ID	Data / Remote	Byte 0
700h+Node-ID	r	00h (0d)
700h+Node-ID	d	FFh (255d)
700h+Node-ID	r	00h (0d)
700h+Node-ID	d	7Fh (127d)

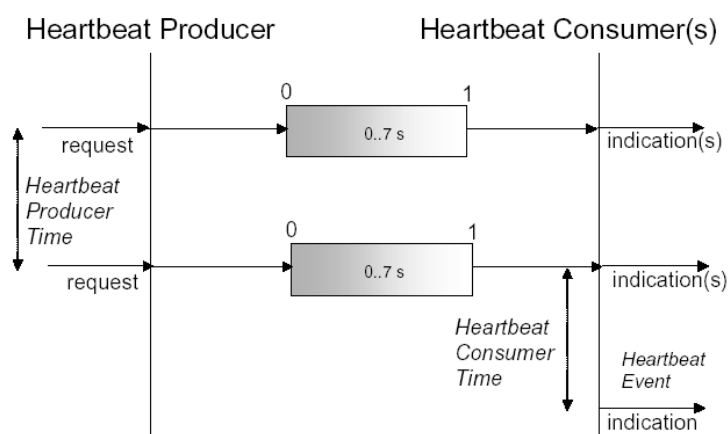
Mögliche Zustände der NMT-Teilnehmer:

- 0: BootUp-Event
- 4: Stopped/Prepared
- 5: Operational
- 127: Pre-Operational

Die unteren 7 Bits ergeben in obigem Beispiel 7Fh, d.h. der Antrieb befindet sich im Zustand Pre-Operational.

2.2.9 Heartbeat Protokoll

Für alle neuen Applikationen ist es dringend empfehlenswert, das modernere Überwachungsprotokoll Heartbeat zu verwenden. Heartbeat wird eingeschaltet im Objekt 2110h Version control durch Setzen von Bit5 auf Null. Ein Heartbeat Producer produziert zyklisch eine so genannte Heartbeat-Meldung (siehe Figur 3). Ein oder mehrere Heartbeat Consumer können diese Heartbeat Meldung empfangen. Falls das zyklische Senden dieser Heartbeat Meldung ausbleibt, wird ein so genannter Heartbeat Event ausgelöst. Das Verhalten im Fehlerfall wird definiert im Objekt Error behavior - Communication error (Objekt 1029h-01h). Beispielsweise kann der Slave (Antrieb) als ein Consumer definiert werden, der beim Ausbleiben der Producer Meldung der Masters (Steuerung) stoppt. Gleichzeitig überwacht der Master den Slave entweder ebenfalls über Heartbeat oder alternativ über seine SDO Antworten.



Figur 3: Heartbeat Protokoll.

Beispiel eines Heartbeat Protokolls:

COB-ID	Data/Remote	Byte 0
700h+Node-ID	d	7Fh (127d)

Die Heartbeat Meldungen bestehen aus der COB-ID und einem Byte, mit dem der NMT-Zustand übertragen wird:

- 0: BootUp Meldung
- 4: Stopped/Prepared
- 5: Operational
- 127: Pre-Operational (7Fh = 127).

2.2.10 Layer Setting Services (LSS)

Um mehrere Busteilnehmer mit gleicher Node-ID auf demselben Bussystem ansprechen zu können, wird LSS verwendet (siehe Vorgehen unter Layer Setting Services and Protocol, CiA Draft Standard Proposal 305).

Jeder Busteilnehmer mit LSS besitzt eine eindeutige Seriennummer, über die er angesprochen und initialisiert werden kann. Anschliessend können sowohl Node-ID als auch Baudrate neu gesetzt werden. LSS kann nur im Zustand Stopped/Prepared ausgeführt werden.

Meldungsaufbau

COB-ID

Master → Antrieb : 2021 = 7E5h

Master ← Antrieb : 2020 = 7E4h

Nach der COB-ID wird ein LSS Command Specifier gesandt. Danach folgen bis zu sieben Datenbyte:

COB-ID	Command Specifier	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
--------	-------------------	--------	--------	--------	--------	--------	--------	--------

Switch Mode Global

7E5h →	04h	Mode	reserviert
--------	-----	------	------------

Mode: : 0 → Operationsmode

1 → Konfigurationsmode

Switch Mode Selektiv

Mit folgendem Ablauf kann ein ganz bestimmter Busteilnehmer im System angesprochen werden.

7E5h →	40h	Vendor-ID	reserviert
--------	-----	-----------	------------

7E5h →	41h	Product code	reserviert
--------	-----	--------------	------------

7E5h →	42h	Revision number	reserviert
--------	-----	-----------------	------------

7E5h →	43h	Serial number	reserviert
--------	-----	---------------	------------

7E4h ←	44h	Mode	reserviert
--------	-----	------	------------

Vendor-ID : 5Fh

Product code : Interner Produktcode für den jeweiligen Busteilnehmer

Revision number : Aktuelle Revisionsnummer des Busteilnehmers

Serial number : Eindeutige, fortlaufende Seriennummer

Mode : Antwort des Busteilnehmers ist der neue Mode (0=Operating mode; 1=Konfigurationsmode)

Node-ID setzen

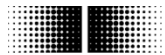
7E5h →	11h	Node-ID	reserviert
--------	-----	---------	------------

7E4h ←	11h	Error code	Specific error	reserviert
--------	-----	------------	----------------	------------

Node-ID : Neue Node-ID des Busteilnehmers

Error code : 0=OK; 1=Node-ID ausserhalb des Bereiches; 2..254=reserviert; 255→Specific error

Specific error : Falls ErrorCode=255 → Applikationsspezifischer Error code.



Baudrate setzen

7E5h →	13h	Table Sel	Table Ind	reserviert
--------	-----	-----------	-----------	------------

7E4h ←	13h	Error code	Specific error	reserviert
--------	-----	------------	----------------	------------

Table Sel : Selektiert die Baudraten-Tabelle (default = 0: Standard CiA Baudraten-Tabelle)

Table Ind : Baudraten-Index in Baudraten-Tabelle (siehe Tabelle 8).

Error code : 0=OK; 1= Baudrate ausserhalb des Bereiches; 2..254=reserviert; 255→Specific error

Specific error : Falls Error code=255 → Applikationsspezifischer Error code.

Tabelle 8: Standard CiA Baudraten.

Baudrate [kBit/s]	10	20	50	100	125	250	500	800	1000
Index (LSS) nach CiA-Tabelle	8h	7h	6h	5h	4h	3h	2h	1h	0h
Index (Definition in Objekt 2100h)	0h	1h	2h	3h	4h	5h	6h	7h	8h

Die Baudrate kann auch im Objekt 2100h geändert werden. Hierbei wird ebenfalls ein Index ins Objekt geschrieben, nicht die effektive Baudrate. ACHTUNG: Die Indices für Objekt 2100h und LSS unterscheiden sich.

Speichern des Konfigurationsprotokoll

Dieses Protokoll speichert die Konfigurationsparameter im EEPROM.

7E5h →	17h	reserviert
--------	-----	------------

7E4h ←	17h	Error code	Specific error	reserviert
--------	-----	------------	----------------	------------

Error Code : 0=OK;1=Speichern nicht unterstützt;2=Zugriffsfehler;3..254=reserviert;255→Specific error

Specific error : Falls ErrorCode=255 → Applikationsspezifischer Error code.

Aktiviere BitTiming Parameter

Die neuen BitTiming Parameter werden mit dem Command Specifier 21 aktiviert.

7E5h →	15h	16 Bit Switch Delay	reserviert
--------	-----	---------------------	------------

Switch Delay : Verzögerung des Antrieb Resets in ms.

Nach der Verzögerungszeit meldet sich der Busteilnehmer mit der neuen Baudrate an.

Vendor-ID anfordern

Vendor-ID eines selektierten Busteilnehmers anfordern

7E5h →	5Ah	reserviert
--------	-----	------------

7E4h ←	5Ah	32 Bit Vendor-ID	reserviert
--------	-----	------------------	------------

Vendor-ID : = 5Fh

Produktcode anfordern

Produktcode eines selektierten Busteilnehmers anfordern

7E5h →	5Bh	reserviert
--------	-----	------------

7E4h ←	5Bh	Product code	reserviert
--------	-----	--------------	------------

Product code : Herstellerabhängiger Produktcode

Revisionsnummer anfordern

Revisionsnummer eines selektierten Busteilnehmers anfordern

7E5h →	5Ch	reserviert
--------	-----	------------

7E4h ←	5Ch	32 Bit Revisionsnummer	reserviert
--------	-----	------------------------	------------

Revisionsnummer : aktuelle Revision

Seriennummer anfordern

Seriennummer eines selektierten Busteilnehmers anfordern

7E5h →	5Dh	reserviert
--------	-----	------------

7E4h ←	5Dh	32 Bit Serial number	reserviert
--------	-----	----------------------	------------

Seriennummer : eindeutige fortlaufende Seriennummer des Busteilnehmers

Bereichsanfrage

Busteilnehmer können auch in einem gewissen Bereich gesucht werden. Hierzu werden folgende Objekte nacheinander versandt:

7E5h →	46h	Vendor-ID	reserviert
--------	-----	-----------	------------

7E5h →	47h	Product code	reserviert
--------	-----	--------------	------------

7E5h →	48h	Revision number LOW	reserviert
--------	-----	---------------------	------------

7E5h →	49h	Revision number HIGH	reserviert
--------	-----	----------------------	------------

7E5h →	4Ah	Serial number LOW	reserviert
--------	-----	-------------------	------------

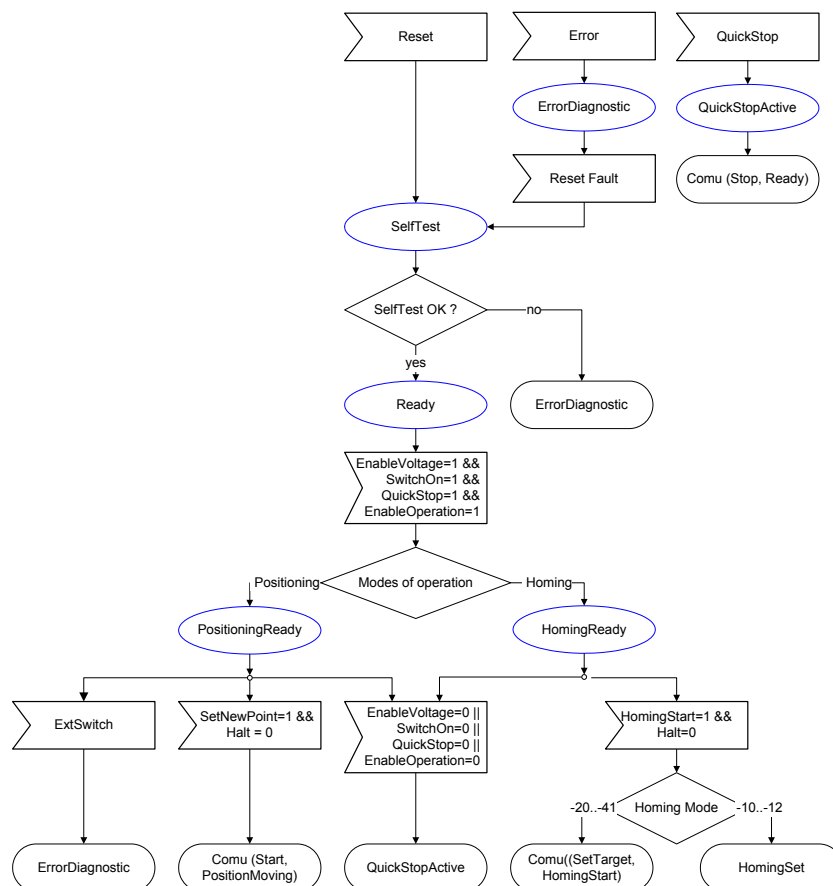
7E5h →	4Bh	Serial number HIGH	reserviert
--------	-----	--------------------	------------

Jeder Busteilnehmer mit den entsprechenden Parametern meldet sich mit folgender Meldung:

7E4h ←	4Fh	reserviert
--------	-----	------------

2.3 Zustandsmaschine für Positioning / Homing

Beim Starten des Antriebs wird zuerst das NMT Zustandsdiagramm aktiviert. In den NMT-Zuständen Pre-Operational oder Operational gilt folgendes Zustandsdiagramm für die Mode of operation Positioning und Homing (siehe Objekt 6060h).



Figur 4: Zustandsmaschine für Mode of operation Positioning und Homing.

Selftest

Nach einem Reset des Antriebs wird ein Selbsttest durchgeführt. Hier wird zuerst ein Reset an den Regler gesendet. Danach werden alle Parameter mit dem Regler ausgetauscht und auf Richtigkeit überprüft. Zudem werden Abhängigkeiten der einzelnen Werte geprüft. Nach erfolgreichem Test wird automatisch in den Zustand Ready gewechselt.

Ready

Im Zustand Ready können beliebige Parameter gesetzt und gelesen werden. Dieser Zustand ist der Grundzustand des Antriebes. Dieser Zustand wird erst verlassen, wenn vom Master dazu aufgefordert wird.

PositioningReady

Wechsel vom Zustand Ready erfolgt durch:

1. Objekt 6060h Modes of operation auf 1 = Positioning setzen
2. Objekt 6040h Kontrollwort auf 000Fh setzen
SwitchOn, EnableVoltage, QuickStop und EnableOperation = 1

Im Zustand PositioningReady wartet der Antrieb nur noch auf das Start-Signal, welches mit dem Kontrollwort gesetzt wird. QuickStop sowie Ansprechen der Schalteingänge werden in diesem Zustand speziell verarbeitet.



HomingReady (Manueller Mode)

Wechsel vom Zustand Ready erfolgt durch:

1. Objekt 6060h Modes of operation auf 6 = Homing setzen
2. Objekt 6040h Controlword auf 000Fh setzen
SwitchOn, EnableVoltage, QuickStop und EnableOperation = 1

In diesem Zustand wird mit dem Objekt 6098h Homing method die gewünschte Referenzierungsfunktion vorgewählt. Mit dem Kontrollwort, Bit4 (HomingOperationStart) wird in den Zustand HomingMove gewechselt.

QuickStop active

QuickStop ist kein eigentlicher Zustand. Falls in den QuickStop gewechselt wird, wird ein QuickStop an den Regler gesendet und in den Zustand Ready gewechselt. Falls ein Fehler anliegt, wird erst dann in den Zustand ErrorDiagnostic gewechselt.

ErrorDiagnostic

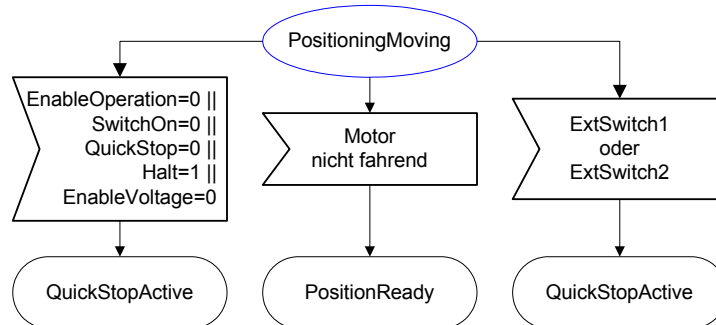
Im ErrorDiagnostic wird eine EMCY-Meldung abgesetzt (Error code siehe Tabelle 17) und auf eine Entscheidung vom Master gewartet. Parameter können auch in diesem Zustand beliebig gelesen und geschrieben werden. Nach einem Error reset (Kontrollwort) wird in den SelfTest gewechselt, wo auch der Regler einen Reset des Fehlers erhält.

PositioningMoving

Das Ziel ist im Regler immer aktuell. Deshalb wird nur noch der Startbefehl gesendet und in den Zustand PositioningMoving gewechselt.

Events:

1. Software-Stoppbefehl über CAN -> QuickStop an Regler senden
2. Motor nicht mehr fahrend -> Zielposition erreicht
3. Externer Hardware-Endschalter spricht an -> QuickStop an Regler senden



Figur 5: PositioningMoving

Homing

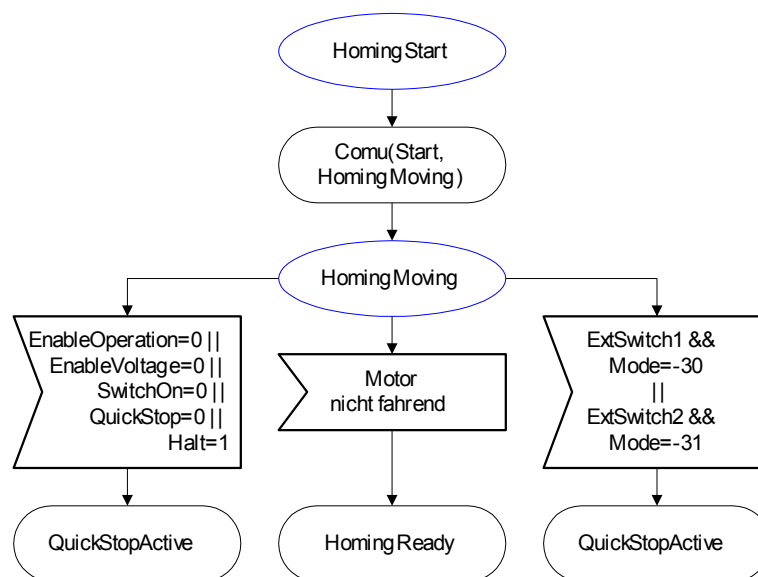
Homing wird in zwei verschiedene Vorgänge unterteilt. Im HomingMoving wird der Motor in Bewegung gesetzt. Im HomingSet hingegen werden nur Werte abgespeichert.

HomingMoving

Neues Ziel ist gesetzt. Danach wird der Startbefehl an den Regler gesendet und in den Zustand HomingMoving übergegangen.

Events:

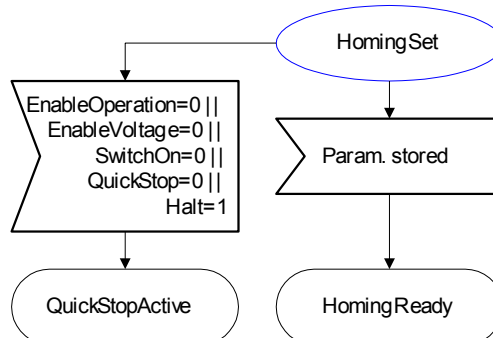
1. Software-Stoppbefehl über CAN -> QuickStop an Regler senden
2. Motor nicht mehr fahrend -> Ziel erreicht
3. Externe Schalter sprechen an -> QuickStop an Regler senden



Figur 6: HomingMoving.

HomingSet

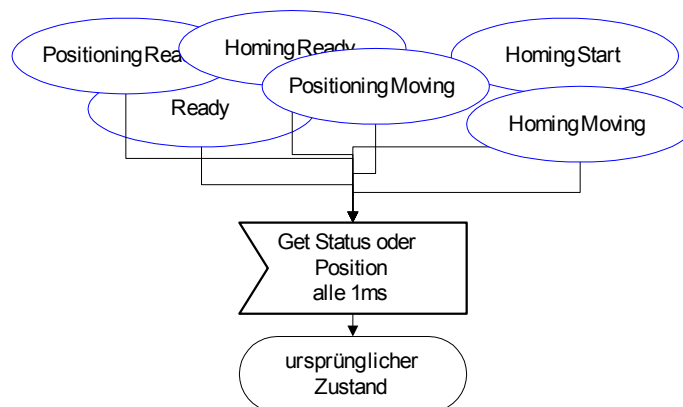
Im Zustand HomingSet werden nur Parameter wie z.B. Position der Software-Endschalter gesetzt.



Figur 7: HomingSet.

Status oder Position einlesen

In den meisten Zuständen werden abwechselnd alle 1ms Status oder aktuelle Position eingelesen.



Figur 8: Status oder aktuelle Position einlesen.

Kontrollwort

Zustandswechsel werden über das Kontrollwort gesteuert.

Ein Master sendet folgendes Kontrollwort um den Antrieb zu aktivieren: **xxxx xxx0 0xxx 1111**

Ein Master sendet folgendes Kontrollwort um den Antrieb zu starten: **xxxx xxx0 0xx1 1111**

Tabelle 9: Kontrollwort (Objekt 6040h). Bit 13...15 sind nicht definiert.

Bit	Beschreibung
0	SwitchOn
1	EnableVoltage
2	QuickStop
3	EnableOperation
4	StartAction (Fahre zum Ziel)
5	Change set immediately
6	Relative/absolute movement
7	Error reset
8	Halt
11	Jogging+
12	Jogging-

Beispiel für einen Zustandswechsel:

Aktion	Kontrollwort Bit					Beschreibung
	7	3	2	1	0	
SwitchOn	0	1	1	1	1	Ready → PositioningReady oder HomingReady (je nach Objekt 6060h)
DisableOperation	0	0	1	1	1	PositioningReady oder HomingReady → Ready
DisableVoltage	0	1	1	0	1	PositioningReady oder HomingReady → Ready
SwitchOff	0	1	1	1	0	PositioningReady oder HomingReady → Ready
QuickStop	0	1	0	1	1	Von PositioningReady oder HomingReady → QuickStop active Falls Motor drehend, wird dies speziell behandelt. Falls QuickStop wieder 1 gesetzt wird, kann direkt zu OperationEnabled gewechselt werden.

2.4 Betrieb des Antriebs

2.4.1 Antriebsstatus auslesen

Je nach NMT-Zustand kann der Status des Antriebs als SDO (Objekt 6041h) ausgelesen oder als T_PDO1 zyklisch bzw. synchron angefordert werden.

COB-ID	DLC	Byte 0	Byte 1	Byte 2
180h+Node-ID	3	xx	xx	xx

2.4.2 Aktuelle Position auslesen

Die effektive Position kann als SDO ausgelesen werden oder wird zyklisch oder synchron als T_PDO2 gesendet.

Anfrage der Position beim Antrieb → Objekt 6064h:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	40h	64h	60h	0	0	0	0	0

Antwort des Antriebs auf die Anfrage der Position (abcd):

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	43h	64h	60h	0	a	b	c	d

Üblicher und praktischer wäre die Aktualisierung der Position mittels Transmit Prozessdatenobjekte (T_PDO). T_PDO2 ist standardmässig auf das Objekt 6064h (Position) eingestellt. Hierbei kann zwischen synchroner und asynchroner T_PDO-Übertragung gewählt werden (siehe Objekt 1400h und folgende).

T_PDO2 Telegrammaufbau:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3
280h+Node-ID	4	ww	xx	yy	zz

Byte1 bis Byte4 beinhalten die Position.

2.4.3 Zielposition anfahren über SDO (mit Fahrauftrag)

Zielposition wird gesetzt (Objekt 607Ah):

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Ah	60h	0	x	x	x	x

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Ah	60h	0	0	0	0	0

Es wird ein Start ins Kontrollwort (Objekt 6040h) geschrieben:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	1Fh	0	0	0

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

Der Antrieb fährt los.

2.4.4 Fahrauftrag stoppen

Es wird ein Stop ins Kontrollwort geschrieben:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	1Fh	01h	0	0

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

Der Antrieb wird gestoppt.

2.4.5 Ziel anfahren über R_PDO (mit Fahrauftrag)

NMT Start:

COB-ID	DLC	Byte 0	Byte 1
0	2	01h	Node-ID

Antwort T_PDO1 (Statuswort Objekt 6041h):

COB-ID	DLC	Byte 0	Byte 1	Byte 2
180h+Node-ID	3	21h	0Ah	01h

Antwort T_PDO2 (Ist-Position Objekt 6064h):

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3
280h+Node-ID	4	Ww	Xx	Yy	Zz

Antrieb in den Zustand PositioningReady setzen mit dem R_PDO1 (Kontrollwort Objekt 6040h):

COB-ID	DLC	Byte 0	Byte 1
200h+Node-ID	2	0Fh	00h

Antwort T_PDO1 (Statuswort Objekt 6041h):

COB-ID	DLC	Byte 0	Byte 1	Byte 2
180h+Node-ID	3	37h	0Ah	02h

Kontrollwort + Ziel wird gesetzt mit dem R_PDO2 (Objekt 6040h, 607Ah):

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
300h+Node-ID	6	1Fh	00h	x	x	x	x

Wenn R_PDO2 auf synchron eingestellt ist, muss noch das SYNC-Telegramm gesendet werden:

COB-ID	DLC
80h	0

Der Antrieb fährt los.



2.5 Befehlssynchronisation zwischen Antrieb und Steuerung

Für die CAN-Bus Kommunikation gelten allgemein die Anforderungen:

- Die Busbelastung soll möglichst klein sein (zyklische Kommunikation sollte daher eher vermieden werden, mit Ausnahme des Nodeguarding).
- Ein Startbefehl soll möglichst schnell ausgeführt werden, um kurze Zykluszeiten zu erreichen.

Bei CANopen erfolgt der Fahrbefehl vom CAN-Bus Master an den Antrieb über ein Kontrollwort (R_PDO oder SDO Objekt 6040h). Für die Statusanzeige sieht CANopen ein T_PDO vor (Statuswort Objekt 6041h), in dem die Statusinformationen des Antriebs (Drive moving, Target position reached, etc.) zusammengefasst sind. Mit diesem Statuswort müssen bei einem Fahrbefehl Master und Antrieb synchronisiert werden. Um die Anforderungen nach minimaler Buslast und Zykluszeit zu erfüllen, sendet der Antrieb das Statuswort nur bei Änderungen.

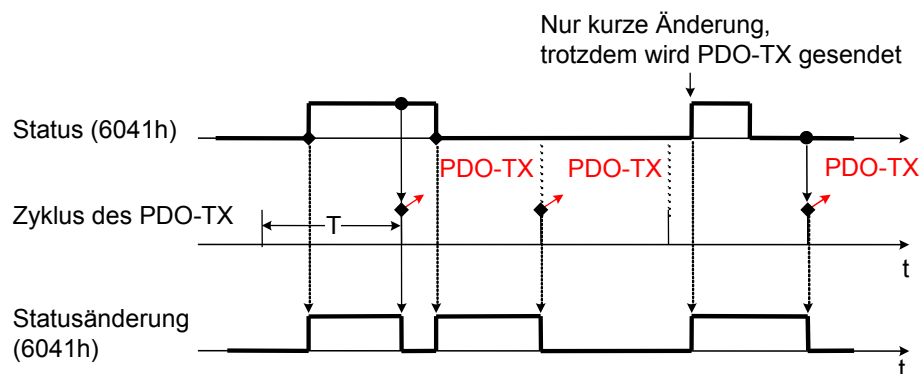
Zur Befehlssynchronisation ist im Antrieb folgendes implementiert:

- Nach jeder Statusänderung wird ein T_PDO gesendet.
- Jeder Fahrbefehl erzeugt eine Statusänderung.

Diese beiden Punkte werden im Folgenden erläutert. Zusätzlich muss bei einem Fahrbefehl auf einen korrekten und vollständigen Ablauf der Kommunikation zwischen Master und Antrieb geachtet werden.

Nach jeder Statusänderung wird ein T_PDO gesendet

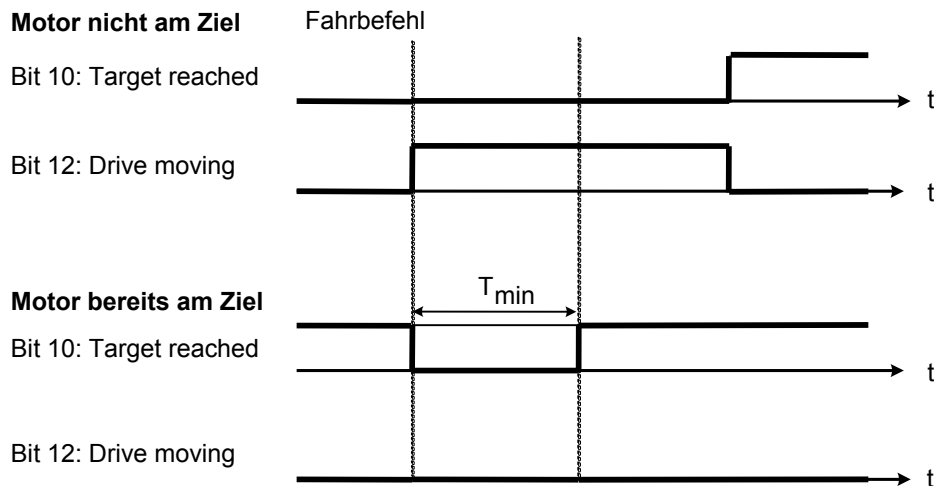
Der Antrieb hat einen einstellbaren Zyklus mit Zykluszeit T (Transmit PDO2 Event Timer, Objekt 1801h-05h) für das Senden eines T_PDO. Gesendet wird dabei normalerweise immer der Status (Objekt 6041h) zum Sendezeitpunkt. Der Antrieb muss jedoch auch dann ein T_PDO senden, wenn nur eine kurze Statusänderung innerhalb eines Zyklus stattfindet und diese Statusänderung noch vor dem nächsten Zyklus des T_PDO wieder verschwindet (siehe Figur 9). Zur Befehlssynchronisation ist dies im Antrieb implementiert. Nur wenn keine Änderung stattgefunden hat, wird kein T_PDO gesendet.



Figur 9: Nach jeder Statusänderung wird ein T_PDO gesendet.

Jeder Fahrbefehl erzeugt eine Statusänderung

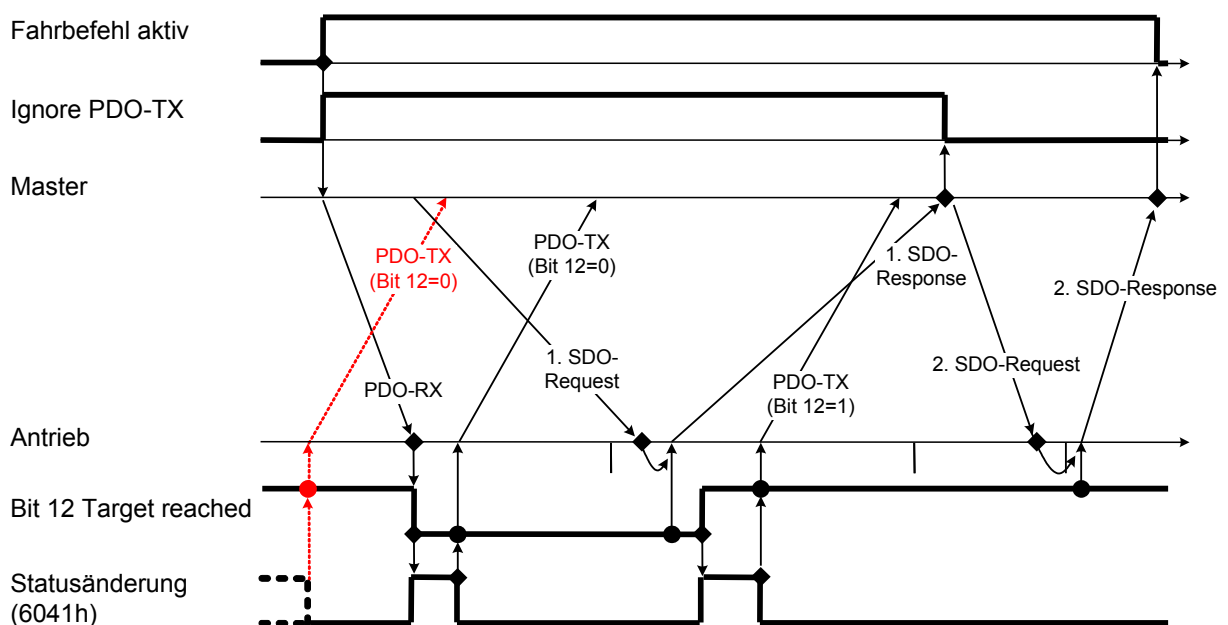
Sendet der Master einen Fahrbefehl an einen Antrieb mit einer Zielposition weit weg von der aktuellen Position, so ergeben sich klare Statusänderungen und das Statusbit 10 „Target position reached“ wird erst gesetzt beim Erreichen der Zielposition. Wenn sich jedoch der Antrieb bereits an der Zielposition befindet, dann würde normalerweise keine Statusänderung erzeugt werden. Dies könnte zu fehlerhafter Synchronisierung zwischen Master und Antrieb führen. Zur Befehlssynchronisation wird daher im Antrieb in einem solchen Fall das Statusbit 10 „Target position reached“ für eine Zeit T_{min} zwangsläufig auf Null gesetzt, nachdem ein Fahrbefehl empfangen wurde. Dies gilt auch beim Referenzieren und Einmessen, wo die entsprechenden Statusbits zunächst für T_{min} auf Null und beim Beenden wieder auf Eins gesetzt werden. Somit ist gewährleistet, dass bei allen Befehlen eine entsprechende Statusänderung erzeugt und ein entsprechender T_PDO gesendet werden.



Figur 10: Jeder Fahrbefehl erzeugt eine Statusänderung (auch beim Referenzieren und Einmessen).

Korrektter Kommunikationsablauf mit bis zu zwei SDO Abfragen des Statusworts 6041h

Eine undefinierte Situation könnte möglicherweise entstehen, wenn zufällig gleichzeitig mit dem Senden eines Fahrauftrags R_PDO durch den Master eine Statusänderung am Antrieb erfolgt und dieser somit ein T_PDO sendet. In diesem Fall empfängt der Master nach dem Absetzen eines Fahrbefehles einen Status, der für ihn jedoch aus der Vergangenheit kommt und den er auf keinen Fall beachten darf. Um diese unerwünschten T_PDO Meldungen nach einem Fahrauftrag zu unterdrücken, müssen gleichzeitig mit dem Senden des Fahrauftrags beim Master die T_PDO Meldungen gesperrt werden (Ignore T_PDO = 1). Um die Sperrung wieder zu aufzuheben, fragt der Master den Status über den 1. SDO-Request ab. Sobald er die 1. SDO Response empfängt, hebt er die Sperrung auf (Ignore T_PDO = 0). Dann werden empfangene T_PDO Meldungen wieder interpretiert. Wenn die 1. SDO-Response den Status „Target position reached = 1“ enthält, ist der Fahrbefehl korrekt abgeschlossen. Wenn ein nach Aufhebung der Sperrung empfangenes T_PDO den Status „Target position reached = 1“ enthält, ist der Fahrbefehl ebenfalls korrekt abgeschlossen. Es kann jedoch vorkommen, dass eine T_PDO Meldung aufgrund ihrer hohen Priorität die 1. SDO-Response überholt und daher unterdrückt wird (siehe Figur 11). Der Fahrbefehl kann in diesem Fall nur durch eine zweite SDO-Request mit anschließender SDO-Response korrekt abgeschlossen werden. Auch beim Referenzieren und Einmessen muss der Kommunikationsablauf unbedingt so realisiert werden.



Figur 11: Korrektter Kommunikationsablauf mit bis zu zwei SDO Abfragen des Statusworts 6041h.

3 Objektverzeichnis

Nach CiA (CAN in Automation) werden die Objekte in drei Gruppen unterteilt:

- **Standard-Objekte**
1000h, 1001h, 1018h
- **Herstellerspezifische Objekte**
2000h - 5FFFh
- **Gerätespezifische Objekte**

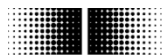
Alle anderen Objekte von 1000h - 1FFFh, 6000h - FFFFh

Sendet der CAN Master ein unzulässiges Kommando an den Antrieb, so wird als Antwort eine Abort Message mit Abort Code gesendet (siehe Tabelle 18). Die Änderung von Objekten mit Schreibzugriff kann nichtflüchtig gespeichert werden in einem EEPROM im Antrieb. Nach Einschalten des Antriebs werden alle Parameter aus diesem EEPROM geladen.

Tabelle 10: Zusammenfassung aller unterstützten Objekte. Speichern der aktuellen Objekt-Konfiguration, siehe Objekt 1010h. Laden der Default-Objekte: siehe Objekt 1011h. Für weitere Details, siehe Kapitel 4.2.

Objekt	Objekt-Nummer in hexadezimaler Darstellung (Beispiel 1000h)					
Name	---					
Format	Variablenformat des Objekts: U/I = Unsigned/Integer, Zahl=Anzahl Bit, ARR = Array, REC = Record					
Zugriff	ro = ReadOnly, wo = WriteOnly, rw = ReadWrite					
Default	Default-Wert des Parameters nach der ersten Initialisierung oder nach Laden der Default-Parameter (Objekt 1011h)					
Save	Speichern bzw. Laden des Parameters (EEPROM) über Objekte 1010h und 1011h, Subindex 1, 2, 3 oder 4					
Beschreibung	Beschreibung der Objektoptionen					

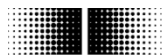
Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
1000h	Device type	U32	ro	00020192h 00h 02h 0192h	-	Gerätetyp gemäss CiA Motortyp = 00h (DC Motor) Antriebstyp = 02h (ServoDrive) Profilnummer = 192h = 402
1001h	Error register	U8	ro	0h	-	Fehlerregister mit Fehlerklasse (siehe auch Tabelle 16). Bit 2=1 Spannungs-Fehler Bit 3=1 Temperatur-Fehler Bit 4=1 CAN-Bus Kommunikations-Fehler Bit 5=1 Gerätespezifischer Fehler
1003h	Predefined error field	ARR			-	Schieberegister für die letzten 8 spezifischen Fehler oder Warnungen
00h	Number of Errors	U8	rw	0h	-	-
01h	Last entry	U32	ro		-	Zuletzt aufgetretener Fehler (siehe Tabelle 17 oder Objekt 603Fh). Error code / Beschreibung 0000h Kein Fehler 2310h Continuous over current 3110h Fehler Überspannung Bus 3111h Fehler Überspannung Leistungselektronik 3120h Fehler Unterspannung Bus 3121h Fehler Unterspannung Leistungselektronik 4210h Warnung / Fehler Übertemperatur (zur Unterscheidung, siehe Zusatz-Information Byte 3...4 in EMCY-Meldung bzw. Tabelle 17) 5441h Warnung unterer Hardware-Endschalter aktiv 5442h Warnung oberer Hardware-Endschalter aktiv 5530h Warnung Speicher (EEPROM) 6010h Warnung Software (Watchdog) 7121h Warnung Motor blockiert (zusätzliche Differenzierung siehe Zusatz-Information Byte 3...4 in EMCY-Meldung bzw. Tabelle 17)



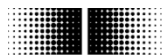
Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
						7320h Fehler Encoder 7510h Fehler interner Kommunikationsfehler 8110h Warnung CAN-Bus Kommunikation 8130h Warnung Lifeguard oder Heartbeat 8500h Warnung Position control FF00h Warnung Data valid multiturn (DVMT) FF02h Fehler Positioning timeout FF05h Warnung Deblockierung aktiv FF06h Fehler Deblocking timeout FF10h Warnung unterer Software-Endschalter aktiv FF11h Warnung oberer Software-Endschalter aktiv
...	-	...
08h	First entry	U32	ro		-	Beschreibung siehe Subindex 01h
1005h	SYNC COB-ID	U32	rw	80h	1, 2	COB-ID des SYNC Objekts
1008h	Device name	U32	ro	MSBA	-	Gerätename in ASCII (Beispiel: MSBA = „ABSM“ = 41h 42h 53h 4Dh)
1009h	-	U32	ro	werksseitig	-	Nicht zugeordnet
100Ah	Software version	U32	ro	werksseitig	-	Die Software version ist die Unterversion der Firmware des Geräts in ASCII. Sie bezieht sich auf die Korrektur eines Softwarefehlers. Die Firmware-Version setzt sich aus der Gesamt- und der Unterversion zusammen. Für Gesamtversion, siehe Objekt 1018h-3h. Beispiel: 0017 = 17.
100Ch	Guard time	U16	rw	0h	1, 2	Guard time für Node Guarding (siehe Figur 2).
100Dh	Life time factor	U8	rw	0h	1, 2	Life time factor für Node Guarding (siehe Figur 2). Die Knoten-Lebensdauer ergibt sich durch Guard Time multipliziert mit Life time factor. Im Fehlerfall, beispielsweise nach Ablauf der Knoten-Lebensdauer wird ein Life Guarding Event ausgelöst. Das Fehlverhalten wird definiert im Objekt Error behavior (Objekt 1029h). ACHTUNG: Einschalten des Node Guarding Protokolls in der Versionskontrolle (Objekt 2110h, Bit5).
1010h	Store parameters	ARR				Speichern der aktuellen Parameter des Antriebs. Spalte Save in dieser Tabelle gibt für jedes Objekt den benötigten Subindex zum Speichern der Parameter an.
00h	Grösster Subindex	U8	ro	4h		-
01h	Store all parameters	U32	rw			Speichern aller Parameter im EEPROM durch Schreiben von save = „evas“ = 65h 76h 61h 73h
02h	Store communication parameters	U32	rw			Speichern der Kommunikationsparameter im EEPROM durch Schreiben von save = „evas“ = 65h 76h 61h 73h
03h	Store application parameters	U32	rw			Speichern der Applikationsparameter im EEPROM durch Schreiben von save = „evas“ = 65h 76h 61h 73h
04h	Store manufacturer specific parameters	U32	rw			Speichern der herstellerspezifischen Parameter im EEPROM durch Schreiben von save = „evas“ = 65h 76h 61h 73h
1011h	Restore parameters	ARR				Laden der Default-Parameter des Antriebs. Spalte Save in dieser Tabelle gibt für jedes Objekt den benötigten Subindex für Laden der Default-Parameter an.
00h	Grösster Subindex	U8	ro	4h		-
01h	Restore all parameters	U32	rw			Laden aller Parameter aus dem EEPROM durch Schreiben von load = „daol“ = 64h 61h 6Fh 6Ch



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
02h	Restore communication parameters	U32	rw			Laden der Kommunikationsparameter aus dem EEPROM durch Schreiben von load = „daol“ = 64h 61h 6Fh 6Ch
03h	Restore application parameters	U32	rw			Laden der Applikationsparameter aus dem EEPROM durch Schreiben von load = „daol“ = 64h 61h 6Fh 6Ch
04h	Restore manufacturer specific parameters	U32	rw			Laden der herstellerspezifischen Parameter aus dem EEPROM durch Schreiben von load = „daol“ = 64h 61h 6Fh 6Ch
1014h	Emergency COB-ID	U32	rw	81h	1, 2	COB-ID des Emergency Objektes EMCY COB-ID = 80h + Node-ID
1016h	Consumer heartbeat	ARR				
00h	Grösster Subindex	U8	ro	1h		-
01h	Consumer heartbeat time	U32	rw	00010000h 0001h 0000h	1, 2	Consumer Heartbeat Zeitintervall (siehe Figur 3). Bit16..23: Node- ID Bit0..15: Consumer Heartbeat Zeitintervall in ms ACHTUNG: Einschalten des Heartbeat Protokolls in der Versionskontrolle (Objekt 2110h, Bit5).
1017h	Producer heartbeat time	U16	rw	0h	1, 2	Producer Heartbeat Zeitintervall in ms
1018h	Identity object	REC				Hersteller- und Geräteidentifikation für LSS
00h	Grösster Subindex	U8	ro	4h		-
01h	Vendor-ID	U32	ro	5Fh		Nummer zur Identifikation des Geräteherstellers nach CiA. (Baumer Electric AG = 5Fh)
02h	Product code	U32	ro	00000102h		Nummer zur Identifikation des Geräts (MSBA = 102h)
03h	Revision number	U32	ro	werkseitig		Die Revision number ist die Gesamtversion der Firmware des Geräts. Sie bezieht sich auf Gerätefunktionen. Die Firmware-Version setzt sich aus der Gesamt- und der Unterversion zusammen. Für Unterversion, siehe Objekt 100Ah Software version.. Beispiel: Firmware-Version 1.00.17 Gesamtversion 1.00 = 0001'0000h (Objekt 1018-3h) Unterversion 17 = 0017 (Objekt 100Ah) Die Revision number wird verwendet zur LSS Identifikation des Geräts und muss identisch sein mit der Version des Handbuchs sowie mit dem Aufdruck RevNr auf dem Typenschild auf der Aussenseite des Antriebs.
04h	Serial number	U32	ro	werkseitig		Eindeutige fortlaufende Seriennummer des Geräts
1029h	Error behaviour	ARR				Fehlerbehandlung von NMT-Kommunikationsfehlern.
00h	Grösster Subindex	U8	ro	1h		-
01h	Communication error	U8	rw	1h	1, 2	Verhalten nach Kommunikationsfehler: 0h = Wechsel in den NMT-Zustand Pre-Operational 1h = kein Zustandswechsel 2h = Wechsel in den NMT-Zustand Stopped/Prepared 3h = Reset des Antriebs
1400h	Receive PDO1 parameter	REC				
00h	Grösster Subindex	U8	ro	2h		-
01h	COB-ID	U32	rw	201h	1, 2	PDO-ID = 200h + Node-ID



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
02h	R_PDO1 type	U8	rw	FEh	1, 2	1h..F0h = Synchroner Betrieb. Mit Wert n wird das R_PDO1 auf jedes n-te SYNC-Telegramm übernommen FEh = Asynchroner Betrieb. R_PDO1 werden sofort nach Erhalt übernommen
1401h	Receive PDO2 parameter	REC				
00h	Grösster Subindex	U8	ro	2h		-
01h	COB-ID	U32	rw	301h	1, 2	PDO-ID = 300h + Node-ID
02h	R_PDO2 type	U8	rw	FEh	1, 2	1h..F0h = Synchroner Betrieb. Mit Wert n wird das R_PDO2 auf jedes n-te SYNC-Telegramm übernommen FEh = Asynchroner Betrieb. R_PDO2 werden sofort nach Erhalt übernommen
1600h	Receive PDO1 mapping	ARR				
00h	Number of mapped objects in R_PDO1	U8	ro	1h		Anzahl gemappter Objekte in R_PDO1
01h	R_PDO1 content 1	U32	ro	60400010h		Controlword. ReadOnly, obwohl von CiA als ReadWrite
1601h	Receive PDO2 mapping	ARR				
00h	Number of mapped objects in R_PDO2	U8	ro	2h		Anzahl gemappter Objekte in R_PDO2
01h	R_PDO2 content 1	U32	ro	60400010h		Controlword. ReadOnly, obwohl von CiA als ReadWrite
02h	R_PDO2 content 2	U32	ro	607A0020h		Target position. ReadOnly, obwohl von CiA als ReadWrite
1800h	Transmit PDO1 parameter	REC				
00h	Grösster Subindex	U8	ro	5h		-
01h	COB-ID	U32	rw	181h	1, 2	PDO-ID = 180h + Node-ID
02h	Transmission type	U8	rw	FEh	1, 2	1h..F0h = Zyklisch, Synchroner Betrieb. Mit Wert n wird das T_PDO1 auf jedes n-te SYNC-Telegramm in Abhängigkeit von Objekt 2800h PDO1 add-on gesendet. FEh = Asynchroner Betrieb (manufacturer specific). T_PDO1 werden mit der Zykluszeit Subindex 05h Event timer in Abhängigkeit von Objekt 2800h PDO1 add-on gesendet. FFh = Asynchroner Betrieb (device profile specific). T_PDO1 werden bei Änderung der Daten unter Berücksichtigung von Subindex 03h Inhibit time und mit der Zykluszeit Subindex 05h Event timer (auch ohne Änderung) gesendet. Objekt 2800h PDO1 add-on hat keine Auswirkung
03h	Inhibit time	U16	rw	0h	1, 2	Sendepause nach Kommunikation von T_PDO1 in 100 Mikrosekunden. Wird berücksichtigt bei Verwendung von Transmission Type FFh. <0Ah = keine Sendepause ≥0Ah = Sendepause ≥ 1 ms (Vielfache von 0Ah sind sinnvoll)
05h	Event timer	U16	rw	203h	1, 2	0h = Transmit PDO1 werden nicht (Transmission Type FEh) gesendet bzw. nur bei Änderung (Transmission Type FFh) gesendet >0h = Zykluszeit in Millisekunden



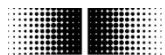
Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
1801h	Transmit PDO2 Parameter	REC				
00h	Grösster Subindex	U8	ro	5h		-
01h	COB-ID	U32	rw	281h	1, 2	PDO-ID = 280h + Node-ID
02h	Transmission type	U8	rw	FEh	1, 2	<p>1h..F0h = Synchroner Betrieb. Mit Wert n wird das T_PDO2 auf jedes n-te SYNC-Telegramm in Abhängigkeit von Objekt 2801h PDO2 add-on gesendet.</p> <p>FEh = Asynchroner Betrieb (manufacturer specific). T_PDO2 werden mit der Zykluszeit Subindex 05h Event timer in Abhängigkeit von Objekt 2800h PDO2 add-on gesendet.</p> <p>FFh = Asynchroner Betrieb (device profile specific). T_PDO2 werden bei Änderung der Daten unter Berücksichtigung von Subindex 03h Inhibit time und mit der Zykluszeit Subindex 05h Event timer (auch ohne Änderung) gesendet. Objekt 2800h PDO2 add-on hat keine Auswirkung</p>
03h	Inhibit time	U16	rw	0h	1, 2	<p>Sendepause nach Kommunikation von T_PDO2 in 100 Mikrosekunden. Wird berücksichtigt bei Verwendung von Transmission Type FFh.</p> <p><0Ah = keine Sendepause</p> <p>>=0Ah = Sendepause >= 1 ms (Vielfache von 0Ah sind sinnvoll)</p>
05h	Event timer	U16	rw	104h	1, 2	<p>0h = Transmit PDO2 werden nicht (Transmission Type FEh) gesendet bzw. nur bei Änderung (Transmission Type FFh) gesendet</p> <p>>0h = Zykluszeit in Millisekunden</p>
1A00h	Transmit PDO1 mapping	ARR				
00h	Number of mapped objects in T_PDO1	U8	ro	2h		Anzahl gemappter Objekte in T_PDO1
01h	T_PDO1 content 1	U32	ro	60410010h		ReadOnly, obwohl von CiA als ReadWrite
02h	T_PDO1 content 2	U32	ro	60FD0020h		ReadOnly, obwohl von CiA als ReadWrite
1A01h	Transmit PDO2 mapping	ARR				
00h	Number of mapped objects in T_PDO2	U8	ro	1h		Anzahl gemappter Objekte in T_PDO2
01h	T_PDO2 content 1	U32	ro	60640020h		ReadOnly, obwohl von CiA als ReadWrite
2100h	Baudrate	U8	rw	2h	1, 2	<p>Baudrate des Antriebs für Buskommunikation (siehe Tabelle 12.</p> <p>01h = 20 kBit/s 02h = 50 kBit/s 03h = 100 kBit/s 04h = 125 kBit/s 05h = 250 kBit/s 06h = 500 kBit/s 07h = 800 kBit/s 08h = 1000 kBit/s</p> <p>ACHTUNG: Nach Änderung der Baudrate muss Antrieb neu initialisiert werden, beispielsweise durch Ein-Aus-Schalten oder mittels NMT-Reset.</p>
2101h	Node-ID	U8	rw	1h	1, 2	<p>Knotennummer bzw. Teilnehmeradresse des Antriebs auf dem CAN-Bus. Mögliche Werte sind 1..127.</p> <p>ACHTUNG: Nach Änderung der Node-ID muss der Antrieb neu initialisiert werden, beispielsweise durch Ein-Aus-Schalten oder mittels NMT-Reset.</p>



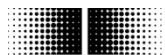
Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
2102h	Node-ID offset	U8	rw	0h		<p>Offset für Node-ID. Mögliche Werte sind 0..126.</p> <p>Node-ID Offset wird sofort nach Eingabe zur Node-ID addiert und aktiviert ohne Initialisierung. Node-ID Offset kann nicht dauerhaft abgespeichert werden und ist nach Initialisierung beispielsweise durch Ein-Aus-Schalten wieder auf 0h.</p> <p>Zur Bestätigung der Änderung des Node-ID offset sendet der Antrieb eine Download Response mit neuer Node-ID (siehe SDO Kommandos in Tabelle 2).</p> <p>ACHTUNG: Der Node-ID Offset kann nicht abgespeichert werden.</p>
2110h	Version control	U32	rw	0Ah	1, 3	<p>Versionskontrolle: Definition von T_PDO1 Inhalt und Verhalten im Fehlerfall</p> <p>Bit 1: T_PDO1 Standardinhalt 0 = T_PDO1 enthält Statuswort (Tabelle 4) mit 2 Byte Länge 1 = T_PDO1 enthält Statuswort (Tabelle 4) und Statuswort-Zusatz (Tabelle 5) mit 3 Byte Gesamtlänge (Default).</p> <p>Bit 3: Verhalten bei BusOFF 0 = bei BusOFF zieht sich Antrieb vom Bus zurück 1 = bei BusOFF automatisch Initialisierung (Default)</p> <p>Bit 5: 0 = Heartbeat (Default) 1 = Nodeguarding</p> <p>Bit 9: Servicebetrieb (Jogging ohne Bus möglich) 0 = ausgeschaltet (Default) 1 = eingeschaltet</p> <p>Bit 10: Emergency-Meldung beim Referenzieren 0 = aktiviert (Default) 1 = deaktiviert</p> <p>Bit 11: Emergency-Meldung der Software-Endschalter 0 = aktiviert (Default) 1 = deaktiviert</p> <p>Bit 12: T_PDO1 Zusatzinhalt: Sende zusätzlich zum T_PDO1 Standardinhalt den Zustand der Hardware-Schalteingänge (Tabelle 6, Object 60FDh) mit 8 Byte Gesamtlänge 0 = deaktiviert (Default) 1 = aktiviert</p> <p>Bit 13: T_PDO2 Zusatzinhalt: Sende zusätzlich zum T_PDO2 Standardinhalt die aktuelle Geschwindigkeit Velocity actual value (Objekt 606Ch Velocity actual value) mit 8 Byte Gesamtlänge. Es kann entweder Bit13 oder Bit14 aktiviert werden. 0 = deaktiviert (Default) 1 = aktiviert</p> <p>Bit 14: T_PDO2 Zusatzinhalt: Sende zusätzlich zum T_PDO2 Standardinhalt den aktuellen Strom Current actual value (Objekt 6078h Current actual value) mit 8 Byte Gesamtlänge Es kann entweder Bit13 oder Bit14 aktiviert werden. 0 = deaktiviert (Default) 1 = aktiviert</p> <p>Bit 18: Automatische Deblockierung: Beschreibung siehe Kapitel 4.8 Automatische Deblockierung 0 = deaktiviert (Default) 1 = aktiviert</p> <p>Bit 19: Strombegrenzung: Beschreibung siehe Kapitel 4.3.5 Strombegrenzung 0 = deaktiviert (Default) 1 = aktiviert</p> <p>Bit 21: Verhalten Motor blocked: 0 = Motor blocked wird als Warning behandelt (Default) 1 = Motor blocked wird als Error behandelt</p>



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
2111h	Positioning parameter	REC				Positionierungsparameter
00h	Grösster Subindex	U8	ro	13h		-
01h	Backlash compensation	U8	rw	0h	1, 3	<p>Spielausgleich, um Spindel- oder Getriebeispiel auszugleichen. Hierbei kann definiert werden, aus welcher Richtung die Zielposition angefahren werden soll. Wenn nötig, wird dazu die Zielposition um Backlash delta überfahren und anschliessend direkt angefahren.</p> <p>00h = Zielposition wird direkt angefahren 01h = Zielposition wird von unten angefahren 02h = Zielposition wird von oben angefahren</p>
02h	Backlash delta	U32	rw	64h	1, 3	Parameter in benutzerdefinierter Positionseinheit. Gibt an, wie weit eine Zielposition überfahren wird bei eingeschaltetem Spielausgleich
03h	Positioning timeout	U32	rw	0h	1, 3	<p>Timeout für Positioniervorgänge in Sekunden: Benötigt ein Positioniervorgang mehr Zeit als Positioning timeout, so wird er abgebrochen und es wird eine EMCY-Meldung mit Fehlercode FF02h ausgegeben</p> <p>0h = Positioning timeout deaktiviert >0h = Positioning timeout in Sekunden</p>
0Bh	Time of jogging start speed	U32	rw	3E8h	1, 3	Zeitdauer der Anfangsgeschwindigkeit beim Tippbetrieb in ms
0Ch	Jogging start speed	U32	rw	14h	1, 3	Anfangsgeschwindigkeit beim Tippbetrieb in benutzerdefinierter Geschwindigkeitseinheit (siehe Objekt 608Ch).
0Dh	Digital input 1	U8	rw	0h	1, 3	<p>Digitaler Hardware-Schalteingang 1 zur Steuerung des Antriebs (vgl. Kontrollwort Objekt 6040h). Eingang ist HIGH aktiv, Invertierung durch Addition mit 80h.</p> <p>Parameter / Beschreibung: 0h = Schalteingang inaktiv 01h = SwitchOn 02h = EnableVoltage 03h = QuickStop 04h = EnableOperation 05h = StartAction 06h = Change set immediately 07h = Relative/absolute 08h = Error reset 09h = Halt 0Ah = Jogging+ (innerhalb Software-Endschalter, wenn aktiv) 0Bh = Jogging- (innerhalb Software-Endschalter, wenn aktiv) 10h = Status: Betrieb freigeben (01h..04h) 12h = Referenzposition setzen mit Presetwert (Objekt 2111h, Subindex 11h). Nach dem Referenzieren wird dieser Parameter wieder auf 00h zurückgesetzt. 13h = Jogging+ (mit Warnung auch über Software-Endschalter hinaus) 14h = Jogging- (mit Warnung auch über Software-Endschalter hinaus) 15h = Oberer externer Hardware-Endschalter 16h = Unterer externer Hardware-Endschalter 17h = nicht definiert</p> <p>ACHTUNG: Bei Einstellwert >0 steuert der Eingang direkt den Antrieb. In diesem Fall verliert der Master auf dem CAN-Bus die Kontrolle.</p>
0Eh	Digital input 2	U8	rw	0h	1, 3	Digitaler Hardware-Schalteingang 2. Beschreibung siehe Digital input 1.



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
11h	Reference position	I32	rw	0h	1, 3	Presetwert für Referenzierung des Antriebs. Für Einzelheiten zur Referenzierung, siehe Objekt 6098h Homing methode und Kapitel 4.4 Antriebsposition referenzieren (Homing). Bei den Homing-Methoden E2h, E1h, D8h, D7h, wird die Referenzposition direkt dem Objekt 6064h Position actual value zugewiesen.
13h	Simulation controlword	U16	rw	0h	1, 3	Kontrollwort, das automatisch gesetzt wird, wenn keine Buskommunikation vorliegt (z.B. für Stand-Alone-Betrieb des Antriebs ohne Feldbus).
1Dh	Service speed	U32	rw	1Eh	1, 3	Geschwindigkeit in benutzerdefinierter Einheit beim Servicebetrieb ohne Busankopplung
1Eh	Free referencing distance	U32	rw	64h	1, 3	Weg der beim Referenzieren frei gefahren wird in benutzerdefinierter Positionseinheit (Freifahrtweg).
2112h	Statistics	REC				Statistik für Betrieb des Antriebs
00h	Grösster Subindex	U8	ro	6h		-
01h	Number of watchdog events	U16	ro	0h	1, 4	Anzahl der Watchdog-Events des Prozessors
02h	Number of position warnings	U16	rw	0h	1, 4	Anzahl der aufgetretenen Reed-Counter Fehler.
05h	Electronics temperature	U16	ro	-		Aktuelle Temperatur der Leistungselektronik in °C
06h	I ² t Overload Level	U16	ro	0		Aktuell aufsummierte I ² t-Überlast in [%], siehe Kapitel 4.3.4 I ² t-Begrenzung. Wertebereich 0...99
2113h	Endtest	REC				Defaultwerte, Motorenparameter
00h	Grösster Subindex	U8	ro	33h		-
20h	Break away current	U16	ro	werkseitig		Losbrechstrom in mA zum Freifahren nach Referenzieren auf Anschlag oder beim Deblockieren
27h	Position target range	I16	rw	4h	1,4	Halbe Breite des Zielfensters in Encoder-Schritten. Wenn aktuelle Position innerhalb Sollposition ± Position target range ist, gilt Zielfenster als erreicht.
31h	-	U32	ro	werkseitig	-	Nicht zugeordnet
33h	Bootloader version	U8	ro	werkseitig	-	Version des Bootloaders Beispiel: Version 5 = 05h
2114h	Warning temperature	U16	rw	50h	1, 3	Temperatur-Grenzwert in °C für Ansprechen der Übertemperatur-Warnung. Überschreitet die Electronics temperature (Objekt 2112h-05h) diesen Grenzwert, wird eine EMCY-Meldung generiert (Objekt 1003h und 603Fh, Error code 4210h-0001h). Der Antrieb kann weiter betrieben werden (50h = 80°C). Bei Temperaturfehler wird der Antrieb gestoppt und in den Zustand ErrorDiagnostic versetzt (Error code 4210h-0000h). Zurücksetzen (Error reset) erfolgt mit dem Kontrollwort (Objekt 6040h).
2300h	Customer EEPROM	ARR				Speicherbereich für Anwenderdaten
00h	Grösster Subindex	U8	ro	7h		-
01h	Data0	U16	rw	0h	1, 4	-
...
07h	Data6	U16	rw	0h	1, 4	-
2800h	PDO1 add-on	U8	rw	1h	1, 2	0 = PDO1 zyklisch senden 1 = PDO1 nur bei Änderung senden n = PDO1 nur bei Änderung senden (n mal, n : 2...255)



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
2801h	PDO2 add-on	U8	rw	1h	1, 2	0 = PDO2 zyklisch senden 1 = PDO2 nur bei Änderung senden n = PDO2 nur bei Änderung senden (n mal, n : 2...255)
603Fh	Error code	U16	ro	0h		Zuletzt aufgetretener Fehler (Tabelle 17, Objekt 1003h).
6040h	Controlword	U16	rw	0h		Kontrollwort (siehe Tabelle 9) Bit 0 = SwitchOn Bit 1 = EnableVoltage Bit 2 = QuickStop Bit 3 = EnableOperation Bit 4 = StartAction (Fahre zum Ziel) Bit 5 = Change set immediately Bit 6 = Relative/absolute movement Bit 7 = Error reset Bit 8 = Halt Bit 9...10 nicht definiert Bit 11 = Jogging+ Bit 12 = Jogging- Bit 13...15 nicht definiert
6041h	Statusword	U16	ro	0h		Statuswort (siehe Tabelle 4) Bit 0 = Ready to switch on Bit 1 = SwitchOn enabled Bit 2 = Operation enabled Bit 3 = Error active Bit 4 = Voltage enabled Bit 5 = QuickStop Bit 6 = SwitchOn disabled Bit 7 = Warning active Bit 8 = CalibrationOK Bit 9 = Not used (permanently 1) Bit 10 = Target position reached Bit 11 = Internal software limit switch active Bit 12 = Drive moving Bit 13 = Not used (permanently 0) Bit 14 = HomingOK Bit 15 = External hardware limit switch active ACHTUNG: HomingOK und CalibrationOK werden auf Null gesetzt, sobald ein Getriebe- oder Positionsparameter verändert wird (siehe Objekt 6098h).
605Dh	Halt option code	I16	rw	2h	1, 3	Verhalten des Antriebs nach Halt-Befehl (Kontrollwort Bit 8) 0h = Austrudeln des Antriebs (Regler wird deaktiviert) 1h = Bremsen mit Quickstop Rampe (nach Stillstand wird der Regler deaktiviert) 2h = wie bei 1h
6060h	Modes of operation	I8	wo	1h		Betriebsmodus des Antriebs 01h = Positioning 06h = Homing Der aktuell aktive Betriebsmodus kann in Objekt 6061h gelesen werden.
6061h	Modes of operation display	I8	ro	1h		Aktuell aktiver Betriebsmodus des Antriebs 01h = Positioning 06h = Homing
6063h	Position actual steps	I32	ro	-		Aktuelle Position in Encoder-Schritten



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
6064h	Position actual value	I32	ro	-		<p>Aktuelle Position in benutzerdefinierter Positionseinheit (siehe Objekt 608Ah). Wert wird berechnet aus aktueller Position in Encoder-Schritten multipliziert mit dem Position factor.</p> <p>Position actual value = Position actual steps * Position factor</p> <p>ACHTUNG: Wenn Positionseinheit Encoder-Schritte, dann Position factor = 1. Gear ratio oder Feed constant werden in diesem Fall nicht berücksichtigt.</p> <p>Der Antrieb ist mit einem Absolut-Multiturn-Encoder ausgerüstet. Nach dem Wiedereinschalten des Antriebs oder nach einem Ausfall der Betriebsspannung steht die aktuelle Position daher jederzeit sofort zur Verfügung – ohne Referenzierung.</p>
6068h	Position window time	U16	rw	0h	1,3	<p>Wartezeit nach der der Regler ausgeschaltet wird, sobald das Zielfenster erreicht wurde.</p> <p>0h = deaktiviert: Wenn der Antrieb nach Zielerreichung wieder aus dem Zielfenster bewegt wird, ist der Regler aktiv und stellt den Antrieb ins Zielfenster zurück.</p> <p>>0h = aktiviert: Wartezeit in Millisekunden</p>
606Ch	Velocity actual value	I32	ro	-		<p>Aktuelle Geschwindigkeit in benutzerdefinierter Geschwindigkeitseinheit (siehe Objekt 608Ch). Wert wird berechnet aus aktueller Motordrehzahl in U/s multipliziert mit dem Velocity factor.</p> <p>Velocity actual value = Motordrehzahl * Velocity factor</p> <p>Für Geschwindigkeitseinheit Encoder-Schritte / Sekunde, Velocity factor = 8. Gear ratio oder Feed constant dann werden nicht berücksichtigt.</p>
6073h	Maximum current	U16	rw	werkseitig	1, 3	<p>Vom Anwender einstellbare i^2t-Begrenzung für den Motorstrom zur Drehmomentbegrenzung und als Überlastschutz.</p> <p>00h = Strombegrenzung mit zulässigem Maximalwert xxh = Maximaler Motorstrom in mA</p> <p>Die Bestimmung erfolgt durch Mittelwertbildung über 8 Messwerte (alle 12 ms). Bei Überschreiten des Grenzwerts wird der aktive Fahrauftrag abgebrochen, eine EMCY Meldung abgesetzt und der Antrieb in den Zustand Errordiagnostik gebracht.</p> <p>ACHTUNG: Werkseitige Einstellung je nach Motorentyp (siehe Kapitel Technische Daten bzw. Datenblatt). Die werksseitige Einstellung ist der Höchstwert für den Einstellbereich von Maximum current und entspricht dem Wert in Objekt 6510h-7h Max motor current. Für maximales Anfahrmoment kann der Motorstrom kurzzeitig Max motor current erreichen, auch wenn Maximum current tiefer eingestellt ist.</p>
6078h	Current actual value	I16	ro	-		<p>Aktueller Strom des Antriebs in Milliampere (Mittelwert über acht Messwerte)</p>
607Ah	Target position	I32	rw	0h	1, 3	<p>Sollposition in benutzerdefinierter Positionseinheit (siehe Objekt 608Ah).</p> <p>Bit 6 im Kontrollwort gibt vor, ob Zielposition absolut oder relativ zu aktueller Position ist.</p> <p>Zwischenposition für Spielausgleich wird bei eingeschaltetem Spielausgleich automatisch bestimmt.</p> <p>Wenn aktuelle Position ausserhalb der Software-Endschalter liegt, sendet der Antrieb eine Abort Message (0609003h = Wert ausserhalb des zulässigen Wertebereichs).</p>



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
607Dh	Software position limit	ARR				Positionen der Software-Endschalter. Diese Positionen können direkt definiert oder über Referenzierung gesetzt werden (siehe Objekt 6098h). Die Software-Endschalter können deaktiviert werden, indem Subindex 01h und 02h auf denselben Wert gesetzt werden.
00h	Grösster Subindex	U8	ro	2h		-
01h	Minimum software position limit	I32	rw	0h	1, 3	Position des unteren Software-Endschalters in benutzerdefinierter Positionseinheit (siehe Objekt 608Ah).
02h	Maximum software position limit	I32	rw	0h	1, 3	Position des oberen Software-Endschalters in benutzerdefinierter Positionseinheit (siehe Objekt 608Ah).
607Eh	Polarity	U8	rw	0h	1, 3	Drehrichtung der Motorachse bei steigenden Positionswerten (Blick auf Achse von Getriebeseite her). Bit7= 0: Drehrichtung im Uhrzeigersinn (CW) = 1: Drehrichtung im Gegenuhrzeigersinn (CCW) Bit0..6: nicht definiert ACHTUNG: Nach Änderung der Drehrichtung müssen Positionen der Software-Endschalter gespiegelt werden. (d.h. die Positionswerte von oberem und unterem Software-Endschalter tauschen und bei beiden das Vorzeichen ändern). Bei Stirnradgetrieben führt jede Getriebestufe zu einer Umkehrung der Drehrichtung an der Ausgangswelle. Bei Änderung wird automatisch HomingOK und CalibrationOK auf Null gesetzt (siehe Objekt 6098h).
607Fh	Max profile velocity	U32	rw	werksseitig	1, 3	Maximal zulässige Geschwindigkeit in benutzerdefinierter Geschwindigkeitseinheit (siehe Objekt 608Ch). Max profile velocity resultiert aus umgerechneter maximaler Motordrehzahl. Maximaler Wert wird werksseitig eingestellt und kann nicht überschritten werden.
6081h	Profile velocity	U32	rw	werksseitig	1, 3	Sollgeschwindigkeit in benutzerdefinierter Geschwindigkeitseinheit (siehe Objekt 608Ch). Für aktuelle Ist-Geschwindigkeit, siehe Objekt 606Ch). ACHTUNG: Sollgeschwindigkeit muss kleiner als maximal zulässige Geschwindigkeit (607Fh) sein. Ansonsten wird eine Abbruchmeldung ausgegeben (siehe SDO Abort Code in Tabelle 18)
6089h	Position notation	I8	rw	FDh	1, 3	Notation der benutzerdefinierten Positionseinheit. FAh = mikro FBh = 10 mikro FCh = 100 mikro FDh = milli FEh = 10 milli FFh = 100 milli 00h = 1 (none) 01h = 10 02h = 100 03h = kilo ACHTUNG: Nur gültig für Linearbewegung mit Längeneinheit (Position dimension = 01h). Bei Änderung wird automatisch HomingOK und CalibrationOK auf Null gesetzt (siehe Objekt 6098h).



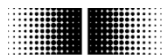
Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
608Ah	Position dimension	U8	rw	01h	1, 3	<p>Dimension der benutzerdefinierten Positionseinheit.</p> <p>Längeneinheit für Linearbewegung 01h = Meter</p> <p>Winkleinheit für Drehbewegung (Getriebeausgang) 00h = Umdrehungen ACh = Encoder-Schritte 10h = Radiant 41h = Grad 42h = Winkelminute 43h = Winkelsekunde</p> <p>Die benutzerdefinierte Positionseinheit ergibt sich durch Multiplikation von Position notation mit Position dimension (nur gültig für Linearbewegung).</p> <p>Beispiel: Positionseinheit in Millimeter (Position notation = FDh und Position dimension = 01h).</p> <p>ACHTUNG: Bei Änderung wird automatisch HomingOK und CalibrationOK auf Null gesetzt (siehe Objekt 6098h).</p>
608Bh	Velocity notation	I8	rw	00h	1, 3	<p>Notation der benutzerdefinierten Geschwindigkeitseinheit.</p> <p>FAh = mikro FBh = 10 mikro FCh = 100 mikro FDh = milli FEh = 10 milli FFh = 100 milli 00h = 1 (none) 01h = 10 02h = 100 03h = kilo</p> <p>ACHTUNG: Nur gültig für Linearbewegung mit Längeneinheit.</p>
608Ch	Velocity dimension	U8	rw	A3h	1, 3	<p>Dimension der benutzerdefinierten Geschwindigkeitseinheit.</p> <p>Längeneinheit für Linearbewegung A6h = Meter pro Sekunde A7h = Meter pro Minute A8h = Meter pro Stunde</p> <p>Winkleinheit für Drehbewegung 00h = Encoder-Schritte pro Sekunde A3h = Umdrehungen pro Sekunde A4h = Umdrehungen pro Minute A5h = Umdrehungen pro Stunde</p> <p>Die benutzerdefinierte Geschwindigkeitseinheit ergibt sich durch Multiplikation von Velocity notation mit Velocity dimension (nur gültig für Linearbewegung).</p> <p>Beispiel: Geschwindigkeitseinheit in Millimeter / Sekunde (FDh und A6h).</p>
608Fh	Position encoder resolution	ARR				<p>Singleturn-Auflösung des absoluten Drehgebers in Schritten/Umdrehung. Die Multiturn-Auflösung beträgt 29 Bit bzw. 536'870'912 Umdrehungen.</p> <p>ACHTUNG: Bei Änderung wird automatisch HomingOK und CalibrationOK auf Null gesetzt (siehe Objekt 6098h).</p>
00h	Grösster Subindex	U8	ro	2h		-
01h	Encoder increments	U32	rw	8h	1, 3	Anzahl Schritte des Encoders pro Motorumdrehung (Zähler)
02h	Encoder revolutions	U32	rw	1h	1, 3	Anzahl Encoder- bzw. Motorumdrehungen (Nenner)



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
6091h	Gear ratio	ARR				<p>Getriebeuntersetzung i in Motorumdrehungen pro Umdrehung der Welle am Getriebeausgang.</p> <p>$\text{Gear ratio} = \text{Motor revolutions} / \text{Gear shaft revolutions}$</p> <p>Darstellung als Bruch, da nur ganzzahlige Werte erlaubt.</p> <p>Beispiel: $i = 61,25 : 1 = 6125 / 100$.</p> <p>ACHTUNG: Bei Änderung wird automatisch HomingOK und CalibrationOK auf Null gesetzt (siehe Objekt 6098h).</p>
00h	Grösster Subindex	U8	ro	2h		-
01h	Motor revolutions	U32	rw	1h	1, 3	Anzahl Motorumdrehungen (Zähler)
02h	Gear shaft revolutions	U32	rw	1h	1, 3	Anzahl Umdrehung der Welle am Getriebeausgang (Nenner)
6092h	Feed constant	ARR				<p>Berechnungsfaktor zur Umrechnung von Umdrehungen des Motors oder der Getriebeausgangswelle in Bewegung auf Benutzerseite.</p> <p>$\text{Feed constant} = \text{Feed} / \text{Spindle shaft revolutions}$.</p> <p>Bei Linearbewegungen mit Spindel entspricht Feed constant der Spindelsteigung. Darstellung als Bruch, da nur ganzzahlige Werte erlaubt. Beispiel: Spindelsteigung 2,5 mm / Umdrehung = 1000 mm / 400 Umdrehungen = 1 Meter / 400 Umdrehungen.</p> <p>ACHTUNG: Die Einheit für Weg (Feed) ist immer Meter. Es wird kein Bezug hergestellt zur eingestellten Position notation (6089h) oder Position dimension (608Ah).</p> <p>Bei Änderung wird automatisch HomingOK und CalibrationOK auf Null gesetzt (siehe Objekt 6098h).</p>
00h	Grösster Subindex	U8	ro	2h		-
01h	Feed	U32	rw	1h	1, 3	<p>Weg in Meter (Zähler)</p> <p>ACHTUNG: Die Einheit für Weg (Feed) ist immer Meter</p>
02h	Spindle shaft revolutions	U32	rw	1h	1, 3	Anzahl Umdrehungen (Nenner)
6093h	Position factor	ARR				Berechnungsfaktor zur Umrechnung von Position actual steps (in Encoder-Schritten) in Position actual value (in benutzerdefinierter Positionseinheit, siehe Objekt 608Ah).
00h	Grösster Subindex	U8	ro	2h		-
01h	Numerator	U32	ro	1h		
02h	Feed constant	U32	ro	1h		
6094h	Velocity factor	ARR				Berechnungsfaktor zur Umrechnung von Motordrehzahl (in Umdrehungen/s) in Velocity actual value (in benutzerdefinierter Geschwindigkeitseinheit, siehe Objekt 608Ch).
00h	Grösster Subindex	U8	ro	2h		-
01h	Numerator	U32	ro	1h		
02h	Divisor	U32	ro	1h		
6098h	Homing method	I8	rw	0h		<p>Methode zur Referenzierung der Antriebs-Position. Nach dem Abschluss der Prozedur wird die Referenzposition gespeichert (Objekt 11h). Für Einzelheiten siehe Kapitel 4.4.</p> <p>Parameter / Beschreibung</p> <p>00h Keine Referenzierung aktiv</p> <p>F6h Setze aktuelle Position (Objekt 6064h) als unteren Software-Endschalter (Objekt 607Dh-1h)</p> <p>F5h Setze aktuelle Position (Objekt 6064h) als oberen Software-Endschalter (Objekt 607Dh-2h)</p> <p>F4h Referenzieren mit Presetwert: Aktuelle Position (Objekt 6064h) mit überschrieben mit Zielposition (Objekt 607Ah).</p>



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
						<p>Anschliessend wird im Statuswort (Objekt 6041h) automatisch Bit14 (HomingOK) gesetzt.</p> <p>F3h Referenzieren mit Einmessmethode (Figur 16): Funktion dient zur Kalibrierung der Position beispielsweise schwer zugänglicher Referenzschalter: Aktuelle Position (Objekt 6064h) wird automatisch überschrieben mit Zielposition (Objekt 607Ah). Zusätzlich wird automatisch Referenzposition (Objekt 2111h-11h) um Einmess-Offset korrigiert.</p> <p>Anschliessend wird im Statuswort (Objekt 6041h) automatisch zusätzlich Bit8 (CalibrationOK) gesetzt.</p> <p>ACHTUNG: Nur zulässig, wenn vorher im Statuswort (Objekt 6041h) bereits Bit14 (HomingOK) gesetzt ist.</p> <p>ECh Fahr zu unterem Software-Endschalter</p> <p>EBh Fahr zu oberem Software-Endschalter</p> <p>EAh Fahr zu Position Null</p> <p>E2h Referenzieren auf unteren Referenzschalter (Figur 14): Fahr zu unterem Referenzschalter. Sobald unterer Referenzschalter anspricht, wird automatisch aktuelle Position (Objekt 6064h) überschrieben mit gespeicherter Referenzposition (Objekt 2111h-11h).</p> <p>Anschliessend wird im Statuswort (Objekt 6041h) automatisch Bit14 (HomingOK) gesetzt.</p> <p>ACHTUNG: Nur möglich wenn Input 1 oder Input 2 (Objekt 0Dh oder 0Eh) eingestellt ist auf Referenzpunkt setzen (12h).</p> <p>E1h Referenzieren auf oberen Referenzschalter (Figur 14): Beschreibung analog Beschreibung E2h Referenzieren auf unteren Referenzschalter.</p> <p>D8h Referenzieren auf unteren Anschlag (Figur 15): Fahr zu unterem Anschlag. Sobald Antrieb blockiert, wird automatisch aktuelle Position (Objekt 6064h) überschrieben mit gespeicherter Referenzposition (Objekt 2111h-11h).</p> <p>Anschliessend wird im Statuswort (Objekt 6041h) automatisch Bit14 (HomingOK) gesetzt.</p> <p>D7h Referenzieren auf oberen Anschlag (Figur 15) Beschreibung analog Beschreibung D8h Referenzieren auf unteren Anschlag.</p> <p>Nach der Referenzierung wird das Objekt Homing method wieder automatisch auf 0h zurückgesetzt.</p> <p>ACHTUNG: HomingOK und CalibrationOK werden automatisch auf Null gesetzt bei Schalten in Homing oder bei Veränderung eines der Parameter zur Positionsberechnung: Polarity 607Eh, Position notation 6089h, Position dimension 608Ah, Position encoder resolution 608Fh, Gear ratio 6091h, Feed constant 6092h (siehe Kapitel 4.3.5).</p>
6099h	Homing speed	ARR				Geschwindigkeit für Referenzierung
00h	Grösster Subindex	U8	ro	2h		-
01h	Speed during search for switch	U32	rw	0Ah	1, 3	Geschwindigkeit in benutzerdefinierter Einheit beim Referenzieren auf Anschlag oder beim Fahren auf Endschalter.
02h	Speed during search for zero position	U32	rw	0Ah	1, 3	Geschwindigkeit in benutzerdefinierter Einheit beim Fahren zur Position Null.
60FDh	Digital input monitor	U32	ro	00000000h		<p>Monitor für externe Hardware-Schalteingänge</p> <p>Bit0..15 nicht definiert</p> <p>Bit16 Monitorbit für Digital input 1 (Objekt 2111h-0Dh)</p> <p>Bit17 Monitorbit für Digital input 2 (Objekt 2111h-0Eh)</p> <p>Bit18..31 nicht definiert</p>
6402h	Motor type	U16	rw	0001h		Motortyp (0001h = DC Motor mit PWM Ansteuerung)



Objekt	Name	Format	Zugriff	Default	Save	Beschreibung
6410h	Motor data	REC				Motorparameter
00h	Grösster Subindex	U8	ro	2h		-
01h	Motor operating time	U32	ro		1, 4	Betriebszeit des Motors in Sekunden. Wert wird alle 6 Minuten gespeichert. Durch Zurücksetzen in den Auslieferungszustand wird der Zähler auf 0 gesetzt.
02h	Software version controller	U32	ro	werkseitig		Version der Firmware des Reglerprozessors (Beispiel: Version 1.00 = 0001h 0000h)
6502h	Drive modes	U32	ro	00000021h		Bit0 Positioning Bit5 Homing
6510h	Drive data	REC				Antriebsparameter
00h	Grösster Subindex	U8	ro	0Ch		-
01h	Drive operating time	U32	ro		1, 4	Betriebszeit des Antriebs in Sekunden. Wert wird alle 6 Minuten gespeichert. Durch Zurücksetzen in den Auslieferungszustand wird der Zähler auf 0 gesetzt.
02h	Software version encoder	U32	ro	-		Version der Firmware des Multiturn Encoder (Beispiel: Version 0.0.07 = 0000'0007h)
03h	Number of positioning tasks	U32	ro		1, 4	Anzahl der Positionieraufträge. Wert wird alle 6 Minuten gespeichert. Durch Zurücksetzen in den Auslieferungszustand wird der Zähler auf 0 gesetzt.
04h	Number of homing tasks	U32	ro		1, 4	Anzahl der durchgeführten Referenzierungen und Endschalter-Definitionen. Wert wird alle 6 Minuten gespeichert. Durch Zurücksetzen in den Auslieferungszustand wird der Zähler auf 0 gesetzt.
05h	Number of saving tasks	U32	ro		1, 4	Anzahl der Speichervorgänge ins EEPROM. Wert wird alle 6 Minuten gespeichert. Durch Zurücksetzen in den Auslieferungszustand wird der Zähler auf 0 gesetzt.
06h	Error temperature	U16	ro	6Eh		Temperatur-Grenzwert in °C für Ansprechen des Übertemperatur-Fehlers (6Eh = 110°C). Überschreitet die Electronics temperature (Objekt 2112h-05h) diesen Grenzwert, wird der Antrieb gestoppt und eine EMCY-Meldung generiert (Objekt 1003h und 603Fh). Anschliessend wechselt der Antrieb in den Zustand Error Diagnostic (Error code 4210h-0001h). Zurücksetzen des Temperatur-Fehlers (Error reset) erfolgt mit Kontrollwort (Objekt 6040h). Der Temperatur-Grenzwert für den Übertemperatur-Fehler ist werkseitig definiert und kann nicht verändert werden.
07h	Max motor current	U16	ro	werkseitig		Maximal zulässiger Motorstrom in mA. Dieser Strom ist werkseitig eingestellt und kann nicht verändert werden. Er gibt den Maximalstrom vor, der in Objekt 6073h eingestellt werden kann.
08h	Min position	S32	ro			Minimal zulässige Positionsangabe in benutzerdefinierter Positionseinheit (siehe Objekt 608Ah)
09h	Max position	S32	ro			Maximal zulässige Positionsangabe in benutzerdefinierter Positionseinheit (siehe Objekt 608Ah)
0Ah	Max velocity	U32	ro			Maximal zulässige Geschwindigkeitsangabe in benutzerdefinierter Geschwindigkeitseinheit (siehe Objekt 608Ch).
0Ch	Continuous motor current	U16	ro	werkseitig		Zulässiger Motorendauerstrom (Nennstrom) in mA. Er ist werkseitig eingestellt und kann nicht verändert werden. Solange dieser Strom nicht überschritten wird, liegt keine I ² t-Überlastung des Antriebs vor.

4 Inbetriebnahme

4.1 Node-ID, Baudrate und Abschlusswiderstand einstellen

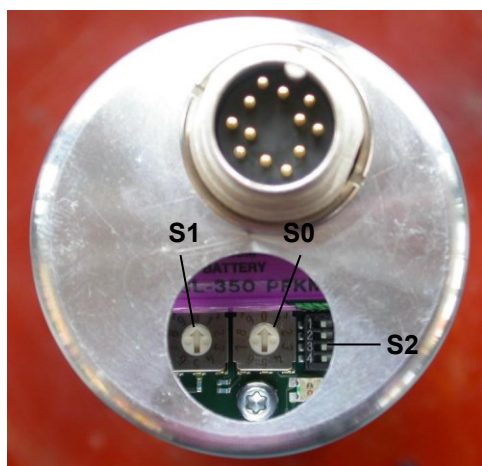
Wenn der Antrieb an den Bus angeschlossen wird, meldet er sich nach dem Einschalten mit einer BootUp-Meldung auf dem Bus an. Nun muss der Antrieb an seine Umgebung angepasst und konfiguriert werden.

Node-ID und Baudrate können entweder über Schalter, im EEPROM oder mit LSS eingestellt werden. Hierbei ist die Einstellung der Einstellschalter dominant über alle anderen Einstellmöglichkeiten. Einstellschalter sind werkseitig deaktiviert und Node-ID sowie Baudrate sind definiert in Objekt 2101h und 2100h bzw. aus dem EEPROM.

Der Abschlusswiderstand wird über den Schalter eingestellt.

Node-ID, Baudrate und Abschlusswiderstand einstellen über Einstellschalter

Die Einstellung der Schalter ist dominant über alle anderen Einstellmöglichkeiten.



Figur 12: Rückansicht des Antriebs mit Dezimal-Drehschaltern S0 und S1 zur Einstellung der Node-ID sowie mit DIP-Schalter S2 zur Einstellung von Baudrate und Abschlusswiderstand.

Tabelle 11: Einstellung von Node-ID über Dezimal-Drehschalter S0 und S1 (Zuordnung, siehe Figur 12). Werkseinstellung ist 00d. Node-ID und Baudrate sind in diesem Fall definiert durch Objekt 2101h und 2100h. Gleichzeitig sind die Stellungen 1...3 von DIP-Schalter S2 (Tabelle 12) deaktiviert.

S1	S0	Node-ID
0	0	Objekt 2101h
0	1...9	1...9
1	0...9	10...19
2	0...9	20...29
3	0...9	30...39
4	0...9	40...49
5	0...9	50...59
6	0...9	60...69
7	0...9	70...79
8	0...9	80...89
9	0...9	90...99

Tabelle 12: Einstellung von Baudrate und Abschlusswiderstand über DIP-Schalter S2. Werkseinstellung ist 1...4=OFF. Stellungen 1...3 vom DIP-Schalter sind nur aktiv, wenn Einstellung Dezimal-Drehschalter S0 und S1 nicht 00d sind. Stellung 4 vom DIP-Schalter ist in jedem Fall aktiv.

S2				Baudrate [kBit/s]	Abschluss- widerstand
1	2	3	4		
OFF	OFF	OFF	-	Objekt 2100h	-
ON	OFF	OFF	-	20	-
OFF	ON	OFF	-	50	-
ON	ON	OFF	-	125	-
OFF	OFF	ON	-	250	-
ON	OFF	ON	-	500	-
OFF	ON	ON	-	800	-
ON	ON	ON	-	1000	-
-	-	-	OFF	-	offen
-	-	-	ON	-	120 Ω

Beim nächsten Initialisieren meldet sich der Antrieb mit neuer Node-ID und Baudrate auf dem Bus an.
ACHTUNG: Auch die Baudrate des Masters muss nun geändert werden, um den Antrieb anzusprechen.

Node-ID und Baudrate einstellen im EEPROM

Die Einstellung der Node-ID im EEPROM erfolgt über das Objekt 2101h. Dies ist nur möglich, wenn alle Einstellschalter sich in Werkseinstellung befinden (S0...S1 = 0, S2 1...3 = OFF).

Beispiel Node-ID auf 23h einstellen:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	2Fh	01h	21h	0h	23h	xx	xx	xx

Bestätigung:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	01h	21h	0h	0h	0h	0h	0h

Tabelle 13: Einstellung der Baudrate im EEPROM über Objekt 2100h. Hierbei wird ein Index ins Objekt geschrieben, nicht die effektive Baudrate (nur möglich, wenn Drehschalter in Werkseinstellung 00).
ACHTUNG: Die Indices für Objekt 2100h und LSS unterscheiden sich.

Baudrate [kBit/s]	20	50	100	125	250	500	800	1000
Index (Definition in Objekt 2100h)	1h	2h	3h	4h	5h	6h	7h	8h
Index (LSS) nach CiA-Tabelle	7h	6h	5h	4h	3h	2h	1h	0h

Beispiel: Einstellen der Baudrate auf 250 kBit/s (5h):

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	2Fh	00h	21h	0h	05h	Xx	Xx	Xx

Bestätigung:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	00h	21h	0h	0h	0h	0h	0h

Nun müssen die neuen Einstellungen über Objekt 1010h nichtflüchtig in Speicher (EEPROM) geschrieben werden. Dazu muss die Botschaft "save" in den Subindex 1 geschrieben werden.

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	10h	10h	01h	73 's'	61 'a'	76 'v'	65 'e'

Bestätigung:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	10h	10h	01h	0h	0h	0h	0h

Beim nächsten Initialisieren meldet sich der Antrieb mit neuer Node-ID und Baudrate auf dem Bus an.
ACHTUNG: Auch die Baudrate des Masters muss nun geändert werden, um den Antrieb anzusprechen.

Node-ID und Baudrate einstellen mit LSS

Wenn mehrere Teilnehmer am selben Bus die gleiche Node-ID und Baudrate besitzen, können diese zwei Objekte mit LSS umgestellt werden. Mit LSS wird der Antrieb über Produktcode, Revisionsnummer, Vendor-ID und Seriennummer angesprochen und konfiguriert (siehe Kapitel 2.2.10).

Beim nächsten Initialisieren meldet sich der Antrieb mit neuer Node-ID und Baudrate auf dem Bus an.
ACHTUNG: Auch die Baudrate des Masters muss nun geändert werden, um den Antrieb anzusprechen.

4.2 Objekte schreiben und lesen

Um ein Objekt (SDO) zu schreiben oder zu lesen werden immer zwei Telegramme ausgetauscht.

Objekt setzen

Zuerst sendet der Master den zu setzenden Wert. Anschliessend sendet der Antrieb die Bestätigung.

Master sendet Wert (wxyz):

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Dh	60h	1h	w	x	y	z

Antwort des Antriebs als Bestätigung:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Dh	60h	1h	0	0	0	0

Objekt lesen

Zuerst sendet der Master eine Aufforderung des gewünschten Objekts. Dann sendet der Antrieb den geforderten Wert.

Anfrage vom Master:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	40h	41h	60h	0	x	x	x	x

Antwort (abcd) des Antriebs auf die Anfrage:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	43h	41h	60h	0	a	b	c	d



Parameter speichern ins EEPROM

Mit dem Objekt 1010h können die in den Objekten enthaltenen Parameter nichtflüchtig in ein EEPROM gespeichert werden. Um unbeabsichtigtes Speichern zu verhindern, muss dabei der Befehl save in Objekt 1010h, Subindex 1, Byte 5 bis 8 geschrieben werden. Welches der Objekte mit welchem Speicherkommando gespeichert wird, ist spezifiziert in Tabelle 10, Spalte Save und Beschreibung zu Objekt1010h.

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600h+Node-ID	8	23	10	10	01	73 (= s)	61 (= a)	76 (= v)	65 (= e)

Default Parameter laden aus dem EEPROM

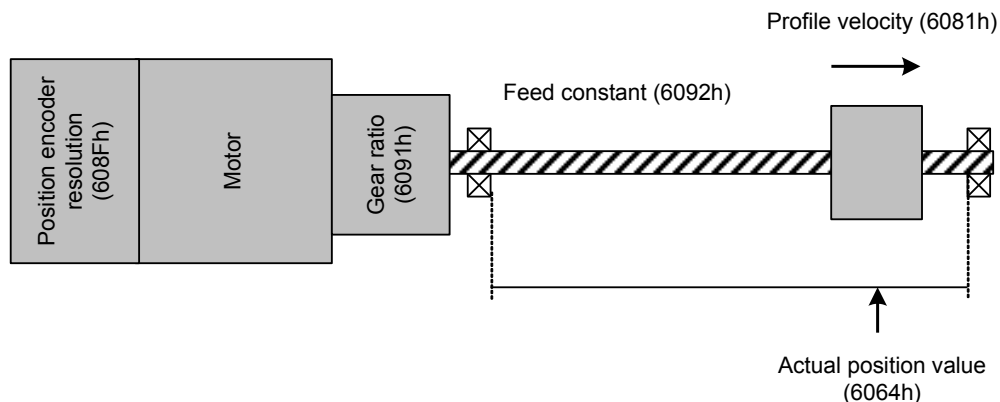
Mit dem Objekt 1011h können die Default-Werte der Objekt-Parameter aus dem EEPROM geladen werden. Um unbeabsichtigtes Laden zu verhindern, muss dabei der Befehl load in Objekt 1011h, Subindex 1, Byte 5 bis 8 geschrieben werden. Welches der Objekte mit welchem Ladekommando geladen wird, ist spezifiziert in Tabelle 10, Spalte Save und Beschreibung zu Objekt1011h.

ACHTUNG: Die aktuellen Werte im RAM werden bei Laden überschrieben.

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600h+Node-ID	8	23	11	10	01	6C (= l)	6F (= o)	61 (= a)	64 (= d)

4.3 Antriebsparameter konfigurieren

4.3.1 Notation, Dimension und Getriebe



Figur 13: Konfigurierung des Antriebs.

Benutzerdefinierte Positionseinheit (Position notation und Position dimension)

Die benutzerdefinierte Positionseinheit setzt sich zusammen aus Position notation (Objekt 6089h) und Position dimension (Objekt 608Ah). Position notation entspricht einem Vorfaktor, d.h. für Linearbewegungen z.B. mikro (10^{-6}), milli (10^{-3}), 1 (10^0) oder kilo (10^3). Für Drehbewegungen ist nur Position notation = 1 sinnvoll.

Die Position dimension entspricht der eigentlichen Positionseinheit, d.h. für Linearbewegungen Meter und für Drehbewegungen Encoder-Schritte, Grad, Minuten, Sekunden oder Radiant. Die benutzerdefinierte Positionseinheit ergibt sich als Multiplikation:

$$\text{Benutzerdefinierte Positionseinheit} = \text{Positionnotation} * \text{Positiondimension}$$

Beispiel: [Position] = milli * meter = Millimeter

Nach Definition der benutzerdefinierten Positionseinheit beziehen sich alle Positionsangaben zur Ansteuerung des Antriebs (ausser Objekt 6063h) nur noch auf diese Einheit und werden intern automatisch umgerechnet unter Berücksichtigung von Getriebeuntersetzung (Gear ratio) und der Spindel bzw. Kundengetriebe (Feed constant).

Dieses Prinzip gilt analog für die benutzerdefinierte Geschwindigkeitseinheit.

Position encoder resolution (Objekt 608Fh)

Dieses Objekt ist in 2 Subindexe unterteilt. Subindex 01h ist die Anzahl der Encoder-Schritte. Subindex 02h beschreibt beinhaltet die Anzahl Umdrehungen. Unser Sensor hat 8 Schritte pro Motor-Umdrehung, also ist Position encoder resolution = 8.

$$\text{Position encoder resolution} = \frac{\text{Encoder increments}}{\text{Motor revolutions}}$$

Gear ratio (Objekt 6091h)

In diesem Objekt wird in Subindex 01h und 02h das Getriebe-Untersetzungsverhältnis dargestellt. Wenn zum Beispiel das Untersetzungsverhältnis den Wert 61.25 hat, wird in Subindex 01h der Wert 6125 und in Subindex 02h der Wert 100 eingetragen. $6125/100 = 61.25$

$$\text{Gear ratio} = \frac{\text{Motor revolutions}}{\text{Gear shaft revolutions}}$$

Feed constant (Objekt 6092h)

Dieses Objekt definiert in Subindex 01h und 02h das Kundengetriebe. Zum Beispiel 400 Umdrehungen der Spindel ergeben einen Meter Weg (Linearumsetzung).

ACHTUNG: Die Wegeinheit für Feed ist immer Meter. Aus Vereinfachungsgründen wird kein Bezug zur eingestellten Position notation (6089h) oder Position dimension (608Ah) hergestellt.

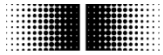
$$\text{Feed constant} = \frac{\text{Feed}}{\text{Spindle shaft revolutions}}$$

$$\text{Beispiel: Feed constant} = \frac{1 \text{ m}}{400 \text{ U}} = \frac{1000 \text{ mm}}{400 \text{ U}} = 2.5 \frac{\text{mm}}{\text{U}}$$

Position factor (Objekt 6093h)

$$\text{Positionfactor} = \text{Positionnotation} \cdot \text{Positiondimension} \cdot \text{Encoderresolution} \cdot \frac{\text{Gearratio}}{\text{Feed constant}}$$

$$\text{Beispiel: Positionfactor} = \frac{1}{1} \cdot \frac{1 \text{ m}}{1000 \text{ mm}} \cdot \frac{\frac{8 \text{ Schritte}}{1 \text{ U}} \cdot \frac{6125}{100}}{\frac{1 \text{ m}}{400 \text{ U}}} = 196 \frac{\text{Schritte}}{\text{mm}}$$

**Position actual value (Objekt 6064h)**

$$\text{Position actual value} = \frac{\text{Position actual steps}}{\text{Position factor}}$$

Beispiel: 100 mm (applikationsseitig) \Leftrightarrow 9'800 Encoder-Schritte (motorseitig)

$$\text{Beispiel: Maximalwert für Position actual value} = \frac{2^{31}-1}{196} \text{ mm} = 10'956'549 \text{ mm}$$

ACHTUNG: Falls als Positionseinheit Encoder-Schritte eingestellt ist, dann gilt automatisch Position factor = 1. Gear ratio oder Feed constant werden dann nicht in der Umrechnung mitberücksichtigt. Anhand obiger Formel kann ausserdem aus der maximalen Anzahl der Encoder-Schritte Positions actual steps ($2^{31}-1$) der Maximalwert für Position actual value berechnet werden (siehe Beispiel).

Benutzerdefinierte Geschwindigkeitseinheit (Velocity notation and Velocity dimension)

Die benutzerdefinierte Geschwindigkeitseinheit setzt sich zusammen aus Velocity notation (Objekt 608Bh) und Velocity dimension (Objekt 608Ch). Velocity notation entspricht einem Vorfaktor, d.h. für Linearbewegungen mikro (10^{-6}), milli (10^{-3}), 1 (10^0) oder kilo (10^3). Für Drehbewegungen ist nur Position notation = 1 sinnvoll.

Die Velocity dimension entspricht der eigentlichen Geschwindigkeitseinheit, d.h. für Linearbewegungen beispielsweise Meter pro Sekunde und für Drehbewegungen Encoder-Schritte pro Sekunde oder Grad pro Sekunde. Die benutzerdefinierte Positionseinheit ergibt sich als Multiplikation:

$$\text{Benutzerdefinierte Geschwindigkeitseinheit} = \text{Velocity notation} \cdot \text{Velocity dimension}$$

$$\text{Beispiel: [Geschwindigkeit]} = \text{milli} \cdot \text{meter/s} = \text{Millimeter/s}$$

Nach Definition der benutzerdefinierten Geschwindigkeitseinheit beziehen sich alle Geschwindigkeitsangaben zur Ansteuerung des Antriebs nur noch auf diese Einheit und werden intern automatisch umgerechnet unter Berücksichtigung von Getriebeuntersetzung (Gear ratio) und der Spindel bzw. Kundengetriebe (Feed constant).

Velocity factor (Objekt 6094h)

$$\text{Velocity factor} = \text{Velocity notation} \cdot \text{Velocity dimension} \cdot \text{Encoder resolution} \cdot \frac{\text{Gear ratio}}{\text{Feed constant} \cdot \text{Encoder resolution}}$$

$$\text{Beispiel: Velocity factor} = \frac{1}{60} \cdot \frac{1}{1000} \cdot \frac{\frac{8 \cdot 6125}{1 \cdot 100}}{\frac{1 \cdot 8}{400 \cdot 1}} = 0.408$$

Velocity actual value (Objekt 606Ch) und Profile velocity (Objekt 6081h)

$$\text{Velocity actual value} = \frac{\text{Motordrehzahl in U/s}}{\text{Velocity factor}}$$

$$\text{Profile velocity} = \frac{\text{Motordrehzahl in U/s}}{\text{Velocity factor}}$$

$$73 \text{ mm/min (applikationsseitig)} \Leftrightarrow 30 \text{ U/s (motorseitig)}$$

ACHTUNG: Falls als Geschwindigkeitseinheit Encoder-Schritte / Sekunde eingestellt ist, dann gilt automatisch Velocity factor = 1/8. Gear ratio oder Feed constant werden dann nicht in der Umrechnung mitberücksichtigt.

4.3.2 Drehrichtung (Polarity)

Im Objekt 607Eh kann die Drehrichtung definiert werden.

4.3.3 Strom und Geschwindigkeit

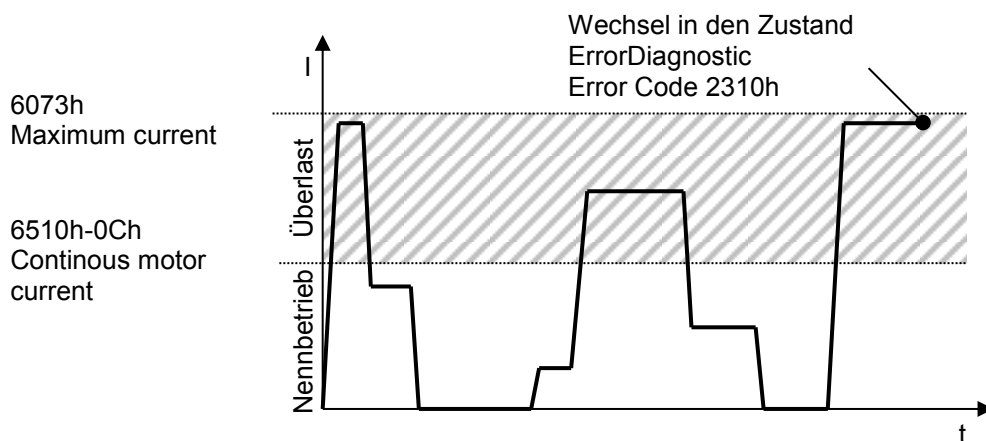
Die Drehmoment-Begrenzung des Antriebs erfolgt über die Begrenzung des maximalen Motorstroms (siehe Objekt 6073h). Es kann ein Wert zwischen 0 mA und dem werksseitig eingestellten maximalen Motorstrom (siehe Objekt 6510h-7h) ins Objekt geschrieben werden. Der Wert 0 entspricht der maximal zulässigen Strombegrenzung. Werte zwischen 1 mA und dem maximalen Motorstrom sorgen für eine (Drehmoment-) Begrenzung.

Die Geschwindigkeiten werden wie die Positionen in benutzerdefinierten Einheiten angegeben. Somit wird die Dimension und Notation (Objekte 608Bh, 608Ch) der Geschwindigkeitsangabe als auch die Getriebe- und Encoder-Einstellungen (Objekte 608Fh, 6091h, 6092h) verrechnet.

4.3.4 I^2t -Begrenzung

Die I^2t -Strombegrenzung erlaubt es, den Antrieb kurzzeitig oberhalb seiner Nennleistung zu betreiben. Die Überlastung kann wiederholt werden, solange auch wieder Ruhephasen eingelegt werden, in denen der Antrieb mit geringerer Leistung betrieben wird und sich abkühlen kann.

Übersteigt der gemessene Strom den zulässigen Dauermotorstrom (Objekt 6510-0Ch Continuous motor current) für eine abhängig von der Stromhöhe definierte Zeit, wird der anliegende Fahrbefehl abgebrochen und in den Zustand ErrorDiagnostic gewechselt (Fehler Continuous over current, Error Code 2310h). Wird der Antrieb überlastet und anschliessend höchstens mit dem zulässigen Dauerstrom betrieben, wird die Überlastung wieder abgebaut.



Figur 14: Stromverlauf bis zur I^2t -Begrenzung.

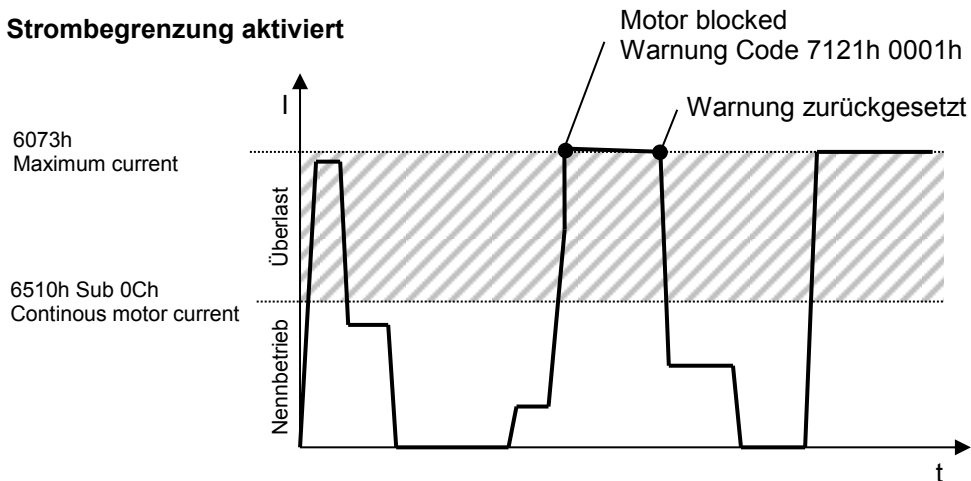
Der aktuelle Wert der Überlastung kann aus dem Objekt 2112h-06h I^2t Overload Level ausgelesen werden. Er gibt den relativen Grad der aktuellen Überlastung bezogen auf die maximal erlaubte Überlastung an.



4.3.5 Strombegrenzung

Bei aktivierter Strombegrenzung lässt der Regler keinen höheren Strom als den vom Anwender bestimmbaren Maximalstrom (Objekt 6073h Maximum current) zu. Wenn der Antrieb auf einen Widerstand fährt, bringt er ein konstantes Moment auf. Wenn Strombegrenzung anspricht und Antrieb sich nicht bewegt, dann wird eine EMCY-Meldung Warnung Motor blockiert (Objekt 7121h 0001h) ausgegeben

Strombegrenzung aktiviert

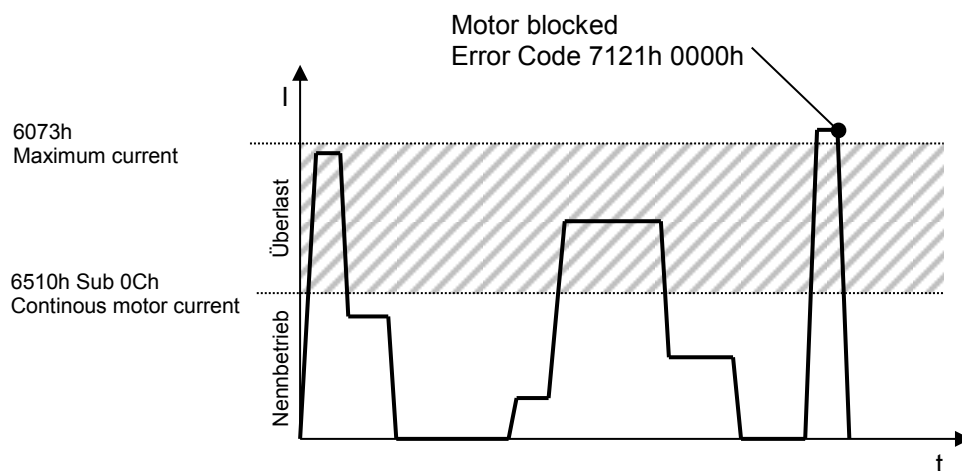


Figur 15: Stromverlauf bei Strombegrenzung aktiviert.

ACHTUNG: Bei aktivierter Strombegrenzung kann die Lebensdauer des bürstenbehafteten Motors in kürzerer Zeit erreicht werden als mit deaktivierter Funktion.

Bei deaktivierter Strombegrenzung wird der anliegende Fahrbefehl abgebrochen, wenn der Strom Maximum current (Objekt 6073h) während typ. 100 ms überschreitet und es wird ein Error motor blocked (Error Code 7121h 0000h) abgesetzt.

Strombegrenzung deaktiviert



Figur 16: Stromverlauf bei Strombegrenzung deaktiviert.

Per Default ist die Strombegrenzung deaktiviert. Sie lässt sich in Version control (Objekt 2110h Bit 19) aktivieren.

4.3.6 Positioning Timeout

Falls dieser Wert nicht 0 ist, wird ein Fahrauftrag abgebrochen, wenn er länger als die angegebene Zeit in ms benötigt.

4.3.7 Spielausgleich

Im Objekt 2111h Subindex 1 kann angegeben werden, wie das Ziel angefahren werden soll. Entweder wird das Ziel direkt, von oben oder von unten angefahren. Zusätzlich wird in Subindex 2 definiert, wie weit der Antrieb zum Spielausgleich über die Zielposition hinausfahren soll.

Beispiel: Der Antrieb befindet sich auf aktueller Position 1000. Die Zielposition ist 2000. Wenn nun „Zielposition direkt anfahren“ aktiviert ist, dann wird die Zielposition von unten her angefahren. Ist dagegen „Zielposition von oben anfahren“ aktiviert, dann fährt der Antrieb zuerst auf Zwischenposition 2100 (bei Backlash delta 100) und dann von oben nach unten auf Zielposition 2000.

Tabelle 14: Beispiele für Antriebsparameter.

Parameter	Objekt	Subindex	Wert	Beschreibung
Position notation	6089h	0	FDh	milli (*10 ⁻³)
Position dimension	608Ah	0	01h	Meter (Längsbewegung)
Velocity notation	608Bh	0	FDh	milli (*10 ⁻³)
Velocity dimension	608Ch	0	A6h	Meter / min
Encoder increments	608Fh	1	04h	8 Encoder-Schritte pro Umdrehung
Motor revolutions	608Fh	2	01h	
Gear ratio – motor revolutions	6091h	1	17EDh	61.25 = 6125 / 100
Gear ratio – shaft revolutions	6091h	2	64h	
Feed constant - feed	6092h	1	01h	2,5 mm / Umdrehung
Feed constant – shaft revolutions	6092h	2	190h	= 1000 mm / 400 Umdrehungen = 1 Meter / 400 Umdrehungen

4.3.8 Einstellungen speichern

Über das Objekt 1010h wird das Speichern in den nicht flüchtigen Speicher (EEPROM) ausgelöst. Um ein unabsichtliches Speichern zu verhindern muss die Botschaft „save“ in den Subindex 1 geschrieben werden.

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	10h	10h	1h	73 ,s'	61 ,a'	76 ,v'	65 ,e'

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	10h	10h	1h	0	0	0	0

4.4 Antriebsposition referenzieren (Homing)

Der Antrieb benötigt einen Bezugspunkt zur Anlage oder Applikation. Hierzu muss der Antrieb im Normalfall einmalig referenziert werden. Nach dem Einschalten befindet sich der Antrieb im Zustand Ready. Das Referenzieren erfolgt im Mode of operation Homing mit dem Objekt 6098h. Der Wechsel des Mode of operation erfolgt durch:

1. Objekt 6060h (Modes of operation) auf 6=Homing setzen
2. Objekt 6040h (Kontrollwort) auf 000Fh setzen
SwitchOn, EnableVoltage, QuickStop und EnableOperation = 1

Anlagen-Bezugspunkt anfahren

Es gibt verschiedene Möglichkeiten zum Setzen der Referenzposition:

1. Herantasten an Bezugspunkt über Jogging, d.h. Ansteuerung der digitalen Hardware-Schalteingänge
 - a. Externe Anschlüsse im Objekt 211h, Subindex 02h als Schalter für Jogging+/- definieren.
2. Herantasten an Bezugspunkt durch Vorgabe von Zielpositionen über den CAN-Bus (relativ / absolut)
 - a. Zielposition vorgeben in benutzerdefinierter Einheit (siehe Objekt 607Ah)
 - b. Relativ oder absolut in Objekt 6040h, Bit6 setzen
 - c. Fahren mit Objekt 6040h, Bit4 (HomingStart)
3. Fahren auf Anschlag (Anschlag als Anlagen-Bezugspunkt)
 - a. Strombegrenzung mit Objekt 6073h auf Minimum setzen
 - b. Mit Objekt 6098h, Wert -40, -41 hin- und herfahren
 - c. ACHTUNG: Durch die Getriebeuntersetzung können sehr grosse Kräfte wirken.
4. Fahren bis externer Referenzschalter anspricht (Position des Referenzschalters als Bezugspunkt)
 - a. Externen Referenzschalter an Referenzposition anbringen
 - b. Externen Hardware-Schalteingang im Objekt 211h, Subindex 02h als Hardware-Endschalter definieren
 - c. Mit Objekt 6098h, Wert -30, -31 hin- und herfahren
 - d. Nachdem die Referenzposition gefunden und gespeichert ist, kann der externe Referenzschalter demontiert werden

Referenzposition direkt setzen

ACHTUNG: Das direkte Setzen der Referenzposition kann gefährlich sein. Bei Eingabe einer inkorrekten Referenzposition ist anschliessend die Funktion der Endschalter wirkungslos. Der Antrieb kann dann gegen einen Anschlag fahren.

Aus diesem Grunde wurde von der CiA ein spezielles Vorgehen definiert. Hierbei muss zunächst die Zielposition auf die zu setzende Referenzposition geändert werden. Danach wird die aktuelle Position des Antriebs mit Objekt 6098, Wert -12 mit der Zielposition (Target position) überschrieben.

Beispiel: Setze Position des Antriebs an der aktuellen Stelle auf 100 mm.

1. Setze Antrieb in den Zustand HomingReady
2. Setze Objekt 607Ah (Target position) auf 100 (Beispiel: Antrieb arbeitet in mm)
3. Setze nächste Referenzierung mit Objekt 6098 (Homing method) auf -12
4. Starte Referenzierung mit Objekt 6040h (Kontrollwort) = 001Fh

Tabelle 15: Übersicht über Referenzierungs-Methoden (Objekt 6098h) im Zustand HomingReady.

Objekt 6098h		Beschreibung
0	0h	Keine Referenzierung aktiv
-10	F6h	Setze aktuelle Position als unteren Software-Endschalter
-11	F5h	Setze aktuelle Position als oberen Software-Endschalter
-12	F4h	Referenzieren mit Presetwert
-13	F3h	Referenzieren mit Einmessmethode
-20	ECh	Fahr zu unterem Software Endschalter
-21	EBh	Fahr zu oberem Software Endschalter
-22	EAh	Fahr zu Position Null
-30	E2h	Referenzieren auf unteren Referenzschalter
-31	E1h	Referenzieren auf oberen Referenzschalter
-40	D8h	Referenzieren auf unteren Anschlag
-41	D7h	Referenzieren auf oberen Anschlag

Mit einem Start-Befehl kann man die gewünschte Auswahl aktivieren bzw. starten. Im Zustand PositioningReady kann das Ziel gesetzt und angefahren werden (siehe Kapitel 2.4.3).

4.4.1 Position oder Referenzposition direkt setzen

Antrieb wird in den Zustand **HomingReady** gesetzt:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0
600h+Node-ID	5	2Fh	60h	60h	0	06h

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	60h	60h	0	0	0	0	0

Es wird das Kommando 0Fh ins Kontrollwort geschrieben:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	0Fh	0	0	0

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

Setzen der Homing-Methode Referenzieren mit Presetwert: Weise Zielposition der aktuellen Position zu.

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0
600h+Node-ID	5	2Fh	98h	60h	0	F4h

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	98h	60h	0	0	0	0	0

Ziel (bzw. 0) setzen:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Ah	60h	0	x	x	x	x

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Ah	60h	0	0	0	0	0

Es wird ein Start ins Kontrollwort geschrieben:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	1Fh	0	0	0

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

Sobald aktuelle Position neu gesetzt ist, wird das Bit14 HomingOK im Statuswort (Objekt 6041h) gesetzt.

Antrieb wird in den Zustand **PositioningReady** gesetzt:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0
600h+Node-ID	5	2Fh	60h	60h	0	01h

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	60h	60h	0	0	0	0	0

Es wird das Kommando 0Fh ins Kontrollwort geschrieben:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	40h	60h	0	0Fh	0	0	0

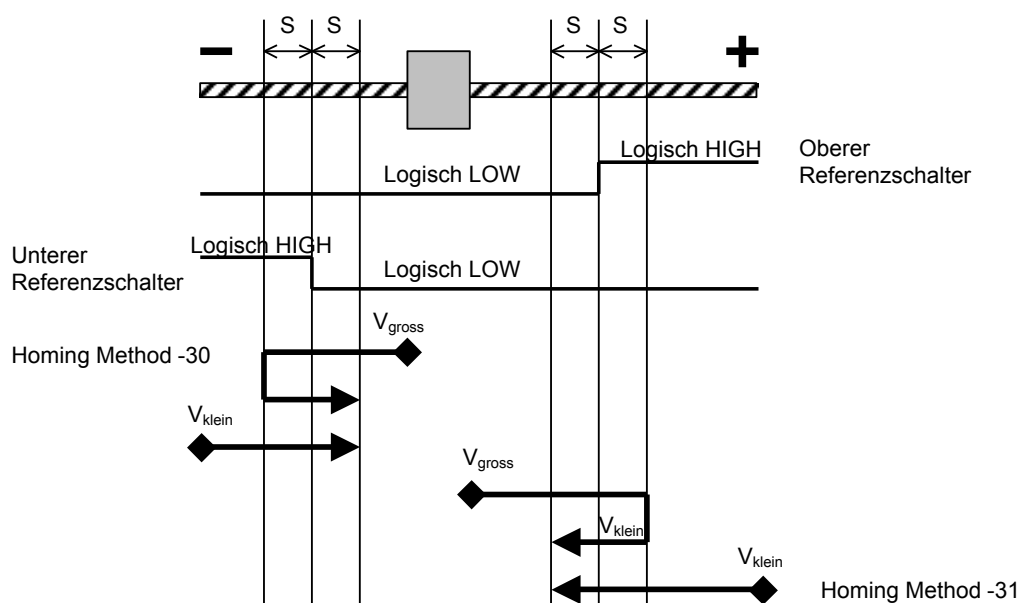
Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	40h	60h	0	0	0	0	0

Nun befindet sich der Antrieb wieder im Zustand **PositioningReady** und ist bereit für Fahraufträge.

4.4.2 Referenzieren auf Referenzschalter

Beim Übergang des Referenzschalters von logisch HIGH auf LOW wird die Referenzposition (Objekt 2111h-11h) der aktuellen Position (Objekt 6064h) zugewiesen. Sobald im Statuswort 6041h das Bit14 HomingOK gesetzt wurde, ist die Referenzierung abgeschlossen.



S: Free referencing distance (Objekt 2111-1Eh)
V_gross Referenzschalter suchen mit Profile velocity (Objekt 6081h)

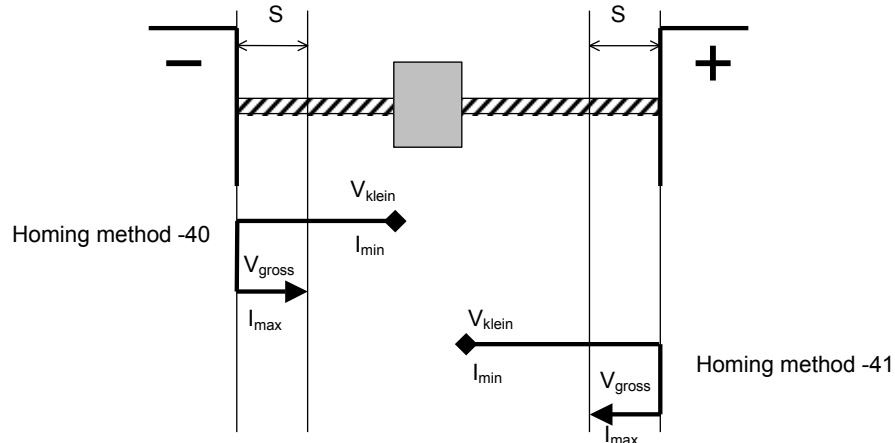
Figur 17: Referenzieren auf unteren bzw. oberen Referenzschalter.

Achtung:

Es darf nur ein einziger digitaler Eingang als Referenzschalter konfiguriert sein.

4.4.3 Referenzieren auf Anschlag

Der Antrieb fährt mit Homing speed auf den Anschlag. Wenn der Motor blockiert ist, wird die Referenzposition (Objekt 2111h-11h) der aktuellen Position (Objekt 6064h) zugewiesen. Anschliessend wird der Antrieb mit dem maximal zulässigem Strom um den Weg s frei gefahren. Sobald im Statuswort 6041h das Bit14 HomingOK gesetzt wurde, ist die Referenzierung abgeschlossen.



Anschlag suchen mit

V_{klein} kleiner Geschwindigkeit (Homing speed, Objekt 6099-1h)
 I_{min} kleinem Strom (Max current, Objekt 6073h)

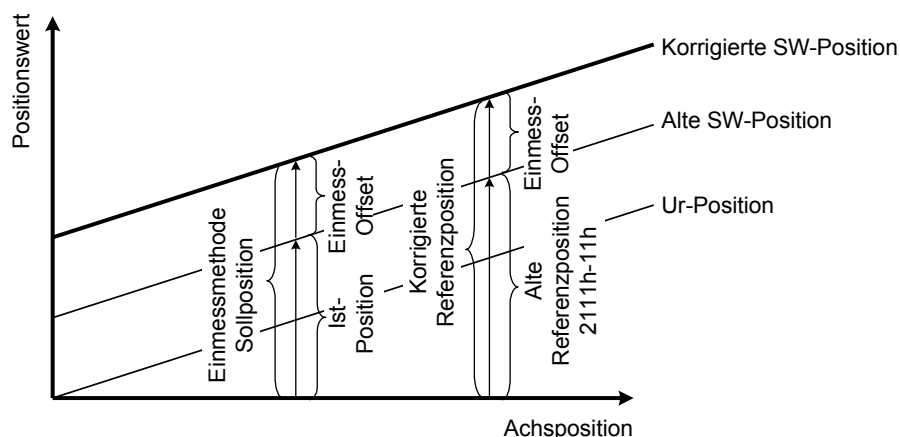
Freifahren mit

V_{gross} maximaler Geschwindigkeit (Profile velocity, Objekt 6081h)
 I_{max} losbrech Strom (Break away current, Objekt 2113h-20h werkseitig)
 S : Freifahrweg (Free referencing distance, Objekt 2111-1Eh)

Figur 18: Referenzieren auf Anschlag in Abwärtsrichtung oder in Aufwärtsrichtung.

4.4.4 Referenzieren mit Einmessmethode

Voraussetzung für diese Homing-Methode ist, dass Antrieb bereits referenziert wurde. Somit muss das Bit14 im Statuswort (Objekt 6041) HomingOK bereits gesetzt sein. Beim Starten dieser Homing-Methode wird zunächst der Einmess-Offset bestimmt. Anschliessend wird die Ist-Position mit der Zielposition (Target position) neu gesetzt. Zusätzlich wird die Referenzposition (2111h-11h) mit dem Einmess-Offset korrigiert. Nach Abschluss der Einmessmethode wird das Bit8 im Statuswort (6041h) CalibrationOK gesetzt.



Figur 19: Referenzieren mit der Einmessmethode.

4.4.5 Zurücksetzen der Statusbits HomingOK und CalibrationOK

Durch folgende Aktionen werden die Statusbits HomingOk (Bit14) und CalibrationOK (Bit8) auf Null zurück gesetzt:

- Wenn während eines Fahrauftrags die Betriebsspannung ausgeschaltet wird. Liegt hierbei noch die getrennte Betriebsspannung Elektronik +VsE an, so werden HomingOK und CalibrationOK nicht zurückgesetzt.
- Wenn im Servicebetrieb ohne Buskommunikation verfahren wird.
- Nach dem Start folgender Homing-Methoden:

Objekt 6098h		Beschreibung
-12	F4h	Referenzieren mit Presetwert (überschreibe aktuelle Position mit Zielposition)
-13	F3h	Referenzieren mit Einmessmethode
-30	E2h	Referenzieren auf unteren Referenzschalter (Fahr zu unterem Referenzschalter)
-31	E1h	Referenzieren auf oberen Referenzschalter (Fahr zu oberem Referenzschalter)
-40	D8h	Referenzieren auf unteren Anschlag (Fahr zu unterem Anschlag)
-41	D7h	Referenzieren auf oberen Anschlag (Fahr zu oberem Anschlag)

- Nach dem Überschreiben folgender Objekte:

Objekt	Beschreibung
607Eh-00h	Polarity
6091h-01h	Gear - Motor revolutions
6091h-02h	Gear - Gear shaft revolutions
6089h-00hh	Position notation
608Ah-00h	Position dimension
608Fh-01h	Position encoder resolution - Encoder increments
608Fh-02h	Position encoder resolution - Encoder revolutions
6092h-01h	Feed constant - Feed
6092h-02h	Feed constant - Spindle shaft revolutions

4.5 Positionen der Software-Endschalter setzen

Nachdem die Referenzposition gesetzt ist, können die Positionen der Software-Endschalter definiert werden. Die Positionen dieser Endschalter beziehen sich immer auf den Offsetwert. Werden Referenzposition oder Offset verschoben, werden auch die Positionen der Software-Endschalter mit verschoben.

Software-Endschalter anfahren

Die Positionen der Software-Endschalter können mit den gleichen Mitteln wie die Referenzposition angefahren werden.

Software-Endschalter setzen

Nachdem sich Antrieb an der Position eines Endschalters befindet, kann diese mit dem Objekt 6098h, Wert – 10 oder –11 als Position für den unterer bzw. oberer Software-Endschalter übernommen werden.

Position der Software-Endschalter direkt eingeben

Als zusätzliche Variante können die Positionen der Software-Endschalter mit dem Objekt 607Dh, Subindex1 oder 2, direkt eingegeben werden.

Position für unteren Software-Endschalter direkt setzen:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Dh	60h	1	x	x	x	x

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Dh	60h	1	0	0	0	0

Position für oberen Software-Endschalter direkt setzen:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+Node-ID	8	23h	7Dh	60h	2	x	x	x	x

Antwort:

COB-ID	DLC	Kommando	Objekt L	Objekt H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+Node-ID	8	60h	7Dh	60h	2	0	0	0	0

4.6 Fahraufträge ausführen

Nachdem die Parameter definiert sind, kann ein Fahrauftrag erfolgen. Dazu muss zunächst ein Wechsel vom Zustand Ready zum Zustand PositioningReady erfolgen durch:

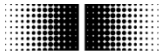
1. Objekt 6060h (Modes of operation) auf 1 = Positioning setzen
2. Objekt 6040h (Kontrollwort) auf 000Fh setzen

SwitchOn, EnableVoltage, QuickStop und EnableOperation = 1

Anschliessend kann eine Zielposition vorgegeben werden durch Beschreiben des Objekts 607Ah relativ oder absolut (Bit6 des Objektes 6040h).

Zusätzliches Setzen von Bit4 im Objekt 6040h löst anschliessend den Fahrauftrag aus.

Das Setzen von Bit8 im Objekt 6040h stoppt den Antrieb (Zwischenstopp). Erneutes Fahren erfolgt durch Rücksetzen dieses Bits. Anschliessend wird die zuvor vorgegebene Zielposition angesteuert.

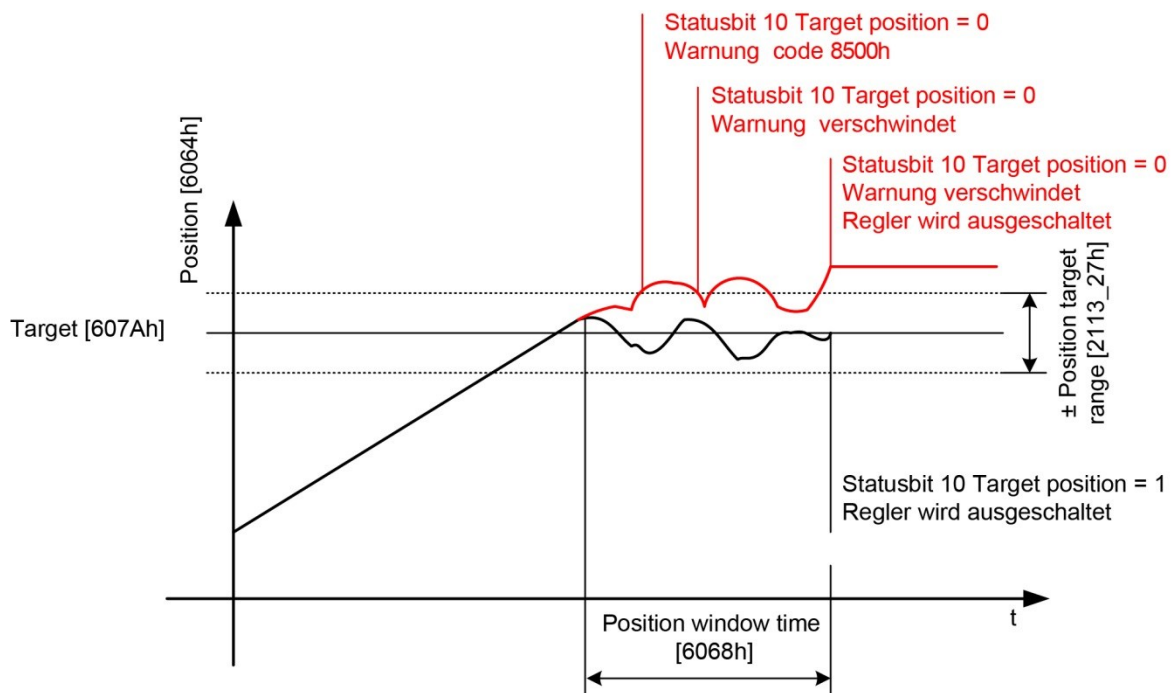


4.7 Reglerabschaltung nach Positionierung

Die Abschaltung der Regler nach der Positionierung schont die Bürsten des Elektromotors verlängert damit die Lebensdauer.

Diese Funktion dient dazu, ein automatisches Nachregeln ins Zielfenster zu verhindern, falls der Antrieb aus dem Zielfenster bewegt wird. Nach Erreichen des Ziels (Antrieb befindet sich im Zielfenster Objekt 2113h-27h und Geschwindigkeit < 2 U/s) wird die Zeit *Position window time* (Objekt 6068h) abgewartet, bevor das Statusbit 10 (*Target reached*) gesetzt und der Regler abgeschaltet wird. Anschliessend wird nicht mehr nachgeregelt.

Per Default ist diese Funktion deaktiviert (Objekt 6068h = 0h). Die Funktion wird aktiviert, sobald eine *Position window time* > 0 eingestellt wird (Objekt 6068h > 0 h in μ s).



Figur 20: Verhalten des Antriebs, wenn er aus dem Zielfenster bewegt wird:

- a) Vor Abschalten des Reglers:
 - Warnung Ausserhalb Zielfenster (ErrorCode 8500h)
 - Antrieb fährt wieder ins Zielfenster (die Position window time wird nicht neu gestartet)
 - Falls nach Ablauf der Position window time das Zielfenster nicht mehr erreicht wurde, bleibt das Statusbit 10 (Target reached) auf 0, verschwindet die Warnung Ausserhalb Zielfenster (ErrorCode 8500h) und der Regler wird abgeschaltet.
- b) Nach Abschalten des Reglers:
 - Antrieb fährt nicht mehr ins Zielfenster
 - Statusbit 10 (Target reached) wird wieder auf 0 zurückgesetzt

4.8 Automatische Deblockierung

Diese Funktion dient dazu, eine Blockade zu erkennen und selbstständig zu lösen.

Unter folgenden Bedingungen wird eine Blockade erkannt:

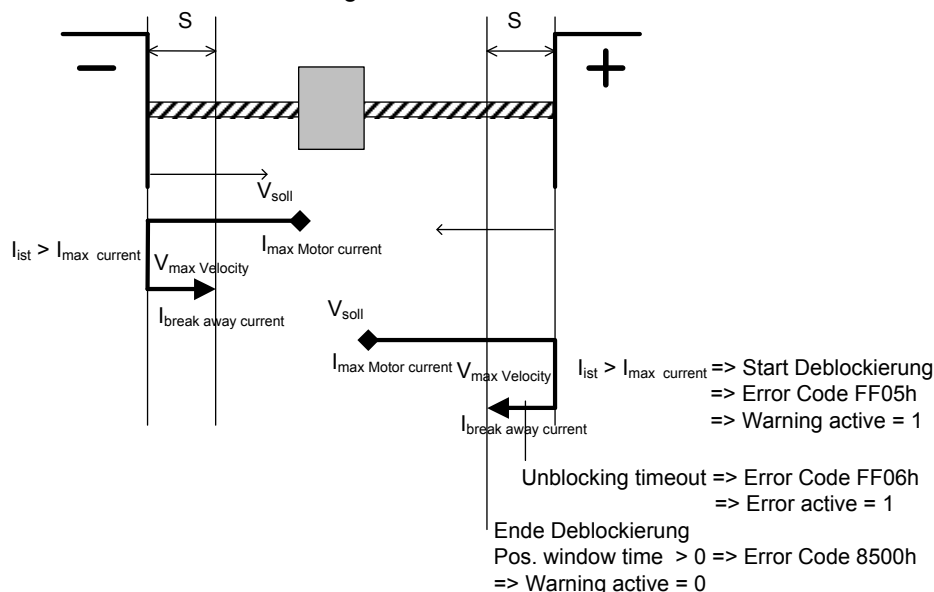
- Aktueller Strom ist grösser als Objekt 6073h Max current.
Der aktuelle Strom wird mit einer fixen Filterzeit gefiltert, damit die Deblockierfunktion nicht zu sensitiv reagiert.
- Aktuelle Geschwindigkeit (Objekt 606Ch Velocity actual value) ist kleiner als 2 U/sec

Ablauf nach Erkennen der Blockade:

1. Bit 7 Statuswort wird gesetzt (Warning active). Warnung motor blocked (Error Code FF05h) wird aktiviert.
2. Antrieb fährt in gegengesetzter Richtung mit dem Losbrechstrom (Objekt 2113h-20h) um den Freifahrweg S (Objekt 2111h-1Eh) frei. Aus Sicherheitsgründen wird während der Deblockierfahrt wiederum der Strom überwacht, um eine erneute Blockierung zu detektieren und anzuhalten. Die Sensitivität kann mittels der Filterzeit Current Time filter (Objekt 2111h-05) eingestellt werden. Aus Sicherheitsgründen sollte die Filterzeit jedoch möglichst klein gehalten werden.

Benötigt eine Deblockierung mehr Zeit als Unblocking timeout (Objekt 2111h-04h), so wird sie abgebrochen und es wird eine EMCYMeldung mit (Error Code FF06h) ausgegeben.

Nach Abschluss des Freifahrens wird bei Position window time (Objekt 6968h) > 0 die Warnung Positioncontroller (Error Code 8500h) abgesetzt. Nach Ablauf der Position window time wird Bit 7 im Statuswort auf 0 gesetzt (Warning inactive). Ist die Position window time = 0, wird die Warnung nach Abschluss der Blockade zurückgesetzt.



Figur 21: Ablauf Deblockierung

Aktivierung Deblockierfunktion:

Version Control Objekt 2110h Bit[18] = 0x00040000

Erkennung Blockade:

```
list > I_max current, (Objekt 6073h)
```

Drehzahl <2 U/sec. (Velocity actual valueObjekt 606Ch)

Freifahren mit:

V_{\max} velocity maximaler Geschwindigkeit (max velocity, Objekt 2113h-Eh werkseitig)

I_{break away current} Losbrech Strom (Break away current, Objekt 2113h-20h werkseitig)

S: Freifahrweg (Free referencing distance, Objekt 2111-1Eh)

Unblocking Timeout	Timeout für Deblockierung in ms, Objekt 2111-4h
--------------------	---

Current Time Filter Messrate in ms, Objekt 2111-5h



Timestamp	RTR Type	Node	Description	COBID	RAW-Data
036.796.486	SDO REQ	0x03	Controlword / 0x6040-00 Send Data 0x001F	0x603	ID: 0x603 / [CANData] 0x2B4060001F00
036.816.900	SDO RESP	0x03	Controlword / 0x6040-00 Send Data OK	0x583	ID: 0x583 / [CANData] 0x60406000'00000000'
036.905.400	TXPDO 1	0x03	0x055237	0x183	ID: 0x183 / [CANData] 0x375205
037.469.700	TXPDO 2	0x03	0xFFFF0312	0x283	ID: 0x283 / [CANData] 0x1203FFFF'
037.730.200	TXPDO 2	0x03	0xFFFF06FA	0x283	ID: 0x283 / [CANData] 0xFA06FFFF'
037.990.700	TXPDO 2	0x03	0xFFFF08DC	0x283	ID: 0x283 / [CANData] 0xDC08FFFF'
038.251.300	TXPDO 2	0x03	0xFFFF0DD0	0x283	ID: 0x283 / [CANData] 0xD00DFFFF'
038.875.800	EMER	0x03	Error code: 0xFF05 Register: 0x00 Alarms: 0x0000 Warnings: 0x0000	0x083	ID: 0x83 / [CANData] 0x05FF0000'00000000'
038.970.000	TXPDO 1	0x03	0x1D42B7	0x183	ID: 0x183 / [CANData] 0xB7421D
039.293.700	TXPDO 2	0x03	0xFFFF09E8	0x283	ID: 0x283 / [CANData] 0xE809FFFF'
039.486.200	TXPDO 1	0x03	0x1E52B7	0x183	ID: 0x183 / [CANData] 0xB7521E
039.554.400	TXPDO 2	0x03	0xFFFF0024	0x283	ID: 0x283 / [CANData] 0x2400FFFF'
039.814.800	TXPDO 2	0x03	0xFFFFF372	0x283	ID: 0x283 / [CANData] 0x72F3FFFF'
040.075.400	TXPDO 2	0x03	0xFFFFE4CC	0x283	ID: 0x283 / [CANData] 0xCCE4FFFF'
040.336.000	TXPDO 2	0x03	0xFFFFED720	0x283	ID: 0x283 / [CANData] 0x20D7FFFF'
040.596.500	TXPDO 2	0x03	0xFFFFEC974	0x283	ID: 0x283 / [CANData] 0x74C9FFFF'
040.857.100	TXPDO 2	0x03	0xFFFFBDBC	0x283	ID: 0x283 / [CANData] 0xBCBDFEFF'
041.117.700	TXPDO 2	0x03	0xFFFFB3F8	0x283	ID: 0x283 / [CANData] 0xF8B3FFFF'
041.270.900	EMER	0x03	Error code: 0x8500 Register: 0x00 Alarms: 0x0000 Warnings: 0x0000	0x083	ID: 0x83 / [CANData] 0x08500000'00000000'
041.378.200	TXPDO 2	0x03	0xFFFFEAD22	0x283	ID: 0x283 / [CANData] 0x22ADFFFF'
041.550.800	TXPDO 1	0x03	0x0242B7	0x183	ID: 0x183 / [CANData] 0xB74202
042.271.100	EMER	0x03	Error code: 0x0000 Register: 0x00 Alarms: 0x0000 Warnings: 0x0000	0x083	ID: 0x83 / [CANData] 0x00000000'00000000'
042.583.100	TXPDO 1	0x03	0x024237	0x183	ID: 0x183 / [CANData] 0x374202

☐ Jump To End ☒ Running MSBA_xxCANopenToolsuite.eds Export Trace File Import Trace File Clear Trace

Figur 22: Trace Aufzeichnung Blockierung

5 Fehlerdiagnose

5.1 Feldbus-Kommunikation

1. Falls der Antrieb über den CAN-Bus nicht angesprochen werden kann, sollten zuerst alle elektrischen Anschlüsse überprüft werden.

Sind die Anschlüsse in Ordnung, sollte als nächstes der Feldbusbetrieb getestet werden. Dazu wird ein CAN-Monitor benötigt, welcher die CAN-Kommunikation aufzeichnet und die Telegramme darstellt.

2. Nun sollte der Antrieb beim Aus- und wieder Einschalten der Spannungsversorgung eine BootUp-Meldung absetzen.

Sollte keine BootUp-Meldung erscheinen, so prüfen Sie, ob die Baudrate des Antriebs mit der Baudrate des CAN-Monitors übereinstimmt.

3. Wenn Sie Schwierigkeiten haben, die Verbindung zu einem Teilnehmer aufzubauen, prüfen Sie die Node-ID und die Baudrate.

Die Baudrate muss überall gleich eingestellt werden. Die Node-ID muss zwischen 1 und 127 liegen. Jeder Busteilnehmer muss eindeutig mit einer Node-ID definiert werden, d.h. es darf auf keinen Fall mehr als ein Busteilnehmer dieselbe Node-ID besitzen.

5.2 LED Statusanzeige

Auf der Rückseite des Antriebs befindet sich eine Duo LED zur Anzeige des Antriebszustands (Verschlussdeckel entfernen).

- Permanent grünes Licht zeigt korrekten Betrieb ohne Warnung oder Fehler.
- Grünes Blinken zeigt den NMT-Zustand Pre-Operational an.
- Permanentes oranges Licht zeigt an, dass sich der Antrieb im Bootloader Zustand befindet.
- Rotes Blinken zeigt eine aktive Warnung an.
- Permanentes rotes Licht zeigt an, dass ein Fehler aktiv ist.

Zur genauen Auswertung der anliegenden Fehler und Warnungen wird dringend empfohlen, die entsprechenden Objekte auszulesen (siehe Kapitel 5.3).

5.3 Fehlerdiagnose am Antrieb

Der Antrieb verfügt über mehrere Objekte und Meldungen, welche seinen Status oder Fehlerzustände umschreiben:

- Error register (Objekt 1001h): Dieses Objekt ist ein Register für den Fehlerstatus des Antriebs.
- Objekt 1003h: In diesem Objekt werden die letzten acht Fehlercodes und Warnungen gespeichert.
- Objekt Emergency (80h + Node-ID): Hoch prioritäre Fehlermeldung eines Teilnehmers mit Error code und Error register.
- SDO Abort Message: Falls die SDO-Kommunikation nicht korrekt abläuft, enthält die SDO-Antwort einen Abort Code.

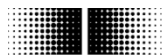
Tabelle 16: Error register (Objekt 1001h) in Emergency-Meldung Byte 2 (Byte 3...7 werden nicht verwendet).

Error register	Beschreibung
Bit 2 = 1	Spannungs-Fehler
Bit 3 = 1	Temperatur-Fehler
Bit 4 = 1	CAN-Bus Kommunikations-Fehler
Bit 5 = 1	Gerätespezifischer Fehler



Tabelle 17: Error code Definition für Emergency-Meldung Byte 0...1 und Byte 3...4 (siehe Kapitel Emergency-Dienst). Für zuletzt aufgetretenen Fehler, siehe Objekt 603Fh. Für letzte acht Fehler, siehe Objekt 1003h.

Error Code Byte 0...1	Zusatz Info. Byte 3...4	Fehler / Warnung	Beschreibung / Massnahme
0000h	0000h	Kein Fehler	-
2310h	0000h	Fehler I ² t andauernder Überstrom	Antrieb wurde aufgrund einer andauernden Überlastung (I ² t) gestoppt. Last überprüfen. Aktuelle I ² t-Überlast siehe Objekt 2112-06h.
3110h	0000h	Fehler Überspannung Bus	Betriebsspannung überprüfen.
3111h	0000h	Fehler Überspannung Leistungselektronik	Betriebsspannung überprüfen.
3120h	0000h	Fehler Unterspannung Bus	Unterspannung während eines aktiven Fahrauftrags
3121h	0000h	Fehler Unterspannung Leistungselektronik	Speisung Leistungselektronik +VsM < 18V und ein Fahrauftrag ist aktiv. Kein Motorbetrieb mehr möglich. CAN-Kommunikation möglich, solange Betriebsspannung an +VsE anliegt.
4210h	0000h	Fehler Übertemperatur	Antrieb kann nicht weiter betrieben werden. Last reduzieren.
4210h	0001h	Warnung Übertemperatur	Antrieb kann weiter betrieben werden. Last reduzieren.
5441h	0000h	Warnung unterer Hardware-Endschalter aktiv	Fahren nur noch in positive Richtung möglich
5442h	0000h	Warnung oberer Hardware-Endschalter aktiv	Fahren nur noch in negative Richtung möglich
5530h	0000h	Warnung Speicher (EEPROM)	Fehler bei Speichervorgang. Speicherung wiederholen. Bei wiederholtem Auftreten Überprüfung des Antriebs im Werk notwendig.
6010h	0000h	Warnung Software (Watchdog)	Reset des Antriebs wurde intern ausgelöst. Antrieb kann weiter betrieben werden. Vorher alle Parameter überprüfen. Bei wiederholtem Auftreten Überprüfung des Antriebs im Werk notwendig.
7121h	0000h	Fehler Motor blockiert	Fahrbehl wurde abgebrochen, da Motor möglicherweise blockiert, Last überprüfen, Strombegrenzung (Objekt 6073h) überprüfen. Neuen Startbehl senden.
7121h	0001h	Warnung Motor blockiert	Motorstrom hat Strombegrenzung (Objekt 6073h) erreicht und steht still. Fahrbehl bleibt aktiv. Motor möglicherweise blockiert, Last überprüfen, Strombegrenzung überprüfen.
7510h	0000h	Error interner Kommunikationsfehler	Bei wiederholtem Auftreten Überprüfung des Antriebs im Werk notwendig.
7320h	0000h	Fehler Encoder	Überprüfung des Antriebs im Werk notwendig.
8110h	0000h	Warnung CAN-Bus Kommunikation	Daten gehen möglicherweise verloren.
8130h	0000h	Warnung Lifeguard oder Heartbeat	Verkabelung überprüfen. Funktion Busmaster überprüfen.
8500h	0000h	Warnung Positionscontroller	Ausserhalb Zielfenster
FF00h	00 00h	Warnung Data valid multiturn (DVMT)	Spannung der Pufferbatterie für Positionsspeicherung hat Warngrenze erreicht. Antrieb kann noch einige Tage bis Wochen weiter betrieben werden. Danach Überprüfung des Antriebs im Werk notwendig.
FF02h	0000h	Fehler Positioning timeout	Motor möglicherweise blockiert, Last überprüfen, Strombegrenzung (Objekt 6073h) überprüfen.
FF05h	0000h	Warnung Deblockierung aktiv	Der Motor führt automatische Deblockierung durch.



FF06h	0000h	Fehler Deblockierung Timeout	Beim automatischen Deblockieren wurde die Unblocking timeout (Objekt 2111-04h) überschritten. Motor möglicherweise blockiert, Freifahrtweg überprüfen, Losbrechstrom (Objekt 2113h-20h) überprüfen
FF10h	0000h	Warnung unterer Software-Endschalter aktiv	Fahren nur noch in positive Richtung möglich (ausser Mode of operation Homing)
FF11h	0000h	Warnung oberer Software-Endschalter aktiv	Fahren nur noch in negative Richtung möglich (ausser Mode of operation Homing)

Tabelle 18: SDO Abort Code Definitionen (siehe Kapitel Servicedaten-Kommunikation). Bei SDO Kommunikationsfehler wird als Antwort eine Abort Message mit Abort Code gesendet.

SDO Abort Code	Beschreibung
05040001h	Kommando-Byte wird nicht unterstützt
06010000h	Falscher Zugriff auf ein Objekt
06010001h	Lesezugriff auf Write Only
06010002h	Schreibzugriff auf Read Only
06020000h	Objekt wird nicht unterstützt
06090011h	Subindex wird nicht unterstützt
06090030h	Wert ausserhalb des zulässigen Wertebereichs
08000020h	Falsche Signatur bei Speichern oder Laden der Default-Parameter
08000021h	Speicherung konnte nicht abgeschlossen werden
08000022h	Zielposition kann nicht gesetzt werden, da Antrieb fahrend

5.4 Automatische Fehler-Quittierung

Der Antrieb kann ohne übergeordnete Steuerung betrieben werden (mittels digitalen Inputs). In diesem Fall ist es unter gewissen Umständen nötig, dass nach dem Abschalten der Motor-Spannungsversorgung oder nach einer Motorblockade den Fehler automatisch zu quittieren. Für diesen Fall können die Fehler Unterspannung Leistungselektronik (Error Code 3121h) und Motor blockiert (Error Code 7121) automatisch vom Antrieb quittiert werden, wenn die übergeordnete Steuerung ausgeschaltet ist (Antrieb bekommt keine Producer Heartbeats). Unter diesen Bedingungen ist die automatische Fehler-Quittierung eingeschaltet (beide müssen erfüllt sein):

- Bit 9 in der Versionskontrolle (Objekt 2110h) „jogging without CAN“ gesetzt
- Consumer Heartbeat konfiguriert (mit ID der Steuerung)

6 EMV-gerechte Verdrahtung

Halten Sie die Spezifikation der CAN-Leitungen ein

Tabelle 19: Spezifikation der CAN-Leitungen.

Gesamtlänge Bussystem	< 300 m	< 1000 m
Kabeltyp	LIYCY 2 x 2 x 0,5 mm ² (paarverseilt mit Abschirmung ist optimal)	CYPIMF 2 x 2 x 0,5 mm ² (paarverseilt mit Abschirmung ist optimal)
Leitungswiderstand	≤ 40 Ohm/km	≤ 40 Ohm/km
Kapazitätsbelag	≤ 130 nF/km	≤ 60 nF/km
Anschluss	Paar 1 (weiss / braun): CAN-GND und +Vs (bei Antrieben +Vs nur auf separater Motorenleitung) Paar 2 (grün / gelb): CAN-HIGH und CAN-LOW	

- Nur Leitungen verwenden, die ein zusätzliches Aderpaar für CAN-GND haben.
- Nur mit korrekt angeschlossenem CAN-GND ist ein störungsfreier Busbetrieb möglich.
- Nur abgeschirmte Leitungen mit auf der Überwurfmutter beidseitig aufgelegtem Schirm verwenden.

Schliessen Sie die Bus-Abschlusswiderstände an

Am physikalischen Anfang und am physikalischen Ende des Bussystems muss jeweils ein Abschlusswiderstand von 120 Ohm angeschlossen sein.

Schirmen Sie Busleitung und Motorleitung korrekt ab

- Schirm der Busleitung unbedingt beidseitig auf Steckergehäuse bzw. Überwurfmutter auflegen.
- Schirm der Motorleitung nach Möglichkeit ebenfalls beidseitig am Steckergehäuse auflegen. Ist dies nicht möglich, so genügt einseitiges Auflegen des Schirm auf der Antriebsseite

Falls eine Erdung über einen der Antriebsflansche nicht sichergestellt werden kann, muss die Kabelabschirmung geerdet werden. Bei beidseitiger Erdung eines Kabels muss der Potentialausgleich zur Verhinderung einer Erdschleife über eine separate Potential-Ausgleichsleitung mit ausreichendem Leiterquerschnitt sichergestellt werden.

Halten Sie unbedingt die zulässigen Bus-Leitungslängen und Stichleitungslängen nach Tabelle 20 ein

Die maximal zulässige Gesamt-Leitungslänge und Gesamt-Stichleitungslänge

- ist abhängig von der Baudrate und
- kann in mehrere Segmente bzw. Einzelstichleitungen aufgeteilt werden.

Tabelle 20: Maximale Gesamt-Busleitungslänge (mit Abschlusswiderstand) und maximale Stichleitungslänge (ohne Abschlusswiderstand) in Abhängigkeit von der Baudrate.

Baudrate [kBit/s]	10	20	50	100	125	250	500	800	1000
Gesamt-Buslänge	5000 m	3000 m	1000 m	500 m	400 m	200 m	75 m	30 m	25 m
Gesamt-Stichleitungslänge	1360 m	875 m	350 m	175 m	140 m	70 m	35 m	20 m	17 m
Einzel-Stichleitungslänge	270 m	175 m	70 m	35 m	28 m	14 m	7 m	4 m	3 m

Die maximal zulässige Segment-Leitungslänge nach Tabelle 21 ist abhängig

- von dem verwendeten Kabelquerschnitt
- von der Anzahl der Kommunikationsteilnehmer

Tabelle 21: Maximale Segment-Leitungslängen in Abhängigkeit von der Anzahl der Kommunikationsteilnehmer und dem Leitungsquerschnitt.

Anzahl der Kommunikationsteilnehmer	Leitungsquerschnitt		
	0,75 mm ²	0,5 mm ²	0,25 mm ²
32	550 m	360 m	200 m
64	470 m	310 m	170 m
100	410 m	270 m	150 m

- Teilen Sie die max. Gesamt-Leitungslänge nach Tabelle 20 in Segmente auf und verwenden Sie Repeater, wenn Sie größere Leitungslängen realisieren müssen als nach Tabelle 21 zulässig sind.
- Repeater erfordern eine Verkürzung der max. Gesamt-Leitungslängen nach Tabelle 20. Diese ist abhängig vom Repeater-Typ. Pro Repeater beträgt die typische Verkürzung der max. Gesamt-Leitungslänge ca. 30 m.
- Die max. Gesamt-Leitungslänge mit Repeatern ergibt sich aus den Längen in Tabelle 20 abzüglich der Summe aller Leitungslängenverkürzungen durch Repeater.

$$L_{\text{Bus max.}} = L_{\text{max. (Tab. 21)}} - (\sum \text{Leitungslängenverkürzung} \cdot \text{Repeateranzahl})$$
- Ein Repeater wird auch empfohlen als Serviceschnittstelle zum störungsfreien Ankoppeln im laufenden Bus-Betrieb oder als Einmessschnittstelle zur galvanischen Trennung des Programmiergeräts.

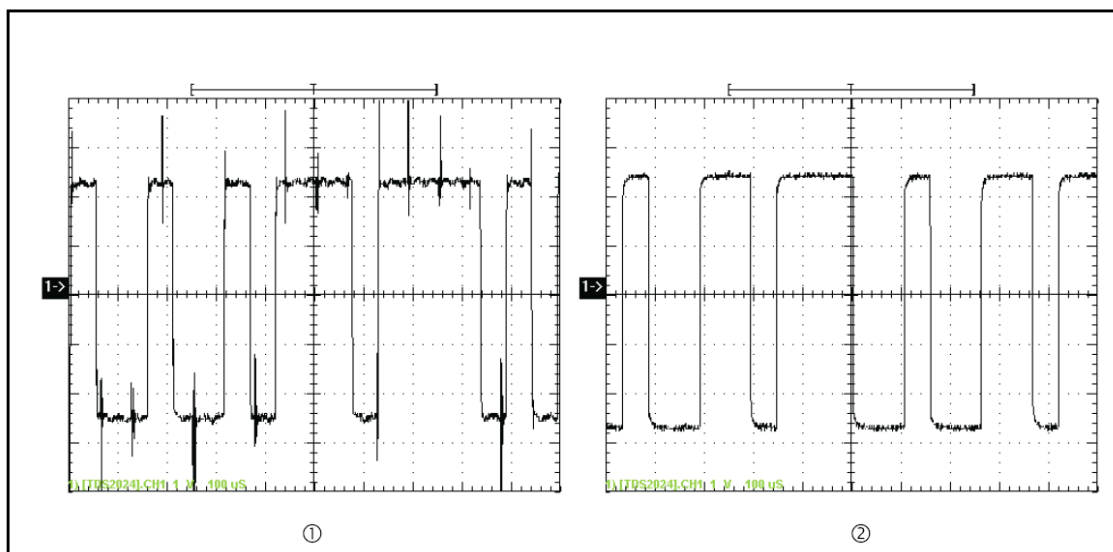
Vermeiden Sie Potentialdifferenzen zwischen den Knoten des CAN-Netzwerks

Potentialdifferenzen (Bezug zu PE) zwischen den Knoten des CAN-Netzwerks können Kommunikationsstörungen verursachen oder zur Zerstörung der CAN-Teilnehmer führen.

Vermeiden Sie Potentialdifferenzen, indem Sie

- jeden CAN-Teilnehmer auf dem kürzesten, möglichst niederohmigen Weg mit dem gleichen Erd-Bezugspotential (PE) der Maschine/Anlage verbinden.
- eine Potentialausgleichsleitung zwischen den Kommunikationsteilnehmern verwenden.
- den Erdbezug der Maschine/Anlage zur Gesamterde niederohmig ausführen.

Erkennen von EMV-Störungen im Signal-Oszillogramm



Figur 23: Oszillogramme der CAN-Signale (1) mit und (2) ohne Störspannung (Messpunkte CAN_H zu CAN_L).

Zur Quantifizierung von Störungen sind Messungen mit einem CAN-Analyser erforderlich. Hiermit können wichtige Bus-Parameter wie beispielsweise die Buslast oder die Anzahl von Errorframes ermittelt und weitergehende Analysen durchgeführt werden.

7 Technische Daten

7.1 Elektrische und mechanische Daten

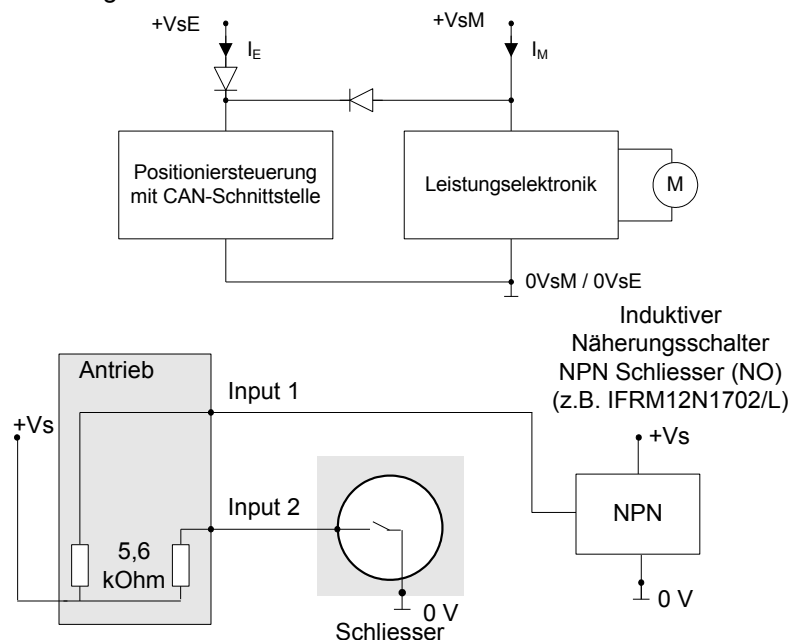
Siehe Datenblatt.

7.2 Anschlussbelegung

Siehe Datenblatt und Typenschild des Produktes.

7.3 Blockschaltbild und Schalteingänge

Der Antrieb verfügt über zwei Anschlüsse zur Spannungsversorgung (Figur 18). Bereits bei Anschluss der Betriebsspannung an +VsM ist der Antrieb voll funktionsfähig. Bei Abschalten von +VsM beispielsweise über einen Not-Aus-Kreis kann die Positioniersteuerung weiterhin über +VsE gespeist werden. Ein Anlaufen des Motors ist in diesem Fall ausgeschlossen.



Figur 24: Blockschaltbild des Antriebs mit NPN Eingängen und Beispiel für Beschaltung der Schalteingänge.

Zusätzlich verfügt der Antrieb über 2 digitale Schalteingänge (Funktion programmierbar, Objekt 2111h-0Dh) zum Anschluss handelsüblicher Positionssensoren. Mögliche Funktionen sind:

- Aktivierung des Tippbetriebs (Jogging+, Jogging-)
- Hardware-Endschalter oder Referenzierung
- Steuerung des Kontrollworts (Objekt 6040h)

7.4 Abmessungen

Siehe Datenblatt.

7.5 Befehlssequenzen als Beispiel

Tabelle 24: Befehlssequenz zur Identifikation des Antriebs.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
SDO Upload Request (Get Software version)	100Ah	00000000h	SDO Upload Response	xxxxxxxh
SDO Upload Request (Get Identity object, Product code)	1018h-02h	00000000h	SDO Upload Response	00000101h
SDO Upload Request (Get Identity object, Revision number)	1018h-03h	00000000h	SDO Upload Response	xxxxxxxh
SDO Upload Request (Get Identity object, Serial number)	1018h-04h	00000000h	SDO Upload Response	xxxxxxxh

Tabelle 25: Befehlssequenz zum Setzen der Applikationsparameter.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
SDO Download Request (Set Maximum current)	6073h	03E8h	SDO Download Response	00000000h
SDO Download Request (Set Polarity)	607Eh	00h	SDO Download Response	00000000h
SDO Download Request (Set Position notation)	6089h	FDh	SDO Download Response	00000000h
SDO Download Request (Set Position dimension)	608Ah	01h	SDO Download Response	00000000h
SDO Download Request (Set Velocity notation)	608Bh	00h	SDO Download Response	00000000h
SDO Download Request (Set Velocity dimension)	608Ch	A3h	SDO Download Response	00000000h
SDO Download Request (Set Position encoder resolution, Encoder increments)	608Fh-01h	00000004h	SDO Download Response	00000000h
SDO Download Request (Set Position encoder resolution, Encoder revolutions)	608Fh-02h	00000001h	SDO Download Response	00000000h
SDO Download Request (Set Gear ratio, Motor revolutions)	6091h-01h	00000012h	SDO Download Response	00000000h
SDO Download Request (Set Gear ratio, Gear shaft revolutions)	6091h-02h	00000001h	SDO Download Response	00000000h
SDO Download Request (Set Feed constant, Feed)	6092h-01h	00000001h	SDO Download Response	00000000h
SDO Download Request (Set Feed constant, Spindle shaft revolutions)	6092h-02h	00000001h	SDO Download Response	00000000h
SDO Download Request (Set Software position limit, Minimum software position limit)	607Dh-01h	00000000h	SDO Download Response	00000000h
SDO Download Request (Set Software position limit, Maximum software position limit)	607Dh-02h	00000000h	SDO Download Response	00000000h
SDO Download Request (Set Positioning parameter, Backlash compensation)	2111h-01h	00h	SDO Download Response	00000000h
SDO Download Request (Set Positioning parameter, Backlash delta)	2111h-02h	00000064h	SDO Download Response	00000000h
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h

Tabelle 26: Befehlssequenz zum Setzen der Kommunikationsparameter.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
SDO Download Request (Set Receive PDO1 parameter, R_PDO1 type)	1400h-02h	FEh	SDO Download Response	00000000h
SDO Download Request (Set Receive PDO2 parameter, R_PDO2 type)	1401h-02h	FEh	SDO Download Response	00000000h
SDO Download Request (Set Transmit PDO1 parameter, T_PDO1 type)	1800h-02h	FEh	SDO Download Response	00000000h
SDO Download Request (Set Transmit PDO1 parameter, T_PDO1 Event Timer)	1800h-05h	0203h	SDO Download Response	00000000h
SDO Download Request (Set T_PDO1 add-on)	2800h	01h	SDO Download Response	00000000h
SDO Download Request (Set Transmit PDO2 parameter, T_PDO2 type)	1801h-02h	FEh	SDO Download Response	00000000h
SDO Download Request (Set Transmit PDO2 parameter, T_PDO2 Event Timer)	1801h-05h	0100h	SDO Download Response	00000000h
SDO Download Request (Set T_PDO2 add-on)	2801h	01h	SDO Download Response	00000000h
SDO Download Request (Set Consumer heartbeat time)	1016h-01h	yyyyzzzh	SDO Download Response	00000000h
SDO Download Request (Set Producer heartbeat time)	1017h	xxxxh	SDO Download Response	00000000h
SDO Download Request (Store communication parameters)	1010h-02h	65766173h	SDO Download Response	00000000h

Tabelle 27: Wechsel vom Zustand Ready in den Zustand PositioningReady (einmalig nach Einschalten oder Reset des Antriebs).

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h

Tabelle 28: Tippen in positiver Richtung (Antrieb im Zustand PositioningReady).

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
SDO Download Request (Set Controlword)	6040h	081Fh	SDO Download Response	00000000h

Tabelle 29: Tippen in negativer Richtung (Antrieb im Zustand PositioningReady).

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
SDO Download Request (Set Controlword)	6040h	101Fh	SDO Download Response	00000000h

Tabelle 30: Positionieren über SDO (Antrieb im Zustand PositioningReady).

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
NMT Cmd (Start Node, einmalig nach Einschalten / Reset)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Target position)	607Ah	xxxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
SDO Upload Request (Get Statusword)	6041h		SDO Upload Response	4237h
-			T_PDO1	5237h
-			T_PDO2	xxxxxxxxh
SDO Upload Request (Get Statusword)	6041h		SDO Upload Response	5237h
-			(T_PDO1, T_PDO2)	
-			T_PDO1	4637h

Tabelle 31: Positionieren über R_PDO (Antrieb im Zustand PositioningReady).

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
NMT Cmd (Start Node, einmalig nach Einschalten / Reset)		01h	(T_PDO1, T_PDO2)	
R_PDO2 (Set Target position + Controlword)		xxxxxxxh 001Fh	T_PDO1	5237h
SDO Upload Request (Get Statusword)	6041h		SDO Upload Response	4237h
-			T_PDO2	xxxxxxxh
SDO Upload Request (Get Statusword)	6041h		SDO Upload Response	5237h
-			(T_PDO1, T_PDO2)	
-			T_PDO1	4637h

Tabelle 32: Stoppen eines Fahrauftrags.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
SDO Download Request (Set Controlword)	6040h	011Fh	SDO Download Response	00000000h

Tabelle 33: Quittieren eines Fehlers.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
SDO Download Request (Set Controlword)	6040h	008Fh	SDO Download Response	00000000h

Tabelle 34: Referenzieren mit Presetwert.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
NMT Cmd (Start Node)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Mode of operation)	6060h	06h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h
SDO Download Request (Set Homing method)	6098h	F4h	SDO Download Response	00000000h
SDO Download Request (Set Target position)	607Ah	xxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
-			(T_PDO1, T_PDO2)	
-			T_PDO2	xxxxxxxh
-			T_PDO1	4237h
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h

Tabelle 35: Referenzieren auf Referenzschalter in negativer Richtung.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
NMT Cmd (Start Node)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Positioning parameter, Input1)	2111h-0Dh	12h	SDO Download Response	00000000h
SDO Download Request (Set Positioning parameter, Reference Position)	2111h-11h	xxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	06h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h
SDO Download Request (Set Homing method)	6098h	E2h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
-			T_PDO1	1237h
-			T_PDO2	xxxxxxxh
-			(T_PDO1, T_PDO2)	
-			T_PDO1	4237h
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h

Tabelle 36: Referenzierung auf unteren Anschlag.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
NMT Cmd (Start Node)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Positioning parameter, Reference Position)	2111h-11h	xxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Maximum current)	6073h	01F4h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	06h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h
SDO Download Request (Set Homing method)	6098h	D8h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
-			T_PDO1	1237h
-			T_PDO2	xxxxxxxh
-			(T_PDO1, T_PDO2)	
-			Emergency Message	7121h
-			Emergency Message	0000h
-			(T_PDO1, T_PDO2)	
-			T_PDO1	4237h
-			T_PDO2	xxxxxxxh
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h

Tabelle 37: Referenzierung mit Einmessmethode.

Master PLD an Slave MSBA	Objekt	Wert	Slave MSBA an Master PLD	Wert
NMT Cmd (Start Node)		01h	(T_PDO1, T_PDO2)	
SDO Download Request (Set Target position)	607Ah	xxxxxxxh	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	06h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h
SDO Download Request (Set Homing method)	6098h	F3h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	001Fh	SDO Download Response	00000000h
-			T_PDO1	4337h
-			T_PDO2	xxxxxxxh
SDO Download Request (Store application parameters)	1010h-03h	65766173h	SDO Download Response	00000000h
SDO Download Request (Set Mode of operation)	6060h	01h	SDO Download Response	00000000h
SDO Download Request (Set Controlword)	6040h	000Fh	SDO Download Response	00000000h