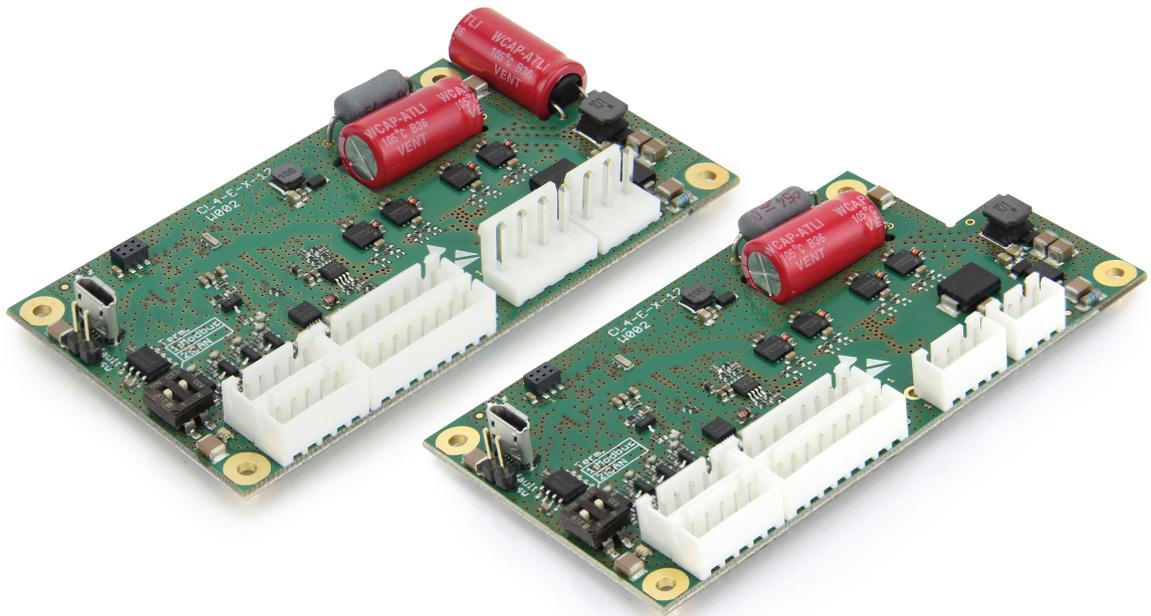


Technisches Handbuch CL4-E

Feldbus: CANopen, USB, Modbus RTU

Zur Benutzung mit folgenden Varianten:

CL4-E-1-12, CL4-E-2-12, CL4-E-1-12-5VDI, CL4-E-2-12-5VDI



Inhalt

1	Einleitung.....	8
1.1	Versionshinweise.....	8
1.2	Urheberrecht, Kennzeichnung und Kontakt.....	8
1.3	Bestimmungsgemäßer Gebrauch.....	8
1.4	Gewährleistung und Haftungsausschluss.....	8
1.5	Fachkräfte.....	9
1.6	EU-Richtlinien zur Produktsicherheit.....	9
1.7	Mitgeltende Vorschriften.....	9
1.8	Verwendete Symbole.....	9
1.9	Hervorhebungen im Text.....	10
1.10	Zahlenwerte.....	10
1.11	Bits.....	10
1.12	Zählrichtung (Pfeile).....	10
2	Sicherheits- und Warnhinweise.....	12
3	Technische Daten und Anschlussbelegung.....	13
3.1	Umgebungsbedingungen.....	13
3.2	Maßzeichnung.....	13
3.3	Elektrische Eigenschaften und technische Daten.....	13
3.4	Übertemperaturschutz.....	14
3.5	LED-Signalisierung.....	15
3.6	Anschlussbelegung.....	17
4	Inbetriebnahme.....	27
4.1	Konfiguration über USB.....	27
4.2	Konfiguration über CANopen.....	31
4.3	Konfiguration über Modbus RTU.....	32
4.4	Motordaten einstellen.....	33
4.5	Motor anschließen.....	34
4.6	Auto-Setup.....	34
4.7	Testlauf.....	37
5	Generelle Konzepte.....	39
5.1	Betriebsarten.....	39
5.2	CiA 402 Power State Machine.....	43
5.3	Benutzerdefinierte Einheiten.....	48
5.4	Begrenzung des Bewegungsbereichs.....	53
5.5	Zykluszeiten.....	54
6	Betriebsmodi.....	55
6.1	Profile Position.....	55
6.2	Velocity.....	63
6.3	Profile Velocity.....	64
6.4	Profile Torque.....	67
6.5	Homing.....	69
6.6	Interpolated Position Mode.....	77

6.7 Cyclic Synchronous Position.....	78
6.8 Cyclic Synchronous Velocity.....	80
6.9 Cyclic Synchronous Torque.....	81
6.10 Takt-Richtungs-Modus.....	82
6.11 Auto-Setup.....	84
7 Spezielle Funktionen.....	85
7.1 Digitale Ein- und Ausgänge.....	85
7.2 I ² t Motor-Überlastungsschutz.....	93
7.3 Objekte speichern.....	95
8 CANopen.....	102
8.1 Allgemeines.....	102
8.2 CANopen Dienste.....	102
8.3 LSS-Protokoll.....	120
9 Modbus RTU.....	134
9.1 RS-485.....	134
9.2 Modbus Modicon-Notation bei SPS.....	134
9.3 Allgemeines.....	134
9.4 Funktionscodes.....	135
9.5 Funktionscode-Beschreibungen.....	135
9.6 Prozessdatenobjekte (PDO).....	159
9.7 NanoJ-Objekte.....	161
10 Programmierung mit NanoJ.....	162
10.1 NanoJ-Programm.....	162
10.2 Mapping im NanoJ-Programm.....	166
10.3 Systemcalls im NanoJ-Programm.....	167
11 Objektverzeichnis Beschreibung.....	169
11.1 Übersicht.....	169
11.2 Aufbau der Objektbeschreibung.....	169
11.3 Objektbeschreibung.....	169
11.4 Wertebeschreibung.....	171
11.5 Beschreibung.....	172
1000h Device Type.....	172
1001h Error Register.....	173
1003h Pre-defined Error Field.....	174
1005h COB-ID Sync.....	178
1007h Synchronous Window Length.....	178
1008h Manufacturer Device Name.....	179
1009h Manufacturer Hardware Version.....	179
100Ah Manufacturer Software Version.....	180
100Ch Guard Time.....	180
100Dh Live Time Factor.....	181
1010h Store Parameters.....	181
1011h Restore Default Parameters.....	185
1014h COB-ID EMCY.....	188
1016h Consumer Heartbeat Time.....	189
1017h Producer Heartbeat Time.....	190
1018h Identity Object.....	191
1019h Synchronous Counter Overflow Value.....	192
1020h Verify Configuration.....	193

1029h Error Behavior.....	194
1400h Receive PDO 1 Communication Parameter.....	195
1401h Receive PDO 2 Communication Parameter.....	196
1402h Receive PDO 3 Communication Parameter.....	197
1403h Receive PDO 4 Communication Parameter.....	198
1404h Receive PDO 5 Communication Parameter.....	199
1405h Receive PDO 6 Communication Parameter.....	201
1406h Receive PDO 7 Communication Parameter.....	202
1407h Receive PDO 8 Communication Parameter.....	203
1600h Receive PDO 1 Mapping Parameter.....	204
1601h Receive PDO 2 Mapping Parameter.....	207
1602h Receive PDO 3 Mapping Parameter.....	209
1603h Receive PDO 4 Mapping Parameter.....	211
1604h Receive PDO 5 Mapping Parameter.....	214
1605h Receive PDO 6 Mapping Parameter.....	216
1606h Receive PDO 7 Mapping Parameter.....	218
1607h Receive PDO 8 Mapping Parameter.....	220
1800h Transmit PDO 1 Communication Parameter.....	222
1801h Transmit PDO 2 Communication Parameter.....	224
1802h Transmit PDO 3 Communication Parameter.....	227
1803h Transmit PDO 4 Communication Parameter.....	229
1804h Transmit PDO 5 Communication Parameter.....	231
1805h Transmit PDO 6 Communication Parameter.....	233
1806h Transmit PDO 7 Communication Parameter.....	235
1807h Transmit PDO 8 Communication Parameter.....	237
1A00h Transmit PDO 1 Mapping Parameter.....	240
1A01h Transmit PDO 2 Mapping Parameter.....	242
1A02h Transmit PDO 3 Mapping Parameter.....	245
1A03h Transmit PDO 4 Mapping Parameter.....	247
1A04h Transmit PDO 5 Mapping Parameter.....	250
1A05h Transmit PDO 6 Mapping Parameter.....	252
1A06h Transmit PDO 7 Mapping Parameter.....	255
1A07h Transmit PDO 8 Mapping Parameter.....	257
1F50h Program Data.....	260
1F51h Program Control.....	261
1F57h Program Status.....	262
1F80h NMT Startup.....	264
2005h CANopen Baudrate.....	264
2007h CANopen Config.....	265
2009h CANopen NodeID.....	266
2028h MODBUS Slave Address.....	266
202Ah MODBUS RTU Baudrate.....	267
202Ch MODBUS RTU Stop Bits.....	267
202Dh MODBUS RTU Parity.....	268
2030h Pole Pair Count.....	269
2031h Max Motor Current.....	269
2034h Upper Voltage Warning Level.....	270
2035h Lower Voltage Warning Level.....	270
2036h Open Loop Current Reduction Idle Time.....	271
2037h Open Loop Current Reduction Value/factor.....	271
2038h Brake Controller Timing.....	272
2039h Motor Currents.....	274
203Ah Homing On Block Configuration.....	275
203Bh I2t Parameters.....	277
203Dh Torque Window.....	279
203Eh Torque Window Time Out.....	280
203Fh Max Slippage Time Out.....	280
2057h Clock Direction Multiplier.....	281
2058h Clock Direction Divider.....	281

205Ah Absolute Sensor Boot Value (in User Units).....	282
205Bh Clock Direction Or Clockwise/Counter Clockwise Mode.....	282
2084h Bootup Delay.....	283
2101h Fieldbus Module Availability.....	283
2102h Fieldbus Module Control.....	284
2103h Fieldbus Module Status.....	286
2290h PDI Control.....	288
2291h PDI Input.....	288
2292h PDI Output.....	290
2300h NanoJ Control.....	291
2301h NanoJ Status.....	291
2302h NanoJ Error Code.....	292
230Fh Uptime Seconds.....	294
2310h NanoJ Input Data Selection.....	294
2320h NanoJ Output Data Selection.....	296
2330h NanoJ In/output Data Selection.....	297
2400h NanoJ Inputs.....	298
2410h NanoJ Init Parameters.....	299
2500h NanoJ Outputs.....	300
2600h NanoJ Debug Output.....	301
2800h Bootloader And Reboot Settings.....	302
3202h Motor Drive Submode Select.....	303
3203h Feedback Selection.....	304
3204h Feedback Mapping.....	306
3210h Motor Drive Parameter Set.....	308
3212h Motor Drive Flags.....	312
3220h Analog Inputs.....	313
3221h Analogue Inputs Control.....	314
3240h Digital Inputs Control.....	315
3242h Digital Input Routing.....	318
3243h Digital Input Homing Capture.....	320
3250h Digital Outputs Control.....	322
3252h Digital Output Routing.....	324
3320h Read Analogue Input.....	326
3321h Analogue Input Offset.....	327
3322h Analogue Input Pre-scaling.....	328
3390h Feedback Hall.....	329
33A0h Feedback Incremental A/B/I 1.....	332
3502h MODBUS Rx PDO Mapping.....	333
3602h MODBUS Tx PDO Mapping.....	338
3700h Deviation Error Option Code.....	342
3701h Limit Switch Error Option Code.....	343
4012h HW Information.....	344
4013h HW Configuration.....	344
4014h Operating Conditions.....	345
4040h Drive Serial Number.....	347
4041h Device Id.....	348
603Fh Error Code.....	348
6040h Controlword.....	349
6041h Statusword.....	350
6042h VI Target Velocity.....	351
6043h VI Velocity Demand.....	352
6044h VI Velocity Actual Value.....	352
6046h VI Velocity Min Max Amount.....	353
6048h VI Velocity Acceleration.....	354
6049h VI Velocity Deceleration.....	355
604Ah VI Velocity Quick Stop.....	356
604Ch VI Dimension Factor.....	357
605Ah Quick Stop Option Code.....	358

605Bh Shutdown Option Code.....	359
605Ch Disable Option Code.....	360
605Dh Halt Option Code.....	360
605Eh Fault Option Code.....	361
6060h Modes Of Operation.....	362
6061h Modes Of Operation Display.....	362
6062h Position Demand Value.....	363
6063h Position Actual Internal Value.....	363
6064h Position Actual Value.....	364
6065h Following Error Window.....	364
6066h Following Error Time Out.....	365
6067h Position Window.....	365
6068h Position Window Time.....	366
606Bh Velocity Demand Value.....	367
606Ch Velocity Actual Value.....	367
606Dh Velocity Window.....	368
606Eh Velocity Window Time.....	368
6071h Target Torque.....	369
6072h Max Torque.....	369
6073h Max Current.....	370
6074h Torque Demand.....	371
6075h Motor Rated Current.....	371
6077h Torque Actual Value.....	371
607Ah Target Position.....	372
607Bh Position Range Limit.....	372
607Ch Home Offset.....	374
607Dh Software Position Limit.....	374
607Eh Polarity.....	375
607Fh Max Profile Velocity.....	376
6080h Max Motor Speed.....	377
6081h Profile Velocity.....	377
6082h End Velocity.....	378
6083h Profile Acceleration.....	378
6084h Profile Deceleration.....	378
6085h Quick Stop Deceleration.....	379
6086h Motion Profile Type.....	379
6087h Torque Slope.....	380
608Fh Position Encoder Resolution.....	380
6090h Velocity Encoder Resolution.....	382
6091h Gear Ratio.....	383
6092h Feed Constant.....	384
6096h Velocity Factor.....	385
6097h Acceleration Factor.....	386
6098h Homing Method.....	387
6099h Homing Speed.....	388
609Ah Homing Acceleration.....	389
60A2h Jerk Factor.....	389
60A4h Profile Jerk.....	391
60A8h SI Unit Position.....	392
60A9h SI Unit Velocity.....	393
60B0h Position Offset.....	393
60B1h Velocity Offset.....	394
60B2h Torque Offset.....	394
60C1h Interpolation Data Record.....	395
60C2h Interpolation Time Period.....	396
60C4h Interpolation Data Configuration.....	397
60C5h Max Acceleration.....	399
60C6h Max Deceleration.....	399
60E4h Additional Position Actual Value.....	400

60E5h Additional Velocity Actual Value.....	401
60E6h Additional Position Encoder Resolution - Encoder Increments.....	402
60E8h Additional Gear Ratio - Motor Shaft Revolutions.....	403
60E9h Additional Feed Constant - Feed.....	405
60EBh Additional Position Encoder Resolution - Motor Revolutions.....	406
60EDh Additional Gear Ratio - Driving Shaft Revolutions.....	407
60EEh Additional Feed Constant - Driving Shaft Revolutions.....	408
60F2h Positioning Option Code.....	409
60F4h Following Error Actual Value.....	411
60F8h Max Slippage.....	411
60FAh Control Effort.....	412
60FCh Position Demand Internal Value.....	412
60FDh Digital Inputs.....	413
60FEh Digital Outputs.....	413
60FFh Target Velocity.....	415
6502h Supported Drive Modes.....	415
6503h Drive Catalogue Number.....	416
6505h Http Drive Catalogue Address.....	417

12 Copyrights.....	418
12.1 Einführung.....	418
12.2 AES.....	418
12.3 MD5.....	418
12.4 uIP.....	419
12.5 DHCP.....	419
12.6 CMSIS DSP Software Library.....	419
12.7 FatFs.....	419
12.8 Protothreads.....	420
12.9 lwIP.....	420

1 Einleitung

Die *CL4-E* ist eine Steuerung für den *Open Loop*- oder *Closed Loop*-Betrieb von Schrittmotoren und den *Closed Loop*-Betrieb von BLDC- Motoren.

Dieses Handbuch beschreibt die Funktionen der Steuerung und die verfügbaren Betriebsmodi. Weiterhin wird gezeigt, wie Sie die Steuerung über die Kommunikationsschnittstelle ansprechen und programmieren können.

Weitere Informationen zum Gerät finden Sie auf der Nanotec Homepage www.nanotec.de.

1.1 Versionshinweise

Version Handbuch	Datum	Änderungen	Version Firmware	Version Hardware
1.0.0	10/2018	erste Veröffentlichung	FIR-v1825	W002

1.2 Urheberrecht, Kennzeichnung und Kontakt

Copyright © 2013 – 2018 Nanotec Electronic GmbH & Co. KG. Alle Rechte vorbehalten.



Nanotec Electronic GmbH & Co. KG
Kapellenstraße 6
85622 Feldkirchen
Deutschland

Tel.+49 89 900 686-0
Fax +49 89 900 686-50

www.nanotec.de

1.3 Bestimmungsgemäßer Gebrauch

Die *CL4-E Steuerung* dient der Steuerung von Schrittmotoren und BLDC-Motoren und ist für den Einsatz unter den freigegebenen **Umgebungsbedingungen** konzipiert.

Ein anderer Gebrauch gilt als nicht bestimmungsgemäß.



Hinweis

Änderungen oder Umbauten der Steuerung sind nicht zulässig.

1.4 Gewährleistung und Haftungsausschluss

Nanotec produziert Komponententeile, die ihren Einsatz in vielfältigen Industrieanwendungen finden. Die Auswahl und Anwendung von Nanotec-Produkten liegt im Verantwortungsbereich des

Anlagenkonstruktors bzw. Endnutzers. Nanotec übernimmt keinerlei Verantwortung für die Integration der Produkte in das Endsystem.

Unter keinen Umständen darf ein Nanotec-Produkt als Sicherheitssteuerung in ein Produkt oder eine Konstruktion integriert werden. Alle Produkte, in denen ein von Nanotec hergestelltes Komponententeil enthalten ist, müssen bei der Übergabe an den Endnutzer entsprechende Warnhinweise und Anweisungen für eine sichere Verwendung und einen sicheren Betrieb aufweisen. Alle von Nanotec bereitgestellten Warnhinweise müssen unmittelbar an den Endnutzer weitergegeben werden.

Es gelten unsere Allgemeinen Geschäftsbedingungen: de.nanotec.com/service/agb/.

1.5 Fachkräfte

Nur Fachkräfte dürfen das Gerät installieren, programmieren und in Betrieb nehmen:

- Personen, die eine entsprechende Ausbildung und Erfahrung im Umgang mit Motoren und deren Steuerung haben.
- Personen, die den Inhalt dieses technischen Handbuchs kennen und verstehen.
- Personen, die die geltenden Vorschriften kennen.

1.6 EU-Richtlinien zur Produktsicherheit

Folgende EU-Richtlinien wurden beachtet:

- RoHS-Richtlinie (2011/65/EU, 2015/863/EU)

1.7 Mitgeltende Vorschriften

Neben diesem technischen Handbuch sind folgende Vorschriften zu beachten:

- Unfallverhütungsvorschriften
- örtliche Vorschriften zur Arbeitssicherheit

1.8 Verwendete Symbole

Alle Hinweise sind in einheitlicher Form. Der Grad der Gefährdung wird in die nachfolgenden Klassen eingeteilt.



VORSICHT

Der Hinweis VORSICHT verweist auf eine möglicherweise gefährliche Situation.

Die Missachtung des Hinweises führt **möglicherweise** zu mittelschweren Verletzungen.

- ▶ Beschreibt, wie Sie die gefährliche Situation vermeiden.



Hinweis

- Weist auf eine Fehlerquelle oder Verwechslungsgefahr hin.
- Die Missachtung des Hinweises führt **möglicherweise** zu Beschädigungen an diesem Gerät oder anderen Geräten.
- Beschreibt, wie Sie Geräteschäden vermeiden können.



Tipp

Zeigt einen Tipp zur Anwendung oder Aufgabe.

1.9 Hervorhebungen im Text

Im Dokument gelten folgende Konventionen:

Ein **fett** hervorgehobener Text markiert Querverweise und Hyperlinks:

- Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:
- Eine Liste verfügbarer Systemcalls findet sich im Kapitel **Systemcalls im NanoJ-Programm**.

Ein *kursiv* hervorgehobener Text markiert benannte Objekte:

- Lesen Sie das *Installationshandbuch*.
- Benutzen Sie die Software *Plug & Drive Studio*, um das Auto-Setup durchzuführen.
- Für Software: Im Tab *Operation* finden Sie die entsprechenden Informationen.
- Für Hardware: Benutzen Sie den *EIN/AUS*-Schalter, um das Gerät einzuschalten.

Ein Text in *courier* markiert einen Code-Abschnitt oder Programmierbefehl:

- Die Zeile mit dem Befehl `od_write(0x6040, 0x00, 5);` ist wirkungslos.
- Die NMT-Nachricht baut sich wie folgt auf: `000 | 81 2A`

Ein Text in "Anführungszeichen" markiert Benutzereingaben:

- NanoJ-Programm starten durch Beschreiben von Objekt 2300_h, Bit 0 = "1".
- Wird in diesem Zustand bereits Haltemoment benötigt, muss in das 3212_h:01_h der Wert "1" geschrieben werden.

1.10 Zahlenwerte

Zahlenwerte werden grundsätzlich in dezimaler Schreibweise angegeben. Sollte eine hexadezimale Notation verwendet werden, wird das mit einem tiefgestellten *h* am Ende der Zahl markiert.

Die Objekte im Objektverzeichnis werden mit Index und Subindex folgendermaßen notiert:

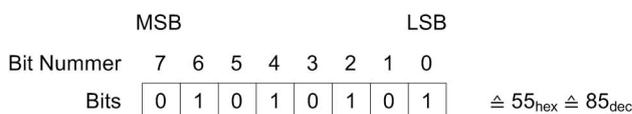
<Index>:<Subindex>

Sowohl der Index als auch der Subindex werden in hexadezimaler Schreibweise angegeben. Sollte kein Subindex notiert sein, gilt der Subindex 00_h.

Beispiel: Der Subindex 5 des Objekts 1003_h wird adressiert mit 1003_h:05_h, der Subindex 00 des Objekts 6040_h mit 6040_h.

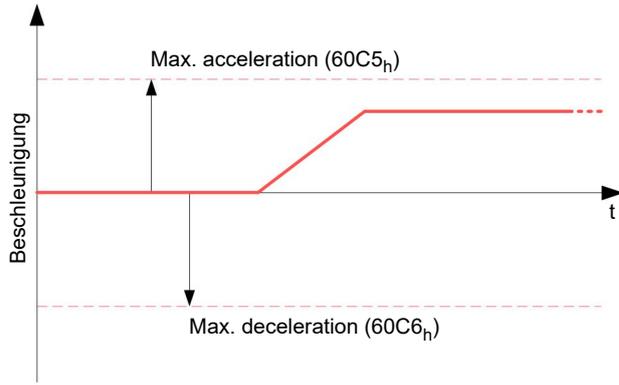
1.11 Bits

Einzelne Bits in einem Objekt beginnen bei der Nummerierung immer bei dem LSB (Bitnummer 0). Siehe nachfolgende Abbildung am Beispiel des Datentyps *UNSIGNED8*.



1.12 Zählrichtung (Pfeile)

In Abbildungen gilt die Zählrichtung immer in Richtung eines Pfeiles. Die in der nachfolgenden Abbildung beispielhaft dargestellten Objekte 60C5_h und 60C6_h werden beide positiv angegeben.



2 Sicherheits- und Warnhinweise



Hinweis

- Beschädigung der Steuerung.
- Ein Wechsel der Verdrahtung im Betrieb kann die Steuerung beschädigen.
- Ändern Sie die Verdrahtung nur im spannungsfreien Zustand und warten Sie nach dem Abschalten, bis sich die Kondensatoren entladen haben.



Hinweis

- Störung der Steuerung durch Erregerspannung des Motors.
- Während des Betriebs können Spannungsspitzen die Steuerung beschädigen.
- Verbauen Sie geeignete Schaltungen (z. B. Stützkondensator), die Spannungsspitzen abbauen.



Hinweis

- Das Gerät enthält Bauteile, die empfindlich gegen elektrostatische Entladung sind.
- Unsachgemäßer Umgang kann das Gerät beschädigen.
- Beachten Sie die Grundprinzipien des ESD-Schutzes beim Umgang mit dem Gerät.

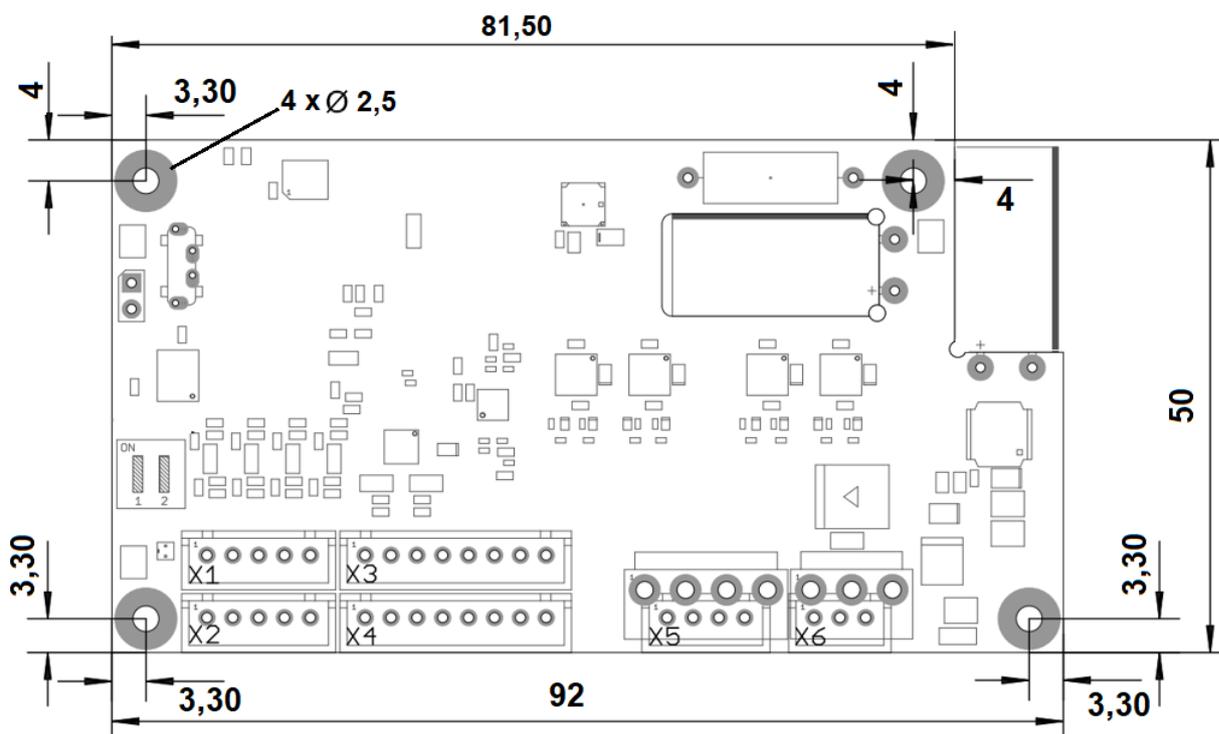
3 Technische Daten und Anschlussbelegung

3.1 Umgebungsbedingungen

Umgebungsbedingung	Wert
Schutzklasse	kein IP-Schutz
Umgebungstemperatur (Betrieb)	-10 ... +40°C
Luftfeuchtigkeit (nicht kondensierend)	0 ... 95 %
Aufstellhöhe über NN (ohne Leistungsbeschränkung)	1500 m
Umgebungstemperatur (Lagerung)	-25 ... +85°C

3.2 Maßzeichnung

Alle Maße sind in Millimetern.



3.3 Elektrische Eigenschaften und technische Daten

Eigenschaft	Beschreibung/Wert
Betriebsspannung	12 ... 58 V DC
Spannungsbereich der Logikversorgung	12 ... 30 V DC

Eigenschaft	Beschreibung/Wert
Nennstrom	CL4-E-1-xx, <i>low current</i> : 3 A _{eff} CL4-E-2-xx, <i>high current</i> : 6 A _{eff}
Spitzenstrom	CL4-E-1-xx, <i>low current</i> : 6 A _{eff} für 5 Sekunden CL4-E-2-xx, <i>high current</i> : 18 A _{eff} für 5 Sekunden
Kommutierung	Schrittmotor Open Loop, Schrittmotor Closed Loop mit Encoder, BLDC-Motor Closed Loop mit Hall Sensor und BLDC-Motor Closed Loop mit Encoder
Betriebsmodi	<i>Profile Position Mode, Profile Velocity Mode, Profile Torque Mode, Velocity Mode, Homing Mode, Interpolated Position Mode, Cyclic Sync Position Mode, Cyclic Sync Velocity Mode, Cyclic Synchronous Torque Mode, Takt-Richtung-Modus</i>
Sollwertvorgabe/ Programmierung	<i>Takt-Richtung, Analog, NanoJ-Programm</i>
Schnittstellen	CANopen, RS-485 (Modbus RTU), USB (Konfigurationsschnittstelle)
Eingänge	<ul style="list-style-type: none"> • 4 digitale Eingänge: <ul style="list-style-type: none"> • 24 V für die Varianten mit den Artikelnummern <i>CL4-E-1-12</i> und <i>CL4-E-2-12</i> • 5 V für die Varianten mit den Artikelnummern <i>CL4-E-1-12-5VDI</i> und <i>CL4-E-2-12-5VDI</i> • 1 analoger Eingang, 10 Bit Auflösung, 0 - 10 V
Ausgänge	2 digitale Ausgänge, positiv schaltend (typische Ausgangsspannung entspricht der angeschlossenen Logikversorgung – 0,6 V)
Schutzschaltung	Über- und Unterspannungsschutz Übertemperaturschutz (> 75° Celsius auf der Leistungsplatine)

3.4 Übertemperaturschutz

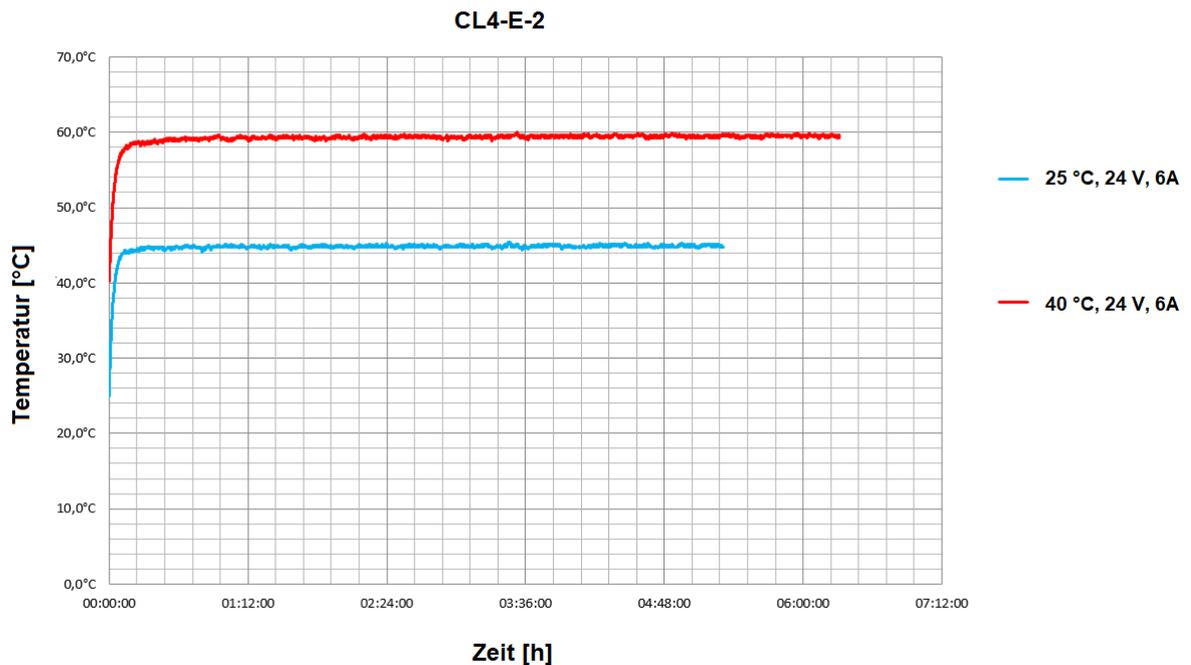
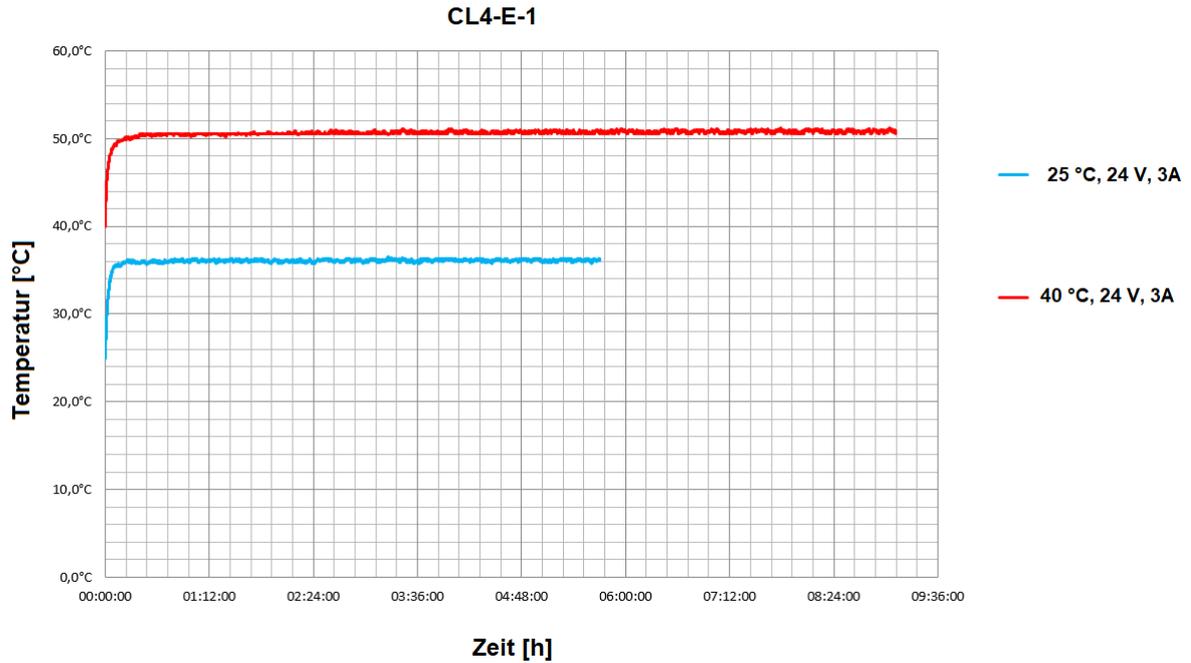
Ab einer Temperatur von ca. 75 °C auf der Leistungsplatine wird das Leistungsteil der Steuerung abgeschaltet und das Fehlerbit gesetzt (siehe Objekt **1001_h** und **1003_h**). Nach Abkühlung und dem Bestätigen des Fehlers (siehe **Tabelle für das Contolword**, "Fault reset") funktioniert die Steuerung wieder normal.

Die folgenden Ergebnisse von Temperaturtests geben einen Hinweis auf das Temperaturverhalten dieser Steuerung.

Es wurden Temperaturtests unter folgenden Bedingungen durchgeführt:

- Betriebsspannung: 24 V DC
- Motorstrom: 3 A (CL4-E-1-xx *low current*)/6 A (CL4-E-2-xx *high current*) effektiv
- Operationsmodus: Drehzahlmodus Vollschritt, 30 U/min
- Umgebungstemperatur: 25 °C / 45 °C
- Aufstellhöhe: 500 m über NN

Die folgenden Grafiken zeigen die Ergebnisse der Temperaturtests:

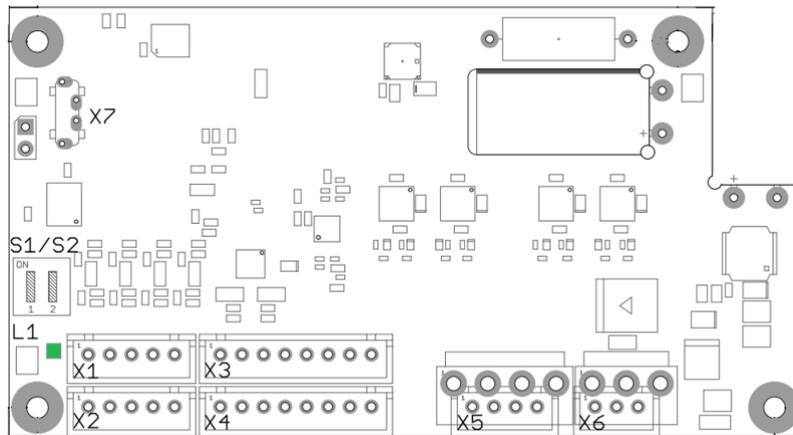


Hinweis

Da das genaue Temperaturverhalten außer vom Motor auch von der Anflanschung und dem dortigen Wärmeübergang sowie von der Konvektion in der Applikation abhängt, empfehlen wir bei Applikationen, die hinsichtlich Stromhöhe und Umgebungstemperatur problematisch sind, immer einen Dauertest in der realen Umgebung.

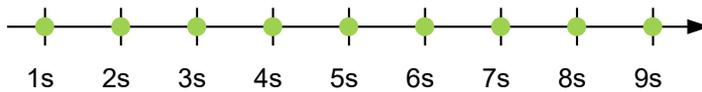
3.5 LED-Signalisierung

3.5.1 Betriebs-LED



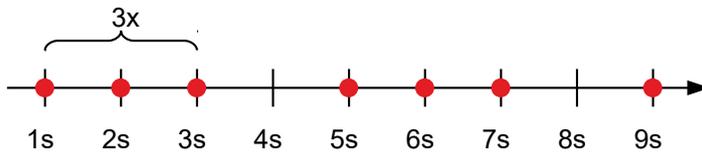
Normaler Betrieb

Im normalen Betrieb blinkt die grüne Betriebs-LED L1 einmal in der Sekunde sehr kurz auf.



Fehlerfall

Liegt ein Fehler vor, schaltet die LED auf Rot um und signalisiert eine Fehlernummer. In der folgenden Darstellung wird der Fehler mit der Nummer 3 signalisiert.



Folgende Tabelle zeigt die Bedeutung der Fehlernummern.

Blinktakt	Fehler
1	Allgemein
2	Spannung
3	Temperatur
4	Überstrom
5	Regler
6	Watchdog-Reset



Hinweis

Für jeden aufgetretenen Fehler wird im Objekt **1003_n** ein genauerer Fehlercode hinterlegt.

3.6 Anschlussbelegung

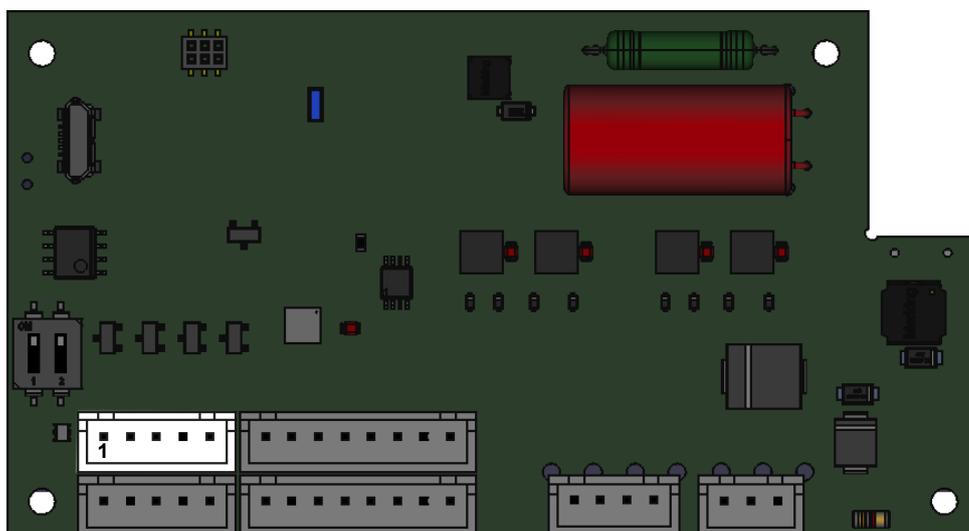
3.6.1 Übersicht

Stecker	Funktion
X1	CANopen / RS-485 IN
X2	CANopen / RS-485 OUT
X3	Encoder/Hall-Sensor
X4	Ein- und Ausgänge
X5	Motor
X6	Spannungsversorgung
X7	Micro-USB
S1	Schalter für Terminierungswiderstand 150 Ohm (RS-485)
S2	Schalter für Terminierungswiderstand 120 Ohm (CANopen)
DS	Default-Switch: Schalter zum Zurücksetzen der Kommunikationseinstellungen

3.6.2 X1 – CANopen/RS-485 IN

Anschluss für CANopen und RS-485 (Modbus RTU). Typ: JST B5B-XH-A

Pin 1 ist in der nachfolgenden Abbildung mit einer "1" markiert.



Pin	Funktion	Bemerkung
1	RS-485-	
2	RS-485+	
3	CAN_L	CAN-Low
4	CAN_H	CAN-High
5	GND	

Leitungspolarisation RS-485



Hinweis

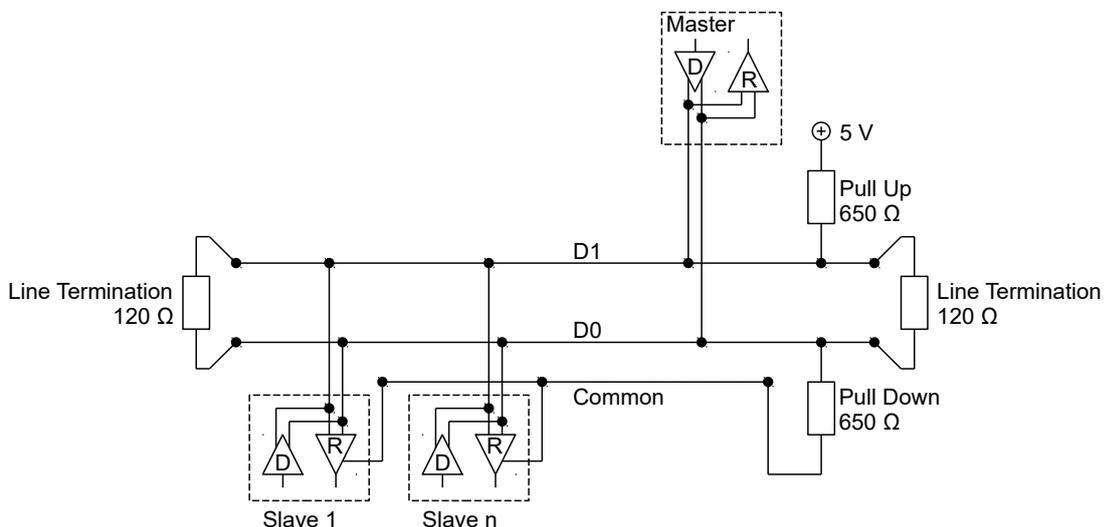
Die Steuerung ist **nicht** mit einer Leitungspolarisation ausgestattet und erwartet, dass das Master Gerat eine besitzt.

Sollte das Master Gerat am Bus von sich aus keine Leitungspolarisation besitzen, muss ein Widerstandspaar an die symmetrischen RS-485 Leitungen angebracht werden:

- Ein Pull-Up Widerstand zu einer 5V Spannung auf RS-485+ (D1) Leitung
- Ein Pull-Down Widerstand zu Masse (GND) auf der RS-485- (D0) Leitung

Der Wert dieser Widerstande muss zwischen 450 Ohm und 650 Ohm liegen. Ein 650 Ohm Widerstand erlaubt eine hohere Anzahl an Geraten am Bus.

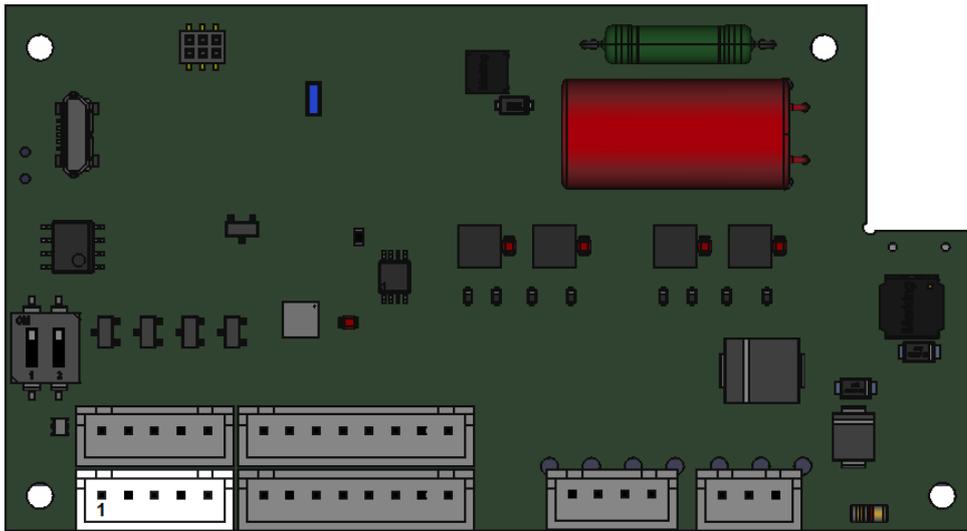
In diesem Fall muss eine Leitungspolarisation an einer Stelle fur den gesamten seriellen Bus angebracht werden. Generell sollte dieser Punkt an dem Master Gerat oder seinem Anschluss sein. Alle anderen Gerate mussen dann keine Leitungspolarisation mehr umsetzen.



3.6.3 X2 – CANopen/RS-485 OUT

Anschluss fur CANopen und RS-485 (Modbus RTU). Typ: JST B5B-XH-A

Pin 1 ist in der nachfolgenden Abbildung mit einer "1" markiert.

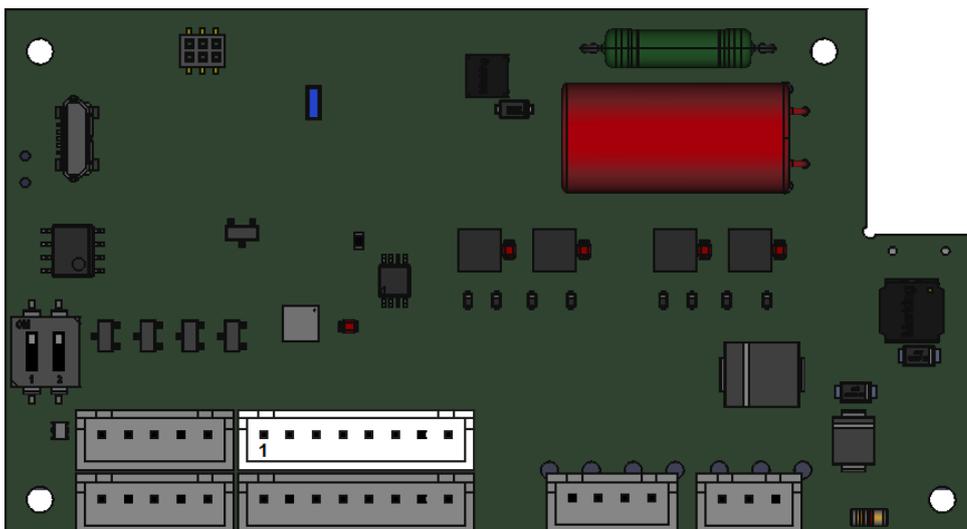


Pin	Funktion	Bemerkung
1	RS-485-	
2	RS-485+	
3	CAN_L	CAN-Low
4	CAN_H	CAN-High
5	GND	

3.6.4 X3 – Encoder/Hall-Sensor

Typ: JST B8B-XH-A

Pin 1 ist in der nachfolgenden Abbildung mit einer "1" markiert.



Pin	Funktion	Bemerkung
1	+5 V DC	Versorgungsspannung für Encoder/Hall Sensor, max. 200 mA
2	A	5 V Signal, max 1 MHz
3	B	5 V Signal, max 1 MHz
4	Index	5 V Signal
5	H1	5 V Signal
6	H2	5 V Signal
7	H3	5 V Signal
8	GND	

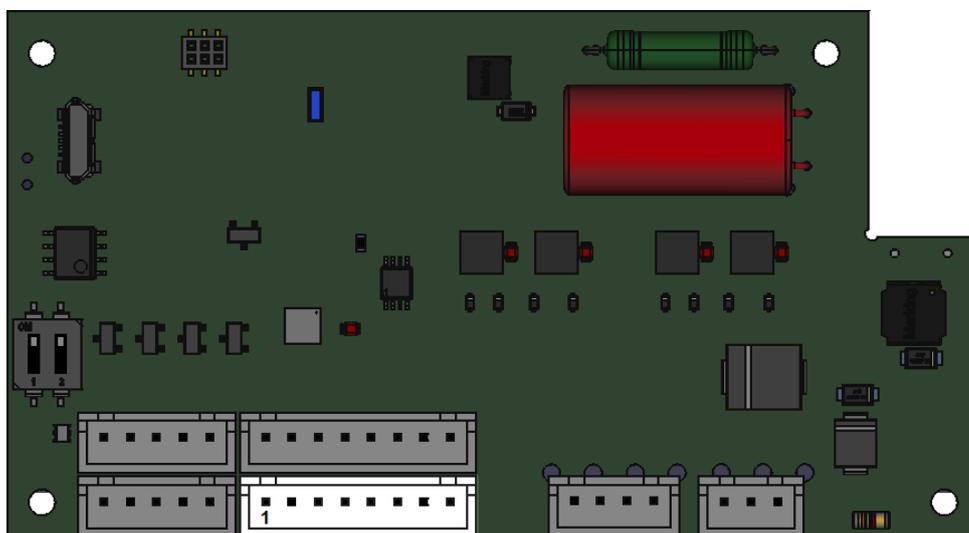
Es gelten folgende Schaltschwellen für die Encoder-Eingänge:

Max. Spannung	Scharschwellen	
	Ein	Aus
5 V	> 2 V	< 0,8 V

3.6.5 X4 – Ein- und Ausgänge

Typ: JST B8B-XH-A

Pin 1 ist in der nachfolgenden Abbildung mit einer "1" markiert.



Pin	Funktion	Bemerkung
1	Digitaler Eingang 1	
2	Digitaler Eingang 2	
3	Digitaler Eingang 3	Richtungseingang in Takt-Richtungs-Modus, max. 1 MHz
4	Digitaler Eingang 4	Takteingang in Takt-Richtungs-Modus, max. 1 MHz

Pin	Funktion	Bemerkung
5	Analoger Eingang 1	10 Bit, 0 - 10 V
6	Digitaler Ausgang 1	positiv schaltend (<i>High-Side-Switch</i>), max. 100 mA
7	Digitaler Ausgang 2	positiv schaltend (<i>High-Side-Switch</i>), max. 100 mA
8	GND	



Hinweis

Um die digitalen Ausgänge benutzen zu können, müssen Sie eine Spannung (12 ...30 V DC) an den Pin 2 von **X6** anschließen (Logikversorgung).

Die typische Ausgangsspannung entspricht der angeschlossenen Logikversorgung – 0,6 V. Der Strom soll 100 mA nicht überschreiten.

Für die digitalen Eingänge der Varianten mit den Artikelnummern *CL4-E-1-12* und *CL4-E-2-12* gelten folgende Schaltschwellen:

Max. Spannung	Schaltschwellen	
	Ein	Aus
24 V	> 9 V	< 3,7 V

Für die digitalen Eingänge der Varianten mit den Artikelnummern *CL4-E-1-12-5VDI* und *CL4-E-2-12-5VDI* gelten folgende Schaltschwellen:

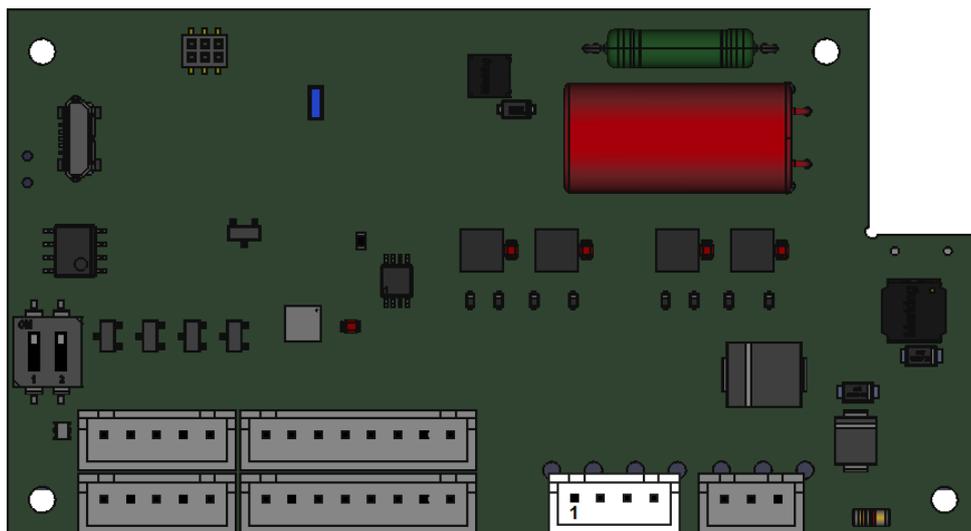
Max. Spannung	Schaltschwellen	
	Ein	Aus
5 V	> 2 V	< 0,8 V

3.6.6 X5 – Motoranschluss

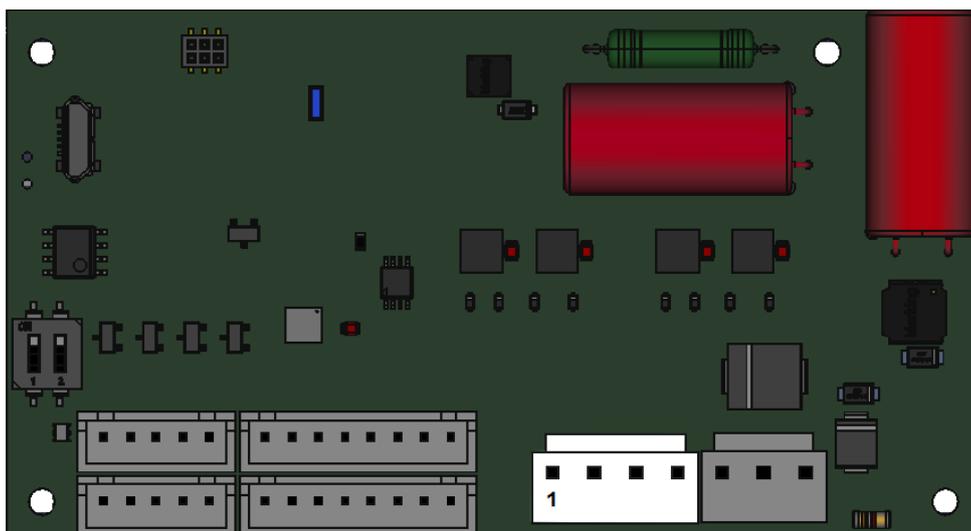
Pin 1 ist in den nachfolgenden Abbildungen mit einer "1" markiert.

Typ:

- CL4-E-1-xx *low current*. JST B4B-XH-A



- CL4-E-2-xx high current: JST B4P-VH



Pin	Funktion Schrittmotor	Funktion BLDC	Bemerkung
1	A	U	
2	A\	V	
3	B	W	
4	B\	n.c.	

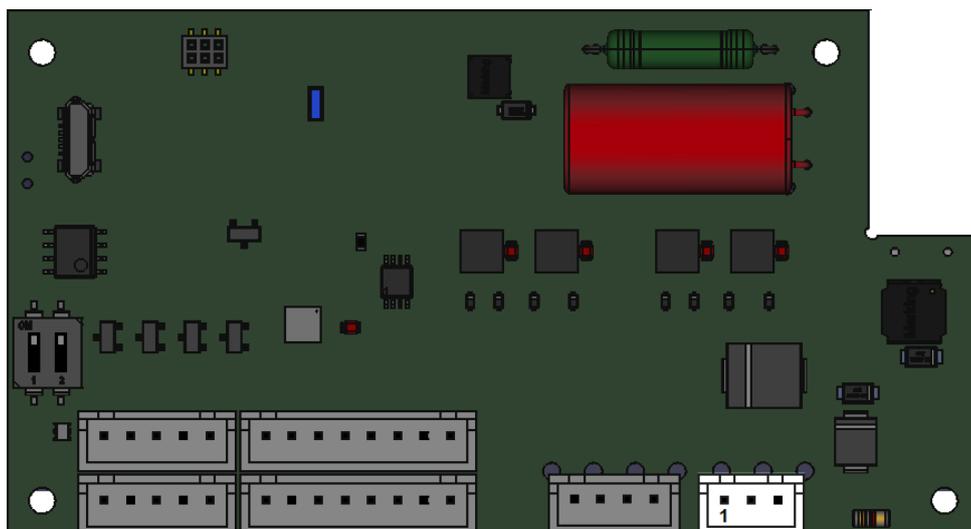
3.6.7 X6 – Spannungsversorgung

Anschluss für die Hauptversorgung und die Logikversorgung.

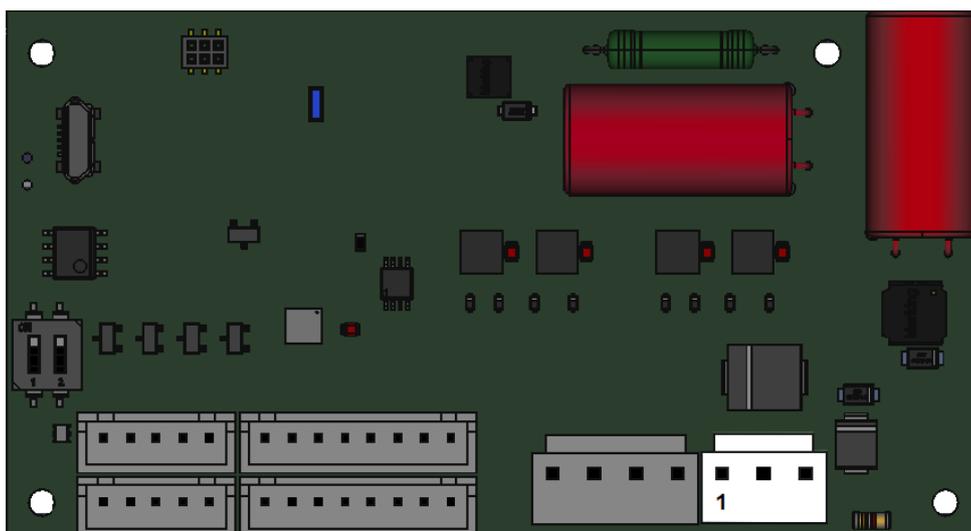
Pin 1 ist in den nachfolgenden Abbildungen mit einer "1" markiert.

Typ:

- CL4-E-1-xx *low current*. JST B3B-XH-A



- CL4-E-2-xx *high current*. JST B3P-VH



Spannungsquelle

Die Betriebs- oder Versorgungsspannung liefert eine Batterie, ein Transformator mit Gleichrichtung und Siebung, oder ein Schaltnetzteil.



Hinweis

- EMV: Bei einer DC-Stromversorgungsleitung mit einer Länge von >30 m oder Verwendung des Motors an einem DC-Bus sind zusätzliche Entstör- und Schutzmaßnahmen notwendig.
- Ein EMI-Filter ist in die DC-Zuleitung mit möglichst geringem Abstand zur Steuerung/Motor einzufügen.
- Lange Daten- oder Versorgungsleitungen sind durch Ferrite zu führen.
- Ein Kondensator von mindestens 4700 µF ist an die Versorgungsspannung (parallel) anzuschließen, so nah wie möglich an der Steuerung.

Pin-Belegung

Pin	Funktion	Bemerkung
1	+Ub	12 ...58 V DC
2	+UB Logic	12 ...30 V DC, Eingangsspannung für die optionale Logikversorgung und die digitalen Ausgänge Stromverbrauch ohne die Ausgänge: ca. 27 mA
3	GND	

Zulässige Betriebsspannung

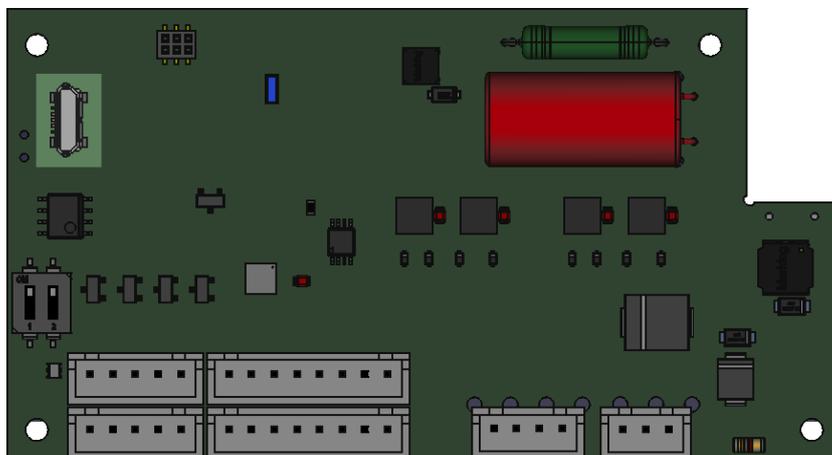
Die maximale Betriebsspannung beträgt 59 V DC. Steigt die Eingangsspannung der Steuerung über diesen Schwellwert, wird der Motor abgeschaltet und ein Fehler ausgelöst. Ab 60 V wird die integrierte Ballast-Schaltung (3W Leistung) aktiviert.

Die minimale Betriebsspannung beträgt 10 V DC. Fällt die Eingangsspannung der Steuerung unter diesen Schwellwert, wird der Motor abgeschaltet und ein Fehler ausgelöst.

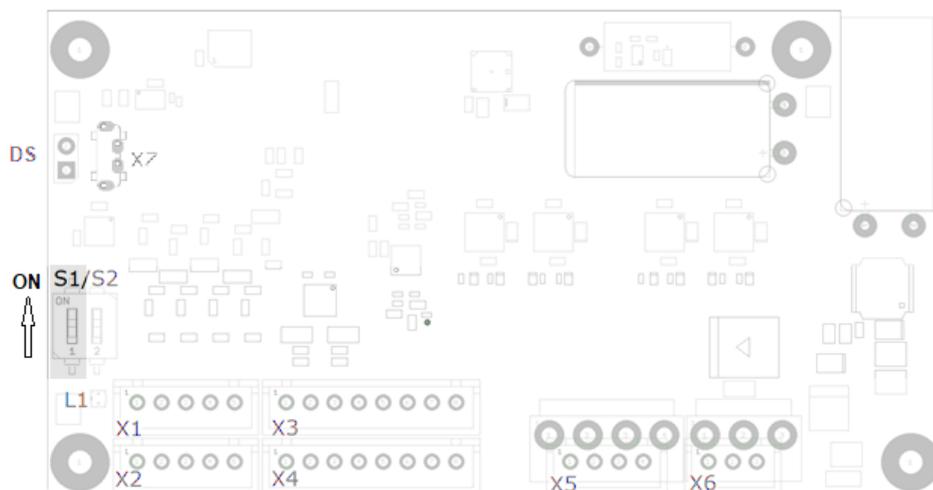
An die Versorgungsspannung muss ein Ladekondensator von mindestens 4700 µF (ca. 1000 µF pro Ampere Nennstrom) angeschlossen sein, um ein Überschreiten der zulässigen Betriebsspannung (z.B. beim Bremsvorgang) zu vermeiden.

3.6.8 X7 – Micro-USB

Für diesen USB-Anschluss wird ein Kabel des Typs "Micro-USB" benötigt.

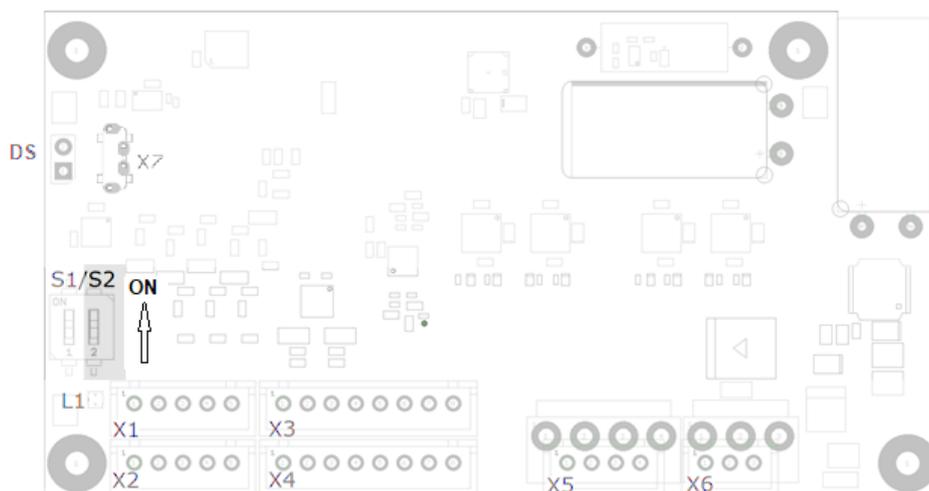


3.6.9 Schalter S1 – Terminierungswiderstand RS-485



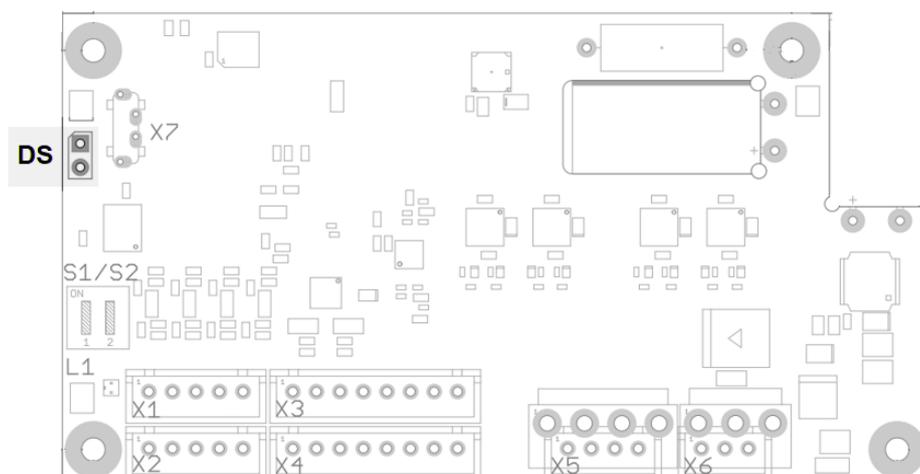
Mit diesem DIP-Schalter kann eine Terminierung mit 150 Ohm zwischen RS-485- und RS-485+ ein- (Schalter auf ON) oder ausgeschaltet werden.

3.6.10 Schalter S2 – Terminierungswiderstand CANopen



Mit diesem DIP-Schalter kann eine Terminierung mit 120 Ohm zwischen CAN-L und CAN_H ein- (Schalter auf ON) oder ausgeschaltet werden.

3.6.11 Schalter DS – Default-Switch



Durch Kurzschließen (z.B. mit einer Drahtbrücke) der beiden Kupferfreistellungen werden die Adresse und die Baudrate zurückgesetzt. Nach ca. 3 Sekunden startet die Steuerung selbstständig neu.

Folgende Objekte sind betroffen:

Objekt	Funktion	Werkseinstellung
2009_h	CANopen Node-ID	127
2005_h	CANopen Baudrate	136 (1 Mbaud)
2028_h	Modbus Slave-Adresse	5
202A_h	Modbus Baudrate	19200
202D_h	Modbus Parität	04 _h (Even)

4 Inbetriebnahme

In diesem Kapitel wird beschrieben, wie Sie die Kommunikation zur Steuerung aufbauen und die notwendigen Parameter einstellen, damit der Motor betriebsbereit ist. Sie können die Steuerung über USB, CANopen oder Modbus RTU (RS-485) konfigurieren.

Die Software *Plug & Drive Studio* bietet Ihnen eine Möglichkeit, die Konfiguration vorzunehmen und die Steuerung an den angeschlossenen Motor anzupassen. Weiterführende Informationen finden Sie im Dokument *Plug & Drive Studio: Quick Start Guide* auf www.nanotec.de.

Beachten Sie folgenden Hinweis:



Hinweis

- EMV: Stromführende Leitungen – insbesondere um Versorgungs- und Motorenleitungen – erzeugen elektromagnetische Wechselfelder.
- Diese können den Motor und andere Geräte stören. Nanotec empfiehlt folgende Maßnahmen:
- Geschirmte Leitungen verwenden und den Leitungsschirm beidseitig auf kurzem Weg erden.
- Kabel mit paarweise verdrehten Adern verwenden.
- Stromversorgungs- und Motorleitungen so kurz wie möglich halten.
- Motorgehäuse großflächig auf kurzem Weg erden.
- Versorgungs-, Motor- und Steuerleitungen getrennt verlegen.

4.1 Konfiguration über USB

4.1.1 Allgemeines

Es gibt folgende Möglichkeiten, die Steuerung über USB zu konfigurieren:

Konfigurationsdatei

Diese Datei lässt sich mittels dem USB-Anschluss auf die Steuerung speichern. Lesen Sie dazu die Kapitel **USB Anschluss** und **Konfigurationsdatei**.

NanoJ-Programm

Dieses Programm lässt sich mit *NanoJ* programmieren, kompilieren und anschließend über USB auf die Steuerung übertragen. Lesen Sie dazu die Kapitel **NanoJ-Programm** und **Programmierung mit NanoJ**.

Nach dem Anschließen an eine Spannungsversorgung liest die Steuerung die Konfiguration in folgender Reihenfolge aus:

1. Die Konfigurationsdatei wird ausgelesen und verarbeitet.
2. Das NanoJ-Programm wird gestartet.

4.1.2 USB-Anschluss

Wird die Steuerung über ein USB-Kabel mit einem PC verbunden, verhält sich die Steuerung wie ein Wechseldatenträger. Es werden keine weiteren Treiber benötigt.

Es werden drei Dateien angezeigt, die Konfigurationsdatei (`cfg.txt`), das *NanoJ-Programm* (`vmmcode.usr`) und die Informationsdatei (`info.bin`), wo die Seriennummer und Firmware-Version des Produkts zu finden sind.

Sie können somit die Konfigurationsdatei oder das *NanoJ-Programm* auf die Steuerung speichern. Die Spannungsversorgung der Steuerung muss beim USB-Betrieb ebenfalls angeschlossen sein.



Hinweis

- Benutzen Sie ausschließlich ein standardisiertes Micro-USB-Kabel. Benutzen Sie keinesfalls USB-Kabel, die Hersteller von Mobiltelefonen ihren Produkten beilegen. Diese USB-Kabel können eine andere Steckerform oder Pin-Belegung aufweisen.
- Speichern Sie keine anderen Dateien auf der Steuerung als die nachfolgend aufgelisteten:
 1. `cfg.txt`
 2. `vmmcode.usr`
 3. `info.bin`
 4. `reset.txt`
 5. `firmware.bin`

Jede andere Datei wird beim Einschalten der Spannungsversorgung der Steuerung gelöscht!



Tipp

Da es bei der Inbetriebnahme häufig vorkommt, dass die gleiche Datei nach einer Aktualisierung wieder auf die Steuerung kopiert wird, empfiehlt es sich, eine Skript-Datei zu verwenden, die diese Arbeit erledigt.

- Unter Windows können Sie sich eine Text-Datei mit der Dateierdung `bat` und folgendem Inhalt erzeugen:

```
copy <QUELLE> <ZIEL>
```

- Unter Linux können Sie sich ein Skript mit der Dateierdung `sh` und folgendem Inhalt erzeugen:

```
#!/bin/bash  
cp <QUELLE> <ZIEL>
```

4.1.3 Konfigurationsdatei

Allgemeines

Die Konfigurationsdatei `cfg.txt` dient dazu, Werte für das Objektverzeichnis beim Start auf einen bestimmten Wert vorzubelegen. Diese Datei ist in einer speziellen Syntax gehalten, um den Zugriff auf die Objekte des Objektverzeichnisses möglichst einfach zu gestalten. Die Steuerung wertet alle Zuweisungen in der Datei von oben nach unten aus.



Hinweis

Sollten Sie die Konfigurationsdatei löschen, wird bei dem nächsten Neustart der Steuerung die Datei neu (ohne Inhalt) erstellt.

Lesen und Schreiben der Datei

So erhalten Sie Zugriff auf die Datei:

1. Schließen Sie die Spannungsversorgung an und schalten Sie die Spannungsversorgung ein.
2. Verbinden Sie die Steuerung mit Ihrem PC über das USB-Kabel.
3. Nachdem der PC das Gerät als Wechseldatenträger erkannt hat, navigieren Sie im Explorer das Verzeichnis der Steuerung an. Dort ist die Datei `cfg.txt` (im Falle einer PD4C heißt die Datei `pd4ccfg.txt`) hinterlegt.
4. Öffnen Sie diese Datei mit einem einfachen Text-Editor, wie Notepad oder Vi. Benutzen Sie keine Programme, welche Textauszeichnung benutzen (LibreOffice oder dergleichen).

Nachdem Sie Änderungen an der Datei vorgenommen haben, gehen Sie wie folgt vor, um die Änderungen wirksam werden zu lassen:

1. Speichern Sie die Datei, falls nicht schon geschehen.
2. Trennen Sie das USB-Kabel von der Steuerung.
3. Trennen Sie die Spannungsversorgung der Steuerung für ca. 1 Sekunde, bis die Betriebs-LED aufhört zu blinken.
4. Verbinden Sie die Spannungsversorgung wieder. Mit diesem Start der Steuerung werden die neuen Werte der Konfigurationsdatei ausgelesen und wirksam.



Tip

Um die Steuerung neu zu starten, können Sie auch eine leere Datei `reset.txt` auf die Steuerung kopieren. Damit startet die Steuerung neu. Die Datei `reset.txt` wird beim Neustart gelöscht.

Aufbau der Konfigurationsdatei

Kommentare

Zeilen, welche mit einem Semikolon beginnen, werden von der Steuerung ignoriert.

Beispiel

```
; Dies ist eine Kommentarzeile
```

Zuweisungen



Hinweis

Informieren Sie sich vor dem Setzen eines Wertes über dessen Datentyp (siehe Kapitel **Objektverzeichnis Beschreibung**)! Die Steuerung validiert keine Einträge auf logische Fehler!

Werte im Objektverzeichnis lassen sich mit folgender Syntax setzen:

```
<Index>:<Subindex>=<Wert>
```

<Index>

Dieser Wert entspricht dem Index des Objektes und wird als Hexadezimalzahl interpretiert. Der Wert muss immer vierstellig angegeben werden.

<Subindex>

Dieser Wert entspricht dem Subindex des Objektes und wird als Hexadezimalzahl interpretiert. Der Wert muss immer zweistellig angegeben werden und kann entfallen wenn der Subindex `00h` ist.

<Wert>

Der Wert, der in das Objekt geschrieben werden soll, wird als Dezimalzahl interpretiert. Für Hexadezimalzahlen ist ein "0x" voranzustellen.

Sie können auch einzelne Bits setzen:

Bit setzen

```
3202:00.3=1
```

Bit zurücksetzen

```
3202:00.3=0
```

Bitweise OR

```
3202:00|=0x08
```

Bitweise AND

```
3202:00&=0x08
```

Beispiel

Setzen des Objekts 203B_h:01 (Nennstrom) auf den Wert "600" (mA):

```
203B:01=600
```

Setzen des Objekts 3202_h:00 auf den Wert "8" (Stromabsenkung im Stillstand in *Open Loop* aktivieren):

```
3202:00=8
```

oder nur Bit 3 setzen

```
3202:00.3=1
```



Hinweis

- Links und rechts vom Gleichheitszeichen dürfen sich keine Leerzeichen befinden. Folgende Zuweisungen sind nicht korrekt:
6040:00 =5
6040:00= 5
6040:00 = 5
- Die Anzahl der Stellen darf nicht verändert werden. Der Index muss vier, der Subindex zweistellig sein. Folgende Zuweisungen sind nicht korrekt
6040:0=6
6040=6
- Leerzeichen am Anfang der Zeile sind nicht zulässig.

4.1.4 NanoJ-Programm

Auf der Steuerung kann ein *NanoJ-Programm* ausgeführt werden. Um ein Programm auf die Steuerung zu laden und zu starten, gehen Sie nach folgenden Schritten vor:

1. Schreiben und kompilieren Sie Ihr Programm, wie es in Kapitel **Programmierung mit NanoJ** beschrieben ist.

2. Schließen Sie die Spannungsversorgung an die Steuerung an und schalten Sie die Spannungsversorgung ein.
3. Verbinden Sie die Steuerung mit Ihrem PC über das USB-Kabel.
4. Nachdem der PC das Gerät als Wechseldatenträger erkannt hat, öffnen Sie einen Explorer und löschen Sie auf der Steuerung die Datei `vmmcode.usr`.
5. Navigieren Sie im Explorer in das Verzeichnis mit Ihrem Programm. Die compilierte Datei hat den gleichen Namen wie die Sourcecode-Datei, nur mit der Dateinamen-Endung `.usr`. Benennen Sie diese Datei in `vmmcode.usr` um.
6. Kopieren Sie die Datei `vmmcode.usr` auf die Steuerung.
7. Trennen Sie die Spannungsversorgung der Steuerung für ca. 1 Sekunde, bis die Betriebs-LED aufhört zu blinken.
8. Verbinden Sie die Spannungsversorgung wieder. Mit diesem Start der Steuerung wird das neue *NanoJ-Programm* eingelesen und gestartet.



Tip

Um die Steuerung neu zu starten, können Sie auch eine leere Datei `reset.txt` auf die Steuerung kopieren. Damit startet die Steuerung neu. Die Datei `reset.txt` wird beim Neustart gelöscht.



Hinweis

- Das *NanoJ-Programm* auf der Steuerung muss den Dateinamen `vmmcode.usr` haben.
- Falls das *NanoJ-Programm* gelöscht wurde, wird mit dem nächsten Start eine leere Datei namens `vmmcode.usr` angelegt.



Tip

Das Löschen des alten *NanoJ-Programms* und das Kopieren des neuen lässt sich mit einer Skript-Datei automatisieren:

- Unter Windows können Sie sich eine Datei mit der Dateiendung `bat` und folgendem Inhalt erzeugen:

```
copy <QUELLPFAD>\<OUTPUT>.usr <ZIEL>:\vmmcode.usr
```

Also zum Beispiel:

```
copy c:\test\main.usr n:\vmmcode.usr
```

- Unter Linux können Sie sich ein Skript mit der Dateiendung `sh` und folgendem Inhalt erzeugen:

```
#!/bin/bash  
cp <QUELLPFAD>/<OUTPUT>.usr <ZIELPFAD>/vmmcode.usr
```

4.2 Konfiguration über CANopen

Alle Einstellungen für CANopen können über den Speicher-Mechanismus geschrieben werden (siehe dazu Kapitel **Objekte speichern**).

4.2.1 Kommunikationseinstellungen

In den folgenden Kapiteln wird beschrieben, wie Sie die Kommunikationseinstellungen ändern können.

Ab Werk ist die Steuerung für die Node-ID 127 und eine Baudrate von 1 Mbaud konfiguriert.

Mittels des **Default-Switch** können Sie die Steuerung in den Auslieferungszustand zurücksetzen.

Node-ID und Baudrate einstellen

Die Steuerung startet in der Standardeinstellung mit einer Node-ID von 127. Falls eine andere Node-ID benötigt wird, wird der neue Wert der Node-ID in das Objekt **2009_h** eingetragen.

Die Steuerung startet in der Standardeinstellung mit der Baudrate von 1 MBd. Die Baudrate wird in das Objekt **2005_h** eingetragen. Der Wert für die entsprechende Baudrate kann aus nachfolgender Tabelle entnommen werden.

Wert		Baudrate in kBd
dec	hex	
129	81	10
130	82	20
131	83	50
132	84	125
133	85	250
134	86	500
136	88	1000

Sie müssen die Änderungen speichern, indem Sie den Wert "65766173_h" in das Objekt 1010_h:0A_h schreiben. Die Änderungen werden erst nach einem Neustart der Steuerung übernommen.

4.2.2 Kommunikation aufbauen

CANopen

Vor Beginn der Inbetriebnahme wird empfohlen, die Kapitel **Anschlussbelegung** und Konfiguration **CANopen** durchzulesen.

1. Verbinden Sie den CANopen-Master mit der Steuerung über die CAN_L- und CAN_H-Leitungen. Überprüfen Sie den Anschluss von Ihrem CAN-GND und dass der notwendige **120 Ohm Terminierungswiderstand** zwischen CAN_H und CAN_L vorhanden ist.
2. Versorgen Sie die Steuerung mit Spannung.
3. Ändern Sie ggf. die Konfigurationswerte, siehe Konfiguration **CANopen**.
Ab Werk ist die Steuerung auf Node-ID 127, Baudrate 1 MBaud eingestellt.
4. Zum Testen der Schnittstelle senden Sie die Bytes 40 41 60 00 00 00 00 00 an die Steuerung.
Das Statusword (6041_h) wurde ausgelesen, Sie erhalten diese Antwort: 4B 41 60 00 XX XX 00 00.

4.2.3 Einstellung mittels LSS

Die *Layer Setting Services* bieten Ihnen die Möglichkeit, die Node-ID und/oder die Baudrate dynamisch über den CANopen-Bus zu vergeben. Siehe Kapitel **LSS-Protokoll**.

4.3 Konfiguration über Modbus RTU

In den folgenden Kapiteln wird beschrieben, wie Sie die Kommunikation aufbauen.

Ab Werk ist die Steuerung ist auf Slave-Adresse 5, Baudrate 19200 Baud, even Parity, 1 Stop Bit eingestellt. Mittels des **Default-Switch** können Sie die Steuerung in den Auslieferungszustand zurücksetzen.

4.3.1 Kommunikationseinstellungen

Folgende Einstellungen können vorgenommen werden:

Konfiguration	Objekt	Wertebereich	Werkseinstellung
Slave Adresse	2028_h	1 bis 247	5
Baudrate	202A_h	7200 bis 256000	19200
Parity	202D_h	<ul style="list-style-type: none"> • None: 0x00 • Even: 0x04 • Odd: 0x06 	0x04 (Even)

Die Anzahl der Datenbits ist dabei immer "8". Die Anzahl der Stop-Bits ist abhängig von der Parity-Einstellung:

- Keine Parity: 2 Stop Bits
- "Even" oder "Odd" Parity: 1 Stop Bit

Unterstützt werden folgende Baudraten:

- 7200
- 9600
- 14400
- 19200
- 38400
- 56000
- 57600
- 115200
- 128000
- 256000

Sie müssen die Änderungen speichern, indem Sie den Wert "65766173_h" in das Objekt 1010_h:0B_h schreiben. Die Änderungen werden erst nach einem Neustart der Steuerung übernommen.

4.3.2 Kommunikation aufbauen

1. Verbinden Sie den *Modbus-Master* mit der Steuerung über die RS-485 + und RS-485- (siehe **X1 – CANopen/RS-485 IN**) Leitungen.
2. Versorgen Sie die Steuerung mit Spannung.
3. Ändern Sie ggf. die Konfigurationswerte.
Ab Werk ist die Steuerung auf Slave Address 5, Baudrate 19200 Baud, even Parity, 1 Stop Bit eingestellt.
4. Zum Testen der Schnittstelle senden Sie die Bytes 05 65 55 00 2F A7 an die Steuerung (eine detaillierte Beschreibung der Modbus-Funktionscodes finden Sie im Kapitel **Modbus RTU**). Das Objektverzeichnis wird ausgelesen.

4.4 Motordaten einstellen

Die Steuerung benötigt vor der Inbetriebnahme des Motors einige Werte aus dem Motordatenblatt.

- Polpaarzahl: Objekt **2030_h:00_h** (Pole pair count) Hier ist die Anzahl der Motorpolpaare einzutragen. Bei einem Schrittmotor wird die Polpaarzahl über den Schrittwinkel berechnet, z.B. $1,8^\circ = 50$ Polpaare, $0,9^\circ = 100$ Polpaare (siehe Schrittwinkel im Motordatenblatt). Bei BLDC-Motoren ist die Polpaarzahl direkt im Motordatenblatt angegeben.
- Objekt **2031_h:00_h**: Maximal zulässiger Motorstrom (Motorschutz) in mA (siehe Motordatenblatt)

- Objekt **6075_h:00_h** Nennstrom des Motors in mA (siehe Motordatenblatt)
- Objekt **6073_h:00_h**: Maximaler Strom (bei einem Schrittmotor entspricht in der Regel dem Nennstrom, Bipolar) in Promille des eingestellten Nennstroms (siehe Motordatenblatt). Werkseinstellung: "1000", was 100% entspricht.
- Objekt **203B_h:02_h** Maximale Dauer des maximalen Stroms (**6073_h**) in ms (für die Erstinbetriebnahme empfiehlt Nanotec einen Wert von 100 Millisekunden; Dieser Wert ist später an die konkrete Applikation anzupassen).
- Motortyp einstellen:
 - Schrittmotor:
 - Objekt **3202_h:00_h** (Motor Drive Submode Select): Definiert den Motortyp Schrittmotor, aktiviert die Stromabsenkung bei Stillstand des Motors: 0000008h. Siehe auch Kapitel **Inbetriebnahme Open Loop**.
 - BLDC-Motor:
 - Objekt **3202_h:00_h** (Motor Drive Submode Select): Definiert den Motortyp BLDC: 00000041h

4.5 Motor anschließen

Nach der Einstellung der Motorparameter, siehe **Motordaten einstellen**, schließen Sie den Motor und ggf. die vorhandenen Sensoren (Encoder/Hallsensoren) an.

- Motor anschließen:
 - an den Anschluss X5, siehe **X5 – Motoranschluss**
- Encoder/Hallsensoren anschließen:
 - an den Anschluss X3, siehe **X3 – Encoder/Hall-Sensor**

4.6 Auto-Setup

Um einige Parameter im Bezug zum Motor und den angeschlossenen Sensoren (Encoder/Hallsensoren) zu ermitteln, müssen Sie ein Auto-Setup durchführen. Der **Closed Loop**-Betrieb setzt ein erfolgreich abgeschlossenes Auto-Setup voraus.



Tipp

Solange sich der an der Steuerung angeschlossene Motor oder die Sensoren für die Rückführung (Encoder/Hallsensoren) nicht ändern, ist das Auto-Setup nur einmal bei der Erstinbetriebnahme durchzuführen.



Hinweis

- Beachten Sie die folgenden Voraussetzungen für das Durchführen des Auto-Setups:
- Der Motor muss lastfrei sein.
- Der Motor darf nicht berührt werden.
- Der Motor muss sich frei in beliebige Richtungen drehen können.
- Es darf kein NanoJ-Programm laufen (Objekt 2300_h:00_h Bit 0 = "0", siehe **2300h NanoJ Control**).



Tipp

Die Ausführung des Auto-Setups benötigt relativ viel Prozessorrechenleistung. Während des Auto-Setups können dadurch eventuell die Feldbusse nicht zeitgerecht bedient werden.

4.6.1 Parameter-Ermittlung

Das Auto-Setup ermittelt über mehrere Test- und Messläufe verschiedene Parameter des angeschlossenen Motors und der vorhandenen Sensoren. Art und Anzahl der Parameter sind teilweise von der jeweiligen Motorkonfiguration abhängig.

Parameter	Alle Motoren unabhängig von der Konfiguration
Motortyp (Schrittmotor oder BLDC-Motor)	X
Wicklungswiderstand	X
Wicklungsinduktivität	X
Verkettungsfluss	X

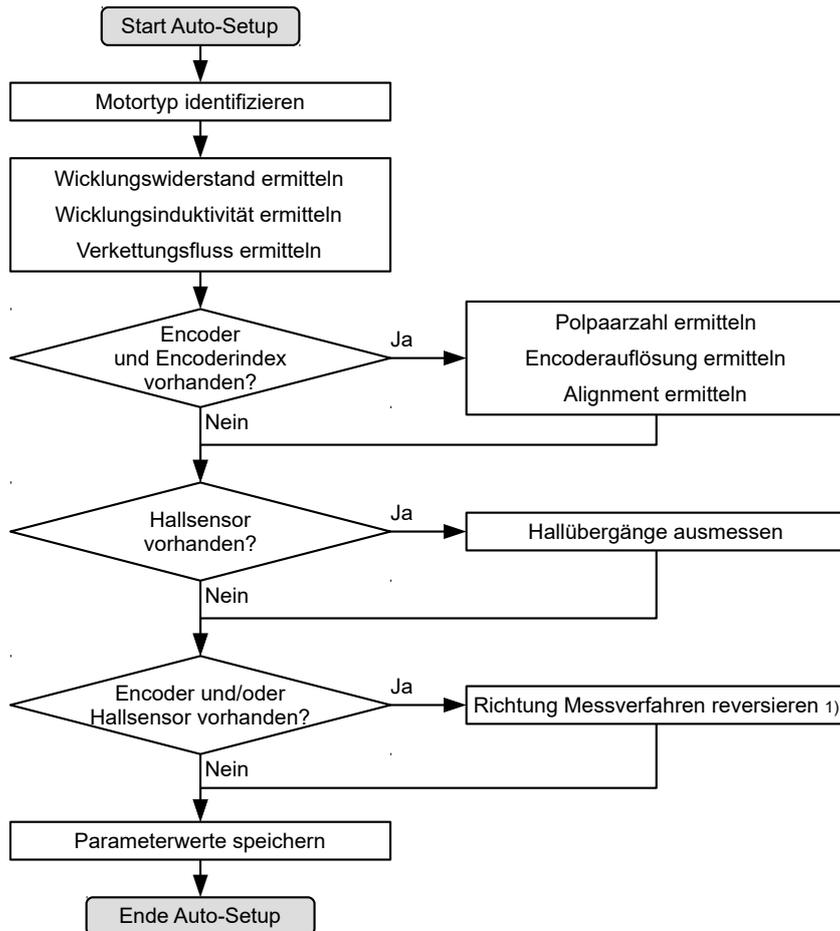
Parameter	Motor ohne Encoder	Motor mit Encoder und Index	Motor mit Encoder ohne Index
Encoderauflösung	-	X	---
Alignment (Verschiebung des elektrischen Nullpunkts zum Index.)	-	X	---

Parameter	Motor ohne Hallsensor	Motor mit Hallsensor
Hallübergänge	-	X

4.6.2 Durchführung

Stellen Sie vor der Durchführung des *Auto-Setups* sicher, dass Sie die notwendigen Parameter richtig eingestellt haben (siehe **Motordaten einstellen**).

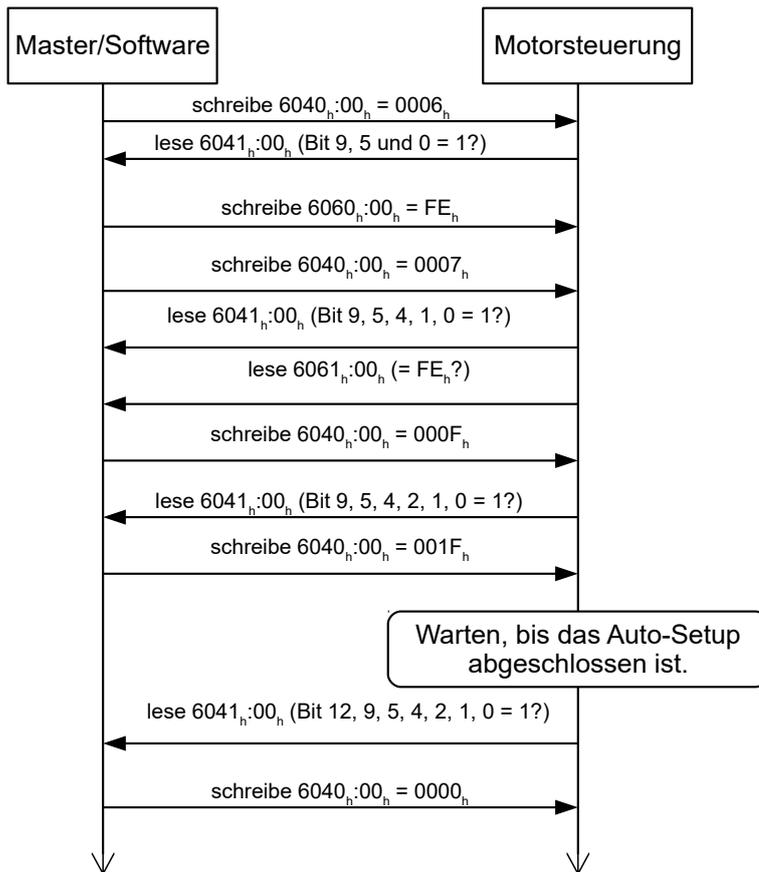
1. Zum Vorwählen des Betriebsmodus *Auto-Setup* tragen Sie in das Objekt 6060_h:00_h den Wert "-2" ("FE_h") ein.
Die *Power state machine* muss nun in den Zustand *Operation enabled* versetzt werden, siehe **CiA 402 Power State Machine**.
2. Starten Sie das *Auto-Setup* mit Setzen von Bit 4 *OMS* im Objekt 6040_h:00_h (Controlword).



Während der Ausführung des Auto-Setups werden nacheinander folgende Tests und Messungen durchgeführt:

1) Zum Ermitteln der Werte wird die Richtung des Messverfahrens reversiert und die Flankenerkennung erneut ausgewertet.

Der Wert 1 im Bit 12 OMS im Objekt 6041_h:00_h (Statusword) zeigt an, dass das Auto-Setup vollständig durchgeführt und beendet wurde. Zusätzlich kann über das Bit 10 TARG im Objekt 6041_h:00_h abgefragt werden, ob ein Encoder-Index gefunden wurde (= "1") oder nicht (= "0").



4.6.3 Parameterspeicherung

Nach erfolgreichem *Auto-Setup* werden die ermittelten Parameterwerte automatisch in die zugehörigen Objekte übernommen und mit dem Speichermechanismus gespeichert, siehe **Objekte speichern** und **1010h Store Parameters**. Benutzt werden die Kategorien *Drive* 1010_h:05_h und *Tuning* 1010_h:06_h.



VORSICHT

Unkontrollierte Motorbewegungen!

Das interne Koordinatensystem ist nach dem Auto-Setup nicht mehr gültig. Es kann zu unvorhersehbaren Reaktionen kommen.

- ▶ Starten Sie das Gerät nach einem Auto-Setup neu. Homing alleine genügt nicht.

4.7 Testlauf

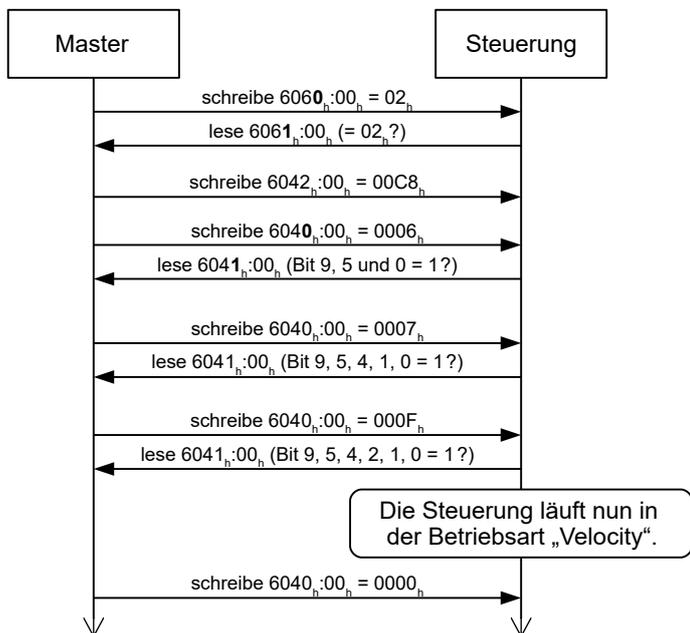
Nach der Konfiguration und dem Auto-Setup kann ein Testlauf durchgeführt werden. Beispielhaft wird der Betriebsmodus **Velocity** angewendet.

Die Werte werden von Ihrem *CANopen-Master* oder *Modbus-Master* an die Steuerung übertragen. Dabei sollte der *Master* nach jeder Übertragung über Status-Objekte der Steuerung die erfolgreiche Parametrierung überprüfen.

1. Wählen Sie den Modus *Velocity*, indem Sie das Objekt **6060_h** (Modes Of Operation) auf den Wert "2" setzen.
2. Schreiben Sie die gewünschte Drehzahl in **6042_h**.

3. Versetzen Sie die *Power state machine* in den Zustand *Operation enabled*, siehe **CiA 402 Power State Machine**.

Folgender Ablauf startet den *Velocity* Modus, der Motor dreht dabei mit 200 U/min.



4. Um den Motor zu stoppen, setzen Sie das Controlword (**6040_h**) auf "0".

5 Generelle Konzepte

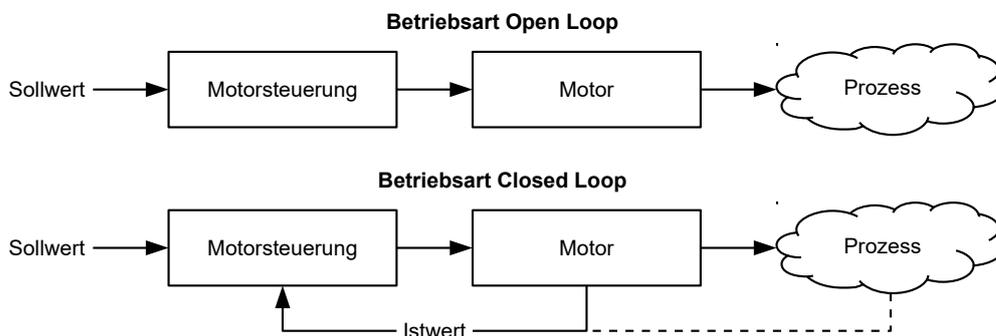
5.1 Betriebsarten

5.1.1 Allgemein

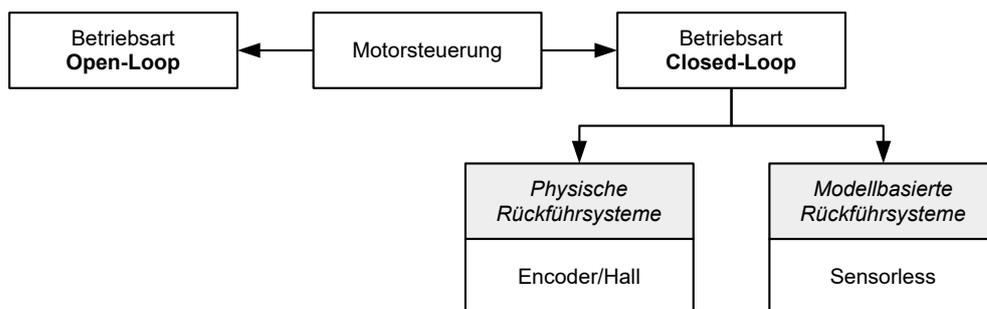
Die Betriebsart von Systemen ohne Rückführung wird als *Open Loop*, die mit Rückführung als *Closed Loop* bezeichnet. In der Betriebsart *Closed Loop* ist es zunächst unerheblich, ob die zurückgeführten Signale vom Motor selbst oder aus dem beeinflussten Prozess kommen.

Bei Steuerungen mit Rückführung wird die gemessene Regelgröße (Istwert) permanent mit einer Führungsgröße (Sollwert) verglichen. Bei Abweichungen zwischen diesen Größen regelt die Steuerung entsprechend den vorgegebenen Regelparametern nach.

Dagegen fehlt den reinen Steuerungen die Rückführung der zu regelnden Größe. Die Führungsgröße (Sollwert) wird lediglich vorgegeben.



Neben den physischen Rückführsystemen (beispielsweise über Encoder oder Hallsensoren) kommen auch modellbasierte Rückführsysteme, die alle unter dem Überbegriff *Sensorless* bekannt sind, zum Einsatz. Beide Rückführsystemen können auch in Kombination eingesetzt werden, um die Qualität der Regelung weiter zu verbessern.



Nachfolgend werden alle möglichen Kombinationen von Betriebsarten und Rückführsysteme im Bezug auf die Motorentchnik zusammengefasst. Die Unterstützung der jeweiligen Betriebsart und Rückführung ist steuerungsspezifisch und in den Kapiteln *Anschlussbelegung* und **Betriebsmodi** nachzulesen.

Betriebsart	Schrittmotor	BLDC-Motor
Open Loop	ja	nein
Closed Loop	ja	ja

Rückführung	Schrittmotor	BLDC-Motor
Hall	nein	ja
Encoder	ja	ja
Sensorless	ja	ja

In Abhängigkeit der Betriebsart können verschiedene Betriebsmodi angewendet werden. Die nachfolgende Liste fasst alle Betriebsmodi, die in den verschiedenen Betriebsarten möglich sind, zusammen.

Betriebsmodus	Betriebsart	
	Open Loop	Closed Loop
Profile Position	ja	ja
Velocity	ja	ja
Profile Velocity	ja	ja
Profile Torque	nein ¹⁾	ja
Homing	ja ²⁾	ja
Interpolated Position Mode	ja ³⁾	ja
Cyclic Synchronous Position	ja ³⁾	ja
Cyclic Synchronous Velocity	ja ³⁾	ja
Cyclic Synchronous Torque	nein ¹⁾	ja
Takt-Richtung	ja	ja

1) Die Drehmoment-Betriebsmodi **Profile Torque** und **Cyclic Synchronous Torque** sind in der Betriebsart *Open Loop* aufgrund einer fehlenden Rückführung nicht möglich.

2) Ausnahme: Homing auf Block ist aufgrund einer fehlenden Rückführung nicht möglich.

3) Da sich Rampen und Geschwindigkeiten in den Betriebsmodi **Cyclic Synchronous Position** und **Cyclic Synchronous Velocity** aus den vorgegebenen Punkten des Masters ergeben, ist es normalerweise nicht möglich, diese Parameter so vorzuwählen und zu erproben, dass ein Schritverlust ausgeschlossen werden kann. Es wird deshalb davon abgeraten, diese Betriebsmodi in Verbindung mit der Betriebsart *Open Loop* zu verwenden.

5.1.2 Open Loop

Einführung

Die Betriebsart *Open Loop* wird nur bei Schrittmotoren angewendet und ist ein reiner Stellbetrieb. Die Felddrehung im Stator wird durch die Steuerung vorgegeben. Der Rotor folgt der magnetischen Felddrehung ohne Schrittverluste unmittelbar, solange keine Grenzparameter - wie beispielsweise das maximal mögliche Drehmoment - überschritten werden. Im Vergleich zum *Closed Loop* werden keine komplexen internen Regelungsprozesse in der Steuerung benötigt. Dadurch sind die Anforderungen an die Steuerungshardware wie auch an die Steuerungslogik sehr gering. Im Besonderen bei preissensitiven Anwendungen und einfachen Bewegungsaufgaben wird deshalb die Betriebsart *Open Loop* vorwiegend eingesetzt.

Da es im Gegensatz zu *Closed Loop* keine Rückkopplung über die aktuelle Rotorposition gibt, kann auch kein Rückschluss auf das an der Abtriebsseite der Motorwelle anstehende Gegenmoment gezogen werden. Um eventuell an der Abtriebswelle des Motors auftretende Drehmomentschwankungen auszugleichen, liefert die Steuerung in der Betriebsart *Open Loop* über den gesamten Drehzahlbereich immer den maximal möglichen (bzw. durch Parameter vorgegebenen) eingestellten Strom an die Statorwicklungen. Die dadurch erzeugte hohe magnetische Feldstärke zwingt den Rotor, in kürzester Zeit den neuen Beharrungszustand einzunehmen. Diesem Moment

steht jedoch das Trägheitsmoment des Rotors entgegen. Unter bestimmten Betriebsbedingungen neigt diese Kombination zu Resonanzen, vergleichbar einem Feder-Masse-System.

Inbetriebnahme

Um die Betriebsart *Open Loop* anzuwenden, sind folgende Einstellungen notwendig:

- Im Objekt **2030_h** (Pole Pair Count) die Polpaarzahl eingeben (siehe Motordatenblatt: Ein Schrittwinkel von 1,8° entspricht bei einem Schrittmotor mit 2 Phasen 50 Polpaaren und von 0,9° entspricht 100 Polpaaren).
- Im Objekt **2031_h:00_h** den maximal zulässigen Motorstrom (Motorschutz) in mA eingeben (siehe Motordatenblatt)
- Im Objekt **6075_h:00_h** den Nennstrom des Motors in mA (siehe Motordatenblatt) eingeben.
- Im Objekt **6073_h:00_h**: den Maximalstrom (bei einem Schrittmotor entspricht in der Regel dem Nennstrom, Bipolar) in Promille des eingestellten Nennstroms eingeben (siehe Motordatenblatt). Werkseinstellung: "1000", was 100% entspricht.
- Im Objekt **3202_h** (Motor Drive Submode Select) das Bit 0 (CL/OL) mit dem Wert "0" belegen.
- Soll der Takt-Richtungs-Modus angewendet werden, dann Kapitel **Takt-Richtungs-Modus** berücksichtigen.

Bei Bedarf sollte die Stromabsenkung bei Stillstand des Motors aktiviert werden, um die Verlustleistung und Wärmeentwicklung zu reduzieren. Um die Stromabsenkung zu aktivieren, sind folgende Einstellungen notwendig:

- Im Objekt **3202_h** (Motor Drive Submode Select) das Bit 3 (CurRed) auf "1" setzen.
- Im Objekt **2036_h** (Open Loop Current Reduction Idle Time) wird die Zeit in Millisekunden angegeben, die sich der Motor im Stillstand befinden muss, bis die Stromabsenkung aktiviert wird.
- Im Objekt **2037_h** (Open Loop Current Reduction Value/factor) wird der Effektivwert angegeben, auf den der Nennstrom reduziert werden soll, wenn die Stromabsenkung im *Open Loop* aktiviert wird und sich der Motor im Stillstand befindet.

Optimierungen

Systembedingt können in der Betriebsart *Open Loop* Resonanzen auftreten, besonders bei geringer Belastung ist die Resonanzneigung hoch. Aus praktischen Erfahrungen heraus haben sich in Abhängigkeit der Applikation verschiedene Maßnahmen bewährt, um Resonanzen weitgehend zu reduzieren:

- Strom reduzieren oder erhöhen, siehe Objekte **6073_h** bzw. **203B_h**. Zu hohe Drehmomentreserve begünstigt Resonanzen.
- Die Betriebsspannung unter Berücksichtigung der produktspezifisch zugelassenen Bereiche reduzieren (bei genügender Drehmomentreserve) oder erhöhen. Der zulässige Betriebsspannungsbereich kann dem Produktdatenblatt entnommen werden.
- Die Regelparameter des Stromreglers über die Objekte **3210_h:09_h** (I_P) und **3210_h:0A_h** (I_L) optimieren.
- Anpassen der Beschleunigung, Verzögerung und/oder Zielgeschwindigkeit in Abhängigkeit des gewählten Betriebsmodus:

Betriebsmodus Profile Position

Objekte **6083_h** (Profile Acceleration), **6084_h** (Profile Deceleration) und **6081_h** (Profile Velocity).

Betriebsmodus Velocity

Objekte **6048_h** (Velocity Acceleration), **6049_h** (Velocity Deceleration) und **6042_h** (Target Velocity).

Betriebsmodus Profile Velocity

Objekte **6083_h** (Profile Acceleration), **6084_h** (Profile Deceleration) und **6081_h** (Profile Velocity).

Betriebsmodus Homing

Objekte **609A_h** (Homing Acceleration), **6099_h:01_h** (Speed During Search For Switch) und **6099_h:02_h** (Speed During Search For Zero).

Betriebsmodus Interpolated Position Mode

Mit der übergeordneten Steuerung können die Beschleunigungs- und Verzögerungsrampen beeinflusst werden.

Betriebsmodus Cycle Synchronous Position

Über die externen Zielvorgaben "Positionsvorgabe/Zeiteinheit" können die Beschleunigungs- und Verzögerungsrampen beeinflusst werden.

Betriebsmodus Cycle Synchronous Velocity

Über die externen Zielvorgaben "Positionsvorgabe/Zeiteinheit" können die Beschleunigungs- und Verzögerungsrampen beeinflusst werden.

Betriebsmodus Takt-Richtung

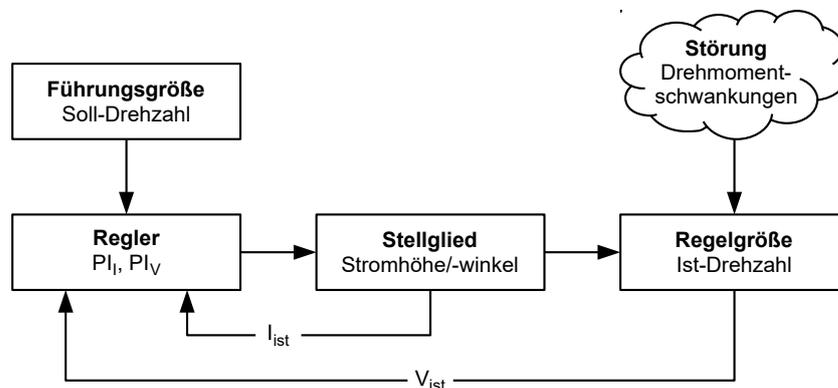
Änderung der Schrittauflösung über die Objekte **2057_h** (Clock Direction Multiplier) und **2058_h** (Clock Direction Divider). Beschleunigungs-/Verzögerungsrampen durch Anpassen der Impulsfrequenz optimieren, um den Resonanzbereich möglichst schnell zu durchlaufen.

5.1.3 Closed Loop

Einführung

Die *Closed Loop*-Theorie geht auf die Vorstellung eines Regelkreises zurück. Eine am System einwirkende Störgröße soll möglichst schnell und ohne bleibende Abweichung ausgeregelt werden, um die Regelgröße wieder an die Führungsgröße anzugleichen.

Closed Loop am Beispiel einer Drehzahlregelung:



- PI_I = Proportional-/Integralregler Stromregelkreis
- PI_V = Proportional-/Integralregler Drehzahlregelkreis
- I_{ist} = Aktueller Strom
- V_{ist} = Aktuelle Drehzahl

Das *Closed Loop*-Verfahren wird auch als "Sinuskommütierung über Encoder mit feldorientierter Regelung" bezeichnet. Kern der *Closed Loop*-Technologie ist die leistungsangepasste Stromregelung sowie die Rückführung der Istwerte des Prozesses. Über die Signale des Encoders wird die Rotorlage erfasst und es werden in den Motorwicklungen sinusförmige Phasenströme erzeugt. Durch die Vektorregelung des Magnetfelds ist gewährleistet, dass das Stator magnetfeld immer senkrecht zum

Rotormagnetfeld steht und die Feldstärke genau dem gewünschten Drehmoment entspricht. Der in den Wicklungen so gesteuerte Strom sorgt für eine gleichmäßige Motorkraft und führt zu einem besonders ruhig laufenden Motor, der sich genau regeln lässt.

Die für die Betriebsart *Closed Loop* notwendige Rückführung der Regelgrößen kann mit verschiedenen Technologien realisiert werden. Neben der physischen Rückführung mit Encoder oder Hall-Sensoren, ist auch eine virtuelle Erfassung der Motorparameter durch softwarebasierte Modellberechnung möglich. Physikalische Größen, wie Geschwindigkeit oder Gegen-EMK, können mit Hilfe eines sogenannten "Beobachters" aus den Daten des Stromreglers rekonstruiert werden. Mit dieser Sensorless-Technologie erhält man einen "virtuellen Drehgeber", der ab einer gewissen Minimalgeschwindigkeit die Positions- und Drehzahlinformation mit der gleichen Präzision liefert wie ein realer optischer oder magnetischer Drehgeber.

Alle Steuerungen von Nanotec, welche die Betriebsart *Closed Loop* unterstützen, implementieren eine feldorientierte Regelung mit einer sinuskommutierten Stromregelung. Die Schrittmotoren und BLDC-Motoren werden also genauso geregelt wie ein Servomotor. Mit der Betriebsart *Closed Loop* können Schrittwinkelfehler während der Fahrt kompensiert und Lastwinkelfehler innerhalb eines Vollschritts korrigiert werden.

Inbetriebnahme

Vor dem Anwenden der Betriebsart *Closed Loop* muss ein Auto-Setup durchgeführt werden. Der Betriebsmodus Auto-Setup ermittelt automatisch die notwendigen Parameter (z.B. Motorkenndaten, Rückführsysteme), welche für eine optimale Arbeitsweise der feldorientierten Regelung notwendig sind. Alle Informationen zur Durchführung des Auto-Setups sind im Kapitel **Auto-Setup** beschrieben.

Um die Betriebsart *Closed Loop* anzuwenden, sind je nach Motortyp und Rückführung bestimmte Einstellungen notwendig, siehe Kapitel **Motordaten einstellen**.

Das Bit 0 im **3202_h** muss gesetzt sein.

5.2 CiA 402 Power State Machine

5.2.1 Zustandsmaschine

CiA 402

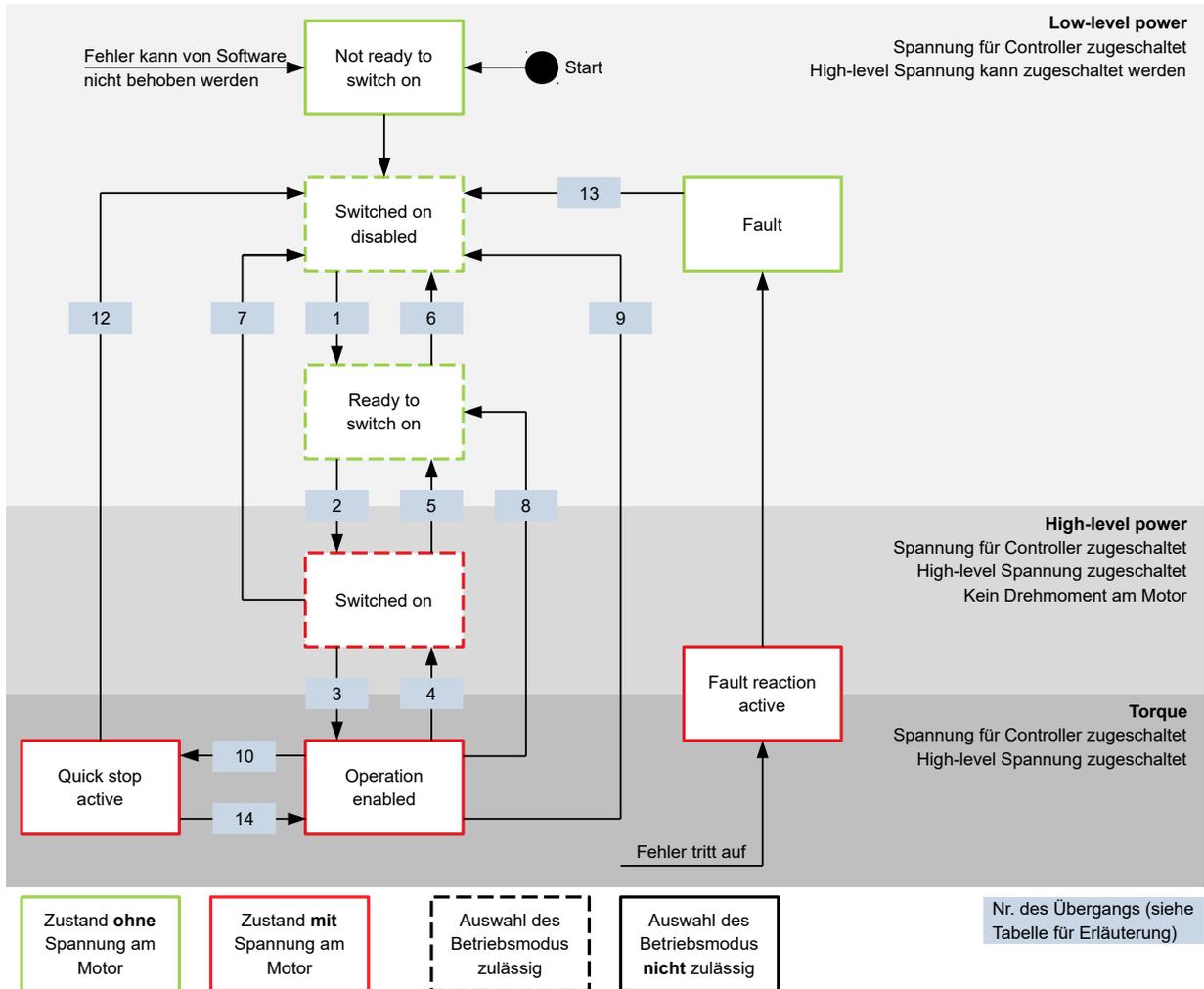
Um die Steuerung betriebsbereit zu schalten, ist es notwendig, eine Zustandsmaschine (*State Machine*) zu durchlaufen. Diese ist im *CANopen-Standard 402* definiert. Zustandsänderungen werden im Objekt **6040_h** (Controlword) angefordert. Der tatsächliche Zustand der Zustandsmaschine lässt sich aus dem Objekt **6041_h** (Statusword) entnehmen.

Controlword

Zustandsänderungen werden über Objekt **6040_h** (Controlword) angefordert.

Zustandsübergänge

Das Diagramm zeigt die möglichen Zustandsübergänge.



In der nachfolgenden Tabelle sind die Bit-Kombinationen für das Controlword aufgelistet, die zu den entsprechenden Zustandsübergängen führen. Ein X entspricht dabei einem nicht weiter zu berücksichtigenden Bit-Zustand. Einzige Ausnahme ist das Rücksetzen des Fehlers (Fault reset): Der Übergang wird nur durch steigende Flanke des Bits angefordert.

Kommando	Bit im Objekt 6040 _h					Übergang
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	1, 5, 8
Switch on	0	0	1	1	1	2
Disable voltage	0	X	X	0	X	6, 7, 9, 12
Quick stop	0	X	0	1	X	10
Disable operation	0	0	1	1	1	4
Enable operation	0	1	1	1	1	3
Enable operation after Quick stop	0	1	1	1	1	14

Kommando	Bit im Objekt 6040 _h					Übergang
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
(wenn 605A _h 5 oder 6 ist) Fault reset		X	X	X	X	13

Statusword

In der nachfolgenden Tabelle sind die Bitmasken aufgelistet, die den Zustand der Steuerung aufschlüsseln.

Statusword (6041 _h)	Zustand
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Die Steuerung erreicht nach Einschalten und erfolgreichem Selbsttest den Zustand *Switch on disabled*.



Hinweis

Tritt ein nicht behebbarer Fehler auf, wechselt die Steuerung in den Zustand *Not ready to switch on* und verbleibt dort.

Betriebsmodus

Der Betriebsmodus wird im Objekt 6060_h eingestellt. Der tatsächlich aktive Betriebsmodus wird im 6061_h angezeigt.

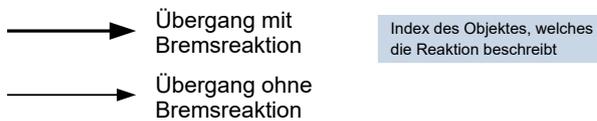
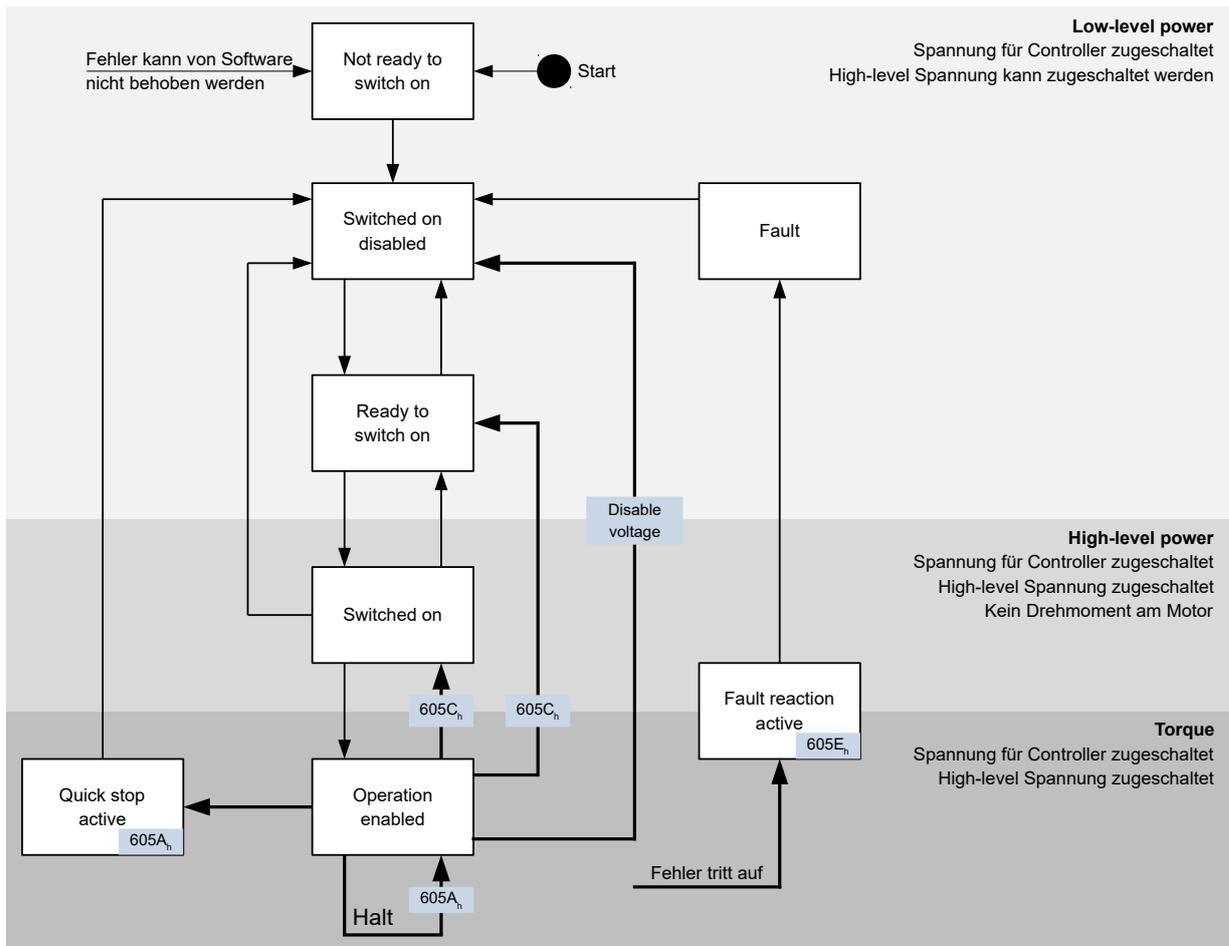
Die Einstellung oder Änderung des Betriebsmodus ist jederzeit möglich.

5.2.2 Verhalten beim Verlassen des Zustands *Operation enabled*

Bremsreaktionen

Beim Verlassen des Zustands *Operation enabled* lassen sich unterschiedliche Bremsreaktionen programmieren.

Die nachfolgende Grafik zeigt eine Übersicht der Bremsreaktionen.



Quick stop active

Übergang in den Zustand *Quick stop active* (quick stop option):

In diesem Fall wird die in Objekt **605A_h** hinterlegte Aktion ausgeführt (siehe nachfolgende Tabelle).

Wert in Objekt 605A _h	Beschreibung
0	Soforthalt mit anschließendem Zustandswechsel in <i>Switch on disabled</i>
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt

Wert in Objekt 605A _h	Beschreibung
6	bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten. Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

Ready to switch on

Übergang in den Zustand *Ready to switch on* (shutdown option):

In diesem Fall wird die in Objekt **605B_h** hinterlegte Aktion ausgeführt (siehe nachfolgende Tabelle).

Wert in Objekt 605B _h	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2 bis 32767	Reserviert

Switched on

Übergang in den Zustand *Switched on* (disable operation option):

In diesem Fall wird die in Objekt **605C_h** hinterlegte Aktion ausgeführt (siehe nachfolgende Tabelle).

Wert in Objekt 605C _h	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2 bis 32767	Reserviert

Halt

Das Bit ist gültig in folgenden Modi:

- **Profile Position**
- **Velocity**
- **Profile Velocity**
- **Profile Torque**
- **Interpolated Position Mode**

Beim Setzen des Bit 8 in Objekt **6040_h** (Controlword) wird die in **605D_h** hinterlegte Reaktion ausgeführt (siehe nachfolgende Tabelle):

Wert in Objekt 605D _h	Beschreibung
-32768 bis 0	Reserviert

Wert in Objekt 605D _h	Beschreibung
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit <i>quick stop ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
3 bis 32767	Reserviert

Fault

Fehlerfall (fault):

Sollte ein Fehler auftreten, wird der Motor abgebremst, wie es in Objekt **605E_h** hinterlegt ist.

Wert in Objekt 605E _h	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsart)
2	Abbremsen mit <i>quick stop ramp</i> (Bremsbeschleunigung je nach Betriebsart)
3 bis 32767	Reserviert

Schlepp-/Schlupffehler

Sollte ein Schlepp- oder Schlupffehler auftreten, wird der Motor abgebremst, wie es in Objekt **3700_h** hinterlegt ist.

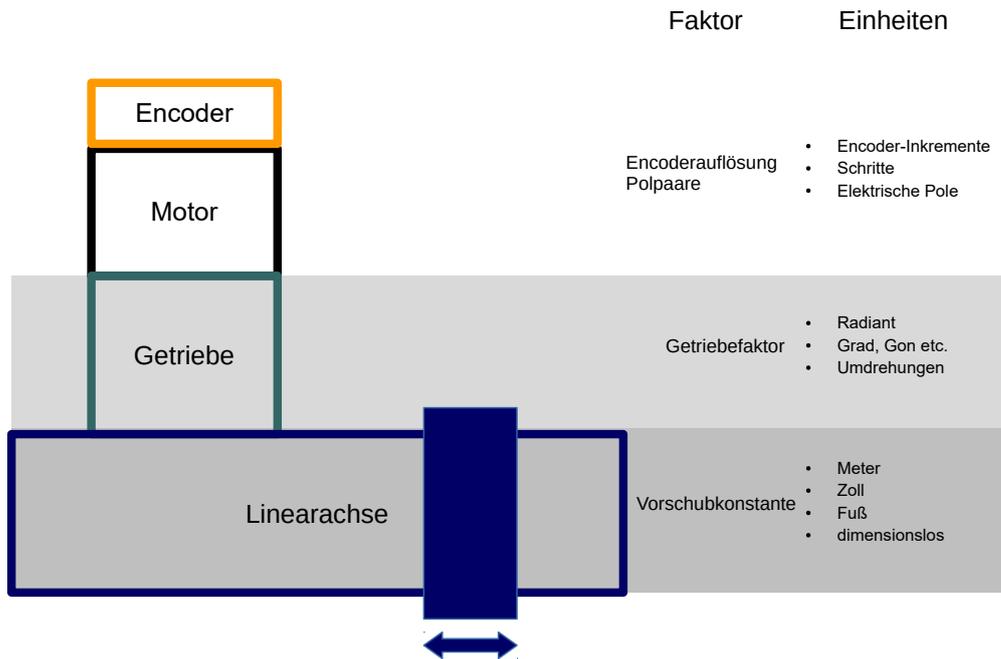
Wert	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsart)
2	Abbremsen mit <i>quick stop ramp</i> (Bremsbeschleunigung je nach Betriebsart)
3 bis 32767	Reserviert

Sie können die Fehlerüberwachung deaktivieren, indem Sie das Objekt **6065_h** auf den Wert "-1" (FFFFFFFF_h), bzw. das Objekt **60F8_h** auf den Wert "7FFFFFFF_h" setzen.

5.3 Benutzerdefinierte Einheiten

Die Steuerung bietet Ihnen die Möglichkeit, benutzerdefinierte Einheiten einzustellen. Damit lassen sich die entsprechenden Parameter z. B. direkt in Grad [°], Millimeter [mm], usw. setzen und auslesen.

Sie können auch, entsprechend den mechanischen Gegebenheiten, eine **Getriebeübersetzung** und/oder eine **Vorschubkonstante** einstellen.



Hinweis

Wertänderungen aller Objekte, die in diesem Kapitel beschrieben werden, werden im Zustand *Operation enabled* der **CiA 402 Power State Machine** nicht sofort angewendet. Der Zustand *Operation enabled* muss dazu verlassen werden.

5.3.1 Einheiten

Es werden sowohl Einheiten des internationalen Einheitensystems (*S*) als auch einige spezifische Einheiten unterstützt. Ebenfalls möglich ist die Angabe einer Zehnerpotenz als Faktor.

In der nachfolgenden Tabelle sind alle unterstützten Einheiten für die Position und deren Werte für **60A8_h** (**Positionseinheit**) bzw. **60A9_h** (**Geschwindigkeitseinheit**) aufgelistet. Abhängig von der verwendeten Einheit wird die **Vorschubkonstante (6092_h)** und/oder die **Getriebeübersetzung (6091_h)** berücksichtigt.

Name	Einheitenzeichen	Wert	6091 _h	6092 _h	Beschreibung
metre	m	01 _h	ja	ja	<i>Meter</i>
inch	in	C1 _h	ja	ja	<i>Zoll</i> (=0,0254 m)
foot	ft	C2 _h	ja	ja	<i>Fuß</i> (=0,3048 m)
grade	g	40 _h	ja	nein	<i>Gon</i> (Winkleinheit, 400 entsprechen 360°)
radian	rad	10 _h	ja	nein	<i>Radian</i>
degree	°	41 _h	ja	nein	<i>Grad</i>
arcminute	'	42 _h	ja	nein	<i>Winkelminute</i> (60'=1°)
arcsecond	"	43 _h	ja	nein	<i>Winkelsekunde</i> (60"=1')

Name	Einheitenzeichen	Wert	6091 _h	6092 _h	Beschreibung
mechanical revolution		B4 _h	ja	nein	<i>Umdrehung</i>
encoder increment		B5 _h	nein	nein	<i>Encoder-Inkremente.</i> Abhängig vom verwendeten Sensor (Encoder/ Hall-Sensor) und Betriebsart . Im <i>Open Loop</i> - und <i>Sensorless</i> -Betrieb entsprechen 32000 _h Inkremente einer Motorumdrehung.
step		AC _h	nein	nein	<i>Schritte.</i> Bei 2-phasigen Schrittmotoren entspricht die Anzahl der Polpaare (2030_h) multipliziert mit 4 einer Umdrehung. Bei 3-phasigen BLDC-Motoren entspricht die Anzahl der Polpaare (2030_h) multipliziert mit 6 einer Umdrehung.
electrical pole		C0 _h	nein	nein	<i>Elektrische Pole.</i> Bei einem Schrittmotor, der z.B. 50 Polpaare (2030_h) hat, entspricht die Einheit 1/50 einer Umdrehung.
dimensionless		00 _h	ja	ja	<i>dimensionslose Längeneinheit</i>

In der nachfolgenden Tabelle sind alle unterstützten Einheiten für die Zeit und deren Werte für **60A9_h** (**Geschwindigkeitseinheit**) aufgelistet:

Name	Einheitenzeichen	Wert	Beschreibung
second	s	03 _h	<i>Sekunde</i>
minute	min	47 _h	<i>Minute</i>
hour	h	48 _h	<i>Stunde</i>
day	d	49 _h	<i>Tag</i>
year	a	4A _h	<i>Jahr (=365,25 Tage)</i>

In der nachfolgenden Tabelle sind die möglichen Exponenten und deren Werte für **60A8_h** (**Positionseinheit**), bzw. **60A9_h** (**Geschwindigkeitseinheit**) aufgelistet:

Faktor	Exponent	Wert
10 ⁶	6	06 _h
10 ⁵	5	05 _h
...
10 ¹	1	01 _h
10 ⁰	0	00 _h
10 ⁻¹	-1	FF _h
...
10 ⁻⁵	-5	FB _h
10 ⁻⁶	-6	FA _h

5.3.2 Encoderauflösung

Die physikalische Auflösung des verwendeten Encoders/Sensors berechnet sich aus den Encoder-Inkrementen (**608F_h:1_h** (Encoder Increments)) pro Motorumdrehungen (**608F_h:2_h** (Motor Revolutions)):

$$\text{Auflösung Positionencoder} = \frac{\text{Encoder-Inkmente (608F}_h\text{:01)}}{\text{Motorumdrehungen (608F}_h\text{:02)}}$$

5.3.3 Getriebeübersetzung

Die Getriebeübersetzung berechnet sich aus Motorumdrehungen (**6091_h:1** (Motor Revolutions)) pro Achsendrehung (**6091_h:2** (Shaft Revolutions)) wie folgt:

$$\text{Getriebeübersetzung} = \frac{\text{Motorumdrehung (6091}_h\text{:1)}}{\text{Achsendrehung (6091}_h\text{:2)}}$$

5.3.4 Vorschubkonstante

Die Vorschubkonstante berechnet sich aus dem Vorschub (**6092_h:1** (Feed) pro Umdrehung der Abtriebsachse (**6092_h:2** (Shaft Revolutions)) wie folgt:

$$\text{Vorschubkonstante} = \frac{\text{Vorschub (6092}_h\text{:01)}}{\text{Umdrehung der Abtriebswelle (6092}_h\text{:02)}}$$

Die Vorschubkonstante ist zur Angabe der Spindelsteigung bei einer Linearachse nützlich und wird verwendet, wenn die Einheit auf Längenmaßen basiert oder wenn diese dimensionslos ist.

5.3.5 Berechnungsformeln für Benutzereinheiten

Positionseinheit

Das Objekt **60A8_h** enthält:

- Bits 16 bis 23: die Positionseinheit (siehe Kapitel **Einheiten**)
- Bits 24 bis 31: den Exponenten einer Zehnerpotenz (siehe Kapitel **Einheiten**)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Exponent einer Zehnerpotenz								Einheit							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserviert (00h)								reserviert (00h)							

Beispiel

Wird **60A8_h** mit dem Wert "FF410000_h" beschrieben (Bits 16-23=41_h und Bits 24-31=FF_h), wird die Einheit auf *Zehntelgrad* eingestellt (Werkseinstellung).

Bei einer relativen Zielposition (**607A_h**) von 3600 fährt der Motor genau eine mechanische Umdrehung, wenn die **Getriebeübersetzung** 1:1 ist. Die **Vorschubkonstante** spielt in diesem Fall keine Rolle.

Beispiel

Wird **60A8_h** mit dem Wert "FD010000_h" beschrieben (Bits 16-23=01_h und Bits 24-31=FD_h(=-3)), wird die Einheit auf *Millimeter* eingestellt.

Bei einer relativen Zielposition (**607A_h**) von 1 fährt der Motor genau eine mechanische Umdrehung (wenn die **Getriebeübersetzung** und **Vorschubkonstante** 1:1 sind).

Wird die **Vorschubkonstante** entsprechend der Spindelsteigung einer Linearachse eingestellt, dreht der Motor so weit, dass ein Vorschub von 1 mm erreicht wird.

Geschwindigkeitseinheit

Das Objekt **60A9_h** enthält:

- Bits 8 bis 15: die Zeiteinheit (siehe Kapitel **Einheiten**)
- Bits 16 bis 23: die Positionseinheit (siehe Kapitel **Einheiten**)
- Bits 24 bis 31: den Exponenten einer Zehnerpotenz (siehe Kapitel **Einheiten**)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Exponent einer Zehnerpotenz								Positionseinheit							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Zeiteinheit								reserviert (00h)							

Beispiel

Wird **60A9_h** mit dem Wert "00B44700_h" beschrieben (Bits 8-15=00_h, Bits 16-23=B4_h und Bits 24-31=47_h), wird die Einheit auf *Umdrehungen pro Minute* eingestellt (Werkseinstellung).

Beispiel

Wird das **60A9_h** mit dem Wert "FD010300_h" beschrieben (Bits 8-15=FD_h(=-3), Bits 16-23=01_h und Bis 24-31=03_h), wird die Einheit auf *Millimeter pro Sekunde* eingestellt.



Hinweis

Die Geschwindigkeitseinheit im Modus **Velocity** ist auf *Umdrehungen pro Minute* voreingestellt. Sie können die Einheit nur über den **604Ch VI Dimension Factor** umstellen.

Umrechnungsfaktor für die Geschwindigkeitseinheit

Sie können einen zusätzlichen Faktor für die Geschwindigkeitseinheit einstellen, damit z.B. eine Einheit von 1/3 Umdrehungen/Minute möglich ist. Der Faktor n errechnet sich aus Faktor für Zähler (**6096_h:01_h**) geteilt durch Faktor für Nenner (**6096_h:02_h**).

$$n_{\text{Geschwindigkeitseinheit}} = \frac{6096_{\text{h}}:01}{6096_{\text{h}}:02}$$

Beschleunigungseinheit

Die Beschleunigungseinheit ist **Geschwindigkeitseinheit** pro Sekunde.

Umrechnungsfaktor für die Beschleunigungseinheit

Der Faktor n für die Beschleunigungseinheit errechnet sich aus Zähler (**6097_h:01_h**) geteilt durch Nenner (**6097_h:02_h**).

$$n_{\text{Beschleunigungseinheit}} = \frac{6097_{\text{h}}:01}{6097_{\text{h}}:02}$$

Ruckeinheit

Die Ruckeinheit ist **Beschleunigungseinheit** pro Sekunde.

Umrechnungsfaktor für den Ruck

Der Faktor n für den Ruck errechnet sich aus Zähler (**60A2_h:01_h**) geteilt durch Nenner (**60A2_h:02_h**).

$$n_{\text{Ruckeinheit}} = \frac{60A2_{\text{h}}:01}{60A2_{\text{h}}:02}$$

5.4 Begrenzung des Bewegungsbereichs

Die digitalen Eingänge können als Endschalter verwendet werden, im Kapitel **Digitale Eingänge** wird beschrieben, wie Sie diese Funktion der Eingänge aktivieren. Die Steuerung unterstützt auch Software-Endschalter.

5.4.1 Verhalten beim Erreichen der Endschalter

Wird ein Endschalter überfahren, wird das Bit 7 (*Warning*) in **6041_h** (*Statusword*) gesetzt und die in Objekt **3701_h** hinterlegte Aktion ausgeführt (siehe nachfolgende Tabelle).

Wert in Objekt 3701 _h	Beschreibung
-1 (Werkseinstellung)	keine Reaktion (um z. B. eine Referenzfahrt durchzuführen)
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.
6	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

Solange der Endschalter noch aktiv ist, ist das Fahren in die Richtung des Endschalters blockiert, es kann aber in die gegengesetzte Richtung gefahren werden.

Das Bit 7 (*Warning*) in **6041_h** wird erst gelöscht, wenn der Endschalter deaktiviert ist und über die Endschalter-Position zurückgefahren wurde.

5.4.2 Software-Endschalter

Die Steuerung berücksichtigt Software-Endschalter (**607D_n** (Software Position Limit)). Zielpositionen (**607A_n**) werden durch **607D_n** limitiert, die absolute Zielposition darf nicht größer sein als die Grenzen in **607D_n**. Sollte sich der Motor beim Einrichten der Endschalter außerhalb des zulässigen Bereichs befinden, werden nur Fahrbefehle in Richtung des zulässigen Bereichs angenommen.

5.5 Zykluszeiten

Die Steuerung arbeitet mit einer Zykluszeit von 1 ms. Das bedeutet, dass Daten jeweils alle 1 ms verarbeitet werden, mehrfache Änderungen eines Wertes (z.B. Wert eines Objektes oder Pegel an einem digitalen Eingang) innerhalb einer ms können nicht erfasst werden.

In der nachfolgenden Tabelle finden Sie eine Übersicht der Zykluszeiten der verschiedenen Prozesse.

Task	Zykluszeit
Applikation	1 ms
NanoJ Applikation	1 ms
Stromregler	31,25 µs (32 KHz)
Geschwindigkeitsregler	250 µs (4 KHz)
Positionsregler	1 ms

6 Betriebsmodi

6.1 Profile Position

6.1.1 Übersicht

Beschreibung

Der *Profile Position Mode* dient dazu, Positionen relativ zur letzten Zielposition oder absolut zur letzten Referenzposition anzufahren. Während der Bewegung werden Grenzwerte für die Geschwindigkeit, Anfahr- und Bremsbeschleunigung und Rucke berücksichtigt.

Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "1" gesetzt werden (siehe "**CiA 402 Power State Machine**").

Controlword

Folgende Bits im Objekt **6040_h** (Controlword) haben eine gesonderte Funktion:

- Bit 4 startet einen Fahrauftrag. Dieser wird bei einem Übergang von "0" nach "1" übernommen. Eine Ausnahme besteht, wenn es von einem anderen Betriebsmodus nach *Profile Position* gewechselt wird: Ist das Bit 4 bereits gesetzt, muss es nicht auf "0" und wieder auf "1" gesetzt werden, damit der Fahrauftrag gestartet wird.
- Bit 5: Ist dieses Bit auf "1" gesetzt, wird ein durch Bit 4 ausgelöster Fahrauftrag sofort ausgeführt. Ist es auf "0" gesetzt, wird der gerade ausgeführte Fahrauftrag zu Ende gefahren und erst im Anschluss der nächste Fahrauftrag gestartet.
- Bit 6: Bei "0" ist die Zielposition (**607A_h**) absolut und bei "1" ist die Zielposition relativ. Die Referenzposition ist abhängig von den Bits 0 und 1 des Objekts **60F2_h**.
- Bit 8 (Halt): Ist dieses Bit auf "1" gesetzt, bleibt der Motor stehen. Bei einem Übergang von "1" auf "0" beschleunigt der Motor mit der eingestellten Startrampe bis zur Zielgeschwindigkeit. Bei einem Übergang von "0" auf "1" bremst der Motor ab und bleibt stehen. Die Bremsbeschleunigung ist dabei abhängig von der Einstellung des "Halt Option Code" im Objekt **605D_h**.
- Bit 9 (Change on setpoint): Ist dieses Bit gesetzt, wird die Geschwindigkeit erst beim Erreichen der ersten Zielposition geändert. Das bedeutet, dass vor Erreichen des ersten Ziels keine Bremsung durchgeführt wird, da der Motor auf dieser Position nicht stehen bleiben soll.

Controlword 6040 _h		
Bit 9	Bit 5	Definition
X	1	Die neue Zielposition wird sofort angefahren.
0	0	Das Positionieren wird erst vollständig abgeschlossen, bevor die nächste Zielposition mit den neuen Limitierungen angefahren wird.
1	0	Die momentane Zielposition wird nur durchfahren, danach wird die neue Zielposition mit den neuen Werten angefahren.

Siehe dazu das Bild in "**Setzen von Fahrbefehlen**".



Hinweis

Das Bit 9 im Controlword wird ignoriert, wenn die Rampengeschwindigkeit im Zielpunkt unterschritten wird. In diesem Fall müsste die Steuerung zurücksetzen und Anlauf nehmen, um die Vorgabe zu erreichen.

Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

- Bit 10 (Target Reached): Dieses Bit ist auf "1" gesetzt, wenn das letzte Ziel erreicht wurde und der Motor eine vorgegebene Zeit (**6068_h**) innerhalb eines Toleranzfensters (**6067_h**) steht.
- Bit 11: Limit überschritten: Die Sollposition über- oder unterschreitet die in **607D_h** eingegebenen Grenzwerte.
- Bit 12 (Set-point acknowledge): Dieses Bit bestätigt den Erhalt eines neuen und gültigen Zielpunktes. Es wird synchron zu dem Bit "New set-point" im Controlword gesetzt und zurückgesetzt.

Eine Ausnahme besteht, wenn eine neue Fahrt gestartet wird, während eine andere noch nicht abgeschlossen ist, und die nächste Fahrt erst nach dem Abschluss der ersten Fahrt ausgeführt werden soll. In diesem Fall wird das Bit erst zurückgesetzt, wenn der Befehl angenommen wurde und die Steuerung bereit ist, neue Fahrbefehle auszuführen. Wird ein neuer Fahrauftrag gesendet, obwohl dieses Bit noch gesetzt ist, wird der neueste Fahrauftrag ignoriert.

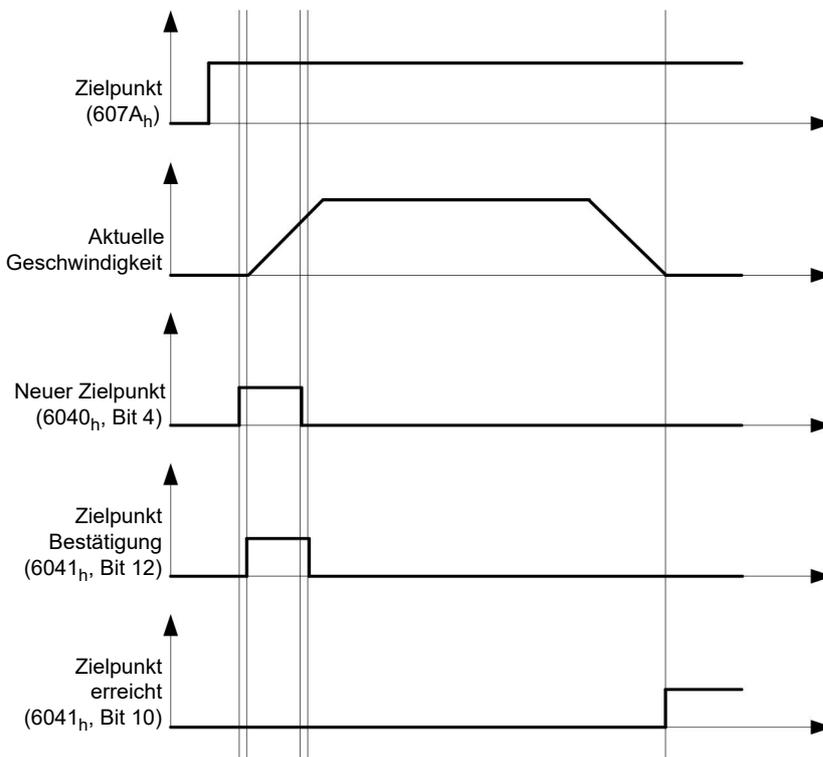
Das Bit wird nicht gesetzt, wenn eine der folgenden Bedingungen erfüllt ist:

- Die neue Zielposition kann unter Einhaltung aller Randbedingungen nicht mehr erreicht werden.
- Es wird bereits eine Zielposition angefahren und zudem ist bereits eine Zielposition vorgegeben. Eine neue Zielposition lässt sich erst vorgeben, nachdem die aktuelle Positionierung abgeschlossen ist.
- Bit 13 (Following Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schleppfehler größer als die eingestellten Grenzen ist (**6065_h** (Following Error Window) und **6066_h** (Following Error Time Out)).

6.1.2 Setzen von Fahrbefehlen

Fahrbefehl

In Objekt **607A_h** (Target Position) wird die neue Zielposition in Benutzereinheiten angegeben (siehe **Benutzerdefinierte Einheiten**). Anschließend wird mit dem Setzen von Bit 4 im Objekt **6040_h** (Controlword) der Fahrbefehl ausgelöst. Wenn die Zielposition gültig ist, antwortet die Steuerung mit Bit 12 im Objekt **6041_h** (Statusword) und beginnt die Positionierfahrt. Sobald die Position erreicht ist, wird im Statusword das Bit 10 auf "1" gesetzt.



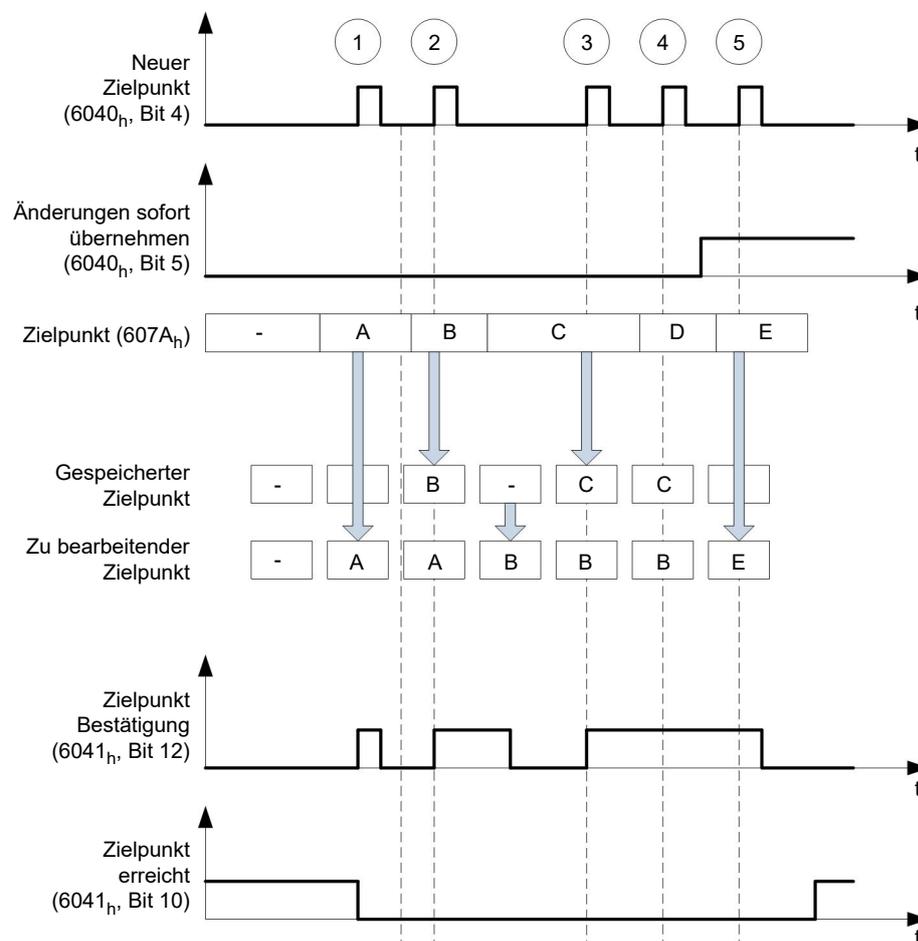
Die Steuerung kann das Bit 4 im Objekt **6040_h** (Controlword) auch selbstständig zurücksetzen. Das wird mit den Bits 4 und 5 des Objektes **60F2_h** eingestellt.

Weitere Fahrbefehle

Bit 12 im Objekt **6041_h** (Statusword, Set-point acknowledge) fällt auf "0", falls ein weiterer Fahrbefehl zwischengespeichert werden kann (siehe Zeitpunkt 1 im nachfolgenden Bild). Solange eine Zielposition angefahren wird, lässt sich eine zweite Zielposition vorbereitend an die Steuerung übergeben. Dabei können alle Parameter - wie Geschwindigkeit, Beschleunigung, Bremsbeschleunigung usw. - neu gesetzt werden (Zeitpunkt 2). Ist der Zwischenspeicher wieder leer, lässt sich der nächste Zeitpunkt einreihen (Zeitpunkt 3).

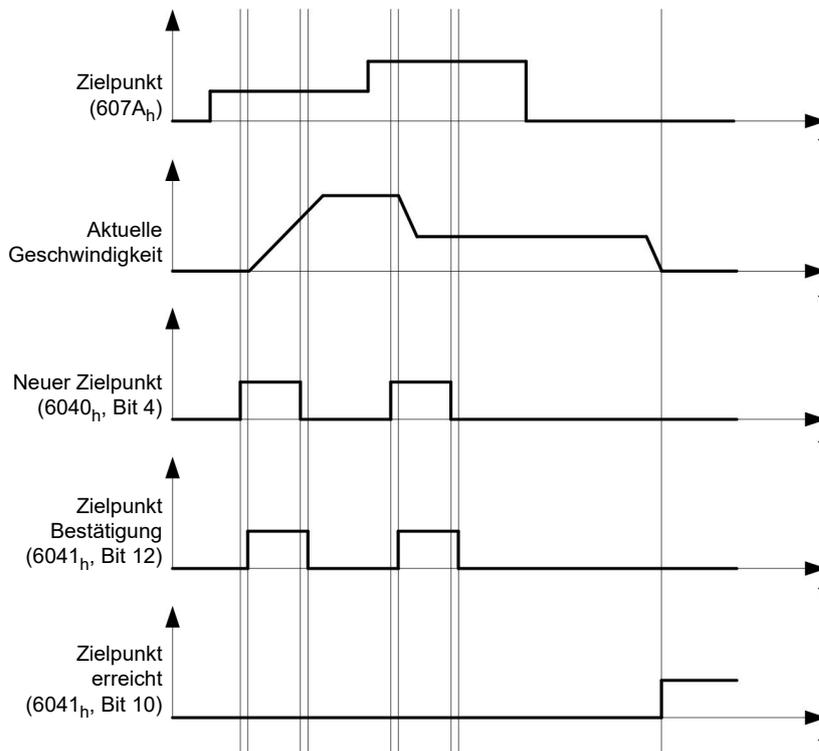
Sollte der Zwischenspeicher schon voll sein, wird ein neuer Zielpunkt ignoriert (Zeitpunkt 4). Wird Bit 5 im Objekt **6040_h** (Controlword, Bit: "Change Set-Point Immediately") gesetzt, arbeitet die Steuerung ohne den Zwischenspeicher, neue Fahrbefehle werden direkt umgesetzt (Zeitpunkt 5).

Zeitpunkte



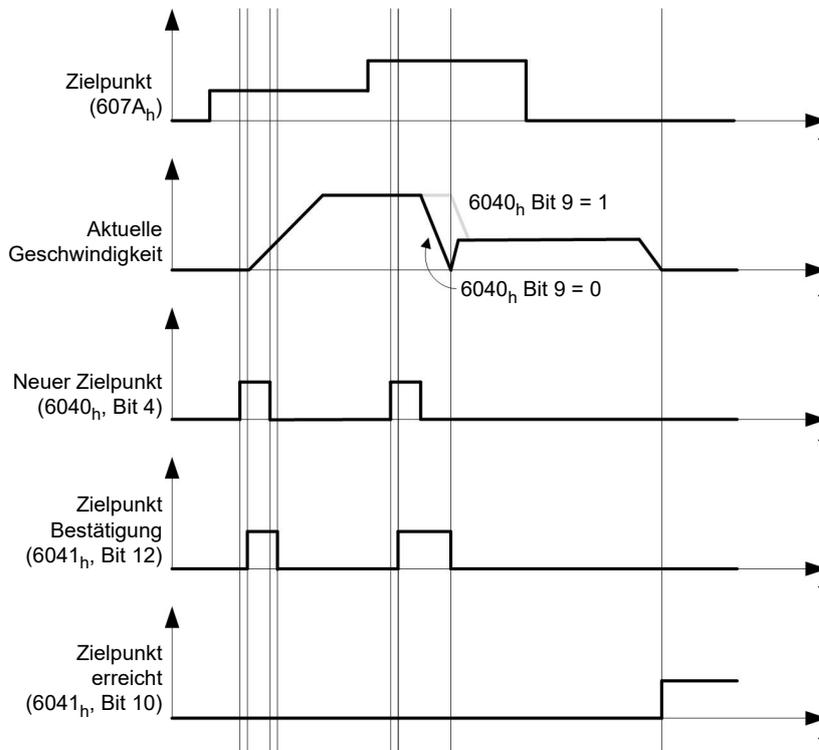
Übergangsprozedur für zweite Zielposition

Die folgende Grafik zeigt die Übergangsprozedur für die zweite Zielposition, während die erste Zielposition angefahren wird. In dieser Abbildung ist Bit 5 von Objekt **6040_h** (Controlword) auf "1" gesetzt, der neue Zielwert wird demnach sofort übernommen.



Möglichkeiten zum Anfahren einer Zielposition

Ist Bit 9 in Objekt **6040_h** (Controlword) gleich "0", wird die momentane Zielposition erst vollständig angefahren. In diesem Beispiel ist die Endgeschwindigkeit (**6082_h**) der ersten Zielposition gleich Null. Wird Bit 9 auf "1" gesetzt, wird die Profilgeschwindigkeit (**6081_h**) gehalten, bis die Zielposition erreicht wurde; erst ab dann gelten die neuen Randbedingungen.



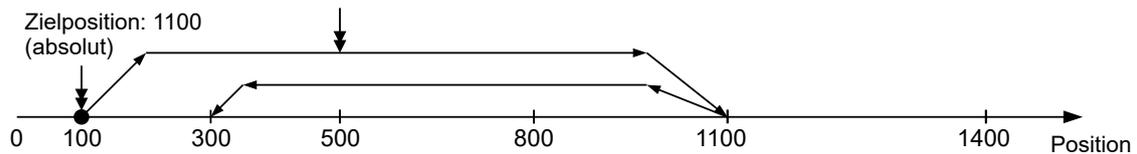
Mögliche Kombinationen von Fahrbefehlen

Um eine bessere Übersicht für die Fahrbefehle zu bekommen, werden in diesem Kapitel Kombinationen von Fahrbefehlen aufgelistet und dargestellt.

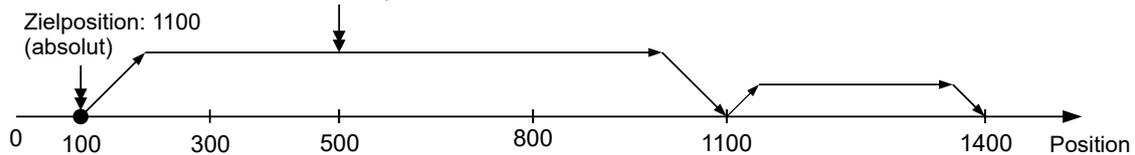
Die nachfolgenden Bilder setzen voraus:

- Ein Doppelpfeil markiert einen neuen Fahrbefehl.
- Der erste Fahrbefehl am Start ist immer ein absoluter Fahrbefehl auf die Position 1100.
- Die zweite Bewegung wird mit einer niedrigeren Geschwindigkeit durchgeführt, um einen übersichtlicher dargestellten Graphen zu erhalten.

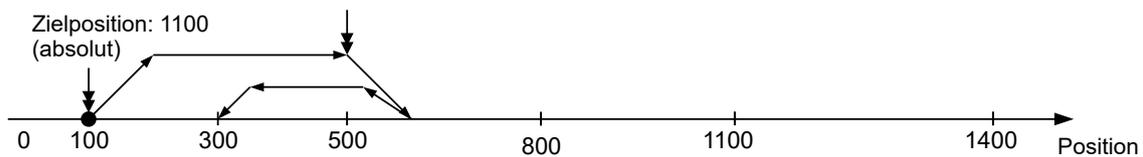
- Änderung im Zielpunkt übernehmen (6040_h:00 Bit 5 = 0)
- Positionierung absolut (6040_h:00 Bit 6 = 0)
- Zielposition: 300



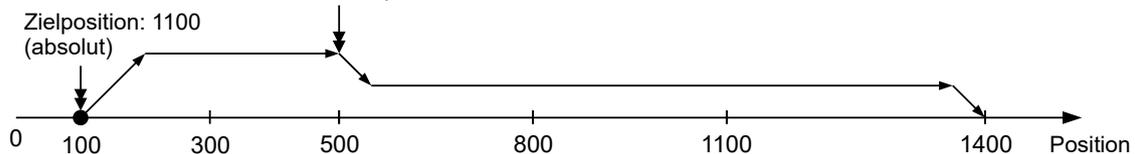
- Relativ zu der vorhergehenden Zielposition (60F2_h:00 = 0)
- Änderung im Zielpunkt übernehmen (6040_h:00 Bit 5 = 0)
- Positionierung relativ (6040_h:00 Bit 6 = 1)
- Zielposition: 300



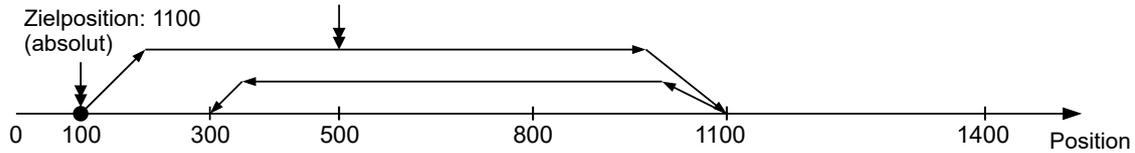
- Änderung sofort übernehmen (6040_h:00 Bit 5 = 1)
- Positionierung absolut (6040_h:00 Bit 6 = 0)
- Zielposition: 300



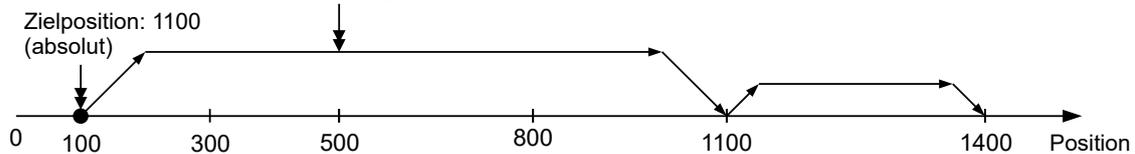
- Relativ zu der vorhergehenden Zielposition (60F2_h:00 = 0)
- Änderung sofort übernehmen (6040_h:00 Bit 5 = 1)
- Positionierung relativ (6040_h:00 Bit 6 = 1)
- Zielposition: 300



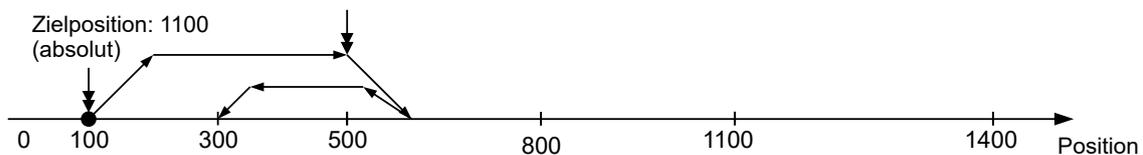
- Änderung im Zielpunkt übernehmen (6040_h:00 Bit 5 = 0)
- Positionierung absolut (6040_h:00 Bit 6 = 0)
- Zielposition: 300



- Relativ zu der aktuellen Position (60F2_h:00 = 1)
- Änderung im Zielpunkt übernehmen (6040_h:00 Bit 5 = 0)
- Positionierung relativ (6040_h:00 Bit 6 = 1)
- Zielposition: 300

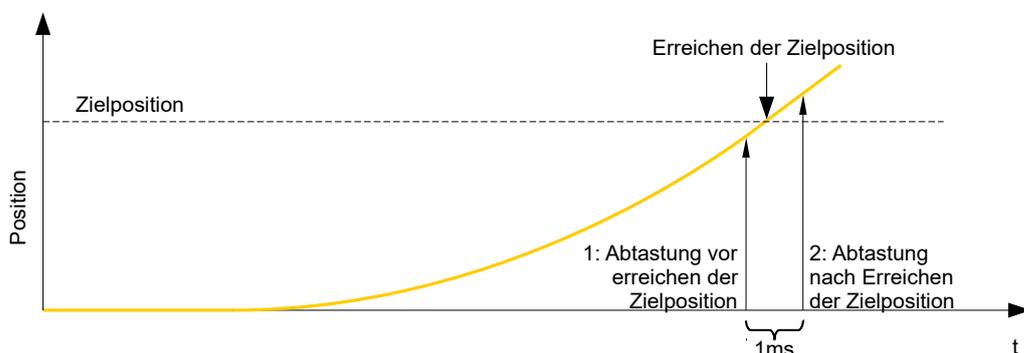


- Änderung sofort übernehmen (6040_h:00 Bit 5 = 1)
- Positionierung absolut (6040_h:00 Bit 6 = 0)
- Zielposition: 300



6.1.3 Genauigkeitsverlust bei Relativbewegungen

Beim Verketteten von relativen Bewegungen kann es zu einem Verlust an Genauigkeit kommen, sollte die Endgeschwindigkeit nicht auf Null gesetzt sein. Die folgende Grafik zeigt, aus welchem Grund.



Die aktuelle Position wird einmal pro Millisekunde abgetastet. Es kann passieren, dass die Zielposition zwischen zwei Abtastungen erreicht wird. Im Falle einer Endgeschwindigkeit ungleich Null wird die Abtastung nach Erreichen der Zielposition als Grundlage für die nachfolgende Bewegung als Offset herangezogen. Demzufolge kann die nachfolgende Bewegung etwas weiter gehen, als erwartet.

6.1.4 Randbedingungen für eine Positionierfahrt

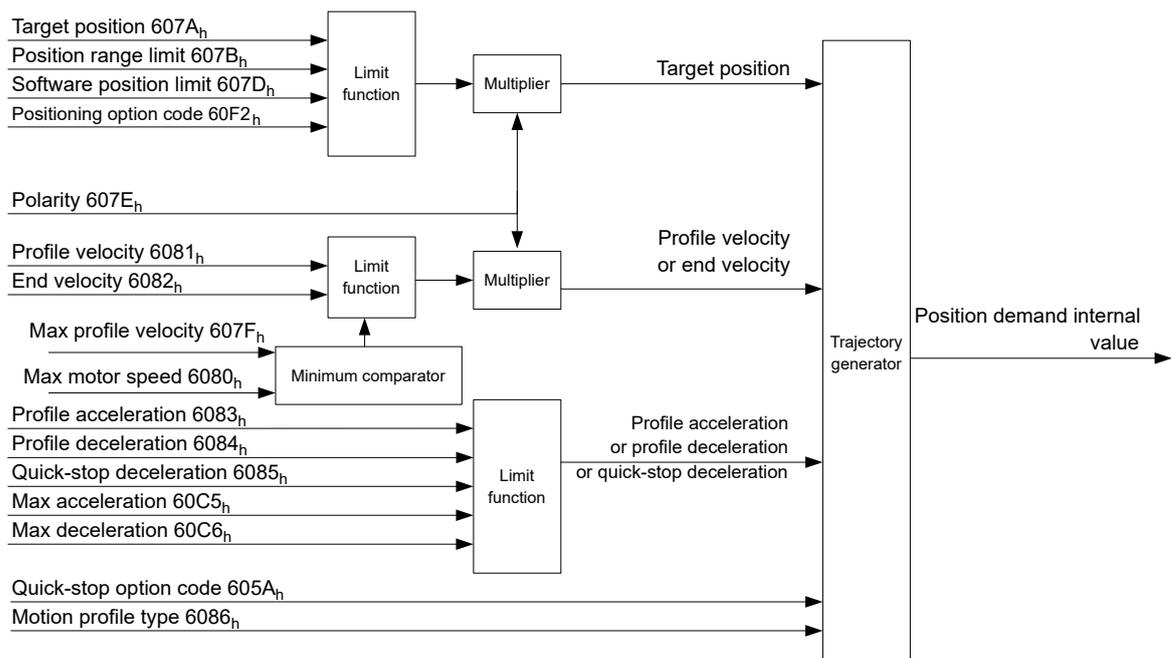
Objekteinträge

Die Randbedingungen für die gefahrene Position lassen sich in folgenden Einträgen des Objektverzeichnisses einstellen:

- **607A_h** (Target Position): vorgesehene Zielposition
- **607D_h** (Software Position Limit): Definition der Endanschläge (siehe Kapitel **Software-Endschalter**)
- **607C_h** (Home Offset): Gibt die Differenz zwischen Null-Position der Steuerung und dem Referenzpunkt der Maschine in **benutzerdefinierten Einheiten** an. (siehe "**Homing**")
- **607B_h** (Position Range Limit): Grenzen einer Modulo-Operation zur Nachbildung einer endlosen Rotationsachse
- **607E_h** (Polarity): Drehrichtung
- **6081_h** (Profile Velocity): maximale Geschwindigkeit, mit der die Position angefahren werden soll
- **6082_h** (End Velocity): Geschwindigkeit beim Erreichen der Zielposition
- **6083_h** (Profile Acceleration): gewünschte Anfahrbeschleunigung
- **6084_h** (Profile deceleration): gewünschte Bremsbeschleunigung
- **6085_h** (Quick Stop Deceleration): Nothalt-Bremsbeschleunigung im Falle des Zustandes "Quick stop active" der "CiA 402 Power State machine"
- **6086_h** (Motion Profile Type): Typ der zu fahrenden Rampe; ist der Wert "0", wird der Ruck nicht limitiert, ist der Wert "3", werden die Werte von 60A4_h:1_h- 4_h als Limitierungen des Rucks gesetzt.
- **60C5_h** (Max Acceleration): die maximale Beschleunigung, die beim Anfahren der Endposition nicht überschritten werden darf
- **60C6_h** (Max Deceleration): die maximale Bremsbeschleunigung, die beim Anfahren der Endposition nicht überschritten werden darf
- **60A4_h** (Profile Jerk), Subindex 01_h bis 04_h: Objekte zur Beschreibung der Grenzwerte für den Ruck.
- Die Geschwindigkeit wird durch **607F_h** (Max Profile Velocity) und **6080_h** (Max Motor Speed) begrenzt, der kleinere Wert wird als Grenze herangezogen.
- **60F2_h** (Positioning Option Code): definiert das Positionierverhalten

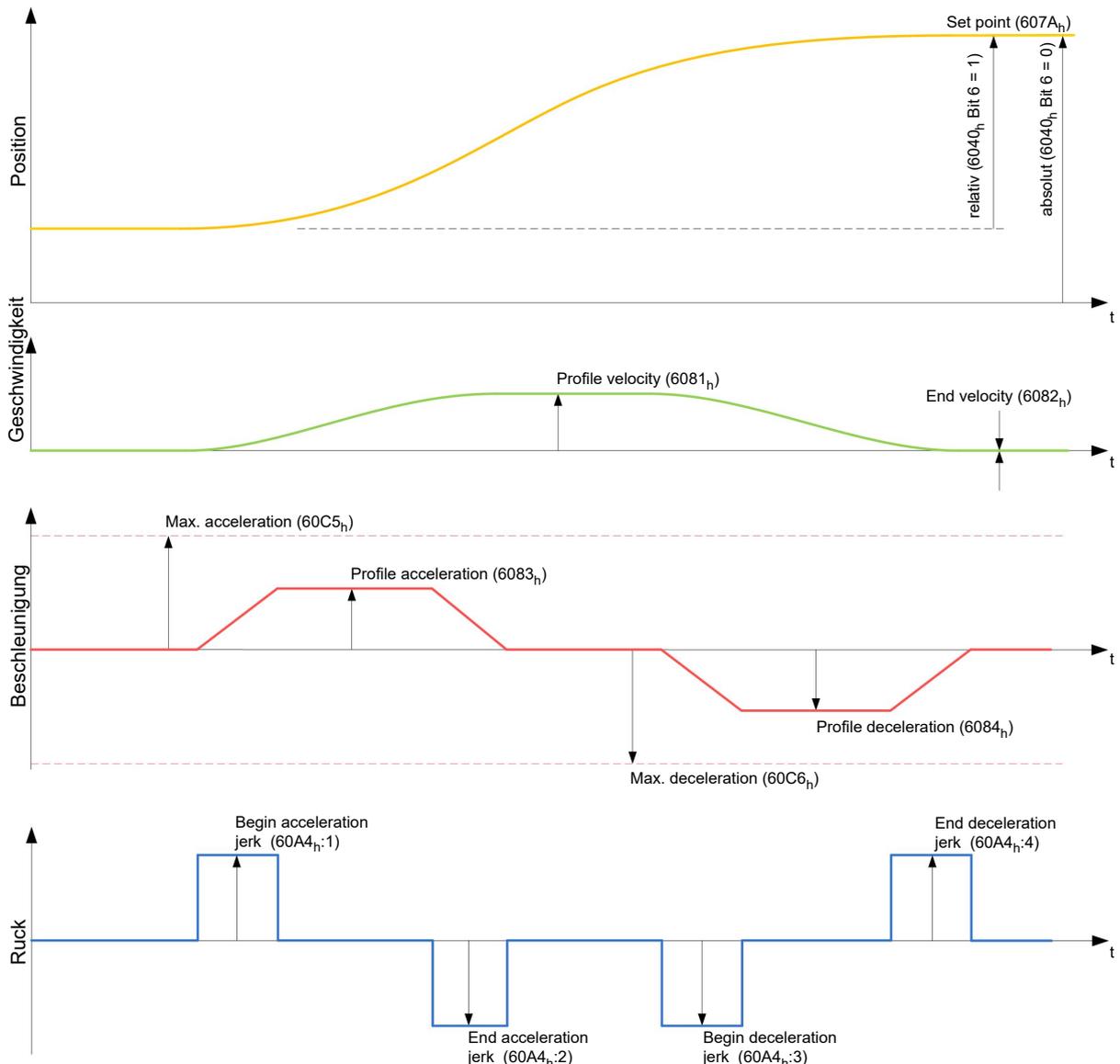
Objekte für die Positionierfahrt

Die nachfolgende Grafik zeigt die beteiligten Objekte für die Randbedingungen der Positionierfahrt.



Parameter für die Zielposition

Nachfolgende Grafik zeigt eine Übersicht über die Parameter, die für das Anfahren einer Zielposition angewendet werden (Abbildung nicht maßstabsgerecht).



6.1.5 Ruck-begrenzer und nicht ruck-begrenzter Modus

Beschreibung

Es wird grundsätzlich zwischen den Modi "ruck-begrenzt" und "nicht ruck-begrenzt" unterschieden.

Ruck-begrenzter Modus

Eine ruck-begrenzte Positionierung lässt sich erreichen, indem das Objekt **6086_h** auf "3" gesetzt wird. Damit werden die Einträge für die Rucke im Subindex :1_h - 4_h vom Objekt **60A4** gültig.

Nicht ruck-begrenzter Modus

Eine "nicht ruck-begrenzte" Rampe wird gefahren, wenn der Eintrag im Objekt **6086_h** auf "0" gesetzt wird (Standard-Einstellung).

6.2 Velocity

6.2.1 Beschreibung

Dieser Modus betreibt den Motor unter Vorgabe einer Zielgeschwindigkeit ähnlich einem Frequenzumrichter. Im Gegensatz zum *Profile Velocity Mode* erlaubt dieser Modus nicht, ruckbegrenzte Rampen auszuwählen.

6.2.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "2" gesetzt werden (siehe **CiA 402 Power State Machine**).

6.2.3 Controlword

Folgende Bits im Objekt **6040_h** (Controlword) haben eine gesonderte Funktion:

- Bit 8 (Halt): Ist dieses Bit auf "1" gesetzt bleibt der Motor stehen. Bei einem Übergang von "1" auf "0" beschleunigt der Motor mit der eingestellten Beschleunigungsrampe bis zur Zielgeschwindigkeit. Bei einem Übergang von "0" auf "1" bremst der Motor entsprechend der Bremsrampe ab und bleibt stehen.

6.2.4 Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

- Bit 11: Limit überschritten: Die Zielgeschwindigkeit über- oder unterschreitet die eingegebenen Grenzwerte.

6.2.5 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

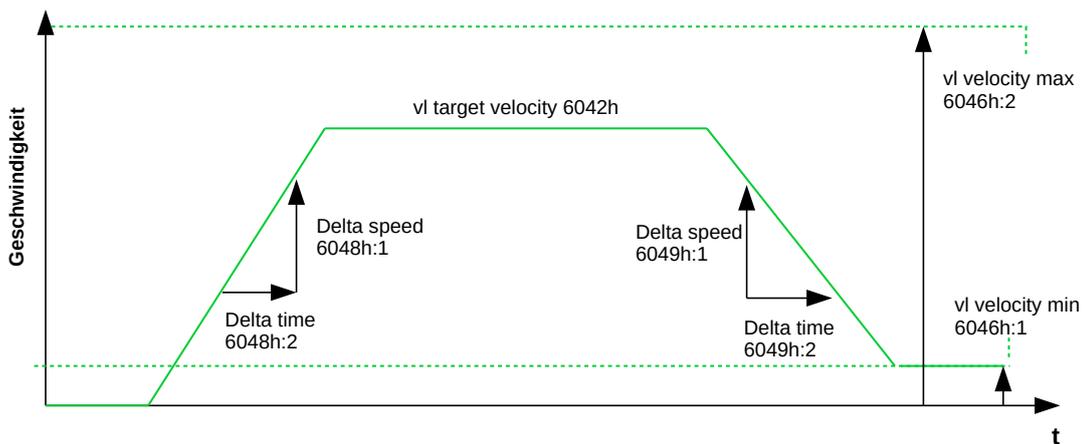
- **604C_h** (Dimension Factor):
Hier wird die Einheit der Geschwindigkeitsangaben für die nachfolgenden Objekte festgelegt. Der Subindex 1 enthält den Nenner (Multiplikator) und der Subindex 2 den Zähler (Divisor), mit dem interne Geschwindigkeitsangaben in Umdrehungen pro Minute verrechnet werden. Wird z.B. Subindex 1 auf den Wert "60" und Subindex 2 auf den Wert "1" eingestellt, erfolgt die Geschwindigkeitsangabe in Umdrehungen pro Sekunde (60 Umdrehungen pro 1 Minute).
- **6042_h**: Target Velocity.
Hier wird die Zielgeschwindigkeit in benutzerdefinierten Einheiten eingestellt.
- **6048_h**: Velocity Acceleration
Dieses Objekt definiert die Beschleunigung. Der Subindex 1 enthält dabei die Geschwindigkeitsänderung, der Subindex 2 die zugehörige Zeit in Sekunden. Beides zusammen wird als Beschleunigung verrechnet:
$$\text{VL velocity acceleration} = \frac{\text{Delta speed (6048}_{h}:1)}{\text{Delta time (6048}_{h}:2)}$$
- **6049_h** (Velocity Deceleration):
Dieses Objekt definiert die Verzögerung (Bremsrampe). Die Subindizes sind dabei so aufgebaut, wie im Objekt **6048_h** beschrieben, die Geschwindigkeitsänderung ist mit positiven Vorzeichen anzugeben.
- **6046_h** (Velocity Min Max Amount):
In diesem Objekt werden die Limitierungen der Zielgeschwindigkeiten angegeben.
In **6046_h:1_h** wird die minimale Geschwindigkeit eingestellt. Unterschreitet die Zielgeschwindigkeit (**6042_h**) die Minimalgeschwindigkeit, wird der Wert auf die Minimalgeschwindigkeit **6046_h:1_h** begrenzt.
In **6046_h:2_h** wird die maximale Geschwindigkeit eingestellt. Überschreitet die Zielgeschwindigkeit (**6042_h**) die Maximalgeschwindigkeit, wird der Wert auf die Maximalgeschwindigkeit **6046_h:2_h** begrenzt.

- **604A_h** (Velocity Quick Stop):
Mit diesem Objekt kann die Schnellstop-Rampe eingestellt werden. Die Subindizes 1 und 2 sind dabei identisch wie bei Objekt **6048_h** beschrieben.
- **6080_h** (Max Motor Speed): maximale Geschwindigkeit

Folgende Objekte können zur Kontrolle der Funktion genutzt werden:

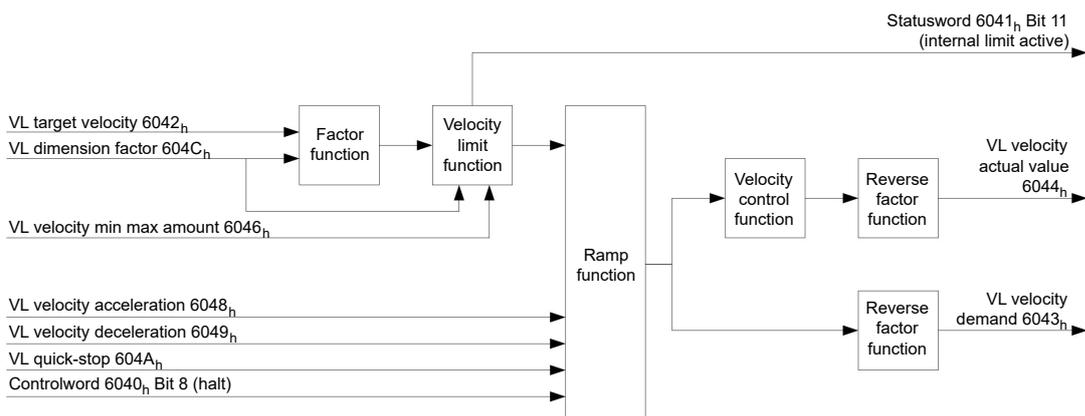
- **6043_h** (VI Velocity Demand)
- **6044_h** (VI Velocity Actual Value)

Geschwindigkeiten im Velocity Mode



Objekte für den Velocity Mode

Der Rampengenerator folgt der Zielgeschwindigkeit unter Einhaltung der eingestellten Geschwindigkeits- und Beschleunigungsgrenzen. Solange eine Begrenzung aktiv ist, wird das Bit 11 im Objekt **6041_h** gesetzt (internal limit active).



6.3 Profile Velocity

6.3.1 Beschreibung

Dieser Modus betreibt den Motor im Geschwindigkeitsmodus mit erweiterten (ruck-limitierten) Rampen. Im Gegensatz zum *Velocity Mode* (siehe "**Velocity**") wird bei diesem Modus im **Statusword** angezeigt, ob die Zielgeschwindigkeit erreicht ist.

6.3.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "3" gesetzt werden (siehe "**CiA 402 Power State Machine**").

6.3.3 Controlword

Folgende Bits im Objekt **6040_h** (Controlword) haben eine gesonderte Funktion:

- Bit 8 (Halt): Ist dieses Bit auf "1" gesetzt, bleibt der Motor stehen. Bei einem Übergang von "1" auf "0" beschleunigt der Motor mit der eingestellten Startrampe bis zur Zielgeschwindigkeit. Bei einem Übergang von "0" auf "1" bremst der Motor ab und bleibt stehen.

6.3.4 Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

- Bit 10 (Zielgeschwindigkeit erreicht; Target Reached): Dieses Bit gibt in Kombination mit dem Bit 8 im Controlword an, ob die Zielgeschwindigkeit erreicht ist, gebremst wird oder der Motor steht (siehe Tabelle).

6041_h Bit 10	6040_h Bit 8	Beschreibung
0	0	Zielgeschwindigkeit nicht erreicht
0	1	Achse bremst
1	0	Zielgeschwindigkeit innerhalb Zielfenster (definiert in 606D_h und 606E_h)
1	1	Geschwindigkeit der Achse ist 0

- Bit 13 (Deviation Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schlupffehler größer als die eingestellten Grenzen ist (**60F8_h Max Slippage** und **203F_h Max Slippage Time Out**).

6.3.5 Objekteinträge

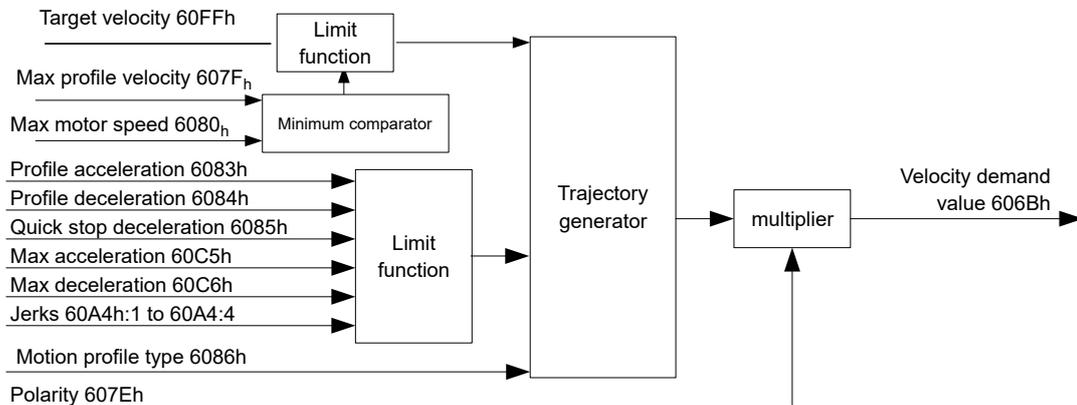
Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **606B_h** (Velocity Demand Value):
Dieses Objekt enthält die Ausgabe des Rampengenerators, die gleichzeitig der Vorgabewert für den Geschwindigkeitsregler ist.
- **606C_h** (Velocity Actual Value):
Gibt die aktuelle Istgeschwindigkeit an.
- **606D_h** (Velocity Window):
Dieser Wert gibt an, wie stark die tatsächliche Geschwindigkeit von der Sollgeschwindigkeit abweichen darf, damit das Bit 10 (Zielgeschwindigkeit erreicht; Target Reached) im Objekt **6041_h** (Statusword) auf "1" gesetzt ist.
- **606E_h** (Velocity Window Time):
Dieses Objekt gibt an, wie lange die reale Geschwindigkeit und die Sollgeschwindigkeit nahe beieinander liegen müssen (siehe **606D_h** "Velocity Window"), damit Bit 10 "Zielgeschwindigkeit erreicht" im Objekt **6041_h** (Statusword) auf "1" gesetzt wird.
- **607E_h** (Polarity):
Wird hier Bit 6 auf "1" gestellt, wird das Vorzeichen der Zielgeschwindigkeit umgekehrt.
- **6083_h** (Profile acceleration):
Setzt den Wert für die Beschleunigungsrampe im Velocity Mode.
- **6084_h** (Profile Deceleration):
Setzt den Wert für die Bremsrampe im Velocity-Mode.
- **6085_h** (Quick Stop Deceleration):
Setzt den Wert für die Bremsrampe für die Schnellbremsung im Velocity Mode.
- **6086_h** (Motion Profile Type):

Hier kann der Rampentyp ausgewählt werden ("0" = Trapez-Rampe, "3" = ruck-begrenzte Rampe).

- **60FF_h** (Target Velocity):
Gibt die zu erreichende Zielgeschwindigkeit an.
- Die Geschwindigkeit wird durch **607F_h** (Max Profile Velocity) und **6080_h** (Max Motor Speed) begrenzt, der kleinere Wert wird als Grenze herangezogen.

Objekte im Profile Velocity Mode

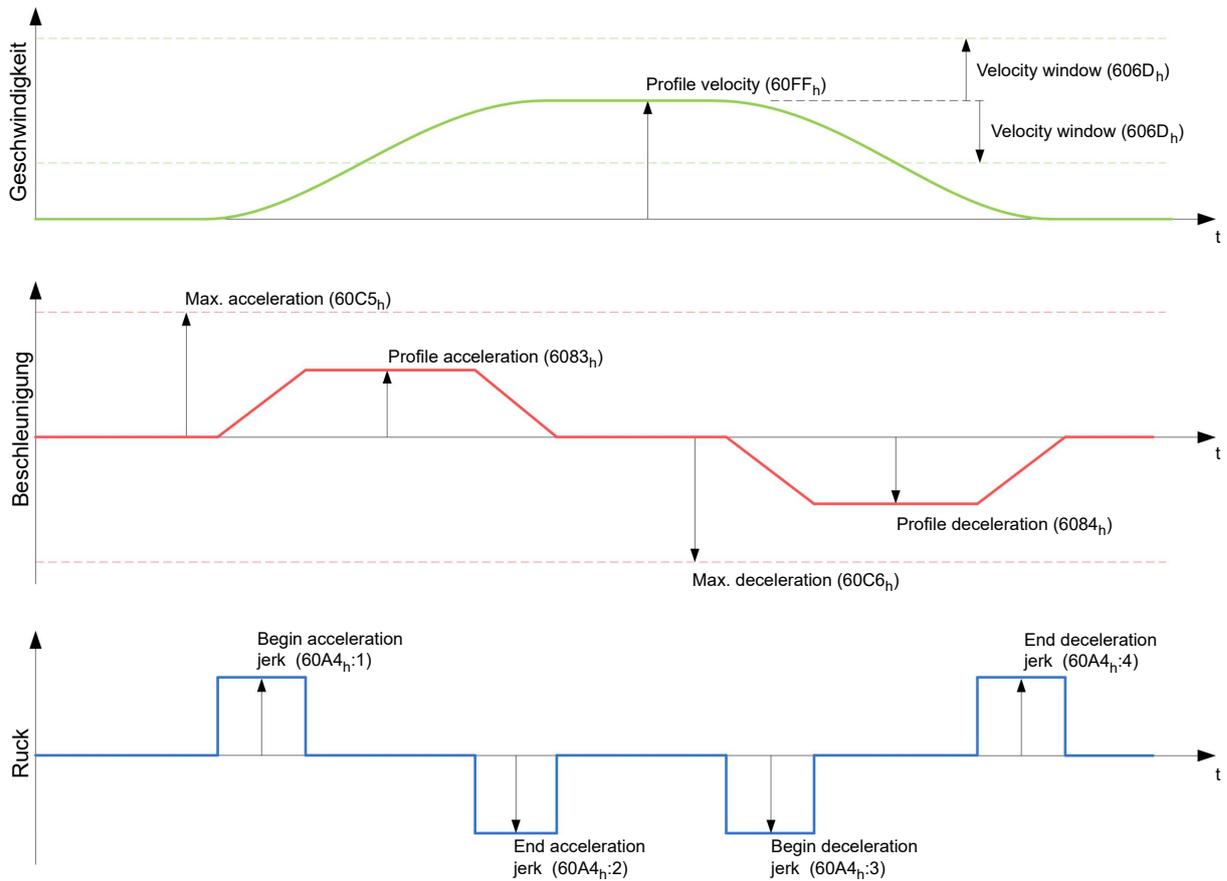


Aktivierung

Nachdem der Modus im Objekt **6060_h** (Modes Of Operation) ausgewählt wurde und die "Power State machine" (siehe "**CiA 402 Power State Machine**") auf *Operation enabled* geschaltet wurde, wird der Motor auf die Zielgeschwindigkeit im Objekt **60FF_h** beschleunigt (siehe nachfolgende Bilder). Dabei werden die Geschwindigkeits-, Beschleunigungs- und bei ruck-begrenzten Rampen auch die Ruckgrenzwerte berücksichtigt.

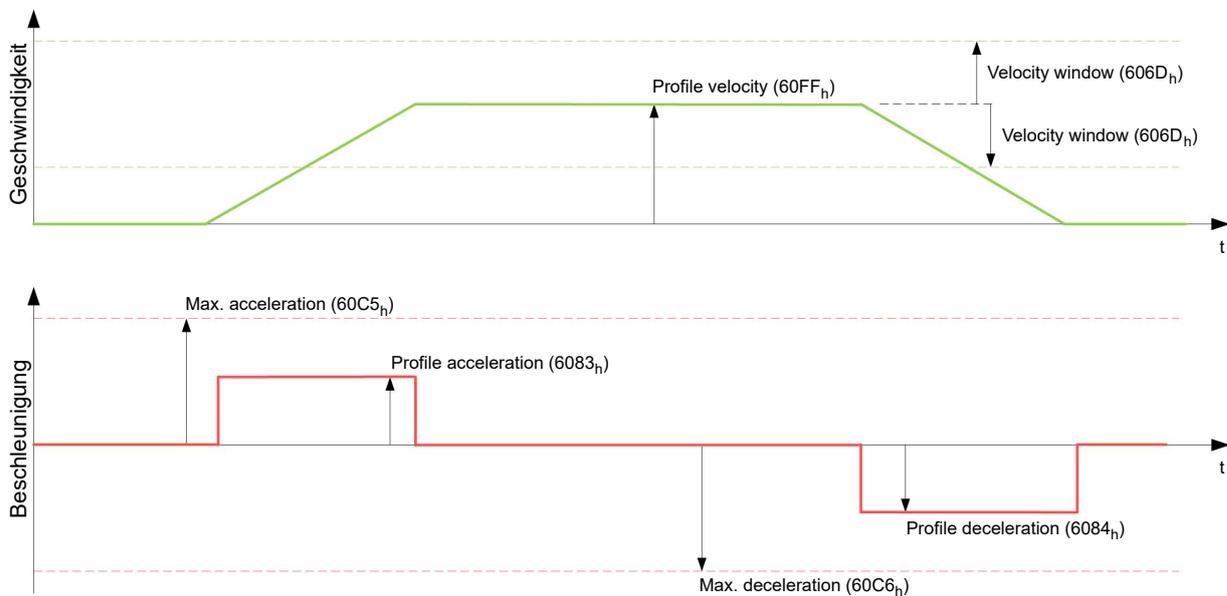
Limitierungen im ruck-limitierten Fall

Das folgende Bild zeigt die einstellbaren Limitierungen im ruck-limitierten Fall (**6086_h = 3**).



Limitierungen im Trapez-Fall

Dieses Bild zeigt die einstellbaren Limitierungen für den Trapez-Fall ($6086_h = 0$).



6.4 Profile Torque

6.4.1 Beschreibung

In diesem Modus wird das Drehmoment als Sollwert vorgegeben und über eine Rampenfunktion angefahren.



Hinweis

Dieser Modus funktioniert, nur wenn der **Closed Loop** aktiviert ist, siehe auch **Inbetriebnahme Closed Loop**.

6.4.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "4" gesetzt werden (siehe "**CiA 402 Power State Machine**").

6.4.3 Controlword

Folgende Bits im Objekt **6040_h** (Controlword) haben eine gesonderte Funktion:

- Bit 8 (Halt): Ist dieses Bit auf "1" gesetzt, bleibt der Motor stehen. Wird dieses Bit von "1" auf "0" gesetzt, wird der Motor den Vorgaben entsprechend angefahren. Beim Setzen von "0" auf "1" wird der Motor unter Berücksichtigung der Vorgabewerte wieder zum Stillstand gebracht.

6.4.4 Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

- Bit 10 (Target Reached): Dieses Bit gibt in Kombination mit dem Bit 8 des Objekts **6040_h** (Controlword) an, ob das vorgegebene Drehmoment erreicht ist (siehe nachfolgende Tabelle). Das Ziel gilt als erreicht wenn das Istdrehmoment (**6077_h Torque Actual Value**) eine vorgegebene Zeit (**203E_h Torque Window Time Out**) innerhalb eines Toleranzfensters (**203D_h Torque Window**) ist.

6040 _h Bit 8	6041 _h Bit 10	Beschreibung
0	0	Vorgegebenes Drehmoment nicht erreicht
0	1	Vorgegebenes Drehmoment erreicht
1	0	Achse beschleunigt
1	1	Geschwindigkeit der Achse ist 0

- Bit 11: Limit überschritten: Das Zieldrehmoment (**6071_h**) überschreitet das in **6072_h** eingegebene maximale Drehmoment.

6.4.5 Objekteinträge

Alle Werte der folgenden Einträge im Objektverzeichnis sind als Tausendstel des maximalen Drehmoments anzugeben, welches dem Nennstrom (**203B_h:01_h**) entspricht. Dazu zählen die Objekte:

- **6071_h** (Target Torque): Zielvorgabe des Drehmomentes
- **6072_h** (Max Torque): Maximales Drehmoment während der gesamten Rampe (Beschleunigen, Drehmoment halten, Abbremsen)
- **6073_h** (Max Current): Maximalstrom. Das Minimum von **6073_h** und **6072_h** wird als Limit für das Drehmoment in **6071_h** verwendet.
- **6074_h** (Torque Demand): Momentaner Ausgabewert des Rampengenerators (Drehmoment) für den Regler
- **6087_h** (Torque Slope): Max. Änderung des Drehmoments pro Sekunde



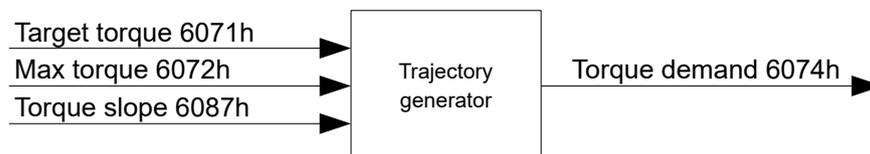
Hinweis

Diese Werte sind nicht limitiert auf 100% des Nennstroms (**203B_h:01_h**). Drehmomentwerte höher als das Nenn Drehmoment (generiert von dem Nennstrom) können erreicht werden, wenn die Maximaldauer (**203B_h:02_h**) des maximalen Stroms (**6073_h**) gesetzt wird (siehe **I2t Motor-Überlastungsschutz**). Alle Drehmoment-Objekte werden von dem maximalen Motorstrom (**2031_h**) limitiert.

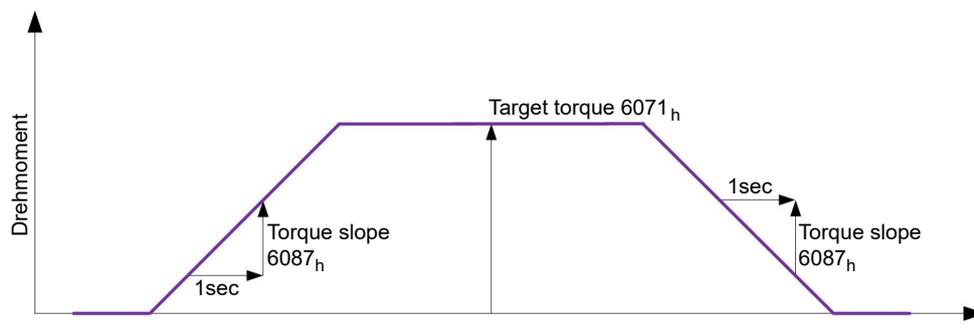
Die folgenden Objekte werden zudem für diesen Operationsmodus benötigt:

- **3202_h** Bit 5 (Motor Drive Submode Select):
Ist dieses Bit auf "0" gesetzt, wird der Antriebsregler im Drehmoment-begrenzten Velocity Mode betrieben, d.h. die maximale Geschwindigkeit kann in Objekt **6080_h** begrenzt werden und der Regler kann im Feldschwächebetrieb arbeiten.
Wird dieses Bit auf "1" gesetzt, arbeitet der Regler im ("Real") Torque Mode, die maximale Geschwindigkeit kann hier nicht begrenzt werden und der Feldschwächebetrieb ist nicht möglich.

Objekte des Rampengenerators



Torque-Verlauf



6.5 Homing

6.5.1 Übersicht

Beschreibung

Aufgabe der Referenzfahrt (Homing Method) ist es, den Positionsnullpunkt der Steuerung auf einen Encoder-Index bzw. Positionsschalter auszurichten.

Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "6" gesetzt werden (siehe "**CiA 402 Power State Machine**").



Tip

Werden Referenz- und/oder Endschalter verwendet, müssen diese Spezialfunktionen erst in der E/A-Konfiguration aktiviert werden (siehe "**Digitale Ein- und Ausgänge**").

Um die Endschalter zu verwenden, müssen Sie zusätzlich das Objekt **3701_h** auf "-1" setzen (Werkseinstellung), damit die weitere Fahrt des Motors nicht blockiert wird.

Controlword

Folgende Bits im Objekt **6040_h** (Controlword) haben eine gesonderte Funktion:

- Bit 4: Wird das Bit auf "1" gesetzt, wird die Referenzierung gestartet. Diese wird solange ausgeführt, bis entweder die Referenzposition erreicht wurde oder Bit 4 wieder auf "0" gesetzt wird.

Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

Bit 13	Bit 12	Bit 10	Beschreibung
0	0	0	Referenzfahrt wird ausgeführt
0	0	1	Referenzfahrt ist unterbrochen oder nicht gestartet
0	1	0	Referenzfahrt bestätigt, aber Ziel wurde noch nicht erreicht
0	1	1	Referenzfahrt vollständig abgeschlossen
1	0	0	Fehler während der Referenzfahrt, Motor dreht sich noch
1	0	1	Fehler während der Referenzfahrt, Motor im Stillstand



Hinweis

Das Bit 12 im Modus *Homing* wird nach der ersten vollständig abgeschlossenen Referenzfahrt seit dem Neustart auf 1 gesetzt. Es wird wieder auf 0 gesetzt nur

- während aller folgenden Referenzfahrt-Vorgänge,
- im Falle eines Fehlers während einer Referenzfahrt (dauerhaft gelöscht, bis eine neue Referenzfahrt vollständig abgeschlossen wird).

Objekteinträge

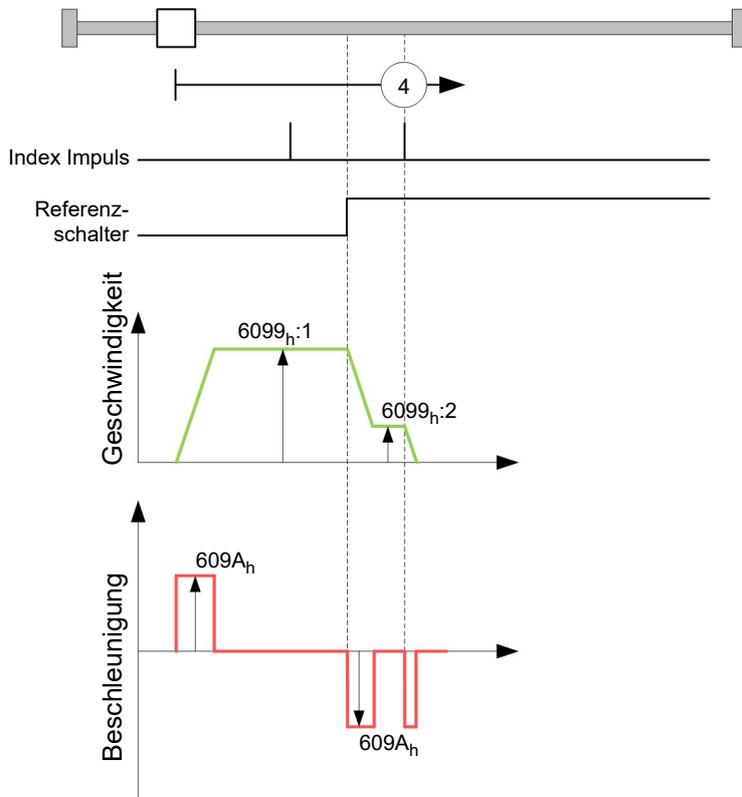
Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **607C_h** (Home Offset): Gibt die Differenz zwischen Null-Position der Steuerung und dem Referenzpunkt der Maschine in **benutzerdefinierten Einheiten** an.
- **6098_h** (Homing Method): Methode, mit der referenziert werden soll (siehe "**Referenzfahrt-Methode**")
- **6099_h:01_h** (Speed During Search For Switch): Geschwindigkeit für die Suche nach dem Schalter
- **6099_h:02_h** (Speed During Search For Zero): Geschwindigkeit für die Suche nach dem Index
- **6080_h** (Max Motor Speed): maximale Geschwindigkeit
- **609A_h** (Homing Acceleration): Anfahr- und Bremsbeschleunigung für die Referenzfahrt
- **203A_h:01_h** (Minimum Current For Block Detection): Minimale Stromschwelle, durch deren Überschreiten, das Blockieren des Motors an einem Block erkannt werden soll.
- **203A_h:02_h** (Period Of Blocking):

Gibt die Zeit in ms an, die der Motor nach der Blockdetektion trotzdem noch gegen den Block fahren soll.

Geschwindigkeiten der Referenzfahrt

Das Bild zeigt die Geschwindigkeiten der Referenzfahrt am Beispiel der Methode 4:



6.5.2 Referenzfahrt-Methode

Beschreibung

Die Referenzfahrt-Methode wird als Zahl in das Objekt 6098_h geschrieben und entscheidet darüber, ob auf eine Schalterflanke (steigend/fallend), eine Stromschwelle für Blockdetektion bzw. einen Index-Impuls referenziert wird oder in welche Richtung die Referenzfahrt startet. Methoden, die den Index-Impuls des Encoders benutzen, liegen im Zahlenbereich 1 bis 14, 33 und 34. Methoden, die den Index-Impuls des Encoders nicht benutzen, liegen zwischen 17 und 30, sind in den Fahrprofilen aber identisch mit den Methoden 1 bis 14. Diese Zahlen sind in den nachfolgenden Abbildungen eingekreist dargestellt. Methoden, bei denen keine Endschalter eingesetzt werden und stattdessen das Fahren gegen einen Block erkannt werden soll, müssen mit einem Minus vor der Methodenanzahl aufgerufen werden.

Für die nachfolgenden Grafiken gilt die negative Bewegungsrichtung nach links. Der Endschalter (*limit switch*) liegt jeweils vor der mechanischen Blockierung, der Referenzschalter (*home switch*) liegt zwischen den beiden Endschaltern. Die Index-Impulse kommen vom angeschlossenen Encoder.

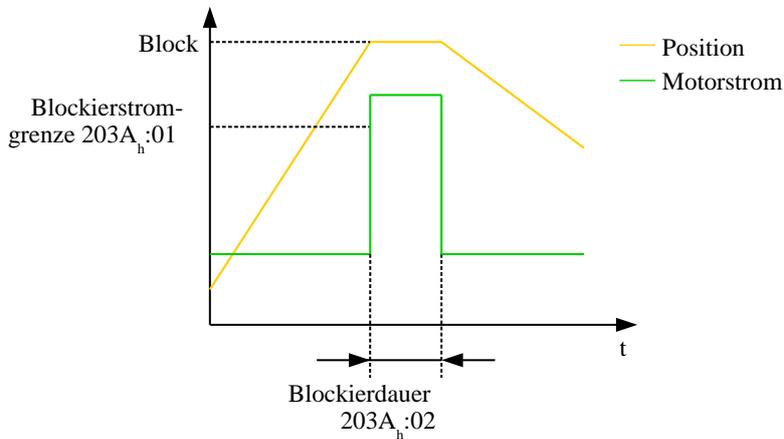
Bei Methoden, die Homing auf Block benutzen, gelten die gleichen Abbildungen wie für die Methoden mit Endschalter. Da sich außer den fehlenden Endschaltern nichts ändert, wurde auf neue Abbildungen verzichtet. Hier gilt für die Abbildungen, dass die Endschalter durch eine mechanische Blockierung ersetzt werden müssen.

Homing auf Block

Homing auf Block funktioniert derzeit nur im *Closed Loop*-Betrieb.

"Homing auf Block" funktioniert wie jede Homing-Methode mit dem Unterschied, dass zur Positionierung - anstelle auf einen Endschalter - auf einen Block (Endanschlag) gefahren wird. Dabei sind zwei Einstellungen vorzunehmen:

1. Stromhöhe: im Objekt **203A_h:01** wird die Stromhöhe definiert, ab der ein Fahren gegen den Block erkannt wird.
2. Blockierdauer: im Objekt **203A_h:02** wird die Dauer, während der Motor gegen den Block fährt, eingestellt.



Methoden-Überblick

Die Methoden 1 bis 14, sowie 33 und 34 benutzen den Index-Impuls des Encoders.

Die Methoden 17 bis 32 sind identisch mit den Methoden 1 bis 14, mit dem Unterschied, dass nur noch auf den End- oder Referenzschalter referenziert wird und nicht auf den Index-Impuls.

- Methoden 1 bis 14 verwenden einen Index-Impuls.
- Methoden 17 bis 30 verwenden keinen Index-Impuls.
- Methoden 33 und 34 referenzieren nur auf den nächsten Index-Impuls.
- Methode 35 referenziert auf die aktuelle Position.

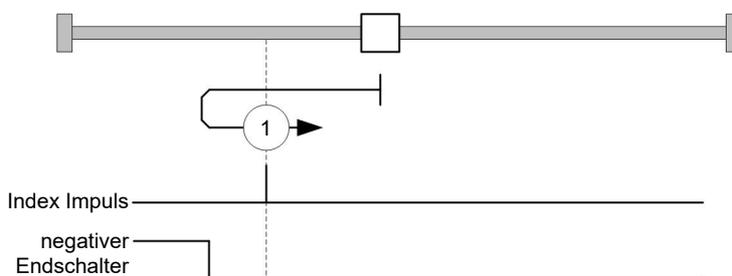
Folgende Methoden können für Homing auf Block benutzt werden:

- Methoden -1 bis -2 und -7 bis -14 enthalten einen Index-Impuls
- Methoden -17 bis -18 und -23 bis -30 haben keinen Index-Impuls

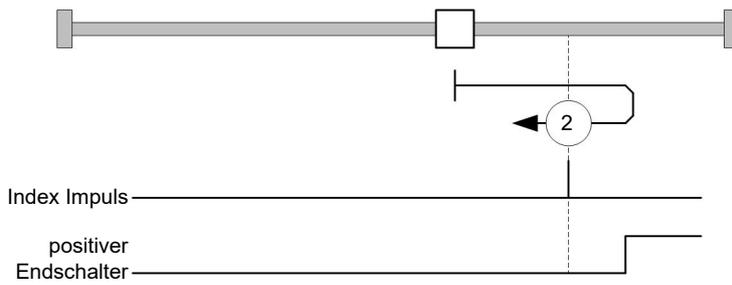
Methoden 1 und 2

Referenzieren auf Endschalter und Index-Impuls.

Methode 1 referenziert auf negativen Endschalter und Index-Impuls:



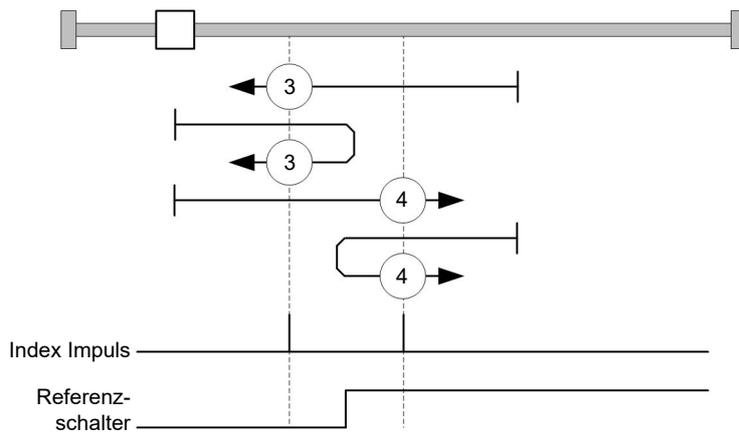
Methode 2 referenziert auf positiven Endschalter und Index-Impuls:



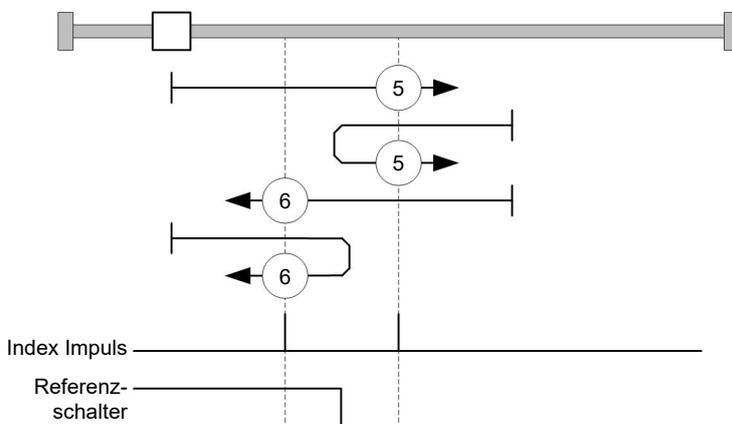
Methoden 3 bis 6

Referenzieren auf die Schaltflanke des Referenzschalters und Index-Impuls.

Bei den Methoden 3 und 4 wird die linke Schaltflanke des Referenzschalters als Referenz verwendet:



Bei den Methoden 5 und 6 wird die rechte Schaltflanke des Referenzschalters als Referenz verwendet:

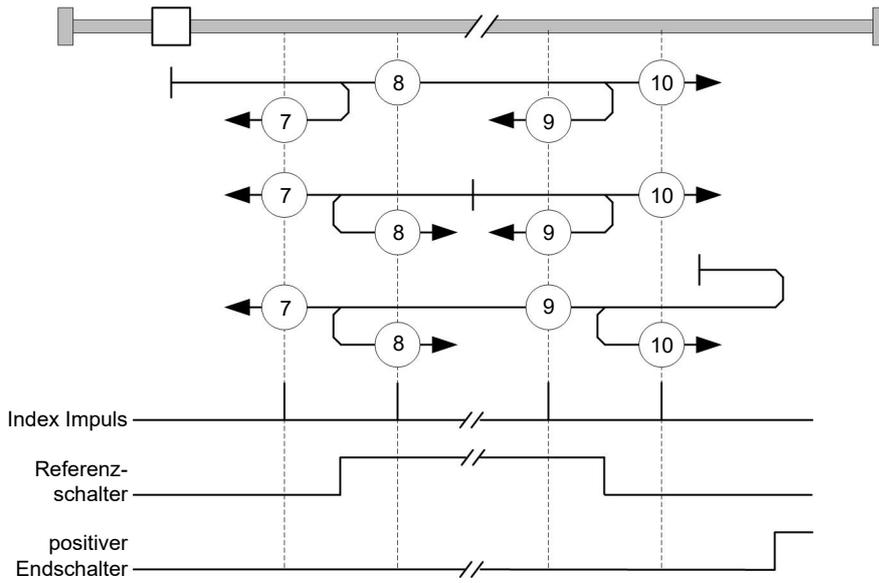


Methoden 7 bis 14

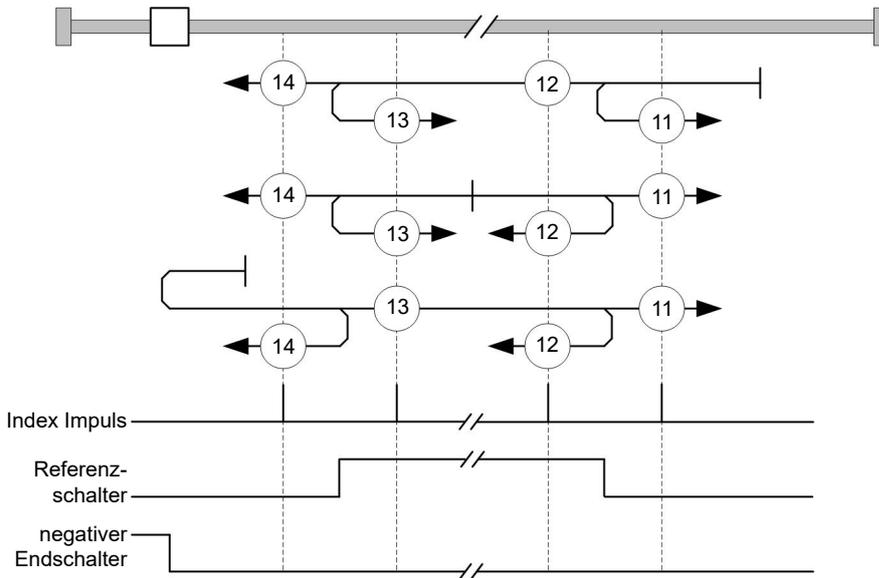
Referenzieren auf Referenzschalter und Index-Impuls (mit Endschaltern).

Bei diesen Methoden ist die derzeitige Position relativ zum Referenzschalter unwichtig. Mit der Methode 10 wird beispielsweise immer auf den Index-Impuls rechts neben der rechten Flanke des Referenzschalters referenziert.

Die Methoden 7 bis 10 berücksichtigen den positiven Endschalter:



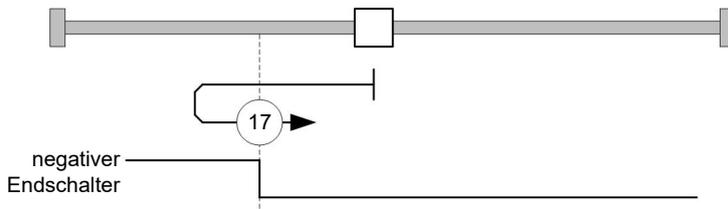
Die Methoden 11 bis 14 berücksichtigen den negativen Endschalter:



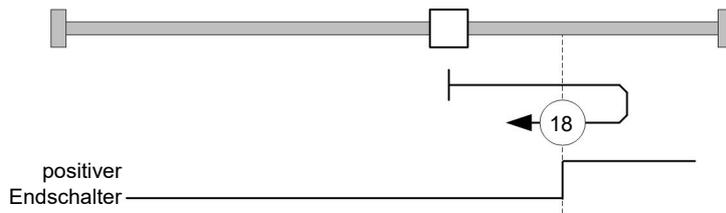
Methoden 17 und 18

Referenzieren auf den Endschalter ohne den Index-Impuls.

Methode 17 referenziert auf den negativen Endschalter:



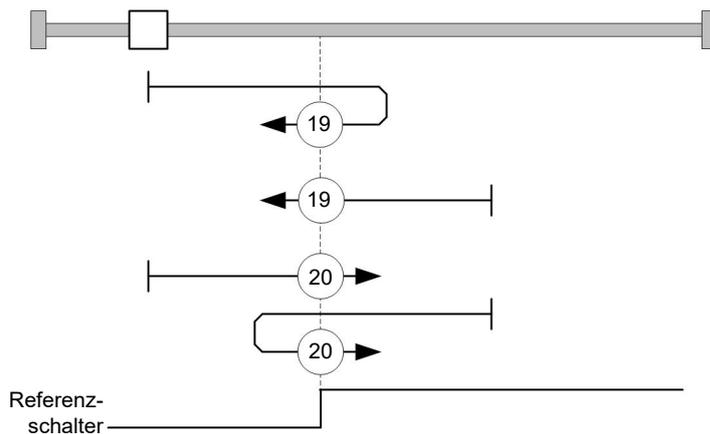
Methode 18 referenziert auf den positiven Endschalter:



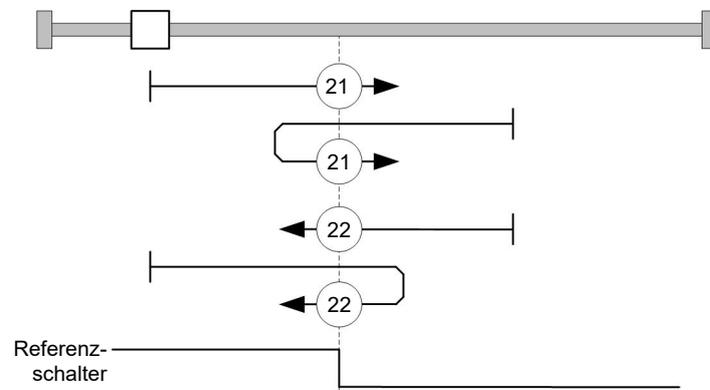
Methoden 19 bis 22

Referenzieren auf die Schaltflanke des Referenzschalters ohne den Index-Impuls.

Bei den Methoden 19 und 20 (äquivalent zu Methoden 3 und 4) wird die linke Schaltflanke des Referenzschalters als Referenz verwendet:



Bei den Methoden 21 und 22 (äquivalent zu Methoden 5 und 6) wird die rechte Schaltflanke des Referenzschalters als Referenz verwendet:

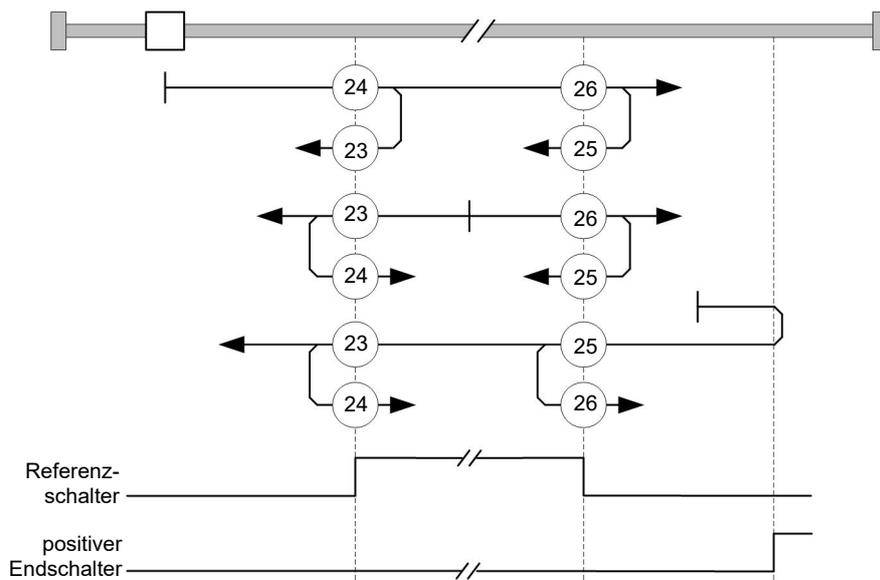


Methoden 23 bis 30

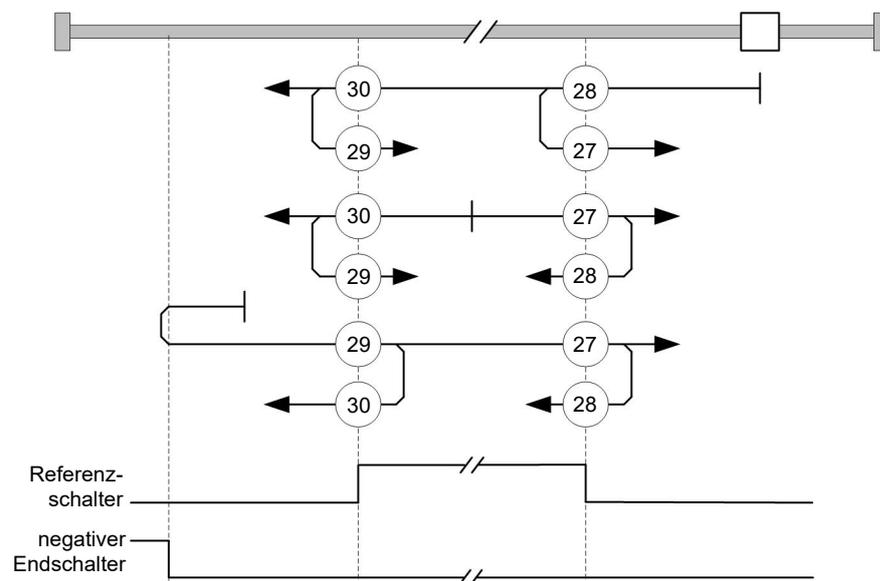
Referenzieren auf Referenzschalter ohne den Index-Impuls (mit Endschaltern).

Bei diesen Methoden ist die derzeitige Position relativ zum Referenzschalter unwichtig. Mit der Methode 26 wird beispielsweise immer auf den Index-Impuls rechts neben der rechten Flanke des Referenzschalters referenziert.

Die Methoden 23 bis 26 berücksichtigen den positiven Referenzschalter:



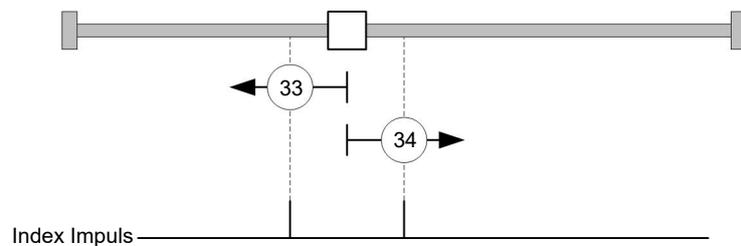
Die Methoden 27 bis 30 berücksichtigen den negativen Referenzschalter:



Methoden 33 und 34

Referenzieren auf den nächsten Index-Impuls.

Bei diesen Methoden wird nur auf den jeweils folgenden Index-Impuls referenziert:



Methode 35

Referenziert auf die aktuelle Position.



Hinweis

Für den Homing Mode 35 ist es nicht notwendig, die **CiA 402 Power State Machine** in den Status "Operation Enabled" zu schalten. Auf diese Weise kann vermieden werden, dass durch eine Bestromung der Motorwicklungen im *Open Loop*-Betrieb, die aktuelle Position nach dem Homing Mode 35 nicht genau 0 ist.

6.6 Interpolated Position Mode

6.6.1 Übersicht

Beschreibung

Der *Interpolated Position Mode* dient zum Synchronisieren mehrerer Achsen. Hierzu übernimmt eine übergeordnete Steuerung die Rampen- bzw. Bahnberechnung und überträgt die jeweilige Sollposition, bei der sich die Achse zu einem bestimmten Zeitpunkt befinden soll, zur Steuerung. Zwischen diesen Positions-Stützstellen interpoliert die Steuerung.

Synchronisierung zum SYNC-Objekt

Für den *Interpolated Position Mode* ist es notwendig, dass sich die Steuerung auf das SYNC-Objekt (abhängig vom Feldbus) aufsynchronisiert. Dieses SYNC-Objekt ist in regelmäßigen Zeitabständen von der übergeordneten Steuerung zu senden. Die Synchronisation erfolgt, sobald die Steuerung in den NMT-Modus *Operational* geschaltet wird.



Hinweis

Es wird empfohlen, wenn möglich ein Zeitintervall des *SYNC-Objekts* zu nutzen.

6.6.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "7" gesetzt werden (siehe "**CiA 402 Power State Machine**").

6.6.3 Controlword

Folgende Bits im Objekt **6040_h** (Controlword) haben eine gesonderte Funktion:

- Bit 4 aktiviert die Interpolation, wenn es auf "1" gesetzt wird.
- Bit 8 (Halt): Ist dieses Bit auf "1" gesetzt, bleibt der Motor stehen. Bei einem Übergang von "1" auf "0" beschleunigt der Motor mit der eingestellten Startrampe bis zur Zielgeschwindigkeit. Bei einem Übergang von "0" auf "1" bremst der Motor ab und bleibt stehen. Die Bremsbeschleunigung ist dabei abhängig von der Einstellung des "Halt Option Code" im Objekt **605D_h**.

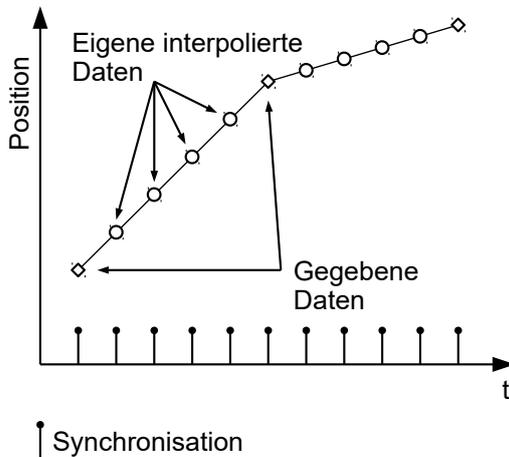
6.6.4 Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

- Bit 10: Zielposition erreicht: Dieses bit ist auf "1" gesetzt, wenn die Zielposition erreicht wurde (sollte das Halt-Bit im Controlword "0" sein) oder die Achse hat die Geschwindigkeit 0 (falls das Halt-Bit im letzten Controlword "1" war).
- Bit 12 (IP Modus aktiv): Dieses Bit wird auf "1" gesetzt, wenn die Interpolation aktiv ist.
- Bit 13 (Following Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schleppfehler größer als die eingestellten Grenzen ist (**6065_h** (Following Error Window) und **6066_h** (Following Error Time Out)).

6.6.5 Benutzung

Die Steuerung folgt einem linear interpolierten Pfad zwischen der aktuellen und der vorgegebenen Zielposition. Die (nächste) Zielposition muss in das Datensatz **60C1_h:01_h** geschrieben werden.



In der derzeitigen Implementation wird nur

- lineare Interpolation
- und eine Zielposition

unterstützt.

6.6.6 Setup

Das folgende Setup ist nötig:

- **60C2_h:01_h**: Zeit zwischen zwei übergebenen Zielpositionen in ms.
- **60C4_h:06_h**: dieses Objekt ist auf "1" zu setzen um die Zielposition im Objekt **60C1_h:01_h** modifizieren zu dürfen.
- **6081_h** (Profile Velocity): maximale Geschwindigkeit, mit der die Position angefahren werden soll
- Die Geschwindigkeit wird durch **607F_h** (Max Profile Velocity) und **6080_h** (Max Motor Speed) begrenzt, der kleinere Wert wird als Grenze herangezogen.
- Um den Motor drehen zu können, ist die *Power state machine* auf den Status *Operation enabled* zu setzen (siehe **CiA 402 Power State Machine**)

6.6.7 Operation

Nach dem Setup ist die Aufgabe der übergerodeten Steuerung, die Zielpositionen rechtzeitig in das Objekt **60C1_h:01_h** zu schreiben.

6.7 Cyclic Synchronous Position

6.7.1 Übersicht

Beschreibung

In diesem Modus wird der Steuerung in festen Zeitabständen (im Folgenden *Zyklus* genannt) über den Feldbus eine absolute Positionsvorgabe übergeben. Die Steuerung berechnet dabei keine Rampen mehr, sondern folgt nur noch den Vorgaben.

Die Zielposition wird zyklisch (per *PDO*) übertragen. Das Bit 4 im Controlword muss nicht gesetzt werden (im Gegensatz zum **Profile Position** Modus).



Hinweis

Die Zielvorgabe ist absolut und damit unabhängig davon, wie oft sie pro *Zyklus* versendet wurde.

Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "8" gesetzt werden (siehe "**CiA 402 Power State Machine**").

Controlword

In diesem Modus haben die Bits des Controlword **6040_h** keine gesonderte Funktion.

Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

Bit	Wert	Beschreibung
8	0	Steuerung ist nicht synchron zum Feldbus
8	1	Steuerung ist synchron zum Feldbus
10	0	Reserviert
10	1	Reserviert
12	0	Steuerung folgt nicht der Zielvorgabe, die Vorgabe des 607A_h (Target Position) wird ignoriert
12	1	Steuerung folgt der Zielvorgabe, das Objekt 607A_h (Target Position) wird als Eingabe für die Positionsregelung genutzt.
13	0	Kein Schleppfehler
13	1	Schleppfehler

Bit 11: Limit überschritten: Die Sollposition über- oder unterschreitet die in **607D_h** eingegebenen Grenzwerte.

6.7.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **607A_h** (Target Position): Dieses Objekt muss zyklisch mit dem Positions-Sollwert beschrieben werden.
- **607B_h** (Position Range Limit): Dieses Objekt enthält die Vorgabe für einen Über- oder Unterlauf der Positionsangabe.
- **607D_h** (Software Position Limit): Dieses Objekt legt die Limitierungen fest, innerhalb deren sich die Positionsvorgabe (**607A_h**) befinden muss.
- **6065_h** (Following Error Window): Dieses Objekt gibt einen Toleranz-Korridor in positiver wie negativer Richtung von der Sollvorgabe vor. Befindet sich die Ist-Position länger als die vorgegebene Zeit (**6066_h**) außerhalb dieses Korridors, wird ein Schleppfehler gemeldet.
- **6066_h** (Following Error Time Out): Dieses Objekt gibt den Zeitbereich in Millisekunden vor. Sollte sich die Ist-Position länger als dieser Zeitbereich außerhalb des Positions-Korridors (**6065_h**) befinden, wird ein Schleppfehler ausgelöst.
- **6085_h** (Quick-Stop Deceleration): Dieses Objekt hält die Bremsbeschleunigung für den Fall, dass ein Quick-Stop ausgelöst wird.
- **605A_h** (Quick-Stop Option Code): Dieses Objekt enthält die Option, die im Falle eines Quick-Stops ausgeführt werden soll.
- **6080_h** (Max Motor Speed): maximale Geschwindigkeit

- **60C2_h:01_h** (Interpolation Time Period): Dieses Objekt gibt die Zeit eines *Zyklus* vor, in diesen Zeitabständen muss ein neuer Sollwert in das **607A_h** geschrieben werden. Es gilt dabei: $\text{Zykluszeit} = \text{Wert des } 60C2_{h:01_{h}} * 10^{\text{Wert des } 60C2_{h:02_{h}}} \text{ Sekunden}$.
- **60C2_h:02_h** (Interpolation Time Index): Dieses Objekt gibt die Zeitbasis der Zyklen an. Derzeit wird nur der Wert **60C2_h:02_h=-3** unterstützt, das ergibt eine Zeitbasis von 1 Millisekunde.

Folgende Objekte können in dem Modus ausgelesen werden:

- **6064_h** (Position Actual Value)
- **606C_h** (Velocity Actual Value)
- **60F4_h** (Following Error Actual Value)

6.8 Cyclic Synchronous Velocity

6.8.1 Übersicht

Beschreibung

In diesem Modus wird der Steuerung in festen Zeitabständen (im Folgenden *Zyklus* genannt) über den Feldbus eine Geschwindigkeitsvorgabe übergeben. Die Steuerung berechnet dabei keine Rampen mehr, sondern folgt nur noch den Vorgaben.

Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "9" gesetzt werden (siehe "**CiA 402 Power State Machine**").

Controlword

In diesem Modus haben die Bits des Controlword **6040_h** keine gesonderte Funktion.

Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

Bit	Wert	Beschreibung
8	0	Steuerung ist nicht synchron zum Feldbus
8	1	Steuerung ist synchron zum Feldbus
10	0	Reserviert
10	1	Reserviert
12	0	Steuerung folgt nicht der Zielvorgabe, die Vorgabe des 60FF_h (Target Velocity) wird ignoriert
12	1	Steuerung folgt der Zielvorgabe, das Objekt 60FF_h (Target Velocity) wird als Eingabe für die Positionsregelung genutzt.
13	0	Reserviert
13	1	Reserviert

6.8.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **60FF_h** (Target Velocity): Dieses Objekt muss zyklisch mit dem Geschwindigkeits-Sollwert beschrieben werden.
- **6085_h** (Quick-Stop Deceleration): Dieses Objekt hält die Bremsbeschleunigung für den Fall, dass ein Quick-Stop ausgelöst wird (siehe "**CiA 402 Power State Machine**").

- **605A_h** (Quick-Stop Option Code): Dieses Objekt enthält die Option, die im Falle eines Quick-Stops ausgeführt werden soll (siehe "**CiA 402 Power State Machine**").
- **6080_h** (Max Motor Speed): maximale Geschwindigkeit
- **60C2_h:01_h** (Interpolation Time Period): Dieses Objekt gibt die Zeit eines *Zyklus* vor, in diesen Zeitabständen muss ein neuer Sollwert in das **60FF_h** geschrieben werden.
Es gilt dabei: Zykluszeit = Wert des **60C2_h:01_h** * 10^{Wert des 60C2:02} Sekunden.
- **60C2_h:02_h** (Interpolation Time Index): Dieses Objekt gibt die Zeitbasis der Zyklen an. Derzeit wird nur der Wert **60C2_h:02_h=-3** unterstützt, das ergibt eine Zeitbasis von 1 Millisekunde.

Folgende Objekte können in dem Modus ausgelesen werden:

- **606C_h** (Velocity Actual Value)
- **607E_h** (Polarity)

6.9 Cyclic Synchronous Torque

6.9.1 Übersicht

Beschreibung

In diesem Modus wird der Steuerung in festen Zeitabständen (im Folgenden *Zyklus* genannt) über den Feldbus eine absolute Drehmomentvorgabe übergeben. Die Steuerung berechnet dabei keine Rampen mehr, sondern folgt nur noch den Vorgaben.



Hinweis

Dieser Modus funktioniert nur wenn der **Closed Loop** aktiviert ist, siehe auch **Inbetriebnahme Closed Loop**.

Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "10" gesetzt werden (siehe "**CiA 402 Power State Machine**").

Controlword

In diesem Modus haben die Bits des Controlword **6040_h** keine gesonderte Funktion.

Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

Bit	Wert	Beschreibung
8	0	Steuerung ist nicht synchron zum Feldbus
8	1	Steuerung ist synchron zum Feldbus
10	0	Reserviert
10	1	Reserviert
12	0	Steuerung folgt nicht der Zielvorgabe, die Vorgabe des 6071_h (Target Torque) wird ignoriert
12	1	Steuerung folgt der Zielvorgabe, das Objekt 6071_h (Target Torque) wird als Eingabe für die Positionsregelung genutzt.
13	0	Reserviert
13	1	Reserviert

6.9.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **6071_h** (Target Torque): Dieses Objekt muss zyklisch mit dem Drehmoment-Sollwert beschrieben werden und ist relativ zu **6072_h** einzustellen.
- **6072_h** (Max Torque): Beschreibt das maximal zulässige Drehmoment.
- **6073_h** (Max Current):
Maximaler Strom. Das Minimum von **6073_h** und **6072_h** wird als Limit für das Drehmoment in **6071_h** verwendet.
- **6080_h** (Max Motor Speed): maximale Geschwindigkeit
- **60C2_h:01_h** (Interpolation Time Period): Dieses Objekt gibt die Zeit eines *Zyklus* vor, in diesen Zeitabständen muss ein neuer Sollwert in das **6071_h** geschrieben werden.
Es gilt dabei: $\text{Zykluszeit} = \text{Wert des } 60C2_{h:01_{h}} * 10^{\text{Wert des } 60C2_{h:02_{h}}} \text{ Sekunden}$.
- **60C2_h:02_h** (Interpolation Time Index): Dieses Objekt gibt die Zeitbasis der Zyklen an. Derzeit wird nur der Wert **60C2_h:02_h=-3** unterstützt, das ergibt eine Zeitbasis von 1 Millisekunde.

Folgende Objekte können in dem Modus ausgelesen werden:

- **606C_h** (Velocity Actual Value)
- **6074_h** (Torque Demand)

6.10 Takt-Richtungs-Modus

6.10.1 Beschreibung

Im Takt-Richtungs-Modus wird der Motor über zwei Eingänge durch eine übergeordnete Positioniersteuerung mit einem Takt- und einem Richtungssignal betrieben. Bei jedem Takt führt der Motor einen Schritt in die dem Richtungssignal entsprechende Richtung aus.

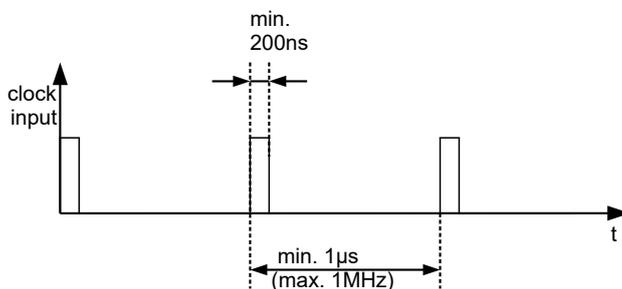
6.10.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "-1" (bzw. "FFh") gesetzt werden (siehe "**CiA 402 Power State Machine**").

6.10.3 Generelles

Folgende Daten gelten für jede Unterart des Takt-Richtungs-Modus:

- Die maximale Frequenz der Eingangspulse liegt bei 1MHz, der ON-Puls sollte dabei nicht kleiner als 200 ns werden.



- Die Skalierung der Schritte erfolgt über die Objekte **2057_h** und **2058_h**. Dabei gilt die folgende Formel:

$$\text{Schrittweite pro Puls} = \frac{2057_{\text{h}}}{2058_{\text{h}}}$$

Ab Werk ist der Wert "Schrittweite pro Puls" = 128 (**2057_h**=128 und **2058_h**=1) eingestellt, was einem Viertelschritt pro Puls entspricht. Ein Vollschritt ist der Wert "512", ein Halbschritt pro Puls entsprechend "256" usw.



Hinweis

Bei einem Schrittmotor mit 50 Polpaaren entsprechen 200 Vollschritte einer mechanischen Umdrehung der Motorwelle.

Die BLDC-Motoren werden von der Steuerung im *Takt-Richtungs-Modus* auch als Schrittmotoren behandelt. Das bedeutet, dass, bei einem BLDC-Motor mit z.B. 3 Polpaaren, 12 (=4*3) Vollschritte einer Umdrehung entsprechen.



Hinweis

Bei einem Richtungswechsel ist es nötig, mindestens eine Zeit von 35µs verstreichen zu lassen, bevor der neue Takt angelegt wird.

6.10.4 Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

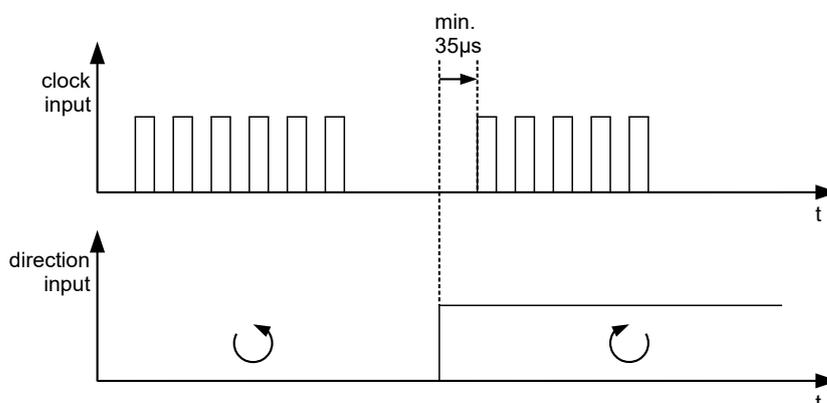
- Bit 13 (Following Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schlepfehler größer als die eingestellten Grenzen ist (**6065_h** (Following Error Window) und **6066_h** (Following Error Time Out)).

6.10.5 Unterarten des Takt-Richtungs-Modus

Takt-Richtungs-Modus (TR-Modus)

Um den Modus zu aktivieren muss das Objekt **205B_h** auf den Wert "0" gesetzt sein (Werkseinstellung).

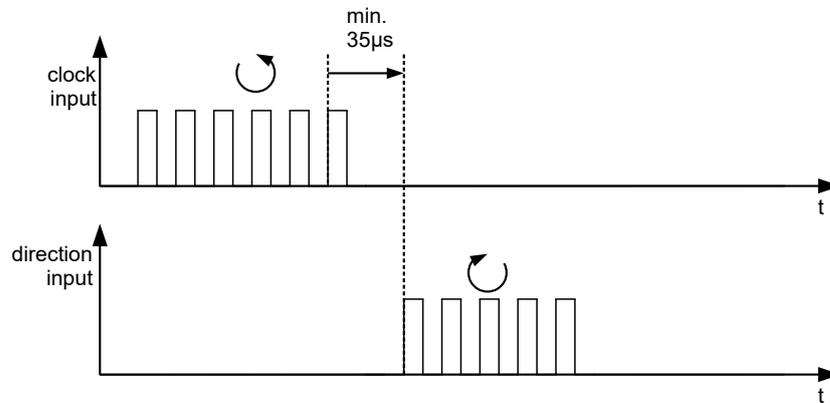
In diesem Modus müssen über den Takteingang die Pulse vorgegeben werden, das Signal des Richtungseingangs gibt dabei die Drehrichtung vor (siehe nachfolgende Grafik).



Rechts-/Linkslauf-Modus (CW/CCW-Modus)

Um den Modus zu aktivieren muss das Objekt **205B_h** auf den Wert "1" gesetzt sein.

In diesem Modus entscheidet der verwendete Eingang über die Drehrichtung (siehe nachfolgende Grafik).



6.11 Auto-Setup

6.11.1 Beschreibung

Um einige Parameter im Bezug zum Motor und den angeschlossenen Sensoren (Encoder/Hallsensoren) zu ermitteln, wird ein *Auto-Setup* durchgeführt. Der **Closed Loop** Betrieb setzt ein erfolgreich abgeschlossenes *Auto-Setup* voraus. Das *Auto-Setup* ist nur einmal bei der Inbetriebnahme durchzuführen, solange sich der an der Steuerung angeschlossene Motor/Sensor nicht ändert. Für Details siehe **entsprechenden Abschnitt im Kapitel Inbetriebnahme**.

6.11.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060_h** (Modes Of Operation) der Wert "-2" ("FE_h") gesetzt werden (siehe **CiA 402 Power State Machine**).

6.11.3 Controlword

Folgende Bits im Objekt **6040_h** (Controlword) haben eine gesonderte Funktion:

- Bit 4 startet einen Fahrauftrag. Dieser wird bei einem Übergang von "0" nach "1" übernommen.

6.11.4 Statusword

Folgende Bits im Objekt **6041_h** (Statusword) haben eine gesonderte Funktion:

- Bit 10: Indexed: zeigt an, ob ein Encoder-Index gefunden wurde (= "1") oder nicht (= "0").
- Bit 12: Aligned: dieses Bit wird auf "1" gesetzt, nachdem das *Auto-Setup* beendet ist

7 Spezielle Funktionen

7.1 Digitale Ein- und Ausgänge

Diese Steuerung verfügt über digitale Ein- und Ausgänge.

7.1.1 Bitzuordnung

Die Software der Steuerung ordnet jedem Eingang und Ausgang zwei Bits im jeweiligen Objekt (z.B. **60FD_h Digital Inputs** bzw. **60FE_h Digital Outputs**) zu:

1. Das erste Bit entspricht der Spezialfunktion eines Ausganges oder Eingangs. Diese Funktionen sind immer verfügbar auf den Bits 0 bis einschließlich 15 des jeweiligen Objekts. Darunter fallen die Endschalter und der Referenzschalter bei den digitalen Eingängen und die Bremsensteuerung bei den Ausgängen.
2. Das zweite Bit zeigt den Aus-/Eingang an sich als Pegel, diese sind auf Bit 16 bis 31 verfügbar.

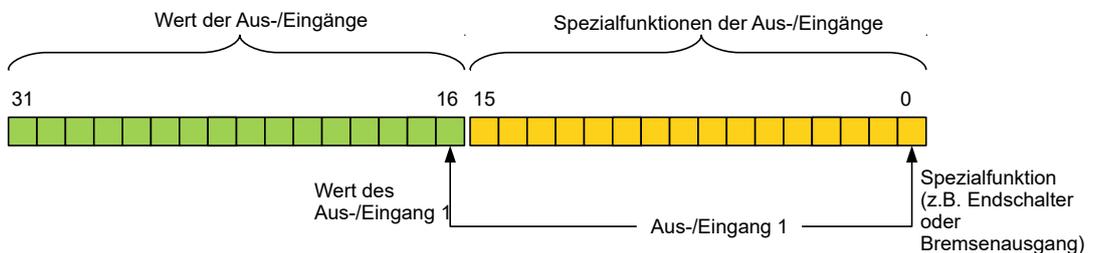
Beispiel

Um den Wert des Ausgangs 2 zu manipulieren, ist immer Bit 17 in **60FE_h** zu benutzen.

Um die Spezialfunktion "Negativer Endschalter" des Eingangs 1 zu aktivieren, ist Bit 0 in **3240_h:01_h** zu setzen, und um den Zustand des Eingangs abzufragen ist Bit 0 in **60FD_h** zu lesen. Das Bit 16 in **60FD_h** zeigt ebenfalls den Zustand des Eingangs 1 (unabhängig davon, ob die Spezialfunktion des Eingangs aktiviert wurde oder nicht).

In der nachfolgenden Zeichnung ist diese Zuordnung graphisch dargestellt.

Bits eines beliebigen Objektes zur Steuerung eines Aus-/Eingangs



7.1.2 Digitale Eingänge

Übersicht



Hinweis

Bei Digitaleingängen mit 5 V darf die Länge der Zuleitungen 3 Meter nicht überschreiten.



Hinweis

Die digitalen Eingänge werden einmal pro Millisekunde erfasst. Signaländerungen am Eingang kürzer als eine Millisekunde werden nicht verarbeitet.

Folgende Eingänge stehen zur Verfügung:

Eingan Sonderfunktion		Schaltswelle umschaltbar	Differenziell / single-ended
1	Negativer Endschalter	nein	single-ended
2	Positiver Endschalter	nein	single-ended
3	Referenzschalter / Richtungseingang im Takt- Richtungs-Modus	nein	single-ended
4	Takteingang im Takt- Richtungs-Modus	nein	single-ended

Objekteinträge

Über die folgenden OD-Einstellungen kann der Wert eines Eingangs manipuliert werden, wobei hier immer nur das entsprechende Bit auf den Eingang wirkt.

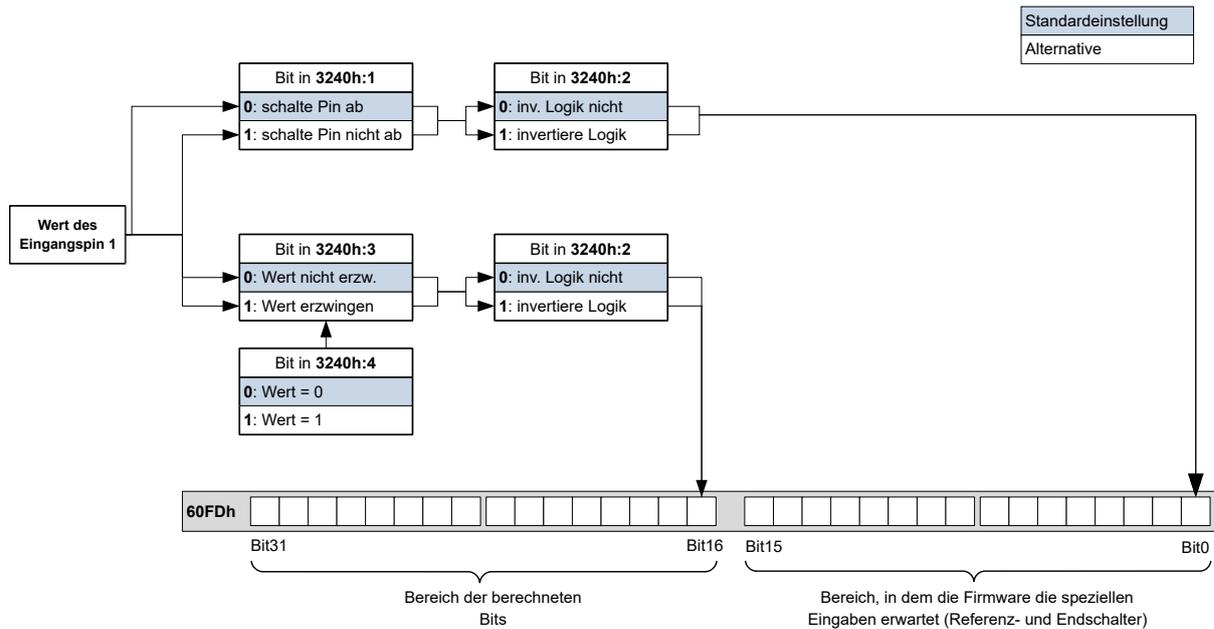
- **3240_h:01_h** (Special Function Enable): Dieses Bit erlaubt Sonderfunktionen eines Eingangs aus- (Wert "0") oder einzuschalten (Wert "1"). Soll Eingang 1 z.B. nicht als negativer Endschalter verwendet werden, so muss die Sonderfunktion abgeschaltet werden, damit nicht fälschlicherweise auf den Signalgeber reagiert wird. Auf die Bits 16 bis 31 hat das Objekt keine Auswirkungen. Die Firmware wertet folgende Bits aus:
 - Bit 0: Negativer Endschalter
 - Bit 1: Positiver Endschalter
 - Bit 2: Referenzschalter

Sollen z.B. zwei Endschalter und ein Referenzschalter verwendet werden, müssen Bits 0-2 in **3240_h:01_h** auf "1" gesetzt werden
- **3240_h:02_h** (Function Inverted): Dieser Subindex wechselt von Schließer-Logik (ein logischer High-Pegel am Eingang ergibt den Wert "1" im Objekt **60FD_h**) auf Öffner-Logik (der logische High-Pegel am Eingang ergibt den Wert "0"). Das gilt für die Sonderfunktionen (außer den Takt- und Richtungseingängen) und für die normalen Eingänge. Hat das Bit den Wert "0" gilt Schließer-Logik, entsprechend bei dem Wert "1" die Öffner-Logik. Bit 0 wechselt die Logik des Eingangs 1, Bit 1 die Logik des Eingangs 2 usw. .
- **3240_h:03_h** (Force Enable): Dieser Subindex schaltet die Softwaresimulation von Eingangswerten ein, wenn das entsprechende Bit auf "1" gesetzt ist. Dann werden nicht mehr die tatsächlichen, sondern die in Objekt **3240_h:04_h** eingestellten Werte für den jeweiligen Eingang verwendet. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw..
- **3240_h:04_h** (Force Value): Dieses Bit gibt den Wert vor, der als Eingangswert eingelesen werden soll, wenn das gleiche Bit im Objekt **3240_h:03_h** gesetzt wurde.
- **3240_h:05_h** (Raw Value): Dieses Objekt beinhaltet den unmodifizierten Eingabewert.
- **60FD_h** (Digital Inputs): Dieses Objekt enthält eine Zusammenfassung der Eingänge und der Spezialfunktionen.

Verrechnung der Eingänge

Verrechnung des Eingangssignals am Beispiel von Eingang 1:

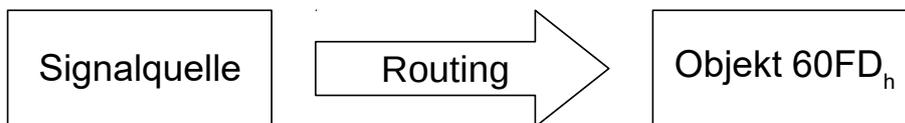
Der Wert an Bit 0 des Objekts **60FD_h** wird von der Firmware als negativer Endschalter interpretiert, das Ergebnis der vollständigen Verrechnung wird in Bit 16 abgelegt.



Input Routing

Prinzip

Um die Zuordnung der Eingänge flexibler vornehmen zu können, existiert der sogenannte *Input Routing Modus*. Dieser weist ein Signal einer Quelle auf ein Bit in dem Objekt 60FD_h zu.



Aktivierung

Dieser Modus wird aktiviert, indem das Objekt 3240_h:08_h (Routing Enable) auf 1 gesetzt wird.



Hinweis

Die Einträge 3240_h:01_h bis 3240_h:04_h haben dann **keine** Funktion mehr, bis das Eingangsrouting wieder abgeschaltet wird.



Hinweis

Wird das *Input Routing* eingeschaltet, werden initial die Werte des 3242_h geändert und entsprechen der Funktion der Inputs, wie diese vor der Aktivierung des *Input Routing* war. Die Eingänge der Steuerung verhalten sich mit der Aktivierung des *Input Routing* gleich. Es sollte daher nicht zwischen dem normalen Modus und dem *Input Routing* hin- und her geschaltet werden.

Routing

Das Objekt 3242_h bestimmt, welche Signalquelle auf welches Bit des 60FD_h geroutet wird. Der Subindex 01_h des 3242_h bestimmt Bit 0, Subindex 02_h das Bit 1, und so weiter. Die Signalquellen und deren Nummern finden Sie in den nachfolgenden Listen.

Nummer		
dec	hex	Signalquelle
00	00	Signal ist immer 0
01	01	Physikalischer Eingang 1
02	02	Physikalischer Eingang 2
03	03	Physikalischer Eingang 3
04	04	Physikalischer Eingang 4
05	05	Physikalischer Eingang 5
06	06	Physikalischer Eingang 6
07	07	Physikalischer Eingang 7
08	08	Physikalischer Eingang 8
09	09	Physikalischer Eingang 9
10	0A	Physikalischer Eingang 10
11	0B	Physikalischer Eingang 11
12	0C	Physikalischer Eingang 12
13	0D	Physikalischer Eingang 13
14	0E	Physikalischer Eingang 14
15	0F	Physikalischer Eingang 15
16	10	Physikalischer Eingang 16
65	41	Hall Eingang "U"
66	42	Hall Eingang "V"
67	43	Hall Eingang "W"
68	44	Encoder Eingang "A"
69	45	Encoder Eingang "B"
70	46	Encoder Eingang "Index"
71	47	USB Power Signal

Die nachfolgende Tabelle beschreibt die invertierten Signale der vorherigen Tabelle.

Nummer		
dec	hex	Signalquelle
128	80	Signal ist immer 1
129	81	Invertierter physikalischer Eingang 1
130	82	Invertierter physikalischer Eingang 2
131	83	Invertierter physikalischer Eingang 3
132	84	Invertierter physikalischer Eingang 4
133	85	Invertierter physikalischer Eingang 5
134	86	Invertierter physikalischer Eingang 6
135	87	Invertierter physikalischer Eingang 7
136	88	Invertierter physikalischer Eingang 8
137	89	Invertierter physikalischer Eingang 9
138	8A	Invertierter physikalischer Eingang 10
139	8B	Invertierter physikalischer Eingang 11
140	8C	Invertierter physikalischer Eingang 12
141	8D	Invertierter physikalischer Eingang 13
142	8E	Invertierter physikalischer Eingang 14

Nummer		
dec	hex	Signalquelle
143	8F	Invertierter physikalischer Eingang 15
144	90	Invertierter physikalischer Eingang 16
193	C1	Invertierter Hall Eingang "U"
194	C2	Invertierter Hall Eingang "V"
195	C3	Invertierter Hall Eingang "W"
196	C4	Invertierter Encoder Eingang "A"
197	C5	Invertierter Encoder Eingang "B"
198	C6	Invertierter Encoder Eingang "Index"
199	C7	Invertiertes USB Power Signal

Beispiel

Es soll der Eingang 1 auf Bit 16 des Objekts **60FD_h** geroutet werden:

Die Nummer der Signalquelle für Eingang 1 ist die "1". Das Routing für Bit 16 wird in das 3242_h:11_h geschrieben.

Demnach muss das Objekt 3242_h:11_h auf den Wert "1" gesetzt werden.

7.1.3 Digitale Ausgänge

Ausgänge

Die Ausgänge werden über das Objekt **60FE_h** gesteuert. Dabei entspricht Ausgang 1 dem Bit 16 im Objekt **60FE_h**, Ausgang 2 dem Bit 17 usw. wie bei den Eingängen. Die Ausgänge mit Sonderfunktionen sind in der Firmware wieder in den unteren Bits 0 bis 15 eingetragen. Im Moment ist nur Bit 0 belegt, das die Motorbremse steuert.

Beschaltung



Hinweis

Beachten Sie immer die maximale Belastbarkeit des Ausgangs (siehe **Anschlussbelegung**).

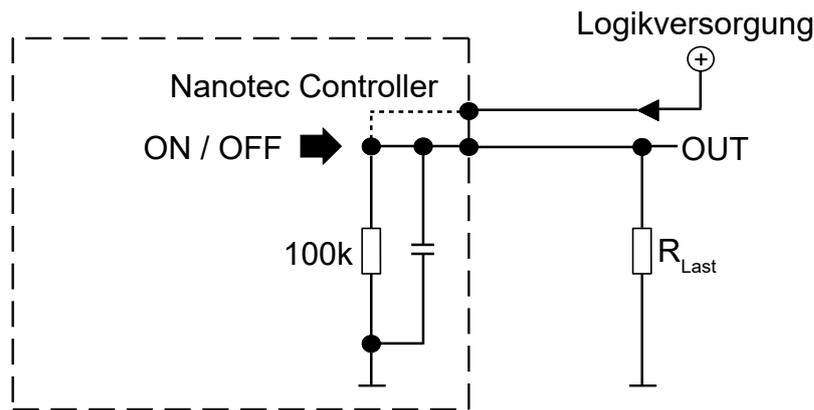
Die digitalen Ausgänge sind positiv schaltend (*High-Side-Switch*).



Hinweis

Um die digitalen Ausgänge benutzen zu können, müssen Sie eine Spannung (12 ...30 V DC) an den Pin 2 von **X6** anschließen (Logikversorgung).

Die Ausgangsspannung entspricht dieser Spannung. Der Strom soll 100 mA nicht überschreiten.



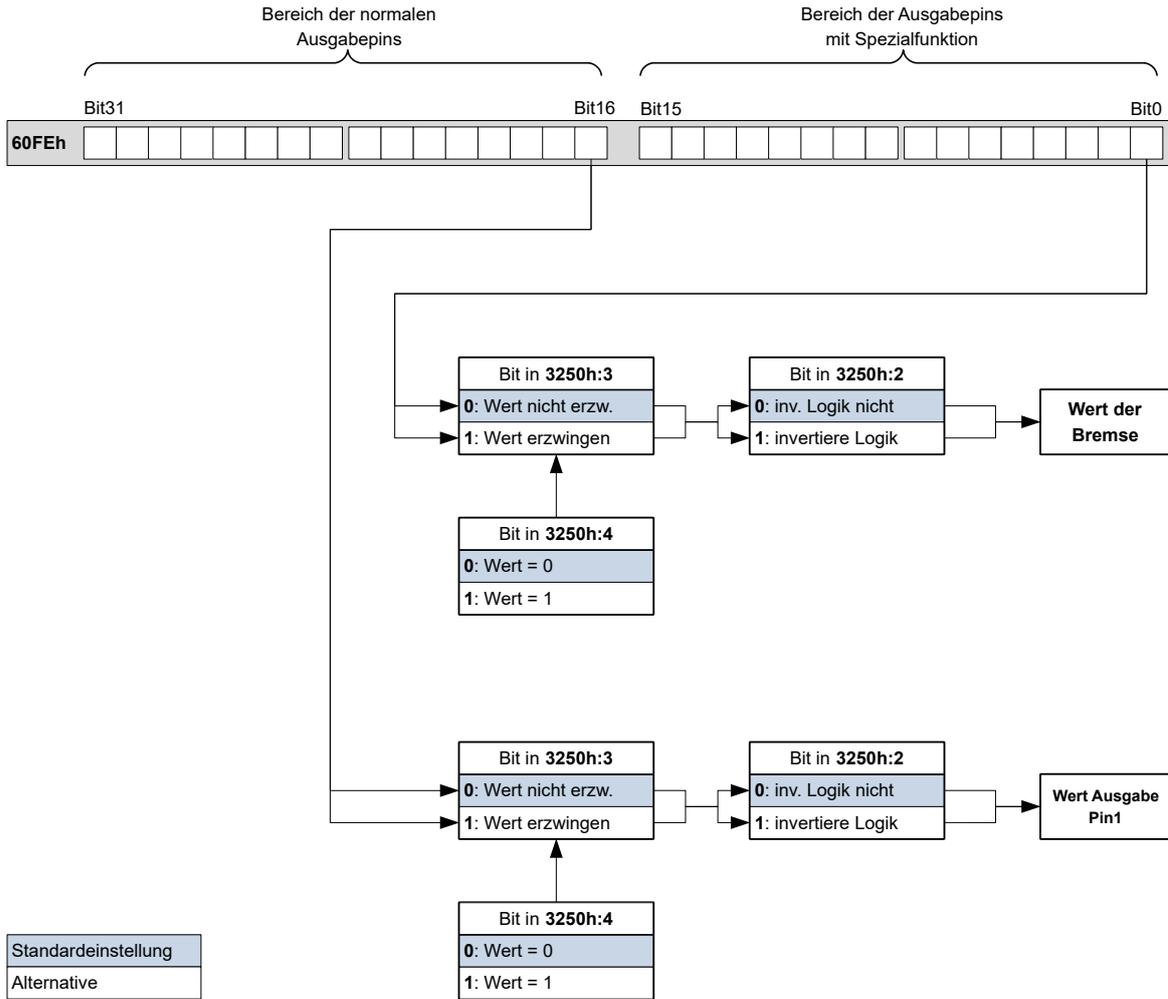
Objekteinträge

Es existieren zusätzliche OD-Einträge, um den Wert der Ausgänge zu manipulieren (siehe dazu das nachfolgende Beispiel). Ähnlich wie bei den Eingängen wirkt immer nur das Bit an der entsprechenden Stelle auf den jeweiligen Ausgang:

- **3250_h:01_h**: Keine Funktion.
- **3250_h:02_h**: Damit lässt sich die Logik von *Schließer* auf *Öffner* umstellen. Als *Schließer* konfiguriert, gibt der Ausgang einen logischen High-Pegel ab, sollte das Bit "1" sein. Bei der *Öffner*-Konfiguration wird bei einer "1" im Objekt **60FE_h** entsprechend ein logischer Low-Pegel ausgegeben.
- **3250_h:03_h**: Ist hier ein Bit gesetzt, wird der Ausgang manuell gesteuert. Der Wert für den Ausgang steht dann in Objekt **3250_h:4_h**, dies ist auch für den Bremsenausgang möglich.
- **3250_h:04_h**: Die Bits in diesem Objekt geben den Ausgabewert vor, welcher am Ausgang angelegt sein soll, wenn die manuelle Steuerung des Ausgangs über das Objekt **3250_h:03_h** aktiviert ist.
- **3250_h:05_h**: Dieses Objekt besitzt keine Funktion und ist aus Gründen der Kompatibilität enthalten.

Verrechnung der Ausgänge

Beispiel für die Verrechnung der Bits für die Ausgänge:

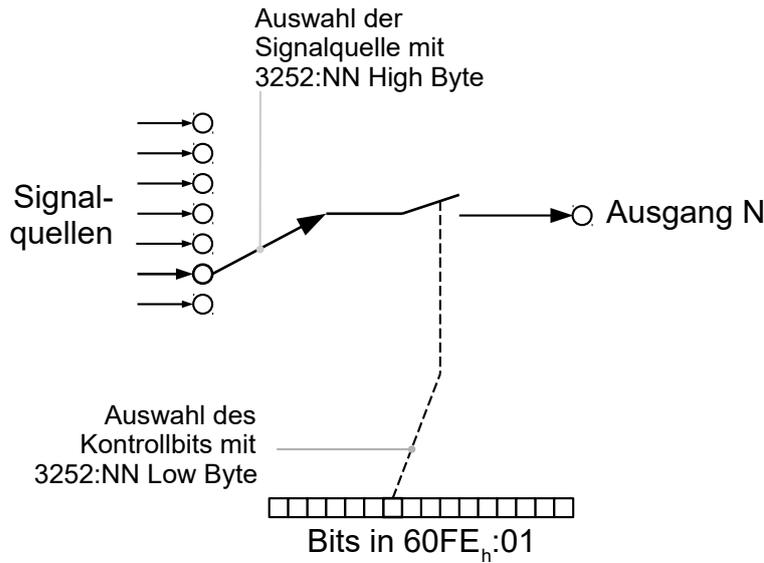


Output Routing

Prinzip

Der "Output Routing Mode" weist einem Ausgang eine Signalquelle zu, ein Kontrollbit im Objekt **60FE_h:01_h** schaltet das Signal ein oder aus.

Die Auswahl der Quelle wird mit **3252_h:01** bis **05** im "High Byte" (Bit 15 bis Bit 8) gemacht. Die Zuordnung eines Kontrollbit aus dem Objekt **60FE_h:01_h** erfolgt im "Low Byte" (Bit 7 bis Bit 0) des **3252_h:01_h** bis **05** (siehe nachfolgende Abbildung).



Aktivierung

Dieser Modus wird aktiviert, indem das Objekt **3250_h:08_h** (Routing Enable) auf 1 gesetzt wird.



Hinweis

Die Einträge **3250_h:01_h** bis **3250_h:04_h** haben dann **keine** Funktion mehr, bis das "Ausgangsrouting" wieder abgeschaltet wird.

Routing

Der Subindex des Objekts **3252_h** bestimmt, welche Signalquelle auf welchen Ausgang geroutet wird. Die Zuordnung der Ausgänge ist nachfolgend gelistet:

Subindex 3252 _h	Output Pin
01 _h	Konfiguration des PWM-Ausgangs (falls verfügbar)
02 _h	Konfiguration des Ausgangs 1
03 _h	Konfiguration des Ausgangs 2 (falls verfügbar)
04 _h	Konfiguration des Ausgangs 3 (falls verfügbar)
05 _h	Konfiguration des Ausgangs 4 (falls verfügbar)



Hinweis

Die maximale Ausgangsfrequenz des Ausgangs 1 und Ausgangs 2 ist 10kHz, des PWM-Ausgangs 2 kHz. Alle anderen Ausgänge können nur bis zu 500Hz Signale erzeugen.

Die Subindizes **3252_h:01_h** bis **05_h** sind 16 Bit breit, wobei das High Byte die Signalquelle auswählt (z.B. den PWM-Generator) und das Low Byte bestimmt das Kontrollbit im Objekt **60FE_h:01**.

Bit 7 von **3252_h:01_h** bis **05** invertiert die Steuerung aus dem Objekt **60FE_h:01**. Normalerweise schaltet der Wert "1" im Objekt **60FE_h:01** das Signal "ein", ist das Bit 7 gesetzt, schaltet der Wert "0" das Signal ein.

Nummer in 3252:01 bis 05

00XX _h	Ausgang ist immer "1"
01XX _h	Ausgang ist immer "0"
02XX _h	Encodersignal (6063 _h) mit Frequenzteiler 1
03XX _h	Encodersignal (6063 _h) mit Frequenzteiler 2
04XX _h	Encodersignal (6063 _h) mit Frequenzteiler 4
05XX _h	Encodersignal (6063 _h) mit Frequenzteiler 8
06XX _h	Encodersignal (6063 _h) mit Frequenzteiler 16
07XX _h	Encodersignal (6063 _h) mit Frequenzteiler 32
08XX _h	Encodersignal (6063 _h) mit Frequenzteiler 64
09XX _h	Position Actual Value (6064 _h) mit Frequenzteiler 1
0AXX _h	Position Actual Value (6064 _h) mit Frequenzteiler 2
0BXX _h	Position Actual Value (6064 _h) mit Frequenzteiler 4
0CXX _h	Position Actual Value (6064 _h) mit Frequenzteiler 8
0DXX _h	Position Actual Value (6064 _h) mit Frequenzteiler 16
0EXX _h	Position Actual Value (6064 _h) mit Frequenzteiler 32
0FXX _h	Position Actual Value (6064 _h) mit Frequenzteiler 64

Beispiel

Das Encodersignal (**6063**_h) soll auf Ausgang 1 mit einem Frequenzteiler 4 gelegt werden. Der Ausgang soll mit Bit 5 des Objektes **60FE**:01 gesteuert werden.

- **3250**_h:08_h = 1 (Routing aktivieren)
- **3252**_h:02_h = 0405_h (04XX_h + 0005_h) Dabei ist:
- 04XX_h: Encodersignal mit Frequenzteiler 4
- 0005_h: Auswahl von Bit 5 des **60FE**:01

Das Einschalten des Ausgangs wird mit dem Setzen des Bit 5 in Objekt **60FE**:01 erledigt.

7.2 I²t Motor-Überlastungsschutz

7.2.1 Beschreibung



Hinweis

Für Schrittmotoren wird nur der Nennstrom und kein Maximalstrom angegeben. Daher erfolgt die Nutzung von I²t mit Schrittmotoren ohne Gewähr.

Das Ziel des I²t Motor-Überlastungsschutz ist es, den Motor vor einem Schaden zu bewahren und gleichzeitig, ihn normal bis zu seinem thermischen Limit zu betreiben.

Diese Funktion ist nur verfügbar, wenn sich die Steuerung in der **Closed Loop-Betriebsart** befindet (Bit 0 des Objekts **3202**_h muss auf "1" gesetzt sein).

7.2.2 Objekteinträge

Folgende Objekte haben Einfluss auf den I²t Motor-Überlastungsschutz:

- **2031**_h: Max Motor Current - Gibt den maximal zulässigen Motorstrom in mA an.
- **203B**_h:1_h Motor Rated Current - Gibt den Nennstrom in mA an.

- **6073_h** Max Current - Gibt den Maximalstrom in Promille des eingestellten Nennstroms an.
- **203B_h:2_h** Maximum Duration Of Peak Current - Gibt die maximale Dauer des Maximalstroms in ms an.

Folgende Objekte zeigen den gegenwärtigen Zustand von I^2t an:

- **203B_h:3_h** Threshold - Gibt die Grenze in mAs an, von der abhängt, ob auf Maximalstrom oder Nennstrom geschaltet wird.
- **203B_h:4_h** CalcValue - Gibt den berechneten Wert an, welcher mit Threshold verglichen wird, um den Strom einzustellen.
- **203B_h:5_h** LimitedCurrent - Zeigt den gegenwärtigen Stromwert an, der von I^2t eingestellt wurde.
- **203B_h:6_h** Status:
 - Wert = "0": I^2t deaktiviert
 - Wert = "1": I^2t aktiviert

7.2.3 Aktivierung

Der *Closed Loop* muss aktiviert sein (Bit 0 des Objekts **3202_h** auf "1" gesetzt, siehe auch Kapitel **Closed Loop**).

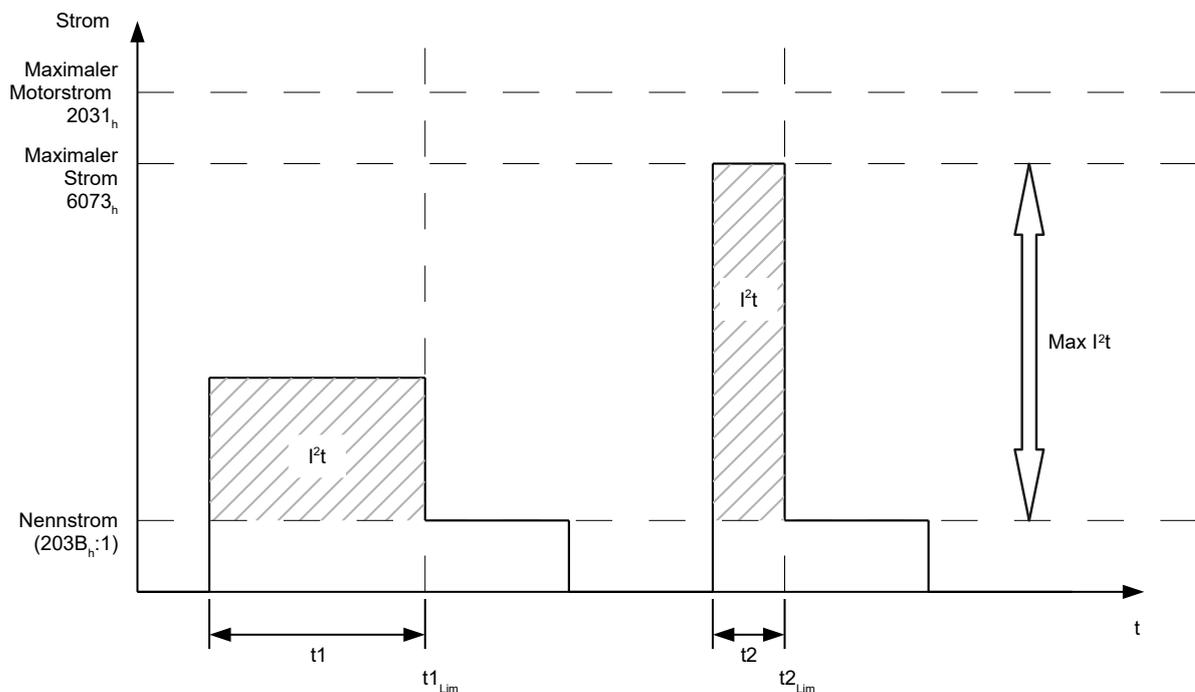
Zum Aktivieren des Modus müssen Sie die vier oben genannten Objekteinträge (**2031_h**, **6073_h**, **203B_h:1_h**, **203B_h:2_h**) sinnvoll beschreiben. Das bedeutet, dass der Maximalstrom größer als der Nennstrom sein muss und ein Zeitwert für die maximale Dauer des Maximalstroms eingetragen sein muss. Wenn diese Bedingungen nicht erfüllt sind, bleibt die I^2t Funktionalität deaktiviert.

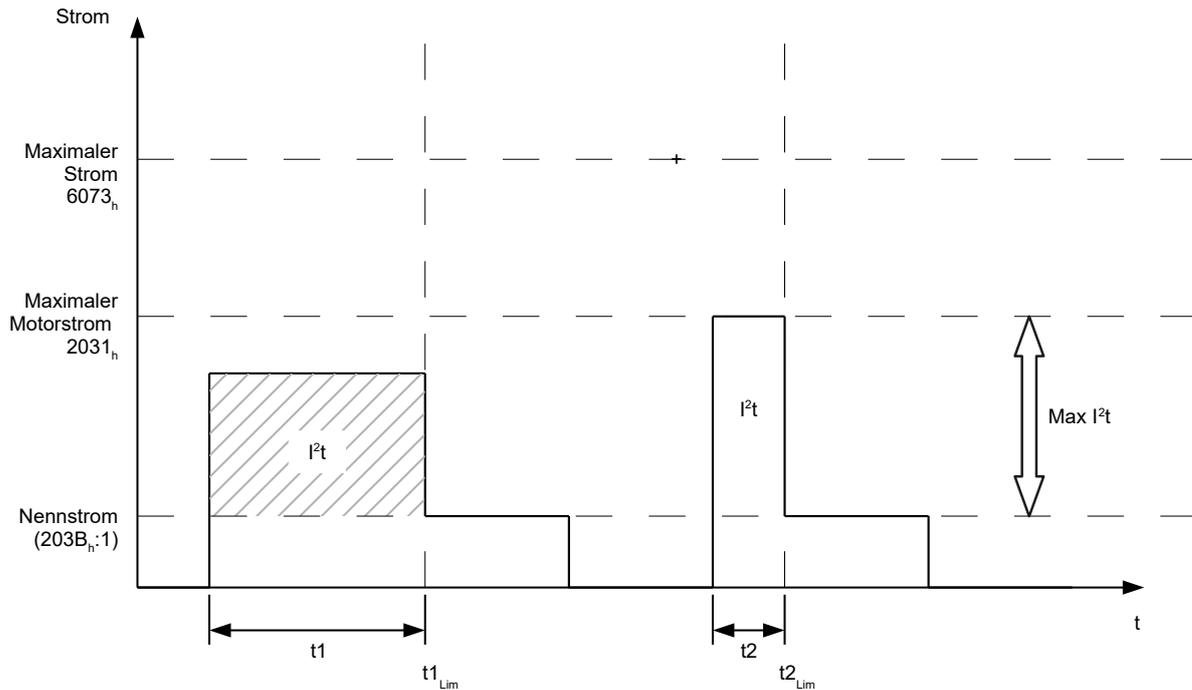
7.2.4 Funktion von I^2t

Durch die Angabe von Nennstrom, Maximalstrom und maximaler Dauer des Maximalstromes wird ein I^2T_{Lim} berechnet.

Der Motor kann solange mit Maximalstrom laufen, bis das berechnete I^2T_{Lim} erreicht wird. Darauf folgend wird der Strom sofort auf Nennstrom gesenkt. Der Maximalstrom wird durch den maximalen Motorstrom (**2031_h**) begrenzt.

In den folgenden Diagrammen sind die Zusammenhänge noch einmal dargestellt.





Im ersten Abschnitt t_1 ist der Stromwert höher als der Nennstrom. Am Zeitpunkt t_{1_Lim} wird $I^2_{t_Lim}$ erreicht und der Strom wird auf Nennstrom begrenzt. Danach kommt während der Dauer t_2 ein Strom, der dem Maximalstrom entspricht. Dementsprechend ist der Wert für $I^2_{t_Lim}$ schneller erreicht, als im Zeitraum t_1 .

7.3 Objekte speichern



Hinweis

Die unsachgemäße Anwendung dieser Funktion kann dazu führen, dass die Steuerung sich nicht mehr starten lässt. Lesen Sie daher vor der Benutzung der Funktion das Kapitel vollständig durch.



Hinweis

Als eine Alternative lassen sich Objekte auch über die Konfigurationsdatei setzen und speichern. Zu beachten ist, dass diese Datei die höhere Priorität hat. Objekte, welche sowohl mit dem hier beschriebenen Mechanismus gespeichert, als auch in der Konfigurationsdatei gespeichert werden, werden den Wert der Konfigurationsdatei annehmen.

7.3.1 Allgemeines

Viele Objekte im Objektverzeichnis lassen sich speichern und werden beim nächsten Einschalten/Reset automatisch wieder geladen. Zudem bleiben die gespeicherten Werte auch bei einem Firmware-Update erhalten.

Es lassen sich immer nur ganze Sammlungen von Objekten (im Folgenden *Kategorien* genannt) zusammen abspeichern, einzelne Objekte können nicht gespeichert werden.

Ein Objekt kann einer der folgenden *Kategorien* zugeordnet sein:

- Kommunikation: Parameter mit Bezug auf externe Schnittstellen, wie PDO-Konfiguration etc.
- Applikation: Parameter mit Bezug auf Betriebsmodi.
- Benutzer: Parameter, die ausschließlich vom Kunden/Benutzer geschrieben und gelesen, und von der Steuerungsfirmware ignoriert werden.

- Bewegung: Parameter mit Bezug auf den Motor und die Sensoren (BLDC/Stepper, *Closed/Open Loop*...). Einige werden vom Auto-Setup gesetzt und gespeichert.
- Tuning: Parameter mit Bezug auf Motor und Encoder, die entweder vom Auto-Setup gesetzt werden, oder den Datenblättern entnommen werden können, zum Beispiel Polpaare und Maximum Current.
- CANopen: Parameter mit Bezug auf die CANopen-Kommunikation
- Modbus RTU: Parameter mit Bezug auf die Modbus RTU-Kommunikation

Wenn ein Objekt keiner dieser *Kategorien* zugeordnet ist, kann es nicht gespeichert werden, zum Beispiel Statusword und alle Objekte, deren Wert abhängig vom aktuellen Zustand der Steuerung ist.

Die Objekte in jeder *Kategorie* werden unten aufgelistet. Im Kapitel **Objektverzeichnis Beschreibung** wird ebenfalls für jedes Objekt die zugehörige *Kategorie* angegeben.

7.3.2 Kategorie: Kommunikation

- **1005_h**: COB-ID Sync
- **1007_h**: Synchronous Window Length
- **100C_h**: Guard Time
- **100D_h**: Live Time Factor
- **1014_h**: COB-ID EMCY
- **1016_h**: Consumer Heartbeat Time
- **1017_h**: Producer Heartbeat Time
- **1019_h**: Synchronous Counter Overflow Value
- **1029_h**: Error Behavior
- **1400_h**: Receive PDO 1 Communication Parameter
- **1401_h**: Receive PDO 2 Communication Parameter
- **1402_h**: Receive PDO 3 Communication Parameter
- **1403_h**: Receive PDO 4 Communication Parameter
- **1404_h**: Receive PDO 5 Communication Parameter
- **1405_h**: Receive PDO 6 Communication Parameter
- **1406_h**: Receive PDO 7 Communication Parameter
- **1407_h**: Receive PDO 8 Communication Parameter
- **1600_h**: Receive PDO 1 Mapping Parameter
- **1601_h**: Receive PDO 2 Mapping Parameter
- **1602_h**: Receive PDO 3 Mapping Parameter
- **1603_h**: Receive PDO 4 Mapping Parameter
- **1604_h**: Receive PDO 5 Mapping Parameter
- **1605_h**: Receive PDO 6 Mapping Parameter
- **1606_h**: Receive PDO 7 Mapping Parameter
- **1607_h**: Receive PDO 8 Mapping Parameter
- **1800_h**: Transmit PDO 1 Communication Parameter
- **1801_h**: Transmit PDO 2 Communication Parameter
- **1802_h**: Transmit PDO 3 Communication Parameter
- **1803_h**: Transmit PDO 4 Communication Parameter
- **1804_h**: Transmit PDO 5 Communication Parameter
- **1805_h**: Transmit PDO 6 Communication Parameter
- **1806_h**: Transmit PDO 7 Communication Parameter
- **1807_h**: Transmit PDO 8 Communication Parameter
- **1A00_h**: Transmit PDO 1 Mapping Parameter
- **1A01_h**: Transmit PDO 2 Mapping Parameter
- **1A02_h**: Transmit PDO 3 Mapping Parameter
- **1A03_h**: Transmit PDO 4 Mapping Parameter
- **1A04_h**: Transmit PDO 5 Mapping Parameter
- **1A05_h**: Transmit PDO 6 Mapping Parameter

- **1A06_h**: Transmit PDO 7 Mapping Parameter
- **1A07_h**: Transmit PDO 8 Mapping Parameter
- **1F80_h**: NMT Startup
- **2102_h**: Fieldbus Module Control
- **3502_h**: MODBUS Rx PDO Mapping
- **3602_h**: MODBUS Tx PDO Mapping

7.3.3 Kategorie: Applikation

- **2034_h**: Upper Voltage Warning Level
- **2035_h**: Lower Voltage Warning Level
- **2036_h**: Open Loop Current Reduction Idle Time
- **2037_h**: Open Loop Current Reduction Value/factor
- **2038_h**: Brake Controller Timing
- **203A_h**: Homing On Block Configuration
- **203D_h**: Torque Window
- **203E_h**: Torque Window Time Out
- **203F_h**: Max Slippage Time Out
- **2057_h**: Clock Direction Multiplier
- **2058_h**: Clock Direction Divider
- **205B_h**: Clock Direction Or Clockwise/Counter Clockwise Mode
- **2084_h**: Bootup Delay
- **2290_h**: PDI Control
- **2291_h**: PDI Input
- **2300_h**: NanoJ Control
- **2410_h**: NanoJ Init Parameters
- **2800_h**: Bootloader And Reboot Settings
- **3210_h**: Motor Drive Parameter Set
- **3212_h**: Motor Drive Flags
- **3221_h**: Analogue Inputs Control
- **3240_h**: Digital Inputs Control
- **3242_h**: Digital Input Routing
- **3243_h**: Digital Input Homing Capture
- **3250_h**: Digital Outputs Control
- **3252_h**: Digital Output Routing
- **3321_h**: Analogue Input Offset
- **3322_h**: Analogue Input Pre-scaling
- **3700_h**: Deviation Error Option Code
- **3701_h**: Limit Switch Error Option Code
- **4013_h**: HW Configuration
- **6040_h**: Controlword
- **6042_h**: VI Target Velocity
- **6046_h**: VI Velocity Min Max Amount
- **6048_h**: VI Velocity Acceleration
- **6049_h**: VI Velocity Deceleration
- **604A_h**: VI Velocity Quick Stop
- **604C_h**: VI Dimension Factor
- **605A_h**: Quick Stop Option Code
- **605B_h**: Shutdown Option Code
- **605C_h**: Disable Option Code
- **605D_h**: Halt Option Code
- **605E_h**: Fault Option Code
- **6060_h**: Modes Of Operation

- **6065_h**: Following Error Window
- **6066_h**: Following Error Time Out
- **6067_h**: Position Window
- **6068_h**: Position Window Time
- **606D_h**: Velocity Window
- **606E_h**: Velocity Window Time
- **6071_h**: Target Torque
- **6072_h**: Max Torque
- **607A_h**: Target Position
- **607B_h**: Position Range Limit
- **607C_h**: Home Offset
- **607D_h**: Software Position Limit
- **607E_h**: Polarity
- **607F_h**: Max Profile Velocity
- **6081_h**: Profile Velocity
- **6082_h**: End Velocity
- **6083_h**: Profile Acceleration
- **6084_h**: Profile Deceleration
- **6085_h**: Quick Stop Deceleration
- **6086_h**: Motion Profile Type
- **6087_h**: Torque Slope
- **6091_h**: Gear Ratio
- **6092_h**: Feed Constant
- **6096_h**: Velocity Factor
- **6097_h**: Acceleration Factor
- **6098_h**: Homing Method
- **6099_h**: Homing Speed
- **609A_h**: Homing Acceleration
- **60A2_h**: Jerk Factor
- **60A4_h**: Profile Jerk
- **60A8_h**: SI Unit Position
- **60A9_h**: SI Unit Velocity
- **60B0_h**: Position Offset
- **60B1_h**: Velocity Offset
- **60B2_h**: Torque Offset
- **60C1_h**: Interpolation Data Record
- **60C2_h**: Interpolation Time Period
- **60C4_h**: Interpolation Data Configuration
- **60C5_h**: Max Acceleration
- **60C6_h**: Max Deceleration
- **60E8_h**: Additional Gear Ratio - Motor Shaft Revolutions
- **60E9_h**: Additional Feed Constant - Feed
- **60ED_h**: Additional Gear Ratio - Driving Shaft Revolutions
- **60EE_h**: Additional Feed Constant - Driving Shaft Revolutions
- **60F2_h**: Positioning Option Code
- **60F8_h**: Max Slippage
- **60FE_h**: Digital Outputs
- **60FF_h**: Target Velocity

7.3.4 Kategorie: Bewegung

- **3202_h**: Motor Drive Submode Select
- **3203_h**: Feedback Selection

- **6073_h**: Max Current
- **6080_h**: Max Motor Speed

7.3.5 Kategorie: Tuning

- **2030_h**: Pole Pair Count
- **2031_h**: Max Motor Current
- **203B_h**: I2t Parameters
- **3390_h**: Feedback Hall
- **33A0_h**: Feedback Incremental A/B/I 1
- **6075_h**: Motor Rated Current
- **608F_h**: Position Encoder Resolution
- **6090_h**: Velocity Encoder Resolution
- **60E6_h**: Additional Position Encoder Resolution - Encoder Increments
- **60EB_h**: Additional Position Encoder Resolution - Motor Revolutions

7.3.6 Kategorie: CANopen

- **2005_h**: CANopen Baudrate
- **2007_h**: CANopen Config
- **2009_h**: CANopen NodeID

7.3.7 Kategorie: Modbus RTU

- **2028_h**: MODBUS Slave Address
- **202A_h**: MODBUS RTU Baudrate
- **202D_h**: MODBUS RTU Parity

7.3.8 Speichervorgang starten



VORSICHT

Unkontrollierte Motorbewegungen!

Während des Speicherns kann die Regelung beeinträchtigt werden. Es kann zu unvorhersehbaren Reaktionen kommen.

► Bevor Sie den Speichervorgang starten, muss der Motor sich im Stillstand befinden. Der Motor darf während des Speicherns nicht angefahren werden.



Hinweis

- Das Speichern kann einige Sekunden dauern. Unterbrechen Sie während des Speicherns keinesfalls die Spannungsversorgung. Andernfalls ist der Stand der gespeicherten Objekte undefiniert.
- Warten Sie immer, dass die Steuerung das erfolgreiche Speichern mit dem Wert "1" in dem entsprechenden Subindex im Objekt **1010_h** signalisiert.

Für jede *Kategorie* gibt es einen Subindex im Objekt **1010_h**. Um alle Objekte dieser *Kategorie* zu speichern, muss der Wert "65766173_h"¹ in den Subindex geschrieben werden. Das Ende des Speichervorgangs wird signalisiert, indem der Wert von der Steuerung durch eine "1" überschrieben wird.

¹ Das entspricht dezimal der 1702257011_d bzw. dem ASCII String `save`.

Nachfolgende Tabelle listet auf, welcher Subindex des Objektes **1010_h** für welche *Kategorie* zuständig ist.

Subindex	Kategorie
01 _h	Alle Kategorien mit der Ausnahme von 06 _h (Tuning) , 0A _h (CANopen) und 0B _h (Modbus RTU)
02 _h	Kommunikation
03 _h	Applikation
04 _h	Benutzer
05 _h	Bewegung
06 _h	Tuning
0A _h	CANopen
0B _h	Modbus RTU

7.3.9 Speicherung verwerfen

Falls alle Objekte oder eine *Kategorie* an gespeicherten Objekten gelöscht werden sollen, muss in das Objekt **1011_h** der Wert "64616F6C_h"² geschrieben werden. Folgende Subindizes entsprechen dabei einer *Kategorie*:

Subindex	Kategorie
01 _h	Alle Kategorien (Zurücksetzen auf Werkseinstellung) mit der Ausnahme von 06 _h (Tuning) , 0A _h (CANopen) und 0B _h (Modbus RTU)
02 _h	Kommunikation
03 _h	Applikation
04 _h	Benutzer
05 _h	Bewegung
06 _h	Tuning
0A _h	CANopen
0B _h	Modbus RTU

Die gespeicherten Objekte werden daraufhin verworfen, die Änderung wirkt erst nach einem Neustart der Steuerung aus. Sie können sie Steuerung neu starten, indem Sie den Wert "746F6F62_h" in **2800_h:01_h** eintragen.



Hinweis

- Die Objekte der *Kategorie* 06_h (Tuning) werden vom **Auto-Setup** ermittelt und werden beim Zurücksetzen auf Werkseinstellungen mittels Subindex 01_h nicht zurückgesetzt (damit eine erneutes Auto-Setup nicht notwendig wird). Sie können diese Objekte mit Subindex 06_h zurücksetzen.
- Die Objekte der *Kategorien* 0A_h (CANopen) und 0B_h (Modbus RTU) werden mittels Subindex 01_h nicht zurückgesetzt.

² Das entspricht dezimal der 1684107116_d bzw. dem ASCII String load.

7.3.10 Konfiguration verifizieren

Das Objekt **1020_h** kann herangezogen werden, um die Konfiguration zu verifizieren. Es agiert wie ein Modifikationsmarker in üblichen Text-Editoren: Sobald eine Datei in dem Editor modifiziert wird, wird ein Marker (normalerweise ein Stern) hinzugefügt.

Die Einträge des Objektes **1020_h** können mit einem Datum und einer Uhrzeit beschrieben und anschließend über **1010_h:01** zusammen mit allen anderen speicherbaren Objekten gespeichert werden.

Die Einträge von **1020_h** werden auf "0" zurückgesetzt, sobald ein beliebiges speicherbares Objekt (einschließlich **1010_h:0x_h**, außer **1010_h:01_h** und **1020_h**) beschrieben wird.

Die folgende Reihenfolge macht die Verifikation möglich:

1. Ein externes Tool oder Master konfiguriert die Steuerung.
2. Das Tool oder der Master setzt den Wert in das Objekt **1020_h**.
3. Das Tool oder der Master aktiviert das Speichern aller Objekte **1010_h:01_h = 65766173_h**. Das Datum und die Uhrzeit im Objekt **1020_h** werden ebenfalls abgespeichert.

Nach einem Neustart der Steuerung kann der Master den Wert in **1020_h:01_h** und **1020:01_h** prüfen. Sollte einer der Werte "0" sein, wurde das Objektverzeichnis verändert, nachdem die gespeicherten Werte geladen wurden. Sollten das Datum oder die Uhrzeit in **1020** nicht den erwarteten Werten entsprechen, wurden Objekte wahrscheinlich mit anderen als den erwarteten Werten gespeichert.

8 CANopen

Die Steuerung lässt sich mittels CANopen ansprechen und kann in einem Netzwerk als *Slave* arbeiten. In diesem Kapitel werden die Dienste der CANopen-Kommunikationsstruktur beschrieben. Zudem werden die Nachrichten für CANopen im Einzelnen aufgeschlüsselt.

CANopen-Referenzen: www.can-cia.org

- *CiA 301 CANopen application layer and communication profile - Application layer and communication profile*, Date: 21.02.2011, Version: 4.2.0
- *CiA 402 Device profile for drives and motion control - Part 1: General definitions*, Date: 14.12.2007, Version: 3.0.0
- *CiA 402 Drives and motion control device profile - Part 2: Operation modes and application data*, Date: 14.12.2007, Version: 3.0
- *CiA 402 Drives and motion control device profile - Part 3: PDO mapping*, Date: 14.12.2007, Version: 3.0
- *CiA 306 Electronic device description - Part 1: Electronic Data Sheet and Device Configuration File*, Date: 08.02.2012, Version: 1.3.5
- *CiA 305 Layer setting services (LSS) and protocols*, Date: 08.05.2013, Version: 3.0.0

8.1 Allgemeines



Tipp

- Unterstützt werden zurzeit nur 11-Bit CAN-ID.
- Bei CANopen werden die Daten immer in Little Endian-Format über den Bus geschickt.

8.1.1 CAN-Nachricht

In diesem Kapitel werden oft CAN-Nachrichten beschrieben, diese werden wie folgt notiert:

583 | 41 09 10 00 1E 00 00 00

183R | DLC=0

Dabei gilt folgende Konvention:

- Alle Zahlen werden in hexadezimaler Schreibweise notiert, auf das einleitende 0x wird wegen der verkürzten Schreibweise verzichtet.
- Normale Datennachricht: Vorangestellt wird die CAN-ID der CAN-Nachricht, in dem obigen Beispiel die 583 (also 583_h bzw. 1411_d). Die Daten und die CAN-ID werden mit einem senkrechten Strich von den Daten getrennt.
- RTR-Nachricht (Remote transmission request): Folgt auf die CAN-ID ein R anstelle der Daten wird die Länge des DLC (Download Content) angegeben, in dem obigen Beispiel ist die Länge des DLC 0.

8.2 CANopen Dienste

Der CANopen-Stack bietet die in der nachfolgenden Tabelle abgedruckten Dienste (auch Services genannt) an, genauere Beschreibungen sind in den jeweiligen Kapiteln hinterlegt.

Default CAN-ID	Service	Beschreibung in
000 _h	Network Management (NMT)	Abschnitt Network Management (NMT)
080 _h	Synchronizing Object	Abschnitt Synchronisations-Objekt (SYNC)
080 _h +Node-ID	Emergency	Abschnitt Emergency Object (EMCY)
180 _h +Node-ID	TX Process Data Objects (PDO)	Abschnitt Process Data Object (PDO)

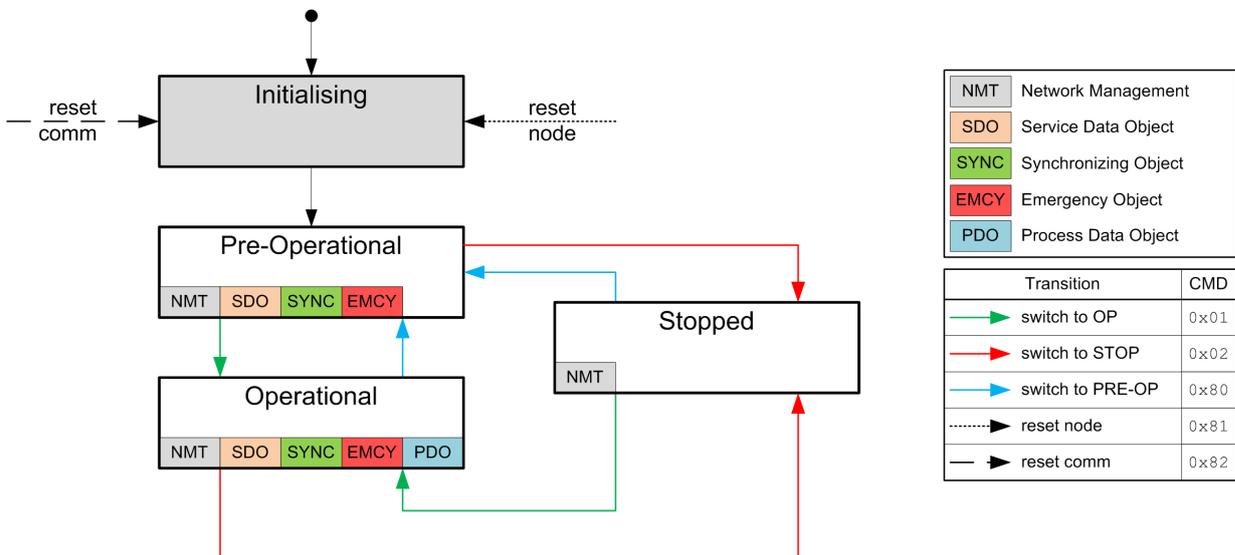
Default CAN-ID	Service	Beschreibung in
200 _h +Node-ID	RX Process Data Objects (PDO)	
280 _h +Node-ID	TX Process Data Objects (PDO)	
300 _h +Node-ID	RX Process Data Objects (PDO)	
380 _h +Node-ID	TX Process Data Objects (PDO)	
400 _h +Node-ID	RX Process Data Objects (PDO)	
480 _h +Node-ID	TX Process Data Objects (PDO)	
500 _h +Node-ID	RX Process Data Objects (PDO)	
580 _h +Node-ID	TX Service Data Objects (SDO)	Abschnitt Service Data Object (SDO)
600 _h +Node-ID	RX Service Data Objects (SDO)	
700 _h +Node-ID	BOOT-UP Protocol	Abschnitt Boot-Up Protocol
700 _h +Node-ID	Nodeguarding und Heartbeat	Abschnitt Heartbeat und Nodeguarding

8.2.1 Network Management (NMT)

Das Network Management ist CANopen-Geräte orientiert und folgt einer Master-Slave-Struktur. NMT benötigt ein CANopen-Gerät im Netzwerk, welches die Rolle des CANopen-Masters einnimmt. Alle anderen Geräte haben die Rolle des NMT-Slaves. Jeder NMT-Slave kann durch seine individuelle Node-ID im Bereich von [1..127] angesprochen werden. Durch NMT-Services können CANopen-Geräte initialisiert, gestartet, beobachtet, resettet oder gestoppt werden.

Dabei folgt die Steuerung dem Zustandsdiagramm aus der nachfolgenden Abbildung. Der Zustand "Initialization" wird nur nach dem Einschalten erreicht oder durch Senden eines NMT-Befehl "Reset Communication" oder "Reset Node". Der Zustand "Pre-Operational" wird nach der Initialisierung automatisch angesteuert.

Sie können im Objekt **1F80_h** einstellen, ob danach automatisch in den Zustand "Operational" gewechselt werden soll, damit Sie den Versand eines zusätzlichen NMT-Kommandos vermeiden.

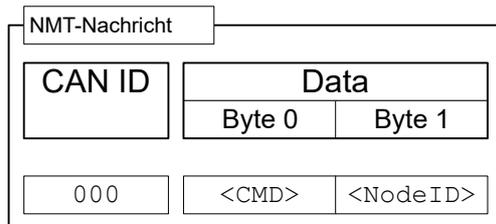


In der nachfolgenden Tabelle finden Sie eine Übersicht, welche die Aktivität der Services in den entsprechenden Zuständen darstellt. Zu beachten ist, dass der Zustand *Stopped* die Kommunikation gänzlich einstellt und nur noch die Steuerung der NMT-Zustandsmaschine zulässt.

Service	Initializing	Pre-Operational	Operational	Stopped
PDO			aktiv	
SDO		aktiv	aktiv	
SYNC		aktiv	aktiv	

Service	Initializing	Pre-Operational	Operational	Stopped
EMCY		aktiv	aktiv	
BOOT-UP	aktiv			
NMT		aktiv	aktiv	aktiv

Die "Network Management"-Nachricht hat die CAN-ID 0. Eine Nachricht ist immer zwei Bytes lang und hat folgenden Aufbau:



Das <CMD> entspricht dabei einem der folgenden Bytes (siehe auch Legende in der Abbildung des **NMT-Zustandsdiagramms**):

<CMD>	Bedeutung
01 _h	Schalte in den Zustand "Operational"
02 _h	Schalte in den Zustand "Stop"
80 _h	Schalte in den Zustand "Pre-Operational"
81 _h	Reset Node
82 _h	Reset Communication

Der Wert für <Node-ID> kann die 00_h sein, dann gilt der NMT-Befehl für alle Geräte am CAN-Bus (Broadcast). Wird eine Zahl ungleich Null verwendet, wird nur das Gerät mit der entsprechenden Node-ID adressiert.

Der Befehl "Reset Node" startet die Steuerung komplett neu, der Befehl "Reset Communication" setzt nur die Einstellungen von CANopen zurück und startet die Kommunikation neu.

Beispiel: Sollen alle Geräte am CAN-Bus in den Betriebszustand "Stop" gebracht werden, kann ein Broadcast mit dem Befehl "Schalte in den Zustand Stop" verwendet werden. Die NMT-Nachricht baut sich wie folgt auf:

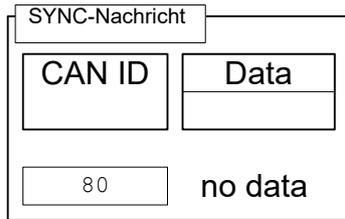
000 | 02 00

Soll nur das Gerät mit der Node-ID 42 vollständig neu gestartet werden, ist folgende CAN-Nachricht zu verschicken:

000 | 81 2A

8.2.2 Synchronisations-Objekt (SYNC)

Das Synchronisationsobjekt wird benutzt, um den Zeitpunkt von PDO-Daten für alle Geräte am Bus gleichzeitig gültig werden zu lassen. Die Sync-Nachricht baut sich folgendermaßen auf:

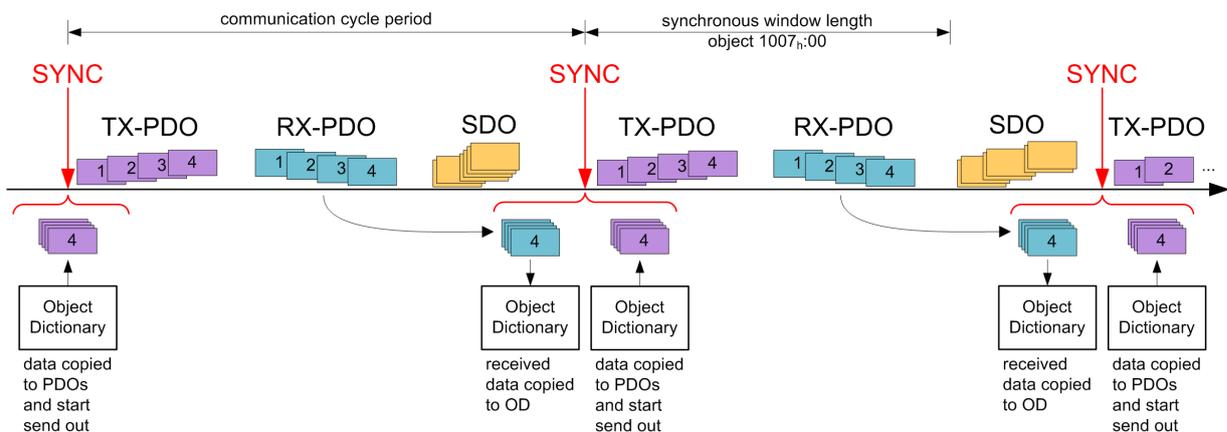


Für den SYNC-Betrieb wird normalerweise für die RX-PDOs der Übertragungsmodus (Transmission Type) 0 verwendet (Daten werden mit dem nächsten SYNC gültig), für die TX-PDOs wird ein Übertragungsmodus zwischen 1-240 gewählt. (Details: siehe Kapitel **Process Data Object (PDO)**).

Nach dem Erhalt einer SYNC-Nachricht gib es ein Zeitfenster ("synchronous window"), innerhalb dessen PDO-Nachrichten gesendet und empfangen werden dürfen, ist die Zeit des Fensters abgelaufen, müssen alle Geräte das Senden von PDOs einstellen. Die "synchronous window length" kann im Objekt **1007_h:00_h** in Millisekunden eingestellt werden.

Ein typischer CAN-SYNC-Betrieb gliedert sich in vier Phasen (siehe auch nachfolgende Abbildung):

1. Die SYNC-Nachricht wird empfangen. Damit werden die vorher empfangenen RX-PDO-Daten in das Objektverzeichnis kopiert (falls vorhanden). Zu dem Zeitpunkt werden auch die Daten gesammelt und in die TX-PDOs kopiert und das Senden dieser Nachrichten veranlasst.
2. Anschließend werden von allen Slaves am Bus die TX-PDOs verschickt.
3. Danach werden vom CANopen-Master die PDOs versendet. Nachdem die Zeit des "synchronous window length" abgelaufen ist, sind keine PDOs mehr zulässig.
4. Spätestens wenn das "synchronous window" wieder geschlossen ist, können SDO-Nachrichten ausgetauscht werden.



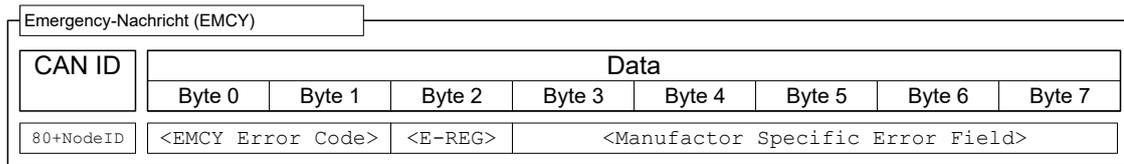
Falls der *Sync Producer* einen *Sync Counter* unterstützt, enthält die Sync-Nachricht einen zusätzlichen 1-Byte-Zählwert. Dieser Zähler erhöht sich um den Wert "1" pro gesendete Sync-Nachricht und wird jedes Mal zurückgesetzt, wenn er den Wert von **1019_h Synchronous Counter Overflow Value** erreicht.

Für jedes **TX-PDO** kann dann im Subindex 06_h des dazugehörigen Kommunikationsparameters (z.B. in **1800_h:06_h**) ein Startwert des *Sync Counter* festgelegt werden, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Die Funktion wird global erst aktiviert, wenn in **1019_h** ein Wert größer 1 eingestellt wird.

8.2.3 Emergency Object (EMCY)

Eine Nachricht des Types "Emergency" wird immer dann gesendet, wenn ein Fehler in der Steuerung auftritt, welcher nicht durch ein SDO-Zugriff verursacht wurde. Dieser Service ist unbestätigt und wird mit der CAN-ID 80_h+Node-ID verschickt.

Die Emergency-Nachricht hat den folgenden Aufbau:



Dabei werden insgesamt drei Fehlercodes übertragen, der "Emergency Error Code" (<EMCY Error Code>), der Inhalt des Objektes "Error Register" (1001_h , <E-REG>) und ein herstellerepezifischer Code (Manufacturer Specific Error)

Fehlerbehandlung

Ein Modul zur Fehlerbehandlung verarbeitet alle intern auftretenden Fehler. Jeder Fehler ist in eine Fehlerklasse eingeteilt.

Jeder auftretende Fehler wird folgendermaßen behandelt:

- Das zum Fehler gehörige Bit im Objekt "Error Register" (1001_h) wird gesetzt.
- Anschließend werden drei Informationen zusammen in das Objekt "Pre-defined Error Field" ($1003_h:01$) geschrieben:
 1. Der *Emergency Error Code*
 2. Das *Error Register*
 3. Der herstellerepezifische Fehlercode
- Steht kein weiterer Fehler mehr an, wird folgende Nachricht verschickt:
80 + Node-ID | 00 00 00 00 00 00 00 00

Im Objekt 1029_h können Sie konfigurieren, ob und wie die Steuerung im Fehlerfall ihren *NMT-Zustand* ändern soll.

8.2.4 Service Data Object (SDO)

Ein "Service Data Object" lässt einen lesenden oder schreibenden Zugriff auf das Objektverzeichnis zu.

Im Nachfolgenden wird der Besitzer des Objektverzeichnisses "Server" genannt, der CAN-Knoten - welcher die Daten anfordert oder schreiben will - "Client". Mit einem "Upload" wird das Lesen eines Wertes eines Objektes aus dem Objektverzeichnis bezeichnet, ein "Download" ist entsprechend das Schreiben eines Wertes in das Objektverzeichnis. Zudem werden folgende Kürzel in den Diagrammen benutzt:

- <IDX>: Index des zu lesenden oder schreibenden Objektes im Objektverzeichnis; das LSB des Indexes steht dabei im Byte 1. Beispiel: das Statusword der Steuerung hat den Index 6041_h , Byte 1 wird dann mit 41_h und Byte 2 mit 60_h beschrieben. Die SDO-Antwort enthält bei **Expedited Transfer** den gleichen Index, wie den der Anforderung.
- <SUBIDX>: Subindex des Objektes im Objektverzeichnis von 00_h bis FF_h . Die Antwort der SDO-Nachricht der Steuerung enthält bei **Expedited Transfer** ebenfalls den Subindex der Anforderung.

Da CAN-Nachrichten des Types SDO sehr viele Meta-Daten beinhalten, sollte mit SDO-Nachricht nur die Konfiguration der Steuerung vorgenommen werden. Sollte es notwendig sein, im laufenden Betrieb Daten zyklisch auszutauschen, ist es sinnvoller, auf CANopen-Nachricht des Types PDO zurück zu greifen (siehe Unterabschnitt **Process Data Object**).

Die SDO-Transfers unterteilen sich in drei Sorten des Zugriffs:

- "expedited transfer" für die Übertragung von einem Objekt mit bis zu vier Bytes.
- "normal transfer" für die Übertragung von beliebig vielen Bytes, wobei jede CAN-Nachricht einzeln bestätigt wird.
- "block transfer" ebenfalls für beliebig viele Bytes, dabei wird jeweils ein Block an CAN-Tickets auf einmal bestätigt.

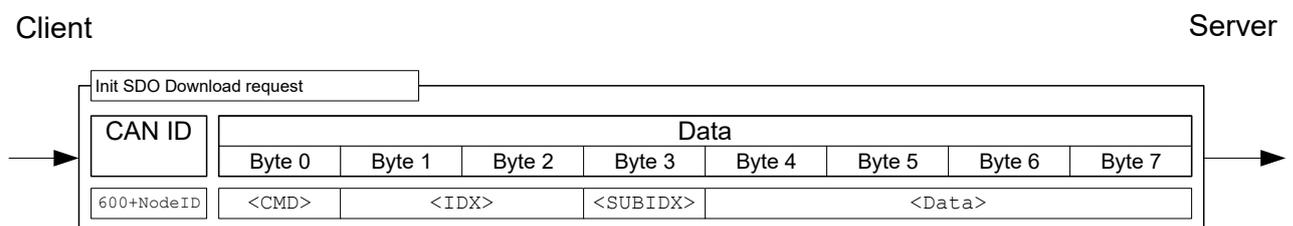
Eine SDO-Nachricht wird an die CAN-ID $600_h + \text{Node-ID}$ verschickt, die Antwort kommt mit der CAN-ID $580_h + \text{Node-ID}$.

Expedited Transfer

Mit dieser Methode lassen sich Werte in Objekte des Types (UN)SIGNED8, INTEGER16 oder INTEGER32 in das Objektverzeichnis schreiben (download) oder auslesen (upload). Dieser Service ist bestätigt, d.h. auf jeden Zugriff wird entweder mit Daten, einer Bestätigung oder mit einer Fehlermeldung geantwortet.

SDO Download

Eine Expedited-SDO-Nachricht zum Schreiben der Daten in das Objektverzeichnis des Servers ist wie folgt aufgebaut:

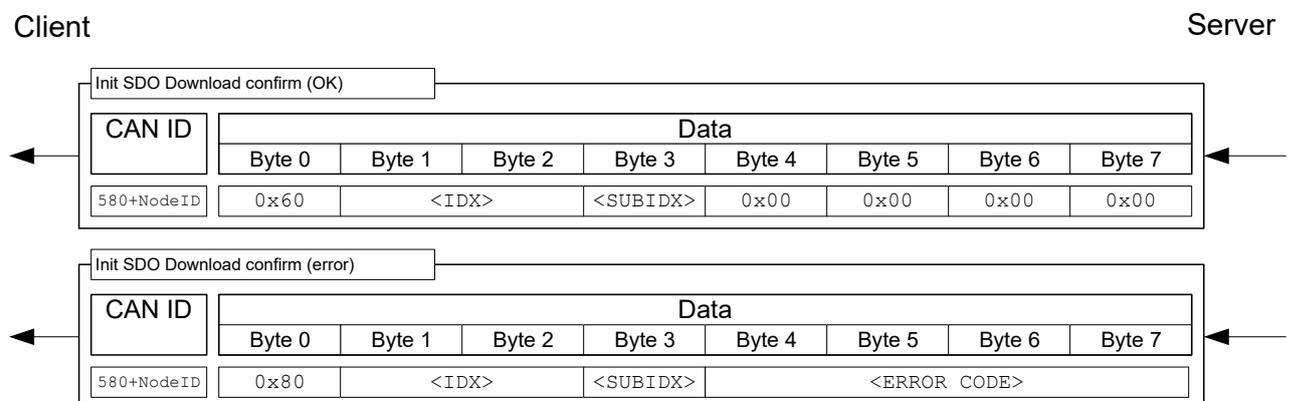


Dabei ist das Byte <CMD> abhängig von der Länge der Daten, welche geschrieben werden sollen. <CMD> kann einer der folgenden Werte sein:

- 1 Byte Datenlänge: $2F_h$
- 2 Byte Datenlänge: $2B_h$
- 3 Byte Datenlänge: 27_h
- 4 Byte Datenlänge: 23_h

Das Feld <Data> wird mit den zu schreibenden Daten beschrieben, das LSB der Daten wird in Byte 4 eingetragen.

Die Antwort des Servers ist entweder eine Bestätigung des Schreibvorganges oder eine Fehlermeldung (Aufbau der Nachrichten: siehe nachfolgende Abbildung). Im letzteren Fall wird der Grund des Fehlers in den Daten mitgesendet (siehe Liste der SDO-Fehlermeldungen in Abschnitt **SDO-Fehlermeldungen**).



Beispiel: Setzen des Objekts $607A_h:00_h$ (Target position, SIGNED32) auf den Wert $3E8_h$ (=1000_d) einer Steuerung mit der Node-ID 3:

603 | 23 7A 60 00 E8 03 00 00

Dabei entspricht

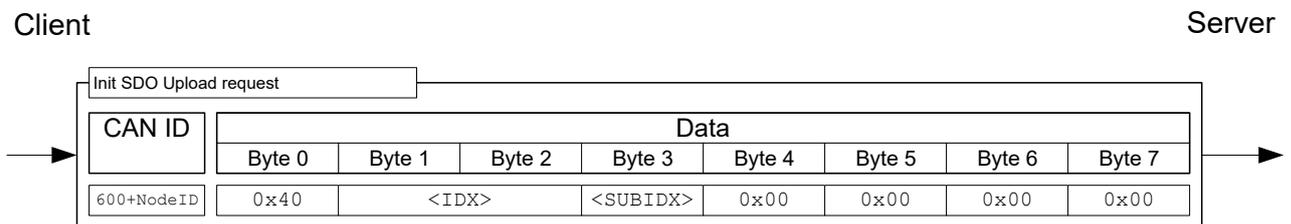
- Byte 1 (23_h): SDO expedited download, 4Bytes Daten (SIGNED32)
- Byte 2 und 3 (7A_h 60_h): Index des Objektes ist 607A_h
- Byte 4 (00_h): Subindex des Objektes ist 00_h
- Byte 5 bis 8 (E8_h 03_h 00_h 00_h): Wert des Objektes: 000003E8_h

Im Erfolgsfall antwortet die Steuerung mit dieser Nachricht:

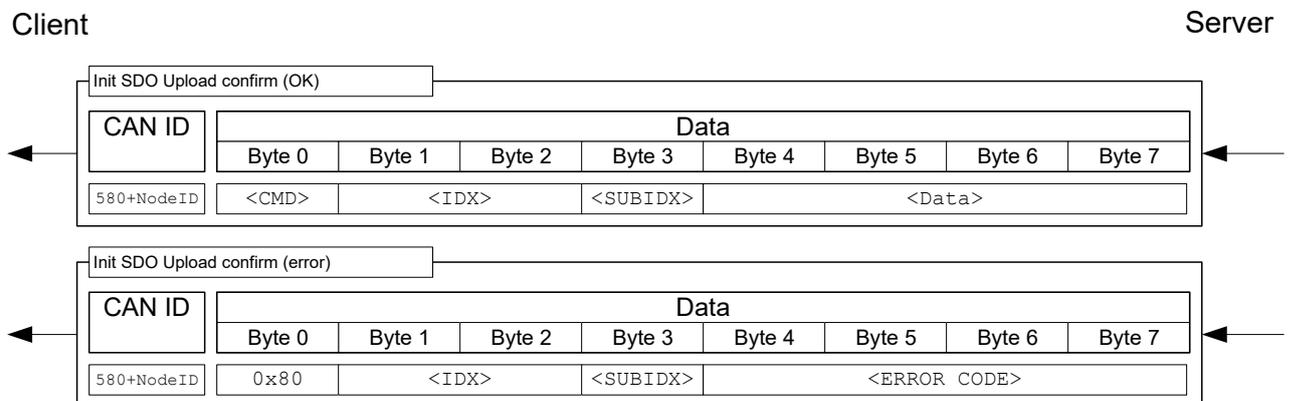
583 | 60 7A 60 00 00 00 00 00

SDO-Upload

Eine CAN-Nachricht zum Lesen eines Objektes aus dem Objektverzeichnis hat den nachfolgenden Aufbau:



Der Server antwortet dabei mit einer der nachfolgenden Nachrichten.



Die Länge der Daten ist im <CMD> der Antwort verschlüsselt:

- 1 Byte Datenlänge: 4F_h
- 2 Byte Datenlänge: 4B_h
- 3 Byte Datenlänge: 47_h
- 4 Byte Datenlänge: 43_h

Das LSB der Daten steht dabei wieder im Byte 4.

Im Fehlerfall ist der Grund des Fehlers in den Daten mit angegeben (siehe Liste der SDO-Fehlermeldungen in **SDO-Fehlermeldungen**).

Beispiel: Um das Objekt "Statusword" (6041_h:00) aus dem Objektverzeichnis zu lesen, reicht es aus, folgende Nachricht zu senden (immer 8 Bytes):

603 | 40 41 60 00 00 00 00 00

Die Steuerung antwortet im Regelfall mit folgender Nachricht:

583 | 4B 41 60 00 40 02 00 00

Dabei entspricht

- Byte 1 (4B_h): SDO expedited upload, 2 Bytes Daten (UNSIGNED16)
- Byte 2 und 3 (41_h 60_h): Index des Objektes ist 6041_h
- Byte 4 (00_h): Subindex des Objektes ist 00_h
- Byte 5 bis 6 (40_h 02_h): Wert des Objektes: 0240_h
- Byte 7 bis 8 (00_h 2_h h h): leer. Eine SDO-Nachricht besteht immer aus 8 Bytes.

Normal Transfer

Die CANopen-Übertragung "expedited" ist auf maximal vier Byte beschränkt, um diese Schranke zu überschreiten, muss der sogenannte "normal transfer" unterstützt werden. Bei dieser Übertragungsart wird der Inhalt mehrerer Nachrichten inhaltlich zusammengefasst, ein solcher Block an Nachrichten wird im Folgenden als "Transfer" bezeichnet. Jede Nachricht innerhalb eines Transfers wird dabei einzeln bestätigt.

Das ist zum Zeitpunkt der Erstellung des Dokumentes nur für Objekte des Types "String" notwendig. Da ein String die Zugriffsbeschränkung "read only" hat, ist ein SDO-Download nicht nötig, in diesem Dokument wird daher nur auf den SDO-Upload eingegangen.

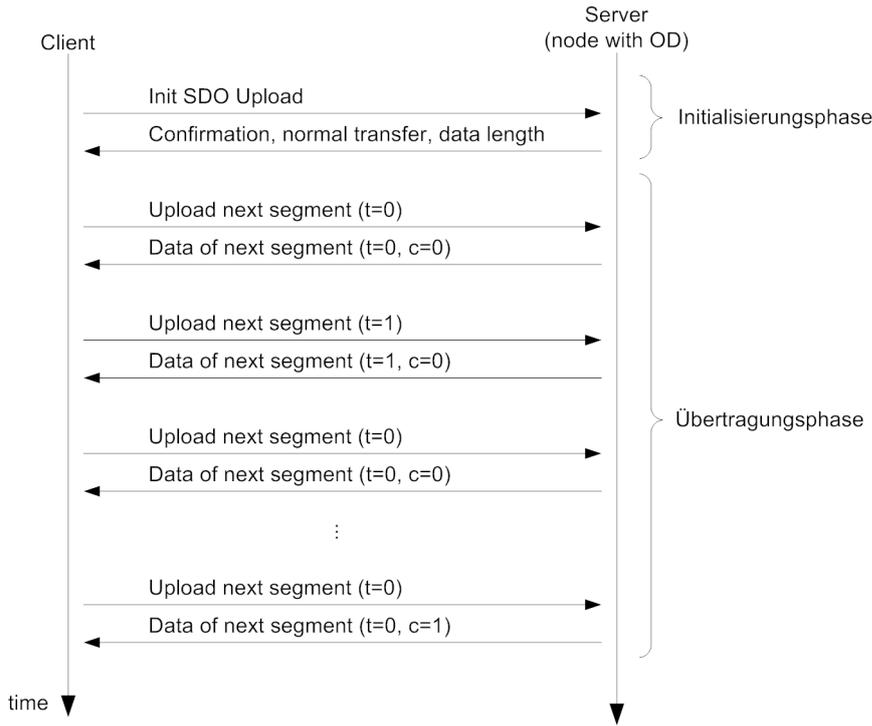
Fehlende Unterstützung des "normal transfer" eines Masters

Sollte die Steuerung von einem Master bedient werden, welcher keinen "normal transfer" unterstützt, kann der Zugriff auf Objekte mit dem Datentyp String auch anders gelöst werden: Jeder String kann mit einem SDO-Upload auf den Subindex 1 und den folgenden Subindizes zeichenweise ausgelesen werden.

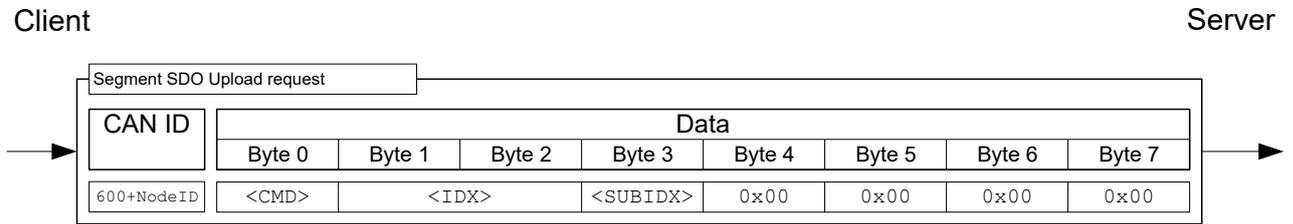
Beispiel: Das Objekt 6505_h (http drive catalogue address) soll ausgelesen werden. Unterstützt der Master "normal transfer", reicht es aus, den Upload des Objektes über den Subindex 00 zu beginnen, die Steuerung stellt automatisch auf "normal transfer" um. Sollte der Master nur "expedited transfer" unterstützen, kann über die Objekte 6505_h:01, 6505_h:02, 6505_h:03 usw. der String Zeichen für Zeichen ausgelesen werden.

SDO-Upload

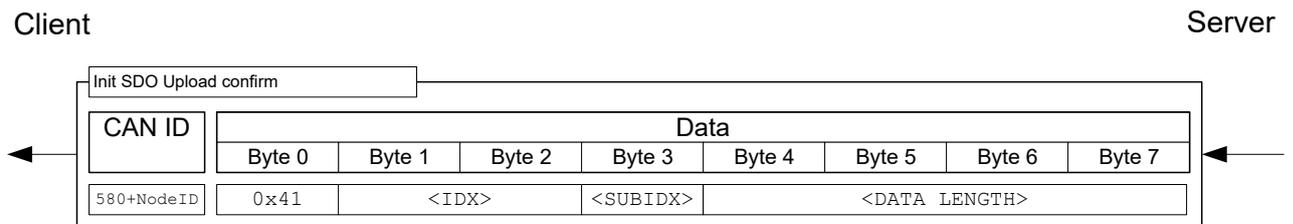
In nachfolgender Abbildung ist die Vorgehensweise eines "SDO Uploads" dargestellt (Client lässt sich den Inhalt eines Objektes schicken). Die Übertragung zerfällt in zwei Phasen: Einer Initialisierungs- und einer Übertragungsphase.



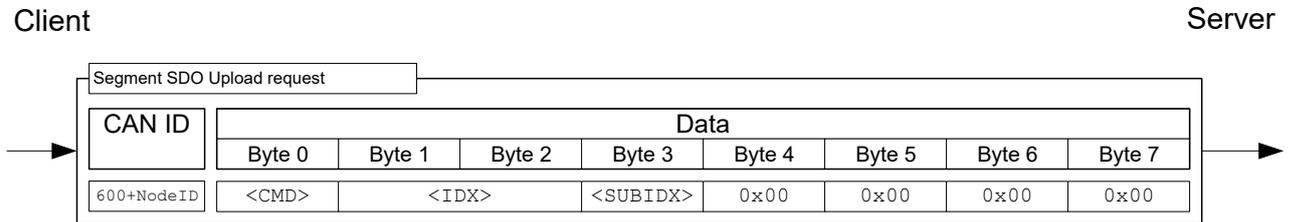
Der Upload beginnt, in dem der Client - wie bei einem "expedited transfer" auch - einen "Init SDO Update" an den Server schickt (siehe nachfolgende Abbildung).



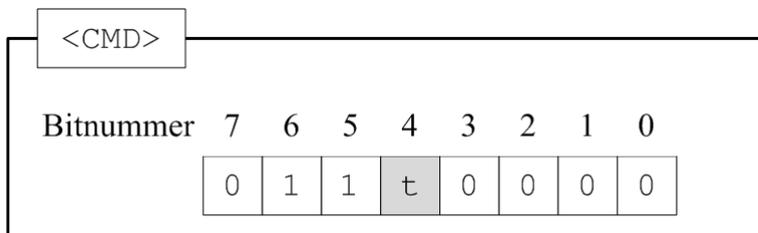
Die Antwort für einen "normal transfer" enthält die Menge der zu empfangenen Bytes nicht im <CMD> codiert, sondern im Datenbereich eingetragen wie es in der nachfolgenden Abbildung im Bereich <DATA LENGTH> zu sehen ist.



Damit gilt die Initialisierung als abgeschlossen, im Anschluss erfolgt nur noch der Upload der Daten.
Ein Datenpaket wird mit folgenden SDO-Request angefordert:

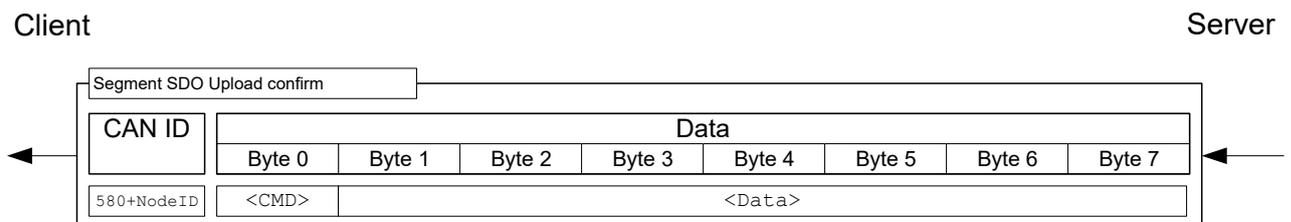


Das Byte 0 mit dem Kommando <CMD> setzt sich folgendermaßen zusammen:

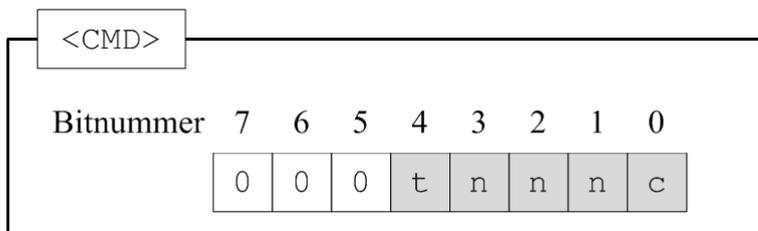


Das Bit mit der Bezeichnung t alterniert mit jeder Anforderung ("toggle bit"). Es beginnt mit jedem Transfer bei 0, auch wenn der vorherige Transfer abgebrochen wurde.

Die Steuerung antwortet auf die obige Nachricht mit den Daten, wobei die Nachricht folgendermaßen aufgebaut ist:



Das Byte 0 mit <CMD> setzt sich folgendermaßen zusammen:



Die Bits haben dabei folgende Bedeutung:

t (toggle bit)

Das Bit alterniert mit jeder Nachrichtensequenz, es ändert sich nicht innerhalb einer Sequenz zwischen "Request" und "Response".

n (number of bytes)

Diese drei Bits geben an, wie viele Bytes *keine* Daten enthalten. Beispiel: sind Bit 2 und 1 auf 0, Bit 3 auf 1 dann sind 011_b = 03_d Bytes nicht gültig. Im Umkehrschluss bedeutet das, dass Byte 1 bis Byte 4 zulässige Werte enthalten und Byte 5 bis Byte 7 nicht beachtet werden sollen.

c (more segments)

Wenn keine weiteren SDO-Segmente mehr verschickt werden und es sich dann hierbei um das letzte Segment handelt, wird das Bit auf 1 gesetzt.

Beispiel: In diesem Beispiel soll das Objekt "Manufacturer Software Version" (100A_h) ausgelesen werden. Die Node-ID des Knotens ist in diesem Beispiel die 3.

Die dazugehörige SDO-Nachrichten-Sequenz wird in nachfolgender Tabelle aufgelistet. Der auszulesende String variiert von Steuerung zu Steuerung.

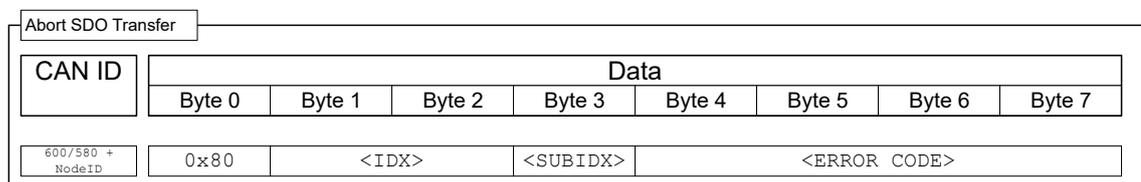
COB-ID	Daten	Beschreibung
603 _h	40 0A 10 00 00 00 00 00	Init Upload; Index: 100A _h ; Subindex: 00
583 _h	41 0A 10 00 11 00 00 00	Init Upload; Size: indicated; transfer type: normal; Num of bytes: 17; Index: 100A _h ; Subindex: 00
603 _h	60 0A 10 00 00 00 00 00	Upload Segment Req.; Toggle bit: not set
583 _h	00 46 49 52 2D 76 31 37	Upload Segment Conf.; More segments: yes; num of bytes: 7; Toggle bit: not set
603 _h	70 0A 10 00 00 00 00 00	Upload Segment Req.; Toggle bit: set
583 _h	10 34 38 2D 42 35 33 38	Upload Segment Conf.; More segments: yes; num of bytes: 7; Toggle bit: set
603 _h	60 0A 10 00 00 00 00 00	Upload Segment Req.; Toggle bit: not set
583 _h	09 36 36 32 00 00 00 00	Upload Segment Conf.; More segments: no (last segment); num of bytes: 3; Toggle bit: not set

46 49 52 2D 76 31 37 34 38 2D 42 35 33 38 36 36 32

Das entspricht dem String: "FIR-v1748-B538662"

Abbruch der SDO-Übertragung

Sowohl der Server als auch der Client sind *jederzeit* berechtigt, den derzeitigen Transfer abubrechen. Dazu muss ein "Abort SDO Transfer" gesendet werden, was nachfolgend abgebildet ist.



Nach dem Empfang der Nachricht gilt die SDO-Übertragung als beendet, der Service ist *nicht* bestätigt. Eine neue SDO-Übertragung muss anschließend komplett von vorne begonnen werden. Das Übertragen des <ERROR CODE> ist optional, die Steuerung wertet den Code nicht aus.

SDO-Fehlermeldungen

Im Falle eines Fehlers wird im Bereich der Daten eine Fehlernummer mitgesendet, die den Grund des Fehlers angibt.

Error Code	Beschreibung
05030000 _h	<i>toggle bit not changed</i> : Gültig nur bei "normal transfer" oder "block transfer". Das Bit, welches nach jeder Übertragung zu alternieren hat, hat seinen Zustand nicht geändert.
05040001 _h	<i>command specifier unknown</i> : Das Byte 0 des Datenblocks enthielt einen nicht zulässigen Befehl.
06010000 _h	<i>unsupported access</i> : Falls über CAN over EtherCAT (CoE) ein "complete access" angefordert wurde (wird nicht unterstützt.)
06010002 _h	<i>read only entry</i> : Es wurde versucht, auf ein konstantes oder nur lesbares Objekt zu schreiben.
06020000 _h	<i>object not existing</i> : Es wurde versucht, auf ein nicht vorhandenes Objekt zu zugreifen (Index fehlerhaft).
06040041 _h	<i>objekt cannot be pdo mapped</i> : Es wurde versucht, ein Objekt in das PDO zu mappen, für das das nicht zulässig ist.
06040042 _h	<i>mapped pdo exceed pdo</i> : Würde das gewünschte Objekt in das PDO-Mapping angehängt werden, würden die 8Byte des PDO-Mappings überschritten.
06070012 _h	<i>parameter length too long</i> : Es wurde versucht, auf ein Objekt mit zu vielen Daten zu schreiben; zum Beispiel mit <CMD>=23 _h (4 Byte) auf ein Objekt des Types Unsigned8, korrekt wäre das <CMD>=2F _h .
06070013 _h	<i>parameter length too short</i> : Es wurde versucht, auf ein Objekt mit zu wenig Daten zu schreiben; zum Beispiel mit <CMD>=2F _h (1 Byte) auf ein Objekt des Types Unsigned32, korrekt wäre das <CMD>=23 _h .
06090011 _h	<i>subindex not existing</i> : Es wurde versucht, auf ein ungültiges Subindex eines Objektes zu zugreifen, der Index hingegen würde existieren.
06090031 _h	<i>value too great</i> : Einige Objekte unterliegen Restriktionen in der Größe des Wertes, in diesem Fall wurde versucht, einen zu hohen Wert in das Objekt zu schreiben. Zum Beispiel darf das Objekt "Pre-defined error field: Number of errors" bei 1003 _h :00 nur auf den Wert "0" gesetzt werden, alle anderen Zahlenwerte provozieren diesen Fehler.
06090032 _h	<i>value too small</i> : Einige Objekte unterliegen Restriktionen in der Größe des Wertes. In diesem Fall wurde versucht, einen zu niedrigen Wert in das Objekt zu schreiben.
08000000 _h	<i>general error</i> : Allgemeiner Fehler, der in keine andere Kategorie passt.
08000022 _h	<i>data cannot be read or stored in this state</i> : Die Parameter des PDOs dürfen nur im State <i>Stopped</i> oder "Pre-Operational" verändert werden. Ein Schreibzugriff auf die Objekte 1400 _h bis 1407 _h , 1600 _h bis 1607 _h , 1800 _h bis 1807 _h und 1A00 _h bis 1A07 _h ist im Zustand "Operational" nicht zulässig.

8.2.5 Process Data Object (PDO)

Eine Nachricht, die nur Prozessdaten enthält, wird als "Process Data Object" (PDO) bezeichnet. Gedacht ist das PDO für Daten, die zyklisch ausgetauscht werden müssen. Die Idee einer PDO-Nachricht ist es, sämtliche Zusatzinformationen (Index, Subindex und Datenlänge) aus einer CAN-Nachricht zu entfernen und die CAN-Nachricht nur noch mit Daten zu füllen. Die Quell- und Zielinformationen zu dem PDO werden separat im sogenannten PDO-Mapping gespeichert.

PDOs lassen sich nur verwenden, wenn sich die NMT-State Maschine im Zustand "Operational" befindet (siehe Abschnitt **Network Management (NMT)**), die Konfiguration der PDOs muss im NMT-Zustand "Pre-Operational" erfolgen.

Die Steuerung unterstützt insgesamt 8 unabhängige PDO-Mappings, jede zugehörige PDO-Nachricht kann maximal acht Bytes (=64Bit) an Nutzdaten tragen. Damit lassen sich beispielsweise zwei Unsigned32-Werte übertragen oder ein UNSIGNED32 und ein UNSIGNED08, die Nachricht muss dabei nicht alle acht Datenbytes voll ausnutzen. Die PDOs unterscheiden sich noch einmal in der Konfiguration in die Sende- und Empfangs-Konfiguration. Die Empfangs-Konfiguration beschreibt

die Verarbeitung für PDO-Nachrichten, die empfangen werden, und die Sende-Konfiguration der zu sendenden PDO-Nachrichten.

RX-Konfiguration

Um ein RX-PDO zu konfigurieren, müssen drei Objektkategorien im Objektverzeichnis berücksichtigt werden:

1. Die Objekte, welche die Funktionalität des Mappings beschreiben.
2. Die Objekte, welche den Inhalt des Mappings beschreiben.
3. Die Objekte, welche die empfangenen Daten erhalten sollen.

Konfiguration der Funktionalität (Communication Parameter)

Die Konfiguration des ersten Mappings wird in den Subindizes des Objektes 1400_h gespeichert. Das zweite Mapping wird in 1401_h konfiguriert und so weiter. Im Folgenden wird jeweils vom 140N_h gesprochen. Die Konfiguration betrifft dabei die COB-ID der PDO-Nachricht und die Übertragungsart.

Die Objekte 140N_h besitzen nur drei Subindizes:

- Subindex 0 (max. subindex): Anzahl der gesamten Subindizes
- Subindex 1 (COB-ID): Hier wird die COB-ID hinterlegt. Für PDO-Mapping 1-4 (1600_h..1603_h) gilt, dass die CAN-ID abhängig von der Node-ID fix ist und nur das Valid-Bit (Bit 31) in der COB-ID gesetzt werden kann. Von 1604_h..1607_h kann die CAN-ID eigenständig gesetzt werden (mit der Einschränkung, dass diese nicht von anderen Diensten verwendet wird, siehe Tabelle am Anfang des Kapitels **CANopen Dienste**) und auch das Valid-Bit. Die Änderung einer COB-ID wird erst *nach* dem Neustart der Steuerung oder der Kommunikation aktiv (siehe **Network Management (NMT)**).

Mapping	COB-ID
1600 _h	200 _h + Node-ID
1601 _h	300 _h + Node-ID
1602 _h	400 _h + Node-ID
1603 _h	500 _h + Node-ID
1604 _h	xxx _h + Node-ID
1605 _h	xxx _h + Node-ID
1606 _h	xxx _h + Node-ID
1607 _h	xxx _h + Node-ID

- Subindex 2 (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden. Die Nummer und die zugehörige Bedeutung können Sie aus der nachfolgenden Tabelle entnehmen.

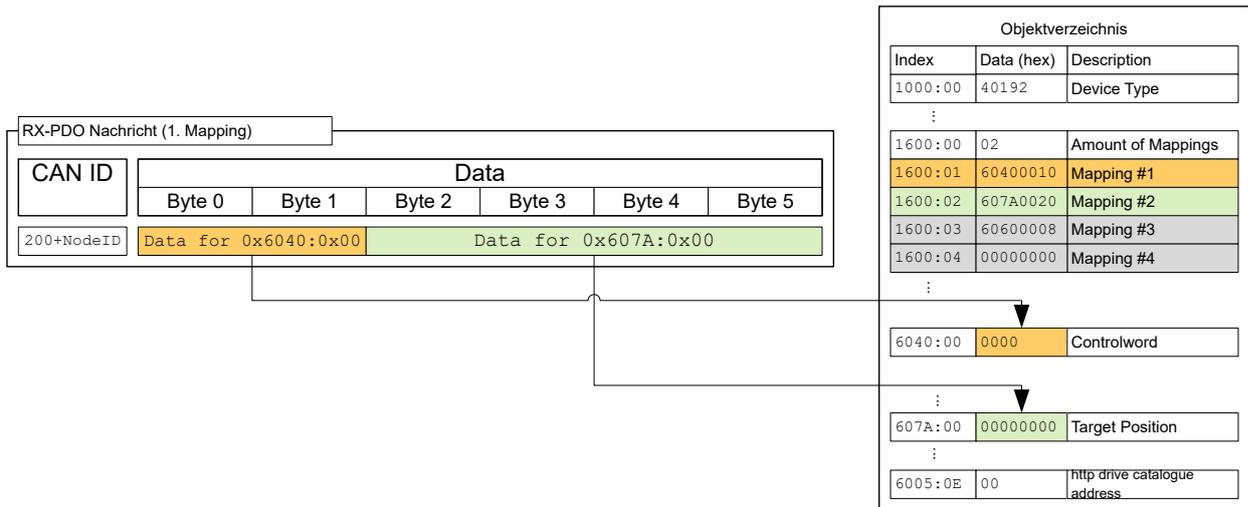
140N _h :02 _h	Bedeutung
00 _h -F0 _h	Synchronous: Die Daten werden zwischengespeichert und erst mit dem Erhalt der nächsten SYNC-Nachricht gültig und in das Objektverzeichnis übernommen.
F1 _h -FD _h	Reserviert
FE _h , FF _h	Asynchronous : Die Daten werden mit dem Erhalt der PDO-Nachricht gültig und in das Objektverzeichnis übernommen.

Inhalt eines Mappings

Die Konfiguration des Inhalts eines Mappings setzt sich wie folgt zusammen (siehe auch nachfolgende Abbildung als Beispiel):

- Alle Subindizes eines Konfigurationsobjektes gehören zusammen, so beschreibt das 1600_h mit allen Subindizes das erste Mapping, das 1601_h das zweite RX-PDO-Mapping usw.

- Der Subindex 00_h gibt an, wie viele Objekte sich in einem Mapping befinden. Er gibt gleichzeitig an, wie viele der Subindizes gültig sind. Wird das Objekt 1600_h:00_h auf "0" gesetzt, ist das RX-Mapping damit vollständig abgeschaltet. In dem Beispiel aus der nachfolgenden Abbildung werden somit zwei Objekte gemappt, das Objekt 1600_h:03_h und 1600_h:04_h ist damit nicht aktiv (grau dargestellt).
- Jeder Subindex von 1600_h:01_h bis 1600_h:0F_h beschreibt fortlaufend ohne Lücken jeweils ein Ziel des Mappings. Dabei wird der Index, Subindex und die Bitlänge codiert. Beispiel aus nachfolgender Abbildung: die ersten zwei Bytes der Nachricht sollen in das Objekt 6040_h:00_h geschrieben werden. In hexadezimaler Schreibweise setzt sich der Inhalt des 1600_h:01_h dann aus <Index><Subindex><Bitlänge> zusammen, also 60400010. Das zweite Mapping (1600_h:02_h) enthält den Eintrag 607A0020. Es mappt also die folgenden vier Byte (=20_hBit) in das Objekt 607A_h:00_h



TX-Konfiguration

Um ein RX-PDO zu konfigurieren, müssen drei Objektkategorien im Objektverzeichnis berücksichtigt werden:

1. Die Objekte, welche die Funktionalität des Mappings beschreiben.
2. Die Objekte, welche den Inhalt des Mappings beschreiben.
3. Die Objekte, welche die zu sendenden Daten erhalten sollen.

Zudem ist zu beachten, dass der Zeitpunkt - zu dem die Daten in die TX-PDO-Nachricht kopiert werden - und der Zeitpunkt des Versendens nicht der gleiche sein müssen (abhängig vom Modus).

Konfiguration der Funktionalität (Communication Parameter)

Die Konfiguration der Funktionalität des ersten Mappings wird in den Subindizes des Objektes 1800_h gespeichert. Das zweite Mapping wird in 1801_h konfiguriert und so weiter. Im Folgenden wird jeweils vom 180N_h gesprochen. Die Konfiguration betrifft dabei die COB-ID der PDO-Nachricht und die Übertragungsart.

Die Objekte 180N_h besitzen folgende Subindizes:

- Subindex 0 (max. subindex): Anzahl der gesamten Subindizes
- Subindex 1 (COB-ID): Hier wird die COB-ID hinterlegt. Für PDO-Mapping 1-4 (1A00_h..1A03_h) gilt, dass die CAN-ID abhängig von der Node-ID fix ist und nur das Valid-Bit (Bit 31) in der COB-ID gesetzt werden kann. Von 1A04_h..1A07_h kann die CAN-ID eigenständig gesetzt werden (mit der Einschränkung dass diese nicht von anderen Diensten verwendet wird, siehe Tabelle am Anfang des Kapitels **CANopen Dienste**) und auch das Valid-Bit. Die Änderung einer COB-ID wird erst *nach* dem Neustart der Steuerung oder der Kommunikation aktiv (siehe **Network Management (NMT)**).

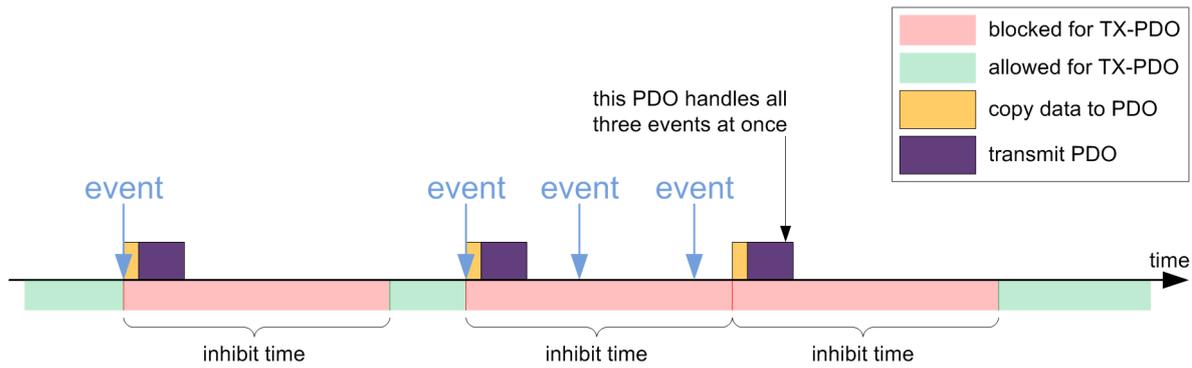
Mapping	COB-ID
1A00 _h	180 _h + Node-ID
1A01 _h	280 _h + Node-ID
1A02 _h	380 _h + Node-ID
1A03 _h	480 _h + Node-ID
1A04 _h	xxx _h + Node-ID
1A05 _h	xxx _h + Node-ID
1A06 _h	xxx _h + Node-ID
1A07 _h	xxx _h + Node-ID

- Subindex 2 (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die Daten in die PDO-Nachricht kopiert und wann dieses gesendet werden soll. Die Nummer und die zugehörige Bedeutung kann aus der nachfolgenden Tabelle entnommen werden. Im Folgenden wird von einem *Event* gesprochen, der das Kopieren und/oder das Senden der Daten anstoßen kann. Zu diesem *Event* zählen drei Ereignisse, die unabhängig voneinander betrachtet werden:
 - Schalten der NMT-Zustandsmaschine auf "operational".
 - Die gegenwärtigen Daten haben sich gegenüber der letzten PDO-Nachricht geändert.
 - Der *Event Timer* ist abgelaufen (siehe 180N_h:5).

Wird der *Event Timer* benutzt, wird dieser unabhängig von den Änderungen behandelt, der *Event Timer* wird erst nach Ablauf desselben neu gestartet, nicht aufgrund eines anderen *Events*.

180N _h :02 _h	Bedeutung
0	Synchronous (acyclic): Die Daten werden mit dem Eintreffen des SYNC in das TX-PDO kopiert aber erst mit dem <i>Event</i> versendet.
01 _h -F0 _h	Synchronous (cyclic): Die Daten werden mit dem Eintreffen der n-ten SYNC-Nachricht kopiert und sofort im Anschluss verschickt (n entspricht der Zahl 1 bis 240, der transmission type "1" sendet bei jedem SYNC die neuen Daten).
F1 _h -FB _h	Reserviert
FC _h	RTR-Only (synchronous): Die Daten werden mit dem Eintreffen jeder SYNC-Nachricht kopiert aber erst auf Anforderung mittels einer RTR-Nachricht verschickt.
FD _h	RTR-Only (event-driven): Die Daten werden mit dem Erhalt einer RTR-Nachricht in die TX-PDO-Nachricht kopiert und daraufhin sofort versendet.
FE _h , FF _h	Die Daten werden beim Eintreten des <i>Events</i> kopiert und sofort versendet.

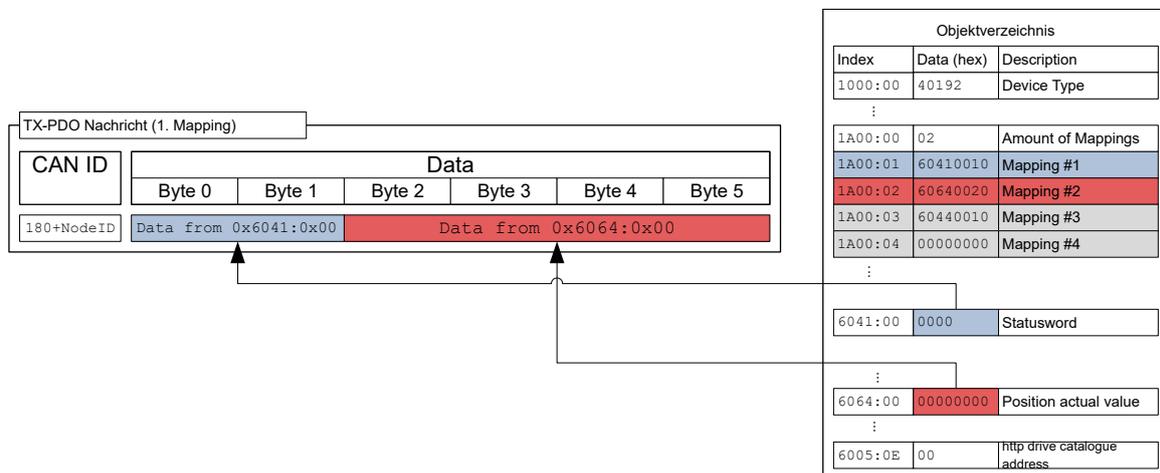
- Subindex 3 (inhibit time): Dieser Subindex enthält eine Zeitsperre in 100-µs-Schritten (siehe nachfolgende Abbildung). Hier kann eine Zeit eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs. Dadurch soll verhindert werden, dass asynchrone PDOs permanent verschickt werden, wenn sich das gemappte Objekt dauernd ändert.
- Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.
- Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.
- Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.



Inhalt eines Mappings

Die Konfiguration des Inhalts eines Mappings setzt sich wie folgt zusammen (siehe nachfolgende Abbildung als Beispiel):

- Alle Subindizes eines Konfigurationsobjektes gehören zusammen, so beschreibt das 1A00_h mit allen Subindizes das erste Mapping, das 1A01_h das zweite RX-PDO-Mapping usw.
- Der Subindex 00 gibt an, wie viele Objekte sich in einem Mapping befinden. Es gibt gleichzeitig an, wie viele der Subindizes gültig sind. Wird das Objekt 1A00_h:00_h auf "0" gesetzt, ist das RX-Mapping damit vollständig abgeschaltet. Im nachfolgendem Beispiel werden somit zwei Objekte in den Einträgen 1A00_h:01_h - 1A00_h:02_h gemappt. Die Objekte in den Einträgen 1A00_h:03_h - 1A00_h:04_h werden somit nicht gemappt (grau dargestellt).
- Jeder Subindex von 1A00_h:01_h bis 1A00_h:0F_h beschreibt fortlaufend ohne Lücken (für eine Lücke können Dummy-Objekte verwendet werden) jeweils eine Quelle des Mappings. Dabei wird der Index, Subindex und die Bitlänge kodiert. Beispiel aus nachfolgender Abbildung: die ersten zwei Byte der Nachricht sollen aus dem Objekt 6041_h:00_h gelesen werden. In hexadezimaler Schreibweise setzt sich der Inhalt des 1A00_h:01_h dann aus <Index><Subindex><Bitlänge> zusammen, also 60410010. Das zweite Mapping (1A00_h:02_h) enthält den Inhalt 60640020. Es mappt also die folgenden vier Byte (entspricht 32 Bits) aus dem Objekt 6064_h:00_h in die TX-PDO-Nachricht.



Voreinstellung

Voreingestellt ist folgende Konfiguration:

RX-PDO

1. Mapping (CAN-ID: 200_h + Node-ID):

- 6040_h:00_h (controlword)
- 6060_h:00_h (mode of operation)
- 3202_h:02_h (motor drive submode select)

2. Mapping (CAN-ID: 300_h + Node-ID):
 - **607A**_h:00_h (target position)
 - **6081**_h:00_h (profile velocity)
3. Mapping (CAN-ID: 400_h + Node-ID): Objekt **6042**_h:00_h (vl target velocity)
4. Mapping (CAN-ID: 500_h + Node-ID): Objekt **60FE**_h:01_h (digital outputs)

TX-PDO

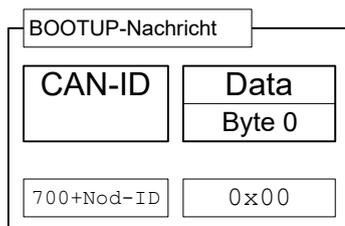
1. Mapping (CAN-ID: 180_h + Node-ID):
 - **6041**_h:00_h (statusword)
 - **6061**_h:00_h (Modes Of Operation Display)
2. Mapping (CAN-ID: 280_h + Node-ID): **6064**_h:00_h (Position actual value)
3. Mapping (CAN-ID: 380_h + Node-ID): **6044**_h:00_h (vl velocity actual value)
4. Mapping (CAN-ID: 480_h + Node-ID): Objekt **60FD**_h:00_h (Digital Inputs)

PDO-Mapping ändern

1. Deaktivieren Sie das PDO, indem Sie das *Valid Bit* (Bit 31) des Subindex 01h des dazugehörigen Communication Parameter (z.B. 1400_h:01_h) auf "1" setzen.
2. Deaktivieren Sie das Mapping indem Sie den Subindex 00h des dazugehörigen Mapping Parameter (z.B. 1600_h:00_h) auf "0" setzen.
3. Ändern Sie das Mapping in den gewünschten Subindizes (z.B. 1600_h:01_h).
4. Aktivieren Sie das Mapping in dem Sie die Anzahl der zu mappenden Objekte in den Subindex 00h des dazugehörigen Mapping Parameter (z.B. 1600_h:00_h) schreiben.
5. Aktivieren Sie das PDO indem Sie Bit 31 des Subindex 01h des dazugehörigen Communication Parameter (z.B. 1400_h:01_h) auf "0" setzen.

8.2.6 Boot-Up Protocol

Erreicht der CAN-Slave den NMT-Zustand "Pre-Operational" (siehe nachfolgende Abbildung), dann wird die nachfolgende Nachricht verschickt, um die Betriebsbereitschaft zu signalisieren.



Dieser Service ist unbestätigt, es erfolgt keine Antwort.



Hinweis

Der Bootloader sendet eine eigene Boot-Up-Nachricht. Diese kann unterdrückt werden, siehe Objekt **2007**_h:00

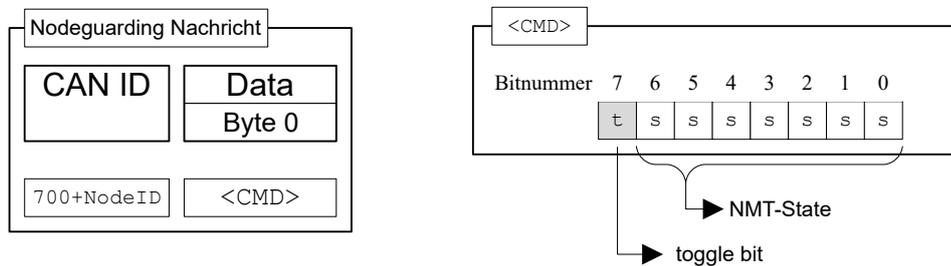
8.2.7 Heartbeat und Nodeguarding

Mit den Services "Heartbeat" und "Nodeguarding" (oft auch mit "Liveguarding" bezeichnet) lassen sich abgeschaltete oder abgestürzte Geräte am CAN-Bus detektieren. Dazu fordert der NMT-Master zyklisch eine Nachricht mit dem aktuellen NMT-Zustand des Slaves an (Nodeguarding). Die

Alternative ist, dass jeder Slave unaufgefordert und zyklisch eine Nachricht versendet (Heartbeat). Eine Kombination aus Nodeguarding und Heartbeat ist nicht zulässig. Es wird zudem empfohlen, den Heartbeat dem Nodeguarding vorzuziehen, da Nodeguarding eine höhere Auslastung des CAN-Busses verursacht.

Nodeguarding

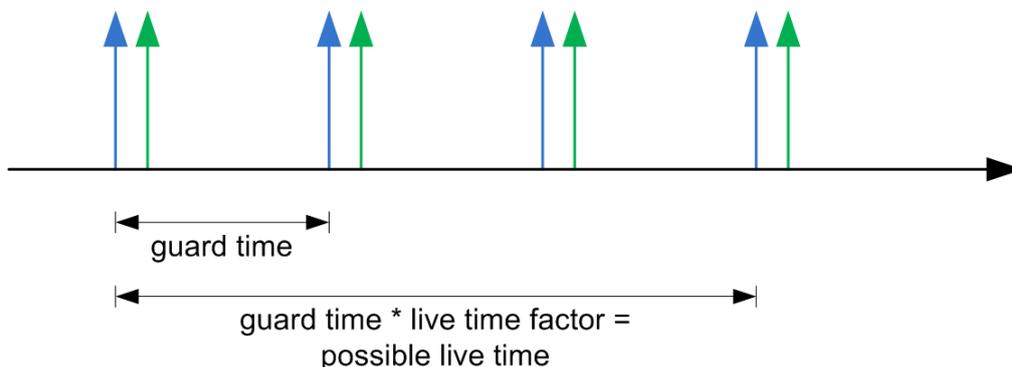
Dieser Service basiert darauf, dass der NMT-Master eine RTR-Nachricht mit der CAN-ID $700_h + \text{Node-ID}$ an den jeweiligen Slave verschickt. Anschließend muss der Slave eine Nachricht als Antwort verschicken, welche nachfolgend abgebildet ist. Das Bit 7 alterniert dabei bei jeder Übertragung, somit kann festgestellt werden, ob eine Nachricht verloren ging. In den Bits 6 bis 0 wird der momentane NMT-Status des Slaves eingetragen.



Es existieren beim Nodeguarding drei Zeitintervalle (siehe auch nachfolgende Abbildung):

1. *guard time*: Die Zeit, zwischen zwei RTR-Nachrichten. Diese kann für jeden CAN-Knoten unterschiedlich sein und wird im Slave im Objekt $100C_h:00$ hinterlegt (Einheit: Millisekunden)
2. *live time factor*: Ein Multiplikator für die *guard time*, diese wird im CAN-Slave im Objekt $100D_h:00$ hinterlegt und kann für jeden Slave am CAN-Bus unterschiedlich sein.
3. *possible live time*: Die Zeitdauer, welche sich aus der Multiplikation aus *guard time* und *live time factor* ergibt.

- RTR Ticket to $0x700 + \text{Node ID}$ sent from NMT Master
- Answer from Client: NMT Ticket with current NMT state

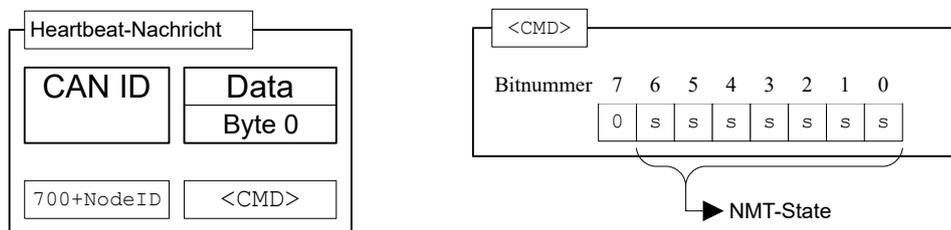


Folgende drei Bedingungen werden beim Nodeguarding geprüft:

1. Der NMT-Master muss innerhalb der "possible live time" die RTR-Anforderung verschicken.
2. Der Slave muss innerhalb der "possible live time" die Antwort auf die RTR-Anforderung verschicken.
3. Der Slave muss mit seinem NMT-Zustand antworten. Zudem muss das "toggle bit" korrekt gesetzt sein.

Heartbeat

Ist der Heartbeat aktiviert, sendet der Slave ohne weitere Aufforderung zyklisch seinen NMT-Zustand auf dem CAN-Bus. Aktiviert wird dieser Service, indem die *Producer Heartbeat Time* im Objekt **1017_h:00_h** auf einen anderen Wert als Null gesetzt wird. Die *Producer Heartbeat Time* wird in Millisekunden gemessen. Die vom Slave verschickte Nachricht hat die nachfolgend abgebildete Form:



Der Slave muss innerhalb der *Heartbeat Consumer Time* die Heartbeat-Nachricht verschicken. Diese Zeit ist nur dem Master bekannt und wird in der Steuerung nicht hinterlegt.

Der Slave kann auch einen *Heartbeat* von einem anderen *Producer* (Master oder anderem Slave) überwachen. Dazu müssen die Zeit *Consumer Heartbeat Time* und die Node-ID des *Producer* im Objekt **1016_h** eingetragen werden.

Fehler, die bei dieser Überwachung auftreten, werden zurückgesetzt, wenn entweder die Funktion deaktiviert wird oder der *Heartbeat* wieder in der korrekten Zeit gesendet wird.

8.3 LSS-Protokoll

Mit den Diensten des *LSS-Protokolls* (*Layer Settings Services*) erfolgt die Vergabe der Node-ID und/oder Baudrate der Steuerung direkt über den CANopen-Bus. Dies ist besonders nützlich bei Geräten, die über keine Möglichkeit zur mechanischen Konfiguration (z.B. Drehschalter) der Parameter verfügen.

8.3.1 Allgemeines

Das *LSS-Protokoll* benötigt ein CANopen-Gerät im Netzwerk, das die Rolle des *LSS-Masters* einnimmt. Alle anderen Geräte haben die Rolle des *LSS-Slaves*.

Jeder *LSS-Slave* verfügt über eine eindeutige *LSS-Adresse*, die aus den vier 32-Bit-Einträgen des Objekts **1018_h Identity Object** besteht.

Ein *LSS-Slave* darf entweder im *Konfigurationsmodus* oder im *Wartemodus* sein. Der *LSS-Master* ist zuständig für das Umschalten zwischen den beiden Modi. Manche *LSS-Dienste* (*Configuration, Inquiry*) sind nur im *Konfigurationsmodus* verfügbar.

8.3.2 LSS-Nachricht

Alle Nachrichten des *LSS-Protokolls* bestehen aus 8 Bytes (*DLC=8*), wobei Byte 0 immer den *Command Specifier (CS)* des Dienstes enthält.

Zwei CAN-ID sind für das *LSS-Protokoll* reserviert:

- 7E5_h: für die Nachrichten vom *LSS-Master* an die *LSS-Slaves* (Request)
- 7E4_h: für die Nachrichten von den *LSS-Slave* an den *LSS-Master* (Response)

8.3.3 LSS-Dienste

Es werden vier Kategorien von Diensten unterstützt:

- **Switch state services**
- **Configuration services**
- **Inquiry services**
- **Identification services**

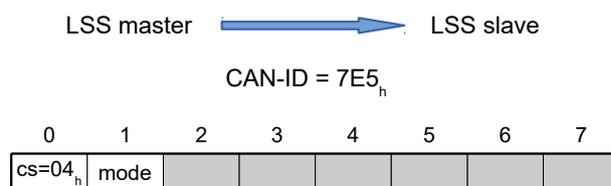
Switch state services

Mit diesen Diensten kann der *LSS-Master* die *LSS-Slaves* in den *Konfigurationsmodus* oder in den *Wartemodus* versetzen.

Das Ändern der Node-ID und Baudrate mittels **Configuration services** und die **Inquiry services** sind nur im *Konfigurationsmodus* zugelassen.

Switch state global service

Mit diesem Dienst versetzt der *LSS-Master* alle *LSS-Slaves* im Netzwerk in den *Konfigurationsmodus* oder in den *Wartemodus*.



Byte 0 : CS (Command Specifier)

Wert = "04_h"

Byte 1: mode

Wert = "00_h": schaltet in den *Wartemodus*

Wert = "01_h": schaltet in den *Konfigurationsmodus*

Bytes 2-7 :

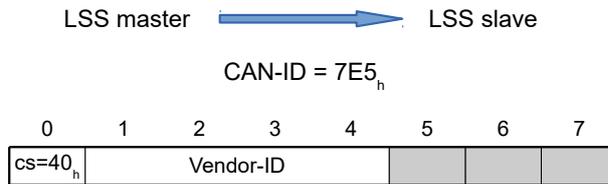
reserviert (=0_h)

Switch state selective service

Mit diesem Dienst versetzt der *LSS-Master* die *LSS-Slaves* mit der (oder Teilen der) entsprechenden *LSS-Adresse* in den *Konfigurationsmodus*.

Der *LSS-Master* sendet vier Nachrichten, welche die *LSS-Adresse* enthalten:

1. Nachricht: Der *LSS-Master* setzt die *LSS-Slaves* mit der entsprechenden *Vendor-ID* in den *Konfigurationsmodus*:



Byte 0 : CS (Command Specifier)

Wert = "40_h"

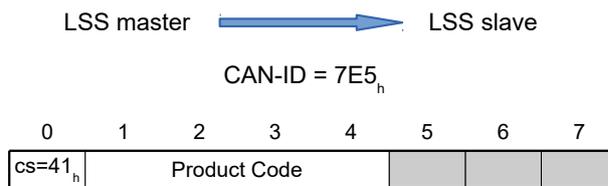
Bytes 1-4: Vendor-ID

Vendor-ID: siehe 1018_h:01_h

Bytes 5-7 :

reserviert (=0_h)

2. Nachricht: Der *LSS-Master* setzt die *LSS-Slaves* mit dem entsprechenden *Product Code* in den *Konfigurationsmodus*:



Byte 0 : CS (Command Specifier)

Wert = "41_h"

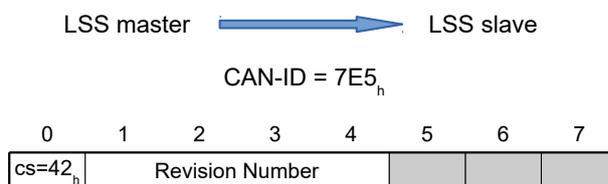
Bytes 1-4: Product Code

Product Code: siehe 1018_h:02_h

Bytes 5-7 :

reserviert (=0_h)

3. Nachricht: Der *LSS-Master* setzt die *LSS-Slaves* mit der entsprechenden *Revision Number* in den *Konfigurationsmodus*:



Byte 0 : CS (Command Specifier)

Wert = "42_h"

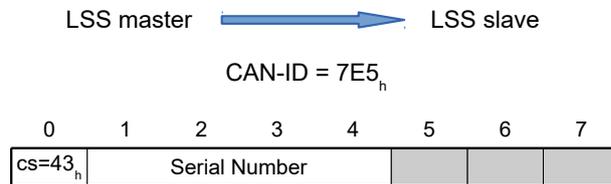
Bytes 1-4: Revision Number

Revision Number: siehe 1018_h:03_h

Bytes 5-7 :

reserviert (=0_h)

4. Nachricht: Der *LSS-Master* setzt die *LSS-Slaves* mit der entsprechenden *Serial Number* in den *Konfigurationsmodus*:



Byte 0 : CS (Command Specifier)

Wert = "43_h"

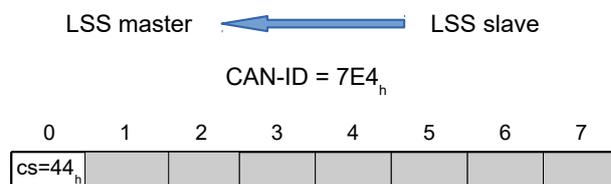
Bytes 1-4: mode

Serial Number: siehe 1018_h:04_h

Bytes 5-7 :

reserviert (=0_h)

Der *LSS-Slave* mit der entsprechenden *LSS-Adresse* wurde in den *Konfigurationsmodus* gesetzt und sendet eine Bestätigung:



Byte 0 : CS (Command Specifier)

Wert = "44_h"

Bytes 1-7 :

reserviert (=0_h)

Configuration services

Mit diesen Diensten kann der *LSS-Master* die Node-ID oder Baudrate der *LSS-Slaves* ändern und ggf. abspeichern.

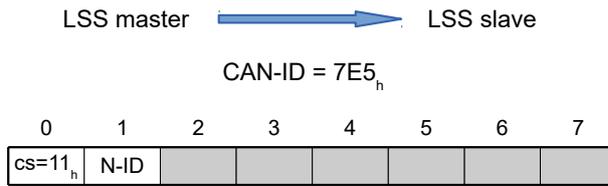


Hinweis

Die *LSS-Slaves* müssen sich im *Konfigurationsmodus* befinden. Siehe Kapitel **Switch state services**.

Configure node-ID service

Der *LSS-Master* sendet eine Nachricht mit der neuen Node-ID an einen *LSS-Slave*:



Byte 0 : CS (Command Specifier)

Wert = "11_h"

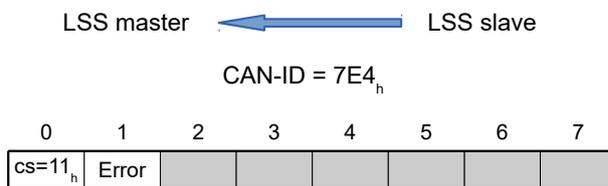
Byte 1: N-ID (Node-ID)

gültige Node-ID zwischen 01_h und 7F_h

Bytes 2-7 :

reserviert (=0_h)

Der LSS-Slave antwortet mit einer Bestätigung/einem Error-Code:



Byte 0 : CS (Command Specifier)

Wert = "11_h"

Byte 1: Error Code

Wert = "00_h": kein Fehler

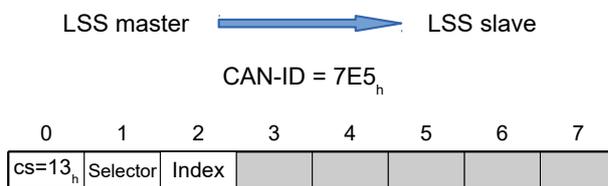
Wert = "01_h": ungültige Node-ID

Bytes 2-7 :

reserviert (=0_h)

[Configure bit timing parameters service](#)

Der LSS-Master sendet eine Nachricht mit der neuen Baudrate an einen LSS-Slave:



Byte 0 : CS (Command Specifier)

Wert = "13_h"

Byte 1: Table Selector

Wert = "00_h": die Tabelle für die Baudrate aus dem CiA 301 Standard wird benutzt.

Byte 2: Table Index

Der Wert für den Index wird der nachfolgenden Tabelle entnommen.

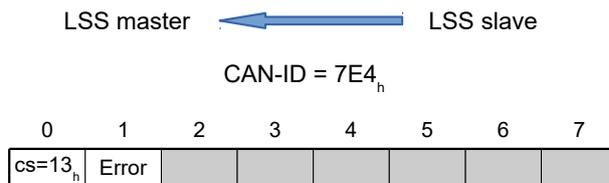
Bytes 3-7 :

reserviert (=0_h)

Folgende Werte werden für den *Table Index* unterstützt:

Table Index	Baudrate in kBd
0	1000
2	500
3	250
4	125
6	50
7	20
8	10

Der *LSS-Slave* antwortet mit einer Bestätigung/einem Error-Code:



Byte 0 : CS (Command Specifier)

Wert = "13_h"

Byte 1: Error Code

Wert = "00_h": kein Fehler

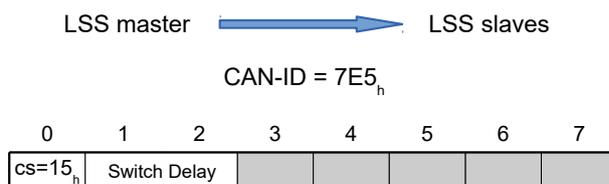
Wert = "01_h": ungültiger *Table Index*/ Baudrate wird nicht unterstützt

Bytes 2-7 :

reserviert (=0_h)

[Activate bit timing parameters service](#)

Der *LSS-Master* aktiviert mit diesem Kommando die eingestellte Baudrate aller *LSS-Slaves* im Netzwerk gleichzeitig:



Byte 0 : CS (Command Specifier)

Wert = "51_h"

Bytes 1-2: Switch Delay

Verzögerung in ms. Dadurch wird gewährleistet, dass alle *LSS-Slaves* im Netzwerk die gleiche Baudrate haben, bevor Nachrichten wieder gesendet werden dürfen.

Nach dem Empfang dieser Nachricht von jedem *LSS-Slave* wird die Zeit abgewartet, die hier hinterlegt wird. Erst danach wird die neue Baudrate übernommen.

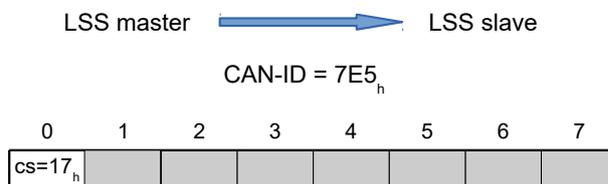
Die gleiche Zeit wird ein zweites Mal abgewartet, erst danach darf ein *LSS-Slave* wieder Nachrichten senden.

Bytes 3-7 :

reserviert (=0_h)

Store configuration service

Der *LSS-Master* speichert mit diesem Kommando die eingestellte Node-ID und Baudrate eines *LSS-Slaves* ab. Der *LSS-Master* muss gewährleisten, dass in dem Moment nur ein *LSS-Slave* im Netzwerk sich im *Konfigurationsmodus* befindet.



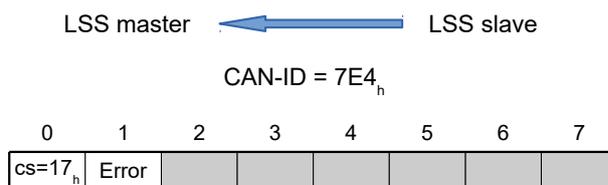
Byte 0 : CS (Command Specifier)

Wert = "17_h"

Bytes 1-7 :

reserviert (=0_h)

Der *LSS-Slave* antwortet mit einer Bestätigung/einem Error-Code:



Byte 0 : CS (Command Specifier)

Wert = "17_h"

Byte 1: Error Code

Wert = "00_h": kein Fehler

Wert = "02_h": Zugriff auf den nichtflüchtigen Speicher fehlgeschlagen

Bytes 2-7 :

reserviert (=0_h)

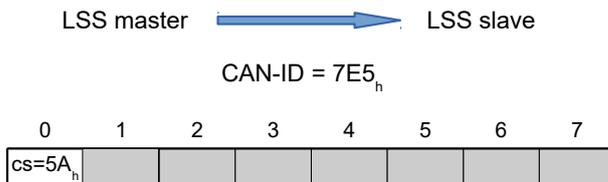
Inquiry services

Mit diesen Diensten kann der *LSS-Master* die *LSS-Adresse* oder die *Node-ID* eines *LSS-Slaves* abfragen. Der *LSS-Master* muss gewährleisten, dass nur ein *LSS-Slave* im Netzwerk sich im *Konfigurationsmodus* befindet.

Inquire LSS address service

Mit diesem Dienst fragt der *LSS-Master* die *LSS-Adresse* eines *Slaves* ab.

1. Der *LSS-Master* fragt die *Vendor-ID* ab:



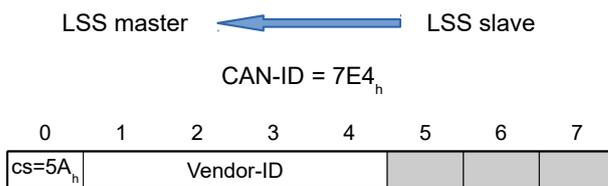
Byte 0 : CS (Command Specifier)

Wert = "5A_h"

Bytes 1-7 :

reserviert (=0_h)

Der *LSS-Slave* sendet seine *Vendor-ID* zurück:



Byte 0 : CS (Command Specifier)

Wert = "5A_h"

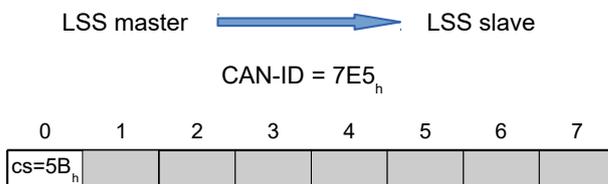
Bytes 1-4: Vendor-ID

Vendor-ID: siehe 1018_h:01_h

Bytes 5-7 :

reserviert (=0_h)

2. Der *LSS-Master* fragt den *Product Code* ab:



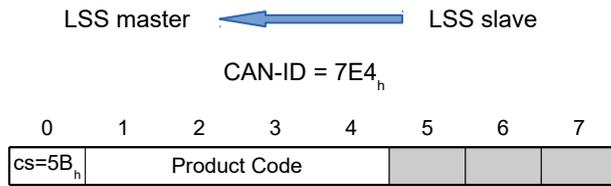
Byte 0 : CS (Command Specifier)

Wert = "5B_h"

Bytes 1-7 :

reserviert (=0_h)

Der *LSS-Slave* sendet seinen *Product Code* zurück:



Byte 0 : CS (Command Specifier)

Wert = "5B_h"

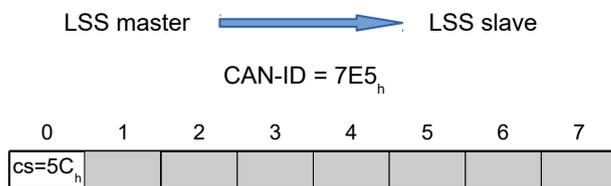
Bytes 1-4: Product Code

Product Code: siehe 1018_h:02_h

Bytes 5-7 :

reserviert (=0_h)

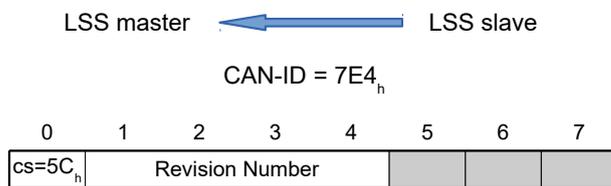
3. Der LSS-Master fragt die Revision Number ab:



Byte 0 : CS (Command Specifier)

Wert = "5C_h"

Der LSS-Slave sendet seine Revision Number zurück:



Byte 0 : CS (Command Specifier)

Wert = "5C_h"

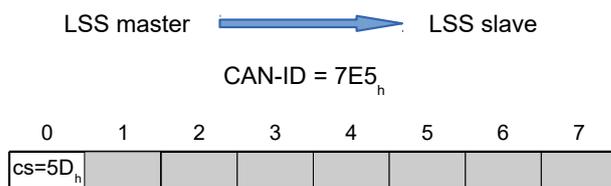
Bytes 1-4: Revision Number

Revision Number: siehe 1018_h:03_h

Bytes 5-7 :

reserviert (=0_h)

4. Der LSS-Master fragt die Serial Number ab:



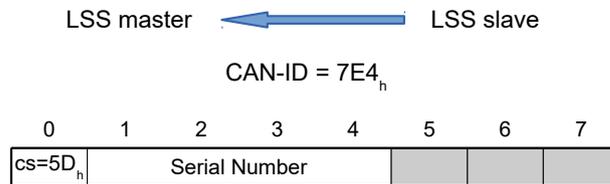
Byte 0 : CS (Command Specifier)

Wert = "5D_h"

Bytes 1-7 :

reserviert (=0_h)

Der LSS-Slave sendet seine Serial Number zurück:



Byte 0 : CS (Command Specifier)

Wert = "5D_h"

Bytes 1-4: Serial Number

Serial Number: siehe 1018_h:04_h

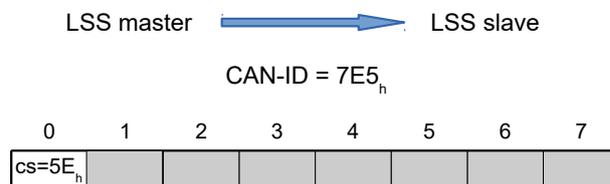
Bytes 5-7 :

reserviert (=0_h)

[Inquire node-ID service](#)

Mit diesem Dienst fragt der LSS-Master die Node-ID eines Slaves ab.

Der LSS-Master fragt die Node-ID ab:



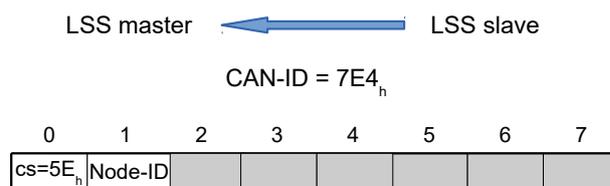
Byte 0 : CS (Command Specifier)

Wert = "5E_h"

Bytes 1-7 :

reserviert (=0_h)

Der LSS-Slave antwortet mit seiner Node-ID:



Byte 0 : CS (Command Specifier)

Wert = "5E_h"

Byte 1: Node-ID

Node-ID des *LSS-Slaves*

Bytes 2-7 :

reserviert (=0_h)

Identification services

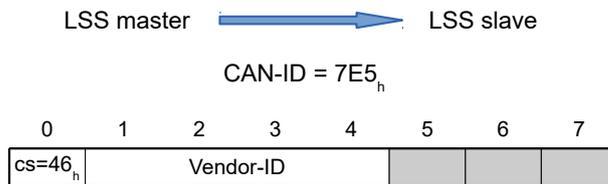
Mit diesen Diensten kann der *LSS-Master* die *LSS-Slaves* anhand ihrer *LSS-Adresse* sich identifizieren lassen.

LSS identify remote slave service

Mit diesem Dienst fordert der *LSS-Master* die *LSS-Slaves* mit der (oder Teilen der) entsprechenden *LSS-Adresse* an, sich mit dem Dienst **LSS identify slave service** zu identifizieren.

Dabei kann ein Bereich für die *Revision Number* und *Serial Number* definiert werden. Alle *LSS-Slaves*, deren Nummern sich im jeweiligen Bereich befinden, müssen sich identifizieren. Es ist die Aufgabe des *LSS-Masters*, den Bereich so zu verkleinern, dass schlussendlich nur ein *LSS-Slave* antwortet.

1. Der *LSS-Master* definiert die *Vendor-ID* der *LSS-Slaves*, die sich identifizieren sollen:



Byte 0 : CS (Command Specifier)

Wert = "46_h"

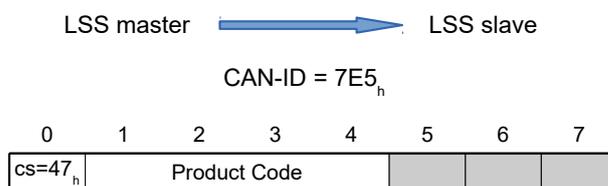
Bytes 1-4: Vendor-ID

Vendor-ID: siehe 1018_h:01_h

Bytes 5-7 :

reserviert (=0_h)

2. Der *LSS-Master* definiert den *Product Code* der *LSS-Slaves*, die sich identifizieren sollen:



Byte 0 : CS (Command Specifier)

Wert = "47_h"

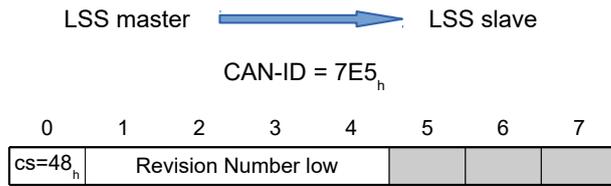
Bytes 1-4: Product Code

Product Code: siehe 1018_h:02_h

Bytes 5-7 :

reserviert (=0_h)

3. Der *LSS-Master* definiert die niedrigste und höchste *Revision Number* eines Bereichs. Alle *LSS-Slaves*, deren *Revision Number* innerhalb dieses Bereichs ist, sollen sich identifizieren:



Byte 0 : CS (Command Specifier)

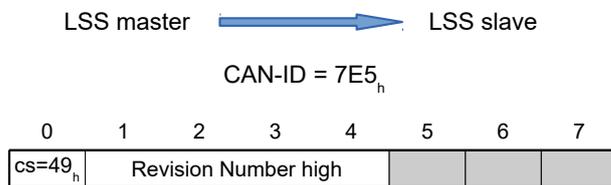
Wert = "48_h"

Bytes 1-4: Revision Number low

niedrigste *Revision Number* des Bereichs: siehe 1018_h:03_h

Bytes 5-7 :

reserviert (=0_h)



Byte 0 : CS (Command Specifier)

Wert = "49_h"

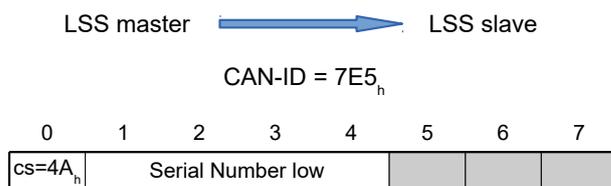
Bytes 1-4: Revision Number high

höchste *Revision Number* des Bereichs: siehe 1018_h:03_h

Bytes 5-7 :

reserviert (=0_h)

4. Der LSS-Master definiert die niedrigste und höchste *Serial Number* eines Bereichs. Alle LSS-Slaves, deren *Serial Number* innerhalb dieses Bereichs ist, sollen sich identifizieren:



Byte 0 : CS (Command Specifier)

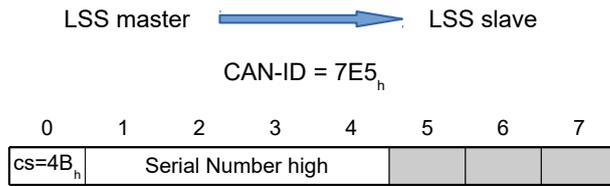
Wert = "4A_h"

Bytes 1-4: Serial Number low

niedrigste *Serial Number* des Bereichs: siehe 1018_h:04_h

Bytes 5-7 :

reserviert (=0_h)



Byte 0 : CS (Command Specifier)

Wert = "4B_h"

Bytes 1-4: Serial Number high

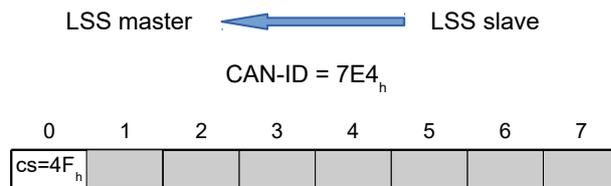
höchste *Serial Number* des Bereichs: siehe 1018_h:04_h

Bytes 5-7 :

reserviert (=0_h)

LSS identify slave service

Der *LSS-Slave*, dessen *LSS-Adresse* mit dem Dienst **LSS identify remote slave service** vom *LSS-Master* definiert wurde, identifiziert sich:



Byte 0 : CS (Command Specifier)

Wert = "4F_h"

Bytes 1-7 :

reserviert (=0_h)

8.3.4 Beispiel

Die Steuerung (*LSS-Slave*) wird mit folgenden Parametern ausgeliefert:

- Node-ID = 7F_h (=127_d)
- Baudrate = 1000 kBd

Die Parameter sollen wie folgt eingestellt werden:

- Node-ID = 05_h (=5_d)
- Baudrate = 125 kBd

Es wird angenommen, dass momentan nur ein *LSS-Slave* im Netzwerk vorhanden ist.

1. Der *LSS-Master* versetzt den *LSS-Slave* in den *Konfigurationsmodus* (siehe **Switch state global service**):

7E5 | 04 01 00 00 00 00 00 00

2. Der *LSS-Master* fragt die Node-ID des *LSS-Slaves* ab (siehe **Inquire node-ID service**):

7E5 | 5E 00 00 00 00 00 00 00

Der *LSS-Slave* antwortet mit seiner Node-ID:

7E4 | 5E 7F 00 00 00 00 00 00

3. Der *LSS-Master* setzt die Node-ID auf "05_h" (siehe **Configure node-ID service**):

7E5 | 11 05 00 00 00 00 00 00

Der *LSS-Slave* bestätigt (Error-Code=00_h):

7E4 | 11 00 00 00 00 00 00 00

4. Der *LSS-Master* setzt die Baudrate auf 125 kBd (*Table Index=4*) (siehe **Configure bit timing parameters service**):

7E5 | 13 00 04 00 00 00 00 00

Der *LSS-Slave* bestätigt (Error-Code=00_h):

7E4 | 13 00 00 00 00 00 00 00

5. Der *LSS-Master* sendet den Befehl zum Abspeichern der Änderungen (siehe **Store configuration service**):

7E5 | 17 00 00 00 00 00 00 00

6. Der *LSS-Master* versetzt den *LSS-Slave* in den *Wartemodus* (siehe **Switch state global service**):

7E5 | 04 00 00 00 00 00 00 00

Der *LSS-Slave* bestätigt (Error-Code=00_h):

7E4 | 17 00 00 00 00 00 00 00

7. Die neuen Parameter werden nach einem Neustart der Steuerung übernommen.

Die Steuerung meldet sich mit der Node-ID 5 und Baudrate 125KBd:

705 | 00

9 Modbus RTU

Modbus-Referenzen: www.modbus.org.

- *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3*, Date: 26.04.2014, Version: 1.1b3
- *MODBUS over Serial Line Specification and Implementation Guide V1.02*, Date: 20.12.2006, Version: 1.02

Die Steuerung lässt sich mittels Modbus RTU ansprechen. Die I/O Daten mit den z.B. vorkonfigurierten Antriebsgrößen (siehe **Prozessdatenobjekte (PDO)**) können mit den Standard Modbus-Funktionscodes durchgeführt werden. Um aber eigene I/O Daten zu konfigurieren, muss der Funktionscode 2Bh (CAN Encapsulation) vom Master unterstützt werden, damit die Parameter unabhängig vom Prozessabbild gelesen und beschrieben werden können.

Wenn der Master diesen Funktionscode nicht unterstützt, kann über das *Plug & Drive Studio* die Konfiguration des I/O Abbildes durchgeführt und gespeichert werden sodass der Master dann über die Standard-Modbus-Funktionscodes auf die Daten zugreifen kann.

9.1 RS-485

Die elektrische Schnittstelle "Two-Wire Modbus Interface" in Übereinstimmung mit dem Standard EIA/TIA-485 (RS-485) wird von der Steuerung unterstützt.

9.2 Modbus Modicon-Notation bei SPS

Viele SPS verwenden das Modicon-Adressierungsmodell. Im Modbus Standard kommt diese Notation nicht vor.

Folgende Adress-Notation ist bei Nanotec Steuerungen relevant:

- Input Register 30001 - 39999 wird auf Modbus Telegram Adresse 0 (0_h) - 9998 (270E_h) gemappt.
- Holding Register 40001- 49999 wird auf Modbus Telegram Adresse 0 (0_h) - 9998 (270E_h) gemappt.



Hinweis

Wenn im Handbuch von Modbus-Adressen gesprochen wird, müssen evtl. in der SPS die Register-Adressen nach *Modicon-Notation* eingesetzt werden.

9.3 Allgemeines

Modbus ist generell Big-Endian basiert.

Die einzigen Ausnahmen bilden dabei die Kommandos mit den Funktionscodes 43 (2B_h) , 101 (65_h) und 102 (66_h) welche auf CANopen basieren. Für die Datenwerte dieser Kommandos gilt das Little-Endian Format. Die restliche Modbus Nachricht ist hingegen nach wie vor Big-Endian basiert.

Beispiel

Kommando 2B_h: Mit diesem Kommando wird der Wert 12345678_h in das Objekt 0123_h (existiert nicht) geschrieben:

SA	FC	Daten	CRC
05	2B	0D 01 00 01 23 01 00 00 00 00 04 78 56 34 12	67 35

SA

Slave-Adresse

FC

Funktionscode

Daten

Datenbereich, Decodierung ist abhängig vom benutzen Funktionscode

CRC

Cyclic redundancy check

9.4 Funktionscodes

Die folgenden "Funktionscodes" werden unterstützt:

	Name	Funktionscode	Unterfunktionscode
Datenzugriff (16-bit)	Read Holding Registers	03 (03 _h)	
	Read Input Register	04 (04 _h)	
	Write Single Register	06 (06 _h)	
	Write Multiple Registers	22 (16 _h)	
	Read/Write Multiple Registers	23 (17 _h)	
Diagnose	Clear Counters and Diagnostic Register	08 (08 _h)	10 (0A _h)
	Return Bus Message Count	08 (08 _h)	11 (0B _h)
	Return Bus Communication Error Count	08 (08 _h)	12 (0C _h)
	Return Bus Exception Error Count	08 (08 _h)	13 (0D _h)
	Return Server Message Count	08 (08 _h)	14 (0E _h)
	Return Server No Response Count	08 (08 _h)	15 (0F _h)
	Return Server NAK Count	08 (08 _h)	16 (10 _h)
	Return Server Busy Count	08 (08 _h)	17 (11 _h)
	Return Bus Character Overrun Count	08 (08 _h)	18 (12 _h)
Sonstiges	Encapsulated Interface Transport	43 (2B _h)	13 (0D _h)
	Read complete object dictionary start	101 (65 _h)	85 (55 _h)
	Read complete object dictionary next	101 (65 _h)	170 (AA _h)
	Read complete array or record start	102 (66 _h)	85 (55 _h)
	Read complete array or record next	102 (66 _h)	170 (AA _h)

9.5 Funktioncode-Beschreibungen

9.5.1 FC 3 (03_h) Read Input Registers / FC 4 (04_h) Read Holding Registers

Mit diesem Funktionscode können ein 16-Bit-Wert oder mehrere 16-Bit-Werte ausgelesen werden. Die Funktion kann auf die NanoJ-Objekte (siehe **NanoJ-Objekte**) oder Prozessdatenobjekte (min. 4 Byte Ausrichtung, siehe **Prozessdatenobjekte (PDO)**) angewendet werden.

Request		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	03 _h / 04 _h

Request		
Name	Länge	Wert
Startadresse	2 Bytes	0000 _h bis FFFF _h
Anzahl der Register	2 Bytes	1 bis (7D _h)
CRC	2 Bytes	

Response ("M" entspricht der Anzahl der zu lesenden Register)		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	03 _h / 04 _h
Anzahl Bytes	1 Byte	2 * M
Registerwert	2 Bytes	
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	83 _h / 84 _h
Ausnahmecode	1 Byte	01, 02, 03 oder 04
CRC	2 Bytes	

Beispiel

Nachfolgend ein Beispiel eines Lese-Request und Response des Registers 5000 (1388_h) und des folgenden Registers (2 Register):

Request

SA	FC	Daten	CRC
05	03	13 88 00 02	41 21

Response

SA	FC	Daten	CRC
05	03	04 02 40 00 00	41 21

9.5.2 FC 6 (06_h) Write Single Register

Mit diesem Funktionscode kann ein einzelner 16-Bit-Wert geschrieben werden. Die Funktion kann auf Prozessdatenobjekte (siehe **Prozessdatenobjekte (PDO)**) angewendet werden.

Request		
Name	Länge	Wert
Slave-Adresse	1 Byte	

Request		
Name	Länge	Wert
Funktionscode	1 Byte	06 _h
Registeradresse	2 Bytes	0000 _h bis FFFF _h
Registerwert	2 Bytes	0000 _h bis FFFF _h
CRC	2 Bytes	

Response		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	06 _h
Registeradresse	2 Bytes	0000 _h bis FFFF _h
Registerwert	2 Bytes	0000 _h bis FFFF _h
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	86 _h
Ausnahmecode	1 Byte	01, 02, 03 oder 04
CRC	2 Bytes	

Beispiel

Nachfolgend ein Beispiel eines Write-Request und Response in das Register 6000 (1770_h) mit dem Wert "0001_h":

Request

SA	FC	Daten	CRC
05	06	17 70 00 01	4D E1

Response

SA	FC	Daten	CRC
05	06	17 70 00 01	4D E1

9.5.3 FC 16 (10_h) Write Multiple Registers

Mit diesem Funktionscode können ein einzelner 16-Bit-Wert oder mehrere 16-Bit-Werte geschrieben werden. Die Funktion kann auf NanoJ-Objekte (siehe **NanoJ-Objekte**) oder Prozessdatenobjekte (siehe **Prozessdatenobjekte (PDO)**) angewendet werden.

Request ("N" ist die Anzahl der zu schreibenden Register)		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	10 _h
Startadresse	2 Bytes	0000 _h bis FFFF _h
Anzahl der Register	2 Bytes	0001 _h bis 007B _h
Anzahl Bytes	1 Byte	2 * N
Registerwert	N * 2 Bytes	
CRC	2 Bytes	

Response		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	10 _h
Startadresse	2 Bytes	0000 _h bis FFFF _h
Anzahl der Register	2 Bytes	0001 _h bis 007B _h
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	90 _h
Ausnahmecode	1 Byte	01, 02, 03 oder 04
CRC	2 Bytes	

Beispiel

Nachfolgend ein Beispiel eines Mehrfach-Schreibens der Werte "0102_h" und "0304_h" startend ab Registeradresse 6000 (1770_h), Anzahl der Register ist 2, Länge der Daten 4:

Request

SA	FC	Daten	CRC
05	10	17 70 00 02 04 01 02 03 04	AB 44

Response

SA	FC	Daten	CRC
05	10	17 70 00 02	44 23

9.5.4 FC 17 (11_h) Report Server ID

Mit diesem Funktionscode kann man die Beschreibung des Typs, der gegenwärtigen Status und andere Informationen des Geräts auslesen.

Request		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	11 _h
CRC	2 Bytes	

Response		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	03 _h
Anzahl Bytes	1 Byte	01 _h
Run Indicator Status	1 Byte	00 _h = OFF, FF _h = ON
Zusatzdaten		
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	91 _h
Ausnahmecode	1 Byte	01 oder 04
CRC	2 Bytes	

Beispiel

Nachfolgend ein Beispiel eines Request/Response für ID und Status:

Request

SA	FC	CRC
05	11	C2 EC

Response

SA	FC	Daten	CRC
05	11	02 05 FF	0F EC

9.5.5 FC 23 (17_h) Read/Write Multiple registers

Mit diesem Funktionscode können ein einzelner 16-Bit-Wert oder mehrere 16-Bit-Werte gleichzeitig gelesen und geschrieben werden. Die Funktion kann auf NanoJ-Objekte (siehe **NanoJ-Objekte**) oder Prozessdatenobjekte (siehe **Prozessdatenobjekte (PDO)**) angewendet werden.

Request ("N" ist die Anzahl der zu lesenden Register):		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	17 _h
Lesen: Startadresse	2 Bytes	0000 _h bis FFFF _h
Lesen: Anzahl Register	2 Bytes	0001 _h bis 0079 _h
Schreiben: Startadresse	2 Bytes	0000 _h bis FFFF _h
Schreiben: Anzahl Register	2 Bytes	0001 _h bis 0079 _h
Schreiben: Anzahl Bytes	1 Byte	2 * N
Schreiben: Registerwert	N * 2 Bytes	
CRC	2 Bytes	

Response ("M" entspricht der Anzahl der zu schreibenden Bytes):		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	17 _h
Anzahl Bytes	1 Byte	2 * M
Gelesene Register	M * 2 Bytes	
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	97 _h
Ausnahmecode	1 Byte	01, 02, 03 oder 04
CRC	2 Bytes	

Beispiel

Nachfolgend ein Beispiel für das Lesen von zwei Registern ab Register 5000 (1388_h) und für das Schreiben von zwei Registern ab Register 6000 (1770_h) mit 4 Bytes und den Daten "0102_h" und "0304_h":

Request

SA	FC	Daten	CRC
05	17	13 88 00 02 17 70 00 02 04 01 02 03 04	56 6A

Response

SA	FC	Daten	CRC
05	17	04 02 40 00 00	0F EC

9.5.6 FC 8 (08_h) Diagnostics

Der Modbus-Funktionscode FC08 bietet eine Menge an Tests zum Überprüfen des Kommunikationssystems zwischen Client und Server oder zum Überprüfen verschiedener interner Fehlerzustände innerhalb des Servers.

Diese Funktion verwendet einen zwei Byte großen Unterfunktionscode im Request, um den Typen des Tests zu definieren. Der Server wiederholt in einer normalen Response beides, den Funktions- und den Unterfunktionscode. Einige der Diagnosen enthalten Daten des Gerätes im Datenfeld der normalen Antwort.

Request:

Name	Länge	Wert
Funktionscode	1 Byte	08 _h
Unterfunktionscode	2 Bytes	
Data	N x 2 Bytes	

Response:

Name	Länge	Wert
Funktionscode	1 Byte	08 _h
Unterfunktionscode	2 Bytes	
Data	N x 2 Bytes	

Fehler:

Name	Länge	Wert
Funktionscode	1 Byte	88 _h
Ausnahmecode	1 Bytes	01 oder 03 oder 04

FC 8.10 (08_h.0A_h) Clear Counters and Diagnostic Register

Das Ziel dieser Anfrage ist, alle Zähler und Diagnose-Register zurückzusetzen. Zähler werden auch beim Einschalten der Steuerung zurückgesetzt.

Unterfunktion	Datenbereich	
	Request	Resonse
00 _h 0A _h	00 _h 00 _h	Echo der Anfragedaten

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 0A 00 00	56 6A

Response

SA	FC	Daten	CRC
05	08	00 0A 00 00	C1 8D

FC 8.11 (08_h-0B_h) Return Bus Message Count

Der Datenbereich der Antwort gibt die Anzahl der Nachrichten zurück, welche seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung am Kommunikationssystem erkannt worden sind.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0B _h	00 _h 00 _h	Total Message Count

FC 8.12 (08_h-0C_h) Return Bus Communication Error Count

Der Datenbereich der Antwort gibt die Anzahl der CRC Fehler seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung zurück.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0C _h	00 _h 00 _h	CRC Error Count

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 0C 00 00	21 8C

Response

SA	FC	Daten	CRC
05	08	00 0C 00 00	21 8C

FC 8.13 (08_h-0D_h) Return Bus Exception Error Count

Der Datenbereich der Antwort gibt die Anzahl der Modbus Ausnahmen seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung zurück.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0D _h	00 _h 00 _h	Exception Error Count

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 0D 00 00	70 4C

Response

SA	FC	Daten	CRC
05	08	00 0D 00 00	70 4C

FC 8.14 (08_h.0E_h) Return Server Message Count

Der Datenbereich der Antwort gibt die Anzahl der an das Gerät gerichteten und Broadcast-Nachrichten zurück, die von der Steuerung verarbeitet wurden. Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0E _h	00 _h 00 _h	Server Message Count

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 0E 00 00	80 4C

Response

SA	FC	Daten	CRC
05	08	00 0E 00 00	80 4C

FC 8.15 (08_h.0F_h) Return Server No Response Count

Der Datenbereich der Antwort gibt die Anzahl der an die Steuerung gerichteten Nachrichten zurück, für die keine Antwort zurückgesendet wurde (weder normale Antwort noch Ausnahme-Antwort). Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0F _h	00 _h 00 _h	No Response Count

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 0F 00 00	D1 8C

Response

SA	FC	Daten	CRC
05	08	00 0F 00 00	D1 8C

FC 8.16 (08_h-10_h) Return Server NAK Count

Der Datenbereich der Antwort gibt die Anzahl der Nachrichten zurück, für die eine "Negative Acknowledge (NAK)"-Ausnahme-Antwort zurückgesendet wurde. Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 10 _h	00 _h 00 _h	Server NAK Count

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 10 00 00	E0 4A

Response

SA	FC	Daten	CRC
05	08	00 10 00 00	E0 4A

FC 8.17 (08_h-11_h) Return Server Busy Count

Der Datenbereich der Antwort gibt die Anzahl der Nachrichten zurück, für die eine "Server Device Busy"-Ausnahme-Antwort zurückgesendet wurde. Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 11 _h	00 _h 00 _h	Server NAK Count

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 11 00 00	B1 8A

Response

SA	FC	Daten	CRC
05	08	00 11 00 00	B1 8A

FC 8.18 (08_h,12_h) Return Bus Character Overrun Count

Der Datenbereich der Antwort gibt die Anzahl der an die Steuerung gerichteten Nachrichten zurück, die aufgrund einem Zeichenüberlauf nicht verarbeitet werden konnten. Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung. Ein Zeichenüberlauf entsteht dadurch, dass Zeichen schneller an der Steuerung ankommen, als sie gespeichert werden können, oder durch den Verlust eines Zeichens aufgrund eines Hardwarefehlers.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 12 _h	00 _h 00 _h	Server Character Overrun Count

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 12 00 00	41 8A

Response

SA	FC	Daten	CRC
05	08	00 12 00 00	41 8A

9.5.7 FC 43 (2B_h) Encapsulated Interface Transport

Diese Funktion ermöglicht einen einfachen Zugriff auf das CANopen-Objektverzeichnis. Weitere Details können in den folgenden Dokumentationen entnommen werden:

1. MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3, Date: 26.04.2014, Version: 1.1b3

2. CiA 309 Draft Standard Proposal - Access from other networks - Part 2: Modbus/TCP mapping
 V1.3, Date: 30.07.2015, Version: 1.3



Hinweis

Für die Nachrichten des Encapsulated Interface-Transport gilt zum Teil eine andere Byte-Reihenfolge, siehe Kapitel **Allgemeines**.

Definition des Request und Response:

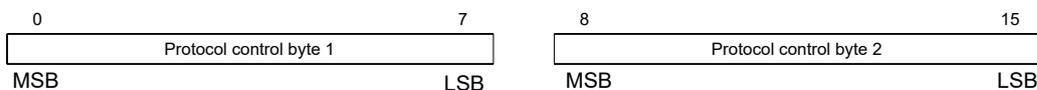
Name	Länge	Beispiel/Zahlenbereich
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h (43 _d)
MEI type	1 Byte	0D _h (13 _d)
Protokolloptionen Bereich	2 bis 5 Byte	
Adressen- und Datenbereich	N Bytes	
CRC	2 Bytes	

Protokolloptionen Bereich

Name	Länge	Beispiel/Zahlenbereich
Protokoll-Kontrolle	1 bis 2 Bytes	Siehe Beschreibung
Reserviert	1 Byte	Immer 0
(Optional) Zählerbyte	1 Byte	
(Optional) Netzwerk ID	1 Byte	
(Optional) Encodierte Daten	1 Byte	

Protokoll-Kontrolle:

Das Feld "Protokoll-Kontrolle" enthält die Merker, welche für die Kontrolle der Nachrichtenprotokolle benötigt werden. Die Bytes des Feldes "Protokoll Kontrolle" sind folgendermaßen definiert, falls der Merker "Verlängerung" gesetzt wurde (andernfalls entfällt das zweite Byte):



Das höchstwertige Bit (MSB) ist Bit 0 für "Protokoll-Kontrolle" Byte 1, und Bit 8 für "Protokoll Kontrolle" Byte 2. Das niedrigstwertige Bit (LSB) ist Bit 7 für "Protokoll Kontrolle" Byte 1, und Bit 15 für "Protokoll Kontrolle" Byte 2.

Bit	Name	Beschreibung
0	Merker "Verlängerung"	Dieses Bit wird genutzt, wenn das Objektverzeichnis Datenset größer ist, als in ein Modbus-Kommando passen würde. Das Datenset wird dann über mehrere Modbus-Nachrichten gestreckt, jede Nachricht enthält einen Teil des Datensets. "0" = Keine mehrfache Nachrichtentransaktion ("multiple message transaction") oder das Ende der mehrfachen

Bit	Name	Beschreibung
		Nachrichtentransaktion. "1" = Teil einer mehrfachen Nachrichtentransaktion.
1	Erweiterte Protokoll-Kontrolle	Länge der Protokoll-Kontrolle, der Wert "0" zeigt eine Länge von 1 Byte an, der Wert "1" eine Länge von 2 Byte.
2	Zähler Byte Option	Dieses Bit ist auf "1" um anzuzeigen, dass das Feld "counter byte" in dieser Nachricht genutzt wird. Ist dieses Bit auf "0" gesetzt, existiert das Feld "counter byte" nicht in dieser Nachricht.
3 und 4	Reserviert	0
5	Network ID Option	Nicht unterstützt, muss "0" sein.
6	Encodierte Datenoption	Nicht unterstützt, muss "0" sein.
7	Zugriffsmerker	Dieses Bit zeigt die Zugriffsmethode des angeforderten Kommandos an. "0" = lesen, "1" = schreiben.
8 to 15	Reserviert	0

Adressen- und Datenbereich

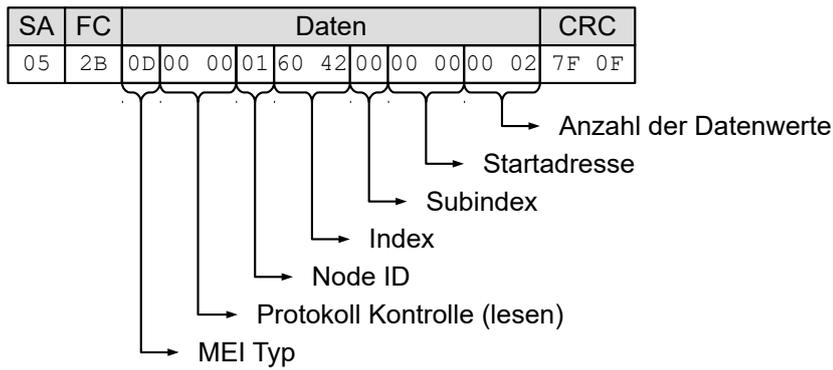
Der Adressen- und Datenbereich ist in der folgenden Tabelle definiert:

Name	Bytegröße und Bytereihenfolge	Beispiel / Bereich
Node-ID	1 Byte	01 _h bis 7F _h
Index	1 Byte, high	0000 _h bis FFFF _h
	1 Byte, low	
Subindex	1 Byte	00 _h bis FF _h
Startadresse	1 Byte, high	0000 _h bis FFFF _h
	1 Byte, low	
Anzahl der Datenwerte	1 Byte, high	0000 _h bis 00FD _h
	1 Byte, low	
Schreib-/Lesedaten	n Byte	Die Daten sind codiert wie in Kapitel Allgemeines beschrieben.

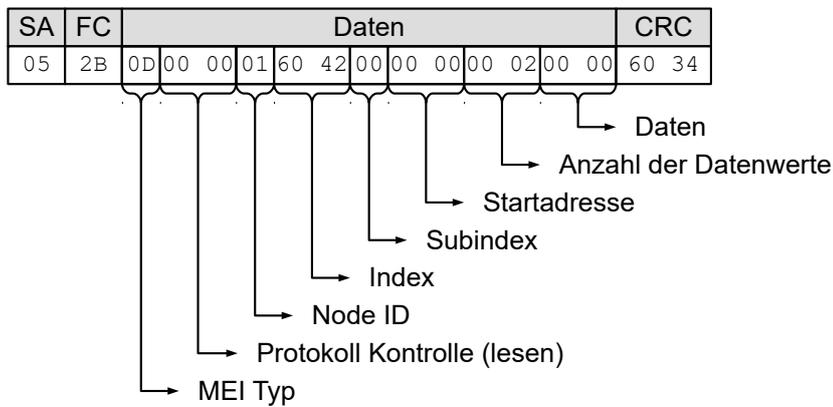
Beispiel:

Um das Objekt 6042_h:00_h auszulesen (16 Bit-Wert), muss folgende Nachricht vom Master verschickt werden (alle Werte sind in hexadezimaler Notation, die Slave-Id der Steuerung ist "5").

Request



Response



Als zusätzliches Beispiel nachfolgend eine Sequenz an Modbus-Nachrichten vom Master zum Slave, um den Motor im "Velocity" Modus sich drehen zu lassen:

Setze 6060 = "02_h" (velocity mode)

Request

SA	FC	Daten										CRC			
05	2B	0D	01	00	01	60	60	00	00	00	00	01	02	C9	2F

Response

SA	FC	Daten										CRC			
05	2B	0D	01	00	01	60	60	00	00	00	00	00	00	A9	89

Setze 2031 = 03E8_h" (1000 mA)

Request

SA	FC	Daten										CRC						
05	2B	0D	01	00	01	20	31	00	00	00	00	04	E8	03	00	00	C3	53

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 20 31 00 00 00 00 00	E5 CC

Setze 6040 = "00_h"

Request

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 00 00	1C 2E

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00 00	AE E9

Setze 6040 = "80_h"

Request

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 80 00	7D EE

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00 00	AE E9

Setze 6040 = "06_h"

Request

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 06 00	1F 8E

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00 00	AE E9

Setze 6040 = "07_h"

Request

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 07 00	1E 1E

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00 00	AE E9

Setze 6040 = "0F_h"

Request

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 0F 00	19 DE

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00 00	AE E9

Nachfolgend zwei Beispiele zum Lesen eines Objektes:

Lese 6041_h:00_h

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 41 00 00 00 00 02	7F 3C

Response

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 41 00 00 00 00 02 37 06	B6 13

Lese 6061_h:00_h

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 61 00 00 00 00 01	38 5D

Response

SI	FC	Daten	CRC
05	2B	0D 00 00 01 60 61 00 00 00 00 01 00	5C D2

Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	

Name	Länge	Beispielwert
Funktionscode	1 Byte	2B _h +80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception Länge	2 Bytes	6
MEI type	1 Byte	0D _h
Exception code	1 Byte	CE _h
Fehlercode	4 Bytes	CANopen-Fehlercode, siehe nachfolgende Tabelle
CRC	2 Bytes	

CANopen-Fehlercode	Beschreibung
FFFF0000 _h	<i>Abort no error</i>
FFFF1003 _h	Service wird nicht unterstützt
FFFF1004 _h	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>
FFFF0003 _h	Unbekanntes oder nicht valides Kommando
FFFF0008 _h	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E _h	Allgemeiner Fehler im Parameter
FFFF0011 _h	Länge des Parameters falsch
FFFF0012 _h	Länge des Parameters zu groß
FFFF0013 _h	Länge des Parameters zu klein
FFFF0015 _h	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 _h	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 _h	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 _h	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 _h	Allgemeiner Fehler
FFFF001E _h	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 _h	Ungültige Sequenz der Nachrichten (z. B. wenn der Wert des <i>Counter-Bytes</i> entsprechend dem vorherigen Request oder Response nicht stimmt)

Im Falle, dass das nicht unterstützte Kontrolloptions-Bit gesetzt ist, wird folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h +80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D _h
Exception code	1 Byte	AE _h
Supported protocol control	1 oder 2 Bytes	Siehe nachfolgende Tabelle

Name	Länge	Beispielwert
CRC	2 Bytes	

Bit	Name	Beschreibung
0	Merker "Verlängerung"	Dieses Bit wird genutzt, wenn das Objektverzeichnis Datenset größer ist, als in ein Modbus-Kommando passen würde. Das Datenset wird dann über mehrere Modbus-Nachrichten gestreckt, jede Nachricht enthält einen Teil des Datensets. "0" = Keine mehrfache Nachrichtentransaktion ("multiple message transaction") oder das Ende der mehrfachen Nachrichtentransaktion. "1" = Teil einer mehrfachen Nachrichtentransaktion.
1	Erweiterte Protokoll-Kontrolle	Länge der Protokoll-Kontrolle, der Wert "0" zeigt eine Länge von 1 Byte an, der Wert "1" eine Länge von 2 Byte.
2	Zähler Byte Option	Dieses Bit ist auf "1" um anzuzeigen, dass das Feld "counter byte" in dieser Nachricht genutzt wird. Ist dieses Bit auf "0" gesetzt, existiert das Feld "counter byte" nicht in dieser Nachricht.
3 und 4	Reserviert	0
5	Network ID Option	Nicht unterstützt, muss "0" sein.
6	Encodierte Datenoption	Nicht unterstützt, muss "0" sein.
7	Zugriffsmerker	Dieses Bit zeigt die Zugriffsmethode des angeforderten Kommandos an. "0" = lesen, "1" = schreiben.
8 to 15	Reserviert	0

Das nachfolgende Beispiel zeigt einen Fehler im Falle eines fehlerhaften Request. Der Request liest das 6061_{h} :00 mit der Länge von 2 Byte, das Objekt ist aber nur 1 Byte groß:

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 60 00 00 00 00 02	79 8D

Response

SA	FC	Daten	CRC
05	2B	FF 00 06 0D CE 12 00 07 06	AC 3C

9.5.8 FC 101 (65_h) Read complete object dictionary

Dieser Funktionscode wird zum Auslesen des gesamten Objektverzeichnisses verwendet.

Um das Auslesen des Objektverzeichnisses zu starten oder neu zu starten, muss der Unterfunktionscode 55_{h} versendet werden. Dieser Code setzt das Auslesen des Objektverzeichnisses auf das Objekt 0000_{h} zurück. Alle nachfolgenden Objektverzeichnis-Frames müssen dann den Unterfunktionscode AA_{h} enthalten. Zum Ende, wenn alle Objekte ausgelesen wurden, wird eine "Error Response" generiert mit dem Abort-Code "No data available".

Das Format jedes "Objekt lesen" ist folgendermaßen:

Request:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	65 _h
Unterfunktionscode	1 Byte	55 _h oder AA _h
Länge der Daten	1 Byte	00 _h
CRC	2 Bytes	

Response:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	65 _h
Funktionscode	1 Byte	
Unterfunktionscode	1 Byte	
Länge der Daten	1 Byte	
n mal "Objektverzeichnis-Frame"	1 - 252 Bytes	
CRC	2 Bytes	

Ein Objektverzeichnis-Frame besteht aus den folgenden Bytes:

Name	Länge	Wert / Bemerkung
Index Low Byte	1 Byte	
Index High Byte	1 Byte	
Subindex	1 Byte	
Anzahl der Bytes	1 Byte	Anzahl m der validen Daten im Datenfeld
Daten Byte	m-1 Byte	

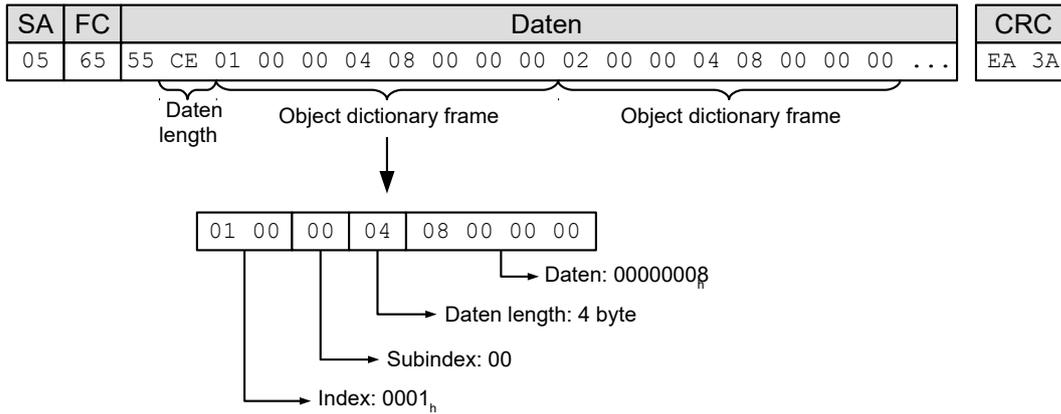
Beispiel

Alle folgenden Zahlenwerte sind in Hexadezimal notiert. Die Adresse des Slaves ist "5".

Start des Auslesens des Objektverzeichnisses mit dem Request:

SA	FC	Daten	CRC
05	65	55 00	2F A7

Die Response ist:



Den nächsten Teil des Objektverzeichnisses auslesen mit dem Request:

SA	FC	Daten	CRC
05	65	AA 00	6E 57

Die Response ist:

SA	FC	Daten	CRC
05	65	AA CD 21 00 0A 02 07 00 21 00 0B 02 07 00 21 00 0C 02 ...	NN NN

Wiederholen des Auslesens des Objektverzeichnisses mit dem vorherigen Request, bis die Response ein Fehler ist:

SA	FC	Daten	CRC
05	E5	0D	EA 94

Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h +80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception Länge	2 Bytes	6
MEI type	1 Byte	0D _h
Exception code	1 Byte	CE _h
Fehlercode	4 Bytes	CANopen-Fehlercode, siehe nachfolgende Tabelle
CRC	2 Bytes	

CANopen-Fehlercode	Beschreibung
FFFF0000 _h	Abort no error
FFFF1003 _h	Service wird nicht unterstützt
FFFF1004 _h	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>

CANopen-Fehlercode	Beschreibung
FFFF0003 _h	Unbekanntes oder nicht valides Kommando
FFFF0008 _h	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E _h	Allgemeiner Fehler im Parameter
FFFF0011 _h	Länge des Parameters falsch
FFFF0012 _h	Länge des Parameters zu groß
FFFF0013 _h	Länge des Parameters zu klein
FFFF0015 _h	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 _h	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 _h	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 _h	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 _h	Allgemeiner Fehler
FFFF001E _h	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 _h	Ungültige Sequenz der Nachrichten (z. B. wenn der Wert des <i>Counter-Bytes</i> entsprechend dem vorherigen Request oder Response nicht stimmt)

Im Falle, dass das nicht unterstützte Kontrolloptions-Bit gesetzt ist, wird folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h + 80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D _h
Exception code	1 Byte	AE _h
Supported protocol control	1 oder 2 Bytes	Siehe nachfolgende Tabelle
CRC	2 Bytes	

Bit	Name	Beschreibung
0	Merker "Verlängerung"	Dieses Bit wird genutzt, wenn das Objektverzeichnis Datenset größer ist, als in ein Modbus-Kommando passen würde. Das Datenset wird dann über mehrere Modbus-Nachrichten gestreckt, jede Nachricht enthält einen Teil des Datensets. "0" = Keine mehrfache Nachrichtentransaktion ("multiple message transaction") oder das Ende der mehrfachen Nachrichtentransaktion. "1" = Teil einer mehrfachen Nachrichtentransaktion.
1	Erweiterte Protokoll-Kontrolle	Länge der Protokoll-Kontrolle, der Wert "0" zeigt eine Länge von 1 Byte an, der Wert "1" eine Länge von 2 Byte.

Bit	Name	Beschreibung
2	Zähler Byte Option	Dieses Bit ist auf "1" um anzuzeigen, dass das Feld "counter byte" in dieser Nachricht genutzt wird. Ist dieses Bit auf "0" gesetzt, existiert das Feld "counter byte" nicht in dieser Nachricht.
3 und 4	Reserviert	0
5	Network ID Option	Nicht unterstützt, muss "0" sein.
6	Encodierte Datenoption	Nicht unterstützt, muss "0" sein.
7	Zugriffsmarker	Dieses Bit zeigt die Zugriffsmethode des angeforderten Kommandos an. "0" = lesen, "1" = schreiben.
8 to 15	Reserviert	0

Das nachfolgende Beispiel zeigt einen Fehler im Falle eines fehlerhaften Request. Der Request liest das **6061_h**:00 mit der Länge von 2 Byte, das Objekt ist aber nur 1 Byte groß:

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 60 00 00 00 00 02	79 8D

Response

SA	FC	Daten	CRC
05	2B	FF 00 06 0D CE 12 00 07 06	AC 3C

9.5.9 FC 102 (66_h) Read complete array or record

Dieser Funktionscode wird zum Auslesen eines gesamten Arrays oder Records vom Objektverzeichnis verwendet.

Um das Auslesen des Arrays zu starten oder neu zu starten, muss der Unterfunktionscode **55_h** versendet werden. Dieser Code setzt das Auslesen auf das Objekt mit Subindex **00_h** zurück. Alle nachfolgenden Requests müssen dann den Unterfunktionscode **AA_h** enthalten. Zum Ende, wenn alle Objekte ausgelesen wurden, wird eine "Error Response" generiert.

Das Format jedes "Objekt lesen" ist folgendermaßen:

Request:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	66 _h
Unterfunktionscode	1 Byte	55 _h oder AA _h
Länge der Daten	1 Byte	00 _h
Index des zu lesenden Arrays	2 Bytes	
CRC	2 Bytes	

Response:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	65 _h

Name	Länge	Wert / Bemerkung
Funktionscode	1 Byte	
Unterfunktionscode	1 Byte	
Länge der Daten	1 Byte	
n mal Objektverzeichnis-Frame	1 - 252 Bytes	
CRC	2 Bytes	

Ein Objektverzeichnis-Frame besteht aus den folgenden Bytes:

Name	Wert / Bemerkung
Index Low Byte	1 Byte
Index High Byte	1 Byte
Subindex	1 Byte
Anzahl der Bytes	1 Byte Anzahl m der validen Daten im Datenfeld
Daten Byte	m-1 Byte

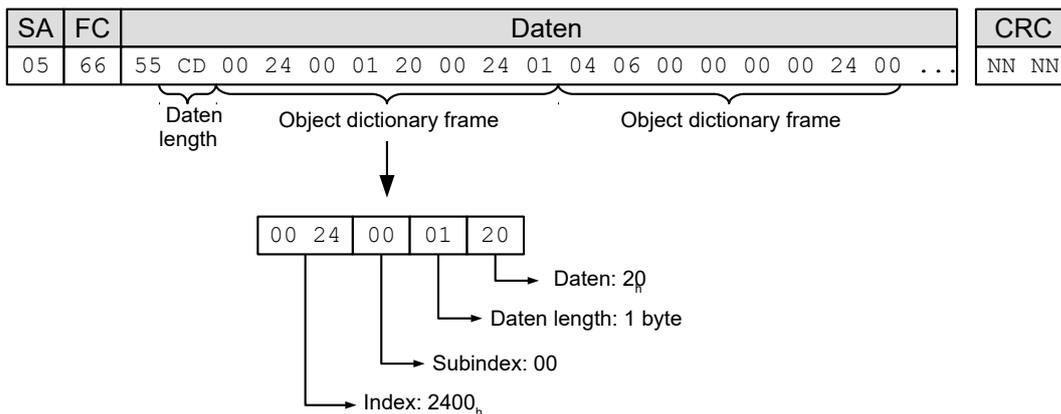
Beispiel

Alle folgenden Zahlenwerte sind in Hexadezimal notiert, der Index des zu lesenden Objektes ist 2400_h. Die Adresse des Slaves ist "5"_h.

Start des Auslesens des Arrays mit dem Request:

SA	FC	Daten	CRC
05	66	55 00 24 00	02 8A

Die Response ist:



Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h +80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)

Name	Länge	Beispielwert
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception Länge	2 Bytes	6
MEI type	1 Byte	0D _h
Exception code	1 Byte	CE _h
Fehlercode	4 Bytes	CANopen-Fehlercode, siehe nachfolgende Tabelle
CRC	2 Bytes	

CANopen-Fehlercode	Beschreibung
FFFF0000 _h	<i>Abort no error</i>
FFFF1003 _h	Service wird nicht unterstützt
FFFF1004 _h	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>
FFFF0003 _h	Unbekanntes oder nicht valides Kommando
FFFF0008 _h	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E _h	Allgemeiner Fehler im Parameter
FFFF0011 _h	Länge des Parameters falsch
FFFF0012 _h	Länge des Parameters zu groß
FFFF0013 _h	Länge des Parameters zu klein
FFFF0015 _h	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 _h	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 _h	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 _h	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 _h	Allgemeiner Fehler
FFFF001E _h	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 _h	Ungültige Sequenz der Nachrichten (z. B. wenn der Wert des <i>Counter-Bytes</i> entsprechend dem vorherigen Request oder Response nicht stimmt)

Im Falle, dass das nicht unterstützte Kontrolloptions-Bit gesetzt ist, wird folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h +80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D _h
Exception code	1 Byte	AE _h
Supported protocol control	1 oder 2 Bytes	Siehe nachfolgende Tabelle
CRC	2 Bytes	

Bit	Name	Beschreibung
0	Merker "Verlängerung"	Dieses Bit wird genutzt, wenn das Objektverzeichnis Datenset größer ist, als in ein Modbus-Kommando passen würde. Das Datenset wird dann über mehrere Modbus-Nachrichten gestreckt, jede Nachricht enthält einen Teil des Datensets. "0" = Keine mehrfache Nachrichtentransaktion ("multiple message transaction") oder das Ende der mehrfachen Nachrichtentransaktion. "1" = Teil einer mehrfachen Nachrichtentransaktion.
1	Erweiterte Protokoll-Kontrolle	Länge der Protokoll-Kontrolle, der Wert "0" zeigt eine Länge von 1 Byte an, der Wert "1" eine Länge von 2 Byte.
2	Zähler Byte Option	Dieses Bit ist auf "1" um anzuzeigen, dass das Feld "counter byte" in dieser Nachricht genutzt wird. Ist dieses Bit auf "0" gesetzt, existiert das Feld "counter byte" nicht in dieser Nachricht.
3 und 4	Reserviert	0
5	Network ID Option	Nicht unterstützt, muss "0" sein.
6	Encodierte Datenoption	Nicht unterstützt, muss "0" sein.
7	Zugriffsmarker	Dieses Bit zeigt die Zugriffsmethode des angeforderten Kommandos an. "0" = lesen, "1" = schreiben.
8 to 15	Reserviert	0

Das nachfolgende Beispiel zeigt einen Fehler im Falle eines fehlerhaften Request. Der Request liest das **6061_h:00** mit der Länge von 2 Byte, das Objekt ist aber nur 1 Byte groß:

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 60 00 00 00 00 02	79 8D

Response

SA	FC	Daten	CRC
05	2B	FF 00 06 0D CE 12 00 07 06	AC 3C

9.6 Prozessdatenobjekte (PDO)

Wie bei CANopen kann bei Modbus ein Prozessimage für Eingangs- und Ausgangsgrößen konfiguriert werden. Dieses Image beinhaltet nur noch Datenwerte einer oder mehrerer Objekte ohne Zusatzinformation wie Länge, Index oder Subindex. Damit lassen sich mittels einer Nachricht gleich mehrere Objekte lesen oder schreiben.

9.6.1 Konfiguration

Die Konfiguration des Image wird als "Mapping" bezeichnet und in folgenden Objekten geschrieben:

- 3502_h für das Modbus Rx (Master → Slave) PDO-Mapping
- 3602_h für das Modbus Tx (Slave → Master) PDO-Mapping

Beide Objekte beinhalten einen Array mit jeweils 16 Einträge. Der Subindex 00 gibt dabei die Anzahl der gültigen Einträge an.

Die Objekte 3502_h und 3602_h lassen sich mit Nachrichten mit dem Modbus-Funktionscode 2B_h beschreiben.

9.6.2 Übertragung

Die Daten werden aufeinander folgend ohne Lücke und Ausrichtung in die Nachricht geschrieben.

Wird ein Alignment (z.B. 16-Bit-Alignment) benötigt, kann man zusätzliche "Dummy-Objekte" mit in die Nachricht einbauen. Dummy-Objekte werden immer mit den Datenwert "0" übertragen. Diese Objekte sind in der nachfolgenden Tabelle abgedruckt.

Index	Datentyp
0002 _h	Vorzeichenbehaftete Ganzzahl (8 Bit)
0003 _h	Vorzeichenbehaftete Ganzzahl (16 Bit)
0004 _h	Vorzeichenbehaftete Ganzzahl (32 Bit)
0005 _h	Vorzeichenlose Ganzzahl (8 Bit)
0006 _h	Vorzeichenlose Ganzzahl (16 Bit)
0007 _h	Vorzeichenlose Ganzzahl (32 Bit)

Das Mapping ist wie folgt:

- Das PDO RX Image fängt an der Modbus-Register-Adresse 6000_d (1770_h) an.
- Das PDO TX Image fängt an der Modbus-Register-Adresse 5000_d (1388_h) an.

Der Zugriff kann mit Funktionscode 17_h lesend/schreibend gleichzeitig erfolgen oder mit den Kommandos 03_h, 04_h, 06_h, 10_h auf die jeweiligen RX/TX Images.

Beispiel

In dem Mapping sollen folgende Objekte eingestellt werden:

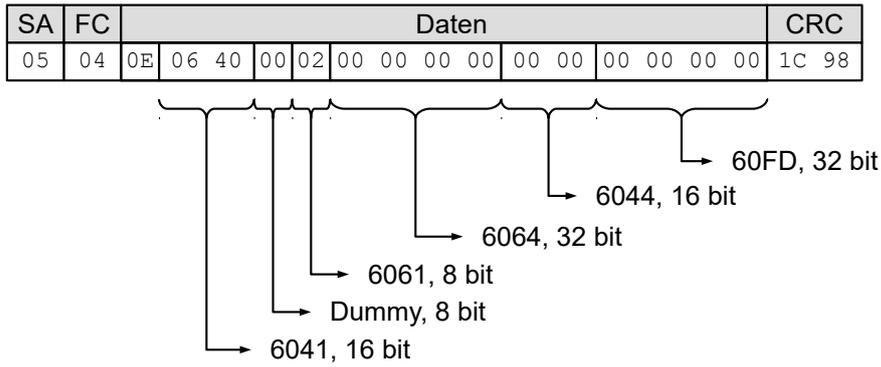
- 3602_h:00_h = "6_h" (6 Werte werden gemappt)
- 3602_h:01_h = "60410010_h" (das Objekt 6041_h:00_h, Länge 16 Bit wird gemappt)
- 3602_h:02_h = "00050008_h" (das Dummy-Objekt 0005_h:00_h, Länge 8 Bit wird gemappt)
- 3602_h:03_h = "60610008_h" (das Objekt 6061_h:00_h, Länge 8 Bit wird gemappt)
- 3602_h:04_h = "60640020_h" (das Objekt 6064_h:00_h, Länge 32 Bit wird gemappt)
- 3602_h:05_h = "60440010_h" (das Objekt 6044_h:00_h, Länge 16 Bit wird gemappt)
- 3602_h:06_h = "60FD0020_h" (das Objekt 60FD_h:00_h, Länge 32 Bit wird gemappt)

Nach dem Mapping für das Objekt 6061_h:00_h wird ein Dummy-Objekt eingefügt, damit das nachfolgende Objekt 6064_h:00_h auf 32 Bit ausgerichtet wird.

RX Nachricht: Der Master schickt an den Slave folgende Nachricht:

SA	FC	Daten	CRC
05	04	13 88 00 07	34 E2

TX Nachricht: Der Slave schickt an den Master folgende Antwort:



9.7 NanoJ-Objekte

Die NanoJ-Objekte **2400_h** NanoJ Input und **2500_h** (NanoJ Output) werden wie das Prozessimage auf Modbus-Register gemappt:

- **2500_h** mit 32 x 32 Bit Werten wird auf die Modbus Register Adresse ab 2000_d (BB8_h) gemappt und kann auf diese Weise nur gelesen werden.
- **2400_h** mit 32 x 32 Bit Werten wird auf die Modbus Register Adresse ab 3000_d (7D0_h) gemappt und kann auf diese Weise nur beschrieben werden.

Für den Zugriff können die Kommandos mit Funktionscode 03_h, 04_h, 10_h und 17_h verwendet werden. Es gilt die Einschränkung, dass die Adresse auf 32 Bit ausgerichtet (aligned) sein muss und auch bei einem Schreibvorgang immer mindestens 32 Bit geschrieben werden muss, damit die Daten konsistent sind.

Beispiel

Request: Der Master schickt an den Slave folgende Nachricht:

SA	FC	Daten																CRC						
05	17	07	D0	00	08	0B	B8	00	08	10	00	01	02	03	04	05	06	07	08	09	0A	0B	41	21
		0C	0D	0E	0F																			

Reply: Der Slave schickt an den Master folgende Antwort:

SA	FC	Daten																CRC							
05	17	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	50	9D

10 Programmierung mit NanoJ

NanoJ ist eine C- bzw. C++-nahe Programmiersprache. NanoJ ist in der Software *Plug & Drive Studio* integriert. Weiterführende Informationen finden Sie im Dokument *Plug & Drive Studio: Quick Start Guide* auf www.nanotec.de.

10.1 NanoJ-Programm

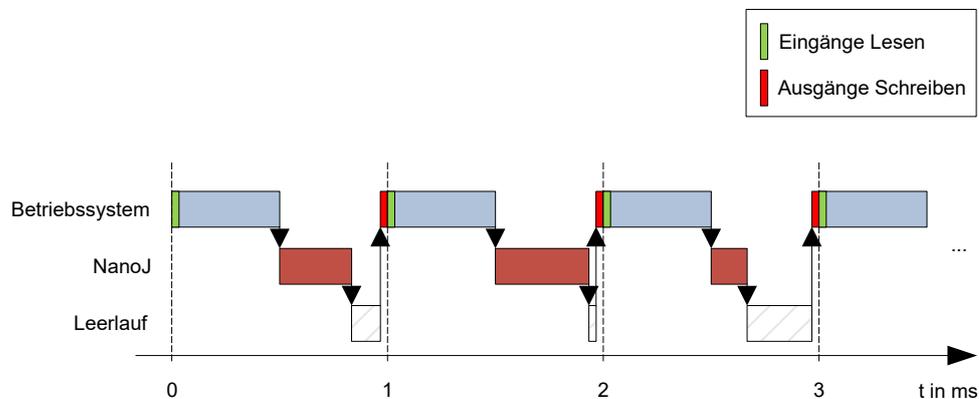
Ein *NanoJ-Programm* stellt eine geschützte Ausführungsumgebung innerhalb der Firmware zur Verfügung. In dieser kann der Anwender eigene Abläufe anlegen. Diese können dann Funktionen in der Steuerung auslösen, indem beispielsweise Einträge im Objektverzeichnis gelesen oder geschrieben werden.

Durch Verwendung von Schutzmechanismen wird verhindert, dass ein *NanoJ-Programm* die Firmware zum Absturz bringt. Im schlimmsten Fall wird die Ausführung mit einem im Objektverzeichnis hinterlegten Fehlercode abgebrochen.

Wenn das *NanoJ-Programm* auf die Steuerung geladen wurde, wird es nach dem Einschalten oder Neustarten der Steuerung automatisch ausgeführt.

10.1.1 Verfügbare Rechenzeit

Ein *NanoJ-Programm* erhält zyklisch im 1 ms-Takt Rechenzeit (siehe folgende Abbildung). Da durch Interrupts und Systemfunktionen der Firmware Rechenzeit verloren geht, stehen dem Benutzerprogramm (abhängig von Betriebsart und Anwendungsfall) nur ca. 30% ... 50% Rechenzeit zur Verfügung. In dieser Zeit muss das Benutzerprogramm den Zyklus durchlaufen und entweder beenden oder durch Aufruf der Funktion `yield()` die Rechenzeit abgeben. Bei Ersterem wird das Benutzerprogramm mit dem Beginn des nächsten 1 ms-Zyklus wieder neu gestartet, letzteres bewirkt eine Fortsetzung des Programms an dem der Funktion `yield()` nachfolgenden Befehl beim nächsten 1 ms-Zyklus.



Falls das *NanoJ-Programm* mehr als die ihm zugeteilte Zeit benötigt, wird es beendet und im Objektverzeichnis ein Fehlercode gesetzt.



Tipp

Bei der Entwicklung von Benutzerprogrammen ist speziell bei zeitintensiveren Aufgaben eine sorgfältige Überprüfung des Laufzeitverhaltens durchzuführen. So empfiehlt sich beispielsweise die Verwendung von Tabellen, anstatt einen Sinuswert über eine `sin` Funktion zu berechnen.



Hinweis

Sollte das *NanoJ-Programm* zu lange die Rechenzeit nicht abgeben, wird es vom Betriebssystem beendet. In diesem Fall wird in das Statusword bei Objekt 2301_h die Ziffer 4 eingetragen, im Fehlerregister bei Objekt 2302_h wird die Ziffer 5 (Timeout) notiert, siehe **2301h NanoJ Status** und **2302h NanoJ Error Code**.

10.1.2 Sandbox

Durch prozessorspezifische Eigenschaften wird eine sogenannte *Sandbox* generiert. Ein Benutzerprogramm in der Sandbox hat nur die Möglichkeit, auf speziell zugewiesene Speicherbereiche und Systemressourcen zuzugreifen. Beispielsweise wird ein Versuch, auf ein Prozessor-IO-Register direkt zu schreiben, mit einem *MPU Fault* quittiert und das Benutzerprogramm wird mit dem entsprechenden Fehlercode im Objektverzeichnis abgebrochen.

10.1.3 NanoJ-Programm - Kommunikationsmöglichkeiten

Ein *NanoJ-Programm* hat mehrere Möglichkeiten, mit der Steuerung zu kommunizieren:

- Lesen und Schreiben von OD-Werten per PDO-Mapping
- direktes Lesen und Schreiben von OD-Werten über Systemcalls
- Aufruf sonstiger Systemcalls (z. B. Debug-Ausgabe schreiben)

Über ein *PDO-Mapping* werden dem Benutzerprogramm OD-Werte in Form von Variablen zur Verfügung gestellt. Bevor ein Benutzerprogramm die 1 ms-Zeitscheibe erhält, werden dazu von der Firmware die Werte aus dem Objektverzeichnis in die Variablen des Benutzerprogramms übertragen. Sobald das Benutzerprogramm Rechenzeit erhält, kann es diese Variablen wie gewöhnliche C-Variablen manipulieren. Am Ende der Zeitscheibe werden letztendlich die neuen Werte von der Firmware wieder automatisch in die jeweiligen OD-Einträge kopiert.

Um die Performance zu optimieren, werden dabei drei Arten von Mappings definiert: Input, Output und Input/Output (In, Out, InOut).

- *Input Mappings* lassen sich nur lesen und werden nicht zurück ins Objektverzeichnis übertragen.
- *Output Mappings* lassen sich nur schreiben.
- *Input/Output Mappings* erlauben hingegen Lesen und Schreiben.

Die gesetzten Mappings können über die GUI bei den Objekten 2310_h, 2320_h, und 2330_h ausgelesen und überprüft werden. Für jedes Mapping sind maximal 16 Einträge erlaubt.

Über die Angabe der *Linker-Section* wird in *NanoJEasy* gesteuert, ob eine Variable im Input-, Output- oder Datenbereich abgelegt wird.

10.1.4 NanoJ-Programm ausführen

Zusammengefasst besteht das *NanoJ-Programm* bei der Ausführung eines Zyklus hinsichtlich des PDO-Mappings aus folgenden drei Schritten:

1. Werte aus dem Objektverzeichnis lesen und in die Bereiche Inputs und Outputs kopieren
2. Benutzerprogramm ausführen
3. Werte aus den Bereichen Outputs und Inputs zurück in das Objektverzeichnis kopieren

Die Konfiguration der Kopiervorgänge ist dem CANopen-Standard angelehnt.

Zusätzlich kann über Systemcalls auf Werte des Objektverzeichnisses zugegriffen werden. Dies ist im Allgemeinen deutlich langsamer und daher sind Mappings vorzuziehen. Die Anzahl an Mappings ist begrenzt (jeweils 16 Einträge in In/Out/InOut).



Tipp

Nanotec empfiehlt: Häufig genutzte und veränderte OD-Einträge mappen und auf weniger häufig genutzte OD-Einträge per Systemcall zuzugreifen.

Eine Liste verfügbarer Systemcalls findet sich im Kapitel **Systemcalls im NanoJ-Programm**.



Tipp

Nanotec empfiehlt, entweder per Mapping oder Systemcall mit `od_write()` auf ein und denselben OD-Wert zuzugreifen. Wird beides gleichzeitig verwendet, so hat der Systemcall keine Auswirkung.

10.1.5 NanoJ-Programm OD-Einträge

Das *NanoJ-Programm* wird durch OD-Einträge im Objekt-Bereich 2300_h bis 2330_h gesteuert und konfiguriert (siehe **2300h NanoJ Control**).

OD-Index	Name und Beschreibung
2300 _h	2300h NanoJ Control
2301 _h	2301h NanoJ Status
2302 _h	2302h NanoJ Error Code
2310 _h	2310h NanoJ Input Data Selection
2320 _h	2320h NanoJ Output Data Selection
2330 _h	2330h NanoJ In/output Data Selection

Beispiel:

Um das Benutzerprogramm *TEST1.USR* zu starten, kann z. B. folgende Sequenz benutzt werden:

- Überprüfen des Eintrags **2302_h** auf Fehlercode.
- Wenn kein Fehler:
NanoJ-Programm starten durch Beschreiben von Objekt **2300_h**, Bit 0 = "1".



Hinweis

Das Starten des NanoJ Programms kann bis zu 200 ms dauern.

- Überprüfen des Eintrags **2302_h** auf Fehlercode und des Objekts **2301_h**, Bit 0 = "1".



Hinweis

Aufgrund Limitierungen in der USB Implementation wird nach einem Neustart der Steuerung die Datei "VMMCODE.USR" auf eine Größe von 16kB gesetzt und das Erstelldatum auf den 13.03.2012 gestellt.

Um ein laufendes Programm anzuhalten: Beschreiben des Eintrags **2300_h** mit dem Bit 0 Wert = "0".

10.1.6 Aufbau NanoJ-Programm

Ein Benutzerprogramm besteht aus mindestens zwei Anweisungen:

- der Präprozessoranweisung `#include "wrapper.h"`
- der Funktion `void user() {}`

In der Funktion `void user()` lässt sich der auszuführende Code hinterlegen.



Hinweis

Die Dateinamen der Benutzerprogramme dürfen nicht länger als acht Zeichen sein und drei Zeichen im Suffix enthalten; Dateiname `main.cpp` ist zulässig, Dateiname `einLangerDateiname.cpp` ist nicht zulässig.



Hinweis

In *NanoJ-Programmen* dürfen globale Variablen ausschließlich innerhalb von Funktionen initialisiert werden. Daraus folgt:

- kein `new` Operator
- keine Konstruktoren
- keine Initialisierung von globalen Variablen außerhalb von Funktionen

Beispiele:

Die globale Variable soll erst innerhalb der Funktion `void user()` initialisiert werden:

```
unsigned int i;  
void user(){  
    i = 1;  
    i += 1;  
}
```

Folgende Zuweisung ist nicht korrekt :

```
unsigned int i = 1;  
void user() {  
    i += 1;  
}
```

10.1.7 NanoJ-Programmbeispiel

Das Beispiel zeigt das Programmieren eines Rechtecksignals in das Objekt `2500h:01h`.

```
// file main.cpp  
map S32 outputReg1 as inout 0x2500:1  
#include "wrapper.h"  
  
// user program  
void user()  
{  
    U16 counter = 0;  
    while( 1 )  
    {  
        ++counter;  
  
        if( counter < 100 )  
            InOut.outputReg1 = 0;  
        else if( counter < 200 )  
            InOut.outputReg1 = 1;  
        else  
            counter = 0;  
    }  
}
```

```
// yield() 5 times (delay 5ms)
for(U08 i = 0; i < 5; ++i )
    yield();
}
} // eof
```

Weitere Beispiele finden Sie auf www.nanotec.de.

10.2 Mapping im NanoJ-Programm

Mit dieser Methode wird eine Variable im *NanoJ-Programm* direkt mit einem Eintrag im Objektverzeichnis verknüpft. Das Anlegen des Mappings muss dabei am Anfang der Datei stehen - noch vor der `#include "wrapper.h"`-Anweisung. Ein Kommentar oberhalb des Mappings ist erlaubt.



Tip

Nanotec empfiehlt:

- Benutzen Sie das Mapping, falls Sie den Zugriff auf ein Objekt im Objektverzeichnis häufiger benötigen, z. B. das *Controlword* 6040_h oder das *Statusword* 6041_h.
- Für den einzelnen Zugriff auf Objekte bieten sich eher die Funktionen `od_write()` und `od_read()` an, siehe **Zugriff auf das Objektverzeichnis**.

10.2.1 Deklaration des Mappings

Die Deklaration des Mappings gliedert sich dabei folgendermaßen:

```
map <TYPE> <NAME> as <input|output|inout> <INDEX>:<SUBINDEX>
```

Dabei gilt:

- `<TYPE>`
Der Datentyp der Variable; U32, U16, U08, S32, S16 oder S08.
- `<NAME>`
Der Name der Variable; wie sie im Benutzerprogramm verwendet wird.
- `<input|output|inout>`
Die Schreib- und Leseberechtigung einer Variable: Eine Variable kann entweder als `input`, `output` oder `inout` deklariert werden. Damit wird festgelegt, ob eine Variable lesbar (`input`), schreibbar (`output`) oder beides ist (`inout`) und über welche Struktur sie im Programm angesprochen werden muss.
- `<INDEX>:<SUBINDEX>`

Index und Subindex des zu mappenden Objekts im Objektverzeichnis.

Jede deklarierte Variable wird im Benutzerprogramm über eine der drei Strukturen *In*, *Out* oder *InOut* angesprochen, je nach definierter Schreib- und Leserichtung.

10.2.2 Beispiel eines Mappings

Beispiel eines Mappings und der zugehörigen Variablenzugriffe:

```
map U16 controlWord as output 0x6040:00
map U08 statusWord as input 0x6041:00
map U08 modeOfOperation as inout 0x6060:00
```

```
#include "wrapper.h"

void user()
{
    [...]
    Out.controlWord = 1;
    U08 tmpVar = In.statusword;
    InOut.modeOfOperation = tmpVar;
    [...]
}
```

10.2.3 Möglicher Fehler bei `od_write()`

Eine mögliche Fehlerquelle ist ein schreibender Zugriff mittels der Funktion `od_write()` (siehe **Systemcalls im NanoJ-Programm**) auf ein Objekt im Objektverzeichnis, welches gleichzeitig als Mapping angelegt wurde. Nachfolgend aufgelisteter Code ist fehlerhaft:

```
map U16 controlWord as output 0x6040:00
#include " wrapper.h"
void user()
{
    [...]
    Out.controlWord = 1;
    [...]
    od_write(0x6040, 0x00, 5 ); // der Wert wird durch das Mapping
    überschrieben
    [...]
}
```

Die Zeile mit dem Befehl `od_write(0x6040, 0x00, 5);` ist wirkungslos. Wie in der Einleitung beschrieben, werden alle Mappings am Ende jeder Millisekunde in das Objektverzeichnis kopiert.

Damit ergibt sich folgender Ablauf:

1. Die Funktion `od_write` schreibt den Wert 5 in das Objekt `6040h:00h`.
2. Am Ende des 1 ms-Zyklus wird das Mapping geschrieben, welches ebenfalls das Objekt `6040h:00h` beschreibt, allerdings mit dem Wert 1.
3. Somit wird - aus Sicht des Benutzers - der `od_write`-Befehl wirkungslos.

10.3 Systemcalls im NanoJ-Programm

Mit Systemcalls ist es möglich, in der Firmware eingebaute Funktionen direkt aus einem Benutzerprogramm aufzurufen. Eine direkte Code-Ausführung ist nur in dem geschützten Bereich der Sandbox möglich und wird über sogenannte *Cortex-Supervisor-Calls* (Svc Calls) realisiert. Dabei wird mit dem Aufruf der Funktion ein Interrupt ausgelöst und die Firmware hat so die Möglichkeit, temporär eine Code-Ausführung außerhalb der Sandbox zuzulassen. Der Entwickler des Benutzerprogramms muss sich jedoch um diesen Mechanismus nicht kümmern - für ihn sind die Systemcalls wie ganz normale C-Funktionen aufrufbar. Lediglich die Datei *wrapper.h* muss - wie üblich - eingebunden werden.

10.3.1 Zugriff auf das Objektverzeichnis

`void od_write (U32 index, U32 subindex, U32 value)`

Diese Funktion schreibt den übergebenen Wert an die angegebene Stelle in das Objektverzeichnis.

index	Index des zu schreibenden Objekts im Objektverzeichnis
subindex	Subindex des zu schreibenden Objekts im Objektverzeichnis
value	zu schreibender Wert



Hinweis

Es wird dringend empfohlen, nach dem Aufruf eines `od_write()` die Prozessorzeit mit `yield()` abzugeben. Der Wert wird zwar sofort ins OD geschrieben. Damit die Firmware jedoch davon abhängige Aktionen auslösen kann, muss diese Rechenzeit erhalten und somit das Benutzerprogramm beendet oder mit `yield()` unterbrochen worden sein.

U32 **od_read** (U32 index, U32 subindex)

Diese Funktion liest den Wert an der angegebenen Stelle aus dem Objektverzeichnis und gibt ihn zurück.

index	Index des zu lesenden Objekts im Objektverzeichnis
subindex	Subindex des zu lesenden Objekts im Objektverzeichnis
Rückgabewert	Inhalt des OD-Eintrags



Hinweis

Aktives Warten auf einen Wert im Objektverzeichnis sollte immer mit einem `yield()` verbunden werden.

Beispiel

```
while (od_read(2400,2) != 0) // wait until 2400:2 is set
{ yield(); }
```

10.3.2 Prozesssteuerung

```
void yield()
```

Diese Funktion gibt die Prozessorzeit wieder an das Betriebssystem ab. Das Programm wird in der nächsten Zeitscheibe wieder an der Stelle nach dem Aufruf fortgesetzt.

```
void sleep (U32 ms)
```

Diese Funktion gibt die Prozessorzeit für die angegebene Zahl an Millisekunden an das Betriebssystem ab. Das Benutzerprogramm wird anschließend an der Stelle nach dem Aufruf fortgesetzt.

ms	Zu wartende Zeit in Millisekunden
----	-----------------------------------

11 Objektverzeichnis Beschreibung

11.1 Übersicht

In diesem Kapitel finden Sie eine Beschreibung aller Objekte.

Sie finden hier Angaben zu:

- Funktionen
- Objektbeschreibungen ("Index")
- Wertebeschreibungen ("Subindices")
- Beschreibungen von Bits
- Beschreibung des Objekts

11.2 Aufbau der Objektbeschreibung

Die Beschreibung der Objekteinträge ist immer gleich aufgebaut und besteht im Normalfall aus folgenden Abschnitten:

Funktion

In diesem Abschnitt wird kurz die Funktion des Objektverzeichnisses beschrieben.

Objektbeschreibung

Diese Tabelle gibt detailliert Auskunft über den Datentyp, Vorgabewerte und dergleichen. Eine genaue Beschreibung findet sich im Abschnitt "**Objektbeschreibung**"

Wertebeschreibung

Diese Tabelle ist nur bei dem Datentyp "Array" oder "Record" verfügbar und gibt genaue Auskunft über die Untereinträge. Eine genauere Beschreibung der Einträge findet sich im Abschnitt "**Wertebeschreibung**"

Beschreibung

Hier werden genauere Angaben zu den einzelnen Bits eines Eintrags gemacht oder eventuelle Zusammensetzungen erläutert. Eine genauere Beschreibung findet sich im Abschnitt "**Beschreibung**"

11.3 Objektbeschreibung

Die Objektbeschreibung besteht aus einer Tabelle, welche folgende Einträge enthält:

Index

Benennt den Index des Objekts in Hexadezimalschreibweise.

Objektname

Der Name des Objekts.

Object Code

Der Typ des Objekts. Das kann einer der folgenden Einträge sein:

- VARIABLE: In dem Fall besteht das Objekt nur aus einer Variable, die mit dem Subindex 0 indiziert wird.
- ARRAY: Diese Objekte bestehen immer aus einem Subindex 0 - welcher die Menge der Untereinträge angibt - und den Untereinträgen selber ab dem Index 1. Der Datentyp innerhalb eines Arrays ändert sich nie, das heißt, Untereintrag 1 und alle folgenden Einträge haben immer den gleichen Datentyp.
- RECORD: Diese Objekte bestehen immer aus einem Untereintrag mit dem Subindex 0 - welcher die Menge der Untereinträge angibt - und den Untereinträgen selber ab dem Index 1. Im Gegensatz zu einem ARRAY kann der Datentyp der Subeinträge variieren, das

bedeutet, dass beispielsweise Untereintrag 1 einen anderen Datentyp als Untereintrag 2 haben kann.

- **VISIBLE_STRING**: Das Objekt beschreibt eine in ASCII codierte Zeichenkette. Die Länge des Strings wird in Subindex 0 angegeben, die einzelnen Zeichen sind ab Subindex 1 gespeichert. Diese Zeichenketten sind **nicht** durch ein Null-Zeichen terminiert.

Datentyp

Hier wird die Größe und die Interpretation des Objekts angegeben. Für den Object Code "VARIABLE" gilt folgende Schreibweise:

- Es wird unterschieden zwischen Einträgen die vorzeichenbehaftet sind, das wird mit dem Präfix "SIGNED" bezeichnet. Für die vorzeichenunbehafteten Einträge wird das Präfix "UNSIGNED" benutzt.
- Die Größe der Variable in Bit wird an das Präfix angestellt und kann entweder 8, 16 oder 32 sein.

Speicherbar

Hier wird beschrieben ob dieses Objekt speicherbar ist und wenn ja, unter welcher Kategorie.

Firmware Version

Hier ist die Firmwareversion eingetragen, ab der das Objekt verfügbar ist.

Änderungshistorie (ChangeLog)

Hier werden eventuelle Änderungen an dem Objekt notiert.

Zudem gibt es noch die Einträge für den Datentyp "VARIABLE" folgende Tabelleneinträge:

Zugriff

Hier wird die Zugriffsbeschränkung eingetragen. Folgende Beschränkungen gibt es:

- "lesen/schreiben": Das Objekt kann sowohl gelesen, als auch geschrieben werden
- "nur lesen": Das Objekt kann nur aus dem Objektverzeichnis gelesen werden. Setzen eines Werte ist nicht möglich.

PDO-Mapping

Einige Bussysteme, wie CANopen oder EtherCAT unterstützen ein PDO-Mapping. In diesem Tabelleneintrag wird beschrieben, ob das Objekt in ein Mapping eingefügt werden darf und in welches. Dabei gibt es folgende Bezeichnungen:

- "no": Das Objekt darf in kein Mapping eingetragen werden.
- "TX-PDO": Das Objekt darf in ein RX Mapping eingetragen werden.
- "RX-PDO": Das Objekt darf in ein TX Mapping eingetragen werden.

Zulässige Werte

In einigen Fällen ist es nur erlaubt, bestimmte Werte in das Objekt zu schreiben. Sollte das der Fall sein, werden diese Werte hier aufgelistet. Besteht keine Beschränkung bleibt das Feld leer.

Vorgabewert

Um die Steuerung beim Einschalten in einen gesicherten Zustand zu bringen ist es nötig, einige Objekte mit Werten vorzubelegen. Der Wert, der beim Start der Steuerung in das Objekt geschrieben wird, wird in diesem Tabelleneintrag notiert.

11.4 Wertebeschreibung



Hinweis

Der Übersichtlichkeit halber werden einige Subindizes zusammengefasst, wenn die Einträge alle den gleichen Namen haben.

In der Tabelle mit der Überschrift "Wertebeschreibung" werden alle Daten für Untereinträge mit Subindex 1 oder höher aufgelistet. Die Tabelle beinhaltet folgende Einträge:

Subindex

Nummer des aktuell beschriebenen Untereintrages.

Name

Der Name des Untereintrages.

Datentyp

Hier wird die Größe und die Interpretation des Untereintrages angegeben. Hier gilt immer folgende Schreibweise:

- Es wird unterschieden zwischen Einträgen die vorzeichenbehaftet sind, das wird mit dem Präfix "SIGNED" bezeichnet. Für die vorzeichenunbehafteten Einträge wird das Präfix "UNSIGNED" benutzt.
- Die Größe der Variable in Bit wird an das Präfix angestellt und kann entweder 8, 16 oder 32 sein.

Zugriff

Hier wird die Zugriffsbeschränkung für den Untereintrag eingetragen. Folgende Beschränkungen gibt es:

- "lesen/schreiben": Das Objekt kann sowohl gelesen, als auch geschrieben werden
- "nur lesen": Das Objekt kann nur aus dem Objektverzeichnis gelesen werden. Setzen eines Wertes ist nicht möglich.

PDO-Mapping

Einige Bussysteme, wie CANopen oder EtherCAT unterstützen ein PDO-Mapping. In diesem Tabelleneintrag wird beschrieben, ob der Untereintrag in ein Mapping eingefügt werden darf und in welches. Dabei gibt es folgende Bezeichnungen:

- "no": Das Objekt darf in kein Mapping eingetragen werden.
- "TX-PDO": Das Objekt darf in ein RX Mapping eingetragen werden.
- "RX-PDO": Das Objekt darf in ein TX Mapping eingetragen werden.

Zulässige Werte

In einigen Fällen ist es nur erlaubt, bestimmte Werte in den Untereintrag zu schreiben. Sollte das der Fall sein, werden diese Werte hier aufgelistet. Besteht keine Beschränkung, bleibt das Feld leer.

Vorgabewert

Um die Steuerung beim Einschalten in einen gesicherten Zustand zu bringen ist es nötig, einige Untereinträge mit Werten vor zu belegen. Der Wert, welcher beim Start der Steuerung in den Untereintrag geschrieben wird, wird in diesem Tabelleneintrag notiert.

11.5 Beschreibung

Dieser Abschnitt kann vorhanden sein, wenn die Benutzung zusätzliche Information verlangt. Sollten einzelne Bits eines Objekts oder Untereintrags unterschiedliche Bedeutung haben, so werden Diagramme wie im nachfolgenden Beispiel verwendet.

Beispiel: Das Objekt ist 8 Bit groß, Bit 0 und 1 haben separat eine Funktion. Bit 2 und 3 sind zu einer Funktion zusammengefasst, für Bit 4 bis 7 gilt das gleiche.

7	6	5	4	3	2	1	0
Beispiel [4]				Beispiel [2]		B	A

Beispiel [4]

Beschreibung der Bits 4 bis einschließlich 7, diese Bits gehören logisch zusammen. Die 4 in den eckigen Klammern gibt die Anzahl der zusammengehörigen Bits an. Oftmals wird an der Stelle noch eine Liste mit möglichen Werten und deren Beschreibung angehängt.

Beispiel [2]

Beschreibung der Bits 3 und 2, diese Bits gehören logisch zusammen. Die 2 in den eckigen Klammern gibt die Anzahl der zusammengehörigen Bits an.

- Wert 00_b: Die Beschreibung an dieser Stelle gilt, wenn Bit 2 und Bit 3 auf "0" sind.
- Wert 01_b: Die Beschreibung an dieser Stelle gilt, wenn Bit 2 auf "0" und Bit 3 auf "1" ist.
- Wert 10_b: Die Beschreibung an dieser Stelle gilt, wenn Bit 2 auf "1" und Bit 3 auf "0" ist.
- Wert 11_b: Die Beschreibung an dieser Stelle gilt, wenn Bit 2 und Bit 3 auf "1" sind.

B

Beschreibung des Bits B, auf die Längenangabe wird bei einem einzelnen Bit verzichtet.

A

Beschreibung des Bits A, Bits mit grauen Hintergrund bleiben ungenutzt.

1000h Device Type

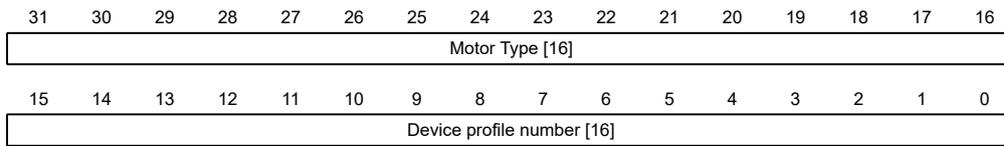
Funktion

Beschreibt den Steuerungstyp.

Objektbeschreibung

Index	1000 _h
Objektname	Device Type
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00060192 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung



Motor Type[16]

Beschreibt den unterstützten Motor-Typ. Die folgenden Werte sind möglich:

- Bit 23 bis Bit 16: Wert "1": Servoantrieb
- Bit 23 bis Bit 16: Wert "2": Schrittmotor

Device profile number[16]

Beschreibt den unterstützten CANopen-Standard.

Werte:

0192_h bzw. 0402_d (Vorgabewert): Der CiA 402-Standard wird unterstützt.

1001h Error Register

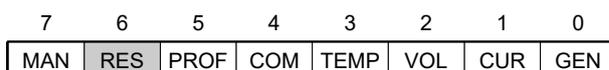
Funktion

Fehlerregister: Im Fehlerfall wird das entsprechende Fehlerbit gesetzt. Sollte der Fehler nicht mehr bestehen, wird es automatisch wieder gelöscht.

Objektbeschreibung

Index	1001 _h
Objektname	Error Register
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung



GEN

Genereller Fehler

CUR

Strom

VOL

Spannung

TEMP

Temperatur

COM

Kommunikation

PROF

Betrifft das Geräteprofil

RES

Reserviert, immer "0"

MAN

Hersteller spezifisch: Der Motor drehte sich in die falsche Richtung.

1003h Pre-defined Error Field

Funktion

Dieses Objekt beinhaltet einen Fehlerstapel mit bis zu acht Einträgen.

Objektbeschreibung

Index	1003 _h
Objektname	Pre-defined Error Field
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Errors
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	07 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

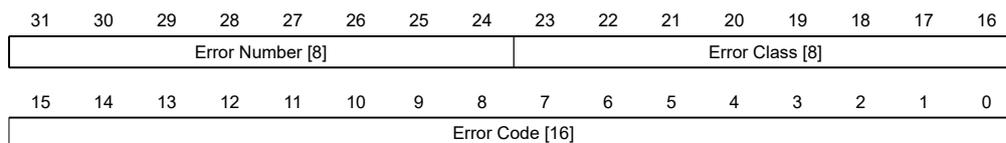
Beschreibung

Allgemeine Funktionsweise

Tritt ein neuer Fehler auf, wird dieser in Subindex 1 eingetragen. Die bereits vorhandenen Einträge in den Subindizes 1 bis 7 werden um eine Stelle nach hinten verschoben. Der Fehler auf Subindex 7 wird dabei entfernt.

Die Anzahl der bereits aufgetreten Fehler lässt sich aus dem Objekt mit dem Subindex 0 ablesen. Ist im Fehlerstapel zur Zeit kein Fehler eingetragen, dann ist das Auslesen eines der acht Subindizes 1-8 nicht möglich und wird mit einem Fehler (Abort-Code=08000024_h) beantwortet. Wird in den Subindex 0 eine "0" geschrieben, beginnt die Zählung von neuem.

Bitbeschreibung



Error Number [8]

Damit lässt sich der Grund des Fehlers genau eingrenzen. Die Bedeutung der Zahl lässt sich aus nachfolgender Tabelle entnehmen.

Fehlernummer	Beschreibung
0	Watchdog-Reset
1	Eingangsspannung zu hoch
2	Ausgangsstrom zu hoch
3	Eingangsspannung zu niedrig
4	Fehler am Feldbus
6	Nur CANopen: NMT-Master braucht zu lange, um Nodeguarding-Anforderung zu schicken
7	Encoderfehler durch elektrische Störung oder defekte Hardware
9	Fehler in der A/B-Spur des Encoders

Fehlernummer	Beschreibung
10	Warnung: Positiver Endschalter überschritten
11	Warnung: Negativer Endschalter überschritten
12	Übertemperatur-Fehler
13	Die Werte des Objekts 6065_h (Following Error Window) und des Objekts 6066_h (Following Error Time Out) wurden überschritten, es wurde ein Fault ausgelöst.
14	Warnung: nichtflüchtiger Speicher voll. Der aktuelle Speichervorgang konnte nicht abgeschlossen werden, Teile der Daten des Speichervorgangs sind verloren. Neustart der Steuerung erforderlich für Aufräumarbeiten.
15	Motor blockiert
16	Warnung: nichtflüchtiger Speicher beschädigt, Neustart der Steuerung erforderlich für Aufräumarbeiten (alle gespeicherten Objekte werden auf Default zurückgesetzt).
17	Nur CANopen: Slave brauchte zu lange um PDO-Nachrichten zu Senden.
18	Hall-Sensor fehlerhaft
19	Nur CANopen: PDO aufgrund eines Längenfehlers nicht verarbeitet
20	Nur CANopen: PDO Länge überschritten
21	Warnung: Starten Sie die Steuerung neu, um zukünftige Fehler beim Speichern (nichtflüchtiger Speicher voll/korrupt) zu vermeiden.
22	Nennstrom muss gesetzt werden (203B _h :01 _h /6075 _h)
23	Encoderauflösung, Polpaarzahl und einige andere Werte sind falsch.
24	Motorstrom ist zu hoch, passen Sie die PI-Parameter an.
25	Interner Softwarefehler, generisch
26	Zu hoher Strom am digitalen Ausgang
27	Nur CANopen: Unerwartete Sync-Länge
30	Fehler in der Drehzahlüberwachung: Schlupffehler zu groß

Error Class[8]

Dieses Byte ist identisch mit dem Objekt **1001_h**

Error Code[16]

Die Bedeutung der beiden Bytes lässt sich aus der nachfolgenden Tabelle entnehmen.

Error Code	Beschreibung
1000 _h	Allgemeiner Fehler
2300 _h	Strom am Ausgang der Steuerung zu groß
3100 _h	Über-/ Unterspannung am Eingang der Steuerung
4200 _h	Temperaturfehler innerhalb der Steuerung
6010 _h	Software-Reset (Watchdog)
6100 _h	Interner Softwarefehler, generisch
6320 _h	Nennstrom muss gesetzt werden (203B _h :01 _h /6075 _h)
7121 _h	Motor blockiert
7305 _h	Inkrementaler oder Hall-Sensor fehlerhaft
7600 _h	Warnung: Nichtflüchtiger Speicher voll oder korrupt, Neustart der Steuerung für Aufräumarbeiten

Error Code	Beschreibung
8000 _h	Fehler bei der Feldbusüberwachung
8130 _h	Nur CANopen: "Life Guard"-Fehler oder "Heartbeat"-Fehler
8200 _h	Nur CANopen: Slave brauchte zu lange um PDO Nachrichten zu Senden.
8210 _h	Nur CANopen: PDO wurde nicht verarbeitet aufgrund eines Längen-Fehlers
8220 _h	Nur CANopen: PDO Länge überschritten
8240 _h	Nur CANopen: Unerwartete Sync-Länge
8400 _h	Fehler in der Drehzahlüberwachung: Schlupffehler zu groß
8611 _h	Fehler in der Positionsüberwachung: Schleppfehler zu groß
8612 _h	Fehler in der Positionsüberwachung: Endschalter überschritten

1005h COB-ID Sync

Funktion

Definiert die COB-ID der SYNC-Nachricht für das SYNC-Protokoll. Der Wert muss einer 11-Bit langen CAN-ID entsprechen und wird bei einem Neustart der Steuerung oder bei einem Reset Communication Kommando ausgewertet. Das Generieren von Sync-Nachrichten wird nicht unterstützt.



Hinweis

Wenn die CAN-ID nicht dem Defaultwert 80_h entsprechen soll, muss man berücksichtigen, dass nur noch nicht vergebene oder reservierte CAN-IDs verwendet werden.

Objektbeschreibung

Index	1005 _h
Objektname	COB-ID Sync
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000080 _h
Firmware Version	FIR-v1426
Änderungshistorie	

1007h Synchronous Window Length

Funktion

Dieses Objekt enthält die Länge des Zeitfensters in Mikrosekunden für synchrone PDOs. Wenn das synchrone Zeitfenster abgelaufen ist, dann werden alle synchronen TxPDOs verworfen und eine EMCY-Nachricht verschickt. Auch die RxPDOs werden bis zur nächsten SYNC-Nachricht verworfen.

Der Wert "0" schaltet das Zeitfenster ab, sodass die PDOs zu jedem beliebigen Zeitpunkt gesendet werden können.

Dieses Objekt ist nur in Gerätevarianten mit CANopen-Anschluss vorhanden.

Objektbeschreibung

Index	1007 _h
Objektname	Synchronous Window Length
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

1008h Manufacturer Device Name

Funktion

Enthält den Gerätenamen als Zeichenkette.

Objektbeschreibung

Index	1008 _h
Objektname	Manufacturer Device Name
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none">• CL4-E-1-12: CL4-E-1-12• CL4-E-2-12: CL4-E-2-12• CL4-E-1-12-5VDI: CL4-E-1-12-5VDI• CL4-E-2-12-5VDI: CL4-E-2-12-5VDI
Firmware Version	FIR-v1426
Änderungshistorie	

1009h Manufacturer Hardware Version

Funktion

Dieses Objekt enthält die Hardware-Version als Zeichenkette.

Objektbeschreibung

Index	1009 _h
Objektname	Manufacturer Hardware Version
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1426
Änderungshistorie	

100Ah Manufacturer Software Version

Funktion

Dieses Objekt enthält die Software-Version als Zeichenkette.

Objektbeschreibung

Index	100A _h
Objektname	Manufacturer Software Version
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FIR-v1825-B588141
Firmware Version	FIR-v1426
Änderungshistorie	

100Ch Guard Time

Funktion

Das Objekt 100C_h multipliziert mit dem Objekt **100Dh Live Time Factor** ergibt die sogenannte Life Time für das Life Guarding / Node Guarding Protokoll. Der Wert wird in Millisekunden angegeben. Siehe auch **Nodeguarding**.



Hinweis

Das *Heartbeat-Protokoll* hat eine höhere Priorität als das *Nodeguarding*. Sind beide Protokolle gleichzeitig aktiviert, wird der Node Guarding Timer unterdrückt, aber auch keine EMCY-Nachricht verschickt.

Objektbeschreibung

Index	100C _h
Objektname	Guard Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

100Dh Live Time Factor

Funktion

Dieses Objekt ist ein Multiplikator, welcher zusammen mit dem Objekt **100C_h** multipliziert das Zeitfenster für das *Nodeguarding* Protokoll in Millisekunden ergibt. Siehe auch **Nodeguarding**.



Hinweis

Das *Heartbeat-Protokoll* hat eine höhere Priorität als das *Nodeguarding*. Sind beide Protokolle gleichzeitig aktiviert, dann wird der Node Guarding Timer unterdrückt, aber auch keine EMCY-Nachricht verschickt.

Dieses Objekt ist nur in Gerätevarianten mit CANopen-Anschluss vorhanden.

Objektbeschreibung

Index	100D _h
Objektname	Live Time Factor
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	FIR-v1426
Änderungshistorie	

1010h Store Parameters

Funktion

Mit diesem Objekt lässt sich das Speichern von Objekten starten. Siehe Kapitel **Objekte speichern**.

Objektbeschreibung

Index	1010 _h
Objektname	Store Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1436: Eintrag "Objektname" geändert von "Store Parameter" auf "Store Parameters".</p> <p>Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 3 auf 4.</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5.</p> <p>Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7.</p> <p>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.</p>

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0D _h
Subindex	01 _h
Name	Save All Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	02 _h
Name	Save Communication Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	03 _h
Name	Save Application Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	04 _h
Name	Save Customer Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	05 _h
Name	Save Drive Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	06 _h
Name	Save Tuning Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	07 _h
Name	Save Miscellaneous Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	08 _h
Name	Save Reserved1 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	09 _h
Name	Save Reserved2 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0A _h
Name	Save CANopen Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	0B _h
Name	Save Modbus RTU Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	0C _h
Name	Save Ethernet Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	0D _h
Name	Save Profibus Configurations To Non-volatile Memory
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Jeder Subindex des Objekts steht für eine bestimmte Speicherklasse. Durch Auslesen eines Eintrages kann festgestellt werden, ob diese Speicherkategorie abgespeichert (Wert "1") werden kann oder nicht (Wert="0").

Um den Speichervorgang einer Speicherkategorie zu starten, muss der Wert "65766173_h" in den jeweiligen Subindex geschrieben werden. Das entspricht dezimal der 1702257011_d bzw. dem ASCII String "save". Sobald der Speichervorgang abgeschlossen wurde, wird der Speicherbefehl wieder durch den Wert "1" überschrieben, da ein Speichern wieder möglich ist.

Für eine detaillierte Beschreibung siehe Kapitel **Objekte speichern**.

1011h Restore Default Parameters

Funktion

Mit diesem Objekt kann das gesamte oder Teile des Objektverzeichnis auf die Defaultwerte zurückgesetzt werden. Siehe Kapitel **Objekte speichern**.

Objektbeschreibung

Index	1011 _h
Objektname	Restore Default Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "Restore Default Parameter" auf "Restore Default Parameters".</p> <p>Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 2 auf 4.</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5.</p> <p>Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Comm Default Parameters" auf "Restore Communication Default Parameters".</p> <p>Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Application Default Parameters" auf "Restore Application Default Parameters".</p> <p>Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7.</p>

Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0D _h

Subindex	01 _h
Name	Restore All Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	02 _h
Name	Restore Communication Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	03 _h
Name	Restore Application Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	04 _h
Name	Restore Customer Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000001 _h
Subindex	05 _h
Name	Restore Drive Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	06 _h
Name	Restore Tuning Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	07 _h
Name	Restore Miscellaneous Configurations
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	08 _h
Name	Restore Reserved1 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	09 _h
Name	Restore Reserved2 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0A _h

Name	Restore CANopen Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	0B _h
Name	Restore Modbus RTU Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	0C _h
Name	Restore Ethernet Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	0D _h
Name	Restore Profibus Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Wird der Wert 64616F6C_h (bzw. 1684107116_d oder ASCII `load`) in dieses Objekt geschrieben, werden Teile oder das gesamte Objektverzeichnis auf die Defaultwerte zurückgesetzt. Der verwendete Subindex entscheidet darüber, welcher Bereich zurück gesetzt wird.

Für eine detaillierte Beschreibung siehe Kapitel **Speicherung verwerfen**.

1014h COB-ID EMCY

Funktion

Dieses Objekt beschreibt die COB-ID des "Emergency Service" unter CANopen.

Mit dem *Valid Bit* (Bit 31) = "1" kann der **Emergency Service** deaktiviert werden, mit dem Wert "0" ist der Service aktiv. Bit 0 bis 30 werden bei jedem Neustart der Steuerung entsprechend der Node-ID generiert.

Objektbeschreibung

Index	1014 _h
Objektname	COB-ID EMCY
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "nur lesen" auf "lesen/schreiben".

1016h Consumer Heartbeat Time

Funktion

Dieses Objekt definiert die Zykluszeit des *Consumer Heartbeat* des CANopen Services *Network Management* und die Node-ID des *Producer* des *Heartbeat*.

Ist der Zykluszeit oder die Node-ID auf den Wert 0 gesetzt, wird auf die Heartbeat-Nachricht nicht reagiert. Siehe auch Kapitel **Heartbeat**.

Objektbeschreibung

Index	1016 _h
Objektname	Consumer Heartbeat Time
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	Consumer Heartbeat Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Der Subindex 01_h enthält:

- Bits 0 bis 15: die Zeit des *Consumer Heartbeat* in ms.
- Bits 16 bis 23: die Node-ID des *Producer*, dessen *Heartbeat* überwacht werden soll.



1017h Producer Heartbeat Time

Funktion

Dieses Objekt definiert die Zykluszeit des *Heartbeat* des CANopen Services *Network Management* in Millisekunden. Ist das Objekt auf den Wert 0 gesetzt, wird keine Heartbeat-Nachricht verschickt. Siehe auch **Heartbeat**.



Hinweis

Das *Heartbeat-Protokoll* hat eine höhere Priorität als das *Nodeguarding*. Sind beide Protokolle gleichzeitig aktiviert, dann wird der Node Guarding Timer unterdrückt, aber auch keine EMCY-Nachricht verschickt.

Dieses Objekt ist nur in Gerätevarianten mit CANopen-Anschluss vorhanden.

Objektbeschreibung

Index	1017 _h
Objektname	Producer Heartbeat Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

1018h Identity Object

Funktion

Dieses Objekt liefert generelle Informationen zu dem Gerät wie Hersteller, Produktcode, Revision und Seriennummer.



Tip

Halten Sie diese Werte bei Serviceanfragen bereit.

Objektbeschreibung

Index	1018 _h
Objektname	Identity Object
Object Code	RECORD
Datentyp	IDENTITY
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h

Subindex	01 _h
Name	Vendor-ID
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000026C _h

Subindex	02 _h
Name	Product Code
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> CL4-E-1-12: 0000005A_h

- CL4-E-2-12: 0000005B_h
- CL4-E-1-12-5VDI: 0000005E_h
- CL4-E-2-12-5VDI: 0000005F_h

Subindex	03 _h
Name	Revision Number
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	07210000 _h

Subindex	04 _h
Name	Serial Number
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

1019h Synchronous Counter Overflow Value

Funktion

Hier wird der Wert eingetragen, ab welchem der *Sync Counter* anfangen soll, neu zu zählen. Siehe Kapitel **Synchronisations-Objekt (SYNC)**.

Objektbeschreibung

Index	1019 _h
Objektname	Synchronous Counter Overflow Value
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Synchronous counter overflow value" auf "Synchronous Counter Overflow Value". Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "UNSIGNED16" auf "UNSIGNED8".

Beschreibung

Zulässige Werte: 02_h bis F0_h.

1020h Verify Configuration

Funktion

Dieses Objekt zeigt den Tag und die Zeit der abgespeicherten Konfiguration an.

Ein Konfigurationstool oder ein Master kann dieses Objekt nutzen, um die Konfiguration nach einem Reset zu verifizieren und gegebenenfalls eine Neukonfiguration durchzuführen.

Das Tool muss das Datum und die Uhrzeit setzen, bevor der Speichermechanismus gestartet wird (siehe Kapitel **Objekte speichern**).

Objektbeschreibung

Index	1020 _h
Objektname	Verify Configuration
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Prüfung
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Configuration Date
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
----------	-----------------

Name	Configuration Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Subindex 01_h (Konfigurationsdatum) soll die Anzahl der Tage seit dem 1. Januar 1984 enthalten.

Subindex 02_h (Konfigurationszeit) soll die Nummer der Millisekunden seit Mitternacht enthalten.

1029h Error Behavior

Funktion

Mit diesem Objekt wird definiert, was der *NMT-Zustand* der Steuerung im Fehlerfall sein soll. Siehe auch Kapitel **Network Management (NMT)**.

Objektbeschreibung

Index	1029 _h
Objektname	Error Behavior
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Communication Error
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	02 _h
Name	Internal Device Error
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Mit diesem Subindex wird definiert, wie im Fall eines Kommunikationsfehlers reagiert werden soll:
 - Wert "00"_h: Die Steuerung wechselt in den Zustand *Pre-Operational* (wenn vorher der Zustand *Operational* war).
 - Wert "01"_h: Die Steuerung wechselt den Zustand nicht.
 - Wert "02"_h: Die Steuerung wechselt in den Zustand *Stopped*.
- 02_h: Mit diesem Subindex wird definiert, wie im Fall der restlichen Fehler (außer Kommunikationsfehler) reagiert werden soll:
 - Wert "00"_h: Die Steuerung wechselt in den Zustand *Pre-Operational* (wenn vorher der Zustand *Operational* war).
 - Wert "01"_h: Die Steuerung wechselt den Zustand nicht.
 - Wert "02"_h: Die Steuerung wechselt in den Zustand *Stopped*.

1400h Receive PDO 1 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das empfangsseitige Mapping (RX-PDO) im Objekt 1600_h.
 Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1400 _h
Objektnamen	Receive PDO 1 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
----------	-----------------

Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
<hr/>	
Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Für Details siehe Kapitel zur **Konfiguration des RX-PDO Mappings**.

1401h Receive PDO 2 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das empfangsseitige Mapping (RX-PDO) im Objekt 1601_h.
 Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1401 _h
Objektnamen	Receive PDO 2 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Für Details siehe Kapitel zur **Konfiguration des RX-PDO Mappings**.

1402h Receive PDO 3 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das empfangsseitige Mapping (RX-PDO) im Objekt 1602_h.
 Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1402 _h
Objektname	Receive PDO 3 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation

Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Für Details siehe Kapitel zur **Konfiguration des RX-PDO Mappings**.

1403h Receive PDO 4 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das empfangsseitige Mapping (RX-PDO) im Objekt 1603_h.
 Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1403 _h
-------	-------------------

Objektname	Receive PDO 4 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Für Details siehe Kapitel zur **Konfiguration des RX-PDO Mappings**.

1404h Receive PDO 5 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das empfangsseitige Mapping (RX-PDO) im Objekt 1604_h.
 Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1404 _h
Objektname	Receive PDO 5 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	80000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Für Details siehe Kapitel zur **Konfiguration des RX-PDO Mappings**.

1405h Receive PDO 6 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das empfangsseitige Mapping (RX-PDO) im Objekt 1605_h.
 Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1405 _h
Objektname	Receive PDO 6 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	80000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	FF _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Für Details siehe Kapitel zur **Konfiguration des RX-PDO Mappings**.

1406h Receive PDO 7 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das empfangsseitige Mapping (RX-PDO) im Objekt 1606_h.
 Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1406 _h
Objektname	Receive PDO 7 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	80000000 _h
Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Für Details siehe Kapitel zur **Konfiguration des RX-PDO Mappings**.

1407h Receive PDO 8 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das empfangsseitige Mapping (RX-PDO) im Objekt 1607_h.
 Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1407 _h
Objektname	Receive PDO 8 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	80000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Für Details siehe Kapitel zur **Konfiguration des RX-PDO Mappings**.

1600h Receive PDO 1 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung empfangen kann (RX-PDO 1). Das PDO wurde vorher über **1400h Receive PDO 1 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1600 _h
Objektname	Receive PDO 1 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Eintrag "Überschrift" geändert von "1600h Drive Control" auf "1600h Receive PDO 1 Mapping Parameter". Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Drive Control" auf "Receive PDO 1 Mapping Parameter".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60400010 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60600008 _h

Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	32020020 _h

Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	05 _h
----------	-----------------

Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

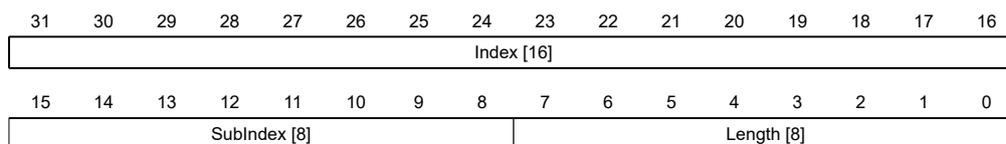
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1601h Receive PDO 2 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung empfangen kann (RX-PDO 2). Das PDO wurde vorher über **1401h Receive PDO 2 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1601 _h
Objektname	Receive PDO 2 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Eintrag "Überschrift" geändert von "1601h Positioning Control" auf "1601h Receive PDO 2 Mapping Parameter". Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Positioning Control" auf "Receive PDO 2 Mapping Parameter".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	607A0020 _h
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32

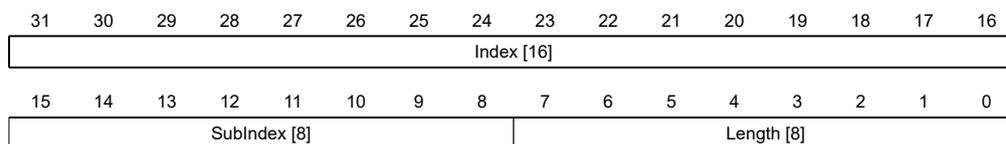
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60810020 _h
<hr/>	
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1602h Receive PDO 3 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung empfangen kann (RX-PDO 3). Das PDO wurde vorher über **1402h Receive PDO 3 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1602 _h
Objektname	Receive PDO 3 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Eintrag "Überschrift" geändert von "1602h Velocity Control" auf "1602h Receive PDO 3 Mapping Parameter".

Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Velocity Control" auf "Receive PDO 3 Mapping Parameter".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60420010 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

1603h Receive PDO 4 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung empfangen kann (RX-PDO 4). Das PDO wurde vorher über **1403h Receive PDO 4 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1603 _h
-------	-------------------

Objektname	Receive PDO 4 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Eintrag "Überschrift" geändert von "1603h Output Control" auf "1603h Receive PDO 4 Mapping Parameter". Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Output Control" auf "Receive PDO 4 Mapping Parameter".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60FE0120 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

1604h Receive PDO 5 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung empfangen kann (RX-PDO 5). Das PDO wurde vorher über **1404h Receive PDO 5 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1604 _h
Objektname	Receive PDO 5 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h

Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

1605h Receive PDO 6 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung empfangen kann (RX-PDO 6). Das PDO wurde vorher über **1405h Receive PDO 6 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1605 _h
Objektnamen	Receive PDO 6 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

1606h Receive PDO 7 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung empfangen kann (RX-PDO 7). Das PDO wurde vorher über **1406h Receive PDO 7 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1606 _h
Objektname	Receive PDO 7 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
----------	-----------------

Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

1607h Receive PDO 8 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung empfangen kann (RX-PDO 8). Das PDO wurde vorher über **1407h Receive PDO 8 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1607 _h
Objektname	Receive PDO 8 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported

Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

1800h Transmit PDO 1 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das sendeseitige Mapping (TX-PDO) 1. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1800 _h
Objektname	Transmit PDO 1 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	

Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 6 auf 7.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Subindex	03 _h
Name	Inhibit Time
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0064 _h

Subindex	04 _h
Name	Compatibility Entry
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	05 _h
Name	Event Timer
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	06 _h
Name	SYNC Start Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Subindex 3 (inhibit time): Hier kann eine Zeit in 100- μ s-Schritten eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs.

Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.

Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Danach erfolgt das Senden des PDO abhängig nur vom *Transmission Type* (Subindex 02_h). Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.

Für Details siehe Kapitel zur **Konfiguration des Tx-PDO Mappings**.

1801h Transmit PDO 2 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das sendeseitige Mapping (TX-PDO) 2. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1801 _h
-------	-------------------

Objektname	Transmit PDO 2 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 6 auf 7.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Subindex	03 _h
Name	Inhibit Time
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	0064 _h
Subindex	04 _h
Name	Compatibility Entry
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	05 _h
Name	Event Timer
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	06 _h
Name	SYNC Start Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Subindex 3 (inhibit time): Hier kann eine Zeit in 100- μ s-Schritten eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs.

Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.

Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Danach erfolgt das Senden des PDO abhängig nur vom *Transmission Type* (Subindex 02_h). Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.

Für Details siehe Kapitel zur **Konfiguration des Tx-PDO Mappings**.

1802h Transmit PDO 3 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das sendeseitige Mapping (TX-PDO) 3. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1802 _h
Objektname	Transmit PDO 3 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 6 auf 7.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	FF _h
Subindex	03 _h
Name	Inhibit Time
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0064 _h
Subindex	04 _h
Name	Compatibility Entry
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	05 _h
Name	Event Timer
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	06 _h
Name	SYNC Start Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Subindex 3 (inhibit time): Hier kann eine Zeit in 100- μ s-Schritten eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs.

Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.

Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Danach erfolgt das Senden des PDO abhängig nur vom *Transmission Type* (Subindex 02_h). Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.

Für Details siehe Kapitel zur **Konfiguration des Tx-PDO Mappings**.

1803h Transmit PDO 4 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das sendeseitige Mapping (TX-PDO) 4. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1803 _h
Objektname	Transmit PDO 4 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 6 auf 7.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h
Subindex	03 _h
Name	Inhibit Time
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0064 _h
Subindex	04 _h
Name	Compatibility Entry
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	05 _h
Name	Event Timer
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	06 _h
Name	SYNC Start Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Subindex 3 (inhibit time): Hier kann eine Zeit in 100-µs-Schritten eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs.

Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.

Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Danach erfolgt das Senden des PDO abhängig nur vom *Transmission Type* (Subindex 02_h). Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.

Für Details siehe Kapitel zur **Konfiguration des Tx-PDO Mappings**.

1804h Transmit PDO 5 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das sendeseitige Mapping (TX-PDO) 5. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1804 _h
Objektname	Transmit PDO 5 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 6 auf 7.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	C0000000 _h
Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h
Subindex	03 _h
Name	Inhibit Time
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0064 _h
Subindex	04 _h
Name	Compatibility Entry
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	05 _h
Name	Event Timer
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	06 _h
Name	SYNC Start Value
Datentyp	UNSIGNED8

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Subindex 3 (inhibit time): Hier kann eine Zeit in 100- μ s-Schritten eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs.

Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.

Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Danach erfolgt das Senden des PDO abhängig nur vom *Transmission Type* (Subindex 02_h). Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.

Für Details siehe Kapitel zur **Konfiguration des Tx-PDO Mappings**.

1805h Transmit PDO 6 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das sendeseitige Mapping (TX-PDO) 6. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1805 _h
Objektname	Transmit PDO 6 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 6 auf 7.

Wertebeschreibung

Subindex	00 _h
----------	-----------------

Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h
<hr/>	
Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	C0000000 _h
<hr/>	
Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h
<hr/>	
Subindex	03 _h
Name	Inhibit Time
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0064 _h
<hr/>	
Subindex	04 _h
Name	Compatibility Entry
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	05 _h
Name	Event Timer
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	06 _h
Name	SYNC Start Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Subindex 3 (inhibit time): Hier kann eine Zeit in 100- μ s-Schritten eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs.

Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.

Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Danach erfolgt das Senden des PDO abhängig nur vom *Transmission Type* (Subindex 02_h). Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.

Für Details siehe Kapitel zur **Konfiguration des Tx-PDO Mappings**.

1806h Transmit PDO 7 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das sendeseitige Mapping (TX-PDO) 7. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1806 _h
Objektname	Transmit PDO 7 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	

Firmware Version	FIR-v1614
Änderungshistorie	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 6 auf 7.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	C0000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Subindex	03 _h
Name	Inhibit Time
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0064 _h

Subindex	04 _h
Name	Compatibility Entry
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	05 _h
Name	Event Timer
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	06 _h
Name	SYNC Start Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Subindex 3 (inhibit time): Hier kann eine Zeit in 100- μ s-Schritten eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs.

Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.

Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Danach erfolgt das Senden des PDO abhängig nur vom *Transmission Type* (Subindex 02_h). Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.

Für Details siehe Kapitel zur **Konfiguration des Tx-PDO Mappings**.

1807h Transmit PDO 8 Communication Parameter

Funktion

Enthält die Kommunikationsparameter für das sendeseitige Mapping (TX-PDO) 8. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1807 _h
-------	-------------------

Objektname	Transmit PDO 8 Communication Parameter
Object Code	RECORD
Datentyp	PDO_COMMUNICATION_PARAMETER
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 6 auf 7.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	COB-ID
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	C0000000 _h

Subindex	02 _h
Name	Transmission Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF _h

Subindex	03 _h
Name	Inhibit Time
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	0064 _h
Subindex	04 _h
Name	Compatibility Entry
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	05 _h
Name	Event Timer
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	06 _h
Name	SYNC Start Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Subindex 01_h (COB-ID): Hier wird die COB-ID hinterlegt.

Subindex 02_h (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden.

Subindex 3 (inhibit time): Hier kann eine Zeit in 100- μ s-Schritten eingestellt werden, die nach einem Senden eines PDOs abgelaufen sein muss, damit ein weiteres Mal das PDO verschickt wird. Diese Zeit gilt nur für asynchrone PDOs.

Subindex 4 (compatibility entry): Dieser Subindex hat keine Funktion und ist nur aus Gründen der Kompatibilität vorhanden.

Subindex 5 (event timer): Diese Zeit (in ms) kann benutzt werden um einen *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Danach erfolgt das Senden des PDO abhängig nur vom *Transmission Type* (Subindex 02_h). Wird global erst aktiviert, wenn in **1019h Synchronous Counter Overflow Value** ein Wert größer 1 eingestellt wird.

Für Details siehe Kapitel zur **Konfiguration des Tx-PDO Mappings**.

1A00h Transmit PDO 1 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung senden kann (TX-PDO 1). Das PDO wurde vorher über **1800h Transmit PDO 1 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1A00 _h
Objektname	Transmit PDO 1 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Eintrag "Überschrift" geändert von "1A00h Drive Status" auf "1A00h Transmit PDO 1 Mapping Parameter". Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Drive Status" auf "Transmit PDO 1 Mapping Parameter".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60410010 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

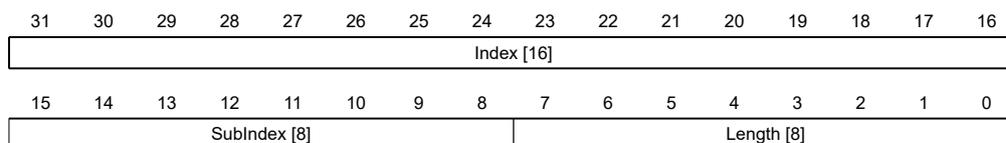
Vorgabewert	60610008 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h

Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1A01h Transmit PDO 2 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung senden kann (TX-PDO 2). Das PDO wurde vorher über **1801h Transmit PDO 2 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1A01 _h
Objektnamen	Transmit PDO 2 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Eintrag "Überschrift" geändert von "1A01h Positioning Status" auf "1A01h Transmit PDO 2 Mapping Parameter". Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Positioning Status" auf "Transmit PDO 2 Mapping Parameter".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60640020 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	05 _h
----------	-----------------

Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

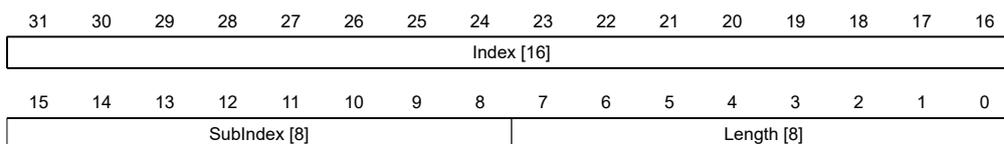
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1A02h Transmit PDO 3 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung senden kann (TX-PDO 3). Das PDO wurde vorher über **1802h Transmit PDO 3 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1A02 _h
Objektname	Transmit PDO 3 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Eintrag "Überschrift" geändert von "1A02h Velocity Status" auf "1A02h Transmit PDO 3 Mapping Parameter". Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Velocity Status" auf "Transmit PDO 3 Mapping Parameter".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h
Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60440010 _h
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32

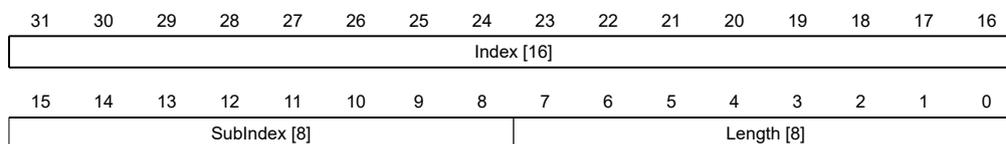
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1A03h Transmit PDO 4 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung senden kann (TX-PDO 4). Das PDO wurde vorher über **1803h Transmit PDO 4 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1A03 _h
Objektname	Transmit PDO 4 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Eintrag "Überschrift" geändert von "1A03h Input Status" auf "1A03h Transmit PDO 4 Mapping Parameter".

Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Input Status" auf "Transmit PDO 4 Mapping Parameter".

Wertebeschreibung

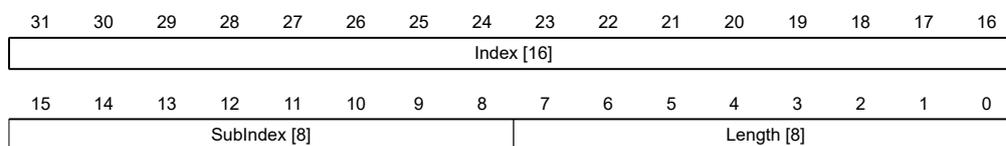
Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h
Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60FD0020 _h
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1A04h Transmit PDO 5 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung senden kann (TX-PDO 5). Das PDO wurde vorher über **1804h Transmit PDO 5 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1A04 _h
Objektname	Transmit PDO 5 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32

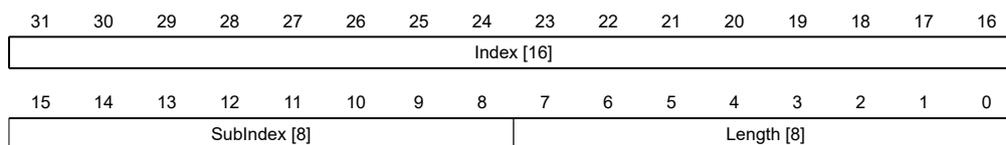
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1A05h Transmit PDO 6 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung senden kann (TX-PDO 6). Das PDO wurde vorher über **1805h Transmit PDO 6 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1A05 _h
Objektname	Transmit PDO 6 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte
 Vorgabewert
 Firmware Version FIR-v1614
 Änderungshistorie

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

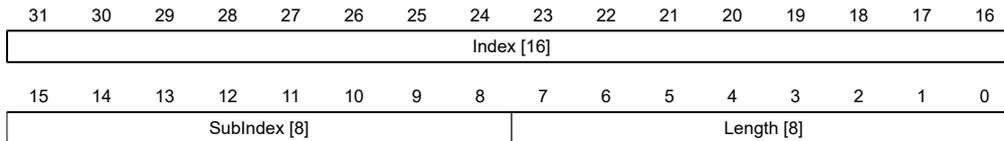
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1A06h Transmit PDO 7 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung senden kann (TX-PDO 7). Das PDO wurde vorher über **1806h Transmit PDO 7 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1A06 _h
Objektname	Transmit PDO 7 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	1st Object To Be Mapped

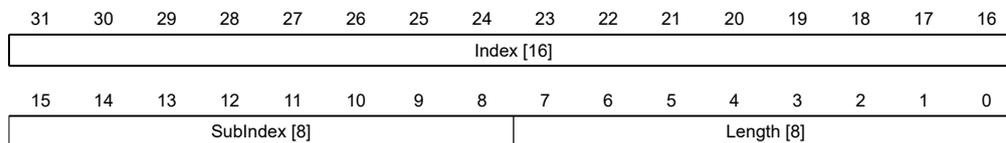
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1A07h Transmit PDO 8 Mapping Parameter

Funktion

Dieses Objekt enthält die Mapping-Parameter für PDOs, welche die Steuerung senden kann (TX-PDO 8). Das PDO wurde vorher über **1807h Transmit PDO 8 Communication Parameter** konfiguriert. Siehe Kapitel **Process Data Object (PDO)**.

Objektbeschreibung

Index	1A07 _h
-------	-------------------

Objektname	Transmit PDO 8 Mapping Parameter
Object Code	RECORD
Datentyp	PDO_MAPPING
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1614
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

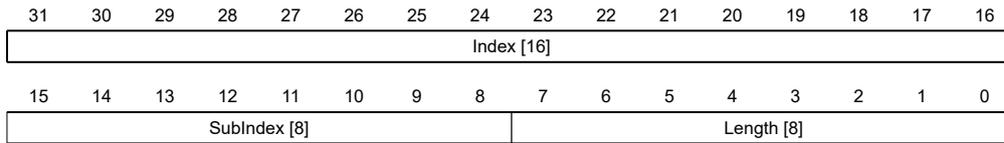
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammensetzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten.

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten.

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

1F50h Program Data

Funktion

Dieses Objekt wird zum Programmieren von Speicherbereichen der Steuerung verwendet. Jeder Eintrag steht für einen bestimmten Speicherbereich.

Objektbeschreibung

Index	1F50 _h
Objektnamen	Program Data
Object Code	ARRAY
Datentyp	DOMAIN
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h
Name	Program Data Bootloader/firmware
Datentyp	DOMAIN

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0

Subindex	02 _h
Name	Program Data NanoJ
Datentyp	DOMAIN
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0

Subindex	03 _h
Name	Program Data DataFlash
Datentyp	DOMAIN
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0

1F51h Program Control

Funktion

Dieses Objekt wird zum Steuern des Programmierens von Speicherbereichen der Steuerung verwendet. Jeder Eintrag steht für einen bestimmten Speicherbereich.

Objektbeschreibung

Index	1F51 _h
Objektname	Program Control
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
----------	-----------------

Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 _h
<hr/>	
Subindex	01 _h
Name	Program Control Bootloader/firmware
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	02 _h
Name	Program Control NanoJ
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	03 _h
Name	Program Control DataFlash
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

1F57h Program Status

Funktion

Dieses Objekt zeigt den Programmierstatus während dem Programmieren von Speicherbereichen der Steuerung an. Jeder Eintrag steht für einen bestimmten Speicherbereich.

Objektbeschreibung

Index	1F57 _h
Objektname	Program Status
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein

Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h
Name	Program Status Bootloader/firmware
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Program Status NanoJ
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	Program Status DataFlash
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

1F80h NMT Startup

Funktion

In diesem Objekt können Sie einstellen, ob nach einem Start der Steuerung automatisch in den NMT-Status *Operational* gewechselt wird. Siehe auch Kapitel **Network Management (NMT)**.

Objektbeschreibung

Index	1F80 _h
Objektname	NMT Startup
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1748-B531667
Änderungshistorie	

Beschreibung

- Wert "0"_h: der Zustand der NMT-Zustandsmaschine nach der Initialisierung ist *Pre-Operational*.
- Wert "8"_h (Bit 3): der Zustand der NMT-Zustandsmaschine ist nach der Initialisierung *Operational*.

2005h CANopen Baudrate

Funktion

Dieses Objekt enthält die Baudrate des CANopen-Busses.

Objektbeschreibung

Index	2005 _h
Objektname	CANopen Baudrate
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: CANopen
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	88 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: CANopen".

Beschreibung

Die Baudraten sind nach folgender Tabelle einzustellen. Jeder Wert außerhalb dieser Tabelle wird als 1000 kBd interpretiert.

Wert		Baudrate in kBd
dec	hex	
129	81	10
130	82	20
131	83	50
132	84	125
133	85	250
134	86	500
136	88	1000

2007h CANopen Config

Funktion

Mit diesem Objekt lassen sich verschiedene Einstellungen für CANopen vornehmen.

Objektbeschreibung

Index	2007 _h
Objektname	CANopen Config
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: CANopen
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: CANopen".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	BL Config
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- Subindex 01: Wird der Wert "1" in das Objekt geschrieben, unterdrückt der Bootloader die Boot-Up-Nachricht und nur die Firmware sendet eine BOOTUP-Nachricht. Bei einer "0" sendet der Bootloader und die Firmware jeweils eine Boot-Up-Nachricht.

2009h CANopen NodeID

Funktion

Dieses Objekt enthält die Node-ID der Steuerung. Siehe Kapitel *Inbetriebnahme*.

Objektbeschreibung

Index	2009 _h
Objektname	CANopen NodeID
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: CANopen
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	7F _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: CANopen".

2028h MODBUS Slave Address

Funktion

Dieses Objekt enthält die Slave-Adresse für Modbus. Siehe Kapitel **Kommunikationseinstellungen**.

Objektbeschreibung

Index	2028 _h
Objektname	MODBUS Slave Address
Object Code	VARIABLE

Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Modbus RTU
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	1-247
Vorgabewert	05 _h
Firmware Version	FIR-v1436
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Modbus RTU".

202Ah MODBUS RTU Baudrate

Funktion

Dieses Objekt enthält die Baudrate des Modbus in Bd. Siehe Kapitel **Kommunikationseinstellungen**.

Objektbeschreibung

Index	202A _h
Objektname	MODBUS RTU Baudrate
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Modbus RTU
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00004B00 _h
Firmware Version	FIR-v1436
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Modbus RTU".

202Ch MODBUS RTU Stop Bits

Funktion

Dieses Objekt enthält die Anzahl der Stop-Bits des Modbus.

Objektbeschreibung

Index	202C _h
Objektname	MODBUS RTU Stop Bits
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	nein
Zugriff	nur lesen

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h
Firmware Version	FIR-v1436
Änderungshistorie	Firmware Version FIR-v1540: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "nein". Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".

Beschreibung

Die Anzahl der Stopbits ist abhängig von der Parity welche im Objekt **202D_h** eingestellt werden kann.

Anzahl der Stopbits	Wert in Objekt 202C _h
1	0
2	2

202Dh MODBUS RTU Parity

Funktion

Dieses Objekt stellt bei Modbus RTU die Anzahl der Paritybits und Stopbits ein.

Objektbeschreibung

Index	202D _h
Objektname	MODBUS RTU Parity
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Modbus RTU
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Modbus RTU".

Beschreibung

Folgende Werte gelten:

- Wert "0x00": Parity None, Stop Bits 2
- Wert "0x04": Parity Even, Stop Bits 1
- Wert "0x06": Parity Odd, Stop Bits 1

2030h Pole Pair Count

Funktion

Enthält die Polpaarzahl des angeschlossenen Motors.

Objektbeschreibung

Index	2030 _h
Objektname	Pole Pair Count
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000032 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "nein" auf "ja, Kategorie: Tuning".

2031h Max Motor Current

Funktion

Hier tragen Sie den maximal zulässigen Motorstrom in Milliampere ein. Alle Stromwerte werden durch diesen Wert begrenzt.

Steuerungsintern wird der eingegebene Wert immer als Effektivwert interpretiert.

Objektbeschreibung

Index	2031 _h
Objektname	Max Motor Current
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000012C _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning". Firmware Version FIR-v1614: Eintrag "Object Name" geändert von "Peak Current" auf "Max Current". Firmware Version FIR-v1748-B538662: Eintrag "Object Name" geändert von "Maximum Current" auf "Max Motor Current".

2034h Upper Voltage Warning Level

Funktion

Dieses Objekt enthält den Schwellwert für den Fehler "Überspannung" in Millivolt.

Objektbeschreibung

Index	2034 _h
Objektname	Upper Voltage Warning Level
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000E678 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Steigt die Eingangsspannung der Steuerung über diesen Schwellwert, wird der Motor abgeschaltet und ein Fehler ausgelöst. Dieser Fehler setzt sich automatisch zurück, wenn die Eingangsspannung kleiner als (Spannung des Objekts 2034_h minus 2 Volt) ist.

2035h Lower Voltage Warning Level

Funktion

Dieses Objekt enthält den Schwellwert für den Fehler "Unterspannung" in Millivolt.

Objektbeschreibung

Index	2035 _h
Objektname	Lower Voltage Warning Level
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00002710 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Fällt die Eingangsspannung der Steuerung unter diesen Schwellwert, wird der Motor abgeschaltet und ein Fehler ausgelöst. Der Fehler setzt sich automatisch zurück, wenn die Eingangsspannung größer als die Spannung des Objekts 2035_h plus 1,5 Volt ist.

2036h Open Loop Current Reduction Idle Time

Funktion

Dieses Objekt beschreibt die Zeit in Millisekunden, die sich der Motor im Stillstand befinden muss, bis die Stromabsenkung aktiviert wird.

Objektbeschreibung

Index	2036 _h
Objektname	Open Loop Current Reduction Idle Time
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Firmware Version	FIR-v1426
Änderungshistorie	

2037h Open Loop Current Reduction Value/factor

Funktion

Dieses Objekt beschreibt den Effektivstrom, auf den der Motorstrom reduziert werden soll, wenn die Stromabsenkung im Open Loop aktiviert wird (Bit 3 in 3202_h = "1") und sich der Motor im Stillstand befindet.

Objektbeschreibung

Index	2037 _h
Objektname	Open Loop Current Reduction Value/factor
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FFFFFFCE _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert von 2037_h größer/gleich 0 und kleiner als Wert 6075_h

Strom wird auf den hier eingetragenen Wert reduziert. Der Wert wird in mA und als Effektivwert interpretiert.

Wert von 2037_h im Bereich von -1 bis -100

Der eingetragene Wert wird als eine Prozentzahl interpretiert und bestimmt die Reduktion des Nennstroms in 2037_h. Für die Berechnung wird der Wert in 6075_h herangezogen.

Beispiel: Das Objekt 6075_h hat den Wert 4200 mA. Der Wert -60 in 2037_h senkt den Strom um 60% von 6075_h ab, somit ergibt sich eine Stromabsenkung auf einen Effektivwert von 6075_h * (2037_h + 100) / 100 = 1680 mA.

Die Angabe -100 in 2037_h würde z.B. bedeuten, dass eine Stromabsenkung auf einen Effektivwert von 0 mA eingestellt wird.

2038h Brake Controller Timing

Funktion

Dieses Objekt enthält die Zeiten für die *Bremsensteuerung* in Millisekunden sowie die PWM-Frequenz und den Tastgrad.

Objektbeschreibung

Index	2038 _h
Objektname	Brake Controller Timing
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	Close Brake Idle Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	000003E8 _h
Subindex	02 _h
Name	Shutdown Power Idle Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	03 _h
Name	Open Brake Delay Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	04 _h
Name	Start Operation Delay Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	PWM Frequency
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	zwischen 0 und 2000 (7D0 _h)
Vorgabewert	00000000 _h
Subindex	06 _h
Name	PWM Duty Cycle
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	0, zwischen 2 und 100 (64 _h)
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Zeit zwischen dem Motorstillstand und dem Schließen der Bremse.
- 02_h: Zeit zwischen dem Schließen der Bremse und dem Abschalten des Motorstroms.
- 03_h: Zeit zwischen dem Einschalten des Motorstroms und dem Öffnen der Bremse.
- 04_h: Zeit zwischen dem Öffnen der Bremse und dem Erreichen des Zustands *Operation enabled* der **CiA 402 Power State Machine**.
- 05_h: Frequenz der Bremsen-PWM in Hertz.
- 06_h: Tastgrad der Bremsen-PWM in Prozent.

2039h Motor Currents

Funktion

Dieses Objekt enthält die gemessenen Motorströme in mA.

Objektbeschreibung

Index	2039 _h
Objektname	Motor Currents
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "nein" auf "TX-PDO".</p> <p>Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "TX-PDO".</p> <p>Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 03 geändert von "nein" auf "TX-PDO".</p> <p>Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 04 geändert von "nein" auf "TX-PDO".</p>

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h
Subindex	01 _h
Name	I_d
Datentyp	INTEGER32

Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	I_q
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	I_a
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	I_b
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung



Hinweis

Die Motorströme I_d (Subindex 01_h) und I_q (Subindex 02_h) werden nur angezeigt, wenn der **Closed Loop aktiviert** wurde, sonst wird der Wert 0 ausgegeben.

203Ah Homing On Block Configuration

Funktion

Dieses Objekt enthält die Parameter für das *Homing auf Block* (siehe Kapitel **Homing**)

Objektbeschreibung

Index	203A _h
-------	-------------------

Objektname	Homing On Block Configuration
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	
PDO-Mapping	
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 4 auf 3.</p> <p>Firmware Version FIR-v1540: Eintrag "Name" geändert von "Period Of Blocking" auf "Block Detection Time".</p> <p>Firmware Version FIR-v1614: Eintrag "Data Type" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".</p> <p>Firmware Version FIR-v1614: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1614: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p>

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Minimum Current For Block Detection
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FFFFFFBA _h

Subindex	02 _h
Name	Block Detection Time
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	000000C8 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Gibt den Stromgrenzwert an, ab dem ein Blockieren detektiert werden soll. Positive Zahlenwerte geben die Stromgrenze in mA an, negative Zahlen einen Prozentwert von Objekt **2031_h:01_h**. Beispiel: der Wert "1000" entspricht 1000 mA (=1 A), der Wert "-70" entspricht 70% von **2031_h**.
- 02_h: Gibt die Zeit in ms an, die der Motor nach der Blockdetektion trotzdem noch gegen den Block fahren soll.

203Bh I2t Parameters

Funktion

Dieses Objekt hält die Parameter für die I²t-Überwachung.

Die I²t-Überwachung wird aktiviert, in dem in **203B_h:01** und **203B_h:02** ein Wert größer 0 eingetragen wird und in **6073_h** ein Wert größer 1000 (siehe **I2t Motor-Überlastungsschutz**).

I²t kann nur für den *Closed Loop*-Betrieb verwendet werden, mit einer Ausnahme: Wenn I²t im *Open Loop*-Betrieb aktiviert ist, wird der Strom auf den kleineren der Werte von **203B_h:01_h**, **6073_h** und **2031_h** begrenzt.

Objektbeschreibung

Index	203B _h
Objektname	I2t Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1512: Eintrag "Savable" geändert von "nein" auf "ja, Kategorie: Applikation".</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 7 auf 8.</p> <p>Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Nominal Current" auf "Motor Rated Current".</p>

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	07 _h
<hr/>	
Subindex	01 _h
Name	Motor Rated Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000012C _h
<hr/>	
Subindex	02 _h
Name	Maximum Duration Of Peak Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03 _h
Name	Threshold
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	CalcValue
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	LimitedCurrent
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	06 _h
Name	Status
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	07 _h
Name	ActualResistance
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes sind in zwei Gruppen geteilt: Subindex 01_h und 02_h enthalten Parameter zur Steuerung, Subindex 03_h bis 06_h sind Statuswerte. Die Funktionen sind wie folgt:

- 01_h: Hier wird der im Motordatenblatt angegebene Nennstrom in mA eingetragen. Dieser muss kleiner als der eingegebene Strom in **2031_h** und **6073_h** sein, sonst wird die Überwachung nicht aktiviert. Der angegebene Wert wird als Effektivwert interpretiert.
- 02_h: Gibt die maximale Dauer des Maximalstroms (**6073_h**) in ms an.
- 03_h: Threshold, gibt die Grenze in mA an, von der abhängt, ob auf Maximalstrom oder Nennstrom geschaltet wird.
- 04_h: CalcValue, gibt den berechneten Wert an, welcher mit Threshold verglichen wird, um den Strom einzustellen.
- 05_h: LimitedCurrent, zeigt den gegenwärtigen Strom als Effektivwert an, der von I²t eingestellt wurde.
- 06_h: aktueller Status. Ist der Subentry-Wert "0", ist I²t deaktiviert, ist der Wert "1", wird I²t aktiviert.

203Dh Torque Window

Funktion

Gibt relativ zum Zieldrehmoment einen symmetrischen Bereich an, innerhalb dessen das Ziel als erreicht gilt.

Wird der Wert auf "FFFFFFF"_h gesetzt, wird die Überwachung abgeschaltet, das Bit "Target reached" im Objekt **6041_h** (Statusword) wird nie gesetzt.

Objektbeschreibung

Index	203D _h
Objektnamen	Torque Window
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben

PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

203Eh Torque Window Time Out

Funktion

Das Istdrehmoment muss sich für diese Zeit (in Millisekunden) innerhalb des "Torque Window" (**203D_h**) befinden, damit das Zieldrehmoment als erreicht gilt.

Objektbeschreibung

Index	203E _h
Objektname	Torque Window Time Out
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation". Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Torque Window Time" auf "Torque Window Time Out".

203Fh Max Slippage Time Out

Funktion

Zeit in Millisekunden, bis ein zu großer Schlupffehler im Modus **Profile Velocity** zu einer Fehlermeldung führt.

Objektbeschreibung

Index	203F _h
Objektname	Max Slippage Time Out
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	0064 _h
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Beschreibung

Weicht die Istgeschwindigkeit von der Sollgeschwindigkeit so stark ab, dass der Wert (Absolutbetrag) dieses Objekts überschritten wird, wird das Bit 13 im Objekt **6041_h** gesetzt. Die Abweichung muss länger andauern als die Zeit im Objekt **203F_h**.

Im Objekt **3700_h** kann eine Reaktion auf den Schlupffehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt **1003_h** eingetragen.

2057h Clock Direction Multiplier

Funktion

Mit diesem Wert wird der Takt-Zählwert im **Takt-Richtungs-Modus** multipliziert, bevor er weiterverarbeitet wird.

Objektbeschreibung

Index	2057 _h
Objektname	Clock Direction Multiplier
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000080 _h
Firmware Version	FIR-v1426
Änderungshistorie	

2058h Clock Direction Divider

Funktion

Durch diesen Wert wird der Takt-Zählwert im **Takt-Richtungs-Modus** dividiert, bevor er weiterverarbeitet wird.

Objektbeschreibung

Index	2058 _h
Objektname	Clock Direction Divider
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Firmware Version	FIR-v1426
Änderungshistorie	

205Ah Absolute Sensor Boot Value (in User Units)

Funktion



Tipp

Dieses Objekt hat nur bei Verwendung eines Absolut-Encoders eine Funktion. Wird kein Absolut-Encoder verwendet, ist der Wert immer 0.

Aus diesem Objekt kann die initiale Encoderposition beim Einschalten der Steuerung (in **benutzerdefinierten Einheiten**) ausgelesen werden.

Objektbeschreibung

Index	205A _h
Objektname	Absolute Sensor Boot Value (in User Units)
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1446
Änderungshistorie	<p>Firmware Version FIR-v1512: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Encoder Boot Value" auf "Absolute Sensor Boot Value (in User Units)".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".</p>

205Bh Clock Direction Or Clockwise/Counter Clockwise Mode

Funktion

Mit diesem Objekt lässt sich der Takt-Richtungs-Modus (Wert = "0") auf den **Rechts-/Linkslauf-Modus** (Wert = "1") umschalten.

Objektbeschreibung

Index	205B _h
-------	-------------------

Objektnamen	Clock Direction Or Clockwise/Counter Clockwise Mode
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1504
Änderungshistorie	

2084h Bootup Delay

Funktion

Definiert den Zeitraum zwischen Anlegen der Versorgungsspannung an die Steuerung und der Funktionsbereitschaft der Steuerung in Millisekunden.

Objektbeschreibung

Index	2084 _h
Objektnamen	Bootup Delay
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

2101h Fieldbus Module Availability

Funktion

Zeigt die verfügbaren Feldbusse an.

Objektbeschreibung

Index	2101 _h
Objektnamen	Fieldbus Module Availability
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	0019000B _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Object Name" geändert von "Fieldbus Module" auf "Fieldbus Module Availability".

Beschreibung

Die Bits 0 bis 15 zeigen die physikalische Schnittstelle an, die Bits 16 bis 31 das benutzte Protokoll (falls notwendig).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													E-IP	MTCP	MRTU
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SPI	E-CAT	E-NET	CAN	RS232	RS485	USB

USB

Wert = "1": Der Feldbus USB ist verfügbar.

RS-485

Wert = "1": Eine RS-485 Schnittstelle ist verfügbar.

RS-232

Wert = "1": Eine RS-232 Schnittstelle ist verfügbar.

CAN

Wert = "1": Der Feldbus CANopen ist verfügbar.

E-NET

Wert = "1": Eine Ethernet Schnittstelle ist verfügbar.

E-CAT

Wert = "1": Eine EtherCAT Schnittstelle ist verfügbar.

SPI

Wert = "1": Eine SPI Schnittstelle ist verfügbar.

MRTU

Wert = "1": Das benutzte Protokoll ist Modbus RTU.

MTCP

Wert = "1": Das benutzte Protokoll ist Modbus TCP

E-IP

Wert = "1": Das benutzte Protokoll ist EtherNet/IP™

2102h Fieldbus Module Control

Funktion

Mit diesem Objekt können bestimmte Feldbusse (physikalischen Schnittstellen und Protokolle) aktiviert/deaktiviert werden.

Objektbeschreibung

Index	2102 _h
Objektname	Fieldbus Module Control
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0019000B _h
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Kommunikation".

Beschreibung

Im Objekt **2103_h:1_h** werden alle physikalischen Schnittstellen/Protokolle angezeigt, welche aktiviert/deaktiviert werden können. Diese können in diesem Objekt (2102_h) geschaltet werden. Der gegenwärtige Status der aktivierten Feldbusse steht im Objekt **2103_h:2_h**.

Dabei gilt die folgende Verteilung der Bits:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													E-IP	MTCP	MRTU
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SPI	E-CAT	E-NET	CAN	RS232	RS485	USB

USB

USB Schnittstelle

RS-485

RS-485 Schnittstelle

RS-232

RS-232 Schnittstelle

CAN

CANopen Schnittstelle

E-NET

EtherNET Schnittstelle

E-CAT

EtherCAT Schnittstelle

SPI

SPI Schnittstelle

MRTU

Modbus RTU Protokoll

MTCP

Modbus TCP Protokoll

E-IP

EtherNet/IP™ Protokoll

2103h Fieldbus Module Status

Funktion

Zeigt die aktiven Feldbusse an.

Objektbeschreibung

Index	2103 _h
Objektname	Fieldbus Module Status
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Fieldbus Module Disable Mask
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000000A _h

Subindex	02 _h
Name	Fieldbus Module Enabled

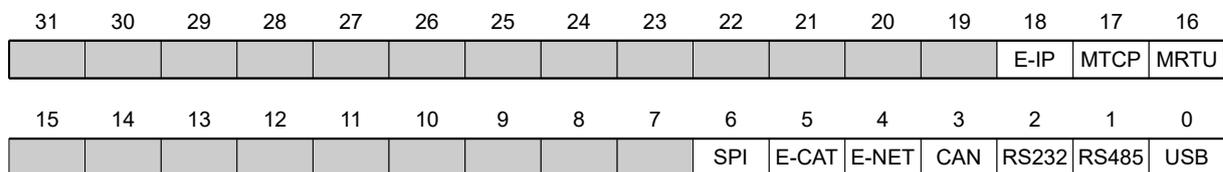
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0019000B _h

Beschreibung

Subindex 1 (Fieldbus Module Disable Mask): Im diesem Subindex werden alle physikalischen Schnittstellen und Protokolle angezeigt, welche aktiviert oder deaktiviert werden können. Ein Wert "1" bedeutet, dass dieser Feldbus deaktivierbar ist.

Subindex 2 (Fieldbus Module Enabled): Dieser Subindex zeigt alle zur Zeit aktivierten physikalischen Schnittstellen und Protokolle an. Der Wert "1" bedeutet, dass der Feldbus aktiv ist.

Für Subindex 1 und 2 gilt folgende Verteilung der Bits:



USB

USB Schnittstelle

RS-485

RS-485 Schnittstelle

RS-232

RS-232 Schnittstelle

CAN

CANopen Schnittstelle

E-NET

EtherNET Schnittstelle

E-CAT

EtherCAT Schnittstelle

SPI

SPI Schnittstelle

MRTU

Modbus RTU Protokoll

MTCP

Modbus TCP Protokoll

E-IP

EtherNet/IP™ Protokoll

2290h PDI Control

Funktion

Mit diesem Objekt können Sie das *Plug&Drive-Interface* aktivieren. Weitergehende Informationen finden Sie im Dokument *Funktionsbeschreibung Plug&Drive-Interface*.

Objektbeschreibung

Index	2290 _h
Objektnamen	PDI Control
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h
Firmware Version	FIR-v1748-B531667
Änderungshistorie	Firmware Version FIR-v1748-B538662: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "nur lesen" auf "lesen/schreiben".

Beschreibung

Um das *Plug&Drive-Interface* zu aktivieren, setzen Sie das Bit 0 auf "1".

2291h PDI Input

Funktion

Wenn Sie das *Plug&Drive-Interface* benutzen, können Sie mit diesem Objekt den Betriebsmodus wählen und starten sowie die entsprechenden Zielwerte einstellen (Zielposition, Geschwindigkeit usw.). Weitergehende Informationen finden Sie im Dokument *Funktionsbeschreibung Plug&Drive-Interface*.

Objektbeschreibung

Index	2291 _h
Objektnamen	PDI Input
Object Code	RECORD
Datentyp	PDI_INPUT
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B531667
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	04 _h

Subindex	01 _h
Name	PDI Set Value 1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	PDI Set Value 2
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	03 _h
Name	PDI Set Value 3
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h

Subindex	04 _h
Name	PDI Command
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h

2292h PDI Output

Funktion

Wenn Sie das *Plug&Drive-Interface* benutzen, können Sie in diesem Objekt den Status und einen vom verwendeten Betriebsmodus abhängigen Rückgabewert lesen. Weitergehende Informationen finden Sie im Dokument *Funktionsbeschreibung Plug&Drive-Interface*.

Objektbeschreibung

Index	2292 _h
Objektname	PDI Output
Object Code	RECORD
Datentyp	PDI_OUTPUT
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B531667
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	PDI Status
Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	02 _h
Name	PDI Return Value
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	

Vorgabewert 00000000_h

2300h NanoJ Control

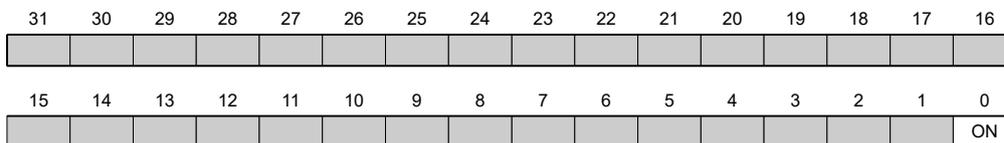
Funktion

Steuert die Ausführung eines NanoJ-Programms.

Objektbeschreibung

Index	2300 _h
Objektname	NanoJ Control
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Control" auf "NanoJ Control".

Beschreibung



ON

Schaltet das NanoJ-Programm ein (Wert = "1") oder aus (Wert = "0").

Bei einer steigenden Flanke in Bit 0 wird das Programm zuvor neu geladen und der Variablenbereich zurückgesetzt.



Hinweis

Das Starten des NanoJ Programms kann bis zu 200ms dauern.

2301h NanoJ Status

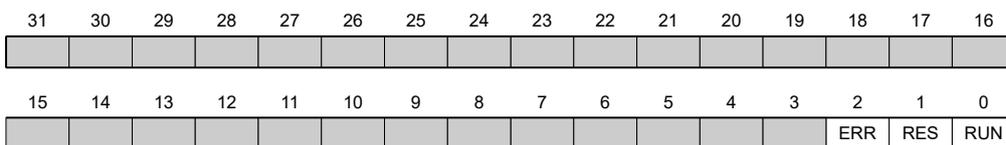
Funktion

Zeigt den Betriebszustand des Benutzerprogramms an.

Objektbeschreibung

Index	2301 _h
Objektname	NanoJ Status
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Status" auf "NanoJ Status".

Beschreibung



RUN

Wert = "0": Programm ist angehalten, Wert = "1": NanoJ-Programm läuft.

RES

Reserviert.

ERR

Programm wurde mit Fehler beendet. Fehlerursache kann aus dem Objekt **2302_h** ausgelesen werden.

2302h NanoJ Error Code

Funktion

Zeigt an, welcher Fehler bei der Ausführung des Benutzerprogramms aufgetreten ist.

Objektbeschreibung

Index	2302 _h
Objektname	NanoJ Error Code
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	

Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Error Code" auf "NanoJ Error Code".

Beschreibung

Fehlercodes bei Programmausführung:

Nummer	Beschreibung
0000 _h	Kein Fehler
0001 _h	Firmware unterstützt verwendete Funktion (noch) nicht
0002 _h	Nicht oder falsch initialisierter Pointer
0003 _h	Unerlaubter Zugriff auf System-Resource
0004 _h	Hardfault (interner Fehler)
0005 _h	Code wird zu lange ohne yield() oder sleep() ausgeführt
0006 _h	Unerlaubter Zugriff auf System-Resource
0007 _h	Zu viele Variablen auf dem Stack
0100 _h	Ungültige NanoJ Programmdatei

Fehler bei dem Zugriff auf ein Objekt:

Nummer	Beschreibung
1xxxxyy _h	Ungültiges Mapping in der NanoJ-Programmdatei: Der Wert in "xxxx" benennt den Index, der Wert in "yy" den Subindex des Objekts, das gemappt werden soll aber nicht gemappt werden kann.
1000 _h	Zugriff auf ein nicht existierendes Objekt im Objektverzeichnis
1001 _h	Schreibzugriff auf schreibgeschützten Eintrag im OD
1002 _h	Interner Dateisystemfehler

Dateisystem Fehlercodes beim Laden des Benutzerprogramms:

Nummer	Beschreibung
10002 _h	Interner Dateisystemfehler
10003 _h	Speichermedium nicht bereit
10004 _h	Datei nicht gefunden
10005 _h	Ordner nicht gefunden
10006 _h	Ungültiger Dateiname/Ordnername
10008 _h	Zugriff auf Datei nicht möglich
10009 _h	Datei/Verzeichnis Objekt ist ungültig
1000A _h	Speichermedium ist schreibgeschützt
1000B _h	Laufwerksnummer ist ungültig
1000C _h	Arbeitsbereich des Laufwerks ist ungültig
1000D _h	Kein gültiges Dateisystem auf dem Laufwerk
1000E _h	Erstellung des Dateisystems ist fehlgeschlagen
1000F _h	Zugriff innerhalb der geforderten Zeit nicht möglich

Nummer	Beschreibung
10010 _h	Zugriff wurde zurückgewiesen

230Fh Uptime Seconds

Funktion

Dieses Objekt enthält die Betriebszeit seit dem letzten Start der Steuerung in Sekunden.



Hinweis

Dieses Objekt wird nicht gespeichert, die Zählung beginnt nach dem Einschalten wieder mit "0".

Objektbeschreibung

Index	230F _h
Objektname	Uptime Seconds
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1436
Änderungshistorie	

2310h NanoJ Input Data Selection

Funktion

Beschreibt die Object Dictionary-Einträge, die in das Input PDO-Mapping des NanoJ-Programms kopiert werden.

Objektbeschreibung

Index	2310 _h
Objektname	NanoJ Input Data Selection
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1650-B472161

Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Input Data Selection" auf "NanoJ Input Data Selection".
	Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".
	Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".
	Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".

Wertebeschreibung

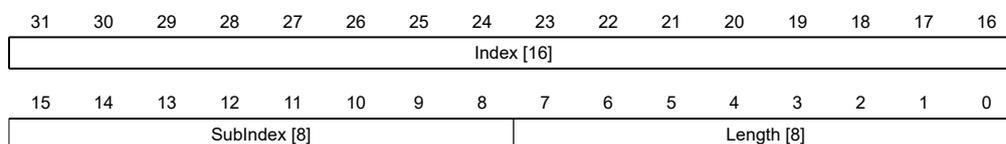
Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 _h

Subindex	01 _h - 10 _h
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-16) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammen setzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

2320h NanoJ Output Data Selection

Funktion

Beschreibt die Object Dictionary-Einträge, die in das Output PDO-Mapping des *NanoJ-Programms* kopiert werden, nachdem es ausgeführt worden ist.

Objektbeschreibung

Index	2320 _h
Objektname	NanoJ Output Data Selection
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1650-B472161
Änderungshistorie	<p>Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Output Data Selection" auf "NanoJ Output Data Selection".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p>

Wertebeschreibung

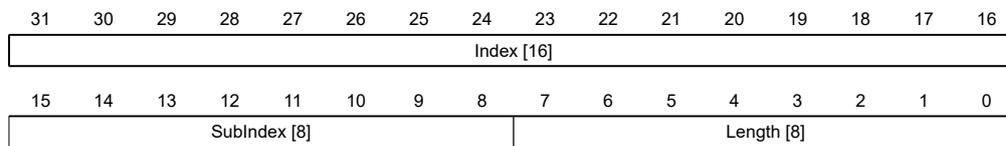
Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 _h

Subindex	01 _h - 10 _h
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-16) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping Eintrag besteht aus vier Byte welche sich nach nachfolgender Grafik zusammen setzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

2330h NanoJ In/output Data Selection

Funktion

Beschreibt die Object Dictionary-Einträge, die zunächst in das Input PDO-Mapping des NanoJ-Programms kopiert und nach dessen Ausführung wieder in das Output PDO-Mapping zurückkopiert werden.

Objektbeschreibung

Index	2330 _h
Objektname	NanoJ In/output Data Selection
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1650-B472161
Änderungshistorie	<p>Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM In/output Data Selection" auf "NanoJ In/output Data Selection".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p>

Wertebeschreibung

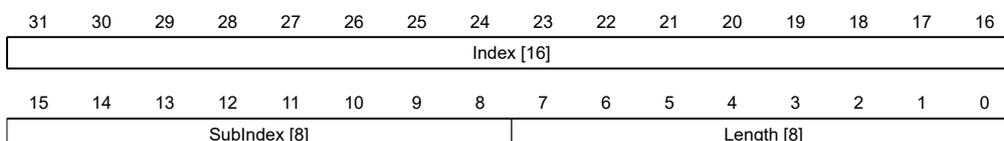
Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 _h

Subindex	01 _h - 10 _h
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-16) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammen setzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

2400h NanoJ Inputs

Funktion

Hier befindet sich ein Array mit 32 32-Bit Integerwerten, das innerhalb der Firmware nicht verwendet wird und ausschließlich zur Kommunikation mit dem Benutzerprogramm über den Feldbus dient.

Objektbeschreibung

Index	2400 _h
Objektname	NanoJ Inputs
Object Code	ARRAY

Datentyp	INTEGER32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 33 Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Inputs" auf "NanoJ Inputs". Firmware Version FIR-v1436: Eintrag "Name" geändert von "VMM Input N#" auf "NanoJ Input N#".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20 _h

Subindex	01 _h - 20 _h
Name	NanoJ Input #1 - #32
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Hier können dem *NanoJ-Programm* z. B. Vorgabewerte übergeben werden.

2410h NanoJ Init Parameters

Funktion

Dieses Objekt funktioniert identisch dem Objekt **2400_h** mit dem Unterschied, dass dieses Objekt gespeichert werden kann.

Objektbeschreibung

Index	2410 _h
Objektname	NanoJ Init Parameters
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1450
Änderungshistorie	Firmware Version FIR-v1450: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED8".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20 _h

Subindex	01 _h - 20 _h
Name	NanoJ Init Parameter #1 - #32
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

2500h NanoJ Outputs

Funktion

Hier befindet sich ein Array mit 32 32-Bit Integerwerten, das innerhalb der Firmware nicht verwendet wird und ausschließlich zur Kommunikation mit dem Benutzerprogramm über den Feldbus dient.

Objektbeschreibung

Index	2500 _h
Objektname	NanoJ Outputs
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Outputs" auf "NanoJ Outputs". Firmware Version FIR-v1436: Eintrag "Name" geändert von "VMM Output N#" auf "NanoJ Output N#".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20 _h

Subindex	01 _h - 20 _h
Name	NanoJ Output #1 - #32
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Hier kann das *NanoJ-Programm* Ergebnisse ablegen, die dann über den Feldbus ausgelesen werden können.

2600h NanoJ Debug Output

Funktion

Dieses Objekt enthält Debug-Ausgaben eines Benutzerprogramms.

Objektbeschreibung

Index	2600 _h
Objektname	NanoJ Debug Output
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Debug Output" auf "NanoJ Debug Output".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	01 _h - 40 _h
Name	Value #1 - #64
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Hier legt das NanoJ-Programm die Debug-Ausgaben ab, welche mit der Funktion `VmmDebugOutputString()`, `VmmDebugOutputInt()` und dergleichen aufgerufen wurden.

2800h Bootloader And Reboot Settings

Funktion

Mit diesem Objekt lässt sich ein Reboot der Firmware auslösen und das Kurzschließen der Motorwicklungen im Bootloader-Modus aus- und einschalten.

Objektbeschreibung

Index	2800 _h
Objektname	Bootloader And Reboot Settings
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h
Name	Reboot Command
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Reboot Delay Time In Ms
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	Bootloader HW Config
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Wird hier der Wert "746F6F62_h" eingetragen, wird die Firmware rebootet.
- 02_h: Zeit in Millisekunden: verzögert den Reboot der Firmware um die jeweilige Zeit.
- 03_h: mit dem Bit 0 kann das Kurzschließen der Motorwicklungen im Bootloader-Modus aus- und eingeschaltet werden:
 - Bit 0= 1: Das Kurzschließen der Motorwicklungen im Bootloader-Modus wird ausgeschaltet.
 - Bit 0= 0: Das Kurzschließen der Motorwicklungen im Bootloader-Modus wird eingeschaltet.

3202h Motor Drive Submode Select

Funktion

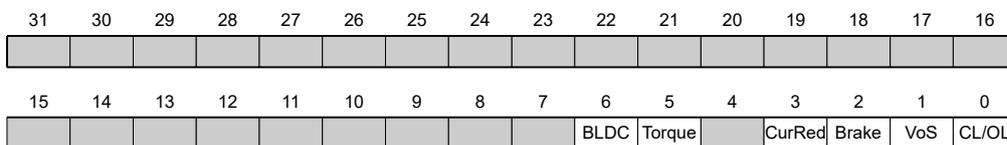
Steuert die Reglerbetriebsart, wie z. B. die *Closed Loop/ Open Loop*-Umschaltung und ob der Velocity-Mode über den S-Regler simuliert wird oder mit einem echten V-Regler im *Closed Loop* arbeitet.

Objektbeschreibung

Index	3202 _h
Objektname	Motor Drive Submode Select
Object Code	VARIABLE
Datentyp	UNSIGNED32

Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Fahrt". Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Fahrt" auf "ja, Kategorie: Bewegung".

Beschreibung



CL/OL

Umschaltung zwischen *Open Loop* und *Closed Loop*

- Wert = "0": *Open Loop*
- Wert = "1": *Closed Loop*

VoS

Wert = "1": V-Regler über eine S-Rampe simulieren: die Geschwindigkeitsmodi über kontinuierliche Positionsänderungen simulieren

Brake

Wert = "1": Einschalten der automatischen Bremsensteuerung.

CurRed (Current Reduction)

Wert = "1": Stromabsenkung im *Open Loop* aktiviert

Torque

nur in den Betriebsmodi **Profile Torque** und **Cyclic Synchronous Torque** aktiv

Wert = "1": M-Regler ist aktiv, andernfalls ist ein V-Regler überlagert: in den Torque-Modi wird kein V-Regler zur Geschwindigkeitsbegrenzung verwendet, das Objekt **6080_h** wird also ignoriert, **3210_h:3** und **3210_h:4** haben keinen Einfluss auf die Regelung.

BLDC

Wert = "1": Motortyp "BLDC" (Bürstenloser Gleichstrommotor)

3203h Feedback Selection

Funktion

In diesem Objekt werden die Quellen der Vorgaben für die Kommutierung, Geschwindigkeits- und Positionsregelung festgelegt.

Objektbeschreibung

Index	3203 _h
Objektname	Feedback Selection
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Bewegung
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662
Änderungshistorie	Firmware Version FIR-v1748-B538662: Die Anzahl der Einträge haben sich geändert von 5 auf 4.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h
Name	1st Feedback Interface
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h

Subindex	02 _h
Name	2nd Feedback Interface
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h

Subindex	03 _h
Name	3rd Feedback Interface
Datentyp	UNSIGNED8

Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:
Subindex n enthält eine Bitmaske für die jeweilige Rückführung n. Die Bits haben dabei folgende Bedeutung:
 - Bit 0: wird das Bit auf "1" gesetzt, wird die Rückführung n für die Positionsregelung verwendet.
 - Bit 1: wird das Bit auf "1" gesetzt, wird die Rückführung n für die Geschwindigkeitsregelung verwendet.
 - Bit 2: wird das Bit auf "1" gesetzt, wird die Rückführung n für die Kommutierung im **Closed Loop** verwendet.

Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Welche Rückführung die Steuerung für die einzelnen Regler (Kommutierung, Geschwindigkeit, Position) berücksichtigt, ist implizit durch die Reihenfolge der Rückführungen vorgegeben.

Das Aufsuchen beginnt immer mit Rückführung 2 und setzt sich aufsteigend fort, bis alle produktspezifisch vorhandenen Rückführungen abgefragt wurden. Wird eine Rückführung gefunden deren Parametrierung gesetzt ist, dann wird diese dem entsprechenden Regler zugeordnet und die Suche abgebrochen.



Hinweis

Wird das Bit 0 in 3202_h auf 0 gesetzt, ist der *Closed Loop* deaktiviert und somit hat das Bit 2 (Kommutierung) keine Bedeutung. Das Bit 1 für die Geschwindigkeit und das Bit 0 für die Position in den jeweiligen Subindizes werden weiterhin für die Anzeige der Positions- und Geschwindigkeits-Ist-Werten herangezogen.

3204h Feedback Mapping

Funktion

Das Objekt enthält Informationen zu den vorhandenen Rückführungen.

Objektbeschreibung

Index	3204 _h
Objektname	Feedback Mapping
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	

Firmware Version	FIR-v1748-B538662
Änderungshistorie	Firmware Version FIR-v1748-B538662: Die Anzahl der Einträge haben sich geändert von 5 auf 4.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h
Name	Index Of 1st Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	3380 _h

Subindex	02 _h
Name	Index Of 2nd Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	3390 _h

Subindex	03 _h
Name	Index Of 3rd Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	33A0 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:

Subindex n verweist auf den Index des zugehörigen Objekts für die Konfiguration der entsprechenden Rückführung.
 Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

3210h Motor Drive Parameter Set

Funktion

Beinhaltet die P- und I-Anteile der Strom-, Geschwindigkeits- und Positionsregler für *Open Loop* (nur Stromregler aktiviert) und *Closed Loop*.

Objektbeschreibung

Index	3210 _h
Objektname	Motor Drive Parameter Set
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "S_P" auf "Position Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "S_I" auf "Position Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "V_P" auf "Velocity Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "V_I" auf "Velocity Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Id_P" auf "Flux Current Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Id_I" auf "Flux Current Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Iq_P" auf "Torque Current Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Iq_I" auf "Torque Current Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "I_P" auf "Torque Current Loop, Proportional Gain (dspDrive - Stepper Motor, Open Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "I_I" auf "Torque Current Loop, Integral Gain (dspDrive - Stepper Motor, Open Loop)".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Torque Current Loop, Proportional Gain (dspDrive - Stepper Motor, Open Loop)" auf "Torque Current Loop, Proportional Gain (open Loop)".</p>

Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Torque Current Loop, Integral Gain (dspDrive - Stepper Motor, Open Loop)" auf "Torque Current Loop, Integral Gain (open Loop)".

Firmware Version FIR-v1650-B472161: Eintrag "Datentyp" geändert von "INTEGER32" auf "UNSIGNED32".

Firmware Version FIR-v1650-B472161: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED32".

Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 11 auf 13.

Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 bis 0A geändert von "nein" auf "RX-PDO".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0C _h

Subindex	01 _h
Name	Position Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000800 _h

Subindex	02 _h
Name	Position Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	Velocity Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	

Vorgabewert	00001B58 _h
Subindex	04 _h
Name	Velocity Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000004 _h
Subindex	05 _h
Name	Flux Current Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000668A0 _h
Subindex	06 _h
Name	Flux Current Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00002EE0 _h
Subindex	07 _h
Name	Torque Current Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000668A0 _h
Subindex	08 _h
Name	Torque Current Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00002EE0 _h
Subindex	09 _h

Name	Torque Current Loop, Proportional Gain (open Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001D4C0 _h
<hr/>	
Subindex	0A _h
Name	Torque Current Loop, Integral Gain (open Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000AFC8 _h
<hr/>	
Subindex	0B _h
Name	Velocity Feed Forward Factor In Per Mille
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000003E8 _h
<hr/>	
Subindex	0C _h
Name	Acceleration Feed Forward Factor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

- Subindex 00_h: Anzahl der Einträge
- Subindex 01_h: Proportionalanteil des S-Reglers (Position)
- Subindex 02_h: Integralanteil des S-Reglers (Position)
- Subindex 03_h: Proportionalanteil des V-Reglers (Geschwindigkeit)
- Subindex 04_h: Integralanteil des V-Reglers (Geschwindigkeit)
- Subindex 05_h: (Closed Loop) Proportionalanteil des Stromreglers der feldbildenden Komponente
- Subindex 06_h: (Closed Loop) Integralanteil des Stromreglers der feldbildenden Komponente
- Subindex 07_h: (Closed Loop) Proportionalanteil des Stromreglers der momentbildenden Komponente
- Subindex 08_h: (Closed Loop) Integralanteil des Stromreglers der momentbildenden Komponente
- Subindex 09_h: (Open Loop) Proportionalanteil des Stromreglers der feldbildenden Komponente
- Subindex 0A_h: (Open Loop) Integralanteil des Stromreglers der feldbildenden Komponente

- Subindex 0B_h: (Closed Loop) Geschwindigkeitsvorsteuerung in Promille. Default ist 1000 und damit ein Faktor von 1.
- Subindex 0C_h: (Closed Loop) Beschleunigungsvorsteuerung. Default ist 0 (Vorsteuerung inaktiv). Ist auch beim Verzögern wirksam.

3212h Motor Drive Flags

Funktion

Mit diesem Objekt wird bestimmt, ob das **Auto-Setup** die Regler-Parameter anpassen soll, oder nicht. Zudem kann die Richtung des Drehfeldes geändert werden.



Hinweis

Änderungen im Subindex 02_h werden erst nach einem Neustart der Steuerung aktiv. Das **Auto-Setup** muss danach erneut durchgeführt werden.

Objektbeschreibung

Index	3212 _h
Objektname	Motor Drive Flags
Object Code	ARRAY
Datentyp	INTEGER8
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1450
Änderungshistorie	Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 2 auf 3. Firmware Version FIR-v1738-B501312: Eintrag "Name" geändert von "Enable Legacy Power Mode" auf "Reserved".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h
Name	Reserved
Datentyp	INTEGER8

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	02 _h
Name	Override Field Inversion
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	03 _h
Name	Do Not Touch Controller Settings
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Für den Subindex 02_h gültige Werte:

- Wert = "0": Default-Werte der Firmware benutzen
- Wert = "1": nicht Invertieren des Drehfeldes erzwingen (mathematisch positiv)
- Wert = "-1": Invertieren des Drehfeldes erzwingen (mathematisch negativ)

Für den Subindex 03_h gültige Werte:

- Wert = "0": **Auto-Setup** erkennt den Motortyp (Schrittmotor oder BLDC-Motor) und verwendet den entsprechenden vorkonfigurierten Parametersatz.
- Wert = "1": **Auto-Setup** mit den Werten für den Regler durchführen, die vor dem Auto-Setup im Objekt 3210_h eingetragen wurden, die Werte in 3210_h werden nicht geändert.

3220h Analog Inputs

Funktion

Zeigt die Momentanwerte der Analogeingänge in Digits an.

Durch Objekt 3221_h kann der jeweilige Analogeingang als Strom- oder Spannungseingang konfiguriert werden.

Objektbeschreibung

Index	3220 _h
Objektnamen	Analog Inputs
Object Code	ARRAY
Datentyp	INTEGER16

Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Analogue Input 1
Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	02 _h
Name	Analogue Input 2
Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Beschreibung

Formeln zum Umrechnen von [digits] in die jeweilige Einheit:

- Spannungseingang: $x \text{ digits} * 10 \text{ V} / 1024 \text{ digits}$
- Stromeingang: $x \text{ digits} * 20 \text{ mA} / 1024 \text{ digits}$

3221h Analogue Inputs Control

Funktion

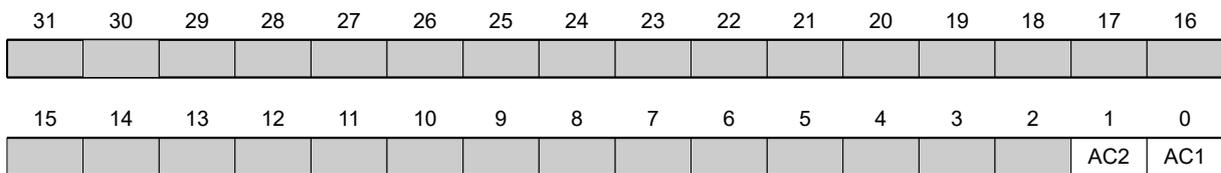
Mit diesem Objekt lässt sich ein Analog-Eingang von Spannungs- auf Strommessung umschalten.

Objektbeschreibung

Index	3221 _h
Objektname	Analogue Inputs Control

Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung



Generell gilt: Wird ein Bit auf den Wert "0" gesetzt, misst der Analogeingang die Spannung, ist das Bit auf den Wert "1" gesetzt, wird der Strom gemessen.

AC1

Einstellung für Analogeingang 1

AC2

Einstellung für Analogeingang 2

3240h Digital Inputs Control

Funktion

Mit diesem Objekt lassen sich digitale Eingänge manipulieren wie in Kapitel **Digitale Ein- und Ausgänge** beschrieben.

Objektbeschreibung

Index	3240 _h
Objektname	Digital Inputs Control
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Subindex 01 _h : Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable" Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 8 auf 9.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	08 _h

Subindex	01 _h
Name	Special Function Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	05 _h
----------	-----------------

Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06 _h
Name	Input Range Select
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07 _h
Name	Differential Select
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	Routing Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- **3240_h:01_h** (Special Function Enable): Dieses Bit erlaubt Sonderfunktionen eines Eingangs aus- (Wert "0") oder einzuschalten (Wert "1"). Soll Eingang 1 z.B. nicht als negativer Endschalter verwendet werden, so muss die Sonderfunktion abgeschaltet werden, damit nicht fälschlicherweise auf den Signalgeber reagiert wird. Auf die Bits 16 bis 31 hat das Objekt keine Auswirkungen. Die Firmware wertet folgende Bits aus:

- Bit 0: Negativer Endschalter
- Bit 1: Positiver Endschalter
- Bit 2: Referenzschalter

Sollen z.B. zwei Endschalter und ein Referenzschalter verwendet werden, müssen Bits 0-2 in **3240_h:01_h** auf "1" gesetzt werden

- **3240_h:02_h** (Function Inverted): Dieser Subindex wechselt von Schließer-Logik (ein logischer High-Pegel am Eingang ergibt den Wert "1" im Objekt **60FD_h**) auf Öffner-Logik (der logische High-Pegel am Eingang ergibt den Wert "0").
 Das gilt für die Sonderfunktionen (außer den Takt- und Richtungseingängen) und für die normalen Eingänge. Hat das Bit den Wert "0" gilt Schließer-Logik, entsprechend bei dem Wert "1" die Öffner-Logik. Bit 0 wechselt die Logik des Eingangs 1, Bit 1 die Logik des Eingangs 2 usw. .
- **3240_h:03_h** (Force Enable): Dieser Subindex schaltet die Softwaresimulation von Eingangswerten ein, wenn das entsprechende Bit auf "1" gesetzt ist.
 Dann werden nicht mehr die tatsächlichen, sondern die in Objekt **3240_h:04_h** eingestellten Werte für den jeweiligen Eingang verwendet. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw..
- **3240_h:04_h** (Force Value): Dieses Bit gibt den Wert vor, der als Eingangswert eingelesen werden soll, wenn das gleiche Bit im Objekt **3240_h:03_h** gesetzt wurde.
- **3240_h:05_h** (Raw Value): Dieses Objekt beinhaltet den unmodifizierten Eingabewert.

3242h Digital Input Routing

Funktion

Dieses Objekt bestimmt die Quelle des Inputroutings, die im **60FD_h** endet.

Objektbeschreibung

Index	3242 _h
Objektname	Digital Input Routing
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1504
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24 _h

Subindex	01 _h - 24 _h
Name	Input Source #1 - #36
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO

Zulässige Werte

Vorgabewert 00_h

Beschreibung

Der Subindex 01_h enthält die Quelle für das Bit 0 des Objekts **60FD**. Der Subindex 02_h enthält die Quelle für das Bit 1 des Objekts **60FD** und so weiter.

Die Nummer, die in eine Subindex geschrieben wird, bestimmt die Quelle für das zugehörige Bit. Die folgende Tabelle listet alle möglichen Signalquellen auf.

Nummer		
dec	hex	Signalquelle
00	00	Signal ist immer 0
01	01	Physikalischer Eingang 1
02	02	Physikalischer Eingang 2
03	03	Physikalischer Eingang 3
04	04	Physikalischer Eingang 4
05	05	Physikalischer Eingang 5
06	06	Physikalischer Eingang 6
07	07	Physikalischer Eingang 7
08	08	Physikalischer Eingang 8
09	09	Physikalischer Eingang 9
10	0A	Physikalischer Eingang 10
11	0B	Physikalischer Eingang 11
12	0C	Physikalischer Eingang 12
13	0D	Physikalischer Eingang 13
14	0E	Physikalischer Eingang 14
15	0F	Physikalischer Eingang 15
16	10	Physikalischer Eingang 16
65	41	Hall Eingang "U"
66	42	Hall Eingang "V"
67	43	Hall Eingang "W"
68	44	Encoder Eingang "A"
69	45	Encoder Eingang "B"
70	46	Encoder Eingang "Index"
71	47	USB Power Signal
128	80	Signal ist immer 1
129	81	Invertierter physikalischer Eingang 1
130	82	Invertierter physikalischer Eingang 2
131	83	Invertierter physikalischer Eingang 3
132	84	Invertierter physikalischer Eingang 4
133	85	Invertierter physikalischer Eingang 5
134	86	Invertierter physikalischer Eingang 6
135	87	Invertierter physikalischer Eingang 7
136	88	Invertierter physikalischer Eingang 8
137	89	Invertierter physikalischer Eingang 9
138	8A	Invertierter physikalischer Eingang 10

Nummer		
dec	hex	Signalquelle
139	8B	Invertierter physikalischer Eingang 11
140	8C	Invertierter physikalischer Eingang 12
141	8D	Invertierter physikalischer Eingang 13
142	8E	Invertierter physikalischer Eingang 14
143	8F	Invertierter physikalischer Eingang 15
144	90	Invertierter physikalischer Eingang 16
193	C1	Invertierter Hall Eingang "U"
194	C2	Invertierter Hall Eingang "V"
195	C3	Invertierter Hall Eingang "W"
196	C4	Invertierter Encoder Eingang "A"
197	C5	Invertierter Encoder Eingang "B"
198	C6	Invertierter Encoder Eingang "Index"
199	C7	Invertiertes USB Power Signal

3243h Digital Input Homing Capture

Funktion

Mit diesem Objekt kann automatisch die Encoderposition notiert werden, wenn am digitalen Eingang, der für den Referenzschalter verwendet wird, ein Pegelwechsel stattfindet.



Hinweis

Verwenden Sie diese Funktion nicht in Kombination mit einer Referenzfahrt. Sonst kann die Referenzfahrt nicht erfolgreich abgeschlossen werden.

Objektbeschreibung

Index	3243 _h
Objektname	Digital Input Homing Capture
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8

Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h
<hr/>	
Subindex	01 _h
Name	Control
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	Capture Count
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03 _h
Name	Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	Sensor Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

- Subindex 01_h: damit wird der Typ des Pegelwechsels ausgewählt:
 - Funktion deaktivieren: Wert "0"
 - Mit steigender Flanke: Wert "1"
 - Mit fallender Flanke: Wert "2"
 - Beide Flanken: Wert "3"

- Subindex 02_h: gibt die Anzahl der notierten Pegelwechsel seit dem letzten Start der Funktion wieder; wird auf 0 zurückgesetzt, wenn Subindex 01_h auf 1,2 oder 3 gesetzt wird
- Subindex 03_h: Encoder Position des Pegelwechsels (in absoluten Benutzereinheiten aus **6064_h**)
- Subindex 04_h: Encoder Position des Pegelwechsels

3250h Digital Outputs Control

Funktion

Mit diesem Objekt lassen sich die digitalen Ausgänge steuern, wie in Kapitel " **Digitale Ein- und Ausgänge**" beschrieben.

Dabei gilt für alle Subindizes:

- Bit 0 bis 15 steuern die Spezialfunktionen.
- Bit 16 bis 31 steuern die Pegel der Ausgänge.

Objektbeschreibung

Index	3250 _h
Objektnamen	Digital Outputs Control
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Subindex 01 _h : Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable" Firmware Version FIR-v1446: Eintrag "Name" geändert von "Special Function Enable" auf "No Function". Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 6 auf 9.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	08 _h

Subindex	01 _h
Name	No Function
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	05 _h
Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	06 _h
Name	Reserved1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	07 _h
Name	Reserved2
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	Routing Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Ohne Funktion.
- 02_h: Mit diesem Subindex wird die Logik invertiert (von Öffner-Logik auf Schließer-Logik).
- 03_h: Mit diesem Subindex wird der Ausgangswert erzwungen, wenn das Bit den Wert "1" hat. Der Pegel des Ausgangs wird in Subindex 4_h festgelegt.
- 04_h: Mit diesem Subindex wird der am Ausgang anzulegende Pegel definiert. Der Wert "0" liefert am digitalen Ausgang einen logischen Low-Pegel, der Wert "1" entsprechend einen logischen High-Pegel.
- 05_h: In diesem dem Subindex wird die an die Ausgänge gelegte Bitkombination abgelegt.

3252h Digital Output Routing

Funktion

Dieses Objekt weist einem Ausgang eine Signalquelle zu, die mit dem **60FE_h** kontrolliert werden kann.

Objektbeschreibung

Index	3252 _h
Objektname	Digital Output Routing
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	05 _h

Subindex	01 _h
Name	Output Control #1
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	1080 _h

Subindex	02 _h
Name	Output Control #2
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0090 _h

Subindex	03 _h
Name	Output Control #3
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0091 _h

Subindex	04 _h
Name	Output Control #4
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0092 _h

Subindex	05 _h
----------	-----------------

Name	Output Control #5
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0093 _h

3320h Read Analogue Input

Funktion

Zeigt die Momentanwerte der Analogeingänge in benutzerdefinierten Einheiten an.

Objektbeschreibung

Index	3320 _h
Objektnamen	Read Analogue Input
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Analogue Inputs
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Analogue Input 1
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Analogue Input 2
Datentyp	INTEGER32
Zugriff	nur lesen

PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die benutzerdefinierten Einheiten setzen sich aus Offset (**3321_h**) und Pre-scaling Wert (**3322_h**) zusammen. Sind beide Objekteinträge noch mit Default-Werten beschrieben, wird der Wert in **3320_h** in der Einheit "ADC digits" angegeben.

Formel zum Umrechnen von digits in die jeweilige Einheit:

- Spannungseingang: $x \text{ digits} * 10 \text{ V} / 1024 \text{ digits}$
- Stromeingang: $x \text{ digits} * 20 \text{ mA} / 1024 \text{ digits}$

Für die Subeinträge gilt:

- Subindex 00_h: Anzahl der Analogeingänge
- Subindex 01_h: Analogwert 1
- Subindex 02_h: Analogwert 2

3321h Analogue Input Offset

Funktion

Offset, der zum eingelesenen Analogwert (**3320_h**) addiert wird, bevor die Teilung mit dem Teiler aus dem Objekt **3322_h** vorgenommen wird.

Objektbeschreibung

Index	3321 _h
Objektname	Analogue Input Offset
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Analogue Inputs
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Analogue Input 1
Datentyp	INTEGER32

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Analogue Input 2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

- Subindex 00_h: Anzahl der Offsets
- Subindex 01_h: Offset für Analogeingang 1
- Subindex 02_h: Offset für Analogeingang 2

3322h Analogue Input Pre-scaling

Funktion

Wert, mit dem der eingelesene Analogwert (3320_h, 3321_h) dividiert wird, bevor er in das Objekt 3320_h geschrieben wird.

Objektbeschreibung

Index	3322 _h
Objektname	Analogue Input Pre-scaling
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Analogue Inputs
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Analogue Input 1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	alle Werte zulässig außer 0
Vorgabewert	00000001 _h

Subindex	02 _h
Name	Analogue Input 2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	alle Werte zulässig außer 0
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes enthalten:

- Subindex 00_h: Anzahl der Teiler
- Subindex 01_h: Teiler für Analogeingang 1
- Subindex 02_h: Teiler für Analogeingang 2

3390h Feedback Hall

Funktion

Enthält Konfigurationswerte für die Hall-Sensoren. Die Werte werden vom **Auto-Setup** ermittelt.

Objektbeschreibung

Index	3390 _h
Objektname	Feedback Hall
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B531667
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported

Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0C _h
<hr/>	
Subindex	01 _h
Name	1st Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	02 _h
Name	2nd Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	03 _h
Name	3rd Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	04 _h
Name	4th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	05 _h
Name	5th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	06 _h
Name	6th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	07 _h
Name	7th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	08 _h
Name	8th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	09 _h
Name	9th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	0A _h
Name	10th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	0B _h
Name	11th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	0C _h
Name	12th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

33A0h Feedback Incremental A/B/I 1

Funktion

Enthält Konfigurationswerte für den ersten inkrementalen Encoder. Die Werte werden vom **Auto-Setup** ermittelt.

Objektbeschreibung

Index	33A0 _h
Objektname	Feedback Incremental A/B/I 1
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Configuration
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	02 _h
Name	Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h (Configuration): folgende Bits haben eine Bedeutung:
 - Bit 0: Wert = "0": Der Encoder verfügt nicht über einen Index. Wert = "1" : Encoder-Index gefunden und soll verwendet werden.
 - Bit 15: Wert = "1": der Encoder ist ein Singleturn-Absolut-Encoder.
- 01_h (Alignment): Dieser Wert gibt den Versatz zwischen dem Index des Encoders und dem elektrischen Feld an.
 Die exakte Bestimmung ist nur über das **Auto-Setup** möglich. Das Vorhandensein dieses Wertes ist für den *Closed Loop*-Betrieb mit Encoder erforderlich.

3502h MODBUS Rx PDO Mapping

Funktion

In dieses Objekt können die Objekte für das RX Mapping geschrieben werden.

Objektbeschreibung

Index	3502 _h
Objektname	MODBUS Rx PDO Mapping
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662

Änderungshistorie

Firmware Version FIR-v1738-B505321: Eintrag "Object Name" geändert von "MODBUS Rx PDO-Mapping" auf "MODBUS Rx PDO Mapping".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #1" auf "1st Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #2" auf "2nd Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #3" auf "3rd Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #4" auf "4th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #5" auf "5th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #6" auf "6th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #7" auf "7th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #8" auf "8th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #9" auf "9th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #10" auf "10th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #11" auf "11th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #12" auf "12th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #13" auf "13th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #14" auf "14th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #15" auf "15th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #16" auf "16th Object To Be Mapped".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	08 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60400010 _h
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00050008 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60600008 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	32020020 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	607A0020 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60810020 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60420010 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60FE0120 _h
<hr/>	
Subindex	09 _h
Name	9th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0A _h
Name	10th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0B _h
Name	11th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	0C _h
Name	12th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0D _h
Name	13th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0E _h
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0F _h
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	10 _h
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

3602h MODBUS Tx PDO Mapping

Funktion

In dieses Objekt können die Objekte für das TX Mapping geschrieben werden.

Objektbeschreibung

Index	3602 _h
Objektname	MODBUS Tx PDO Mapping
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662
Änderungshistorie	<p>Firmware Version FIR-v1738-B505321: Eintrag "Object Name" geändert von "MODBUS Tx PDO-Mapping" auf "MODBUS Tx PDO Mapping".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #1" auf "1st Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #2" auf "2nd Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #3" auf "3rd Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #4" auf "4th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #5" auf "5th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #6" auf "6th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #7" auf "7th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #8" auf "8th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #9" auf "9th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #10" auf "10th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #11" auf "11th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #12" auf "12th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #13" auf "13th Object To Be Mapped".</p>

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #14" auf "14th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #15" auf "15th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #16" auf "16th Object To Be Mapped".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60410010 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00050008 _h

Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60610008 _h

Subindex	04 _h
Name	4th Object To Be Mapped

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60640020 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60440010 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60FD0020 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	09 _h
Name	9th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0A _h
Name	10th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0B _h
Name	11th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0C _h
Name	12th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0D _h
Name	13th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0E _h
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	0F _h
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	10 _h
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

3700h Deviation Error Option Code

Funktion

Das Objekt enthält die auszuführende Aktion, wenn ein Schlepp- oder Schlupffehler ausgelöst wird.

Objektbeschreibung

Index	3700 _h
Objektname	Deviation Error Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FFFF _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Following Error Option Code" auf "Deviation Error Option Code".

Beschreibung

Wert	Beschreibung
-32768 bis -2	Reserviert
-1	Keine Reaktion
0	Soforthalt
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)

Wert	Beschreibung
2	Abbremsen mit "quick stop ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
3 bis 32767	Reserviert

3701h Limit Switch Error Option Code

Funktion

Wird ein Endschalter überfahren, wird das Bit 7 (*Warning*) in **6041_h** (*Statusword*) gesetzt und die in diesem Objekt hinterlegte Aktion ausgeführt.

Objektbeschreibung

Index	3701 _h
Objektname	Limit Switch Error Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FFFF _h
Firmware Version	FIR-v1748-B538662
Änderungshistorie	

Beschreibung

Wert in Objekt 3701 _h	Beschreibung
-1	keine Reaktion (um z. B. eine Referenzfahrt durchzuführen)
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.
6	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

4012h HW Information

Funktion

Dieses Objekt zeigt Informationen über die Hardware an.

Objektbeschreibung

Index	4012 _h
Objektname	HW Information
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	EEPROM Size In Bytes
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Subindex 01: Zeigt die Größe des angeschlossenen EEPROMS in Bytes an. Der Wert "0" bedeutet, dass kein EEPROM angeschlossen ist.

4013h HW Configuration

Funktion

Mit diesem Objekt kann man bestimmte Hardware-Konfigurationen einstellen.

Objektbeschreibung

Index	4013 _h
Objektname	HW Configuration
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	HW Configuration #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Bit 0 : reserviert

4014h Operating Conditions

Funktion

Dieses Objekt dient zum Auslesen aktueller Umgebungswerte der Steuerung.

Objektbeschreibung

Index	4014 _h
Objektname	Operating Conditions
Object Code	ARRAY

Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	<p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 02 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Temperature PCB [d°C]" auf "Temperature PCB [Celsius * 10]".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 03 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 4 auf 6.</p>

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	05 _h

Subindex	01 _h
Name	Voltage UB Power [mV]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Voltage UB Logic [mV]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	Temperature PCB [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04 _h
Name	Temperature Motor [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	Temperature Microcontroller Chip [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes enthalten:

- 01_h: aktuelle Versorgungsspannung in [mV]
- 02_h: aktuelle Logikspannung in [mV]
- 03_h: aktuelle Temperatur der Steuerungsplatine in [d°C] (Zehntelgrad)
- 04_h: reserviert
- 05_h: reserviert

4040h Drive Serial Number

Funktion

Dieses Objekt hält die Seriennummer der Steuerung.

Objektbeschreibung

Index	4040 _h
Objektname	Drive Serial Number
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1450
Änderungshistorie	

4041h Device Id

Funktion

Dieses Objekt hält die ID des Geräts.

Objektbeschreibung

Index	4041 _h
Objektname	Device Id
Object Code	VARIABLE
Datentyp	OCTET_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1540
Änderungshistorie	

603Fh Error Code

Funktion

Dieses Objekt liefert den Error-Code des letzten aufgetretenen Fehlers.

Er entspricht den unteren 16-Bits des Objekts **1003_h**. Für die Beschreibung der Error-Codes schauen Sie unter Objekt **1003_h** nach.

Objektbeschreibung

Index	603F _h
Objektname	Error Code
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Bedeutung des Fehlers siehe Objekt **1003_h** (Pre-defined Error Field).

6040h Controlword

Funktion

Dieses Objekt steuert die **CiA 402 Power State Machine**.

Objektbeschreibung

Index	6040 _h
Objektname	Controlword
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Teile des Objektes sind in der Funktion abhängig vom aktuell gewählten Modus.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						OMS	HALT	FR		OMS [3]		EO	QS	EV	SO

SO (Switched On)

Wert = "1": Schaltet in den Zustand "Switched on"

EV (Enable Voltage)

Wert = "1": Schaltet in den Zustand "Enable voltage"

QS (Quick Stop)

Wert = "0": Schalten in den Zustand "Quick stop"

EO (Enable Operation)

Wert = "1": Schalten in den Zustand "Enable operation"

OMS (Operation Mode Specific)

Bedeutung abhängig vom gewählten Betriebsmodus

FR (Fault Reset)

Setzt einen Fehler zurück (falls möglich)

HALT

Wert = "1": Löst einen Halt aus, gültig in folgenden Modi:

- **Profile Position**

- **Velocity**
- **Profile Velocity**
- **Profile Torque**
- **Interpolated Position Mode**

6041h Statusword

Funktion

Dieses Objekt liefert Informationen zum Status der **CiA 402 Power State Machine**.

Objektbeschreibung

Index	6041 _h
Objektname	Statusword
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Teile des Objektes sind in der Funktion abhängig vom aktuell gewählten Modus. Schlagen Sie im entsprechenden Unterkapitel im Kapitel **Betriebsmodi** nach.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLA		OMS [2]	ILA	TARG	REM	SYNC	WARN	SOD	QS	VE	FAULT	OE	SO	RTSO	

RTSO (Ready To Switch On)

Wert = "1": Steuerung befindet sich in dem Zustand "Ready To Switch On"

SO (Switched On)

Wert = "1": Steuerung befindet sich in dem Zustand "Switched On"

OE (Operation Enabled)

Wert = "1": Steuerung befindet sich in dem Zustand "Operation Enabled"

FAULT

Fehler vorgefallen

VE (Voltage Enabled)

Spannung angelegt

QS (Quick Stop)

Wert = "0": Steuerung befindet sich in dem Zustand "Quick Stop"

SOD (Switched On Disabled)

Wert = "1": Steuerung befindet sich in dem Zustand "Switched On Disabled"

WARN (Warning)

Wert = "1": Warnung

SYNC (Synchronisation)

Wert = "1": Steuerung ist synchron zum Feldbus, Wert = "0": Steuerung ist nicht synchron zum Feldbus

REM (Remote)

Remote (Wert des Bits immer "1")

TARG

Zielvorgabe erreicht

ILA (Internal Limit Active)

Limit überschritten

OMS (Operation Mode Specific)

Bedeutung abhängig vom gewählten Betriebsmodus

CLA (Closed Loop Active)

Wert = "1": die Steuerung befindet sich im Status *Operation enabled* und der **Closed Loop** ist aktiviert.

In der nachfolgenden Tabelle sind die Bitmasken aufgelistet, die den Zustand der Steuerung aufschlüsseln.

Statusword (6041 _h)	Zustand
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

6042h VI Target Velocity

Funktion

Gibt die Zielgeschwindigkeit für den **Velocity** Modus in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6042 _h
Objektnamen	VI Target Velocity
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation

Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00C8 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

6043h VI Velocity Demand

Funktion

Vorgabegeschwindigkeit in **benutzerdefinierten Einheiten** für den Regler im **Velocity** Mode.

Objektbeschreibung

Index	6043 _h
Objektname	VI Velocity Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6044h VI Velocity Actual Value

Funktion

Gibt die aktuelle Istgeschwindigkeit im **Velocity** Modus in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6044 _h
Objektname	VI Velocity Actual Value
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6046h VI Velocity Min Max Amount

Funktion

Mit diesem Objekt können Minimalgeschwindigkeit und Maximalgeschwindigkeit in **benutzerdefinierten Einheiten** eingestellt werden.

Objektbeschreibung

Index	6046 _h
Objektname	VI Velocity Min Max Amount
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	MinAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	MaxAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00004E20 _h

Beschreibung

Subindex 1 enthält die Minimalgeschwindigkeit.

Subindex 2 enthält die Maximalgeschwindigkeit.

Wird eine Zielgeschwindigkeit (Objekt **6042_h**) vom Betrag her kleiner als die Minimalgeschwindigkeit angegeben, gilt die Minimalgeschwindigkeit und das Bit 11 (Internal Limit Reached) in **6041_h Statusword_h** wird gesetzt.

Eine Zielgeschwindigkeit größer als die Maximalgeschwindigkeit setzt die Geschwindigkeit auf die Maximalgeschwindigkeit und das Bit 11 (Internal Limit Reached) in **6041_h Statusword_h** wird gesetzt.

6048h VI Velocity Acceleration

Funktion

Setzt die Beschleunigungsrampe im Velocity Mode (siehe **Velocity**).

Objektbeschreibung

Index	6048 _h
Objektname	VI Velocity Acceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h

Subindex	02 _h
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	0001 _h

Beschreibung

Die Beschleunigung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01_h: enthält die Geschwindigkeitsänderung.

Subindex 02_h: enthält die Zeitänderung.

6049h VI Velocity Deceleration

Funktion

Setzt die Verzögerung (Bremsrampe) im Velocity Mode (siehe **Velocity**).

Objektbeschreibung

Index	6049 _h
Objektname	VI Velocity Deceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h

Subindex	02 _h
----------	-----------------

Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 _h

Beschreibung

Die Verzögerung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01_h: enthält die Geschwindigkeitsänderung.

Subindex 02_h: enthält die Zeitänderung.

604Ah VI Velocity Quick Stop

Funktion

Dieses Objekt definiert die Verzögerung (Bremsrampe), wenn im **Velocity Mode** der Quick Stop-Zustand eingeleitet wird.

Objektbeschreibung

Index	604A _h
Objektname	VI Velocity Quick Stop
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	00001388 _h
<hr/>	
Subindex	02 _h
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 _h

Beschreibung

Die Verzögerung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01_h: enthält die Geschwindigkeitsänderung.

Subindex 02_h: enthält die Zeitänderung.

604Ch VI Dimension Factor

Funktion

Hier wird die Einheit der Geschwindigkeitsangaben für die Objekte festgelegt, welche den **Velocity Mode** betreffen.

Objektbeschreibung

Index	604C _h
Objektname	VI Dimension Factor
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
<hr/>	
Subindex	01 _h

Name	VI Dimension Factor Numerator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	02 _h
Name	VI Dimension Factor Denominator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Der Subindex 1 enthält den Zähler (Multiplikator) und der Subindex 2 den Nenner (Divisor), mit dem interne Geschwindigkeitsangaben in Umdrehungen pro Minute verrechnet werden. Wird z.B. Subindex 1 auf den Wert "60" und Subindex 2 auf den Wert "1" eingestellt, erfolgt die Geschwindigkeitsangabe in Umdrehungen pro Sekunde (60 Umdrehungen pro 1 Minute).

605Ah Quick Stop Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der **CiA 402 Power State Machine** in den Zustand *Quick Stop active*.

Objektbeschreibung

Index	605A _h
Objektname	Quick Stop Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0002 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert in Objekt 605A _h	Beschreibung
0	Soforthalt mit anschließendem Zustandswechsel in <i>Switch on disabled</i>

Wert in Objekt 605A _h	Beschreibung
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.
6	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

605Bh Shutdown Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der **CiA 402 Power State Machine** vom Zustand *Operation enabled* in den Zustand *Ready to switch on*.

Objektbeschreibung

Index	605B _h
Objektname	Shutdown Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart) und anschließendem Zustandswechsel zu "Switch on disabled"
2 bis 32767	Reserviert

605Ch Disable Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der **CiA 402 Power State Machine** vom Zustand *Operation enabled* in den Zustand *Switched on*.

Objektbeschreibung

Index	605C _h
Objektname	Disable Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart) und anschließendem Zustandswechsel zu "Switch on disabled"
2 bis 32767	Reserviert

605Dh Halt Option Code

Funktion

Das Objekt enthält die auszuführende Aktion, wenn im Controlword **6040_h** das Bit 8 (Halt) gesetzt wird.

Objektbeschreibung

Index	605D _h
Objektname	Halt Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	FIR-v1426

Änderungshistorie

Beschreibung

Wert	Beschreibung
-32768 bis 0	Reserviert
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
2	Abbremsen mit "quick stop ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
3 bis 32767	Reserviert

605Eh Fault Option Code

Funktion

Das Objekt enthält die auszuführende Aktion, wie der Motor im Fehlerfall zum Stillstand gebracht werden soll.

Objektbeschreibung

Index	605E _h
Objektname	Fault Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0002 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
2	Abbremsen mit "quick stop ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
3 bis 32767	Reserviert

6060h Modes Of Operation

Funktion

In dieses Objekt wird der gewünschte Betriebsmodus eingetragen.

Objektbeschreibung

Index	6060 _h
Objektname	Modes Of Operation
Object Code	VARIABLE
Datentyp	INTEGER8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Modus	Beschreibung
-2	Auto-Setup
-1	Takt-Richtungsmodus
0	No mode change/no mode assigned
1	Profile Position Mode
2	Velocity Mode
3	Profile Velocity Mode
4	Profile Torque Mode
5	Reserved
6	Homing Mode
7	Interpolated Position Mode
8	Cyclic Synchronous Position Mode
9	Cyclic Synchronous Velocity Mode
10	Cyclic Synchronous Torque Mode

6061h Modes Of Operation Display

Funktion

Zeigt den aktuellen Betriebsmodus. Siehe auch **6060h Modes Of Operation**.

Objektbeschreibung

Index	6061 _h
-------	-------------------

Objektnamen	Modes Of Operation Display
Object Code	VARIABLE
Datentyp	INTEGER8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6062h Position Demand Value

Funktion

Gibt die aktuelle Sollposition in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6062 _h
Objektnamen	Position Demand Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6063h Position Actual Internal Value

Funktion

Enthält die aktuelle Drehgeberposition in Inkrementen. Im Gegensatz zu den Objekten **6062_h** und **6064_h** wird dieser Wert nach einem **Homing** nicht auf "0" gesetzt.



Hinweis

Ist die Encoderauflösung im Objekt **608F_h** = 0, sind die Zahlenwerte dieses Objekts ungültig.

Objektbeschreibung

Index	6063 _h
Objektnamen	Position Actual Internal Value
Object Code	VARIABLE
Datentyp	INTEGER32

Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6064h Position Actual Value

Funktion

Enthält die aktuelle Istposition in **benutzerdefinierten Einheiten**.

Objektbeschreibung

Index	6064 _h
Objektname	Position Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6065h Following Error Window

Funktion

Definiert den maximal erlaubten **Schleppfehler** in **benutzerdefinierten Einheiten** symmetrisch zur **Sollposition**.

Objektbeschreibung

Index	6065 _h
Objektname	Following Error Window
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000100 _h
Firmware Version	FIR-v1426

Änderungshistorie

Firmware Version FIR-v1504: Eintrag "Savable" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Weicht die Istposition von der Sollposition so stark ab, dass der Wert dieses Objekts überschritten wird, wird das Bit 13 im Objekt **6041_h** gesetzt. Die Abweichung muss länger andauern als die Zeit in dem Objekt **6066_h**.

Wird der Wert des "Following Error Window" auf "FFFFFFF"_h gesetzt, wird die Schleppfehler-Überwachung abgeschaltet.

In dem Objekt **3700_h** kann eine Reaktion auf den Schleppfehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt **1003_h** eingetragen.

6066h Following Error Time Out

Funktion

Zeit in Millisekunden, bis ein zu großer Schleppfehler zu einer Fehlermeldung führt.

Objektbeschreibung

Index	6066 _h
Objektname	Following Error Time Out
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0064 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1504: Eintrag "Savable" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Weicht die Istposition von der Sollposition so stark ab, dass der Wert des Objekts **6065_h** überschritten wird, wird das Bit 13 im Objekt **6041_h** gesetzt. Die Abweichung muss länger als die Zeit in diesem Objekt anhalten.

In dem Objekt **3700_h** kann eine Reaktion auf den Schleppfehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt **1003_h** eingetragen.

6067h Position Window

Funktion

Gibt relativ zur Zielposition einen symmetrischen Bereich an, innerhalb dessen das Ziel als erreicht gilt in den Modi **Profile Position** und **Interpolated Position Mode**.

Objektbeschreibung

Index	6067 _h
Objektname	Position Window
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000000A _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1504: Eintrag "Savable" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Ist die Abweichung der Istposition zur Zielposition kleiner als der Wert dieses Objekts, wird das Bit 10 im Objekt **6041_h** gesetzt. Die Bedingung muss länger erfüllt sein als die im Objekt **6066_h** definierte Zeit.

Wird der Wert auf "FFFFFFFF"_h gesetzt, wird die Überwachung abgeschaltet.

6068h Position Window Time

Funktion

Die Istposition muss sich für diese Zeit in Millisekunden innerhalb des "Position Window" (**6067_h**) befinden, damit die Zielposition als erreicht gilt in den Modi **Profile Position** und **Interpolated Position Mode**.

Objektbeschreibung

Index	6068 _h
Objektname	Position Window Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0064 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1504: Eintrag "Savable" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Ist die Abweichung der Istposition zur Zielposition kleiner als der Wert des Objekts **6067_h**, wird das Bit 10 im Objekt **6041_h** gesetzt. Die Bedingung muss länger erfüllt sein als die im Objekt **6066_h** definierte Zeit.

606B_h Velocity Demand Value

Funktion

Vorgabegeschwindigkeit in **benutzerdefinierten Einheiten** für den Regler im **Profile Velocity Mode**.

Objektbeschreibung

Index	606B _h
Objektname	Velocity Demand Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Dieses Objekt enthält die Ausgabe des Rampengenerators, die gleichzeitig der Vorgabewert für den Geschwindigkeitsregler ist.

606C_h Velocity Actual Value

Funktion

Aktuelle Istgeschwindigkeit in **benutzerdefinierten Einheiten**.

Objektbeschreibung

Index	606C _h
Objektname	Velocity Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

606Dh Velocity Window

Funktion

Gibt relativ zur Zielgeschwindigkeit einen symmetrischen Bereich an, innerhalb dessen das Ziel als erreicht gilt im Modus **Profile Velocity**.

Objektbeschreibung

Index	606D _h
Objektname	Velocity Window
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	001E _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Ist die Abweichung der Istgeschwindigkeit zur Sollgeschwindigkeit kleiner als der Wert dieses Objekts, wird das Bit 10 im Objekt **6041_h** gesetzt. Die Bedingung muss länger erfüllt sein als die im Objekt **6066_h** definierte Zeit (siehe auch **Statusword im Modus Profile Velocity**).

606Eh Velocity Window Time

Funktion

Die Istgeschwindigkeit muss sich für diese Zeit in Millisekunden innerhalb des "Velocity Window" (**606D_h**) befinden, damit das Ziel als erreicht gilt.

Objektbeschreibung

Index	606E _h
Objektname	Velocity Window Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Beschreibung

Ist die Abweichung der Istgeschwindigkeit zur Sollgeschwindigkeit kleiner als der Wert des Objekts **606D_h**, wird das Bit 10 im Objekt **6041_h** gesetzt. Die Bedingung muss länger erfüllt sein als die im Objekt **6066** definierte Zeit (siehe auch **Statusword im Modus Profile Velocity**).

6071h Target Torque

Funktion

Dieses Objekt enthält das Zieldrehmoment für den **Profile Torque** und **Cyclic Synchronous Torque** Modus in Promille des Nenndrehmoments.

Objektbeschreibung

Index	6071 _h
Objektname	Target Torque
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt **203B_h:01**.

Das Minimum von **6073_h** und **6072_h** wird als Limit für das Drehmoment in **6071_h** verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in **2031_h**) nicht übersteigen.

6072h Max Torque

Funktion

Das Objekt beschreibt das maximale Drehmoment für den **Profile Torque** und **Cyclic Synchronous Torque** Modus in Promille des Nenndrehmoments.

Objektbeschreibung

Index	6072 _h
Objektname	Max Torque
Object Code	VARIABLE

Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0064 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt **203B_h:01**.

Das Minimum von **6073_h** und **6072_h** wird als Limit für das Drehmoment in **6071_h** verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in **2031_h**) nicht übersteigen.

6073h Max Current

Funktion

Gibt den Maximalstrom in Promille des eingestellten Nennstroms an. Wird durch den maximalen Motorstrom (**2031_h**) begrenzt. Siehe auch **I2t Motor-Überlastungsschutz**.



Hinweis

Für Schrittmotoren wird nur der Nennstrom und kein Maximalstrom angegeben. Daher sollte der Wert von **6073_h** den Wert 1000 (100%) nicht überschreiten.

Objektbeschreibung

Index	6073 _h
Objektname	Max Current
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03E8 _h
Firmware Version	FIR-v1825-B577172
Änderungshistorie	

Beschreibung

Der Maximalstrom wird in Promille des Nennstroms wie folgt berechnet:

$$(6073_h * 203B_h:01) / 1000$$

Der Maximalstrom bestimmt:

- den Maximalstrom für den **I2t Motor-Überlastungsschutz**,
- den Sollstrom im *Open Loop*-Betrieb.

6074h Torque Demand

Funktion

Momentaner vom Rampengenerator geforderter Drehmomentsollwert in Promille des Nenndrehmoments für den internen Regler.

Objektbeschreibung

Index	6074 _h
Objektname	Torque Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt **203B_h:01**.

Das Minimum von **6073_h** und **6072_h** wird als Limit für das Drehmoment in **6071_h** verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in **2031_h**) nicht übersteigen.

6075h Motor Rated Current

Funktion

Enthält den in **203B_h:01_h** eingetragenen Nennstrom in mA.

6077h Torque Actual Value

Funktion

Dieses Objekt zeigt den aktuellen Drehmomentwert in Promille des Nenndrehmoments für den internen Regler.

Objektbeschreibung

Index	6077 _h
Objektname	Torque Actual Value

Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1540
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt **203B_h:01**.

Das Minimum von **6073_h** und **6072_h** wird als Limit für das Drehmoment in **6071_h** verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in **2031_h**) nicht übersteigen.

607Ah Target Position

Funktion

Dieses Objekt gibt die Zielposition in **benutzerdefinierten Einheiten** für den **Profile Position** und **Cyclic Synchronous Position** Modus an.

Objektbeschreibung

Index	607A _h
Objektnamen	Target Position
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000FA0 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

607Bh Position Range Limit

Funktion

Enthält die Minimal- und Maximalposition in **benutzerdefinierten Einheiten**.

Objektbeschreibung

Index	607B _h
Objektname	Position Range Limit
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Min Position Range Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Max Position Range Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Wird dieser Bereich über- oder unterschritten, erfolgt ein Überlauf. Um diesen Überlauf zu verhindern, können im Objekt **607D_h** ("Software Position Limit") Grenzwerte für die Zielposition eingestellt werden.

607Ch Home Offset

Funktion

Gibt die Differenz zwischen Null-Position der Steuerung und dem Referenzpunkt der Maschine in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	607C _h
Objektname	Home Offset
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

607Dh Software Position Limit

Funktion

Legt die Grenzpositionen relativ zum Referenzpunkt der Applikation in **benutzerdefinierten Einheiten** fest.

Objektbeschreibung

Index	607D _h
Objektname	Software Position Limit
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Min Position Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Max Position Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die absolute Zielposition muss innerhalb der hier gesetzten Grenzen liegen. Der Home Offset (**607C_h**) wird nicht berücksichtigt.

607E_h Polarity

Funktion

Mit diesem Objekt lässt sich die Drehrichtung umkehren.

Objektbeschreibung

Index	607E _h
Objektname	Polarity
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "RX-PDO".

Beschreibung

Generell gilt für die Richtungsumkehr: Ist ein Bit auf den Wert "1" gesetzt, ist die Umkehrung aktiviert. Ist der Wert "0", ist die Drehrichtung wie im jeweiligen Modus beschrieben.

7	6	5	4	3	2	1	0
POS	VEL						

VEL (Velocity)

Umkehr der Drehrichtung in folgenden Modi:

- **Profile Velocity Mode**
- **Cyclic Synchronous Velocity Mode**
- **Velocity Mode**

POS (Position)

Umkehr der Drehrichtung in folgenden Modi:

- **Profile Position Mode**
- **Cyclic Synchronous Position Mode**



Tipp

Sie können ein Invertieren des Drehfeldes erzwingen, dass alle Betriebsmodi betrifft. Siehe Objekt **3212_h:02_h**.

607F_h Max Profile Velocity

Funktion

Gibt die maximale Geschwindigkeit für den Modus **Profile Position**, **Interpolated Position Mode** und **Profile Velocity** in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	607F _h
Objektname	Max Profile Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00007530 _h
Firmware Version	FIR-v1540
Änderungshistorie	<p>Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Max profile velocity" auf "Max Profile Velocity".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "INTEGER16" auf "UNSIGNED32".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "nur lesen" auf "lesen/schreiben".</p>

Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "TX-PDO" auf "RX-PDO".

6080h Max Motor Speed

Funktion

Gibt die maximal zulässige Geschwindigkeit des Motors in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6080 _h
Objektname	Max Motor Speed
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00007530 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning". Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Maximum Speed" auf "Max Motor Speed". Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "RX-PDO". Firmware Version FIR-v1748-B538662: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Bewegung".

6081h Profile Velocity

Funktion

Gibt die maximale Fahrgeschwindigkeit in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6081 _h
Objektname	Profile Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h

Firmware Version	FIR-v1426
Änderungshistorie	

6082h End Velocity

Funktion

Gibt die Geschwindigkeit am Ende der gefahrenen Rampe in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6082 _h
Objektnamen	End Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6083h Profile Acceleration

Funktion

Gibt die maximale Beschleunigung in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6083 _h
Objektnamen	Profile Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6084h Profile Deceleration

Funktion

Gibt die maximale Verzögerung (Bremsrampe) in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6084 _h
Objektname	Profile Deceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6085h Quick Stop Deceleration

Funktion

Gibt die maximale Quick Stop-Verzögerung in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6085 _h
Objektname	Quick Stop Deceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6086h Motion Profile Type

Funktion

Gibt den Rampentyp für die Modi **Profile Position** und **Profile Velocity** an.

Objektbeschreibung

Index	6086 _h
Objektname	Motion Profile Type
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben

PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert = "0": = Trapez-Rampe

Wert = "3": Rampe mit begrenztem Ruck

6087h Torque Slope

Funktion

Dieses Objekt enthält die Steigung des Drehmoments im Torque Mode.

Objektbeschreibung

Index	6087 _h
Objektname	Torque Slope
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt **203B_h:01**.

Das Minimum von **6073_h** und **6072_h** wird als Limit für das Drehmoment in **6071_h** verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in **2031_h**) nicht übersteigen.

608Fh Position Encoder Resolution

Funktion

Enthält die physikalische Auflösung des Encoders/Sensors, der für die Positionsregelung verwendet wird.

Objektbeschreibung

Index	608F _h
Objektname	Position Encoder Resolution
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Tuning
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1738-B501312: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "nein" auf "RX-PDO".</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "RX-PDO".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p>

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Encoder Increments
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000007D0 _h

Subindex	02 _h
Name	Motor Revolutions
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	

Vorgabewert	00000001 _h
-------------	-----------------------

Beschreibung

Position Encoder Resolution = Encoder Increments (**608F_h:01_h**) / Motor Revolutions (**608F_h:02_h**)

6090h Velocity Encoder Resolution

Funktion

Enthält die physikalische Auflösung des Encoders/Sensors, der für die Drehzahlregelung verwendet wird.

Objektbeschreibung

Index	6090 _h
Objektnamen	Velocity Encoder Resolution
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	<p>Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p>

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Encoder Increments Per Second
Datentyp	INTEGER32
Zugriff	lesen/schreiben

PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	Motor Revolutions Per Second
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Velocity Encoder Resolution = Encoder Increments per second (6090_h:01_h) / Motor Revolutions per second (6090_h:02_h)

6091h Gear Ratio

Funktion

Anzahl der Motorumdrehungen pro Umdrehung der Abtriebsachse.

Objektbeschreibung

Index	6091 _h
Objektname	Gear Ratio
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "nein" auf "RX-PDO". Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "RX-PDO".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Motor Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	02 _h
Name	Shaft Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Gear Ratio = Motor Revolutions (**6091_h:01_h**) / Shaft Revolutions (**6091_h:02_h**)

6092h Feed Constant

Funktion

Vorschub im Falle eines Linearantriebs, in **benutzerdefinierten Einheiten** pro Umdrehungen der Abtriebsachse.

Objektbeschreibung

Index	6092 _h
Objektname	Feed Constant
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Feed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	02 _h
Name	Shaft Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Feed Constant = Feed (6092_h:01_h) / Shaft Revolutions (6092_h:02_h)

6096h Velocity Factor

Funktion

Dieses Objekt beinhaltet den Faktor, der zum Umrechnen von benutzerdefinierten Geschwindigkeitseinheiten verwendet wird. Siehe Kapitel **Benutzerdefinierte Einheiten**.

Objektbeschreibung

Index	6096 _h
Objektname	Velocity Factor
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
<hr/>	
Subindex	01 _h
Name	Numerator
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	02 _h
Name	Divisor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Zähler des Faktors
- 02_h: Nenner des Faktors

6097h Acceleration Factor

Funktion

Dieses Objekt beinhaltet den Faktor, der zum Umrechnen von benutzerdefinierten Beschleunigungseinheiten verwendet wird. Siehe Kapitel **Benutzerdefinierte Einheiten**.

Objektbeschreibung

Index	6097 _h
Objektnamen	Acceleration Factor
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
<hr/>	
Subindex	01 _h
Name	Numerator
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	02 _h
Name	Divisor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Zähler des Faktors
- 02_h: Nenner des Faktors

6098h Homing Method

Funktion

Dieses Objekt definiert die **Referenzfahrt-Methode** im **Homing** Mode.

Objektbeschreibung

Index	6098 _h
Objektname	Homing Method
Object Code	VARIABLE
Datentyp	INTEGER8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	23 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6099h Homing Speed

Funktion

Gibt die Geschwindigkeiten für den Homing Mode (6098_h) in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	6099 _h
Objektname	Homing Speed
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Speed During Search For Switch
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000032 _h

Subindex	02 _h
Name	Speed During Search For Zero
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	

Vorgabewert 0000000A_h

Beschreibung

In Subindex 1 wird die Geschwindigkeit für die Suche nach dem Schalter angegeben.

In Subindex 2 wird die (niedrigere) Geschwindigkeit für die Suche nach der Referenzposition angegeben.



Hinweis

- Die Geschwindigkeit in Subindex 2 ist gleichzeitig die Anfangsgeschwindigkeit beim Start der Beschleunigungsrampe. Wird diese zu hoch eingestellt, verliert der Motor Schritte bzw. dreht sich überhaupt nicht. Eine zu hohe Einstellung führt dazu, dass die Indexmarkierung übersehen wird. Die Geschwindigkeit in Subindex 2 soll daher unter 1000 Schritten pro Sekunde sein.
 - Die Geschwindigkeit in Subindex 1 muss größer sein als die Geschwindigkeit in Subindex 2.
-

609Ah Homing Acceleration

Funktion

Gibt die Beschleunigungsrampe für den Homing Mode in **benutzerdefinierten Einheiten** an.

Objektbeschreibung

Index	609A _h
Objektname	Homing Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Die Rampe wird nur beim Losfahren verwendet. Beim Erreichen des Schalters wird sofort auf die niedrigere Geschwindigkeit umgeschaltet und beim Erreichen der Endposition wird sofort gestoppt.

60A2h Jerk Factor

Funktion

Dieses Objekt beinhaltet den Faktor, der zum Umrechnen von benutzerdefinierten Ruckeinheiten verwendet wird. Siehe Kapitel **Benutzerdefinierte Einheiten**.

Objektbeschreibung

Index	60A2 _h
Objektname	Jerk Factor
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Numerator
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	02 _h
Name	Divisor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Zähler des Faktors
- 02_h: Nenner des Faktors

60A4h Profile Jerk

Funktion

Im Falle einer Rampe mit begrenztem Ruck kann in diesem Objekt die Größe des Rucks eingetragen werden. Ein Eintrag mit dem Wert "0" bedeutet, dass der Ruck nicht begrenzt ist.

Objektbeschreibung

Index	60A4 _h
Objektname	Profile Jerk
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Name" geändert von "End Acceleration Jerk" auf "Begin Deceleration Jerk". Firmware Version FIR-v1614: Eintrag "Name" geändert von "Begin Deceleration Jerk" auf "End Acceleration Jerk".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h
Subindex	01 _h
Name	Begin Acceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	02 _h
Name	Begin Deceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h

Subindex	03 _h
Name	End Acceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h

Subindex	04 _h
Name	End Deceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h

Beschreibung

- Subindex 01_h (*Begin Acceleration Jerk*): Anfangsruck bei Beschleunigung
- Subindex 02_h (*Begin Deceleration Jerk*): Anfangsruck bei Bremsung
- Subindex 03_h (*End Acceleration Jerk*): Abschlussruck bei Beschleunigung
- Subindex 04_h (*End Deceleration Jerk*): Abschlussruck bei Bremsung

60A8h SI Unit Position

Funktion

Dieses Objekt beinhaltet die Positionseinheit. Siehe Kapitel **Benutzerdefinierte Einheiten**.

Objektbeschreibung

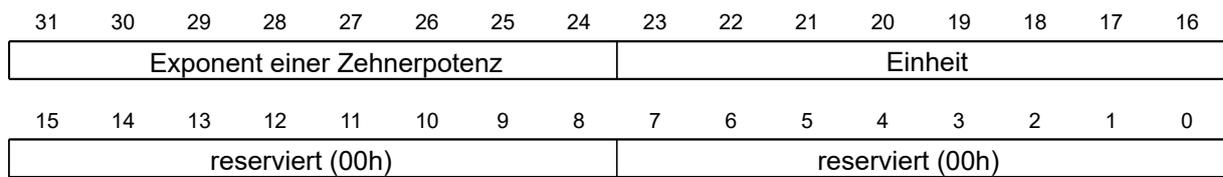
Index	60A8 _h
Objektname	SI Unit Position
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF410000 _h
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Beschreibung

Das Objekt 60A8_h enthält :

- Bits 16 bis 23: die Positionseinheit (siehe Kapitel **Einheiten**)

- Bits 24 bis 31: den Exponenten einer Zehnerpotenz (siehe Kapitel **Einheiten**)



60A9h SI Unit Velocity

Funktion

Dieses Objekt beinhaltet die Geschwindigkeitseinheit. Siehe Kapitel **Benutzerdefinierte Einheiten**.

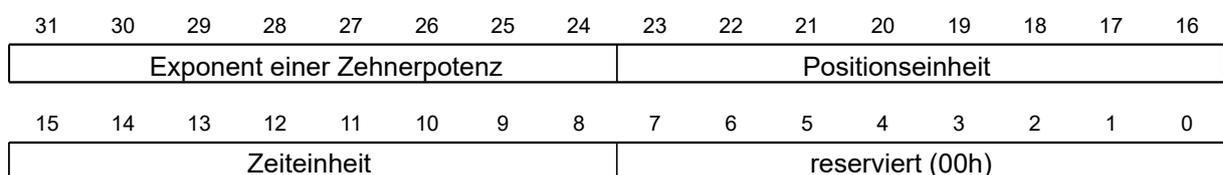
Objektbeschreibung

Index	60A9 _h
Objektname	SI Unit Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00B44700 _h
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Beschreibung

Das Objekt 60A9_h enthält :

- Bits 8 bis 15: die Zeiteinheit (siehe Kapitel **Einheiten**)
- Bits 16 bis 23: die Positionseinheit (siehe Kapitel **Einheiten**)
- Bits 24 bis 31: den Exponenten einer Zehnerpotenz (siehe Kapitel **Einheiten**)



60B0h Position Offset

Funktion

Offset für den Positionssollwert in **benutzerdefinierten Einheiten**. Wird in den Modi **Cyclic Synchronous Position**, und **Takt-Richtungs-Modus** berücksichtigt.

Objektbeschreibung

Index	60B0 _h
-------	-------------------

Objektname	Position Offset
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1738-B505321
Änderungshistorie	

60B1h Velocity Offset

Funktion

Offset für den Geschwindigkeitssollwert in **benutzerdefinierten Einheiten**. Wird in den Modi **Cyclic Synchronous Position**, **Cyclic Synchronous Velocity** und **Takt-Richtungs-Modus** berücksichtigt.

Objektbeschreibung

Index	60B1 _h
Objektname	Velocity Offset
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1738-B505321
Änderungshistorie	

60B2h Torque Offset

Funktion

Offset für den Drehmomentsollwert in Promille. Wird in den Modi **Cyclic Synchronous Position**, **Cyclic Synchronous Velocity**, **Cyclic Synchronous Torque** und **Takt-Richtungs-Modus** berücksichtigt.

Objektbeschreibung

Index	60B2 _h
Objektname	Torque Offset
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben

PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1738-B505321
Änderungshistorie	

60C1h Interpolation Data Record

Funktion

Dieses Objekt enthält die Sollposition in **benutzerdefinierten Einheiten** für den Interpolationsalgorithmus für den Betriebsmodus **Interpolated Position**.

Objektbeschreibung

Index	60C1 _h
Objektname	Interpolation Data Record
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1512
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	1st Set-point
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Der Wert wird zum nächsten Synchronisationszeitpunkt übernommen.

60C2h Interpolation Time Period

Funktion

Dieses Objekt enthält die Interpolationszeit.

Objektbeschreibung

Index	60C2 _h
Objektname	Interpolation Time Period
Object Code	RECORD
Datentyp	INTERPOLATION_TIME_PERIOD
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Interpolation Time Period Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	02 _h
Name	Interpolation Time Index
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	FD _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Interpolationszeit.
- 02_h: Zehnerexponent der Interpolationszeit: muss den Wert -3 halten (entspricht der Zeitbasis in Millisekunden).

Es gilt dabei: Zykluszeit = Wert des **60C2_h:01_h** * 10^{Wert des 60C2:02} Sekunden.

60C4h Interpolation Data Configuration

Funktion

Dieses Objekt bietet die maximale Puffergröße, gibt die konfigurierte Puffer-Organisation der interpolierten Daten an und bietet Objekte zur Definition der Größe des Datensatzes und zum Löschen des Puffers. Es wird zudem verwendet, um die Position weiterer Datenpunkte zu speichern.

Objektbeschreibung

Index	60C4 _h
Objektname	Interpolation Data Configuration
Object Code	RECORD
Datentyp	INTERPOLATION_DATA_CONFIGURATION
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1512
Änderungshistorie	<p>Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 05 geändert von "lesen/schreiben" auf "nur schreiben".</p> <p>Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 06 geändert von "lesen/schreiben" auf "nur schreiben".</p> <p>Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p>

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	06 _h
<hr/>	
Subindex	01 _h
Name	MaximumBufferSize
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	02 _h
Name	ActualBufferSize
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	03 _h
Name	BufferOrganization
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	04 _h
Name	BufferPosition
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
<hr/>	
Subindex	05 _h
Name	SizeOfDataRecord
Datentyp	UNSIGNED8
Zugriff	nur schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h

Subindex	06 _h
Name	BufferClear
Datentyp	UNSIGNED8
Zugriff	nur schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Der Wert des Subindex 01_h enthält die maximale mögliche Anzahl der interpolierten Datensätze.

Der Wert des Subindex 02_h enthält die momentane Anzahl der interpolierten Datensätze.

Wenn Subindex 03_h "00_h" ist, bedeutet das eine FIFO-Puffer-Organisation, wenn es "01_h" ist, gibt es eine Ring-Puffer-Organisation an.

Der Wert des Subindex 04_h ist ohne Einheit und gibt den nächsten freien Puffer-Einstiegspunkt an.

Der Wert des Subindex 05_h wird in der Einheit "Byte" angegeben. Wenn der Wert "00_h" in den Subindex 06_h geschrieben wird, löscht es die eingegangenen Daten im Puffer, deaktiviert den Zugriff und löscht alle Interpolierten Datensätze. Wenn der Wert "01_h" in den Subindex 06_h geschrieben wird, aktiviert es den Zugriff auf den Eingangs-Puffer.

60C5h Max Acceleration

Funktion

Dieses Objekt enthält die maximal zulässige Beschleunigung für den Modus **Profile Position** und **Profile Velocity**.

Objektbeschreibung

Index	60C5 _h
Objektnamen	Max Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 _h
Firmware Version	FIR-v1426
Änderungshistorie	

60C6h Max Deceleration

Funktion

Dieses Objekt enthält die maximal zulässige Verzögerung (Bremsrampe) für den Modus **Profile Position** und **Profile Velocity**.

Objektbeschreibung

Index	60C6 _h
Objektname	Max Deceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 _h
Firmware Version	FIR-v1426
Änderungshistorie	

60E4h Additional Position Actual Value

Funktion

Enthält die aktuelle Istposition aller vorhandenen Rückführungen in **benutzerdefinierten Einheiten**.

Objektbeschreibung

Index	60E4 _h
Objektname	Additional Position Actual Value
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32". Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h - 03 _h
Name	Additional Position Actual Value #1 - #3
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:
 Subindex n enthält die aktuelle Istposition der entsprechenden Rückführung.
 Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

60E5h Additional Velocity Actual Value

Funktion

Enthält die aktuelle Istgeschwindigkeit aller vorhandenen Rückführungen in **benutzerdefinierten Einheiten**.

Objektbeschreibung

Index	60E5 _h
Objektname	Additional Velocity Actual Value
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32". Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	03 _h
<hr/>	
Subindex	01 _h - 03 _h
Name	Additional Velocity Actual Value #1 - #3
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:
Subindex n enthält die aktuelle Istgeschwindigkeit der entsprechenden Rückführung.
Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

60E6h Additional Position Encoder Resolution - Encoder Increments

Funktion

Mit diesem Objekt und mit **60EB_h** wird die Auflösung jeder vorhandenen Rückführung berechnet.

Objektbeschreibung

Index	60E6 _h
Objektname	Additional Position Encoder Resolution - Encoder Increments
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662
Änderungshistorie	Firmware Version FIR-v1748-B538662: Die Anzahl der Einträge haben sich geändert von 5 auf 4.

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	03 _h
<hr/>	
Subindex	01 _h
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00320000 _h
<hr/>	
Subindex	02 _h
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000000C8 _h
<hr/>	
Subindex	03 _h
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #3
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:
 Subindex n enthält die Anzahl der Inkremente der entsprechenden Rückführung.
 Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Auflösung der Rückführung "n" berechnet sich wie folgt:

$$\text{Position Encoder Resolution} = \text{Encoder Increments (60E6}_h\text{:01}_h\text{)} / \text{Motor Revolutions (60EB}_h\text{:02}_h\text{)}$$

60E8h Additional Gear Ratio - Motor Shaft Revolutions

Funktion

In diesem Objekt und in 60ED_h können Sie die Getriebeübersetzung jeder vorhandenen Rückführung einstellen.

Objektbeschreibung

Index	60E8 _h
Objektname	Additional Gear Ratio - Motor Shaft Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h - 03 _h
Name	Additional Gear Ratio - Motor Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h: Subindex "n" enthält die Anzahl der Motorumdrehungen für die entsprechende Rückführung. Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Getriebeübersetzung der Rückführung "n" berechnet sich wie folgt:

$$\text{Gear Ratio} = \text{Motor Shaft Revolutions (60E8}_h:n_h) / \text{Driving Shaft Revolutions (60ED}_h:n_h)$$

60E9h Additional Feed Constant - Feed

Funktion

In diesem Objekt und in **60EE_h** können Sie eine Vorschubkonstante für jede vorhandene Rückführung einstellen.

Objektbeschreibung

Index	60E9 _h
Objektname	Additional Feed Constant - Feed
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h - 03 _h
Name	Additional Feed Constant - Feed Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h: Subindex "n" enthält den Vorschub in **benutzerdefinierten Einheiten** für die entsprechende Rückführung.
Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Vorschubkonstante der Rückführung "n" berechnet sich wie folgt:

Feed Constant = Feed (60E9_h:n_h) / Driving Shaft Revolutions (60EE_h:n_h)

60EB_h Additional Position Encoder Resolution - Motor Revolutions

Funktion

Mit diesem Objekt und mit **60E6_h** wird die Auflösung jeder vorhandenen Rückführung berechnet.

Objektbeschreibung

Index	60EB _h
Objektname	Additional Position Encoder Resolution - Motor Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h - 03 _h
Name	Additional Position Encoder Resolution - Motor Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h: Subindex n enthält die Anzahl der Motorumdrehungen der entsprechenden Rückführung.

Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Auflösung der Rückführung "n" berechnet sich wie folgt:

Position Encoder Resolution = Encoder Increments (60E6_h:01_h) / Motor Revolutions (60EB_h:02_h)

60ED_h Additional Gear Ratio - Driving Shaft Revolutions

Funktion

In diesem Objekt und in 60E8_h können Sie die Getriebeübersetzung jeder vorhandenen Rückführung einstellen.

Objektbeschreibung

Index	60ED _h
Objektname	Additional Gear Ratio - Driving Shaft Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h - 03 _h
Name	Additional Gear Ratio - Driving Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h: Subindex "n" enthält die Anzahl der Umdrehungen der Abtriebsachse für die entsprechende Rückführung.
 Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Getriebeübersetzung der Rückführung "n" berechnet sich wie folgt:

$$\text{Gear Ratio} = \text{Motor Shaft Revolutions (60E8}_h:n_h) / \text{Driving Shaft Revolutions (60ED}_h:n_h)$$

60EEh Additional Feed Constant - Driving Shaft Revolutions

Funktion

In diesem Objekt und in 60E9_h können Sie eine Vorschubkonstante für jede vorhandene Rückführung einstellen.

Objektbeschreibung

Index	60EE _h
Objektname	Additional Feed Constant - Driving Shaft Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 _h

Subindex	01 _h - 03 _h
Name	Additional Feed Constant - Driving Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_{h_i} : Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_{h_i} : Subindex "n" enthält die Anzahl der Umdrehungen der Abtriebsachse für die entsprechende Rückführung.
Subindex 01_{h_i} entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Vorschubkonstante der Rückführung "n" berechnet sich wie folgt:

$$\text{Feed Constant} = \text{Feed} (60E9_{h_i:n_{h_i}}) / \text{Driving Shaft Revolutions} (60EE_{h_i:n_{h_i}})$$

60F2h Positioning Option Code

Funktion

Das Objekt beschreibt das Positionierverhalten im **Profile Position** Modus.

Objektbeschreibung

Index	60F2 _h
Objektname	Positioning Option Code
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	FIR-v1446
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Derzeit werden nur nachfolgende Bits unterstützt:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS	RESERVED [3]			IP OPTION [4]			RADO [2]		RRO [2]		CIO [2]		REL. OPT. [2]		

REL. OPT. (Relative Option)

Diese Bits bestimmen das Verhalten bei relativer Drehbewegung im "Profile Position" Modus, sollte Bit 6 des Kontrollwortes **6040_h** = "1" gesetzt sein.

Bit 1	Bit 0	Definition
0	0	Positionsbewegungen werden relativ zu der vorherigen (intern absoluten) Zielposition ausgeführt (jeweils relativ zu 0 falls keine Zielposition voran gegangen ist)
0	1	Positionsbewegungen werden relativ zum Vorgabewert (bzw. Ausgang) des Rampengenerators ausgeführt.
1	0	Positionsbewegungen werden relativ zur Istposition (Objekt 6064_h) ausgeführt.

Bit 1	Bit 0	Definition
1	1	Reserviert

RRO (Request-Response Option)

Diese Bits bestimmen das Verhalten bei der Übergabe des Controlwords **6040_h** Bit 5 ("new setpoint") - die Steuerung übernimmt in diesem Fall die Freigabe des Bits selbständig. Damit fällt die Notwendigkeit weg, das Bit anschließend extern wieder auf "0" zu setzen. Nachdem das Bit von der Steuerung aus auf den Wert "0" gesetzt wurde, wird auch das Bit 12 ("setpoint acknowledgement") im Statusword **6041_h** auf den Wert "0" gesetzt.



Hinweis

Diese Optionen bringen die Steuerung dazu, das Objekt Controlword **6040_h** zu modifizieren.

Bit 5	Bit 4	Definition
0	0	Die Funktionalität ist wie unter Setzen von Fahrbefehlen beschrieben.
0	1	Die Steuerung wird das Bit "new setpoint" frei geben, sobald die momentane Zielfahrt ihr Ziel erreicht hat.
1	0	Die Steuerung wird das Bit "new setpoint" frei geben, sobald es der Steuerung möglich ist.
1	1	Reserviert

RADO (Rotary Axis Direction Option)

Diese Bits bestimmen die Drehrichtung im "Profile Position" Modus.

Bit 7	Bit 6	Definition
0	0	Normale Positionierung ähnlich einer linearen Achse: Falls eines der "Position Range Limits" 607B_h:01_h und 02_h erreicht oder überschritten wird, wird der Vorgabewert automatisch an das andere Ende der Limits übertragen. Nur mit dieser Bitkombination ist eine Bewegung größer als der Modulo-Wert möglich.
0	1	Positionierung nur in negativer Richtung: falls die Zielposition größer als die aktuelle Position ist fährt die Achse über das "Min Position Range Limit" aus Objekt 607D_h:01_h zu der Zielposition.
1	0	Positionierung nur in positiver Richtung: falls die Zielposition kleiner als die aktuelle Position ist fährt die Achse über das "Max Position Range Limit" aus Objekt 607D_h:01_h zu der Zielposition.
1	1	Positionierung mit dem kürzesten Weg zur Zielposition. Falls die Differenz zwischen aktueller Position und Zielposition in einem 360° System kleiner als 180° ist, fährt die Achse in positiver Richtung.

60F4h Following Error Actual Value

Funktion

Dieses Objekt enthält den aktuellen Schleppfehler in **benutzerdefinierten Einheiten**.

Objektbeschreibung

Index	60F4 _h
Objektname	Following Error Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

60F8h Max Slippage

Funktion

Definiert den maximal erlaubten Schlupffehler in **benutzerdefinierten Einheiten** symmetrisch zur **Sollgeschwindigkeit** im Modus **Profile Velocity**.

Objektbeschreibung

Index	60F8 _h
Objektname	Max Slippage
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000190 _h
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Beschreibung

Weicht die Istgeschwindigkeit von der Sollgeschwindigkeit so stark ab, dass der Wert (Absolutbetrag) dieses Objekts überschritten wird, wird das Bit 13 im Objekt **6041_h** gesetzt. Die Abweichung muss länger andauern als die Zeit im Objekt **203F_h**.

Wird der Wert des 60F8_h auf "7FFFFFFF"_h gesetzt, wird die Schlupffehler-Überwachung abgeschaltet.

Im Objekt **3700_h** kann eine Reaktion auf den Schlupffehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt **1003_h** eingetragen.

60FAh Control Effort

Funktion

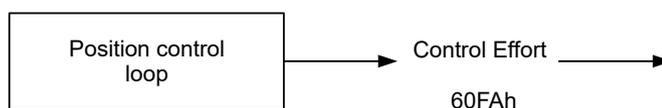
Dieses Objekt beinhaltet die Korrekturgeschwindigkeit in **benutzerdefinierten Einheiten**, die vom Positionsregler dem Geschwindigkeitsregler zugeführt wird.

Objektbeschreibung

Index	60FA _h
Objektnamen	Control Effort
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1748-B531667
Änderungshistorie	

Beschreibung

Der Positionsregler bildet aus der Differenz zwischen Ist- und Sollposition eine Korrekturgeschwindigkeit (in **benutzerdefinierten Einheiten**), die an den Geschwindigkeitsregler weitergeleitet wird. Dieser Korrekturwert hängt vom Proportionalanteil (**3210_h:01_h**) und Integralanteil (**3210_h:02_h**) des Positionsreglers ab. Siehe auch Kapitel **Closed Loop**.



60FCh Position Demand Internal Value

Funktion

Gibt die aktuelle Sollposition in Inkrementen an.

Objektbeschreibung

Index	60FC _h
Objektnamen	Position Demand Internal Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen

PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

60FDh Digital Inputs

Funktion

Mit diesem Objekt können die **Digitalen Eingänge** des Motors gelesen werden.

Objektbeschreibung

Index	60FD _h
Objektname	Digital Inputs
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								IN 8	IN 7	IN 6	IN 5	IN 4	IN 3	IN 2	IN 1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													HS	PLS	NLS

NLS (Negative Limit Switch)

negativer Endschalter

PLS (Positive Limit Switch)

positiver Endschalter

HS (Home Switch)

Referenzschalter

IN n (Input n)

Eingang n - die Anzahl der verwendeten Bits ist abhängig von der jeweiligen Steuerung.

60FEh Digital Outputs

Funktion

Mit diesem Objekt können die **Digitalausgänge** des Motors geschrieben werden.

Objektbeschreibung

Index	60FE _h
Objektname	Digital Outputs
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	Digital Outputs #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Zum Schreiben der Ausgänge müssen noch die Einträge in Objekt **3250_h**, Subindex 02_h bis 05_h berücksichtigt werden.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												OUT4	OUT3	OUT2	OUT1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															BRK

BRK (Brake)

Bit für den Bremsenausgang (falls der Controller diese Funktion unterstützt).

OUT n (Output No n)

Bit für den jeweiligen digitalen Ausgang, die genaue Zahl der Digitalausgänge ist abhängig von der Steuerung.

60FFh Target Velocity

Funktion

In dieses Objekt wird die Zielgeschwindigkeit für den **Profile Velocity** und **Cyclic Synchronous VelocityMode** in **benutzerdefinierten Einheiten** eingetragen.

Objektbeschreibung

Index	60FF _h
Objektname	Target Velocity
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

6502h Supported Drive Modes

Funktion

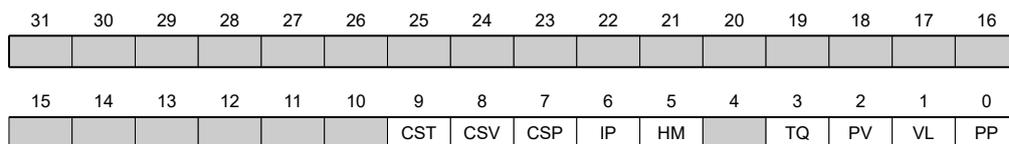
Das Objekt beschreibt die unterstützten Betriebsmodi im Objekt **6060_h**.

Objektbeschreibung

Index	6502 _h
Objektname	Supported Drive Modes
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	000003EF _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Ein gesetztes Bit gibt an, ob der jeweilige Modus unterstützt wird. Ist der Wert des Bits "0", wird der Modus nicht unterstützt.



PP

Profile Position Modus

VL

Velocity Modus

PV

Profile Velocity Modus

TQ

Torque Modus

HM

Homing Modus

IP

Interpolated Position Modus

CSP

Cyclic Synchronous Position Modus

CSV

Cyclic Synchronous Velocity Modus

CST

Cyclic Synchronous Torque Modus

6503h Drive Catalogue Number

Funktion

Enthält den Gerätenamen als Zeichenkette.

Objektbeschreibung

Index	6503 _h
Objektname	Drive Catalogue Number
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1426
Änderungshistorie	

6505h Http Drive Catalogue Address

Funktion

Dieses Objekt enthält die Web-Adresse des Herstellers als Zeichenkette.

Objektbeschreibung

Index	6505 _h
Objektname	Http Drive Catalogue Address
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	http://www.nanotec.de
Firmware Version	FIR-v1426
Änderungshistorie	

12 Copyrights

12.1 Einführung

In der Nanotec Software sind Komponenten aus Produkten externer Software-Hersteller integriert. In diesem Kapitel finden Sie die Copyright-Informationen zu den verwendeten externen Software-Quellen.

12.2 AES

FIPS-197 compliant AES implementation

Based on XySSL: Copyright (C) 2006-2008 Christophe Devine

Copyright (C) 2009 Paul Bakker <polarssl_maintainer at polarssl dot org>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution; or, the application vendor's website must provide a copy of this notice.
- Neither the names of PolarSSL or XySSL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The AES block cipher was designed by Vincent Rijmen and Joan Daemen.

<http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

12.3 MD5

MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

12.4 uIP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12.5 DHCP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12.6 CMSIS DSP Software Library

Copyright (C) 2010 ARM Limited. All rights reserved.

12.7 FatFs

FatFs - FAT file system module include file R0.08 (C)ChaN, 2010

FatFs module is a generic FAT file system module for small embedded systems.

This is a free software that opened for education, research and commercial developments under license policy of following terms.

Copyright (C) 2010, ChaN, all right reserved.

The FatFs module is a free software and there is NO WARRANTY.

No restriction on use. You can use, modify and redistribute it for personal, non-profit or commercial product UNDER YOUR RESPONSIBILITY.

Redistributions of source code must retain the above copyright notice.

12.8 Protothreads

Protothread class and macros for lightweight, stackless threads in C++.

This was "ported" to C++ from Adam Dunkels' protothreads C library at: <http://www.sics.se/~adam/pt/>

Originally ported for use by Hamilton Jet (www.hamiltonjet.co.nz) by Ben Hoyt, but stripped down for public release. See his blog entry about it for more information: <http://blog.micropledge.com/2008/07/protothreads/>

Original BSD-style license

Copyright (c) 2004-2005, Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the Institute and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Institute or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

12.9 lwIP

Copyright (c) 2001-2004 Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This file is part of the lwIP TCP/IP stack.

Author: Adam Dunkels <adam@sics.se>