Technisches Handbuch PD6-E

Feldbus: EtherNet/IP™

Zu nutzen mit folgenden Varianten:

PD6-E891S95-M(S)-65-3, PD6-E891M95-M(S)-65-3, PD6-E891L95-M(S)-65-3, PD6-EB80SD-M(S)-65-3, PD6-EB80MD-M(S)-65-3, PD6-EB80LD-M(S)-65-3, PD6-EB80CD-M(S)-65-3

Gültig ab Firmware-Version FIR-v2538

Technisches Handbuch Version: 1.1.0

Inhalt

1 Einleitung ... 9
1.1 Versionshinweise ... 9
1.2 Urheberrecht, Kennzeichnung und Kontakt ... 9
1.3 Bestimmungsgemäße Verwendung ... 10
1.4 Gewährleistung und Haftungsausschluss ... 10
1.5 Zielgruppe und Qualifikation ... 10
1.6 EU-Richtlinien zur Produktsicherheit ... 10
1.7 Mitgeltende Vorschriften ... 10
1.8 Verwendete Symbole ... 10
1.9 Hervorhebungen im Text ... 11
1.10 Zahlenwerte ... 12
1.11 Bits ... 12
1.12 Zählrichtung (Pfeile) ... 12

2 Sicherheits- und Warnhinweise ... 13

3 Technische Daten und Anschlussbelegung ... 15
3.1 Umgebungsbedingungen ... 15
3.2 Maßzeichnungen ... 15
3.2.1 PD6-E891S95-M-65-3 ... 15
3.2.2 PD6-E891M95-M-65-3 ... 16
3.2.3 PD6-E891L95-M-65-3 ... 16
3.2.4 PD6-EB80SD-M-65-3 ... 16
3.2.5 PD6-EB80MD-M-65-3 ... 17
3.2.6 PD6-EB80LD-M-65-3 ... 17
3.2.7 PD6-EB80CD-M-65-3 ... 17
3.3 Elektrische Eigenschaften und technische Daten ... 18
3.3.1 Elektrische Daten ... 18
3.3.2 Technische Daten ... 18
3.4 Übertemperaturschutz ... 19
3.5 LED-Signalisierung ... 23
3.5.1 Betriebs-LED ... 23
3.5.2 EtherNet/IP™ LEDs ... 24
3.6 Anschlussbelegung ... 25
3.6.1 Übersicht ... 25
3.6.2 X1 — Spannungsversorgung ... 25
3.6.3 X2 — Ein-/Ausgänge und Logikversorgung ... 27
3.6.4 X3 — EtherNet/IP IN ... 29
3.6.5 X4 — EtherNet/IP OUT ... 30
3.6.6 X5 — externe Haltebremse ... 30
3.6.7 X6 − Mini USB ... 31
3.6.8 X7 - STO (*Safe Torque Off*) ... 31

4 Inbetriebnahme ... 33
4.1 Konfiguration über USB ... 34
4.1.1 Allgemeines ... 34
4.1.2 USB-Anschluss ... 35
4.1.3 Konfigurationsdatei ... 36
4.1.4 NanoJ-Programm ... 38

4.2 Inbetriebnahme EtherNet/IP™ ... 39
4.2.1 Anschluss ... 39
4.2.2 Software Verbindung ... 39
4.3 Konfigurieren der Sensoren ... 42
4.4 Auto-Setup ... 44
4.4.1 Parameter-Ermittlung ... 44
4.4.2 Durchführung ... 45
4.4.3 Parameterspeicherung ... 47
4.5 Spezielle Fahrmodi ... 47
4.5.1 Aktivierung ... 48
4.5.2 Analog-Drehzahl ... 48
4.5.3 Automatische Fahrt mit 30 U/min (Testfahrt) ... 50

5 Generelle Konzepte ... 51
5.1 Betriebsarten ... 51
5.1.1 Allgemein ... 51
5.1.2 Open Loop ... 52
5.1.3 Closed Loop ... 54
5.1.4 Slow Speed ... 60
5.2 CiA 402 Power State Machine ... 61
5.2.1 Zustandsmaschine ... 61
5.2.2 Verhalten beim Verlassen des Zustands *Operation enabled* ... 63
5.3 Benutzerdefinierte Einheiten ... 66
5.3.1 Einheiten ... 67
5.3.2 Encoderauflösung ... 68
5.3.3 Getriebeübersetzung ... 69
5.3.4 Vorschubkonstante ... 69
5.3.5 Berechnungsformeln für Benutzereinheiten ... 69
5.4 Begrenzung des Bewegungsbereichs ... 71
5.4.1 Verhalten beim Erreichen der Endschalter ... 71
5.4.2 Software-Endschalter ... 72
5.5 Zykluszeiten ... 72

6 Betriebsmodi ... 73
6.1 Profile Position ... 73
6.1.1 Übersicht ... 73
6.1.2 Setzen von Fahrbefehlen ... 74
6.1.3 Genauigkeitsverlust bei Relativbewegungen ... 78
6.1.4 Randbedingungen für eine Positionierfahrt ... 79
6.1.5 Ruck-begrenzter und nicht ruck-begrenzter Modus ... 80
6.2 Velocity ... 81
6.2.1 Beschreibung ... 81
6.2.2 Aktivierung ... 81
6.2.3 Controlword ... 81
6.2.4 Statusword ... 81
6.2.5 Objekteinträge ... 81
6.3 Profile Velocity ... 82
6.3.1 Beschreibung ... 82
6.3.2 Aktivierung ... 82
6.3.3 Controlword ... 83
6.3.4 Statusword ... 83
6.3.5 Objekteinträge ... 83
6.4 Profile Torque ... 85
6.4.1 Beschreibung ... 85
6.4.2 Aktivierung ... 86
6.4.3 Controlword ... 86
6.4.4 Statusword ... 86

6.4.5 Objekteinträge....86
6.5 Homing....87
6.5.1 Übersicht....87
6.5.2 Referenzfahrt-Methode....89
6.6 Auto-Setup....95
6.6.1 Beschreibung....95
6.6.2 Aktivierung....95
6.6.3 Controlword....95
6.6.4 Statusword....95

7 Spezielle Funktionen....96
7.1 Digitale Ein- und Ausgänge....96
7.1.1 Bitzuordnung....96
7.1.2 Digitale Eingänge....96
7.1.3 Digitale Ausgänge....101
7.2 Analoge Eingänge....105
7.2.1 Objekteinträge....105
7.2.2 Analogwert skalieren....105
7.3 Automatische Bremsensteuerung....106
7.3.1 Beschreibung....106
7.3.2 Aktivierung und Anschluss....106
7.3.3 Steuerung der Bremse....106
7.3.4 Bremsen-PWM....107
7.4 I^2t Motor-Überlastungsschutz....108
7.4.1 Beschreibung....108
7.4.2 Objekteinträge....108
7.4.3 Aktivierung....109
7.4.4 Funktion von I^2t....109
7.5 Objekte speichern....109
7.5.1 Allgemeines....110
7.5.2 Kategorie: Kommunikation....110
7.5.3 Kategorie: Applikation....110
7.5.4 Kategorie: Benutzer....112
7.5.5 Kategorie: Bewegung....112
7.5.6 Kategorie: Tuning....112
7.5.7 Kategorie: Ethernet....113
7.5.8 Speichervorgang starten....113
7.5.9 Speicherung verwerfen....114
7.5.10 Konfiguration verifizieren....114

8 EtherNet/IP™....115
8.1 Geräteprofil....115
8.2 Service: Get object dictionary entry....116
8.3 Service: Set object dictionary entry....116
8.4 Service: Get object dictionary entry Rockwell....116
8.5 Assembly-Objekte....117
8.6 Konfiguration der Assembly-Objekte....119
8.7 Rockwell Studio 5000....120
8.8 Panasonic-SPS....120

9 Programmierung mit NanoJ....121
9.1 NanoJ-Programm....121
9.2 Mapping im NanoJ-Programm....125
9.3 NanoJ-Funktionen im NanoJ-Programm....126
9.4 Einschränkungen und mögliche Probleme....128

10 Objektverzeichnis Beschreibung.....130

10.1 Übersicht.....130
10.2 Aufbau der Objektbeschreibung.....130
10.3 Objektbeschreibung.....130
10.4 Wertebeschreibung.....131
10.5 Beschreibung.....132
1000h Device Type.....133
1001h Error Register.....134
1003h Pre-defined Error Field.....135
1008h Manufacturer Device Name.....140
1009h Manufacturer Hardware Version.....140
100Ah Manufacturer Software Version.....141
1010h Store Parameters.....141
1011h Restore Default Parameters.....145
1018h Identity Object.....148
1020h Verify Configuration.....150
1F50h Program Data.....151
1F51h Program Control.....152
1F57h Program Status.....153
200Fh IEEE 802 MAC Address.....154
2010h IP-Configuration.....155
2011h Static-IPv4-Address.....156
2012h Static-IPv4-Subnet-Mask.....157
2013h Static-IPv4-Gateway-Address.....158
2014h Current-IPv4-Address.....159
2015h Current-IPv4-Subnet-Mask.....160
2016h Current-IPv4-Gateway-Address.....161
2017h LLDP Non-volatile Storage.....161
202Fh MODBUS Address Control.....163
2030h Pole Pair Count.....164
2031h Max Motor Current.....165
2034h Upper Voltage Warning Level.....166
2035h Lower Voltage Warning Level.....167
2036h Open Loop Current Reduction Idle Time.....167
2037h Open Loop Current Reduction Value/factor.....168
2038h Brake Controller Timing.....168
2039h Motor Currents.....170
203Ah Homing On Block Configuration.....172
203Bh I2t Parameters.....174
203Dh Torque Window.....177
203Eh Torque Window Time Out.....177
203Fh Max Slippage Time Out.....178
205Ah Absolute Sensor Boot Value (in User Units).....178
2084h Bootup Delay.....179
2101h Fieldbus Module Availability.....179
2102h Fieldbus Module Control.....181
2103h Fieldbus Module Status.....182
2290h PDI Control.....184
2291h PDI Input.....184
2292h PDI Output.....186
2300h NanoJ Control.....187
2301h NanoJ Status.....188
2302h NanoJ Error Code.....189
230Eh Timer.....190
230Fh Uptime Seconds.....191
2310h NanoJ Input Data Selection.....191
2320h NanoJ Output Data Selection.....193

2330h NanoJ In/output Data Selection....194
2400h NanoJ Inputs....195
2410h NanoJ Init Parameters....196
2500h NanoJ Outputs....197
2600h NanoJ Debug Output....198
2701h Customer Storage Area....199
2800h Bootloader And Reboot Settings....200
3202h Motor Drive Submode Select....201
3203h Feedback Selection....203
3204h Feedback Mapping....204
320Dh Torque Of Inertia Factor....206
3210h Motor Drive Parameter Set....207
3212h Motor Drive Flags....211
321Ah Current Controller Parameters....213
321Bh Velocity Controller Parameters....215
321Ch Position Controller Parameters....217
321Dh Pre-control....218
321Eh Voltage Limit....219
3220h Analog Input Digits....220
3221h Analog Inputs Control....221
3240h Digital Inputs Control....222
3242h Digital Input Routing....225
3243h Home Switch Position Capture....234
3250h Digital Outputs Control....236
3252h Digital Output Routing....239
3320h Analog Input Values....240
3321h Analog Input Offsets....241
3322h Analog Input Numerators....242
3323h Analog Input Denominators....243
3380h Feedback Sensorless....244
33A0h Feedback Incremental A/B/I 1....246
33B0h Feedback SSI 1....247
3501h EtherNetIP Rx PDO Mapping....251
3502h MODBUS Rx PDO Mapping....258
3601h EtherNetIP Tx PDO Mapping....262
3602h MODBUS Tx PDO Mapping....269
3700h Deviation Error Option Code....273
3701h Limit Switch Error Option Code....273
4012h HW Information....274
4013h HW Configuration....275
4014h Operating Conditions....276
4015h Special Drive Modes....278
4016h Factory Settings....279
4021h Ballast Configuration....280
4040h Drive Serial Number....281
4041h Device Id....282
4042h Bootloader Infos....282
603Fh Error Code....284
6040h Controlword....284
6041h Statusword....285
6042h Vl Target Velocity....287
6043h Vl Velocity Demand....287
6044h Vl Velocity Actual Value....288
6046h Vl Velocity Min Max Amount....288
6048h Vl Velocity Acceleration....289
6049h Vl Velocity Deceleration....290
604Ah Vl Velocity Quick Stop....291
604Ch Vl Dimension Factor....292
605Ah Quick Stop Option Code....293

605Bh Shutdown Option Code.....294
605Ch Disable Option Code.....295
605Dh Halt Option Code.....296
605Eh Fault Option Code.....296
6060h Modes Of Operation.....297
6061h Modes Of Operation Display.....298
6062h Position Demand Value.....298
6063h Position Actual Internal Value.....299
6064h Position Actual Value.....299
6065h Following Error Window.....300
6066h Following Error Time Out.....300
6067h Position Window.....301
6068h Position Window Time.....301
606Bh Velocity Demand Value.....302
606Ch Velocity Actual Value.....302
606Dh Velocity Window.....303
606Eh Velocity Window Time.....303
606Fh Velocity Threshold.....304
6070h Velocity Threshold Time.....304
6071h Target Torque.....305
6072h Max Torque.....306
6073h Max Current.....306
6074h Torque Demand.....307
6075h Motor Rated Current.....307
6077h Torque Actual Value.....308
607Ah Target Position.....308
607Bh Position Range Limit.....309
607Ch Home Offset.....310
607Dh Software Position Limit.....310
607Eh Polarity.....311
607Fh Max Profile Velocity.....312
6080h Max Motor Speed.....313
6081h Profile Velocity.....314
6082h End Velocity.....314
6083h Profile Acceleration.....314
6084h Profile Deceleration.....315
6085h Quick Stop Deceleration.....315
6086h Motion Profile Type.....316
6087h Torque Slope.....316
608Fh Position Encoder Resolution.....317
6090h Velocity Encoder Resolution.....318
6091h Gear Ratio.....319
6092h Feed Constant.....320
6096h Velocity Factor.....321
6097h Acceleration Factor.....323
6098h Homing Method.....324
6099h Homing Speed.....324
609Ah Homing Acceleration.....325
60A2h Jerk Factor.....326
60A4h Profile Jerk.....327
60A8h SI Unit Position.....328
60A9h SI Unit Velocity.....329
60B0h Position Offset.....330
60B1h Velocity Offset.....330
60B2h Torque Offset.....331
60C1h Interpolation Data Record.....331
60C2h Interpolation Time Period.....332
60C4h Interpolation Data Configuration.....333
60C5h Max Acceleration.....335

60C6h Max Deceleration..... 336
60E4h Additional Position Actual Value..... 336
60E5h Additional Velocity Actual Value..... 337
60E6h Additional Position Encoder Resolution - Encoder Increments..... 338
60E8h Additional Gear Ratio - Motor Shaft Revolutions..... 339
60E9h Additional Feed Constant - Feed..... 341
60EBh Additional Position Encoder Resolution - Motor Revolutions..... 342
60EDh Additional Gear Ratio - Driving Shaft Revolutions..... 343
60EEh Additional Feed Constant - Driving Shaft Revolutions..... 344
60F2h Positioning Option Code..... 345
60F4h Following Error Actual Value..... 347
60F8h Max Slippage..... 347
60FAh Control Effort..... 348
60FCh Position Demand Internal Value..... 348
60FDh Digital Inputs..... 349
60FEh Digital Outputs..... 350
60FFh Target Velocity..... 351
6502h Supported Drive Modes..... 351
6503h Drive Catalogue Number..... 352
6505h Http Drive Catalogue Address..... 353

11 Copyrights..... 354
11.1 Einführung..... 354
11.2 AES..... 354
11.3 MD5..... 354
11.4 uIP..... 355
11.5 DHCP..... 355
11.6 CMSIS DSP Software Library..... 355
11.7 FatFs..... 355
11.8 Protothreads..... 356
11.9 lwIP..... 356
11.10 littlefs..... 357

1 Einleitung

Der *PD6-E-M* ist ein bürstenloser Motor mit integrierter Steuerung in Schutzart IP65. Durch den integrierten Absolut-Encoder ist der sofortige Betrieb im *Closed Loop*-Modus ohne Referenzfahrt möglich.

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.

1.1 Versionshinweise

Version Handbuch	Datum	Änderungen	Version Firmware
1.0.0	04/2024	Veröffentlichung	v2339
1.0.1	04/2025	Hinweis zum USB-Anschluss hinzugefügt.	v2339
1.1.0	10/2025	Varianten mit Sicherheitsfunktion *STO* hinzugefügt.	v2538
		Geschirmte Kabel auch für die Spannungsversorgung der STO-Varianten gefordert.	

Varianten	Hardware-Revision
Schrittmotor: PD6-E...-65...	4
BLDC: PD6-E**B**...-65...	5
STO: PD6-E...**S**-65...	0

1.2 Urheberrecht, Kennzeichnung und Kontakt

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

Nanotec Electronic GmbH & Co. KG

Kapellenstraße 6

85622 Feldkirchen

Deutschland

Tel.+49 89 900 686-0

Fax +49 89 900 686-50

www.nanotec.de

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

EtherNet/IP™ und CIP™ sind eingetragene Warenzeichen der Open DeviceNet Vendor Association, Inc (ODVA).

CompactLogix®, Studio 5000®, Logix Designer® und RSLinx Classic® sind eingetragene Warenzeichen der Rockwell Automation® Corporation.

1.3 Bestimmungsgemäße Verwendung

Der Motor mit integrierter Steuerung *PD6-E-M* 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.

Unter keinen Umständen darf dieses Nanotec-Produkt als Sicherheitsbauteil in ein Produkt oder eine Anlage integriert werden. 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.

1.4 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 Anlagen-konstrukteur, 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.

1.5 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 EU-Richtlinien zur Produktsicherheit

Folgende EU-Richtlinien wurden beachtet:

- RoHS-Richtlinie (2011/65/EU, 2015/863/EU)
- Maschinenrichtlinie (2006/42/EG), nur bei den Varianten PD6-E-...-xS-... mit Sicherheitsfunktion STO
- EMV-Richtlinie (2014/30/EU)

1.7 Mitgeltende Vorschriften

Neben diesem technischen Handbuch sind folgende Vorschriften zu beachten:

- Unfallverhütungsvorschriften
- örtliche Vorschriften zur Arbeitssicherheit

1.8 Verwendete Symbole

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

VORSICHT!

Der Hinweis VORSICHT verweist auf eine möglicherweise gefährliche Situation.

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

► Beschreibt, wie Sie die gefährliche Situation vermeiden.

HINWEIS

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

STO

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

STO

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.

1.9 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.

1.10 Zahlenwerte

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

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

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

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

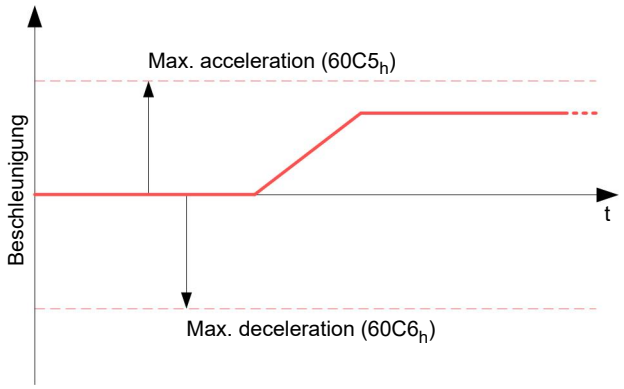
1.11 Bits

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

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

1.12 Zählrichtung (Pfeile)

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



2 Sicherheits- und Warnhinweise

VORSICHT

STO

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

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.

VORSICHT!

Überhitzungs- oder Brandgefahr bei unzureichender Kühlung!

Falls die Kühlung nicht ausreichend ist oder die Umgebungstemperatur zu hoch ist, besteht Überhitzungs- oder Brandgefahr.

► Achten Sie beim Einsatz darauf, dass die Kühlung und die Umgebungsbedingungen gewährleistet sind.

► Achten Sie bei der Einbaulage des Motors *PD6-EB...* darauf, dass die Kühlrippen die Wärmeabführung begünstigen.

VORSICHT

STO

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 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.

HINWEIS

Beschädigung der Elektronik durch verpolten Anschluss der Versorgungsspannung!

Bei Verpolung entsteht ein Kurzschluss zwischen Versorgungsspannung und GND (Masse) über die Leistungsdiode.

► Beachten Sie die Pin-Belegung.

► Installieren Sie eine Leitungsschutzeinrichtung (Sicherung) in der Zuleitung.

3 Technische Daten und Anschlussbelegung

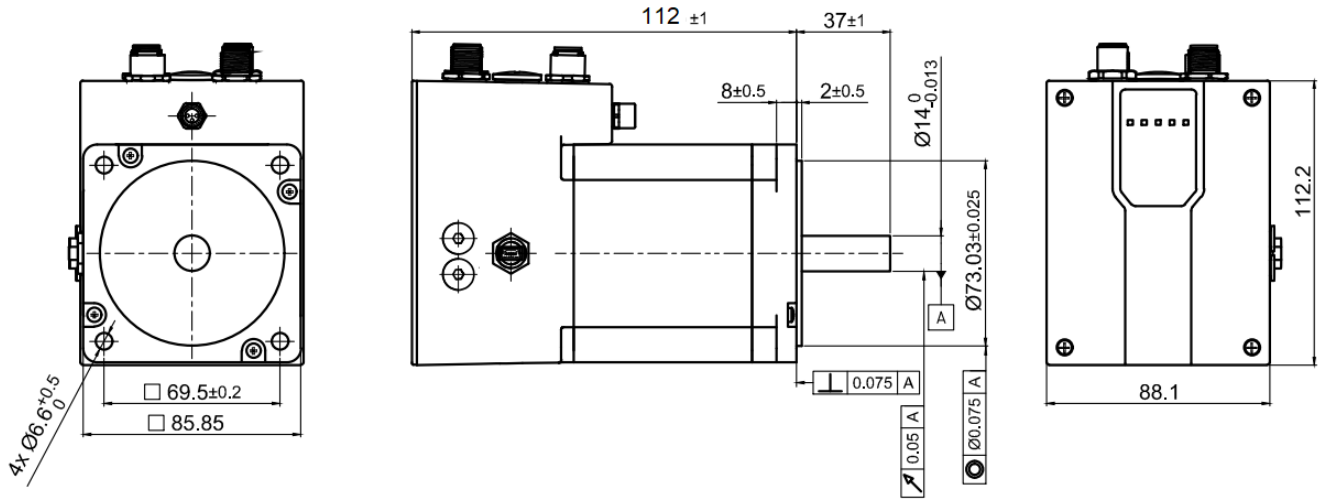
3.1 Umgebungsbedingungen

Umgebungsbedingung	Wert
Schutzklasse	IP65 (IP54 am Wellenausgang)
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 … 85%
Abs.. Luftfeuchtigkeit (Lagerung und Transport), nicht kondensierend	30 g/m^3
Max. Aufstellhöhe über *NN* (ohne Leistungsbeschränkung im Betrieb)	1500 m
Max. Aufstellhöhe über *NN* (Lagerung und Transport)	3000 m

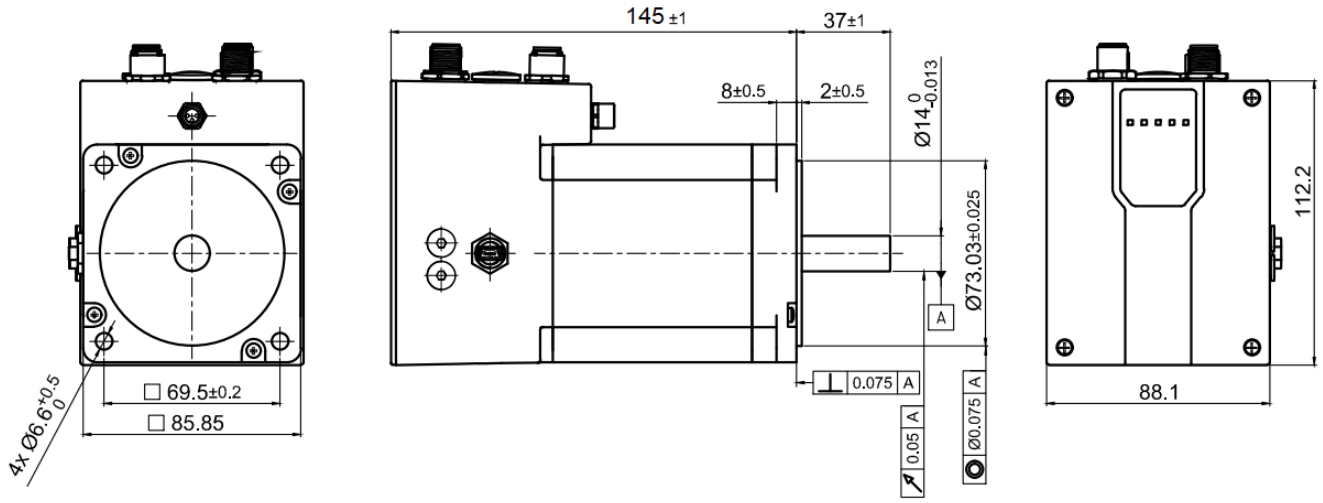
3.2 Maßzeichnungen

Alle Maße sind in Millimetern.

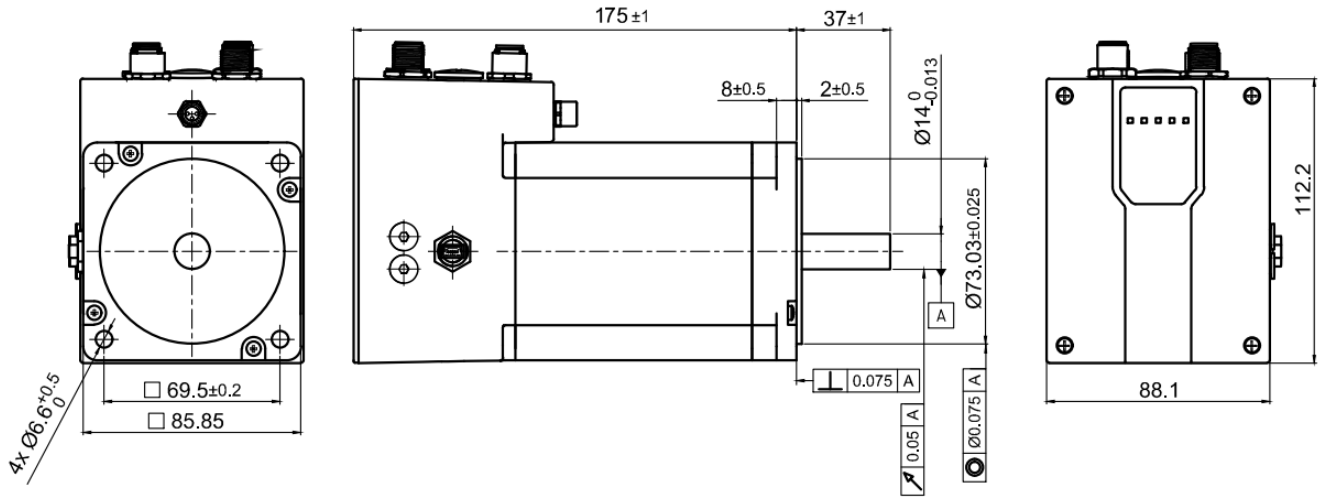
3.2.1 PD6-E891S95-M-65-3



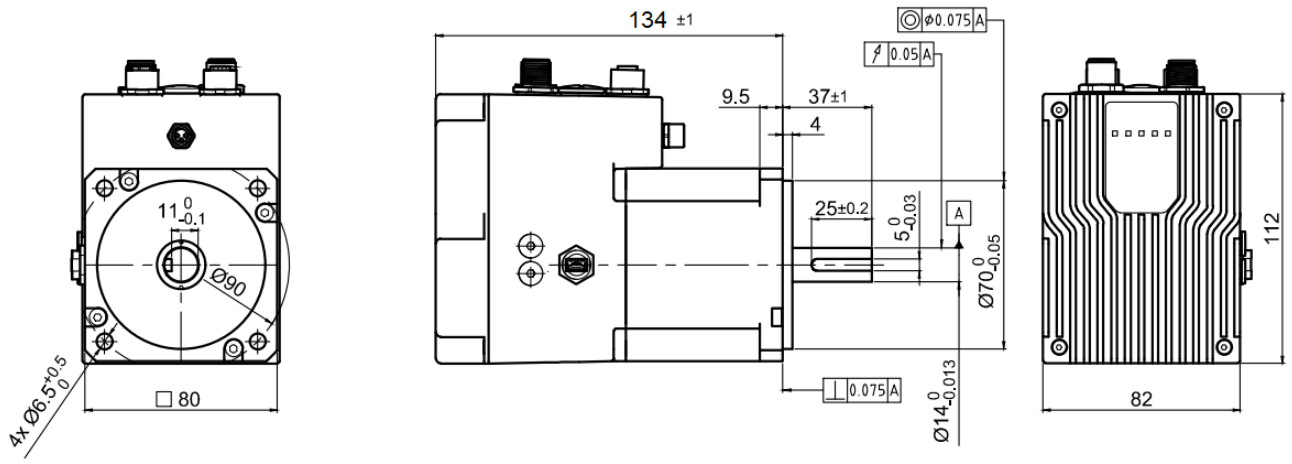
3.2.2 PD6-E891M95-M-65-3



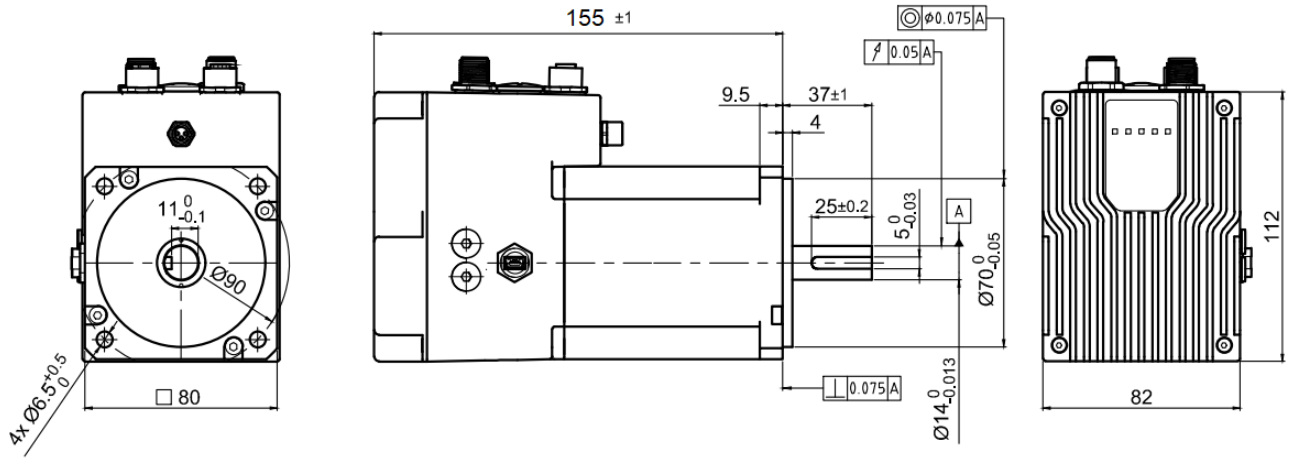
3.2.3 PD6-E891L95-M-65-3



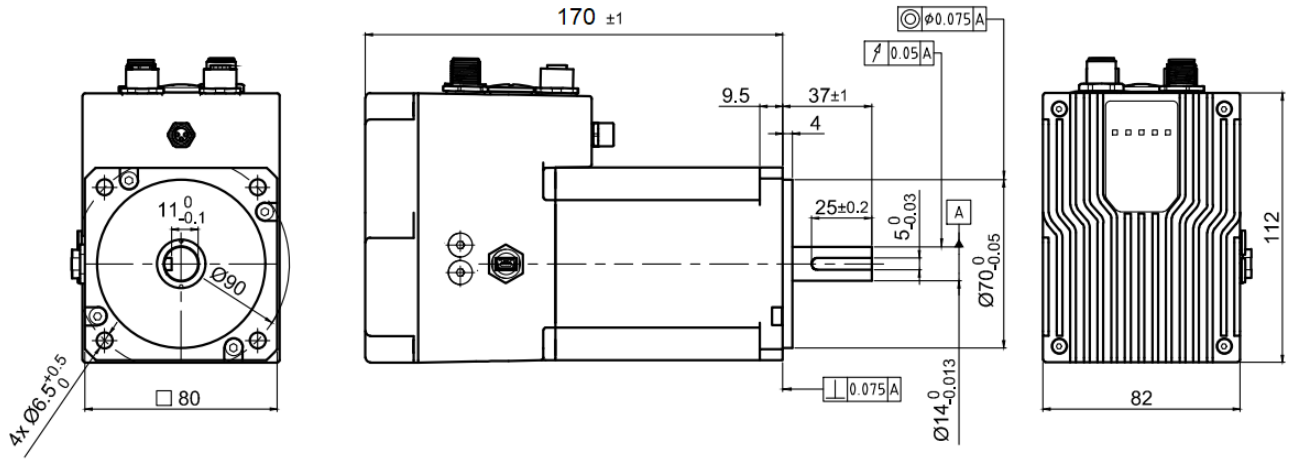
3.2.4 PD6-EB80SD-M-65-3



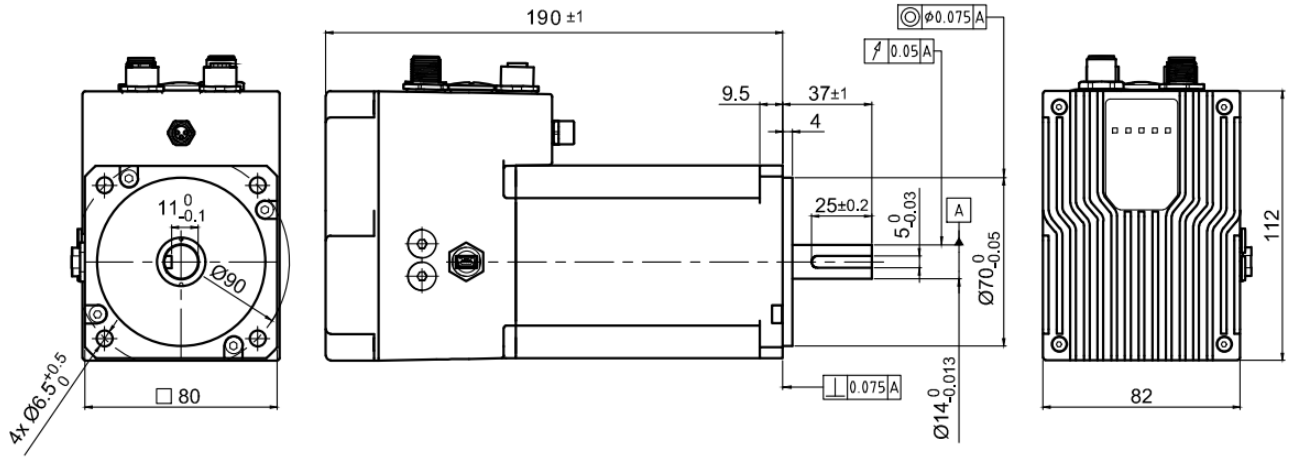
3.2.5 PD6-EB80MD-M-65-3



3.2.6 PD6-EB80LD-M-65-3



3.2.7 PD6-EB80CD-M-65-3



3.3 Elektrische Eigenschaften und technische Daten

3.3.1 Elektrische Daten

Betriebsspannung (Nennwert)	48 V DC
Zulässiger Bereich der Betriebsspannung	12…57,6 V DC
Logikspannung (Nennwert)	24 V DC
Spannungsbereich der Logikversorgung	12…30 V DC
Nennstrom	■ PD6-E891S95-… : 9,5 A eff. ■ PD6-E8918M95-… : 9,5 A eff. ■ PD6-E8918L95-… : 9,5 A eff. ■ PD6-EB80SD -… : 6,9 A eff. ■ PD6-EB80MD -… : 14 A eff. ■ PD6-EB80LD -… : 18,75 A eff. ■ PD6-EB80CD-… : 25 A eff.
Spitzenstrom (für max. 5 s)	■ PD6-E8918S95-… : 12,5 A eff. ■ PD6-E8918M95-… : 12,5 A eff. ■ PD6-E8918L95-… : 12,5 A eff. ■ PD6-EB80SD-… : 20 A eff. ■ PD6-EB80MD-… : 40 A eff. ■ PD6-EB80LD-… : 65 A eff. ■ PD6-EB80CD-… : 75 A eff.

3.3.2 Technische Daten

Betriebsmodi	*Profile Position Mode*, *Profile Velocity Mode*, *Profile Torque Mode*, *Velocity Mode*, *Homing Mode*,
Sollwertvorgabe/ Programmierung	*EtherNet/IP*, *Analogeingang*, *NanoJ-Programm*
Eingänge	6 Digitaleingänge (+5 V/+24 V DC), einzeln per Software umschaltbar, Werkseinstellung: 5 V 1 Analogeingang 0…10 V oder 0…20 mA (per Software umschaltbar)
Ausgänge	2 digitale Ausgänge, *Open Drain*, max. 100 mA 1 Bremsenausgang für externe Haltebremse
Integrierter Encoder	magnetischer Multiturn-Absolut-Encoder, Singleturn-Auflösung: 17 Bit (131072 pro Umdrehung), Multiturn-Auflösung: 16 Bit (+/-2^{15} Umdrehungen)
Schutzschaltung	Über- und Unterspannungsschutz, interne Ballast-Schaltung Übertemperaturschutz (> 75°C auf der Leistungsplatine, entspricht ca. 65 - 72 °C außen am Deckel) Verpolungsschutz: bei Verpolung Kurzschluss zwischen Versorgungsspannung und GND über Leistungsdiode, daher ist eine

Leitungsschutzeinrichtung (Sicherung) in Zuleitung nötig. Die Werte der Sicherung sind abhängig von der Applikation und müssen

- größer als die maximale Stromaufnahme der Steuerung,
- kleiner als der maximale Strom der Spannungsversorgung ausgelegt werden.

Falls der Sicherungswert sehr nahe an der maximalen Stromaufnahme der Steuerung liegt, sollte eine Auslösecharakteristik mittel/träge eingesetzt werden.

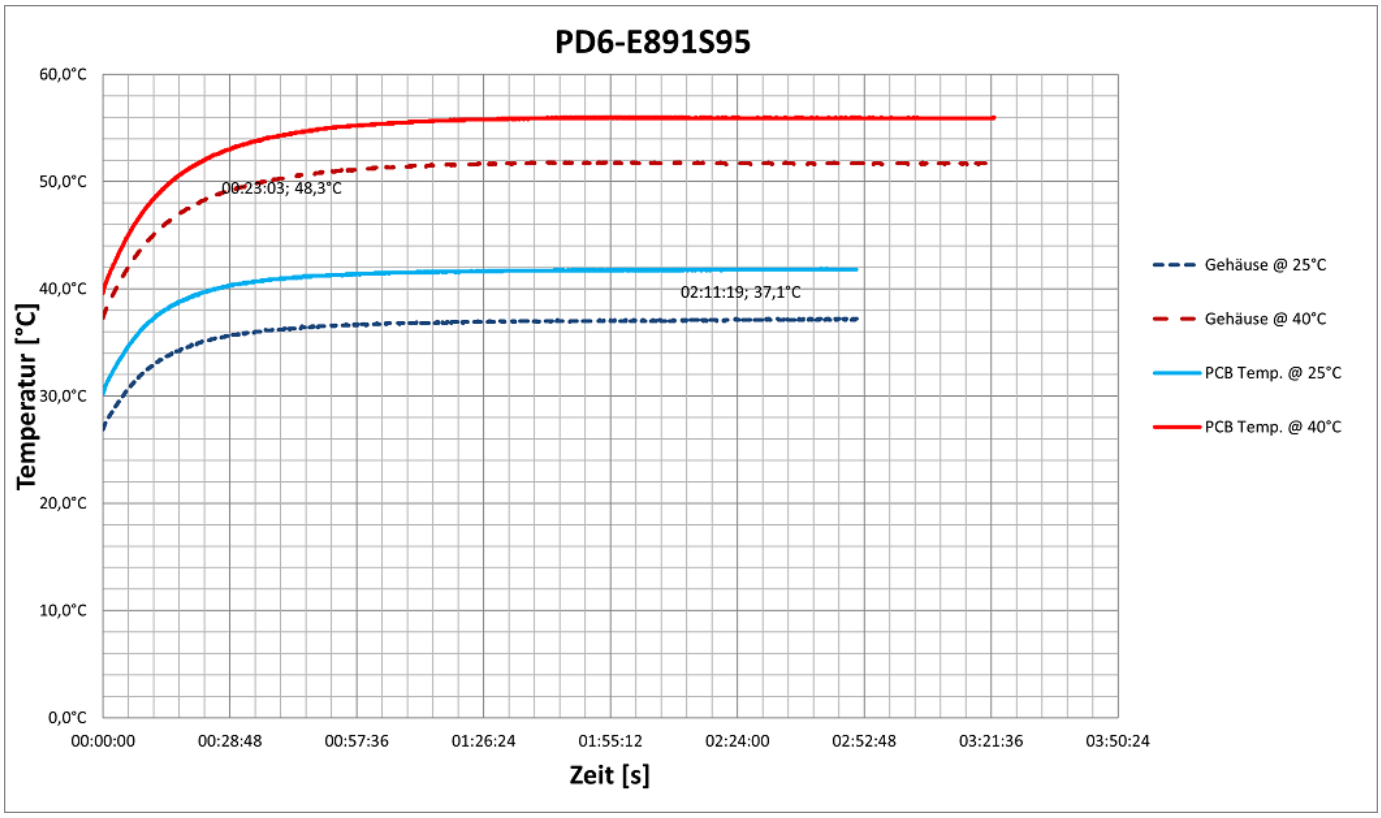
3.4 Übertemperaturschutz

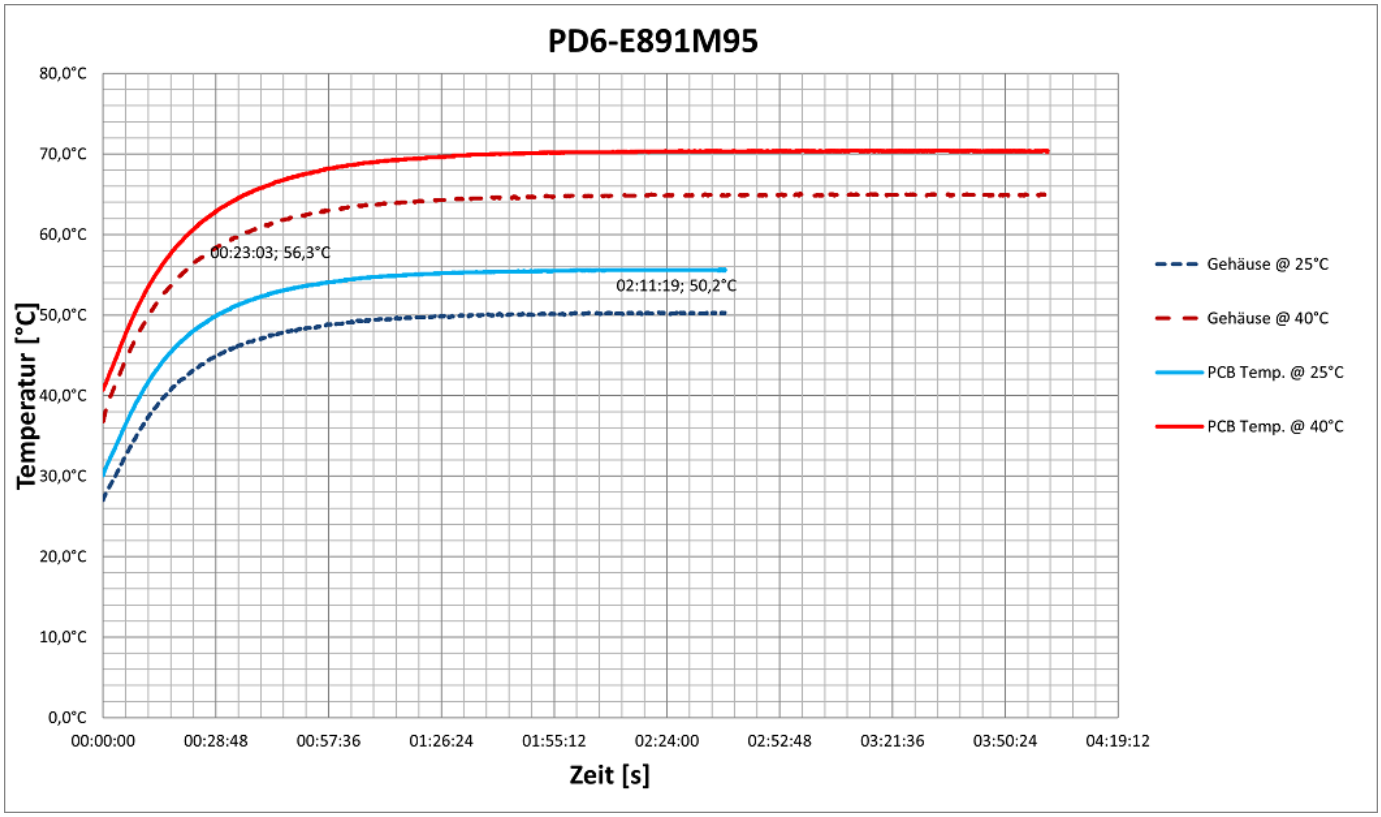
Ab einer Temperatur von ca. 75°C auf der Leistungsplatine (entspricht 65 - 72°C außen am Deckel) wird das Leistungsteil der Steuerung abgeschaltet und das Fehlerbit gesetzt (siehe Objekt 1001_h und 1003_h). Nach Abkühlung und dem Bestätigen des Fehlers (siehe Tabelle für das Contolword, "Fault reset") funktioniert die Steuerung wieder normal.

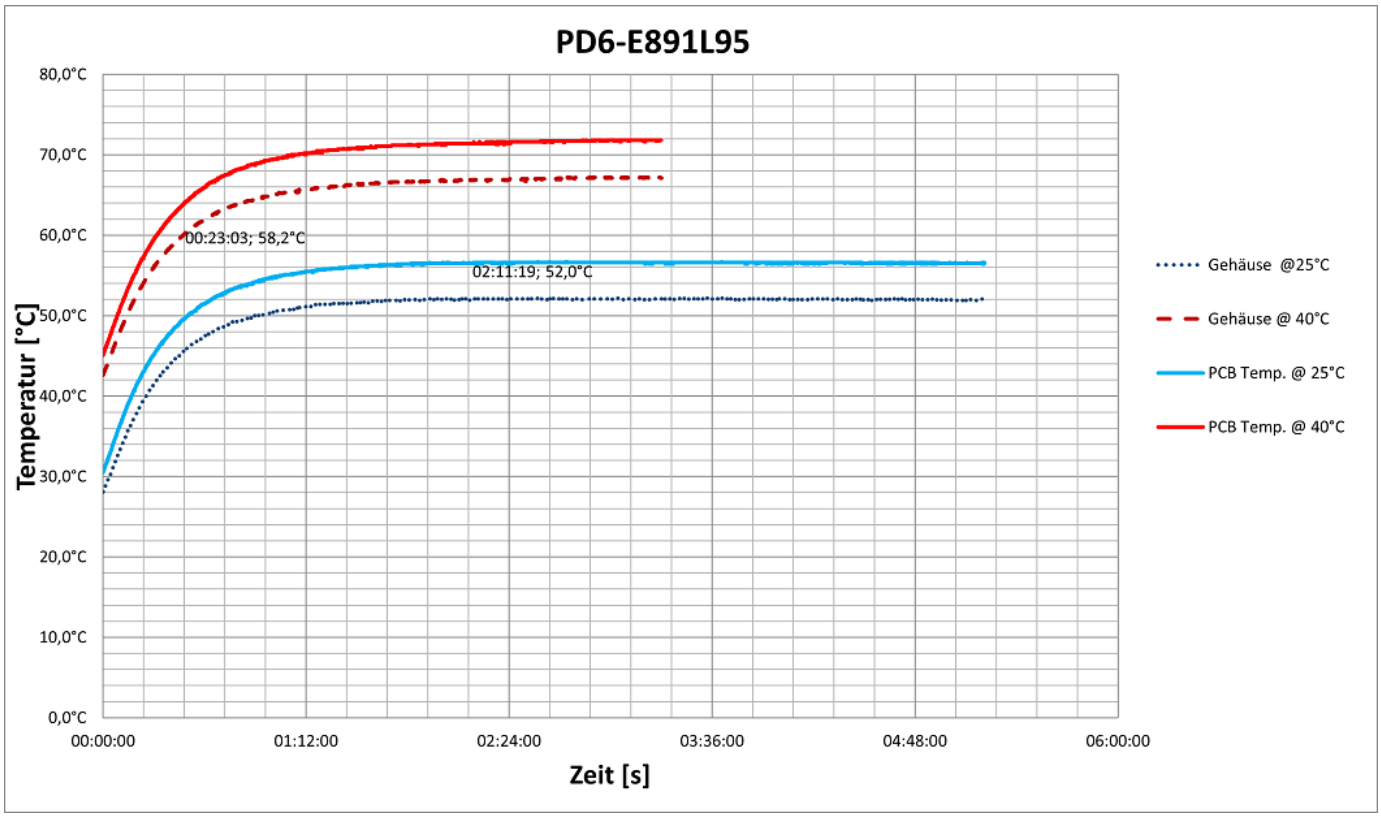
Es wurden Temperaturtests unter folgenden Bedingungen durchgeführt:

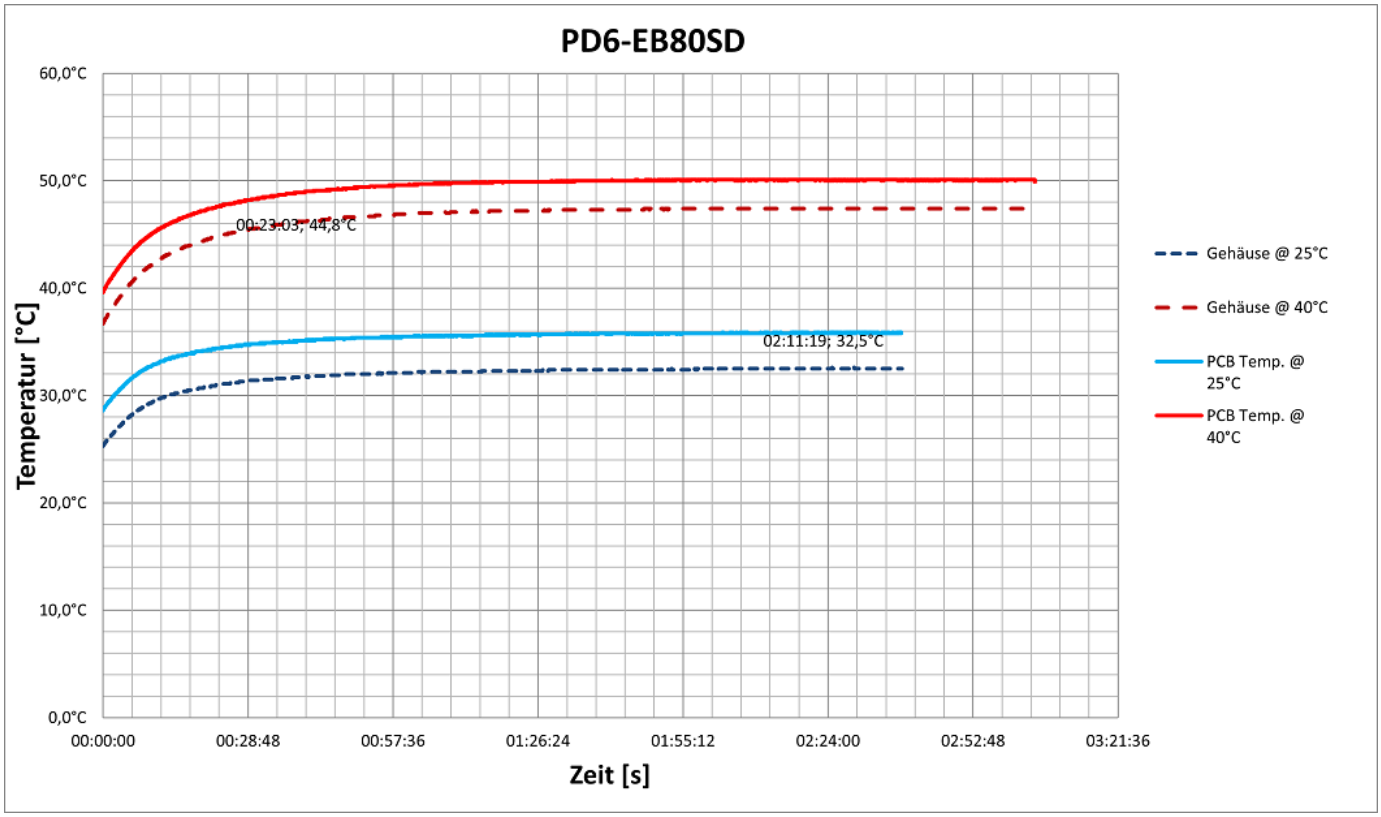
- Betriebsspannung / Logikspannung: 48/24 V DC
- Motorstrom: Nennstrom (siehe Elektrische Eigenschaften und technische Daten)
- Operationsmodus: *Open Loop*, Drehzahlmodus 100 U/min
- Umgebungstemperatur: 25°C / 40°C
- Aufstellhöhe: 500 m über NN
- keine externe Kühlung im Klimaschrank, z. B. über Lüfter
- Motor angeflanscht an Alu-Platte (30 cm x 30 cm x 2 cm)

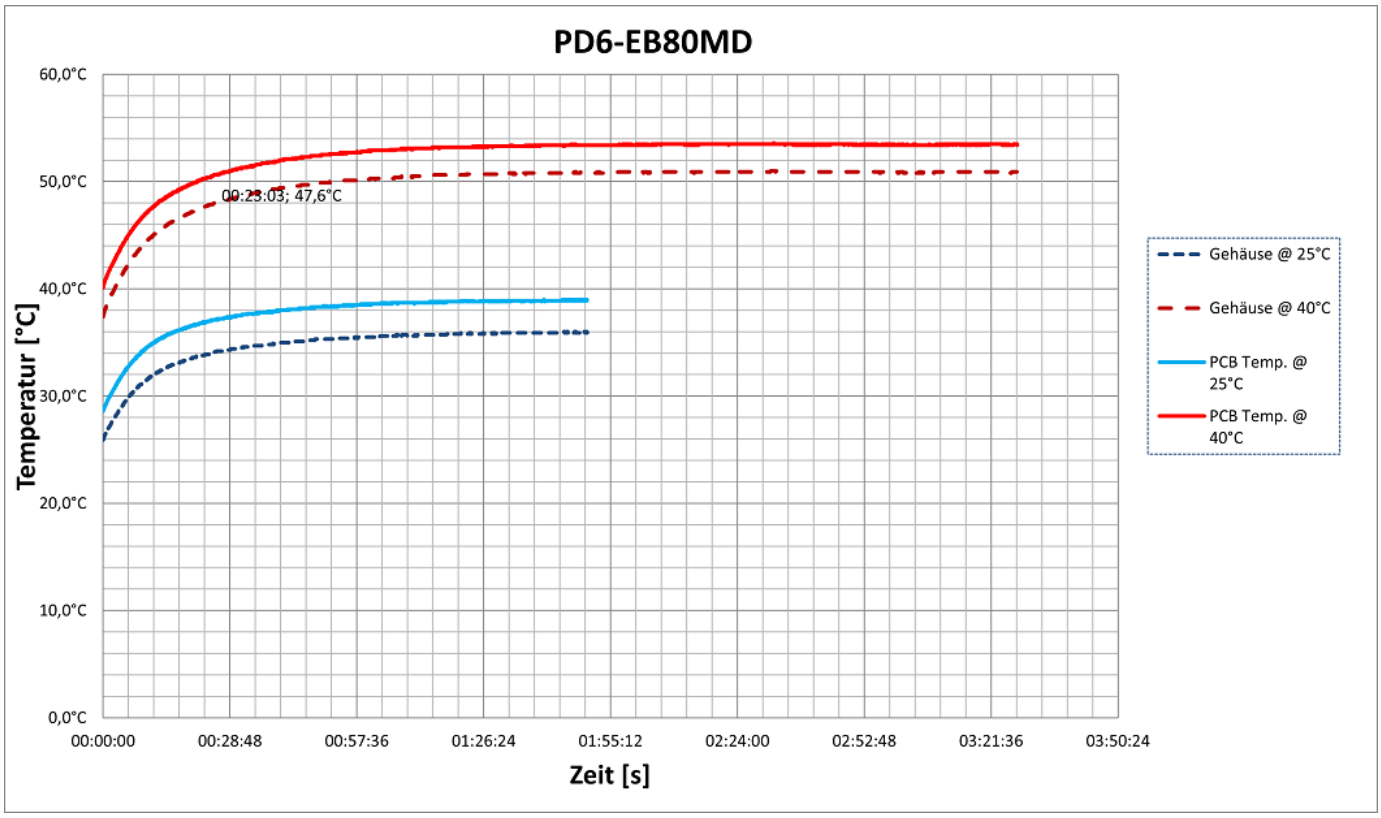
Die folgenden Grafiken zeigen die Ergebnisse der Temperaturtests:

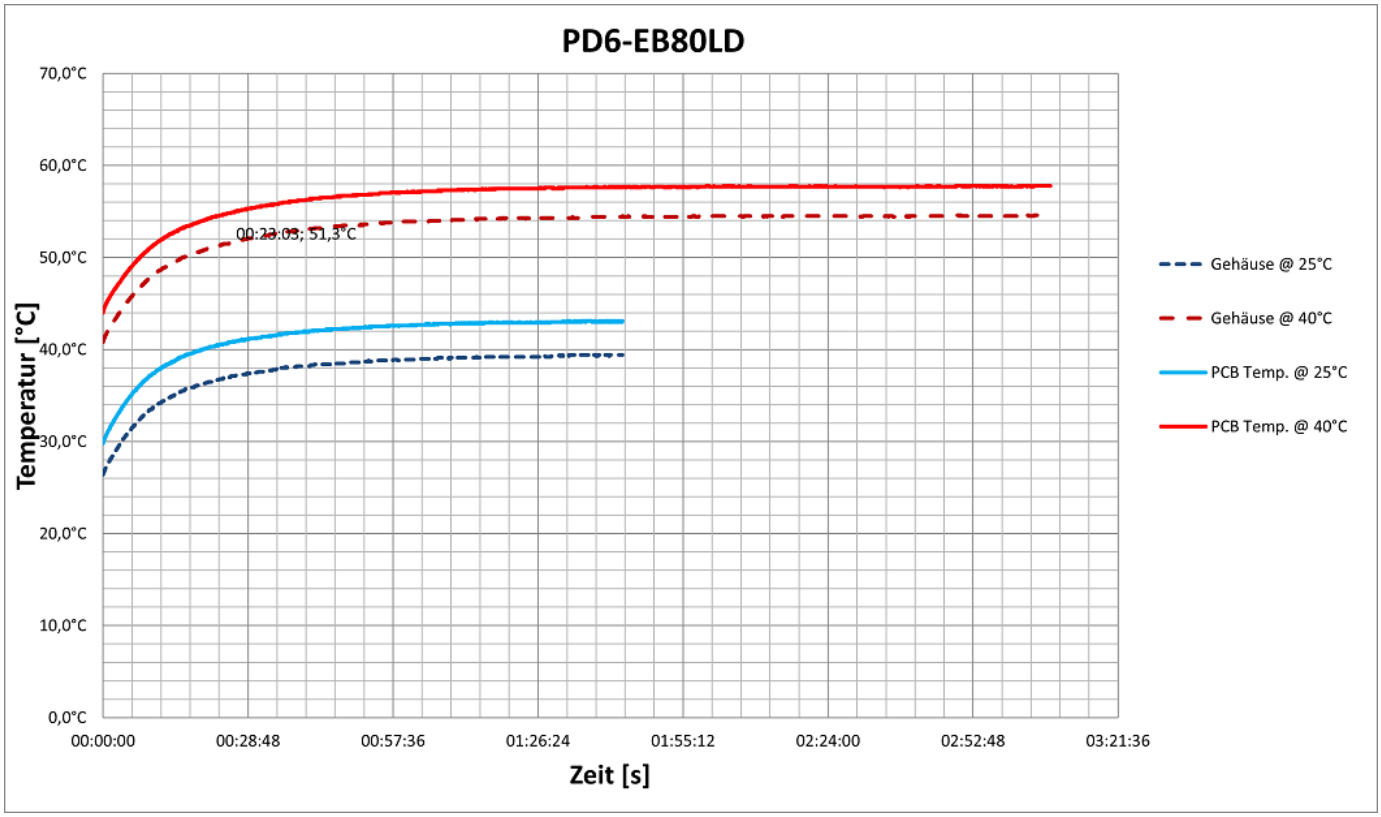


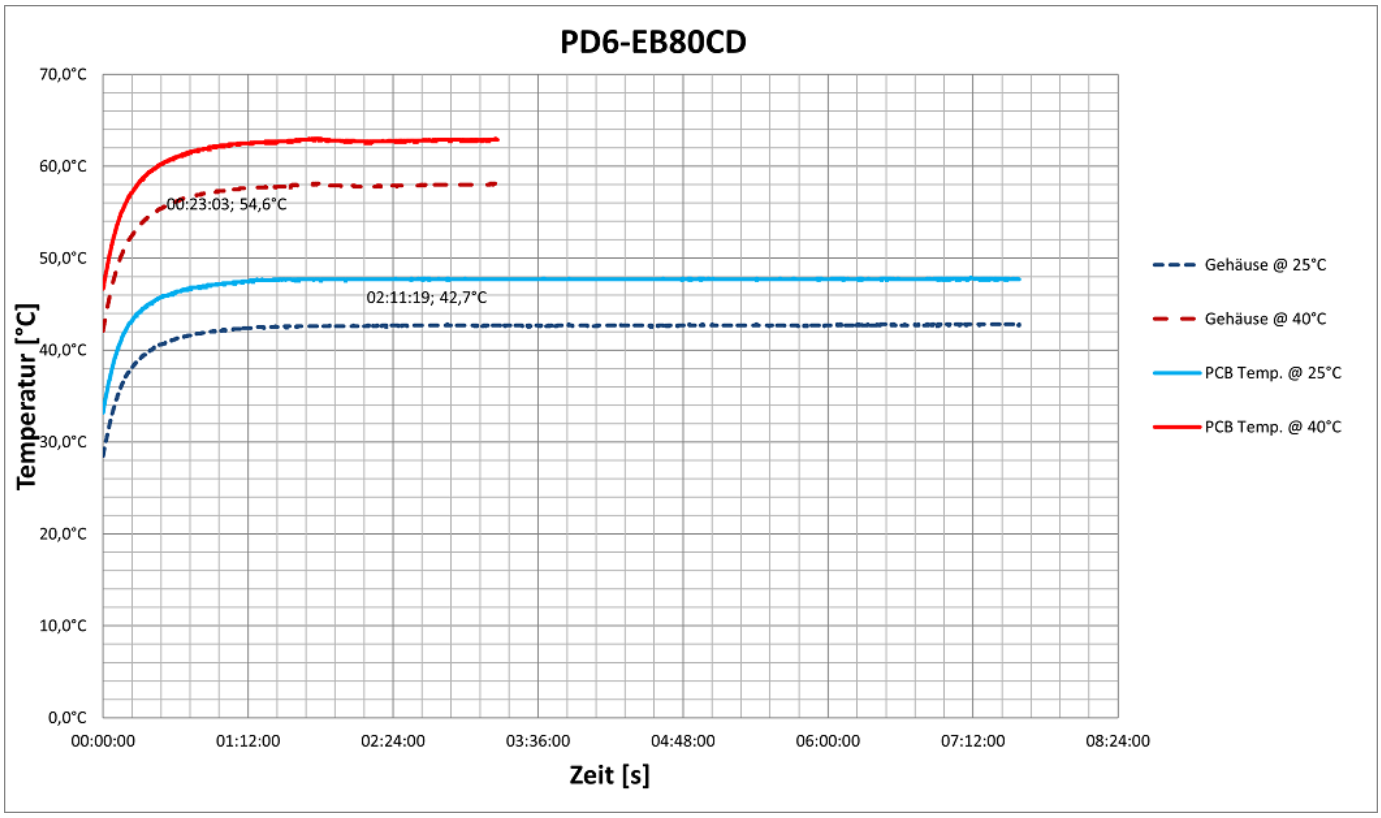

PD6-E891M95
80,0°C
70,0°C
60,0°C
50,0°C
40,0°C
30,0°C
20,0°C
10,0°C
0,0°C
Temperatur [°C]
00:23:03; 56,3°C
02:11:19; 50,2°C
00:00:00
00:28:48
00:57:36
01:26:24
01:55:12
02:24:00
02:52:48
03:21:36
03:50:24
04:19:12
Zeit [s]
Gehäuse @ 25°C
Gehäuse @ 40°C
PCB Temp. @ 25°C
PCB Temp. @ 40°C


PD6-E891L95
80,0°C
70,0°C
60,0°C
50,0°C
40,0°C
30,0°C
20,0°C
10,0°C
0,0°C
Temperatur [°C]
00:23:03; 58,2°C
02:11:19; 52,0°C
00:00:00
01:12:00
02:24:00
03:36:00
04:48:00
06:00:00
Zeit [s]
Gehäuse @25°C
Gehäuse @ 40°C
PCB Temp. @ 25°C
PCB Temp. @ 40°C


PD6-EB80SD
60,0°C
50,0°C
40,0°C
30,0°C
20,0°C
10,0°C
0,0°C
Temperatur [°C]
00:23:03; 44,8°C
02:11:19; 32,5°C
00:00:00
00:28:48
00:57:36
01:26:24
01:55:12
02:24:00
02:52:48
03:21:36
Zeit [s]
Gehäuse @ 25°C
Gehäuse @ 40°C
PCB Temp. @ 25°C
PCB Temp. @ 40°C


PD6-EB80MD
60,0°C
50,0°C
40,0°C
30,0°C
20,0°C
10,0°C
0,0°C
Temperatur [°C]
00:23:03; 47,6°C
00:00:00
00:28:48
00:57:36
01:26:24
01:55:12
02:24:00
02:52:48
03:21:36
03:50:24
Zeit [s]
Gehäuse @ 25°C
Gehäuse @ 40°C
PCB Temp. @ 25°C
PCB Temp. @ 40°C





Sowohl bei 25 °C als auch bei 40 °C ist der Motor länger als 3 Stunden in Betrieb gewesen ohne Abschaltung. Die Temperatur hat nach ca. 30 Minuten ihr Maximum erreicht und ist danach stabil geblieben.

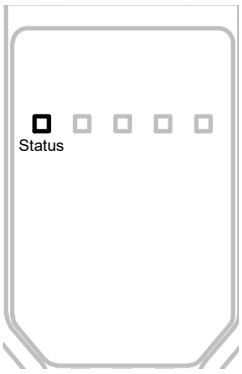
HINWEIS

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

3.5 LED-Signalisierung

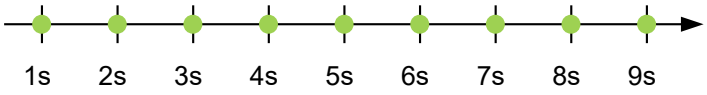
3.5.1 Betriebs-LED

Die Betriebs-LED zeigt den aktuellen Status an.



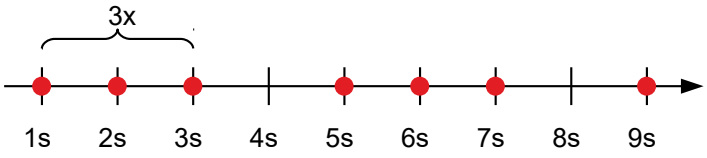
3.5.1.1 Normaler Betrieb

Im normalen Betrieb blinkt die grüne Betriebs-LED 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

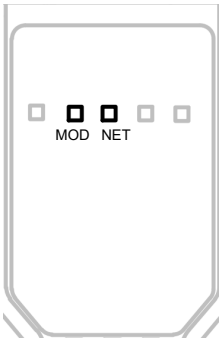
HINWEIS

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

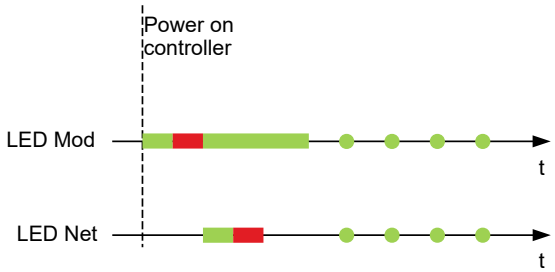
TIPP

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

3.5.2 EtherNet/IP™ LEDs



Die LEDs "Mod" und "Net" blitzen in dem unten dargestellten Muster nach dem Anschalten der Steuerung.



Nach der Einschaltsequenz blitzen die LEDs "Mod" und "Net" kontinuierlich, solange keine EtherNet/IP™ Verbindung aufgebaut wurde. Alle anderen Blink-Kombinationen können in den nachfolgenden Kapiteln nachgelesen werden.

3.5.2.1 Mod LED

Die MOD LED zeigt einen der folgenden Status der Steuerung an:

LED Verhalten	Zusammenfassung	Voraussetzung
Dauerhaft aus	Keine Spannungsversorgung	Falls die Steuerung keine Spannungsversorgung hat, ist die Mod LED dauerhaft aus.
Dauerhaft grün	Gerät betriebsbereit	Falls die Steuerung korrekt operiert ist die LED dauerhaft grün.
Blinkend grün	Standby	Falls die Steuerung nicht konfiguriert ist, blinkt die Mod LED grün.
Blinkend rot	Fehler (minor fault)	Falls ein Fehler auftritt, blinkt die Mod LED rot.
Dauerhaft rot	Watchdog (major fault)	Falls ein Watchdog-Fehler auftritt, leuchtet die Mod LED dauerhaft rot.

3.5.2.2 Net LED

Die Net LED zeigt einen der folgenden Status der Steuerung an:

LED Verhalten	Zusammenfassung	Voraussetzung
Dauerhaft aus	Fehlende Spannungs-versorgung, keine IP Adresse	Die Steuerung ist abgeschalten oder hat Spannungsversorgung aber keine IP Adresse ist konfiguriert worden (Interface Konfigurations Attribut des TCP/IP Interface Objekts).
Blinkend grün	Keine Verbindung	Eine IP Adresse ist konfiguriert aber keine CIP™ Verbindung wurde aufgebaut und eine exklusive Benutzerverbindung lief in einen Timeout.
Dauerhaft grün	Verbunden	Zumindest eine CIP™ Verbindung (beliebige Transportklasse) ist aufgebaut und eine exklusive Benutzerverbindung lief nicht in einen Timeout.
Blinkend rot	Verbindungstimeout	Eine exklusive Benutzerverbindung, dessen Ziel die Steuerung ist, ist in einen Timeout gelaufen. Die "Net Led" kehrt zu dauerhaften grün nur zurück, wenn alle exklusiven Benutzerverbindungen wieder aufgebaut sind.

3.6 Anschlussbelegung

3.6.1 Übersicht

Stecker / Bedienelement	Funktion
X1	Spannungsversorgung
X2	Ein-/Ausgänge und Logikversorgung
X3	EtherNet/IP IN
X4	EtherNet/IP OUT
X5	Anschluss für externe Haltebremse
X6	Mini-USB

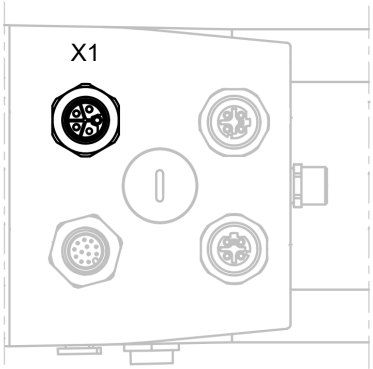
HINWEIS

Alle Pins mit der Bezeichnung *GND* sind intern verbunden.

3.6.2 X1 — Spannungsversorgung

Anschluss für die Hauptversorgung (Leistung). Stecker-Typ: M12, 4-polig, L-kodiert, male

Passendes Nanotec-Kabel: ZK-M12-4-...-...-LFF (nicht im Lieferumfang enthalten)



HINWEIS

Die integrierte Steuerung wird von der Logikversorgung (am X2) versorgt, deren Anschluss für die Kommunikation und den Betrieb notwendig ist.

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.

► Ein EMI-Filter ist in die DC-Zuleitung mit möglichst geringem Abstand zur Steuerung/Motor einzufügen.

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

3.6.2.2 Pin-Belegung

Pin	Funktion	Bemerkung
1	+UB	12…57,6 V DC rot (bei Verwendung des Kabels ZK-M12-4-...-...-LFF)

Pin	Funktion	Bemerkung
2	+UB	12…57,6 V DC rot/weiß (bei Verwendung des Kabels ZK-M12-4-...-...-LFF)
3	GND	schwarz (bei Verwendung des Kabels ZK-M12-4-...-...-LFF)
4	GND	schwarz/weiß (bei Verwendung des Kabels ZK-M12-4-...-...-LFF)

HINWEIS

Beschädigung der Elektronik durch verpolten Anschluss der Versorgungsspannung!

Bei Verpolung entsteht ein Kurzschluss zwischen Versorgungsspannung und GND (Masse) über die Leistungsdiode.

► Beachten Sie die Pin-Belegung.

► Installieren Sie eine Leitungsschutzeinrichtung (Sicherung) in der Zuleitung.

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 die integrierte Ballast-Schaltung aktiviert (Drahtwiderstand CRS500JT-73-15RAA von VITROHM mit 5 W Dauerleistung).

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.

HINWEIS

Beschädigung der Steuerung und/oder Ihres Netzteils durch Erregerspannung des Motors!

Während des Betriebs können Spannungsspitzen die Steuerung und möglicherweise Ihr Netzteil beschädigen.

► Verbauen Sie geeignete Schaltungen (z. B. Ladekondensator), die Spannungsspitzen abbauen.

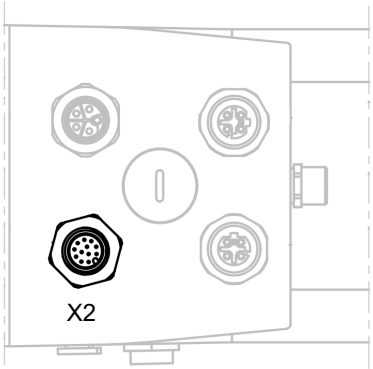
► Bei BLDC-Motoren (PD6-EB…): Wählen Sie eine Spannungsquelle, die der Nennspannung des jeweiligen Motors entspricht, wie im Motordatenblatt angegeben.

► Verwenden Sie ein Netzteil mit Schutzschaltung gegen Überspannung.

3.6.3 X2 — Ein-/Ausgänge und Logikversorgung

Anschluss für die digitalen und analogen Ein-/Ausgänge und die Logikversorgung. Stecker-Typ: M12, 12-polig, A-kodiert, male

Passendes Nanotec-Kabel: ZK-M12-12-2M-1-AFF (nicht im Lieferumfang enthalten)



Pin	Funktion	Bemerkung
1	GND	
2	Digitaler Eingang 1	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240_h, max. 1 MHz
3	Digitaler Eingang 2	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240_h, max. 1 MHz
4	Digitaler Eingang 3	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240_h, max. 1 MHz
5	Digitaler Eingang 4	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240_h, max. 1 MHz
6	Digitaler Eingang 5	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240_h, max. 1 MHz
7	Digitaler Eingang 6	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240_h, max. 1 MHz
8	Analoger Eingang	10 Bit, 0 V…+10 V oder 0…20 mA, umschaltbar per Software mit Objekt 3221_h
9	Digitaler Ausgang 1	Digitalausgang, Open-Drain, max. 30 V / 100 mA
10	Digitaler Ausgang 2	Digitalausgang, Open-Drain, max. 30 V / 100 mA
11	Spannungsausgang	+5 V±5%, max. 250 mA, kurzschlussfest
12	+UB Logic	+24 V DC (12…30 V), Eingangsspannung für die Logikversorgung

HINWEIS

Die Logikversorgung versorgt die Elektronik, den Encoder und die Kommunikationsschnittstelle.

Die Wicklungen des Motors werden nicht von der Logikversorgung versorgt.

Für Eingang 1 bis 6 gelten folgende Schaltschwellen:

Max. Spannung	Schaltschwellen	
	Einschalten	Ausschalten
5 V	> 4,09 V	< 0,95 V
24 V	> 14,74 V	< 3,78 V

Externen Encoder anschließen

Sie können einen externen Encoder an die Pins der digitalen Eingänge anschließen, den Sie einem Regelkreis zuordnen können. Siehe Kapitel Konfigurieren der Sensoren und Zuordnung der Rückführungen zu den Regelkreisen.

Unterstützt werden inkrementale Encoder mit oder ohne Index, single-ended oder differenziell:

Pin	Differenziell	Single-ended
1	GND	GND
2	A\	A
3	A	B
4	B\	I
5	B	
6	I\	
7	I	

HINWEIS

Sie können 5-V-Encoder an den Spannungsausgang am Pin 11 anschließen, sofern der benötigte Strom 250 mA nicht überschreitet.

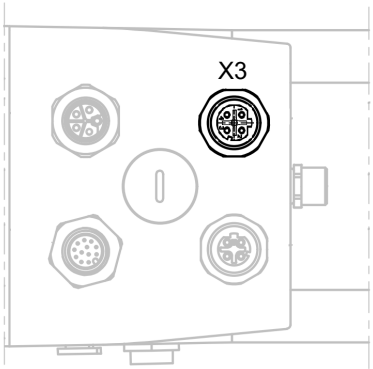
Für 24-V-Encoder müssen Sie:

- eine externe Versorgungsspannung zur Verfügung stellen
- und die entsprechenden Bits in 3240_h:06_h auf den Wert "1" setzen, um den Pegel auf 24 V umzuschalten.

3.6.4 X3 — EtherNet/IP IN

Anschluss für EtherNet/IP. Stecker-Typ: M12, 4-polig, D-kodiert, female

Passendes Nanotec-Kabel: ZK-M12-4-2M-1-D-RJ45 (nicht im Lieferumfang enthalten)

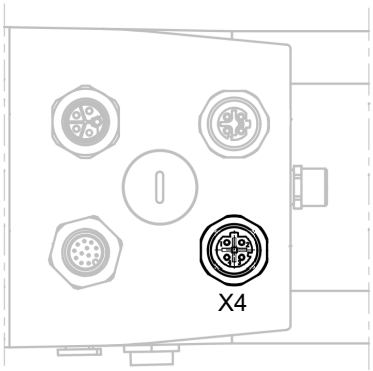


Pin	Funktion	Bemerkung
1	TD+	Transmission Data +
2	RD+	Receiver Data +
3	TD-	Transmission Data -
4	RD-	Receiver Data -

3.6.5 X4 — EtherNet/IP OUT

Anschluss für EtherNet/IP. Typ: M12, 4-polig, D-kodiert, female

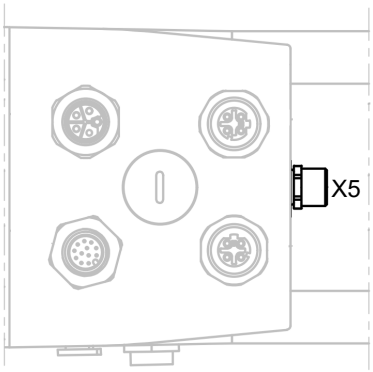
Passendes Nanotec-Kabel: ZK-M12-4-2M-1-D-RJ45 (nicht im Lieferumfang enthalten)



Pin	Funktion	Bemerkung
1	TD+	Transmission Data +
2	RD+	Receiver Data +
3	TD-	Transmission Data -
4	RD-	Receiver Data -

3.6.6 X5 — externe Haltebremse

Anschluss für eine externe optionale Haltebremse. Stecker-Typ: M8, 3-polig, A-kodiert, female

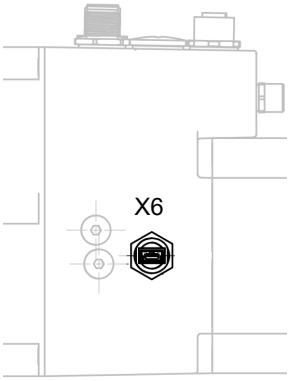


Pin	Funktion	Bemerkung
1	Brake	intern verbunden mit +UB
2	Brake GND	PWM-gesteuerter Open-Drain Ausgang, max 1,5 A
4	n.c.	nicht verbunden

Weitere Details zur Ansteuerung der Haltebremse finden Sie im Kapitel Automatische Bremsensteuerung.

3.6.7 X6 – Mini USB

Mini-USB-Anschluss.

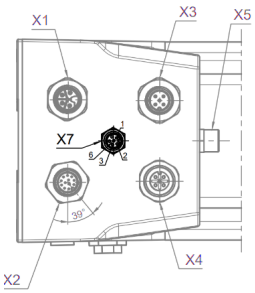


3.6.8 X7 - STO (*Safe Torque Off*)

Anschluss für die STO-Signale. Dieser Anschluss und die entsprechende integrierte Sicherheitsfunktion *STO* stehen nur bei den STO-Varianten des Produkts zur Verfügung (PD6-E-...-xS-...).

Eine ausführliche Beschreibung der STO-Funktion finden Sie im Dokument *Funktionsbeschreibung STO* auf http://www.nanotec.de.

Stecker-Typ: M8, 6-polig, A-kodiert, female



VORSICHT

STO

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

Pin	Funktion	Bemerkung
1	STO-A-IN	STO-Kanal A, PWM-Signalunterdrückung
2	STO-A-RTN	GND-Anschluss für STO-Kanal A
3	STO-FBH	STO-Feedback, High: Rückmeldeausgang für den STO-Status
4	STO-FBL	STO-Feedback, Low: Rückmeldeausgang für den STO-Status
5	STO-B-IN	STO-Kanal B, Gate-Treiber-Schalter
6	STO-B-RTN	GND-Anschluss für STO-Kanal B

STO-Funktion

Fällt die Spannung an einem der zwei STO-Eingängen (Kanal A und B) unter 7 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* auf nanotec.de.

4 Inbetriebnahme

In diesem Kapitel wird beschrieben, wie Sie die Kommunikation zur Steuerung aufbauen und die notwendigen Parameter einstellen, damit der Motor betriebsbereit ist.

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

Die Steuerung bietet Ihnen auch die Möglichkeit, *spezielle Fahrmodi* über das Objekt 4015_h aus-/einzuschalten. Damit können Sie den Motor direkt über die Eingänge (Analogeingang / Takt-Richtung) ansteuern. Siehe Kapitel Spezielle Fahrmodi für Details.

Beachten Sie folgende Hinweise:

VORSICHT!

Bewegte Teile können zu Handverletzungen führen!

Wenn Sie im laufenden Betrieb bewegte Teile anfassen, kann dies zu Handverletzungen führen.

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

VORSICHT!

Motorbewegungen sind im freistehenden Betrieb unkontrolliert und können Verletzungen hervorrufen.

Wenn der Motor unbefestigt ist, kann der Motor z. B. herunterfallen. Das kann zu Fußverletzungen oder zu Beschädigungen am Motor führen.

► Wenn Sie den Motor frei stehend betreiben, beobachten Sie den Motor, schalten Sie ihn bei Gefahr sofort ab und achten Sie darauf, dass der Motor nicht herunterfallen kann.

VORSICHT!

Bewegte Teile können Haare und lose Kleidung erfassen.

Im laufenden Betrieb können Haare oder lose Kleidung erfasst werden, dies kann zu Verletzungen führen.

► Bei langen Haaren tragen Sie ein Haarnetz oder andere geeignete Schutzmaßnahmen, wenn Sie in dem Bereich bewegter Teile sind. Arbeiten Sie nicht mit loser Kleidung oder Krawatten in der Nähe bewegter Teile.

VORSICHT

STO

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

STO

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 und die Spannungsversorgung.

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

HINWEIS

EMV: Stromführende Leitungen – insbesondere um Versorgungsleitungen – 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-Leitungen so kurz wie möglich halten.

► Kabel mit paarweise verdrillten Adern verwenden.

► Motorgehäuse großflächig auf kurzem Weg erden.

► Versorgungs- und Steuerleitungen getrennt verlegen.

HINWEIS

Störung des Motors und anderer Geräte durch elektromagnetische Wechselfelder!

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

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

► Kabel mit paarweise verdrillten Adern verwenden.

► Stromversorgungs- und Motorleitungen so kurz wie möglich halten.

► Motorgehäuse großflächig auf kurzem Weg erden.

► Versorgungs-, Motor- und Steuerleitungen räumlich getrennt verlegen.

4.1 Konfiguration über USB

4.1.1 Allgemeines

Es gibt folgende Möglichkeiten, die Steuerung 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.

► USB anschließen, bevor Sie die Versorgungsspannung einschalten.

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

► USB an Laptop nur im Akkubetrieb anschließen oder USB-Isolatior verwenden.

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

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

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

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

HINWEIS

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

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

TIPP

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

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

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

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

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

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 $203B_h$:01 (Nennstrom) auf den Wert "600" (mA):

```
203B:01=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 `vmmcode.usr`.
5. Navigieren Sie im Explorer in das Verzeichnis mit Ihrem Programm. Die compilierte Datei hat den gleichen Namen wie die Sourcecode-Datei, nur mit der Dateinamen-Endung `.usr`. Benennen Sie diese Datei in `vmmcode.usr` um.
6. Kopieren Sie die Datei `vmmcode.usr` auf die Steuerung.
   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 `vmmcode.usr` haben.
- Falls das *NanoJ-Programm* gelöscht wurde, wird mit dem nächsten Start eine leere Datei namens `vmmcode.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>:\vmmcode.usr
```

Also zum Beispiel:

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

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

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

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

4.2 Inbetriebnahme EtherNet/IP™

Diese Steuerung ist mit einer EtherNet/IP™-Schnittstelle ausgestattet. Lesen Sie das Kapitel EtherNet/IP für weitere Details.

HINWEIS

Nanotec Steuerungen basieren immer auf dem CANopen-Standard CiA402. Daher werden alle Attribute in einem sogenannten *Objektverzeichnis* (engl. *object dictionary*) gespeichert. Diese können mit einem Index und einem Subindex adressiert werden. z.B. 1018_h:01_h. Der Index ist ein 16-bit Wert und der Subindex ein 8-bit Wert.

Die Steuerung bietet EtherNet/IP™, aber dieses Protokoll hat ein anderes Vorgehen: alle Attribute werden auf Basis von Objekten gespeichert. Aus diesem Grund nutzt der *PD6-E* eine Zuordnung um eine Brücke zwischen EtherNet/IP™ und CANopen zu schlagen. Als Folge ist die Konfiguration dieser Steuerung anders als bei normalen EtherNet/IP™-Geräten.

Die folgende Inbetriebnahme-Prozedur geht davon aus dass eine *CompactLogix* PLC und *Studio 5000* Software von Rockwell benutzt werden.

4.2.1 Anschluss

1. Schließen Sie die Logik-Versorgungsspannung an den Stecker X12 (siehe Kapitel X2 — Ein-/Ausgänge und Logikversorgung) an.
2. Verbinden Sie die Rockwell *CompactLogix* PLC mit dem Anschluss X3 der Steuerung (siehe Kapitel X3 — EtherNet/IP IN).

4.2.2 Software Verbindung

Standardmäßig ist die Steuerung im DHCP-Modus, daher wird ein DHCP-Server in dem Netzwerk benötigt. Falls kein DHCP-Server verfügbar ist oder die Steuerung mit einer festen IP-Adresse arbeiten soll, kann das Tool *BOOTP/DHCP* von Rockwell benutzt werden. Mit diesem Tool lässt sich entweder eine IP-Adresse mittels DHCP der Steuerung zuweisen oder es lässt sich eine statische Adresse zuweisen und DHCP deaktivieren. BOOTP wird von der Steuerung nicht unterstützt.

Falls Sie einen eigenen DHCP-Server besitzen und die IP-Adresse herausfinden wollen, lässt sich das am einfachsten über das Tool *ping* bewerkstelligen. Dazu muss der NetBIOS-Service auf dem PC aktiviert sein und die MAC-Adresse der Steuerung muss bekannt sein.

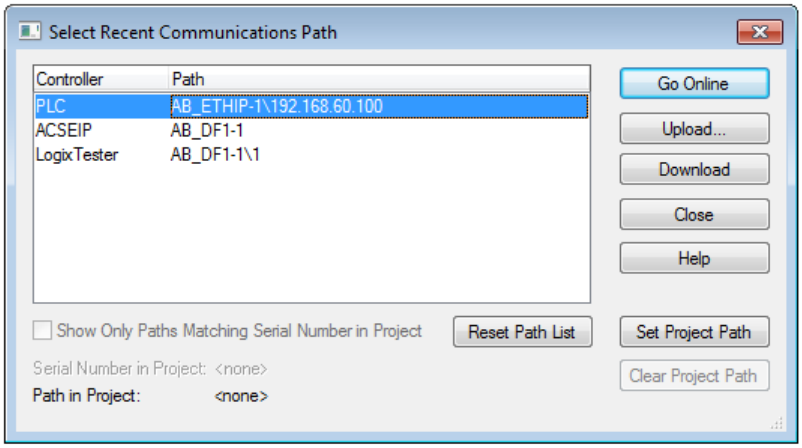
Beispiel

Falls die Steuerung mit der MAC Adresse 44:AA:E8:00:02:9F angesprochen werden soll, ist der Aufruf für das Tool in einer Shell oder Command-Line:

```
ping MAC-44AAE800029F
```

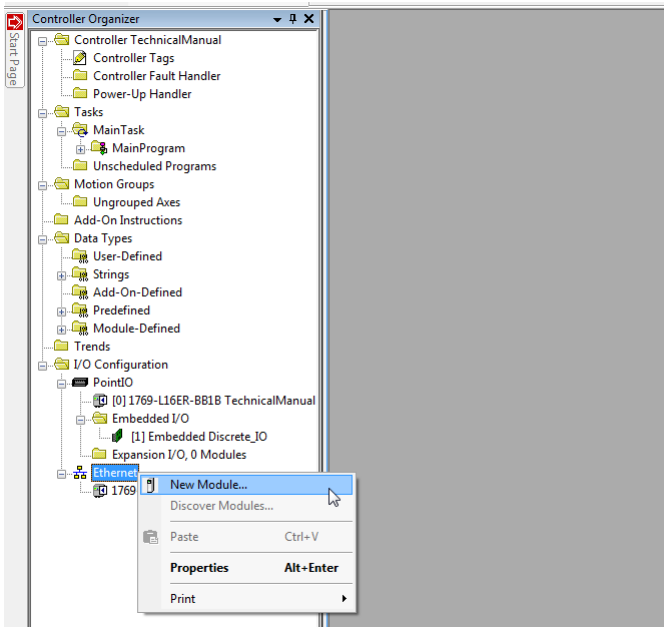
Sie müssen die nächsten Schritte im Rockwell *Logix Designer* machen:

1. Benutzen sie die Software *RSLinx Classic*, um einen EtherNet/IP™-Treiber zu erstellen. Schlagen Sie im entsprechenden Handbuch zur Hilfe nach.
2. Wählen Sie den Projekt-Pfad der PLC.

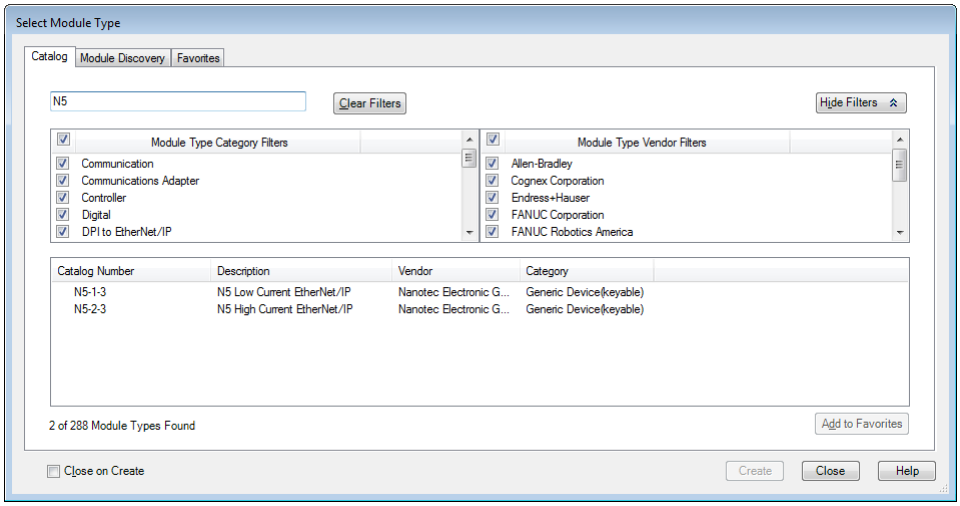


3. Importieren der EDS-Datei der Steuerung: Klicken Sie auf *Tools\EDS Hardware Installation Tool*, wählen sie *Register an EDS file(s)*. Wählen sie anschließend die korrekte EDS-Datei aus und importieren Sie diese.

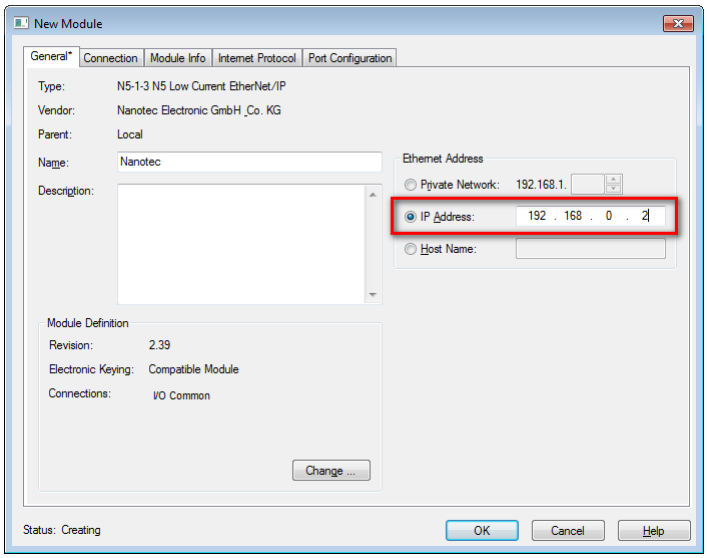
4. Klicken Sie mit der rechten Maustaste auf *Ethernet* im *Controller Organizer* und wählen Sie *New Module…*.



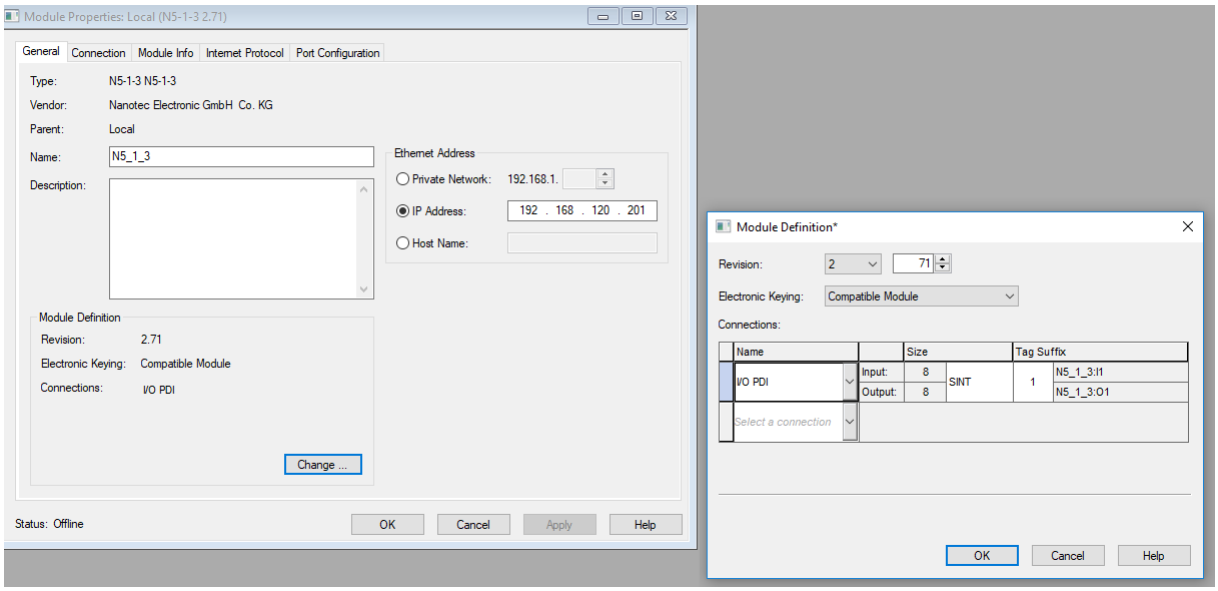
Wählen Sie die Registerkarte *Catalog*, suchen Sie den Eintrag *PD6-E* und wählen Sie das Gerät, mit dem Sie arbeiten wollen.



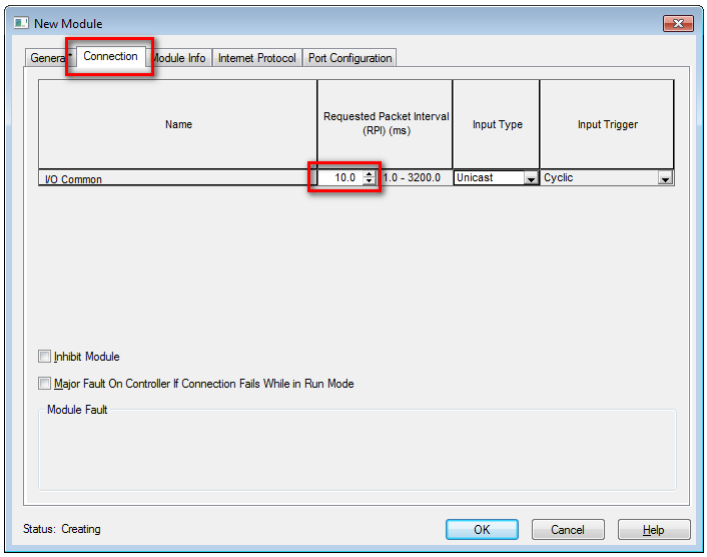
Anschließend wird die Eingabemaske *New Module* geöffnet. Wählen Sie die Registerkarte *General* und geben Sie einen Namen und die IP-Adresse für das Gerät ein. Die Benutzung des *Host Name* wird nicht unterstützt.



Klicken Sie auf *Change*, um eine der Assemblies auszuwählen (entweder *IIO Common* oder *I/O PDI*, siehe Kapitel Assembly-Objekte).



Wählen Sie die Registerkarte *Connection* und geben Sie einen RPI-Wert für die *I/O Common*-Daten an, wie es in der nachfolgenden Abbildung gezeigt wird. Zudem kann der *input type* zwischen *unicast* und *multicast* geändert werden.



Zum Schluss schließen Sie die Maske mit einem Klick auf OK.

4.3 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
$33A0_h$	Inkrementaler Encoder 1	enthält Konfigurationswerte für den ersten inkrementalen Encoder
$33B0_h$	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 *Tuning*, siehe Objekte speichern).

Für externe Sensoren, die nicht direkt auf der Motorwelle montiert sind, müssen Sie entsprechend der konstruktiven Gegebenheiten die Getriebeübersetzung (Objekte $60E8_h$ und $60ED_h$) und/oder die Vorschubkonstante (Objekte $60E9_h$ und $60EE_h$) 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
60E8h Additional Gear Ratio - Motor Shaft Revolutions	4
60EDh Additional Gear Ratio - Driving Shaft Revolutions	1
60E9h Additional Feed Constant - Feed	2513274 Inkr. (entspricht 1256,637 mm)
60EEh Additional Feed Constant - Driving 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, 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.

4.4 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.

HINWEIS

Bei den Plug & Drive Motoren ist es nicht notwendig ein Auto-Setup auszuführen, da dieses bereits werksseitig durchgeführt wurde. Eine Ausnahme besteht bei Anschluss eines externen Encoders an X2.

4.4.1 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)	✓

Parameter	Alle Motoren unabhängig von der Konfiguration
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.

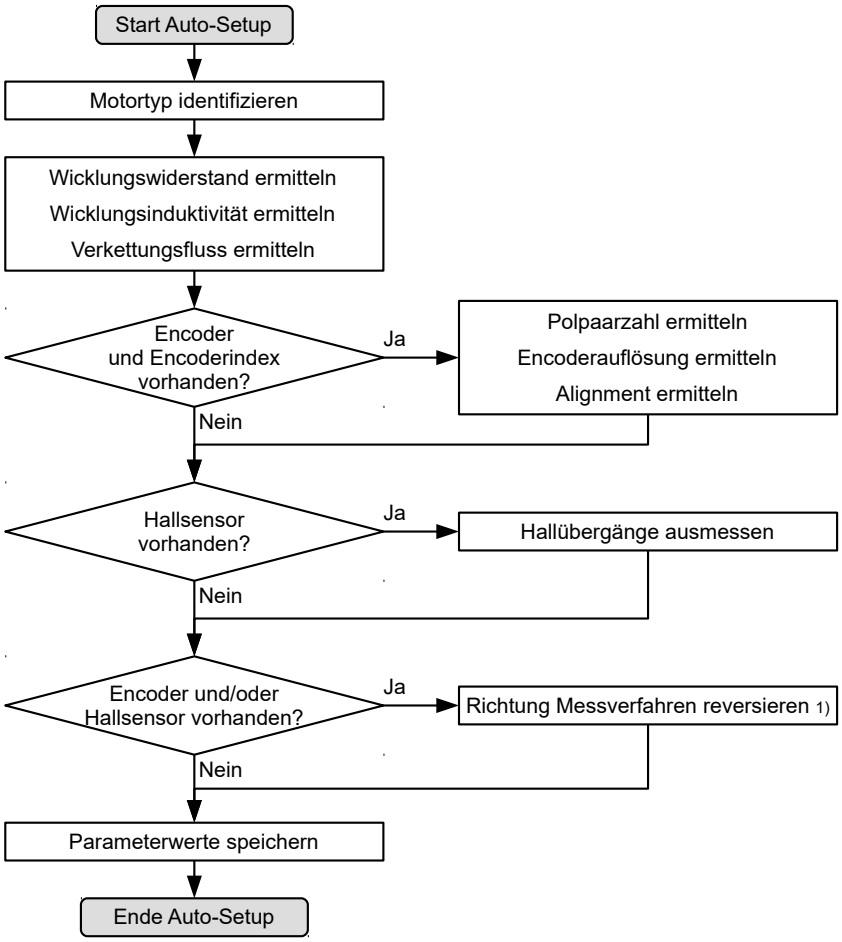
Parameter	Motor ohne Encoder	Motor mit Encoder und Index	Motor mit Encoder ohne Index
Encoderauflösung	-	✓	---
Alignment (Verschiebung des elektrischen Nullpunkts zum Index)	-	✓	---

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

4.4.2 Durchführung

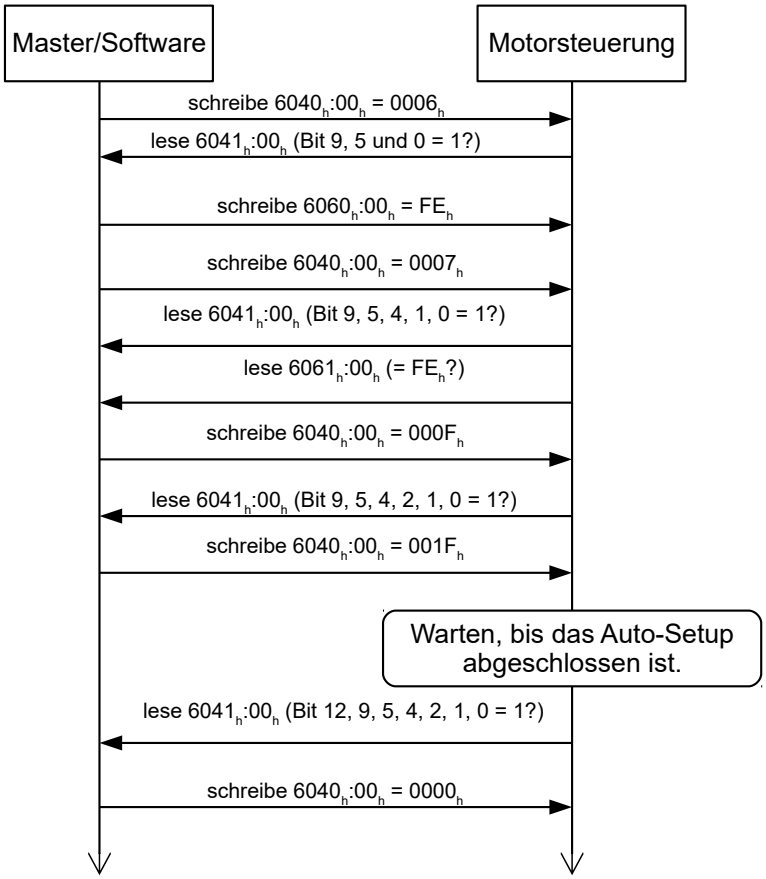
1. Zum Vorwählen des Betriebsmodus *Auto-Setup* tragen Sie in das Objekt 6060_h:00_h den Wert "-2" (="FE_h") ein.
   Die *Power state machine* muss nun in den Zustand *Operation enabled* versetzt werden, siehe CiA 402 Power State Machine.
2. Starten Sie das *Auto-Setup* mit Setzten von Bit 4 *OMS* im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



4.4.3 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.

4.5 Spezielle Fahrmodi

Sie haben die Möglichkeit, den Motor direkt über den Analogeingang anzusteuern, indem Sie die *speziellen Fahrmodi* aktivieren. Darunter zählen:

- Analog-Drehzahl
- Automatische Fahrt mit 30 U/min (Testfahrt)

Sie können ebenso die Betriebsart, *Open Loop* oder *Closed Loop*, bestimmen.

Der digitale Eingang 3 dient dabei als Freigabe (siehe X2 — Ein-/Ausgänge und Logikversorgung).

HINWEIS

Der Zustand der CiA 402 Power State Machine wird nach Aktivierung der *speziellen Fahrmodi* nur über einen digitalen Eingang (Freigabe) gesteuert. Zustandsänderungen, die im Objekt 6040_h (Controlword) angefordert werden, haben keine Auswirkung.

4.5.1 Aktivierung

Um die *speziellen Fahrmodi* zu aktivieren, müssen Sie in 4015_h:01_h den Wert "2" eintragen. In 4015_h:02_h stellen Sie den Modus ein, indem Sie einen Wert zwischen "00"$_h$ und "0F"$_h$ schreiben.

Die folgende Tabelle listet alle möglichen Modi und den Wert für 4015:02_h auf:

Wert	Modus			
02_h	Takt-Richtung (Testfahrt)	Automatische Fahrt mit 30 U/min	Drehrichtung im Uhrzeigersinn	Open Loop
03_h	Takt-Richtung (Testfahrt)	Automatische Fahrt mit 30 U/min	Drehrichtung gegen den Uhrzeigersinn	Open Loop
04_h	Analog-Drehzahl	Richtung über "Richtungs"-Eingang	Maximale Drehzahl1000 U/min	Open Loop
05_h	Analog-Drehzahl	Richtung über "Richtungs"-Eingang	Maximale Drehzahl100 U/min	Open Loop
06_h	Analog-Drehzahl	Offset 5 V (Joystick-Modus)	Maximale Drehzahl1000 U/min	Open Loop
07_h	Analog-Drehzahl	Offset 5 V (Joystick-Modus)	Maximale Drehzahl 100 U/min	Open Loop
$0A_h$	Takt-Richtung (Testfahrt)	Automatische Fahrt mit 30 U/min	Drehrichtung im Uhrzeigersinn	Closed Loop
$0B_h$	Takt-Richtung (Testfahrt)	Automatische Fahrt mit 30 U min	Drehrichtung gegen den Uhrzeigersinn	Closed Loop
$0C_h$	Analog-Drehzahl	Richtung über "Richtungs"-Eingang	Maximale Drehzahl 1000 U/min	Closed Loop
$0D_h$	Analog-Drehzahl	Richtung über "Richtungs"-Eingang	Maximale Drehzahl 100 U/min	Closed Loop
$0E_h$	Analog-Drehzahl	Offset 5 V (Joystick-Modus)	Maximale Drehzahl1000 U/min	Closed Loop
$0F_h$	Analog-Drehzahl	Offset 5 V (Joystick-Modus)	Maximale Drehzahl 100 U/min	Closed Loop

Sie müssen das Objekt 4015_h (*Kategorie Applikation*) speichern (siehe Kapitel Objekte speichern), die Änderungen werden erst nach einem Neustart der Steuerung aktiviert.

4.5.2 Analog-Drehzahl

Die Steuerung setzt intern den Betriebsmodus auf Velocity. Zur Vorgabe der Drehzahl wird die Spannung am analogen Eingang benutzt und die entsprechende Zielgeschwindigkeit wird in 6042_h geschrieben.

4.5.2.1 Maximale Drehzahl

Die maximale Drehzahl kann zwischen 100 U/min und 1000 U/min gewechselt werden, dabei passt die Steuerung die Skalierung in $604C_h$ automatisch an.

HINWEIS

Sollten Sie danach in einen anderen Modus wechseln wollen, müssen Sie ggf. die Skalierung in $604C_h$ anpassen bzw. zurücksetzen.

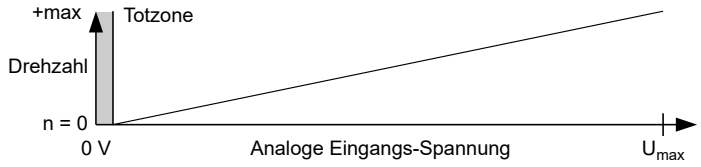
Ist eine andere Drehzahl notwendig, dann lässt sich diese über den Skalierungsfaktor für die Geschwindigkeit (Objekt $604C_h$) oder den Analogwert (siehe Analoge Eingänge) einstellen.

4.5.2.2 Verrechnung der Analogspannung

Es gibt zwei Modi, wie die analoge Eingangsspannung verrechnet wird.

Normaler Modus

Sie müssen die Eingänge *Freigabe*, *Richtung* und den *Analogeingang* beschalten (siehe Kapitel X2 — Ein-/Ausgänge und Logikversorgung). Das Maximum der analogen Spannung entspricht der maximalen Drehzahl. Die Richtung wird dabei über den Richtungseingang vorgegeben. Wenn kein Signal am Richtungseingang anliegt, dreht sich der Motor im Uhrzeigersinn bei Blick auf die Antriebswelle). Es existiert eine Totzone von 0 V bis 20 mV, in welcher der Motor nicht fährt.



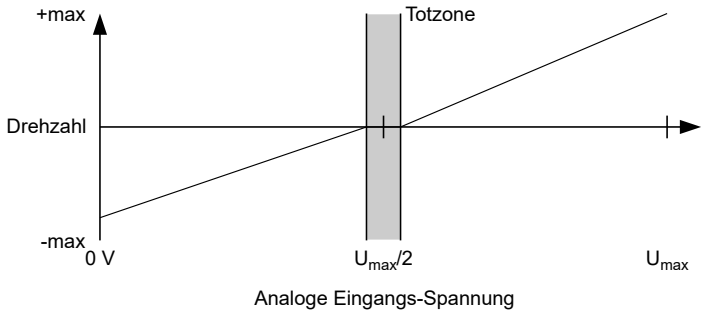
Joystick Modus

Sie müssen den Eingang *Freigabe* und den *Analogeingang* beschalten (siehe Kapitel X2 — Ein-/Ausgänge und Logikversorgung). Die Hälfte der maximalen, analogen Spannung entspricht der Drehzahl 0, dabei passt die Steuerung den Offset in 3321_h automatisch an.

HINWEIS

Sollten Sie danach in einen anderen Modus wechseln wollen, müssen Sie ggf. den Offset in 3321_h anpassen bzw. zurücksetzen.

Sinkt die Spannung unter die Hälfte, steigt die Drehzahl in negativer Richtung. Wenn die Spannung entsprechend über die Hälfte steigt, steigt auch die Drehzahl in positiver Richtung. Die Totzone geht dabei von $U_{max}/2 \pm 20$ mV.



4.5.3 Automatische Fahrt mit 30 U/min (Testfahrt)

Der Motor dreht mit 30 U/min wenn der Eingang *Freigabe* gesetzt ist.

5 Generelle Konzepte

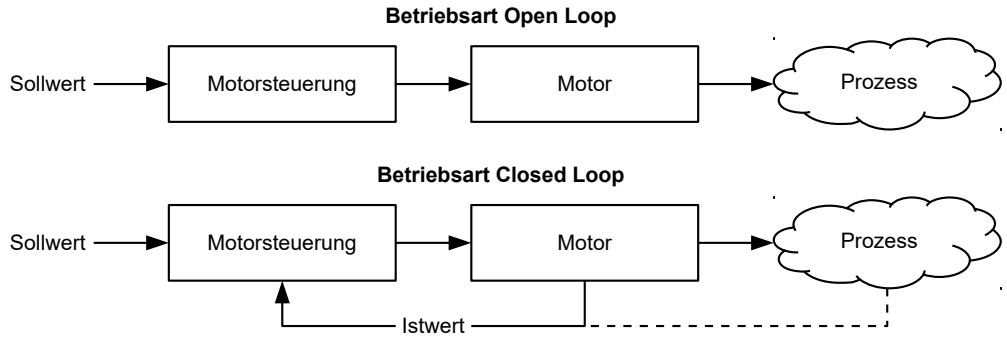
5.1 Betriebsarten

5.1.1 Allgemein

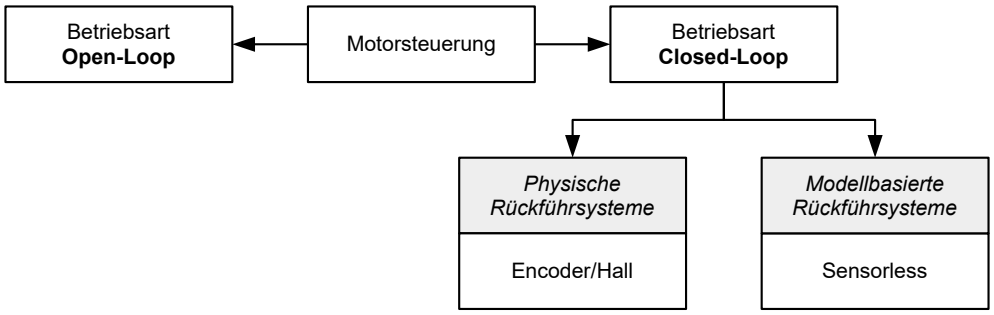
Die Betriebsart von Systemen ohne Rückführung wird als *Open Loop*, die mit Rückführung als *Closed Loop* bezeichnet. In der Betriebsart *Closed Loop* ist es zunächst unerheblich, ob die zurückgeführten Signale vom Motor selbst oder aus dem beeinflussten Prozess kommen.

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

Speziell für Applikationen im niedrigen Drehzahlbereich hat Nanotec die Betriebsart Slow Speed entwickelt, die eine Mischung aus *Open Loop* und *Closed Loop* ist. Diese Betriebsart kann angewendet werden, wenn ein Encoder als Rückführung vorhanden ist.

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	Slow Speed
Profile Position	ja	ja	ja
Velocity	ja	ja	ja
Profile Velocity	ja	ja	ja
Profile Torque	nein[1)]	ja	nein
Homing	ja[2)]	ja	ja
Interpolated Position Mode	ja[3)]	ja	ja
Cyclic Synchronous Position	ja[3)]	ja	ja
Cyclic Synchronous Velocity	ja[3)]	ja	ja
Cyclic Synchronous Torque	nein[1)]	ja	nein

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 2031_h:00_h den maximal zulässigen Motorstrom (Motorschutz) in mA eingeben (siehe Motordatenblatt)
- Im Objekt 6075_h:00_h den Nennstrom des Motors in mA (siehe Motordatenblatt) eingeben.
- Im Objekt 6073_h:00_h: den Maximalstrom (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 6073_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 2036_h (Open Loop Current Reduction Idle 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 2037_h (Open Loop Current Reduction Value/factor) wird der Effektivwert angegeben, auf den der Nennstrom reduziert werden soll, wenn die Stromabsenkung im *Open Loop* aktiviert wird und sich der Motor im Stillstand befindet.

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.
- Die Regelparameter des Stromreglers über die Objekte 3210_h:09_h (I_P) und 3210_h:$0A_h$ (I_I) optimieren (in der Regel nicht notwendig).
- Anpassen der Beschleunigung, Verzögerung und/oder Zielgeschwindigkeit in Abhängigkeit des gewählten Betriebsmodus:

  Betriebsmodus Profile Position

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

  Betriebsmodus Velocity

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

  Betriebsmodus Profile Velocity

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

  Betriebsmodus Homing

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

  Betriebsmodus Interpolated Position Mode

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

  Betriebsmodus Cyclic Synchronous Position

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

Betriebsmodus Cyclic Synchronous Velocity

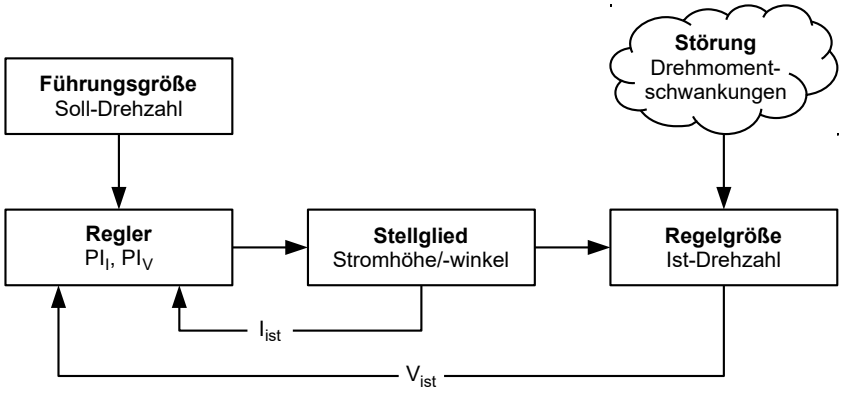
Über die externen Zielvorgaben "Positionsvorgabe/Zeiteinheit" können die Beschleunigungs- und Verzögerungsrampen beeinflusst werden.

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

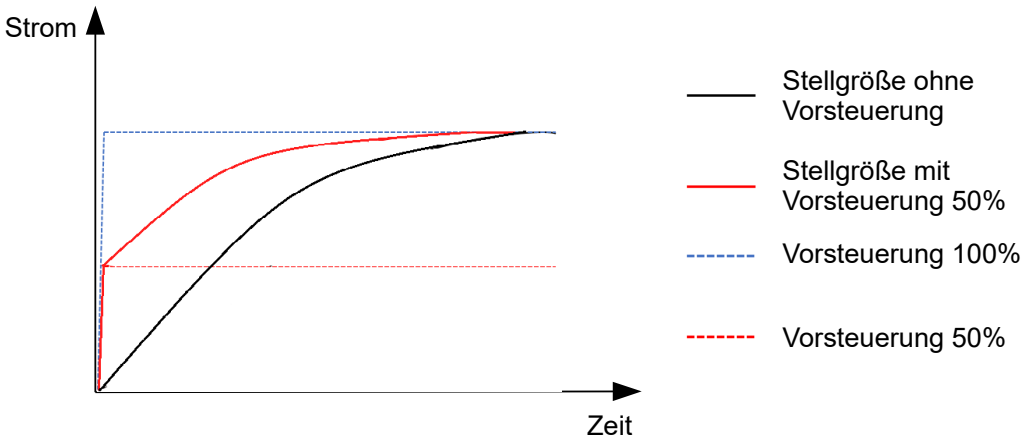
5.1.3.3 Vorsteuerung

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".

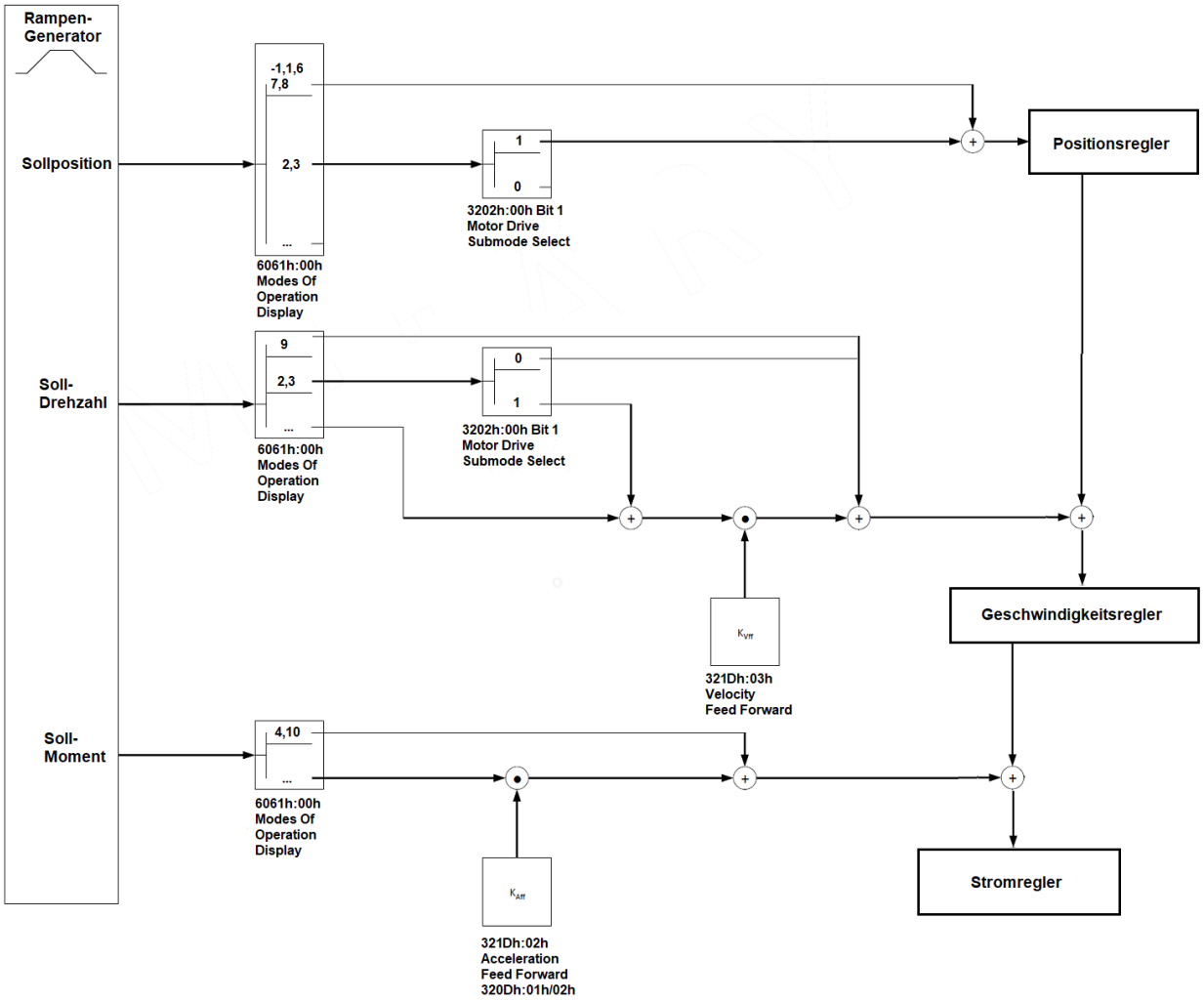


Der Faktor für die *Geschwindigkeitsvorsteuerung* wird im Objekt $321D_h$:03_h in Promille des Ausgangs des Rampengenerators ($606B_h$) eingestellt und vor dem Geschwindigkeitsregler zum Ausgang des Positionsreglers addiert. Die *Geschwindigkeitsvorsteuerung* ist in allen Modi mit Positionsregelkreis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

- Profile Position
- Homing
- Interpolated Position Mode
- Cyclic Synchronous Position
- Takt-Richtungs-Modus
- Velocity/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.



Der Faktor für die *Spannungsvorsteuerung* wird im Objekt 321D$_h$:01$_h$ in Promille der Spannung definiert, die benötigt wird, um den Sollstrom zu erzeugen. Ist der Faktor 1000‰ (Werkseinstellung), steht die Spannung sofort zur Verfügung und der Iststrom erreicht sehr schnell den Sollstrom. Dadurch existiert praktisch keine Regelabweichung beim Beschleunigen und der Stromregler wird entlastet.

Die *Spannungsvorsteuerung* ist in allen Modi aktiv. Um sie auszuschalten, setzen Sie 321D$_h$:01$_h$ auf "0".

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.

Beispiel

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

Bit	Regler	Rückführung 1 Sensorless	Rückführung 2 Hall	Rückführung 3 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.

Ist kein zweiter Kommutierungssensor ausgewählt, oder fehlt den ausgewählten Sensoren das Alignment, so wird nötigenfalls ein Auto-Alignment im *Open Loop* ermittelt (unabhängig vom Bit 4 in 3202_h).

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.

Bei den Plug & Drive Motoren ist es nicht notwendig das *Auto-Setup* auszuführen, da dieses bereits werksseitig durchgeführt wurde. Eine Ausnahme besteht bei Anschluss eines externen Encoders an X2.

Das Bit 0 im 3202_h muss gesetzt sein. Das Bit wird nach einem erfolgreich abgeschlossenen Auto-Setup automatisch gesetzt.

5.1.3.6 Optimierungen

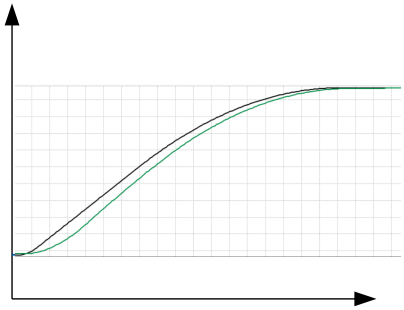
Im *Closed Loop* wird die gemessene Regelgröße (Istwert) permanent mit der Führungsgröße (Sollwert) verglichen. Bei Abweichungen zwischen diesen Größen regelt die Steuerung entsprechend den vorgegebenen Regelparametern nach.

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.

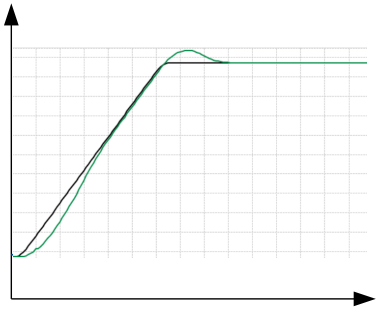
Es ist aufgrund der kaskadierten Reglerstruktur sinnvoll, mit der Optimierung des innersten Reglers (Stromreglers) zu beginnen, bevor der Geschwindigkeits- und ggf. der Positionsregler optimiert werden. Jeder der drei Regler besteht aus einem Proportional- und einem Integral-Anteil, die normalerweise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

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".

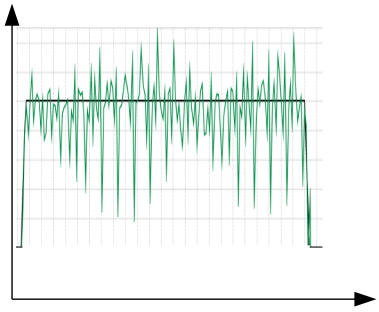


P-Anteil zu klein

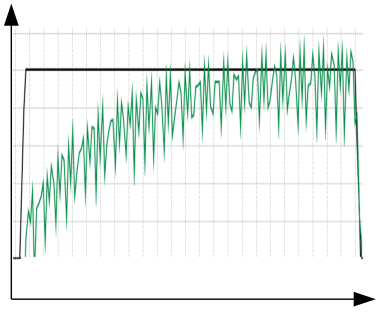


P-Anteil zu groß

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.

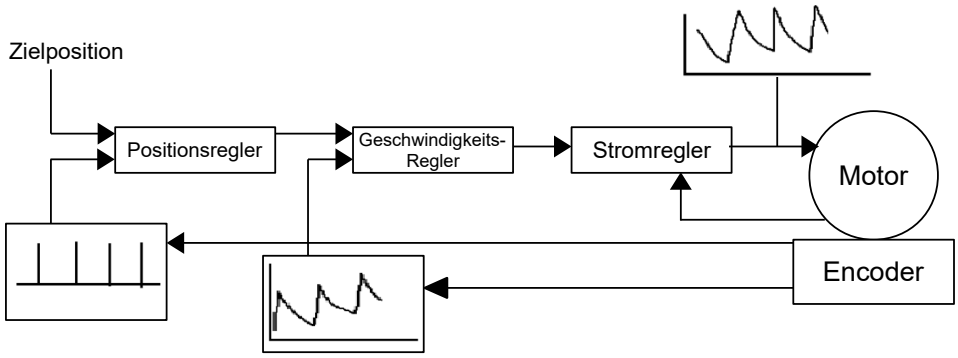
5.1.4 Slow Speed

5.1.4.1 Einführung

Die Betriebsart *Slow Speed* vereint die Vorteile der *Open Loop*- und *Closed Loop*-Technologie im niedrigen Drehzahlbereich und kann angewendet werden, wenn ein Encoder als Rückführung vorhanden ist. *Slow Speed* bietet eine Schleppfehlerüberwachung, weist aber einen laufruhigeren Betrieb auf, als im puren *Closed Loop*-Betrieb bei niedrigen Drehzahlen.

Die Rotorlage wird über die Signale des Encoders erfasst. Um die Geschwindigkeit zu berechnen, wird die Änderung der Position durch die (feste) Zykluszeit dividiert. Bei niedrigen Drehzahlen zählt der Controller weniger (oder gar keine) Encoder-Inkremente in einem Zyklus, was zu einer Geschwindigkeitskurve mit relativ vielen Spitzen führt (trotz des verwendeten Tiefpassfilters).

Wegen des kaskadierten Regelkreises führt dies im *Closed Loop*-Betrieb zu Stromspitzen, die einen unruhigen Lauf zufolge haben, wie die folgende Abbildung zeigt.



In der Betriebsart *Slow Speed* fährt der Motor im Gegenteil mit konstantem Phasenstrom, wie im *Open Loop*. Der Schleppfehler wird aber über den Encoder überwacht und die Vektorregelung des Magnetfelds wird ggf. aktiviert, wie im *Closed Loop*.

5.1.4.2 Aktivierung

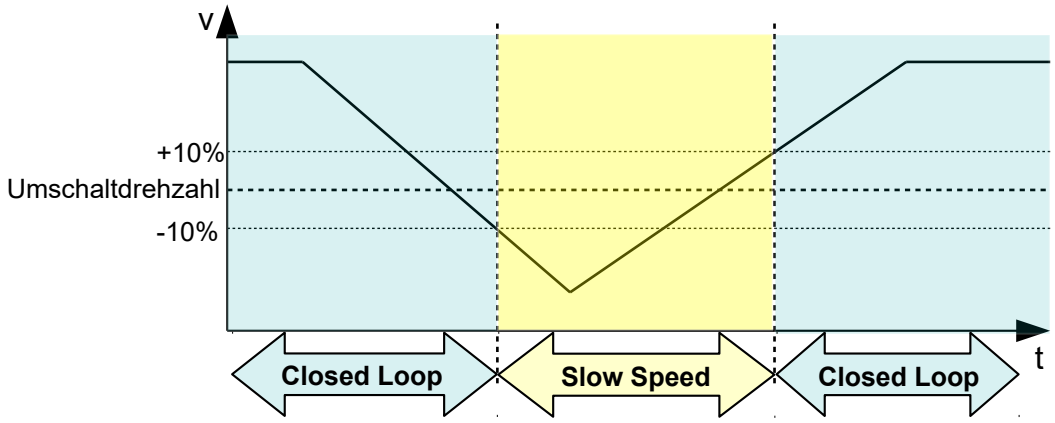
Um die Betriebsart *Slow Speed* zu aktivieren, müssen Sie:

1. den Closed Loop aktivieren,
2. das Bit 7 im Objekt 3202_h (Motor Drive Submode Select) auf "1" setzen.

Die Umschaltung zwischen *Slow Speed* und *Closed Loop* erfolgt automatisch bei einer von der physikalischen Encoderauflösung abhängigen Drehzahl, mit einer Hysterese von 10%. Diese feste Umschaltdrehzahl wird in Umdrehungen pro Minute berechnet wie folgt:

$$\frac{4000}{\text{Encoderauflösung (ppr)}} \times 60$$

Die nachfolgende Abbildung zeigt die Umschaltung in Abhängigkeit von der Drehzahl in beiden Richtungen.



Im Stillstand befindet sich der Motor im *Closed Loop*-Betrieb.

5.1.4.3 Optimierungen

Der gesamte Phasenstrom bleibt konstant, wie im *Open Loop*. Systembedingt können dann Resonanzen auftreten, die Sie durch Anpassung des Motorstroms und/oder der Beschleunigungsrampe vermeiden können. Siehe auch Kapitel Open Loop.

Bei Betrieb in unterschiedlichen Drehzahlbereichen, wenn zwischen *Closed Loop* und *Slow Speed* gewechselt wird, ist eventuell notwendig:

- den Motorstrom (Objekte 6075_h, 6073_h) zu reduzieren, wenn von *Closed Loop* in *Slow Speed* gewechselt wird,
- unterschiedliche Regelparameter (siehe Reglerstruktur) für jeden Drehzahlbereich zu ermitteln.

5.2 CiA 402 Power State Machine

5.2.1 Zustandsmaschine

5.2.1.1 CiA 402

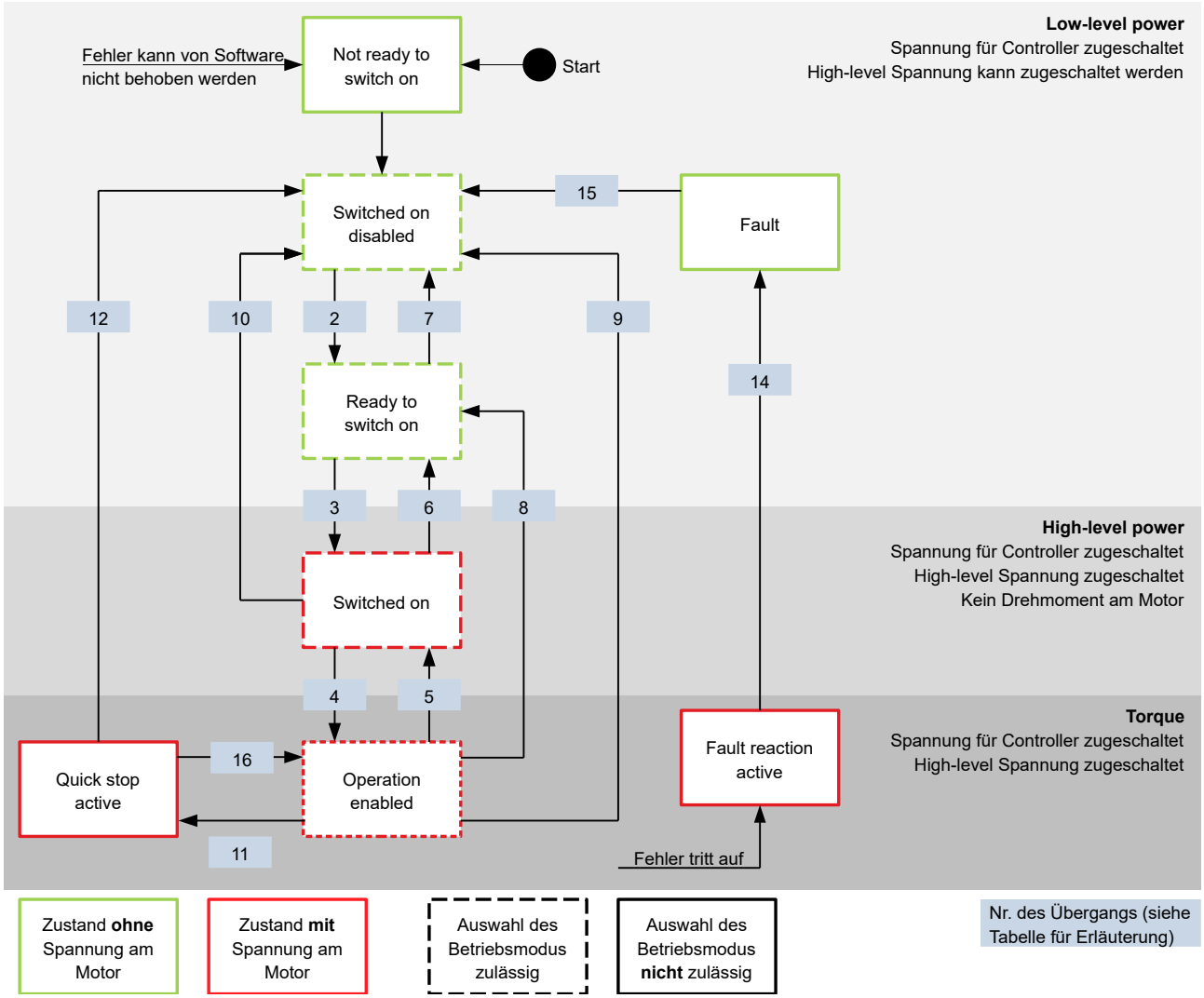
Um die Steuerung betriebsbereit zu schalten, ist es notwendig, eine Zustandsmaschine (*State Machine*) zu durchlaufen. Diese ist im *CANopen-Standard 402* definiert. Zustandsänderungen werden im Objekt 6040_h (Controlword) angefordert. Der tatsächliche Zustand der Zustandsmaschine lässt sich aus dem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

Kommando	Bit im Objekt 6040_h					Übergang
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Enable operation	0	1	1	1	1	4
Enable operation after Quick stop	0	1	↑	1	1	16
Fault / warning 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.

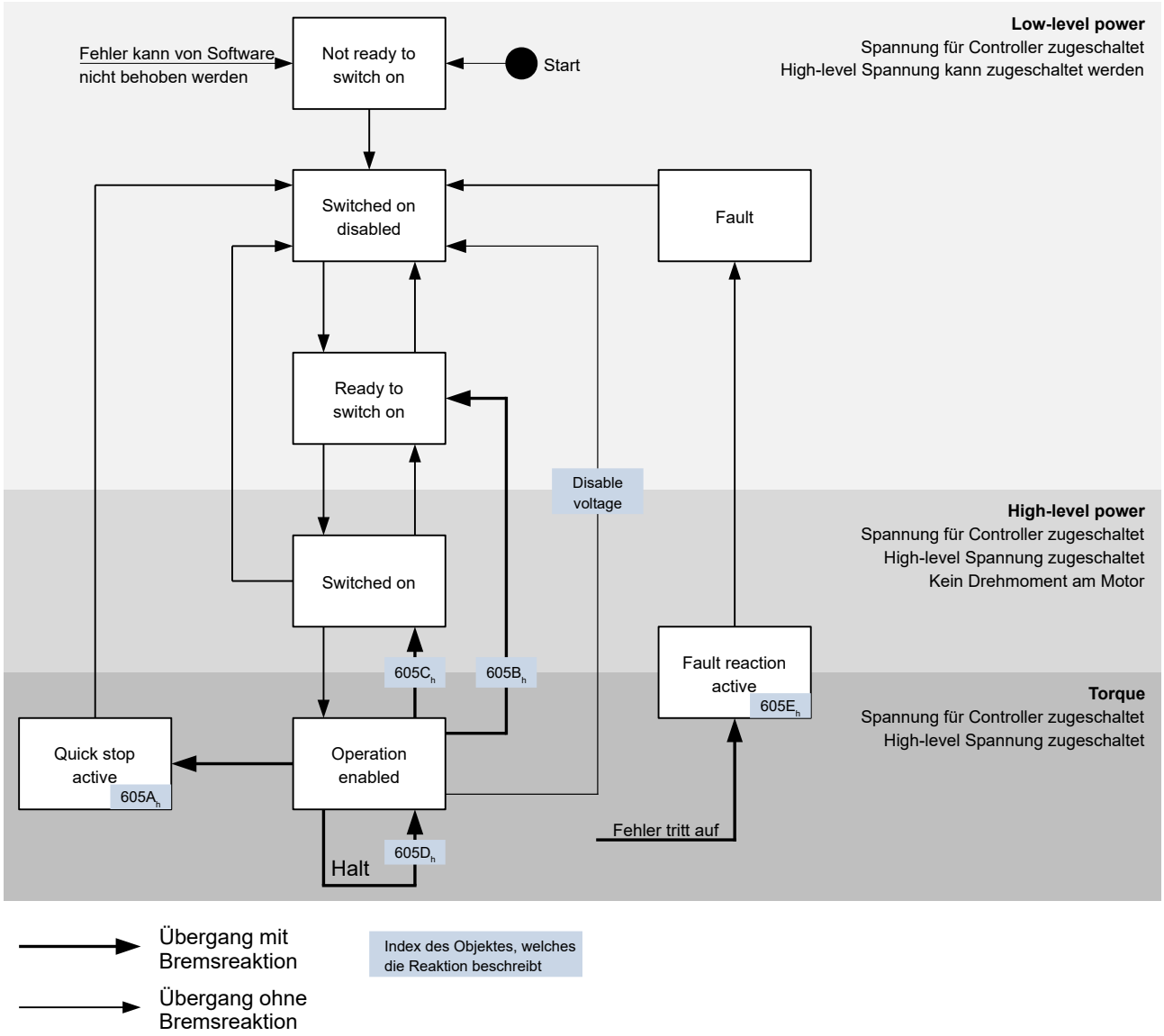
Die Einstellung oder Änderung des Betriebsmodus ist jederzeit möglich.

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
- Velocity
- 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.

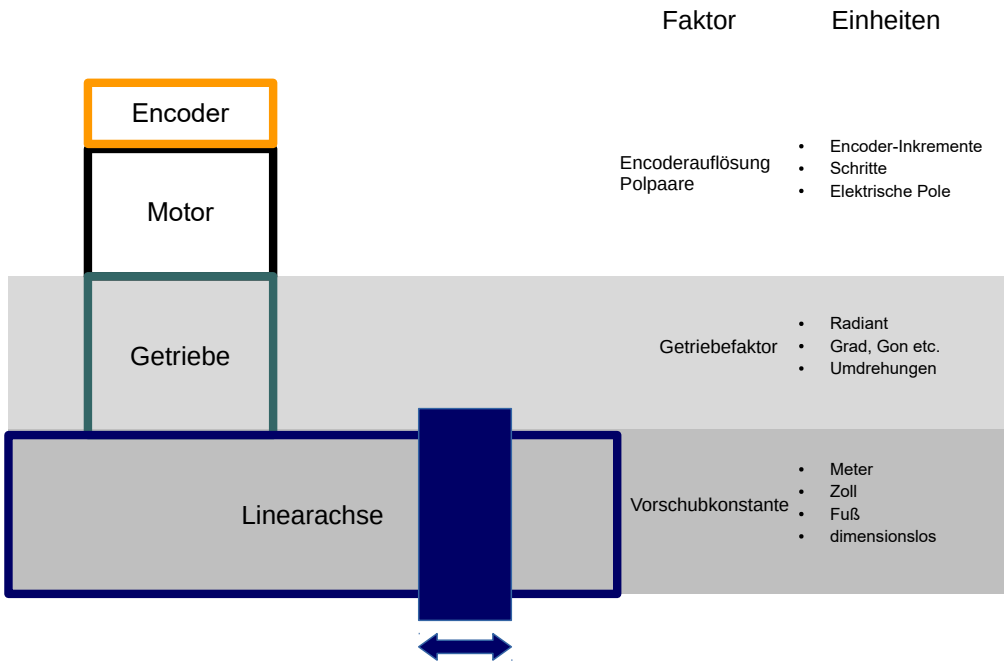
Wert	Beschreibung
-32768 bis -2	Reserviert
-1	keine Reaktion
0	Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen
1	Abbremsen mit *slow down ramp* (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit *quick stop ramp* (6085_h)
3 bis 32767	reserviert

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 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 pole		$C0_h$	nein	nein	*Elektrische Pole.* Bei einem Schrittmotor, der z.B. 50 Polpaare (2030_h) hat, entspricht die Einheit 1/50 einer 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 ($60E8_h$ (Motor Shaft Revolutions)) pro Achsenumdrehungen ($60ED_h$ (Driving Shaft Revolutions)).

5.3.4 Vorschubkonstante

Die Vorschubkonstante berechnet sich aus dem Vorschub in benutzerdefinierten Positionseinheiten ($60E9_h$ (Feed) pro Umdrehung der Abtriebsachse ($60EE_h$ (Driving Shaft Revolutions).

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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Exponent einer Zehnerpotenz								Einheit							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserviert (00h)								reserviert (00h)							

Beispiel

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

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

Beispiel

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

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

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

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. Wenn Sie keinen externen Encoder an X2 angeschlossen haben, wird der interne Absolut-Encoder verwendet.

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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Exponent einer Zehnerpotenz								Positionseinheit							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Zeiteinheit								reserviert (00h)							

Beispiel

Wird 60A9$_h$ mit dem Wert "00B44700$_h$" beschrieben (Bits 8-15=47$_h$, Bits 16-23=B4$_h$ und Bits 24-31=00$_h$), wird die Einheit auf *Umdrehungen pro Minute* eingestellt (Werkseinstellung).

Beispiel

Wird das 60A9$_h$ mit dem Wert "FD010300$_h$" beschrieben (Bits 8-15=03$_h$, Bits 16-23=01$_h$ und Bits 24-31=FD$_h$(=-3)), 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. Wenn Sie keinen externen Encoder an X2 angeschlossen haben, wird der interne Absolut-Encoder verwendet.

HINWEIS

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

Umrechnungsfaktor für die Geschwindigkeitseinheit

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

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

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	50 µs (20 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

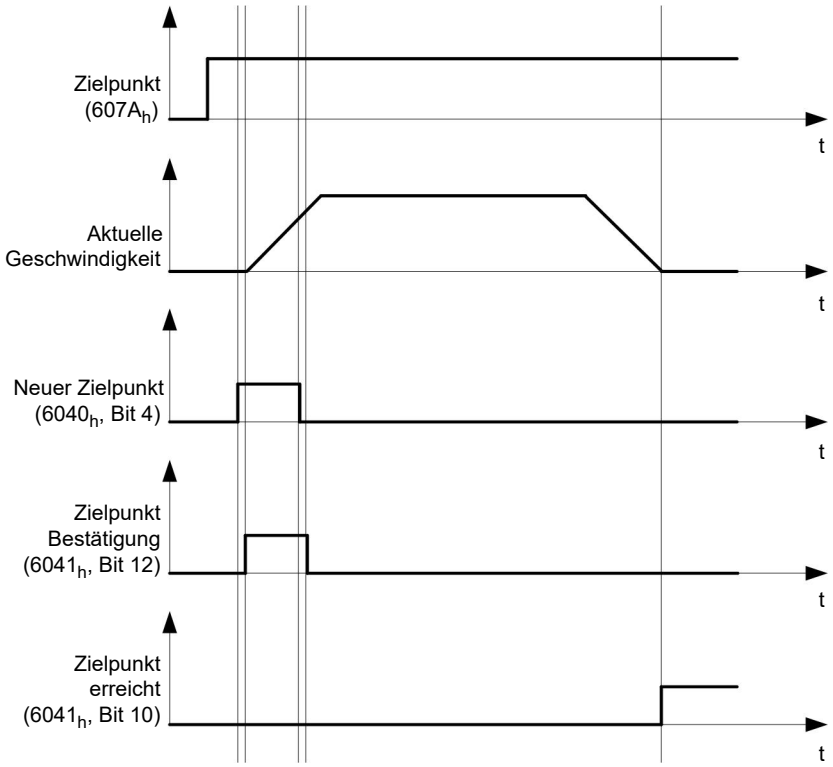
Folgende Bits im Objekt 6041_h (Statusword) haben eine gesonderte Funktion:

- 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.



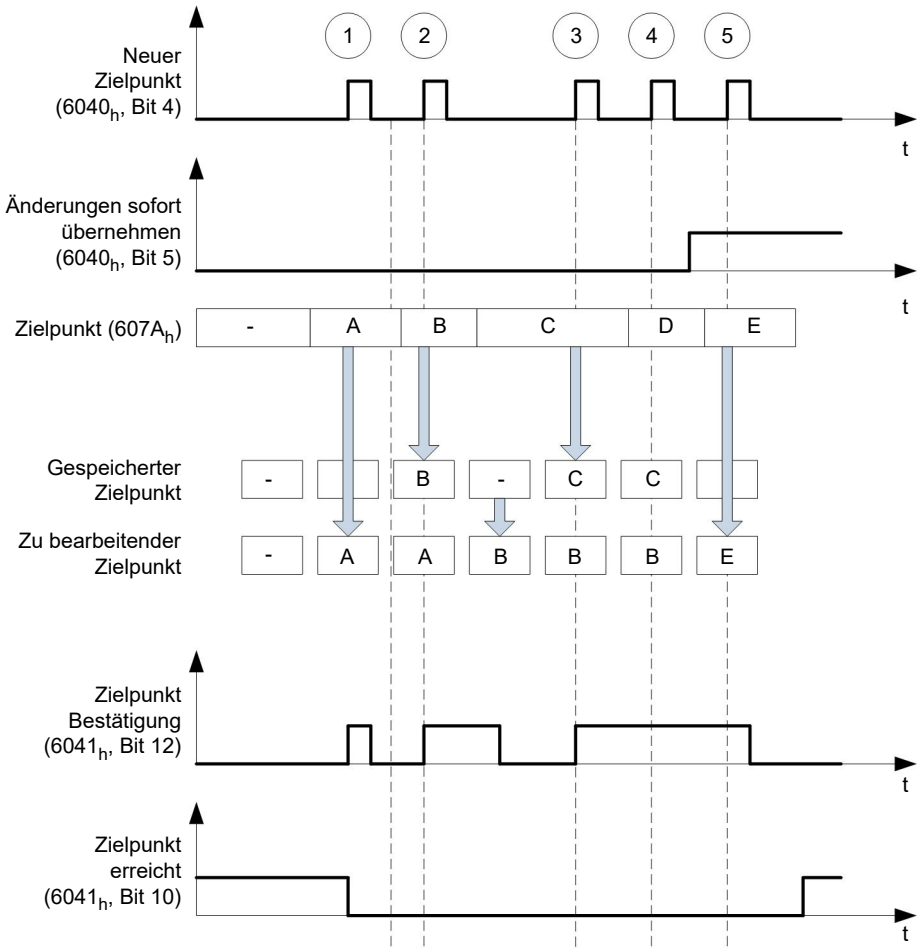
Die Steuerung kann das Bit 4 im Objekt 6040_h (Controlword) auch selbstständig zurücksetzen. Das wird mit den Bits 4 und 5 des Objektes $60F2_h$ eingestellt.

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).

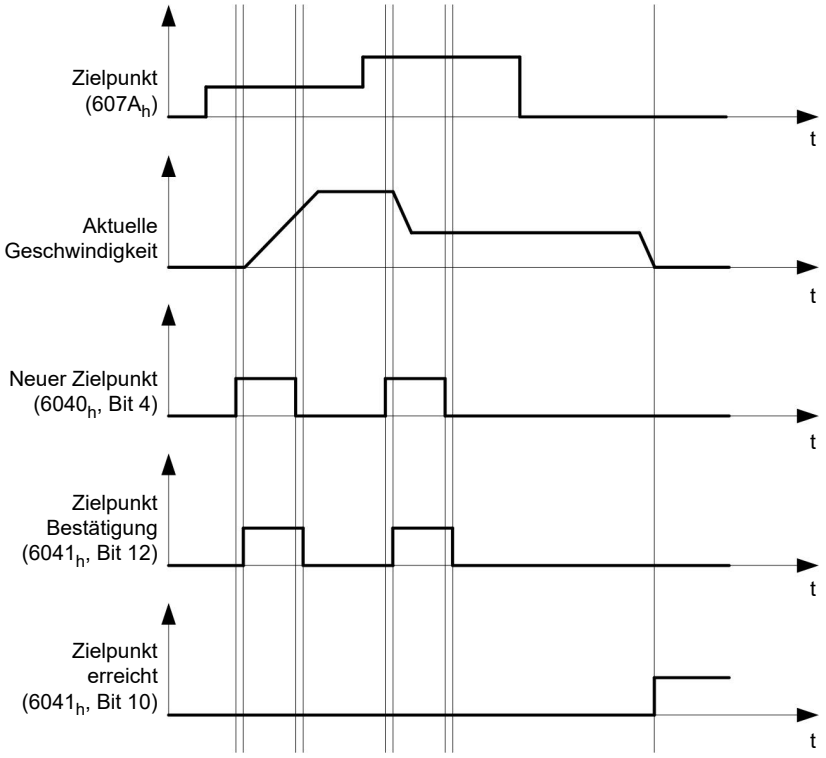
Sollte der Zwischenspeicher schon voll sein, wird ein neuer Zielpunkt ignoriert (Zeitpunkt 4). Wird Bit 5 im Objekt 6040_h (Controlword, Bit: "Change Set-Point Immediately") gesetzt, arbeitet die Steuerung ohne den Zwischenspeicher, neue Fahrbefehle werden direkt umgesetzt (Zeitpunkt 5).

Zeitpunkte



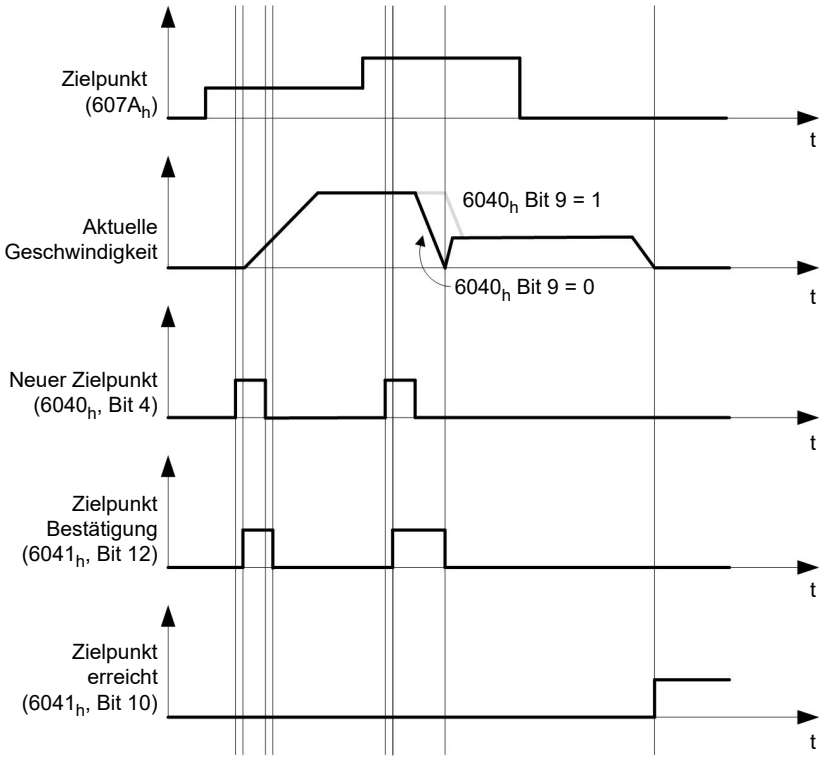
Übergangsprozedur für zweite Zielposition

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.

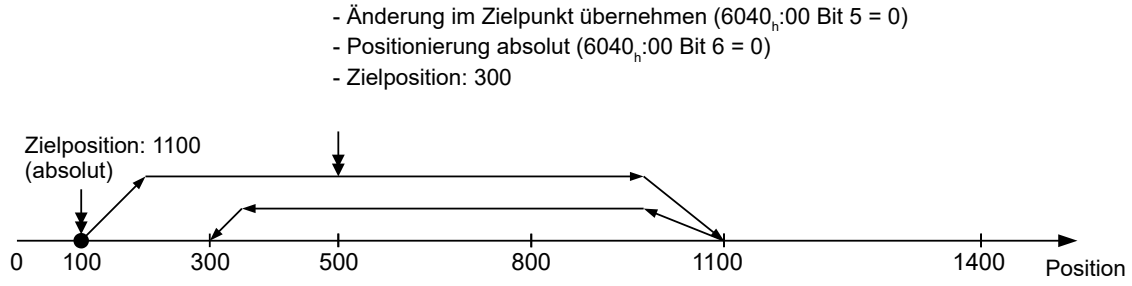


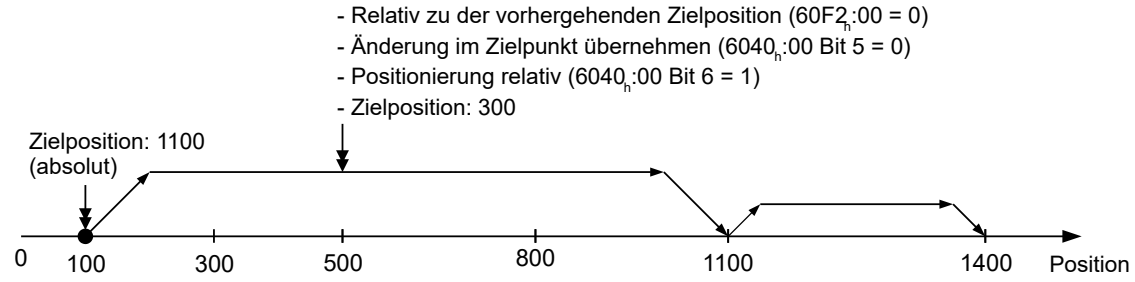
Mögliche Kombinationen von Fahrbefehlen

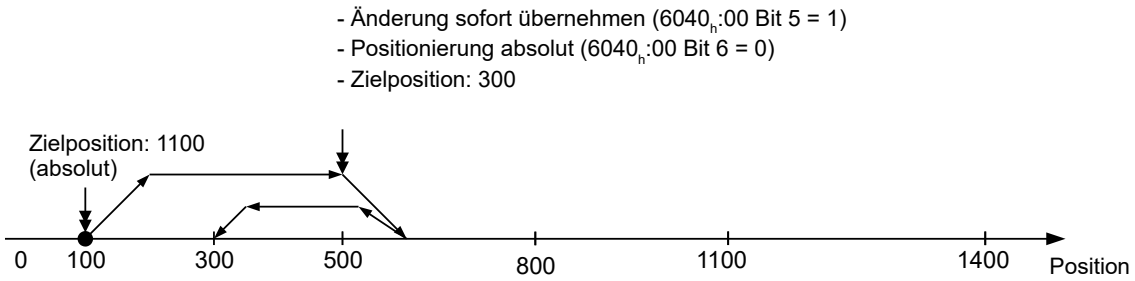
Um eine bessere Übersicht für die Fahrbefehle zu bekommen, werden in diesem Kapitel Kombinationen von Fahrbefehlen aufgelistet und dargestellt.

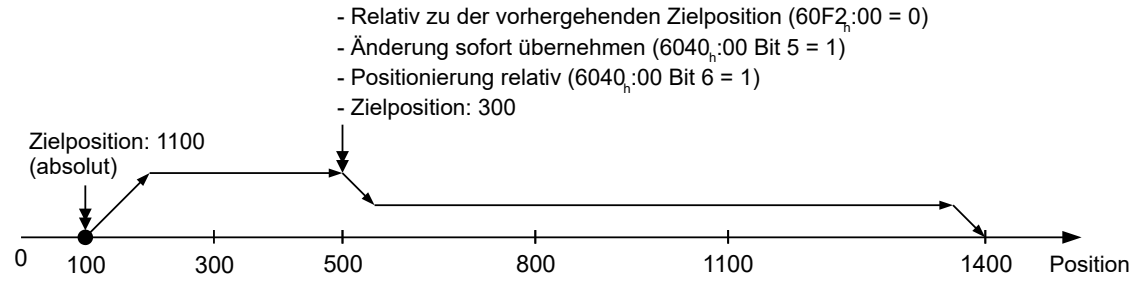
Die nachfolgenden Bilder setzen voraus:

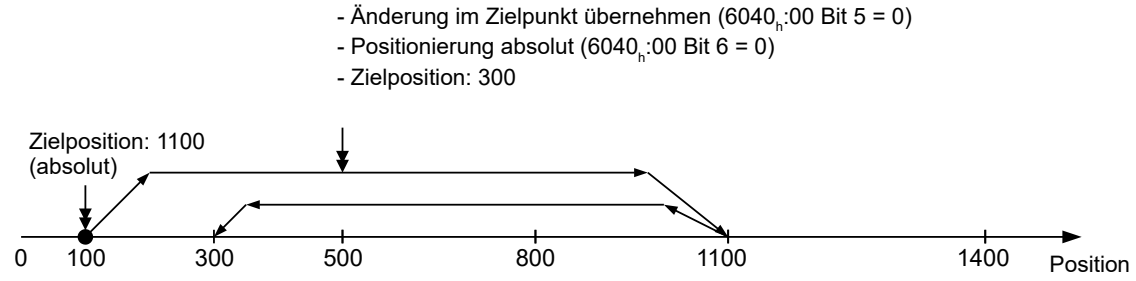
- Ein Doppelpfeil markiert einen neuen Fahrbefehl.
- Der erste Fahrbefehl am Start ist immer ein absoluter Fahrbefehl auf die Position 1100.
- Die zweite Bewegung wird mit einer niedrigeren Geschwindigkeit durchgeführt, um einen übersichtlicher dargestellten Graphen zu erhalten.

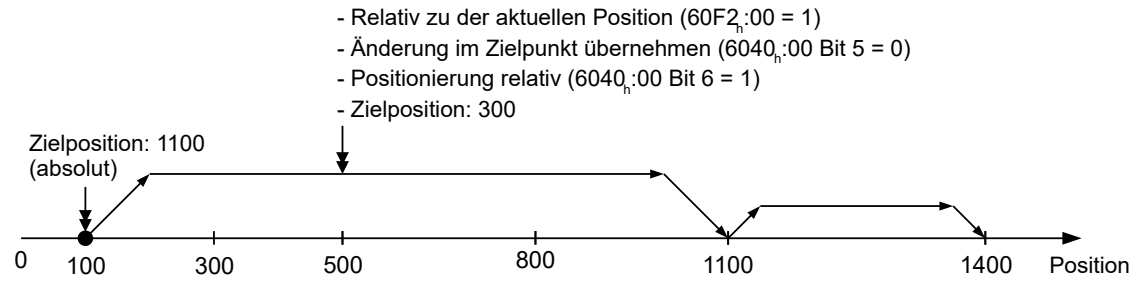


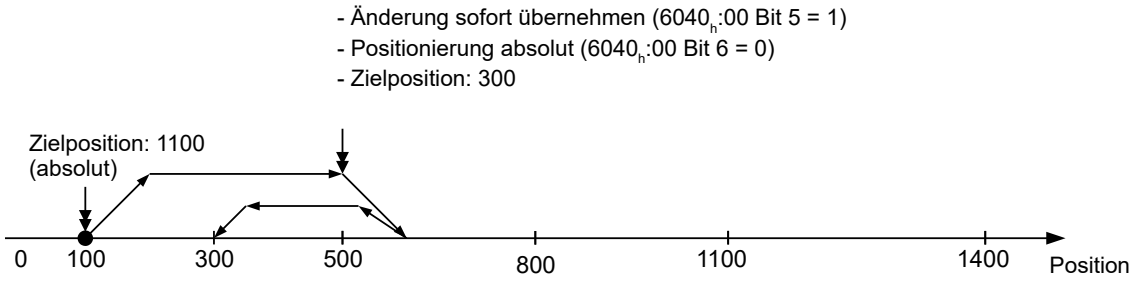






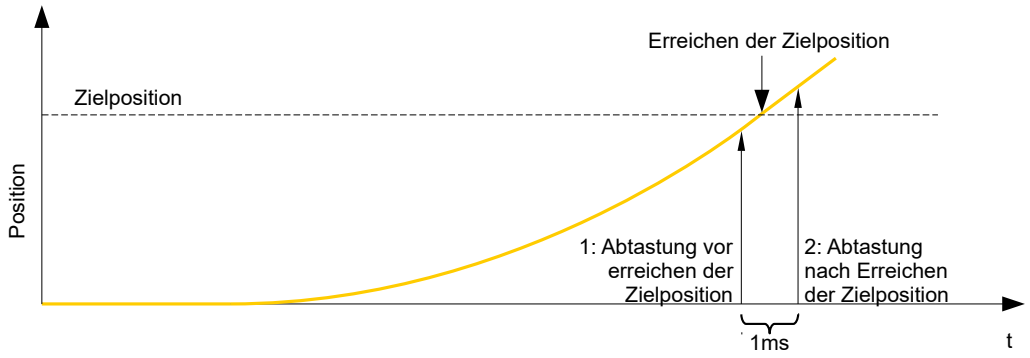






6.1.3 Genauigkeitsverlust bei Relativbewegungen

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

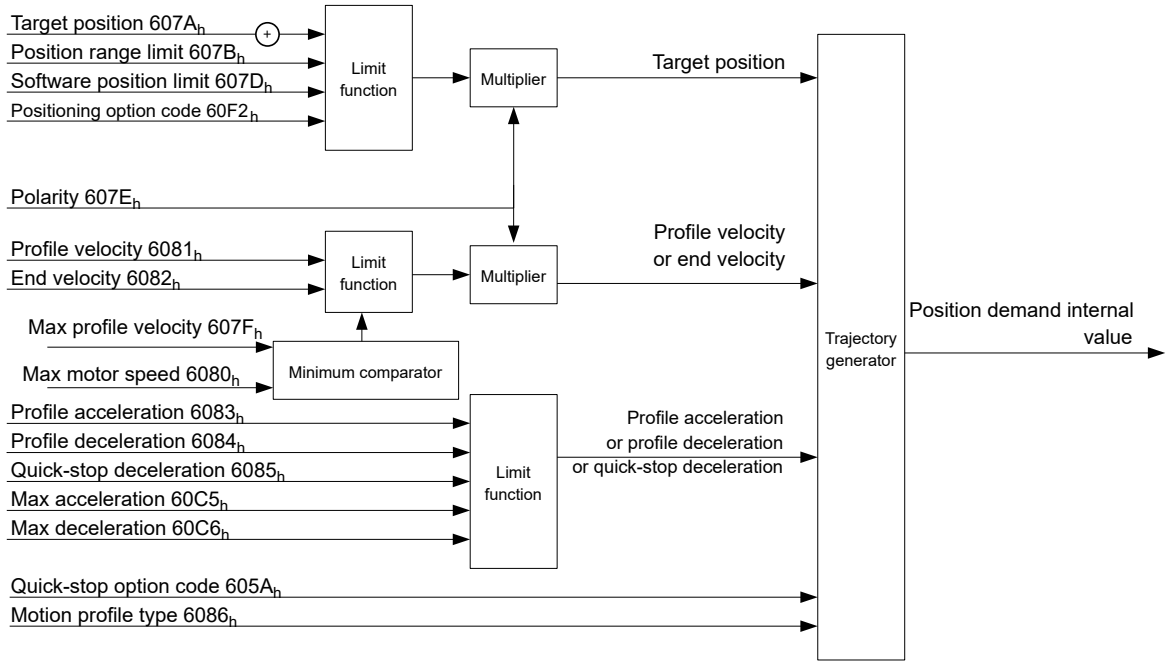
6.1.4.1 Objekteinträge

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

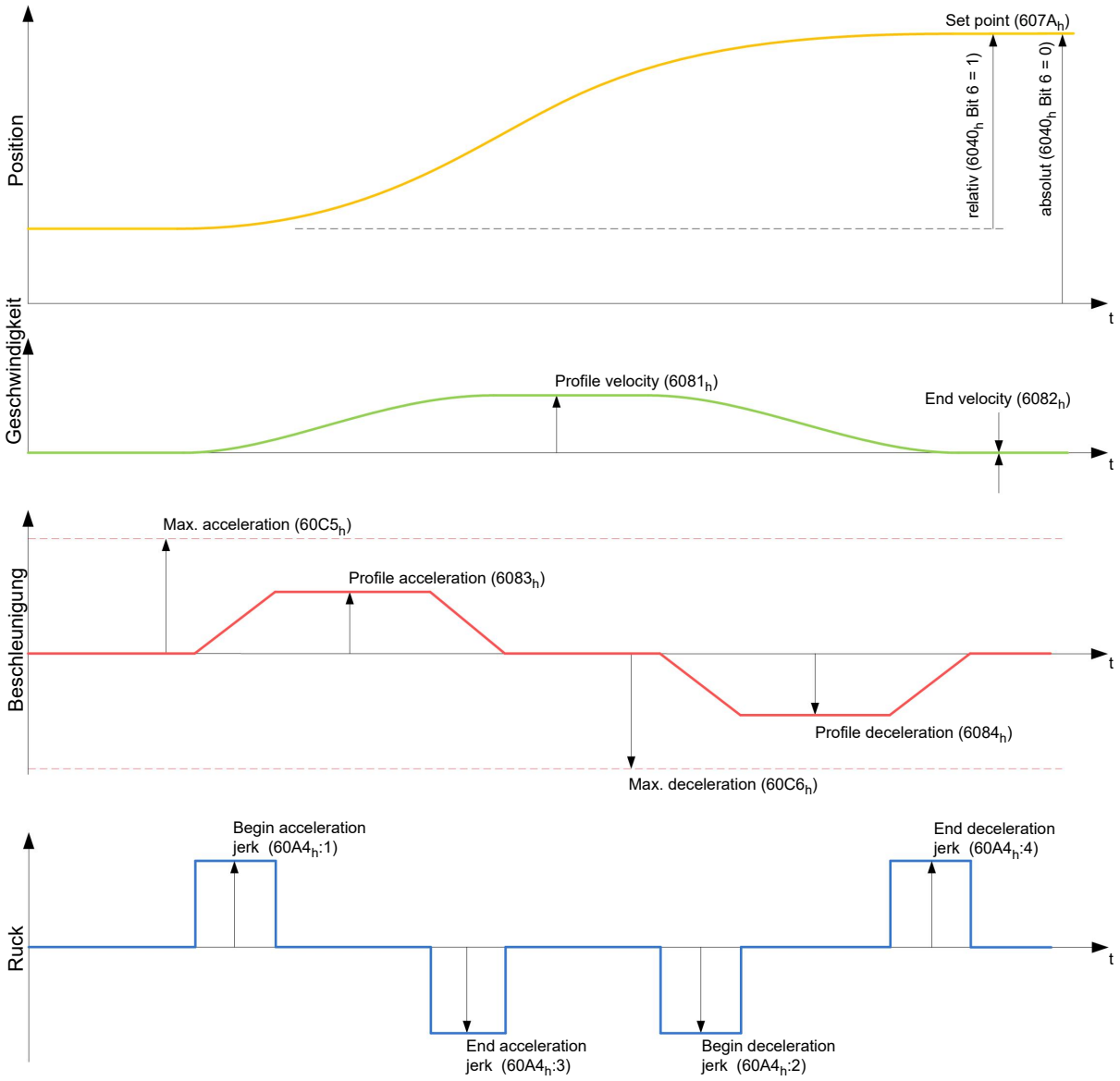
6.1.4.2 Objekte für die Positionierfahrt

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 Velocity

6.2.1 Beschreibung

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

6.2.2 Aktivierung

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

6.2.3 Controlword

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

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

6.2.4 Statusword

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

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

6.2.5 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- $604C_h$ (Dimension Factor):
  Hier wird die Einheit der Geschwindigkeitsangaben für die nachfolgenden Objekte festgelegt.
  Der Subindex 1 enthält den Nenner (Multiplikator) und der Subindex 2 den Zähler (Divisor), mit dem interne Geschwindigkeitsangaben in Umdrehungen pro Minute verrechnet werden. Wird z.B. Subindex 1 auf den Wert "60" und Subindex 2 auf den Wert "1" eingestellt, erfolgt die Geschwindigkeitsangabe in Umdrehungen pro Sekunde (60 Umdrehungen pro 1 Minute).
- 6042_h: Target Velocity.
  Hier wird die Zielgeschwindigkeit in benutzerdefinierten Einheiten eingestellt.
- 6048_h: Velocity Acceleration
  Dieses Objekt definiert die Beschleunigung. Der Subindex 1 enthält dabei die Geschwindigkeitsänderung, der Subindex 2 die zugehörige Zeit in Sekunden. Beides zusammen wird als Beschleunigung verrechnet:

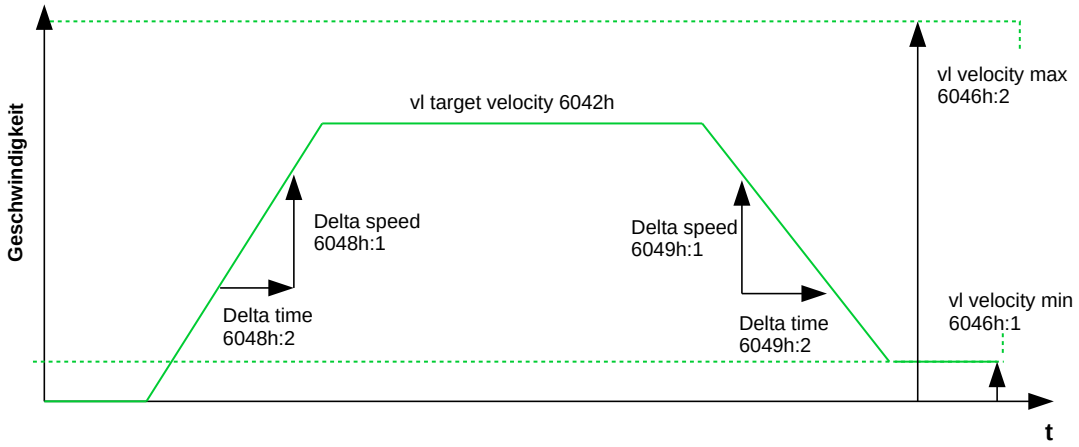
$$\text{VL velocity acceleration} = \frac{\text{Delta speed } (6048_h{:}1)}{\text{Delta time } (6048_h{:}2)}$$

- 6049_h (Velocity Deceleration):
  Dieses Objekt definiert die Verzögerung (Bremsrampe). Die Subindizes sind dabei so aufgebaut, wie im Objekt 6048_h beschrieben, die Geschwindigkeitsänderung ist mit positiven Vorzeichen anzugeben.
- 6046_h (Velocity Min Max Amount):
  In diesem Objekt werden die Limitierungen der Zielgeschwindigkeiten angegeben.
  In $6046_h{:}1_h$ wird die minimale Geschwindigkeit eingestellt. Unterschreitet die Zielgeschwindigkeit (6042_h) die Minimalgeschwindigkeit, wird der Wert auf die Minimalgeschwindigkeit $6046_h{:}1_h$ begrenzt.
  In $6046_h{:}2_h$ wird die maximale Geschwindigkeit eingestellt. Überschreitet die Zielgeschwindigkeit (6042_h) die Maximalgeschwindigkeit, wird der Wert auf die Maximalgeschwindigkeit $6046_h{:}2_h$ begrenzt.
- $604A_h$ (Velocity Quick Stop):
  Mit diesem Objekt kann die Schnellstopp-Rampe eingestellt werden. Die Subindizes 1 und 2 sind dabei identisch wie bei Objekt 6048_h beschrieben.

Folgende Objekte können zur Kontrolle der Funktion genutzt werden:

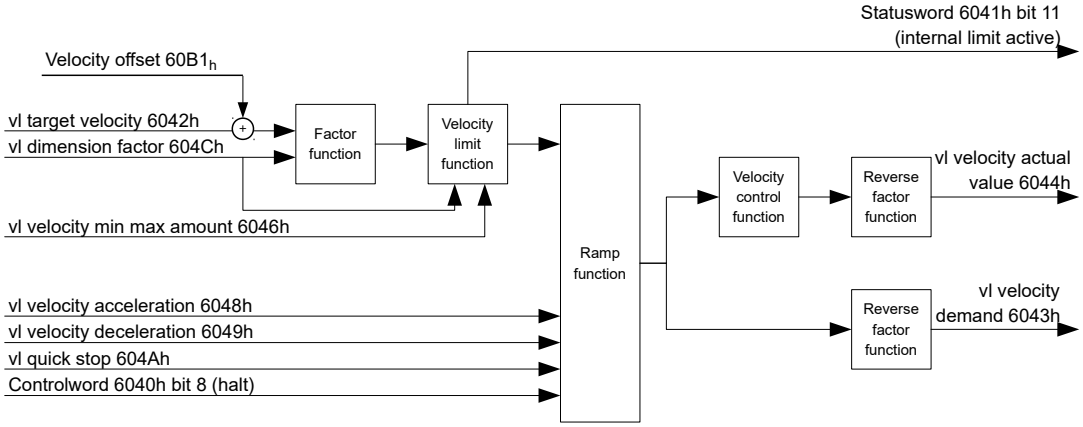
- 6043_h (Vl Velocity Demand)
- 6044_h (Vl Velocity Actual Value)

6.2.5.1 Geschwindigkeiten im Velocity Mode



6.2.5.2 Objekte für den Velocity Mode

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



6.3 Profile Velocity

6.3.1 Beschreibung

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

6.3.2 Aktivierung

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

6.3.3 Controlword

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

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

6.3.4 Statusword

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

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

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

- Bit 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).

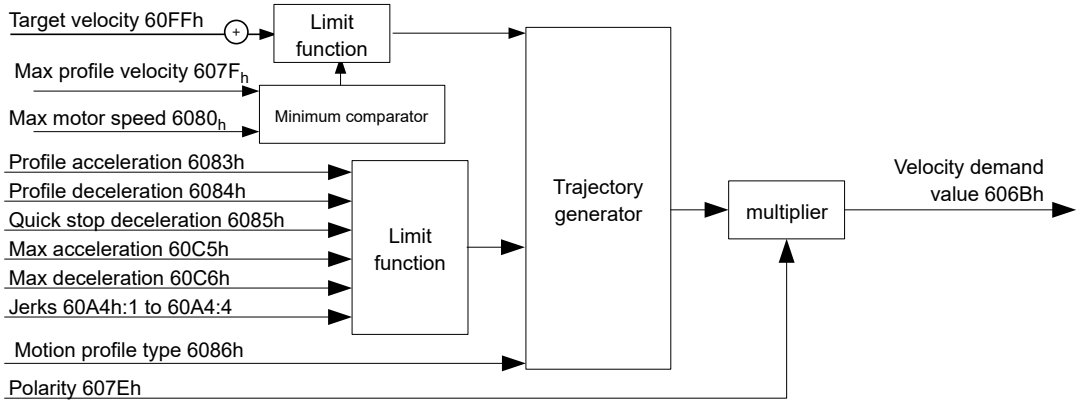
6.3.5 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

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

6.3.5.1 Objekte im Profile Velocity Mode

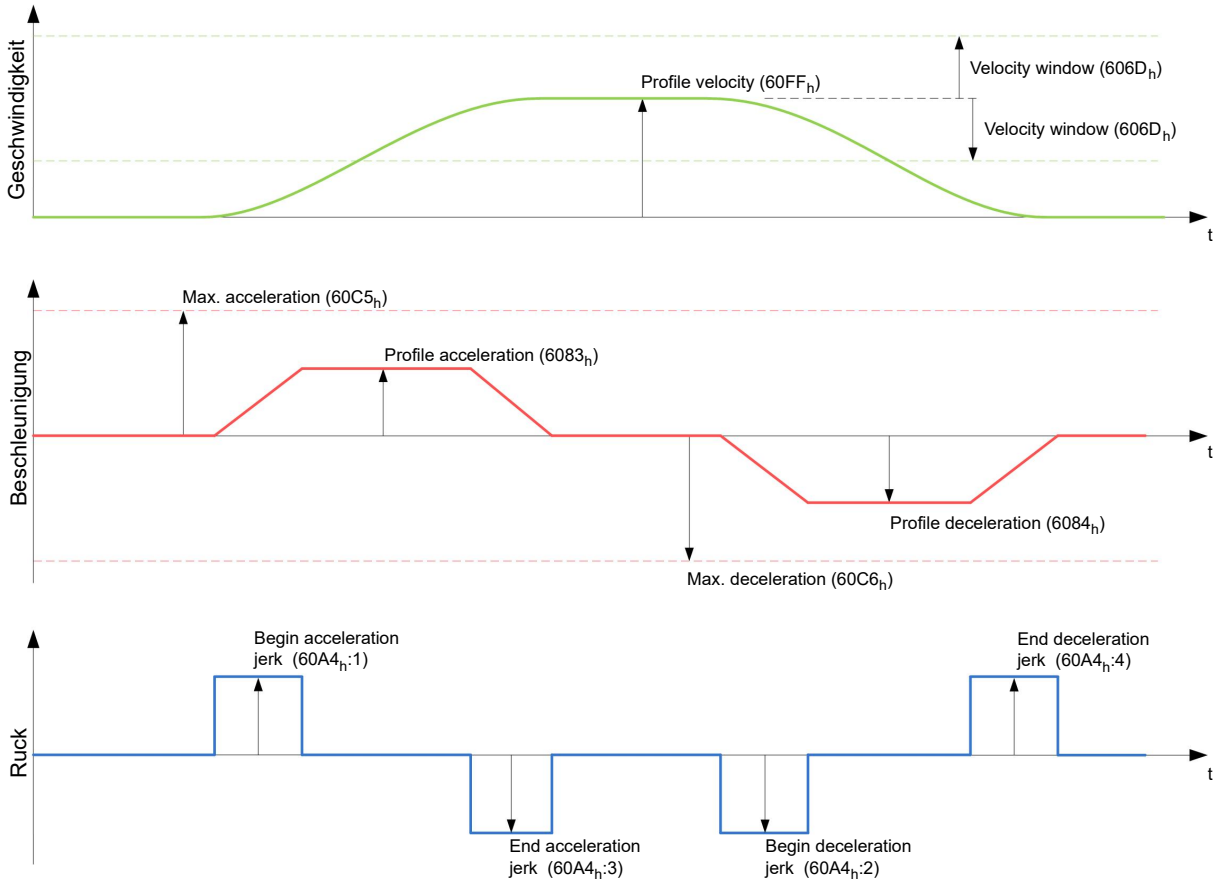


6.3.5.2 Aktivierung

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.

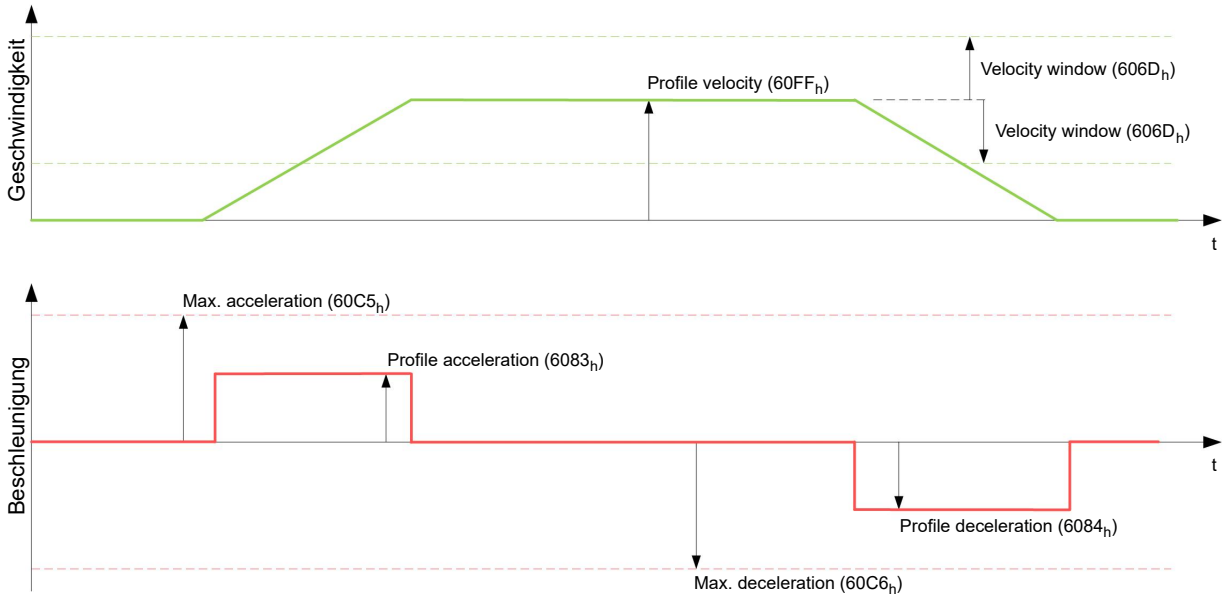
6.3.5.3 Limitierungen im ruck-limitierten Fall

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



6.3.5.4 Limitierungen im Trapez-Fall

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



6.4 Profile Torque

6.4.1 Beschreibung

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

HINWEIS

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

6.4.2 Aktivierung

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

6.4.3 Controlword

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

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

6.4.4 Statusword

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

- Bit 10 (Target Reached): Dieses Bit gibt in Kombination mit dem Bit 8 des Objekts 6040_h (Controlword) an, ob das vorgegebene Drehmoment erreicht ist (siehe nachfolgende Tabelle). Das Ziel gilt als erreicht wenn das Istdrehmoment (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.

6.4.5 Objekteinträge

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

- 6071_h (Target Torque):
  Zielvorgabe des Drehmomentes
- 6072_h (Max Torque):
  Maximales Drehmoment während der gesamten Rampe (Beschleunigen, Drehmoment halten, Abbremsen)
- 6073_h (Max Current):
  Maximalstrom. Das Minimum von 6073_h und 6072_h wird als Limit für das Drehmoment in 6071_h verwendet.
- 6074_h (Torque Demand):
  Momentaner Ausgabewert des Rampengenerators (Drehmoment) für den Regler
- 6077 (Torque Actual Value):
  Aktueller Drehmomentwert
- 6087_h (Torque Slope):
  Max. Änderung des Drehmoments pro Sekunde

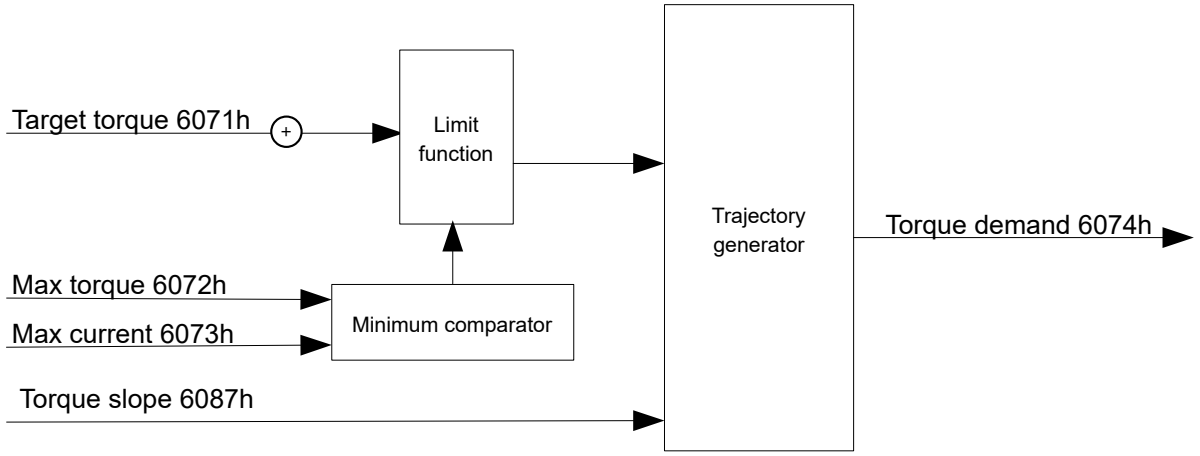
HINWEIS

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

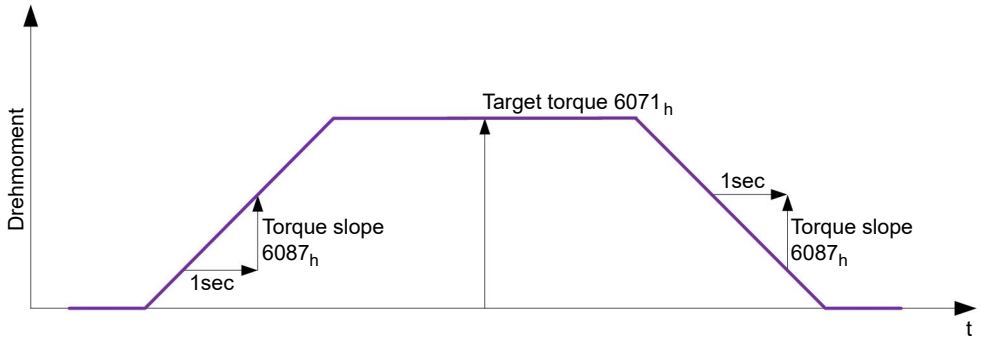
Die folgenden Objekte werden zudem für diesen Operationsmodus benötigt:

- 3202_h Bit 5 (Motor Drive Submode Select):
  Ist dieses Bit auf "0" gesetzt, wird der Antriebsregler im Drehmoment-begrenzten Velocity Mode betrieben, d.h. die maximale Geschwindigkeit kann in 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.

6.4.5.1 Objekte des Rampengenerators



6.4.5.2 Torque-Verlauf



6.5 Homing

6.5.1 Übersicht

6.5.1.1 Beschreibung

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

6.5.1.2 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.

6.5.1.3 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.

6.5.1.4 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).

6.5.1.5 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

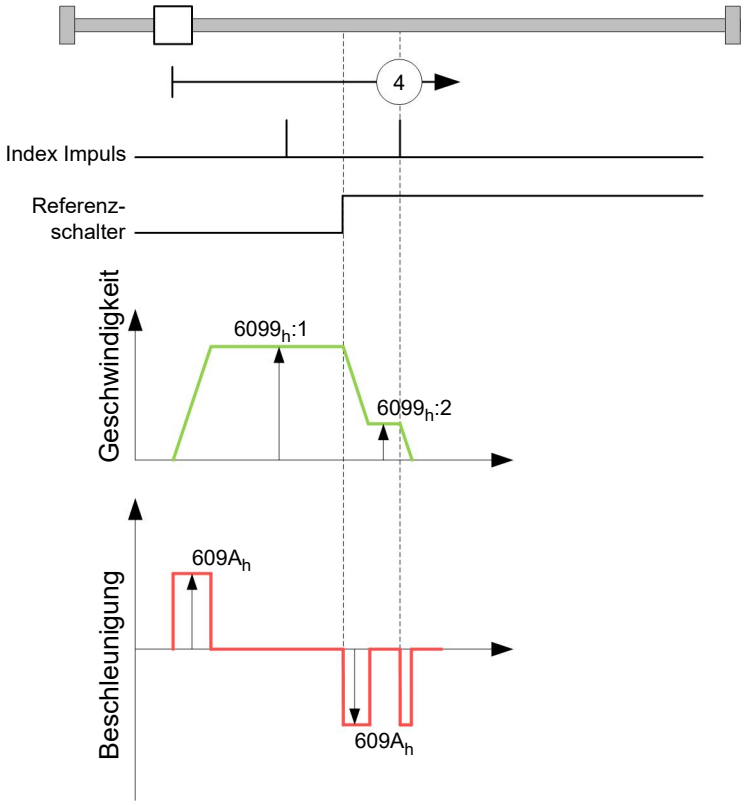
- $607C_h$ (Home Offset): Gibt die Differenz zwischen Null-Position der Steuerung und dem Referenzpunkt der Maschine in benutzerdefinierten Einheiten an.
- 6098_h (Homing Method):
  Methode, mit der referenziert werden soll (siehe "Referenzfahrt-Methode")
- 6099_h:01_h (Speed During Search For Switch):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

- $203A_h$:01_h (Minimum Current For Block Detection):
  Minimale Stromschwelle, durch deren Überschreiten, das Blockieren des Motors an einem Block erkannt werden soll.
- $203A_h$:02_h (Period Of Blocking):
  Gibt die Zeit in ms an, die der Motor nach der Blockdetektion trotzdem noch gegen den Block fahren soll.

Geschwindigkeiten der Referenzfahrt

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



6.5.2 Referenzfahrt-Methode

6.5.2.1 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.

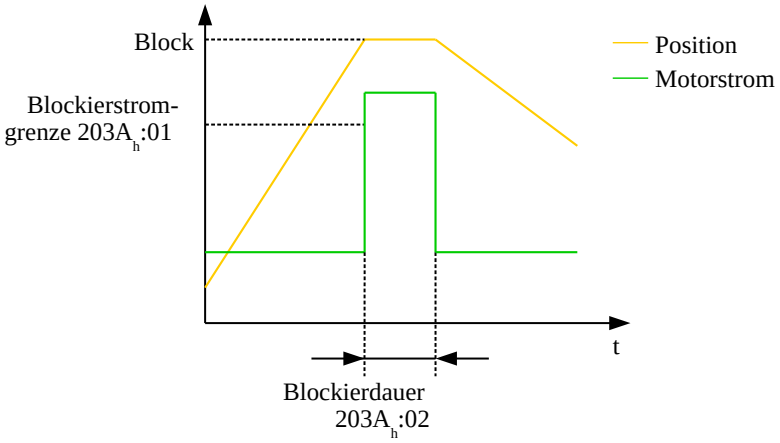
Bei Methoden, die Homing auf Block benutzen, gelten die gleichen Abbildungen wie für die Methoden mit Endschalter. Da sich außer den fehlenden Endschaltern nichts ändert, wurde auf neue Abbildungen verzichtet. Hier gilt für die Abbildungen, dass die Endschalter durch eine mechanische Blockierung ersetzt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

6.5.2.2 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 $203A_h$:01 wird die Stromhöhe definiert, ab der ein Fahren gegen den Block erkannt wird.
2. Blockierdauer: im Objekt $203A_h$:02 wird die Dauer, während der Motor gegen den Block fährt, eingestellt.



6.5.2.3 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.

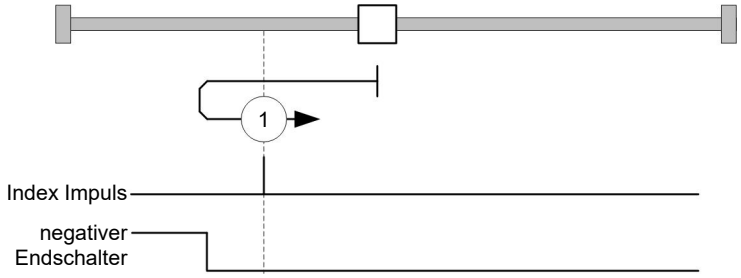
Folgende Methoden können für Homing auf Block benutzt werden:

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

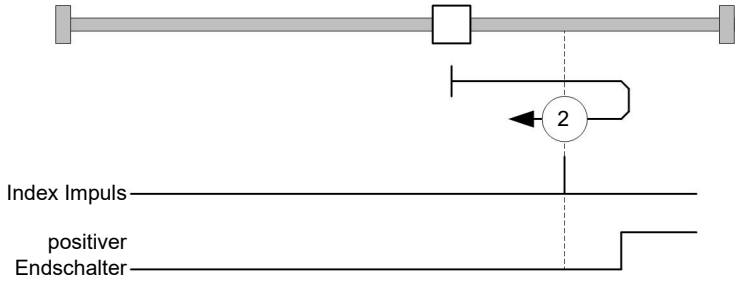
6.5.2.4 Methoden 1 und 2

Referenzieren auf Endschalter und Index-Impuls.

Methode 1 referenziert auf negativen Endschalter und Index-Impuls:



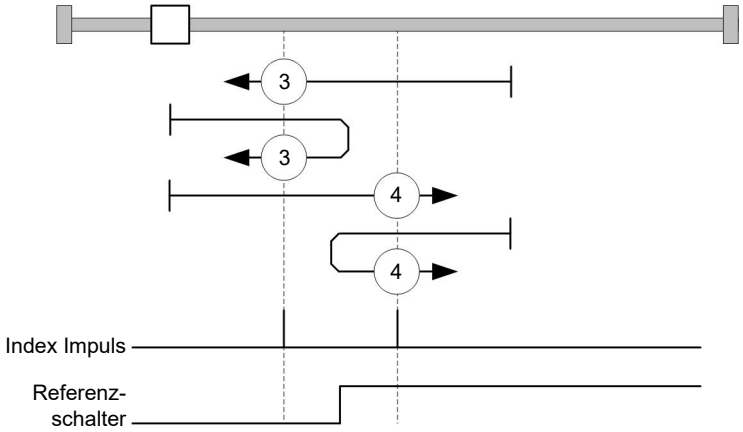
Methode 2 referenziert auf positiven Endschalter und Index-Impuls:



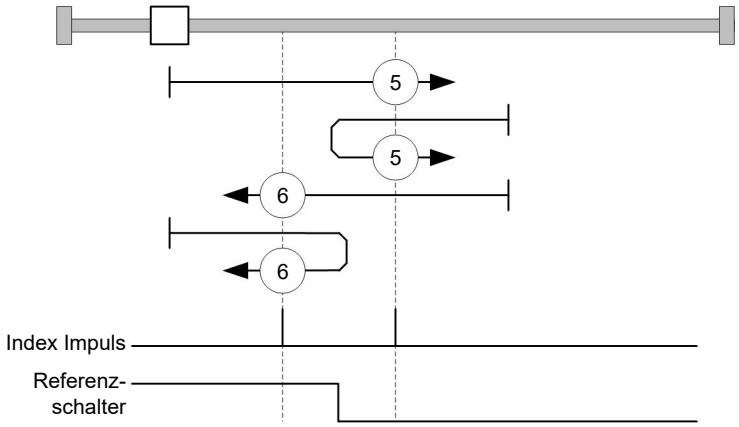
6.5.2.5 Methoden 3 bis 6

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:

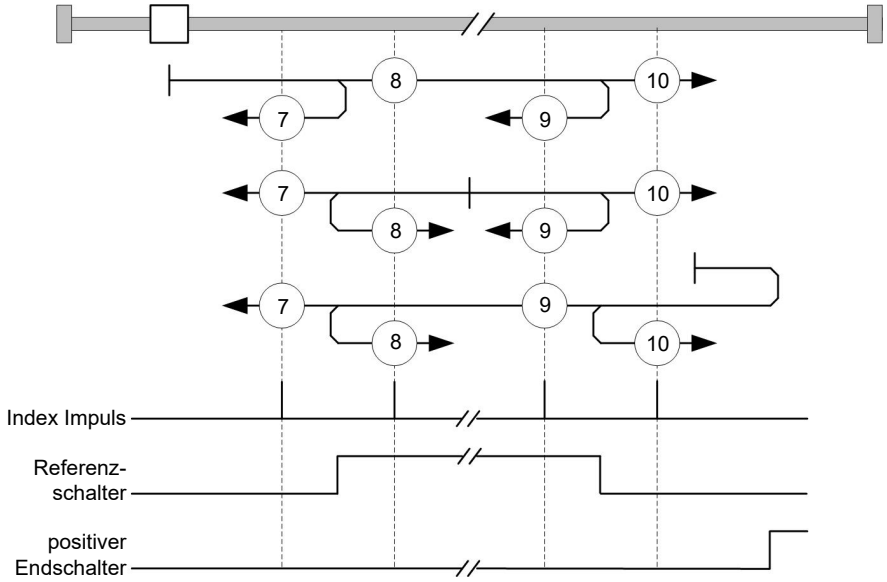


6.5.2.6 Methoden 7 bis 14

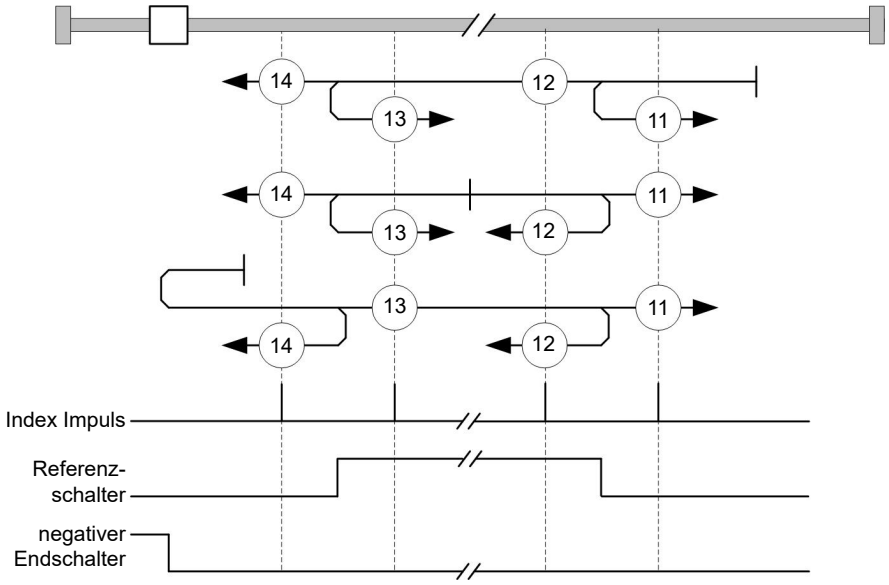
Referenzieren auf Referenzschalter und Index-Impuls (mit Endschaltern).

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:



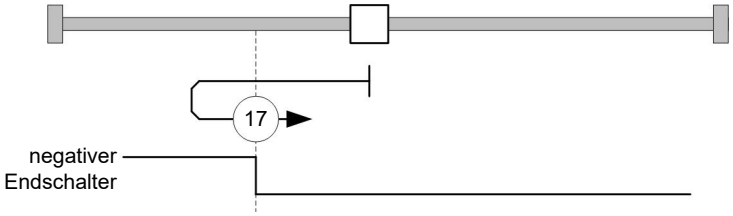
Die Methoden 11 bis 14 berücksichtigen den negativen Endschalter:



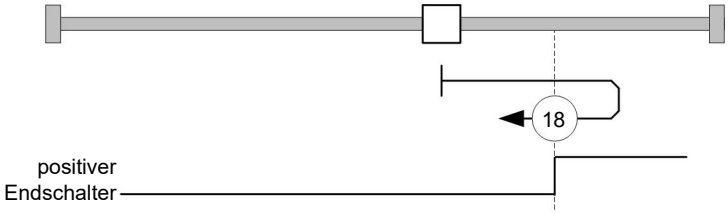
6.5.2.7 Methoden 17 und 18

Referenzieren auf den Endschalter ohne den Index-Impuls.

Methode 17 referenziert auf den negativen Endschalter:



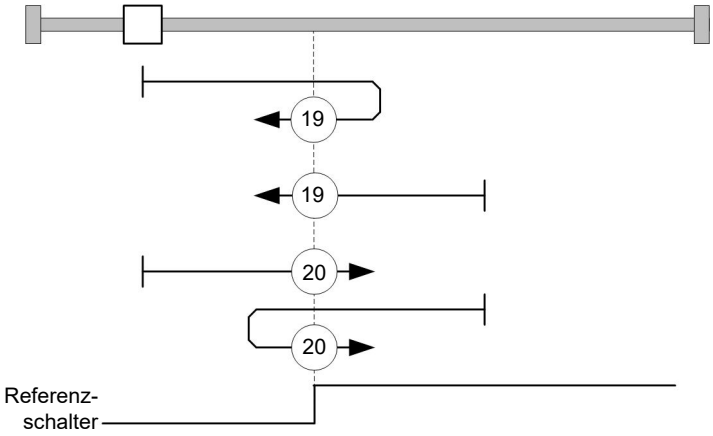
Methode 18 referenziert auf den positiven Endschalter:



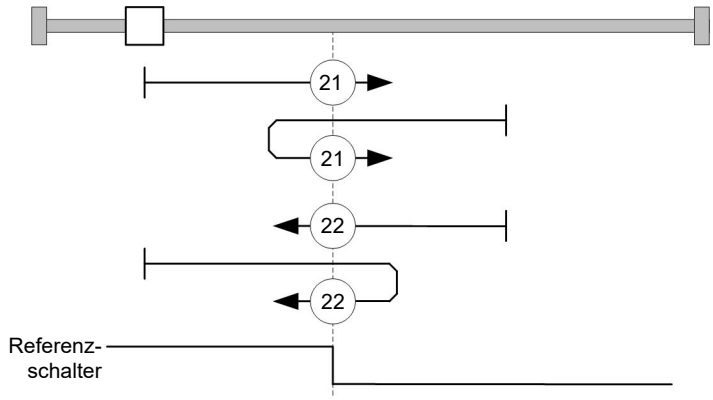
6.5.2.8 Methoden 19 bis 22

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:

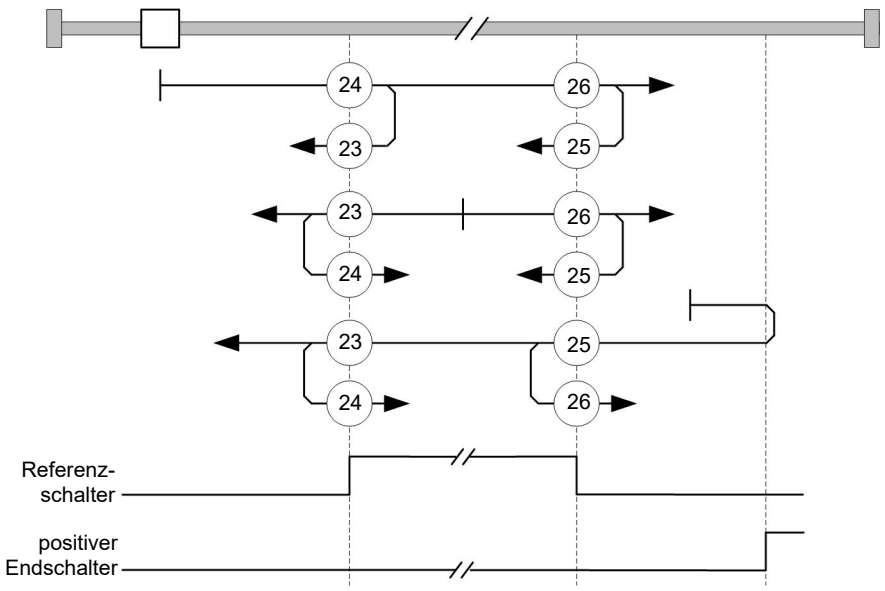


6.5.2.9 Methoden 23 bis 30

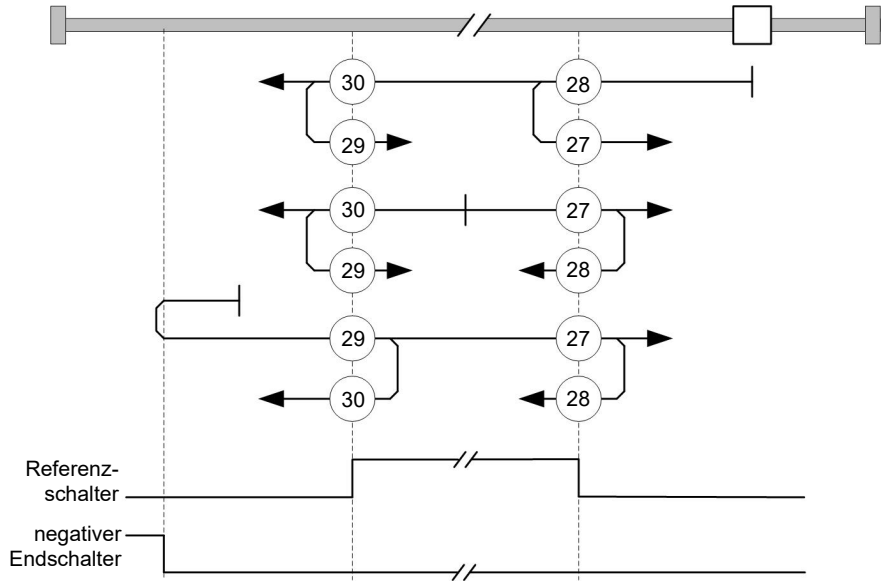
Referenzieren auf Referenzschalter ohne den Index-Impuls (mit Endschaltern).

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:



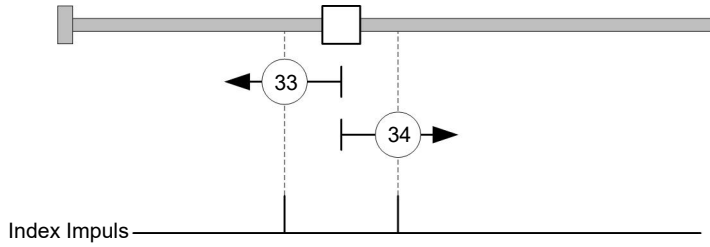
Die Methoden 27 bis 30 berücksichtigen den negativen Referenzschalter:



6.5.2.10 Methoden 33 und 34

Referenzieren auf den nächsten Index-Impuls.

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



6.5.2.11 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.

6.6 Auto-Setup

6.6.1 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. Bei den Plug & Drive Motoren ist es nicht notwendig ein *Auto-Setup* auszuführen, da dieses bereits werksseitig durchgeführt wurde. Für Details siehe entsprechenden Abschnitt im Kapitel Inbetriebnahme.

6.6.2 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).

6.6.3 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.

6.6.4 Statusword

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

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

7 Spezielle Funktionen

7.1 Digitale Ein- und Ausgänge

Diese Steuerung verfügt über digitale Ein- und Ausgänge.

7.1.1 Bitzuordnung

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

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

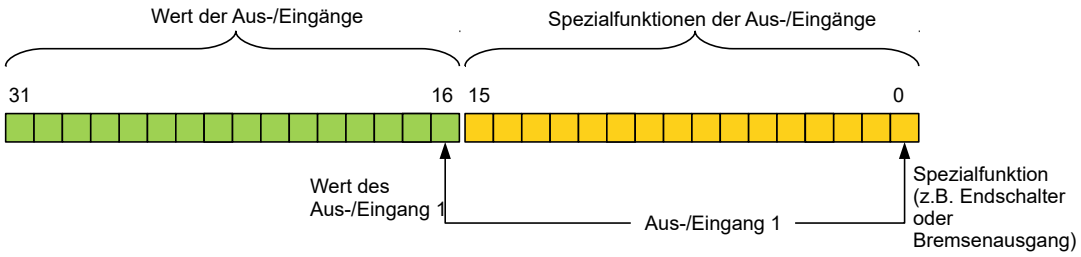
Beispiel

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

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

In der nachfolgenden Zeichnung ist diese Zuordnung graphisch dargestellt.

Bits eines beliebigen Objektes zur Steuerung eines Aus-/Eingangs



7.1.2 Digitale Eingänge

7.1.2.1 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.

Folgende Eingänge stehen zur Verfügung:

Eingang	Sonderfunktion	Schaltschwelle umschaltbar	Differenziell / single-ended
1	Negativer Endschalter / Takteingang im Takt-Richtungs-Modus	ja, 5 V oder 24 V (siehe 3240_h:06_h)	Die Eingänge sind nur gemeinsam umschaltbar. (siehe 3240_h:07_h)
2	Positiver Endschalter / Richtungseingang im Takt-Richtungs-Modus	ja, 5 V oder 24 V (siehe 3240_h:06_h)	
3	Referenzschalter	ja, 5 V oder 24 V (siehe 3240_h:06_h)	
4	keine	ja, 5 V oder 24 V (siehe 3240_h:06_h)	
5	keine	ja, 5 V oder 24 V (siehe 3240_h:06_h)	
6	keine	ja, 5 V oder 24 V (siehe 3240_h:06_h)	

Wenn Sie das 3240_h:07_h auf den Wert "1" setzen, stehen Ihnen, anstatt sechs single-ended, drei differentielle Eingänge zur Verfügung:

Pin	Funktion
2	- Eingang 1
3	Eingang 1
4	- Eingang 2
5	Eingang 2
6	- Eingang 3
7	Eingang 3

7.1.2.2 Objekteinträge

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

- 3240_h:01_h (Special Function Enable): Dieses Bit erlaubt Sonderfunktionen eines Eingangs aus- (Wert "0") oder einzuschalten (Wert "1"). Soll Eingang 1 z. B. nicht als negativer Endschalter verwendet werden, so muss die Sonderfunktion abgeschaltet werden, damit nicht fälschlicherweise auf den Signalgeber reagiert wird. Auf die Bits 16 bis 31 hat das Objekt keine Auswirkungen.
  Die Firmware wertet folgende Bits aus:
  - Bit 0: Negativer Endschalter (siehe Begrenzung des Bewegungsbereichs)
  - Bit 1: Positiver Endschalter (siehe Begrenzung des Bewegungsbereichs)
  - Bit 2: Referenzschalter (siehe Homing)
  - Bit 3: Interlock (siehe Interlock-Funktion)

  Sollen z. B. zwei Endschalter und ein Referenzschalter verwendet werden, müssen Bits 0-2 in 3240_h:01_h auf "1" gesetzt werden.
- 3240_h:02_h (Function Inverted): Dieser Subindex wechselt von Schließer-Logik (ein logischer High-Pegel am Eingang ergibt den Wert "1" im Objekt $60FD_h$) auf Öffner-Logik (der logische High-Pegel am Eingang ergibt den Wert "0").
  Das gilt für die Sonderfunktionen (außer den Takt- und Richtungseingängen) und für die normalen Eingänge. Hat das Bit den Wert "0" gilt Schließer-Logik, entsprechend bei dem Wert "1" die Öffner-Logik. Bit 0 wechselt die Logik des Eingangs 1, Bit 1 die Logik des Eingangs 2 usw.
- 3240_h:03_h (Force Enable): Dieser Subindex schaltet die Softwaresimulation von Eingangswerten ein, wenn das entsprechende Bit auf "1" gesetzt ist.

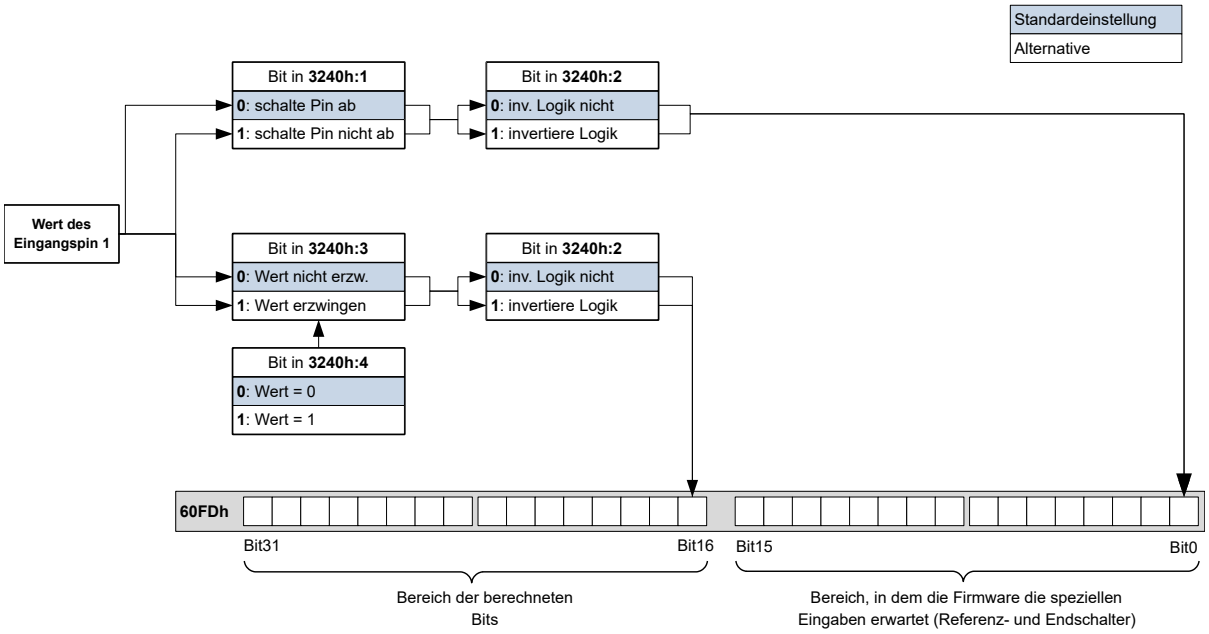
Dann werden nicht mehr die tatsächlichen, sondern die in Objekt 3240_h:04_h eingestellten Werte für den jeweiligen Eingang verwendet. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw.

- 3240_h:04_h (Force Value): Dieses Bit gibt den Wert vor, der als Eingangswert eingelesen werden soll, wenn das gleiche Bit im Objekt 3240_h:03_h gesetzt wurde.
- 3240_h:05_h (Raw Value): Dieses Objekt beinhaltet den unmodifizierten Eingabewert.
- 3240_h:07_h (Differential Select): Dieser Subindex schaltet bei den Eingängen zwischen "single-ended Eingang" (Wert "0" in dem Subindex) zu "Differentieller Eingang" (Wert "1" in dem Subindex) für alle Eingänge auf einmal um.
- $60FD_h$ (Digital Inputs): Dieses Objekt enthält eine Zusammenfassung der Eingänge und der Spezialfunktionen.

7.1.2.3 Verrechnung der Eingänge

Verrechnung des Eingangssignals am Beispiel von Eingang 1:

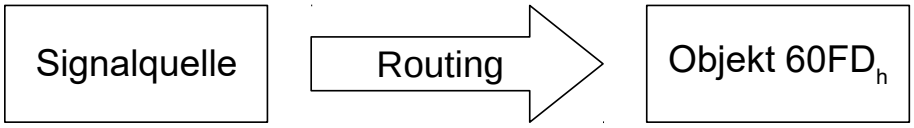
Der Wert an Bit 0 des Objekts $60FD_h$ wird von der Firmware als negativer Endschalter interpretiert, das Ergebnis der vollständigen Verrechnung wird in Bit 16 abgelegt.



7.1.2.4 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.



Aktivierung

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

HINWEIS

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

HINWEIS

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

Routing

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

Nummer		
dec	**hex**	**Signalquelle**
00	00	Signal ist immer 0
01	01	Physikalischer Eingang 1
02	02	Physikalischer Eingang 2
03	03	Physikalischer Eingang 3
04	04	Physikalischer Eingang 4
05	05	Physikalischer Eingang 5
06	06	Physikalischer Eingang 6
07	07	Physikalischer Eingang 7
08	08	Physikalischer Eingang 8
09	09	Physikalischer Eingang 9
10	0A	Physikalischer Eingang 10
11	0B	Physikalischer Eingang 11
12	0C	Physikalischer Eingang 12
13	0D	Physikalischer Eingang 13
14	0E	Physikalischer Eingang 14
15	0F	Physikalischer Eingang 15
16	10	Physikalischer Eingang 16
68	44	Encoder Eingang "A"
69	45	Encoder Eingang "B"
70	46	Encoder Eingang "Index"
71	47	USB Power Signal
100	64	STO_A
101	65	STO_B

Die nachfolgende Tabelle beschreibt die invertierten Signale der vorherigen Tabelle.

Nummer		
dec	**hex**	**Signalquelle**
128	80	Signal ist immer 1
129	81	Invertierter Physikalischer Eingang 1
130	82	Invertierter Physikalischer Eingang 2
131	83	Invertierter Physikalischer Eingang 3

Nummer		
dec	hex	Signalquelle
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
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_A
229	E5	Invertierter STO_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.

7.1.2.5 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.

Um die Interlock-Funktion zu aktivieren, müssen Sie die Sonderfunktion einschalten, indem Sie das Bit 3 in 3240:01_h auf "1" setzen.

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 das *Input Routing* zu aktivieren, setzen Sie das 3240_h:08_h auf "1".
2. Um den Eingang 4 auf Bit 3 zu routen, setzen Sie das 3242_h:04_h auf "4".

7.1.3 Digitale Ausgänge

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

7.1.3.2 Beschaltung

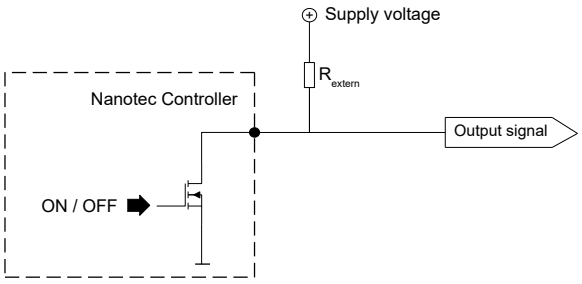
HINWEIS

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

Die Outputs sind als "Open Drain" realisiert. Demzufolge ist immer eine externe Spannungsversorgung nötig.

Beispiel

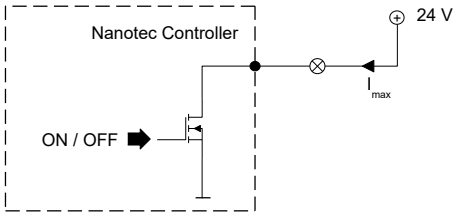
Es soll das digitale Ausgangssignal weiter verwendet werden. Dazu ist eine Beschaltung wie im nachfolgenden Bild zu realisieren.



Bei einer Versorgungsspannung von +24 V wird ein Widerstandswert R_{extern} von 10 kΩ empfohlen.

Beispiel

Es soll ein einfacher Verbraucher mit dem digitalen Ausgang gestellt werden.



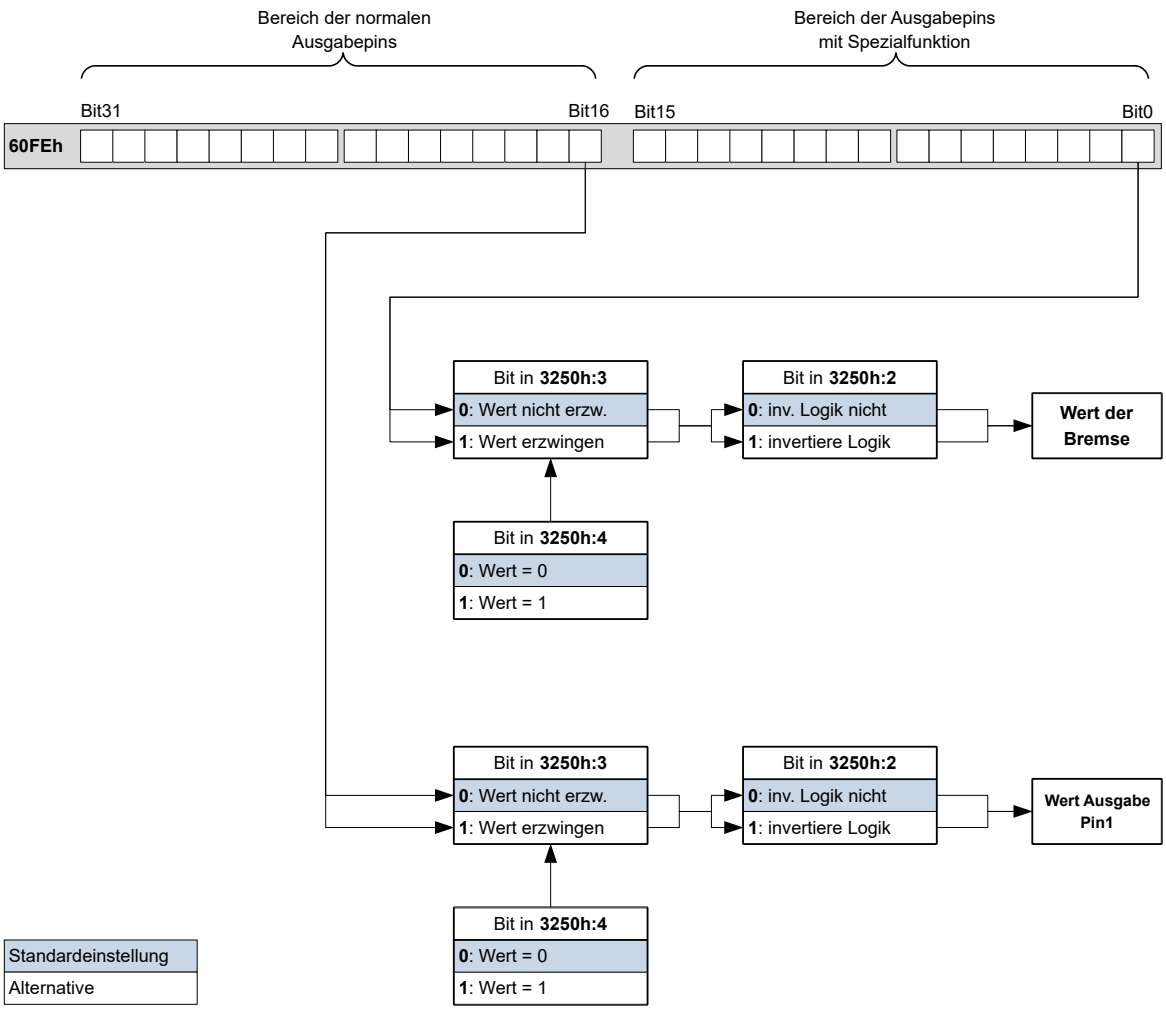
7.1.3.3 Objekteinträge

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

- 3250_h:01_h: Keine Funktion.
- 3250_h:02_h: Damit lässt sich die Logik von *Schließer* auf *Öffner* umstellen. Als *Schließer* konfiguriert, gibt der Ausgang einen logischen High-Pegel ab, sollte das Bit "1" sein. Bei der *Öffner* -Konfiguration wird bei einer "1" im Objekt $60FE_h$ entsprechend ein logischer Low-Pegel ausgegeben.
- 3250_h:03_h: Ist hier ein Bit gesetzt, wird der Ausgang manuell gesteuert. Der Wert für den Ausgang steht dann in Objekt 3250_h:4_h, dies ist auch für den Bremsenausgang möglich.
- 3250_h:04_h: Die Bits in diesem Objekt geben den Ausgabewert vor, welcher am Ausgang angelegt sein soll, wenn die manuelle Steuerung des Ausgangs über das Objekt 3250_h:03_h aktiviert ist.
- 3250_h:05_h: In diesen Subindex wird die an die Ausgänge gelegte Bitkombination abgelegt.
- 3250_h:08_h: Zum Aktivieren des Output Routing.
- 3250_h:09_h: Zum Ein-/Ausschalten der Ansteuerung der Betriebs-LED. Ist das Bit 0 auf "1" gesetzt, wird die grüne LED angesteuert (blinkt im normalen Betrieb). Ist das Bit 1 auf "1" gesetzt, wird die rote LED angesteuert (blinkt im Fehlerfall). Wird das Bit auf "0" gesetzt, bleibt die jeweilige LED aus.
-

7.1.3.4 Verrechnung der Ausgänge

Beispiel für die Verrechnung der Bits für die Ausgänge:

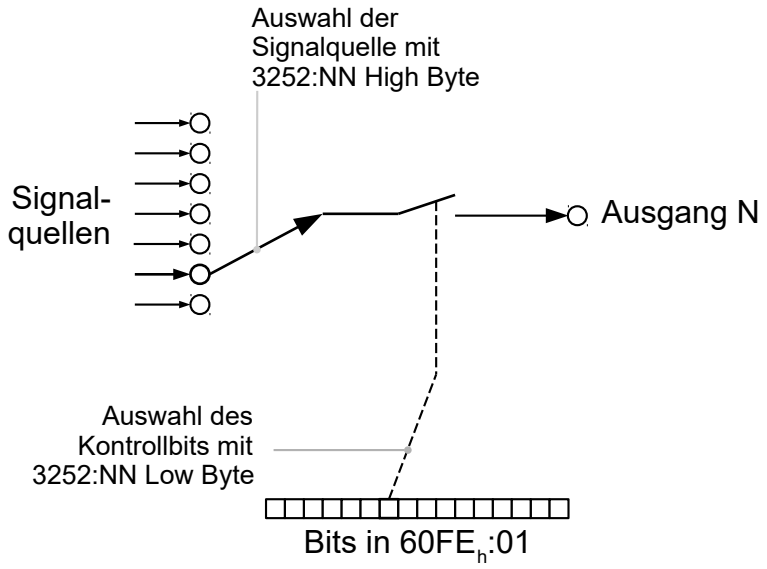


7.1.3.5 Output Routing

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).



Aktivierung

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

HINWEIS

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

Routing

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

Subindex 3252_h	Output Pin
01_h	Konfiguration des PWM-Ausgangs (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.

TIPP

Um das Routing zu deaktivieren, tragen Sie den Wert $FFFF_h$ 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
$09XX_h$	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

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

7.2 Analoge Eingänge

Die Steuerung besitzt einen Analogeingang mit einer Auflösung von 10 Bit. Er befindet sich am Pin 8 von X2. Sie können den Analogeingang als Strom- oder Spannungseingang konfigurieren mit dem Objekt 3221_h.

Den Analogwert können Sie in einem NanoJ-Programm auslesen und beliebig verwenden, um z. B. die Zielgeschwindigkeit vorzugeben. Die Steuerung bietet auch vorprogrammierte Betriebsmodi (Analog-Drehzahl / Joystick), die den Analogeingang auswerten (siehe Spezielle Fahrmodi).

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.
- 3221_h (Analogue Inputs Control):
  Mit diesem Objekt schalten Sie einen Analogeingang von Spannungs- auf Strommessung um.
- 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 *ADC Digits* angegeben.

Beispiel

Der Analogeingang 1 hat einen Messbereich von 0 V…+10 V. Am Analogeingang liegt eine Spannung von 0 V…+10 V an, der bei der Auflösung von 10 Bit dem Wertebereich 0…1023 *ADC Digits* entspricht.

Um den Analogwert in der physikalischen Einheit Millivolt anzeigen zu lassen, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "10000" (entspricht dem gesamten Messbereich in Millivolt) in 3322_h:01_h (Analogue Input Factor Numerator).
2. Schreiben Sie den Wert "1023" (entspricht der Auflösung in Digits) in 3323_h:01_h (Analogue Input Factor Denominator).

Bei der maximalen Spannung von 10 V, lesen Sie nun im Objekt 3320_h (Read Analogue Input) den Wert "10000" aus:

1023 Digits * 10000 mV / 1023 = 10000 mV

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:

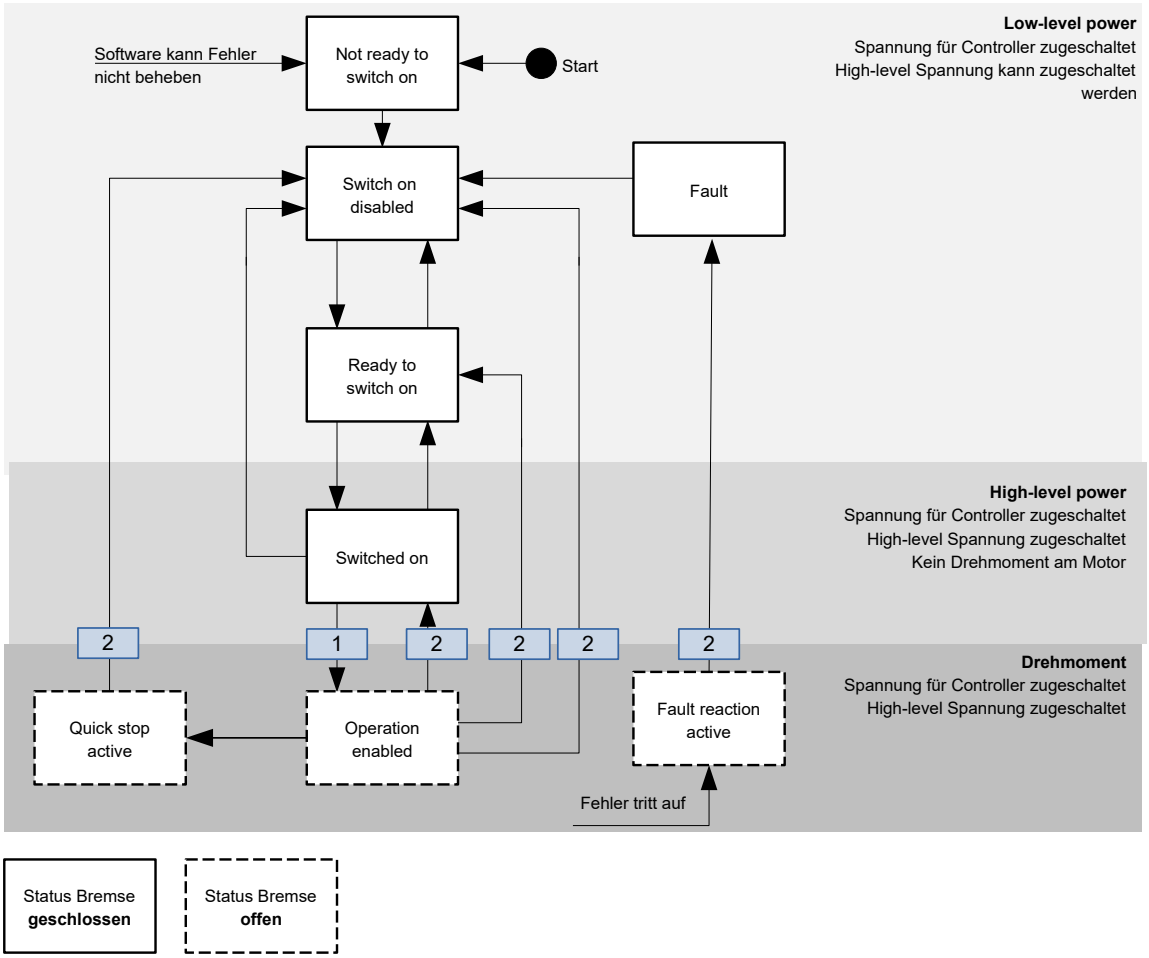
- Automatisch: Bit 2 des Objekts 3202_h auf "1" setzen aktiviert die Bremsensteuerung.
- Manuell: Bit 2 des Objekts 3202_h auf "0" setzen deaktiviert die Bremsensteuerung, die Bremse lässt sich jetzt mit dem Bit 0 im Objekt $60FE_h$:01_h kontrollieren.

7.3.2.1 Anschluss

Der Bremsenausgang befindet sich am X5 (siehe X5 — externe Haltebremse)

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.

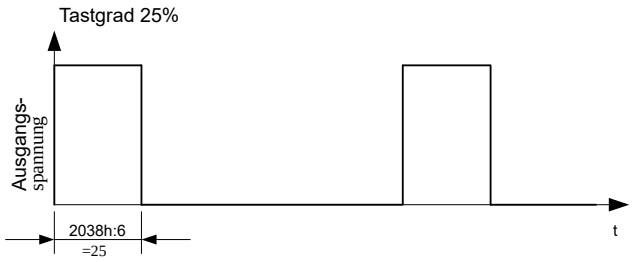
7.3.4.1 Frequenz

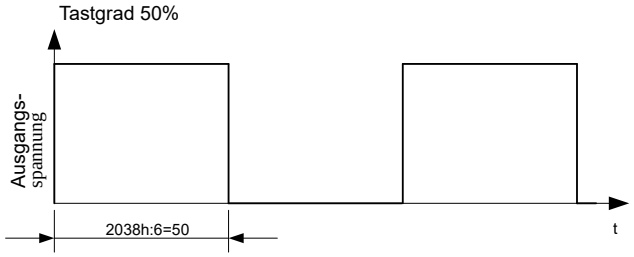
Die Frequenz der Bremsen-PWM kann im Objekt 2038_h:5_h eingestellt werden. Die Einheit ist Hertz, ein Wert kleiner 50 oder größer 20000 ist nicht möglich.

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 2 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 I^2t Motor-Überlastungsschutz

7.4.1 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).

7.4.2 Objekteinträge

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

- 2031_h: Max Motor Current - Gibt den maximal zulässigen Motorstrom in mA an.
- $203B_h$:1_h Motor Rated Current - Gibt den Nennstrom in mA an.
- 6073_h Max Current - Gibt den Maximalstrom in Promille des eingestellten Nennstroms an.
- $203B_h$:2_h Maximum Duration Of Peak Current - Gibt die maximale Dauer des Maximalstroms in ms an.

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

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

- Wert = "0": I^2t deaktiviert
- Wert = "1": I^2t aktiviert

7.4.3 Aktivierung

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

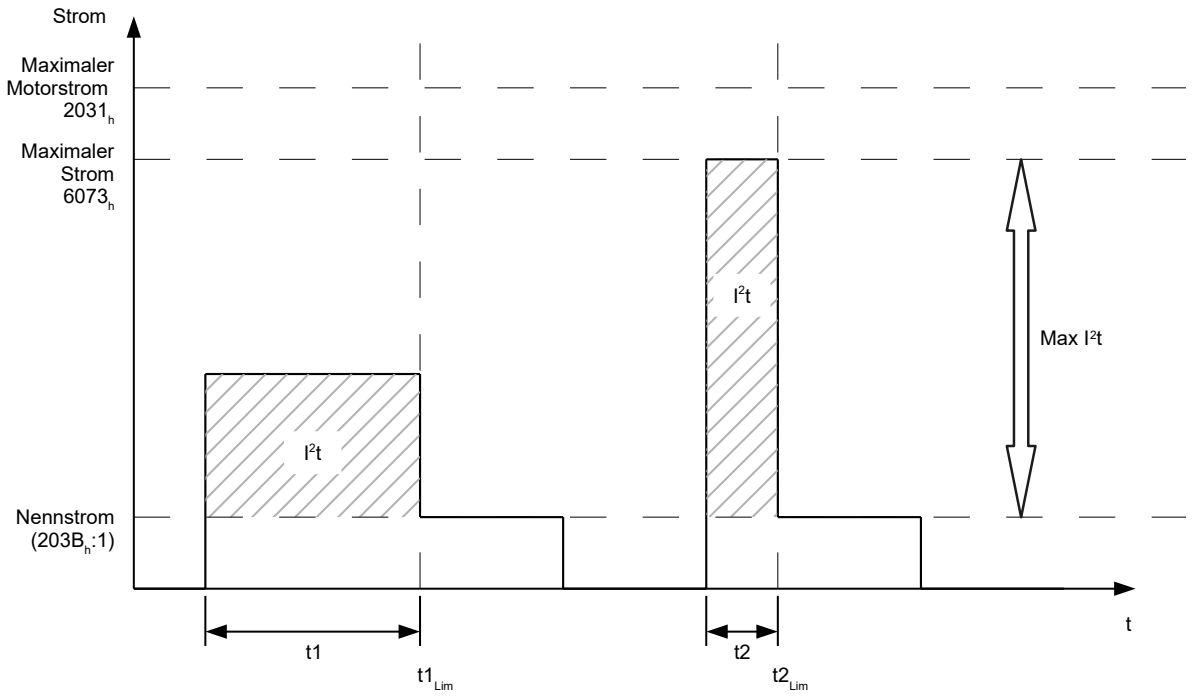
Zum Aktivieren des Modus müssen Sie die vier oben genannten Objekteinträge (2031_h, 6073_h, $203B_h$:1_h, $203B_h$:2_h) sinnvoll beschreiben. Das bedeutet, dass der Maximalstrom größer als der Nennstrom sein muss und ein Zeitwert für die maximale Dauer des Maximalstroms eingetragen sein muss. Wenn diese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

7.4.4 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 (2031_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.

7.5 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.

HINWEIS

Objekte können dauerhaft über die Konfigurationsdatei `cfg.txt` gespeichert werden. Der Speicher-Mechanismus, der in diesem Kapitel beschrieben wird, ist bei dieser Steuerung nur mit einem *NanoJ-Programm* oder mit der Software *Plug & Drive Studio* nutzbar.

7.5.1 Allgemeines

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

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

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

- Kommunikation: Parameter mit Bezug auf externe Schnittstellen, wie PDO-Konfiguration etc.
- Applikation: Parameter mit Bezug auf Betriebsmodi.
- Benutzer: Parameter, die ausschließlich vom Kunden/Benutzer geschrieben und gelesen, und von der Steuerungsfirmware ignoriert werden.
- Bewegung: Parameter mit Bezug auf den Motor und die Sensoren (BLDC/Stepper, *Closed/Open Loop*...). Einige werden vom Auto-Setup gesetzt und gespeichert.
- Tuning: Parameter mit Bezug auf Motor und Encoder, die entweder vom Auto-Setup gesetzt werden, oder den Datenblättern entnommen werden können, zum Beispiel Polpaare und Maximum Current.

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.

7.5.2 Kategorie: Kommunikation

- $202F_h$: MODBUS Address Control
- 2102_h: Fieldbus Module Control
- 3501_h: EtherNetIP Rx PDO Mapping
- 3502_h: MODBUS Rx PDO Mapping
- 3601_h: EtherNetIP Tx PDO Mapping
- 3602_h: MODBUS Tx PDO Mapping

7.5.3 Kategorie: Applikation

- 2034_h: Upper Voltage Warning Level
- 2035_h: Lower Voltage Warning Level
- 2036_h: Open Loop Current Reduction Idle Time
- 2037_h: Open Loop Current Reduction Value/factor
- 2038_h: Brake Controller Timing
- $203A_h$: Homing On Block Configuration
- $203D_h$: Torque Window
- $203E_h$: Torque Window Time Out

- $203F_h$: Max Slippage Time Out
- 2084_h: Bootup Delay
- 2290_h: PDI Control
- 2300_h: NanoJ Control
- 2410_h: NanoJ Init Parameters
- 2800_h: Bootloader And Reboot Settings
- 3210_h: Motor Drive Parameter Set
- 3212_h: Motor Drive Flags
- $321A_h$: Current Controller Parameters
- $321B_h$: Velocity Controller Parameters
- $321C_h$: Position Controller Parameters
- $321D_h$: Pre-control
- $321E_h$: Voltage Limit
- 3221_h: Analog Inputs Control
- 3240_h: Digital Inputs Control
- 3242_h: Digital Input Routing
- 3243_h: Home Switch Position Capture
- 3250_h: Digital Outputs Control
- 3252_h: Digital Output Routing
- 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
- 4013_h: HW Configuration
- 4015_h: Special Drive Modes
- 6040_h: Controlword
- 6042_h: Vl Target Velocity
- 6046_h: Vl Velocity Min Max Amount
- 6048_h: Vl Velocity Acceleration
- 6049_h: Vl Velocity Deceleration
- $604A_h$: Vl Velocity Quick Stop
- $604C_h$: Vl Dimension Factor
- $605A_h$: Quick Stop Option Code
- $605B_h$: Shutdown Option Code
- $605C_h$: Disable Option Code
- $605D_h$: Halt Option Code
- $605E_h$: Fault Option Code
- 6060_h: Modes Of Operation
- 6065_h: Following Error Window
- 6066_h: Following Error Time Out
- 6067_h: Position Window
- 6068_h: Position Window Time
- $606D_h$: Velocity Window
- $606E_h$: Velocity Window Time
- $606F_h$: 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
- 6081_h: Profile Velocity
- 6082_h: End Velocity
- 6083_h: Profile Acceleration
- 6084_h: Profile Deceleration
- 6085_h: Quick Stop Deceleration
- 6086_h: Motion Profile Type
- 6087_h: Torque Slope
- 6091_h: Gear Ratio
- 6092_h: Feed Constant
- 6096_h: Velocity Factor
- 6097_h: Acceleration Factor
- 6098_h: Homing Method
- 6099_h: Homing Speed
- $609A_h$: Homing Acceleration
- $60A2_h$: Jerk Factor
- $60A4_h$: Profile Jerk
- $60A8_h$: SI Unit Position
- $60A9_h$: SI Unit Velocity
- $60B0_h$: Position Offset
- $60B1_h$: Velocity Offset
- $60B2_h$: Torque Offset
- $60C1_h$: Interpolation Data Record
- $60C2_h$: Interpolation Time Period
- $60C4_h$: Interpolation Data Configuration
- $60C5_h$: Max Acceleration
- $60C6_h$: Max Deceleration
- $60E8_h$: Additional Gear Ratio - Motor Shaft Revolutions
- $60E9_h$: Additional Feed Constant - Feed
- $60ED_h$: Additional Gear Ratio - Driving Shaft Revolutions
- $60EE_h$: Additional Feed Constant - Driving Shaft Revolutions
- $60F2_h$: Positioning Option Code
- $60F8_h$: Max Slippage
- $60FE_h$: Digital Outputs
- $60FF_h$: Target Velocity

7.5.4 Kategorie: Benutzer

- 2701_h: Customer Storage Area

7.5.5 Kategorie: Bewegung

- 3202_h: Motor Drive Submode Select
- $320D_h$: Torque Of Inertia Factor
- 6073_h: Max Current
- 6080_h: Max Motor Speed

7.5.6 Kategorie: Tuning

- 2030_h: Pole Pair Count
- 2031_h: Max Motor Current
- $203B_h$: I2t Parameters
- 3203_h: Feedback Selection
- 3380_h: Feedback Sensorless

- $33A0_h$: Feedback Incremental A/B/I 1
- $33B0_h$: Feedback SSI 1
- 4021_h: Ballast Configuration
- 6075_h: Motor Rated Current
- $608F_h$: Position Encoder Resolution
- 6090_h: Velocity Encoder Resolution
- $60E6_h$: Additional Position Encoder Resolution - Encoder Increments
- $60EB_h$: Additional Position Encoder Resolution - Motor Revolutions

7.5.7 Kategorie: Ethernet

- 2010_h: IP-Configuration
- 2011_h: Static-IPv4-Address
- 2012_h: Static-IPv4-Subnet-Mask
- 2013_h: Static-IPv4-Gateway-Address
- 2017_h: LLDP Non-volatile Storage

7.5.8 Speichervorgang starten

VORSICHT!

Unkontrollierte Motorbewegungen!

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

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

HINWEIS

- Das Speichern kann einige Sekunden dauern. Unterbrechen Sie während des Speicherns keinesfalls die 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
02_h	Kommunikation
03_h	Applikation
04_h	Benutzer
05_h	Bewegung
06_h	Tuning

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

7.5.9 Speicherung verwerfen

Falls alle Objekte oder eine *Kategorie* an gespeicherten Objekten gelöscht werden sollen, muss in das Objekt 1011_h der Wert "$64616F6C_h$" [2] geschrieben werden.

Folgende Subindizes entsprechen dabei einer *Kategorie*:

Subindex	Kategorie
01_h	Alle Kategorien (Zurücksetzen auf Werkseinstellung) mit der Ausnahme von 06_h (Tuning)
02_h	Kommunikation
03_h	Applikation
04_h	Benutzer
05_h	Bewegung
06_h	Tuning

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

HINWEIS

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

7.5.10 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. Sollten das Datum oder die Uhrzeit in 1020 nicht den erwarteten Werten entsprechen, wurden Objekte wahrscheinlich mit anderen als den erwarteten Werten gespeichert.

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

8 EtherNet/IP™

Mittels EtherNet/IP™ ist es möglich, die wichtigsten CiA402-Operationsmodi der Steuerung zu benutzen. Die Benutzung der im CiA402-Standard beschriebenen zyklischen Modi (interpolated position mode, cyclic syncronous torque, velocity und position) ist aktuell nicht möglich.

Folgende Nachrichtentypen werden unterstützt:

- CIP™-Nachrichten (CIP UCMM, CIP class 3 und weitere)
- I/O-Datenübertragung (CIP class 1)

EtherNet/IP™-Referenzen: www.odva.org.

- *THE CIP NETWORKS LIBRARY Volume 1: Common Industrial Protocol (CIP), Date: April.2014,Edition: 3.16*
- *THE CIP NETWORKS LIBRARY Volume 2: EtherNet/IP Adaptation of CIP, Date: April.2015,Edition: 1.19*
- *Network Infrastructure for EtherNet/IP™: Introduction and Considerations*

HINWEIS

Nanotec Steuerungen basieren immer auf dem CANopen-Standard CiA402. Daher werden alle Attribute in einem sogenannten *Objektverzeichnis* (engl. *object dictionary*) gespeichert. Diese können mit einem Index und einem Subindex adressiert werden. z. B. 1018_h:01_h. Der Index ist ein 16-Bit-Wert und der Subindex ein 8-Bit-Wert.

Die Steuerung bietet EtherNet/IP™, aber dieses Protokoll hat ein anderes Vorgehen: alle Attribute werden auf Basis von Objekten gespeichert. Aus diesem Grund nutzt die Steuerung eine Zuordnung um eine Brücke zwischen EtherNet/IP™ und CANopen zu schlagen. Als Folge ist die Konfiguration dieser Steuerung anders als bei normalen EtherNet/IP™-Geräten.

8.1 Geräteprofil

Die Nanotec EtherNet/IP™-Implementation enthält ein herstellerspezifisches Geräteprofil, um Zugriff auf die herstellerspezifischen Steuerungseigenschaften und Motorkontrolle zu bekommen.

Mit den nachfolgenden drei Services ist es möglich jedes Attribut zu erreichen, um es zu lesen oder zu schreiben. Auf jedes Attribut, das in dem Kapitel Objektverzeichnis Beschreibung aufgelistet ist, kann zugegriffen werden.

Name	Service	Class	Instance
Get object dictionary entry	0x32	0x64	1
Set object dictionary entry	0x33	0x64	1
Get object dictionary entry Rockwell	0x34	0x64	1

Zusätzlich zu dem herstellerspezifischen Profil werden noch die folgenden Standard-Services angeboten:

- Message Router Object
- Connection Manager
- Connection Configuration
- Port
- Ethernet Link Object
- TCP/IP Object
- Assembly

8.2 Service: Get object dictionary entry

Die nachfolgende Tabelle zeigt die Zuordnung der Daten, um ein Attribut zu lesen. Die Länge der Daten hängt von dem Attribut ab.

Name	Service	Class	Instance	Daten	
GetOD entry	0x32	0x64	1	Index (16-Bit)	Subindex (8-Bit)

Beispiel

Nachfolgend ein Beispiel, um den Wert des Objekts 2014_h:00_h (Current-IPv4-Address) zu lesen:

Request

```
32 02 20 64 24 01 14 20 00
```

Reply

```
B2 00 00 00 F4 00 A8 C0
```

8.3 Service: Set object dictionary entry

Die folgende Tabelle zeigt die Zuordnung der Daten, um einen Wert in ein Attribut zu schreiben. Die Länge der Daten hängt vom Attribut ab. Es ist mindestens 1 Byte und höchstens 4 Byte lang.

Name	Service	Class	Instance	Daten		
SetOD entry	0x33	0x64	1	Index (16-Bit)	Subindex (8-Bit)	Wert

Beispiel

Nachfolgend ein Beispiel, um den Wert 00000304_h in das Objekt 2031_h:00_h (Max Motor Current) zu schreiben:

Request

```
33 02 20 64 24 01 31 20 00 04 03 00 00
```

Reply

```
B3 00 00 00
```

8.4 Service: Get object dictionary entry Rockwell

Die nachfolgende Tabelle zeigt die Zuordnung der Daten, um ein Attribut zu lesen. Die Antwort enthält einen 16-Bit-Objekttyp, eine 16-Bit Datenlänge und die Daten. Die Antwort darauf ist: Objekttyp (16 Bit), gefolgt von der Datenlänge (16 Bit) und den Daten selbst.

Name	Service	Class	Instance	Daten	
GetOD entry	0x34	0x64	1	Index (16-Bit)	Subindex (8-Bit)

Beispiel

Nachfolgend ein Beispiel eines Lese-Request und Reply für das Objekt 2014_h:00_h (Current-IPv4-Address):

Request

```
34 02 20 64 24 01 14 20 00
```

Reply

```
B4 00 00 00 07 00 04 00 F4 00 A8 C8
```

8.5 Assembly-Objekte

Es gibt ein Set aus Produzent/Verbraucher-Gruppen (*I/O Common*), welche Folgendes enthalten:

- Objekte zur Bedienung des *Plug&Drive-Interface*, womit Sie
  - die unterstützen Betriebsmodi benutzen,
  - den Status der Steuerung überwachen
  - und auf die Objekte des Objektverzeichnisses zugreifen können.

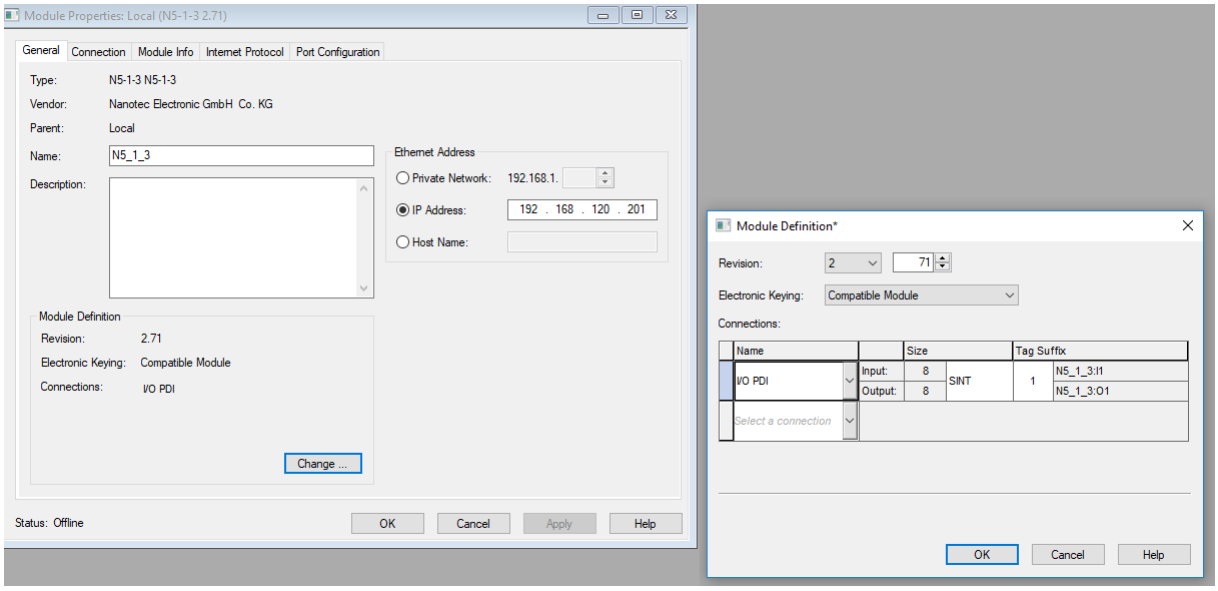
  Weitergehende Informationen zum *Plug&Drive-Interface* finden Sie im Dokument *Funktionsbeschreibung Plug&Drive-Interface* auf www.nanotec.de.
- NanoJ Inputs und Outputs. Damit können Sie Werte an das *NanoJ-Programm* übergeben bzw. Werte auslesen.
- Ein- und Ausgänge
- aktuelle Ist-Werte wie Position, Geschwindigkeit, Drehmoment, Schleppfehler
- den Error-Code des letzten aufgetretenen Fehlers

Ein zusätzliches Set (*I/O PDI*) enthält nur die Input- und Output-Objekte für das *Plug&Drive-Interface* (PDI), siehe PDI-Assemblies.

HINWEIS

Es ist nicht möglich, das Assembly-Set *I/O Common* und die PDI-Assemblies (*I/O PDI*) gleichzeitig zu verwenden.

Wählen Sie nur eine *Connection* aus, entweder *I/O Common* oder *I/O PDI*.



Eine *Configuration Assembly* wird nicht unterstützt. Die Attribute, welche nicht in den *Assemblies* enthalten sind, müssen über das herstellerspezifische Geräteprofil gelesen und geschrieben werden.

Assembly	Pfad
`Assem100`: Target -> Originator Assembly - Common	20 04 24 64 30 03
`Assem101`: Originator -> Target Assembly - Common	20 04 24 65 30 03
Configuration - Common	

Assembly - Common Target -> Originator - Data assignment

Offset (Bytes)	Objektverzeichnis-Eintrag	Größe (Bits)
0	2292_h:01_h PDI Status	16
2	$603F_h$:00_h Error Code	16
4	2292_h:02_h PDI Return Value	32
8	6064_h:00_h Position Actual Value	32
12	$606C_h$:00_h Velocity Actual Value	32
16	$60FD_h$:00_h Digital Inputs	32
20	6077_h:00_h Torque Actual Value	16
22	3220_h:01_h Analog Input 1	16
24	$60F4_h$:00_h Following Error Actual Value	32
28	2500_h:01_h NanoJ Output #1	32
32	2500_h:02_h NanoJ Output #2	32
⋮	⋮	32
116	2500_h:17_h NanoJ Output #23	32
120	Reserviert für zukünftige Benutzung	64 (2 x 32)

Assembly - Common Originator -> Target - Data assignment

Offset (Bytes)	Objektverzeichnis-Eintrag	Größe (Bits)
0	2291_h:01_h PDI Set Value 1	32
4	2291_h:02_h PDI Set Value 2	16
6	2291_h:03_h PDI Set Value 3	8
7	2291_h:04_h PDI Command	8
8	$60FE_h$:01_h Digital Output	32
12	2400_h:01_h NanoJ Input #1	32
16	2400_h:02_h NanoJ Input #2	32
⋮	⋮	32
116	2400_h:$1B_h$ NanoJ Input #27	32
120	Reserviert für zukünftige Benutzung	64 (2 x 32)

PDI-Assemblies

Die Assemblies für das *Plug&Drive-Interface* enthalten das Input- bzw. das Output-Objekt des PDI. Weitergehende Informationen zum *Plug&Drive-Interface* finden Sie im Dokument *Funktionsbeschreibung Plug&Drive-Interface* auf www.nanotec.de.

Assembly	Pfad
`Assem104`: Target -> Originator Assembly - Common	20 04 24 68 30 03
`Assem105`: Originator -> Target Assembly - Common	20 04 24 69 30 03

Assembly - Common Target -> Originator - Data assignment

Offset (Bytes)	Objektverzeichnis-Eintrag	Größe (Bits)
0	2292_h:01_h PDI Status	16
2	$603F_h$:00_h Error Code	16
4	2292_h:02_h PDI Return Value	32

Assembly - Common Originator -> Target - Data assignment

Offset (Bytes)	Objektverzeichnis-Eintrag	Größe (Bits)
0	2291_h:01_h PDI Set Value 1	32
4	2291_h:02_h PDI Set Value 2	16
6	2291_h:03_h PDI Set Value 3	8
8	2291_h:04_h PDI Command	8

8.6 Konfiguration der Assembly-Objekte

Die Assembly-Objekte in der Steuerung sind statisch und unterstützen keine dynamischen Assemblies. Die Größe der statischen Assemblies ist immer 128 Bytes. Die wichtigsten Daten für die verschiedenen Betriebsmodi sind bereits konfiguriert, aber es ist möglich, zusätzliche Daten anzuhängen.

Die gemappten Daten ändern ist nur möglich mit dem *SetOD entry*-Kommando. Es ist darauf zu achten, dass bei einer Änderung der Konfiguration auch die EDS-Datei angepasst werden muss. Es wird empfohlen, die neuen Daten an das Ende des momentanen Mappings anzuhängen. Wie im Kapitel Assembly-Objekte aufgelistet, gibt es einen Datenbereich für zukünftige Benutzung.

HINWEIS

Die PDI-Assemblies können nicht modifziert werden.

Die Daten werden aufeinanderfolgend ohne Lücken ausgerichtet. Falls ein 8-Bit-Datenwert auf beispielsweise 32 Bit ausgerichtet werden soll, ist es möglich, sogenannte *Dummy-Objekte* zu nutzen. Die Benutzung kann auch in der Standardkonfiguration für die Objekte 3501_h und 3601_h eingesehen werden.

HINWEIS

Um das Mapping ändern zu können, müssen Sie es zuerst deaktivieren, indem Sie den entsprechenden Subindex 0_h auf "0" setzen.

Nach dem Schreiben der Objekte in die jeweiligen Subindizes, tragen Sie die Anzahl der gemappten Objekte in den Subindex 0_h ein.

Die folgende Tabelle listet alle verfügbaren *Dummy-Objekte* auf:

Index	Datentyp
0002_h	vorzeichenbehaftete Ganzzahl (8 Bit)
0003_h	vorzeichenbehaftete Ganzzahl (16 Bit)
0004_h	vorzeichenbehaftete Ganzzahl (32 Bit)
0005_h	vorzeichenlose Ganzzahl (8 Bit)
0006_h	vorzeichenlose Ganzzahl (16 Bit)
0007_h	vorzeichenlose Ganzzahl (32 Bit)

8.7 Rockwell Studio 5000

8.7.1 Einschränkungen

Die Steuerung arbeitet mit allen möglichen Datenformaten, sie kann die nachfolgend gelisteten Datenformate behandeln.

	Abkürzung	
Größe in Byte	mit Vorzeichen	ohne Vorzeichen
8	SINT	USINT
16	INT	UINT
32	DINT	UDINT

Rockwell PLCs unterstützen nur Zahlenformate mit Vorzeichen, demzufolge werden alle Attribute in ein DINT (32 Byte, mit Vorzeichen) umgewandelt. Das kann Probleme verursachen bei der Umrechnung einer 32-Bit-Ganzzahl ohne Vorzeichen in eine mit Vorzeichen, falls das höchste Bit gesetzt ist. In diesem Fall wird die Zahl in eine negative - anstelle einer positiven - Zahl konvertiert. Aktuell sind derlei UDINT-Attribute - welche wahrscheinlich das höchste Bit gesetzt haben können - normalerweise ein Status und es ist möglich auf die einzelnen Bits mit einem BOOL-Typ zu referenzieren.

8.8 Panasonic-SPS

HINWEIS

Verbindungsprobleme beim Importieren des EDS-Files bei Panasonic-SPS.

Die Panasonic-SPS unterstützt den Connection Type "NULL" nicht.

► Entfernen Sie vor dem Importieren den Connection Type "NULL" eigenständig aus dem EDS-File.

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.com.

9.1 NanoJ-Programm

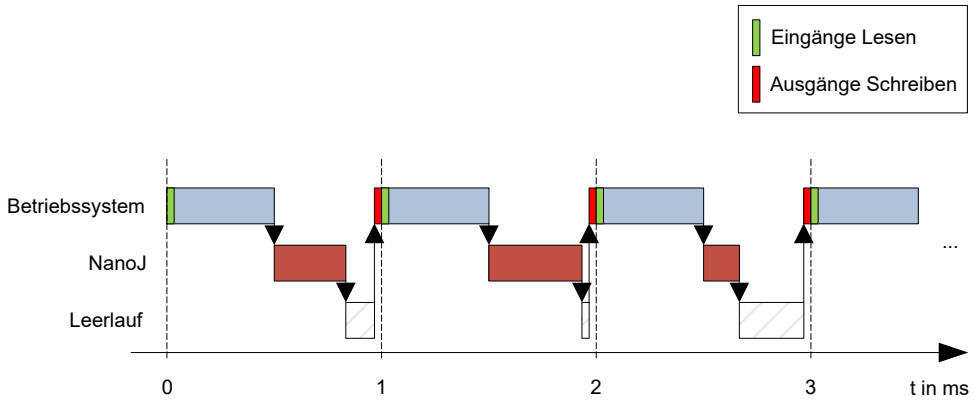
Ein *NanoJ-Programm* stellt eine geschützte Ausführungsumgebung innerhalb der Firmware zur Verfügung. In dieser kann der Anwender eigene Abläufe anlegen. Diese können dann Funktionen in der Steuerung auslösen, indem beispielsweise Einträge im Objektverzeichnis gelesen oder geschrieben werden.

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 (z. B. Debug-Ausgabe schreiben)

Über ein *PDO-Mapping* werden dem Benutzerprogramm OD-Werte in Form von Variablen zur Verfügung gestellt. Bevor ein Benutzerprogramm die 1 ms-Zeitscheibe erhält, werden dazu von der Firmware die Werte aus dem Objektverzeichnis in die Variablen des Benutzerprogramms übertragen. Sobald das Benutzerprogramm Rechenzeit erhält, kann es diese Variablen wie gewöhnliche C-Variablen manipulieren. Am Ende der Zeitscheibe werden letztendlich die neuen Werte von der Firmware wieder automatisch in die jeweiligen OD-Einträge kopiert.

Um die Performance zu optimieren, werden dabei drei Arten von Mappings definiert: Input, Output und Input/Output (In, Out, InOut).

- *Input Mappings* lassen sich nur lesen und werden nicht zurück ins Objektverzeichnis übertragen.
- *Output Mappings* lassen sich nur schreiben.
- *Input/Output Mappings* erlauben hingegen Lesen und Schreiben.

Die gesetzten Mappings können über die GUI bei den Objekten 2310_h, 2320_h, und 2330_h ausgelesen und überprüft werden. Für jedes Mapping sind maximal 16 Einträge erlaubt.

Über die Angabe der *Linker-Section* wird in *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*
- 2410h NanoJ Init Parameters: Array mit zweiunddreißig S32-Werten. Dieses Objekt kann gespeichert werden, im Gegensatz zu 2400_h.
- 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:

- Überprüfen des Eintrags 2302_h auf Fehlercode.
- Wenn kein Fehler:
  NanoJ-Programm starten durch Beschreiben von Objekt 2300_h, Bit 0 = "1" bzw. durch Neustarten der Steuerung.

HINWEIS

Das Starten des NanoJ Programms kann bis zu 200 ms dauern.

- Überprüfen des Eintrags 2302_h auf Fehlercode und des Objekts 2301_h, Bit 0 = "1".

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 hinterlegen.

HINWEIS

Die Dateinamen der Benutzerprogramme dürfen nicht länger als acht Zeichen sein und drei Zeichen im Suffix enthalten; Dateiname `main.cpp` ist zulässig, Dateiname `einLangerDateiname.cpp` ist nicht zulässig.

HINWEIS

In *NanoJ-Programmen* dürfen globale Variablen ausschließlich innerhalb von Funktionen initialisiert werden. Daraus folgt:

- kein `new` Operator
- keine Konstruktoren
- keine Initialisierung von globalen Variablen außerhalb von Funktionen

Beispiele:

Die globale Variable soll erst innerhalb der Funktion `void user()` initialisiert werden:

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

Folgende Zuweisung 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.com.

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

9.3.3 Debug-Ausgabe

Die folgenden Funktionen geben einen Wert in die Debug-Konsole aus. Sie unterscheiden sich anhand des Datentyps des zu übergebenden Parameters.

```
bool VmmDebugOutputString (const char *outstring)
```

```
bool VmmDebugOutputInt (const U32 val)
```

```
bool VmmDebugOutputByte (const U08 val)
```

```
bool VmmDebugOutputHalfWord (const U16 val)
```

```
bool VmmDebugOutputWord (const U32 val)
```

```
bool VmmDebugOutputFloat (const float val)
```

HINWEIS

Die Debug-Ausgaben werden zunächst in einen eigenen Bereich des Objektverzeichnisses geschrieben und dann von dort von *Plug & Drive Studio* ausgelesen.

Dieser OD-Eintrag hat den Index 2600_h und ist 64 Zeichen lang, siehe 2600h NanoJ Debug Output. In Subindex 00 ist immer die Anzahl der bereits geschriebenen Zeichen enthalten.

Ist der Puffer vollgeschrieben, so schlägt `VmmDebugOutputxxx()` zunächst fehl, das Benutzerprogramm wird dann nicht weiter ausgeführt und hält an der Stelle der Debug-Ausgabe an. Erst wenn die GUI den Puffer ausgelesen hat und danach Subindex 00 wieder zurückgesetzt hat, wird das Programm wieder fortgesetzt und `VmmDebugOutputxxx()` kehrt ins Benutzerprogramm zurück.

HINWEIS

Debug-Ausgaben dürfen daher nur während der Testphase bei der Entwicklung eines Benutzerprogramms verwendet werden.

HINWEIS

Nutzen Sie die Debug-Ausgabe nicht, wenn der *AutoYield*-Modus aktiviert ist (siehe Verfügbare Rechenzeit).

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.

Einschränkung/Problem	Maßnahme
Wenn ein Objekt als Output gemappt wurde und der Wert des Objekts niemals vor dem Start des *NanoJ-Programms* festgelegt wird, kann der Wert dieses Objekts zufällig sein.	Initialisieren Sie die Werte der gemappten Objekte 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.
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.

7	6	5	4	3	2	1	0
Beispiel [4]				Beispiel [2]		B	A

Beispiel [4]

Beschreibung der Bits 4 bis einschließlich 7, diese Bits gehören logisch zusammen. Die 4 in den eckigen Klammern gibt die Anzahl der zusammengehörigen Bits an. Oftmals wird an der Stelle noch eine Liste mit möglichen Werten und deren Beschreibung angehängt.

Beispiel [2]

Beschreibung der Bits 3 und 2, diese Bits gehören logisch zusammen. Die 2 in den eckigen Klammern gibt die Anzahl der zusammengehörigen Bits an.

- Wert 00_b: Die Beschreibung an dieser Stelle gilt, wenn Bit 2 und Bit 3 auf "0" sind.
- Wert 01_b: Die Beschreibung an dieser Stelle gilt, wenn Bit 2 auf "0" und Bit 3 auf "1" ist.
- Wert 10_b: Die Beschreibung an dieser Stelle gilt, wenn Bit 2 auf "1" und Bit 3 auf "0" ist.
- Wert 11_b: Die Beschreibung an dieser Stelle gilt, wenn Bit 2 und Bit 3 auf "1" sind.

B

Beschreibung des Bits B, auf die Längenangabe wird bei einem einzelnen Bit verzichtet.

A

Beschreibung des Bits A, Bits mit grauen Hintergrund bleiben ungenutzt.

1000h Device Type

Funktion

Beschreibt den Steuerungstyp.

Objektbeschreibung

Index	1000_h
Objektname	Device Type
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00040192_h ■ PD6-E891M95-M-65-3: 00040192_h ■ PD6-E891S95-M-65-3: 00040192_h ■ PD6-EB80CD-M-65-3: 00020192_h ■ PD6-EB80LD-M-65-3: 00020192_h ■ PD6-EB80MD-M-65-3: 00020192_h ■ PD6-EB80SD-M-65-3: 00020192_h ■ PD6-E891S95-MS-65-3: 00040192_h ■ PD6-E891M95-MS-65-3: 00040192_h ■ PD6-E891L95-MS-65-3: 00040192_h ■ PD6-EB80SD-MS-65-3: 00020192_h ■ PD6-EB80MD-MS-65-3: 00020192_h ■ PD6-EB80LD-MS-65-3: 00020192_h ■ PD6-EB80CD-MS-65-3: 00020192_h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Motor Type [16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Device profile number [16]															

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Error Number [8]								Error Class [8]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Error Code [16]															

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

Fehlernummer	Beschreibung
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
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 ($203B_h$:01_h/6075_h)
23	Encoderauflösung, Polpaarzahl und einige andere Werte sind falsch.
24	Motorstrom ist zu hoch, passen Sie die PI-Parameter an.
25	Interner Softwarefehler, generisch
26	Zu hoher Strom am digitalen Ausgang
27	Nur CANopen: Unerwartete Sync-Länge
30	Fehler in der Drehzahlüberwachung: Schlupffehler zu groß
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.
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.
43	Eingangsspannung (+Ub Logic) zu hoch

Fehlernummer	Beschreibung
44	Eingangsspannung (+Ub Logic) zu niedrig
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*

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 ($203B_h$:01_h/6075_h)
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

Error Code	Beschreibung
8612_h	Fehler in der Positionsüberwachung: Endschalter überschritten

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	■ PD6-E891L95-M-65-3: PD6-E891L95-M-65-3 ■ PD6-E891M95-M-65-3: PD6-E891M95-M-65-3 ■ PD6-E891S95-M-65-3: PD6-E891S95-M-65-3 ■ PD6-EB80CD-M-65-3: PD6-EB80CD-M-65-3 ■ PD6-EB80LD-M-65-3: PD6-EB80LD-M-65-3 ■ PD6-EB80MD-M-65-3: PD6-EB80MD-M-65-3 ■ PD6-EB80SD-M-65-3: PD6-EB80SD-M-65-3 ■ PD6-E891S95-MS-65-3: PD6-E891S95-MS-65-3 ■ PD6-E891M95-MS-65-3: PD6-E891M95-MS-65-3 ■ PD6-E891L95-MS-65-3: PD6-E891L95-MS-65-3 ■ PD6-EB80SD-MS-65-3: PD6-EB80SD-MS-65-3 ■ PD6-EB80MD-MS-65-3: PD6-EB80MD-MS-65-3 ■ PD6-EB80LD-MS-65-3: PD6-EB80LD-MS-65-3 ■ PD6-EB80CD-MS-65-3: PD6-EB80CD-MS-65-3
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-v2538-B1077563
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". Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 3 auf 4. Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5. Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7. Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14. Firmware Version FIR-v2508-B1072143: Eintrag "Name" geändert von "Save Drive Parameters To Non-volatile Memory" auf "Save Motor 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 Drive Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

Subindex	06_h
Name	Save Tuning Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

Subindex	07_h
Name	Save Miscellaneous Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

Subindex	08_h
Name	Save Reserved1 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	09_h
Name	Save Reserved2 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0A_h$
Name	Save CANopen Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

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

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

Subindex	$0D_h$
Name	Save Profibus Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

Beschreibung

Jeder Subindex des Objekts steht für eine bestimmte Speicherklasse. Durch Auslesen eines Eintrages kann festgestellt werden, ob diese Speicherkategorie abgespeichert (Wert "1") werden kann oder nicht (Wert="0").

Um den Speichervorgang einer Speicherkategorie zu starten, muss der Wert "65766173_h" in den jeweiligen Subindex geschrieben werden. Das entspricht dezimal der 1702257011_d bzw. dem ASCII String " `save`. Sobald der Speichervorgang abgeschlossen wurde, wird der Speicherbefehl wieder durch den Wert "1" überschrieben, da ein Speichern wieder möglich ist.

Für eine detaillierte Beschreibung siehe Kapitel Objekte speichern.

1011h Restore Default Parameters

Funktion

Mit diesem Objekt kann das gesamte oder Teile des Objektverzeichnis auf die Defaultwerte zurückgesetzt werden. Siehe Kapitel Objekte speichern.

Objektbeschreibung

Index	1011_h
Objektname	Restore Default Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "Restore Default Parameter" auf "Restore Default Parameters". Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 2 auf 4. Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5. Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Comm Default Parameters" auf "Restore Communication Default Parameters". Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Application Default Parameters" auf "Restore Application Default Parameters". Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7. Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.

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 Drive Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

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

Subindex	07_h
Name	Restore Miscellaneous Configurations
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

Subindex	08_h
Name	Restore Reserved1 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	09_h
Name	Restore Reserved2 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

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

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.

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	■ PD6-E891L95-M-65-3: 00000322_h ■ PD6-E891M95-M-65-3: $0000032B_h$ ■ PD6-E891S95-M-65-3: 00000334_h ■ PD6-EB80CD-M-65-3: $0000033D_h$ ■ PD6-EB80LD-M-65-3: 00000346_h ■ PD6-EB80MD-M-65-3: $0000034F_h$ ■ PD6-EB80SD-M-65-3: 00000358_h ■ PD6-E891S95-MS-65-3: $000003FE_h$ ■ PD6-E891M95-MS-65-3: 00000403_h ■ PD6-E891L95-MS-65-3: 00000408_h ■ PD6-EB80SD-MS-65-3: $0000040D_h$ ■ PD6-EB80MD-MS-65-3: 00000412_h ■ PD6-EB80LD-MS-65-3: 00000417_h ■ PD6-EB80CD-MS-65-3: $0000041C_h$

Subindex	03_h
Name	Revision Number
Datentyp	UNSIGNED32
Zugriff	nur lesen

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$09EA0000_h$

Subindex	04_h
Name	Serial Number
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

1020h Verify Configuration

Funktion

Dieses Objekt zeigt den Tag und die Zeit der abgespeicherten Konfiguration an.

Ein Konfigurationstool oder ein Master kann dieses Objekt nutzen, um die Konfiguration nach einem Reset zu verifizieren und gegebenenfalls eine Neukonfiguration durchzuführen.

Das Tool muss das Datum und die Uhrzeit setzen, bevor der Speichermechanismus gestartet wird (siehe Kapitel Objekte speichern).

Objektbeschreibung

Index	1020_h
Objektname	Verify Configuration
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Prüfung
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00_h
Name	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.

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

200Fh IEEE 802 MAC Address

Funktion

Dieses Objekt enthält die MAC-Adresse der Steuerung als Zeichenkette.

Objektbeschreibung

Index	$200F_h$
Objektname	IEEE 802 MAC Address
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1748-B533384
Änderungshistorie	

2010h IP-Configuration

Funktion

Über dieses Objekt konfigurieren Sie die Ethernet-Schnittstelle. Das Objekt wird nur einmalig beim Neustart der Steuerung berücksichtigt. Weitere Details finden Sie im Kapitel *Inbetriebnahme*.

Objektbeschreibung

Index	2010_h
Objektname	IP-Configuration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Ethernet
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000064_h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1748-B533384: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Ethernet".

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									LLMNR	NBIOS			DHCP		IP

IP

Wert = "1": Eine statische IP-Adresse aus dem Objekt 2011_h wird genutzt und die Netzwerkmaske aus dem Objekt 2012_h wird genutzt.

DHCP

Wert = "1": Die IP-Adressvergabe mittels eines DHCP-Servers wird aktiviert

AUTO

Wert = "1": Die IP-Adressvergabe über das AUTO-IP Protokoll wird aktiviert

NBIOS

Wert = "1": Das NetBIOS-Protokoll wird aktiviert, notwendig vor einer Hostname-Auflösung (z.B. bei einem ping-Kommando).

LLMNR

Wert = "1": Das LLMNR-Protokoll wird aktiviert, notwendig vor einer Hostname-Auflösung (z.B. bei einem ping-Kommando).

HINWEIS

Um Konflikte zu vermeiden, setzen Sie niemals gleichzeitig auf "1" das Bit 0 (statische IP-Adresse) und die Bits 2 und 3 (DHCP bzw. Auto-IP)..

2011h Static-IPv4-Address

Funktion

Enthält die statische IPv4-Adresse in Form eines 32-Bit Wortes.

Objektbeschreibung

Index	2011_h
Objektname	Static-IPv4-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Ethernet
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$C0A80792_h$
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1450: Eintrag "Object Name" geändert von "Static-IP-Address" auf "Static-IPv4-Address". Firmware Version FIR-v1748-B533384: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Ethernet".

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP Address Part 1 [8]								IP Address Part 2 [8]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP Address Part 3 [8]								IP Address Part 4 [8]							

IP Address Part 1 [8]

Gibt den ersten Teil der IP-Adresse an

IP Address Part 2 [8]

Gibt den zweiten Teil der IP-Adresse an

IP Address Part 3 [8]

Gibt den dritten Teil der IP-Adresse an

IP Address Part 4 [8]

Gibt den vierten Teil der IP-Adresse an

Beispiel

Die Adresse `192.168.2.0` wird zuerst in das Hexadezimalsystem umgewandelt und ergibt dann folgenden Konfigurationswert:

192 => $C0_h$

168 => $A8_h$

2 => 02_h

0 => 0

Der zugehörige Einstellwert lautet dann `C0A80200`$_h$.

2012h Static-IPv4-Subnet-Mask

Funktion

Enthält die Subnetzmaske der statischen IP-Adresse in Form eines 32-Bit Wortes.

Objektbeschreibung

Index	2012_h
Objektname	Static-IPv4-Subnet-Mask
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Ethernet
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$FFFFFF00_h$
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1450: Eintrag "Object Name" geändert von "Static-IP-Subnet-Mask" auf "Static-IPv4-Subnet-Mask". Firmware Version FIR-v1748-B533384: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Ethernet".

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Subnet Mask Part 1 [8]								Subnet Mask Part 2 [8]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Subnet Mask Part 3 [8]								Subnet Mask Part 4 [8]							

Subnet Mask Part 1 [8]

Gibt den ersten Teil der Subnetzmaske an

Subnet Mask Part 2 [8]

Gibt den zweiten Teil der Subnetzmaske an

Subnet Mask Part 3 [8]

Gibt den dritten Teil der Subnetzmaske an

Subnet Mask Part 4 [8]

Gibt den vierten Teil der Subnetzmaske an

Beispiel

Die Klasse-C Netzwerkmaske `255.255.255.0` wird zuerst in das Hexadezimalsystem umgewandelt und ergibt dann folgenden Konfigurationswert:

255 => FF_h

0 => 0

Der zugehörige Einstellwert lautet dann $FFFFFF00_h$.

2013h Static-IPv4-Gateway-Address

Funktion

Enthält die statische IP-Gateway-Adresse in Form eines 32-Bit Wortes.

Objektbeschreibung

Index	2013_h
Objektname	Static-IPv4-Gateway-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Ethernet
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h
Firmware Version	FIR-v1446
Änderungshistorie	Firmware Version FIR-v1512: Eintrag "Object Name" geändert von "Static-IP-Gateway-Address" auf "Static-IPv4-Gateway-Address". Firmware Version FIR-v1748-B533384: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Ethernet".

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP-Gateway-Address Part 1 [8]								IP-Gateway-Address Part Part 2 [8]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP-Gateway-Address Part 3 [8]								IP-Gateway-Address Part Part 4 [8]							

IP-Gateway-Adress Part 1 [8]

Gibt den ersten Teil der IP-Gateway-Adresse an

IP-Gateway-Adress Part 2 [8]

Gibt den zweiten Teil der IP-Gateway-Adresse an

IP-Gateway-Adress 3 [8]

Gibt den dritten Teil der IP-Gateway-Adresse an

IP-Gateway-Adress 4 [8]

Gibt den vierten Teil der IP-Gateway-Adresse an

Beispiel

Die Adresse `192.168.2.0` wird zuerst in das Hexadezimalsystem umgewandelt und ergibt dann folgenden Konfigurationswert:

192 => $C0_h$

168 => $A8_h$

2 => 02_h

0 => 0

Der zugehörige Einstellwert lautet dann $C0A80200_h$.

2014h Current-IPv4-Address

Funktion

Enthält die derzeit aktive IP-Adresse in Form eines 32-Bit Wortes.

Objektbeschreibung

Index	2014_h
Objektname	Current-IPv4-Address
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-v1450: Eintrag "Object Name" geändert von "Current-IP-Address" auf "Current-IPv4-Address".

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP Address Part 1 [8]								IP Address Part 2 [8]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP Address Part 3 [8]								IP Address Part 4 [8]							

IP Address Part 1 [8]

Gibt den ersten Teil der IP-Adresse an

IP Address Part 2 [8]

Gibt den zweiten Teil der IP-Adresse an

IP Address Part 3 [8]

Gibt den dritten Teil der IP-Adresse an

IP Address Part 4 [8]

Gibt den vierten Teil der IP-Adresse an

Beispiel

Die Adresse `192.168.2.0` wird zuerst in das Hexadezimalsystem umgewandelt und ergibt dann folgenden Konfigurationswert:

192 => $C0_h$

168 => $A8_h$

2 => 02_h

0 => 0

Der zugehörige Einstellwert lautet dann $C0A80200_h$.

2015h Current-IPv4-Subnet-Mask

Funktion

Enthält die derzeit aktive Subnetzmaske der statischen IP-Adresse in Form eines 32-Bit Wortes.

Objektbeschreibung

Index	2015_h
Objektname	Current-IPv4-Subnet-Mask
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-v1450: Eintrag "Object Name" geändert von "Current-IP-Subnet-Mask" auf "Current-IPv4-Subnet-Mask".

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Subnet Mask Part 1 [8]								Subnet Mask Part 2 [8]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Subnet Mask Part 3 [8]								Subnet Mask Part 4 [8]							

Subnet Mask Part 1 [8]

Gibt den ersten Teil der Subnetzmaske an

Subnet Mask Part 2 [8]

Gibt den zweiten Teil der Subnetzmaske an

Subnet Mask Part 3 [8]

Gibt den dritten Teil der Subnetzmaske an

Subnet Mask Part 4 [8]

Gibt den vierten Teil der Subnetzmaske an

Beispiel

Die Klasse-C Netzwerkmaske `255.255.255.0` wird zuerst in das Hexadezimalsystem umgewandelt und ergibt dann folgenden Konfigurationswert:

255 => FF_h

0 => 0

Der zugehörige Einstellwert lautet dann `FFFFFF00`$_h$.

2016h Current-IPv4-Gateway-Address

Funktion

Dieses Objekt enthält die derzeit aktive Gateway IP-Adresse in Form eines 32-Bit Wortes.

Objektbeschreibung

Index	2016_h
Objektname	Current-IPv4-Gateway-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h
Firmware Version	FIR-v1540
Änderungshistorie	

2017h LLDP Non-volatile Storage

Funktion

In diesem Objekt werden die Einstellungen für das LLDP *Link-Layer Discovery Protocol* gespeichert (Speicherkategorie *Ethernet)*.

Objektbeschreibung

Index	2017_h
Objektname	LLDP Non-volatile Storage
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Ethernet
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	
Firmware Version	FIR-v2239-B1032745
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	LLDP_Enable -used Lower Byte Only
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0007_h

Subindex	02_h
Name	Tx_Timeout
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$001E_h$

Subindex	03_h
Name	MsgTxHold -used Lower Byte Only
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0004_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h (LLDP_Enable):
  - Bit "0": Global Enable, wenn gesetzt auf "1"
  - Bit "1": LLDP Tx & Rx Enabled, wenn gesetzt auf "1"

- Bit "2": LLDP Tx Enabled, wenn gesetzt auf "1"
- 02_h (Tx_Timeout): Das Intervall in Sekunden, in dem LLDP-Frames von diesem Gerät gesendet werden.
- 03_h (MsgTxHold): Message-Transmission-Multiplier für die LLDP-Frames

202Fh MODBUS Address Control

Funktion

In diesem Objekt können Sie die Startadressen für die verschiedenen Modbus-Register manuell anpassen, falls Ihre SPS die vorgegebenen Adressenbereiche nicht unterstützt.

Objektbeschreibung

Index	$202F_h$
Objektname	MODBUS Address Control
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2213-B1062280
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	NanoJ Read Base Address
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$07D0_h$

Subindex	02_h
Name	NanoJ Write Base Address
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$0BB8_h$

Subindex	03_h
Name	PDI Read Base Address
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	1384_h

Subindex	04_h
Name	PDI Write Base Address
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$176C_h$

Subindex	05_h
Name	PDO Read Base Address
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	1388_h

Subindex	06_h
Name	PDO Write Base Address
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	1770_h

2030h Pole Pair Count

Funktion

Enthält die Polpaarzahl des angeschlossenen Motors.

Objektbeschreibung

Index	2030_h

Objektname	Pole Pair Count
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00000032_h ■ PD6-E891M95-M-65-3: 00000032_h ■ PD6-E891S95-M-65-3: 00000032_h ■ PD6-EB80CD-M-65-3: 00000004_h ■ PD6-EB80LD-M-65-3: 00000004_h ■ PD6-EB80MD-M-65-3: 00000004_h ■ PD6-EB80SD-M-65-3: 00000004_h ■ PD6-E891S95-MS-65-3: 00000032_h ■ PD6-E891M95-MS-65-3: 00000032_h ■ PD6-E891L95-MS-65-3: 00000032_h ■ PD6-EB80SD-MS-65-3: 00000004_h ■ PD6-EB80MD-MS-65-3: 00000004_h ■ PD6-EB80LD-MS-65-3: 00000004_h ■ PD6-EB80CD-MS-65-3: 00000004_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". Firmware Version FIR-v2239-B1032745: Eintrag "Datentyp" geändert von "UNSIGNED16" auf "UNSIGNED32".

2031h Max Motor Current

Funktion

Hier tragen Sie den maximal zulässigen Motorstrom in Milliampere ein. Alle Stromwerte werden durch diesen Wert begrenzt.

Steuerungsintern wird der eingegebene Wert immer als Effektivwert interpretiert.

Objektbeschreibung

Index	2031_h
Objektname	Max Motor Current
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: $0000251C_h$

	■ PD6-E891M95-M-65-3: $000024FE_h$ ■ PD6-E891S95-M-65-3: 00002508_h ■ PD6-EB80CD-M-65-3: $000124F8_h$ ■ PD6-EB80LD-M-65-3: $0000FDE8_h$ ■ PD6-EB80MD-M-65-3: $00009C40_h$ ■ PD6-EB80SD-M-65-3: $00004E20_h$ ■ PD6-E891S95-MS-65-3: 00002508_h ■ PD6-E891M95-MS-65-3: $000024FE_h$ ■ PD6-E891L95-MS-65-3: $0000251C_h$ ■ PD6-EB80SD-MS-65-3: $00004E20_h$ ■ PD6-EB80MD-MS-65-3: $00009C40_h$ ■ PD6-EB80LD-MS-65-3: $0000FDE8_h$ ■ PD6-EB80CD-MS-65-3: $000124F8_h$
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning". Firmware Version FIR-v1614: Eintrag "Object Name" geändert von "Peak Current" auf "Max Current". Firmware Version FIR-v1748-B538662: Eintrag "Object Name" geändert von "Maximum Current" auf "Max Motor Current". Firmware Version FIR-v1825-B577172: Eintrag "Object Name" geändert von "Max Motor Current" auf "Maximum Current". Firmware Version FIR-v1825-B577172: Eintrag "Object Name" geändert von "Maximum Current" auf "Max Motor Current". Firmware Version FIR-v1825-B577172: Eintrag "Object Name" geändert von "Max Motor Current" auf "Maximum Current". Firmware Version FIR-v1825-B577172: Eintrag "Object Name" geändert von "Maximum Current" auf "Max Motor Current".

2034h Upper Voltage Warning Level

Funktion

Dieses Objekt enthält den Schwellwert für den Fehler "Überspannung" in Millivolt.

Objektbeschreibung

Index	2034_h
Objektname	Upper Voltage Warning Level
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$0000F5B3_h$
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	$00002C52_h$
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Fällt die Eingangsspannung der Steuerung unter diesen Schwellwert, wird der Motor abgeschaltet und ein Fehler ausgelöst. Der Fehler setzt sich automatisch zurück, wenn die Eingangsspannung größer als die Spannung des Objekts 2035_h plus 1,5 Volt ist.

2036h Open Loop Current Reduction Idle Time

Funktion

Dieses Objekt beschreibt die Zeit in Millisekunden, die sich der Motor im Stillstand befinden muss, bis die Stromabsenkung aktiviert wird.

Objektbeschreibung

Index	2036_h
Objektname	Open Loop Current Reduction Idle Time
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$000003E8_h$
Firmware Version	FIR-v1426
Änderungshistorie	

2037h Open Loop Current Reduction Value/factor

Funktion

Dieses Objekt beschreibt den Effektivstrom, auf den der Motorstrom im Open Loop reduziert werden soll, wenn sich der Motor im Stillstand befindet.

Objektbeschreibung

Index	2037_h
Objektname	Open Loop Current Reduction Value/factor
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$FFFFFFCE_h$
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert von 2037_h größer/gleich 0 und kleiner als Wert 6075_h

Strom wird auf den hier eingetragenen Wert reduziert. Der Wert wird in mA und als Effektivwert interpretiert.

Wert von 2037_h im Bereich von -1 bis -100

Der eingetragene Wert wird als eine Prozentzahl interpretiert und bestimmt die Reduktion des Nennstroms in 2037_h. Für die Berechnung wird der Wert in 6075_h herangezogen.

Beispiel: Das Objekt 6075_h hat den Wert 4200 mA. Der Wert -60 in 2037_h senkt den Strom um 60% von 6075_h ab, somit ergibt sich eine Stromabsenkung auf einen Effektivwert von 6075_h * (2037_h + 100) / 100 = 1680 mA.

Die Angabe -100 in 2037_h würde z.B. bedeuten, dass eine Stromabsenkung auf einen Effektivwert von 0 mA eingestellt wird.

2038h Brake Controller Timing

Funktion

Dieses Objekt enthält die Zeiten für die *Bremsensteuerung* in Millisekunden sowie die PWM-Frequenz und den Tastgrad.

Objektbeschreibung

Index	2038_h
Objektname	Brake Controller Timing
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426

Änderungshistorie	

Wertebeschreibung

Subindex	00_h
Name	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 0 bzw. 50 (Bremsenausgang) und 2000 ($7D0_h$) 20000 ($4E20_h$)
Vorgabewert	00000000_h

Subindex	06_h
Name	PWM Duty Cycle
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	0, zwischen 2 und 100 (64_h)
Vorgabewert	00000000_h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Zeit zwischen dem Motorstillstand und dem Schließen der Bremse.
- 02_h: Zeit zwischen dem Schließen der Bremse und dem Abschalten des Motorstroms.
- 03_h: Zeit zwischen dem Einschalten des Motorstroms und dem Öffnen der Bremse.
- 04_h: Zeit zwischen dem Öffnen der Bremse und dem Erreichen des Zustands *Operation enabled* der CiA 402 Power State Machine.
- 05_h: Frequenz des PWM-Signals (Software-PWM) in Hertz.
- 06_h: Tastgrad des PWM-Signals (Software-PWM) 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".

	Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "TX-PDO". Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 03 geändert von "nein" auf "TX-PDO". Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 04 geändert von "nein" auf "TX-PDO". 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 durch √2, also 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, 2031_h und $203B_h$:05_h auf eine Skala gelegt werden.
  Open Loop: $I = \sqrt{(I\alpha^2 + I\beta^2)} / \sqrt{2}$
  Closed Loop: $I = \mathrm{sgn}(Iq) * \sqrt{(I\alpha^2 + I\beta^2)} / \sqrt{2}$

HINWEIS

Die Motorströme I_d (Subindex 01_h) und I_q (Subindex 02_h) werden nur angezeigt, wenn der Closed Loop aktiviert wurde, sonst wird der Wert 0 ausgegeben.

203Ah Homing On Block Configuration

Funktion

Dieses Objekt enthält die Parameter für das *Homing auf Block* (siehe Kapitel Homing).

Objektbeschreibung

Index	$203A_h$
Objektname	Homing On Block Configuration
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	
PDO-Mapping	
Zulässige Werte	

Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 4 auf 3. Firmware Version FIR-v1540: Eintrag "Name" geändert von "Period Of Blocking" auf "Block Detection Time". Firmware Version FIR-v1614: Eintrag "Data Type" geändert von "UNSIGNED32" auf "INTEGER32". Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation". Firmware Version FIR-v1614: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32". Firmware Version FIR-v1614: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".

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	Minimum Current For Block Detection
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00000947_h ■ PD6-E891M95-M-65-3: $0000093F_h$ ■ PD6-E891S95-M-65-3: 00000942_h ■ PD6-EB80CD-M-65-3: $0000186A_h$ ■ PD6-EB80LD-M-65-3: $0000124F_h$ ■ PD6-EB80MD-M-65-3: $00000DAC_h$ ■ PD6-EB80SD-M-65-3: $000006BD_h$ ■ PD6-E891S95-MS-65-3: 00000942_h ■ PD6-E891M95-MS-65-3: $0000093F_h$ ■ PD6-E891L95-MS-65-3: 00000947_h ■ PD6-EB80SD-MS-65-3: $000006BD_h$ ■ PD6-EB80MD-MS-65-3: $00000DAC_h$ ■ PD6-EB80LD-MS-65-3: $0000124F_h$ ■ PD6-EB80CD-MS-65-3: $0000186A_h$

Subindex	02_h
Name	Block Detection Time
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$000000C8_h$

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Gibt den Stromgrenzwert an, ab dem ein Blockieren detektiert werden soll. Positive Zahlenwerte geben die Stromgrenze in mA an, negative Zahlen einen Prozentwert von Objekt 2031_h. Beispiel: der Wert "1000" entspricht 1000 mA (=1 A), der Wert "-70" entspricht 70% von 2031_h.
- 02_h: Gibt die Zeit in ms an, die der Motor nach der Blockdetektion trotzdem noch gegen den Block fahren soll.

203Bh I2t Parameters

Funktion

Dieses Objekt hält die Parameter für die I^2t-Überwachung.

Die I^2t-Überwachung wird aktiviert, in dem in $203B_h$:01 und $203B_h$:02 ein Wert größer 0 eingetragen wird und in 6073_h ein Wert größer 1000 (siehe I2t Motor-Überlastungsschutz).

I^2t kann nur für den *Closed Loop*-Betrieb verwendet werden, mit einer Ausnahme: Wenn I^2t im *Open Loop*-Betrieb aktiviert ist, wird der Strom auf den kleineren der Werte von $203B_h$:01_h, 6073_h und 2031_h begrenzt.

Objektbeschreibung

Index	$203B_h$
Objektname	I2t Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1512: Eintrag "Savable" geändert von "nein" auf "ja, Kategorie: Applikation". Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 7 auf 8. Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning". Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Nominal Current" auf "Motor Rated Current". Firmware Version FIR-v1825-B577172: Eintrag "Name" geändert von "Motor Rated Current" auf "Nominal Current". Firmware Version FIR-v1825-B577172: Eintrag "Name" geändert von "Nominal Current" auf "Motor Rated Current". Firmware Version FIR-v1825-B577172: Eintrag "Name" geändert von "Motor Rated Current" auf "Nominal Current".

	Firmware Version FIR-v1825-B577172: Eintrag "Name" geändert von "Nominal Current" auf "Motor Rated Current". Firmware Version FIR-v1825-B577172: Die Anzahl der Einträge haben sich geändert von 8 auf 7. Firmware Version FIR-v1926-B648637: Eintrag "Name" geändert von "Maximum Duration Of Peak Current" auf "Maximum Duration Of Max Current".

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	Motor Rated Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: $0000251C_h$ ■ PD6-E891M95-M-65-3: $000024FE_h$ ■ PD6-E891S95-M-65-3: 00002508_h ■ PD6-EB80CD-M-65-3: $000061A8_h$ ■ PD6-EB80LD-M-65-3: $0000493E_h$ ■ PD6-EB80MD-M-65-3: $000036B0_h$ ■ PD6-EB80SD-M-65-3: $00001AF4_h$ ■ PD6-E891S95-MS-65-3: 00002508_h ■ PD6-E891M95-MS-65-3: $000024FE_h$ ■ PD6-E891L95-MS-65-3: $0000251C_h$ ■ PD6-EB80SD-MS-65-3: $00001AF4_h$ ■ PD6-EB80MD-MS-65-3: $000036B0_h$ ■ PD6-EB80LD-MS-65-3: $0000493E_h$ ■ PD6-EB80CD-MS-65-3: $000061A8_h$

Subindex	02_h
Name	Maximum Duration Of Max Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00000000_h

	■ PD6-E891M95-M-65-3: 00000000_h ■ PD6-E891S95-M-65-3: 00000000_h ■ PD6-EB80CD-M-65-3: 00001388_h ■ PD6-EB80LD-M-65-3: 00001388_h ■ PD6-EB80MD-M-65-3: 00001388_h ■ PD6-EB80SD-M-65-3: 00001388_h ■ PD6-E891S95-MS-65-3: 00000000_h ■ PD6-E891M95-MS-65-3: 00000000_h ■ PD6-E891L95-MS-65-3: 00000000_h ■ PD6-EB80SD-MS-65-3: 00001388_h ■ PD6-EB80MD-MS-65-3: 00001388_h ■ PD6-EB80LD-MS-65-3: 00001388_h ■ PD6-EB80CD-MS-65-3: 00001388_h

Subindex	03_h
Name	Threshold
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	04_h
Name	CalcValue
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	05_h
Name	LimitedCurrent
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	06_h
Name	Status
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

Die Subindizes sind in zwei Gruppen geteilt: Subindex 01_h und 02_h enthalten Parameter zur Steuerung, Subindex 03_h bis 06_h sind Statuswerte. Die Funktionen sind wie folgt:

- 01_h: Hier wird der im Motordatenblatt angegebene Nennstrom in mA eingetragen. Dieser muss kleiner als der eingegebene Strom in 2031_h und 6073_h sein, sonst wird die Überwachung nicht aktiviert. Der angegebene Wert wird als Effektivwert interpretiert.
- 02_h: Gibt die maximale Dauer des Maximalstroms (6073_h) in ms an.
- 03_h: Threshold, gibt die Grenze in A^2ms an, von der abhängt, ob auf Maximalstrom oder Nennstrom geschalten wird.
- 04_h: CalcValue, gibt den berechneten Wert in A^2ms an, welcher mit Threshold verglichen wird, um den Strom einzustellen.
- 05_h: LimitedCurrent, zeigt den gegenwärtigen Strom als Effektivwert an, der von I^2t eingestellt wurde.
- 06_h: aktueller Status. Ist der Subentry-Wert "0", ist I^2t deaktiviert, ist der Wert "1", wird I^2t aktiviert.

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". 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.

205Ah Absolute Sensor Boot Value (in User Units)

Funktion

Aus diesem Objekt kann die initiale Encoderposition beim Einschalten der Steuerung (in benutzerdefinierten Einheiten) ausgelesen werden.

Objektbeschreibung

Index	$205A_h$

Objektname	Absolute Sensor Boot Value (in User Units)
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h
Firmware Version	FIR-v1446
Änderungshistorie	Firmware Version FIR-v1512: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen". Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Encoder Boot Value" auf "Absolute Sensor Boot Value (in User Units)". Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".

2084h Bootup Delay

Funktion

Definiert den Zeitraum zwischen Anlegen der Versorgungsspannung an die Steuerung und der Funktionsbereitschaft der Steuerung in Millisekunden.

Objektbeschreibung

Index	2084_h
Objektname	Bootup Delay
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h
Firmware Version	FIR-v1426
Änderungshistorie	

2101h Fieldbus Module Availability

Funktion

Zeigt die verfügbaren Feldbusse an.

Objektbeschreibung

Index	2101_h
Objektname	Fieldbus Module Availability

Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$001D0011_h$
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Object Name" geändert von "Fieldbus Module" auf "Fieldbus Module Availability".

Beschreibung

Die Bits 0 bis 15 zeigen die physikalische Schnittstelle an, die Bits 16 bis 31 das benutzte Protokoll (falls notwendig).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													E-IP	MTCP	MRTU

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SPI	E-CAT	E-NET	CAN	RS232	RS485	USB

USB

Wert = "1": Der Feldbus USB ist verfügbar.

RS-485

Wert = "1": Eine RS-485 Schnittstelle ist verfügbar.

RS-232

Wert = "1": Eine RS-232 Schnittstelle ist verfügbar.

CAN

Wert = "1": Der Feldbus CANopen ist verfügbar.

E-NET

Wert = "1": Eine Ethernet Schnittstelle ist verfügbar.

E-CAT

Wert = "1": Eine EtherCAT Schnittstelle ist verfügbar.

SPI

Wert = "1": Eine SPI Schnittstelle ist verfügbar.

MRTU

Wert = "1": Das benutzte Protokoll ist Modbus RTU.

MTCP

Wert = "1": Das benutzte Protokoll ist Modbus TCP

E-IP

Wert = "1": Das benutzte Protokoll ist EtherNet/IP™

2102h Fieldbus Module Control

Funktion

Mit diesem Objekt können bestimmte Feldbusse (physikalischen Schnittstellen und Protokolle) aktiviert/deaktiviert werden.

Objektbeschreibung

Index	2102_h
Objektname	Fieldbus Module Control
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$001D0011_h$
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Kommunikation".

Beschreibung

Im Objekt 2103_h:1_h werden alle physikalischen Schnittstellen/Protokolle angezeigt, welche aktiviert/deaktiviert werden können. Diese können in diesem Objekt (2102_h) geschaltet werden. Der gegenwärtige Status der aktivierten Feldbusse steht im Objekt 2103_h:2_h.

Dabei gilt die folgende Verteilung der Bits:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													E-IP	MTCP	MRTU

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SPI	E-CAT	E-NET	CAN	RS232	RS485	USB

USB

USB Schnittstelle

RS-485

RS-485 Schnittstelle

RS-232

RS-232 Schnittstelle

CAN

CANopen Schnittstelle

E-NET

EtherNET Schnittstelle

E-CAT

EtherCAT Schnittstelle

SPI
SPI Schnittstelle

MRTU
Modbus RTU Protokoll

MTCP
Modbus TCP Protokoll

E-IP
EtherNet/IP™ Protokoll

2103h Fieldbus Module Status

Funktion

Zeigt die aktiven Feldbusse an.

Objektbeschreibung

Index	2103_h
Objektname	Fieldbus Module Status
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00_h
Name	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	00000000_h

Subindex	02_h
Name	Fieldbus Module Enabled
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$001D0011_h$

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:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													E-IP	MTCP	MRTU

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SPI	E-CAT	E-NET	CAN	RS232	RS485	USB

USB
USB Schnittstelle

RS-485
RS-485 Schnittstelle

RS-232
RS-232 Schnittstelle

CAN
CANopen Schnittstelle

E-NET
EtherNET Schnittstelle

E-CAT
EtherCAT Schnittstelle

SPI
SPI Schnittstelle

MRTU
Modbus RTU Protokoll

MTCP
Modbus TCP Protokoll

E-IP

EtherNet/IP™ Protokoll

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". Firmware Version FIR-v2315-B1040535: Eintrag "Data type" geändert von "INTEGER8" auf "UNSIGNED8". Firmware Version FIR-v2239-B1032745: Eintrag "Data type" geändert von "UNSIGNED8" auf "INTEGER8".

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	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00_h

2292h PDI Output

Funktion

Wenn Sie das *Plug&Drive-Interface* benutzen, können Sie in diesem Objekt den Status und einen vom verwendeten Betriebsmodus abhängigen Rückgabewert lesen. Weitergehende Informationen finden Sie im Dokument *Funktionsbeschreibung Plug&Drive-Interface*.

Objektbeschreibung

Index	2292_h
Objektname	PDI Output
Object Code	RECORD
Datentyp	PDI_OUTPUT
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B531667
Änderungshistorie	Firmware Version FIR-v2315-B1040535: Eintrag "Data type" geändert von "INTEGER16" auf "UNSIGNED16". Firmware Version FIR-v2239-B1032745: Eintrag "Data type" geändert von "UNSIGNED16" auf "INTEGER16".

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	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	0000_h

Subindex	02_h
Name	PDI Return Value
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

2300h NanoJ Control

Funktion

Steuert die Ausführung eines *NanoJ-Programms*.

Objektbeschreibung

Index	2300_h
Objektname	NanoJ Control
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Control" auf "NanoJ Control".

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													AYield		ON

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).

HINWEIS

Nutzen Sie die Debug-Ausgabe nicht, wenn der *AutoYield*-Modus aktiviert ist.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ERR	RES	RUN

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
0001_h	Firmware unterstützt verwendete Funktion nicht (z. B. `sin`, `cos` etc.)
0005_h	Time Out: Code wird zu lange ohne `yield()` oder `sleep()` ausgeführt
0007_h	Zu viele Variablen auf dem Stack
0100_h	Ungültige NanoJ Programmdatei
0101_h	Ungültige NanoJ-Version der Programmdatei
0102_h	CRC-Fehler in der NanoJ-Programmdatei

Fehler bei dem Zugriff auf ein Objekt:

Nummer	Beschreibung
$1xxxxyy_h$	Ungültiges Mapping in der NanoJ-Programmdatei: Der Wert in "xxxx" benennt den Index, der Wert in "yy" den Subindex des Objekts, das gemappt werden soll aber nicht gemappt werden kann.
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)

Nummer	Beschreibung
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

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". Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein". Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen". Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".

Wertebeschreibung

Subindex	00_h
Name	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.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Index [16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SubIndex [8]								Length [8]							

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". Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein". Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen". Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".

Wertebeschreibung

Subindex	00_h
Name	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.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Index [16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SubIndex [8]								Length [8]							

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". Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein". Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen". Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".

Wertebeschreibung

Subindex	00_h
Name	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.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Index [16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SubIndex [8]								Length [8]							

Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

Darin ist die Länge des zu mappenden Objektes in der Einheit Bit enthalten.

2400h NanoJ Inputs

Funktion

Hier befindet sich ein Array mit 32 32-Bit Integerwerten, das innerhalb der Firmware nicht verwendet wird und ausschließlich zur Kommunikation mit dem Benutzerprogramm über den Feldbus dient.

Objektbeschreibung

Index	2400_h
Objektname	NanoJ Inputs
Object Code	ARRAY
Datentyp	INTEGER32

Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 33 Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Inputs" auf "NanoJ Inputs". Firmware Version FIR-v1436: Eintrag "Name" geändert von "VMM Input N#" auf "NanoJ Input N#".

Wertebeschreibung

Subindex	00_h
Name	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.

2410h NanoJ Init Parameters

Funktion

Dieses Objekt funktioniert identisch dem Objekt 2400_h mit dem Unterschied, dass dieses Objekt gespeichert werden kann.

Objektbeschreibung

Index	2410_h
Objektname	NanoJ Init Parameters
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	
Firmware Version	FIR-v1450
Änderungshistorie	Firmware Version FIR-v1450: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED8".

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20_h

Subindex	01_h - 20_h
Name	NanoJ Init Parameter #1 - #32
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

2500h NanoJ Outputs

Funktion

Hier befindet sich ein Array mit 32 32-Bit Integerwerten, das innerhalb der Firmware nicht verwendet wird und ausschließlich zur Kommunikation mit dem Benutzerprogramm über den Feldbus dient.

Objektbeschreibung

Index	2500_h
Objektname	NanoJ Outputs
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Outputs" auf "NanoJ Outputs". Firmware Version FIR-v1436: Eintrag "Name" geändert von "VMM Output N#" auf "NanoJ Output N#".

Wertebeschreibung

Subindex	00_h
Name	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.

2600h NanoJ Debug Output

Funktion

Dieses Objekt enthält Debug-Ausgaben eines Benutzerprogramms.

Objektbeschreibung

Index	2600_h
Objektname	NanoJ Debug Output
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Debug Output" auf "NanoJ Debug Output".

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00_h

Subindex	01_h - 40_h
Name	Value #1 - #64
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00_h

Beschreibung

Hier legt das NanoJ-Programm die Debug-Ausgaben ab, welche mit der Funktion `VmmDebugOutputString()` oder `VmmDebugOutputInt()` aufgerufen wurden.

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 und das Kurzschließen der Motorwicklungen im Bootloader-Modus aus- und einschalten.

Objektbeschreibung

Index	2800_h
Objektname	Bootloader And Reboot Settings
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

Subindex	00_h
Name	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	00000000_h

Subindex	03_h
Name	Bootloader HW Config
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Wird hier der Wert "$746F6F62_h$" eingetragen, wird die Firmware rebootet.
- 02_h: Zeit in Millisekunden: verzögert den Reboot der Firmware um die jeweilige Zeit.
- 03_h: mit dem Bit 0 kann das Kurzschließen der Motorwicklungen im Bootloader-Modus aus- und eingeschaltet werden:
  - Bit 0= 1: Das Kurzschließen der Motorwicklungen im Bootloader-Modus wird ausgeschaltet.
  - Bit 0= 0: Das Kurzschließen der Motorwicklungen im Bootloader-Modus wird eingeschaltet.

3202h Motor Drive Submode Select

Funktion

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

Objektbeschreibung

Index	3202_h
Objektname	Motor Drive Submode Select
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00000001_h ■ PD6-E891M95-M-65-3: 00000001_h ■ PD6-E891S95-M-65-3: 00000001_h ■ PD6-EB80CD-M-65-3: 00000041_h ■ PD6-EB80LD-M-65-3: 00000041_h ■ PD6-EB80MD-M-65-3: 00000041_h

	■ PD6-EB80SD-M-65-3: 00000041_h ■ PD6-E891S95-MS-65-3: 00000001_h ■ PD6-E891M95-MS-65-3: 00000001_h ■ PD6-E891L95-MS-65-3: 00000001_h ■ PD6-EB80SD-MS-65-3: 00000041_h ■ PD6-EB80MD-MS-65-3: 00000041_h ■ PD6-EB80LD-MS-65-3: 00000041_h ■ PD6-EB80CD-MS-65-3: 00000041_h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Fahrt". Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Fahrt" auf "ja, Kategorie: Bewegung".

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								Slow	BLDC	Torque	AutoAl	CurRed	Brake	VoS	CL/OL

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*.

VoS

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

Brake

Wert = "1": Einschalten der automatischen Bremsensteuerung.

CurRed (Current Reduction)

Stromabsenkung im *Open Loop*: seit der Version *FIR-v2213* bleibt diese aktiviert, solange die Objekte 2036_h und 2037_h nicht beide auf "0" stehen.

Torque

nur in den Betriebsmodi Profile Torque und Cyclic Synchronous Torque aktiv

Wert = "1": M-Regler ist aktiv, andernfalls ist ein V-Regler überlagert: in den Torque-Modi wird kein V-Regler zur Geschwindigkeitsbegrenzung verwendet, das Objekt 6080_h wird also ignoriert, 3210_h:3 und 3210_h:4 haben keinen Einfluss auf die Regelung.

BLDC

Wert = "1": Motortyp "BLDC" (Bürstenloser Gleichstrommotor)

Slow (*Slow Speed*)

Wert = "1": die Betriebsart Slow Speed wird aktiviert (der *Closed Loop* muss bereits aktiviert sein)

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: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-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	RX-PDO
Zulässige Werte	
Vorgabewert	03_h

Subindex	01_h
Name	1st Feedback Interface
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00_h

Subindex	02_h
Name	2nd Feedback Interface
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	00_h

Subindex	03_h
Name	3rd Feedback Interface
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:
  Subindex n enthält eine Bitmaske für die jeweilige Rückführung n. Die Bits haben dabei folgende Bedeutung:
- Bit 0: wird das Bit auf "1" gesetzt, wird 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*.

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	03_h

Subindex	01_h
Name	Index Of 1st Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	3380_h

Subindex	02_h
Name	Index Of 2nd Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	$33A0_h$

Subindex	03_h
Name	Index Of 3rd 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*.

320Dh Torque Of Inertia Factor

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	Torque Of Inertia Factor
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Bewegung
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1825-B577172
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	Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

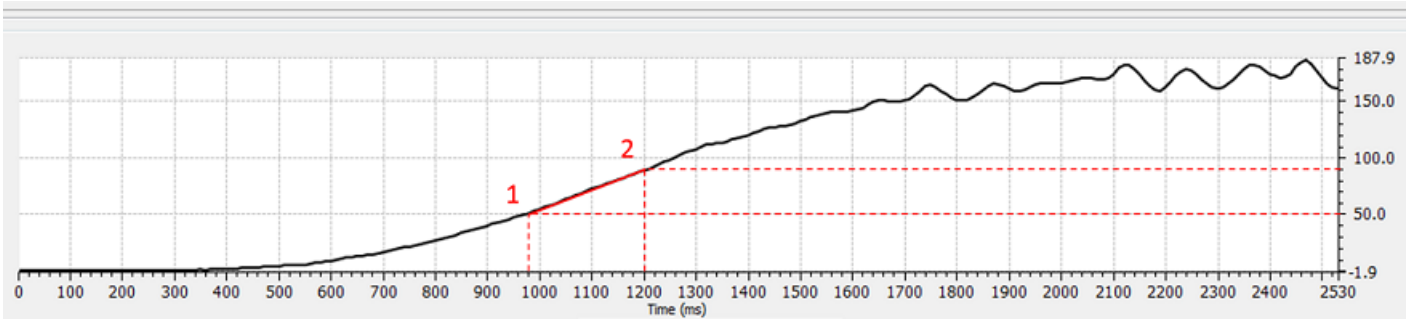
Subindex	02_h

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 entsprechenden Stromwert (mA) 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.



3210h Motor Drive Parameter Set

Funktion

Beinhaltet die P- und I-Anteile der Strom-, Geschwindigkeits- und Positionsregler für *Open Loop* (nur Stromregler aktiviert) und *Closed Loop*.

Objektbeschreibung

Index	3210_h
Objektname	Motor Drive Parameter Set
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Name" geändert von "S_P" auf "Position Loop, Proportional Gain (closed Loop)". Firmware Version FIR-v1626: Eintrag "Name" geändert von "S_I" auf "Position Loop, Integral Gain (closed Loop)".

	Firmware Version FIR-v1626: Eintrag "Name" geändert von "V_P" auf "Velocity Loop, Proportional Gain (closed Loop)". Firmware Version FIR-v1626: Eintrag "Name" geändert von "V_I" auf "Velocity Loop, Integral Gain (closed Loop)". Firmware Version FIR-v1626: Eintrag "Name" geändert von "Id_P" auf "Flux Current Loop, Proportional Gain (closed Loop)". Firmware Version FIR-v1626: Eintrag "Name" geändert von "Id_I" auf "Flux Current Loop, Integral Gain (closed Loop)". Firmware Version FIR-v1626: Eintrag "Name" geändert von "Iq_P" auf "Torque Current Loop, Proportional Gain (closed Loop)". Firmware Version FIR-v1626: Eintrag "Name" geändert von "Iq_I" auf "Torque Current Loop, Integral Gain (closed Loop)". Firmware Version FIR-v1626: Eintrag "Name" geändert von "I_P" auf "Torque Current Loop, Proportional Gain (dspDrive - Stepper Motor, Open Loop)". Firmware Version FIR-v1626: Eintrag "Name" geändert von "I_I" auf "Torque Current Loop, Integral Gain (dspDrive - Stepper Motor, Open Loop)". Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Torque Current Loop, Proportional Gain (dspDrive - Stepper Motor, Open Loop)" auf "Torque Current Loop, Proportional Gain (open Loop)". Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Torque Current Loop, Integral Gain (dspDrive - Stepper Motor, Open Loop)" auf "Torque Current Loop, Integral Gain (open Loop)". Firmware Version FIR-v1650-B472161: Eintrag "Datentyp" geändert von "INTEGER32" auf "UNSIGNED32". Firmware Version FIR-v1650-B472161: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED32". Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 11 auf 13. Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 bis 0A geändert von "nein" auf "RX-PDO".

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$0C_h$

Subindex	01_h
Name	Position Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000800_h

Subindex	02_h
Name	Position Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	03_h
Name	Velocity Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$00002EE0_h$

Subindex	04_h
Name	Velocity Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$0000001E_h$

Subindex	05_h
Name	Flux Current Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$00881EE0_h$

Subindex	06_h
Name	Flux Current Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	

Vorgabewert	$0007C740_h$

Subindex	07_h
Name	Torque Current Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$00881EE0_h$

Subindex	08_h
Name	Torque Current Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$0007C740_h$

Subindex	09_h
Name	Torque Current Loop, Proportional Gain (open Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$004DC880_h$

Subindex	$0A_h$
Name	Torque Current Loop, Integral Gain (open Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$001D2B30_h$

Subindex	$0B_h$
Name	Velocity Feed Forward Factor In Per Mille
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$000003E8_h$

Subindex	$0C_h$

Name	Acceleration Feed Forward Factor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

- Subindex 00_h: Anzahl der Einträge
- Subindex 01_h: Proportional-Anteil des S-Reglers (Position)
- Subindex 02_h: Integral-Anteil des S-Reglers (Position)
- Subindex 03_h: Proportional-Anteil des V-Reglers (Geschwindigkeit)
- Subindex 04_h: Integral-Anteil des V-Reglers (Geschwindigkeit)
- Subindex 05_h: (Closed Loop) Proportional-Anteil des Stromreglers der feldbildenden Komponente
- Subindex 06_h: (Closed Loop) Integral-Anteil des Stromreglers der feldbildenden Komponente
- Subindex 07_h: (Closed Loop) Proportional-Anteil des Stromreglers der momentbildenden Komponente
- Subindex 08_h: (Closed Loop) Integral-Anteil des Stromreglers der momentbildenden Komponente
- Subindex 09_h: (Open Loop) Proportional-Amteil des Stromreglers der feldbildenden Komponente
- Subindex $0A_h$: (Open Loop) Integral-Anteil des Stromreglers der feldbildenden Komponente
- Subindex $0B_h$: (Closed Loop) Geschwindigkeitsvorsteuerung in Promille. Default ist 1000 und damit ein Faktor von 1.
- Subindex $0C_h$: (Closed Loop) Beschleunigungsvorsteuerung. Default ist 0 (Vorsteuerung inaktiv). Ist auch beim Verzögern wirksam.

3212h Motor Drive Flags

Funktion

Mit diesem Objekt wird bestimmt, ob das Auto-Setup die Regler-Parameter anpassen soll, oder nicht. Zudem kann die Richtung des Drehfeldes geändert werden.

HINWEIS

Änderungen im Subindex 02_h werden erst nach einem Neustart der Steuerung aktiv. Das Auto-Setup muss danach erneut durchgeführt werden.

Objektbeschreibung

Index	3212_h
Objektname	Motor Drive Flags
Object Code	ARRAY
Datentyp	INTEGER8
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1450

Änderungshistorie	Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 2 auf 3. Firmware Version FIR-v1738-B501312: Eintrag "Name" geändert von "Enable Legacy Power Mode" auf "Reserved".

Wertebeschreibung

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

Subindex	01_h
Name	Reserved
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00_h

Subindex	02_h
Name	Override Field Inversion
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00_h

Subindex	03_h
Name	Auto-setup With Current Controller Parameters From The OD
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00_h

Subindex	04_h
Name	Use 321Ah, 321Bh, 321Ch, 321Dh, Instead Of 3210h
Datentyp	INTEGER8
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01_h

Beschreibung

Für den Subindex 02_h gültige Werte:

- Wert = "0": Default-Werte der Firmware benutzen
- Wert = "1": nicht Invertieren des Drehfeldes erzwingen (mathematisch positiv)
- Wert = "-1": Invertieren des Drehfeldes erzwingen (mathematisch negativ)

Für den Subindex 03_h gültige Werte:

- Wert = "0": Auto-Setup erkennt den Motortyp (Schrittmotor oder BLDC-Motor) und verwendet den entsprechenden vorkonfigurierten Parametersatz.
- Wert = "1": Auto-Setup mit den Regelparametern durchführen, die vor dem Auto-Setup im Objekt 3210_h bzw. $321A_h$ bis $321E_h$ eingetragen wurden. Die Regelparameter werden nicht geändert.

Für den Subindex 04_h gültige Werte:

- Wert = "0": Die alten Regelparameter aus dem Objekt 3210_h werden verwendet.
- Wert = "1": Die neuen Regelparameter (siehe Reglerstruktur) werden verwendet.

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	■ PD6-E891L95-M-65-3: $00002A30_h$ ■ PD6-E891M95-M-65-3: $000020D0_h$ ■ PD6-E891S95-M-65-3: 00000820_h ■ PD6-EB80CD-M-65-3: 00000258_h ■ PD6-EB80LD-M-65-3: 00000320_h ■ PD6-EB80MD-M-65-3: $000004B0_h$ ■ PD6-EB80SD-M-65-3: $00000A28_h$ ■ PD6-E891S95-MS-65-3: 00000820_h ■ PD6-E891M95-MS-65-3: $000020D0_h$ ■ PD6-E891L95-MS-65-3: $00002A30_h$ ■ PD6-EB80SD-MS-65-3: $00000A28_h$ ■ PD6-EB80MD-MS-65-3: $000004B0_h$ ■ PD6-EB80LD-MS-65-3: 00000320_h ■ PD6-EB80CD-MS-65-3: 00000258_h

Subindex	02_h
Name	Integrator Time Ti For Iq [µs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00002891_h ■ PD6-E891M95-M-65-3: $000023AA_h$ ■ PD6-E891S95-M-65-3: 00001450_h ■ PD6-EB80CD-M-65-3: 00001207_h ■ PD6-EB80LD-M-65-3: 00001262_h ■ PD6-EB80MD-M-65-3: 00001770_h ■ PD6-EB80SD-M-65-3: 00001223_h ■ PD6-E891S95-MS-65-3: 00001450_h ■ PD6-E891M95-MS-65-3: $000023AA_h$ ■ PD6-E891L95-MS-65-3: 00002891_h ■ PD6-EB80SD-MS-65-3: 00001223_h ■ PD6-EB80MD-MS-65-3: 00001770_h ■ PD6-EB80LD-MS-65-3: 00001262_h ■ PD6-EB80CD-MS-65-3: 00001207_h

Subindex	03_h
Name	Proportional Gain Kp For Id [mV/A]

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: $00002A30_h$ ■ PD6-E891M95-M-65-3: $000020D0_h$ ■ PD6-E891S95-M-65-3: 00000820_h ■ PD6-EB80CD-M-65-3: 00000258_h ■ PD6-EB80LD-M-65-3: 00000320_h ■ PD6-EB80MD-M-65-3: $000004B0_h$ ■ PD6-EB80SD-M-65-3: $00000A28_h$ ■ PD6-E891S95-MS-65-3: 00000820_h ■ PD6-E891M95-MS-65-3: $000020D0_h$ ■ PD6-E891L95-MS-65-3: $00002A30_h$ ■ PD6-EB80SD-MS-65-3: $00000A28_h$ ■ PD6-EB80MD-MS-65-3: $000004B0_h$ ■ PD6-EB80LD-MS-65-3: 00000320_h ■ PD6-EB80CD-MS-65-3: 00000258_h

Subindex	04_h
Name	Integrator Time Ti For Id [µs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00002891_h ■ PD6-E891M95-M-65-3: $000023AA_h$ ■ PD6-E891S95-M-65-3: 00001450_h ■ PD6-EB80CD-M-65-3: 00001207_h ■ PD6-EB80LD-M-65-3: 00001262_h ■ PD6-EB80MD-M-65-3: 00001770_h ■ PD6-EB80SD-M-65-3: 00001223_h ■ PD6-E891S95-MS-65-3: 00001450_h ■ PD6-E891M95-MS-65-3: $000023AA_h$ ■ PD6-E891L95-MS-65-3: 00002891_h ■ PD6-EB80SD-MS-65-3: 00001223_h ■ PD6-EB80MD-MS-65-3: 00001770_h ■ PD6-EB80LD-MS-65-3: 00001262_h ■ PD6-EB80CD-MS-65-3: 00001207_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	■ PD6-E891L95-M-65-3: 00000140_h ■ PD6-E891M95-M-65-3: 00000140_h ■ PD6-E891S95-M-65-3: 00000140_h ■ PD6-EB80CD-M-65-3: $00000FA0_h$ ■ PD6-EB80LD-M-65-3: $00000FA0_h$ ■ PD6-EB80MD-M-65-3: $00000FA0_h$ ■ PD6-EB80SD-M-65-3: $00000FA0_h$ ■ PD6-E891S95-MS-65-3: 00000140_h ■ PD6-E891M95-MS-65-3: 00000140_h ■ PD6-E891L95-MS-65-3: 00000140_h ■ PD6-EB80SD-MS-65-3: $00000FA0_h$ ■ PD6-EB80MD-MS-65-3: $00000FA0_h$ ■ PD6-EB80LD-MS-65-3: $00000FA0_h$ ■ PD6-EB80CD-MS-65-3: $00000FA0_h$

Subindex	02_h

Name	Integrator Time Ti [µs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: $00007A12_h$ ■ PD6-E891M95-M-65-3: $00007A12_h$ ■ PD6-E891S95-M-65-3: $00007A12_h$ ■ PD6-EB80CD-M-65-3: $00004E20_h$ ■ PD6-EB80LD-M-65-3: $00004E20_h$ ■ PD6-EB80MD-M-65-3: $00004E20_h$ ■ PD6-EB80SD-M-65-3: $00004E20_h$ ■ PD6-E891S95-MS-65-3: $00007A12_h$ ■ PD6-E891M95-MS-65-3: $00007A12_h$ ■ PD6-E891L95-MS-65-3: $00007A12_h$ ■ PD6-EB80SD-MS-65-3: $00004E20_h$ ■ PD6-EB80MD-MS-65-3: $00004E20_h$ ■ PD6-EB80LD-MS-65-3: $00004E20_h$ ■ PD6-EB80CD-MS-65-3: $00004E20_h$

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	■ PD6-E891L95-M-65-3: 00000166_h ■ PD6-E891M95-M-65-3: 00000166_h ■ PD6-E891S95-M-65-3: 00000166_h ■ PD6-EB80CD-M-65-3: $0000001D_h$ ■ PD6-EB80LD-M-65-3: $0000001D_h$ ■ PD6-EB80MD-M-65-3: $0000001D_h$ ■ PD6-EB80SD-M-65-3: $0000001D_h$ ■ PD6-E891S95-MS-65-3: 00000166_h ■ PD6-E891M95-MS-65-3: 00000166_h ■ PD6-E891L95-MS-65-3: 00000166_h ■ PD6-EB80SD-MS-65-3: $0000001D_h$ ■ PD6-EB80MD-MS-65-3: $0000001D_h$ ■ PD6-EB80LD-MS-65-3: $0000001D_h$ ■ PD6-EB80CD-MS-65-3: $0000001D_h$

Subindex	02_h
Name	Integrator Time Ti [µs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

321Dh Pre-control

Funktion

Enthält die Parameter für die Vorsteuerung. Siehe Kapitel Reglerstruktur.

Objektbeschreibung

Index	$321D_h$
Objektname	Pre-control
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	03_h

Subindex	01_h
Name	Voltage Pre-control [‰]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$000003E8_h$

Subindex	02_h
Name	Acceleration Pre-control [‰]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$000003E8_h$

Subindex	03_h
Name	Velocity Pre-control [‰]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$000003E8_h$

321Eh Voltage Limit

Funktion

Maximal zulässige PWM-Spannung (Tastverhältnis). Werte ≤ 1000 werden als Promille-Werte (der verfügbaren Spannung) interpretiert. Werte > 1000 als 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:

Betriebsspannung*0,9425

U_{o_high}

Ab dieser Spannung findet die maximale Übermodulation statt. Wird wie folgt berechnet:

bei zweiphasigen Schrittmotoren: Betriebsspannung*1,063

bei dreiphasigen BLDC-Motoren: Betriebsspannung*0,99

3220h Analog Input Digits

Funktion

Zeigt die Momentanwerte der Analogeingänge in *ADC Digits* an.

Durch Objekt 3221_h kann der jeweilige Analogeingang als Strom- oder Spannungseingang konfiguriert werden.

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	01_h

Subindex	01_h
Name	Analog Input #1 Digit
Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000_h

Beschreibung

Formeln zum Umrechnen von [digits] in die jeweilige Einheit:

- Stromeingang (falls konfigurierbar): x Digits * 20 mA / 1023 Digits

3221h Analog Inputs Control

Funktion

Mit diesem Objekt lässt sich ein Analogeingang von Spannungs- auf Strommessung umschalten, falls die Hardware das erlaubt (siehe technische Daten).

Objektbeschreibung

Index	3221_h
Objektname	Analog Inputs Control
Object Code	VARIABLE
Datentyp	INTEGER32

Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														AC2	AC1

Generell gilt: Wird ein Bit auf den Wert "0" gesetzt, misst der Analogeingang die Spannung, ist das Bit auf den Wert "1" gesetzt, wird der Strom gemessen.

AC1

Einstellung für Analogeingang 1

AC2

Einstellung für Analogeingang 2

3240h Digital Inputs Control

Funktion

Mit diesem Objekt lassen sich digitale Eingänge manipulieren wie in Kapitel Digitale Ein- und Ausgänge beschrieben .

Objektbeschreibung

Index	3240_h
Objektname	Digital Inputs Control
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
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" Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 8 auf 9.

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	Special Function Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	02_h
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	03_h
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	04_h
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	05_h

Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	06_h
Name	Input Range Select
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	07_h
Name	Differential Select
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	08_h
Name	Routing Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00000000_h ■ PD6-E891M95-M-65-3: 00000000_h ■ PD6-E891S95-M-65-3: 00000000_h ■ PD6-EB80CD-M-65-3: 00000000_h ■ PD6-EB80LD-M-65-3: 00000000_h ■ PD6-EB80MD-M-65-3: 00000000_h ■ PD6-EB80SD-M-65-3: 00000000_h ■ PD6-E891S95-MS-65-3: 00000001_h ■ PD6-E891M95-MS-65-3: 00000001_h ■ PD6-E891L95-MS-65-3: 00000001_h ■ PD6-EB80SD-MS-65-3: 00000001_h ■ PD6-EB80MD-MS-65-3: 00000001_h ■ PD6-EB80LD-MS-65-3: 00000001_h ■ PD6-EB80CD-MS-65-3: 00000001_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 3240_h:01_h (Special Function Enable): Dieses Bit erlaubt Sonderfunktionen eines Eingangs aus- (Wert "0") oder einzuschalten (Wert "1"). Soll Eingang 1 z.B. nicht als negativer Endschalter verwendet werden, so muss die Sonderfunktion abgeschaltet werden, damit nicht fälschlicherweise auf den Signalgeber reagiert wird. Auf die Bits 16 bis 31 hat das Objekt keine Auswirkungen.
  Die Firmware wertet folgende Bits aus:
  - Bit 0: Negativer Endschalter
  - Bit 1: Positiver Endschalter
  - Bit 2: Referenzschalter
  - Bit 3: Interlock

  Sollen z.B. zwei Endschalter und ein Referenzschalter verwendet werden, müssen Bits 0-2 in 3240_h:01_h auf "1" gesetzt werden.
- 3240_h:02_h (Function Inverted): Dieser Subindex wechselt von Schließer-Logik (ein logischer High-Pegel am Eingang ergibt den Wert "1" im Objekt $60FD_h$) auf Öffner-Logik (der logische High-Pegel am Eingang ergibt den Wert "0").
  Das gilt für die Sonderfunktionen (außer den Takt- und Richtungseingängen) und für die normalen Eingänge. Hat das Bit den Wert "0" gilt Schließer-Logik, entsprechend bei dem Wert "1" die Öffner-Logik. Bit 0 wechselt die Logik des Eingangs 1, Bit 1 die Logik des Eingangs 2 usw. .
- 3240_h:03_h (Force Enable): Dieser Subindex schaltet die Softwaresimulation von Eingangswerten ein, wenn das entsprechende Bit auf "1" gesetzt ist.
  Dann werden nicht mehr die tatsächlichen, sondern die in Objekt 3240_h:04_h eingestellten Werte für den jeweiligen Eingang verwendet. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw..
- 3240_h:04_h (Force Value): Dieses Bit gibt den Wert vor, der als Eingangswert eingelesen werden soll, wenn das gleiche Bit im Objekt 3240_h:03_h gesetzt wurde.
- 3240_h:05_h (Raw Value): Dieses Objekt beinhaltet den unmodifizierten Eingabewert.
- 3240_h:06_h (Input Range Select): Damit können Eingänge - welche über diese Funktion verfügen - von der Schaltschwelle von 5 V (Bit auf "0") auf die Schaltschwelle 24 V (Bit auf "1") umgeschaltet werden. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw.
- 3240_h:07_h (Differential Select): Dieser Subindex schaltet bei den Eingängen zwischen "single-ended Eingang" (Wert "0" in dem Subindex) zu "Differentieller Eingang" (Wert "1" in dem Subindex) für alle Eingänge auf einmal um.
- 3240_h:08_h (Routing Enable): Der Wert "1" in diesem Subindex aktiviert das Input Routing.

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	Firmware Version FIR-v2538-B1077563: Eintrag "Name" geändert von "Input Source For Bit #0 In 60FDh - Negative Limit Switch" auf "Input Source For Bit #0 In 60FDh - Negative Limit Switch". Firmware Version FIR-v2538-B1077563: Eintrag "Name" geändert von "Input Source For Bit #1 In 60FDh - Positive Limit Switch" auf "Input Source For Bit #1 In 60FDh - Positive Limit Switch". Firmware Version FIR-v2538-B1077563: Eintrag "Name" geändert von "Input Source For Bit #2 In 60FDh - Home Switch" auf "Input Source For Bit #2 In 60FDh - Home Switch". Firmware Version FIR-v2538-B1077563: Eintrag "Name" geändert von "Input Source For Bit #3 In 60FDh - Interlock" auf "Input Source For Bit #3 In 60FDh - Interlock".

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	■ PD6-E891L95-M-65-3: 00_h ■ PD6-E891M95-M-65-3: 00_h ■ PD6-E891S95-M-65-3: 00_h ■ PD6-EB80CD-M-65-3: 00_h ■ PD6-EB80LD-M-65-3: 00_h ■ PD6-EB80MD-M-65-3: 00_h ■ PD6-EB80SD-M-65-3: 00_h ■ PD6-E891S95-MS-65-3: 01_h ■ PD6-E891M95-MS-65-3: 01_h ■ PD6-E891L95-MS-65-3: 01_h ■ PD6-EB80SD-MS-65-3: 01_h ■ PD6-EB80MD-MS-65-3: 01_h ■ PD6-EB80LD-MS-65-3: 01_h ■ PD6-EB80CD-MS-65-3: 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	■ PD6-E891L95-M-65-3: 00_h ■ PD6-E891M95-M-65-3: 00_h ■ PD6-E891S95-M-65-3: 00_h ■ PD6-EB80CD-M-65-3: 00_h ■ PD6-EB80LD-M-65-3: 00_h ■ PD6-EB80MD-M-65-3: 00_h ■ PD6-EB80SD-M-65-3: 00_h ■ PD6-E891S95-MS-65-3: 02_h ■ PD6-E891M95-MS-65-3: 02_h ■ PD6-E891L95-MS-65-3: 02_h ■ PD6-EB80SD-MS-65-3: 02_h ■ PD6-EB80MD-MS-65-3: 02_h ■ PD6-EB80LD-MS-65-3: 02_h ■ PD6-EB80CD-MS-65-3: 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	■ PD6-E891L95-M-65-3: 00_h ■ PD6-E891M95-M-65-3: 00_h ■ PD6-E891S95-M-65-3: 00_h ■ PD6-EB80CD-M-65-3: 00_h ■ PD6-EB80LD-M-65-3: 00_h ■ PD6-EB80MD-M-65-3: 00_h ■ PD6-EB80SD-M-65-3: 00_h ■ PD6-E891S95-MS-65-3: 03_h ■ PD6-E891M95-MS-65-3: 03_h ■ PD6-E891L95-MS-65-3: 03_h ■ PD6-EB80SD-MS-65-3: 03_h ■ PD6-EB80MD-MS-65-3: 03_h ■ PD6-EB80LD-MS-65-3: 03_h ■ PD6-EB80CD-MS-65-3: 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	■ PD6-E891L95-M-65-3: 00_h ■ PD6-E891M95-M-65-3: 00_h ■ PD6-E891S95-M-65-3: 00_h

	■ PD6-EB80CD-M-65-3: 00_h ■ PD6-EB80LD-M-65-3: 00_h ■ PD6-EB80MD-M-65-3: 00_h ■ PD6-EB80SD-M-65-3: 00_h ■ PD6-E891S95-MS-65-3: 04_h ■ PD6-E891M95-MS-65-3: 04_h ■ PD6-E891L95-MS-65-3: 04_h ■ PD6-EB80SD-MS-65-3: 04_h ■ PD6-EB80MD-MS-65-3: 04_h ■ PD6-EB80LD-MS-65-3: 04_h ■ PD6-EB80CD-MS-65-3: 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	■ PD6-E891L95-M-65-3: 00_h ■ PD6-E891M95-M-65-3: 00_h ■ PD6-E891S95-M-65-3: 00_h ■ PD6-EB80CD-M-65-3: 00_h ■ PD6-EB80LD-M-65-3: 00_h ■ PD6-EB80MD-M-65-3: 00_h ■ PD6-EB80SD-M-65-3: 00_h ■ PD6-E891S95-MS-65-3: 05_h ■ PD6-E891M95-MS-65-3: 05_h ■ PD6-E891L95-MS-65-3: 05_h ■ PD6-EB80SD-MS-65-3: 05_h ■ PD6-EB80MD-MS-65-3: 05_h ■ PD6-EB80LD-MS-65-3: 05_h ■ PD6-EB80CD-MS-65-3: 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	■ PD6-E891L95-M-65-3: 00_h ■ PD6-E891M95-M-65-3: 00_h ■ PD6-E891S95-M-65-3: 00_h ■ PD6-EB80CD-M-65-3: 00_h ■ PD6-EB80LD-M-65-3: 00_h ■ PD6-EB80MD-M-65-3: 00_h ■ PD6-EB80SD-M-65-3: 00_h ■ PD6-E891S95-MS-65-3: 06_h ■ PD6-E891M95-MS-65-3: 06_h

	■ PD6-E891L95-MS-65-3: 06_h ■ PD6-EB80SD-MS-65-3: 06_h ■ PD6-EB80MD-MS-65-3: 06_h ■ PD6-EB80LD-MS-65-3: 06_h ■ PD6-EB80CD-MS-65-3: 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	00_h

Subindex	18_h
Name	Input Source For Bit #23 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00_h

Subindex	19_h
Name	Input Source For Bit #24 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00_h

Subindex	$1A_h$
Name	Input Source For Bit #25 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00_h

Subindex	$1B_h$
Name	Input Source For Bit #26 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	

Vorgabewert	00_h

Subindex	$1C_h$
Name	Input Source For Bit #27 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00_h

Subindex	$1D_h$
Name	Input Source For Bit #28 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00_h

Subindex	$1E_h$
Name	Input Source For Bit #29 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00_h

Subindex	$1F_h$
Name	Input Source For Bit #30 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00_h ■ PD6-E891M95-M-65-3: 00_h ■ PD6-E891S95-M-65-3: 00_h ■ PD6-EB80CD-M-65-3: 00_h ■ PD6-EB80LD-M-65-3: 00_h ■ PD6-EB80MD-M-65-3: 00_h ■ PD6-EB80SD-M-65-3: 00_h ■ PD6-E891S95-MS-65-3: $E4_h$ ■ PD6-E891M95-MS-65-3: $E4_h$ ■ PD6-E891L95-MS-65-3: $E4_h$ ■ PD6-EB80SD-MS-65-3: $E4_h$ ■ PD6-EB80MD-MS-65-3: $E4_h$ ■ PD6-EB80LD-MS-65-3: $E4_h$

	■ PD6-EB80CD-MS-65-3: $E4_h$

Subindex	20_h
Name	Input Source For Bit #31 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00_h ■ PD6-E891M95-M-65-3: 00_h ■ PD6-E891S95-M-65-3: 00_h ■ PD6-EB80CD-M-65-3: 00_h ■ PD6-EB80LD-M-65-3: 00_h ■ PD6-EB80MD-M-65-3: 00_h ■ PD6-EB80SD-M-65-3: 00_h ■ PD6-E891S95-MS-65-3: $E5_h$ ■ PD6-E891M95-MS-65-3: $E5_h$ ■ PD6-E891L95-MS-65-3: $E5_h$ ■ PD6-EB80SD-MS-65-3: $E5_h$ ■ PD6-EB80MD-MS-65-3: $E5_h$ ■ PD6-EB80LD-MS-65-3: $E5_h$ ■ PD6-EB80CD-MS-65-3: $E5_h$

3243h Home Switch Position Capture

Funktion

Mit diesem Objekt kann die aktuelle Position automatisch notiert werden, wenn am digitalen Eingang, der für den Referenzschalter verwendet wird, ein Pegelwechsel stattfindet.

HINWEIS

Verwenden Sie diese Funktion nicht in Kombination mit einer Referenzfahrt. Sonst kann die Referenzfahrt nicht erfolgreich abgeschlossen werden.

Objektbeschreibung

Index	3243_h
Objektname	Home Switch Position Capture
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312

Änderungshistorie	

Wertebeschreibung

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

Subindex	01_h
Name	Control
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	02_h
Name	Capture Count
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	03_h
Name	Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	04_h
Name	Sensor Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

- Subindex 01_h: damit wird der Typ des Pegelwechsels ausgewählt:
  - Funktion deaktivieren: Wert "0"
  - Mit steigender Flanke: Wert "1"
  - Mit fallender Flanke: Wert "2"
  - Beide Flanken: Wert "3"
- Subindex 02_h: gibt die Anzahl der notierten Pegelwechsel seit dem letzten Start der Funktion wieder; wird auf 0 zurückgesetzt, wenn Subindex 01_h auf 1,2 oder 3 gesetzt wird
- Subindex 03_h: Encoder Position des Pegelwechsels (in absoluten Benutzereinheiten aus 6064_h)
- Subindex 04_h: Encoder Position des Pegelwechsels

3250h Digital Outputs Control

Funktion

Mit diesem Objekt lassen sich die digitalen Ausgänge steuern , wie in Kapitel " Digitale Ein- und Ausgänge" beschrieben .

Dabei gilt für alle Subindizes:

- Bit 0 bis 15 steuern die Spezialfunktionen.
- Bit 16 bis 31 steuern die Pegel der Ausgänge.

Objektbeschreibung

Index	3250_h
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" Firmware Version FIR-v1446: Eintrag "Name" geändert von "Special Function Enable" auf "No Function". Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 6 auf 9. 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	09_h

Subindex	01_h
Name	No Function
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	02_h
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	03_h
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	04_h
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	05_h
Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	06_h
Name	Reserved1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	07_h
Name	Reserved2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	08_h
Name	Routing Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00000000_h ■ PD6-E891M95-M-65-3: 00000000_h ■ PD6-E891S95-M-65-3: 00000000_h ■ PD6-EB80CD-M-65-3: 00000000_h ■ PD6-EB80LD-M-65-3: 00000000_h ■ PD6-EB80MD-M-65-3: 00000000_h ■ PD6-EB80SD-M-65-3: 00000000_h ■ PD6-E891S95-MS-65-3: 00000001_h ■ PD6-E891M95-MS-65-3: 00000001_h ■ PD6-E891L95-MS-65-3: 00000001_h ■ PD6-EB80SD-MS-65-3: 00000001_h ■ PD6-EB80MD-MS-65-3: 00000001_h ■ PD6-EB80LD-MS-65-3: 00000001_h ■ PD6-EB80CD-MS-65-3: 00000001_h

Subindex	09_h
Name	Enable Mask [Bit0=StatusLed, Bit1=ErrorLed]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$FFFFFFFF_h$

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Ohne Funktion.
- 02_h: Mit diesem Subindex wird die Logik invertiert (von Öffner-Logik auf Schließer-Logik).
- 03_h: Mit diesem Subindex wird der Ausgangswert erzwungen, wenn das Bit den Wert "1" hat. Der Pegel des Ausganges wird in Subindex 4_h festgelegt.
- 04_h: Mit diesem Subindex wird der am Ausgang anzulegende Pegel definiert. Der Wert "0" liefert am digitalen Ausgang einen logischen Low-Pegel, der Wert "1" entsprechend einen logischen High-Pegel.
- 05_h: In diesen Subindex wird die an die Ausgänge gelegte Bitkombination abgelegt.
- 08_h: Wird der Subindex auf "1" gesetzt, wird das *Output Routing* aktiviert.

HINWEIS

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

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-v2538-B1077563: Eintrag "Name" geändert von "Control Bit Of 60FEh:1h And Source For Brake Output" auf "Brake Output: Signal In High Byte, Bit Number Of 60FEh:1h In Low Byte". Firmware Version FIR-v2538-B1077563: Eintrag "Name" geändert von "Control Bit Of 60FEh:1h And Source For Output #1" auf "Output #1: Signal In High Byte, Bit Number Of 60FEh:1h In Low Byte". Firmware Version FIR-v2538-B1077563: Eintrag "Name" geändert von "Control Bit Of 60FEh:1h And Source For Output #2" auf "Output #2: Signal In High Byte, Bit Number Of 60FEh:1h In Low Byte".

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	Brake Output: Signal In High Byte, Bit Number Of 60FEh:1h In Low Byte
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	1080_h

Subindex	02_h
Name	Output #1: Signal In High Byte, Bit Number Of 60FEh:1h In Low Byte
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0090_h

Subindex	03_h
Name	Output #2: Signal In High Byte, Bit Number Of 60FEh:1h In Low Byte
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0091_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	01_h

Subindex	01_h
Name	Analog Input #1 Value
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:

- 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	01_h

Subindex	01_h
Name	Analog Input #1 Offset
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	01_h

Subindex	01_h
Name	Analog Input #1 Numerator
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001_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	01_h

Subindex	01_h
Name	Analog Input #1 Denominator
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001_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: Tuning
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.

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.

33A0h Feedback Incremental A/B/I 1

Funktion

Enthält Konfigurationswerte für den ersten inkrementalen Encoder. Die Werte werden vom Auto-Setup ermittelt.

Objektbeschreibung

Index	$33A0_h$
Objektname	Feedback Incremental A/B/I 1
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

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

Subindex	01_h
Name	Configuration
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000_h

Subindex	02_h
Name	Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 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.
  - Bit 15: Wert = "1": der Encoder ist ein Singleturn-Absolut-Encoder.
- 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.

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: Tuning
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	$0D_h$

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	15_h

Subindex	06_h
Name	Baud Rate
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$00280DE8_h$

Subindex	07_h
Name	Position Bitmask Low
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$001FFFFE_h$

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	00000000_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	00000000_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:
  21 MHz, 10,5 MHz, 5,25 MHz, 2,625 MHz, 1,3125 MHz, 656,25 KHz, 328,125 KHz, 164,0625 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".

3501h EtherNetIP Rx PDO Mapping

Funktion

In dieses Objekt können die Objekte für das RX Mapping geschrieben werden.

HINWEIS

Um das Mapping ändern zu können, müssen Sie es zuerst deaktivieren, indem Sie den Subindex 0_h auf "0" setzen.

Nach dem Schreiben der Objekte in die jeweiligen Subindizes, tragen Sie die Anzahl der gemappten Objekte in den Subindex 0_h ein.

Objektbeschreibung

Index	3501_h
Objektname	EtherNetIP Rx PDO Mapping
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662
Änderungshistorie	Firmware Version FIR-v2139-B1019507: Eintrag "Name" geändert von "Highest Sub-index Supported" auf "Number Of Entries".

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20_h

Subindex	01_h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	22910120_h

Subindex	02_h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	22910210_h

Subindex	03_h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	22910308_h

Subindex	04_h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	22910408_h

Subindex	05_h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$60FE0120_h$

Subindex	06_h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000120_h

Subindex	07_h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000220_h

Subindex	08_h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000320_h

Subindex	09_h
Name	9th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000420_h

Subindex	$0A_h$
Name	10th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000520_h

Subindex	$0B_h$
Name	11th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000620_h

Subindex	$0C_h$
Name	12th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000720_h

Subindex	$0D_h$
Name	13th Object To Be Mapped
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000820_h

Subindex	$0E_h$
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24000920_h

Subindex	$0F_h$
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$24000A20_h$

Subindex	10_h
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$24000B20_h$

Subindex	11_h
Name	17th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$24000C20_h$

Subindex	12_h
Name	18th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	$24000D20_h$

Subindex	13_h
Name	19th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$24000E20_h$

Subindex	14_h
Name	20th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$24000F20_h$

Subindex	15_h
Name	21th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001020_h

Subindex	16_h
Name	22th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001120_h

Subindex	17_h
Name	23th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001220_h

Subindex	18_h

Name	24th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001320_h

Subindex	19_h
Name	25th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001420_h

Subindex	$1A_h$
Name	26th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001520_h

Subindex	$1B_h$
Name	27th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001620_h

Subindex	$1C_h$
Name	28th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001720_h

Subindex	$1D_h$
Name	29th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001820_h

Subindex	$1E_h$
Name	30th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24001920_h

Subindex	$1F_h$
Name	31th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$24001A20_h$

Subindex	20_h
Name	32th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$24001B20_h$

3502h MODBUS Rx PDO Mapping

Funktion

In dieses Objekt können die Objekte für das RX Mapping geschrieben werden.

HINWEIS

Um das Mapping ändern zu können, müssen Sie es zuerst deaktivieren, indem Sie den Subindex 0_h auf "0" setzen.

Nach dem Schreiben der Objekte in die jeweiligen Subindizes, tragen Sie die Anzahl der gemappten Objekte in den Subindex 0_h ein.

Objektbeschreibung

Index	3502_h
Objektname	MODBUS Rx PDO Mapping

Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662
Änderungshistorie	Firmware Version FIR-v1738-B505321: Eintrag "Object Name" geändert von "MODBUS Rx PDO-Mapping" auf "MODBUS Rx PDO Mapping".

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	07_h

Subindex	01_h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60400010_h

Subindex	02_h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00050008_h

Subindex	03_h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	60600008_h

Subindex	04_h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$607A0020_h$

Subindex	05_h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60810020_h

Subindex	06_h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60420010_h

Subindex	07_h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$60FE0120_h$

Subindex	08_h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	09_h

Name	9th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0A_h$
Name	10th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0B_h$
Name	11th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0C_h$
Name	12th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0D_h$
Name	13th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0E_h$
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0F_h$
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	10_h
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

3601h EtherNetIP Tx PDO Mapping

Funktion

In dieses Objekt können die Objekte für das TX Mapping geschrieben werden.

HINWEIS

Um das Mapping ändern zu können, müssen Sie es zuerst deaktivieren, indem Sie den Subindex 0_h auf "0" setzen.

Nach dem Schreiben der Objekte in die jeweiligen Subindizes, tragen Sie die Anzahl der gemappten Objekte in den Subindex 0_h ein.

Objektbeschreibung

Index	3601_h
Objektname	EtherNetIP Tx PDO Mapping
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662

Änderungshistorie	Firmware Version FIR-v2139-B1019507: Eintrag "Name" geändert von "Highest Sub-index Supported" auf "Number Of Entries".

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20_h

Subindex	01_h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	22920110_h

Subindex	02_h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$603F0010_h$

Subindex	03_h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	22920220_h

Subindex	04_h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	60640020_h

Subindex	05_h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$606C0020_h$

Subindex	06_h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$60FD0020_h$

Subindex	07_h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60770010_h

Subindex	08_h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	32200110_h

Subindex	09_h
Name	9th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$60F40020_h$

Subindex	$0A_h$

Name	10th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000120_h

Subindex	$0B_h$
Name	11th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000220_h

Subindex	$0C_h$
Name	12th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000320_h

Subindex	$0D_h$
Name	13th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000420_h

Subindex	$0E_h$
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000520_h

Subindex	$0F_h$
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000620_h

Subindex	10_h
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000720_h

Subindex	11_h
Name	17th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000820_h

Subindex	12_h
Name	18th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25000920_h

Subindex	13_h
Name	19th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$25000A20_h$

Subindex	14_h
Name	20th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$25000B20_h$

Subindex	15_h
Name	21th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$25000C20_h$

Subindex	16_h
Name	22th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$25000D20_h$

Subindex	17_h
Name	23th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$25000E20_h$

Subindex	18_h
Name	24th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$25000F20_h$

Subindex	19_h
Name	25th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25001020_h

Subindex	$1A_h$
Name	26th Object To Be Mapped
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25001120_h

Subindex	$1B_h$
Name	27th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25001220_h

Subindex	$1C_h$
Name	28th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25001320_h

Subindex	$1D_h$
Name	29th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25001420_h

Subindex	$1E_h$
Name	30th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25001520_h

Subindex	$1F_h$
Name	31th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	25001620_h

Subindex	20_h
Name	32th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	25001720_h

3602h MODBUS Tx PDO Mapping

Funktion

In dieses Objekt können die Objekte für das TX Mapping geschrieben werden.

HINWEIS

Um das Mapping ändern zu können, müssen Sie es zuerst deaktivieren, indem Sie den Subindex 0_h auf "0" setzen.

Nach dem Schreiben der Objekte in die jeweiligen Subindizes, tragen Sie die Anzahl der gemappten Objekte in den Subindex 0_h ein.

Objektbeschreibung

Index	3602_h
Objektname	MODBUS Tx PDO Mapping
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662
Änderungshistorie	Firmware Version FIR-v1738-B505321: Eintrag "Object Name" geändert von "MODBUS Tx PDO-Mapping" auf "MODBUS Tx PDO Mapping".

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	06_h

Subindex	01_h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60410010_h

Subindex	02_h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00050008_h

Subindex	03_h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60610008_h

Subindex	04_h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60640020_h

Subindex	05_h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60440010_h

Subindex	06_h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$60FD0020_h$

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

Subindex	09_h
Name	9th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0A_h$
Name	10th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0B_h$
Name	11th Object To Be Mapped
Datentyp	UNSIGNED32

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

Subindex	$0C_h$
Name	12th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0D_h$
Name	13th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0E_h$
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	$0F_h$
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	10_h
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000000_h

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
-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.

4012h HW Information

Funktion

Dieses Objekt zeigt Informationen über die Hardware an.

Objektbeschreibung

Index	4012_h
Objektname	HW Information
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

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

Subindex	01_h
Name	EEPROM Size In Bytes
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

Subindex 01: Zeigt die Größe des angeschlossenen EEPROMS in Bytes an. Der Wert "0" bedeutet, dass kein EEPROM angeschlossen ist.

4013h HW Configuration

Funktion

Mit diesem Objekt kann man bestimmte Hardware-Konfigurationen einstellen.

Objektbeschreibung

Index	4013_h
Objektname	HW Configuration
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

Wertebeschreibung

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

Subindex	01_h
Name	HW Configuration #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

reserviert

4014h Operating Conditions

Funktion

Dieses Objekt dient zum Auslesen aktueller Umgebungswerte der Steuerung.

Objektbeschreibung

Index	4014_h
Objektname	Operating Conditions
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen". Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 02 geändert von "lesen/schreiben" auf "nur lesen". Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Temperature PCB [d?C]" auf "Temperature PCB [Celsius * 10]". Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 03 geändert von "lesen/schreiben" auf "nur lesen".

	Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 4 auf 6.

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	Voltage UB Power [mV]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	02_h
Name	Voltage UB Logic [mV]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	03_h
Name	Temperature PCB [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	04_h
Name	Temperature Motor [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	

Vorgabewert	00000000_h

Subindex	05_h
Name	Temperature Microcontroller Chip [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

Die Subindizes enthalten:

- 01_h: aktuelle Versorgungsspannung in [mV]
- 02_h: aktuelle Logikspannung in [mV]
- 03_h: aktuelle Temperatur der Steuerungsplatine in [d°C] (Zehntelgrad)
- 04_h: reserviert
- 05_h: 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 Spezielle Fahrmodi.

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.

4016h Factory Settings

Funktion

Dieses Objekt zeigt an, ob das Auto-Setup in der Produktion ausgeführt wurde.

Objektbeschreibung

Index	4016_h
Objektname	Factory Settings
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	Firmware Version FIR-v2013-B726332: 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	01_h

Subindex	01_h
Name	Factory Autosetup Done
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00_h

Beschreibung

Für den Subindex 01_h zulässige Werte:

- Wert = "0" : Das *Auto-Setup* wurde nicht ausgeführt.
- Wert = "1" : Das *Auto-Setup* wurde ausgeführt.

4021h Ballast Configuration

Funktion

Objektbeschreibung

Index	4021_h
Objektname	Ballast Configuration
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
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	Settings [Bit0: On/Off]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

Subindex	02_h
Name	UB Power Limit [mV]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$0000F3BF_h$

Subindex	03_h
Name	UB Power Hysteresis [mV]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	$000001F4_h$

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)

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.

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.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						OMS	HALT	FR	OMS [3]			EO	QS	EV	SO

SO (Switched On)

Wert = "1": Schaltet in den Zustand "Switched on"

EV (Enable Voltage)

Wert = "1": Schaltet in den Zustand "Enable voltage"

QS (Quick Stop)

Wert = "0": Schalten in den Zustand "Quick stop"

EO (Enable Operation)

Wert = "1": Schalten in den Zustand "Enable operation"

OMS (Operation Mode Specific)

Bedeutung abhängig vom gewählten Betriebsmodus

FR (Fault Reset)

Setzt einen Fehler oder eine Warnung zurück (falls möglich)

HALT

Wert = "1": Löst einen Halt aus, gültig in folgenden Modi:

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

6041h Statusword

Funktion

Dieses Objekt liefert Informationen zum Status der CiA 402 Power State Machine.

Objektbeschreibung

Index	6041_h
Objektname	Statusword
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000_h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Teile des Objektes sind in der Funktion abhängig vom aktuell gewählten Modus. Schlagen Sie im entsprechenden Unterkapitel im Kapitel Betriebsmodi nach.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLA		OMS [2]		ILA	TARG	REM	SYNC	WARN	SOD	QS	VE	FAULT	OE	SO	RTSO

RTSO (Ready To Switch On)
Wert = "1": Steuerung befindet sich in dem Zustand "Ready To Switch On"

SO (Switched On)
Wert = "1": Steuerung befindet sich in dem Zustand "Switched On"

OE (Operation Enabled)
Wert = "1": Steuerung befindet sich in dem Zustand "Operation Enabled"

FAULT
Fehler vorgefallen (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" , außer wenn die speziellen Fahrmodi aktiviert sind)

TARG
Zielvorgabe erreicht

ILA (Internal Limit Active)
Limit überschritten

OMS (Operation Mode Specific)
Bedeutung abhängig vom gewählten Betriebsmodus

CLA (Closed Loop Active)
Wert = "1": die Steuerung befindet sich im Status *Operation enabled* und der Closed Loop ist aktiviert.

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

Statusword (6041_h)	Zustand
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on

Statusword (6041_h)	Zustand
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

6042h Vl Target Velocity

Funktion

Gibt die Zielgeschwindigkeit für den Velocity Modus in benutzerdefinierten Einheiten an.

Objektbeschreibung

Index	6042_h
Objektname	Vl Target Velocity
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$00C8_h$
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

6043h Vl Velocity Demand

Funktion

Vorgabegeschwindigkeit in benutzerdefinierten Einheiten für den Regler im Velocity Mode.

Objektbeschreibung

Index	6043_h
Objektname	Vl Velocity Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000_h
Firmware Version	FIR-v1426
Änderungshistorie	

6044h Vl Velocity Actual Value

Funktion

Gibt die aktuelle Istgeschwindigkeit im Velocity Modus in benutzerdefinierten Einheiten an.

Objektbeschreibung

Index	6044_h
Objektname	Vl Velocity Actual Value
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000_h
Firmware Version	FIR-v1426
Änderungshistorie	

6046h Vl Velocity Min Max Amount

Funktion

Mit diesem Objekt können Minimalgeschwindigkeit und Maximalgeschwindigkeit in benutzerdefinierten Einheiten eingestellt werden.

Objektbeschreibung

Index	6046_h
Objektname	Vl Velocity Min Max Amount
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	MinAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	02_h
Name	MaxAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00007530_h

Beschreibung

Subindex 1 enthält die Minimalgeschwindigkeit.

Subindex 2 enthält die Maximalgeschwindigkeit.

Wird eine Zielgeschwindigkeit (Objekt 6042_h) vom Betrag her kleiner als die Minimalgeschwindigkeit angegeben, gilt die Minimalgeschwindigkeit und das Bit 11 (Internal Limit Reached) in 6041h Statusword$_h$ wird gesetzt.

Eine Zielgeschwindigkeit größer als die Maximalgeschwindigkeit setzt die Geschwindigkeit auf die Maximalgeschwindigkeit und das Bit 11 (Internal Limit Reached) in 6041h Statusword$_h$ wird gesetzt.

6048h Vl Velocity Acceleration

Funktion

Setzt die Beschleunigungsrampe im Velocity Mode (siehe Velocity).

Objektbeschreibung

Index	6048_h
Objektname	Vl Velocity Acceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
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	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$000001F4_h$

Subindex	02_h
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001_h

Beschreibung

Die Beschleunigung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01_h: enthält die Geschwindigkeitsänderung.

Subindex 02_h: enthält die Zeitänderung.

6049h Vl Velocity Deceleration

Funktion

Setzt die Verzögerung (Bremsrampe) im Velocity Mode (siehe Velocity).

Objektbeschreibung

Index	6049_h
Objektname	Vl Velocity Deceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
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	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$000001F4_h$

Subindex	02_h
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001_h

Beschreibung

Die Verzögerung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01_h: enthält die Geschwindigkeitsänderung.

Subindex 02_h: enthält die Zeitänderung.

604Ah Vl Velocity Quick Stop

Funktion

Dieses Objekt definiert die Verzögerung (Bremsrampe), wenn im Velocity Mode der Quick Stop-Zustand eingeleitet wird.

Objektbeschreibung

Index	$604A_h$
Objektname	Vl Velocity Quick Stop
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
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	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388_h

Subindex	02_h
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001_h

Beschreibung

Die Verzögerung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01_h: enthält die Geschwindigkeitsänderung.

Subindex 02_h: enthält die Zeitänderung.

604Ch VI Dimension Factor

Funktion

Hier wird die Einheit der Geschwindigkeitsangaben für die Objekte festgelegt, welche den Velocity Mode betreffen.

Objektbeschreibung

Index	$604C_h$
Objektname	VI Dimension Factor
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation

Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

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

Subindex	01_h
Name	Vl Dimension Factor Numerator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001_h

Subindex	02_h
Name	Vl Dimension Factor Denominator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001_h

Beschreibung

Der Subindex 1 enthält den Zähler (Multiplikator) und der Subindex 2 den Nenner (Divisor), mit dem interne Geschwindigkeitsangaben in Umdrehungen pro Minute verrechnet werden. Wird z.B. Subindex 1 auf den Wert "60" und Subindex 2 auf den Wert "1" eingestellt, erfolgt die Geschwindigkeitsangabe in Umdrehungen pro Sekunde (60 Umdrehungen pro 1 Minute).

605Ah Quick Stop Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der CiA 402 Power State Machine in den Zustand *Quick Stop active*.

Objektbeschreibung

Index	$605A_h$
Objektname	Quick Stop Option Code

Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0002_h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert in Objekt $605A_h$	Beschreibung
0	Soforthalt mit anschließendem Zustandswechsel in *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	0001_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 Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der CiA 402 Power State Machine vom Zustand *Operation enabled* in den Zustand *Switched on*.

Objektbeschreibung

Index	$605C_h$
Objektname	Disable Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001_h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert 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 Option Code

Funktion

Das Objekt enthält die auszuführende Aktion, wie der Motor im Fehlerfall zum Stillstand gebracht werden soll.

Objektbeschreibung

Index	$605E_h$
Objektname	Fault Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0002_h
Firmware Version	FIR-v1426

Änderungshistorie	

Beschreibung

Wert 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
2	Velocity Mode
3	Profile Velocity Mode
4	Profile Torque Mode
5	Reserved
6	Homing Mode
7	Interpolated Position Mode

Modus	Beschreibung
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 $608F_h$ 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	0000_h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt $203B_h$:01.

Das Minimum von 6073_h und 6072_h wird als Limit für das Drehmoment in 6071_h verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031_h) nicht übersteigen.

6072h Max Torque

Funktion

Das Objekt beschreibt das maximale Drehmoment für den Profile Torque und Cyclic Synchronous Torque Modus in Promille des Nenndrehmoments.

Objektbeschreibung

Index	6072_h
Objektname	Max Torque
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0064_h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt $203B_h$:01.

Das Minimum von 6073_h und 6072_h wird als Limit für das Drehmoment in 6071_h verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031_h) nicht übersteigen.

6073h Max Current

Funktion

Enthält den Maximalstrom in Promille des eingestellten Nennstroms. Wird durch den maximalen Motorstrom (2031_h) begrenzt. Siehe auch I2t Motor-Überlastungsschutz.

HINWEIS

Für Schrittmotoren wird nur der Nennstrom und kein Maximalstrom angegeben. Daher sollte der Wert von 6073_h den Wert 1000 (100%) in der Regel nicht überschreiten.

Objektbeschreibung

Index	6073_h
Objektname	Max Current
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben

PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$03E8_h$
Firmware Version	FIR-v1825-B577172
Änderungshistorie	

Beschreibung

Der Maximalstrom wird in Promille des Nennstroms wie folgt berechnet:

(6073_h*$203B_h$:01)/1000

Der Maximalstrom bestimmt:

- den Maximalstrom für den I2t Motor-Überlastungsschutz,
- den Sollstrom im *Open Loop*-Betrieb.

6074h Torque Demand

Funktion

Momentaner vom Rampengenerator geforderter Drehmomentsollwert in Promille des Nenndrehmoments für den internen Regler.

Objektbeschreibung

Index	6074_h
Objektname	Torque Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000_h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt $203B_h$:01.

Das Minimum von 6073_h und 6072_h wird als Limit für das Drehmoment in 6071_h verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031_h) nicht übersteigen.

6075h Motor Rated Current

Funktion

Enthält den in $203B_h$:01_h eingetragen Nennstrom in mA.

6077h Torque Actual Value

Funktion

Dieses Objekt zeigt den aktuellen Drehmomentwert in Promille des Nenndrehmoments für den internen Regler.

Objektbeschreibung

Index	6077_h
Objektname	Torque Actual Value
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000_h
Firmware Version	FIR-v1540
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt $203B_h$:01.

Das Minimum von 6073_h und 6072_h wird als Limit für das Drehmoment in 6071_h verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031_h) nicht übersteigen.

607Ah Target Position

Funktion

Dieses Objekt gibt die Zielposition in benutzerdefinierten Einheiten für den Profile Position und Cyclic Synchronous Position Modus an.

Objektbeschreibung

Index	$607A_h$
Objektname	Target Position
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$00000FA0_h$
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.

7	6	5	4	3	2	1	0
POS	VEL						

VEL (Velocity)

Umkehr der Drehrichtung in folgenden Modi:

- Profile Velocity Mode
- Cyclic Synchronous Velocity Mode

POS (Position)

Umkehr der Drehrichtung in folgenden Modi:

- Profile Position Mode
- Cyclic Synchronous Position Mode

TIPP

Sie können ein Invertieren des Drehfeldes erzwingen, dass alle Betriebsmodi betrifft. Siehe Objekt 3212_h:02_h.

607Fh Max Profile Velocity

Funktion

Gibt die maximale Geschwindigkeit in benutzerdefinierten Einheiten an, für die Mod i Profile Torque, Profile Position , Interpolated Position Mode (nur wenn der Closed Loop aktiviert ist) 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". Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "INTEGER16" auf "UNSIGNED32". 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". 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: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	■ PD6-E891L95-M-65-3: 00007530_h ■ PD6-E891M95-M-65-3: 00007530_h ■ PD6-E891S95-M-65-3: 00007530_h ■ PD6-EB80CD-M-65-3: $0000109A_h$ ■ PD6-EB80LD-M-65-3: $000010CC_h$ ■ PD6-EB80MD-M-65-3: $0000109A_h$ ■ PD6-EB80SD-M-65-3: 00001004_h ■ PD6-E891S95-MS-65-3: 00007530_h ■ PD6-E891M95-MS-65-3: 00007530_h ■ PD6-E891L95-MS-65-3: 00007530_h ■ PD6-EB80SD-MS-65-3: 00001004_h ■ PD6-EB80MD-MS-65-3: $0000109A_h$ ■ PD6-EB80LD-MS-65-3: $000010CC_h$ ■ PD6-EB80CD-MS-65-3: $0000109A_h$
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning". Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Maximum Speed" auf "Max Motor Speed". Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "RX-PDO". Firmware Version FIR-v1748-B538662: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Bewegung". Firmware Version FIR-v1825-B577172: Eintrag "Speicherbar" geändert von "ja, Kategorie: Bewegung" auf "ja, Kategorie: Tuning". Firmware Version FIR-v1825-B577172: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Bewegung".

6081h Profile Velocity

Funktion

Gibt die maximale Fahrgeschwindigkeit in benutzerdefinierten Einheiten an.

Objektbeschreibung

Index	6081_h
Objektname	Profile Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	$000001F4_h$
Firmware Version	FIR-v1426
Änderungshistorie	

6082h End Velocity

Funktion

Gibt die Geschwindigkeit am Ende der gefahrenen Rampe in benutzerdefinierten Einheiten an.

Objektbeschreibung

Index	6082_h
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	00000064_h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt $203B_h$:01.

Das Minimum von 6073_h und 6072_h wird als Limit für das Drehmoment in 6071_h verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031_h) nicht übersteigen.

608Fh Position Encoder Resolution

Funktion

Enthält die physikalische Auflösung (siehe Objekte $60E6_h$/ $60EB_h$) des Encoders/Sensors, der für die Positionsregelung verwendet wird (siehe 3203h Feedback Selection).

Objektbeschreibung

Index	$608F_h$
Objektname	Position Encoder Resolution
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Tuning
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning". Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "nein" auf "RX-PDO". Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "RX-PDO". Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".

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	Encoder Increments
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	00000000_h

Subindex	02_h
Name	Motor Revolutions
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001_h

Beschreibung

Position Encoder Resolution = Encoder Increments ($608F_h$:01_h) / Motor Revolutions ($608F_h$:02_h)

6090h Velocity Encoder Resolution

Funktion

Enthält die physikalische Auflösung (siehe Objekte $60E6_h$/ $60EB_h$) des Encoders/Sensors, der für die Drehzahlregelung verwendet wird (siehe 3203h Feedback Selection).

Objektbeschreibung

Index	6090_h
Objektname	Velocity Encoder Resolution
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32". Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32". Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32". Firmware Version FIR-v1825-B577172: Eintrag "Datentyp" geändert von "INTEGER32" auf "UNSIGNED32". Firmware Version FIR-v1825-B577172: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED32". Firmware Version FIR-v1825-B577172: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED32". Firmware Version FIR-v1825-B577172: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".

	Firmware Version FIR-v1825-B577172: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".
	Firmware Version FIR-v1825-B577172: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".

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	Encoder Increments Per Second
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Subindex	02_h
Name	Motor Revolutions Per Second
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001_h

Beschreibung

Velocity Encoder Resolution = Encoder Increments per second (6090_h:01_h) / Motor Revolutions per second (6090_h:02_h)

6091h Gear Ratio

Funktion

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". 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 beim Losfahren und im Falle eines Quick-Stopps 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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Exponent einer Zehnerpotenz								Einheit							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserviert (00h)								reserviert (00h)							

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)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Exponent einer Zehnerpotenz								Positionseinheit							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Zeiteinheit								reserviert (00h)							

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: muss den Wert -3 halten (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". Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 06 geändert von "lesen/schreiben" auf "nur schreiben". Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation". 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	06_h

Subindex	01_h
Name	MaximumBufferSize
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

Subindex	02_h
Name	ActualBufferSize
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001_h

Subindex	03_h
Name	BufferOrganization
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00_h

Subindex	04_h
Name	BufferPosition
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001_h

Subindex	05_h
Name	SizeOfDataRecord
Datentyp	UNSIGNED8
Zugriff	nur schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04_h

Subindex	06_h
Name	BufferClear
Datentyp	UNSIGNED8
Zugriff	nur schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00_h

Beschreibung

Der Wert des Subindex 01_h enthält die 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". 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	03_h

Subindex	01_h - 03_h
Name	Additional Position Actual Value #1 - #3
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:
  Subindex n enthält die aktuelle Istposition der entsprechenden Rückführung.
  Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

60E5h Additional Velocity Actual Value

Funktion

Enthält die aktuelle Istgeschwindigkeit aller vorhandenen Rückführungen in benutzerdefinierten Einheiten.

Objektbeschreibung

Index	$60E5_h$
Objektname	Additional Velocity Actual Value
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	03_h

Subindex	01_h - 03_h
Name	Additional Velocity Actual Value #1 - #3
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:
  Subindex n enthält die aktuelle Istgeschwindigkeit der entsprechenden Rückführung.
  Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

60E6h Additional Position Encoder Resolution - Encoder Increments

Funktion

Mit diesem Objekt und mit $60EB_h$ wird die Auflösung jeder vorhandenen Rückführung berechnet.

Objektbeschreibung

Index	$60E6_h$
Objektname	Additional Position Encoder Resolution - Encoder Increments
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	

Firmware Version	FIR-v1748-B538662
Änderungshistorie	

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 - 03_h
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #1 - #3
Datentyp	INTEGER32
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 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.

60E8h Additional Gear Ratio - Motor Shaft Revolutions

Funktion

In diesem Objekt und in $60ED_h$ können Sie die Getriebeübersetzung jeder vorhandenen Rückführung einstellen.

Objektbeschreibung

Index	$60E8_h$
Objektname	Additional Gear Ratio - Motor Shaft Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03_h

Subindex	01_h - 03_h
Name	Additional Gear Ratio - Motor Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h: Subindex "n" enthält die Anzahl der Motorumdrehungen für die entsprechende Rückführung. Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Getriebeübersetzung der Rückführung "n" berechnet sich wie folgt:

Gear Ratio = Motor Shaft Revolutions ($60E8_h$:n_h) / Driving Shaft Revolutions ($\underline{60ED_h}$:n_h)

60E9h Additional Feed Constant - Feed

Funktion

In diesem Objekt und in 60EE$_h$ können Sie eine Vorschubkonstante für jede vorhandene Rückführung einstellen.

Objektbeschreibung

Index	60E9$_h$
Objektname	Additional Feed Constant - Feed
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00$_h$
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03$_h$

Subindex	01$_h$ - 03$_h$
Name	Additional Feed Constant - Feed Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001$_h$

Beschreibung

Die Subindizes haben folgende Funktion:

- 00$_h$: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n$_h$: Subindex "n" enthält den Vorschub in benutzerdefinierten Einheiten für die entsprechende Rückführung.
  Subindex 01$_h$ entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Vorschubkonstante der Rückführung "n" berechnet sich wie folgt:

Feed Constant = Feed ($60E9_h$:n_h) / Driving Shaft Revolutions ($60EE_h$:n_h)

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: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-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	RX-PDO
Zulässige Werte	
Vorgabewert	03_h

Subindex	01_h - 03_h
Name	Additional Position Encoder Resolution - Motor Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h:
  Subindex n enthält die Anzahl der Motorumdrehungen der entsprechenden Rückführung.
  Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Auflösung der Rückführung "n" berechnet sich wie folgt:

Position Encoder Resolution = Encoder Increments ($60E6_h$:n_h) / Motor Revolutions ($60EB_h$:n_h)

60EDh Additional Gear Ratio - Driving Shaft Revolutions

Funktion

In diesem Objekt und in $60E8_h$ können Sie die Getriebeübersetzung jeder vorhandenen Rückführung einstellen.

Objektbeschreibung

Index	$60ED_h$
Objektname	Additional Gear Ratio - Driving Shaft Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03_h

Subindex	01_h - 03_h
Name	Additional Gear Ratio - Driving Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.

- n_h: Subindex "n" enthält die Anzahl der Umdrehungen der Abtriebsachse für die entsprechende Rückführung.
  Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Getriebeübersetzung der Rückführung "n" berechnet sich wie folgt:

Gear Ratio = Motor Shaft Revolutions ($60E8_h$:n_h) / Driving Shaft Revolutions ($60ED_h$:n_h)

60EEh Additional Feed Constant - Driving Shaft Revolutions

Funktion

In diesem Objekt und in $60E9_h$ können Sie eine Vorschubkonstante für jede vorhandene Rückführung einstellen.

Objektbeschreibung

Index	$60EE_h$
Objektname	Additional Feed Constant - Driving Shaft Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00_h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03_h

Subindex	01_h - 03_h
Name	Additional Feed Constant - Driving Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001_h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h: Subindex "n" enthält die Anzahl der Umdrehungen der Abtriebsachse für die entsprechende Rückführung.
  Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Vorschubkonstante der Rückführung "n" berechnet sich wie folgt:

Feed Constant = Feed ($60E9_h$:n_h) / Driving Shaft Revolutions ($60EE_h$:n_h)

60F2h Positioning Option Code

Funktion

Das Objekt beschreibt das Positionierverhalten im Profile Position Modus.

Objektbeschreibung

Index	$60F2_h$
Objektname	Positioning Option Code
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001_h
Firmware Version	FIR-v1446
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

Beschreibung

Derzeit werden nur nachfolgende Bits unterstützt:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS	RESERVED [3]			IP OPTION [4]				RADO [2]		RRO [2]		CIO [2]		REL. OPT. [2]	

REL. OPT. (Relative Option)

Diese Bits bestimmen das Verhalten bei relativer Drehbewegung im "Profile Position" Modus, sollte Bit 6 des Kontrollwortes 6040_h = "1" gesetzt sein.

Bit 1	Bit 0	Definition
0	0	Positionsbewegungen werden relativ zu der vorherigen (intern absoluten) Zielposition ausgeführt (jeweils relativ zu 0 falls keine 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.

Bit 1	Bit 0	Definition
1	1	Reserviert

RRO (Request-Response Option)

Diese Bits bestimmen das Verhalten bei der Übergabe des Controlwords 6040_h Bit 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.
1	1	Positionierung mit dem kürzesten Weg zur Zielposition. Falls die Differenz zwischen aktueller Position und Zielposition in einem 360° System kleiner als 180° ist, fährt die Achse in positiver Richtung.

60F4h Following Error Actual Value

Funktion

Dieses Objekt enthält den aktuellen Schleppfehler in benutzerdefinierten Einheiten.

Objektbeschreibung

Index	$60F4_h$
Objektname	Following Error Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000_h
Firmware Version	FIR-v1426
Änderungshistorie	

60F8h Max Slippage

Funktion

Definiert den maximal erlaubten Schlupffehler in benutzerdefinierten Einheiten symmetrisch zur Sollgeschwindigkeit im Modus Profile Velocity.

Objektbeschreibung

Index	$60F8_h$
Objektname	Max Slippage
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000190_h
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Beschreibung

Weicht die Istgeschwindigkeit von der Sollgeschwindigkeit so stark ab, dass der Wert (Absolutbetrag) dieses Objekts überschritten wird, wird das Bit 13 im Objekt 6041_h gesetzt. Die Abweichung muss länger andauern als die Zeit im Objekt $203F_h$.

Wird der Wert des $60F8_h$ auf "7FFFFFFF"$_h$ gesetzt, wird die Schlupffehler-Überwachung abgeschaltet.

Im Objekt 3700_h kann eine Reaktion auf den Schlupffehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt 1003_h eingetragen.

60FAh Control Effort

Funktion

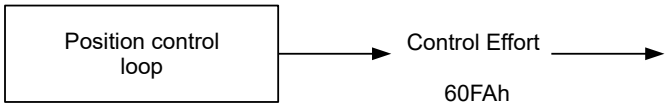
Dieses Objekt beinhaltet die Korrekturgeschwindigkeit (Stellgröße) in benutzerdefinierten Einheiten, die vom Positionsregler dem Geschwindigkeitsregler zugeführt wird.

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

Zeigt den aktuellen Vorgabewert für den Positionsregler in Encoder-Inkrementen an (siehe Reglerstruktur).

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
STO B	STO A						...	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	00000001_h

Beschreibung

Zum Schreiben der Ausgänge müssen noch die Einträge in Objekt 3250_h, Subindex 02_h bis 05_h berücksichtigt werden.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											...	OUT4	OUT3	OUT2	OUT1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															BRK

BRK (Brake)

Bit für den Bremsenausgang (falls der Controller diese Funktion unterstützt):

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	$0000002F_h$
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.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						CST	CSV	CSP	IP	HM		TQ	PV	VL	PP

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 uIP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

11.5 DHCP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

11.6 CMSIS DSP Software Library

Copyright (C) 2010 ARM Limited. All rights reserved.

11.7 FatFs

FatFs - FAT file system module include file R0.08 (C)ChaN, 2010

FatFs module is a generic FAT file system module for small embedded systems.

This is a free software that opened for education, research and commercial

developments under license policy of following trems.

Copyright (C) 2010, ChaN, all right reserved.

The FatFs module is a free software and there is NO WARRANTY.

No restriction on use. You can use, modify and redistribute it for

personal, non-profit or commercial product UNDER YOUR RESPONSIBILITY.

Redistributions of source code must retain the above copyright notice.

11.8 Protothreads

Protothread class and macros for lightweight, stackless threads in C++.

This was "ported" to C++ from Adam Dunkels' protothreads C library at: http://www.sics.se/~adam/pt/

Originally ported for use by Hamilton Jet (www.hamiltonjet.co.nz) by Ben Hoyt, but stripped down for public release. See his blog entry about it for more information: http://blog.micropledge.com/2008/07/protothreads/

Original BSD-style license

Copyright (c) 2004-2005, Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the Institute and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Institute or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

11.9 lwIP

Copyright (c) 2001-2004 Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO

EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This file is part of the lwIP TCP/IP stack.

Author: Adam Dunkels <adam@sics.se>

11.10 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.