# Technisches Handbuch N6

## Feldbus: CANopen

## Zu nutzen mit folgenden Varianten:

N6-1-2-1, N6-1-2-1-S, N6-2-2-1, N6-2-2-1-S

Gültig ab Firmware-Version FIR-v2508
und ab Hardware-Version 1

Technisches Handbuch Version: 1.1.0

# Inhalt


1 Einleitung ..... 10
1.1 Produktdokumentation ..... 10
1.2 Versionshinweise ..... 10
1.3 Urheberrecht, Kennzeichnung und Kontakt ..... 10
1.4 Bestimmungsgemäße Verwendung ..... 11
1.5 Zielgruppe und Qualifikation ..... 11
1.6 Gewährleistung und Haftungsausschluss ..... 11
1.7 Mitgeltende Vorschriften ..... 11
1.8 EU-Richtlinien zur Produktsicherheit ..... 11
1.9 Verwendete Symbole ..... 12
1.10 Hervorhebungen im Text ..... 12
1.11 Zahlenwerte ..... 13
1.12 Bits ..... 13
1.13 Zählrichtung (Pfeile) ..... 13

2 Sicherheits- und Warnhinweise ..... 14

3 Technische Daten und Anschlussbelegung ..... 16
3.1 Umgebungsbedingungen ..... 16
3.2 Maßzeichnungen und Montagemöglichkeiten ..... 17
3.3 Elektrische Eigenschaften und technische Daten ..... 17
3.4 Übertemperaturschutz ..... 18
3.5 LED-Signalisierung ..... 18
3.5.1 Betriebs-LED ..... 18
3.6 Anschlussbelegung ..... 19
3.6.1 X1 – Motoranschluss ..... 19
3.6.2 X2 – Spannungsversorgung ..... 20
3.6.3 X3 - STO (*Safe Torque Off*) ..... 22
3.6.4 X4 – Ein- und Ausgänge ..... 23
3.6.5 X5 – SSI-Encoder ..... 24
3.6.6 X6 – Inkrmentalencoder/Hall-Sensoren ..... 25
3.6.7 X7 – CANopen IN/OUT ..... 26
3.6.8 S1 – 120 Ohm Terminierungswiderstand ..... 26
3.6.9 S2 – CANopen Node-ID und Baudrate ..... 27
3.6.10 S3 – CANopen Node-ID und Baudrate ..... 27
3.6.11 Micro-USB ..... 27

4 Inbetriebnahme ..... 29
4.1 Konfiguration über USB ..... 30
4.1.1 Allgemeines ..... 30
4.1.2 USB-Anschluss ..... 30
4.1.3 Konfigurationsdatei ..... 31
4.1.4 NanoJ-Programm ..... 34
4.2 Konfiguration über CANopen ..... 35
4.2.1 Kommunikationseinstellungen ..... 35
4.2.2 Kommunikation aufbauen ..... 36
4.3 Motordaten einstellen ..... 37
4.4 Motor anschließen ..... 38
4.5 Auto-Setup ..... 38

4.5.1 Parameter-Ermittlung....39
4.5.2 Durchführung....39
4.5.3 Parameterspeicherung....41
4.6 Konfigurieren der Sensoren....41
4.7 Testlauf....43

**5 Generelle Konzepte....44**
5.1 Betriebsarten....44
5.1.1 Allgemein....44
5.1.2 Open Loop....45
5.1.3 Closed Loop....47
5.2 CiA 402 Power State Machine....53
5.2.1 Zustandsmaschine....53
5.2.2 Verhalten beim Verlassen des Zustands *Operation enabled*....55
5.3 Benutzerdefinierte Einheiten....58
5.3.1 Einheiten....59
5.3.2 Encoderauflösung....60
5.3.3 Getriebeübersetzung....61
5.3.4 Vorschubkonstante....61
5.3.5 Berechnungsformeln für Benutzereinheiten....61
5.4 Begrenzung des Bewegungsbereichs....63
5.4.1 Verhalten beim Erreichen der Endschalter....63
5.4.2 Software-Endschalter....64
5.5 Zykluszeiten....64

**6 Betriebsmodi....65**
6.1 Profile Position....65
6.1.1 Übersicht....65
6.1.2 Setzen von Fahrbefehlen....66
6.1.3 Genauigkeitsverlust bei Relativbewegungen....70
6.1.4 Randbedingungen für eine Positionierfahrt....71
6.1.5 Ruck-begrenzter und nicht ruck-begrenzter Modus....72
6.2 Profile Velocity....73
6.2.1 Beschreibung....73
6.2.2 Aktivierung....73
6.2.3 Controlword....73
6.2.4 Statusword....73
6.2.5 Objekteinträge....73
6.3 Profile Torque....75
6.3.1 Beschreibung....75
6.3.2 Aktivierung....76
6.3.3 Controlword....76
6.3.4 Statusword....76
6.3.5 Objekteinträge....76
6.4 Homing....77
6.4.1 Übersicht....77
6.4.2 Referenzfahrt-Methode....79
6.5 Interpolated Position Mode....85
6.5.1 Übersicht....85
6.5.2 Aktivierung....85
6.5.3 Controlword....85
6.5.4 Statusword....85
6.5.5 Benutzung....85
6.5.6 Setup....86
6.5.7 Operation....86
6.6 Cyclic Synchronous Position....86
6.6.1 Übersicht....86

6.6.2 Objekteinträge....87
6.7 Cyclic Synchronous Velocity....88
6.7.1 Übersicht....88
6.7.2 Objekteinträge....88
6.8 Cyclic Synchronous Torque....89
6.8.1 Übersicht....89
6.8.2 Objekteinträge....90
6.9 Takt-Richtungs-Modus....90
6.9.1 Beschreibung....90
6.9.2 Aktivierung....90
6.9.3 Generelles....90
6.9.4 Statusword....91
6.9.5 Unterarten des Takt-Richtungs-Modus....91
6.10 Auto-Setup....92
6.10.1 Beschreibung....92
6.10.2 Aktivierung....92
6.10.3 Controlword....92
6.10.4 Statusword....92

**7 Spezielle Funktionen....93**
7.1 Digitale Ein- und Ausgänge....93
7.1.1 Digitale Eingänge....93
7.1.2 Digitale Ausgänge....97
7.1.3 Pulsgenerator....99
7.2 Analoge Eingänge....100
7.2.1 Objekteinträge....100
7.2.2 Analogwert skalieren....100
7.3 Automatische Bremsensteuerung....101
7.3.1 Beschreibung....101
7.3.2 Aktivierung und Anschluss....101
7.3.3 Steuerung der Bremse....101
7.3.4 Bremsen-PWM....102
7.4 Externe Ballast-Schaltung....103
7.4.1 Steuerung des Ballast-Widerstands....103
7.4.2 Ballast aktivieren....103
7.4.3 Ballast-Überwachung....104
7.5 $I^2t$ Motor-Überlastungsschutz....105
7.5.1 Beschreibung....105
7.5.2 Objekteinträge....105
7.5.3 Aktivierung....105
7.5.4 Funktion von $I^2t$....105
7.6 Objekte speichern....106
7.6.1 Allgemeines....106
7.6.2 Kategorie: Kommunikation....107
7.6.3 Kategorie: Applikation....108
7.6.4 Kategorie: Benutzer....109
7.6.5 Kategorie: Bewegung....109
7.6.6 Kategorie: Motor....110
7.6.7 Kategorie: Tuning....110
7.6.8 Kategorie: Sensor....110
7.6.9 Kategorie: CANopen....110
7.6.10 Kategorie: Modbus RTU....110
7.6.11 Speichervorgang starten....110
7.6.12 Speicherung verwerfen....111
7.6.13 Konfiguration verifizieren....112
7.7 Generic SPI....112

8 CANopen ..... 114
8.1 Allgemeines ..... 114
8.1.1 CAN-Nachricht ..... 114
8.2 CANopen Dienste ..... 114
8.2.1 Network Management (NMT) ..... 115
8.2.2 Synchronisations-Objekt (SYNC) ..... 116
8.2.3 Emergency Object (EMCY) ..... 117
8.2.4 Service Data Object (SDO) ..... 118
8.2.5 Process Data Object (PDO) ..... 125
8.2.6 Boot-Up Protocol ..... 130
8.2.7 Heartbeat und Nodeguarding ..... 131
8.3 LSS-Protokoll ..... 133
8.3.1 Allgemeines ..... 133
8.3.2 LSS-Nachricht ..... 133
8.3.3 LSS-Dienste ..... 133
8.3.4 Beispiel ..... 144

9 Programmierung mit NanoJ ..... 146
9.1 NanoJ-Programm ..... 146
9.2 Mapping im NanoJ-Programm ..... 149
9.3 NanoJ-Funktionen im NanoJ-Programm ..... 151
9.4 Einschränkungen und mögliche Probleme ..... 152

10 Objektverzeichnis Beschreibung ..... 154
10.1 Übersicht ..... 154
10.2 Aufbau der Objektbeschreibung ..... 154
10.3 Objektbeschreibung ..... 154
10.4 Wertebeschreibung ..... 155
10.5 Beschreibung ..... 156
1000h Device Type ..... 157
1001h Error Register ..... 158
1003h Pre-defined Error Field ..... 159
1005h COB-ID Sync ..... 163
1006h Communication Cycle Period ..... 164
1007h Synchronous Window Length ..... 165
1008h Manufacturer Device Name ..... 165
1009h Manufacturer Hardware Version ..... 166
100Ah Manufacturer Software Version ..... 166
100Ch Guard Time ..... 166
100Dh Live Time Factor ..... 167
1010h Store Parameters ..... 168
1011h Restore Default Parameters ..... 171
1014h COB-ID EMCY ..... 175
1016h Consumer Heartbeat Time ..... 175
1017h Producer Heartbeat Time ..... 177
1018h Identity Object ..... 177
1019h Synchronous Counter Overflow Value ..... 179
1020h Verify Configuration ..... 180
1029h Error Behavior ..... 181
1400h Receive PDO 1 Communication Parameter ..... 182
1401h Receive PDO 2 Communication Parameter ..... 183
1402h Receive PDO 3 Communication Parameter ..... 184
1403h Receive PDO 4 Communication Parameter ..... 185
1404h Receive PDO 5 Communication Parameter ..... 187
1405h Receive PDO 6 Communication Parameter ..... 188

1406h Receive PDO 7 Communication Parameter....189
1407h Receive PDO 8 Communication Parameter....190
1600h Receive PDO 1 Mapping Parameter....191
1601h Receive PDO 2 Mapping Parameter....194
1602h Receive PDO 3 Mapping Parameter....196
1603h Receive PDO 4 Mapping Parameter....198
1604h Receive PDO 5 Mapping Parameter....201
1605h Receive PDO 6 Mapping Parameter....203
1606h Receive PDO 7 Mapping Parameter....205
1607h Receive PDO 8 Mapping Parameter....207
1800h Transmit PDO 1 Communication Parameter....209
1801h Transmit PDO 2 Communication Parameter....211
1802h Transmit PDO 3 Communication Parameter....213
1803h Transmit PDO 4 Communication Parameter....216
1804h Transmit PDO 5 Communication Parameter....218
1805h Transmit PDO 6 Communication Parameter....220
1806h Transmit PDO 7 Communication Parameter....222
1807h Transmit PDO 8 Communication Parameter....224
1A00h Transmit PDO 1 Mapping Parameter....226
1A01h Transmit PDO 2 Mapping Parameter....229
1A02h Transmit PDO 3 Mapping Parameter....231
1A03h Transmit PDO 4 Mapping Parameter....234
1A04h Transmit PDO 5 Mapping Parameter....236
1A05h Transmit PDO 6 Mapping Parameter....239
1A06h Transmit PDO 7 Mapping Parameter....241
1A07h Transmit PDO 8 Mapping Parameter....244
1C00h Sync Manager Communication Type....246
1F50h Program Data....248
1F51h Program Control....249
1F57h Program Status....250
1F80h NMT Startup....251
2005h CANopen Baudrate....252
2007h CANopen Config....252
2009h CANopen NodeID....254
2028h MODBUS Slave Address....254
202Ah MODBUS RTU Baudrate....255
202Ch MODBUS RTU Stop Bits....255
202Dh MODBUS RTU Parity....256
2030h Pole Pair Count....256
2034h Upper Voltage Warning Level....257
2035h Lower Voltage Warning Level....257
2038h Brake Controller Timing....258
2039h Motor Currents....260
203Dh Torque Window....262
203Eh Torque Window Time Out....262
203Fh Max Slippage Time Out....263
2057h Clock Direction Multiplier....263
2058h Clock Direction Divider....264
205Bh Clock Direction Or Clockwise/Counter Clockwise Mode....264
205Ch Pulse Generator Configuration....264
2101h Fieldbus Module Availability....266
2102h Fieldbus Module Control....268
2103h Fieldbus Module Status....269
2290h PDI Control....271
2291h PDI Input....272
2292h PDI Output....273
2300h NanoJ Control....274
2301h NanoJ Status....275
2302h NanoJ Error Code....276

2303h NanoJ Metadata....277
230Eh Timer....279
230Fh Uptime Seconds....280
2310h NanoJ Input Data Selection....281
2320h NanoJ Output Data Selection....282
2330h NanoJ In/output Data Selection....283
2400h NanoJ Inputs....285
2500h NanoJ Outputs....286
2701h Customer Storage Area....287
2800h Bootloader And Reboot Settings....288
306Ch Velocity Actual Internal Value....289
3202h Motor Drive Submode Select....289
3203h Feedback Selection....290
3204h Feedback Mapping....292
3205h Feedback Use....294
320Dh Moment Of Inertia....295
3211h Motor Drive Set Point....297
3219h Current Configuration....298
321Ah Current Controller Parameters....300
321Bh Velocity Controller Parameters....302
321Ch Position Controller Parameters....303
321Dh Feedforward....304
321Eh Voltage Limit....305
3220h Analog Input Digits....306
323Ah User Pin Settings....307
3241h Digital Input Position Capture....309
3242h Digital Input Routing....312
324Ah Inputs....319
3250h Digital Outputs Control....320
3252h Digital Output Routing....321
325Ah Outputs....322
3260h Pwm Output 0....323
3261h Pwm Output 1....324
3262h Pwm Output 2....326
3273h Generic SPI Hardware Configuration....327
3274h Generic SPI Mosi Data....328
3275h Generic SPI Miso Data....328
3320h Analog Input Values....329
3321h Analog Input Offsets....330
3322h Analog Input Numerators....331
3323h Analog Input Denominators....332
3380h Feedback Sensorless....333
3390h Feedback Hall....335
33A0h Feedback Incremental A/B/I 1....339
33B0h Feedback SSI 1....340
3700h Deviation Error Option Code....344
3701h Limit Switch Error Option Code....345
4014h Operating Conditions....346
4015h Special Drive Modes....348
4021h Ballast Configuration....349
4040h Drive Serial Number....352
4041h Device Id....352
4042h Bootloader Infos....352
6007h Abort Connection Option Code....354
603Fh Error Code....355
6040h Controlword....356
6041h Statusword....357
605Ah Quick Stop Option Code....358
605Bh Shutdown Option Code....359

605Ch Disable Operation Option Code....360
605Dh Halt Option Code....360
605Eh Fault Reaction Option Code....361
6060h Modes Of Operation....362
6061h Modes Of Operation Display....363
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....366
6068h Position Window Time....366
606Bh Velocity Demand Value....367
606Ch Velocity Actual Value....367
606Dh Velocity Window....368
606Eh Velocity Window Time....368
606Fh Velocity Threshold....369
6070h Velocity Threshold Time....369
6071h Target Torque....370
6072h Max Torque....370
6073h Max Current....371
6074h Torque Demand....372
6075h Motor Rated Current....372
6077h Torque Actual Value....373
607Ah Target Position....373
607Bh Position Range Limit....374
607Ch Home Offset....375
607Dh Software Position Limit....375
607Eh Polarity....376
607Fh Max Profile Velocity....377
6080h Max Motor Speed....378
6081h Profile Velocity....378
6082h End Velocity....379
6083h Profile Acceleration....379
6084h Profile Deceleration....380
6085h Quick Stop Deceleration....380
6086h Motion Profile Type....380
6087h Torque Slope....381
6091h Gear Ratio....382
6092h Feed Constant....383
6096h Velocity Factor....384
6097h Acceleration Factor....385
6098h Homing Method....386
6099h Homing Speed....386
609Ah Homing Acceleration....388
60A2h Jerk Factor....388
60A4h Profile Jerk....389
60A8h SI Unit Position....391
60A9h SI Unit Velocity....391
60B0h Position Offset....392
60B1h Velocity Offset....392
60B2h Torque Offset....393
60C1h Interpolation Data Record....393
60C2h Interpolation Time Period....394
60C4h Interpolation Data Configuration....395
60C5h Max Acceleration....398
60C6h Max Deceleration....398
60E4h Additional Position Actual Value....399
60E5h Additional Velocity Actual Value....400

60E6h Additional Position Encoder Resolution - Encoder Increments....401
60EBh Additional Position Encoder Resolution - Motor Revolutions....402
60F1h Additional Encoder Inversion....403
60F2h Position Option Code....404
60F4h Following Error Actual Value....406
60F8h Max Slippage....406
60FAh Control Effort....407
60FCh Position Demand Internal Value....407
60FDh Digital Inputs....408
60FEh Digital Outputs....409
60FFh Target Velocity....410
6502h Supported Drive Modes....410
6503h Drive Catalogue Number....411
6505h Http Drive Catalogue Address....412

**11 Copyrights....413**
11.1 Einführung....413
11.2 AES....413
11.3 MD5....413
11.4 DHCP....414
11.5 CMSIS DSP Software Library....414
11.6 Protothreads....414
11.7 lwIP....415
11.8 littlefs....415

# 1 Einleitung

Die *N6* 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 Produkt finden Sie auf www.nanotec.de.

## 1.1 Produktdokumentation

Alle Produktdokumentationen stehen auf der Produktsetite auf www.nanotec.de zur Verfügung.

| Dokument | Inhalt |
|---|---|
| Technisches Handbuch N6 | Installation, Inbetriebnahme, Parametrierung/Programmierung, Beschreibung der Funktionen der Motorsteuerung |
| Funktionsbeschreibung STO | Beschreibung der integrierten Sicherheitsfunktion *STO* für die Varianten, die über diese verfügen (N6-...-**S**) |

## 1.2 Versionshinweise

| Version Handbuch | Datum | Änderungen | Version Firmware |
|---|---|---|---|
| 1.0.0 | 01/2025 | Veröffentlichung | FIR-v2451 |
| 1.1.0 | 04/2025 | Korrekturen<br>USB-Mass-Storage ersetzt durch MTP-Gerät, PowerShell- und Linux-Scripts zu finden im Kapitel Konfiguration über USB, | FIR-v2508 |

## 1.3 Urheberrecht, Kennzeichnung und Kontakt

© 2013 – 2025 Nanotec Electronic GmbH & Co. KG. Alle Rechte vorbehalten.

CE

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

Microsoft® Windows® 98/NT/ME/2000/XP/7/10/11 und PowerShell® sind eingetragene Warenzeichen der Microsoft Corporation.

## 1.4 Bestimmungsgemäße Verwendung

Die *N6* dient der Steuerung von Schritt- und BLDC-Motoren und findet Verwendung als Komponente von Antriebssystemen in vielfältigen Industrieanwendungen.

Verwenden Sie das Produkt bestimmungsgemäß innerhalb der durch die technischen Daten definierten Grenzen (siehe insbesondere Zulässige Betriebsspannung) und unter den freigegebenen Umgebungsbedingungen.

Mit Ausnahmen der Produktvarianten mit integrierter Sicherheitsfunktion *STO* (N6-x-x-x-**S**) darf dieses Nanotec-Produkt unter keinen Umständen als Sicherheitsbauteil in ein Produkt oder eine Anlage integriert werden. Für die STO-Varianten beachten Sie zusätzlich die Voraussetzungen für die Verwendung und die Sicherheitshinweise im Dokument *Funktionsbeschreibung STO*.

Alle Produkte, in denen eine von Nanotec hergestellte Komponente enthalten ist, müssen bei Übergabe an Endnutzer entsprechende Warnhinweise samt Anleitung für sichere Verwendung und sicheren Betrieb enthalten. Alle von Nanotec bereitgestellten Warnhinweise müssen unmittelbar an den Endnutzer weitergegeben werden.

## 1.5 Zielgruppe und Qualifikation

Das Produkt und diese Dokumentation richten sich an technisch geschulte Fachkräfte wie:

- Entwicklungsingenieure
- Anlagenkonstrukteure
- Monteure/Servicekräfte
- Applikationsingenieure

Nur Fachkräfte dürfen das Produkt installieren, programmieren und in Betrieb nehmen. Fachkräfte sind Personen, die

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

## 1.6 Gewährleistung und Haftungsausschluss

Nanotec haftet nicht für Schäden und Fehlfunktion durch Montagefehler, Nichtbeachten dieses Dokuments oder sachwidrige Reparatur. Verantwortlich für Auswahl, Betrieb, Nutzung unserer Produkte sind Anlagenkonstrukteur, Betreiber und Nutzer. Nanotec verantwortet keine Produktintegration im Endsystem. Es gelten die allgemeinen Geschäftsbedingungen auf www.nanotec.de. **Anm.:** Produktumbau / -änderung, sowie das Öffnen sind untersagt.

## 1.7 Mitgeltende Vorschriften

Neben diesem technischen Handbuch sind folgende Vorschriften zu beachten:

- Unfallverhütungsvorschriften
- örtliche Vorschriften zur Arbeitssicherheit

## 1.8 EU-Richtlinien zur Produktsicherheit

Folgende EU-Richtlinien wurden beachtet:

- RoHS-Richtlinie (2011/65/EU, 2015/863/EU)
- EMV-Richtlinie (2014/30/EU)

## 1.9 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**

**Verweist auf eine mögliche Fehlbedienung des Produkts.**

Die Missachtung des Hinweises führt möglicherweise zu Beschädigungen an diesem Produkt oder anderen Produkten.

► Beschreibt, wie Sie die Fehlbedienung vermeiden.

**TIPP**

Zeigt einen Tipp zur Anwendung oder Aufgabe.

### STO-spezifische Hinweise

Folgende Hinweise gelten nur für die Produkte mit dr sicherheitsfunktion STO (*Safe Torque Off*).

**VORSICHT**

**Der Hinweis VORSICHT mit dem STO-Symbol verweist auf eine möglicherweise gefährliche Situation, die durch den Verlust der Sicherheitsfunktion entstehen kann.**

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

► Beschreibt, wie Sie die gefährliche Situation vermeiden.

**Hinweis**

**Der Hinweis mit dem STO-Symbol verweist auf eine mögliche Fehlbedienung des Produkts mit STO-Funktion.**

Die Missachtung des Hinweises führt möglicherweise zu Beschädigungen an diesem Produkt oder anderen Produkten.

► Beschreibt, wie Sie die Fehlbedienung vermeiden.

## 1.10 Hervorhebungen im Text

Im Dokument gelten folgende Konventionen:

Ein unterstrichener Text markiert Querverweise und Hyperlinks:

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

## 1.11 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 `1003h:05h`, der Subindex 00 des Objekts $6040_h$ mit `6040h`.

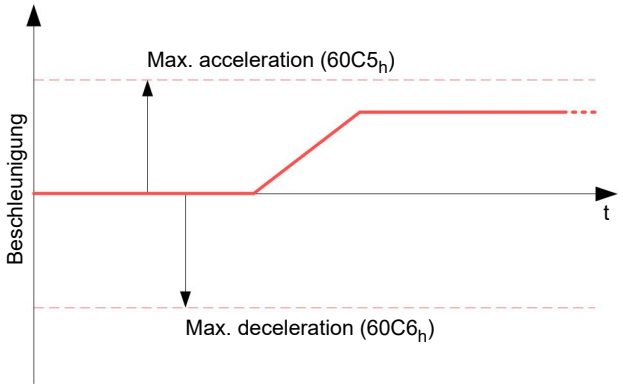
## 1.12 Bits

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

| | MSB | | | | | | | LSB | |
|---|---|---|---|---|---|---|---|---|---|
| Bit Nummer | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Bits | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | ≙ $55_{hex}$ ≙ $85_{dec}$ |

## 1.13 Zählrichtung (Pfeile)

In Abbildungen gilt die Zählrichtung immer in Richtung eines Pfeiles. Die in der nachfolgenden Abbildung beispielhaft dargestellten Objekte `60C5h` und `60C6h` werden beide positiv angegeben.



# 2 Sicherheits- und Warnhinweise

**VORSICHT**

**Der Verlust der Sicherheitsfunktion führt möglicherweise zu Verletzungen!**

► Beachten Sie die Einsatzbedingungen für die STO-Funktion und das Produkt, wie im Dokument *Funktionsbeschreibung STO* beschrieben

► Überbrücken Sie die STO-Funktion nicht.

► Verwenden Sie die STO-Funktion erst, wenn alle Schutzmaßnahmen eingerichtet sind und die Validierung der Maschine/Anlage nach der Inbetriebnahme abgeschlossen ist.

**VORSICHT**

**Der Verlust der Sicherheitsfunktion führt möglicherweise zu Verletzungen!**

Wenn die Versorgungsspannung 60 V überschreitet, kann die STO-Funktion nicht gewährleistet werden.

► Verwenden Sie für die Versorgung der Prodktvarianten mit STO-Funktion ausschließlich PELV-Stromkreise nach IEC 60204-1, die (auch im Fehlerfall) eine absolut maximale Spannung von 60 V erlauben.

► Bei Verwendung von Batterien achten sie auf eine maximale Nennspannung von 48 V DC (mit einer Toleranz von + 20% ), konform mit DVC A oder B nach IEC 61800-5-1.

**VORSICHT**

**Der Verlust der Sicherheitsfunktion führt möglicherweise zu Verletzungen!**

Zu hohe Temperatur kann die Elektronik des Controllers und der STO beschädigen. Dies führt möglicherweise zu Verlust der STO-Funktion.

► Beachten Sie die freigegebenen Umgebungsbedingungen und die zugelassenen Leistungswerte..

► Bauen Sie das Produkt so ein, dass für die Kühlung saubere uns trockene Kühlluft zur Verfügung steht und diese Luft das Produkt umströmen kann.

**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**

**Beschädigung 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**

**Beschädigung der Elektronik durch unsachgemäßen Umgang mit ESD-empfindlichen Bauteilen!**

Das Gerät enthält Bauteile, die empfindlich gegen elektrostatische Entladung sind. Unsachgemäßer Umgang kann das Gerät beschädigen.

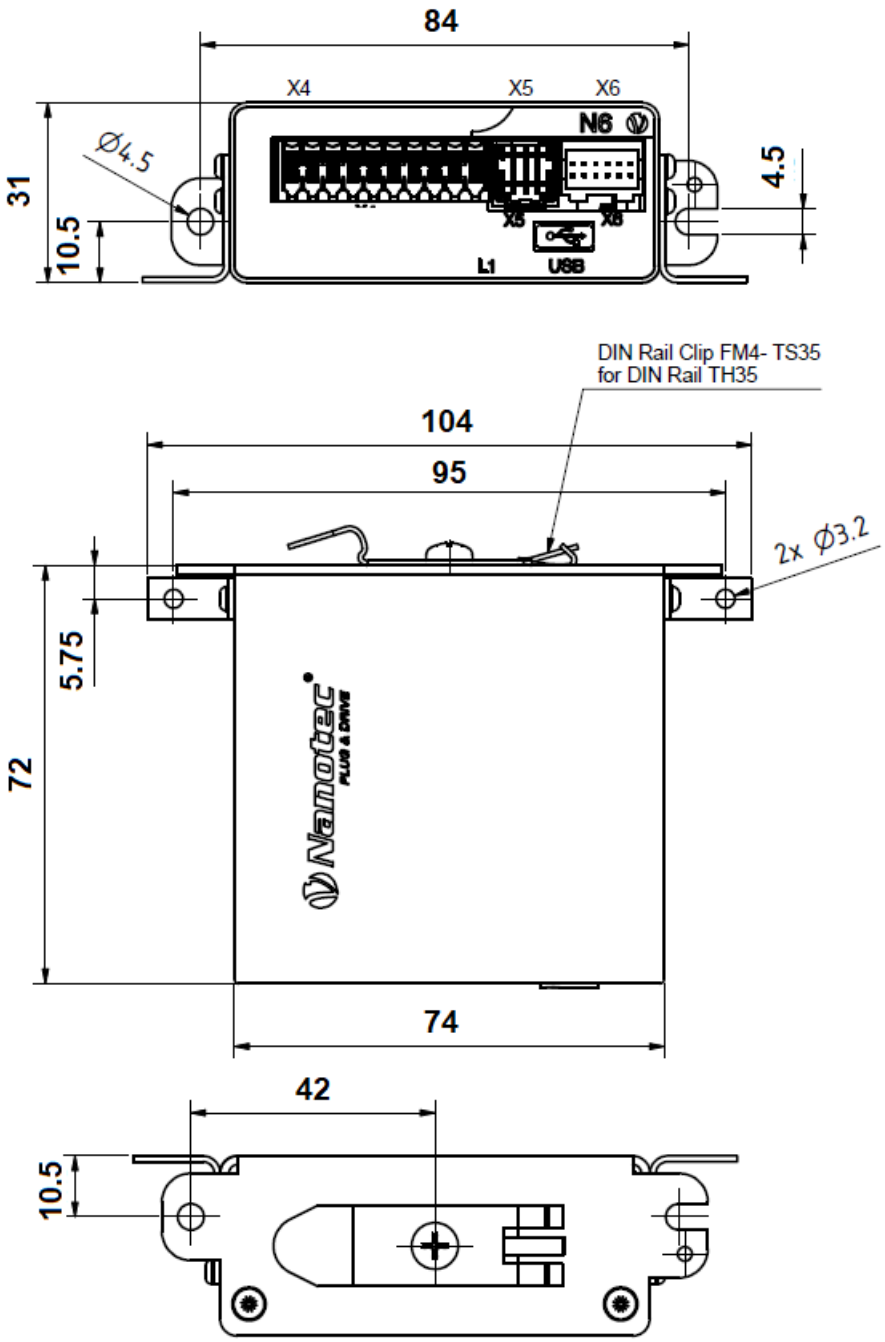
► Beachten Sie die Grundprinzipien des ESD-Schutzes beim Umgang mit dem Gerät.

# 3 Technische Daten und Anschlussbelegung

## 3.1 Umgebungsbedingungen

| Umgebungsbedingung | Wert |
|---|---|
| Schutzklasse | IP20 |
| Verschmutzungsgrad | 2 |
| Umgebungstemperatur (Betrieb) | -10 … +40°C |
| Umgebungstemperatur (Lagerung und Transport) | -25 … +85°C |
| Rel.Luftfeuchtigkeit (Betrieb), nicht kondensierend | 0 … 85% |
| Rel. Luftfeuchtigkeit (Lagerung und Transport), nicht kondensierend | 0 … % |
| Abs.. Luftfeuchtigkeit (Lagerung und Transport), nicht kondensierend | 30 g/m$^3$ |
| Max. Aufstellhöhe über *NN* | 2000 m (Leistungseinschränkung ab 1000 m: -1%/100 m) |
| Max. Aufstellhöhe über *NN* (Lagerung und Transport) | 3000 m |

## 3.2 Maßzeichnungen und Montagemöglichkeiten




Sie können die Steuerung mit Schrauben an den seitlichen Laschen auf eine ebene Montagefläche oder mit der mitgelieferten Hutschienenklammer an einer TH35-Hutschiene in Ihrem Schaltschrank befestigen.

## 3.3 Elektrische Eigenschaften und technische Daten

| Eigenschaft | Beschreibung/Wert |
|---|---|
| Betriebsspannung | 12 V -5%…57,6 V DC |
| Nennstrom | 6 $A_{eff}$ |
| Spitzenstrom | N6-1-… (*low current*): 6 $A_{eff}$ |

| Eigenschaft | Beschreibung/Wert |
|---|---|
| | N6-2-… (*high current*): 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 | *Profile Position Mode*, *Profile Velocity Mode*, *Profile Torque Mode*, *Homing Mode*, *Interpolated Position Mode*, *Cyclic Sync Position Mode*, *Cyclic Sync Velocity Mode*, *Cyclic Synchronous Torque Mode*, *Takt-Richtung-Modus* |
| Sollwertvorgabe/ Programmierung | *CANopen*, *Takt-Richtung*, *Analog*, *NanoJ-Programm* |
| Schnittstellen | CANopen, USB |
| Eingänge | ■ 6 Eingänge 5 V/24 V (=UB_Logic), per Software umschaltbar, Werkseinstellung: 5 V<br>■ 2 Analogeingänge 0 bis +24 V, 12-Bit-Auflösung |
| Ausgänge | 3 Ausgänge, 5 V/24 V (=UB_Logic), per Software umschaltbar,100 mA |
| Bremsenanschluss | 1 PWM-Ausgang, max. 1,5 A, 20 KHz |
| Sensoreneingänge | 1 Inkrementallencoder (5 V), 3 Hallsensoren (5 V), 1 SSI-Encoder (10 V) |
| Schutzschaltung | Über- und Unterspannungsschutz<br>Übertemperaturschutz (> 75° Celsius auf der Leistungsplatine)<br>Verpolungsschutz |

## 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.

**HINWEIS**

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

Achten Sie darauf, dass eine Temperaturableitung durch die Montagefläche und eine passive Kühlung oder aktive Lüftung möglich sind, sodass die maximale Umgebungstemperatur nicht überschritten wird.

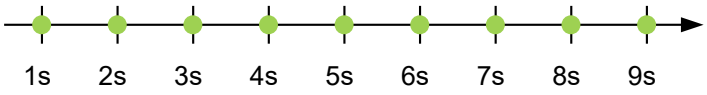
## 3.5 LED-Signalisierung

### 3.5.1 Betriebs-LED

Die Betriebs-LED zeigt den aktuellen Status an.

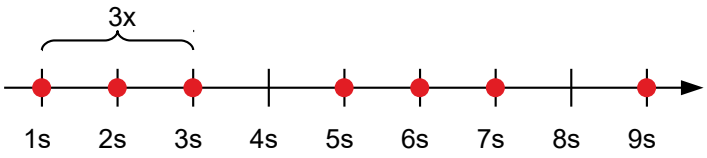
#### 3.5.1.1 Normaler Betrieb

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




#### 3.5.1.2 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 $1003_h$ ein genauerer Fehlercode hinterlegt.

**TIPP**

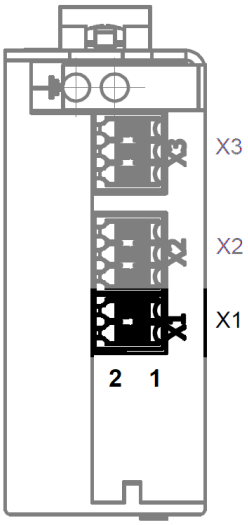
Sie können die Betriebs-LEDs mit $3250_h$:$09_h$ ausschalten.

## 3.6 Anschlussbelegung

### 3.6.1 X1 – Motoranschluss

- Typ: Degson 15EDGRHD-3.5-04P-04
- Gegenstecker (im Lieferumfang enthalten): Degson 15EDGKNHBM-3.5-04P (oder äquivalent)
- Leiterquerschnitt, min. : 1 $mm^2$ (AWG 17)

Pins 1 und 2 sind unten markiert.




| Pin | Funktion (Schrittmotor) | Funktion (BLDC-Motor) | Bemerkung |
|---|---|---|---|
| 1 | A | U | |
| 2 | A\ | V | |
| 3 | B | W | |
| 4 | B\ | nicht benutzt | |

**HINWEIS**

Motorleitungen sind durch Ferrite zu führen (74271222 von Würth oder äquivalent).

### 3.6.2 X2 – Spannungsversorgung

- Typ: Degson 15EDGRHD-3.5-06P
- Gegenstecker (im Lieferumfang enthalten): Degson 15EDGKNHB-3.5-06P (oder äquivalent)
- □ Leiterquerschnitt, min. :
  - 1 $mm^2$ (AWG 17) für Hauptversorgung UB
  - 0.8 $mm^2$ (AWG 18) für UB_Logic

#### 3.6.2.1 Spannungsquelle

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

**VORSICHT**

STO

**Der Verlust der Sicherheitsfunktion führt möglicherweise zu Verletzungen!**

Wenn die Versorgungsspannung 60 V überschreitet, kann die STO-Funktion nicht gewährleistet werden.

► Verwenden Sie für die Versorgung der Prodktvarianten mit STO-Funktion ausschließlich PELV-Stromkreise nach IEC 60204-1, die (auch im Fehlerfall) eine absolut maximale Spannung von 60 V erlauben.

► Bei Verwendung von Batterien achten sie auf eine maximale Nennspannung von 48 V DC (mit einer Toleranz von + 20% ), konform mit DVC A oder B nach IEC 61800-5-1.

**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.**

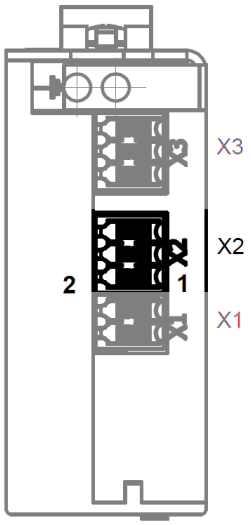
► Es ist empfohlen, ein EMI-Filter (810911010 von Würth oder äquivalent) ist in die DC-Zuleitung(en) mit möglichst geringem Abstand zur Steuerung/Motor einzufügen.

► Lange Daten- oder Versorgungsleitungen sind durch Ferrite zu führen.

► Motorleitungen sind durch Ferrite zu führen (74271222 von Würth oder äquivalent).

### 3.6.2.2 Anschlüsse

Pins 1 und 2 sind unten markiert.




| Pin | Funktion | Bemerkung |
|---|---|---|
| 1 | GND_L | GND für die Logikversorgung UB_Logic |
| 2 | +UB_Logic | Eingangsspannung für die Logikversorgung 12 V - 30 V DC, |
| 3 | Ballast- | Anschluss für optionalen externen Ballastwiderstand, max. 6,5 A für 1 s, danach min. 1 s ausschalten |
| 4 | Ballast+ | Anschluss für optionalen externen Ballastwiderstand, max. 6,5 A für 1 s, danach min. 1 s ausschalten |
| 5 | GND_P | GND für die Hauptversorgung UB |
| 6 | +UB | Eingangsspannung für die Hauptversorgung (Leistungsteil), 12 V - 57,6 V DC |

**HINWEIS**

Die Logikversorgung versorgt die Elektronik, die Sensoren und die Kommunikationsschnittstelle.

Die Wicklungen des Motors werden nicht von der Logikversorgung versorgt.

### 3.6.2.3 Zulässige Betriebsspannung

Die maximale Betriebsspannung beträgt 57,6 V DC. Steigt die Eingangsspannung der Steuerung über den in $2034_h$ eingestellten Schwellwert, wird der Motor abgeschaltet und ein Fehler ausgelöst. Ab der in $4021_h$:$02_h$ eingestellten Ansprechschwelle wird der optionale externe Ballastwiderstand) dazugeschaltet.

Die minimale Betriebsspannung beträgt 12 V DC. Fällt die Eingangsspannung der Steuerung unter 10 V, wird der Motor abgeschaltet und ein Fehler ausgelöst.

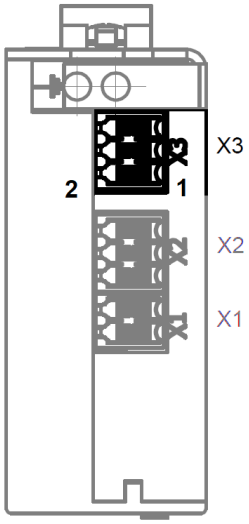
An die Versorgungsspannung muss ein Ladekondensator von mindestens 4700 µF / 50 V (ca. 1000 µF pro Ampere Nennstrom) angeschlossen sein, um ein Überschreiten der zulässigen Betriebsspannung (z. B. beim Bremsvorgang) zu vermeiden. Es ist empfohlen, ein EMI-Filter (810911010 von Würth oder äquivalent) ist in die DC-Zuleitung(en) mit möglichst geringem Abstand zur Steuerung/Motor einzufügen.

### 3.6.3 X3 - STO (*Safe Torque Off*)

Anschluss für die STO-Signale. Dieser Anschluss und die entsprechende integrierte Sicherheitsfunktion *STO* stehen nur bei den STO-Varianten des Produkts zur Verfügung (N6-x-x-x-**S**).

- Typ: Degson 15EDGRHD-3.5-06P-04
- Gegenstecker (im Lieferumfang enthalten): Degson 15EDGKNHBM-3.5 (oder äquivalent)

Pins 1 und 2 sind unten markiert.




| VORSICHT |
|---|
| **Der Verlust der Sicherheitsfunktion durch elektromagnetische Störungen führt möglicherweise zu Verletzungen!**<br>Externe Störungen können die Sicherheitsfunktion beeinflussen und zu deren Verlust führen.<br>► Beachten Sie die maximal zulässige Leitungslänge von 30 m für alle STO-Signale. Längere Leitungen reduzieren die Störfestigkeit (EMV) und erfordern zusätzliche Entstör- und Schutzmaßnahmen.<br>► Verwenden Sie geschirmte Leitungen für die STO-Signale.<br>► Verlegen Sie Versorgungs-, Signal- und Steuerleitungen räumlich getrennt. |

| VORSICHT |
|---|
| **Der Verlust der Sicherheitsfunktion durch falsche Verdrahtung führt möglicherweise zu Verletzungen!**<br>Falsche Verdrahtung oder die Verwendung ungeeigneter externer Bauteile können zu Verlust der Sicherheitsfunktion führen. Dies führt möglicherweise zu Verletzungen.<br>► Verwenden Sie ausschließlich Bauteile entsprechend der Sicherheitskategorie der Applikation. |

| VORSICHT |
| --- |
| ► Prüfen Sie die elektrische Installation (Verkabelung, Anschlussbelegung) und validieren Sie die STO-Funktion vor dem erstmaligen Betrieb und nach jedem Eingriff in die Verdrahtung und jedem Austausch von Bauteilen/Betriebsmitteln.<br>► Überbrücken Sie die STO-Funktion nicht. Falls die Beschaltung für die erste Inbetriebnahme nicht der geforderten Sicherheitskategorie der Applikation entspricht, entfernen Sie diese unmittelbar nach der ersten Inbetriebnahme. |

| Pin | Funktion | Bemerkung |
| --- | --- | --- |
| 1 | STOA_IN | STO-Kanal A, PWM-Signalunterdrückung<br>Nennspannung 24 V DC, max. 60 V DC |
| 2 | STOA_RTN | GND-Anschluss für STO-Kanal A |
| 3 | STO_FBH | STO-Feedback, High |
| 4 | STO_FBL | STO-Feedback, Low |
| 5 | STOB_IN | STO-Kanal B, Gate-Treiber-Schalter<br>Nennspannung 24 V DC, max. 60 V DC |
| 6 | STOB_RTN | GND-Anschluss für STO-Kanal B |

**STO-Funktion**

Fällt die Spannung an einem der zwei STO-Eingängen (Kanal A und B) unter 5 V, wird die STO-Funktion ausgelöst und der Motor kann kein Drehmoment mehr erzeugen.

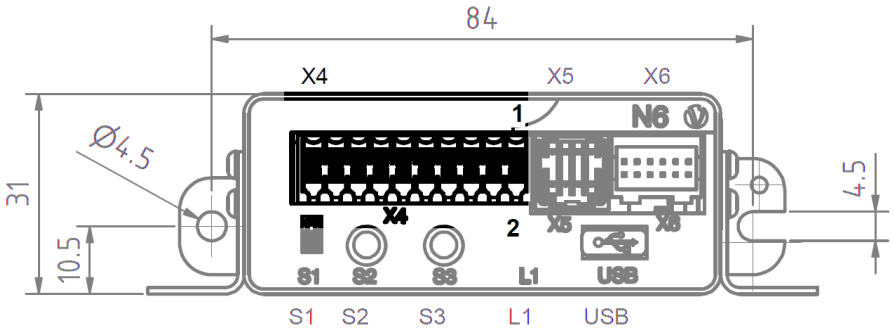
Der Feedback-Ausgang dient der Überwachung des Status der STO-Funktion. Wenn die STO-Funktion aktiv ist, schließt der Kontakt zwischen *STO_FBH* und *STO_FBL*.

Eine ausführliche Beschreibung der STO-Funktion finden Sie im Dokument *Funktionsbeschreibung STO*.

## 3.6.4 X4 – Ein- und Ausgänge

- Typ: Degson 15EDGRHD-3.5-20P-14
- Gegenstecker (im Lieferumfang enthalten): Degson 15EDGKNHB-3.5-20P (oder äquivalent)

Pins 1 und 2 sind unten markiert.




| Pin | Funktion | Bemerkung |
| --- | --- | --- |
| 1 | +10 V | +10 V DC, Ausgangsspannung, max. 350 mA |
| 2 | GNDD | GND für Digitaleingänge/-Ausgänge |
| 3 | +5 V | +5 V DC, Ausgangsspannung, max. 350 mA |
| 4 | GNDD | GND für Digitaleingänge/-Ausgänge |

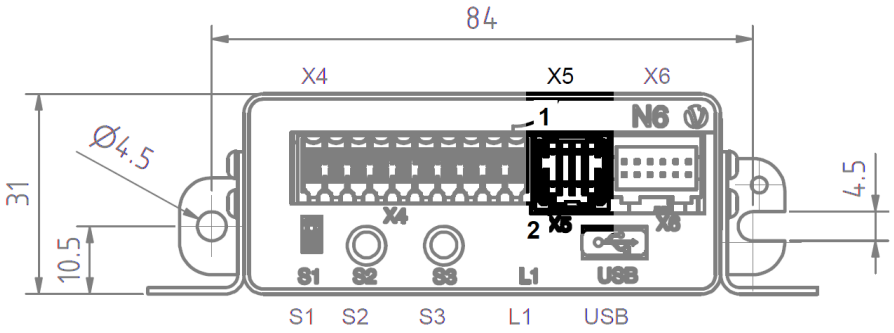
| Pin | Funktion | Bemerkung |
|---|---|---|
| 5 | Ausgang 1 | Digitalausgang, 5 V / 24 V (UB_Logic) umschaltbar per Software mit Objekt 323A$_{hh}$, max. 100 mA |
| 6 | Ausgang 2 | Digitalausgang, 5 V / 24 V (UB_Logic) umschaltbar per Software mit Objekt 323A$_{hh}$, max. 100 mA |
| 7 | Ausgang 3 | Digitalausgang, 5 V / 24 V (UB_Logic) umschaltbar per Software mit Objekt 323A$_{hh}$, max. 100 mA |
| 8 | GNDD | GND für Digitaleingänge/-Ausgänge |
| 9 | Eingang 1 | Digitaleingang 5 V / 24 V umschaltbar per Software mit Objekt 323A$_{hh}$, Takteingang in Takt-Richtungs-Modus |
| 10 | Eingang 2 | Digitaleingang 5 V / 24 V umschaltbar per Software mit Objekt 323A$_{h}$, Richtungseingang in Takt-Richtungs-Modus |
| 11 | Eingang 3 | Digitaleingang 5 V / 24 V umschaltbar per Software mit Objekt 323A$_{h}$, max. 1 MHz |
| 12 | Eingang 4 | Digitaleingang 5 V / 24 V, umschaltbar per Software mit Objekt 323A$_{h}$, max. 1 MHz |
| 13 | Eingang 5 | Digitaleingang 5 V bis 24 V, umschaltbar per Software mit Objekt 323A$_{h}$ |
| 14 | Eingang 6 | Digitaleingang 5 V bis 24 V, umschaltbar per Software mit Objekt 323A$_{h}$ |
| 15 | GNDA | GND für Analogeingang |
| 16 | Analogeingang 1 | 0 V…+24 V, 12-Bit-Auflösung |
| 17 | GNDA | GND für Analogeingang |
| 18 | Analogeingang 2 | 0 V…+24 V, 12-Bit-Auflösung |
| 19 | Bremse - | GND für Bremse |
| 20 | Bremse + | PWM-gesteuerter Ausgang, 5 V / 24 V umschaltbar per Software mit Objekt 323A$_{hh}$, bis 20 KHz, max. 1500 mA |

### 3.6.5 X5 – SSI-Encoder

Anschluss für SSI-Encoder

- Typ: Molex MicroClasp 55959-0830
- Gegenstecker (im Lieferumfang nicht enthalten): Molex 51353-0800 (oder äquivalent)

Pins 1 und 2 sind unten markiert.



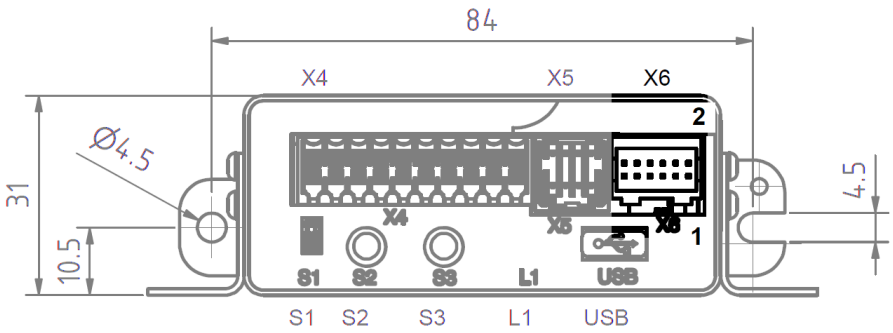

| Pin | Funktion | Bemerkung |
|---|---|---|
| 1 | GND | |
| 2 | SHIELD | Anschluss für die Schirmung |

| Pin | Funktion | Bemerkung |
|---|---|---|
| 3 | n.c. | reserviert |
| 4 | DATA B | |
| 5 | DATA A | |
| 6 | CLCK B | bis zu 10 MHz |
| 7 | CLCK A | bis zu 10 MHz |
| 8 | Vcc | +10 V DC, Ausgangs- und Versorgungsspannung für SSI-Encoder, max. 350 mA |

### 3.6.6 X6 – Inkrmentalencoder/Hall-Sensoren

- Typ: JST S12B-PADSS-1
- Gegenstecker (im Lieferumfang nicht enthalten):
  - Gehäuse: JST PADP-12V-1-S (oder äquivalent)
  - Buchsenkontakte: JST SPH-001T-P0.5L (oder äquivalent)
- Passende Nanotec-Kabel (nicht im Lieferumfang enthalten):
  - ZK-PADP-12-500-S
  - ZK-M12-8-2M-2-PADP
  - ZK-M12-12-2M-2-PADP
  - ZK-NTO3-10-500-PADP / ZK-NTO3-10-1000-PADP
  - ZK-NOE-10-500-S-PADP
  - ZK-WEDL-500-S-PADP

Pin 1 und Pin 2 sind im Bild markiert.




| Pin | Funktion | Bemerkung |
|---|---|---|
| 1 | GND | |
| 2 | Vcc | 5 V DC, Ausgangs- und Versorgungsspannung für Encoder / Hall Sensor; max. 350 mA |
| 3 | A | 5 V Signal, max. 1 MHz |
| 4 | B | 5 V Signal, max. 1 MHz |
| 5 | A\ | 5 V Signal, max. 1 MHz |
| 6 | B\ | 5 V Signal, max. 1 MHz |
| 7 | I | 5 V Signal, max. 1 MHz |
| 8 | I\ | 5 V Signal, max. 1 MHz |
| 9 | Hall 1 | 5 V Signal |
| 10 | Hall 2 | 5 V Signal |
| 11 | Hall 3 | 5 V Signal |
| 12 | Shielding | Schirmung |

**HINWEIS**

**Bei Verwendung eines Single-Ended-Encoders werden die Kanäle A/, B/ und I/ nicht ausgewertet!**

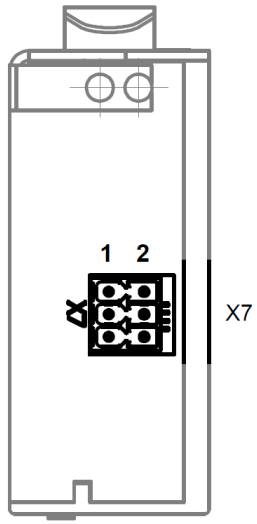
Damit ein Single-Ended-Encoder richtig erkannt wird:

► Schließen Sie an die Pins A\, B\, I\ nichts an, legen Sie diese Pins ebenfalls nicht auf die Masse (GND).

### 3.6.7 X7 – CANopen IN/OUT

- Typ: Amphenol NR061A510000H
- Gegenstecker (im Lieferumfang enthalten): NL06105xxxxxH (oder äquivalent)

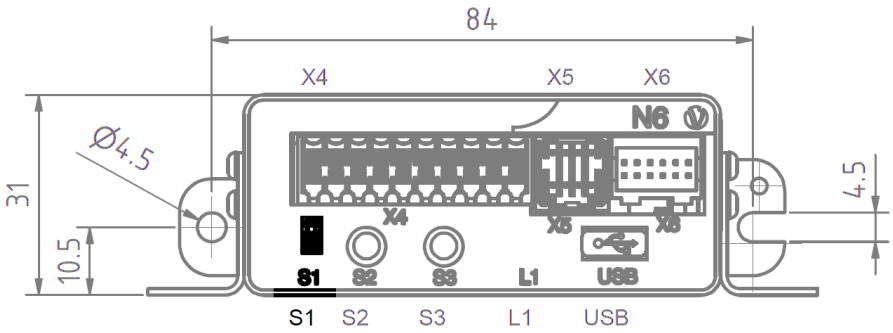
Pins 1 und 2 sind unten markiert.




| Pin | Funktion | Bemerkung |
|---|---|---|
| 1 | GND | |
| 2 | GND | |
| 3 | CAN_H | CAN-High |
| 4 | CAN_H | CAN-High |
| 5 | CAN_L | CAN-Low |
| 6 | CAN_L | CAN-Low |

### 3.6.8 S1 – 120 Ohm Terminierungswiderstand

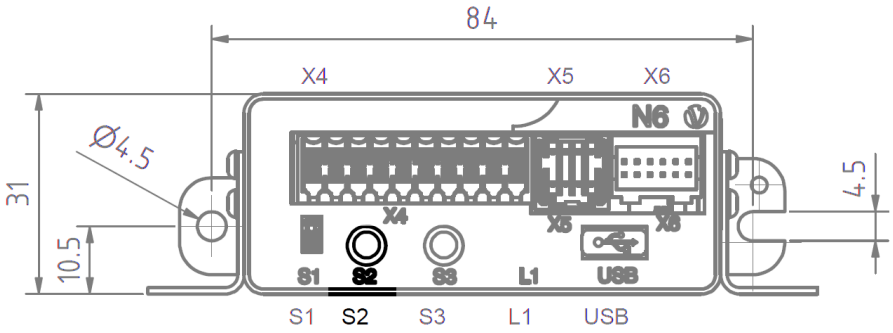
Diese DIP-Schalter schalten die Terminierung von 120 Ω zwischen CAN_H und CAN_L des CAN-Busses zu (Schalter oben) oder ab.




### 3.6.9 S2 – CANopen Node-ID und Baudrate

Hex-Codierschalter zum Einstellen der CANopen Node-ID und Baudrate. Siehe Kapitel Node-ID und Baudrate einstellen.

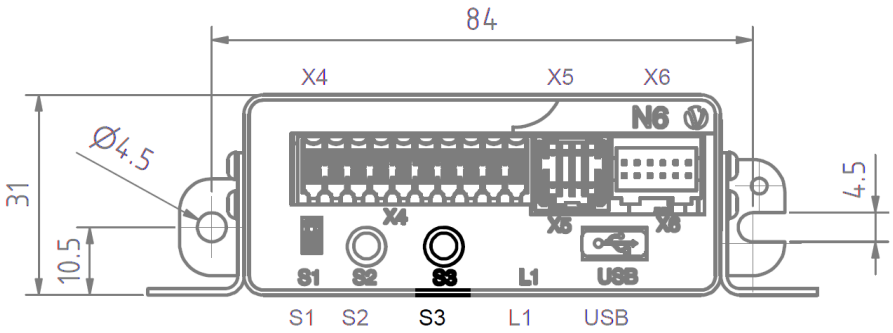
Der Wert dieses Schalters wird mit 16 multipliziert und zum Wert des Schalters S3 addiert, damit stellt dieser Schalter die sechzehner Stelle.




### 3.6.10 S3 – CANopen Node-ID und Baudrate

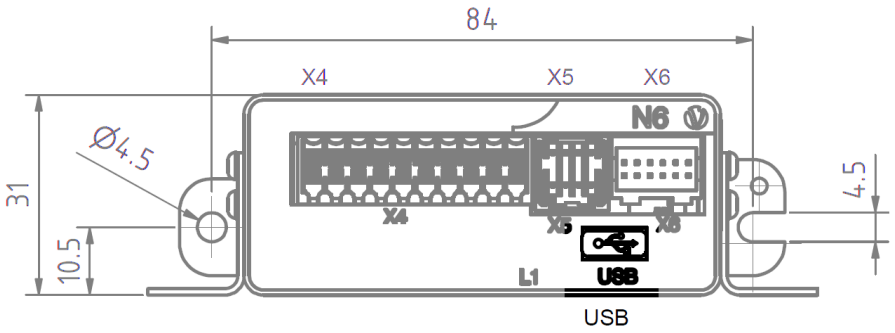
Hex-Codierschalter zum Einstellen der CANopen Node-ID und Baudrate. Siehe Kapitel Node-ID und Baudrate einstellen.

Der Wert dieses Schalters wird zum Wert des Schalters S2 addiert, damit stellt dieser Schalter die einer Stelle.




### 3.6.11 Micro-USB

Micro-USB-Anschluss für die Konfiguration/Parametrierung.


84
X4
X5
X6
N6
Ø4.5
31
10.5
4.5
X4
X5
X6
L1
USB
USB

# 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 oder über den CANopen-Bus konfigurieren.

Die Software *Plug & Drive Studio 3* bietet Ihnen eine Möglichkeit, die Konfiguration vorzunehmen und die Steuerung an den angeschlossenen Motor anzupassen. Weiterführende Informationen finden Sie im Dokument *Plug & Drive Studio: User Manual* auf www.nanotec.de.

**VORSICHT**

**Der Verlust der Sicherheitsfunktion durch falsche Verdrahtung führt möglicherweise zu Verletzungen!**

Falsche Verdrahtung oder die Verwendung ungeeigneter externer Bauteile können zu Verlust der Sicherheitsfunktion führen. Dies führt möglicherweise zu Verletzungen.

► Verwenden Sie ausschließlich Bauteile entsprechend der Sicherheitskategorie der Applikation.

► Prüfen Sie die elektrische Installation (Verkabelung, Anschlussbelegung) und validieren Sie die STO-Funktion vor dem erstmaligen Betrieb und nach jedem Eingriff in die Verdrahtung und jedem Austausch von Bauteilen/Betriebsmitteln.

► Überbrücken Sie die STO-Funktion nicht. Falls die Beschaltung für die erste Inbetriebnahme nicht der geforderten Sicherheitskategorie der Applikation entspricht, entfernen Sie diese unmittelbar nach der ersten Inbetriebnahme.

**VORSICHT**

**Der Verlust der Sicherheitsfunktion durch elektromagnetische Störungen führt möglicherweise zu Verletzungen!**

Externe Störungen können die Sicherheitsfunktion beeinflussen und zu deren Verlust führen.

► Beachten Sie die maximal zulässige Leitungslänge von 30 m für alle STO-Signale. Längere Leitungen reduzieren die Störfestigkeit (EMV) und erfordern zusätzliche Entstör- und Schutzmaßnahmen.

► Verwenden Sie geschirmte Leitungen für die STO-Signale.

► Verlegen Sie Versorgungs-, Signal- und Steuerleitungen räumlich getrennt.

**VORSICHT**

**Verletzungen durch unerwartete und unbeabsichtigte Bewegungen des Antriebs bei Installations- und Wartungsarbeiten!**

Bei fehlender oder nicht aktivierter STO-Funktion kann der Antrieb unerwartet anlaufen.

► Schalten Sie vor den Arbeiten die Energieversorgung aus.

► Sichern Sie alle Energieversorgungen gegen versehentliches Wiedereinschalten.

**HINWEIS**

**EMV: Stromführende Leitungen – insbesondere um Versorgungs- und Motorenleitungen – erzeugen elektromagnetische Wechselfelder. Diese können den Motor und andere Geräte stören.**

Geeignete Maßnahmen können sein:

► Geschirmte Leitungen verwenden und den Leitungsschirm beidseitig auf kurzem Weg erden.

► Stromversorgungs- und Motorleitungen so kurz wie möglich halten.

► Kabel mit paarweise verdrillten Adern verwenden.

► 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

**HINWEIS**

**Beschädigung des Produkts und/oder externer Hardware durch Potenzialdifferenzen am USB.**

Das Anschließen des USB-Kabels während die Elektronik versorgt wird (Hot-Plugging), führt möglicherweise zu Schäden.

► USB anschließen, bevor Sie die Versorgungsspannung einschalten.

► Wenn möglich, Potentialdifferenzen zwischen PC und Produkt ausgleichen oder USB-Isolator verwenden.

► USB-Kabel zuerst an das Produkt anschließen, dann an den PC.

Wird die Steuerung über ein USB-Kabel mit einem PC verbunden, wird im Windows-Dateiexplorer ein MTP-Gerät angelegt, das einen Datenträger enthält.

Es werden drei Dateien angezeigt, die Konfigurationsdatei (`cfg*.txt`), das *NanoJ-Programm* (`nanoj*.usr`) und die Firmware (`*.fw`).

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. `nanoj*.usr`
  3. `reset.txt`

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

**PowerShell-Scripts**

Im Download-Bereich der Produktwebseite auf nanotec.com finden Sie die Datei *mtp.ps1* mit Beispielen für die Windows-PowerShell-Konsolle. Damit können Sie:

- Alle gespeicherten Dateien auflisten:
  `.\mtp.ps1 -Action list -DeviceName N6-1-1-1`
- Eine Datei aus dem aktuellen Script-Ordner auf das Gerät kopieren:
  `.\mtp.ps1 -Action copy -DeviceName N6-1-1-1 -FileName cfg_Example.txt`
- Eine Datei löschen:
  `.\mtp.ps1 -Action delete -DeviceName N6-1-1-1 -FileName cfg_Example.txt`

**MTP in Linux**

Nanotec empfiehlt folgende: MTP-Tools. Sollten das oft in Linux-Distributionen inkudierten *GVfs* bereits installiert sein, vorher deinstallieren:

```
sudo apt remove gvfs
```

Anschließend *MTP-Tools* und *libmtp* installieren und neustarten:

```
sudo apt-get install mtp-tools
sudo apt-get install libmtp-runtime
reboot
```

Nach dem Anschließen des USB prüfen, ob das Gerät erkannt wird:

```
mtp-detect
```

Das Gerät antwortet darauf mit seinen Eigenschaften (Hersteller, Bezeichnung, Seriennummer etc.) Sie können nun:

- Alle gespeicherten Dateien mit ihrem ID auflisten:
  `mtp-files`
- Eine Datei aus dem aktuellen Script-Ordner auf das Gerät kopieren (der Name muss identisch sein):
  `mtp-sendfile cfg_Out.txt cfg_Out.txt`
- Eine Datei aus dem Gerät kopieren (File-ID wie von *mtp-files* gelistet):
  `mtp-getfile 1072922872 cfg_Or.txt`
- Eine Datei löschen:
  `mtp-delfile -f cfg_Example.txt`

## 4.1.3 Konfigurationsdatei

### 4.1.3.1 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.

#### 4.1.3.2 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` 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 durch einen Neustart wirksam werden zu lassen:

1. Speichern Sie die Datei, falls nicht schon geschehen. Der Motor hält an.
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.

**TIPP**

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.

#### 4.1.3.3 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 $00_h$ 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**

```
3202:00.01=1
```

**Bit zurücksetzen**

```
3202:00.01=0
```

**Bitweise OR**

```
3202:00|=0x01
```

**Bitweise AND**

```
3202:00&=0x01
```

**Beispiel**

Setzen des Objekts $6075_h$:00 (Nennstrom) auf den Wert "600" (mA):

```
6075:00=600
```

Setzen des Objekts $3202_h$:00 auf den Wert "1" ( *Closed Loop* aktivieren):

```
3202:00=1
```

oder nur Bit 0 setzen

```
3202:00.00=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 `nanoj*.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 `nanoj*.usr` um.
6. Kopieren Sie die Datei `nanoj*.usr` auf die Steuerung.
   Um das *NanoJ-Programm* beim nächsten Neustart der Steuerung zu starten, fügen Sie folgende Zeile in die Konfigurationsdatei ein:

   ```
   2300:00=1
   ```
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.

**TIPP**

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 `nanoj*.usr` haben.
- Falls das *NanoJ-Programm* gelöscht wurde, wird mit dem nächsten Start eine leere Datei namens `nanoj*.usr` angelegt.

**TIPP**

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>:\nanoj*.usr
```

Also zum Beispiel:

```
copy c:\test\main.usr n:\nanoj*.usr
```

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

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

Sie können Ihr *NanoJ-Programm* vor Auslesen/Kopieren schützen, indem Sie das Attribut *Versteckt* des FAT-Filesystems aktivieren.

## 4.2 Konfiguration über CANopen

### 4.2.1 Kommunikationseinstellungen

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

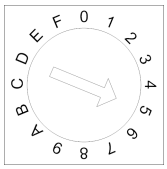
Ab Werk ist die Steuerung für die Node-ID 1 und eine Baudrate von 1 MBaud konfiguriert.

#### 4.2.1.1 Node-ID und Baudrate einstellen

Node-ID und Baudrate ergeben sich abhängig von der Position der *Drehschalter* S2 und S3 und ggf. noch von den Objekten 2005h CANopen Baudrate und 2009h CANopen NodeID.

#### Drehschalter

Die *N6* verfügt über zwei Hex-Codierschalter - ähnlich wie in der nachfolgenden Abbildung.




Mit der Zahlenkombination aus beiden Drehschaltern können Sie die Quelle für die CANopen Node-ID und die Baudrate einstellen.

Dabei gilt: Die Zahlenkombination setzt sich aus beiden Drehschaltern S2 und S3 zusammen, wobei S2 das höherwertigere Byte darstellt und S3 entsprechend das niederwertigere Byte.

**Beispiel**

Schalter S2 steht auf dem Wert "$0_h$", Schalter S3 auf dem Wert "$F_h$", daraus ergibt sich der Wert "$0F_h$"="$15_d$".

Schalter S2 steht auf dem Wert "$A_h$", Schalter S3 auf dem Wert "$1_h$", daraus ergibt sich der Wert "$A1_h$"="$161_d$".

### Generelles

- Steht der Drehschalter auf dem Wert "1", ist die Node-ID "1" und die Baudrate ist fest auf 1000 kBd eingestellt. Sollte es Probleme bei der Konfiguration geben, lässt sich somit immer eine Kommunikation mit der Steuerung aufbauen und eventuelle Fehler rückgängig machen.
- Die Änderungen in den Objekten $2005_h$ und $2009_h$ müssen gespeichert werden, indem der Wert "$65766173_h$" in das Objekt $1010_h$:$0A_h$ geschrieben wird.
- Die Änderungen werden erst übernommen, nachdem entweder
  - die Spannungsversorgung kurz getrennt wird oder
  - die Steuerung wird neugestartet via $2800_h$ oder
  - die CANopen-Nachricht "RESET COMM" (NMT) an den Motor verschickt wird.
- Die Ladesequenz für die Objekte $2005_h$ und $2009_h$ ist folgendermaßen (jeder nachfolgende Wert überschreibt den vorherigen):
  1. Der Standardwert wird geladen.
  2. Ein gespeicherter Wert wird geladen, falls einer vorhanden ist.
  3. Die Konfigurationsdatei wird angewendet, falls die Steuerung eine hat.
  4. Die Einstellungen des oder der Drehschalter werden übernommen.

### Node-ID und Baudrate

Die nachfolgende Tabelle zeigt die Möglichkeiten, die sich für die Drehschalter ergeben.

| Zahlenkombination der Drehschalter | | Node-ID | Baudrate |
|---|---|---|---|
| dec | hex | | |
| 0 | 0 | Object $2009_h$ | 1MBd fest |
| 1-127 | 1-7F | Zahl der Drehschalter | 1MBd fest |
| 128 | 80 | Object $2009_h$ | Objekt $2005_h$ |
| 129-255 | 81-FF | (Zahl der Drehschalter)-128 | Objekt $2005_h$ |

Den Wert des $2005_h$ können Sie aus nachfolgender Tabelle entnehmen.

| Wert im $2005_h$ | | Baudrate in kBd |
|---|---|---|
| dec | hex | |
| 120 | 78 | 100 |
| 129 | 81 | 10 |
| 130 | 82 | 20 |
| 131 | 83 | 50 |
| 132 | 84 | 125 |
| 133 | 85 | 250 |
| 134 | 86 | 500 |
| 135 | 87 | 1000 |

## 4.2.2 Kommunikation aufbauen

### 4.2.2.1 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 Terminierungswiderstand Terminierungswiderstand zwischen CAN_H und CAN_L vorhanden ist.
2. Versorgen Sie die Steuerung mit Spannung (+UB Logic).
3. Ändern Sie ggf. die Konfigurationswerte, siehe Konfiguration CANopenCANopen.
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`.

## 4.3 Motordaten einstellen

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

**HINWEIS**

Die Steuerung bietet keinen Schutz vor Überlast oder Übertemperatur für den Motor. Führen Sie bei kritischen Applikationen einen Dauertest in der realen Umgebung und sorgen Sie ggf. für ausreichende Kühlung des Motors.

- 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° = 50 Polpaare, 0,9° = 100 Polpaare (siehe Schrittwinkel im Motordatenblatt). Bei BLDC-Motoren ist die Polpaarzahl direkt im Motordatenblatt angegeben.
- Objekt $6075_h$:$00_h$ Nennstrom des Motors in mA (siehe Motordatenblatt
- Objekt $6073_h$:$00_h$: Maximaler Strom (entspricht bei einem Schrittmotor in der Regel dem Nennstrom, Bipolar) in Promille des eingestellten Nennstroms (siehe Motordatenblatt). Werkseinstellung: "1000", was 100% des Wertes in $6075_h$ entspricht.
- Objekt **3219**$_h$:$01_h$ 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 $3202_h$ (Motor Drive Submode Select): Bit 6 auf "0" für Schrittmotor setzen und über Bit 0 zwischen Open- und Closed-Loop wählen: Wer $0_h$ oder $1_h$. Siehe auch Kapitel Inbetriebnahme *Open Loop*.
    - Objekt **3219**$_h$:$03_h$ (Open loop idle state current): Effektivwert in Promille angegeben, auf den der Nennstrom reduziert werden soll, wenn die Stromabsenkung im *Open Loop* aktiviert wird.
  - BLDC-Motor:
    - Objekt $3202_h$ (Motor Drive Submode Select): Bit 6 für BLDC, Bit 0 für den empfohlenen *Closed Loop*: $00000041_h$
- Motor mit Encoder ohne Index: Sie müssen nach dem Auto-Setup die Encoder-Parameter einstellen, siehe Kapitel Konfigurieren der Sensoren.
- Motor mit Bremse: Objekt $3202_h$:$00_h$ (Motor Drive Submode Select): Für die Erstinbetriebnahme wird die Bremsensteuerung aktiviert. Abhängig von der konkreten Applikation kann diese Konfiguration wo nötig später wieder deaktiviert werden. Je nach Motortyp ist eines der folgenden Werte einzutragen:
  - Schrittmotor, Bremsensteuerung aktiviert: $00000004_h$
  - BLDC-Motor, Bremsensteuerung aktiviert, *Closed Loop*: $00000045_h$

**HINWEIS**

Aufgrund der Sinuskommutierung und des sinusförmigen Stromverlauf, kann der Strom einer Motorwicklung einen Wechselstromwert erreichen, der kurzfristig größer (um maximal √2-mal) ist, als der eingestellte Strom.

Bei besonders langsamen Drehzahlen oder im Stillstand mit voller Belastung kann deshalb eine der Wicklungen für längere Zeit überbestromt werden. Berücksichtigen Sie dies bei der Auslegung des Motors und wählen Sie ggf. einen Motor mit größerer Drehmoment-Reserve, falls die Anwendung das fordert.

## 4.4 Motor anschließen

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

**HINWEIS**

**Beschädigung der Elektronik durch falschen Anschluss des Motors!**

► Beachten Sie die PIN-Belegung im Kapitel *Anschlussbelegung* und dem Motordatenblatt.

- Motor anschließen:
  - an den Anschluss X1 – Motoranschluss
- Encoder/Hallsensoren anschließen:
  - an den Anschluss X6 – Inkrmentalencoder/Hall-Sensoren
  - bzw. an X5 – SSI-Encoder
- Bremse anschließen:
  - an den Anschluss X4 – Ein- und Ausgänge

  Im Kapitel Automatische Bremsensteuerung wird beschrieben, wie die automatische Bremsensteuerung aktiviert werden kann.

## 4.5 Auto-Setup

Um einige Parameter mit Bezug zum Motor und den angeschlossenen Sensoren (Encoder/Hall-Sensoren) zu ermitteln, müssen Sie ein Auto-Setup durchführen.

**TIPP**

Solange sich der an der Steuerung angeschlossene Motor oder die Sensoren für die Rückführung (Encoder/Hall-Sensoren) 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.

### 4.5.1 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) | ✓ |
| Wicklungswiderstand | ✓ |
| Wicklungsinduktivität | ✓ |
| Verkettungsfluss | ✓ |

**HINWEIS**

Bei Motoren, deren Wicklungen sehr unterschiedliche Induktivitäten ausweisen, ist das Ermitteln der Verkettungsflusses nicht möglich. Deshalb sind diese Motoren für den sensorlosen *Closed Loop*-Betrieb nicht geeignet.

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

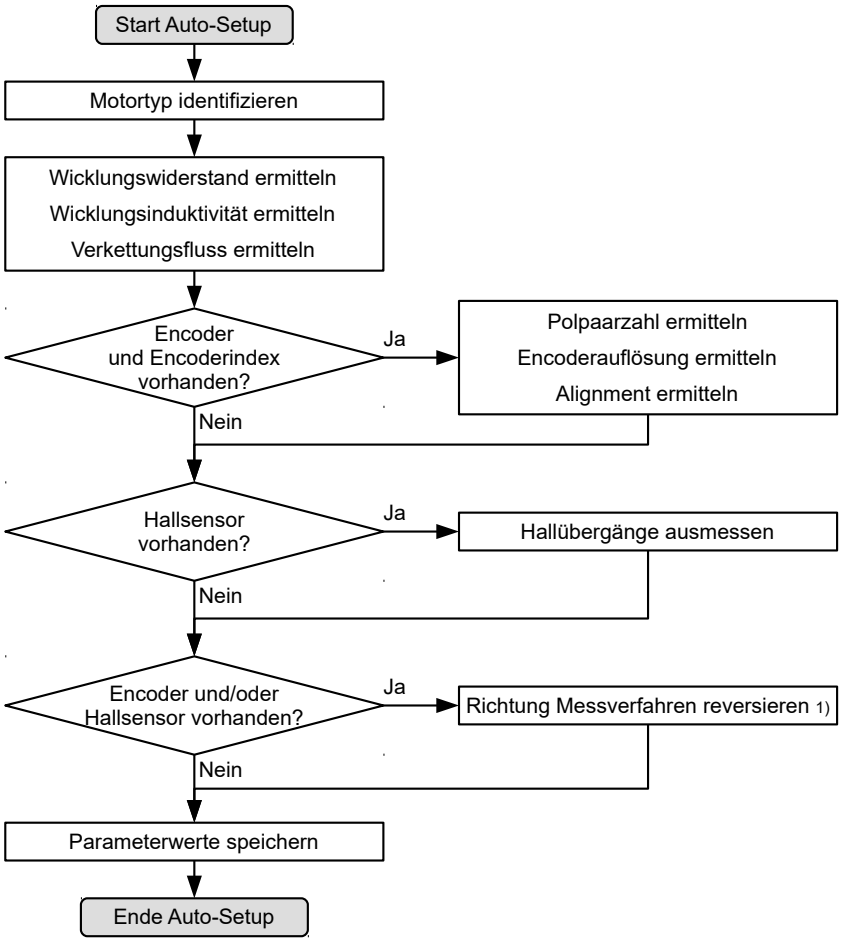
| Parameter | Motor ohne Hall-Sensor | Motor mit Hall-Sensor |
|---|---|---|
| Hallübergänge | - | ✓ |

### 4.5.2 Durchführung

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

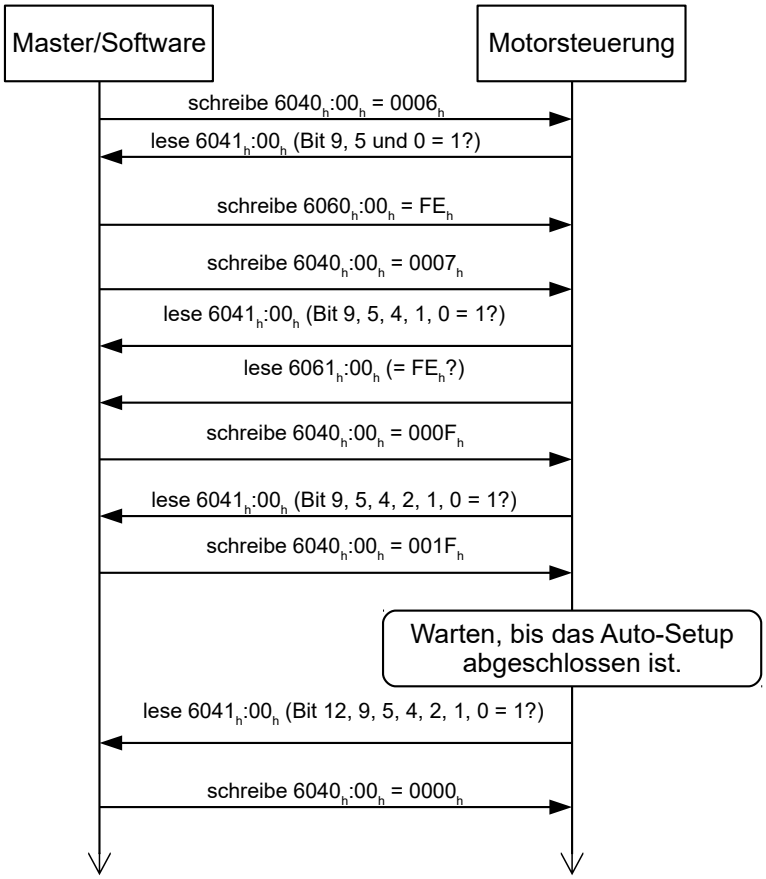
1. Zum Vorwählen des Betriebsmodus *Auto-Setup* tragen Sie in das Objekt $6060_h$:$00_h$ den Wert "-2" (="$FE_h$") ein.
   Die *Power state machine* muss nun in den Zustand *Operation enabled* versetzt werden, siehe CiA 402 Power State Machine.
2. Starten Sie das *Auto-Setup* mit Setzten von Bit 4 *OMS*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




### 4.5.3 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.

## 4.6 Konfigurieren der Sensoren

Die Parameter (Konfiguration, Alignment etc.) jeder Rückführung werden vom Auto-Setup ermittelt und in folgende Objekte gespeichert:

| Objekt | Rückführung | Beschreibung |
|---|---|---|
| $3380_h$ | Sensorless | enthält Mess- und Konfigurations-Werte für die sensorlose Regelung |
| $3390_h$ | Hall-Sensor (digital) | enthält Konfigurationswerte für die Hall-Sensoren |
| $33A0_h$ | Inkrementaler Encoder 1 | enthält Konfigurationswerte für den ersten inkrementalen Encoder |

| Objekt | Rückführung | Beschreibung |
|---|---|---|
| 33B0$_h$ | SSI-Encoder 1 | enthält Konfigurationswerte für den ersten SSI-Encoder |

**HINWEIS**

Die Ermittlung der Auflösung von Encodern ohne Index oder mit mehr als einem Index pro Motorumdrehung ist nicht möglich.

In diesem Fall müssen Sie die Parameter in die entsprechenden Objekte (siehe 3204$_h$, 60E6$_h$ und 60EB$_h$) eintragen und speichern (Kategorie *Sensor*, siehe Objekte speichern).

Für externe Sensoren, die nicht direkt auf der Motorwelle montiert sind, müssen Sie entsprechend der konstruktiven Gegebenheiten die Getriebeübersetzung (6091$_h$) und/oder die Vorschubkonstante (6092$_h$) einstellen und speichern (Kategorie *Applikation*).

**Beispiel**

Ein Encoder mit einer Auflösung von 2000 Inkrementen/mm wurde angeschlossen, der im Feld direkt am Prozess für eine hochgenaue Positionsmessung verwendet werden soll. Der konstruktive Aufbau wurde wie folgt realisiert:

| Motor | Getriebe | Prozess | Encoder |
|---|---|---|---|
| Rotatorisch | Rotatorisch \| Rotatorisch | Rotatorisch \| Translatorisch | Translatorisch |
| 1 | i=4 | Durchmesser 40 mm \| 125,6637... mm/U | 2000 Inkr./mm (62831,85 Inkr. pro Motorumdrehung) |

Sie müssen die Auflösung, Getriebeübersetzung und Vorschubkonstante wie folgt einstellen:

| Objekt | Wert |
|---|---|
| 60E6h Additional Position Encoder Resolution - Encoder Increments | 1256637 |
| 60EBh Additional Position Encoder Resolution - Motor Revolutions | 20 |
| 6091$_h$:01 (Motor Revolutions) | 4 |
| 6091$_h$:02 (Shaft Revolutions) | 1 |
| 6092$_h$:01 (Feed) | 2513274 Inkr. (entspricht 1256,637 mm ) |
| 6092$_h$:02 (Shaft Revolutions) | 10 |

Sie müssen noch die Einheit für die Position auf Millimeter oder eine andere Längeneinheit setzen, siehe Kapitel Benutzerdefinierte Einheiten.

Im Objekt 3203$_h$ können Sie einstellen, welche der vorhandenen Rückführungen die Steuerung für jeden Regler (Stromregler/Kommutierung, Geschwindigkeitsregler, Positionsregler) im *Closed Loop* oder die

Ermittlung der Ist-Position und Ist-Geschwindigkeit im *Open Loop* berücksichtigt. Siehe auch Kapitel Closed Loop und Zuordnung der Rückführungen zu den Regelkreisen.

**HINWEIS**

Der Wert "0" in einem Subindex des Objekts $60E6_h$ bedeutet, dass die jeweilige Rückführung nicht angeschlossen ist und nicht verwendet wird. So kann z. B. die Sensorless-Funktion ausgeschaltet werden, um Rechenzeit zu sparen. Dies kann hilfreich sein, wenn ein *NanoJ*-Programm die Rechenzeit benötigt.

Steht ein Wert ungleich "0" in einem Subindex, verwendet die Steuerung den entsprechenden Sensor. Im Fehlerfall (Signal nicht vorhanden, Konfiguration/Zustand ungültig etc.) wird im Statusword das Fehlerbit gesetzt und im Objekt 1003h ein Fehlercode hinterlegt.

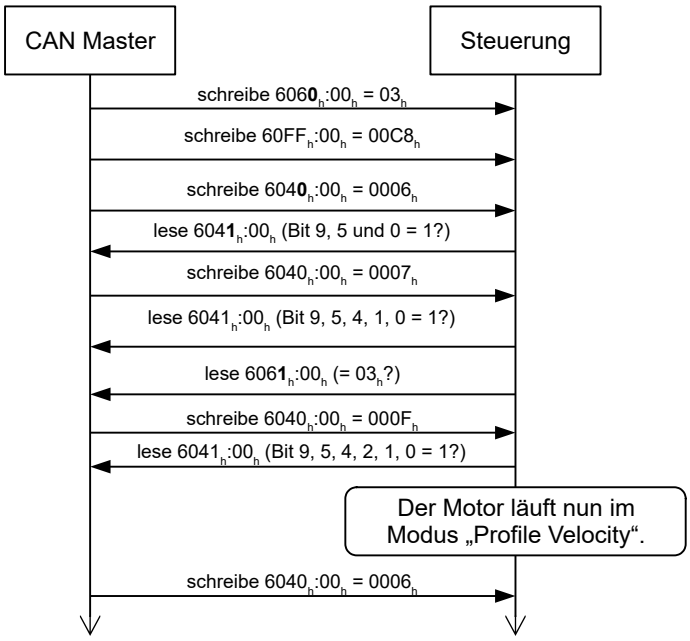
## 4.7 Testlauf

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

Die Werte werden von Ihrem *CANopen-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, indem Sie das Objekt $6060_h$ (Modes Of Operation) auf den Wert "3" setzen.
2. Schreiben Sie die gewünschte Drehzahl in $60FF_h$.
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 "6".

# 5 Generelle Konzepte

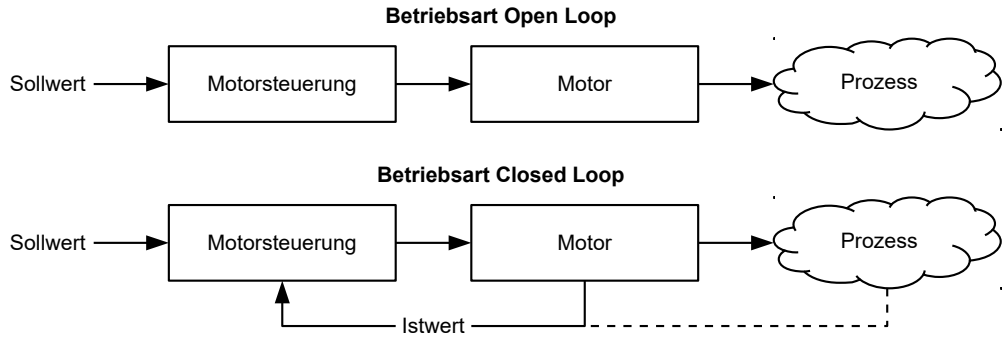
## 5.1 Betriebsarten

### 5.1.1 Allgemein

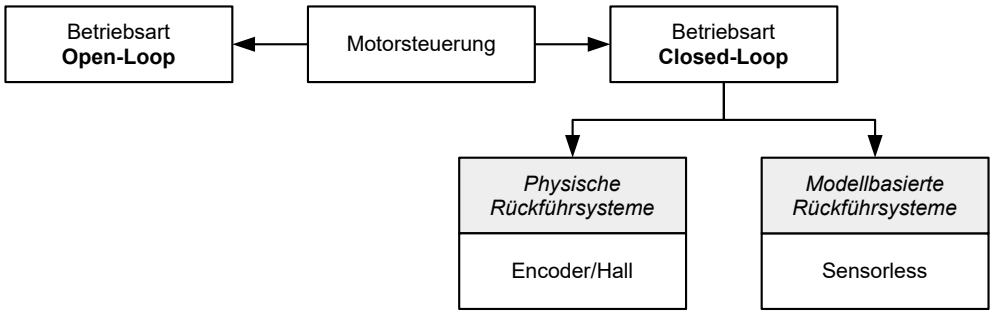
Die Betriebsart von Systemen ohne Rückführung wird als *Open Loop*, die mit Rückführung als *Closed Loop* bezeichnet. In der Betriebsart *Closed Loop* ist es zunächst unerheblich, ob die zurückgeführten Signale vom Motor selbst oder aus dem beeinflussten Prozess kommen.

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 zum Einsatz, die alle unter dem Überbegriff *Sensorless* bekannt sind. Beide Rückführsysteme können auch in Kombination eingesetzt werden, um die Qualität der Regelung weiter zu verbessern.




Nachfolgend werden alle möglichen Kombinationen von Betriebsarten und Rückführsystemen mit Bezug auf die Motortechnik 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 |

| Rückführung | Schrittmotor | BLDC-Motor |
|---|---|---|
| Encoder | ja | ja |
| Sensorless | ja | ja |

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

| Betriebsmodus | Betriebsart | |
|---|---|---|
| | Open Loop | Closed Loop |
| Profile Position | 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 Schrittverlust ausgeschlossen werden kann. Es wird deshalb davon abgeraten, diese Betriebsmodi in Verbindung mit der Betriebsart *Open Loop* zu verwenden.

## 5.1.2 Open Loop

### 5.1.2.1 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 und des Gesamtsystems entgegen. Unter bestimmten Betriebsbedingungen neigt diese Kombination zu Resonanzen, vergleichbar einem Feder-Masse-System.

### 5.1.2.2 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 $6075_h$:$00_h$ den Nennstrom des Motors in mA (siehe Motordatenblatt) eingeben.
- Im Objekt $6073_h$:$00_h$: den Maximalstrom (entspricht bei einem Schrittmotor in der Regel dem Nennstrom, Bipolar) in Promille des eingestellten Nennstroms eingeben (siehe Motordatenblatt). Werkseinstellung: "1000", was 100% des Wertes in $6075_h$ entspricht. Ein Wert größer "1000" wird intern auf "1000" limitiert.
- Im Objekt $3202_h$ (Motor Drive Submode Select) das Bit 0 (CL/OL) mit dem Wert "0" belegen.

Nanotec empfiehlt die Stromabsenkung bei Stillstand des Motors, um die Verlustleistung und Wärmeentwicklung zu reduzieren:

- Im Objekt $3219_h$:02 (Open Loop Idle State Detection Time) wird die Zeit in Millisekunden angegeben, die sich der Motor im Stillstand (der Sollwert wird geprüft) befinden muss, bis die Stromabsenkung aktiviert wird.
- Im Objekt $3219_h$:03 (Open Loop Idle State Current) 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.

#### 5.1.2.3 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 Objekt $6073_h$ bzw. $6075_h$. Eine 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.
- 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 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 Cyclic Synchronous Position**

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

  **Betriebsmodus Cyclic Synchronous Velocity**

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

  **Betriebsmodus Takt-Richtung**

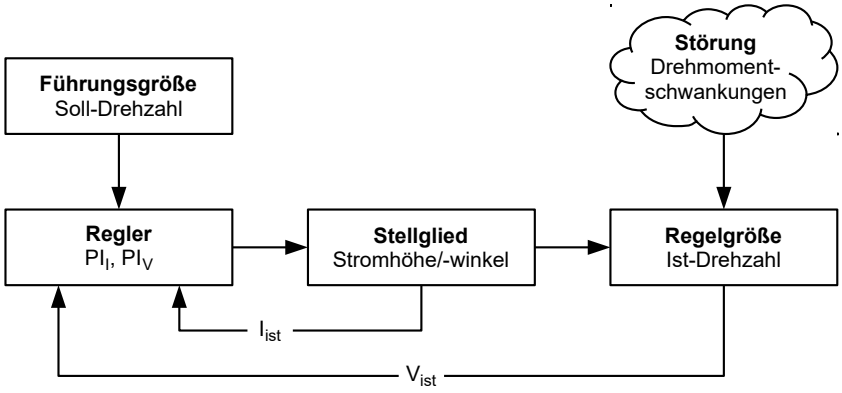
  Änderung der Schrittauflösung über die Objekte $2057_h$ (Clock Direction Multiplier) und $2058_h$ (Clock Direction Divider). Beschleunigungs-/Verzögerungsrampen durch Anpassen der Impulsfrequenz optimieren, um den Resonanzbereich möglichst schnell zu durchlaufen.

## 5.1.3 Closed Loop

### 5.1.3.1 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 "Sinuskommutierung 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 eines Sensors wird die Rotorlage erfasst und es werden in den Motorwicklungen sinusförmige Phasenströme erzeugt. Durch die Vektorregelung des Magnetfelds ist gewährleistet, dass das Statormagnetfeld 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 eine 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.

### 5.1.3.2 Reglerstruktur

Der Regler besteht aus drei kaskadierten PI-Reglern (proportional-integral): dem Stromregler (Kommutierung), dem Geschwindigkeitsregler und dem Positionsregler.

Der Stromregler ist in allen Betriebsmodi aktiv. Der Geschwindigkeitsregler ebenso, mit der einzigen Ausnahme der "Real Torque"-Modi (Drehmomentmodus ohne Drehzahl-Begrenzung, wenn das Bit 5 in $3202_h$ auf "1" steht).

Der Positionsregler ist in folgenden Betriebsmodi aktiv:

- Profile Position
- Homing
- Interpolated Position Mode
- Cyclic Synchronous Position
- Takt-Richtungs-Modus
- Profile Velocity/Cylic Synchronous Velocity, wenn das Bit 1 in $3202_h$ auf "1" steht

Jeder Regler besteht aus einem Proportional-Anteil mit dem *Verstärkungsfaktor* $K_p$ und einem Integral-Anteil mit der *Integrierzeit* $T_i$. Die Stellgröße (das Ausgangssignal des Reglers, das die Vorgabe für den nächsten Regler ist) wird jeweils durch die maximale Geschwindigkeit (Positionsregler), den maximalen Strom (Geschwindigkeitsregler) oder das maximale PWM-Signal (Stromregler) limitiert.

| Objekt | Name | Einheit | Beschreibung |
|---|---|---|---|
| $321A_h:01_h$ | Stromregler<br>Proportional Gain Kp for Iq | [mV/A] | Proportional-Anteil momentbildende Komponente |
| $321A_h:02_h$ | Stromregler<br>Integrator Time Ti for Iq | [µs] | Integrierzeit momentbildende Komponente |
| $321A_h:03_h$ | Stromregler<br>Proportional Gain Kp for Id | [mV/A] | Proportional-Anteil feldbildende Komponente |
| $321A_h:04_h$ | Stromregler<br>Integrator Time Ti for Id | [µs] | Integrierzeit feldbildende Komponente |
| $321B_h:01_h$ | Geschwindigkeitsregler<br>Proportional Gain Kp | [mA/Hz] | Proportional-Anteil |
| $321B_h:02_h$ | Geschwindigkeitsregler<br>Integrator Time Ti | [µs] | Integrierzeit |
| $321C_h:01_h$ | Positionsregler<br>Proportional Gain Kp | [Hz]<br>(Reglerabweichung in mech. Umdrehungen pro Sekunde) | Proportional-Anteil |
| $321C_h:02_h$ | Positionsregler<br>Integrator Time Ti | [µs] | Integrierzeit |

Der *Verstärkungsfaktor* $K_p$ hat einen direkten Einfluss auf die aktuelle Stellgröße: bei gleicher Abweichung ist die Stellgröße proportional zum Verstärkungsfaktor.

Jeder Regler besitzt auch einen Integral-Anteil, der durch die *Integrierzeit* ($T_i$) bestimmt wird. Je kleiner die Integrierzeit, desto schneller steigt die Stellgröße. Ist die Integrierzeit 0, wird der Integral-Anteil intern auf "0" gesetzt und der Regler hat nur den Proportional-Anteil

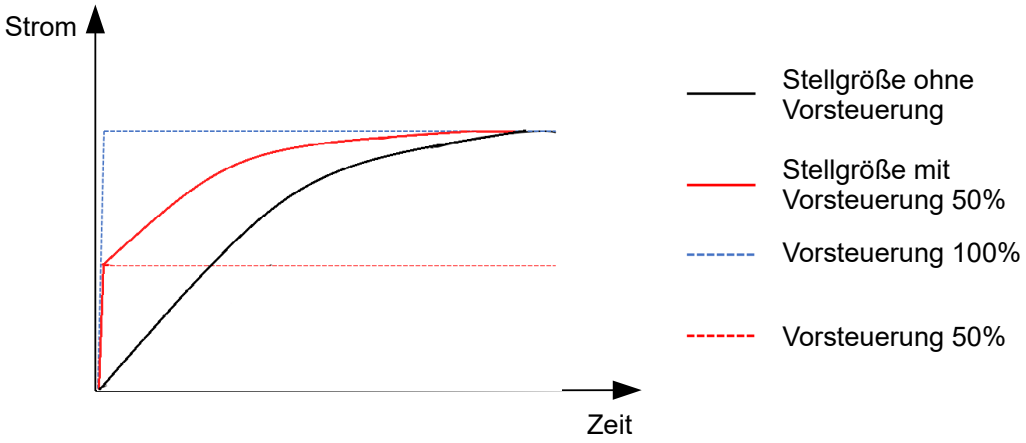
### 5.1.3.3 Vorsteuerung

Sie haben auch die Möglichkeit, eine *Geschwindigkeitsvorsteuerung*, eine *Beschleunigungsvorsteuerung* (die einem Drehmoment-/Stromwert entspricht) und eine *Spannungsvorsteuerung* einzustellen.

Sie können die *Vorsteuerung* verwenden, um eine bereits bekannte oder zu erwartende Stellgröße auf die Führungsgröße ("prädiktiv") aufzuschlagen. Sie können z. B. das Trägheitsmoment der Last kompensieren, indem Sie einen Beschleunigungs-Vorsteuerwert auf den Ausgang des Geschwindigkeitsreglers addieren.

Die Vorsteuerwerte werden zusätzlich in den Geschwindigkeits-/Stromregelkreis eingespeist bzw. auf den Spannungswert addiert und stehen sofort zur Verfügung. Dadurch kann eine dynamischere Regelung erzielt werden.

Die folgende Abbildung zeigt den Strom (der die Beschleunigung erzeugt) während der Beschleunigungsphase in Abhängigkeit von der *Beschleunigungsvorsteuerung*. Bei einem Vorsteuerwert von "50%" steht der Strom bereits zu Beginn der Beschleunigungsphase auf "50%", der Stromregler wird dadurch "entlastet".

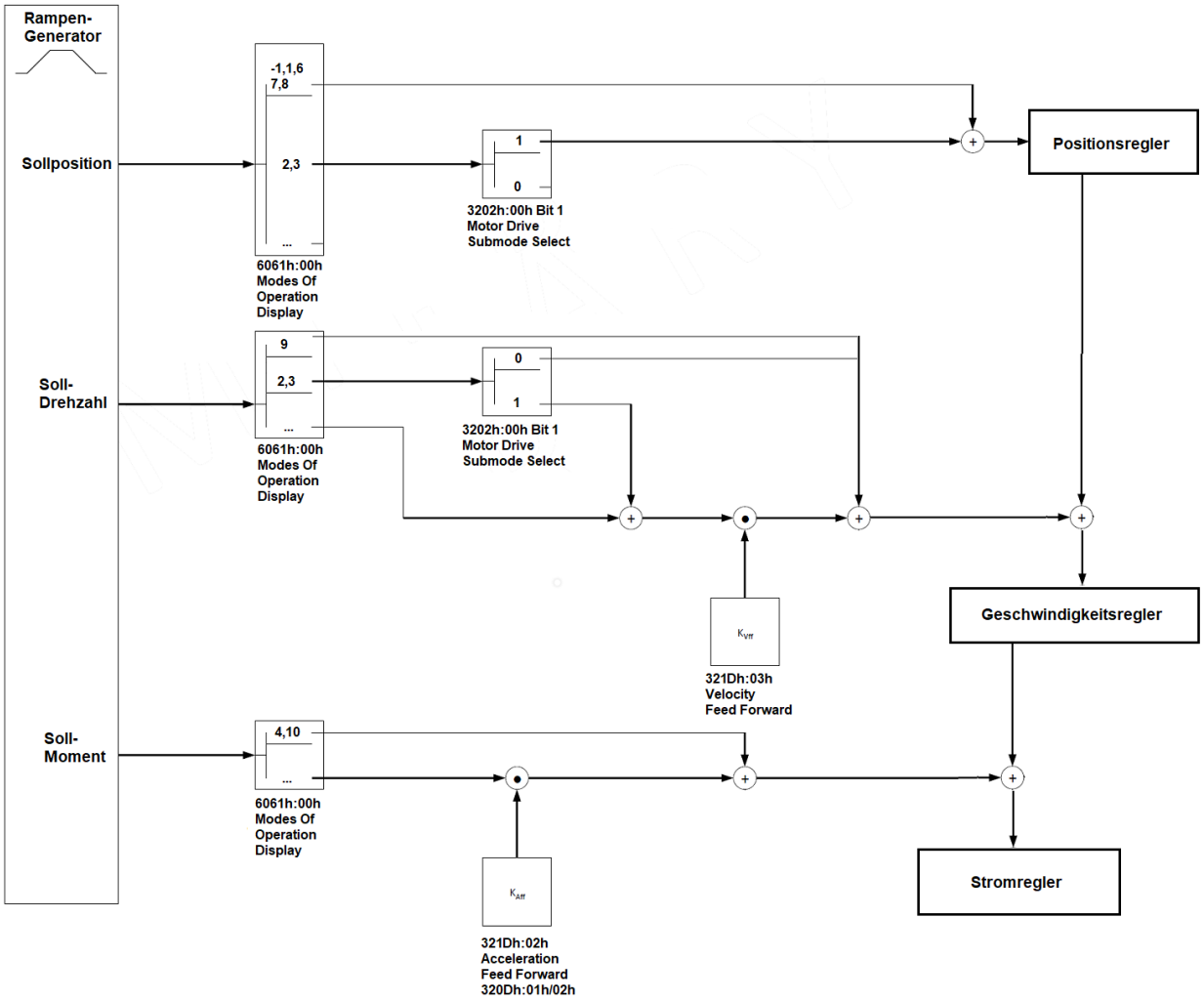



Der Faktor für die *Geschwindigkeitsvorsteuerung* wird im Objekt $321D_h$:$03_h$ in Promille des Ausgangs des Rampengenerators ($606B_h$) eingestellt und vor dem Geschwindigkeitsregler zum Ausgang des Positionsreglers addiert. Die *Geschwindigkeitsvorsteuerung* ist in allen Modi mit Positionsregelkreis aktiv:

- Profile Position
- Homing
- Interpolated Position Mode
- Cyclic Synchronous Position
- Takt-Richtungs-Modus
- Profile Velocity, wenn das Bit 1 in $3202_h$ auf "1" steht

Der Faktor für die *Beschleunigungsvorsteuerung* wird im Objekt $321D_h$:$02_h$ in Promille des Faktors von $320D_h$ eingestellt und mit dem Ausgang des Rampengenerators ($6074_h$) multipliziert. Der Wert wird vor dem Stromregler zum Ausgang des Geschwindigkeitsreglers addiert. Die *Beschleunigungsvorsteuerung* ist in allen Modi aktiv, mit der Ausnahme der Drehmomentmodi.

Die folgende Abbildung zeigt die Fälle, in denen die Vorsteuerung aktiv ist und die Position der Vorsteuerung innerhalb der Regler-Kaskade.




Die d- und q-Stromregler haben einen wechselseitigen Einfluss aufeinander. Um diese Kopplung zu neutralisieren, werden von der Steuerung Spannungen vorausberechnet und auf die vom Stromregler berechneten Spannungen aufgeschlagen. Mittels des Objekts $321D_h$:$01_h$ können Sie diese Entkopplung anpassen (Werkseinstellung 1000 ‰).

Anhand des im Autosetup ermittelten Werts für den ohmschen Widerstand, kann die nötige Spannung für einen gewünschten Strom vorausberechnet werden. Mit dem Objekt $321D_h$:$04_h$ können Sie die vorausberechnete Spannung anpassen (Werkseinstellung 0 ‰). Steht diese Spannung sofort zur Verfügung, kann der Ist-Strom dem Sollwert sehr schnell folgen und den Integralanteil des Stromreglers unterstützen. Bei Verwendung dieser *Spannungsvorsteuerung*, sollen Sie die Ti-Zeitwerte des Stromreglers im Objekt $321A_h$ entsprechend (deutlich) erhöhen.

### 5.1.3.4 Zuordnung der Rückführungen zu den Regelkreisen

Im Objekt $3203_h$ legen Sie fest, welche der vorhandenen Rückführungen die Steuerung für die einzelnen Regler (Stromregler/Kommutierung, Geschwindigkeit, Position) berücksichtigt. Sie können auch einen zweiten Sensor für die Kommutierung verwenden (siehe Kommutierungshilfe).

Jeder Subindex des Objekts enthält eine Bitmaske für die jeweilige Rückführung eines Sensors. Die Bits haben dabei folgende Bedeutung:

- Bit 0: wird das Bit auf "1" gesetzt, wird dieser Sensor für die Rückführung der Position verwendet.
- Bit 1: wird das Bit auf "1" gesetzt, wird dieser Sensor für die Rückführung der Geschwindigkeit verwendet.
- Bit 2: wird das Bit auf "1" gesetzt, wird dieser Sensor für die Rückführung der Kommutierung im Closed Loop verwendet.

Subindex $01_h$ entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

Welchen Sensor die Steuerung für die einzelnen Regler (Kommutierung, Geschwindigkeit, Position) berücksichtigt, ist implizit durch die Reihenfolge der Sensoren vorgegeben.

Das Aufsuchen beginnt immer mit Sensor 2 und setzt sich aufsteigend fort, bis alle vorhandenen Sensoren abgefragt wurden. Wird ein Sensor gefunden dessen Rückführung gesetzt ist, dann wird diese dem entsprechenden Regler zugeordnet und die Suche abgebrochen.

Das Objekt $3205_h$ zeigt, welcher Sensor für welchen Steuerkreis aktuell verwendet wird.

**Beispiel**

Die Steuerung hat zwei physikalische Schnittstellen. Angeschlossen wurden Hall-Sensoren und ein (nicht-absoluter) Inkremental-Encoder.

| Bit | Regler | Rückführung 1<br>Sensorless | Rückführung 2<br>Hall | Rückführung 3<br>Inkremental-Encoder |
|---|---|---|---|---|
| 0 | Position | 0 | 0 | 1 |
| 1 | Geschwindigkeit | 0 | 1 | 1[1] |
| 2 | Kommutierung | 0 | 1[2] | 1 |
| | **Index:Subindex** | $3203_h:01_h$ | $3203_h:02_h$ | $3203_h:03_h$ |

[1]Die Hall-Sensoren sollen für die Geschwindigkeitsregelung verwendet werden, der Encoder für die Positionierung und Kommutierung. Obwohl das Bit für die Geschwindigkeit auch bei der dritten Rückführung gesetzt wurde, wird dieses nicht berücksichtigt.

[2]Direkt nach dem Einschalten − und bis der Index des Encoders zum ersten Mal überfahren wird − soll die Kommutierung über die Hall-Sensoren erfolgen und den sofortigen *Closed Loop*-Betrieb ermöglichen.

**Kommutierungshilfe**

Einigen Sensoren fehlt anfangs das für die Kommutierung nötige Alignment (Versatz zwischen dem Index des Encoders und den Magneten des Rotors). Das heißt, dass die Rotorlage nicht allein anhand der Positionsinformation des Sensors bestimmt werden kann.

Als Hilfestellung können Sie einen zweiten Sensor als Kommutierungssensor einstellen (Bit 2 des entsprechenden Subindex in $3203_h$). So kann beispielsweise jeder (elektrisch) absolute Sensor mit Alignment (wie ein Hall-Sensor) eine Kommutierungshilfe bieten, z. B. für einen Inkremental-Encoder ohne Index bzw. mit noch fehlendem Alignment (Index-Signal seit einem Neustart noch nicht gesehen). Die Steuerung verwendet automatisch den besseren Sensor für die Kommutierung.

### 5.1.3.5 Inbetriebnahme

Vor dem Anwenden der Betriebsart *Closed Loop* sollte ein Auto-Setup durchgeführt werden. Der Betriebsmodus Auto-Setup ermittelt automatisch die notwendigen Parameter (z.B. Motorkenndaten, Rückführsysteme), die 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. Das Bit wird nach einem erfolgreich abgeschlossenen Auto-Setup automatisch gesetzt.

**Aktivierung**

Wird ein (elektrisch) absoluter Sensor (z. B. Hall-Sensor) für die Kommutierung verwendet, wird der *Closed Loop* automatisch bereits beim Einschalten aktiviert.

Wird ein Encoder für die Kommutierung verwendet, muss der Index des Encoders mindestens einmal nach dem Einschalten überfahren werden, bevor der *Closed Loop* aktiviert werden kann (solange erfolgt ein *Open Loop*-Betrieb).

Wenn kein Index vorhanden ist, oder dieser nicht verwendet werden soll, können Sie einen zweiten Sensor zur Kommutierung verwenden (siehe Zuordnung der Rückführungen zu den Regelkreisen).

Das Bit 15 im 6041h Statusword zeigt an, ob der *Closed Loop* aktiv ist oder nicht (wenn der Zustand der CiA 402 Power State Machine *Operation Enabled* ist).

### 5.1.3.6 Optimierungen

Im *Closed Loop* wird die gemessene Regelgröße (Istwert) permanent mit der Führungsgröße (Sollwert) verglichen. Bei Abweichungen zwischen diesen Größen regelt die Steuerung entsprechend den vorgegebenen Regelparametern nach.

Ziel der Optimierung der Regelparameter (das sogenannte *Tuning* des Reglers) ist ein möglichst ruhiger Motorlauf, eine hohe Genauigkeit und eine hohe Dynamik in der Reaktion der Steuerung auf Störungen. Alle Regelabweichungen sollen so schnell wie möglich eliminiert werden.

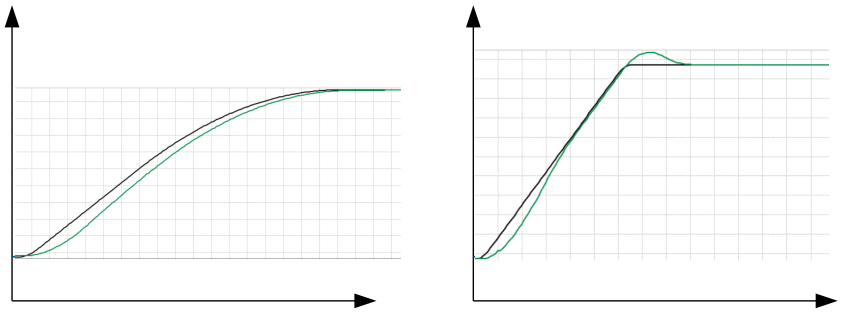
Es ist aufgrund der kaskadierten Reglerstruktur sinnvoll, mit der Optimierung des innersten Reglers (Stromreglers) zu beginnen, bevor der Geschwindigkeits- und ggf. der Positionsregler optimiert werden. Jeder der drei Regler besteht aus einem Proportional- und einem Integral-Anteil, die normalerweise in dieser Reihenfolge angepasst werden sollten.

Die Stromregler-Parameter können Sie anhand der Motorparameter wie folgt berechnen:

1. $K_p=L/(4*T_c)$, wo L die Wicklungsinduktivität in Henry und $T_c$ die Abtastzeit (siehe Zykluszeiten) in Sekunden.
2. $K_i=L/R$ , wo R der Wicklungswiderstand in Ohm.
3. Einheiten umrechnen und Werte in die entsprechenden Subindizes eintragen:
   $321A_h:01_h=K_p*10^3$
   $321A_h:02_h=K_1*10^6$

Folgende Abbildungen zeigen die Reaktion des Reglers auf eine Sollwert-Änderung.

Ist der Proportional-Anteil zu klein, bewegt sich der Istwert unterhalb des Sollwerts. Ein zu großer Proportional-Anteil führt dagegen zu einem "Überschwingen".

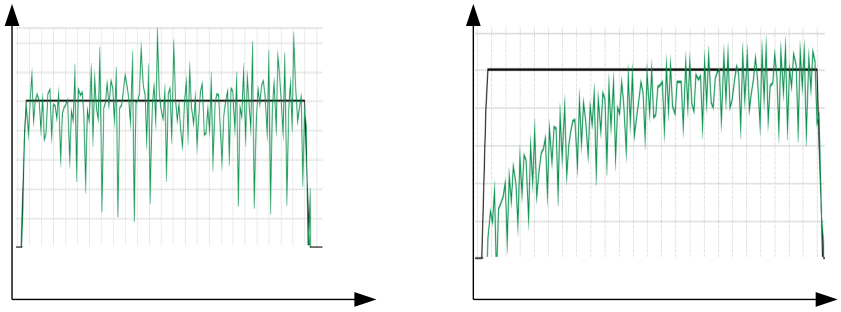


P-Anteil zu klein



P-Anteil zu groß

Ist die Integrierzeit zu klein, neigt das System zu Schwingungen zu. Ist die Integrierzeit zu groß, wird die Abweichung zu langsam ausgeregelt.



Ti zu klein | Ti zu groß

**VORSICHT!**

**Verletzungsgefahr durch unkontrollierte Motorbewegungen!**

Falsche Regelparameter können zu einem instabilen Regelverhalten führen. Es kann zu unvorhersehbaren Reaktionen kommen.

► Erhöhen Sie die Regelparameter langsam und schrittweise. Erhöhen Sie diese nicht weiter, wenn Sie starke Schwingungen/Oszillationen beobachten.

► Greifen Sie während des Betriebs nicht nach bewegten Teilen. Warten Sie nach dem Abschalten, bis alle Bewegungen beendet sind.

## 5.2 CiA 402 Power State Machine

### 5.2.1 Zustandsmaschine

#### 5.2.1.1 CiA 402

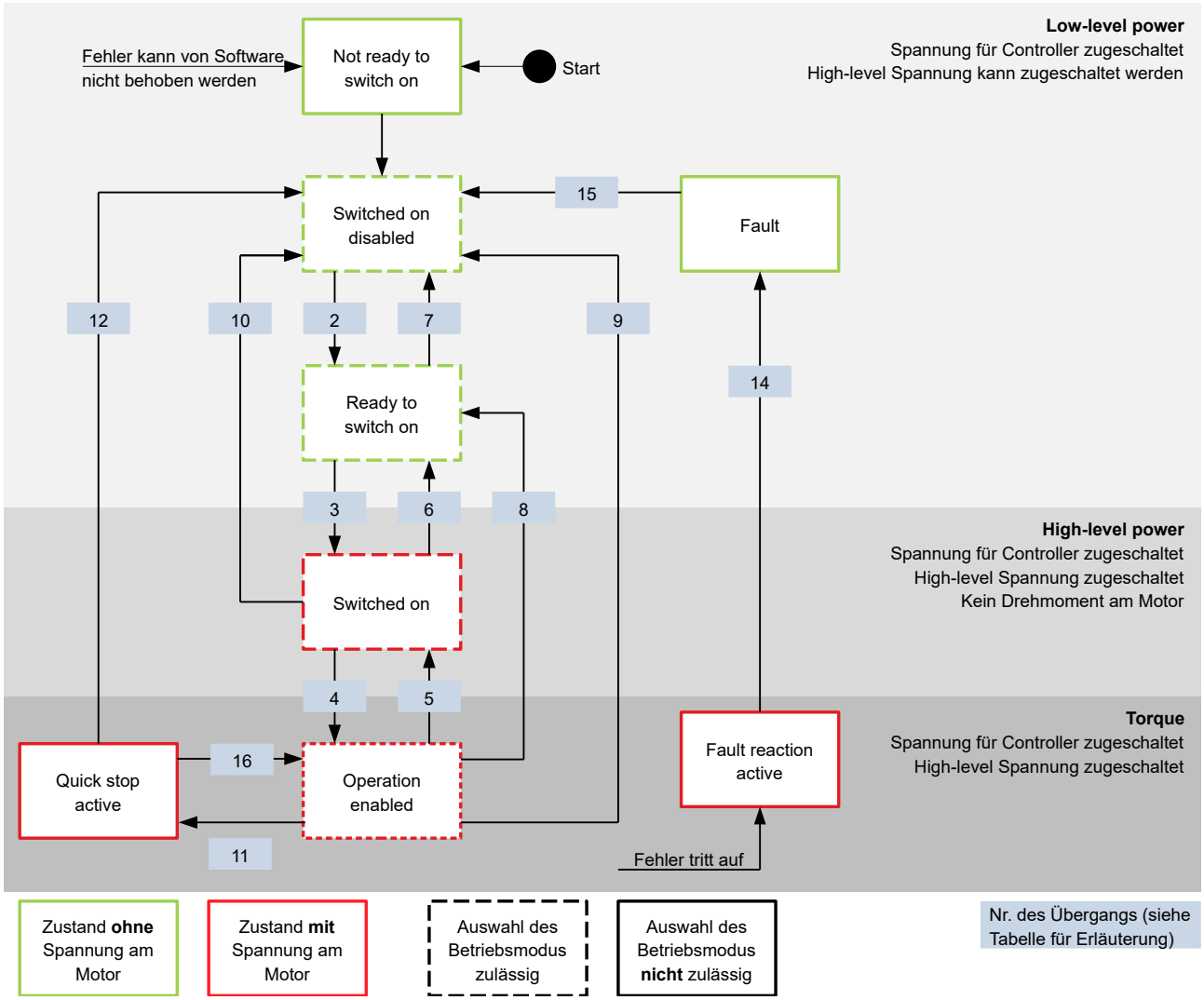
Um die Steuerung betriebsbereit zu schalten, ist es notwendig, eine Zustandsmaschine (*State Machine*) zu durchlaufen. Diese ist im *CANopen-Standard 402* definiert. Zustandsänderungen werden im Objekt $6040_h$ (Controlword) angefordert. Der tatsächliche Zustand der Zustandsmaschine lässt sich aus dem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

#### 5.2.1.2 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. Ausnahmen sind das Rücksetzen des Fehlers (Fault reset) und der Wechsel von *Quick Stop Active* nach *Operation Enabled*: 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 | 2, 6, 8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3 |
| Disable voltage | 0 | X | X | 0 | X | 7, 10, 9, 12 |
| Quick stop | 0 | X | 0 | 1 | X | 11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4 |

| Kommando | Bit im Objekt $6040_h$ | | | | | Übergang |
|---|---|---|---|---|---|---|
| | Bit 7 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
| Enable operation after Quick stop | 0 | 1 | | 1 | 1 | 16 |
| Fault reset | | X | X | X | X | 15 |

### 5.2.1.3 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*.

### 5.2.1.4 Betriebsmodus

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

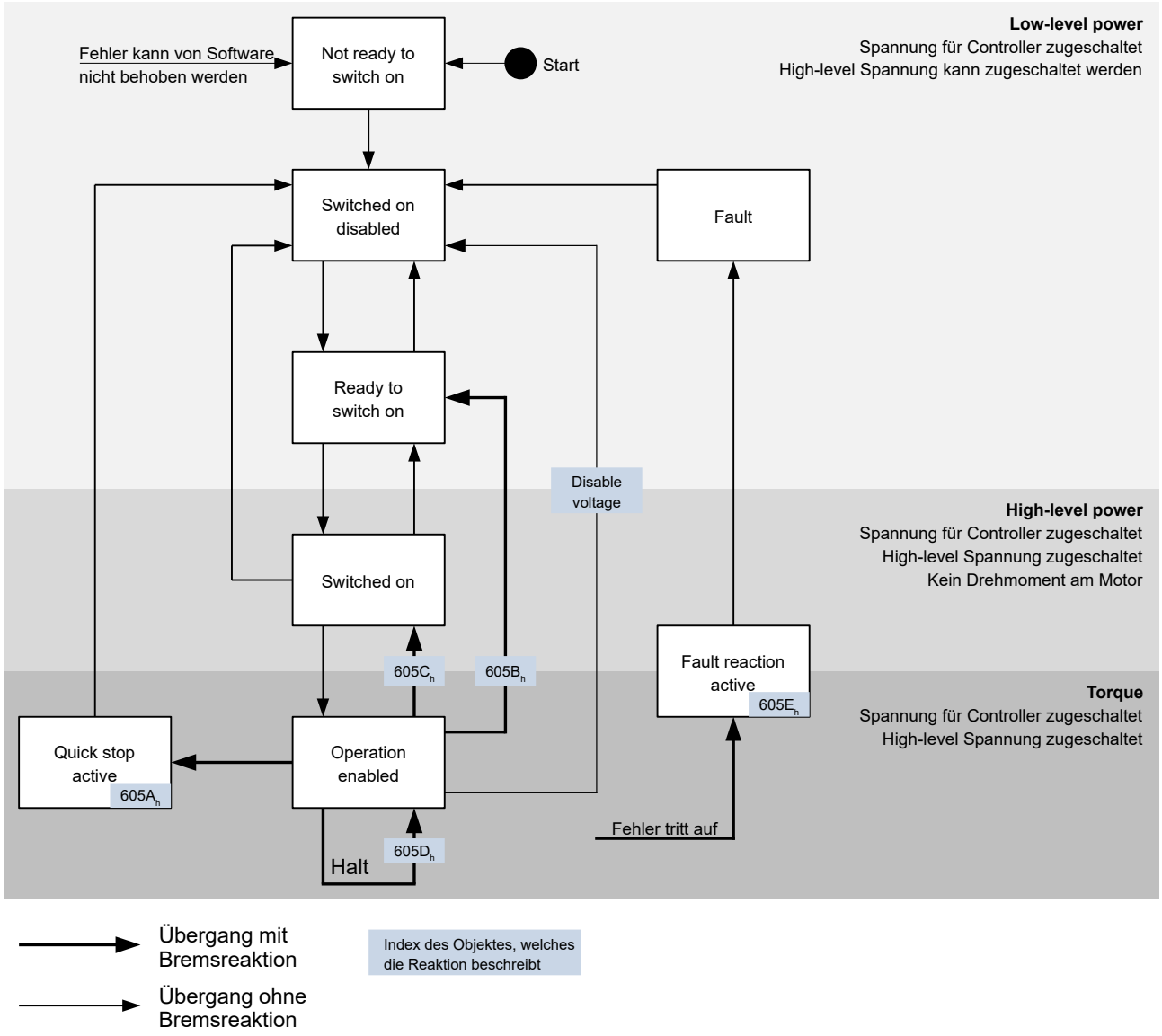
Die Einstellung oder Änderung des Betriebsmodus ist jederzeit möglich.

## 5.2.2 Verhalten beim Verlassen des Zustands *Operation enabled*

### 5.2.2.1 Bremsreaktionen

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

Die nachfolgende Grafik zeigt eine Übersicht der Bremsreaktionen.




### 5.2.2.2 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 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen |
| 1 | Abbremsen mit *slow down ramp* (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in *Switch on disabled* |
| 2 | Abbremsen mit *quick stop ramp* (6085$_h$) und anschließendem Zustandswechsel in *Switch on disabled* |
| 5 | Abbremsen mit *slow down ramp* (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in *Quick Stop Active*; Die Regelung schaltet nicht ab und der Motor bleibt |

| Wert in Objekt $605A_h$ | Beschreibung |
|---|---|
| | bestromt. Sie können wieder in den Zustand *Operation enabled* schalten. |
| 6 | Abbremsen mit *quick stop ramp* ($6085_h$) und anschließendem Zustandswechsel in *Quick Stop Active*; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand *Operation enabled* schalten. |

Der Zustand *Quick stop active* kann auch beim Betätigen eines Endschalters erreicht werden, siehe Begrenzung des Bewegungsbereichs.

### 5.2.2.3 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 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in *Ready to switch on* |
| 2 bis 32767 | Reserviert |

### 5.2.2.4 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 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in *Switched on* |
| 2 bis 32767 | Reserviert |

### 5.2.2.5 Halt

Das Bit ist gültig in folgenden Modi:

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

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

| Wert in Objekt $605D_h$ | Beschreibung |
|---|---|
| -32768 bis 0 | Reserviert |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) |
| 2 | Abbremsen mit *quick stop ramp* ($6085_h$) |
| 3 bis 32767 | Reserviert |

#### 5.2.2.6 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 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) |
| 2 | Abbremsen mit *quick stop ramp* ($6085_h$) |
| 3 bis 32767 | Reserviert |

Für jeden aufgetretenen Fehler wird im Objekt $1003_h$ ein genauerer Fehlercode hinterlegt.

#### 5.2.2.7 Schlepp-/Schlupffehler

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

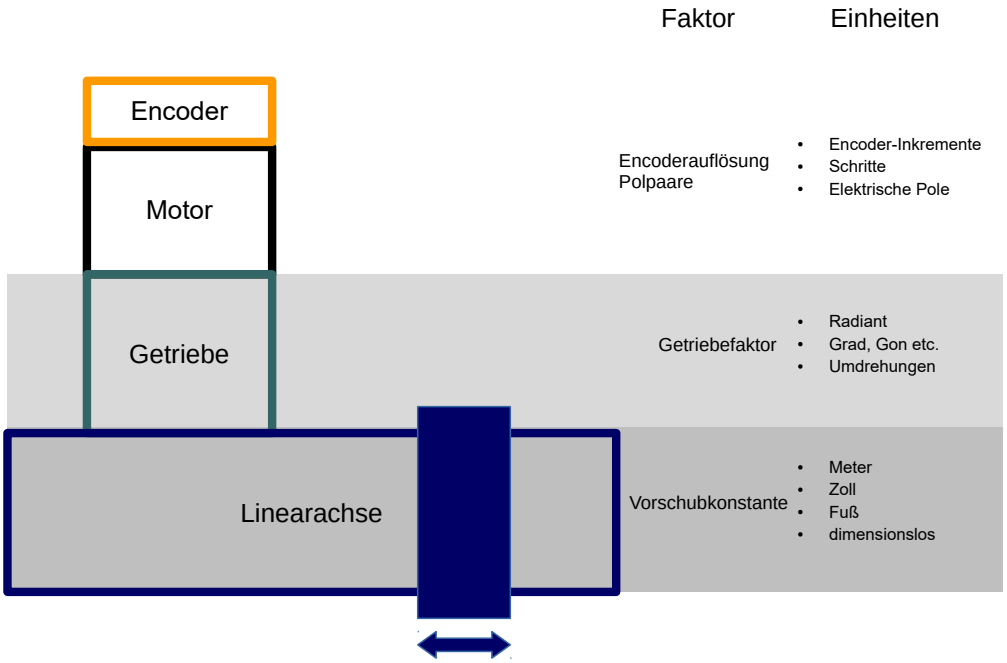
| Wert | Beschreibung |
|---|---|
| -32768 bis -2 | Reserviert |
| -1 | keine Reaktion |
| 0 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) |
| 2 | Abbremsen mit *quick stop ramp* ($6085_h$) |
| 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 (*SI*) als auch einige spezifische Einheiten unterstützt. Ebenfalls möglich ist die Angabe einer Zehnerpotenz als Faktor.

In der nachfolgenden Tabelle sind alle unterstützen 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 | *Meter* |
| inch | in | $C1_h$ | ja | ja | *Zoll* (=0,0254 m) |
| foot | ft | $C2_h$ | ja | ja | *Fuß* (=0,3048 m) |
| grade | g | $40_h$ | ja | nein | *Gon* (Winkeleinheit, 400 entsprechen 360°) |
| radian | rad | $10_h$ | ja | nein | *Radiant* |
| degree | ° | $41_h$ | ja | nein | *Grad* |
| arcminute | ' | $42_h$ | ja | nein | *Winkelminute* (60'=1°) |
| arcsecond | '' | $43_h$ | ja | nein | *Winkelsekunde* (60''=1') |
| mechanical revolution | | $B4_h$ | ja | nein | *Umdrehung* |

| Name | Einheitenzeichen | Wert | 6091$_h$ | 6092$_h$ | Beschreibung |
|---|---|---|---|---|---|
| encoder increment | | B5$_h$ | nein | nein | *Encoder-Inkremente.* Abhängig vom verwendeten Sensor (Encoder/Hall-Sensor) und Betriebsart. Im *Open Loop*- und *Sensorless*-Betrieb und wenn alle weiteren Sensoren in 3203$_h$ deaktiviert sind, entspricht die Anzahl der Polpaare (2030$_h$) multipliziert mit 65536 einer Motorumdrehung. |
| step | | AC$_h$ | nein | nein | *Schritte.* 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 revolution | | C0$_h$ | nein | nein | *Elektrische Umdrehung.* Bei einem Schrittmotor, der z.B. 50 Polpaare (2030$_h$) hat, entspricht die Einheit 1/50 einer Motor-Umdrehung. |
| dimensionless | | 00$_h$ | ja | ja | *dimensionslose Längeneinheit* |

In der nachfolgenden Tabelle sind alle unterstützen Einheiten für die Zeit und deren Werte für 60A9$_h$ (Geschwindigkeitseinheit) aufgelistet:

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

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 für die Positionsmessung verwendeten Encoders/Sensors berechnet sich aus den Encoder-Inkrementen (60E6$_h$ (Encoder Increments)) pro Motorumdrehungen (60EB$_h$ (Motor Revolutions)).

### 5.3.3 Getriebeübersetzung

Die Getriebeübersetzung berechnet sich aus Motorumdrehungen ($6091_h$:01) pro Achsenumdrehungen ($6091_h$:02).

### 5.3.4 Vorschubkonstante

Die Vorschubkonstante berechnet sich aus dem Vorschub in benutzerdefinierten Positionseinheiten ($6092_h$:01) pro Umdrehung der Abtriebsachse ($6092_h$:02).

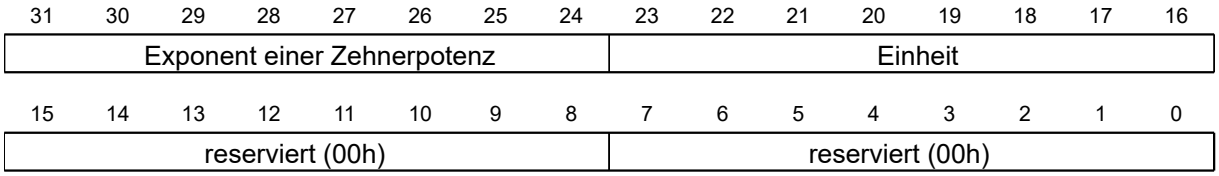
Die Vorschubkonstante ist zur Angabe der Spindelsteigung bei einer Linearachse nützlich und wird verwendet, wenn die Einheit auf Längenmaßen basiert oder wenn diese dimensionslos ist.

### 5.3.5 Berechnungsformeln für Benutzereinheiten

#### 5.3.5.1 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)




**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).

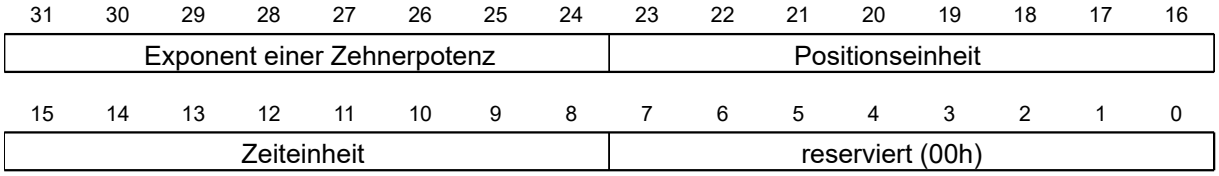
Wird die Vorschubkonstante entsprechend der Spindelsteigung einer Linearachse eingestellt, dreht der Motor so weit, dass ein Vorschub von 1 mm erreicht wird.

Im Kapitel Zuordnung der Rückführungen zu den Regelkreisen wird beschrieben, wie Sie festlegen, welcher Encoder/Sensor für die Positions-Regelung und -Messung verwendet werden soll.

#### 5.3.5.2 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)




**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.

Im Kapitel Zuordnung der Rückführungen zu den Regelkreisen wird beschrieben, wie Sie festlegen, welcher Encoder/Sensor für die Geschwindigkeits-Regelung und -Messung verwendet werden soll.

### 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$_h$).

$$n_{Geschwindigkeitseinheit} = \frac{6096_h:01}{6096_h:02}$$

## 5.3.5.3 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$_h$).

$$n_{Beschleunigungseinheit} = \frac{6097_h:01}{6097_h:02}$$

## 5.3.5.4 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$_h$).

$$n_{Ruckeinheit} = \frac{60A2_h:01}{60A2_h:02}$$

## 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 ausgelöst, so wird die Endschalterposition intern gespeichert, das Bit 7 (*Warning*) in $6041_h$ (*Statusword*) gesetzt und die CiA 402 Power State Machine in den Zustand *Quick Stop Active* versetz. Die in Objekt $3701_h$ hinterlegte Aktion wird dabei ausgeführt (siehe nachfolgende Tabelle).

| Wert in Objekt $3701_h$ | Beschreibung |
|---|---|
| -2 | keine Reaktion, verwerfen der Endschalterposition |
| -1 (Werkseinstellung) | keine Reaktion (um z. B. eine Referenzfahrt durchzuführen), außer Vermerken der Endschalterposition |
| 0 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen (Zustand *Switch on disabled*) |
| 1 | Abbremsen mit *slow down ramp* (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in *Switch on disabled* |
| 2 | Abbremsen mit *quick stop ramp* und anschließendem Zustandswechsel in *Switch on disabled* |
| 5 | Abbremsen mit *slow down ramp* (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in *Quick Stop Active*; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand *Operation enabled* schalten. |
| 6 | Abbremsen mit *quick stop ramp* und anschließendem Zustandswechsel in *Quick Stop Active*; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand *Operation enabled* schalten. |

Ein Weiterfahren hinter die Endschalterposition wird verhindert, sofern der Wert in $3701_h$ nicht "-1" oder "-2" ist. Es kann aber in jedem Fall in die entgegengesetzte Richtung gefahren werden.

Wird der Wert "-2" verwendet, wird das Bit 7 in $6041_h$ (Warning) bereits gelöscht, wenn die Endschalter nicht mehr auslösen. Ansonsten wird es erst gelöscht, wenn zusätzlich über die intern gemerkte Endschalterposition zurückgefahren wurde.

**HINWEIS**

Um ein automatisches Zurückwechseln bei der Verwendung der Optionen "5" oder "6" aus dem Zustand *Quick Stop Active* nach *Operation Enabled* zu vermeiden — das Quick-Stop-Bit (Bit 2) in $6040_h$ wird bei einem Auslösen der Endschalter nicht verwendet —, wird ein Wechsel des Quick-Stop-Bits von "0" auf "1" erwartet, um zurück in den Zustand *Operation Enabled* zu wechseln (605Ah Quick Stop Option Code muss auf "5" oder "6" stehen).

**Endschalterposition verwerfen**

| HINWEIS |
| --- |
| Ein Verwerfen der Endschalterpositionen ist nötig, wenn beide Endschalter gleichzeitig betätigt wurden oder der Bewegungsbereich dynamisch durch eine Verschiebung der Endschalter begrenzt wird. |

Um die beim Auslösen intern gespeicherten Endschalterpositionen zu löschen und die Endschalter freizugeben oder frei zu fahren, setzen Sie das Objekt $3701_h$ kurzzeitig auf "-2".

Falls bei Verwendung der Werte "5" oder "6" in $3701_h$, der Zustand der *State Machine* *Quick Stop Active* ist und der Motor bestromt bleiben soll, gehen Sie wie folgt vor, um einen automatischen Wechsel in den Zustand *Switch On Disabled* zu vermeiden:

1. Schalten Sie durch eine steigende Flanke von Bit 2 (Quick Stop) in $6040_h$ zurück in den Zustand *Operation Enabled*, ohne aber eine Fahrt zu starten (Bit 4 in $6040_h$ auf 0 bzw. Zielgeschwindigkeit oder -Drehmoment auf "0" setzen).
2. Setzen Sie $3701_h$ auf "-2" .
3. Geben Sie die Endschalter wieder frei.
4. Setzen Sie $3701_h$ wieder auf "5" oder "6" zurück.

### 5.4.2 Software-Endschalter

Die Steuerung berücksichtigt Software-Endschalter ($607D_h$ (Software Position Limit)). Zielpositionen ($607A_h$) werden durch $607D_h$ limitiert, die absolute Zielposition darf nicht größer sein als die Grenzen in $607D_h$. 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 vom 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 | 62,5 µs (16 KHz) |
| Geschwindigkeitsregler | 250 µs (4 KHz) |
| Positionsregler | 1 ms |

# 6 Betriebsmodi

## 6.1 Profile Position

### 6.1.1 Übersicht

#### 6.1.1.1 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.

#### 6.1.1.2 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").

#### 6.1.1.3 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. 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.

#### 6.1.1.4 Statusword

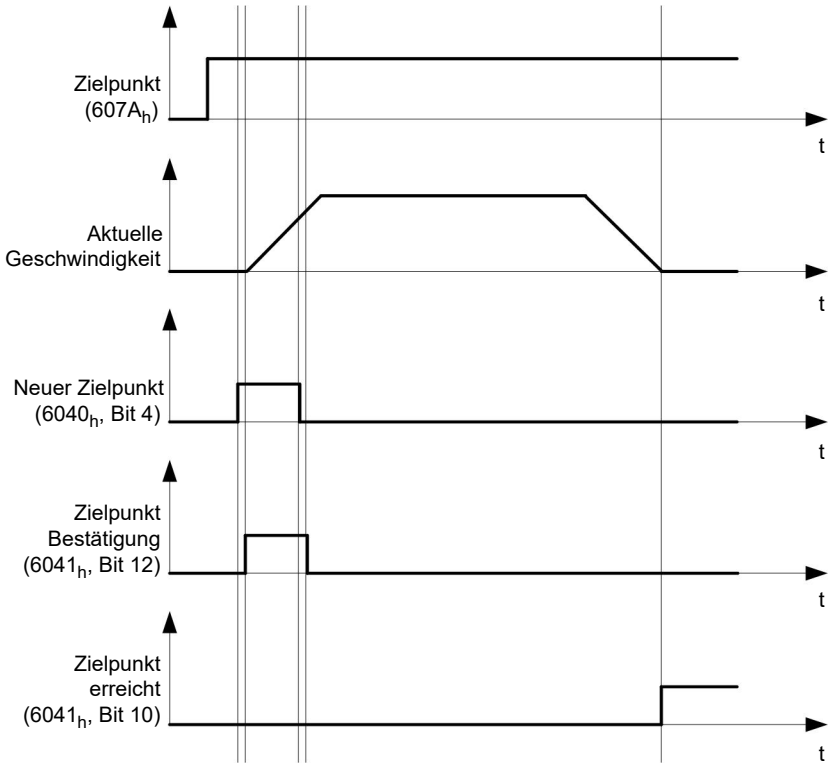
Folgende Bits im Objekt 6041$_h$ (Statusword) haben eine gesonderte Funktion:

- 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. Das Bit wird auch auf "1" gesetzt, wenn das Halt-Bit (Bit 8) in 6040$_h$ gesetzt wurde und sobald der Motor sich im Stillstand befindet.
- 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

#### 6.1.2.1 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.



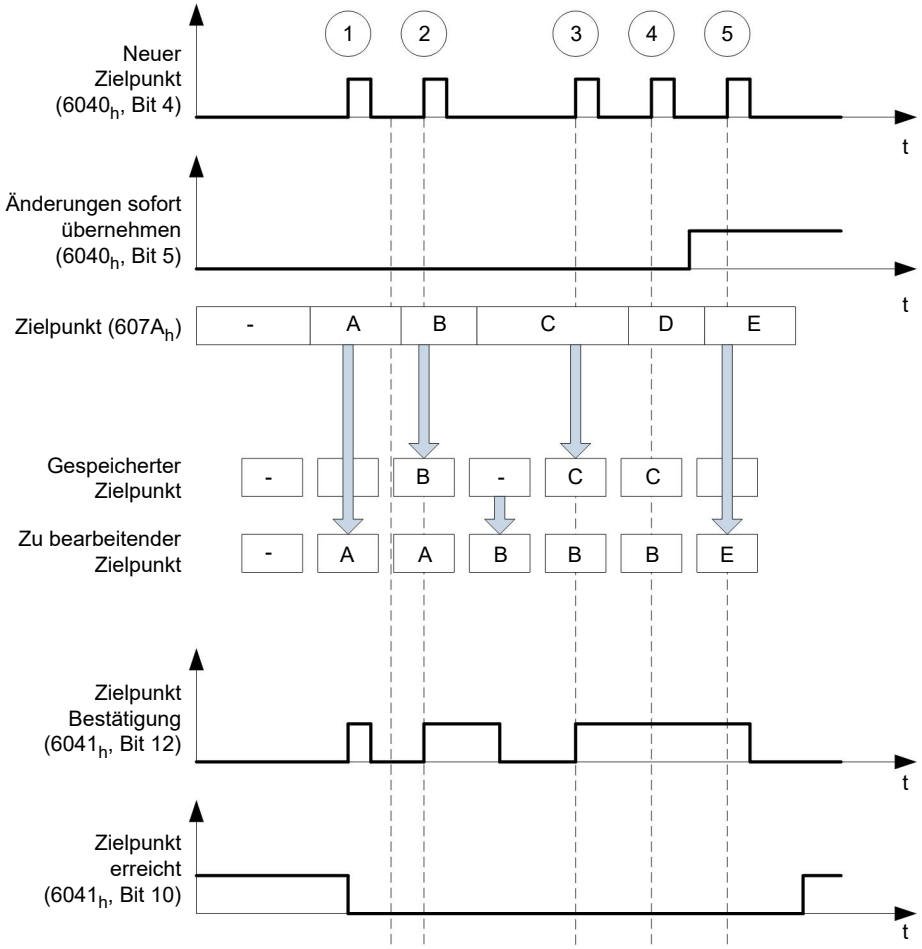

Die Steuerung kann das Bit 4 im Objekt 6040$_h$ (Controlword) auch selbstständig zurücksetzen. Das wird mit den Bits 4 und 5 des Objektes 60F2$_h$ eingestellt.

### 6.1.2.2 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).

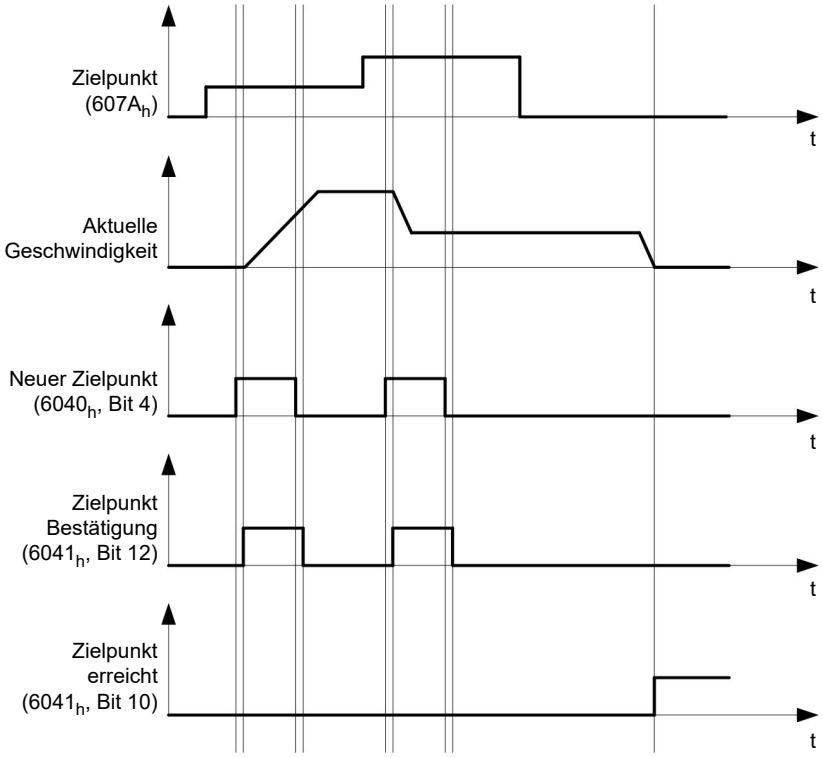
Sollte der Zwischenspeicher schon voll sein, wird ein neuer Zielpunkt ignoriert (Zeitpunkt 4). Wird Bit 5 im Objekt 6040$_h$ (Controlword, Bit: "Change Set-Point Immediately") gesetzt, arbeitet die Steuerung ohne den Zwischenspeicher, neue Fahrbefehle werden direkt umgesetzt (Zeitpunkt 5).

#### Zeitpunkte



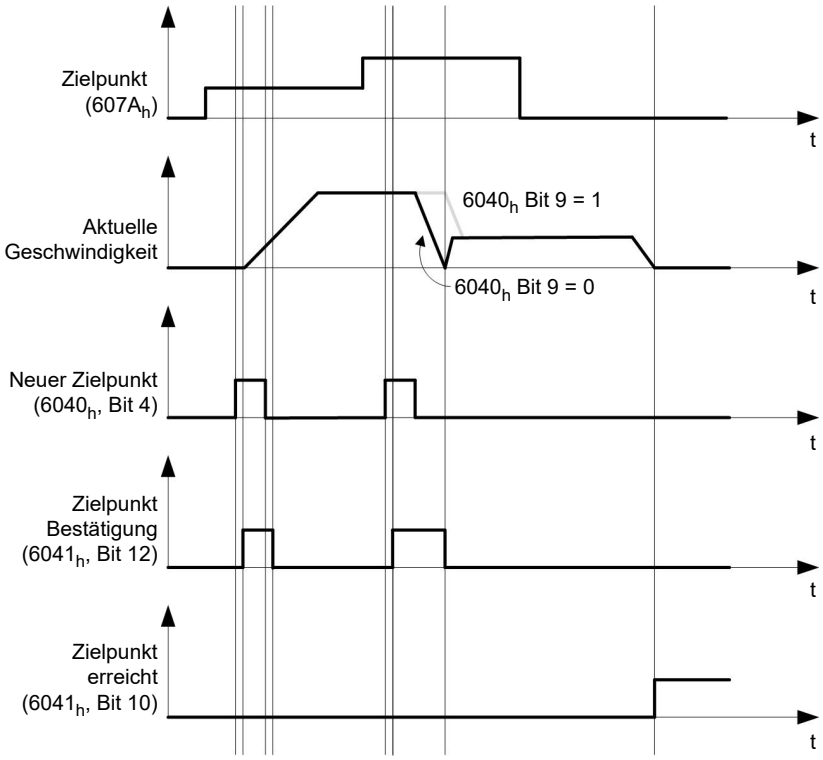

#### Übergangsprozedur für zweite Zielposition

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.



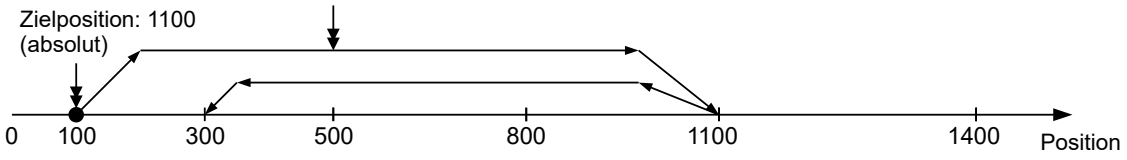

## Mögliche Kombinationen von Fahrbefehlen

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

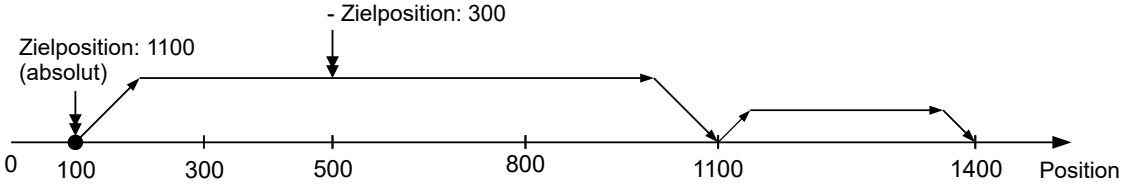
Die nachfolgenden Bilder setzen voraus:

- 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.

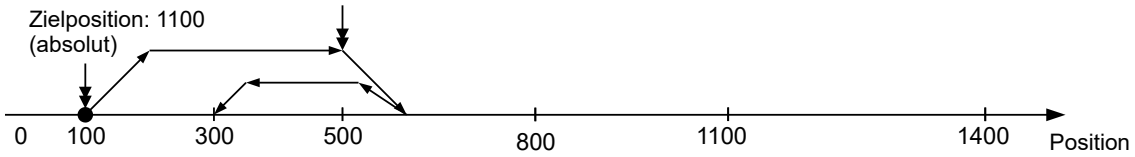
- Änderung im Zielpunkt übernehmen ($6040_h$:00 Bit 5 = 0)
- Positionierung absolut ($6040_h$:00 Bit 6 = 0)
- Zielposition: 300



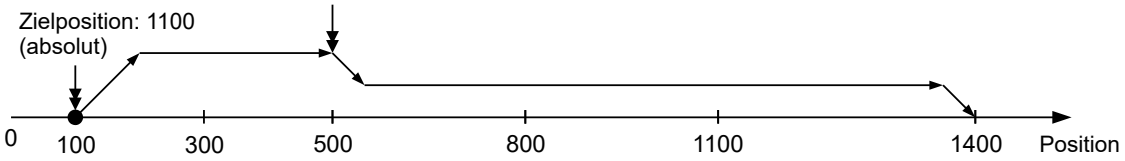

- Relativ zu der vorhergehenden Zielposition ($60F2_h$:00 = 0)
- Änderung im Zielpunkt übernehmen ($6040_h$:00 Bit 5 = 0)
- Positionierung relativ ($6040_h$:00 Bit 6 = 1)
- Zielposition: 300



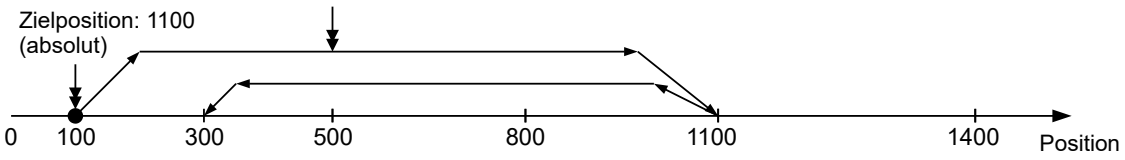

- Änderung sofort übernehmen ($6040_h$:00 Bit 5 = 1)
- Positionierung absolut ($6040_h$:00 Bit 6 = 0)
- Zielposition: 300



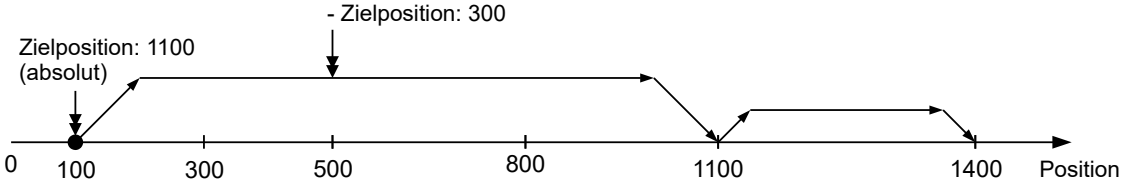

- Relativ zu der vorhergehenden Zielposition ($60F2_h$:00 = 0)
- Änderung sofort übernehmen ($6040_h$:00 Bit 5 = 1)
- Positionierung relativ ($6040_h$:00 Bit 6 = 1)
- Zielposition: 300



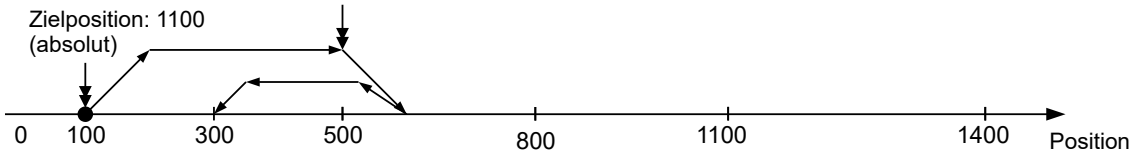
- Änderung im Zielpunkt übernehmen (6040$_h$:00 Bit 5 = 0)
- Positionierung absolut (6040$_h$:00 Bit 6 = 0)
- Zielposition: 300




- 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

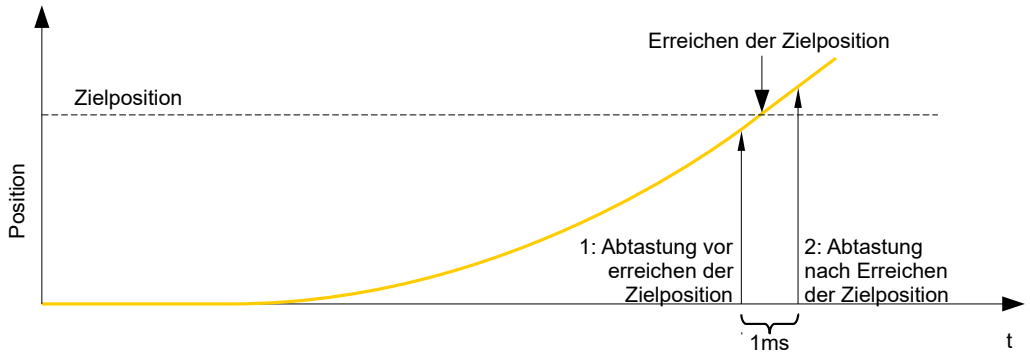



- Änderung sofort übernehmen (6040$_h$:00 Bit 5 = 1)
- Positionierung absolut (6040$_h$:00 Bit 6 = 0)
- Zielposition: 300




### 6.1.3 Genauigkeitsverlust bei Relativbewegungen

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

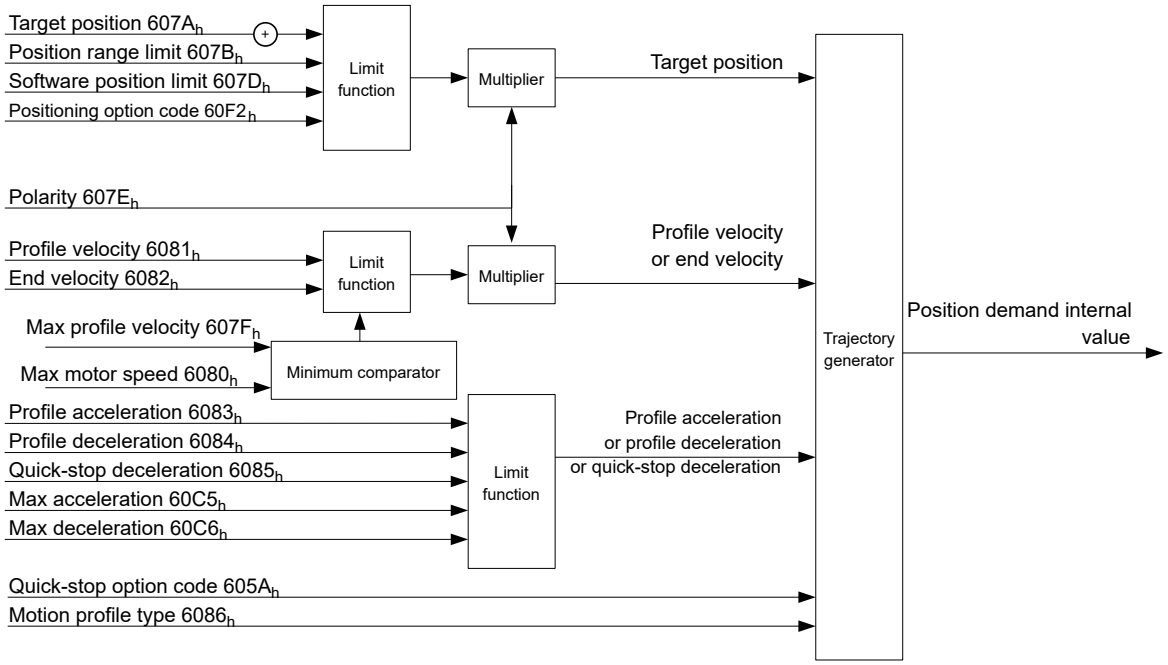
### 6.1.4.1 Objekteinträge

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

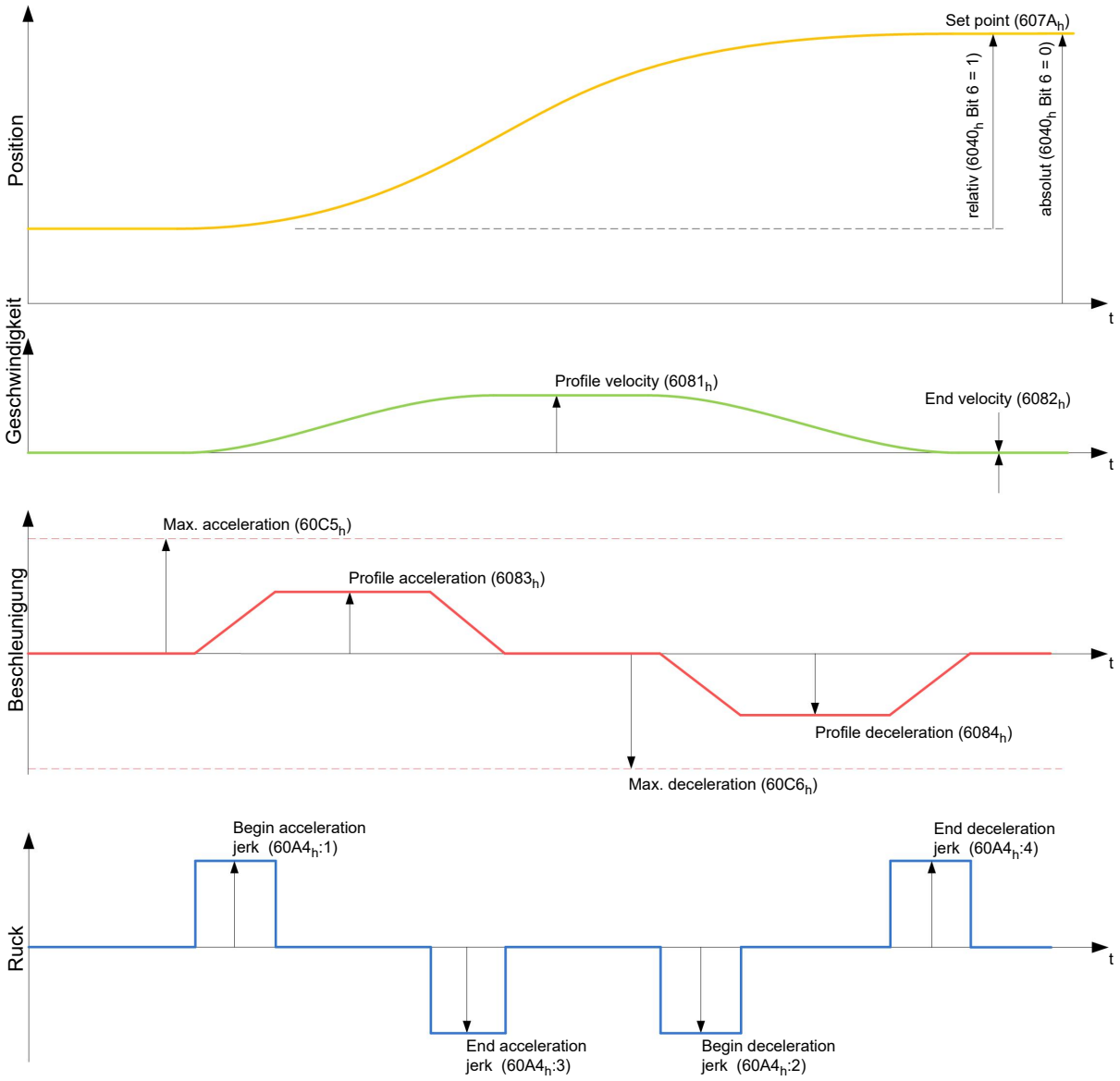
### 6.1.4.2 Objekte für die Positionierfahrt

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



### 6.1.4.3 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-begrenzter und nicht ruck-begrenzter Modus

### 6.1.5.1 Beschreibung

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

### 6.1.5.2 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.

### 6.1.5.3 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 Profile Velocity

### 6.2.1 Beschreibung

Dieser Modus betreibt den Motor im Geschwindigkeitsmodus mit erweiterten (ruck-limitierten) Rampen.

### 6.2.2 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").

### 6.2.3 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 Startrampe bis zur Zielgeschwindigkeit. Bei einem Übergang von "0" auf "1" bremst der Motor ab und bleibt stehen.

### 6.2.4 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 12: Dieses Bit zeigt, ob die Istgeschwindigkeit Null ist.
  Ist die Istgeschwindigkeit größer als der Wert in $606F_h$ (Velocity Threshold) für eine Zeit von $6070_h$ (Velocity Threshold Time), hat dieses Bit den Wert "0". Sonst bleibt das Bit auf "1".
- Bit 13 (Deviation Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schlupffehler größer als die eingestellten Grenzen ist (60F8h Max Slippage und 203Fh Max Slippage Time Out).

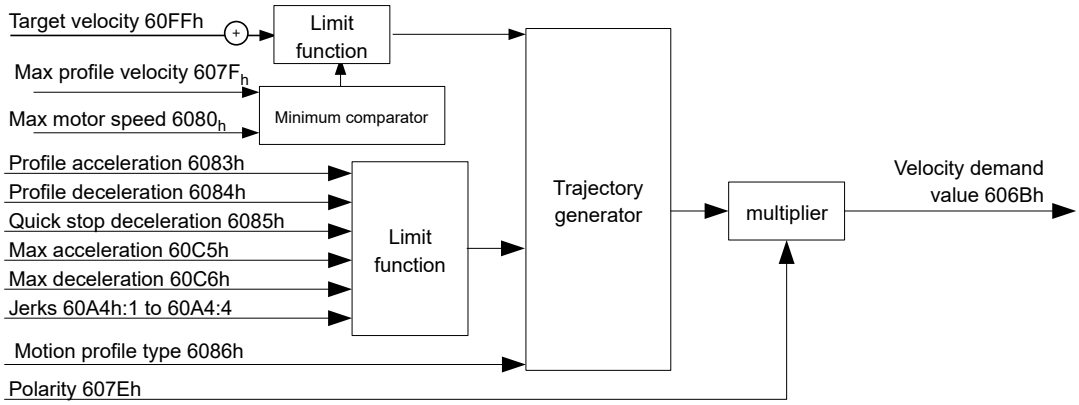
### 6.2.5 Objekteinträge

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.
- 6084$_h$ (Profile Deceleration):
  Setzt den Wert für die Bremsrampe.
- 6085$_h$ (Quick Stop Deceleration):
  Setzt den Wert für die Bremsrampe für die Schnellbremsung.
- 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.

### 6.2.5.1 Objekte im Profile Velocity Mode

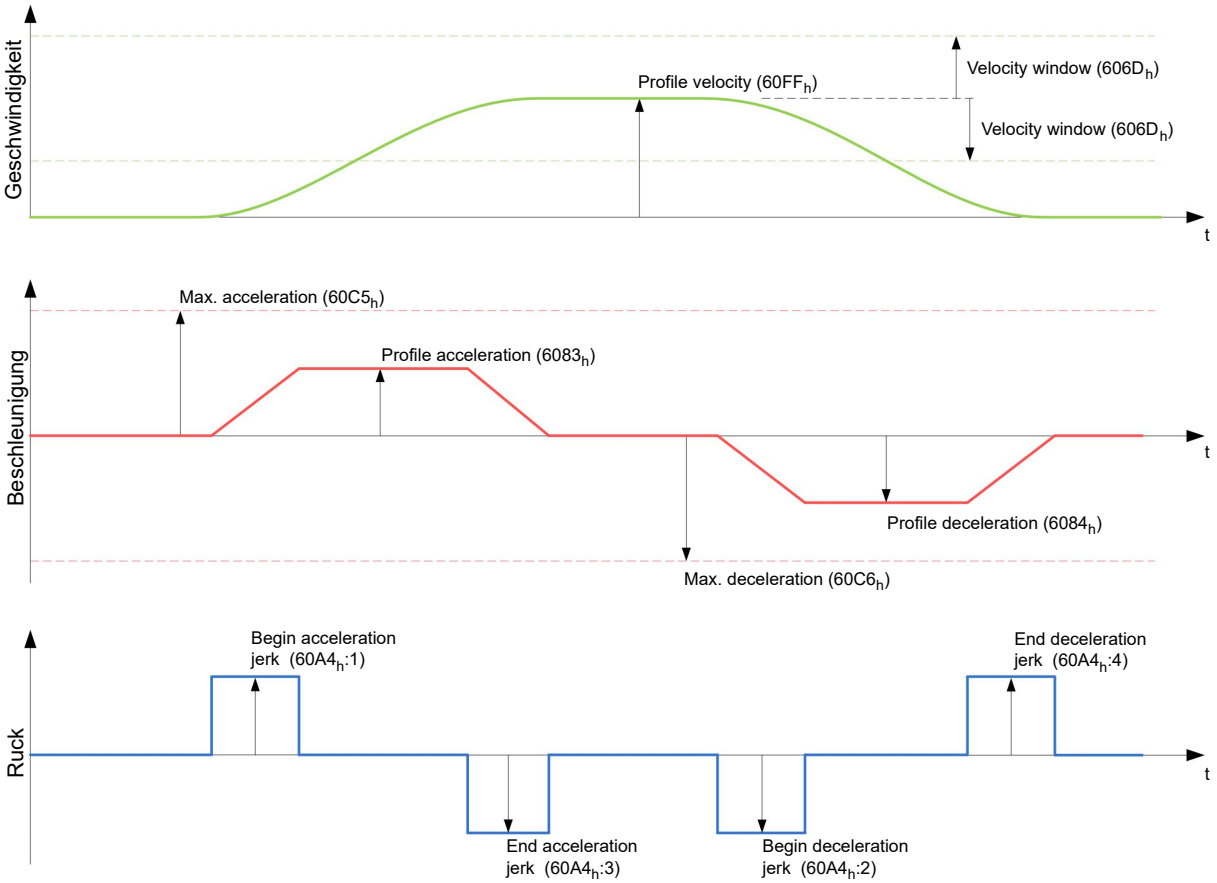



### 6.2.5.2 Aktivierung

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.

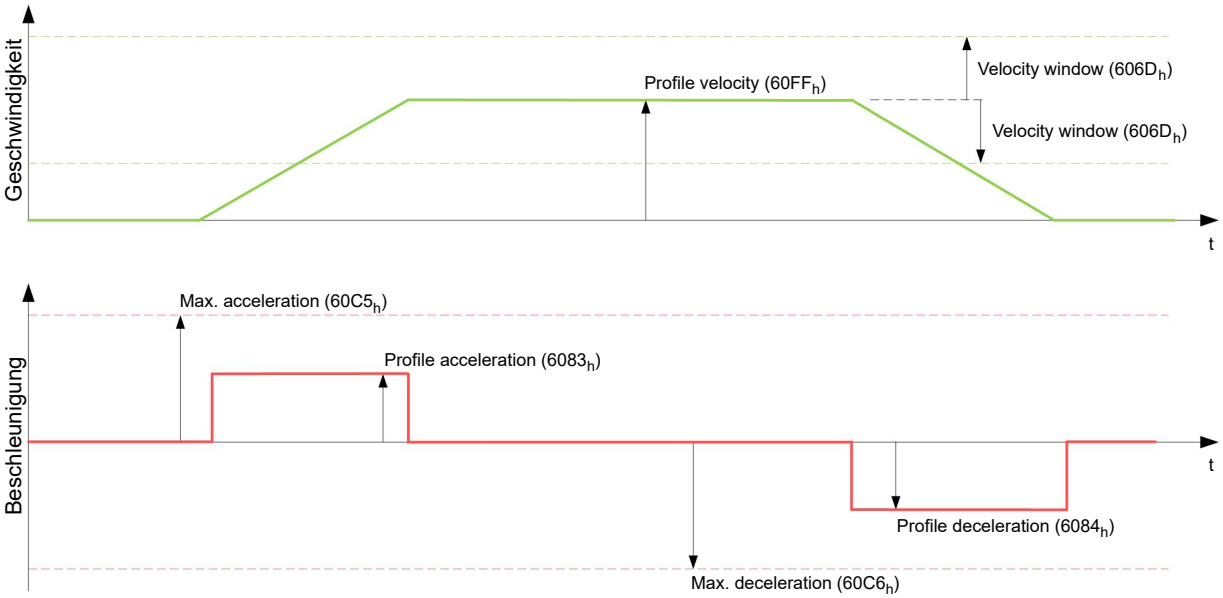
### 6.2.5.3 Limitierungen im ruck-limitierten Fall

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




### 6.2.5.4 Limitierungen im Trapez-Fall

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




## 6.3 Profile Torque

### 6.3.1 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.

### 6.3.2 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").

### 6.3.3 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. 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.

### 6.3.4 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 (6077h Torque Actual Value) eine vorgegebene Zeit (203Eh Torque Window Time Out) innerhalb eines Toleranzfensters (203Dh 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 bremst ab |
| 1 | 1 | Geschwindigkeit der Achse ist 0 |

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

### 6.3.5 Objekteinträge

Alle Werte der folgenden Einträge im Objektverzeichnis sind als Tausendstel des maximalen Drehmoments anzugeben, welches dem Nennstrom ($6075_h$) 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
- 6077 (Torque Actual Value):
  Aktueller Drehmomentwert
- $6087_h$ (Torque Slope):
  Max. Änderung des Drehmoments pro Sekunde

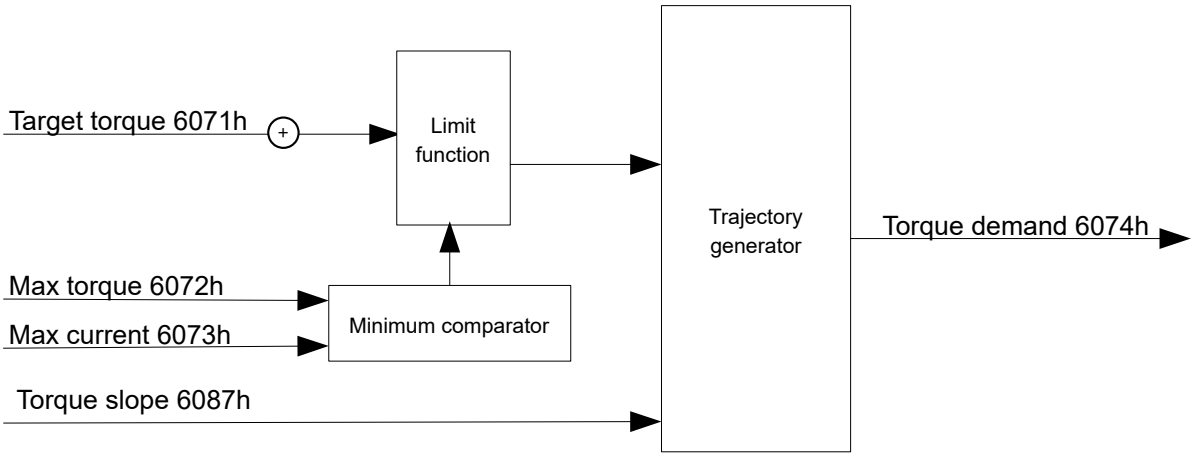
**HINWEIS**

Diese Werte sind nicht limitiert auf 100% des Nennstroms (6075$_h$). Drehmomentwerte höher als das Nenndrehmoment (generiert von dem Nennstrom) können erreicht werden, wenn die Maximaldauer (3219$_h$:01$_h$) des maximalen Stroms (6073$_h$) gesetzt wird (siehe I2t Motor-Überlastungsschutz).

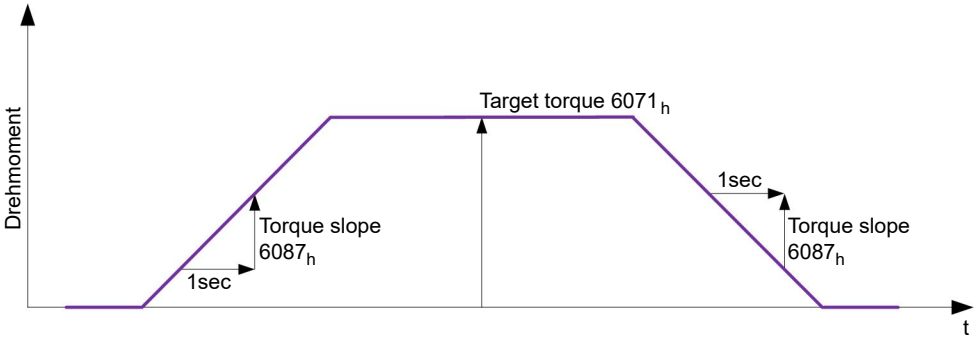
Die folgenden Objekte werden zudem für diesen Operationsmodus benötigt:

- 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 den Objekten 607F$_h$ und 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.

#### 6.3.5.1 Objekte des Rampengenerators




#### 6.3.5.2 Torque-Verlauf




## 6.4 Homing

### 6.4.1 Übersicht

#### 6.4.1.1 Beschreibung

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

### 6.4.1.2 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").

**TIPP**

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.

### 6.4.1.3 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.

### 6.4.1.4 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 ist seit dem letzten Neustart bereits durchgeführt worden, aber Ziel ist aktuell 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).

### 6.4.1.5 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

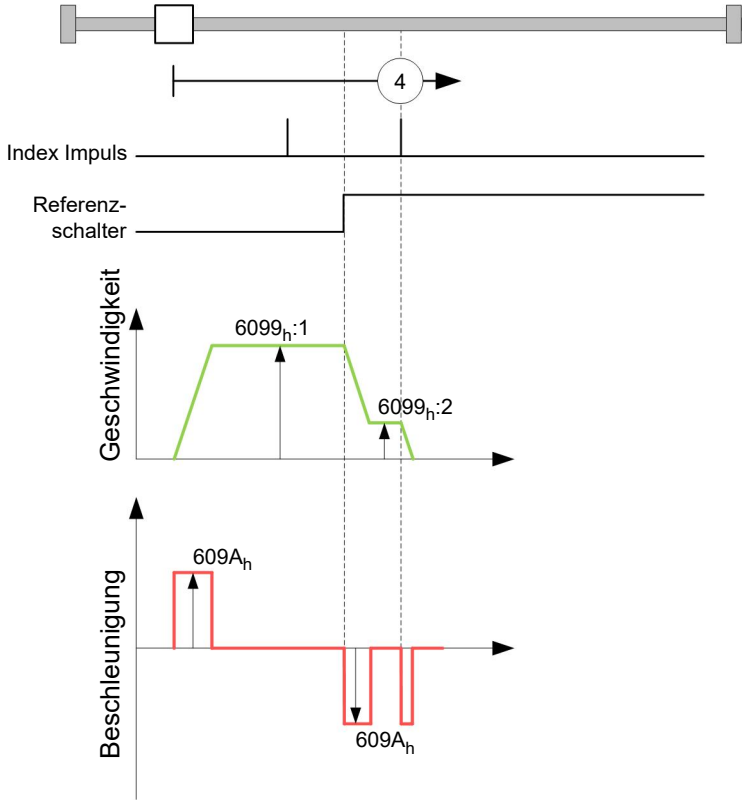
- $607C_h$ (Home Offset): Gibt die Differenz zwischen Null-Position der Steuerung und dem Referenzpunkt der Maschine in benutzerdefinierten Einheiten an.
- $6098_h$ (Homing Method):
  Methode, mit der referenziert werden soll (siehe "Referenzfahrt-Methode")
- $6099_h$:$01_h$ (Speed During Search For 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

- 3219$_h$:05$_h$ (Block Detection Threshold):
  Minimale Stromschwelle, durch deren Überschreiten, das Blockieren des Motors an einem Block erkannt werden soll.
- 3219$_h$:04$_h$ (Block Detection Time):
  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:




## 6.4.2 Referenzfahrt-Methode

### 6.4.2.1 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 Methodenzahl 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.

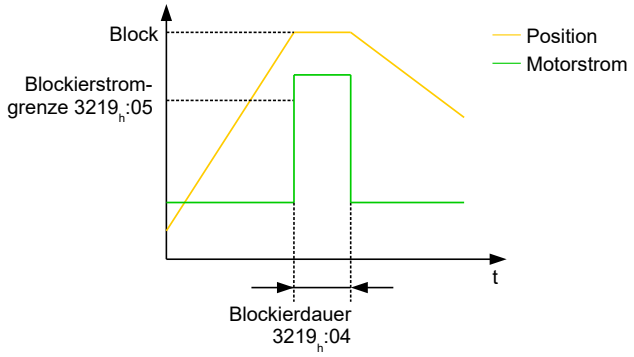
Bei Methoden, die Homing auf Block benutzen, gelten die gleichen Abbildungen wie für die Methoden mit Endschalter. Da sich außer den fehlenden Endschaltern nichts ändert, wurde auf neue Abbildungen verzichtet. Hier gilt für die Abbildungen, dass die Endschalter durch eine mechanische Blockierung ersetzt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

### 6.4.2.2 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 3219$_h$:05 wird die Stromhöhe definiert, ab der ein Fahren gegen den Block erkannt wird.
2. Blockierdauer: im Objekt 3219$_h$:04 wird die Dauer, während der Motor gegen den Block fährt, eingestellt.




### 6.4.2.3 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.

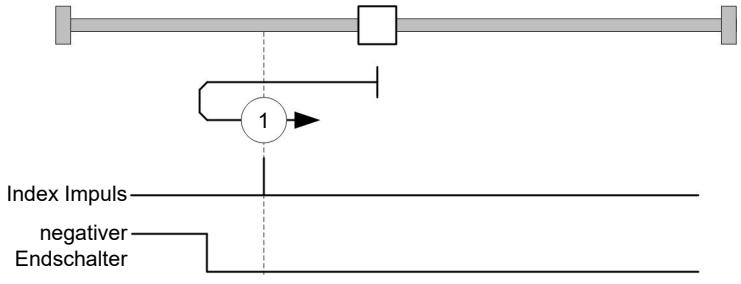
Folgende Methoden können für Homing auf Block benutzt werden:

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

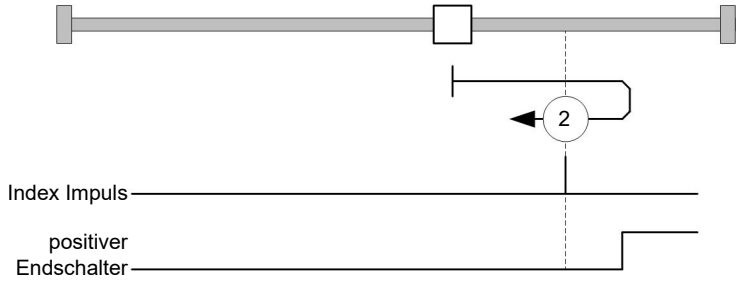
### 6.4.2.4 Methoden 1 und 2

Referenzieren auf Endschalter und Index-Impuls.

Methode 1 referenziert auf negativen Endschalter und Index-Impuls:



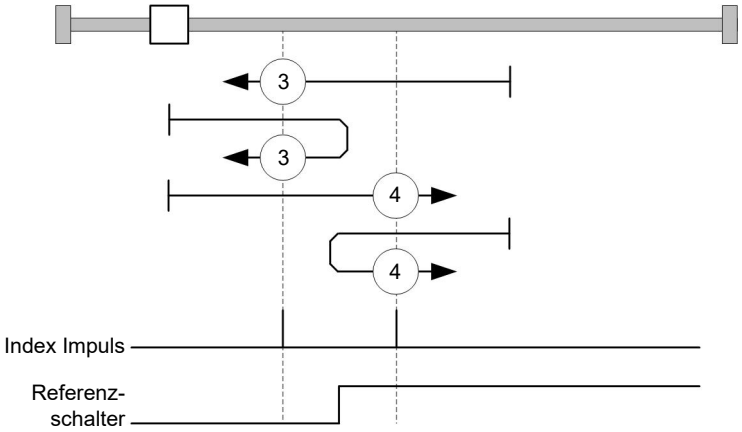

Methode 2 referenziert auf positiven Endschalter und Index-Impuls:



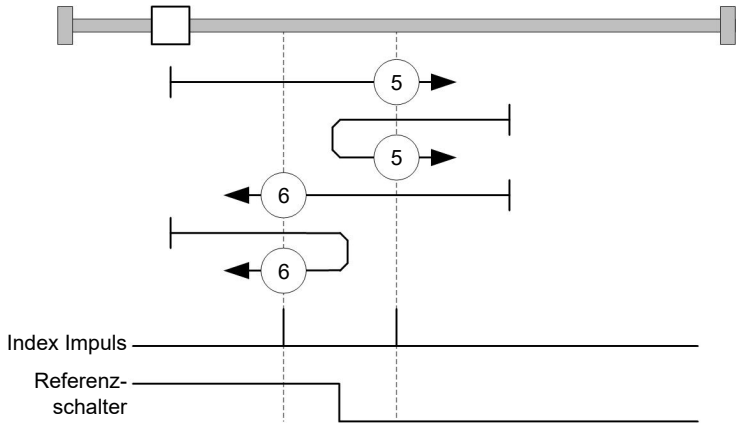

### 6.4.2.5 Methoden 3 bis 6

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:

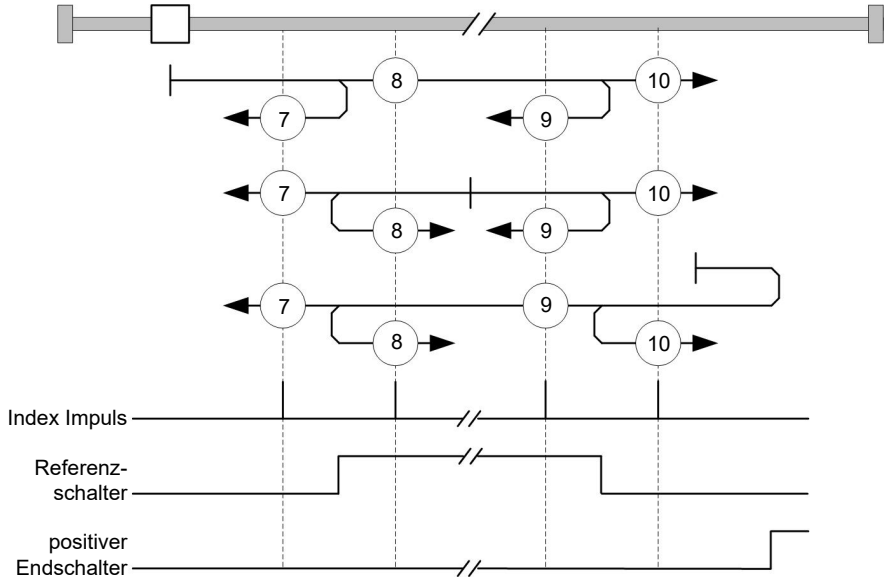



### 6.4.2.6 Methoden 7 bis 14

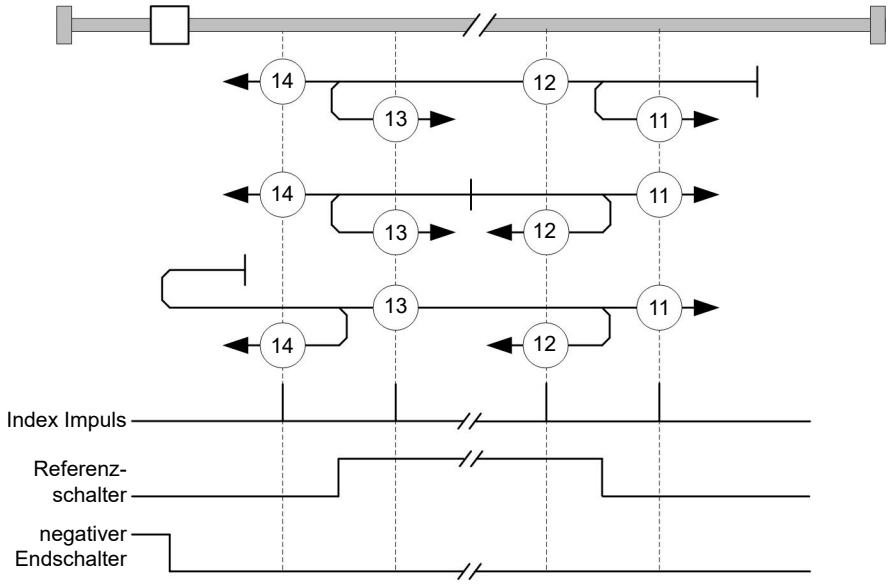
Referenzieren auf Referenzschalter und Index-Impuls (mit Endschaltern).

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:



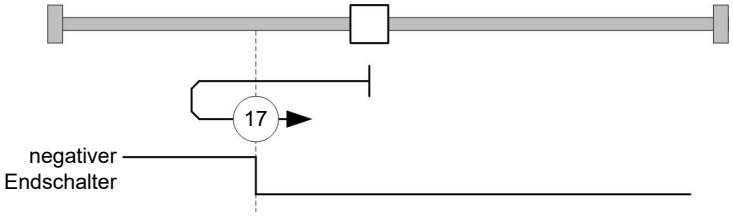

Die Methoden 11 bis 14 berücksichtigen den negativen Endschalter:



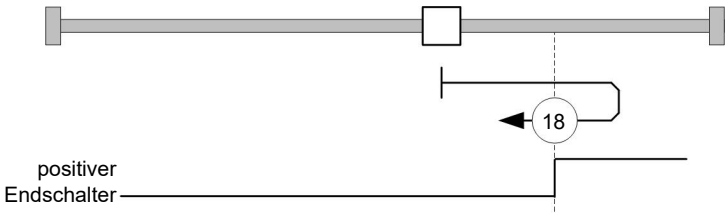

### 6.4.2.7 Methoden 17 und 18

Referenzieren auf den Endschalter ohne den Index-Impuls.

Methode 17 referenziert auf den negativen Endschalter:



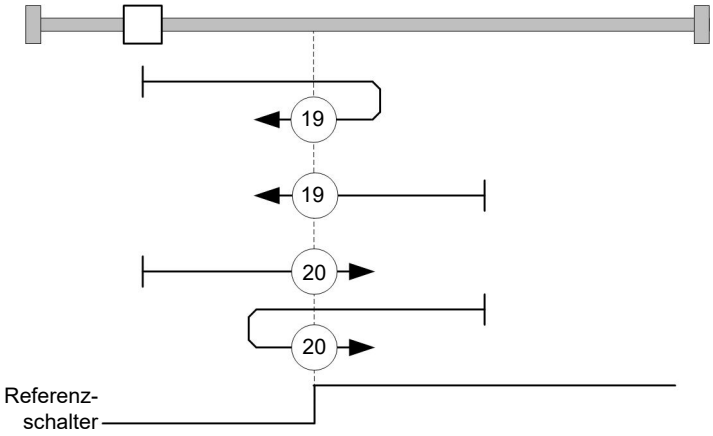

Methode 18 referenziert auf den positiven Endschalter:



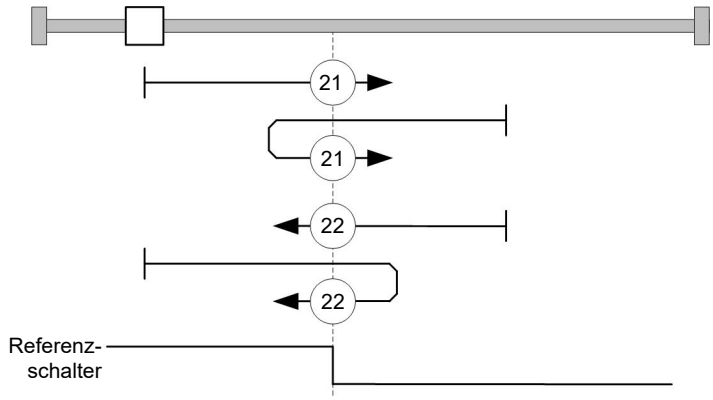

### 6.4.2.8 Methoden 19 bis 22

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:

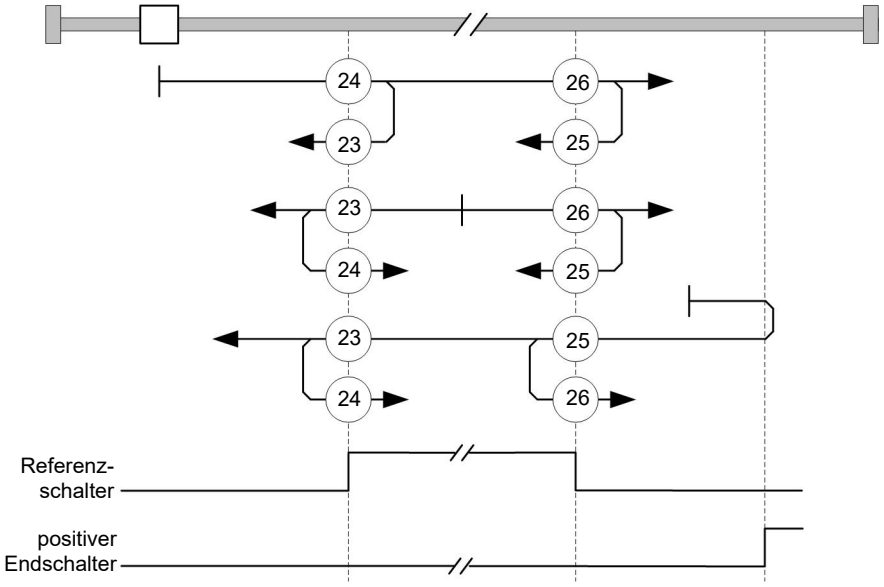



### 6.4.2.9 Methoden 23 bis 30

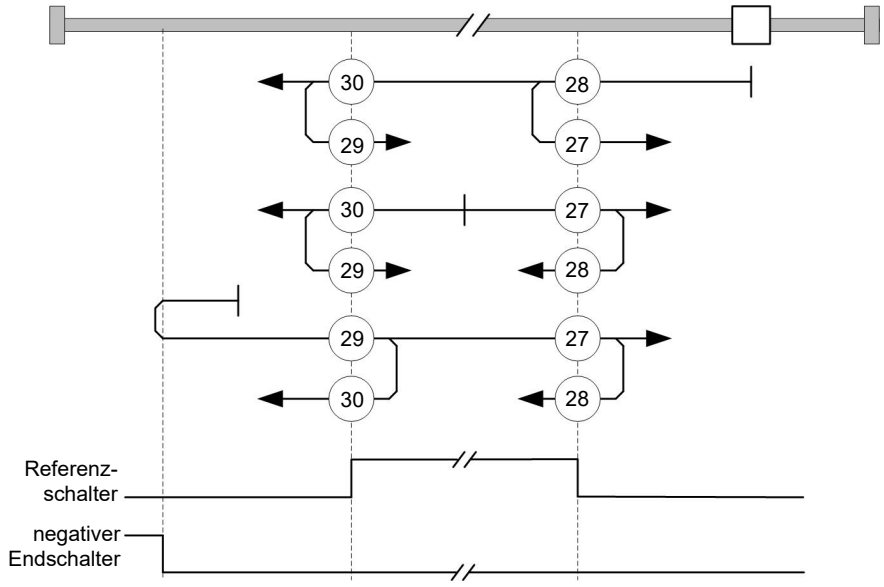
Referenzieren auf Referenzschalter ohne den Index-Impuls (mit Endschaltern).

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:



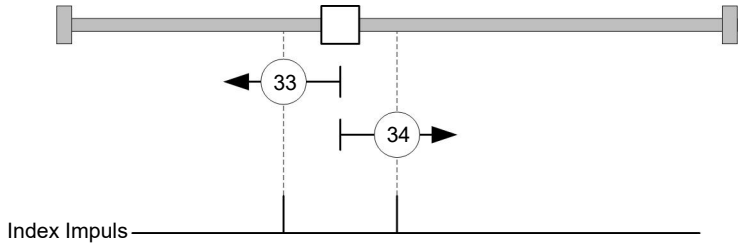

Die Methoden 27 bis 30 berücksichtigen den negativen Referenzschalter:




### 6.4.2.10 Methoden 33 und 34

Referenzieren auf den nächsten Index-Impuls.

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




### 6.4.2.11 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.

## 6.5 Interpolated Position Mode

### 6.5.1 Übersicht

#### 6.5.1.1 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.

#### 6.5.1.2 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.

### 6.5.2 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").

### 6.5.3 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 $605D_h$.

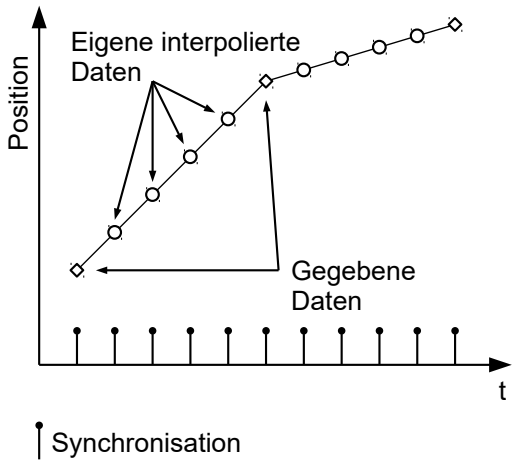
### 6.5.4 Statusword

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)).

### 6.5.5 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.

### 6.5.6 Setup

Das folgende Setup ist nötig:

- 60C2$_h$: Zeit zwischen zwei übergebenen Zielpositionen (Werkseinstellung) 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
- 6084$_h$ (Profile deceleration): gewünschte Bremsbeschleunigung beim Abbremsen
- 60C6$_h$ (Max Deceleration): die maximal erlaubte Bremsbeschleunigung
- Nur wenn der Closed Loop aktiviert ist: 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).

### 6.5.7 Operation

Nach dem Setup ist die Aufgabe der übergerodeten Steuerung, die Zielpositionen rechtzeitig in das Objekt 60C1$_h$:01$_h$ zu schreiben.

## 6.6 Cyclic Synchronous Position

### 6.6.1 Übersicht

#### 6.6.1.1 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.

#### 6.6.1.2 Synchronisierung zum SYNC-Objekt

Um eine gleichmäßige Bewegung zu erzielen, ist es sinnvoll, 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* für die Übertragung der Zielposition zu nutzen.

#### 6.6.1.3 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").

#### 6.6.1.4 Controlword

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

#### 6.6.1.5 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.

### 6.6.2 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.
- Nur wenn der Closed Loop aktiviert ist: $6080_h$ (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: Zykluszeit = Wert des $60C2_h:01_h * 10^{\text{Wert des 60C2:02}}$ 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.
- $60B0_h$ (Position Offset): Offset für den Positionssollwert in benutzerdefinierten Einheiten
- $60B1_h$ (Velocity Offset): Offset für den Geschwindigkeitssollwert in benutzerdefinierten Einheiten
- $60B2_h$ (Torque Offset): Offset für den Drehmomentsollwert in Promille

Folgende Objekte können in dem Modus ausgelesen werden:

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

## 6.7 Cyclic Synchronous Velocity

### 6.7.1 Übersicht

#### 6.7.1.1 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.

#### 6.7.1.2 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").

#### 6.7.1.3 Controlword

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

#### 6.7.1.4 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 |

### 6.7.2 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 $60C2_h$:$01_h$ * $10^{\text{Wert des 60C2:02}}$ 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.
- $60B1_h$ (Velocity Offset): Offset für den Geschwindigkeitssollwert in benutzerdefinierten Einheiten
- $60B2_h$ (Torque Offset): Offset für den Drehmomentsollwert in Promille

Folgende Objekte können in dem Modus ausgelesen werden:

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

## 6.8 Cyclic Synchronous Torque

### 6.8.1 Übersicht

#### 6.8.1.1 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.

#### 6.8.1.2 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").

#### 6.8.1.3 Controlword

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

#### 6.8.1.4 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. |

| Bit | Wert | Beschreibung |
|---|---|---|
| 13 | 0 | Reserviert |
| 13 | 1 | Reserviert |

### 6.8.2 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: Zykluszeit = Wert des $60C2_h:01_h$ * $10^{\text{Wert des 60C2:02}}$ 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.
- $60B2_h$ (Torque Offset): Offset für den Drehmomentsollwert in Promille

Folgende Objekte können in dem Modus ausgelesen werden:

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

## 6.9 Takt-Richtungs-Modus

### 6.9.1 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.

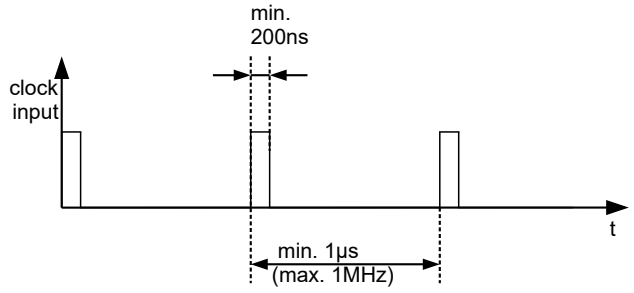
### 6.9.2 Aktivierung

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

### 6.9.3 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 aus den Eingangspulsen resultierende Sollposition wird zyklisch aktualisiert, die Zykluszeit entspricht der Interpolation Time Period ($60C2_h$). Die Eingangspulse, die innerhalb eines Zyklus ankommen, werden in der Steuerung gesammelt und zwischengespeichert.
- Die Skalierung der Schritte erfolgt über die Objekte $2057_h$ und $2058_h$. Dabei gilt die folgende Formel:

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

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.

### 6.9.4 Statusword

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

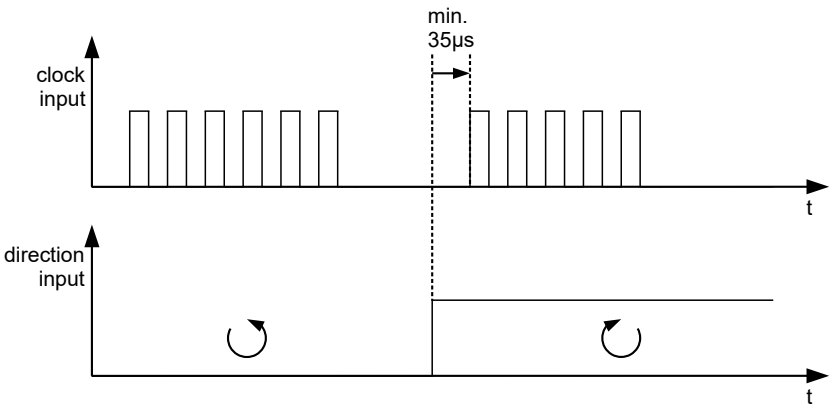
- Bit 13 (Following Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schleppfehler größer als die eingestellten Grenzen ist ($6065_h$ (Following Error Window) und $6066_h$ (Following Error Time Out)).

### 6.9.5 Unterarten des Takt-Richtungs-Modus

#### 6.9.5.1 Takt-Richtungs-Modus (TR-Modus)

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

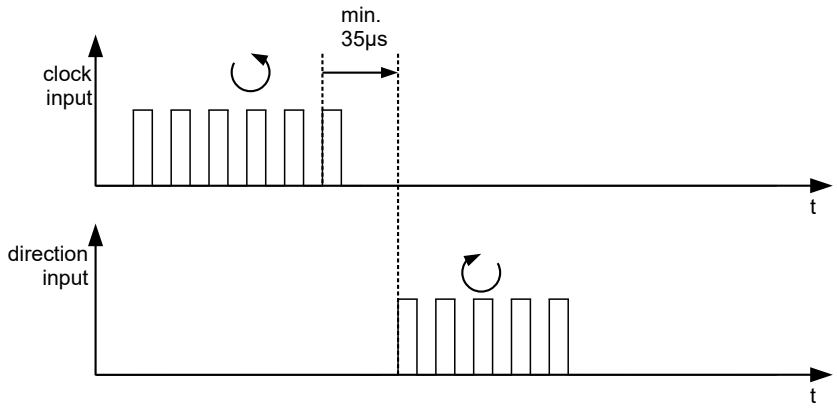
In diesem Modus müssen über den Takteingang die Pulse vorgegeben werden, das Signal des Richtungseingangs gibt dabei die Drehrichtung vor (siehe nachfolgende Grafik).



#### 6.9.5.2 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).




## 6.10 Auto-Setup

### 6.10.1 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.

### 6.10.2 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).

### 6.10.3 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.

### 6.10.4 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

Dieses Produkt verfügt über digitale Ein- und Ausgänge. Die genaue Anzahl abhängig von der Produktvariante entnehmen Sie dem Kapitel Anschlussbelegung.

In 323Ah User Pin Settings konfigurieren Sie die Hardware wie folgt:

- Subindex $01_h$: hier legen Sie den Pegel für die Ein-/Ausgänge fest:
  - Wert "0": 5 V
  - Wert "1": 24 V (Eingänge) bzw. +UB_Logic (Ausgänge)

**HINWEIS**

Verwenden Sie für die Eingänge immer eine Spannung, die kleiner ist als die Betriebsspannung +UB_Logic.

- Subindex $02_h$: hier legen Sie die Beschaltung der digitalen Eingänge fest:
  - Wert "0" (Pull-Down): High-Pegel bei 5/24 V am Pin.
  - Wert "1" (Pull-Up): High-Pegel ohne externe Spannung am Pin.

### 7.1.1 Digitale Eingänge

#### 7.1.1.1 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.

| Eingang | Sonderfunktion | Schaltschwelle umschaltbar | Differenziell / single-ended |
|---|---|---|---|
| 1 | Negativer Endschalter | ja, 5 V oder 24 V (siehe $3240_h$:$06_h$) | single-ended |
| 2 | Positiver Endschalter | ja, 5 V oder 24 V (siehe $3240_h$:$06_h$) | single-ended |
| 3 | Referenzschalter | ja, 5 V oder 24 V (siehe $3240_h$:$06_h$) | single-ended |
| 4 | keine | ja, 5 V oder 24 V (siehe $3240_h$:$06_h$) | single-ended |
| 5 | keine | nein, 5 V bis 24 V Weitbereichseingang | single-ended |
| 6 | keine | nein, 5 V bis 24 V Weitbereichseingang | single-ended |

| Eingang | Sonderfunktion | Schaltschwelle umschaltbar (siehe 323A$_h$) | Differenziell / single-ended |
|---|---|---|---|
| 1 | Negativer Endschalter | ja, 5 V oder 24 V (=UB_Logic) | single-ended |
| 2 | Positiver Endschalter | ja, 5 V oder 24 V (=UB_Logic) | single-ended |
| 3 | Referenzschalter | ja, 5 V oder 24 V (=UB_Logic) | single-ended |
| 4 | keine | ja, 5 V oder 24 V (=UB_Logic) | single-ended |
| 5 | keine | ja, 5 V oder 24 V (=UB_Logic) | single-ended |
| 6 | keine | ja, 5 V oder 24 V (=UB_Logic) | single-ended |

### 7.1.1.2 Verrechnung der Eingänge

Das Objekt 60FD$_h$ (Digital Inputs) enthält eine Zusammenfassung der Eingänge und der Sonderfunktionen. Aus dem Objekt 324Ah Inputs lesen Sie ebenfalls den aktuellen Status der Eingänge aus (samt Hallsensoren und Inkrementalencoder, falls vorhanden).

Die folgende Tabelle zeigt den Wert des entsprechenden Bits im jeweiligen Objekt für die Eingänge, abhängig von der Konfiguration in 323Ah User Pin Settings:

| Spannung am Pin | Subindex 02 (Pull-Up Enable) | Subindex 01 (Voltage Level Select) | Bit-Wert |
|---|---|---|---|
| n.c | 0 (Pull-Down) | X | 0 |
| GND | 0 (Pull-Down) | X | 0 |
| 5 V | 0 (Pull-Down) | 0 (5 V) | 1 |
| 5 V | 0 (Pull-Down) | 1 (24 V) | 0 |
| 24 V | 0 (Pull-Down) | 1 (24 V) | 1 |
| n.c. | 1 (Pull-Up) | X | 1 |
| GND | 1 (Pull-Up) | X | 0 |
| 5 V | 1 (Pull-Up) | 0 (5 V) | 1 |
| 5 V | 1 (Pull-Up) | 1 (24 V) | 0 |
| 24 V | 1 (Pull-Up) | 1 (24 V) | 1 |

### 7.1.1.3 Sonderfunktionen

Die Firmware wertet folgende Bits in 60FD$_h$ aus:

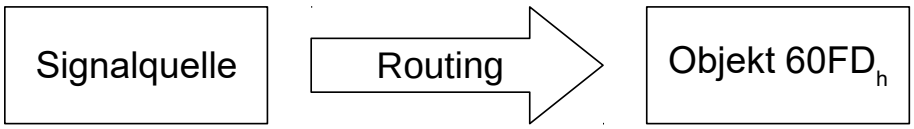
- Bit 0: Negativer Endschalter (siehe Begrenzung des Bewegungsbereichs)
- Bit 1: Positiver Endschalter (siehe Begrenzung des Bewegungsbereichs)
- Bit 2: Referenzschalter (siehe Homing)
- Bit 3: Interlock (siehe Interlock-Funktion)

Die Zuordnung der Bits zu den Pins legen sie mit dem *Input Routing* fest.

### 7.1.1.4 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.




### 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 |
| 81 | 51 | Negativer Block |
| 82 | 52 | Positiver Block |
| 100 | 64 | STO_IN_A |
| 101 | 65 | STO_IN_B |

Die nachfolgende Tabelle beschreibt die invertierten Signale der vorherigen Tabelle.

| Nummer | | |
|---|---|---|
| **dec** | **hex** | **Signalquelle** |
| 128 | 80 | Signal ist immer 1 |

| Nummer | | Signalquelle |
|---|---|---|
| dec | hex | |
| 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 |
| 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 |
| 228 | E4 | Invertierter STO_IN_A |
| 229 | E5 | Invertierter STO_IN_B |

**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.

### 7.1.1.5 Interlock-Funktion

Bei der Interlock-Funktion handelt es sich um eine Freigabe, die Sie über das Bit 3 in $60FD_h$ steuern. Steht dieses Bit auf "1", darf der Motor fahren. Steht das Bit auf "0", wird die Steuerung in den Fehlerzustand versetzt und die in $605E_h$ hinterlegte Aktion ausgeführt.

Mittels *Input Routing* legen Sie fest, welche Signalquelle auf Bit 3 des $60FD_h$ geroutet wird und die Interlock-Funktion steuern soll.

**Beispiel**

Eingang 4 soll auf Bit 3 des Objekts $60FD_h$ geroutet werden, um die Interlock-Funktion zu steuern. Ein Low-Pegel soll zum Fehlerzustand führen.

1. Um den Eingang 4 auf Bit 3 zu routen, setzen Sie das $3242_h$:$04_h$ auf "4".

## 7.1.2 Digitale Ausgänge

### 7.1.2.1 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 in den unteren Bits 0 bis 15 eingetragen. Im Moment ist nur Bit 0 belegt, das die Motorbremse steuert.

### 7.1.2.2 Beschaltung

**HINWEIS**

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

**HINWEIS**

Um die digitalen Ausgänge benutzen zu können, müssen Sie eine Spannung (12 …30 V DC) an den Pins 1 und 2 von X2 – Spannungsversorgung anschließen (Logikversorgung).

Die Ausgangsspannung entspricht dann

- dieser Logik-Spannung, wenn Sie $323A_{hh}$:$01_h$ auf "1" setzen,
- 5 V, wenn Sie $323A_{hh}$:$01_h$ auf "0" setzen (Werkseinstellung).

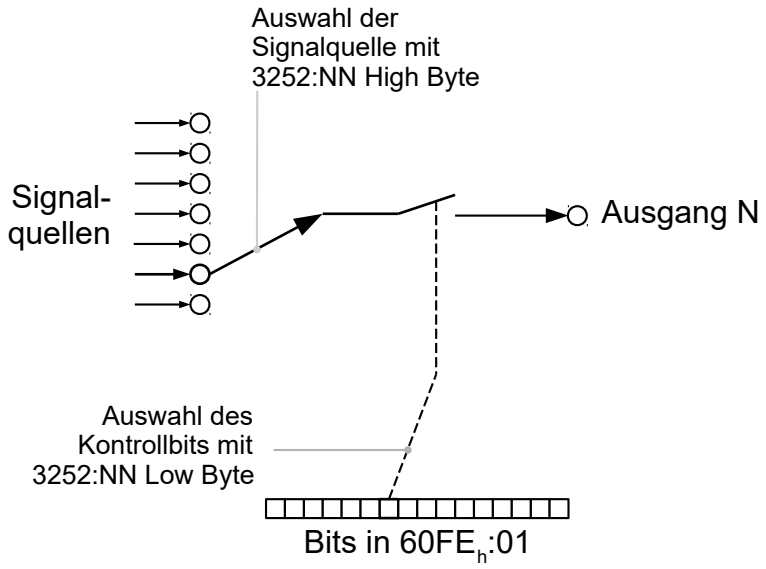
Der Strom soll 100 mA nicht überschreiten.

### 7.1.2.3 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 n 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 n (siehe nachfolgende Abbildung).




### 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 (Software-PWM) |
| 02$_h$ | Konfiguration des Ausgangs 1 |
| 03$_h$ | Konfiguration des Ausgangs 2 (falls verfügbar) |
| … | … |
| 0n$_h$ | Konfiguration des Ausgangs n (falls verfügbar) |

**HINWEIS**

Die maximale Ausgangsfrequenz des PWM-Ausgangs (Software-PWM) ist 2 kHz. Alle anderen Ausgänge können nur bis zu 500Hz Signale erzeugen.

Die Subindizes 3252$_h$:01$_h$ bis 0n$_h$ sind 16 Bit breit, wobei das High Byte die Signalquelle auswählt (z. B. den PWM-Generator) und das Low Byte das Kontrollbit im Objekt 60FE$_h$:01 bestimmt.

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

| Nummer in 3252:01 bis 0n | |
|---|---|
| 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 |

| Nummer in 3252:01 bis 0n | |
|---|---|
| $09XX_h$ | 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 |
| $10XX_h$ | PWM-Signal, das mit Objekt $2038_h$:$05_h$ und $06_h$ konfiguriert wird |
| $11XX_h$ | Invertiertes PWM-Signal, das mit Objekt $2038_h$:$05_h$ und $06_h$ konfiguriert wird |
| $20XX_h$ | Pulsgenerator |

**HINWEIS**

Bei jeder Änderung des "Encodersignals" ($6063_h$) oder der aktuellen Position ($6064_h$, in benutzerdefinierten Einheiten) um ein Inkrement wird ein Puls am digitalen Ausgang ausgegeben (bei Frequenzteiler 1). Berücksichtigen Sie dies bei der Auswahl des Frequenzteilers und der Einheit, besonders bei Verwendung von Sensoren mit niedriger Auflösung (wie z. B. Hall-Sensoren).

**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)
- $3252_h$:$02_h$ = $0405_h$ ($04XX_h$ + $0005_h$)
- $04XX_h$: 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.

**Beispiel**

Das Bremsen-PWM-Signal soll auf Ausgang 2 gelegt werden. Da die automatische Bremsensteuerung das Bit 0 des 60FE:$01_h$ benutzt, soll dieses als Kontrollbit benutzt werden.

- $3250_h$:$08_h$ = 1 (Routing aktivieren)
- $3252_h$:$03_h$ = $1080_h$ (=$10XX_h$ + $0080_h$). Dabei gilt:
  - $10XX_h$: Bremsen-PWM-Signal
  - $0080_h$: Auswahl des invertierten Bits 0 des Objekts 60FE:01

### 7.1.3 Pulsgenerator

Sie haben die Möglichkeit, die Istposition über zwei Pins der Steuerung auszugeben und an Ihre SPS oder eine weitere Steuerung weiterzuleiten. Die maximale Frequenz beträgt dabei 200 kHz.

#### Funktion der Pins aktivieren

Um die Funktion zu aktivieren, verwenden Sie das *Output Routing* und setzen Sie die Quelle in den entsprechenden Subindizes von $3252_h$ auf $20XX_h$.

#### Typ der Ausgangssignale wählen

Sie können im Objekt $205C_h02_h$ einen der folgenden Typen wählen :

- Wert "0": zwei um 90° phasenversetzte Kanäle auf Kanal A (voreilend bei Fahrten in positiver Richtung) und B, analog zu einem inkrementellen Encoder
- Wert "1": ein Takt- und ein Richtungssignal auf Kanal A und B, analog zu den Signalen im Takt-Richtungs-Modus
- Wert "2": zwei Taktsignale, analog zu den Signalen im Rechts-/Linkslauf-Modus (CW/CCW-Modus)

#### Quelle der Positionsdaten wählen

Über den Pulsgeneratorausgang werden die Positionsdaten einer der vorhandenen Rückführungen wiedergeben. Um die Quelle zu wählen, setzen Sie das Bit 3 im entsprechenden Subindex des Objekts $3203_h$ auf "1".

Wenn Sie kein Bit setzen, wird der Wert aus 205C:$01_h$ verwendet (max. 255 Pulse). In diesem Fall werden die Pulse kontinuierlich gesendet, unabhängig von der Motorbewegung.

#### Auflösung der Ausgangssignale einstellen

Über den Numerator in 205C:$03_h$ und den Denominator in 205C:$04_h$ definieren Sie die Umrechnung des Quellsignals zu virtuellen Encodersignalen des Pulsgenerators.

## 7.2 Analoge Eingänge

Die Steuerung besitzt 2 Analogeingänge mit einer Auflösung von 12 Bit. Sie befinden sich an den Pins 16 und 18 am X4 – Ein- und Ausgänge..

Den Analogwert können Sie in einem NanoJ-Programm auslesen und beliebig verwenden, um z. B. die Zielgeschwindigkeit vorzugeben.

### 7.2.1 Objekteinträge

Um den Wert des Analogeingangs auszulesen und ggf. zu manipulieren, benutzen Sie folgende OD-Einstellungen:

- $3220_h$ (Analog Inputs):
  Dieses Objekt zeigt die Momentanwerte der Analogeingänge in *ADC Digits* an.
- $3320_h$ (Read Analogue Input):
  Dieses Objekt zeigt die Momentanwerte der Analogeingänge in benutzerdefinierten Einheiten an.
- $3321_h$ (Analogue Input Offset):
  Dies ist der Offset, der zum eingelesenen Analogwert ($3220_h$) addiert wird, bevor die Skalierung (Multiplikator aus dem Objekt $3322_h$ und Teiler aus dem Objekt $3323_h$) vorgenommen wird.
- $3322_h$(Analogue Input Factor Numerator):
  Dies ist der Wert, mit dem der eingelesene Analogwert ($3220_h$ + 3321) multipliziert wird, bevor er in das Objekt $3320_h$ geschrieben wird.
- $3323_h$(Analogue Input Factor Denominator):
  Dies ist der Wert, mit dem der eingelesene Analogwert ($3220_h$+ $3321_h$) dividiert wird, bevor er in das Objekt $3320_h$ geschrieben wird.

### 7.2.2 Analogwert skalieren

Den Wert lesen Sie im Objekt $3320_h$ (Read Analogue Input): Dieses Objekt zeigt die Momentanwerte der Analogeingänge in benutzerdefinierten Einheiten an.

Die benutzerdefinierten Einheiten setzten sich aus Offset ($3321_h$) und Skalierungswert ($3322_h$/ $3323_h$) zusammen. Sind beide noch mit Default-Werten beschrieben, wird der Wert in $3320_h$ in der Einheit *Millivolt* angegeben.

## 7.3 Automatische Bremsensteuerung

### 7.3.1 Beschreibung

Die automatische Bremsensteuerung wird aktiv, wenn die Steuerung in den Zustand *Operation Enabled* der CiA 402 Power State Machine gebracht wird, sonst bleibt die Bremse immer geschlossen.

Der Bremsen-Ausgang der Steuerung resultiert in einem PWM-Signal, welches sich in der Frequenz und in dem Tastverhältnis einstellen lässt.

Für das Zusammenspiel der Bremse mit dem Motorstoppverhalten, lesen Sie auch das Kapitel Power State machine - Bremsreaktionen.

### 7.3.2 Aktivierung und Anschluss

Die Bremse kann entweder automatisch oder manuell gesteuert werden:

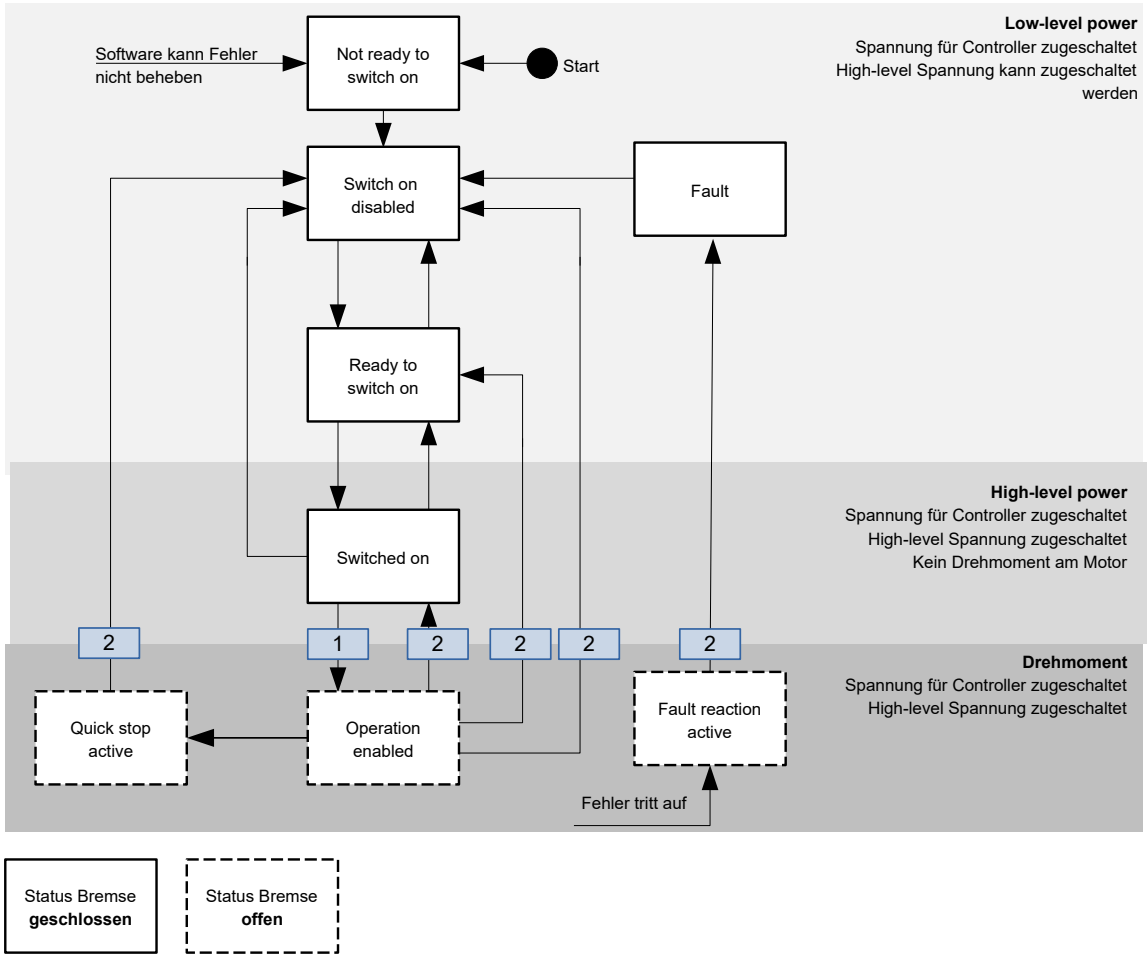
- Automatisch: Bit 2 des Objekts $3202_h$ auf "1" setzen aktiviert die Bremsensteuerung.
- Manuell: Bit 2 des Objekts $3202_h$ auf "0" setzen deaktiviert die Bremsensteuerung, die Bremse lässt sich jetzt mit dem Bit 0 im Objekt $60FE_h$:$01_h$ kontrollieren.

#### 7.3.2.1 Anschluss

Der Bremsenausgang befindet sich am X4 – Ein- und Ausgänge.

### 7.3.3 Steuerung der Bremse

Die nachfolgende Grafik zeigt die Zustände der CiA 402 Power State Machine zusammen mit den Zuständen der Bremse für den automatischen Modus.




Bei dem Übergang, welcher mit 1 markiert ist, werden folgende Schritte durchgeführt:

1. Der Motorstrom wird eingeschaltet.
2. Die Zeit, welche in $2038_h$:$3_h$ hinterlegt wird, wird abgewartet.
3. Die Bremse löst sich.
4. Die Zeit, welche in $2038_h$:$4_h$ hinterlegt wird, wird abgewartet.
5. Der Zustand *Operation Enabled* wird erreicht, die Motorsteuerung kann Fahrbefehle umsetzen.

Bei allen Übergängen, welche mit 2 markiert sind, werden folgende Schritte durchgeführt:

1. Der Motor wird zum Stillstand gebracht.
2. Die Zeit, welche in $2038_h$:$1_h$ hinterlegt wird, wird abgewartet.
3. Die Bremse wird aktiviert.
4. Die Zeit, welche in $2038_h$:$2_h$ hinterlegt wird, wird abgewartet.
5. Der Motorstrom wird abgeschaltet.

## 7.3.4 Bremsen-PWM

Die eingeschaltete Bremse erzeugt am Ausgang der Steuerung ein PWM-Signal, welches im Tastgrad und der Frequenz eingestellt werden kann. Sollte ein Ausgangspin ohne PWM benötigt werden, lässt sich ein Tastgrad von 100 Prozent einstellen. Die maximale Spannung am Pin *Bremse +* schalten Sie zwischen 5 und 24 V (=UB_Logic, Pin 2 am Stecker X2 – Spannungsversorgung) mittels Software mit Objekt $323A_h$.

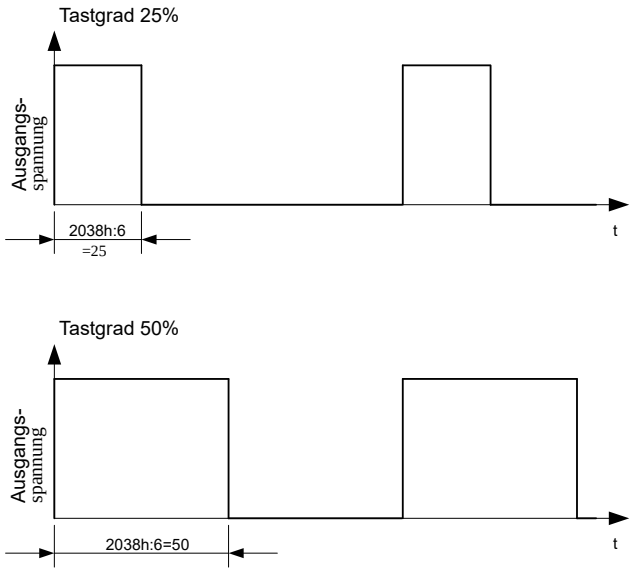
### 7.3.4.1 Frequenz

Die Frequenz der Bremsen-PWM kann im Objekt $2038_h$:$5_h$ eingestellt werden. Die Einheit ist Hertz, ein Wert kleiner 50 oder größer 20000 ist nicht möglich.

### 7.3.4.2 Tastgrad

Der Tastgrad – das Verhältnis Impuls- zu Periodendauer – wird im $2038_h$:$6_h$ eingestellt. Der Wert wird als Prozentzahl angesehen und kann zwischen 1 und 100 gewählt werden. Bei einem Wert von 100 ist der Ausgangspin dauerhaft eingeschaltet.

In nachfolgender Abbildung ist beispielhaft ein Tastgrad von 25 und 50 Prozent eingezeichnet, wobei die Frequenz beibehalten wurde.




## 7.4 Externe Ballast-Schaltung

Beim Bremsvorgang wird durch Selbstinduktion des Motors elektrische Energie in den Zwischenkreis zurückgespeist. Sofern kein rückspeisefähiges Netzteil eingesetzt wird, kann die Bremsleistung zu einem Anstieg der Zwischenkreisspannung führen, der ohne zusätzliche Maßnahmen nur durch den Eigenverbrauch sowie Kapazitäten im Zwischenkreis begrenzt wird.

Um eine Beschädigung der Steuerung durch Überspannung zu verhindern, kann es je nach Höhe der Bremsleistung erforderlich sein, überschüssige Energie in Form von Wärme abzuführen. Die Steuerung stellt hierzu am Pin 3 / 4 von X2 – Spannungsversorgung einen Anschluss für einen externen Ballast-Widerstand bereit, falls der interne (PWR1635-215-10R0J von Bourns mit 25 W max. Leistung) nicht ausreicht.

### 7.4.1 Steuerung des Ballast-Widerstands

In der Firmware der Steuerung ist eine Ballaststeuerung und -Überwachung implementiert, die zwei Funktionen besitzt:

- Begrenzung der Zwischenkreisspannung durch Aktivierung des Ballast-Widerstands oder Abschaltung der Endstufe
- Schutz des Ballast-Widerstands vor thermischer Überlastung

In den folgenden Kapiteln werden die zu konfigurierenden Parameter beschrieben.

### 7.4.2 Ballast aktivieren

Um den Ballast zu aktivieren, setzen Sie Bit 0 in $4021_h$:$01_h$ auf "1". Wenn Sie den externen Ballast-Widerstand verwenden möchten, setzen Sie Bit 2 in $4021_h$:$01_h$ auf "1".

Die Ansprechschwelle in Millivolt, sowie die Hysterese beim Ein-/Ausschalten, tragen Sie in $4021_h$:$02_h$ bzw. $4021_h$:$03_h$ ein (für den internen Widerstand vorkonfiguriert).

Schafft es der Ballast trotz Aktivierung nicht, den Anstieg der Zwischenkreisspannung zu begrenzen, wird beim Überschreiten der Überspannungsschwelle ($2034_h$) ein Fehler erzeugt und die Treiber-Endstufe abgeschaltet.

### 7.4.3 Ballast-Überwachung

Die Firmware überwacht den Ballast-Widerstand kontinuierlich, indem die in ihm umgesetzte Energie aufsummiert wird — unter Berücksichtigung der Wärmemenge, die der Widerstand durch Konvektion an seine Umgebung abgibt.

Überschreitet die Energie den zulässigen Grenzwert, wird ein Einschalten des Ballast-Widerstands blockiert und eine Warnung mit dem Error-Code $7113_h$ (siehe $1003_h$) erzeugt. Nachdem der Widerstand ausreichend abgekühlt ist, wird die Blockierung automatisch aufgehoben.

Um die Überwachung für den externen Widerstand zu konfigurieren, müssen Sie folgende Widerstandsparameter dem Datenblatt des Ballast-Widerstands entnehmen bzw. ermitteln und in den entsprechenden Subindex von $4021_h$ eintragen:

**Nominal Resistance $R_{Ballast}$, [mOhm]**

Nennwert des Ballast-Widerstands

**Cooling Power $P_{Stat_TA_Max}$, [mW]**

Die Wärmemenge, die der Widerstand durch Konvektion kontinuierlich an seine Umgebung abgeben kann bzw. darf. Diese können Sie wie folgt berechnen:

$$P_{Stat_TA_Max} = (T_{Ballast_Max} - TA_{Max}) / R_{th,A}$$

- $T_{Ballast_Max}$: Maximal zulässige Oberflächentemperatur des Widerstandes. Begrenzt durch die Daten des Widerstands (Datenblattangabe) oder durch die Einbauposition (Temperaturfestigkeit benachbarter Bauteile).
- $TA_{Max}$: maximale Temperatur in der Umgebung des Ballasts
- $R_{th,A}$: Wärmeübergangswiderstand des Ballast-Widerstands zur Umgebung (Datenblattangabe)

**Short Term Energy Limit $E_{ST_25°C}$, [mWs]**

Energiemenge, die dem Widerstand innerhalb kurzer Lastpulse (<1 Sekunde) zugeführt werden darf, ohne ihn zu überlasten.

Dabei ist das Material des Widerstandselements (Draht, Dickschicht) der begrenzende Faktor, da bei kurzen Pulsen praktisch nur dieses Energie aufnehmen kann und sich erhitzt.

Der Wert wird bei Lastwiderständen typischerweise im Datenblatt angegeben.

**Long Term Energy Limit $E_{LT_TA_Max}$, [mWs]**

Energiemenge, die dem Widerstand innerhalb der *Long Term Reference Time* (siehe unten, typischerweise zwischen 1 und 5 Sekunden) zugeführt werden darf, ohne ihn zu überlasten.

Bei langen Pulsdauern nimmt auch das Trägermaterial (Zement oder Keramikkörper) Energie auf und verlangsamt dadurch den Temperaturanstieg.

Die Long-Term-Überlastfähigkeit eines Lastwiderstands wird üblicherweise in dessen Datenblatt in Form eines Überlastfaktors für einen bestimmten Zeitraum (z. B. 5-fache Nennleistung für 5 Sekunden) angegeben.

**Long Term Reference Time $t_{LT_Ref}$, [ms]**

die Bezugszeit für das Long Term Energy Limit (typischerweise zwischen 1 und 5 Sekunden)

Sind die Parameter nicht gültig oder nicht realistisch, wird ein Fehler mit dem Error-Code $7110_h$ (siehe $1003_h$) erzeugt.

# 7.5 $I^2t$ Motor-Überlastungsschutz

## 7.5.1 Beschreibung

**HINWEIS**

Für Schrittmotoren wird nur der Nennstrom und kein Maximalstrom angegeben. Daher erfolgt die Nutzung von $I^2t$ mit Schrittmotoren ohne Gewähr.

Das Ziel des $I^2t$ 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).

## 7.5.2 Objekteinträge

Folgende Objekte haben Einfluss auf den $I^2t$ Motor-Überlastungsschutz:

- $6075_h$ Motor Rated Current - Gibt den Nennstrom in mA an.
- $6073_h$ Max Current - Gibt den Maximalstrom in Promille des eingestellten Nennstroms an.
- $3219_h$:$01_h$ I²t Duration Of Max Current - Gibt die maximale Dauer des Maximalstroms in ms an.

## 7.5.3 Aktivierung

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

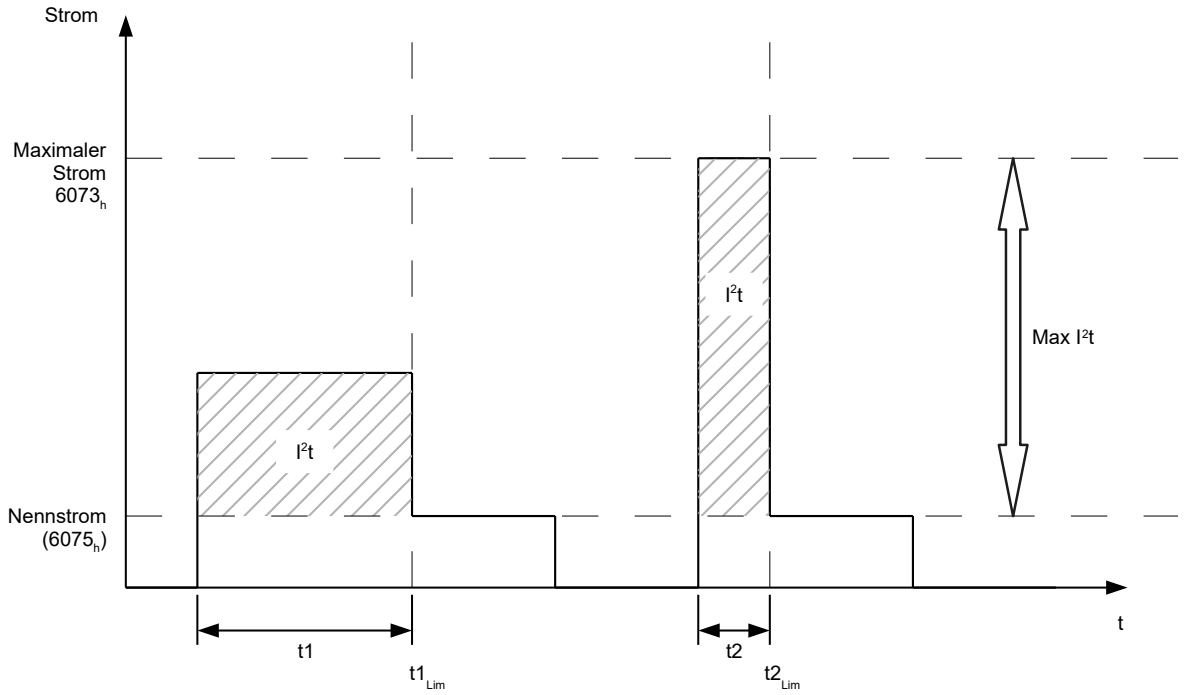
Zum Aktivieren des Modus müssen Sie die drei oben genannten Objekteinträge sinnvoll beschreiben. Das bedeutet, dass der Maximalstrom größer als der Nennstrom sein muss und ein Zeitwert für die maximale Dauer des Maximalstroms eingetragen sein muss. Wenn diese Bedingungen nicht erfüllt sind, bleibt die $I^2t$ Funktionalität deaktiviert.

## 7.5.4 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. Darauffolgend wird der Strom sofort auf Nennstrom gesenkt. Der Maximalstrom wird durch den maximalen Motorstrom ($6073_h$) begrenzt.

In den folgenden Diagrammen sind die Zusammenhänge noch einmal dargestellt.




Im ersten Abschnitt t1 ist der Stromwert höher als der Nennstrom. Am Zeitpunkt $t1_{Lim}$ wird $I^2t_{Lim}$ erreicht und der Strom wird auf Nennstrom begrenzt. Danach kommt während der Dauer t2 ein Strom, der dem Maximalstrom entspricht. Dementsprechend ist der Wert für $I^2t_{Lim}$ schneller erreicht, als im Zeitraum t1.

## 7.6 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.

### 7.6.1 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.

- Motor: Parameter mit Bezug auf den Motor (BLDC/Stepper, Motorstrom, *Closed/Open Loop*...). Einige werden vom Auto-Setup gesetzt und gespeichert.
- Sensor: Parameter mit Bezug auf Sensor/Encoder, die entweder vom Auto-Setup gesetzt werden, oder den Datenblättern entnommen werden können.
- CANopen: Parameter mit Bezug auf die CANopen-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.

### 7.6.2 Kategorie: Kommunikation

- $1005_h$: COB-ID Sync
- $1006_h$: Communication Cycle Period
- $1007_h$: 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

### 7.6.3 Kategorie: Applikation

- 2034$_h$: Upper Voltage Warning Level
- 2035$_h$: Lower Voltage Warning Level
- 2038$_h$: Brake Controller Timing
- 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
- 205C$_h$: Pulse Generator Configuration
- 2290$_h$: PDI Control
- 2300$_h$: NanoJ Control
- 2800$_h$: Bootloader And Reboot Settings
- 320D$_h$: Moment Of Inertia
- 3219$_h$: Current Configuration
- 321A$_h$: Current Controller Parameters
- 321B$_h$: Velocity Controller Parameters
- 321C$_h$: Position Controller Parameters
- 321D$_h$: Feedforward
- 321E$_h$: Voltage Limit
- 323A$_h$: User Pin Settings
- 3241$_h$: Digital Input Position Capture
- 3242$_h$: Digital Input Routing
- 3250$_h$: Digital Outputs Control
- 3252$_h$: Digital Output Routing
- 325A$_h$: Outputs
- 3260$_h$: Pwm Output 0
- 3261$_h$: Pwm Output 1
- 3262$_h$: Pwm Output 2
- 3273$_h$: Generic SPI Hardware Configuration
- 3274$_h$: Generic SPI Mosi Data
- 3321$_h$: Analog Input Offsets
- 3322$_h$: Analog Input Numerators
- 3323$_h$: Analog Input Denominators
- 3700$_h$: Deviation Error Option Code
- 3701$_h$: Limit Switch Error Option Code
- 4015$_h$: Special Drive Modes
- 4021$_h$: Ballast Configuration
- 6007$_h$: Abort Connection Option Code
- 6040$_h$: Controlword
- 605A$_h$: Quick Stop Option Code
- 605B$_h$: Shutdown Option Code
- 605C$_h$: Disable Operation Option Code
- 605D$_h$: Halt Option Code
- 605E$_h$: Fault Reaction 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
- $606F_h$: Velocity Threshold
- $6070_h$: Velocity Threshold 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
- $6080_h$: Max Motor Speed
- $6081_h$: 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
- $60F2_h$: Position Option Code
- $60F8_h$: Max Slippage
- $60FE_h$: Digital Outputs
- $60FF_h$: Target Velocity

### 7.6.4 Kategorie: Benutzer

- $2701_h$: Customer Storage Area

### 7.6.5 Kategorie: Bewegung

- PLACEHOLDER DRIVE

### 7.6.6 Kategorie: Motor

- $2030_h$: Pole Pair Count
- $3202_h$: Motor Drive Submode Select
- $6073_h$: Max Current
- $6075_h$: Motor Rated Current

### 7.6.7 Kategorie: Tuning

- PLACEHOLDER TUNING

### 7.6.8 Kategorie: Sensor

- $3203_h$: Feedback Selection
- $3380_h$: Feedback Sensorless
- $3390_h$: Feedback Hall
- $33A0_h$: Feedback Incremental A/B/I 1
- $33B0_h$: Feedback SSI 1
- $60E6_h$: Additional Position Encoder Resolution - Encoder Increments
- $60EB_h$: Additional Position Encoder Resolution - Motor Revolutions
- $60F1_h$: Additional Encoder Inversion

### 7.6.9 Kategorie: CANopen

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

### 7.6.10 Kategorie: Modbus RTU

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

### 7.6.11 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**

- Während der Speicherung kann die Feldbusfunktion beeinträchtigt sein.
- Das Speichern kann einige Sekunden dauern. Unterbrechen Sie während des Speicherns keinesfalls die Spannungszufuhr. 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.

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 $0A_h$ (CANopen) |
| $02_h$ | Kommunikation |
| $03_h$ | Applikation |
| $04_h$ | Benutzer |
| $05_h$ | Motor |
| $06_h$ | Sensor |
| $0A_h$ | CANopen |

### 7.6.12 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 $06_h$ (Sensor) und $0A_h$ (CANopen) |
| $02_h$ | Kommunikation |
| $03_h$ | Applikation |
| $04_h$ | Benutzer |
| $05_h$ | Motor |
| $06_h$ | Sensor |
| $0A_h$ | CANopen |

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

[1] Das entspricht dezimal der $1702257011_d$ bzw. dem ASCII String `save`.
[2] Das entspricht dezimal der $1684107116_d$ bzw. dem ASCII String `load`.

### 7.6.13 Konfiguration verifizieren

Das Objekt $1020_h$ kann herangezogen werden, um die Konfiguration zu verifizieren. Es agiert wie ein Modifkationsmarker 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. Das gilt auch für Änderungen am Verzeichnis nach einem Firmware-Update, z. B. wenn ein gespeichertes Objekt nicht mehr existiert. Sollten das Datum oder die Uhrzeit in 1020 nicht den erwarteten Werten entsprechen, wurden Objekte wahrscheinlich mit anderen als den erwarteten Werten gespeichert.

## 7.7 Generic SPI

Über diese SPI-Schnittstelle kann die Steuerung mit externen Geräten kommunizieren, beispielsweise Port-Expandern oder Displays.

**HINWEIS**

Die verwendete SPI-Peripherie muss eine Clock-Frequenz von mindestens 164 KHz unterstützen.

Die Einstellungen der SPI-Schnittstelle befinden sich im Objekt $3273_h$:$01_h$ (Generic SPI Hardware Configuration):

- Bit 0 (Clock Phase):
  - Wert = "0": Die Datenübertragung beginnt mit der ersten Clock-Flanke, nachdem *Chip Select* auf Low gezogen wurde.
  - Wert = "1": Die Datenübertragung beginnt mit der zweiten Clock-Flanke, nachdem *Chip Select* auf Low gezogen wurde.
- Bit 1 (Clock Polarity): Mit diesem Bit können Sie die Polarität des Clock-Signals invertieren. Der Wert 0 heißt, der Pegel bleibt auf Low, wenn die Clock ruht.
- Bits 2 bis 4 (Baudrate): Hier stellen Sie die Clock-Frequenz ein:
  - $000_b$: 21 MHz
  - $001_b$: 10,5 MHz
  - $010_b$: 5,25 MHz
  - $011_b$: 2625 KHz
  - $100_b$: 1312,5 KHz
  - $101_b$: 656,25 KHz
  - $110_b$: 328,125 KHz
  - $111_b$: 164,0625 KHz
- Bit 10 (CS Polarity): Mit diesem Bit können Sie die Polarität des *Chip Select* invertieren. Der Wert 0 heißt, der Pegel bleibt auf High, wenn das Signal ruht.

Die Daten werden über folgende Objekte gesendet/empfangen:

- $3274_h$ (Generic SPI Mosi Data):
  - Subindex 1 bis $1F_h$ (Generic SPI Mosi Data Byte #1bis #31): Hier schreiben Sie die Daten, die gesendet werden sollen, geteilt in bis zu 31 Bytes.
  - Subindex 0 (Length of SPI message to be sent): Hier tragen Sie anschließend die Anzahl der Bytes (= Subnidices) ein, die gesendet werden sollen. Im nächsten Millisekunden-Zyklus werden die Daten gesendet und der Subindex wird auf den Wert "0" zurücksetzt.
- $3275_h$ (Generic SPI Miso Data): Hier lesen Sie die empfangenen Daten.
  - Subindex 0 (Length of received SPI message): Dem Wert entnehmen Sie, wie viele Daten-Bytes (=Subindices) empfangen wurden.
  - Subindex 1 bis $1F_h$ (Generic SPI Miso Data Byte #1bis #31): Hier finden Sie die Daten, die empfangen wurden.

# 8 CANopen

Sie können die Steuerung mittels CANopen ansprechen. Die Steuerung kann in einem Netzwerk als *Slave* arbeiten.

In diesem Kapitel werden die Dienste der CANopen-Kommunikationsstruktur beschrieben. Die Nachrichten für CANopen werden 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*, Data 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 CAN-Nachrichten beschrieben, diese werden wie folgt notiert:

```
583 | 41 09 10 00 1E 00 00 00
```

```
183R | DLC=0
```

Die einzelnen Nachrichten werden wie folgt geschrieben:

- Alle Zahlen werden in hexadezimaler Schreibweise angegeben, auf das einleitende `0x` wird wegen der verkürzten Schreibweise verzichtet.
- Normale Datennachricht: Der CAN-Nachricht wird die CAN-ID vorangestellt, in dem obigen Beispiel die ID `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 angegebenen 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) |
| $200_h$+Node-ID | RX Process Data Objects (PDO) | |

| Default CAN-ID | Service | Beschreibung in |
|---|---|---|
| $280_h$+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 |

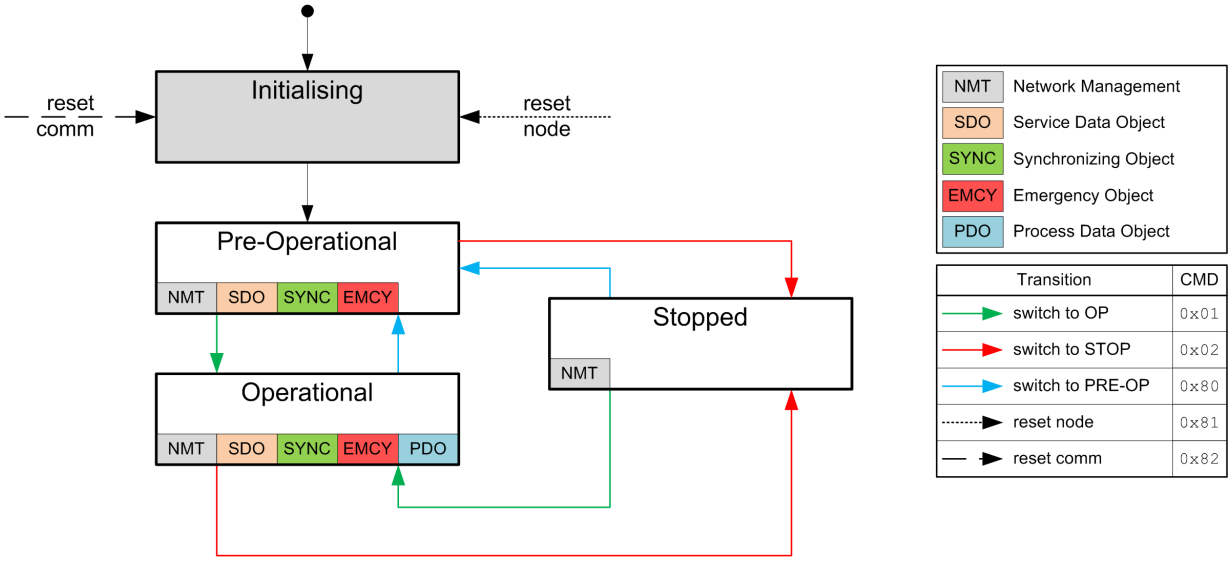
### 8.2.1 Network Management (NMT)

Das Network-Management 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, resetet 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-Befehls "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.



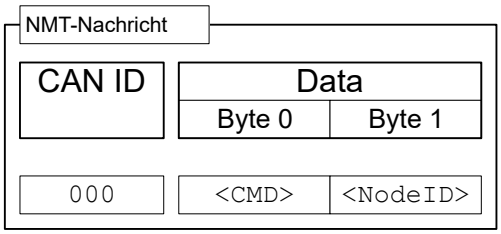

In der nachfolgenden Tabelle finden Sie eine Übersicht, welche die Aktivität der Services in den entsprechenden Zuständen darstellt.

Achten Sie darauf, 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 | |

| Service | Initializing | Pre-Operational | Operational | Stopped |
|---|---|---|---|---|
| SYNC | | aktiv | aktiv | |
| 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 |

Mit dem Befehl "Reset Node" starten Sie die Steuerung komplett neu. Mit dem Befehl "Reset Communication" setzten Sie die Einstellungen von CANopen zurück und starten die Kommunikation neu.

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.

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:

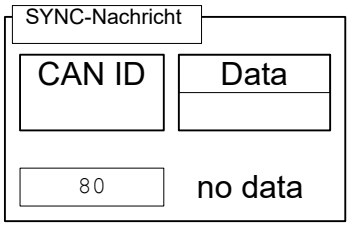
```
000 | 02 00
```

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

```
000 | 01 2A
```

### 8.2.2 Synchronisations-Objekt (SYNC)

Das Synchronisations-Objekt 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:

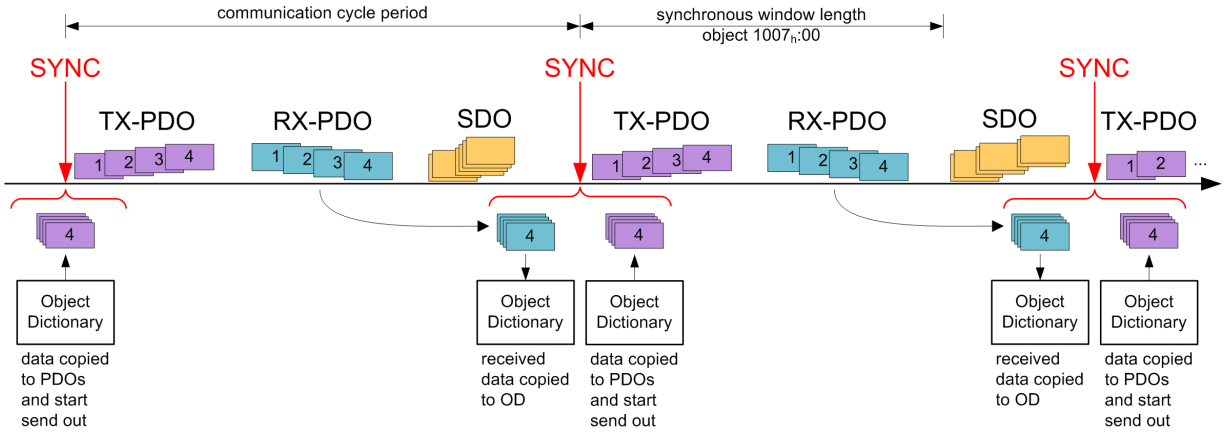



Für den SYNC-Betrieb wird in der Regel für die RX-PDOs der Übertragungsmodus (Transmission Type) 0 verwendet (Daten werden mit dem nächsten SYNC gültig), für die TX-PDOs wird ein Übertragungsmodus zwischen 1-240 gewählt. (Details: siehe Kapitel Process Data Object (PDO)).

Nach dem Erhalt einer SYNC-Nachricht gibt 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 Mikrosekunden 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 gesampelt 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.



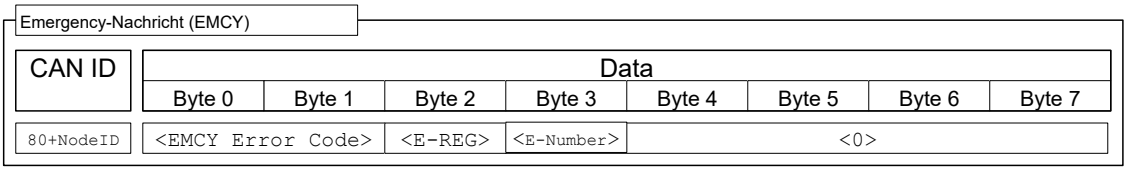

Falls der *Sync Producer* einen *Sync Counter* unterstützt, enthält die Sync-Nachricht einen zusätzlichen 1-Byte-Zählwert. Dieser Zähler erhöht sich um den Wert "1" pro gesendete Sync-Nachricht und wird jedes Mal zurückgesetzt, wenn er den Wert von 1019h Synchronous Counter Overflow Value erreicht.

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 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 einen 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`
- die "Error Number" (`E-Number`)

### 8.2.3.1 Fehlerbehandlung

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

Jeder auftretende Fehler wird folgendermaßen behandelt:

1. Das zum Fehler gehörige Bit im Objekt "Error Register" ($1001_h$) wird gesetzt.
2. Anschließend werden drei Informationen zusammen in das Objekt "Pre-defned Error Field" ($1003_h$:01) geschrieben:
   - Der *Emergency Error Code*
   - Das *Error Register*
   - Der herstellerspezifsche Fehlercode
3. Steht kein weiterer Fehler mehr an, wird folgende Nachricht verschickt:
   `80 + Node-ID | 00 00 E-REG E-Number 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 Objektverzeichnisses 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, sollten Sie mit SDO-Nachrichten nur die Konfiguration der Steuerung vornehmen. Sollte es notwendig sein, im laufenden Betrieb Daten zyklisch auszutauschen, greifen Sie auf CANopen-Nachrichten des Types PDO zurück (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.

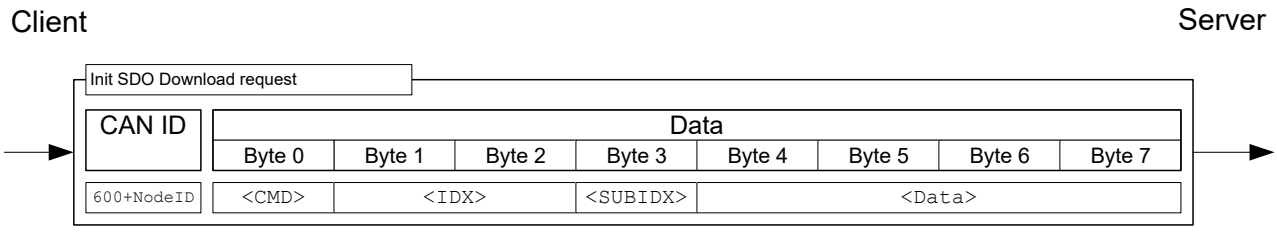
Eine SDO-Nachricht wird an die CAN-ID $600_h$ + Node-ID verschickt, die Antwort kommt mit der CAN-ID $580_h$ + Node-ID.

### 8.2.4.1 Expedited Transfer

Mit dieser Methode können Sie 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:

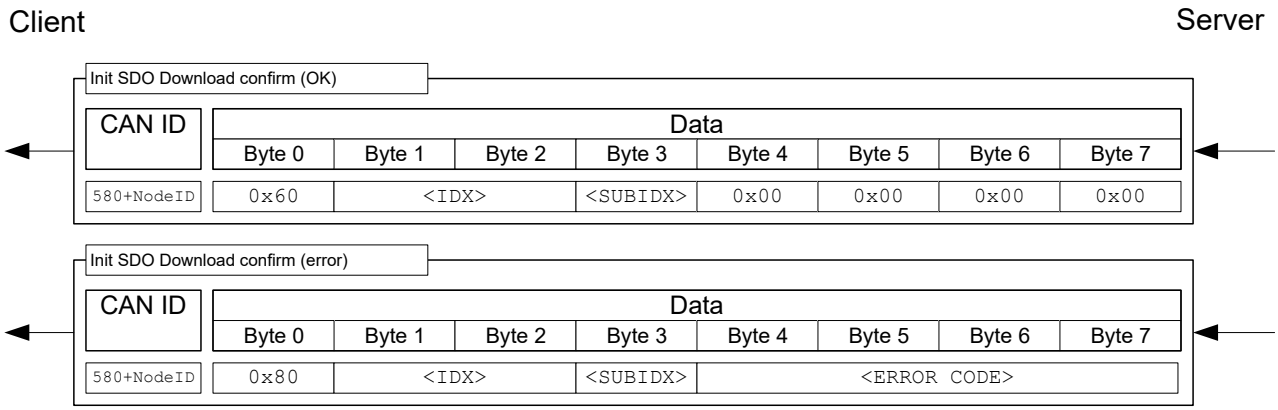



Dabei ist das Byte `<CMD>` abhängig von der Länge der Daten, welche geschrieben werden sollen. `<CMD>` kann einer der folgenden Werte sein:

- 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

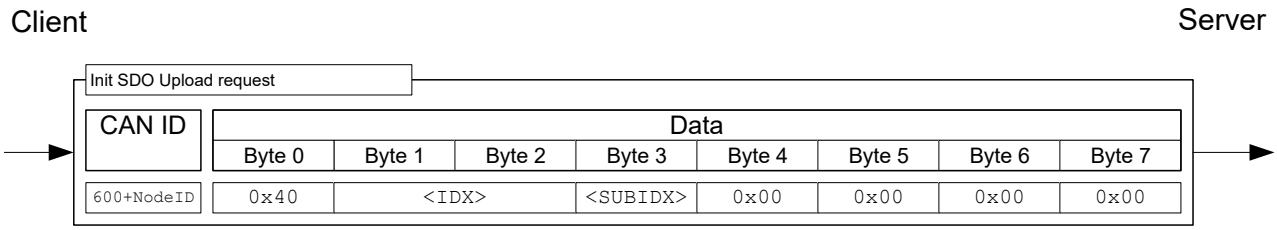
- Byte 1 ($23_h$): SDO expedited download, 4Bytes Daten (SIGNED32)
- Byte 2 und 3 ($7A_h$ $60_h$): Index des Objektes ist $607A_h$
- Byte 4 ($00_h$): Subindex des Objektes ist $00_h$
- Byte 5 bis 8 ($E8_h$ $03_h$ $00_h$ $00_h$): Wert des Objektes: $000003E8_h$

Im Erfolgsfall antwortet die Steuerung mit dieser Nachricht:

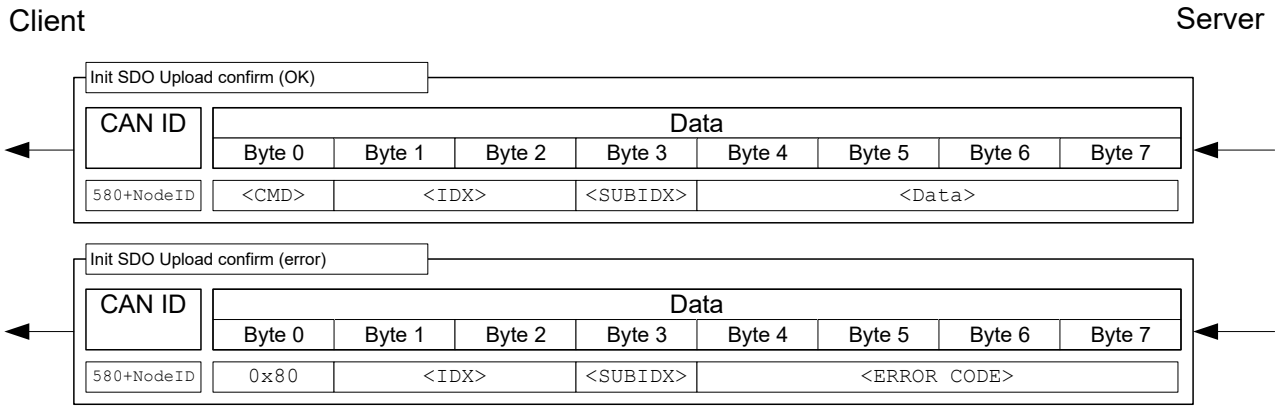
```
583 | 60 7A 60 00 00 00 00 00
```

**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.

### 8.2.4.2 Normal Transfer

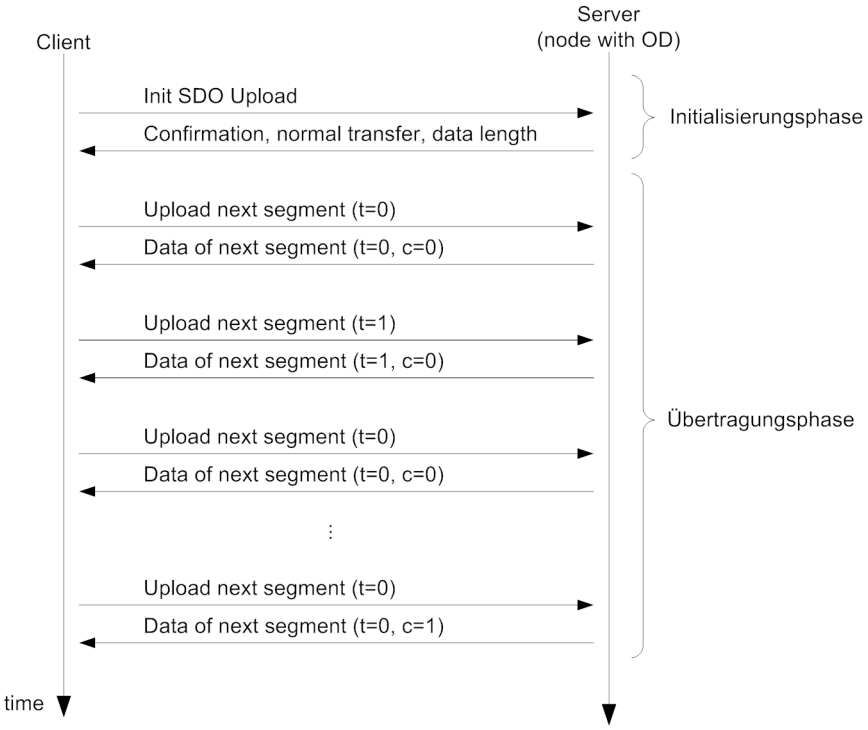
Im Gegensatz zur CANopen-Übertragung "expedited", ist der "normal transfer" nicht auf maximal vier Byte beschränkt. Bei dieser Übertragungsart wird der Inhalt mehrerer Nachrichten inhaltlich zusammengefasst, ein solcher Block an Nachrichten wird im Folgenden als "Transfer" bezeichnet. Jede Nachricht innerhalb eines Transfers wird dabei einzeln bestätigt.

**HINWEIS**

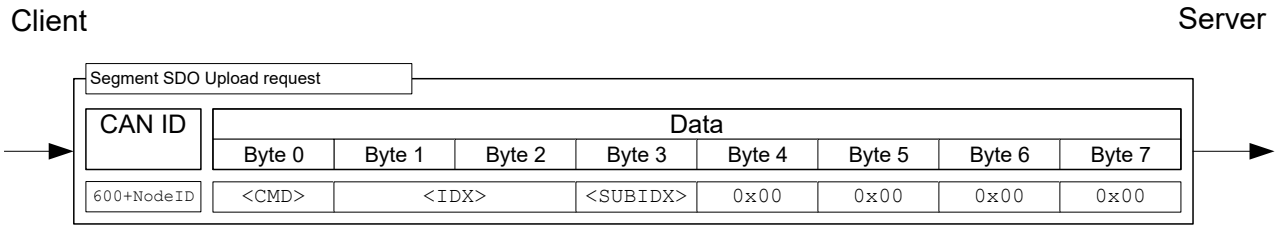
Wenn Ihr CANopen-Master den "normal transfer" nicht unterstützt, können Sie auf Objekte mit dem Datentyp String auch anders zugreifen: Jeder String kann mit einem SDO-Upload auf den Subindex 1 und die folgenden Subindizes zeichenweise ausgelesen werden.

#### SDO-Upload

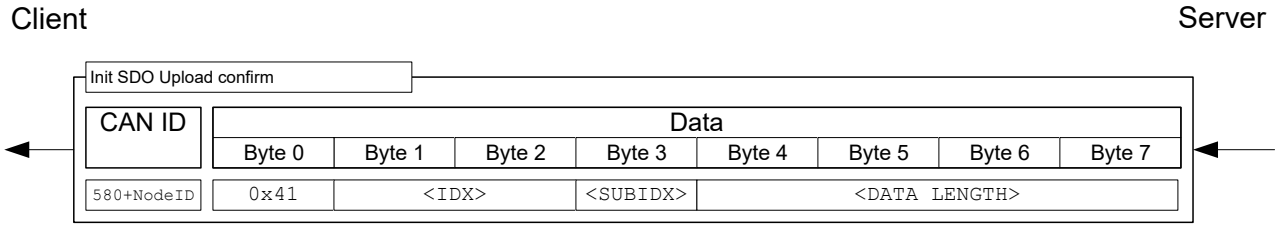
In nachfolgender Abbildung ist die Vorgehensweise eines "SDO-Uploads" dargestellt (Client lässt sich den Inhalt eines Objektes schicken). Die Übertragung zerfällt in zwei Phasen: Einer Initialisierungs- und einer Übertragungsphase.



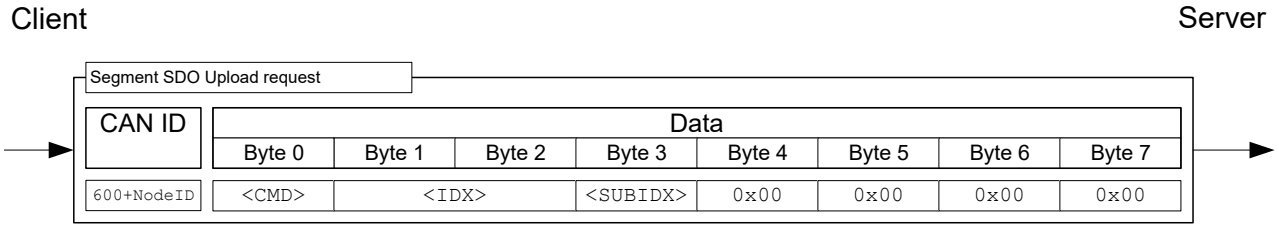
Der Upload beginnt, indem der Client - wie bei einem "expedited transfer" auch - ein "Init SDO Update" an den Server schickt (siehe nachfolgende Abbildung).



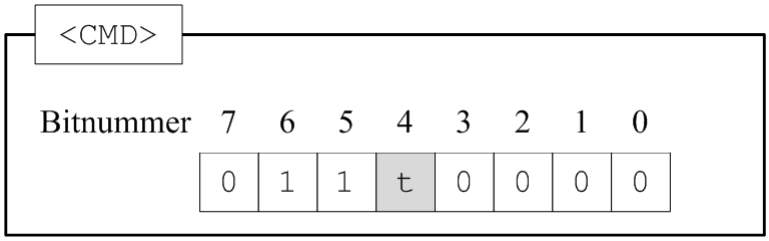

Die Antwort für einen "normal transfer" enthält die Menge der zu empfangenen Bytes nicht im <CMD> codiert, sondern im Datenbereich eingetragen wie es in der nachfolgenden Abbildung im Bereich <DATA LENGTH> zu sehen ist.




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

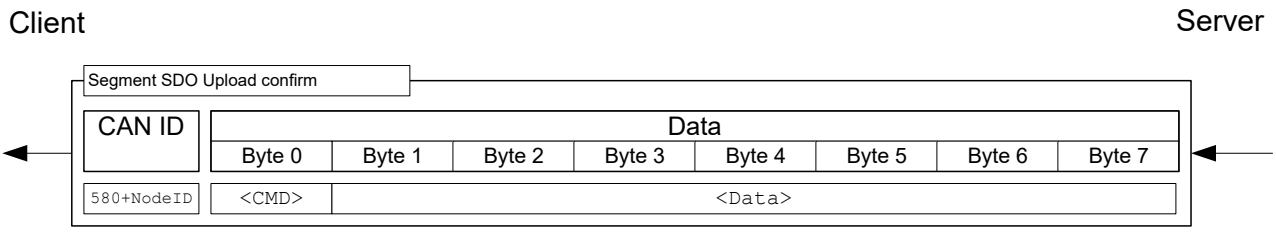



Das Byte 0 mit dem Kommando <CMD> setzt sich folgendermaßen zusammen:

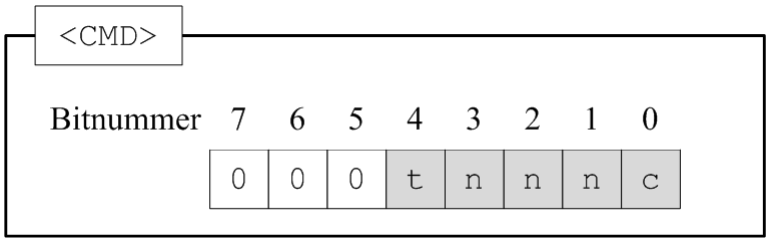



Das Bit mit der Bezeichnung `t` alterniert mit jeder Anforderung ("toggle bit"). Es beginnt mit jedem Transfer bei 0, auch wenn der vorherige Transfer abgebrochen wurde.

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 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 |

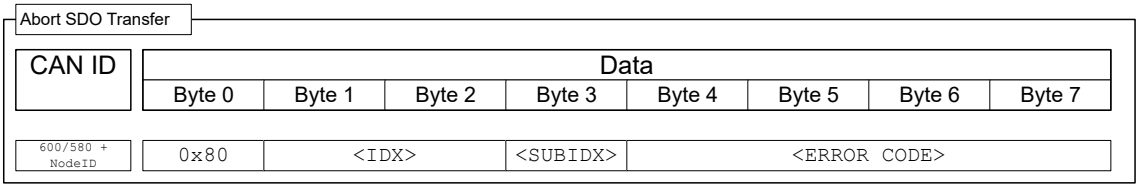
| COB-ID | Daten | Beschreibung |
|---|---|---|
| $583_h$ | 41 0A 10 00 11 00 00 00 | Init Upload; Size: indicated; transfer type: normal; Num of bytes: 17; Index: $100A_h$; Subindex: 00 |
| $603_h$ | 60 0A 10 00 00 00 00 00 | Upload Segment Req.; Toggle bit: not set |
| $583_h$ | 00 46 49 52 2D 76 31 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 berechtigt, den derzeitigen Transfer abzubrechen. 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.

### 8.2.4.3 SDO-Fehlermeldungen

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

| Error Code | Beschreibung |
|---|---|
| $05030000_h$ | *toggle bit not changed*: Gültig nur bei "normal transfer" oder "block transfer". Das Bit, welches nach jeder Übertragung zu alternieren hat, hat seinen Zustand nicht geändert. |
| $05040001_h$ | *command specifier unknown*: Das Byte 0 des Datenblocks enthielt einen nicht zulässigen Befehl. |
| $06010000_h$ | *unsupported access*: Falls über CAN over EtherCAT (CoE) ein "complete access" angefordert wurde (wird nicht unterstützt.) |
| $06010002_h$ | *read only entry*: Es wurde versucht, auf ein konstantes oder nur lesbares Objekt zu schreiben. |
| $06020000_h$ | *object not existing*: Es wurde versucht, auf ein nicht vorhandenes Objekt zuzugreifen (Index fehlerhaft). |
| $06040041_h$ | *objekt cannot be pdo mapped*: Es wurde versucht, ein Objekt in das PDO zu mappen, für welches das nicht zulässig ist. |

| Error Code | Beschreibung |
|---|---|
| $06040042_h$ | *mapped pdo exceed pdo*: Würde das gewünschte Objekt in das PDO-Mapping angehängt werden, würden die 8Byte des PDO-Mappings überschritten. |
| $06070012_h$ | *parameter length too long*: 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>=$2F_h$. |
| $06070013_h$ | *parameter length too short*: 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>=$23_h$. |
| $06090011_h$ | *subindex not existing*: Es wurde versucht, auf einen ungültigen Subindex eines Objektes zuzugreifen, der Index hingegen würde existieren. |
| $06090031_h$ | *value too great*: 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. |
| $06090032_h$ | *value too small*: Einige Objekte unterliegen Restriktionen in der Größe des Wertes. In diesem Fall wurde versucht, einen zu niedrigen Wert in das Objekt zu schreiben. |
| $08000000_h$ | *general error*: Allgemeiner Fehler, der in keine andere Kategorie passt. |
| $08000022_h$ | *data cannot be read or stored in this state*: Die Parameter des PDOs dürfen nur im State "Stopped" 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 (=64 Bit) 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 Sende- und Empfangs-Konfiguration. Die Empfangs-Konfiguration beschreibt die Verarbeitung für PDO-Nachrichten, die empfangen werden, und die Sende-Konfiguration der zusendenden PDO-Nachrichten.

### 8.2.5.1 RX-Konfiguration

Um ein RX-PDO zu konfigurieren, müssen Sie drei Objektkategorien im Objektverzeichnis berücksichtigen:

- Die Objekte, welche die Funktionalität des Mappings beschreiben.
- Die Objekte, welche den Inhalt des Mappings beschreiben.
- 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.

| Mapping | CAN-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$_h$ |
| 1605$_h$ | xxx$_h$ |
| 1606$_h$ | xxx$_h$ |
| 1607$_h$ | xxx$_h$ |

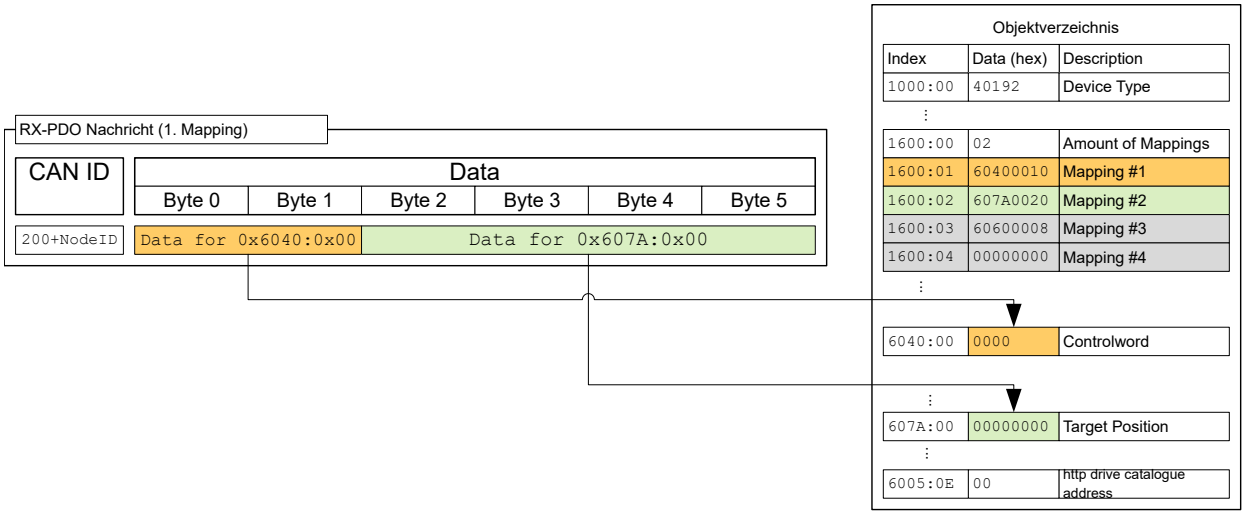
  Die Änderung einer COB-ID wird aktiv erst *nach* dem Neustart der Steuerung oder der Kommunikation (siehe Network Management (NMT)).
- Subindex 2 (transmission type): In diesem Subindex wird eine Nummer hinterlegt, die den Zeitpunkt definiert, zu dem die empfangenen Daten gültig werden. Die Nummer und 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$_h$Bit) in das Objekt 607A$_h$:00$_h$




### Dummy-Objekte

Sie können RX-PDOs so konfigurieren, dass mehr als ein Teilnehmer darauf reagiert. In diesem Fall kann es gewünscht sein, nur einen Teil der im PDO enthaltenen Daten in einem der Geräte auszuwerten. Für lokal nicht genutzte Daten können Sie ein Dummy-Objekt von einem der unterstützten Datentypen in das Mapping des PDOs eintragen:

| Index | Datentyp |
|---|---|
| $0002_h$ | INTEGER8 |
| $0003_h$ | INTEGER16 |
| $0004_h$ | INTEGER32 |
| $0005_h$ | UNSIGNED08 |
| $0006_h$ | UNSIGNED16 |
| $0007_h$ | UNSIGNED32 |

## 8.2.5.2 TX-Konfiguration

Um ein TX-PDO zu konfigurieren, müssen Sie drei Objektkategorien im Objektverzeichnis berücksichtigen:

- Die Objekte, welche die Funktionalität des Mappings beschreiben.
- Die Objekte, welche den Inhalt des Mappings beschreiben.
- 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.

| Mapping | CAN-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 |
| $1A04_h$ | $xxx_h$ |
| $1A05_h$ | $xxx_h$ |
| $1A06_h$ | $xxx_h$ |
| $1A07_h$ | $xxx_h$ |

Die Änderung einer COB-ID wird aktiv erst *nach* dem Neustart der Steuerung oder der Kommunikation (siehe Network Management (NMT)).

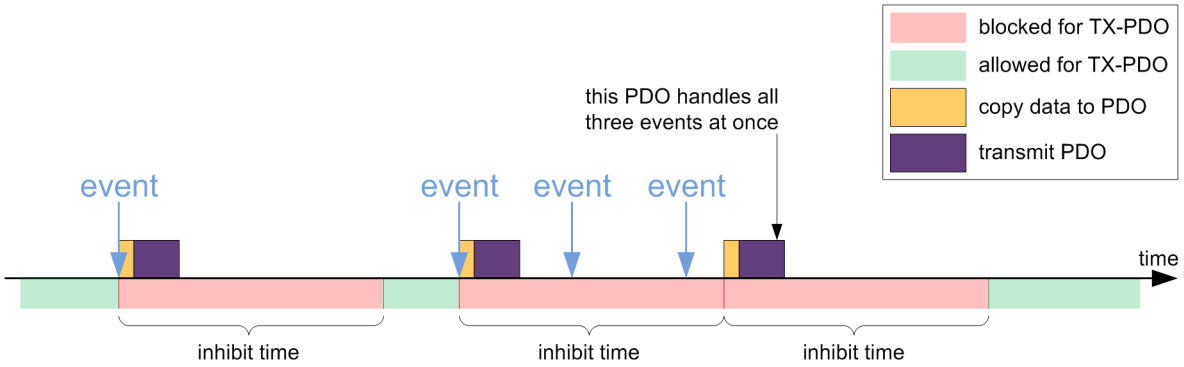
- Subindex 2 (transmission type): In diesem Subindex wird eine Nummer hinterlegt, welche den Zeitpunkt definiert, zu dem die Daten in die PDO-Nachricht kopiert und wann dieses gesendet werden soll. Die Nummer und die zugehörige Bedeutung kann aus der nachfolgenden Tabelle entnommen werden.
  Im Folgenden 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 *Event* 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 *Events* 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 ein *Event* auszulösen, welcher für das Kopieren der Daten und Senden des PDOs sorgt.

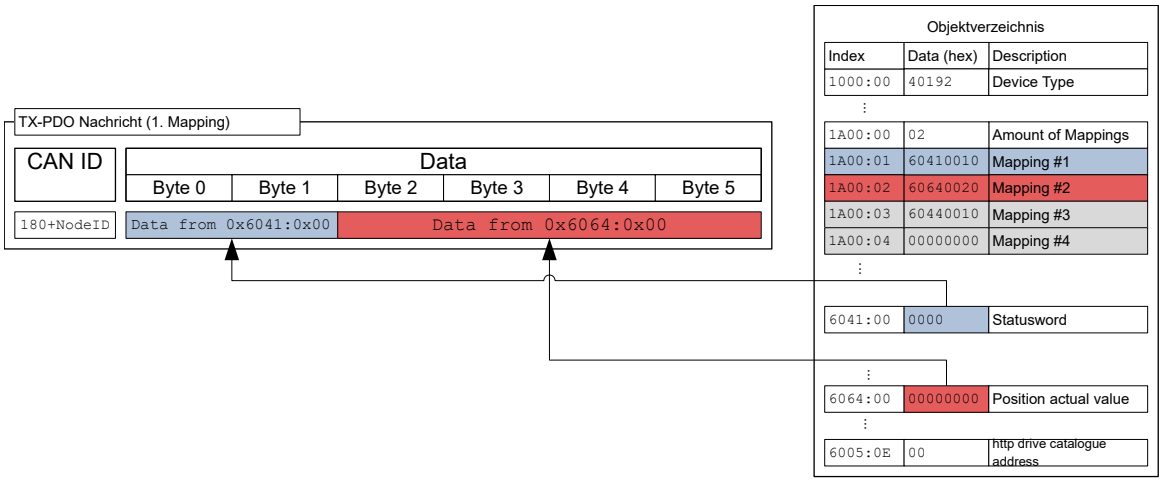
- Subindex 6 (sync start value): hier wird der Startwert des *Sync Counter* eingetragen, ab welchem der *Slave* auf den Sync zum ersten Mal reagieren und das PDO senden soll. Wird global erst aktiviert, wenn in 1019h Synchronous Counter Overflow Value ein Wert größer 1 eingestellt wird.




### 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 TX-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 TX-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.




## 8.2.5.3 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)

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 $60FF_h$:$00_h$ (target velocity)

**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): $606C_h$:$00_h$ (velocity actual value)

4. Mapping (CAN-ID: $480_h$ + Node-ID): Objekt $60FD_h$:$00_h$ (Digital Inputs)

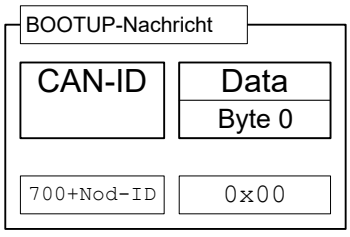
### 8.2.5.4 PDO-Mapping ändern

Sie ändern das PDO-Mapping im NMT-Zustand "Pre-operational" wie folgt:

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, indem 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.
6. Speichern sie die Konfiguration, indem Sie den Wert "65766173" in $1010_h$:$02_h$ schreiben.

## 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

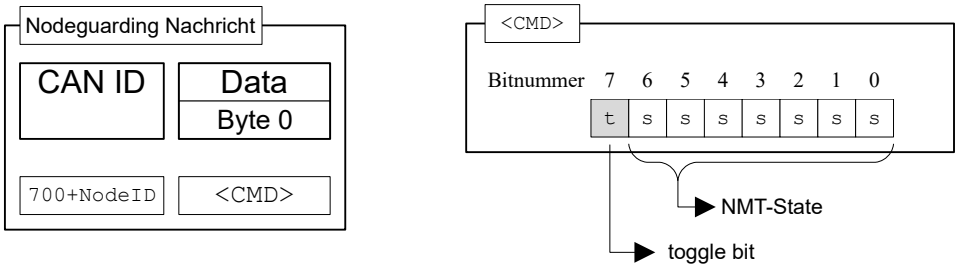
Mit den Services "Heartbeat" und "Nodeguarding" (oft auch mit "Liveguarding" bezeichnet) lassen sich abgeschaltete oder abgestürzte Geräte am CAN-Bus finden. Dazu fordert der NMT-Master zyklisch eine Nachricht mit dem aktuellen NMT-Zustand des Slaves an (Nodeguarding).

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.

#### 8.2.7.1 Nodeguarding

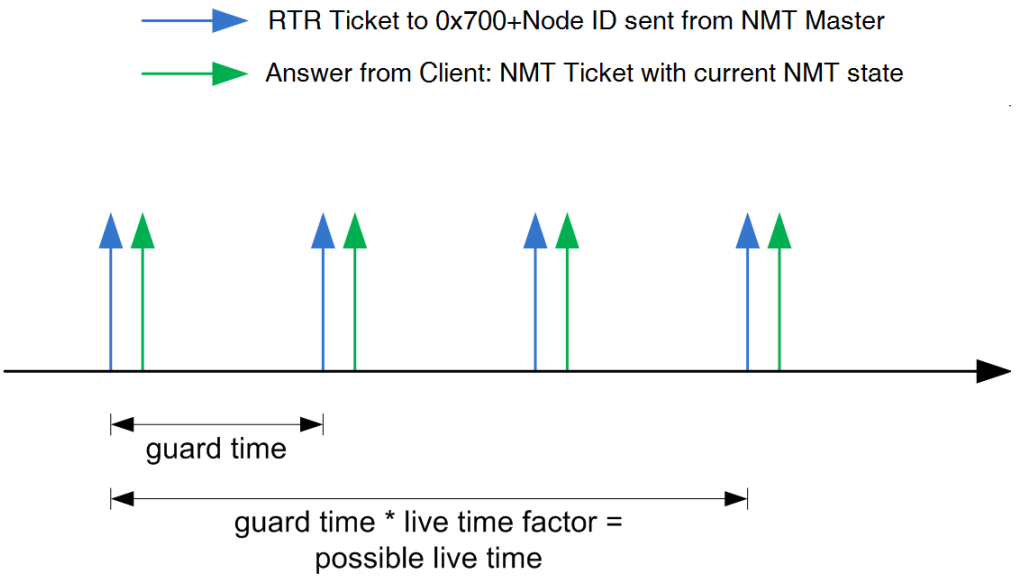
Dieser Service basiert darauf, dass der NMT-Master eine RTR-Nachricht mit der CAN-ID $700_h$ + Node-ID an den jeweiligen Slave verschickt.

Anschließend muss der Slave eine Nachricht als Antwort verschicken, welche nachfolgend abgebildet ist. Das Bit 7 wechselt 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):

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

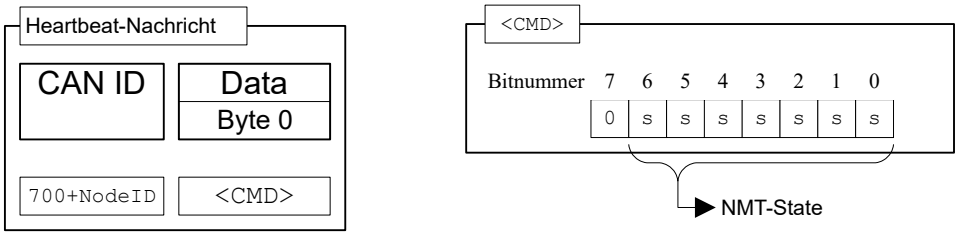



Folgende drei Bedingungen werden beim Nodeguarding geprüft:

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

### 8.2.7.2 Heartbeat

Ist der Heartbeat aktiviert, sendet der Slave ohne weitere Aufforderung zyklisch seinen NMT-Zustand auf dem CAN-Bus. Sie aktivieren diesen Service, indem Sie die Zeit *Producer Heartbeat Time* im Objekt $1017_h$:$00_h$ auf einen anderen Wert als Null setzen. 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 Sie die Zeit *Consumer Heartbeat Time* und die Node-ID des *Producer* im Objekt $1016_h$ eintragen.

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 1018h 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

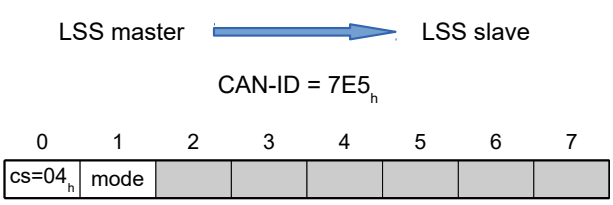
#### 8.3.3.1 Switch state services

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 :**

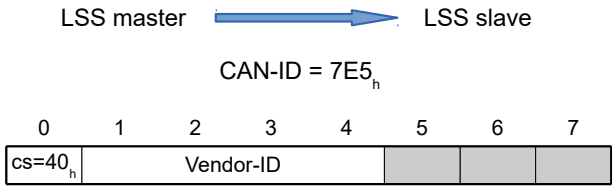
reserviert (=$0_h$)

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




**Byte 0 : CS (Command Specifier)**

Wert = "$40_h$"

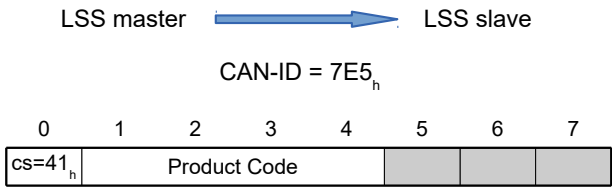
**Bytes 1-4: Vendor-ID**

*Vendor-ID*: siehe $1018_h$:$01_h$

**Bytes 5-7 :**

reserviert (=$0_h$)

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




**Byte 0 : CS (Command Specifier)**

Wert = "$41_h$"

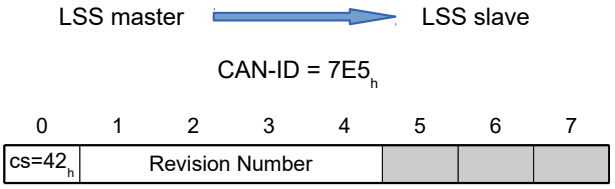
**Bytes 1-4: Product Code**

*Product Code*: siehe $1018_h$:$02_h$

**Bytes 5-7 :**

reserviert (=$0_h$)

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




**Byte 0 : CS (Command Specifier)**

Wert = "$42_h$"

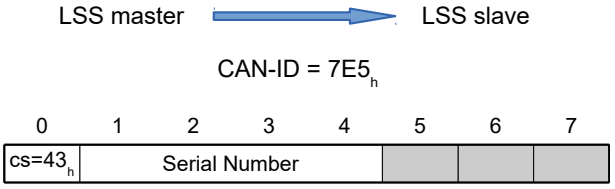
**Bytes 1-4: Revision Number**

*Revision Number*: siehe $1018_h$:$03_h$

**Bytes 5-7 :**

reserviert (=$0_h$)

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




**Byte 0 : CS (Command Specifier)**

Wert = "$43_h$"

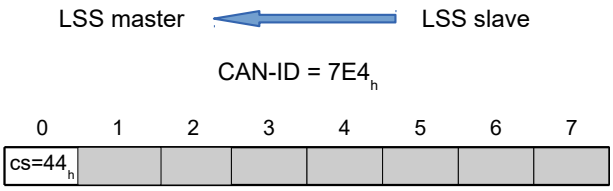
**Bytes 1-4: mode**

*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 (=$0_h$)

### 8.3.3.2 Configuration services

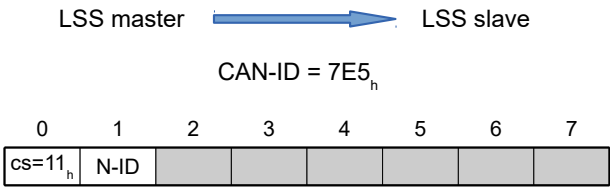
Mit diesen Diensten kann der *LSS-Master* die Node-ID oder Baudrate der *LSS-Slaves* ändern und ggf. abspeichern.

**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 = "$11_h$"

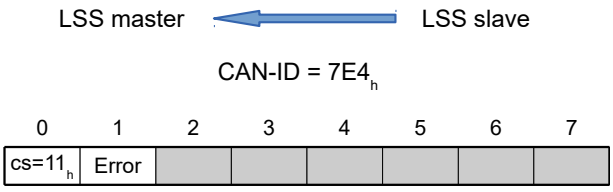
**Byte 1: N-ID (Node-ID)**

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

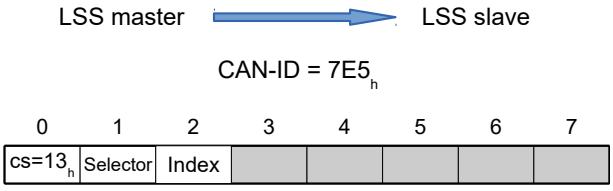
Wert = "$01_h$": ungültige Node-ID

**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.

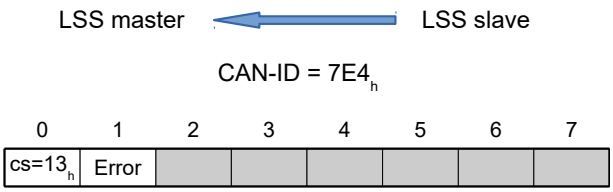
**Bytes 3-7 :**

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

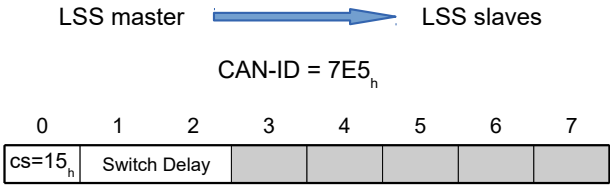
Wert = "$01_h$": ungültiger *Table Index*/ Baudrate wird nicht unterstützt

**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.

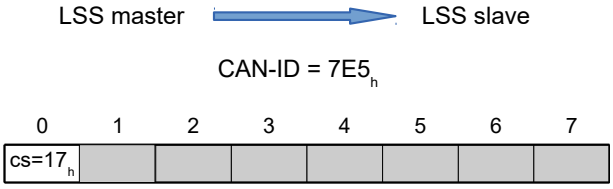
Die gleiche Zeit wird ein zweites Mal abgewartet, erst danach darf ein *LSS-Slave* wieder Nachrichten senden.

**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.



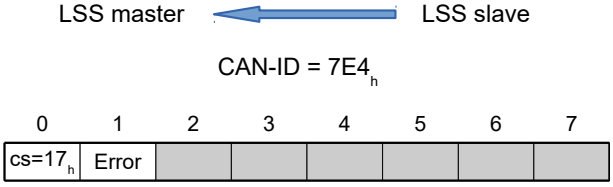

**Byte 0 : CS (Command Specifier)**

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 (=$0_h$)

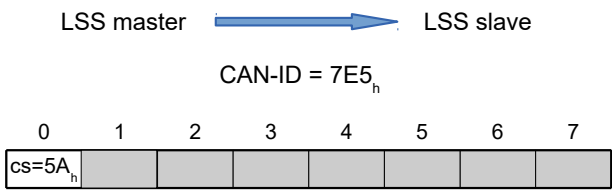
### 8.3.3.3 Inquiry services

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:



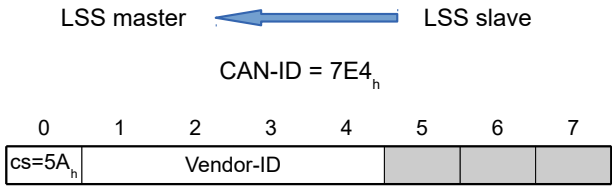

**Byte 0 : CS (Command Specifier)**

Wert = "$5A_h$"

**Bytes 1-7 :**

reserviert (=$0_h$)

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




**Byte 0 : CS (Command Specifier)**

Wert = "$5A_h$"

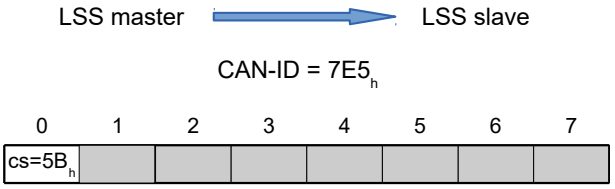
**Bytes 1-4: Vendor-ID**

*Vendor-ID*: siehe $1018_h$:$01_h$

**Bytes 5-7 :**

reserviert (=$0_h$)

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



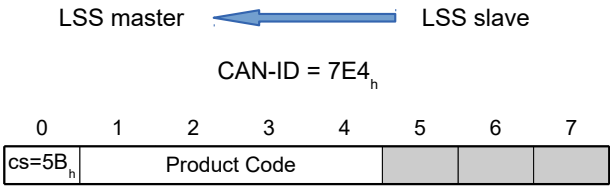

**Byte 0 : CS (Command Specifier)**

Wert = "$5B_h$"

**Bytes 1-7 :**

reserviert (=$0_h$)

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




**Byte 0 : CS (Command Specifier)**

Wert = "$5B_h$"

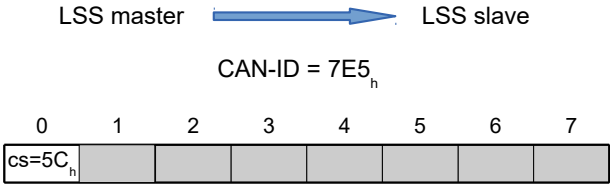
**Bytes 1-4: Produt Code**

*Product Code*: siehe $1018_h$:$02_h$

**Bytes 5-7 :**

reserviert (=$0_h$)

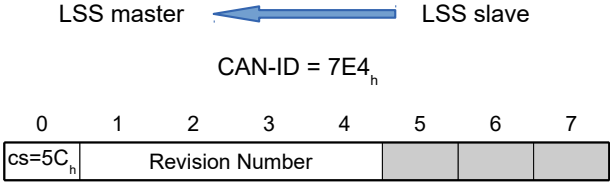
**3.** Der *LSS-Master* fragt die *Revision Number* ab:




**Byte 0 : CS (Command Specifier)**

Wert = "$5C_h$"

Der *LSS-Slave* sendet seine *Revision Number* zurück:




**Byte 0 : CS (Command Specifier)**

Wert = "$5C_h$"

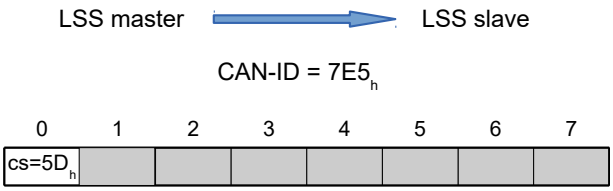
**Bytes 1-4: Revision Number**

*Revision Number*: siehe $\underline{1018_h}$:$03_h$

**Bytes 5-7 :**

reserviert (=$0_h$)

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



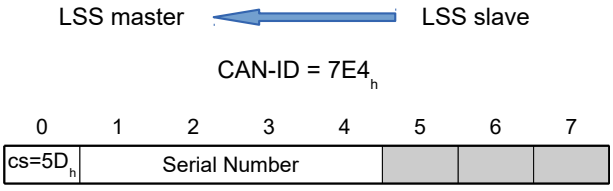

**Byte 0 : CS (Command Specifier)**

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 $\underline{1018_h}$:$04_h$

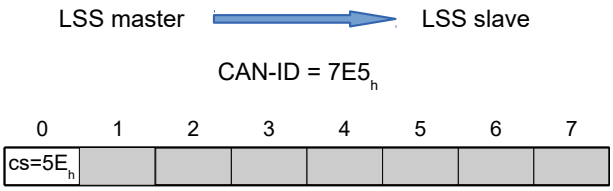
**Bytes 5-7 :**

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:



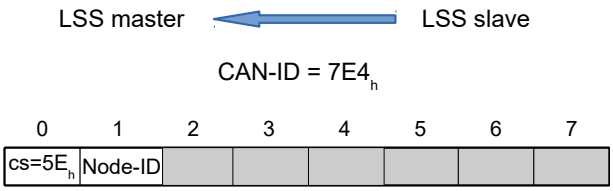

**Byte 0 : CS (Command Specifier)**

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 (=$0_h$)

### 8.3.3.4 Identification services

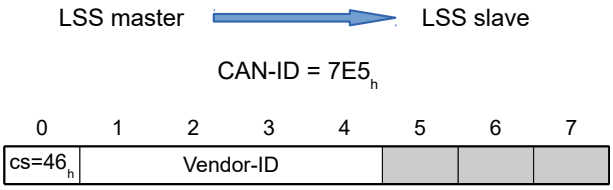
Mit diesen Diensten kann der *LSS-Master* die *LSS-Slaves* anhand ihrer *LSS-Adresse* sich identifizieren lassen.

### 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 = "$46_h$"

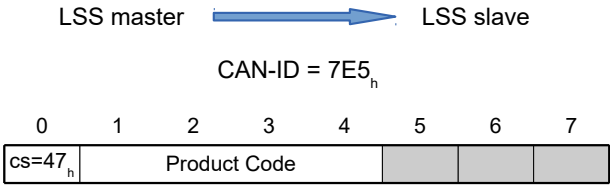
**Bytes 1-4: Vendor-ID**

*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 = "$47_h$"

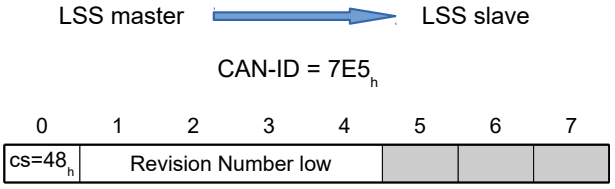
**Bytes 1-4: Product Code**

*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)**

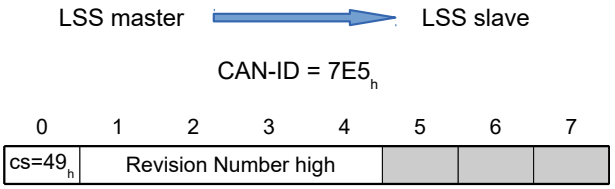
Wert = "$48_h$"

**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 = "$49_h$"

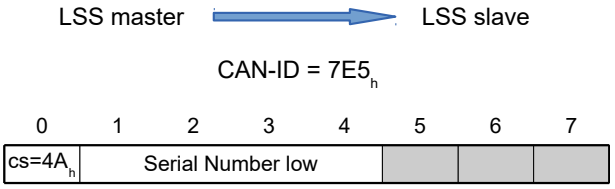
**Bytes 1-4: Revision Number high**

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)**

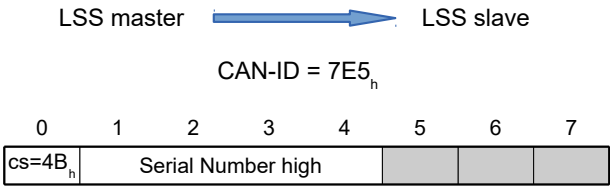
Wert = "$4A_h$"

**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**

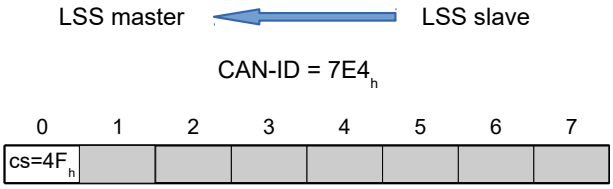
höchste *Serial Number* des Bereichs: siehe $1018_h$:$04_h$

**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 Programmierung mit NanoJ

*NanoJ* ist eine *C*- bzw. *C++*-nahe Programmiersprache. NanoJ ist in der Software *Plug & Drive Studio 3* integriert. Weiterführende Informationen finden Sie im Dokument *Plug & Drive Studio 3: User Manual* auf www.nanotec.de.

## 9.1 NanoJ-Programm

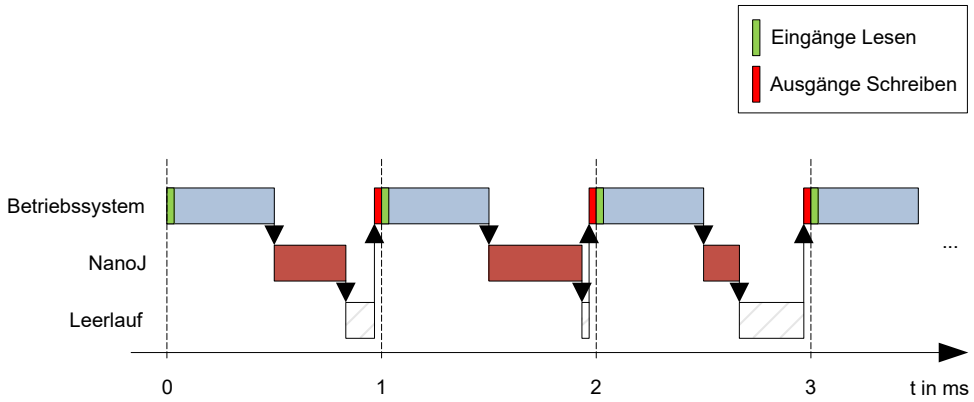
Ein *NanoJ-Programm* stellt eine geschützte Ausführungsumgebung innerhalb der Firmware zur Verfügung. In dieser kann der Anwender eigene Abläufe anlegen. Diese können dann Funktionen in der Steuerung auslösen, indem beispielsweise Einträge im Objektverzeichnis gelesen oder geschrieben werden.

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, sofern Sie Bit 0 im Objekt $2300_h$ nicht auf "0" setzen.

### 9.1.1 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, steht dem Benutzerprogramm (abhängig von Betriebsart und Anwendungsfall) nur ein Teil der 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.

Damit das *NanoJ-Programm* nicht angehalten wird, können Sie den *AutoYield*-Modus aktivieren, indem Sie den Wert "5" in $2300_h$ schreiben. Im *AutoYield*-Modus ist aber das *NanoJ-Programm* nicht mehr echtzeitfähig und läuft nicht mehr im 1-Millisekunde-Takt.

## 9.1.2 Geschützte Ausführungsumgebung

Durch prozessorspezifische Eigenschaften wird eine sogenannte *Geschützte Ausführungsumgebung* generiert. Ein Benutzerprogramm in der geschützten Ausführungsumgebung 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.

## 9.1.3 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 NanoJ-Funktionen
- Aufruf sonstiger NanoJ-Funktionen

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 *Plug & Drive Studio* gesteuert, ob eine Variable im Input-, Output- oder Datenbereich abgelegt wird.

### NanoJ-Inputs und NanoJ-Outputs

Um mit dem *NanoJ-Programm* über die jeweilige Schnittstelle zu kommunizieren, können Sie folgende Objekte benutzen:

- 2400h NanoJ Inputs: Array mit zweiunddreißig S32-Werten zum Übergeben von Werten an das *NanoJ-Programm*
- 2500h NanoJ Outputs: Array mit zweiunddreißig S32-Werten, wo das *NanoJ-Programm* Werte ablegen kann, die über den Feldbus ausgelesen werden können

## 9.1.4 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 NanoJ-Funktionen 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 NanoJ-Funktion zuzugreifen.

Eine Liste verfügbarer NanoJ-Funktionen findet sich im Kapitel NanoJ-Funktionen im NanoJ-Programm.

**TIPP**

Nanotec empfiehlt, entweder per Mapping oder NanoJ-Funktion mit `od_write()` auf ein und denselben OD-Wert zuzugreifen. Wird beides gleichzeitig verwendet, so hat die NanoJ-Funktion keine Auswirkung.

### 9.1.5 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:

- Umbenennen der Datei `TEST1.USR` in `nanoj*.usr`.
- Kopieren der Datei `nanoj*.usr` über USB auf die Steuerung und starten Sie diese neu.
- NanoJ-Programm starten durch Beschreiben von Objekt $2300_h$, Bit 0 = "1".
- Überprüfen des Eintrags $2302_h$ auf Fehlercode und des Objekts $2301_h$, Bit 0 = "1" (NanoJ-Programm läuft).

Um ein laufendes Programm anzuhalten: Beschreiben des Eintrags $2300_h$ mit dem Bit 0 Wert = "0".

### 9.1.6 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 implementieren.

**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 führt zu einem Fehler beim Kompilieren:

```
unsigned int i = 1;
 void user() {
 i += 1;
}
```

### 9.1.7 NanoJ-Programmbeispiel

Das Beispiel zeigt das Programmieren eines Rechtecksignals in das Objekt $2500_h$:$01_h$.

```
// 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.

## 9.2 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.

**TIPP**

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.

### 9.2.1 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.

**HINWEIS**

Ein Kommentar ist nur oberhalb der jeweiligen Mapping-Deklaration im Code erlaubt, nicht in derselben Zeile.

### 9.2.2 Beispiel eines Mappings

Beispiel eines Mappings und der zugehörigen Variablenzugriffe:

```
// 6040h:00h is UNSIGNED16
map U16 controlWord as output 0x6040:00
// 6041h:00h is UNSIGNED16
map U16 statusWord as input 0x6041:00

// 6060h:00h is SIGNED08 (INTEGER8)
map S08 modeOfOperation as inout 0x6060:00

#include "wrapper.h"

void user()
{
  [...]
  Out.controlWord = 1;
  U16 tmpVar = In.statusword;
```

```
  InOut.modeOfOperation = tmpVar;
  [...]
}
```

### 9.2.3 Möglicher Fehler bei `od_write()`

Eine mögliche Fehlerquelle ist ein schreibender Zugriff mittels der Funktion `od_write()` (siehe NanoJ-Funktionen 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 $6040_h$:$00_h$.
2. Am Ende des 1 ms-Zyklus wird das Mapping geschrieben, welches ebenfalls das Objekt $6040_h$:$00_h$ beschreibt, allerdings mit dem Wert `1`.
3. Somit wird - aus Sicht des Benutzers - der `od_write`-Befehl wirkungslos.

## 9.3 NanoJ-Funktionen im NanoJ-Programm

Mit NanoJ-Funktionen ist es möglich, in der Firmware eingebaute Funktionen direkt aus einem Benutzerprogramm aufzurufen. Eine direkte Code-Ausführung ist nur in dem geschützten Bereich der geschützten Ausführungsumgebung 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 geschützten Ausführungsumgebung zuzulassen. Der Entwickler des Benutzerprogramms muss sich jedoch um diesen Mechanismus nicht kümmern - für ihn sind die NanoJ-Funktionen wie ganz normale C-Funktionen aufrufbar. Lediglich die Datei *wrapper.h* muss - wie üblich - eingebunden werden.

### 9.3.1 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(); }
```

### 9.3.2 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 |

## 9.4 Einschränkungen und mögliche Probleme

Im Folgenden werden Einschränkungen und mögliche Probleme bei der Arbeit mit NanoJ aufgelistet:

| Einschränkung/Problem | Maßnahme |
|---|---|
| Wenn ein Objekt gemappt wird, z. B. 0x6040, wird das Objekt alle 1 ms auf seinen vorherigen Wert zurückgesetzt. Das macht die Steuerung dieses Objekts über den Feldbus oder das *Plug & Drive Studio* unmöglich. | Greifen Sie stattdessen mit `od_read` / `od_write` auf das Objekt zu. |
| Wenn ein Objekt als Output gemappt wurde und der Wert des Outputs niemals vor dem Start des *NanoJ-Programms* festgelegt wird, kann der Wert dieses Objekts zufällig sein. | Initialisieren Sie die Werte der gemappten Outputs in Ihrem NanoJ-Programm, damit es sich deterministisch verhält. |
| Die Array-Initialisierung darf nicht mit mehr als 16 Einträgen verwendet werden. | Verwenden Sie stattdessen `constant array`. |
| Zu viele lokale Variablen und Arrays innerhalb von Funktionen können zu einem Stacküberlauf führen. | Deklarieren Sie die Variablem global. Der Speicherbedarf wird bereits beim Kompilieren überwacht, es kommt nicht zu Fehlern zur Laufzeit. |

| Einschränkung/Problem | Maßnahme |
|---|---|
| Zu tief verschachtelte Funktionen können zu einem Stacküberlauf führen. | Maximale Verschachtelungstiefe von 2 beachten. |
| `float` darf nicht mit Vergleichsoperatoren verwendet werden. | Verwenden Sie stattdessen `int`. |
| `double` darf nicht verwendet werden. | |
| Wenn ein NanoJ-Programm den Controller neu startet (entweder direkt durch einen expliziten Neustart oder indirekt, z. B. durch die Verwendung der Reset-Funktion), könnte der Controller in eine Neustartschleife geraten, der man nur schwer oder gar nicht entkommen kann. | |
| `math` oder `cmath` können nicht einbezogen werden. | |

# 10 Objektverzeichnis Beschreibung

## 10.1 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

## 10.2 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"

## 10.3 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 beschreiben 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 dar 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.

## 10.4 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.

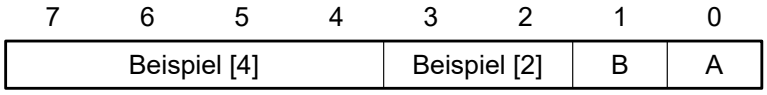
**Vorgabewert**

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.

## 10.5 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.



**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

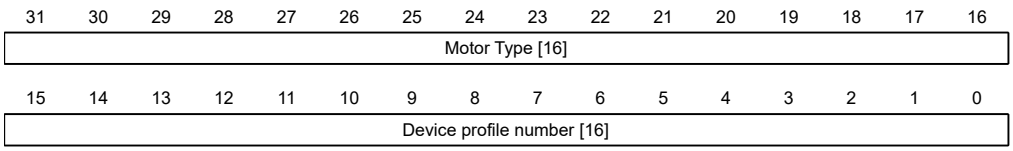
### Funktion

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 "2": BLDC-Motor
- Bit 23 bis Bit 16: Wert "4": Schrittmotor

- Bit 23 bis Bit 16: Wert "6": Sowohl Schrittmotor als auch BLDC-Motor

**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/werden das/die entsprechende(n) Fehlerbit(s) gesetzt. Sollte der Fehler nicht mehr bestehen, wird es automatisch wieder gelöscht.

**HINWEIS**

Für jeden aufgetretenen Fehler wird im Objekt $1003_h$ ein genauerer Fehlercode hinterlegt.

### 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

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| MAN | RES | PROF | COM | TEMP | VOL | CUR | GEN |

**GEN**

Genereller Fehler, wird immer gesetzt im Fehlerfall

**CUR**

Strom

**VOL**

Spannung

**TEMP**

Temperatur

**COM**
Kommunikation

**PROF**
Betrifft das Geräteprofil

**RES**
Reserviert, immer "0"

**MAN**
Hersteller-spezifisch

## 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 | 1st Standard Error Field |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | 2nd Standard Error Field |

| | |
|---|---|
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | 3th Standard Error Field |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | 4th Standard Error Field |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | 5th Standard Error Field |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | 6th Standard Error Field |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Standard Error Field |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |

| | |
|---|---|
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $08_h$ |
| Name | 8th Standard Error Field |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

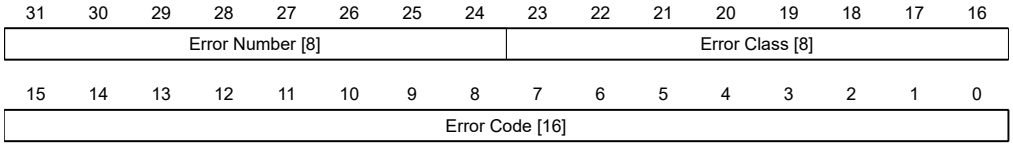
## Beschreibung

### 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 (+Ub) zu hoch |
| 2 | Ausgangsstrom zu hoch |
| 3 | Eingangsspannung (+Ub) zu niedrig |
| 4 | Fehler am Feldbus |
| 6 | Nur CANopen: NMT-Master braucht zu lange, um Nodeguarding-Anforderung zu schicken |
| 7 | Sensor 1 (siehe $3204_h$): Fehler durch elektrische Störung oder defekte Hardware |
| 8 | Sensor 2 (siehe $3204_h$): Fehler durch elektrische Störung oder defekte Hardware |
| 9 | Sensor 3 (siehe $3204_h$): Fehler durch elektrische Störung oder defekte Hardware |
| 10 | Warnung: Positiver Endschalter überschritten |
| 11 | Warnung: Negativer Endschalter überschritten |

| Fehlernummer | Beschreibung |
|---|---|
| 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 | Sensor n (siehe $3204_h$), wo n größer 3: Fehler durch elektrische Störung oder defekte Hardware |
| 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 ($6075_h$) |
| 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 |
| 32 | Interner Fehler: Korrekturfaktor für Referenzspannung fehlt im OTP |
| 35 | *STO Fault*: STO wurde angefordert, aber nicht über beide STO-Eingänge. |
| 36 | *STO Changeover*: STO wurde angefordert, aber nicht über beide STO-Eingänge. Dauert dieser Zustand länger als 100 ms, wechselt die Steuerung in den Zustand *STO Fault*. |
| 37 | *STO Active*: STO ist aktiv, es wird kein Dreh- oder Haltemoment erzeugt. |
| 38 | *STO Self-Test*: Fehler während des Selbsttests der Firmware. Kontaktieren Sie Nanotec. |
| 39 | Fehler in der Ballast-Konfiguration: nicht gültige/realistische Parameter eingetragen (siehe Ballast-Überwachung) |
| 40 | Warnung: Ballast-Widerstand thermisch überlastet |
| 41 | Nur EtherCAT: *Sync Manager Watchdog*: Die Steuerung hat zu lange keine PDO-Daten empfangen, überprüfen Sie die Software- und Hardware-Verbindungen. |
| 46 | Interlock-Fehler: Bit 3 in $60FD_h$ steht auf "0", der Motor darf nicht fahren (siehe Abschnitt *Interlock-Funktion* im Kapitel Digitale Eingänge) |
| 48 | Nur CANopen: NMT-Status wurde gesetzt auf *stopped* |
| 52 | Feldbus-Zykluszeit überschritten. |
| 53 | Nur PROFINET: Falscher Stack-Version |

**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 |
| $5440_h$ | Interlock-Fehler: Bit 3 in $60FD_h$ steht auf "0", der Motor darf nicht fahren (siehe Abschnitt *Interlock-Funktion* im Kapitel Digitale Eingänge) |
| $6010_h$ | Software-Reset (Watchdog) |
| $6100_h$ | Interner Softwarefehler, generisch |
| $6320_h$ | Nennstrom muss gesetzt werden ($6075_h$) |
| $7110_h$ | Fehler in der Ballast-Konfiguration: nicht gültige/realistische Parameter eingetragen (siehe Ballast-Überwachung) |
| $7113_h$ | Warnung: Ballast-Widerstand thermisch überlastet |
| $7121_h$ | Motor blockiert |
| $7200_h$ | Interner Fehler: Korrekturfaktor für Referenzspannung fehlt im OTP |
| $7305_h$ | Sensor 1 (siehe $3204_h$) fehlerhaft |
| $7306_h$ | Sensor 2 (siehe $3204_h$) fehlerhaft |
| $7307_h$ | Sensor n (siehe $3204_h$), wo n größer 2 |
| $7600_h$ | Warnung: Nichtflüchtiger Speicher voll oder korrupt, Neustart der Steuerung für Aufräumarbeiten |
| $8000_h$ | allgemeiner Fehler bei der Überwachung der STO-Funktion |
| $8001_h$ | allgemeiner Fehler bei der Überwachung der STO-Funktion |
| $8002_h$ | allgemeiner Fehler bei der Überwachung der STO-Funktion |
| $8003_h$ | *STO Self-Test*: Fehler während des Selbsttests der Firmware. Kontaktieren Sie Nanotec. |
| $8100_h$ | 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.

**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.

Sie können das Generieren von Sync-Nachrichten aktivieren (die Steuerung wird zu *Sync-Master des Netzwerks*), indem Sie das Bit 30 auf "1" setzen. Die Zykluszeit stellen Sie im Objekt $1006_h$ ein.

### 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 | |

## 1006h Communication Cycle Period

### Funktion

Enthält die Zykluszeit für die generierten Sync-Nachrichten (siehe $1005_h$) in µs. Erlaubt werden nur Vielfache von 1000 µs.

### Objektbeschreibung

| | |
|---|---|
| Index | $1006_h$ |
| Objektname | Communication Cycle Period |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Kommunikation |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |
| Firmware Version | FIR-v2013-B726332 |
| Änderungshistorie | |

## 1007h Synchronous Window Length

### Funktion

Dieses Objekt enthält die Länge des Zeitfensters in Mikrosekunden für synchrone PDOs. Wenn das synchrone Zeitfenster abgelaufen ist, werden alle synchronen TxPDOs verworfen. 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 | ■ N6-1-2-1-S: N6-1-2-1-S<br>■ N6-1-2-1: N6-1-2-1<br>■ N6-2-2-1-S: N6-2-2-1-S<br>■ N6-2-2-1: N6-2-2-1 |
| 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-v2508-B1072143 |
| 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 Protokol. 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 | Firmware Version FIR-v1436: Eintrag "Objektname" geändert von "Store Parameter" auf "Store Parameters".<br>Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 3 auf 4.<br>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5.<br>Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7.<br>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Save Miscellaneous Configurations To Non-volatile Memory" auf "Save Reserved0 Configurations To Non-volatile Memory".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Save Drive Parameters To Non-volatile Memory" auf "Save Motor Parameters To Non-volatile Memory".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Save Tuning Parameters To Non-volatile Memory" auf "Save Sensor Parameters To Non-volatile Memory". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | $00000001_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Save Application Parameters To Non-volatile Memory |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Save Customer Parameters To Non-volatile Memory |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Save Motor Parameters To Non-volatile Memory |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Save Sensor Parameters To Non-volatile Memory |
| Datentyp | UNSIGNED32 |

| | |
|---|---|
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $07_h$ |
| Name | Save Reserved0 Configurations To Non-volatile Memory |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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 | Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "Restore Default Parameter" auf "Restore Default Parameters".<br>Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 2 auf 4.<br>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5.<br>Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Comm Default Parameters" auf "Restore Communication Default Parameters".<br>Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Application Default Parameters" auf "Restore Application Default Parameters".<br>Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7.<br>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Restore Miscellaneous Configurations" auf "Restore Reserved0 Configurations To Non-volatile Memory".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Restore Drive Default Parameters" auf "Restore Motor Default Parameters".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Restore Tuning Default Parameters" auf "Restore Sensor Default Parameters". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 Motor Default Parameters |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Restore Sensor Default Parameters |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $07_h$ |
| Name | Restore Reserved0 Configurations To Non-volatile Memory |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |

| | |
|---|---|
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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 | $00000001_h$ |

| | |
|---|---|
| Subindex | $0B_h$ |
| Name | Restore Modbus RTU Configurations To Non-volatile Memory |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $0C_h$ |
| Name | Restore Ethernet Configurations To Non-volatile Memory |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| 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 | Number Of Entries |
| 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.

**TIPP**

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 | Number Of Entries |
| 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 | ■ N6-1-1-1-S: $00110001_h$<br>■ N6-1-1-1: $00110101_h$<br>■ N6-2-1-1-S: $00120001_h$<br>■ N6-2-1-1: $00120101_h$<br>■ N6-1-2-1-S: $00110002_h$<br>■ N6-1-2-1: $00110102_h$<br>■ N6-2-2-1-S: $00120002_h$<br>■ N6-2-2-1: $00120102_h$<br>■ N6-1-3-1-S: $00110003_h$<br>■ N6-1-3-1: $00110103_h$<br>■ N6-1-3-2-S: $00110203_h$<br>■ N6-1-3-2: $00110303_h$<br>■ N6-2-3-1-S: $00120003_h$<br>■ N6-2-3-1: $00120103_h$<br>■ N6-2-3-2-S: $00120203_h$<br>■ N6-2-3-2: $00120303_h$<br>■ N6-1-4-1-S: $00110004_h$<br>■ N6-1-4-1: $00110104_h$<br>■ N6-2-4-1-S: $00120004_h$<br>■ N6-2-4-1: $00120104_h$<br>■ N6-1-5-1-S: $00110005_h$ |

- N6-1-5-1: $00110105_h$
- N6-2-5-1-S: $00120005_h$
- N6-2-5-1: $00120105_h$
- N6-1-7-1: $00110107_h$
- N6-1-10-1-S: $0011000A_h$
- N6-1-10-1: $0011010A_h$
- N6-2-10-1-S: $0012000A_h$
- N6-2-10-1: $0012010A_h$

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Revision Number |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $09CC0000_h$ |

| | |
|---|---|
| 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 $F0_h$ und 0. Ist der Wert "0", wird die Sync-Nachricht als CAN-Nachricht mit einer Datenlänge von 0 übertragen.

## 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | $00_h$ |

| | |
|---|---|
| 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 | $1400_h$ |
| Objektname | 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 | Number Of Entries |
| 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.

## 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 | $1401_h$ |
| Objektname | 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | $60400010_h$ |

| | |
|---|---|
| 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 | $00000000_h$ |

| | |
|---|---|
| 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 | $00000000_h$ |

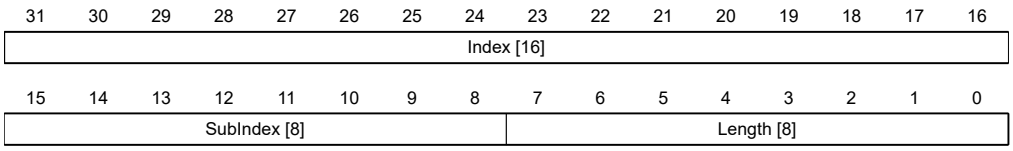
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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.

## 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".<br>Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Positioning Control" auf "Receive PDO 2 Mapping Parameter". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | $60810020_h$ |

| | |
|---|---|
| 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 |

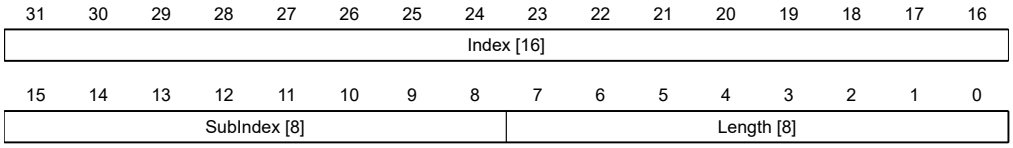
| | |
|---|---|
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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.

# 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".<br>Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Velocity Control" auf "Receive PDO 3 Mapping Parameter". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | $60FF0020_h$ |

| | |
|---|---|
| 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".<br>Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Output Control" auf "Receive PDO 4 Mapping Parameter". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | Number Of Entries |
| 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 | $1605_h$ |
| Objektname | 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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 | $00000000_h$ |

## 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 | Number Of Entries |
| 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 | $00000000_h$ |

## 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 | Number Of Entries |
| 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.

## 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 | Number Of Entries |
| 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.

## 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 | Number Of Entries |
| 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.

## 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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-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.

## 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 | Number Of Entries |
| 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-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.

## 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 | Number Of Entries |
| 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-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.

## 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 | Number Of Entries |
| 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-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.

## 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".<br>Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Drive Status" auf "Transmit PDO 1 Mapping Parameter". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | $00000000_h$ |

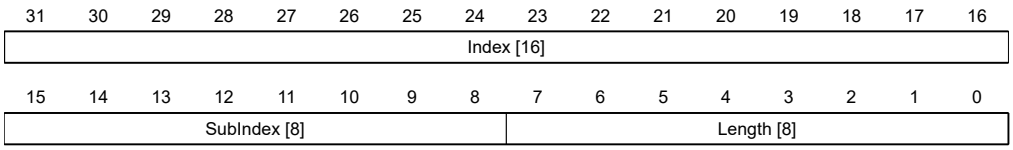
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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 | $1A01_h$ |
| Objektname | 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".<br>Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Positioning Status" auf "Transmit PDO 2 Mapping Parameter". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | $00000000_h$ |

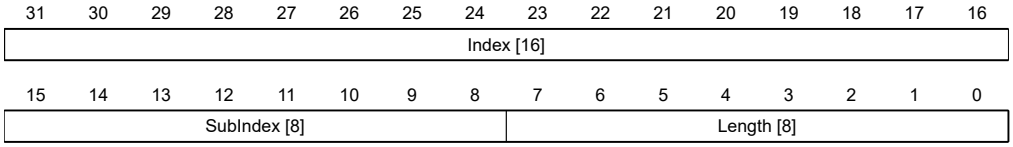
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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".<br>Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Velocity Status" auf "Transmit PDO 3 Mapping Parameter". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | $606C0020_h$ |

| | |
|---|---|
| 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 | $00000000_h$ |

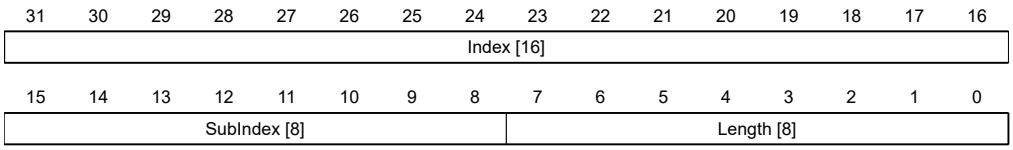
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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.

## 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".<br>Firmware Version FIR-v1426: Eintrag "Object Name" geändert von "Input Status" auf "Transmit PDO 4 Mapping Parameter". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | $00000000_h$ |

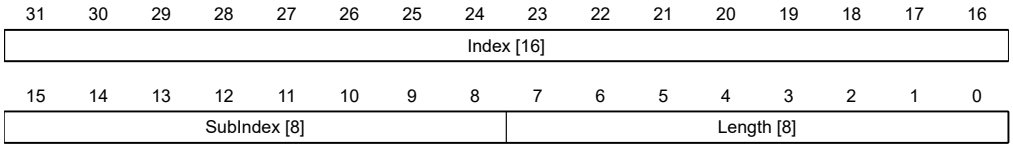
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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 | Number Of Entries |
| 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 | $00000000_h$ |

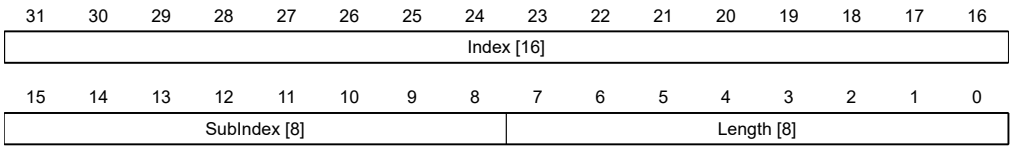
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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 | Number Of Entries |
| 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 | $00000000_h$ |

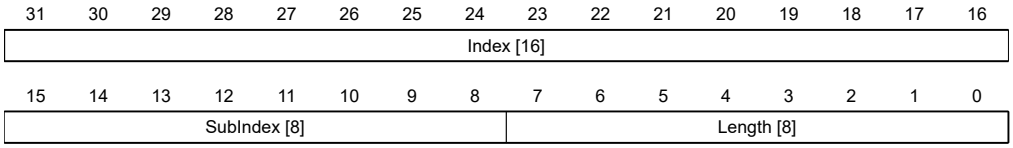
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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.

## 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 | Number Of Entries |
| 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 | $00000000_h$ |

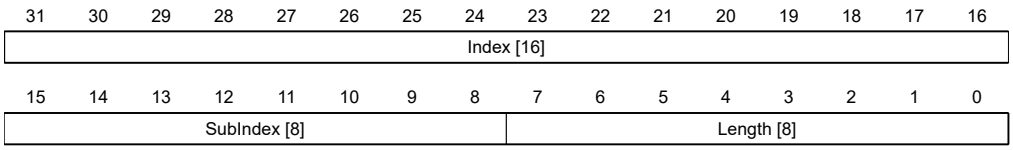
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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.

## 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 | Number Of Entries |
| 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 | $00000000_h$ |

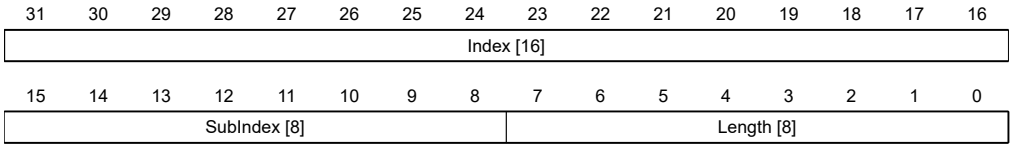
| | |
|---|---|
| Subindex | $07_h$ |
| Name | 7th Object To Be Mapped |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| 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.

## 1C00h Sync Manager Communication Type

### Funktion

Diese Objekt zeigt die Zuordnung der vier EtherCAT SyncManager an.

### Objektbeschreibung

| | |
|---|---|
| Index | $1C00_h$ |
| Objektname | Sync Manager Communication Type |
| Object Code | ARRAY |
| Datentyp | UNSIGNED8 |

| | |
|---|---|
| Speicherbar | nein |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | Die Anzahl der Einträge haben sich geändert von 2 auf 5<br>Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Sync Manager Communication Type" auf "Sync-Manager Type Channel 1: Mailbox Write".<br>Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Sync Manager Communication Type" auf "Sync-Manager Type Channel 2: Mailbox Read".<br>Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Sync Manager Communication Type" auf "Sync-Manager Type Channel 3: Process Data Write (Outputs)".<br>Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Sync Manager Communication Type" auf "Sync-Manager Type Channel 4: Process Data Read (Inputs)". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Sync-Manager Type Channel 1: Mailbox Write |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $01_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Sync-Manager Type Channel 2: Mailbox Read |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Sync-Manager Type Channel 3: Process Data Write (Outputs) |

| | |
|---|---|
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $03_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Sync-Manager Type Channel 4: Process Data Read (Inputs) |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

### Beschreibung

Die Zuordnung der SyncManager ist vom Hersteller festgelegt und kann nicht geändert werden.

| Subindex / Syncmanager | Funktion |
|---|---|
| 1 | Empfangen von Mailboxnachrichten |
| 2 | Senden von Mailboxnachrichten |
| 3 | Empfangen von zyklischen Prozessdaten |
| 4 | Senden von zyklischen Prozessdaten |

## 1F50h Program Data

### Funktion

Dieses Objekt wird zum Programmieren von Speicherbereichen der Steuerung verwendet. Jeder Eintrag steht für einen bestimmten Speicherbereich.

### Objektbeschreibung

| | |
|---|---|
| Index | $1F50_h$ |
| Objektname | 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 | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| 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 |

## 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 | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Program Control Bootloader/firmware |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Program Control NanoJ |
| 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 | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| 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 | $00000000_h$ |

## 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 |
|---|---|---|
| dec | hex | in kBd |
| 129 | 81 | 10 |
| 130 | 82 | 20 |
| 131 | 83 | 50 |
| 132 | 84 | 125 |
| 133 | 85 | 250 |
| 134 | 86 | 500 |
| 135 | 87 | 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".<br>Firmware Version FIR-v2412: Subindex $02_h$ hinzugefügt. |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | BL Config |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Automatic COB-ID |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

## 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.
- Subindex 02: Ist der Wert "1" (Werkseinstellung), wird die COB-ID für die PDOs 1 bis 4 automatisch von der Firmware eingetragen. Um manuell die COB-IDs einzustellen und zu speichern, setzen Sie den Wert auf "0".

## 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.

### 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.

### 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".<br>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.

## 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 | UNSIGNED16 |
| Speicherbar | ja, Kategorie: Motor |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0032_h$ |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "nein" auf "ja, Kategorie: Tuning". |

| | |
|---|---|
| | Firmware Version FIR-v2315-B1040535: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "UNSIGNED16". |

## 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 | $0000EA60_h$ |
| 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.

## 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 | Number Of Entries |
| 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 50 und 20000 ($4E20_h$) |
| Vorgabewert | $00004E20_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | PWM Duty Cycle |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | Zwischen 1 und 100 ($64_h$) |
| Vorgabewert | $00000032_h$ |

## 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 des PWM-Signals in Hertz.
- $06_h$: Tastgrad des PWM-Signals in Prozent.

## 2039h Motor Currents

### Funktion

Dieses Objekt enthält die gemessenen Motorströme in mA. Alle Werte sind Peak-Werte, (#2*Effektivwert).

### Objektbeschreibung

| | |
|---|---|
| Index | $2039_h$ |
| Objektname | Motor Currents |
| Object Code | ARRAY |
| Datentyp | INTEGER32 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "nein" auf "TX-PDO".<br>Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "TX-PDO".<br>Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 03 geändert von "nein" auf "TX-PDO".<br>Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 04 geändert von "nein" auf "TX-PDO".<br>Firmware Version FIR-v2213: Subindex $05_h$, "Actual Current" hinzugefügt. Phasenströme Ia und Ib in Iα und Iβ geändert (Clarke-Transormation). |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $05_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Id |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Iq |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Iα |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Iβ |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Actual Current |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## Beschreibung

- $01_h$: feldbildende Komponente des Stroms
- $02_h$: momentbildende Komponente des Stroms
- $03_h$: Iα
- $04_h$: Iβ
- $05_h$: Gesamtstrom runtergerechnet auf eine Motorphase. Im *Closed Loop* wird zusätzlich das Vorzeichen von Iq verwendet. Der Stromwert kann dann zum Vergleich mit dem Strom aus $6075_h$ auf eine Skala gelegt werden.
  *Open Loop*:
  - Schrittmotor: $I = \sqrt{(I\alpha^2 + I\beta^2)} / \sqrt{2}$
  - BLDC-Motor: $I = \sqrt{(I\alpha^2 + I\beta^2)} / (2/\sqrt{3})$

  *Closed Loop*:
  - Schrittmotor: $I = sgn(Iq) * \sqrt{(I\alpha^2 + I\beta^2)} / \sqrt{2}$

- BLDC-Motor: I = sgn(Iq ) * √(Iα ² + Iβ ²) / (2/√3)

## 203Dh Torque Window

### Funktion

Gibt relativ zum Zieldrehmoment einen symmetrischen Bereich an, innerhalb dessen das Ziel als erreicht gilt.

Wird der Wert auf "FFFFFFFF"$_h$ gesetzt, wird die Überwachung abgeschaltet, das Bit "Target reached" im Objekt 6041$_h$ (Statusword) wird nie gesetzt.

### Objektbeschreibung

| | |
|---|---|
| Index | 203D$_h$ |
| Objektname | 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".<br>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) des Objekts $60F8_h$ (Max Slippage) 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 | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000080_h$ |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | Firmware Version FIR-v2339-B1048823: Eintrag "Datentyp" geändert von "INTEGER32" auf "UNSIGNED32". |

## 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 | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | Firmware Version FIR-v2339-B1048823: Eintrag "Datentyp" geändert von "INTEGER32" auf "UNSIGNED32". |

## 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 | $205B_h$ |
| Objektname | 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 | |

## 205Ch Pulse Generator Configuration

### Funktion

Mit diesem Objekt konfigurieren Sie den virtuellen Encoderausgang. Details finden Sie im Kapitel Digitale Ein- und Ausgänge, Pulsgenerator.

## Objektbeschreibung

| | |
|---|---|
| Index | $205C_h$ |
| Objektname | Pulse Generator Configuration |
| Object Code | ARRAY |
| Datentyp | INTEGER16 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2115-B1016293 |
| Änderungshistorie | Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Number Of Increments To Be Sent When No Sensor In 0x3203 Is Choosen" auf "Number Of Increments To Be Sent When No Sensor In 0x3203 Is Choosen".<br>Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Output Signal Mode: 0=AB, 1=ClkDir, 2=CW/CCW" auf "Output Signal Mode: 0=AB, 1=ClkDir, 2=CW/CCW".<br>Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Numerator For Conversion Of Sensor Increments To Virtual Encoder Increments" auf "Numerator For Conversion Of Sensor Increments To Virtual Encoder Increments".<br>Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Denominator For Conversion Of Sensor Increments To Virtual Encoder Increments" auf "Denominator For Conversion Of Sensor Increments To Virtual Encoder Increments".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Object Name" geändert von "Virtual Encoder Configuration" auf "Pulse Generator Configuration".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Numerator For Conversion Of Sensor Increments To Virtual Encoder Increments" auf "Numerator For Conversion Of Sensor Increments To Pulse Generator Increments".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Denominator For Conversion Of Sensor Increments To Virtual Encoder Increments" auf "Denominator For Conversion Of Sensor Increments To Pulse Generator Increments". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Number Of Increments To Be Sent When No Sensor In 0x3203 Is Choosen |
| Datentyp | INTEGER16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Output Signal Mode: 0=AB, 1=ClkDir, 2=CW/CCW |
| Datentyp | INTEGER16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Numerator For Conversion Of Sensor Increments To Pulse Generator Increments |
| Datentyp | INTEGER16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Denominator For Conversion Of Sensor Increments To Pulse Generator Increments |
| Datentyp | INTEGER16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0001_h$ |

## 2101h Fieldbus Module Availability

### Funktion

Zeigt die verfügbaren Feldbusse an.

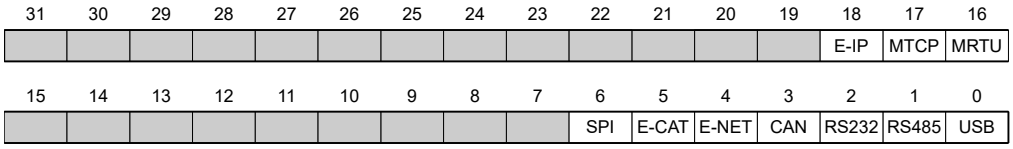
### Objektbeschreibung

| | |
|---|---|
| Index | $2101_h$ |
| Objektname | Fieldbus Module Availability |
| Object Code | VARIABLE |

| | |
|---|---|
| Datentyp | UNSIGNED32 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |
| 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).




**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.

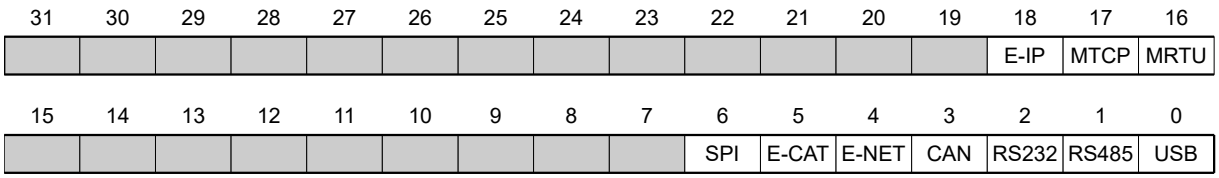
## Objektbeschreibung

| | |
|---|---|
| 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 | $00000000_h$ |
| 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:




**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 | Number Of Entries |
| 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 | ■ N6-1-1-1-S: $00000121_h$<br>■ N6-1-1-1: $00000121_h$<br>■ N6-2-1-1-S: $00000121_h$<br>■ N6-2-1-1: $00000121_h$<br>■ N6-1-2-1-S: $00000109_h$<br>■ N6-1-2-1: $00000109_h$<br>■ N6-2-2-1-S: $00000109_h$<br>■ N6-2-2-1: $00000109_h$<br>■ N6-1-3-1-S: $00000911_h$<br>■ N6-1-3-1: $00000911_h$<br>■ N6-1-3-2-S: $00000911_h$<br>■ N6-1-3-2: $00000911_h$<br>■ N6-2-3-1-S: $00000911_h$<br>■ N6-2-3-1: $00000911_h$<br>■ N6-2-3-2-S: $00000911_h$<br>■ N6-2-3-2: $00000911_h$<br>■ N6-1-4-1-S: $00000511_h$<br>■ N6-1-4-1: $00000511_h$<br>■ N6-2-4-1-S: $00000511_h$<br>■ N6-2-4-1: $00000511_h$<br>■ N6-1-5-1-S: $00000103_h$<br>■ N6-1-5-1: $00000103_h$<br>■ N6-2-5-1-S: $00000103_h$<br>■ N6-2-5-1: $00000103_h$<br>■ N6-1-7-1: $00000101_h$<br>■ N6-1-10-1-S: $00000311_h$<br>■ N6-1-10-1: $00000311_h$<br>■ N6-2-10-1-S: $00000311_h$<br>■ N6-2-10-1: $00000311_h$ |

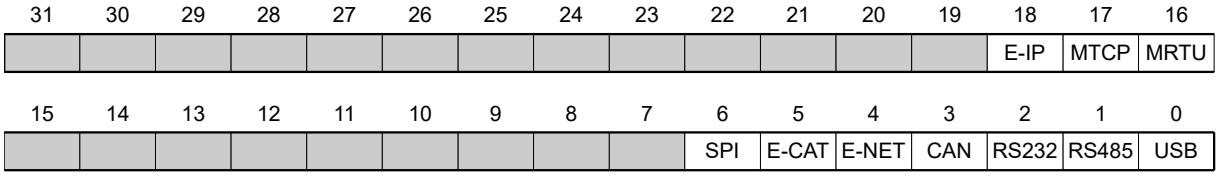
| | |
|---|---|
| Subindex | $02_h$ |
| Name | Fieldbus Module Enabled |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## Beschreibung

Subindex 1 (Fieldbus Module Disable Mask): In diesem Subindex werden alle physikalischen Schnittstellen und Protokolle angezeigt, welche aktiviert oder deaktiviert werden können. Ein gesetztes Bit bedeutet, dass dieser Feldbus deaktivierbar ist.

Subindex 2 (Fieldbus Module Enabled): Dieser Subindex zeigt alle zur Zeit aktivierten physikalischen Schnittstellen und Protokolle an. Ein gesetztes Bit 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 | $2290_h$ |
| Objektname | 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 | $2291_h$ |
| Objektname | PDI Input |
| Object Code | RECORD |
| Datentyp | PDI_INPUT |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1748-B531667 |
| Änderungshistorie | Firmware Version FIR-v2013-B726332: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".<br>Firmware Version FIR-v2315-B1040535: Eintrag "Data type" geändert von "INTEGER8" auf "UNSIGNED8". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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 | UNSIGNED8 |
| 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 | Firmware Version FIR-v2315-B1040535: Eintrag "Data type" geändert von "INTEGER16" auf "UNSIGNED16". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | PDI Status |
| Datentyp | UNSIGNED16 |
| 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

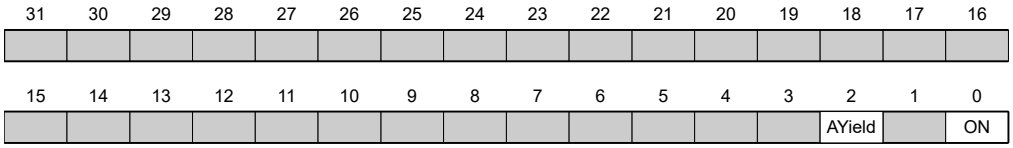
Steuert die Ausführung eines *NanoJ-Programms*.

### 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.

Beim Einschalten wird geprüft, ob ein *NanoJ-Programm* vorhanden ist. Wenn ja,wird in 2300 eine "1" eingetragen und damit das *NanoJ-Programm*gestartet.

**AYield (AutoYield)**

Ist dieses Feature aktiviert (Bit auf "1"), wird das *NanoJ-Programm* nicht mehr angehalten, wenn es länger läuft, als es darf. Somit ist aber das *NanoJ-Programm* nicht mehr echtzeitfähig und läuft nicht mehr im 1-Millesekunde-Takt (siehe Verfügbare Rechenzeit).

# 2301h NanoJ Status

## Funktion

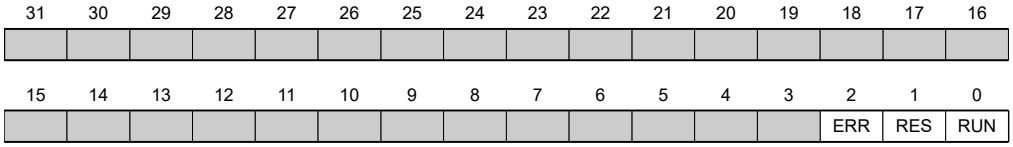
Zeigt den Betriebszustand des Benutzerprogramms an.

## 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 |
|---|---|
| $1xxxxyy_h$ | 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. |

| Nummer | Beschreibung |
|---|---|
| $2000000_h$ | Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs `input` deklariert (siehe 2310h NanoJ Input Data Selection) |
| $3000000_h$ | Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs `output` deklariert (siehe 2320h NanoJ Output Data Selection) |
| $4000000_h$ | Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs `inout` deklariert (siehe 2330h NanoJ In/output Data Selection) |
| $0002_h$ | Fehler im Speicher-Management |
| $0003_h$ | Usage-Fault |
| $0004_h$ | Hatd-Fault |
| $0005_h$ | Time Out: Code wird zu lange ohne yield() oder sleep() ausgeführt |
| $0006_h$ | Bus-Fault |
| $1000_h$ | Zugriff auf ein nicht existierendes Objekt im Objektverzeichnis |
| $1002_h$ | Es wurde versucht, einen zu niedrigen oder zu hohen Wert in ein Objekt zu schreiben. |
| $1003_h$ | Es wurde versucht, ein Objekt auszulesen, das nur Schreibzugriff erlaubt. |
| $1FFF_h$ | unzulässiger Zugriff auf ein Objekt |

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. |
| $2000000_h$ + Anzahl der überzähligen Variablen | Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs `input` deklariert (siehe 2310h NanoJ Input Data Selection) |
| $3000000_h$ + Anzahl der überzähligen Variablen | Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs `output` deklariert (siehe 2320h NanoJ Output Data Selection) |
| $4000000_h$ + Anzahl der überzähligen Variablen | Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs `inout` deklariert (siehe 2330h NanoJ In/output Data Selection) |
| $1000_h$ | Zugriff auf ein nicht existierendes Objekt im Objektverzeichnis |
| $1001_h$ | Schreibzugriff auf schreibgeschützten Eintrag im OD |
| $1002_h$ | Es wurde versucht, einen zu niedrigen oder zu hohen Wert in ein Objekt zu schreiben. |
| $1003_h$ | Es wurde versucht, ein Objekt auszulesen, das nur Schreibzugriff erlaubt. |
| $1FFF_h$ | unzulässiger Zugriff auf ein Objekt |

## 2303h NanoJ Metadata

### Funktion

Dieses Objekt dient der richtigen Erkennung des NanoJ-Programms von externer Software, beispielsweise *Plug & Drive Studio*.

## Objektbeschreibung

| | |
|---|---|
| Index | $2303_h$ |
| Objektname | NanoJ Metadata |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2339-B1048823 |
| Änderungshistorie | |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $06_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Version |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000300_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | FLASH Start Address |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | FLASH End Address |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |

| | |
|---|---|
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | RAM Start Address |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | RAM End Address |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Header Size |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## 230Eh Timer

### Funktion

Dieses Objekt enthält die Betriebszeit seit dem letzen Start der Steuerung in Millisekunden.

**HINWEIS**

Dieses Objekt wird nicht gespeichert, die Zählung beginnt nach dem Einschalten oder einem Überlauf wieder mit "0".

### Objektbeschreibung

| | |
|---|---|
| Index | $230E_h$ |
| Objektname | Timer |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |

| | |
|---|---|
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2139-B1020888 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $01_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | 1ms Timer |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## 230Fh Uptime Seconds

### Funktion

Dieses Objekt enthält die Betriebszeit seit dem letzen Start der Steuerung in Sekunden.

**HINWEIS**

Dieses Objekt wird nicht gespeichert, die Zählung beginnt nach dem Einschalten oder einem Überlauf 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".<br>Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".<br>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".<br>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $10_h$ |

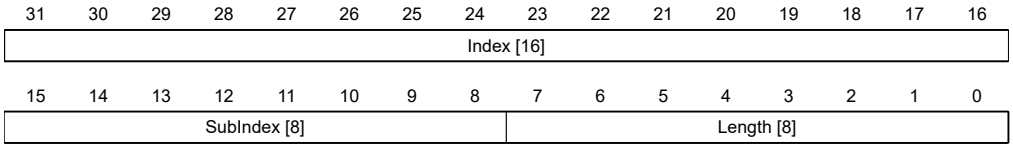
| | |
|---|---|
| Subindex | $01_h$ - $10_h$ |
| Name | Mapping #1 - #16 |
| Datentyp | UNSIGNED32 |

| | |
|---|---|
| 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 | Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Output Data Selection" auf "NanoJ Output Data Selection".<br>Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".<br>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

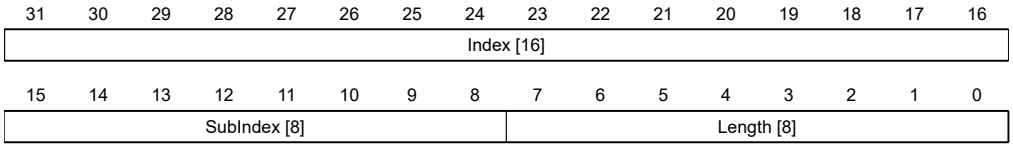
| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $10_h$ |

| | |
|---|---|
| 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 | Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM In/output Data Selection" auf "NanoJ In/output Data Selection".<br>Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".<br>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".<br>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen". |

## Wertebeschreibung

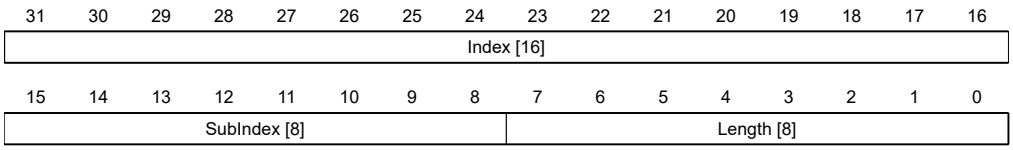
| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $10_h$ |

| | |
|---|---|
| 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<br>Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Inputs" auf "NanoJ Inputs".<br>Firmware Version FIR-v1436: Eintrag "Name" geändert von "VMM Input N#" auf "NanoJ Input N#". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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.

## 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".<br>Firmware Version FIR-v1436: Eintrag "Name" geändert von "VMM Output N#" auf "NanoJ Output N#". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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.

## 2701h Customer Storage Area

### Funktion

In dieses Objekt können Daten abgelegt und gespeichert werden.

### Objektbeschreibung

| | |
|---|---|
| Index | $2701_h$ |
| Objektname | Customer Storage Area |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Benutzer |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1540 |
| Änderungshistorie | Firmware Version FIR-v1540: Eintrag "Data type" geändert von "UNSIGNED32" auf "UNSIGNED8". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $FE_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $FE_h$ |
| Name | Storage #1 - #254 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## 2800h Bootloader And Reboot Settings

### Funktion

Mit diesem Objekt lässt sich ein Reboot der Firmware auslösen.

### 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 | Number Of Entries |
| 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 | $00000064_h$ |

| | |
|---|---|
| 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.
- $03_h$: reserviert

## 306Ch Velocity Actual Internal Value

### Funktion

Enthält die aktuelle Istgeschwindigkeit in U/min. Der Wert ist im Vergleich zu 606Ch Velocity Actual Value weniger stark gefiltert und wird intern vom Geschwindigkeitsregler unf für die Spannungsvorsteuerung und Totzeitkompensation verwendet.

### Objektbeschreibung

| | |
|---|---|
| Index | $306C_h$ |
| Objektname | Velocity 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-v2412-B1057638 |
| Änderungshistorie | |

## 3202h Motor Drive Submode Select

### Funktion

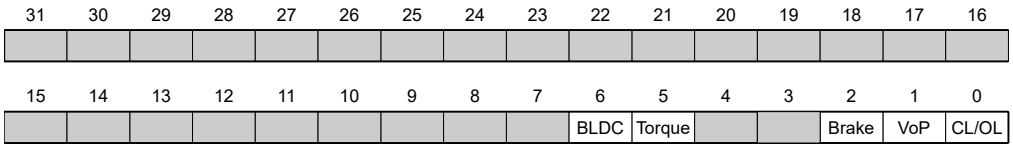
Steuert die Reglerbetriebsart, wie z. B. die *Closed Loop*/ *Open Loop*-Umschaltung und ob der Velocity-Mode über den S-Regler simuliert wird oder mit einem echten V-Regler im *Closed Loop* arbeitet.

### Objektbeschreibung

| | |
|---|---|
| Index | $3202_h$ |
| Objektname | Motor Drive Submode Select |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED32 |

| | |
|---|---|
| Speicherbar | ja, Kategorie: Motor |
| 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".<br>Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Fahrt" auf "ja, Kategorie: Bewegung".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Bewegung" auf "ja, Kategorie: Motor". |

## Beschreibung




### CL/OL

Umschaltung zwischen *Open Loop* und *Closed Loop* (siehe Kapitel Betriebsarten)

- Wert = "0": *Open Loop*
- Wert = "1": *Closed Loop*

Das Umschalten ist nicht möglich im Zustand *Operation enabled*.

### VoP

Wert = "1": V-Regler über eine P-Rampe simulieren: die Geschwindigkeitsmodi über kontinuierliche Positionsänderungen simulieren

### Brake

Wert = "1": Einschalten der automatischen Bremsensteuerung.

### 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, die 321Bh Velocity Controller Parameters 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.

Eine Wertänderung im Zustand *Operation enabled* zeigt keinen sofortigen Effekt. Wertänderungen in den Objekten werden zwischengespeichert und ausgelesen beim Übergang nach Zustand *Operation enabled*.

## Objektbeschreibung

| | |
|---|---|
| Index | $3203_h$ |
| Objektname | Feedback Selection |
| Object Code | ARRAY |
| Datentyp | UNSIGNED8 |
| Speicherbar | ja, Kategorie: Sensors |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1748-B538662 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Sensors". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| 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 | $00_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | 4th 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 dieser Sensor für die Rückführung der Position verwendet.
- Bit 1: wird das Bit auf "1" gesetzt, wird dieser Sensor für die Rückführung der Geschwindigkeit verwendet.
- Bit 2: wird das Bit auf "1" gesetzt, wird dieser Sensor für die Rückführung der Kommutierung im Closed Loop verwendet.

Subindex $01_h$ entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

Welchen Sensor die Steuerung für die einzelnen Regler (Kommutierung, Geschwindigkeit, Position) berücksichtigt, ist implizit durch die Reihenfolge der Sensoren vorgegeben.

Das Aufsuchen beginnt immer mit Sensor 2 und setzt sich aufsteigend fort, bis alle vorhandenen Sensoren abgefragt wurden. Wird ein Sensor gefunden dessen Ruckführung 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 | |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| 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 | $33A0_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Index Of 4th Feedback Interface |
| Datentyp | UNSIGNED16 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $33B0_h$ |

### 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*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

## 3205h Feedback Use

### Funktion

Dieses Objekt zeigt, welcher Sensor für welchen Steuerkreis aktuell verwendet wird.

### Objektbeschreibung

| | |
|---|---|
| Index | $3205_h$ |
| Objektname | Feedback Use |
| Object Code | ARRAY |
| Datentyp | UNSIGNED16 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2412-B1057638 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |

| | |
|---|---|
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Sensor For Position Control Loop |
| Datentyp | UNSIGNED16 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Sensor For Velocity Control Loop |
| Datentyp | UNSIGNED16 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Sensor For Current Control Loop |
| Datentyp | UNSIGNED16 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Sensor For Pulse Generator |
| Datentyp | UNSIGNED16 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

## 320Dh Moment Of Inertia

### Funktion

Dieser Faktor wird für die Berechnung der Beschleunigungsvorsteuerung verwendet (siehe 321D). Default ist 0 (Vorsteuerung inaktiv).

Die Beschleunigungsvorsteuerung ist auch beim Verzögern wirksam.

## Objektbeschreibung

| | |
|---|---|
| Index | $320D_h$ |
| Objektname | Moment Of Inertia |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1825-B577172 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Bewegung" auf "ja, Kategorie: Applikation".<br>Firmware Version FIR-v2508-B1072143: Eintrag "Object Name" geändert von "Torque Of Inertia Factor" auf "Moment Of Inertia".<br>Firmware Version FIR-v2508-B1072143: Eintrag "Name" geändert von "Current" auf "Torque [‰]". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

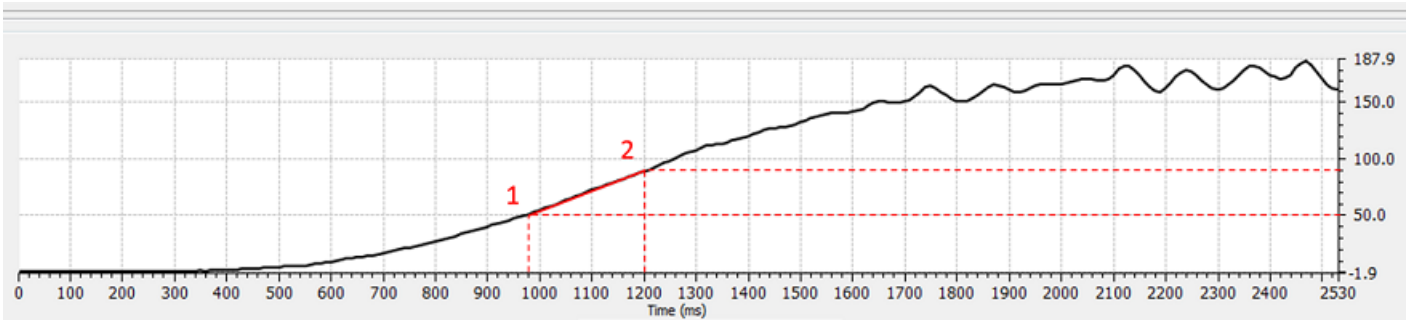
| | |
|---|---|
| Subindex | $01_h$ |
| Name | Torque [‰] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Acceleration |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## Beschreibung

Der Wert hängt von der Trägheit der Last ab. Um den Faktor zu ermitteln:

1. Aktivieren Sie den *Closed Loop* und wählen Sie den Modus *Profile Torque* .
2. Stellen Sie eine Zielvorgabe für das Drehmoment und tragen Sie den Wert in $320D_h:01_h$ ein.
3. Zeichnen Sie (z. B. im *Plug & Drive Studio*) die aktuelle Geschwindigkeit (Objekt $606C_h$) auf. Berechnen Sie die Beschleunigung in den eingestellten benutzerdefinierten Einheiten für den Drehzahlbereich, wo diese konstant ist. Tragen Sie den Wert in $320D_h:02_h$ ein.
   Am Beispiel der Drehzahl-Kurve in der folgenden Abbildung:
   (90-50)/(1200-980)=182 U/min pro Sekunde.




# 3211h Motor Drive Set Point

## Funktion

Zum Auslesen der Id/Iq- und Ud/Uq- Set Points,

## Objektbeschreibung

| | |
|---|---|
| Index | $3211_h$ |
| Objektname | Motor Drive Set Point |
| Object Code | ARRAY |
| Datentyp | INTEGER32 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2451-B1068465 |
| Änderungshistorie | |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Id Set Point [mA] |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Iq Set Point [mA] |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Ud Set Point [mV] |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Uq Set Point [mV] |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## 3219h Current Configuration

### Funktion

Enthält verschiedene Stromeinstellungen.

### Objektbeschreibung

| | |
|---|---|
| Index | $3219_h$ |
| Objektname | Current Configuration |
| Object Code | ARRAY |
| Datentyp | UNSIGNED16 |
| Speicherbar | ja, Kategorie: Applikation |

| | |
|---|---|
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2451-B1068465 |
| Änderungshistorie | |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $05_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | $I^2t$ Duration Of Max Current [ms] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $03E8_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Open Loop Idle State Detection Time [ms] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $03E8_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Open Loop Idle State Current [‰] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $01F4_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Block Detection Time [ms] |

| | |
|---|---|
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00C8_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Block Detection Threshold [‰] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02BC_h$ |

### Beschreibung

Die Funktionen sind wie folgt:

- Subindex $01_h$: Maximale Dauer des Maximalstroms ($6073_h$) in Millisekunden. Nur relevant, wenn $6073_h$ > 1000 ‰.
- Subindex $02_h$: Zeit in Millisekunden, die sich der Motor im Stillstand befinden muss, bis die Stromabsenkung aktiviert wird.
- Subindex $03_h$: Reduzierter Strom als Promille des Nennstroms ($6075_h$), wenn sich der Motor im *Open Loop* im Stillstand befindet.
- Subindex $04_h$: Zeit in Millisekunden, die der Motor nach der Blockdetektion trotzdem noch gegen den Block fahren soll (siehe *Homing auf Block* im Modus Homing).
- Subindex $05_h$: Stromgrenzwert Promille des Nennstroms ($6075_h$), ab dem ein Blockieren detektiert werden soll (siehe *Homing auf Block* im Modus Homing).

## 321Ah Current Controller Parameters

### Funktion

Enthält die Parameter für den Stromregler (Kommutierung). In der Regel sollen die Werte für Iq (Subindex $01_h$/$02_h$) und Id (Subindex $03_h$/$04_h$) gleich sein. Siehe Kapitel Reglerstruktur.

### Objektbeschreibung

| | |
|---|---|
| Index | $321A_h$ |
| Objektname | Current Controller Parameters |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2213-B1028181 |
| Änderungshistorie | |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Proportional Gain Kp For Iq [mV/A] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00001DB0_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Integrator Time Ti For Iq [µs] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $000018BD_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Proportional Gain Kp For Id [mV/A] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00001DB0_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Integrator Time Ti For Id [µs] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $000018BD_h$ |

## 321Bh Velocity Controller Parameters

### Funktion

Enthält die Parameter für den Geschwindigkeitsregler. Siehe Kapitel Reglerstruktur.

### Objektbeschreibung

| | |
|---|---|
| Index | $321B_h$ |
| Objektname | Velocity Controller Parameters |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2213-B1028181 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Proportional Gain Kp [mA/Hz] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000140_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Integrator Time Ti [µs] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000C350_h$ |

## 321Ch Position Controller Parameters

### Funktion

Enthält die Parameter für den Positionsregler. Siehe Kapitel Reglerstruktur.

### Objektbeschreibung

| | |
|---|---|
| Index | $321C_h$ |
| Objektname | Position Controller Parameters |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2213-B1028181 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Proportional Gain Kp [Hz] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000008F_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Integrator Time Ti [µs] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## 321Dh Feedforward

### Funktion

Enthält die Parameter für die Vorsteuerung. Siehe Kapitel Reglerstruktur.

### Objektbeschreibung

| | |
|---|---|
| Index | $321D_h$ |
| Objektname | Feedforward |
| Object Code | ARRAY |
| Datentyp | UNSIGNED16 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2213-B1028181 |
| Änderungshistorie | Firmware Version FIR-v2315-B1040535: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "UNSIGNED16".<br>Firmware Version FIR-v2315-B1040535: Eintrag "Data type" geändert von "UNSIGNED32" auf "UNSIGNED16".<br>Firmware Version FIR-v2315-B1040535: Eintrag "Data type" geändert von "UNSIGNED32" auf "UNSIGNED16".<br>Firmware Version FIR-v2315-B1040535: Eintrag "Data type" geändert von "UNSIGNED32" auf "UNSIGNED16".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Object Name" geändert von "Pre-control" auf "Feedforward".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Voltage Pre-control For Dq-decoupling [‰]" auf "Voltage Feedforward For Dq-decoupling [‰]".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Acceleration Pre-control [‰]" auf "Acceleration Feedforward [‰]".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Velocity Pre-control [‰]" auf "Velocity Feedforward [‰]".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Ohmic Based Voltage Pre-control [‰]" auf "Ohmic Based Voltage Feedforward [‰]". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Voltage Feedforward For Dq-decoupling [‰] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $03E8_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Acceleration Feedforward [‰] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $03E8_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Velocity Feedforward [‰] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $03E8_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Ohmic Based Voltage Feedforward [‰] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

## 321Eh Voltage Limit

### Funktion

Maximal zulässige PWM-Spannung (Tastverhältnis) in Millivolt. Siehe auch Kapitel Reglerstruktur.

### Objektbeschreibung

| | |
|---|---|
| Index | $321E_h$ |
| Objektname | Voltage Limit |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |

| | |
|---|---|
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $000186A0_h$ |
| Firmware Version | FIR-v2213-B1028181 |
| Änderungshistorie | |

## Beschreibung

Von diesem Wert hängt auch ab, ob die *Übermodulation* des Spannungsvektors verwendet wird. Wird die *Übermodulation* verwendet, kann ein höheres Drehmoment erreicht werden. Die resultierende Spannung ist aber nicht mehr sinusförmig, was zu Oberschwingungen und höheren Verlusten führen kann.

| Wert in mV | Übermodulation |
|---|---|
| 1001…$U_{o_low}$ | Keine; der Spannungsvektor beschreibt einen Kreis. |
| $U_{o_low}$…$U_{o_high}$ | Der Spannungsvektor beschreibt einen Kreis, der an vier/ sechs Seiten immer mehr abgeflacht wird, proportional zum eingestellten Wert. |
| ≥$U_{o_high}$ | Volle; Der Spannungsvektor beschreibt einen Quadrat bzw. ein Sechseck. |

**$U_{o_low}$**

Die niedrigste Spannung, ab welcher eine Übermodulation stattfindet. Wird wie folgt berechnet (geringe Abweichungen abhängig von der Hardware möglich):

Betriebsspannung*0,95

**$U_{o_high}$**

Ab dieser Spannung findet die maximale Übermodulation statt. Wird wie folgt berechnet (geringe Abweichungen abhängig von der Hardware möglich):

bei zweiphasigen Schrittmotoren: Betriebsspannung*1,128

bei dreiphasigen BLDC-Motoren: Betriebsspannung*1,05

# 3220h Analog Input Digits

## Funktion

Zeigt die Momentanwerte der Analogeingänge in *ADC Digits* an.

## Objektbeschreibung

| | |
|---|---|
| Index | $3220_h$ |
| Objektname | Analog Input Digits |
| Object Code | ARRAY |
| Datentyp | INTEGER16 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |

| | |
|---|---|
| Vorgabewert | |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Analog Input Digits |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $02_h$ |
| Name | Analog Input #1 - #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 Digits * 24 V / 4095 Digits
- Stromeingang (falls konfigurierbar): x Digits * 20 mA / 1023 Digits

## 323Ah User Pin Settings

### Funktion

Mit diesem Objekt können Sie die digitalen Ein- / Ausgänge konfigurieren, wie in Kapitel Digitale Ein- und Ausgänge beschrieben .

### Objektbeschreibung

| | |
|---|---|
| Index | $323A_h$ |
| Objektname | User Pin Settings |
| Object Code | ARRAY |
| Datentyp | UNSIGNED8 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2339-B1048823 |

| Änderungshistorie | |
|---|---|

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Voltage Level Select |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Pull-Up Enable |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

## Beschreibung

- Subindex $01_h$: hier legen Sie den Pegel für die Ein-/Ausgänge fest::
  - Wert "0": 5 V
  - Wert "1": 24 V (Eingänge) bzw. +UB _Logic (Ausgänge)

**HINWEIS**

Verwenden Sie für die Eingänge immer eine Spannung, die kleiner ist als die Spannung +UB _Logic.

- Subindex $02_h$: hier legen Sie den Typ der digitalen Eingänge fest:
  - Wert "0" (Pull-Down): High-Pegel bei 5/24 V am Pin.
  - Wert "1" (Pull-Up): High-Pegel ohne externe Spannung am Pin.

## 3241h Digital Input Position Capture

### Funktion

Mit diesem Objekt kann automatisch die Sensorposition notiert werden, wenn am digitalen Eingang des Referenzschalters bzw. an zwei weiteren Eingängen ein Pegelwechsel stattfindet.

### Objektbeschreibung

| | |
|---|---|
| Index | $3241_h$ |
| Objektname | Digital Input Position Capture |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1446 |
| Änderungshistorie | Firmware Version FIR-v1446: Eintrag "Data type" geändert von "UNSIGNED32" auf "UNSIGNED8".<br>Firmware Version FIR-v1738-B501312: Eintrag "Name" geändert von "Encoder Raw Value" auf "Sensor Raw Value".<br>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "TX-PDO".<br>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "RX-PDO" auf "TX-PDO".<br>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "RX-PDO" auf "TX-PDO".<br>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 03 geändert von "RX-PDO" auf "TX-PDO".<br>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 04 geändert von "RX-PDO" auf "TX-PDO". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0C_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Control For Capture Of Home Switch |
| Datentyp | UNSIGNED32 |

| | |
|---|---|
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Capture Count For Capture Of Home Switch |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Position Value For Capture Of Home Switch |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Sensor Raw Value For Capture Of Home Switch |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Control For Capture Of Input #3 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Capture Count For Capture Of Input #3 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |

| | |
|---|---|
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $07_h$ |
| Name | Position Value For Capture Of Input #3 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $08_h$ |
| Name | Sensor Raw Value For Capture Of Input #3 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $09_h$ |
| Name | Control For Capture Of Input #4 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $0A_h$ |
| Name | Capture Count For Capture Of Input #4 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $0B_h$ |
| Name | Position Value For Capture Of Input #4 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $0C_h$ |

| | |
|---|---|
| Name | Sensor Raw Value For Capture Of Input #4 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

### Beschreibung

- Subindex $01_h$, $05_h$, $09_h$: 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 $02_h$. $06_h$, $0A_h$: 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 $03_h$, $07_h$, $0B_h$: Encoder Position des Pegelwechsels (in absoluten Benutzereinheiten wie in $6064_h$)
- Subindex $04_h$, $08_h$, $0C_h$: Encoder Position des Pegelwechsels (in Sensor-Inkrementen wie in $6063_h$)

## 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 | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $20_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Input Source For Bit #0 In 60FDh - Negative Limit Switch |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Input Source For Bit #1 In 60FDh - Positive Limit Switch |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Input Source For Bit #2 In 60FDh - Home Switch |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Input Source For Bit #3 In 60FDh - Interlock |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Input Source For Bit #4 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Input Source For Bit #5 In 60FDh |
| Datentyp | UNSIGNED8 |

| | |
|---|---|
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $07_h$ |
| Name | Input Source For Bit #6 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $08_h$ |
| Name | Input Source For Bit #7 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $09_h$ |
| Name | Input Source For Bit #8 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $0A_h$ |
| Name | Input Source For Bit #9 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $0B_h$ |
| Name | Input Source For Bit #10 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |

| | |
|---|---|
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $0C_h$ |
| Name | Input Source For Bit #11 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $0D_h$ |
| Name | Input Source For Bit #12 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $0E_h$ |
| Name | Input Source For Bit #13 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $0F_h$ |
| Name | Input Source For Bit #14 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $10_h$ |
| Name | Input Source For Bit #15 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $11_h$ |

| | |
|---|---|
| Name | Input Source For Bit #16 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $01_h$ |

| | |
|---|---|
| Subindex | $12_h$ |
| Name | Input Source For Bit #17 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $13_h$ |
| Name | Input Source For Bit #18 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $03_h$ |

| | |
|---|---|
| Subindex | $14_h$ |
| Name | Input Source For Bit #19 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $15_h$ |
| Name | Input Source For Bit #20 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $05_h$ |

| | |
|---|---|
| Subindex | $16_h$ |
| Name | Input Source For Bit #21 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |

| | |
|---|---|
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $06_h$ |

| | |
|---|---|
| Subindex | $17_h$ |
| Name | Input Source For Bit #22 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $5A_h$ |

| | |
|---|---|
| Subindex | $18_h$ |
| Name | Input Source For Bit #23 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $5B_h$ |

| | |
|---|---|
| Subindex | $19_h$ |
| Name | Input Source For Bit #24 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $07_h$ |

| | |
|---|---|
| Subindex | $1A_h$ |
| Name | Input Source For Bit #25 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $08_h$ |

| | |
|---|---|
| Subindex | $1B_h$ |
| Name | Input Source For Bit #26 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $09_h$ |

| | |
|---|---|
| Subindex | $1C_h$ |
| Name | Input Source For Bit #27 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0A_h$ |

| | |
|---|---|
| Subindex | $1D_h$ |
| Name | Input Source For Bit #28 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0B_h$ |

| | |
|---|---|
| Subindex | $1E_h$ |
| Name | Input Source For Bit #29 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0C_h$ |

| | |
|---|---|
| Subindex | $1F_h$ |
| Name | Input Source For Bit #30 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0D_h$ |

| | |
|---|---|
| Subindex | $20_h$ |
| Name | Input Source For Bit #31 In 60FDh |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0E_h$ |

## 324Ah Inputs

### Funktion

Aus diesem Objekt lesen sie den aktuellen Status der Eingänge aus.

### Objektbeschreibung

| | |
|---|---|
| Index | $324A_h$ |
| Objektname | Inputs |
| Object Code | ARRAY |
| Datentyp | UNSIGNED16 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2339-B1048823 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | User Inputs |
| Datentyp | UNSIGNED16 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Encoder Inputs |
| Datentyp | UNSIGNED16 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

### Beschreibung

- Subindex $01_h$: zeigt den Status folgender Eingänge::
  - Bits 0 bis 5: digitaler Eingang 1 bis 6 (genaue Anzahl Produktabhängig)
  - Bits 6 und 7: Analogeingang 1 und 2 als digitaler Eingang (genaue Anzahl Produktabhängig)
- Subindex $02_h$: zeigt den Status folgender Sensor-Eingänge::
  - Bits 0 bis 2: Hallsensor 1 bis 3 (falls vorhanden)
  - Bits 3 bis 5: Kanal A, B, Index des ersten Inkrementalencoders (falls vorhanden)
  - Bits 6 bis 8: Kanal A, B, Index des zweiten Inkrementalencoders (falls vorhanden)

## 3250h Digital Outputs Control

### Funktion

### Objektbeschreibung

| | |
|---|---|
| Index | $3250_h$ |
| Objektname | Digital Outputs Control |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | Firmware Version FIR-v1426: Subindex $01_h$: Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable"<br>Firmware Version FIR-v1446: Eintrag "Name" geändert von "Special Function Enable" auf "No Function".<br>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 6 auf 9.<br>Firmware Version FIR-v2039: Subindex 09 hinzugefügt |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $01_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Enable Mask [Bit0=StatusLed, Bit1=ErrorLed] |
| Datentyp | UNSIGNED32 |

| | |
|---|---|
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $FFFFFFFF_h$ |

## 3252h Digital Output Routing

### Funktion

Dieses Objekt weist einem Ausgang eine Signalquelle zu, die mit dem $60FE_h$ kontrolliert werden kann. Details finden Sie im Kapitel *Output Routing*.

### 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-v1540 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Control Bit Of 60FEh:1h And Source For Brake Output" auf "Control Bit Of 60FEh:1h And Source For Brake Output".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Control Bit Of 60FEh:1h And Source For Output #1" auf "Control Bit Of 60FEh:1h And Source For Output #1".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Control Bit Of 60FEh:1h And Source For Output #2" auf "Control Bit Of 60FEh:1h And Source For Output #2".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Control Bit Of 60FEh:1h And Source For Output #3" auf "Control Bit Of 60FEh:1h And Source For Output #3". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Control Bit Of 60FEh:1h And Source For Brake Output |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $1080_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Control Bit Of 60FEh:1h And Source For Output #1 |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0010_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Control Bit Of 60FEh:1h And Source For Output #2 |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0011_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Control Bit Of 60FEh:1h And Source For Output #3 |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0012_h$ |

## 325Ah Outputs

### Funktion

Mit diesem Objekt steuern Sie die digitalen Ausgänge (Alternative zu 60FEh Digital Outputs).

### Objektbeschreibung

| | |
|---|---|
| Index | $325A_h$ |
| Objektname | Outputs |
| Object Code | ARRAY |
| Datentyp | UNSIGNED16 |
| Speicherbar | ja, Kategorie: Applikation |

| | |
|---|---|
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2339-B1048823 |
| Änderungshistorie | Firmware Version FIR-v2412-B1057638: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Applikation". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $01_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | User Outputs |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

### Beschreibung

- Subindex $01_h$:
  - Bits 0 bis 6: digitaler Ausgang 1 bis 7 (genaue Anzahl Produktabhängig)
  - Bit 7: Bremsenausgang (falls vorhanden)

## 3260h Pwm Output 0

### Funktion

Mit diesem Objekt konfigurieren Sie den ersten PWM-Ausgang. Den Ausgang müssen Sie als PWM-Ausgang mittels *Output Routing* festlegen.

### Objektbeschreibung

| | |
|---|---|
| Index | $3260_h$ |
| Objektname | Pwm Output 0 |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |

| | |
|---|---|
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1939-B682906 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Pwm Frequency |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00002710_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Pwm Duty Cycle |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

### Beschreibung

Die Subindizes haben folgende Funktion:

- $01_h$: Frequenz des PWM-Signals in Hertz. 50…20000
- $02_h$: Tastgrad (Duty-Cycle) des PWM-Signals: 0…100

## 3261h Pwm Output 1

### Funktion

Mit diesem Objekt konfigurieren Sie den zweiten PWM-Ausgang. Den Ausgang müssen Sie als PWM-Ausgang mittels *Output Routing* festlegen.

## Objektbeschreibung

| | |
|---|---|
| Index | $3261_h$ |
| Objektname | Pwm Output 1 |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1939-B682906 |
| Änderungshistorie | |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Pwm Frequency |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00002710_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Pwm Duty Cycle |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## Beschreibung

Die Subindizes haben folgende Funktion:

- $01_h$: Frequenz des PWM-Signals in Hertz. 50…20000
- $02_h$: Tastgrad (Duty-Cycle) des PWM-Signals: 0…100

## 3262h Pwm Output 2

### Funktion

Mit diesem Objekt konfigurieren Sie den dritten PWM-Ausgang. Den Ausgang müssen Sie als PWM-Ausgang mittels *Output Routing* festlegen.

### Objektbeschreibung

| | |
|---|---|
| Index | $3262_h$ |
| Objektname | Pwm Output 2 |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1939-B682906 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Pwm Frequency |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00002710_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Pwm Duty Cycle |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

### Beschreibung

Die Subindizes haben folgende Funktion:

Die Subindizes haben folgende Funktion:

- $01_h$: Frequenz des PWM-Signals in Hertz. 50…20000
- $02_h$: Tastgrad (Duty-Cycle) des PWM-Signals: 0…100

## 3273h Generic SPI Hardware Configuration

### Funktion

Siehe Kapitel Generic SPI.

### Objektbeschreibung

| | |
|---|---|
| Index | $3273_h$ |
| Objektname | Generic SPI Hardware Configuration |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2213-B1029645 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $01_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Hardware Feature Control |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## 3274h Generic SPI Mosi Data

### Funktion

Siehe Kapitel Generic SPI.

### Objektbeschreibung

| | |
|---|---|
| Index | $3274_h$ |
| Objektname | Generic SPI Mosi Data |
| Object Code | ARRAY |
| Datentyp | UNSIGNED8 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2213-B1029645 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Length Of SPI Message To Be Sent |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $1F_h$ |
| Name | Generic SPI Mosi Data Byte #1 - #31 |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

## 3275h Generic SPI Miso Data

### Funktion

Siehe Kapitel Generic SPI.

### Objektbeschreibung

| | |
|---|---|
| Index | $3275_h$ |

| | |
|---|---|
| Objektname | Generic SPI Miso Data |
| Object Code | ARRAY |
| Datentyp | UNSIGNED8 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2213-B1029645 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Length Of Received SPI Message |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $1F_h$ |
| Name | Generic SPI Miso Data Byte #1 - #31 |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

## 3320h Analog Input Values

### Funktion

Dieses Objekt zeigt die Momentanwerte der Analogeingänge in benutzerdefinierten Einheiten an.

### Objektbeschreibung

| | |
|---|---|
| Index | $3320_h$ |
| Objektname | Analog Input Values |
| Object Code | ARRAY |
| Datentyp | INTEGER32 |
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |

| | |
|---|---|
| Firmware Version | FIR-v1426 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Analog Input Values |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $02_h$ |
| Name | Analog Input #1 - #2 |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

### Beschreibung

Die benutzerdefinierten Einheiten setzten sich aus Offset ($3321_h$) und Skalierungswert ($3322_h$/ $3323_h$) zusammen. Sind beide 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 Digits * 24 V / 4095 Digits
- Stromeingang (falls konfigurierbar): x Digits * 20 mA / 1023 Digits

Für die Subeinträge gilt:

- Subindex $00_h$: Anzahl der Analogeingänge
- Subindex $01_h$: Analogwert 1
- Subindex $02_h$: Analogwert 2 (falls vorhanden)

## 3321h Analog Input Offsets

### Funktion

Offset, der zum eingelesenen Analogwert ($3220_h$) addiert wird, bevor die Skalierung (Multiplikator aus dem Objekt 3322 und Teiler aus dem Objekt $3323_h$) vorgenommen wird.

### Objektbeschreibung

| | |
|---|---|
| Index | $3321_h$ |
| Objektname | Analog Input Offsets |
| Object Code | ARRAY |
| Datentyp | INTEGER16 |

| | |
|---|---|
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2139-B1022383 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Analog Input Offsets |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $02_h$ |
| Name | Analog Input #1 - #2 |
| Datentyp | INTEGER16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

## 3322h Analog Input Numerators

### Funktion

Wert, mit dem der eingelesene Analogwert ($3220_h$ + $3321_h$) multipliziert wird, bevor er in das Objekt $3320_h$ geschrieben wird.

### Objektbeschreibung

| | |
|---|---|
| Index | $3322_h$ |
| Objektname | Analog Input Numerators |
| Object Code | ARRAY |
| Datentyp | INTEGER16 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Analog Input Numerators |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $02_h$ |
| Name | Analog Input #1 - #2 |
| Datentyp | INTEGER16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $6098_h$ |

### Beschreibung

Die Subindizes enthalten:

- Subindex $01_h$: Multiplikator für Analogeingang 1
- Subindex $02_h$: Multiplikator für Analogeingang 2 (falls vorhanden)

## 3323h Analog Input Denominators

### Funktion

Wert, mit dem der eingelesene Analogwert ($3220_h$+ $3321_h$) dividiert wird, bevor er in das Objekt $3320_h$ geschrieben wird.

### Objektbeschreibung

| | |
|---|---|
| Index | $3323_h$ |
| Objektname | Analog Input Denominators |
| Object Code | ARRAY |
| Datentyp | INTEGER16 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1926-B648637 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Analog Input Denominators |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $02_h$ |
| Name | Analog Input #1 - #2 |
| Datentyp | INTEGER16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0FFF_h$ |

### Beschreibung

Die Subindizes enthalten:

- Subindex $01_h$: Teiler für Analogeingang 1
- Subindex $02_h$: Teiler für Analogeingang 2 (falls vorhanden)

## 3380h Feedback Sensorless

### Funktion

Enthält Mess- und Konfigurations-Werte, die für die sensorlose Regelung und die Feldschwächung im Closed Loop notwendig sind.

### Objektbeschreibung

| | |
|---|---|
| Index | $3380_h$ |
| Objektname | Feedback Sensorless |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Sensors |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2013-B726332 |
| Änderungshistorie | Firmware Version FIR-v2013-B726332: Die Anzahl der Einträge haben sich geändert von 7 auf 6.<br>Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Sensors". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $05_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Resistance [Ohm] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Inductance [H] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Magnetic Flux [Vs] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Switch On Speed [rpm] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000078_h$ |

| | |
|---|---|
| Subindex | $05_h$ |

| | |
|---|---|
| Name | Switch Off Speed [rpm] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000064_h$ |

### Beschreibung

Die Subindizes haben folgende Funktion:

- $01_h$: Wicklungswiderstand. Float-Wert, hier als UNSIGNED32 dargestellt. Wird vom Auto-Setup ermittelt.
- $02_h$: Wicklungsinduktivität. Float-Wert, hier als UNSIGNED32 dargestellt. Wird vom Auto-Setup ermittelt.
- $03_h$: Verkettungsfluss. Float-Wert, hier als UNSIGNED32 dargestellt. Wird vom Auto-Setup ermittelt.
- $04_h$: Einschaltdrehzahl in U/min. Ab dieser Drehzahl wird der *Closed Loop* ( *Sensorless*) aktiviert, wenn vom Auto-Setup keine Sensoren erkannt wurden.
- $05_h$: Ausschaltdrehzahl in U/min. Unter dieser Drehzahl wird der *Closed Loop* ( *Sensorless*) deaktiviert, wenn vom Auto-Setup keine Sensoren erkannt wurden.

## 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: Sensors |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1748-B531667 |
| Änderungshistorie | Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "1st Alignment" auf "Alignment Of H1 Edge While H3 Low And H2 Low".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "2nd Alignment" auf "Alignment Of H1 Edge While H3 Low And H2 High".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "3rd Alignment" auf "Alignment Of H1 Edge While H3 High And H2 Low".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "4th Alignment" auf "Alignment Of H1 Edge While H3 High And H2 High". |

| | |
|---|---|
| | Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "5th Alignment" auf "Alignment Of H2 Edge While H3 Low And H1 Low".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "6th Alignment" auf "Alignment Of H2 Edge While H3 Low And H1 High".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "7th Alignment" auf "Alignment Of H2 Edge While H3 High And H1 Low".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "8th Alignment" auf "Alignment Of H2 Edge While H3 High And H1 High".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "9th Alignment" auf "Alignment Of H3 Edge While H2 Low And H1 Low".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "10th Alignment" auf "Alignment Of H3 Edge While H2 Low And H1 High".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "11th Alignment" auf "Alignment Of H3 Edge While H2 High And H1 Low".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "12th Alignment" auf "Alignment Of H3 Edge While H2 High And H1 High".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Sensors". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0C_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Alignment Of H1 Edge While H3 Low And H2 Low |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Alignment Of H1 Edge While H3 Low And H2 High |

| | |
|---|---|
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Alignment Of H1 Edge While H3 High And H2 Low |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Alignment Of H1 Edge While H3 High And H2 High |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Alignment Of H2 Edge While H3 Low And H1 Low |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Alignment Of H2 Edge While H3 Low And H1 High |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $07_h$ |
| Name | Alignment Of H2 Edge While H3 High And H1 Low |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |

| | |
|---|---|
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $08_h$ |
| Name | Alignment Of H2 Edge While H3 High And H1 High |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $09_h$ |
| Name | Alignment Of H3 Edge While H2 Low And H1 Low |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $0A_h$ |
| Name | Alignment Of H3 Edge While H2 Low And H1 High |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $0B_h$ |
| Name | Alignment Of H3 Edge While H2 High And H1 Low |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $0C_h$ |
| Name | Alignment Of H3 Edge While H2 High And H1 High |
| 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: Sensors |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1738-B501312 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Sensors". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $03_h$ |

| | |
|---|---|
| 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 | $0000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Latency [ns] |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

### Beschreibung

Die Subindizes haben folgende Funktion:

- $01_h$ (Configuration): folgende Bits haben eine Bedeutung:
  - Bit 0: Wert = "0": Der Encoder verfügt nicht über einen Index. Wert = "1" : Encoder-Index ist vorhanden und soll verwendet werden.
- $02_h$ (Alignment): Dieser Wert gibt den Versatz zwischen dem Index des Encoders und den Magneten des Rotors an.
  Die exakte Bestimmung ist über das Auto-Setup möglich. Das Vorhandensein dieses Wertes ist für den *Closed Loop*-Betrieb mit Encoder erforderlich.
- $03_h$ (Latency): Hier können Sie die Latenz des verwendeten Encoders eintragen.

## 33B0h Feedback SSI 1

### Funktion

Enthält Konfigurationswerte für den ersten SSI-Encoder.

### Objektbeschreibung

| | |
|---|---|
| Index | $33B0_h$ |
| Objektname | Feedback SSI 1 |
| Object Code | RECORD |
| Datentyp | SSI ENCODER |
| Speicherbar | ja, Kategorie: Sensors |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1939-B682906 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Sensors". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |

| | |
|---|---|
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0D_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Configuration |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Alignment |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Home Position Low |
| Datentyp | INTEGER32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Home Position High |
| Datentyp | INTEGER32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Number Of Bits For Transfer |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |

| | |
|---|---|
| Zulässige Werte | |
| Vorgabewert | $17_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Baud Rate |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $000F4240_h$ |

| | |
|---|---|
| Subindex | $07_h$ |
| Name | Position Bitmask Low |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $007FFFC0_h$ |

| | |
|---|---|
| Subindex | $08_h$ |
| Name | Position Bitmask High |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $09_h$ |
| Name | Status Bitmask Low |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $0A_h$ |
| Name | Status Bitmask High |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $0B_h$ |
| Name | Status Value Low |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $0C_h$ |
| Name | Status Value High |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $0D_h$ |
| Name | Latency [ns] |
| Datentyp | INTEGER32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## Beschreibung

Die Subindizes haben folgende Funktion:

- $01_h$ (Configuration):
  - Bit 0: Wert = "0": Das Alignment wurde noch nicht ermittelt bzw. soll nicht verwendet werden. Wert = "1" : Das Alignment ist vorhanden und soll verwendet werden.
- $02_h$ (Alignment): Dieser Wert gibt den Versatz zwischen der Nullposition des Encoders und den Magneten des Rotors an.
  Die exakte Bestimmung ist nur über das Auto-Setup möglich. Das Vorhandensein dieses Wertes ist für den *Closed Loop*-Betrieb mit Encoder erforderlich.
- $03_h$ (Home Position Low) und $04_h$ (Home Position High): In diese Subindizes wird die absolute Encoderposition eingetragen, nachdem ein Homing abgeschlossen ist.
- $05_h$ (Number Of Bits For Transfer): Anzahl der Bits einer Nachricht (Encoder-Daten). Maximal 64 Bits.
- $06_h$ (Baud Rate): Baudrate der Schnittstelle in Hertz. Folgende Frequenzen werden unterstützt: 27 MHz, 13,5 MHz, 6,75 MHz, 3,375 MHz, 1,6875 MHz, 656,25 kHz, 843,75 kHz, 421,875 kHz, 210,9375 kHz, 105,46875 kHz. Bei abweichenden Werten wird die gültige Frequenz mit der kleinsten Differenz gewählt.
- $07_h$ (Position Bitmask Low) und $08_h$ (Position Bitmask High): In diese Subindizes tragen Sie eine Bitmaske ein, die bestimmt, welche Bits der Encoder-Daten die Positionsdaten enthalten (siehe nachfolgende Anleitung).
- $09_h$ (Status Bitmask Low) und $0A_h$ (Status Bitmask High): In diese Subindizes tragen Sie eine Bitmaske ein, die bestimmt, welche Bits der Encoder-Daten die Statusinformationen enthalten (siehe nachfolgende Anleitung).
- $0B_h$ (Status Value Low) und $0C_h$ (Status Value High): In diese Subindizes tragen Sie eine Bitmaske ein, die bestimmt, welchen Wert die Statusinformationen-Bits (Subindizes $09_h$ und $0A_h$) haben müssen (siehe

nachfolgende Anleitung). Ein anderer Wert an dieser Stelle de Encoder-Daten wird von der Steuerung als Fehler interpretiert.

- $0D_h$ (Latency): Latenz des Encoders in Nanosekunden, den Encoderdatenblatt zu entnehmen.

Um die Konfiguration entsprechend Ihres Encoders einzustellen:

1. Stellen Sie die Baudrate in Subindex $06_h$ und die Anzahl der Bits in Subindex $05_h$ ein, entsprechend dem Encoder-Datenblatt.
2. Legen Sie fest, welche Bits die Positionsdaten enthalten sollen und setzen Sie die Subindizes $07_h$ und $08_h$ auf den entsprechenden Wert.
3. Legen Sie fest, welche Bits die Statusinformationen (z. B.Status, Error etc.) enthalten sollen und setzen Sie die Subindizes $09_h$ und $0A_h$ auf den entsprechenden Wert.
4. Legen Sie fest, welchen Wert, "0" oder "1", die Statusinformationen-Bits haben müssen und setzen Sie die entsprechenden Bits in den Subindizes $09_h$ und $0A_h$ auf den Wert.
5. Speichern Sie das Objekt, indem Sie den Wert "$65766173_h$" in $1010_h$:$06_h$ schreiben, und starten Sie die Steuerung neu.

**Beispiel**

Der Encoder sendet die Daten in einer 32-Bit-Nachricht. Bits 4…23 enthalten die Position. Die Statusinformationen sind in folgenden Bits geteilt:

- Bits 0…2 sind Status-Bits, die immer den Wert "0" haben müssen
- Bit 3 ist das Error-Bit, das den Wert "0" hat, wenn ein Fehler aufgetreten ist
- Bit 31 signalisiert den Beginn der Nachricht und hat immer den Wert "1"

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | POS | POS | POS | POS | POS | POS |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| POS | POS | POS | POS | POS | POS | POS | POS | POS | POS | POS | POS | S | E | E | PAR |

Sie müssen in die Subindizes folgende Werte eintragen:

- $05_h$ (Number Of Bits For Transfer): $20_h$
- $07_h$(Position Bitmask Low) $00FFFFF0_h$
- $09_h$ (Status Bitmask Low): $8000\ 000F_h$
- $0B_h$ (Status Value Low): $8000\ 000F8_h$

Die Subindizes $08_h$, $0A_h$ und $0C_h$, welche die höchstwertige 32 Bits einer 64-Bit-Nachricht enthalten würden, haben den Wert "0".

## 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 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) |
| 2 | Abbremsen mit *quick stop ramp* ($6085_h$) |
| 3 bis 32767 | reserviert |

## 3701h Limit Switch Error Option Code

### Funktion

Wird ein Endschalter ausgelöst, so wird die Endschalterposition intern gespeichert, das Bit 7 ( *Warning*) in $6041_h$ ( *Statusword*) gesetzt und die CiA 402 Power State Machine in den Zustand *Quick Stop Active* versetz. Die in diesem Objekt hinterlegte Aktion wird dabei ausgeführt. Siehe Kapitel Begrenzung des Bewegungsbereichs.

### 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 |
|---|---|
| -2 | keine Reaktion, verwerfen der Endschalterposition |

| Wert in Objekt $3701_h$ | Beschreibung |
|---|---|
| -1 (Werkseinstellung) | keine Reaktion (um z. B. eine Referenzfahrt durchzuführen), außer Vermerken der Endschalterposition |
| 0 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen (Zustand *Switch on disabled*) |
| 1 | Abbremsen mit *slow down ramp* (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in *Switch on disabled* |
| 2 | Abbremsen mit *quick stop ramp* und anschließendem Zustandswechsel in *Switch on disabled* |
| 5 | Abbremsen mit *slow down ramp* (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in *Quick Stop Active*; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand *Operation enabled* schalten. |
| 6 | Abbremsen mit *quick stop ramp* und anschließendem Zustandswechsel in *Quick Stop Active*; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand *Operation enabled* schalten. |

## 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 | Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".<br>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 02 geändert von "lesen/schreiben" auf "nur lesen".<br>Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Temperature PCB [d?C]" auf "Temperature PCB [Celsius * 10]".<br>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 03 geändert von "lesen/schreiben" auf "nur lesen".<br>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 4 auf 6.<br>Firmware Version FIR-v2451-B1068465: Die Anzahl der Einträge haben sich geändert von 6 auf 5. |

| | |
|---|---|
| | Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Temperature PCB [Celsius * 10]" auf "Temperature PCB [d°C]".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Name" geändert von "Temperature Motor [Celsius * 10]" auf "Temperature Microcontroller Chip [d°c]". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| 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 [d°C] |
| Datentyp | INTEGER32 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Temperature Microcontroller Chip [d°c] |
| 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)
- $04_h$: aktuelle Temperatur des Prozessors in [d°C] (Zehntelgrad)

## 4015h Special Drive Modes

### Funktion

Mit diesem Objekt können Sie die *speziellen Fahrmodi* aus- oder einschalten. Siehe Kapitel .

### Objektbeschreibung

| | |
|---|---|
| Index | $4015_h$ |
| Objektname | Special Drive Modes |
| Object Code | ARRAY |
| Datentyp | UNSIGNED8 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1738-B501312 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $02_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Special Drive Mode Configuration |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |

| | |
|---|---|
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Virtual Config Switch Value |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

### Beschreibung

Die Subindizes haben folgende Funktionen:

- $01_h$:
  - Wert="$0"_h$: die *speziellen Fahrmodi* werden ausgeschaltet
  - Wert="$2"_h$: die *speziellen Fahrmodi* werden eingeschaltet und der Modus wird im Subindex $02_h$ eingestellt.
- $02_h$: Definiert den verwendeten Modus.

## 4021h Ballast Configuration

### Funktion

Mit diesem Objekt schalten Sie die Ballast-Schaltung ein oder aus und bestimmen Sie deren Ansprechschwelle. Ferner konfigurieren Sie die Ballast-Überwachung. Details finden Sie im Kapitel Externe Ballast-Schaltung.

### Objektbeschreibung

| | |
|---|---|
| Index | $4021_h$ |
| Objektname | Ballast Configuration |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2013-B726332 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Applikation". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |

| | |
|---|---|
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $08_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Settings [Bit0: On/Off, Bit1: Polarity, Bit2: 0(Internal)/1(External)] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000004_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | UB Power Limit [mV] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $000186A0_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | UB Power Hysteresis [mV] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $000001F4_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Nominal Resistance [mOhm] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Long Term Energy Limit [mWs] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |

| | |
|---|---|
| Zulässige Werte | |
| Vorgabewert | $00000001_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Long Term Reference Time [ms] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $07_h$ |
| Name | Short Term Energy Limit [mWs] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $08_h$ |
| Name | Cooling Power [mW] |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

## Beschreibung

Die Subindizes haben folgende Funktion:

- $01_h$:
  - Bit 0: schaltet den Ballast ein (Wert = "1") oder aus (Wert = "0")
- $02_h$: Ansprechschwelle (Ein-/Ausschalten) der Ballast-Schaltung
- $03_h$: Hysterese für die Ansprechschwelle (Ein-/Ausschalten)
- $04_h$: Nennwert des Ballast-Widerstands
- $05_h$: Energiemenge, die dem Widerstand innerhalb der *Long Term Reference Time* (Subindex $06_h$) zugeführt werden darf, ohne ihn zu überlasten.
- $06_h$: die Bezugszeit für das *Long Term Energy Limit* (Subindex $05_h$) (typischerweise zwischen 1 und 5 Sekunden)
- $07_h$: Energiemenge, die dem Widerstand innerhalb kurzer Lastpulse (<1 Sekunde) zugeführt werden darf, ohne ihn zu überlasten.
- $08_h$: Die Wärmemenge, die der Widerstand durch Konvektion kontinuierlich an seine Umgebung abgeben kann bzw. darf.

## 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 | |

## 4042h Bootloader Infos

### Objektbeschreibung

| | |
|---|---|
| Index | $4042_h$ |
| Objektname | Bootloader Infos |
| Object Code | ARRAY |
| Datentyp | UNSIGNED32 |

| | |
|---|---|
| Speicherbar | nein |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2013-B726332 |
| Änderungshistorie | |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $03_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Bootloader Version |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Bootloader Supported Fieldbus |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Bootloader Hw-group |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

### Beschreibung

Die Subindizes haben folgende Funktionen:

- $01_h$: Version des Bootloaders. Die 4 höchstwertigen Bytes erhalten die Hauptversionsnummer, die 4 niedrigwertigsten Bytes die Nebenversionsnummer. Beispiel für die Version 4.2: $00040002_h$
- $02_h$: Vom Bootloader unterstütze Feldbusse. Die Bits haben dieselbe Funktion wie die Bits des Objekts 2101h Fieldbus Module Availability.

## 6007h Abort Connection Option Code

### Funktion

Tritt ein Fehler (Watchdog, Heartbeat etc.) auf dem Bus auf, schaltet die Steuerung automatisch den Zustand in SAFEOPERATIONAL. Mit diesem Objekt können Sie die Reaktion einstellen.

### Objektbeschreibung

| | |
|---|---|
| Index | $6007_h$ |
| Objektname | Abort Connection Option Code |
| Object Code | VARIABLE |
| Datentyp | INTEGER16 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | ■ N6-1-1-1-S: $FFFF_h$<br>■ N6-1-1-1: $FFFF_h$<br>■ N6-2-1-1-S: $FFFF_h$<br>■ N6-2-1-1: $FFFF_h$<br>■ N6-1-2-1-S: $0001_h$<br>■ N6-1-2-1: $0001_h$<br>■ N6-2-2-1-S: $0001_h$<br>■ N6-2-2-1: $0001_h$<br>■ N6-1-3-1-S: $0001_h$<br>■ N6-1-3-1: $0001_h$<br>■ N6-1-3-2-S: $0001_h$<br>■ N6-1-3-2: $0001_h$<br>■ N6-2-3-1-S: $0001_h$<br>■ N6-2-3-1: $0001_h$<br>■ N6-2-3-2-S: $0001_h$<br>■ N6-2-3-2: $0001_h$<br>■ N6-1-4-1-S: $0001_h$<br>■ N6-1-4-1: $0001_h$<br>■ N6-2-4-1-S: $0001_h$<br>■ N6-2-4-1: $0001_h$<br>■ N6-1-5-1-S: $0001_h$<br>■ N6-1-5-1: $0001_h$<br>■ N6-2-5-1-S: $0001_h$<br>■ N6-2-5-1: $0001_h$<br>■ N6-1-7-1: $0001_h$<br>■ N6-1-10-1-S: $0001_h$<br>■ N6-1-10-1: $0001_h$ |

- N6-2-10-1-S: $0001_h$
- N6-2-10-1: $0001_h$

| | |
|---|---|
| Firmware Version | FIR-v2013-B726332 |
| Änderungshistorie | |

### Beschreibung

Sie können folgende Reaktionen einstellen:

| Wert | Reaktion |
|---|---|
| -1 | Die Steuerung (Slave) setzt alle Eingangswerte (RX-PDO) auf den Wert "0". Dadurch wird auch das Objekt $6040_h$ (Controlword) auf "0" gesetzt, was ein Austrudeln des Motors zufolge hat. |
| 0 | keine Reaktion |
| 1 | *Fault*: Die in Objekt $605E_h$ hinterlegte Aktion wird ausgeführt. |
| 2 | *Disable voltage*: Übergang in den Zustand *Switched on disabled*, ohne Bremsreaktion (der Motor trudelt aus) |
| 3 | *Quick stop*: Die in Objekt $605A_h$ hinterlegte Aktion wird ausgeführt. |

## 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).

Wird der Fehler durch setzen von Bit 7 im 6040h Controlword zurückgesetzt, wird dieses Objekt auch automatisch auf "0" zurückgesetzt.

# 6040h Controlword

## Funktion

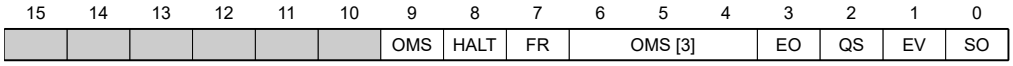
Dieses Objekt steuert die CiA 402 Power State Machine.

## 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.




**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 oder eine Warnung zurück (falls möglich)

**HALT**

Wert = "1": Löst einen Halt aus, gültig in folgenden Modi:

- Profile Position
- 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 (siehe $1003_h$)

**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. Der Wert bleibt auch nach einem Betriebsmoduswechsel auf "1", bis eine neue Zielvorgabe erfolgt.

**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 |

## 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 *Switch on disabled* |
| 1 | Abbremsen mit *slow down ramp* (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in *Switch on disabled* |
| 2 | Abbremsen mit *quick stop ramp* ($6085_h$) und anschließendem Zustandswechsel in *Switch on disabled* |
| 5 | Abbremsen mit *slow down ramp* (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in *Quick Stop Active*; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand *Operation enabled* schalten. |
| 6 | Abbremsen mit *quick stop ramp* ($6085_h$) und anschließendem Zustandswechsel in *Quick Stop Active*; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand *Operation enabled* schalten. |

Im Modus Homing wird die in $609A_h$ (Homing Acceleration) eingestellte Bremsrampoe verwendet, anstatt von $6085_h$.

## 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 | $0000_h$ |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | |

### Beschreibung

| Wert in Objekt $605B_h$ | Beschreibung |
|---|---|
| -32768 bis -1 | Reserviert |
| 0 | Sperren der Antriebsfunktion — Motor kann sich frei drehen |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in *Ready to switch on* |
| 2 bis 32767 | Reserviert |

## 605Ch Disable Operation 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 Operation 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 | Firmware Version FIR-v2412-B1057638: Eintrag "Object Name" geändert von "Disable Option Code" auf "Disable Operation Option Code". |

### Beschreibung

| Wert in Objekt $605C_h$ | Beschreibung |
|---|---|
| -32768 bis -1 | Reserviert |
| 0 | Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in *Switched on* |
| 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 in Objekt $605D_h$ | Beschreibung |
|---|---|
| -32768 bis 0 | Reserviert |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) |
| 2 | Abbremsen mit *quick stop ramp* ($6085_h$) |
| 3 bis 32767 | Reserviert |

## 605Eh Fault Reaction 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 Reaction 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 | Firmware Version FIR-v2412-B1057638: Eintrag "Object Name" geändert von "Fault Option Code" auf "Fault Reaction Option Code". |

### Beschreibung

| Wert in Objekt $605E_h$ | Beschreibung |
|---|---|
| -32768 bis -1 | Reserviert |
| 0 | Sperren der Antriebsfunktion — Motor kann sich frei drehen |
| 1 | Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus) |
| 2 | Abbremsen mit *quick stop ramp* ($6085_h$) |
| 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 |
| 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 | $6061_h$ |
| Objektname | 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 | $6062_h$ |
| Objektname | 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 Istposition in Encoder-Inkrementen. Im Gegensatz zu den Objekten $6062_h$ und $6064_h$ wird dieser Wert nach einem Homing nicht auf "0" gesetzt. Die Quelle wird in 3203h Feedback Selection bestimmt.

**HINWEIS**

Ist die Encoderauflösung im Objekt 60E6$_h$ Null, sind die Zahlenwerte dieses Objekts ungültig.

### Objektbeschreibung

| | |
|---|---|
| Index | 6063$_h$ |
| Objektname | 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. Die Quelle wird in 3203h Feedback Selection bestimmt.

### 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 "FFFFFFFF"$_h$ 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 $6068_h$ 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 6068$_h$ definierte Zeit.

## 606Bh Velocity Demand Value

### Funktion

Vorgabegeschwindigkeit in benutzerdefinierten Einheiten für den Geschwindgkeitsregler.

### 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.

## 606Ch 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 $606E_h$ 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 606E definierte Zeit (siehe auch Statusword im Modus Profile Velocity).

# 606Fh Velocity Threshold

## Funktion

Geschwindigkeit in benutzerdefinierten Einheiten, ab der die Istgeschwindigkeit im Modus Profile Velocity als ungleich Null gilt.

## Objektbeschreibung

| | |
|---|---|
| Index | $606F_h$ |
| Objektname | Velocity Threshold |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED16 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |
| Firmware Version | FIR-v2013-B726332 |
| Änderungshistorie | |

## Beschreibung

Ist die Istgeschwindigkeit größer als der Wert in $606F_h$(Velocity Threshold) für eine Zeit von $6070_h$(Velocity Threshold Time), hat das Bit 12 in $6041_h$(Statusword) den Wert "0". Sonst bleibt das Bit auf "1".

# 6070h Velocity Threshold Time

## Funktion

Zeit in Millisekunden, ab der eine Istgeschwindigkeit größer als der Wert in $606F_h$ im Modus Profile Velocity als ungleich Null gilt.

## Objektbeschreibung

| | |
|---|---|
| Index | $6070_h$ |
| Objektname | Velocity Threshold 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-v2013-B726332 |
| Änderungshistorie | |

### Beschreibung

Ist die Istgeschwindigkeit größer als der Wert in 606F$_h$(Velocity Threshold) für eine Zeit von 6070$_h$(Velocity Threshold Time), hat das Bit 12 in 6041$_h$(Statusword) den Wert "0". Sonst bleibt das Bit auf "1".

## 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 | 000A$_h$ |
| 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 6075$_h$.

Das Minimum von 6073$_h$ und 6072$_h$ wird als Limit für das Drehmoment in 6071$_h$ verwendet.

## 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 $6075_h$.

Das Minimum von $6073_h$ und $6072_h$ wird als Limit für das Drehmoment in $6071_h$ verwendet.

## 6073h Max Current

### Funktion

Enthält den Maximalstrom in Promille des eingestellten Nennstroms. Wird durch den maximalen Motorstrom ) 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%) in der Regel nicht überschreiten.

### Objektbeschreibung

| | |
|---|---|
| Index | $6073_h$ |
| Objektname | Max Current |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED16 |
| Speicherbar | ja, Kategorie: Motor |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $03E8_h$ |
| Firmware Version | FIR-v1825-B577172 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Bewegung" auf "ja, Kategorie: Motor". |

### Beschreibung

Der Maximalstrom wird in Promille des Nennstroms wie folgt berechnet:

($6073_h$*$6075_h$)/1000

Die Einheit ist mA DC, äquivalent zum Strom für eine Blockkommutierung, wie üblicherweise im Motor-Datenblatt angegeben.

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 $6075_h$.

Das Minimum von $6073_h$ und $6072_h$ wird als Limit für das Drehmoment in $6071_h$ verwendet.

## 6075h Motor Rated Current

### Funktion

Enthält den Nennstrom in mA DC, äquivalent zum Strom für eine Blockkommutierung, wie üblicherweise im Motor-Datenblatt angegeben.

### Objektbeschreibung

| | |
|---|---|
| Index | $6075_h$ |
| Objektname | Motor Rated Current |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Motor |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000708_h$ |
| Firmware Version | FIR-v1738-B501312 |

| | |
|---|---|
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Motor". |

## 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 $6075_h$.

Das Minimum von $6073_h$ und $6072_h$ wird als Limit für das Drehmoment in $6071_h$ verwendet.

## 607Ah Target Position

### Funktion

Dieses Objekt gibt die Zielposition in benutzerdefinierten Einheiten für den Profile Position und Cyclic Synchronous Position Modus an.

### Objektbeschreibung

| | |
|---|---|
| Index | $607A_h$ |
| Objektname | Target Position |
| Object Code | VARIABLE |
| Datentyp | INTEGER32 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00000E10_h$ |

| | |
|---|---|
| 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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.

## 607Eh Polarity

### Funktion

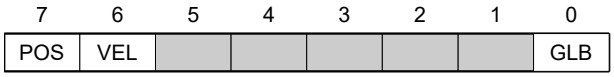
Mit diesem Objekt lässt sich die Drehrichtung umkehren.

### 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.




### GLB (Global)

Umkehr der Drehrichtung in allen Modi. Dabei ist es weder notwendig, das Autosetup erneut durchzuführen, noch die Zielgrößen anzupassen. Die Istwerte werden automatisch angepasst. Die Änderung wird nicht übernommen, wenn die *State Machine* bereits im Zustand *Operation Enabled* ist.

### VEL (Velocity)

Umkehr der Drehrichtung in folgenden Modi:

- Profile Velocity Mode
- Cyclic Synchronous Velocity Mode

### POS (Position)

Umkehr der Drehrichtung in folgenden Modi:

- Profile Position Mode
- Cyclic Synchronous Position Mode

# 607Fh Max Profile Velocity

## Funktion

Gibt die maximale Geschwindigkeit in benutzerdefinierten Einheiten an, für die Mod i Profile Torque, Profile Position und Profile Velocity.

## 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 | Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Max profile velocity" auf "Max Profile Velocity".<br>Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "INTEGER16" auf "UNSIGNED32".<br>Firmware Version FIR-v1738-B501312: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation". |

| | |
|---|---|
| | Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "nur lesen" auf "lesen/schreiben".<br>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "TX-PDO" auf "RX-PDO". |

## 6080h Max Motor Speed

### Funktion

Enthält die maximal zulässige Geschwindigkeit des Motors in benutzerdefinierten Einheiten.

### Objektbeschreibung

| | |
|---|---|
| Index | $6080_h$ |
| Objektname | Max Motor Speed |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Applikation |
| 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".<br>Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Maximum Speed" auf "Max Motor Speed".<br>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "RX-PDO".<br>Firmware Version FIR-v1748-B538662: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Bewegung".<br>Firmware Version FIR-v1825-B577172: Eintrag "Speicherbar" geändert von "ja, Kategorie: Bewegung" auf "ja, Kategorie: Tuning".<br>Firmware Version FIR-v1825-B577172: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Bewegung".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Bewegung" auf "ja, Kategorie: Applikation". |

## 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 | $6082_h$ |
| Objektname | 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 | $6083_h$ |
| Objektname | 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. Wird durch $60C6_h$ limitiert.

### 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. Wird je nach Betriebsmodus limitiert durch $60C6_h$ (Max Deceleration) und ggf. $60A4_h$ (Profile Jerk).

### 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 | $000003E8_h$ |
| 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 $6075_h$.

Das Minimum von $6073_h$ und $6072_h$ wird als Limit für das Drehmoment in $6071_h$ verwendet.

# 6091h Gear Ratio

## Funktion

Enthält die Getriebeübersetzung (Anzahl der Motorumdrehungen pro Umdrehung der Abtriebsachse) des Encoders/Sensors, der für die Positionsregelung verwendet wird (siehe 3203h Feedback Selection).

## 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".<br>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "RX-PDO". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| 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

Enthält die Vorschubskonstante (Vorschub in benutzerdefinierten Einheiten pro Umdrehung der Abtriebsachse) des Encoders/Sensors, der für die Positionsregelung verwendet wird (siehe 3203h Feedback Selection).

### 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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

## 6097h Acceleration Factor

### Funktion

Dieses Objekt beinhaltet den Faktor, der zum Umrechnen von benutzerdefinierten Beschleunigungseinheiten verwendet wird. Siehe Kapitel Benutzerdefinierte Einheiten.

### Objektbeschreibung

| | |
|---|---|
| Index | $6097_h$ |
| Objektname | 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 | Number Of Entries |
| 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

## 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 | Number Of Entries |
| 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, besonders bei hochauflösenden Encodern. Die minimale erkennbare Breite des Indeximpulses beträgt 31,25 µs.
  .
- 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 Ruckeinheitein 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 | Number Of Entries |

| | |
|---|---|
| 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 in benutzerdefinierten Einheiten 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 | Number Of Entries |
| 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.

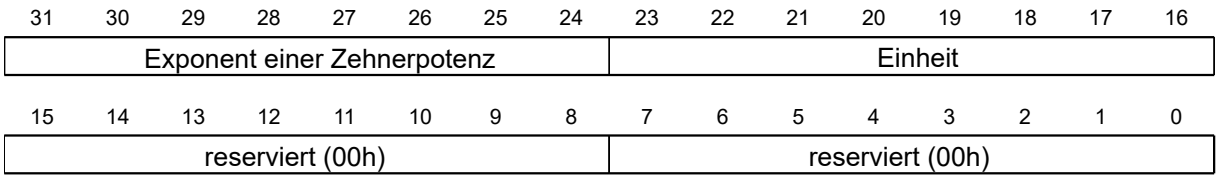
### Objektbeschreibung

| | |
|---|---|
| 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

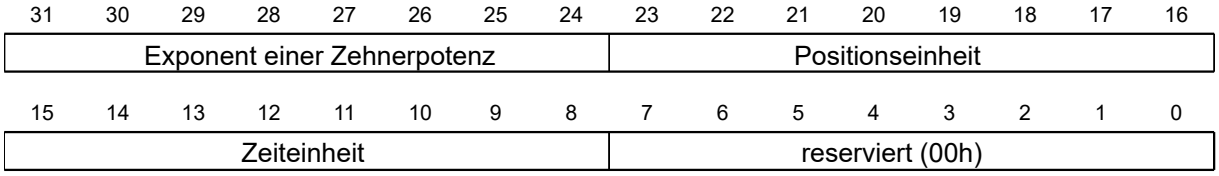
| | |
|---|---|
| Index | 60A9$_h$ |
| Objektname | SI Unit Velocity |

| | |
|---|---|
| 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 im Modus Cyclic Synchronous Position 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 | Number Of Entries |
| 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 | Number Of Entries |
| 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: Werte zwischen -5 und +4 erlaubt (-3 entspricht der Zeitbasis in Millisekunden).

Es gilt dabei: Zykluszeit = Wert des $\underline{60C2_h}$:$01_h$ * $10^{\text{Wert des 60C2:02}}$ 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 | Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 05 geändert von "lesen/schreiben" auf "nur schreiben".<br>Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 06 geändert von "lesen/schreiben" auf "nur schreiben".<br>Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".<br>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Name" geändert von "Buffer Organization" auf "Buffer Organisation". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $06_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Maximum Buffer Size |
| Datentyp | UNSIGNED32 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $02_h$ |
| Name | Actual Buffer Size |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |

| | |
|---|---|
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00000000_h$ |

| | |
|---|---|
| Subindex | $03_h$ |
| Name | Buffer Organisation |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

| | |
|---|---|
| Subindex | $04_h$ |
| Name | Buffer Position |
| Datentyp | UNSIGNED16 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |

| | |
|---|---|
| Subindex | $05_h$ |
| Name | Size Of Data Record |
| Datentyp | UNSIGNED8 |
| Zugriff | nur schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $01_h$ |

| | |
|---|---|
| Subindex | $06_h$ |
| Name | Buffer Clear |
| Datentyp | UNSIGNED8 |
| Zugriff | nur schreiben |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

## Beschreibung

Der Wert des Subindex $01_h$ enthält die maximal 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 | $60C5_h$ |
| Objektname | 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 die Betriebsmodi Profile Position, Profile Velocity und Interpolated Position Mode.

### 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".<br>Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32". |

## Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $04_h$ |
| Name | Additional Position Actual Value #1 - #4 |
| 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*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

## 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 | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $04_h$ |
| Name | Additional Velocity Actual Value #1 - #4 |
| 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*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

## 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 | UNSIGNED32 |
| Speicherbar | ja, Kategorie: Sensors |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1748-B538662 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Datentyp" geändert von "INTEGER32" auf "UNSIGNED32".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Sensors".<br>Firmware Version FIR-v2451-B1068465: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED32". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $04_h$ |
| Name | Additional Position Encoder Resolution - Encoder Increments Feedback Interface #1 - #4 |
| Datentyp | UNSIGNED32 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-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 Anzahl der Inkremente der entsprechenden Rückführung.
  Subindex $01_h$ entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

Die Auflösung der Rückführung "n" berechnet sich wie folgt:

Position Encoder Resolution = Encoder Increments ($60E6_h$:$01_h$) / Motor Revolutions ($60EB_h$:$02_h$)

**HINWEIS**

Der Wert "0" in einem Subindex bedeutet, dass die jeweilige Rückführung nicht angeschlossen ist und nicht verwendet wird. So kann z. B. die Sensorless-Funktion ausgeschaltet werden, um Rechenzeit zu sparen. Dies kann hilfreich sein, wenn ein *NanoJ*-Programm die Rechenzeit benötigt.

Steht ein Wert ungleich "0" in einem Subindex, überprüft die Steuerung beim Einschalten den entsprechenden Sensor. Im Fehlerfall (Signal nicht vorhanden, Konfiguration/Zustand ungültig etc.) wird im Statusword das Fehlerbit gesetzt und im Objekt 1003h ein Fehlercode hinterlegt.

## 60EBh 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: Sensors |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v1738-B501312 |
| Änderungshistorie | Firmware Version FIR-v2451-B1068465: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Sensors". |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |

| | |
|---|---|
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $04_h$ |
| Name | Additional Position Encoder Resolution - Motor Revolutions Feedback Interface #1 - #4 |
| 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 Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

Die Auflösung der Rückführung "n" berechnet sich wie folgt:

Position Encoder Resolution = Encoder Increments ($60E6_h$:$n_h$) / Motor Revolutions ($60EB_h$:$n_h$)

## 60F1h Additional Encoder Inversion

### Funktion

Hier wird die Polarität der Sensoren gespeichert.

### Objektbeschreibung

| | |
|---|---|
| Index | $60F1_h$ |
| Objektname | Additional Encoder Inversion |
| Object Code | ARRAY |
| Datentyp | UNSIGNED8 |
| Speicherbar | ja, Kategorie: Sensors |
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | |
| Firmware Version | FIR-v2451-B1068465 |
| Änderungshistorie | |

### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |

| | |
|---|---|
| Zugriff | nur lesen |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $04_h$ |

| | |
|---|---|
| Subindex | $01_h$ - $04_h$ |
| Name | Additional Encoder Inversion - Feedback Interface #1 - #4 |
| Datentyp | UNSIGNED8 |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $00_h$ |

## 60F2h Position Option Code

### Funktion

Das Objekt beschreibt das Positionierverhalten im Profile Position Modus.

### Objektbeschreibung

| | |
|---|---|
| Index | $60F2_h$ |
| Objektname | Position Option Code |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED16 |
| Speicherbar | ja, Kategorie: Applikation |
| Zugriff | lesen/schreiben |
| PDO-Mapping | RX-PDO |
| Zulässige Werte | |
| Vorgabewert | $0000_h$ |
| Firmware Version | FIR-v1446 |
| Änderungshistorie | Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".<br>Firmware Version FIR-v2412-B1057638: Eintrag "Object Name" geändert von "Positioning Option Code" auf "Position Option Code". |

### 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 Zielpositon 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. |
| 1 | 1 | Reserviert |

**RRO (Request-Response Option)**

Diese Bits bestimmen das Verhalten bei der Übergabe des Controlwords $6040_h$ Bit 4 ("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. |

| Bit 7 | Bit 6 | Definition |
|---|---|---|
| 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"$_h$ oder "80000000"$_h$ 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

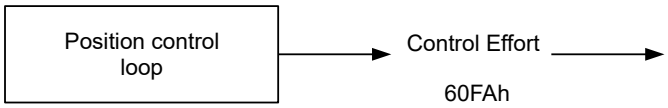
Dieses Objekt beinhaltet die Korrekturgeschwindigkeit (Stellgröße) in benutzerdefinierten Einheiten, die vom Positionsregler dem Geschwindigkeitsregler zugeführt wird.

### Objektbeschreibung

| | |
|---|---|
| Index | $60FA_h$ |
| Objektname | 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 und Integralanteil des Positionsreglers ab. Siehe auch Kapitel Closed Loop.




## 60FCh Position Demand Internal Value

### Funktion

### Objektbeschreibung

| | |
|---|---|
| Index | $60FC_h$ |
| Objektname | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | IL | HS | PLS | NLS |

**NLS (Negative Limit Switch)**
negativer Endschalter

**PLS (Positive Limit Switch)**
positiver Endschalter

**HS (Home Switch)**
Referenzschalter

**IL (Interlock)**

*Interlock*

**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". |

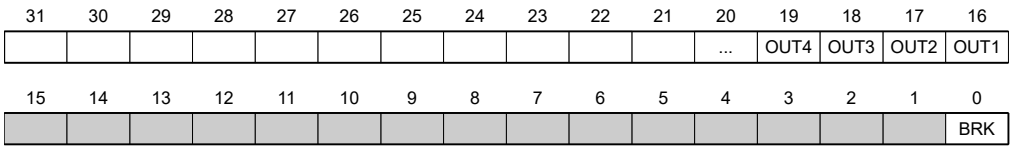
### Wertebeschreibung

| | |
|---|---|
| Subindex | $00_h$ |
| Name | Number Of Entries |
| Datentyp | UNSIGNED8 |
| Zugriff | nur lesen |
| PDO-Mapping | nein |
| Zulässige Werte | |
| Vorgabewert | $01_h$ |

| | |
|---|---|
| Subindex | $01_h$ |
| Name | Physical Outputs |
| 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.




**BRK (Brake)**

Bit für den Bremsenausgang (falls der Controller diese Funktion unterstützt):

Wert "1" heißt, dass die Bremse aktiviert wird (kein Strom kann zwischen den beiden Pins des Bremsen-Anschlusses fließen, die Bremse ist geschlossen).

**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 Velocity Mode 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 $6060_h$.

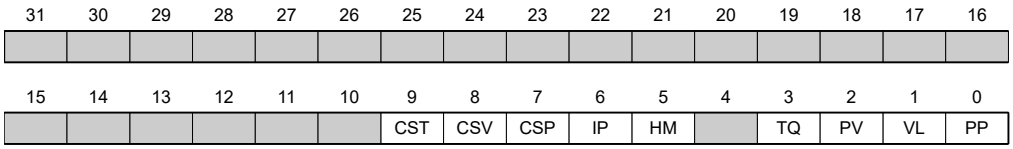
### Objektbeschreibung

| | |
|---|---|
| Index | $6502_h$ |
| Objektname | Supported Drive Modes |
| Object Code | VARIABLE |
| Datentyp | UNSIGNED32 |
| Speicherbar | nein |

| | |
|---|---|
| Zugriff | nur lesen |
| PDO-Mapping | TX-PDO |
| Zulässige Werte | |
| Vorgabewert | $000003ED_h$ |
| 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 | 0 |
| Firmware Version | FIR-v1426 |
| Änderungshistorie | |

# 11 Copyrights

## 11.1 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.

## 11.2 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

## 11.3 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.

## 11.4 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.

## 11.5 CMSIS DSP Software Library

Copyright (C) 2010 ARM Limited. All rights reserved.

## 11.6 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.

## 11.7 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>

## 11.8 littlefs

```
/*
* The little filesystem
*
* Copyright (c) 2017, Arm Limited. All rights reserved.
* SPDX-License-Identifier: BSD-3-Clause
*/
```

Copyright (c) 2017, Arm Limited. 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/orother materials provided with the distribution.
- Neither the name of ARM 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 HOLDER 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.