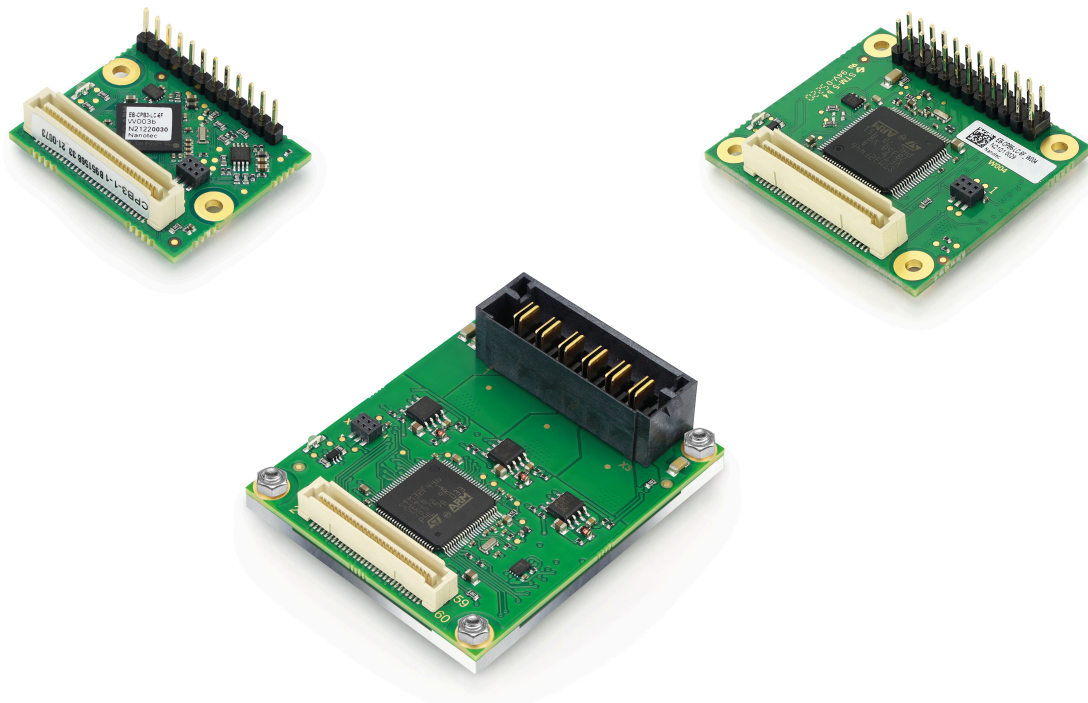


Technisches Handbuch CPB

Feldbus: Modbus RTU

Zu nutzen mit folgenden Varianten:

CPB3-1-5, CPB3-2-5, CPB6-1-5, CPB6-2-5, CPB15-5



Inhalt

1	Einleitung.....	10
1.1	Versionshinweise.....	10
1.2	Urheberrecht, Kennzeichnung und Kontakt.....	11
1.3	Bestimmungsgemäße Verwendung.....	11
1.4	Zielgruppe und Qualifikation.....	12
1.5	Gewährleistung und Haftungsausschluss.....	12
1.6	EU-Richtlinien zur Produktsicherheit.....	12
1.7	Mitgeltende Vorschriften.....	12
1.8	Verwendete Symbole.....	12
1.9	Hervorhebungen im Text.....	13
1.10	Zahlenwerte.....	13
1.11	Bits.....	13
1.12	Zählrichtung (Pfeile).....	13
2	Sicherheits- und Warnhinweise.....	15
3	Technische Daten und Anschlussbelegung.....	16
3.1	Maßzeichnungen.....	16
3.2	Umgebungsbedingungen.....	17
3.3	Elektrische Eigenschaften und technische Daten.....	17
3.4	Übertemperaturschutz.....	18
3.5	LED-Signalisierung.....	19
3.5.1	Betriebs-LED.....	19
3.6	Anschlussbelegung.....	21
3.6.1	Übersicht.....	21
3.6.2	X1 — Spannungsversorgung und Motor.....	22
3.6.3	X2 — Ein- und Ausgänge.....	23
4	Hardware-Installation.....	27
4.1	Anschließen der Steuerung.....	27
4.1.1	Integrieren der CPB in ein Motherboard.....	27
4.1.2	Schalter für Kommunikationseinstellungen anschließen.....	28
5	Inbetriebnahme.....	29
5.1	Konfiguration über USB.....	29
5.1.1	Allgemeines.....	29
5.1.2	USB-Anschluss.....	29
5.1.3	Konfigurationsdatei.....	30
5.1.4	NanoJ-Programm.....	32
5.2	Konfiguration über Modbus RTU.....	33
5.2.1	Kommunikationseinstellungen.....	33
5.3	Motordaten einstellen.....	34
5.4	Auto-Setup.....	35
5.4.1	Parameter-Ermittlung.....	36
5.4.2	Durchführung.....	36
5.4.3	Parameterspeicherung.....	38
5.5	Konfigurieren der Sensoren.....	38
5.6	Testlauf.....	40

6	Generelle Konzepte.....	41
6.1	Betriebsarten.....	41
6.1.1	Allgemein.....	41
6.1.2	Open Loop.....	42
6.1.3	Closed Loop.....	44
6.1.4	Slow Speed.....	50
6.2	CiA 402 Power State Machine.....	52
6.2.1	Zustandsmaschine.....	52
6.2.2	Verhalten beim Verlassen des Zustands <i>Operation enabled</i>	54
6.3	Benutzerdefinierte Einheiten.....	57
6.3.1	Einheiten.....	58
6.3.2	Encoderauflösung.....	59
6.3.3	Getriebeübersetzung.....	60
6.3.4	Vorschubkonstante.....	60
6.3.5	Berechnungsformeln für Benutzereinheiten.....	60
6.4	Begrenzung des Bewegungsbereichs.....	62
6.4.1	Verhalten beim Erreichen der Endschalter.....	62
6.4.2	Software-Endschalter.....	63
6.5	Zykluszeiten.....	63
7	Betriebsmodi.....	64
7.1	Profile Position.....	64
7.1.1	Übersicht.....	64
7.1.2	Setzen von Fahrbefehlen.....	65
7.1.3	Genauigkeitsverlust bei Relativbewegungen.....	69
7.1.4	Randbedingungen für eine Positionierfahrt.....	70
7.1.5	Ruck-begrenzter und nicht ruck-begrenzter Modus.....	71
7.2	Velocity.....	72
7.2.1	Beschreibung.....	72
7.2.2	Aktivierung.....	72
7.2.3	Controlword.....	72
7.2.4	Statusword.....	72
7.2.5	Objekteinträge.....	72
7.3	Profile Velocity.....	73
7.3.1	Beschreibung.....	73
7.3.2	Aktivierung.....	73
7.3.3	Controlword.....	74
7.3.4	Statusword.....	74
7.3.5	Objekteinträge.....	74
7.4	Profile Torque.....	76
7.4.1	Beschreibung.....	76
7.4.2	Aktivierung.....	77
7.4.3	Controlword.....	77
7.4.4	Statusword.....	77
7.4.5	Objekteinträge.....	77
7.5	Homing.....	78
7.5.1	Übersicht.....	78
7.5.2	Referenzfahrt-Methode.....	80
7.6	Interpolated Position Mode.....	86
7.6.1	Übersicht.....	86
7.6.2	Aktivierung.....	86
7.6.3	Controlword.....	86
7.6.4	Statusword.....	86
7.6.5	Benutzung.....	86
7.6.6	Setup.....	87
7.6.7	Operation.....	87

7.7 Cyclic Synchronous Position.....	87
7.7.1 Übersicht.....	87
7.7.2 Objekteinträge.....	88
7.8 Cyclic Synchronous Velocity.....	89
7.8.1 Übersicht.....	89
7.8.2 Objekteinträge.....	89
7.9 Cyclic Synchronous Torque.....	90
7.9.1 Übersicht.....	90
7.9.2 Objekteinträge.....	91
7.10 Takt-Richtungs-Modus.....	91
7.10.1 Beschreibung.....	91
7.10.2 Aktivierung.....	91
7.10.3 Generelles.....	91
7.10.4 Statusword.....	92
7.10.5 Unterarten des Takt-Richtungs-Modus.....	92
7.11 Auto-Setup.....	93
7.11.1 Beschreibung.....	93
7.11.2 Aktivierung.....	93
7.11.3 Controlword.....	93
7.11.4 Statusword.....	93
8 Spezielle Funktionen.....	95
8.1 Digitale Ein- und Ausgänge.....	95
8.1.1 Ein- und Ausgangsbelegung festlegen.....	95
8.1.2 Bitzuordnung.....	98
8.1.3 Digitale Eingänge.....	99
8.1.4 Digitale Ausgänge.....	104
8.1.5 Virtueller Encoderausgang.....	108
8.2 Automatische Bremsensteuerung.....	109
8.2.1 Beschreibung.....	109
8.2.2 Aktivierung und Anschluss.....	109
8.2.3 Steuerung der Bremse.....	109
8.2.4 Bremsen-PWM.....	110
8.3 Externe Ballast-Schaltung.....	111
8.3.1 Steuerung des Ballast-Widerstands.....	111
8.3.2 Ballast aktivieren.....	111
8.3.3 Ballast-Überwachung.....	112
8.4 I ² t Motor-Überlastungsschutz.....	113
8.4.1 Beschreibung.....	113
8.4.2 Objekteinträge.....	113
8.4.3 Aktivierung.....	113
8.4.4 Funktion von I ² t.....	113
8.5 Objekte speichern.....	114
8.5.1 Allgemeines.....	114
8.5.2 Kategorie: Kommunikation.....	115
8.5.3 Kategorie: Applikation.....	115
8.5.4 Kategorie: Benutzer.....	117
8.5.5 Kategorie: Bewegung.....	117
8.5.6 Kategorie: Tuning.....	117
8.5.7 Kategorie: Modbus RTU.....	117
8.5.8 Speichervorgang starten.....	118
8.5.9 Speicherung verwerfen.....	118
8.5.10 Konfiguration verifizieren.....	119
9 Modbus RTU.....	120
9.1 Modbus Modicon-Notation bei SPS.....	120
9.2 Allgemeines.....	120

9.3 Funktionscodes.....	121
9.4 Funktioncode-Beschreibungen.....	122
9.4.1 FC 3 (03 _h) Read Input Registers / FC 4 (04 _h) Read Holding Registers.....	122
9.4.2 FC 6 (06 _h) Write Single Register.....	123
9.4.3 FC 16 (10 _h) Write Multiple Registers.....	124
9.4.4 FC 17 (11 _h) Report Server ID.....	125
9.4.5 FC 23 (17 _h) Read/Write Multiple registers.....	125
9.4.6 FC 8 (08 _h) Diagnostics.....	126
9.4.7 FC 43 (2B _h) Encapsulated Interface Transport.....	131
9.4.8 FC 101 (65 _h) Read complete object dictionary.....	138
9.4.9 FC 102 (66 _h) Read complete array or record.....	142
9.4.10 Ausnahmecodes.....	145
9.5 Prozessdatenobjekte (PDO).....	145
9.5.1 Konfiguration.....	145
9.5.2 Übertragung.....	145
9.6 NanoJ-Objekte.....	147
10 Programmierung mit NanoJ.....	148
10.1 NanoJ-Programm.....	148
10.2 Mapping im NanoJ-Programm.....	152
10.3 NanoJ-Funktionen im NanoJ-Programm.....	153
10.4 Einschränkungen und mögliche Probleme.....	155
11 Objektverzeichnis Beschreibung.....	157
11.1 Übersicht.....	157
11.2 Aufbau der Objektbeschreibung.....	157
11.3 Objektbeschreibung.....	157
11.4 Wertebeschreibung.....	158
11.5 Beschreibung.....	159
1000h Device Type.....	160
1001h Error Register.....	161
1003h Pre-defined Error Field.....	162
1008h Manufacturer Device Name.....	166
1009h Manufacturer Hardware Version.....	167
100Ah Manufacturer Software Version.....	167
1010h Store Parameters.....	167
1011h Restore Default Parameters.....	171
1018h Identity Object.....	174
1020h Verify Configuration.....	176
1F50h Program Data.....	177
1F51h Program Control.....	178
1F57h Program Status.....	179
2028h MODBUS Slave Address.....	180
202Ah MODBUS RTU Baudrate.....	181
202Ch MODBUS RTU Stop Bits.....	181
202Dh MODBUS RTU Parity.....	182
2030h Pole Pair Count.....	182
2031h Max Motor Current.....	183
2034h Upper Voltage Warning Level.....	183
2035h Lower Voltage Warning Level.....	184
2036h Open Loop Current Reduction Idle Time.....	184
2037h Open Loop Current Reduction Value/factor.....	185
2038h Brake Controller Timing.....	186
2039h Motor Currents.....	188
203Ah Homing On Block Configuration.....	190
203Bh I2t Parameters.....	191
203Dh Torque Window.....	194

203Eh Torque Window Time Out.....	194
203Fh Max Slippage Time Out.....	195
2057h Clock Direction Multiplier.....	195
2058h Clock Direction Divider.....	196
205Ah Absolute Sensor Boot Value (in User Units).....	196
205Bh Clock Direction Or Clockwise/Counter Clockwise Mode.....	197
205Ch Virtual Encoder Configuration.....	197
2084h Bootup Delay.....	199
2101h Fieldbus Module Availability.....	199
2102h Fieldbus Module Control.....	200
2103h Fieldbus Module Status.....	202
2104h Additional Fieldbus Configuration.....	203
2290h PDI Control.....	205
2291h PDI Input.....	205
2292h PDI Output.....	207
2300h NanoJ Control.....	208
2301h NanoJ Status.....	209
2302h NanoJ Error Code.....	210
230Eh Timer.....	211
230Fh Uptime Seconds.....	212
2310h NanoJ Input Data Selection.....	212
2320h NanoJ Output Data Selection.....	214
2330h NanoJ In/output Data Selection.....	215
2400h NanoJ Inputs.....	216
2410h NanoJ Init Parameters.....	217
2500h NanoJ Outputs.....	218
2600h NanoJ Debug Output.....	219
2701h Customer Storage Area.....	220
2800h Bootloader And Reboot Settings.....	221
3202h Motor Drive Submode Select.....	222
3203h Feedback Selection.....	224
3204h Feedback Mapping.....	226
320Dh Torque Of Inertia Factor.....	227
3210h Motor Drive Parameter Set.....	229
3212h Motor Drive Flags.....	233
321Ah Current Controller Parameters.....	235
321Bh Velocity Controller Parameters.....	236
321Ch Position Controller Parameters.....	237
321Dh Pre-control.....	238
321Eh Voltage Limit.....	239
3220h Analog Input Digits.....	240
3221h Analog Inputs Control.....	241
3225h Analog Input Switches.....	242
3240h Digital Inputs Control.....	243
3241h Digital Input Capture.....	246
3242h Digital Input Routing.....	248
3243h Digital Input Homing Capture.....	255
3250h Digital Outputs Control.....	257
3252h Digital Output Routing.....	260
3260h Pwm Output 0.....	265
3261h Pwm Output 1.....	266
3273h Generic SPI Hardware Configuration.....	267
3274h Generic SPI Mosi Data.....	268
3275h Generic SPI Miso Data.....	269
3320h Analog Input Values.....	269
3321h Analog Input Offsets.....	271
3322h Analog Input Numerators.....	271
3323h Analog Input Denominators.....	272
3380h Feedback Sensorless.....	273

3390h Feedback Hall.....	275
33A0h Feedback Incremental A/B/I 1.....	278
33B0h Feedback SSI 1.....	279
33B1h Feedback SSI 2.....	283
3502h MODBUS Rx PDO Mapping.....	287
3602h MODBUS Tx PDO Mapping.....	291
3700h Deviation Error Option Code.....	295
3701h Limit Switch Error Option Code.....	296
4012h HW Information.....	297
4013h HW Configuration.....	298
4014h Operating Conditions.....	299
4021h Ballast Configuration.....	301
4040h Drive Serial Number.....	303
4041h Device Id.....	304
4042h Bootloader Infos.....	304
603Fh Error Code.....	305
6040h Controlword.....	306
6041h Statusword.....	307
6042h VI Target Velocity.....	309
6043h VI Velocity Demand.....	309
6044h VI Velocity Actual Value.....	309
6046h VI Velocity Min Max Amount.....	310
6048h VI Velocity Acceleration.....	311
6049h VI Velocity Deceleration.....	312
604Ah VI Velocity Quick Stop.....	313
604Ch VI Dimension Factor.....	314
605Ah Quick Stop Option Code.....	315
605Bh Shutdown Option Code.....	316
605Ch Disable Option Code.....	317
605Dh Halt Option Code.....	317
605Eh Fault Option Code.....	318
6060h Modes Of Operation.....	319
6061h Modes Of Operation Display.....	320
6062h Position Demand Value.....	320
6063h Position Actual Internal Value.....	321
6064h Position Actual Value.....	321
6065h Following Error Window.....	322
6066h Following Error Time Out.....	322
6067h Position Window.....	323
6068h Position Window Time.....	323
606Bh Velocity Demand Value.....	324
606Ch Velocity Actual Value.....	324
606Dh Velocity Window.....	325
606Eh Velocity Window Time.....	325
606Fh Velocity Threshold.....	326
6070h Velocity Threshold Time.....	326
6071h Target Torque.....	327
6072h Max Torque.....	328
6073h Max Current.....	328
6074h Torque Demand.....	329
6075h Motor Rated Current.....	329
6077h Torque Actual Value.....	330
607Ah Target Position.....	330
607Bh Position Range Limit.....	331
607Ch Home Offset.....	332
607Dh Software Position Limit.....	332
607Eh Polarity.....	333
607Fh Max Profile Velocity.....	334
6080h Max Motor Speed.....	335

6081h Profile Velocity.....	335
6082h End Velocity.....	336
6083h Profile Acceleration.....	336
6084h Profile Deceleration.....	337
6085h Quick Stop Deceleration.....	337
6086h Motion Profile Type.....	337
6087h Torque Slope.....	338
608Fh Position Encoder Resolution.....	339
6090h Velocity Encoder Resolution.....	340
6091h Gear Ratio.....	341
6092h Feed Constant.....	342
6096h Velocity Factor.....	343
6097h Acceleration Factor.....	344
6098h Homing Method.....	346
6099h Homing Speed.....	346
609Ah Homing Acceleration.....	347
60A2h Jerk Factor.....	348
60A4h Profile Jerk.....	349
60A8h SI Unit Position.....	350
60A9h SI Unit Velocity.....	351
60B0h Position Offset.....	352
60B1h Velocity Offset.....	352
60B2h Torque Offset.....	353
60C1h Interpolation Data Record.....	353
60C2h Interpolation Time Period.....	354
60C4h Interpolation Data Configuration.....	355
60C5h Max Acceleration.....	357
60C6h Max Deceleration.....	358
60E4h Additional Position Actual Value.....	358
60E5h Additional Velocity Actual Value.....	359
60E6h Additional Position Encoder Resolution - Encoder Increments.....	360
60E8h Additional Gear Ratio - Motor Shaft Revolutions.....	361
60E9h Additional Feed Constant - Feed.....	363
60EBh Additional Position Encoder Resolution - Motor Revolutions.....	364
60EDh Additional Gear Ratio - Driving Shaft Revolutions.....	365
60EEh Additional Feed Constant - Driving Shaft Revolutions.....	366
60F2h Positioning Option Code.....	367
60F4h Following Error Actual Value.....	369
60F8h Max Slippage.....	369
60FAh Control Effort.....	370
60FCh Position Demand Internal Value.....	370
60FDh Digital Inputs.....	371
60FEh Digital Outputs.....	372
60FFh Target Velocity.....	373
6502h Supported Drive Modes.....	373
6503h Drive Catalogue Number.....	374
6505h Http Drive Catalogue Address.....	375
12 Copyrights.....	376
12.1 Einführung.....	376
12.2 AES.....	376
12.3 MD5.....	376
12.4 uIP.....	377
12.5 DHCP.....	377
12.6 CMSIS DSP Software Library.....	377
12.7 FatFs.....	377
12.8 Protothreads.....	378
12.9 lwIP.....	378

12.10 littlefs..... 379

1 Einleitung

Die Produkte der *CPB*-Baureihe sind Motorsteuerungen im Steckmodulformat zur Integration in Ihre eigenen Entwicklungen. Die Varianten *CPB3-...* und *CPB6-...* können sowohl BLDC- als auch Schrittmotoren ansteuern, die *CPB15* ist nur für BLDC-Motoren geeignet.

Dieses Handbuch beschreibt die Integration der *CPB* in Ihr Motherboard und die Funktionen der Steuerung. Weiterhin wird gezeigt, wie Sie die Steuerung über die Kommunikationsschnittstelle ansprechen und programmieren können.

Weitere Informationen zum Produkt finden Sie auf www.nanotec.de.

1.1 Versionshinweise

CPB3

Version Handbuch	Datum	Änderungen	Version Firmware	Version Hardware
1.0.0	01/2022	erste Veröffentlichung	FIR-v2139	W003
1.1.0	07/2022	<ul style="list-style-type: none"> ■ Änderungen und Ergänzungen im Kapitel <u>Digitale Ein- und Ausgänge</u>: <ul style="list-style-type: none"> □ Änderungen in <u>Ein- und Ausgangsbelegung festlegen</u>. □ Neue Funktion <u>Generic SPI</u>. 	FIR-v2213	W003
1.1.1	09/2022	Korrektur des Routing-Beispiels im Kapitel <u>Digitale Eingänge</u> .	FIR-v2213	W003
1.1.2	12/2024	Korrektur der <u>.Anschlussbelegung</u> , Pin 9 und 10 von X2 waren vertauscht.	FIR-v2213	W003

CPB6

Version Handbuch	Datum	Änderungen	Version Firmware	Version Hardware
1.0.0	01/2022	erste Veröffentlichung	FIR-v2139	W004
1.1.0	07/2022	<ul style="list-style-type: none"> ■ Änderungen und Ergänzungen im Kapitel <u>Digitale Ein- und Ausgänge</u>: <ul style="list-style-type: none"> □ Änderungen in <u>Ein- und Ausgangsbelegung festlegen</u>. □ Neue Funktion <u>Generic SPI</u>. 	FIR-v2213	W004
1.1.1	09/2022	Korrektur des Routing-Beispiels im Kapitel <u>Digitale Eingänge</u> .	FIR-v2213	W004
1.1.2	12/2024	Korrektur der <u>.Anschlussbelegung</u> , Pin 9 und 10 von X2 waren vertauscht.	FIR-v2213	W004

CPB15

Version Handbuch	Datum	Änderungen	Version Firmware	Version Hardware
1.0.0	01/2022	erste Veröffentlichung	FIR-v2139	W005

Version Handbuch	Datum	Änderungen	Version Firmware	Version Hardware
1.1.0	07/2022	<ul style="list-style-type: none"> ■ Änderungen und Ergänzungen im Kapitel <u>Digitale Ein- und Ausgänge</u>: <ul style="list-style-type: none"> □ Änderungen in <u>Ein- und Ausgangsbelegung festlegen</u>. □ Neue Funktion <u>Generic SPI</u>. 	FIR-v2213	W005
1.1.1	09/2022	Korrektur des Routing-Beispiels im Kapitel <u>Digitale Eingänge</u> .	FIR-v2213	W005
1.1.2	12/2024	Korrektur der <u>Anschlussbelegung</u> , Pin 9 und 10 von X2 waren vertauscht.	FIR-v2213	W005

1.2 Urheberrecht, Kennzeichnung und Kontakt

© 2013 – 2022 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.

1.3 Bestimmungsgemäße Verwendung

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

Die Steuerung muss über ein geeignetes Motherboard an Motoren angeschlossen werden. Die Systemgrenze der *CPB* endet an den Steckverbindern.

Verwenden Sie das Produkt bestimmungsgemäß innerhalb der durch die technischen Daten definierten Grenzen (siehe insbesondere Elektrische Eigenschaften und technische Daten) 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 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.

1.5 Gewährleistung und Haftungsausschluss

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

1.6 EU-Richtlinien zur Produktsicherheit

Folgende EU-Richtlinien wurden beachtet:

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

1.7 Mitgeltende Vorschriften

Neben diesem technischen Handbuch sind folgende Vorschriften zu beachten:

- Unfallverhütungsvorschriften
- örtliche Vorschriften zur Arbeitssicherheit

1.8 Verwendete Symbole

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

VORSICHT!



Der Hinweis VORSICHT verweist auf eine möglicherweise gefährliche Situation.

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

- ▶ Beschreibt, wie Sie die gefährliche Situation vermeiden.

HINWEIS



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.

1.9 Hervorhebungen im Text

Im Dokument gelten folgende Konventionen:

Ein unterstrichener Text markiert Querverweise und Hyperlinks:

- Folgende Bits im Objekt `6041h` (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 `2300h`, Bit 0 = "1".
- Wird in diesem Zustand bereits Haltemoment benötigt, muss in das `3212h:01h` 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:

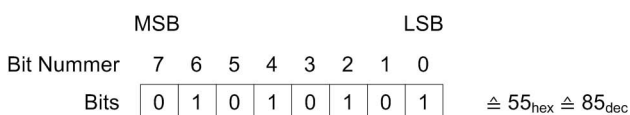
<Index>:<Subindex>

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

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

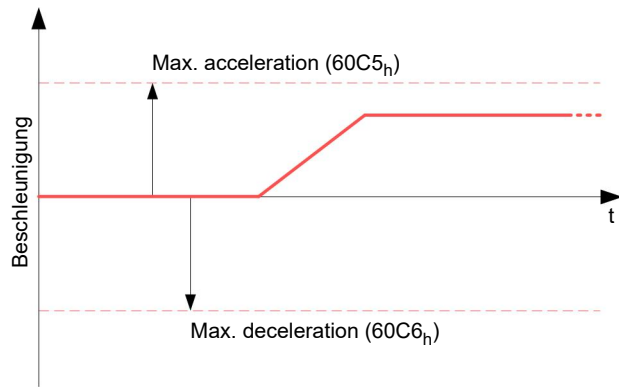
1.11 Bits

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



1.12 Zählrichtung (Pfeile)

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



2 Sicherheits- und Warnhinweise

HINWEIS



Beschädigung der Steuerung!

Ein Wechsel der Verdrahtung im Betrieb kann die Steuerung beschädigen.

- ▶ Ändern Sie die Verdrahtung nur im spannungsfreien Zustand und warten Sie nach dem Abschalten, bis sich die Kondensatoren entladen haben.

HINWEIS



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!

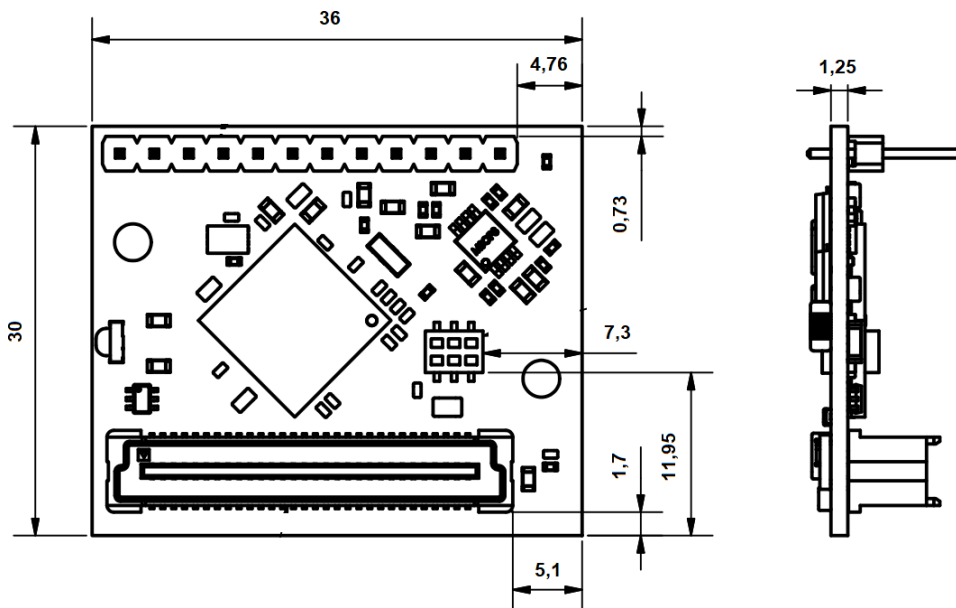
- ▶ Installieren Sie eine Leitungsschutzeinrichtung (Sicherung) in der Zuleitung.

3 Technische Daten und Anschlussbelegung

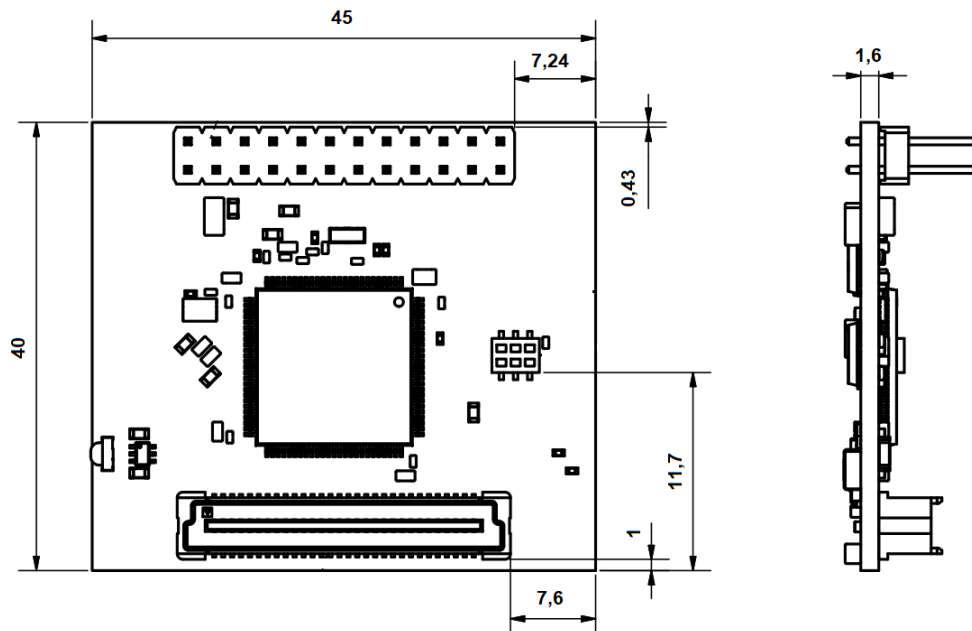
3.1 Maßzeichnungen

Alle Maße sind in Millimetern.

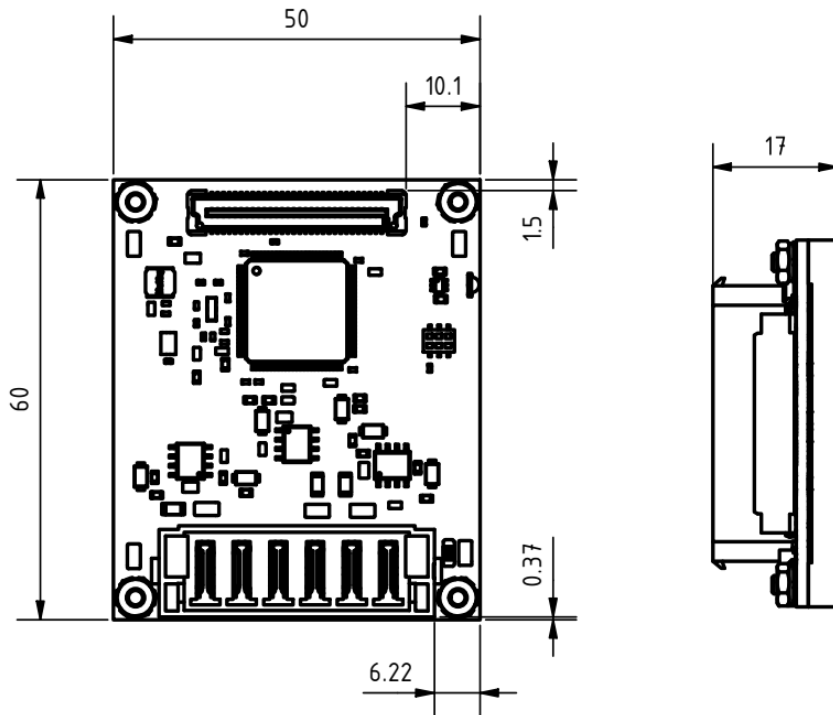
CPB3



CPB6



CPB15



3.2 Umgebungsbedingungen

Umgebungsbedingung	Wert
Schutzklasse	kein IP-Schutz
Umgebungstemperatur (Betrieb)	-10 ... +40°C
Luftfeuchtigkeit (nicht kondensierend)	0 ... 95 %
Max. Aufstellhöhe über NN	2000 m (Leistungseinschränkung ab 1000 m: -1%/100 m)
Umgebungstemperatur (Lagerung)	-25 ... +85°C

3.3 Elektrische Eigenschaften und technische Daten

Eigenschaft	Beschreibung/Wert
Betriebsspannung	12... 57,6 V DC
Nennstrom @40°C	CPB3-...: 3 A _{eff} CPB6-...: 6 A _{eff} (für Temperatur-Derating siehe Übertemperaturschutz) CPB15-...: <ul style="list-style-type: none"> ■ 15 A_{eff} (@25°C) ■ 10 A_{eff} (@40°C)
Spitzenstrom @40°C	<ul style="list-style-type: none"> ■ CPB3-1-...: 3 A_{eff}

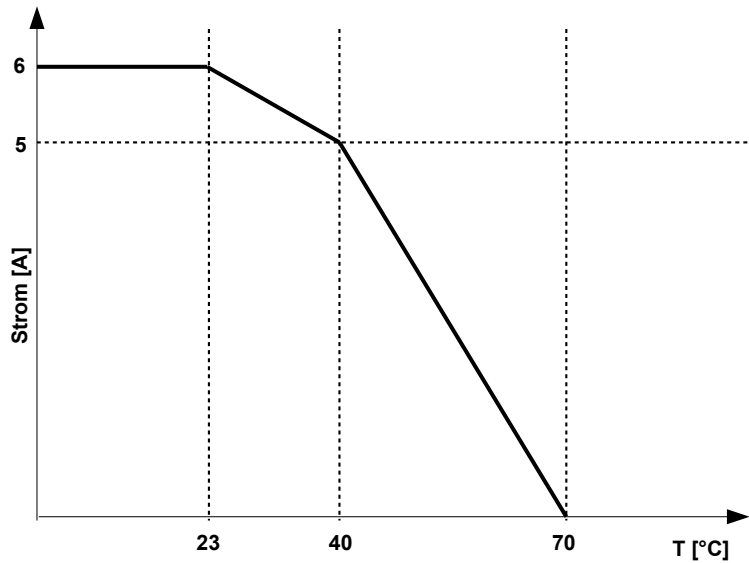
Eigenschaft	Beschreibung/Wert
	<ul style="list-style-type: none"> ■ CPB3-2-...: 9 A_{eff} (für max. 5 Sekunden)
	<ul style="list-style-type: none"> ■ CPB6-1-...: 6 A_{eff} ■ CPB6-2-...: 18 A_{eff} (für max. 5 Sekunden)
	CPB15-...: 45 A _{eff} (für max. 5 Sekunden)
Kommutierung	CPB3-..., CPB6-...: Schrittmotor <i>Open Loop</i> , Schrittmotor <i>Closed Loop</i> mit Encoder, BLDC sinuskommutiert über Hallsensor, BLDC sinuskommutiert über Encoder CPB15-...: BLDC sinuskommutiert über Hallsensor, BLDC sinuskommutiert über Encoder Anmerkung: Für Encoder und Hallsensor ist eine externe Beschaltung erforderlich!
Betriebsmodi	<i>Profile Position Mode, Profile Velocity Mode, Profile Torque Mode, Velocity Mode, Homing Mode, Interpolated Position Mode, Cyclic Sync Position Mode, Cyclic Sync Velocity Mode, Cyclic Synchronous Torque Mode, Takt-Richtung-Modus</i>
Sollwertvorgabe/ Programmierung	<i>Takt-Richtung, Analog, NanoJ-Programm</i>
Schnittstellen	USB, RS485 (Modbus RTU) Anmerkung: Für USB und RS485 ist eine externe Beschaltung erforderlich!
Encoder/Hall	1x SSI-Encoder, 1x Hallsensor, 1x inkrementeller Encoder (zweiter SSI-Encoder stattdessen konfigurierbar) Anmerkung: Für Encoder und Hallsensor ist eine externe Beschaltung erforderlich!
I/O	12x General I/O (einer davon kann als Ausgang für externe Bremse verwendet werden), 2x Analogeingang, 1x Ausgang für die externe Ballast-Schaltung
Übertemperatur	Schutzschaltung bei Temperatur > 80 °C
Stützkondensator	Nanotec empfiehlt pro Ampere Nennstrom am Motor eine Kapazität von ca. 1000 µF.

3.4 Übertemperaturschutz

Ab einer Temperatur von ca. 80 °C auf der Leistungsplatine wird das Leistungsteil der Steuerung abgeschaltet und das Fehlerbit gesetzt (siehe Objekt [1001_n](#) und [1003_n](#)). Nach Abkühlung und dem Bestätigen des Fehlers (siehe [Tabelle für das Contolword](#), "Fault reset") funktioniert die Steuerung wieder normal.

Temperaturabhängige Leistungsreduzierung für die CPB6

Das folgende Diagramm zeigt den zulässigen Dauerstrom abhängig von der Umgebungstemperatur:



HINWEIS



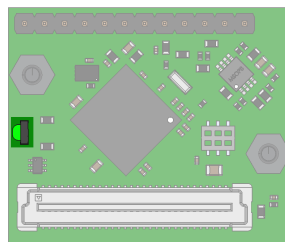
Da das genaue Temperaturverhalten außer vom Motor auch von der Anflanschung und dem dortigen Wärmeübergang sowie von der Konvektion in der Applikation abhängt, empfehlen wir bei Applikationen, die hinsichtlich Stromhöhe und Umgebungstemperatur problematisch sind, immer einen Dauertest in der realen Umgebung.

3.5 LED-Signalisierung

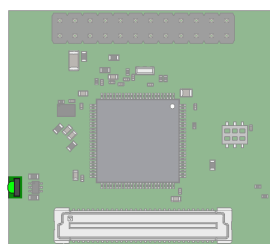
3.5.1 Betriebs-LED

Die Betriebs-LED zeigt den aktuellen Status an.

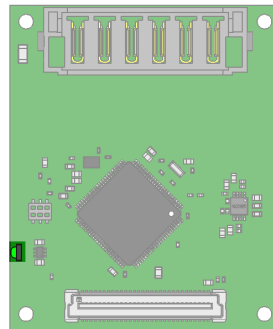
CPB3



CPB6

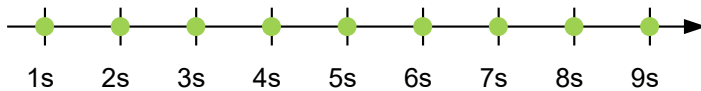


CPB15



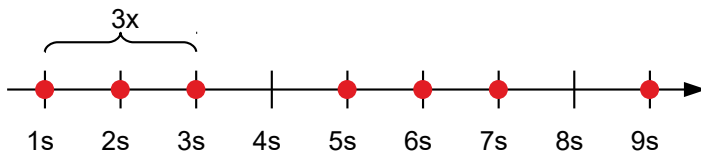
3.5.1.1 Normaler Betrieb

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 `1003h` ein genauerer Fehlercode hinterlegt.



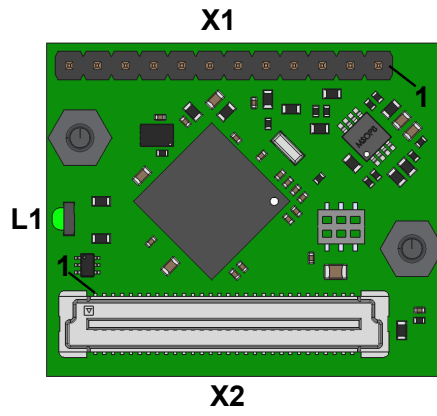
TIPP

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

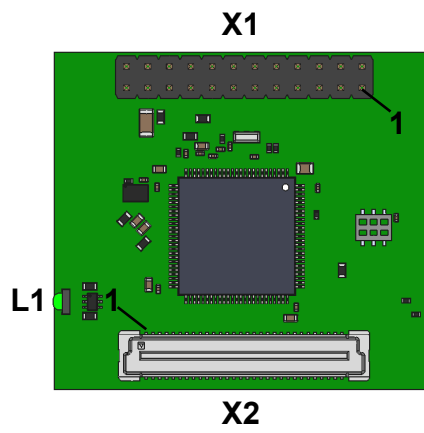
3.6 Anschlussbelegung

3.6.1 Übersicht

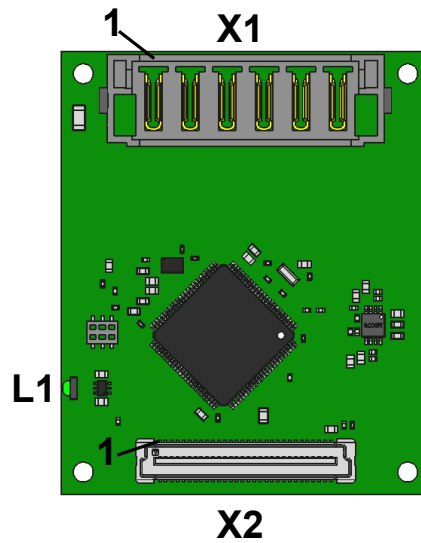
CPB3



CPB6



CPB15



3.6.2 X1 — Spannungsversorgung und Motor

CPB3

Steckverbinder: Würth Elektronik 61301211121

Pin	Name	Beschreibung/Funktion
1	GND	
2	BN_OUT	B\ (Schrittmotor)
3	BN_OUT	B\ (Schrittmotor)
4	B_OUT	B (Schrittmotor) oder W (BLDC)
5	B_OUT	B (Schrittmotor) oder W (BLDC)
6	AN_OUT	A\ (Schrittmotor) oder V (BLDC)
7	AN_OUT	A\ (Schrittmotor) oder V (BLDC)
8	A_OUT	A (Schrittmotor) oder U (BLDC)
9	A_OUT	A (Schrittmotor) oder U (BLDC)
10	+UB	12 - 57,6 V DC
11	+UB	12 - 57,6 V DC
12	GND	

CPB6

Steckverbinder: Würth Elektronik 61302421121

Pin	Name	Beschreibung/Funktion
1	GND	
2	GND	
3	BN_OUT	B\ (Schrittmotor)
4	BN_OUT	B\ (Schrittmotor)

Pin	Name	Beschreibung/Funktion
5	BN_OUT	B\ (Schrittmotor)
6	BN_OUT	B\ (Schrittmotor)
7	B_OUT	B (Schrittmotor) oder W (BLDC)
8	B_OUT	B (Schrittmotor) oder W (BLDC)
9	B_OUT	B (Schrittmotor) oder W (BLDC)
10	B_OUT	B (Schrittmotor) oder W (BLDC)
11	AN_OUT	A\ (Schrittmotor) oder V (BLDC)
12	AN_OUT	A\ (Schrittmotor) oder V (BLDC)
13	AN_OUT	A\ (Schrittmotor) oder V (BLDC)
14	AN_OUT	A\ (Schrittmotor) oder V (BLDC)
15	A_OUT	A (Schrittmotor) oder U (BLDC)
16	A_OUT	A (Schrittmotor) oder U (BLDC)
17	A_OUT	A (Schrittmotor) oder U (BLDC)
18	A_OUT	A (Schrittmotor) oder U (BLDC)
19	+UB	12 - 57,6 V DC
20	+UB	12 - 57,6 V DC
21	+UB	12 - 57,6 V DC
22	+UB	12 - 57,6 V DC
23	GND	
24	GND	

CP15

Steckverbinder: Samtec MPT-06-6.30-01-L-V

Pin	Name	Beschreibung/Funktion
1	+UB	12 - 57,6 V DC
2	GND	
3	U_OUT	U (BLDC)
4	V_OUT	V (BLDC)
5	W_OUT	W (BLDC)
6	GND	

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.

3.6.3 X2 — Ein- und Ausgänge

Steckverbinder:

- CPB3: Amphenol 10144518-063802LF
- CPB6: Amphenol 10144518-063802LF
- CPB15: Amphenol 10144518-062802LF

HINWEIS



Einige Pins sind frei konfigurierbar und unterstützen alternative Funktionen. Details finden Sie im Kapitel Ein- und Ausgangsbelegung festlegen.

Pin	Name	Beschreibung/Funktion
1	GND	
2	GND	
3	reserviert	nicht beschalten
4	ANA1	Analogeingang 1: 0...3,2 V
5	reserviert	nicht beschalten
6	ANA2	Analogeingang 2: 0...3,2 V
7	reserviert	nicht beschalten
8	Spare	nicht beschalten
9	reserviert	nicht beschalten
10	RS485_DE	<i>Driver Enable</i>
11	RS485_RE	<i>Receiver Enable</i>
12	reserviert	nicht beschalten
13	RS485_RX	Modbus RTU Rx
14	reserviert	nicht beschalten
15	RS485_TX	Modbus RTU Tx
16	GND	
17	GND	
18	H1	Hallsensor 1
19	DIO11	General I/O
20	H2	Hallsensor 2
21	DIO13	General I/O
22	H3	Hallsensor 3
23	ENC1B	Encoder 1, B
24	ENC1A	Encoder 1, A
25	SSI1_MISO	SSI-Encoder 1, Data IN
26	ENC1I	Encoder 1, Index
27	SSI1_CLK	SSI-Encoder 1, Clock
28	U_SPIMISO	MISO-Pin für <u>Drehschalter</u>
29	DIO14	General I/O
30	U_SPISCK	SCK-Pin für <u>Drehschalter</u>
31	reserviert	nicht beschalten
32	reserviert	nicht beschalten
33	U_SPINSS	SH/LD-Pin für <u>Drehschalter</u>
34	reserviert	nicht beschalten
35	VBUS	USB VBUS (Versorgung)

Pin	Name	Beschreibung/Funktion
36	DP	USB DP (Data+)
37	GND	
38	DM	USB DM (Data-)
39	DIO1	General I/O
40	GND	
41	DIO3	General I/O
42	DIO2	General I/O
43	DIO5	General I/O
44	DIO4	General I/O
45	DIO7	General I/O
46	DIO6	General I/O
47	DIO9	General I/O
48	BRAKE	Ansteuerung der externen Bremse
49	GND	
50	DIO10	General I/O
51	uC Reset	Systemfunktion, reserviert
52	Ballast	zur Ansteuerung der externen Ballast-Schaltung
53	uC Supply	Spannungsversorgung 3,3 V Muss vom Motherboard bereitgestellt werden, Grenzen: 3,2... 3,4 V DC, min. 140 mA
54	uC Supply	Spannungsversorgung 3,3 V Muss vom Motherboard bereitgestellt werden, Grenzen: 3,2... 3,4 V DC, min. 140 mA
55	Driver Supply	Spannungsversorgung 10,5 V Muss vom Motherboard bereitgestellt werden, Grenzen: 9,6... 14 V DC, min. 70 mA
56	DIO12	General I/O
57	Ub_Logic	optionale externe Logikversorgung, 12...30 V DC Wenn Ub_Logic auf dem Motherboard verwendet wird, schließen Sie diese hier an, damit die Firmware sie detektieren kann.
58	Spare	nicht beschalten
59	GND	
60	GND	

HINWEIS



- Für die digitalen Eingänge liegt die Einschaltswelle bei 1,79 V, die Ausschaltswelle liegt bei 1,11 V DC (, abhängig von der Versorgung an Pins 53/54, typische Werte @3,3 V). Die maximale Abtastfrequenz liegt bei 1 MHz. Wenn die I/O-Pins als Ausgang verwendet werden (siehe Ein- und Ausgangsbelegung festlegen), ist die Strombelastbarkeit ca. 10 mA bei 3,3 V DC.



TIPP

Im Kapitel Hardware-Installation finden Sie Hinweise zum Design Ihres Motherboards.

4 Hardware-Installation



HINWEIS

Beachten Sie, dass alle Bauteile spannungsfrei sind.



HINWEIS

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

4.1 Anschließen der Steuerung

HINWEIS

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



Geeignete Maßnahmen können sein:

- ▶ Geschirmte Leitungen verwenden und den Leitungsschirm beidseitig auf kurzem Weg erden.
- ▶ Stromversorgungs- und Motorleitungen so kurz wie möglich halten.
- ▶ Kabel mit paarweise verdrehten Adern verwenden.
- ▶ Motorgehäuse großflächig auf kurzem Weg erden.
- ▶ Versorgungs-, Motor- und Steuerleitungen getrennt verlegen.

4.1.1 Integrieren der CPB in ein Motherboard

Die minimale Beschaltung variiert je nach Motortyp und vorhandener Rückführung (Schritt- oder BLDC-Motor, Hallensoren/Encoder). Zur Inbetriebnahme ist der Anschluss der Spannungsversorgung (X1 — Spannungsversorgung und Motor), des Motors und eines RS485-Transceivers ausreichend.

Die folgende Tabelle zeigt die notwendigen Spannungsschienen und die Anforderungen, die Ihr Motherboard für diese erfüllen soll. Bei der Berechnung des Stromverbrauchs wurde eine niedrige Effizienz von 80% angenommen für die Abwärtsrichter.

Spannungs-Schiene	Verwendet für	Stromverbrauch	Ripple erlaubt	Toleranz der regulierten Spannung	Empfohlene Topologie
+10-12V	Gate-Drivers, +3.3V, +5V	70...160 mA	100 mV p-p		Abwärtsrichter von Hauptversorgung UB
+5V	Encoder/ Hall, +3.3V	150 mA max.	100 mV p-p		Abwärtsrichter von +10-12V
+3.3V	MCU	140 mA min.		+/-3%	Low-Dropout-Regler (LDO)

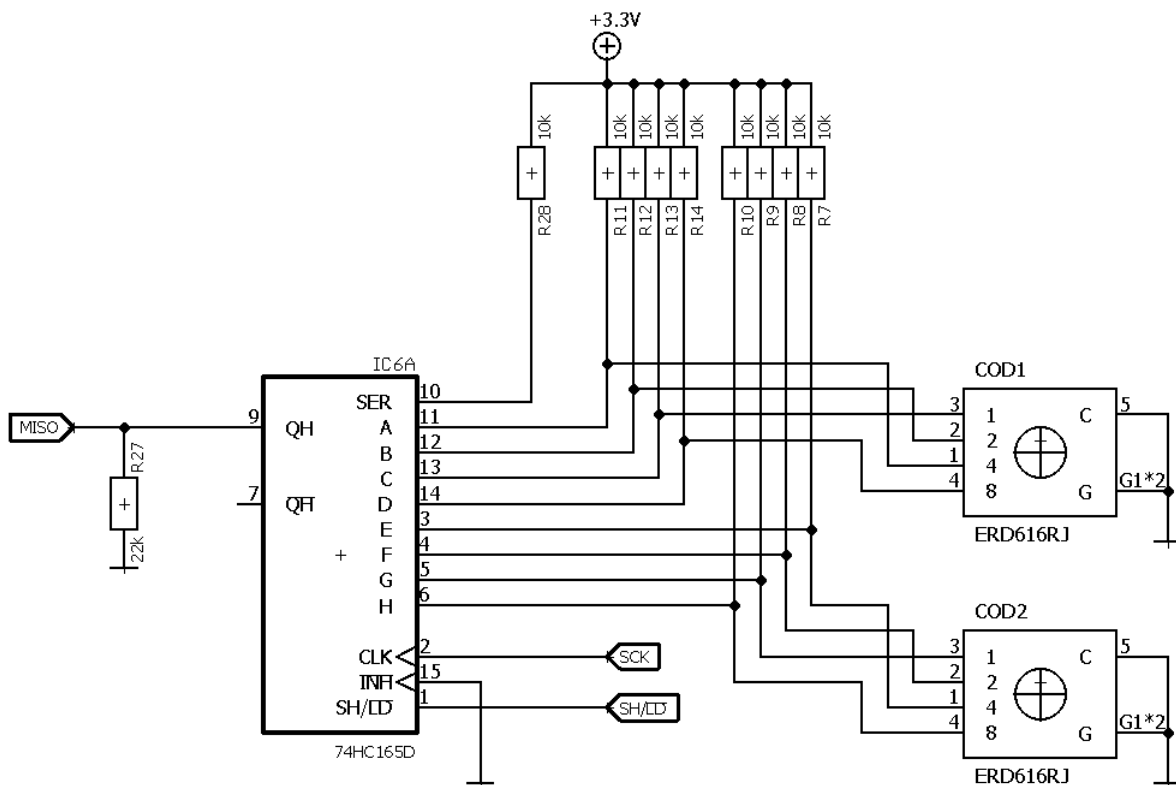
HINWEIS



Ein Verpolungsschutz ist nicht vorhanden. Installieren Sie eine Sicherung in die Zuleitung, dimensioniert entsprechend dem Stromverbrauch in Ihrer Applikation.

4.1.2 Schalter für Kommunikationseinstellungen anschließen

Sie können zwei Drehschalter an die vorgesehenen Pins der Steuerung anschließen, worüber Sie die Kommunikationsparameter einstellen. Die Beschaltung muss exakt der unten vorgegebenen Referenzschaltung entsprechen. Die drei dafür vorgesehenen Pins sind Pin 28 (MISO), Pin 30 (SCK) und Pin 33 (SH/LD).



HINWEIS



- Verwenden Sie den Schieberegister *74HC165D* oder ein äquivalentes Produkt. Legen Sie Pin 10 auf die 3,3-Volt-Versorgung via Pullup-Widerstand, damit die Steuerung die Schaltung detektieren kann.
- Wenn Sie keine Drehschalter verwenden möchten, schließen Sie nichts an die drei vorgesehenen Pins: MISO (Pin 28), SCK (Pin 30) und SH/LD (Pin 33).

Damit die Steuerung die Drehschalter detektiert und verwendet, müssen Sie den Wert "1" ins Objekt `2104h:01h` schreiben (im Auslieferungszustand der Fall). Ist der Wert des Objekts "0", werden die in den entsprechenden Objekten gespeicherten Werte verwendet. Die Objekte und die möglichen Kombinationen finden Sie im Kapitel [Konfiguration über Modbus RTU](#).

5 Inbetriebnahme

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

5.1 Konfiguration über USB

5.1.1 Allgemeines

HINWEIS



Für USB ist eine externe Beschaltung der Pins 36 (CP) und 38 (DM) erforderlich (siehe [Anschlussbelegung](#)).

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

Konfigurationsdatei

Diese Datei lässt sich mittels dem USB-Anschluss auf die Steuerung speichern. Lesen Sie dazu die Kapitel [USB Anschluss](#) und [Konfigurationsdatei](#).

NanoJ-Programm

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

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

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

5.1.2 USB-Anschluss

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

Es werden drei Dateien angezeigt, die Konfigurationsdatei (`cfg.txt`), das *NanoJ-Programm* (`vmmcode.usr`) und die Informationsdatei (`info.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 Dateieindung `bat` und folgendem Inhalt erzeugen:

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

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

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

5.1.3 Konfigurationsdatei

5.1.3.1 Allgemeines

Die Konfigurationsdatei `cfg.txt` dient dazu, Werte für das Objektverzeichnis beim Start auf einen bestimmten Wert vorzublegen. 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.

5.1.3.2 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.

5.1.3.3 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

```
3202:00.03=1
```

Bit zurücksetzen

```
3202:00.03=0
```

Bitweise OR

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

Bitweise AND

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

Beispiel

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

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

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

```
3202:00=8
```

oder nur Bit 3 setzen

```
3202:00.03=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.



5.1.4 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 Dateierdung `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 Dateierdung `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.

5.2 Konfiguration über Modbus RTU

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

Ab Werk ist die Steuerung ist auf Slave-Adresse 5, Baudrate 19200 Baud, even Parity, 1 Stop Bit eingestellt. Alle Änderungen sind erst nach einem Neustart der Steuerung wirksam.

5.2.1 Kommunikationseinstellungen

Wenn Sie keine Drehschalter angeschlossen haben, setzen Sie das Objekt `2104h:01h` auf den Wert "0". Die Kommunikationsparameter stellen Sie in den Objekten `2028h`, `202Ah` und `202Dh` ein.

Folgende Einstellungen können vorgenommen werden:

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

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

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

Unterstützt werden folgende Baudraten:

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

Sie müssen die Änderungen speichern, indem Sie den Wert "65766173_h" in das Objekt 1010_h:0B_h und :02_h schreiben. Die Änderungen werden erst nach einem Neustart der Steuerung übernommen.

Wenn sie die Drehschalter verwenden möchten, setzen Sie das Objekt 2104_h:01_h auf den Wert "1". Die vorgegebene Beschaltung für die Drehschalter finden sie im Kapitel [Schalter für Kommunikationseinstellungen anschließen](#).

Mit der Zahlenkombination aus beiden Drehschaltern können Sie die Quelle für die Slave-Adresse, die Baudrate und die Parität einstellen.

Dabei gilt: die Zahlenkombination setzt sich aus beiden Drehschaltern zusammen, wobei der Hex-Schalter an den Pins E bis H (S2) das höherwertigere Byte darstellt und der andere (S1) entsprechend das niederwertigere Byte.

Beispiel

Schalter S1 steht auf dem Wert "0_h", Schalter S2 auf dem Wert "F_h", daraus ergibt sich der Wert "0F_h"="15_d".

Schalter S1 steht auf dem Wert "A_h", Schalter S2 auf dem Wert "1_h", daraus ergibt sich der Wert "A1_h"="161_d".

Zahlenkombination der Drehschalter		Slave-Adresse	Baudrate und Parität
dec	hex		
0	0	Objekt 2028 _h	Objekt 202A _h bzw. 202D _h
1-247	1-F7	Zahl der Drehschalter	Objekt 202A _h bzw. 202D _h
248-255	F8-FF	5	19200, even Parity

5.3 Motordaten einstellen

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

- Polpaarzahl: Objekt 2030_h:00_h (Pole pair count) Hier ist die Anzahl der Motorpolpaare einzutragen. Bei einem Schrittmotor wird die Polpaarzahl über den Schrittwinkel berechnet, z.B. 1,8° = 50 Polpaare, 0,9° = 100 Polpaare (siehe Schrittwinkel im Motordatenblatt). Bei BLDC-Motoren ist die Polpaarzahl direkt im Motordatenblatt angegeben.
- Objekt 2031_h:00_h: Maximal zulässiger Motorstrom (Motorschutz) in mA (siehe Motordatenblatt)
- Objekt 6075_h:00_h Nennstrom des Motors in mA (siehe Motordatenblatt), begrenzt durch 2031_h

- Objekt 6073_h:00_h: Maximaler Strom (entspricht bei einem Schrittmotor in der Regel dem Nennstrom, Bipolar) in Promille des eingestellten Nennstroms (siehe Motordatenblatt). Werkseinstellung: "1000", was 100% des Wertes in 6075_h entspricht. Wird durch 2031_h begrenzt.
- Objekt 203B_h:02_h Maximale Dauer des maximalen Stroms (6073_h) in ms (für die Erstinbetriebnahme empfiehlt Nanotec einen Wert von 100 Millisekunden; dieser Wert ist später an die konkrete Applikation anzupassen).
- Motortyp einstellen:
 - Schrittmotor:
 - Objekt 3202_h:00_h (Motor Drive Submode Select): Definiert den Motortyp Schrittmotor, aktiviert die Stromabsenkung bei Stillstand des Motors: 0000008h. Siehe auch Kapitel Inbetriebnahme Open Loop.
 - BLDC-Motor:
 - Objekt 3202_h:00_h (Motor Drive Submode Select): Definiert den Motortyp BLDC: 00000040h
- Motor mit Encoder ohne Index: Sie müssen nach dem Auto-Setup die Encoder-Parameter einstellen, siehe Kapitel Konfigurieren der Sensoren.
- □ Schrittmotor, Bremsensteuerung (und Stromabsenkung) aktiviert: 0000000Ch
- BLDC-Motor, Bremsensteuerung aktiviert: 00000044h

HINWEIS

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



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

5.4 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 2300_h NanoJ Control).

TIPP



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

5.4.1 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)	✓
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)	-	✓	---

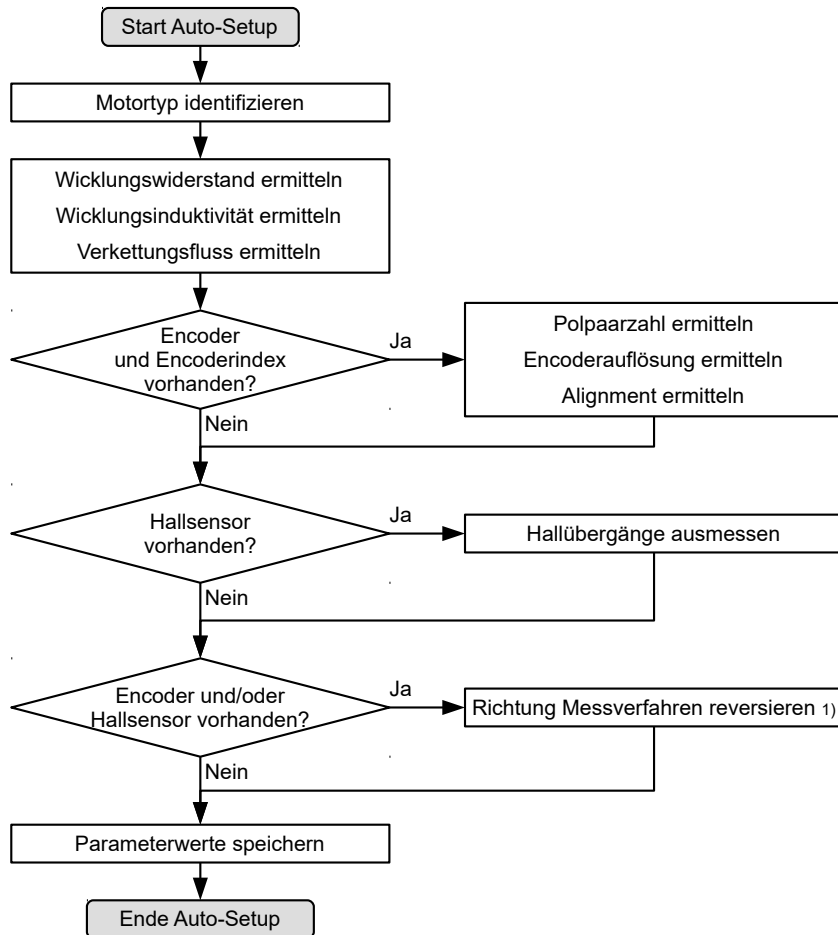
Parameter	Motor ohne Hall-Sensor	Motor mit Hall-Sensor
Hallübergänge	-	✓

5.4.2 Durchführung

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

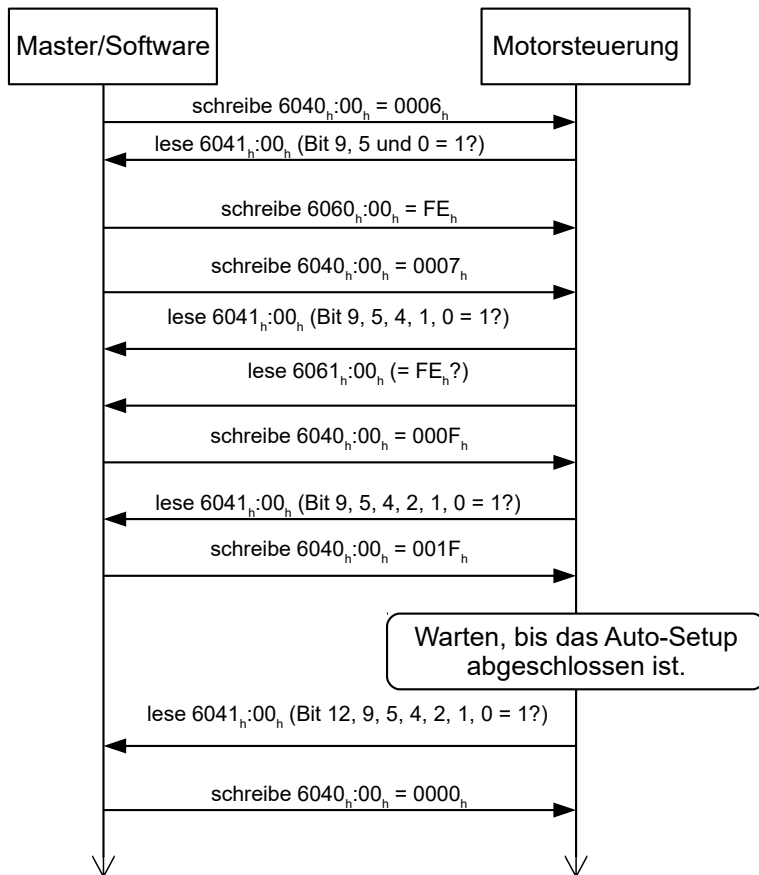
1. Zum Vorwählen des Betriebsmodus *Auto-Setup* tragen Sie in das Objekt 6060_h:00_h den Wert "-2" (= "FE_h") ein.
Die *Power state machine* muss nun in den Zustand *Operation enabled* versetzt werden, siehe [CiA 402 Power State Machine](#).
2. Starten Sie das *Auto-Setup* mit Setzen von Bit 4 *OMS*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

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



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

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



5.4.3 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.

5.5 Konfigurieren der Sensoren

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

Objekt	Rückführung	Beschreibung
3380 _h	Sensorless	enthält Mess- und Konfigurations-Werte für die sensorlose Regelung
3390 _h	Hall-Sensor (digital)	enthält Konfigurationswerte für die Hall-Sensoren
33A0 _h	Inkrementaler Encoder 1	enthält Konfigurationswerte für den ersten inkrementalen Encoder

Objekt	Rückführung	Beschreibung
<u>33B0_h</u>	SSI-Encoder 1	enthält Konfigurationswerte für den ersten SSI-Encoder
<u>33B1_h</u>	SSI-Encoder 2	enthält Konfigurationswerte für den zweiten SSI-Encoder. In <u>Ein- und Ausgangsbelegung festlegen</u> lesen Sie, wie Sie die Funktion der vorgesehenen Pins aktivieren.

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
<u>60E6_h Additional Position Encoder Resolution - Encoder Increments</u>	1256637
<u>60EB_h Additional Position Encoder Resolution - Motor Revolutions</u>	20
<u>60E8_h Additional Gear Ratio - Motor Shaft Revolutions</u>	4
<u>60ED_h Additional Gear Ratio - Driving Shaft Revolutions</u>	1
<u>60E9_h Additional Feed Constant - Feed</u>	2513274 Inkr. (entspricht 1256,637 mm)
<u>60EE_h Additional Feed Constant - Driving Shaft Revolutions</u>	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 1003_h ein Fehlercode hinterlegt.

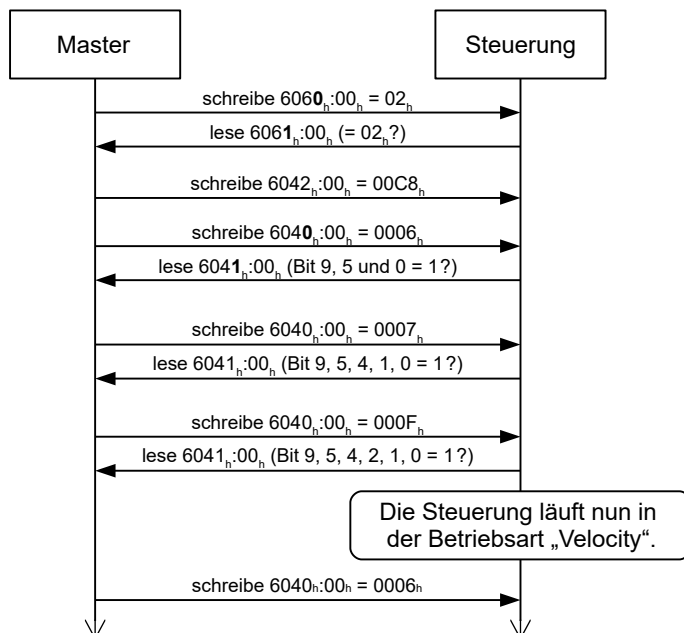
5.6 Testlauf

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

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

1. Wählen Sie den Modus *Velocity*, indem Sie das Objekt 6060_h (Modes Of Operation) auf den Wert "2" setzen.
2. Schreiben Sie die gewünschte Drehzahl in 6042_h .
3. Versetzen Sie die *Power state machine* in den Zustand *Operation enabled*, siehe CiA 402 Power State Machine.

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



4. Um den Motor zu stoppen, setzen Sie das Controlword (6040_h) auf "6".

6 Generelle Konzepte

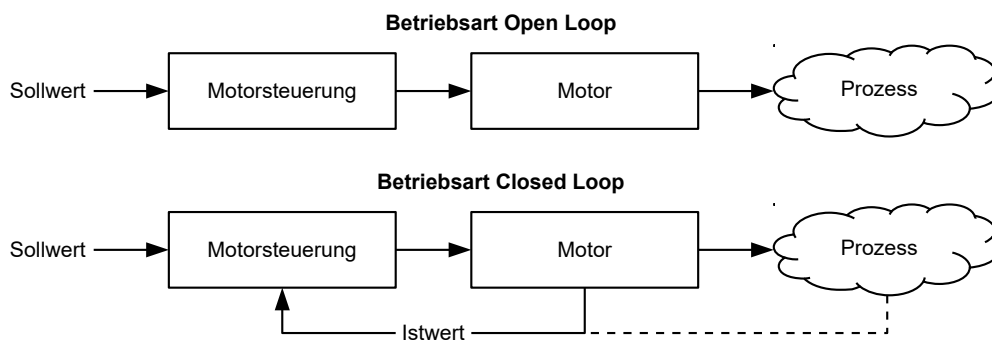
6.1 Betriebsarten

6.1.1 Allgemein

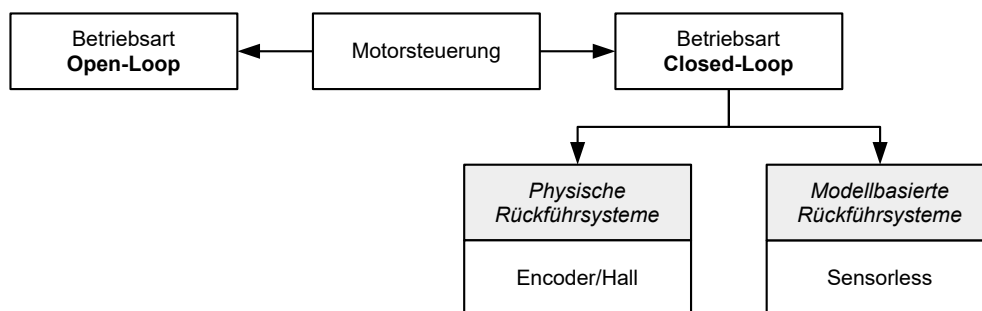
Die Betriebsart von Systemen ohne Rückführung wird als *Open Loop*, die mit Rückführung als *Closed Loop* bezeichnet. In der Betriebsart *Closed Loop* ist es zunächst unerheblich, ob die zurückgeführten Signale vom Motor selbst oder aus dem beeinflussten Prozess kommen.

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
Takt-Richtung	ja	ja	ja

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

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

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

6.1.2 Open Loop

6.1.2.1 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.

6.1.2.2 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 zu aktivieren, um die Verlustleistung und Wärmeentwicklung zu reduzieren. Um die Stromabsenkung zu aktivieren, sind folgende Einstellungen notwendig:

- Im Objekt 3202_h (Motor Drive Submode Select) das Bit 3 (CurRed) auf "1" setzen.
- Im Objekt 2036_h (Open Loop Current Reduction Idle Time) wird die Zeit in Millisekunden angegeben, die sich der Motor im Stillstand (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.

6.1.2.3 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.

Betriebsmodus Takt-Richtung

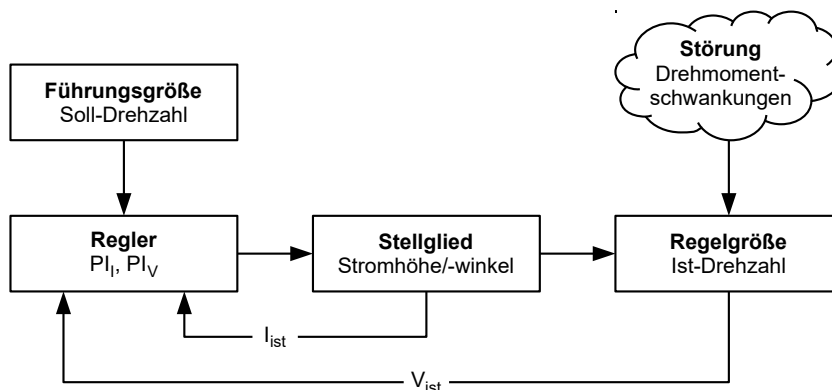
Änderung der Schrittauflösung über die Objekte 2057_h (Clock Direction Multiplier) und 2058_h (Clock Direction Divider). Beschleunigungs-/Verzögerungsrampen durch Anpassen der Impulsfrequenz optimieren, um den Resonanzbereich möglichst schnell zu durchlaufen.

6.1.3 Closed Loop

6.1.3.1 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 Vollschriffs korrigiert werden.

6.1.3.2 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 `3202h` 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/Cyclic Synchronous Velocity, wenn das Bit 1 in `3202h` 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
<code>321A_h:01_h</code>	Stromregler	[mV/A]	Proportional-Anteil momentbildende Komponente
	Proportional Gain K_p for I_q		
<code>321A_h:02_h</code>	Stromregler	[μ s]	Integrierzeit momentbildende Komponente
	Integrator Time T_i for I_q		
<code>321A_h:03_h</code>	Stromregler	[mV/A]	Proportional-Anteil feldbildende Komponente
	Proportional Gain K_p for I_d		
<code>321A_h:04_h</code>	Stromregler	[μ s]	Integrierzeit feldbildende Komponente
	Integrator Time T_i for I_d		
<code>321B_h:01_h</code>	Geschwindigkeitsregler	[mA/Hz]	Proportional-Anteil
	Proportional Gain K_p		
<code>321B_h:02_h</code>	Geschwindigkeitsregler	[μ s]	Integrierzeit
	Integrator Time T_i		
<code>321C_h:01_h</code>	Positionsregler	[Hz]	Proportional-Anteil
	Proportional Gain K_p		

Objekt	Name	Einheit	Beschreibung
		(Reglerabweichung in mech. Umdrehungen pro Sekunde)	
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

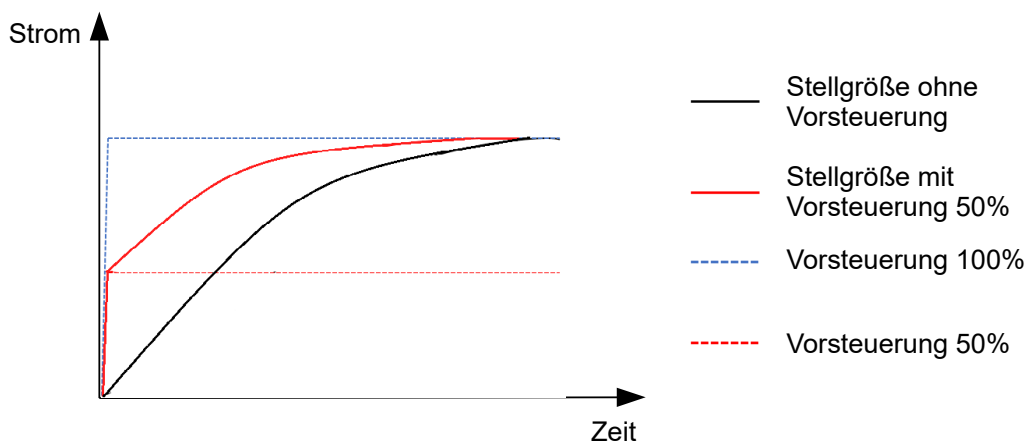
6.1.3.3 Vorsteuerung

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".



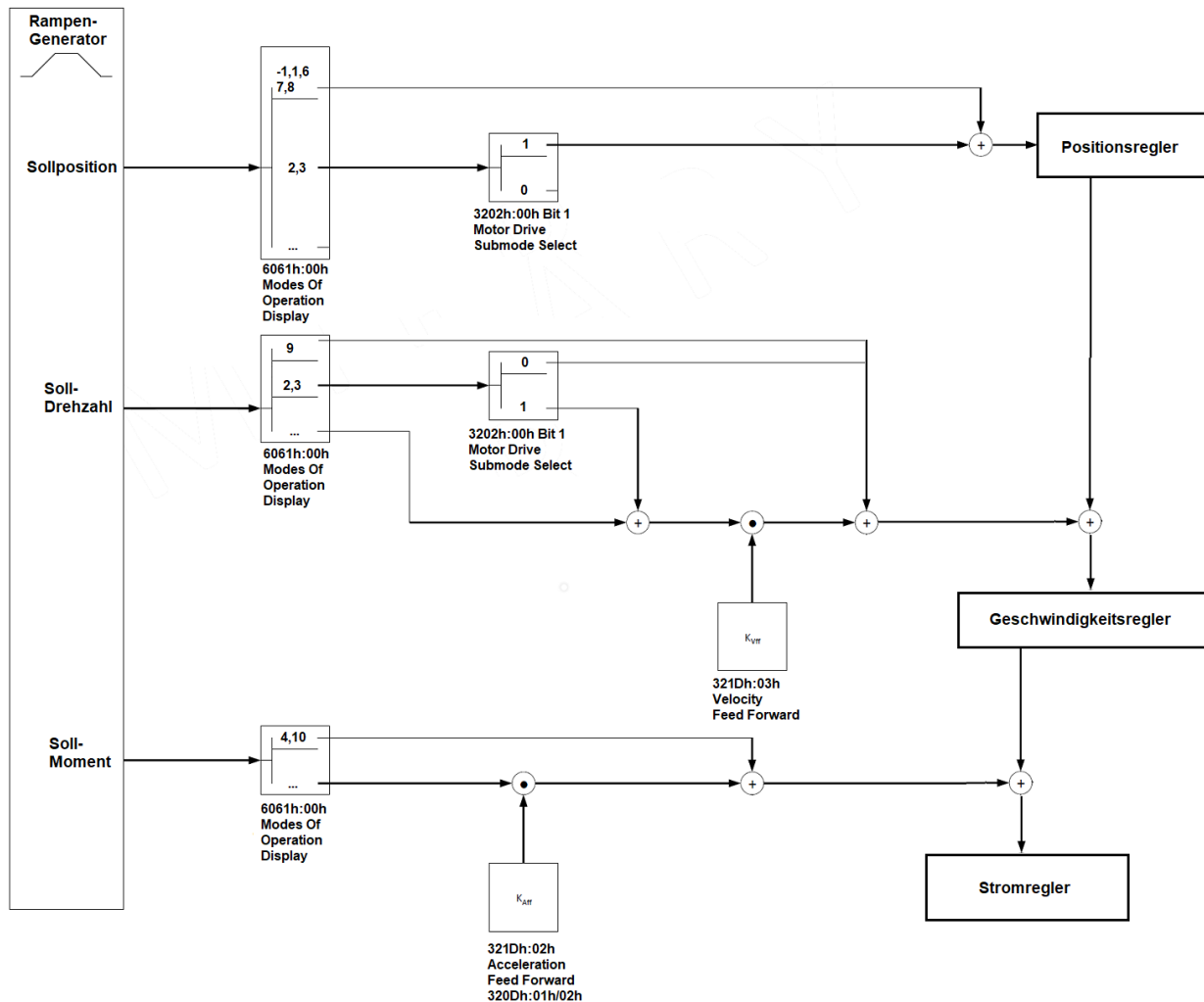
Der Faktor für die *Geschwindigkeitsvorsteuerung* wird im Objekt 321D_h:03_h in Promille des Ausgangs des Rampengenerators (606B_h) eingestellt und vor dem Geschwindigkeitsregler zum Ausgang des Positionsreglers addiert. Die *Geschwindigkeitsvorsteuerung* ist in allen Modi mit Positionsregelkreis aktiv:

- 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".

6.1.3.4 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	Rückführung 2	Rückführung 3
		Sensorless	Hall	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

¹Die 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.

²Direkt 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).

6.1.3.5 Inbetriebnahme

Vor dem Anwenden der Betriebsart *Closed Loop* sollte ein Auto-Setup durchgeführt werden. Der Betriebsmodus Auto-Setup ermittelt automatisch die notwendigen Parameter (z.B. Motorkenndaten,

Rückführsysteme), die für eine optimale Arbeitsweise der feldorientierten Regelung notwendig sind. Alle Informationen zur Durchführung des Auto-Setups sind im Kapitel [Auto-Setup](#) beschrieben.

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

Das Bit 0 im [3202_h](#) muss gesetzt sein. Das Bit wird nach einem erfolgreich abgeschlossenen Auto-Setup automatisch gesetzt.

Aktivierung

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

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

Wenn kein Index vorhanden ist, oder dieser nicht verwendet werden soll, können Sie:

- einen zweiten Sensor zur Kommutierung verwenden (siehe [Zuordnung der Rückführungen zu den Regelkreisen](#))
- oder ein *Auto-Alignment* im *Open Loop* ermitteln lassen, indem Sie das Bit 4 in [3202_h](#) auf "1" setzen. Das *Auto-Alignment* wird einmalig bei jedem Neustart der Steuerung ermittelt, nach dem ersten Befehl, der die *CiA 402 Power State Machine* in den Zustand *Operation Enabled* versetzt. Dabei wird der Rotor um bis zu einem magnetischem Pol bewegt. Nachdem das Alignment ermittelt wurde, wird der Zustand *Operation Enabled* erreicht und ggf. die Fahrt fortgesetzt.

HINWEIS



Damit das *Auto-Alignment* ermittelt werden kann, müssen Sie sicherstellen, dass die (automatische oder manuelle) Bremsensteuerung deaktiviert ist (siehe Kapitel [Automatische Bremsensteuerung](#)).

VORSICHT!

Unkontrollierte Motorbewegungen!

Es kann zu unvorhersehbaren Reaktionen kommen, wenn das Alignment nicht richtig ermittelt wird.



Beachten Sie folgende Voraussetzungen für die Verwendung des *Auto-Alignments*:

- ▶ Die Motorwelle muss möglichst lastfrei sein. Wenn das nicht möglich ist, muss der Motor so ausgelegt werden, dass eine große Drehmoment-Reserve (mindestens 25%) vorhanden ist.
- ▶ Verwenden Sie einen Encoder mit ausreichend hoher Auflösung (mindestens 500 Inkremente pro Umdrehung, nach Quadratur, bei einem Motor mit 50 Polpaaren)

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

6.1.3.6 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.

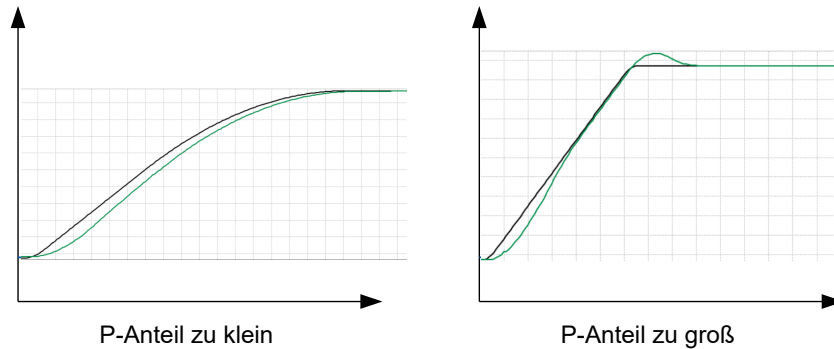
Ziel der Optimierung der Regelparameter (das sogenannte *Tuning* des Reglers) ist ein möglichst ruhiger Motorlauf, eine hohe Genauigkeit und eine hohe Dynamik in der Reaktion der Steuerung auf Störungen. Alle Regelabweichungen sollen so schnell wie möglich eliminiert werden.

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.

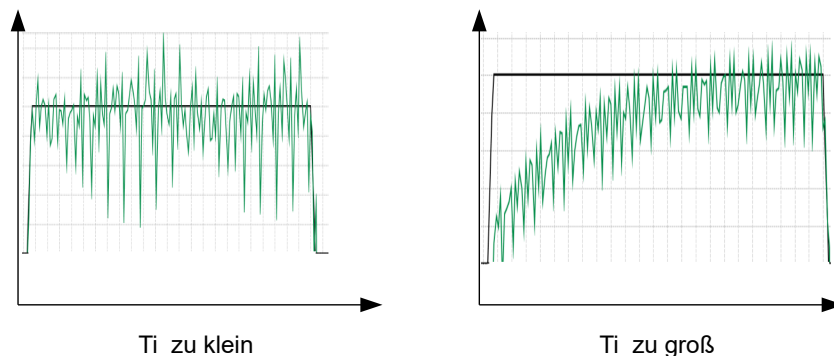
Jeder der drei Regler besteht aus einem Proportional- und einem Integral-Anteil, die normalerweise in dieser Reihenfolge angepasst werden sollten.

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".



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



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.



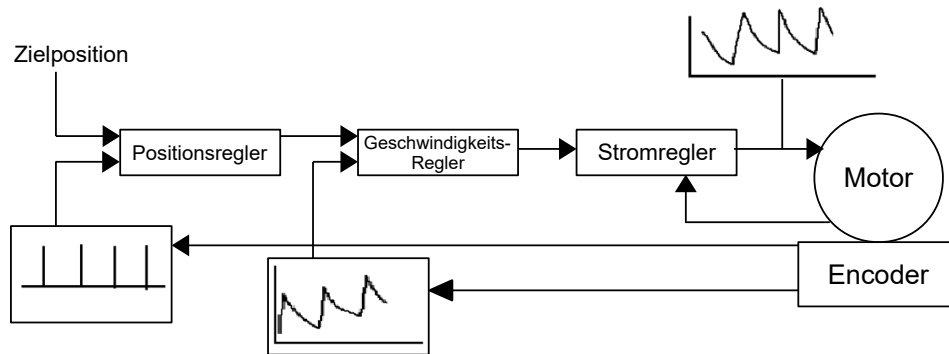
6.1.4 Slow Speed

6.1.4.1 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 reinen *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*.

6.1.4.2 Aktivierung

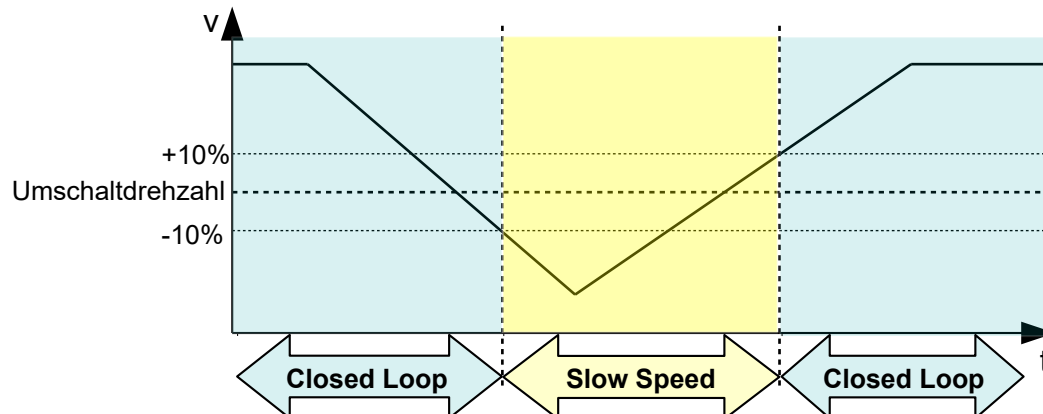
Um die Betriebsart *Slow Speed* zu aktivieren, müssen Sie:

1. den *Closed Loop* aktivieren,
2. das Bit 7 im Objekt `3202h` (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 Umschaltzahl 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.

6.1.4.3 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.

6.2 CiA 402 Power State Machine

6.2.1 Zustandsmaschine

6.2.1.1 CiA 402

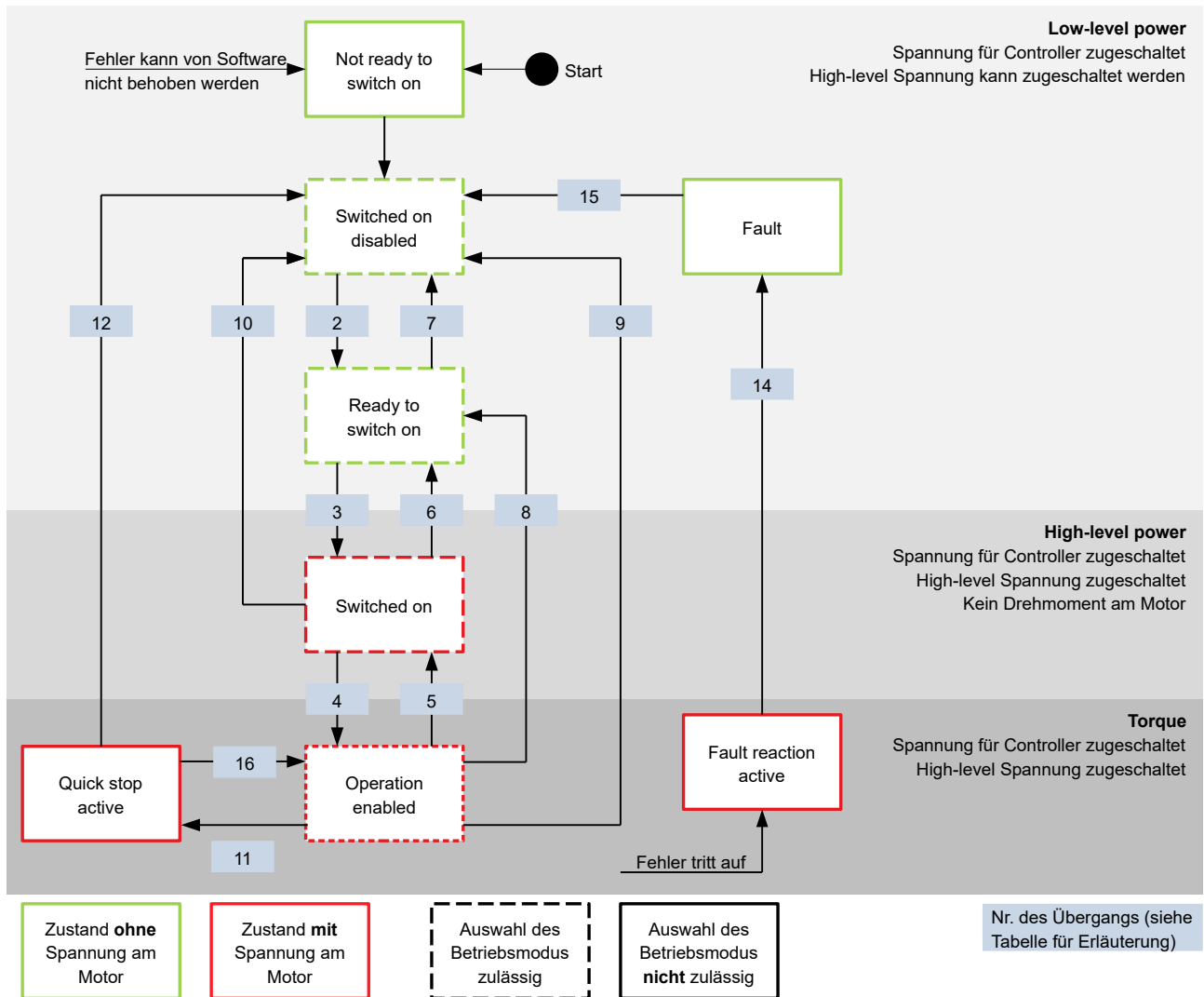
Um die Steuerung betriebsbereit zu schalten, ist es notwendig, eine Zustandsmaschine (*State Machine*) zu durchlaufen. Diese ist im *CANopen-Standard 402* definiert. Zustandsänderungen werden im Objekt [6040_h](#) (Controlword) angefordert. Der tatsächliche Zustand der Zustandsmaschine lässt sich aus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

6.2.1.2 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 Zurücksetzen des Fehlers (Fault reset) und der Wechsel von *Quick Stop Active* nach *Operation Enabled*: Der Übergang wird nur durch steigende Flanke des Bits angefordert.

Kommando	Bit im Objekt 6040 _h					Übergang
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Disable voltage	0	X	X	0	X	7, 10, 9, 12
Quick stop	0	X	0	1	X	11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4

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

6.2.1.3 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*.

6.2.1.4 Betriebsmodus

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

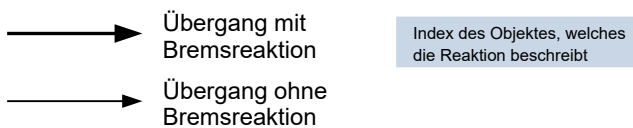
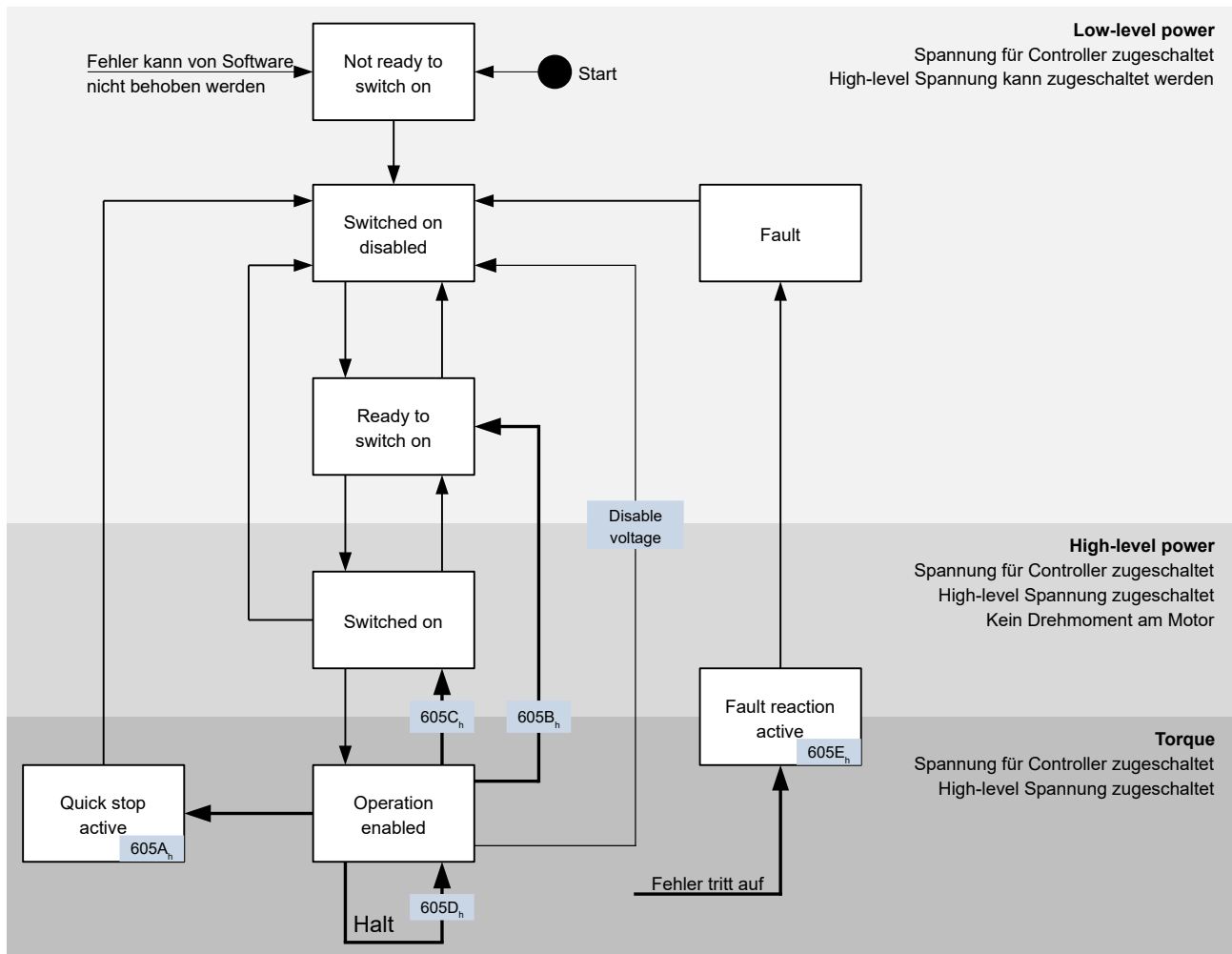
Die Einstellung oder Änderung des Betriebsmodus ist jederzeit möglich.

6.2.2 Verhalten beim Verlassen des Zustands *Operation enabled*

6.2.2.1 Bremsreaktionen

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

Die nachfolgende Grafik zeigt eine Übersicht der Bremsreaktionen.



6.2.2.2 Quick stop active

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

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

Wert in Objekt 605A _n	Beschreibung
0	Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> (6085 _n) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt

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

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

6.2.2.3 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 <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Ready to switch on</i>
2 bis 32767	Reserviert

6.2.2.4 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 — Motor kann sich frei drehen
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switched on</i>
2 bis 32767	Reserviert

6.2.2.5 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 <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit <i>quick stop ramp</i> (6085 _h)
3 bis 32767	Reserviert

6.2.2.6 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 <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit <i>quick stop ramp</i> (6085 _h)
3 bis 32767	Reserviert

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

6.2.2.7 Schlepp-/Schlupffehler

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

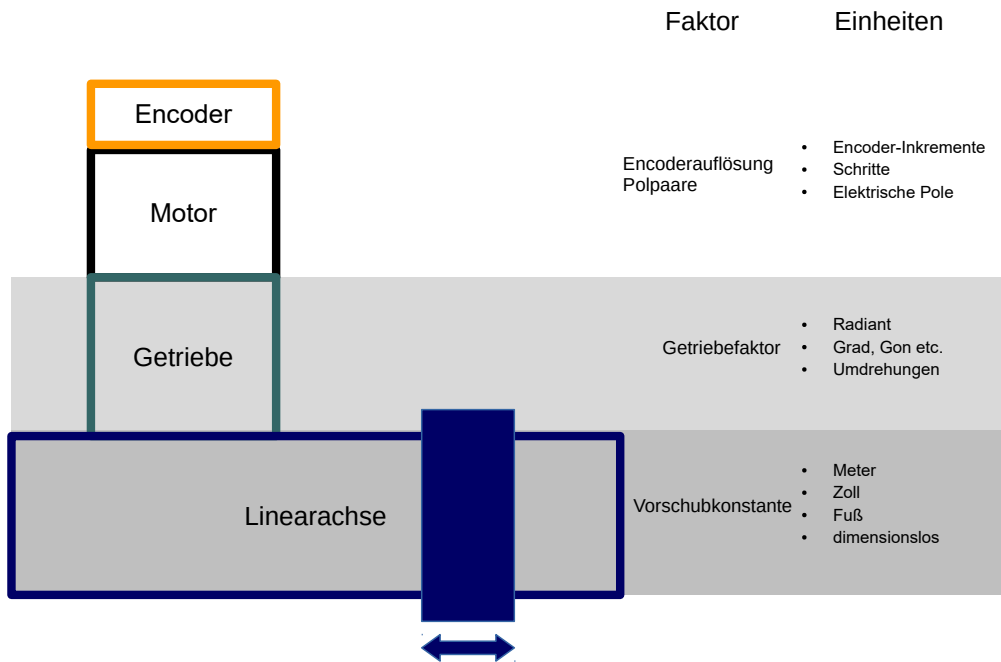
Wert	Beschreibung
-32768 bis -2	Reserviert
-1	keine Reaktion
0	Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit <i>quick stop ramp</i> (6085 _h)
3 bis 32767	reserviert

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 "7FFFFFFFF_h" setzen.

6.3 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.

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

Name	Einheitenzeichen	Wert	6091 _h	6092 _h	Beschreibung
metre	m	01 _h	ja	ja	<i>Meter</i>
inch	in	C1 _h	ja	ja	<i>Zoll</i> (=0,0254 m)
foot	ft	C2 _h	ja	ja	<i>Fuß</i> (=0,3048 m)
grade	g	40 _h	ja	nein	<i>Gon</i> (Winkeleinheit, 400 entsprechen 360°)
radian	rad	10 _h	ja	nein	<i>Radiant</i>
degree	°	41 _h	ja	nein	<i>Grad</i>
arcminute	'	42 _h	ja	nein	<i>Winkelminute</i> (60'=1°)
arcsecond	"	43 _h	ja	nein	<i>Winkelsekunde</i> (60"=1')
mechanical revolution		B4 _h	ja	nein	<i>Umdrehung</i>

Name	Einheitenzeichen	Wert	6091 _h	6092 _h	Beschreibung
encoder increment		B5 _h	nein	nein	<i>Encoder-Inkrement</i> . Abhängig vom verwendeten Sensor (Encoder/Hall-Sensor) und <u>Betriebsart</u> . Im <i>Open Loop</i> - und <i>Sensorless</i> -Betrieb entspricht die Anzahl der Polpaare (2030 _h) multipliziert mit 65536 einer Motorumdrehung.
step		AC _h	nein	nein	<i>Schritte</i> . Bei 2-phasigen Schrittmotoren entspricht die Anzahl der Polpaare (2030 _h) multipliziert mit 4 einer Umdrehung. Bei 3-phasigen BLDC-Motoren entspricht die Anzahl der Polpaare (2030 _h) multipliziert mit 6 einer Umdrehung.
electrical pole		C0 _h	nein	nein	<i>Elektrische Pole</i> . Bei einem Schrittmotor, der z.B. 50 Polpaare (2030 _h) hat, entspricht die Einheit 1/50 einer Umdrehung.
dimensionless		00 _h	ja	ja	<i>dimensionslose Längeneinheit</i>

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

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

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

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

6.3.2 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)).

6.3.3 Getriebeübersetzung

Die Getriebeübersetzung berechnet sich aus Motorumdrehungen ($60E8_h$ (Motor Shaft Revolutions)) pro Achsumdrehungen ($60ED_h$ (Driving Shaft Revolutions)).

6.3.4 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.

6.3.5 Berechnungsformeln für Benutzereinheiten

6.3.5.1 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.

6.3.5.2 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=00_h, Bits 16-23=B4_h und Bits 24-31=47_h), wird die Einheit auf *Umdrehungen pro Minute* eingestellt (Werkseinstellung).

Beispiel

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

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

HINWEIS

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

Umrechnungsfaktor für die Geschwindigkeitseinheit

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

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

6.3.5.3 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 ($6097_h:02_h$).

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

6.3.5.4 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 ($60A2_h:02_h$).

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

6.4 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.

6.4.1 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 in Objekt 3701_h hinterlegte Aktion 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 <i>Switch on disabled</i>)
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.
6	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

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.

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.

6.4.2 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.

6.5 Zykluszeiten

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

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

Task	Zykluszeit
Applikation	1 ms
NanoJ Applikation	1 ms
Stromregler	62,5 µs (16 KHz)
Geschwindigkeitsregler	250 µs (4 KHz)
Positionsregler	1 ms

7 Betriebsmodi

7.1 Profile Position

7.1.1 Übersicht

7.1.1.1 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.

7.1.1.2 Aktivierung

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

7.1.1.3 Controlword

Folgende Bits im Objekt `6040h` (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 (`607Ah`) absolut und bei "1" ist die Zielposition relativ. Die Referenzposition ist abhängig von den Bits 0 und 1 des Objekts `60F2h`.
- 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 `605Dh`.
- 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.

7.1.1.4 Statusword

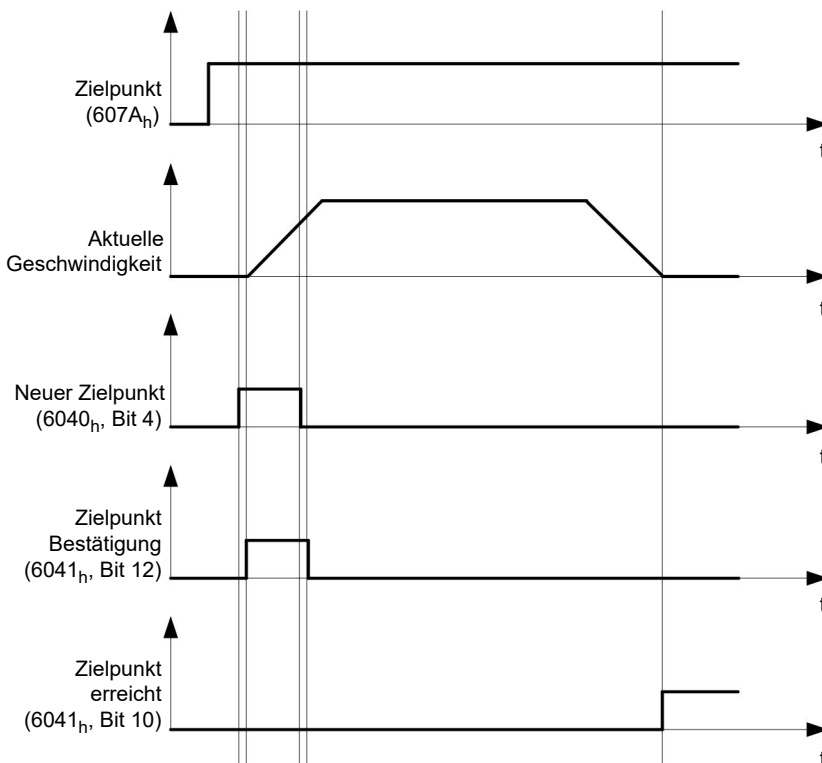
Folgende Bits im Objekt 6041_h (Statusword) haben eine gesonderte Funktion:

- 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)).

7.1.2 Setzen von Fahrbefehlen

7.1.2.1 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.



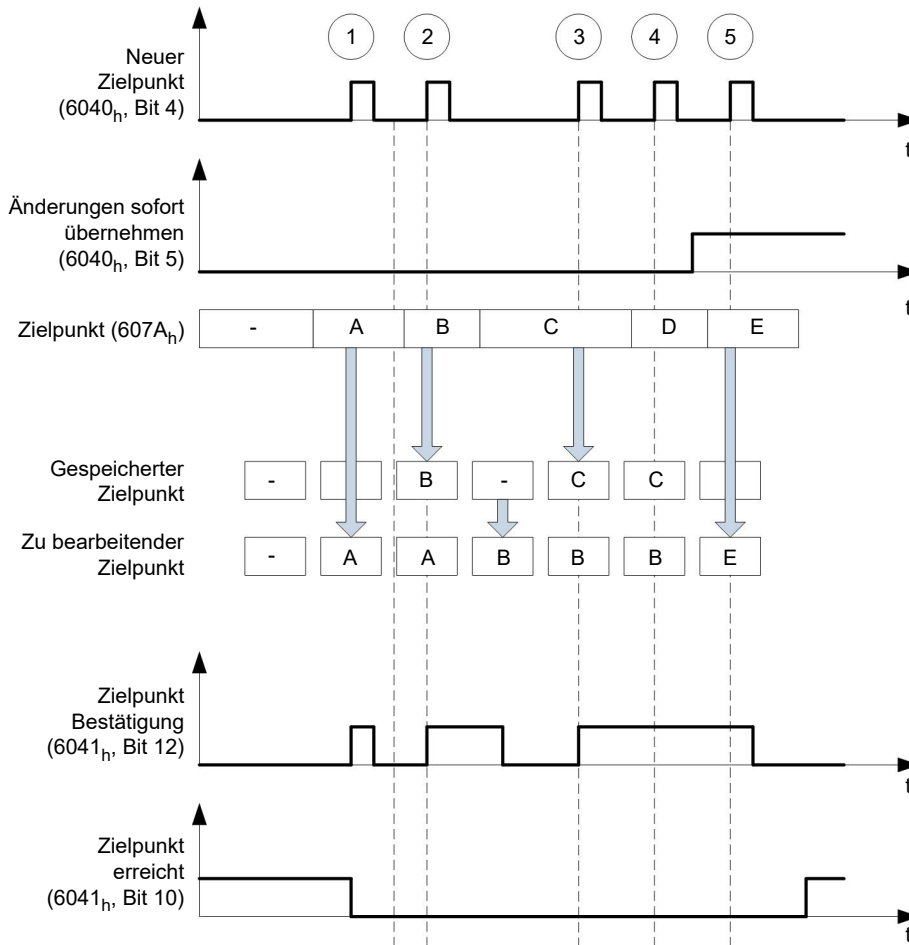
Die Steuerung kann das Bit 4 im Objekt 6040_h (Controlword) auch selbstständig zurücksetzen. Das wird mit den Bits 4 und 5 des Objektes $60F2_h$ eingestellt.

7.1.2.2 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).

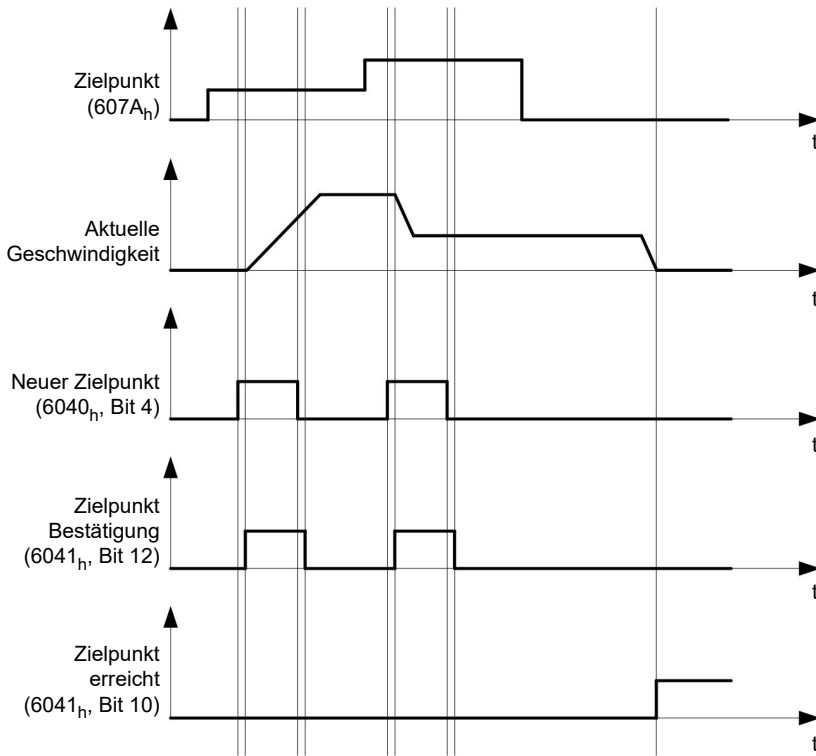
Sollte der Zwischenspeicher schon voll sein, wird ein neuer Zeitpunkt ignoriert (Zeitpunkt 4). Wird Bit 5 im Objekt 6040_h (Controlword, Bit: "Change Set-Point Immediately") gesetzt, arbeitet die Steuerung ohne den Zwischenspeicher, neue Fahrbefehle werden direkt umgesetzt (Zeitpunkt 5).

Zeitpunkte



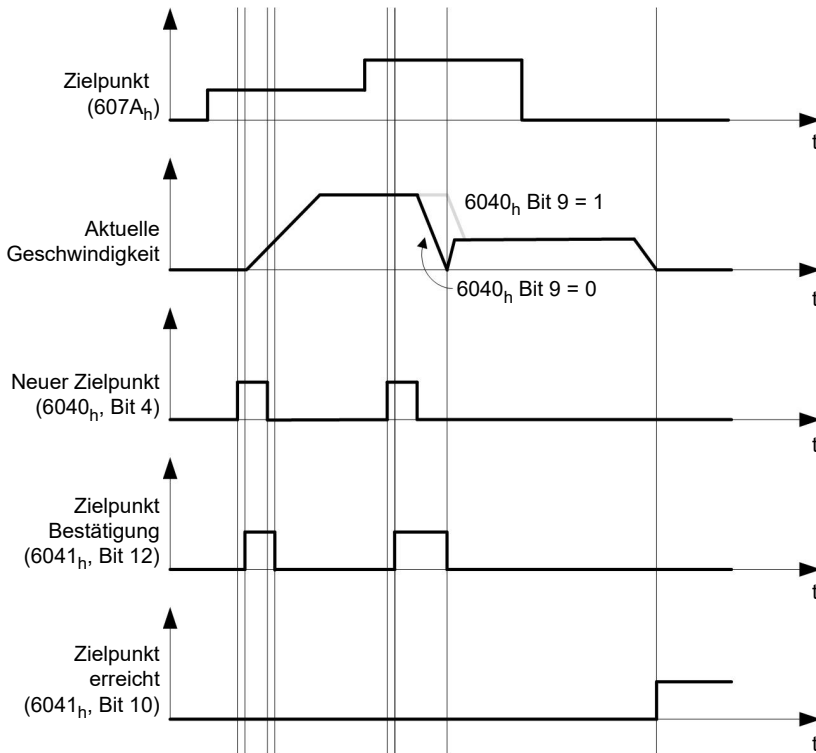
Übergangsprozedur für zweite Zielposition

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.



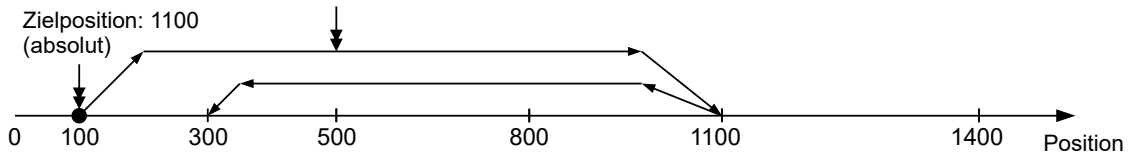
Mögliche Kombinationen von Fahrbefehlen

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

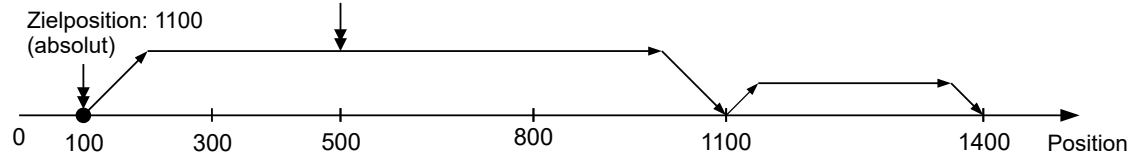
Die nachfolgenden Bilder setzen voraus:

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

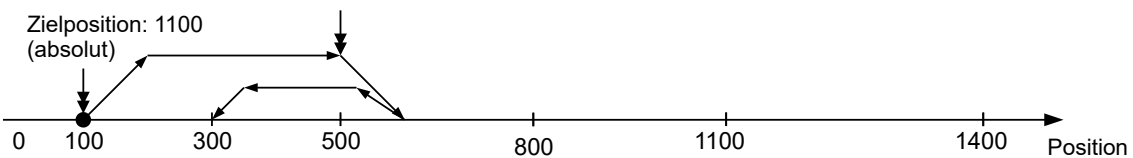
- Änderung im Zielpunkt übernehmen ($6040_n:00$ Bit 5 = 0)
- Positionierung absolut ($6040_n:00$ Bit 6 = 0)
- Zielposition: 300



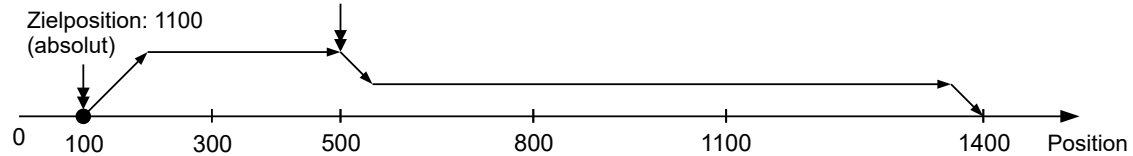
- Relativ zu der vorhergehenden Zielposition ($60F2_n:00 = 0$)
- Änderung im Zielpunkt übernehmen ($6040_n:00$ Bit 5 = 0)
- Positionierung relativ ($6040_n:00$ Bit 6 = 1)
- Zielposition: 300



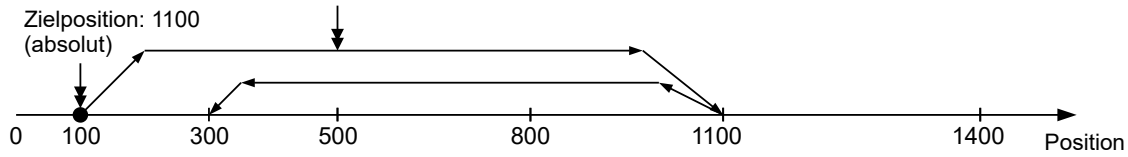
- Änderung sofort übernehmen ($6040_n:00$ Bit 5 = 1)
- Positionierung absolut ($6040_n:00$ Bit 6 = 0)
- Zielposition: 300



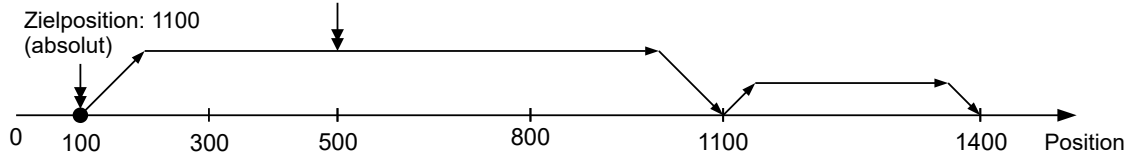
- Relativ zu der vorhergehenden Zielposition ($60F2_n:00 = 0$)
- Änderung sofort übernehmen ($6040_n:00$ Bit 5 = 1)
- Positionierung relativ ($6040_n:00$ Bit 6 = 1)
- Zielposition: 300



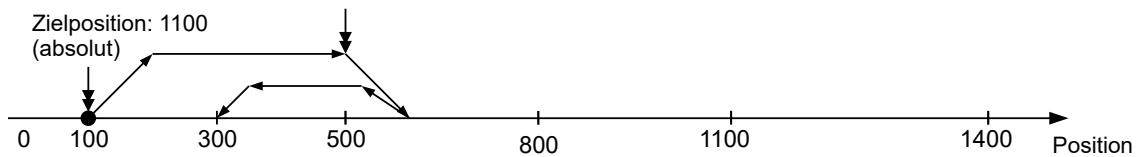
- Änderung im Zielpunkt übernehmen (6040_n:00 Bit 5 = 0)
- Positionierung absolut (6040_n:00 Bit 6 = 0)
- Zielposition: 300



- Relativ zu der aktuellen Position (60F2_n:00 = 1)
- Änderung im Zielpunkt übernehmen (6040_n:00 Bit 5 = 0)
- Positionierung relativ (6040_n:00 Bit 6 = 1)
- Zielposition: 300

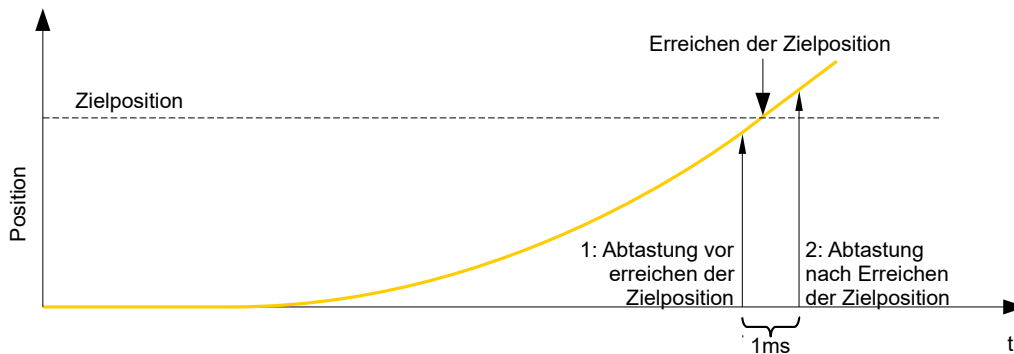


- Änderung sofort übernehmen (6040_n:00 Bit 5 = 1)
- Positionierung absolut (6040_n:00 Bit 6 = 0)
- Zielposition: 300



7.1.3 Genauigkeitsverlust bei Relativbewegungen

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



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

7.1.4 Randbedingungen für eine Positionierfahrt

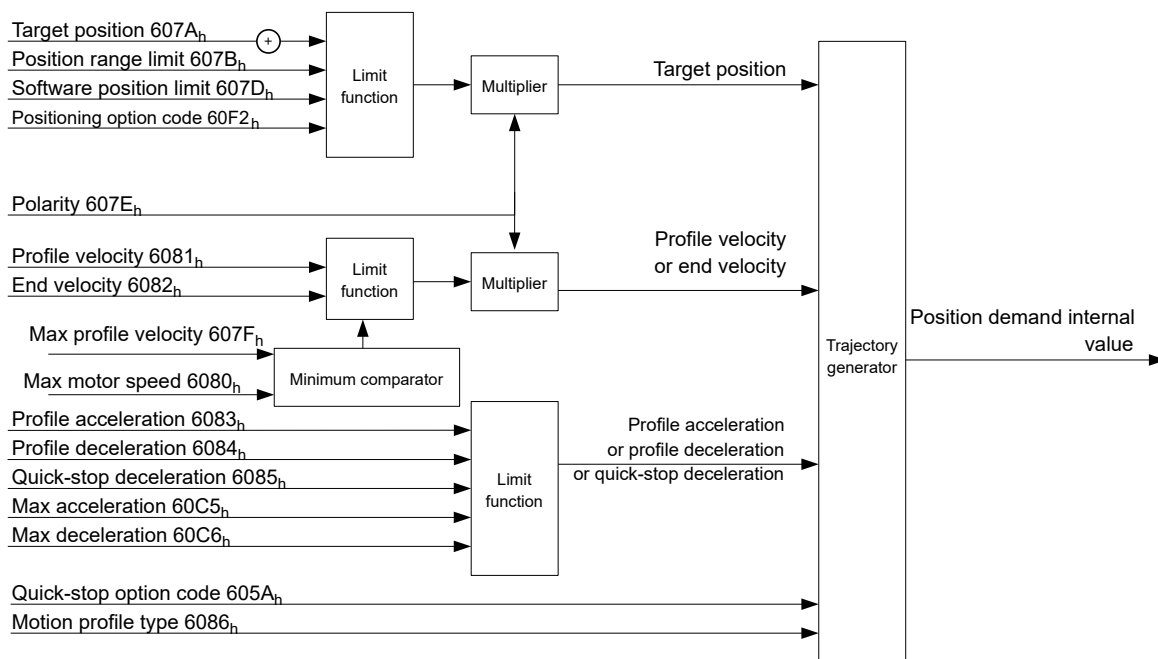
7.1.4.1 Objekteinträge

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

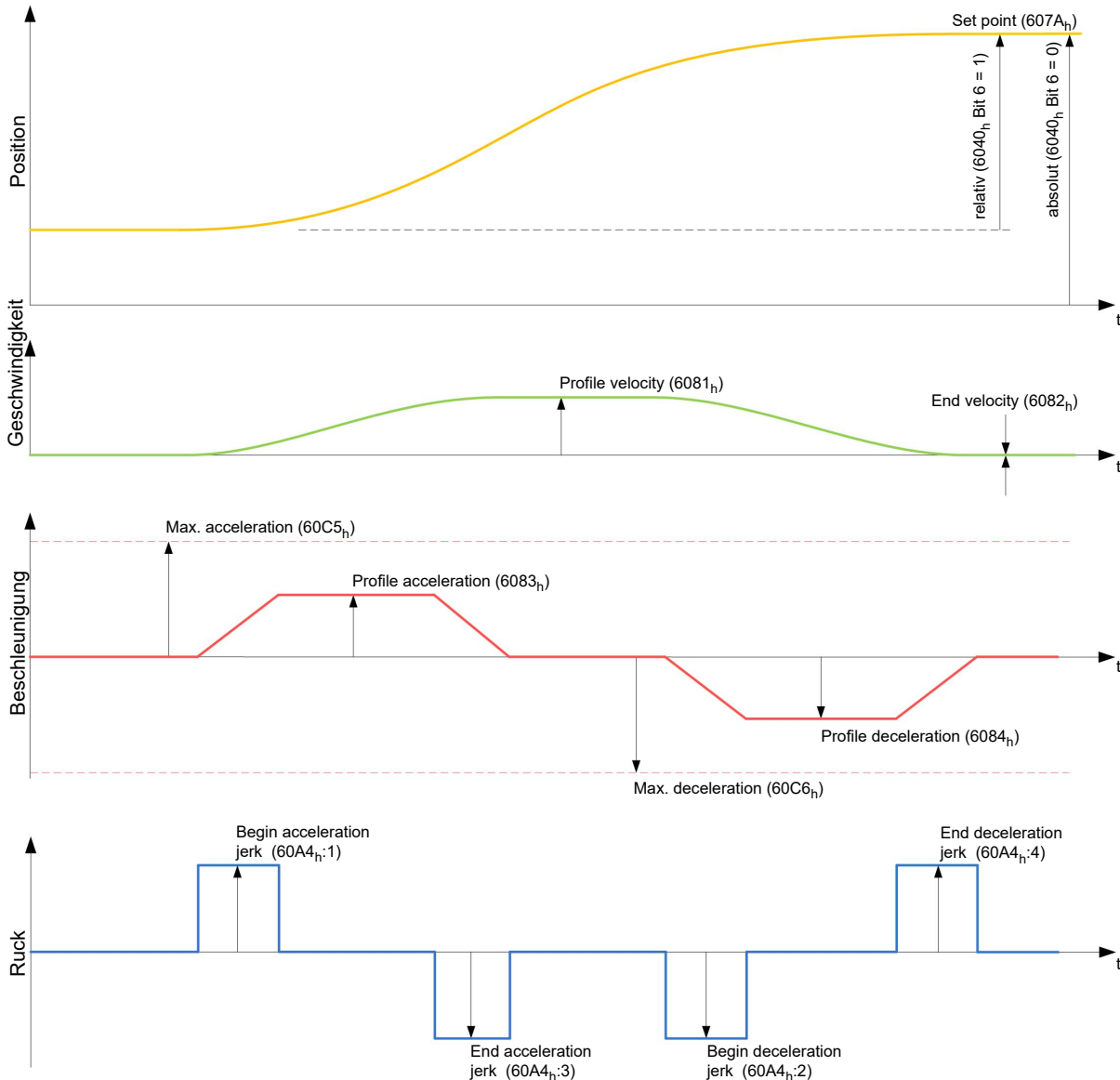
7.1.4.2 Objekte für die Positionierfahrt

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



7.1.4.3 Parameter für die Zielposition

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



7.1.5 Ruck-begrenzter und nicht ruck-begrenzter Modus

7.1.5.1 Beschreibung

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

7.1.5.2 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.

7.1.5.3 Nicht ruck-begrenzter Modus

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

7.2 Velocity

7.2.1 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.

7.2.2 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).

7.2.3 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.

7.2.4 Statusword

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

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

7.2.5 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:

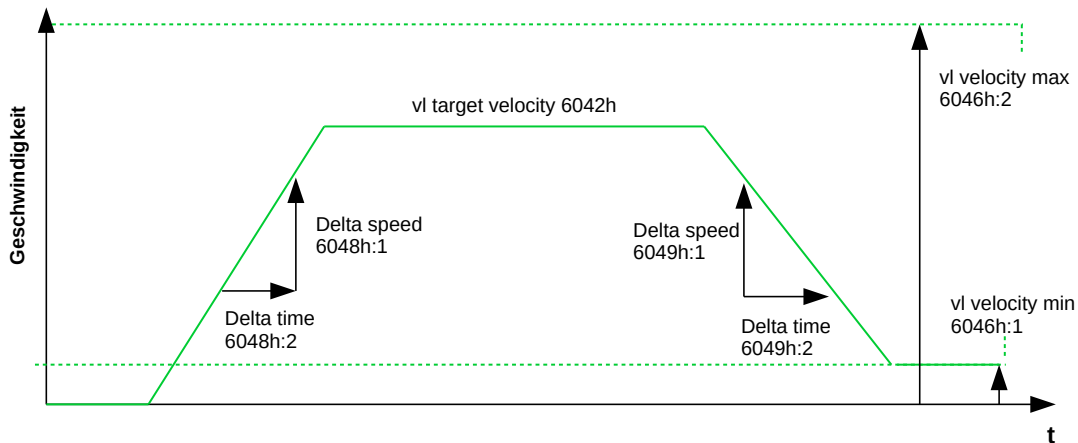
$$\text{VL velocity acceleration} = \frac{\text{Delta speed (6048}_{h}:1)}{\text{Delta time (6048}_{h}:2)}$$

- 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:

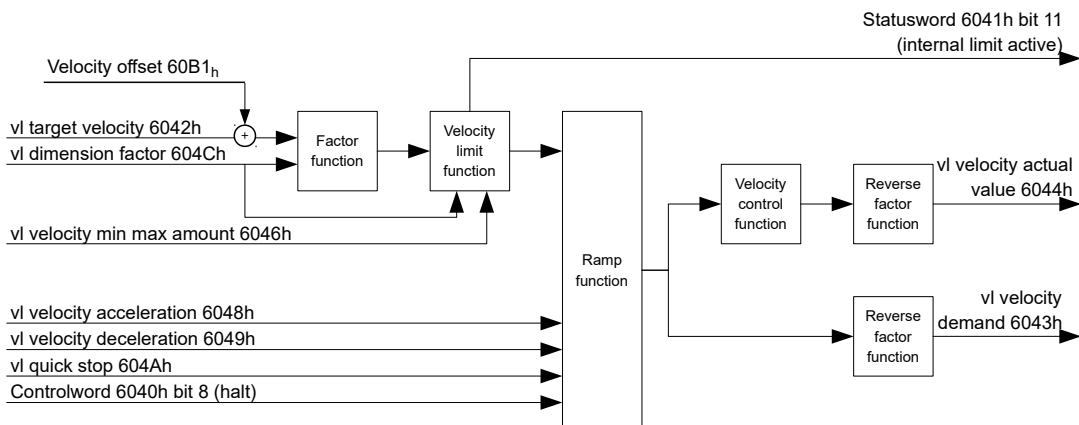
- 6043_h (VI Velocity Demand)
- 6044_h (VI Velocity Actual Value)

7.2.5.1 Geschwindigkeiten im Velocity Mode



7.2.5.2 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).



7.3 Profile Velocity

7.3.1 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.

7.3.2 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").

7.3.3 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.

7.3.4 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 <u>606D_h</u> und <u>606E_h</u>)
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 (60F8_h Max Slippage und 203F_h Max Slippage Time Out).

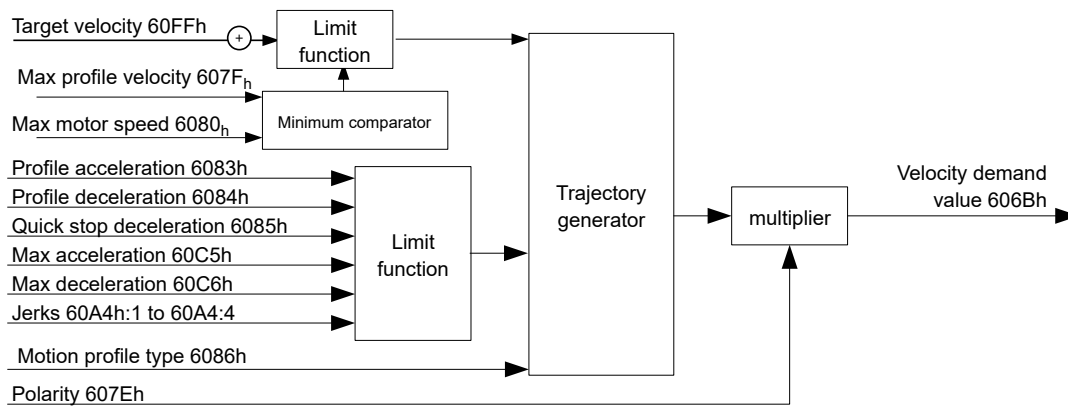
7.3.5 Objekteinträge

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).
- 60FE_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.

7.3.5.1 Objekte im Profile Velocity Mode

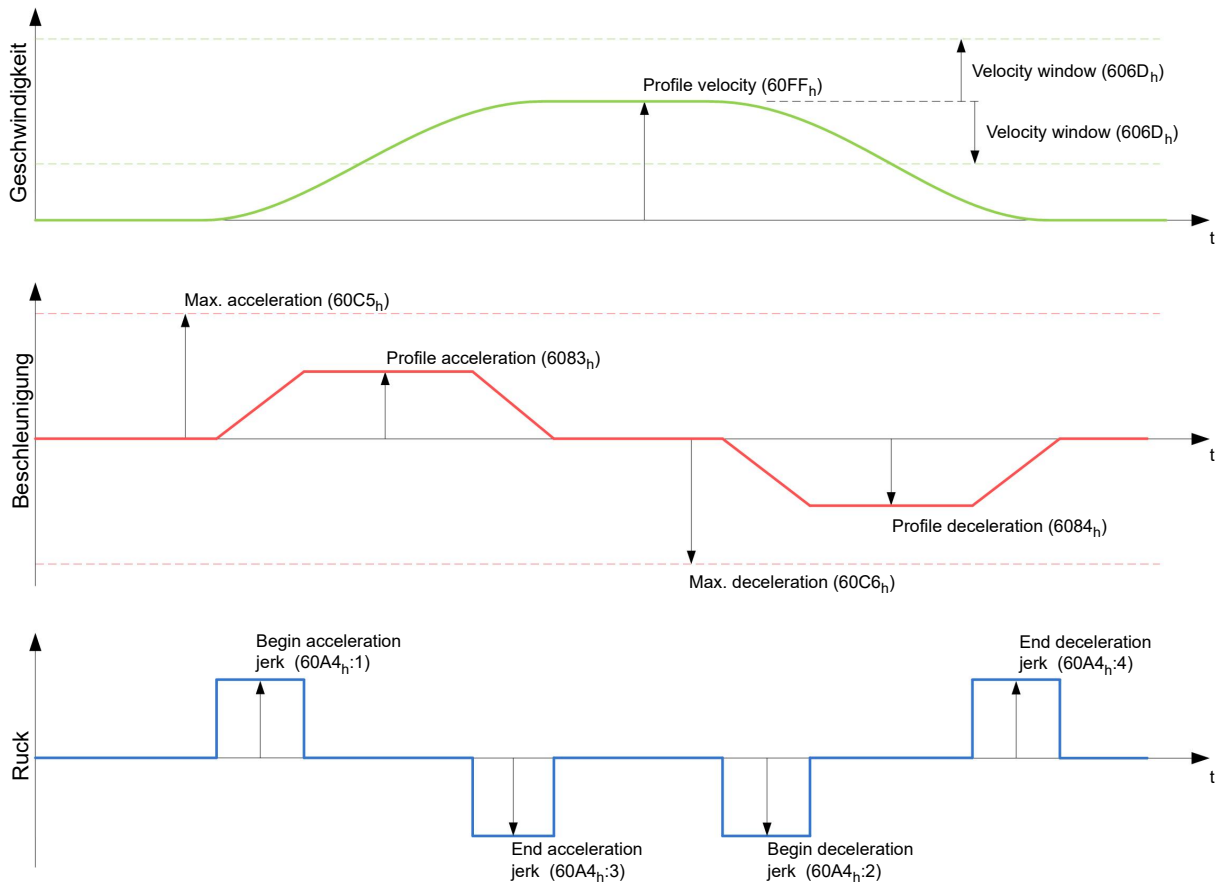


7.3.5.2 Aktivierung

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.

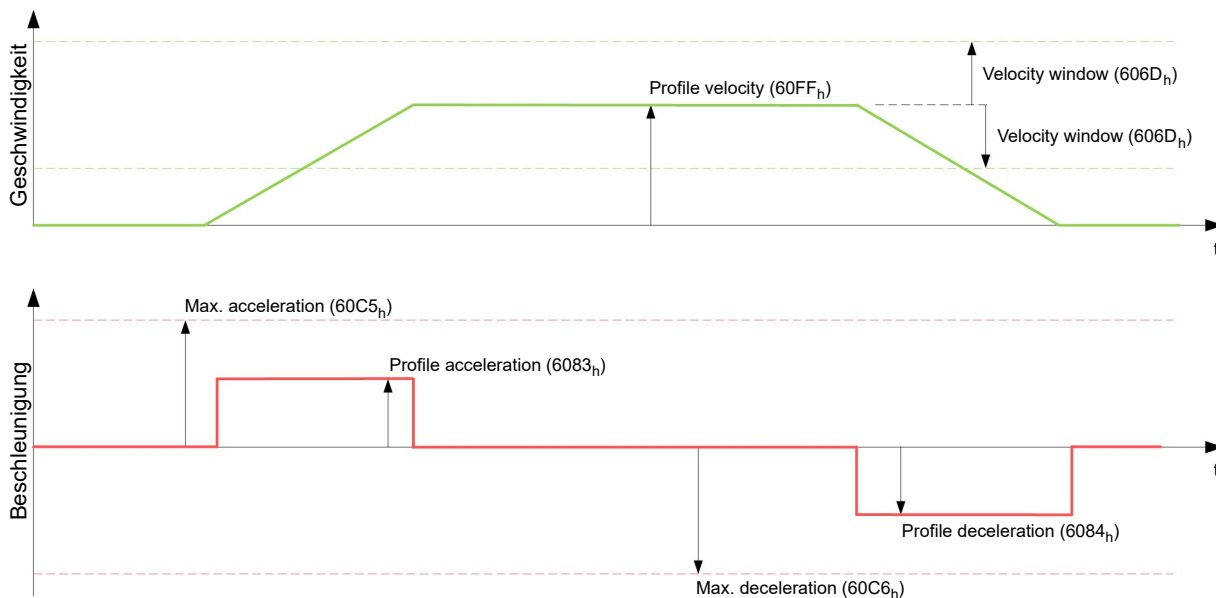
7.3.5.3 Limitierungen im ruck-limitierten Fall

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



7.3.5.4 Limitierungen im Trapez-Fall

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



7.4 Profile Torque

7.4.1 Beschreibung

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

7.4.2 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").

7.4.3 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.

7.4.4 Statusword

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

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

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

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

7.4.5 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

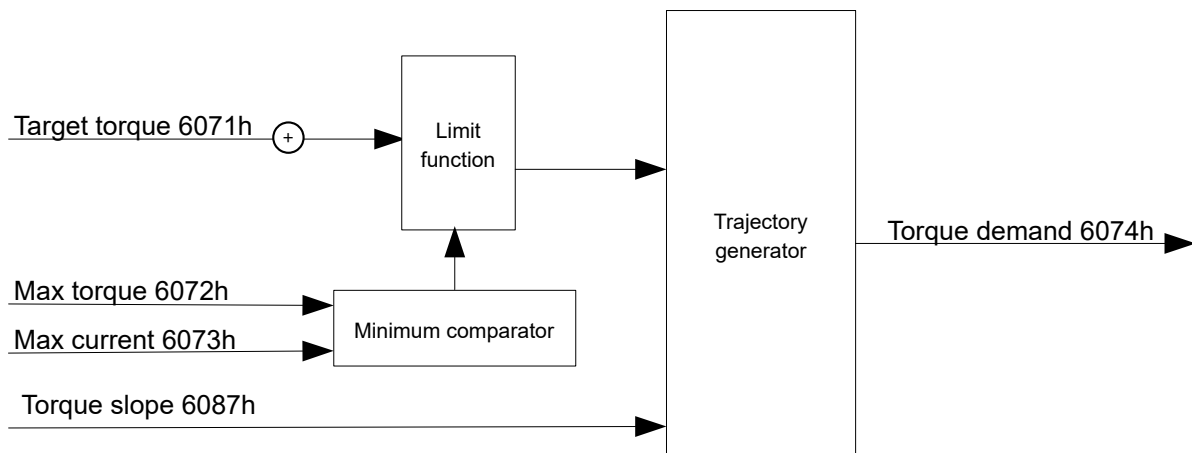


Diese Werte sind nicht limitiert auf 100% des Nennstroms (203B_h:01_h). Drehmomentwerte höher als das Nenndrehmoment (generiert von dem Nennstrom) können erreicht werden, wenn die Maximaldauer (203B_h:02_h) des maximalen Stroms (6073_h) gesetzt wird (siehe 12t Motor-Überlastungsschutz). Alle Drehmoment-Objekte werden von dem maximalen Motorstrom (2031_h) limitiert.

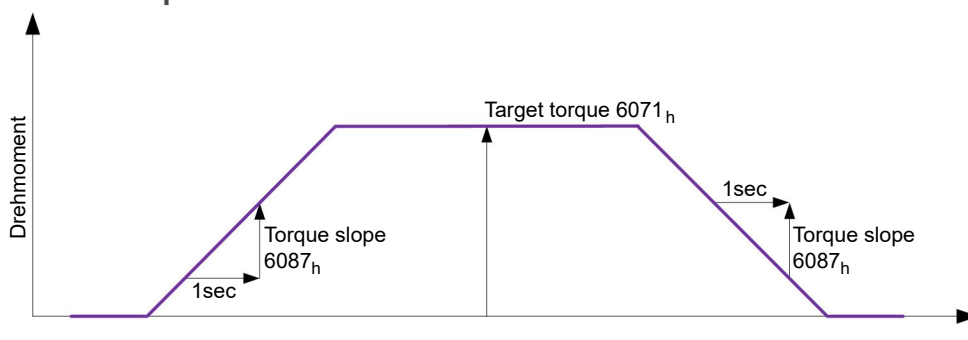
Die folgenden Objekte werden zudem für diesen Operationsmodus benötigt:

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

7.4.5.1 Objekte des Rampengenerators



7.4.5.2 Torque-Verlauf



7.5 Homing

7.5.1 Übersicht

7.5.1.1 Beschreibung

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

7.5.1.2 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 `3701h` auf "-1" setzen (Werkseinstellung), damit die weitere Fahrt des Motors nicht blockiert wird.

7.5.1.3 Controlword

Folgende Bits im Objekt `6040h` (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.

7.5.1.4 Statusword

Folgende Bits im Objekt `6041h` (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).

7.5.1.5 Objekteinträge

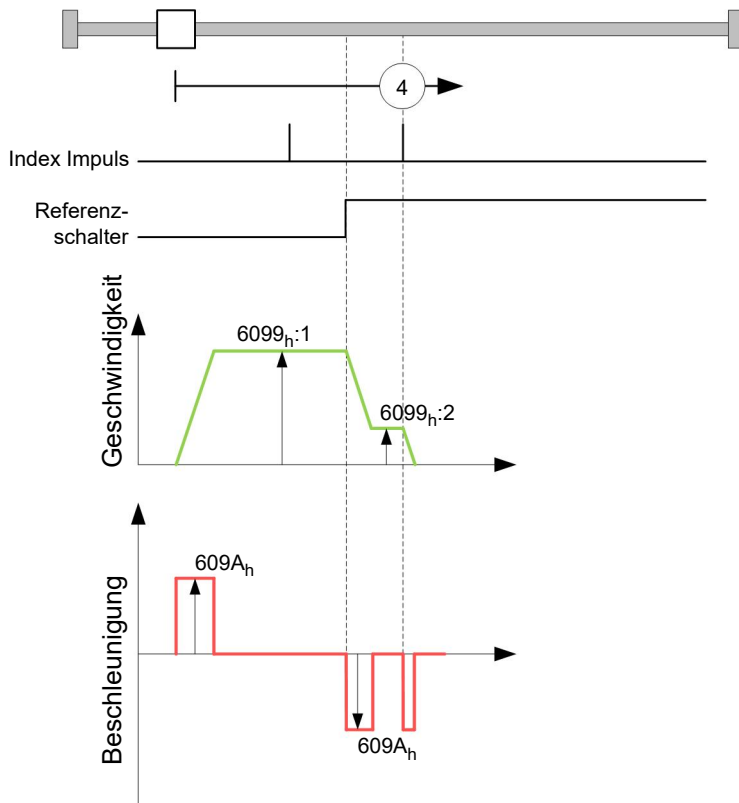
Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

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



7.5.2 Referenzfahrt-Methode

7.5.2.1 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.

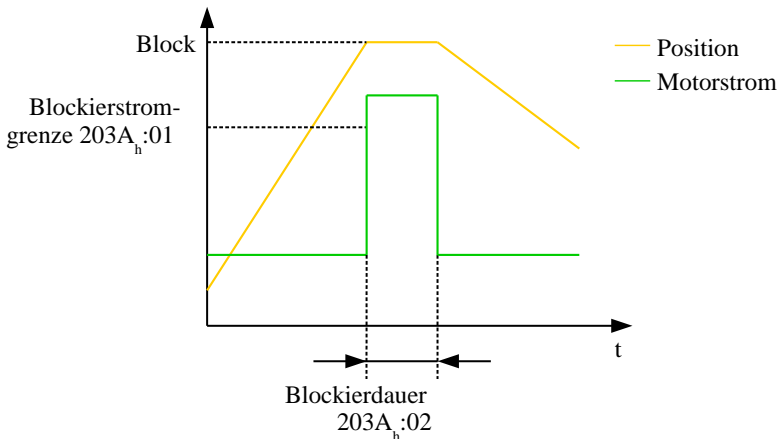
Bei Methoden, die Homing auf Block benutzen, gelten die gleichen Abbildungen wie für die Methoden mit Endschalter. Da sich außer den fehlenden Endschaltern nichts ändert, wurde auf neue Abbildungen verzichtet. Hier gilt für die Abbildungen, dass die Endschalter durch eine mechanische Blockierung ersetzt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

7.5.2.2 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.



7.5.2.3 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.

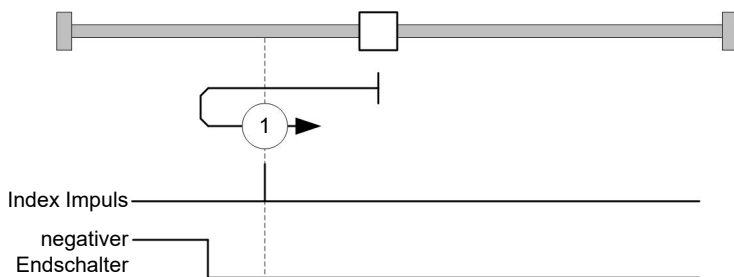
Folgende Methoden können für Homing auf Block benutzt werden:

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

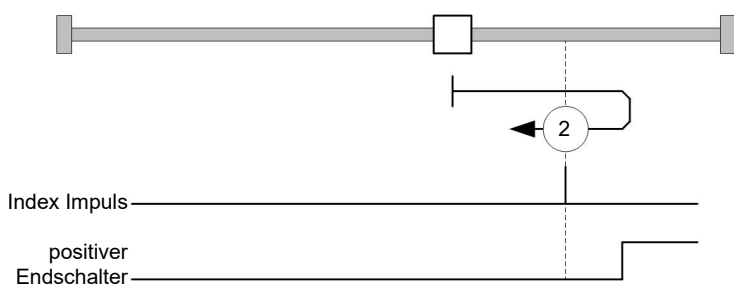
7.5.2.4 Methoden 1 und 2

Referenzieren auf Endschalter und Index-Impuls.

Methode 1 referenziert auf negativen Endschalter und Index-Impuls:



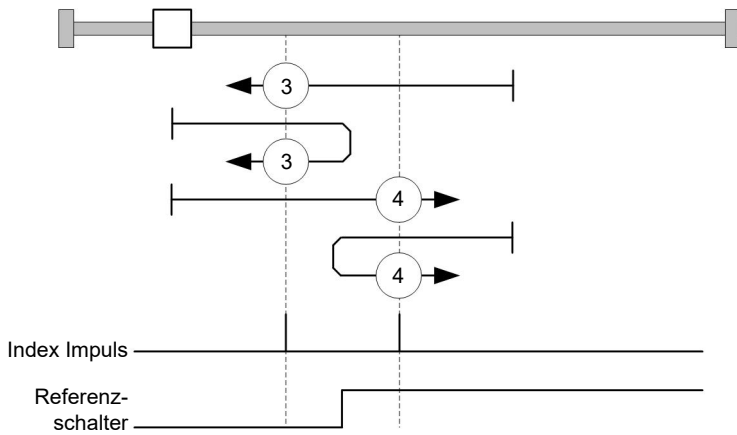
Methode 2 referenziert auf positiven Endschalter und Index-Impuls:



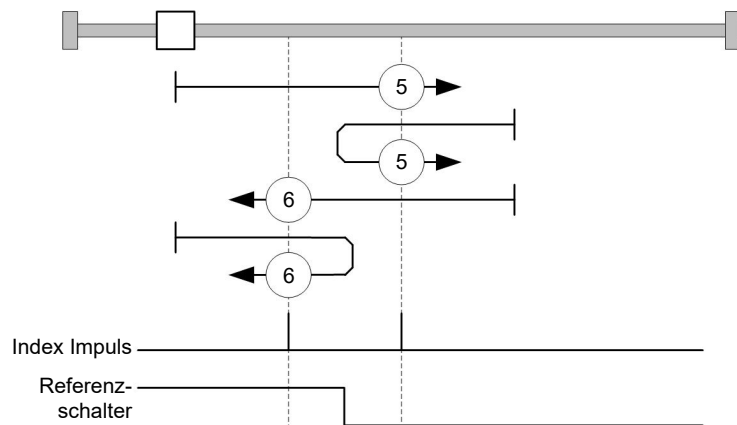
7.5.2.5 Methoden 3 bis 6

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:

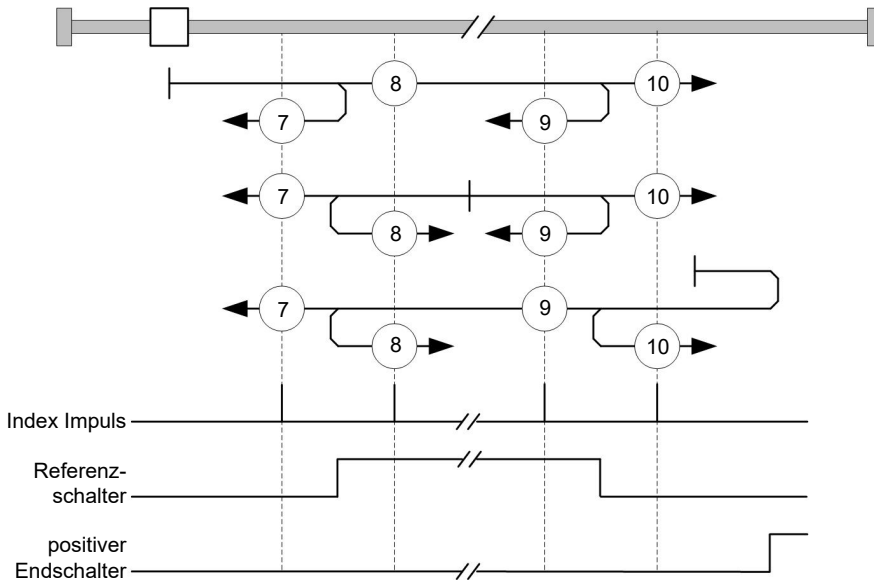


7.5.2.6 Methoden 7 bis 14

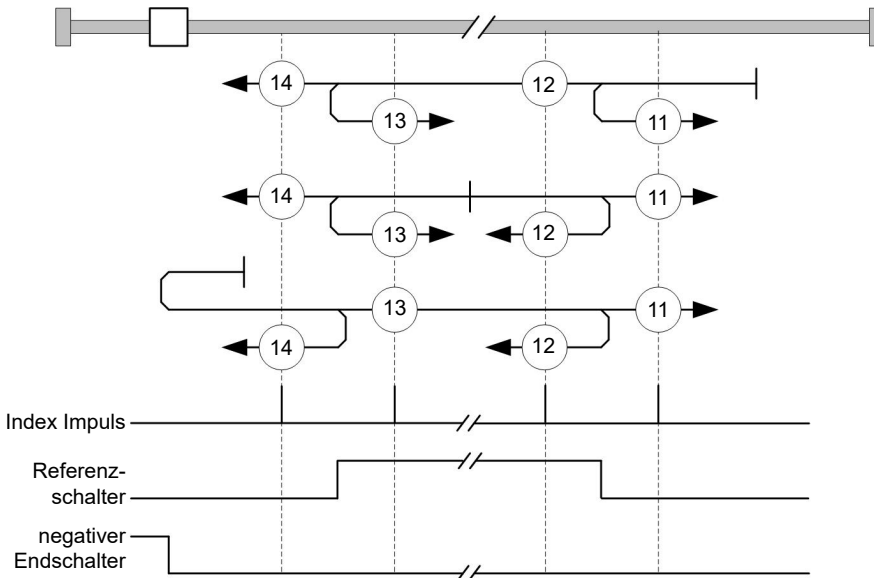
Referenzieren auf Referenzschalter und Index-Impuls (mit Endschaltern).

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:



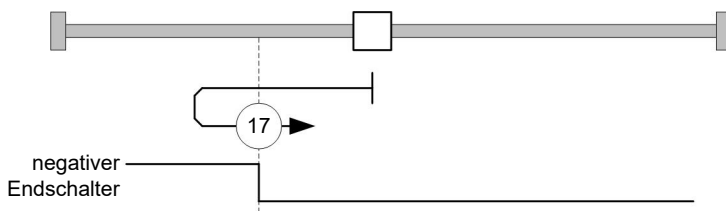
Die Methoden 11 bis 14 berücksichtigen den negativen Endschalter:



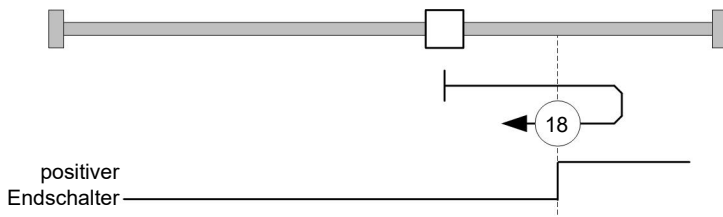
7.5.2.7 Methoden 17 und 18

Referenzieren auf den Endschalter ohne den Index-Impuls.

Methode 17 referenziert auf den negativen Endschalter:



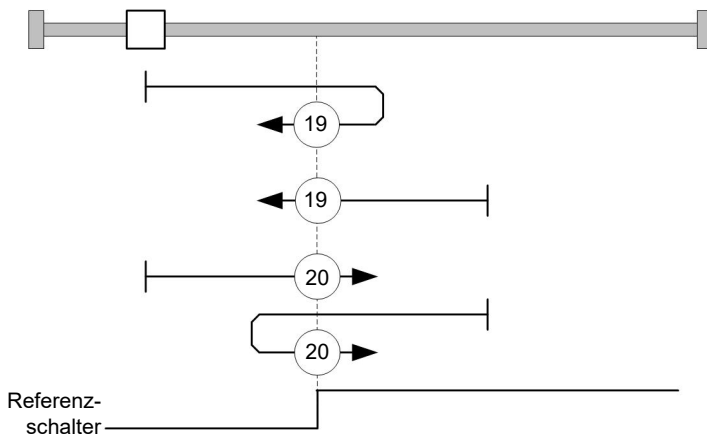
Methode 18 referenziert auf den positiven Endschalter:



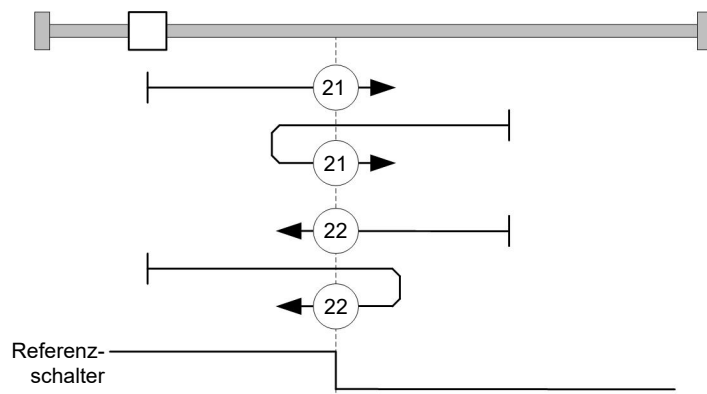
7.5.2.8 Methoden 19 bis 22

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:

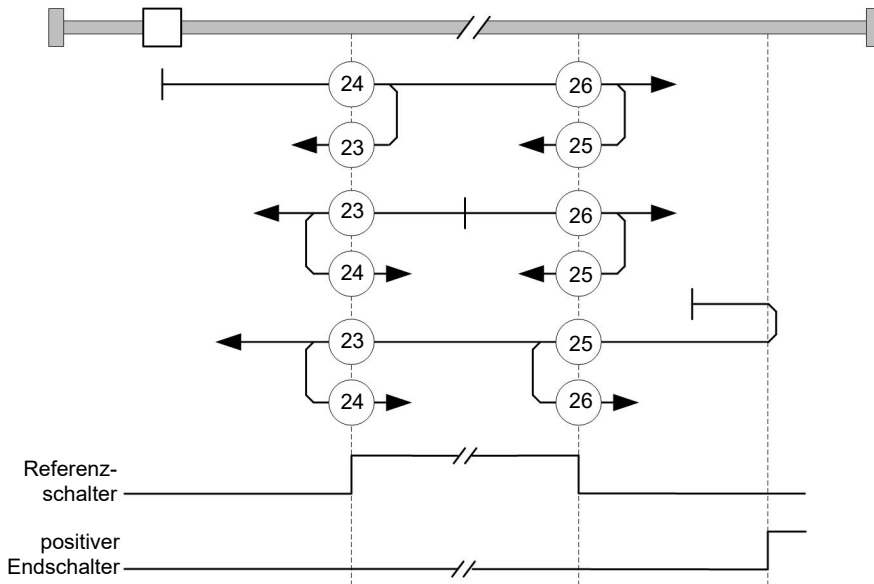


7.5.2.9 Methoden 23 bis 30

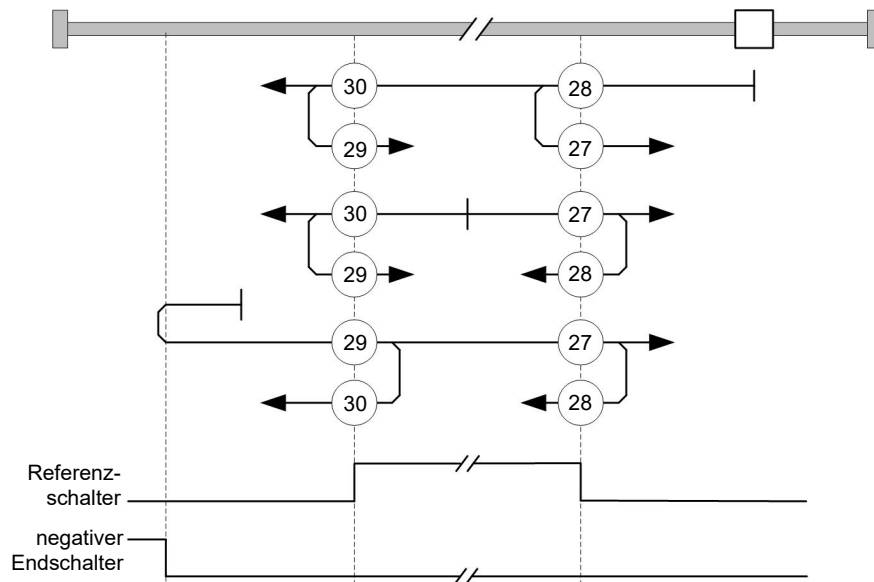
Referenzieren auf Referenzschalter ohne den Index-Impuls (mit Endschaltern).

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:



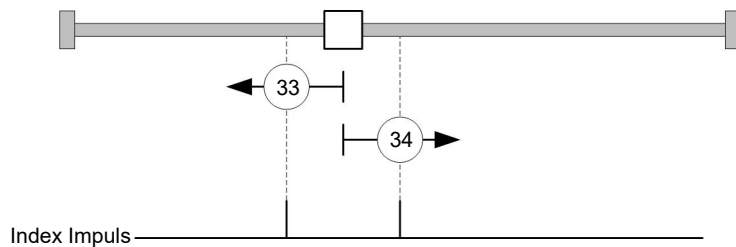
Die Methoden 27 bis 30 berücksichtigen den negativen Referenzschalter:



7.5.2.10 Methoden 33 und 34

Referenzieren auf den nächsten Index-Impuls.

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



7.5.2.11 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.

7.6 Interpolated Position Mode

7.6.1 Übersicht

7.6.1.1 Beschreibung

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

7.6.1.2 Synchronisierung zum SYNC-Objekt

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



HINWEIS

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

7.6.2 Aktivierung

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

7.6.3 Controlword

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

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

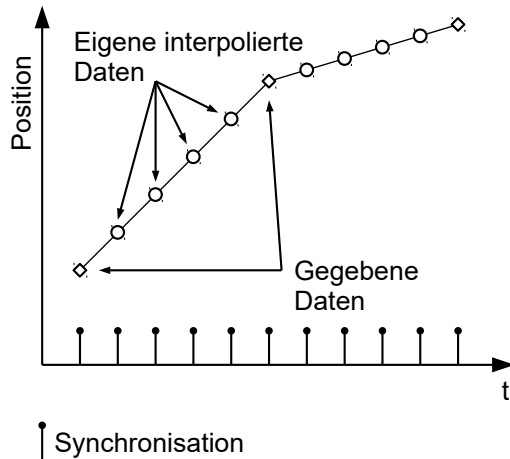
7.6.4 Statusword

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

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

7.6.5 Benutzung

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



In der derzeitigen Implementation wird nur

- lineare Interpolation
- und eine Zielposition

unterstützt.

7.6.6 Setup

Das folgende Setup ist nötig:

- `60C2h:01h`: Zeit zwischen zwei übergebenen Zielpositionen in ms.
- `60C4h:06h`: dieses Objekt ist auf "1" zu setzen um die Zielposition im Objekt `60C1h:01h` modifizieren zu dürfen.
- `6081h` (Profile Velocity): maximale Geschwindigkeit, mit der die Position angefahren werden soll
- `6084h` (Profile deceleration): gewünschte Bremsbeschleunigung beim Abbremsen
- `60C6h` (Max Deceleration): die maximal erlaubte Bremsbeschleunigung
- Nur wenn der Closed Loop aktiviert ist: Die Geschwindigkeit wird durch `607Fh` (Max Profile Velocity) und `6080h` (Max Motor Speed) begrenzt, der kleinere Wert wird als Grenze herangezogen.
- Um den Motor drehen zu können, ist die *Power state machine* auf den Status *Operation enabled* zu setzen (siehe CiA 402 Power State Machine).

7.6.7 Operation

Nach dem Setup ist die Aufgabe der übergerodeten Steuerung, die Zielpositionen rechtzeitig in das Objekt `60C1h:01h` zu schreiben.

7.7 Cyclic Synchronous Position

7.7.1 Übersicht

7.7.1.1 Beschreibung

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

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



HINWEIS

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

7.7.1.2 Synchronisierung zum SYNC-Objekt

Um eine gleichmäßige Bewegung zu erzielen, ist es sinnvoll, dass sich die Steuerung auf das SYNC-Objekt (abhängig vom Feldbus) aufsynchronisiert. Dieses SYNC-Objekt ist in regelmäßigen Zeitabständen von der übergeordneten Steuerung zu senden. Die Synchronisation erfolgt, sobald die Steuerung in den NMT-Modus *Operational* geschaltet wird.

HINWEIS



Es wird empfohlen, wenn möglich ein Zeitintervall des *SYNC-Objekts* für die Übertragung der Zielposition zu nutzen.

7.7.1.3 Aktivierung

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

7.7.1.4 Controlword

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

7.7.1.5 Statusword

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

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

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

7.7.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

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

- **605A_h** (Quick-Stop Option Code): Dieses Objekt enthält die Option, die im Falle eines Quick-Stops ausgeführt werden soll.
- Nur wenn der Closed Loop aktiviert ist: **6080_h** (Max Motor Speed): maximale Geschwindigkeit
- **60C2_h:01_h** (Interpolation Time Period): Dieses Objekt gibt die Zeit eines *Zyklus* vor, in diesen Zeitabständen muss ein neuer Sollwert in das **607A_h** geschrieben werden.
Es gilt dabei: Zykluszeit = Wert des **60C2_h:01_h** * 10^{Wert des 60C2:02} Sekunden.
- **60C2_h:02_h** (Interpolation Time Index): Dieses Objekt gibt die Zeitbasis der Zyklen an. Derzeit wird nur der Wert **60C2_h:02_h**=-3 unterstützt, das ergibt eine Zeitbasis von 1 Millisekunde.
- **60B0_h** (Position Offset): Offset für den Positionssollwert in benutzerdefinierten Einheiten
- **60B1_h** (Velocity Offset): Offset für den Geschwindigkeitssollwert in benutzerdefinierten Einheiten
- **60B2_h** (Torque Offset): Offset für den Drehmomentsollwert in Promille

Folgende Objekte können in dem Modus ausgelesen werden:

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

7.8 Cyclic Synchronous Velocity

7.8.1 Übersicht

7.8.1.1 Beschreibung

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

7.8.1.2 Aktivierung

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

7.8.1.3 Controlword

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

7.8.1.4 Statusword

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

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

7.8.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- $60FF_h$ (Target Velocity): Dieses Objekt muss zyklisch mit dem Geschwindigkeits-Sollwert beschrieben werden.
- 6085_h (Quick-Stop Deceleration): Dieses Objekt hält die Bremsbeschleunigung für den Fall, dass ein Quick-Stop ausgelöst wird (siehe "[CiA 402 Power State Machine](#)").
- $605A_h$ (Quick-Stop Option Code): Dieses Objekt enthält die Option, die im Falle eines Quick-Stops ausgeführt werden soll (siehe "[CiA 402 Power State Machine](#)").
- 6080_h (Max Motor Speed): maximale Geschwindigkeit
- $60C2_h:01_h$ (Interpolation Time Period): Dieses Objekt gibt die Zeit eines *Zyklus* vor, in diesen Zeitabständen muss ein neuer Sollwert in das $60FF_h$ geschrieben werden.
Es gilt dabei: Zykluszeit = Wert des $60C2_h:01_h$ * $10^{\text{Wert des } 60C2:02}$ Sekunden.
- $60C2_h:02_h$ (Interpolation Time Index): Dieses Objekt gibt die Zeitbasis der Zyklen an. Derzeit wird nur der Wert $60C2_h:02_h=-3$ unterstützt, das ergibt eine Zeitbasis von 1 Millisekunde.
- $60B1_h$ (Velocity Offset): Offset für den Geschwindigkeitssollwert in benutzerdefinierten Einheiten
- $60B2_h$ (Torque Offset): Offset für den Drehmomentsollwert in Promille

Folgende Objekte können in dem Modus ausgelesen werden:

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

7.9 Cyclic Synchronous Torque

7.9.1 Übersicht

7.9.1.1 Beschreibung

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

HINWEIS



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

7.9.1.2 Aktivierung

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

7.9.1.3 Controlword

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

7.9.1.4 Statusword

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

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

Bit	Wert	Beschreibung
13	0	Reserviert
13	1	Reserviert

7.9.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **6071_h** (Target Torque): Dieses Objekt muss zyklisch mit dem Drehmoment-Sollwert beschrieben werden und ist relativ zu **6072_h** einzustellen.
- **6072_h** (Max Torque): Beschreibt das maximal zulässige Drehmoment.
- **6073_h** (Max Current):
Maximaler Strom. Das Minimum von **6073_h** und **6072_h** wird als Limit für das Drehmoment in **6071_h** verwendet.
- **6080_h** (Max Motor Speed): maximale Geschwindigkeit
- **60C2_h:01_h** (Interpolation Time Period): Dieses Objekt gibt die Zeit eines *Zyklus* vor, in diesen Zeitabständen muss ein neuer Sollwert in das **6071_h** geschrieben werden.
Es gilt dabei: $\text{Zykluszeit} = \text{Wert des } 60C2_{h:01_{h}} * 10^{\text{Wert des } 60C2:02} \text{ Sekunden.}$
- **60C2_h:02_h** (Interpolation Time Index): Dieses Objekt gibt die Zeitbasis der Zyklen an. Derzeit wird nur der Wert **60C2_h:02_h=-3** unterstützt, das ergibt eine Zeitbasis von 1 Millisekunde.
- **60B2_h** (Torque Offset): Offset für den Drehmomentsollwert in Promille

Folgende Objekte können in dem Modus ausgelesen werden:

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

7.10 Takt-Richtungs-Modus

7.10.1 Beschreibung

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

7.10.2 Aktivierung

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

HINWEIS

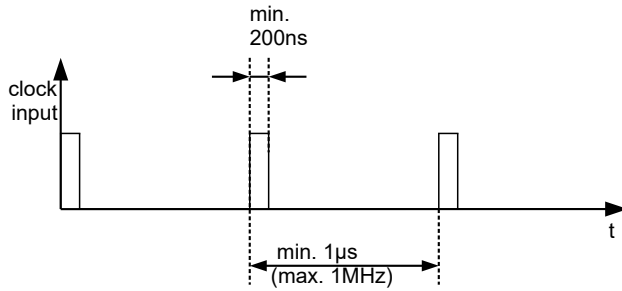


Um diesen Modus zu verwenden, müssen Sie die Pins für Takt und Richtung entsprechend konfigurieren, wie im Kapitel [Ein- und Ausgangsbelegung festlegen](#) beschrieben.

7.10.3 Generelles

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

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



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

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

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

HINWEIS



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

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

HINWEIS



Bei einem Richtungswechsel ist es nötig, mindestens eine Zeit von $35\mu\text{s}$ verstreichen zu lassen, bevor der neue Takt angelegt wird.

7.10.4 Statusword

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

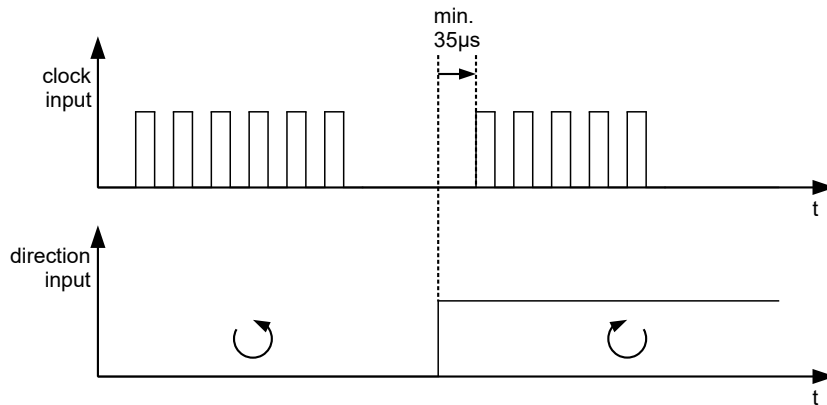
- Bit 13 (Following Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schleppfehler größer als die eingestellten Grenzen ist (6065_h (Following Error Window) und 6066_h (Following Error Time Out)).

7.10.5 Unterarten des Takt-Richtungs-Modus

7.10.5.1 Takt-Richtungs-Modus (TR-Modus)

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

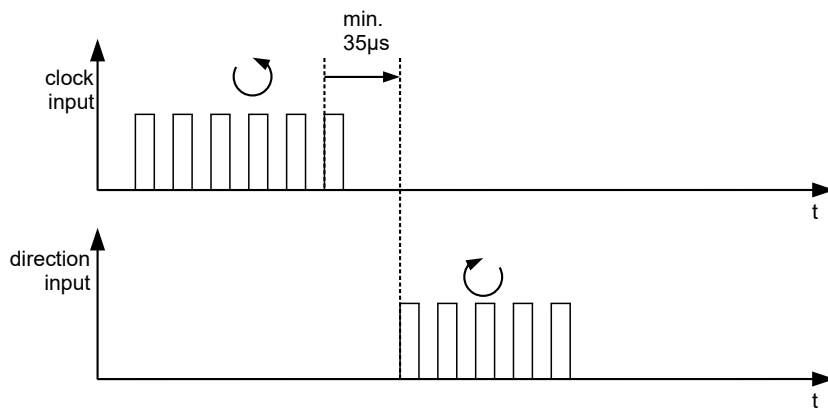
In diesem Modus müssen über den Takteingang die Pulse vorgegeben werden, das Signal des Richtungseingangs gibt dabei die Drehrichtung vor (siehe nachfolgende Grafik).



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

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

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



7.11 Auto-Setup

7.11.1 Beschreibung

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

7.11.2 Aktivierung

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

7.11.3 Controlword

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

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

7.11.4 Statusword

Folgende Bits im Objekt `6041h` (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

8 Spezielle Funktionen

8.1 Digitale Ein- und Ausgänge

Diese Steuerung verfügt über 24 I/O Pins, die wahlweise als Eingang oder Ausgang konfiguriert werden.

8.1.1 Ein- und Ausgangsbelegung festlegen

Einige Pins können frei belegt werden. Welche das sind, entnehmen Sie der nachfolgenden Tabelle. Die Belegung im Auslieferungszustand finden Sie im Kapitel [Anschlussbelegung](#).

Im entsprechenden Subindex des Objekts 3272h (Usage Of Pins Available On Connector) legen Sie für jeden konfigurierbaren Pin dessen Funktion fest. Alle in der nachfolgenden Tabelle aufgelisteten Pins können folgende Funktionen zugewiesen bekommen:

- digitaler Eingang oder Ausgang
- *Input Range Selection*
- *Analog Input Control*
- *Capture Input*
- *SPI Chip Select*

Folgende Pins unterstützen zusätzlich alternative Funktionen:

Pin	Alt. Funktion 1	Alt. Funktion 2	Subindex in 3272 _h
4, ANA1	Analogeingang 1 (Werkseinstellung)		01 _h
6, ANA2	Analogeingang 2 (Werkseinstellung)		02 _h
18, H1	Hallsensor-Eingang 1 (Werkseinstellung)		03 _h
19, DIO11			04 _h
20, H2	Hallsensor-Eingang 2 (Werkseinstellung)		05 _h
21, DIO13		<u>SPI</u> Data OUT (MOSI)	06 _h
22, H3	Hallsensor-Eingang 3 (Werkseinstellung)		07 _h
23, ENC1B	inkr. Encoder 1, B (Werkseinstellung)	SSI-Encoder 2, Data IN	08 _h
24, ENC1A	inkr. Encoder 1, A (Werkseinstellung)	SSI-Encoder 2, Clock	09 _h
25, SSI1_MISO	SSI-Encoder 1, Data IN (Werkseinstellung)	<u>SPI</u> Data IN (MISO)	0A _h
26, ENC1I	inkr. Encoder 1, Index (Werkseinstellung)		0B _h
27, SSI1_CLK	SSI-Encoder 1, Clock (Werkseinstellung)	<u>SPI</u> Clock	0C _h
29, DIO14			0D _h
39, DIO1	Takteingang in <u>Takt- Richtungs-Modus</u>	Kanal A des <u>virtuellen Encoderausgangs</u>	0E _h
41, DIO3			0F _h
42, DIO2	Richtungs- Eingang in <u>Takt-Richtungs-Modus</u>	Kanal B des <u>virtuellen Encoderausgangs</u>	10 _h
43, DIO5			11 _h
44, DIO4			12 _h

Pin	Alt. Funktion 1	Alt. Funktion 2	Subindex in 3272 _h
45, DIO7			13 _h
46, DIO6			14 _h
47, DIO9	<u>PWM-Ausgang 0</u>		15 _h
48, BRAKE	<u>Bremsenausgang</u> (Werkseinstellung)		16 _h
50, DIO10	<u>PWM-Ausgang 1</u>		17 _h
56, DIO12			18 _h

Im Objekt 3272_h legen Sie fest, wie jeder Pin verwendet werden soll, indem Sie den entsprechenden Wert in den entsprechenden Subindex schreiben.

Wert	Funktion
0	digitaler Eingang
128	digitaler Ausgang
256/257	<u>Input Range Selection</u>
384/385	<u>Analog Input Control</u>
512	analoger Eingang
640	Hallsensor-Eingang
768	Encodereingang (inkrementell)
896	Encodereingang (SSI)
1024	PWM-Ausgang / Bremsenausgang
1152	<u>Virtueller Encoderausgang</u>
1280	Takt-/Richtungseingang im <u>Takt-Richtungs-Modus</u>
1408	<u>Generic SPI</u>
1536/1537	<u>Capture Input</u>

Danach speichern Sie Ihre Konfiguration, indem Sie den Wert "65766173_h" in 1010_h:03_h schreiben (siehe Kapitel Objekte speichern) und starten Sie die Steuerung neu.

Beispiel

Pin 39 (DIO1 soll der Takteingang im Takt-Richtungs-Modus sein.

Pin 42 (DIO2) soll der Richtungseingang im Takt-Richtungs-Modus sein.

1. 2372_h:0E_h und 2372_h:10_h auf "1280" setzen.
2. 1010_h:03_h auf "65766173_h" setzen.

Input Range Selection

Sie können bis zu zwei der konfigurierbaren Pins mit dieser Funktion belegen. Diese Pins sind lediglich digitale Ausgänge, die über das entsprechende Bit in 3240_h:06_h kontrolliert werden (auf *High* gesetzt, wenn Bit=1):

Wert in 3272 _h :xx _h	Kontrollbit in 3240 _h :06 _h
384	0
385	1

Mit diesen Pins können Sie passende externe Schaltungen ansteuern, die den Schaltpegel der digitalen Eingänge umschalten, beispielsweise zwischen 5/24 V.

Analog Input Control

Sie können bis zu zwei der konfigurierbaren Pins mit dieser Funktion belegen. Diese Pins sind lediglich digitale Ausgänge, die über das entsprechende Bit in 3221_h:06_h kontrolliert werden (auf *High* gesetzt, wenn Bit=1):

Wert in 3272 _h :xx _h	Kontrollbit in 3221 _h :00 _h
256	0
257	1

Mit diesen Pins können Sie passende externe Schaltungen ansteuern, die den entsprechenden Analogeingang zwischen Spannungs- und Strommessung umschalten. Der erste Pin (in der Reihenfolge der Tabelle der alternativen Funktionen) kontrolliert den ersten Analogeingang und der zweite Pin den zweiten.

Capture Input

Sie können bis zu zwei der konfigurierbaren Pins mit dieser Funktion belegen. Diese Pins sind Capture-Eingänge, die über die entsprechenden Subindices in 3241_h konfiguriert werden:

Wert in 3272 _h :xx _h	Konfiguration in 3241 _h
1536	Capture Input 1: Subindex 1 bis 4
1537	Capture Input 2: Subindex 5 bis 8

Bei Pegelwechsel an diesen Pins wird die aktuelle Encoderposition notiert. Der erste mit der Funktion belegte Pin (in der Reihenfolge der Tabelle der alternativen Funktionen) ist der erste Capture-Eingang (Capture Input 1).

Generic SPI

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



HINWEIS

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

Allen konfigurierbaren Pins kann die Funktion *Chip Select* zugewiesen werden, mit Ausnahme der Pins, die für folgende Funktionen vorgesehen sind:

Pin	SPI-Funktion	Subindex in 3272 _h
21, DIO13	Data OUT (MOSI)	06 _h
25, SS11_MISO	Data IN (MISO)	0A _h
27, SS11_CLK	Clock	0C _h

Um die jeweilige Funktion zu aktivieren, schreiben Sie den Wert "1408" in 3272_h:xx_h.

HINWEIS



Nur der erste (in der Reihenfolge der Tabelle der alternativen Funktionen) als *Chip Select* konfigurierte Pin wird verwendet. Sollte Sie mehrere Pins mit der Funktion benötigen, verwenden Sie einen Pin als digitalen Ausgang.

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

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

Die Daten werden über folgende Objekte gesendet/empfangen:

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

8.1.2 Bitzuordnung

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

1. Das erste Bit entspricht der Spezialfunktion eines 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.

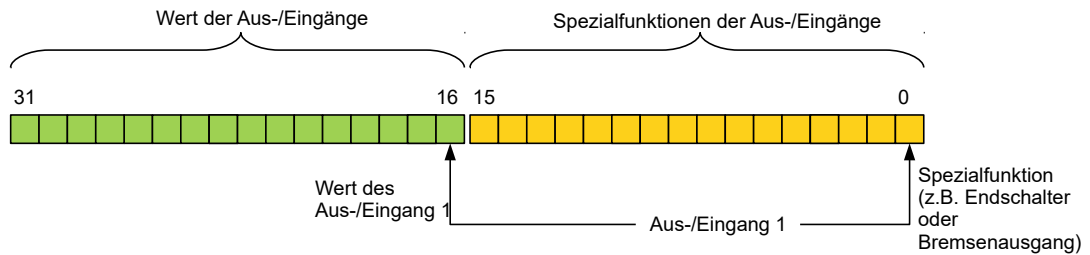
Beispiel

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



8.1.3 Digitale Eingänge

8.1.3.1 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 im Auslieferungszustand zur Verfügung:

	Pin	Name
	39	DIO1
	42	DIO2
	41	DIO3
	44	DIO4
	43	DIO5
	46	DIO6
	45	DIO7
	47	DIO9
	50	DIO10
	19	DIO11
	56	DIO12
	21	DIO13
	29	DIO14

8.1.3.2 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.

HINWEIS



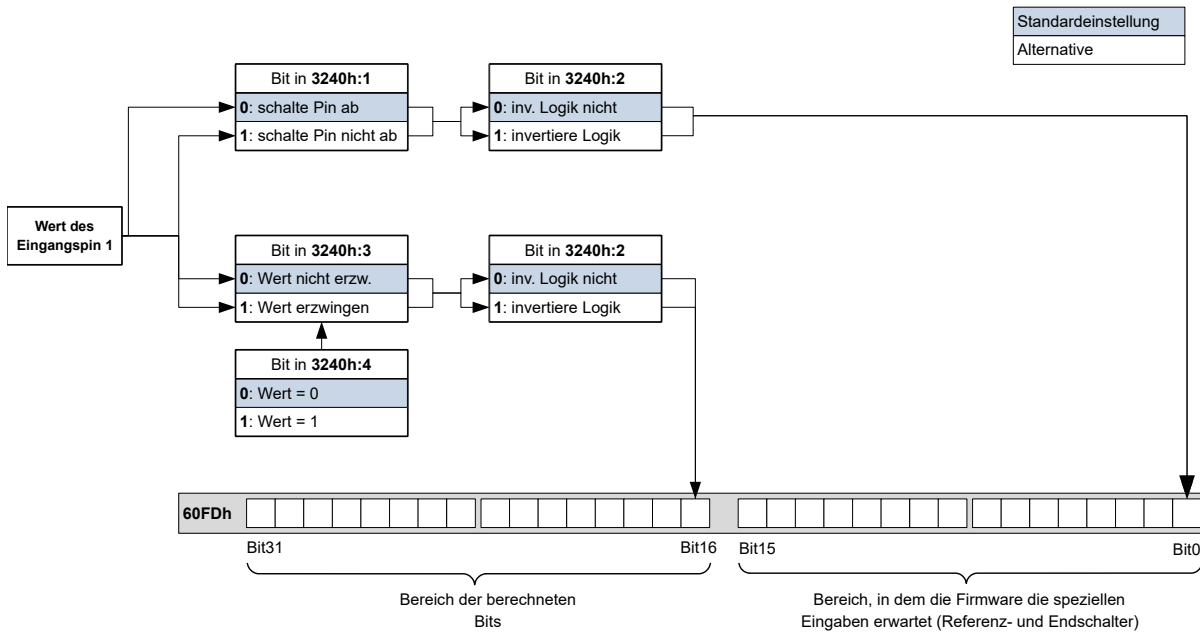
Da das *Input Routing* (siehe nachfolgendes Kapitel) im Auslieferungszustand aktiviert ist, hat das Objekt **3240_h:01_h** bei dieser Steuerung keine Funktion. Um die Sonderfunktionen zu nutzen, konfigurieren Sie in **3242_h:01_h** bis **:04_h** die Quelle für Bits 0 bis 3 von **60FD_h**, entsprechend Ihrer Verkabelung.

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

8.1.3.3 Verrechnung der Eingänge

Verrechnung des Eingangssignals am Beispiel von Eingang 1:

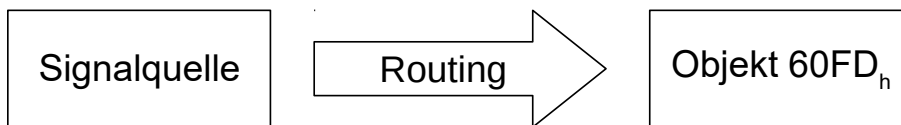
Der Wert an Bit 0 des Objekts **60FD_h** wird von der Firmware als negativer Endschalter interpretiert, das Ergebnis der vollständigen Verrechnung wird in Bit 16 abgelegt.



8.1.3.4 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 (im Auslieferungszustand der Fall).

HINWEIS

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

HINWEIS

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

Routing

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

HINWEIS


Wenn Sie das *Input Routing* deaktivieren, entsprechen die Bits 16 bis 31 den ersten 16 Pins in der Tabelle der alternativen Funktionen.

Nummer		Signalquelle
dec	hex	
00	00	Signal ist immer 0
01	01	ANA1 (Pin 4)
02	02	ANA2 (Pin 6)
03	03	H1 (Pin 18)
04	04	DIO11 (Pin 19)
05	05	H2 (Pin 20)
06	06	DIO13 (Pin 21)
07	07	H3 (Pin 22)
08	08	ENC1B (Pin 23)
09	09	ENC1A (Pin 24)
10	0A	SSI1_MISO (Pin 25)
11	0B	ENC1I (Pin 26)
12	0C	SSI1_CLK (Pin 27)
13	0D	DIO14 (Pin 29)
14	0E	DIO1 (Pin 39)
15	0F	DIO3 (Pin 41)
16	10	DIO2 (Pin 42)
17	11	DIO5 (Pin 43)
18	12	DIO4 (Pin 44)
19	13	DIO7 (Pin 45)
20	14	DIO6 (Pin 46)
21	15	DIO9 (Pin 47)
22	16	BRAKE (Pin 48)
23	17	DIO10 (Pin 50)
24	18	DIO12 (Pin 56)
65	41	Hall Eingang "U"
66	42	Hall Eingang "V"
67	43	Hall Eingang "W"
68	44	Encoder Eingang "A"
69	45	Encoder Eingang "B"
70	46	Encoder Eingang "Index"

Die nachfolgende Tabelle beschreibt die invertierten Signale der vorherigen Tabelle.

Nummer		Signalquelle
dec	hex	
128	80	Signal ist immer 1
129	81	Invertierter ANA1 (Pin 4)
130	82	Invertierter ANA2 (Pin 6)

Nummer		Signalquelle
dec	hex	
131	83	Invertierter H1 (Pin 18)
132	84	Invertierter DIO11 (Pin 19)
133	85	Invertierter H2 (Pin 20)
134	86	Invertierter DIO13 (Pin 21)
135	87	Invertierter H3 (Pin 22)
136	88	Invertierter ENC1B (Pin 23)
137	89	Invertierter ENC1A (Pin 24)
138	8A	Invertierter SSI1_MISO (Pin 25)
139	8B	Invertierter ENC1I (Pin 26)
140	8C	Invertierter SSI1_CLK (Pin 27)
141	8D	Invertierter DIO14 (Pin 29)
142	8E	Invertierter DIO1 (Pin 39)
143	8F	Invertierter DIO3 (Pin 41)
144	90	Invertierter DIO2 (Pin 42)
145	91	Invertierter DIO5 (Pin 43)
146	92	Invertierter DIO4 (Pin 44)
147	93	Invertierter DIO7 (Pin 45)
148	94	Invertierter DIO6 (Pin 46)
149	95	Invertierter DIO9 (Pin 47)
150	96	Invertierter BRAKE (Pin 48)
151	97	Invertierter DIO10 (Pin 50)
152	98	Invertierter DIO12 (Pin 56)
193	C1	Invertierter Hall Eingang "U"
194	C2	Invertierter Hall Eingang "V"
195	C3	Invertierter Hall Eingang "W"
196	C4	Invertierter Encoder Eingang "A"
197	C5	Invertierter Encoder Eingang "B"
198	C6	Invertierter Encoder Eingang "Index"

Beispiel

Eingang 1 soll auf Bit 0 des Objekts `60FDh` geroutet werden, um als negativer Endschalter benutzt zu werden.

1. Um das *Input Routing* zu aktivieren, setzen Sie das `3240h:08h` auf "1" (im Auslieferungszustand bereits gesetzt).
2. Um den Eingang 1 (DIO1) auf Bit 0 zu routen, setzen Sie das `3242h:01h` auf "14".

8.1.3.5 Interlock-Funktion

Bei der Interlock-Funktion handelt es sich um eine Freigabe, die Sie über das Bit 3 in `60FDh` 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 `605Eh` hinterlegte Aktion ausgeführt.

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

Beispiel

Eingang 4 soll auf Bit 3 des Objekts `60FDh` 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 `3240h:08h` auf "1" (im Auslieferungszustand bereits gesetzt).
2. Um den Eingang 4 (DIO4) auf Bit 3 zu routen, setzen Sie das `3242h:04h` auf "18".

8.1.4 Digitale Ausgänge

8.1.4.1 Ausgänge

Die Ausgänge werden über das Objekt `60FEh` gesteuert. Dabei entspricht Ausgang 1 dem Bit 16 im Objekt `60FEh`, 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.

Die gewünschten Pins müssen Sie vorher als Ausgang definiert haben, siehe [Ein- und Ausgangsbelegung festlegen](#).

8.1.4.2 Beschaltung

Die digitalen Ausgänge haben einen digitalen Pegel von 3,3 V DC. Die Strombelastbarkeit liegt bei ca. 10mA.

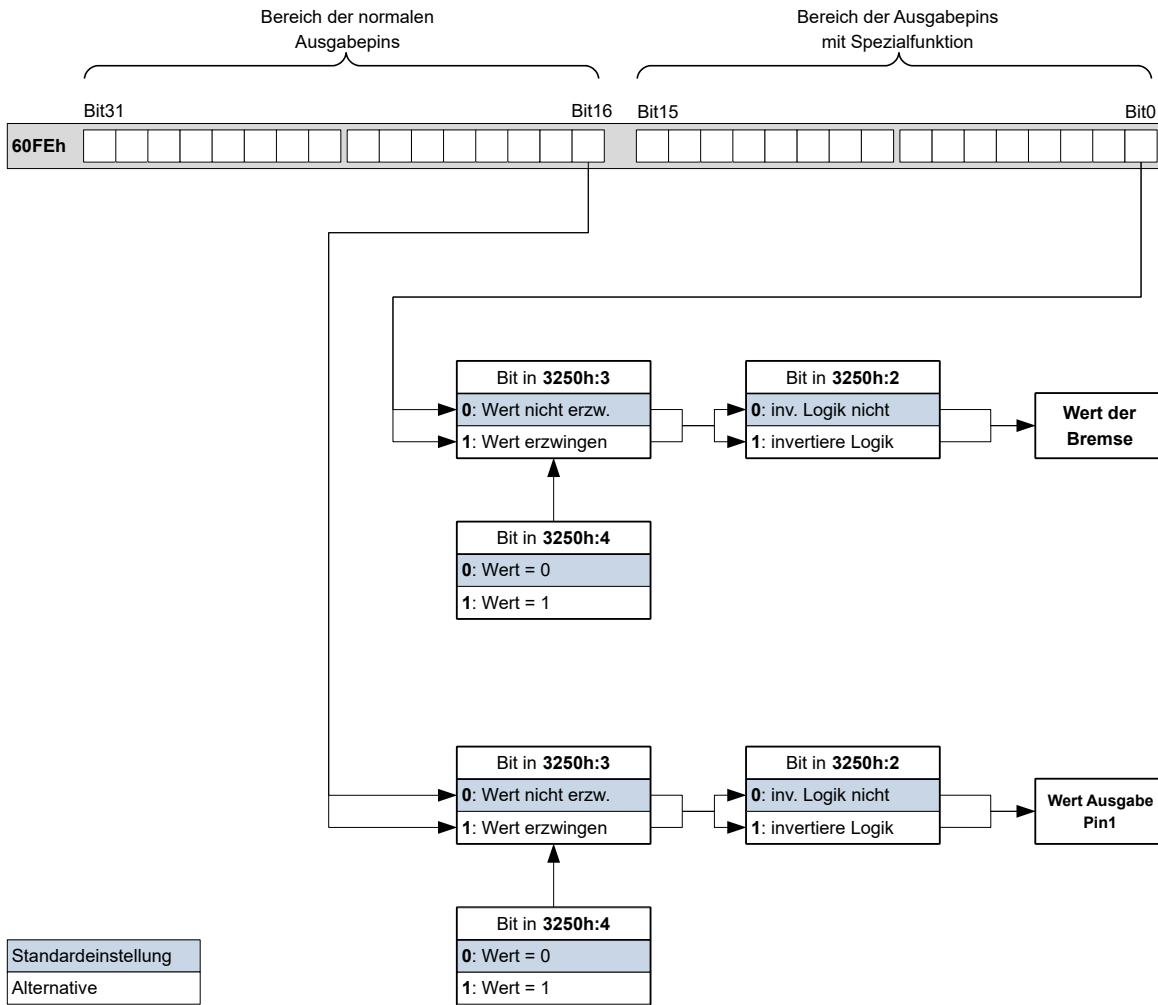
8.1.4.3 Objekteinträge

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:

- `3250h:01h`: Keine Funktion.
- `3250h:02h`: 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 `60FEh` entsprechend ein logischer Low-Pegel ausgegeben.
- `3250h:03h`: Ist hier ein Bit gesetzt, wird der Ausgang manuell gesteuert. Der Wert für den Ausgang steht dann in Objekt `3250h:4h`, dies ist auch für den Bremsenausgang möglich.
- `3250h:04h`: Die Bits in diesem Objekt geben den Ausgabewert vor, welcher am Ausgang angelegt sein soll, wenn die manuelle Steuerung des Ausgangs über das Objekt `3250h:03h` aktiviert ist.
- `3250h:05h`: In diesen Subindex wird die an die Ausgänge gelegte Bitkombination abgelegt.
- `3250h:08h`: Zum Aktivieren des *Output Routing*.
- `3250h:09h`: 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.

8.1.4.4 Verrechnung der Ausgänge

Beispiel für die Verrechnung der Bits für die Ausgänge:

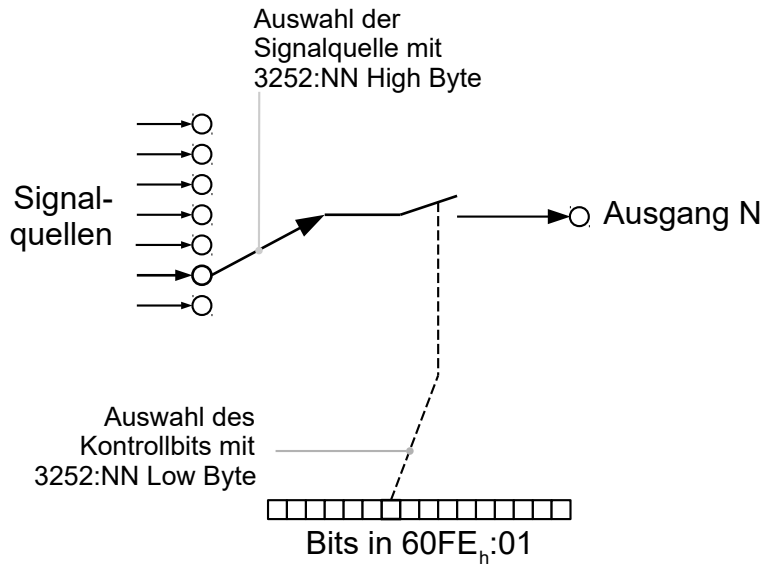


8.1.4.5 Output Routing

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 (im Auslieferungszustand der Fall).

HINWEIS



Die Einträge $3250_h:01_h$ bis $3250_h:04_h$ 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 Ausgangs 1 (Pin 4)
02 _h	Konfiguration des Ausgangs 2 (Pin 6)
03 _h	Konfiguration des Ausgangs 3 (Pin 18)
04 _h	Konfiguration des Ausgangs 4 (Pin 19)
05 _h	Konfiguration des Ausgangs 5 (Pin 20)
06 _h	Konfiguration des Ausgangs 6 (Pin 21)
07 _h	Konfiguration des Ausgangs 7 (Pin 22)
08 _h	Konfiguration des Ausgangs 8 (Pin 23)
09 _h	Konfiguration des Ausgangs 9 (Pin 24)
0A _h	Konfiguration des Ausgangs 10 (Pin 25)
0B _h	Konfiguration des Ausgangs 11 (Pin 26)
0C _h	Konfiguration des Ausgangs 12 (Pin 27)
0D _h	Konfiguration des Ausgangs 13 (Pin 29)
0E _h	Konfiguration des Ausgangs 14 (Pin 39)
0F _h	Konfiguration des Ausgangs 15 (Pin 41)
10 _h	Konfiguration des Ausgangs 16 (Pin 42)
11 _h	Konfiguration des Ausgangs 17 (Pin 43)

Subindex 3252 _h	Output Pin
12 _h	Konfiguration des Ausgangs 18 (Pin 44)
13 _h	Konfiguration des Ausgangs 19 (Pin 45)
14 _h	Konfiguration des Ausgangs 20 (Pin 46)
15 _h	Konfiguration des Ausgangs 21 (Pin 47)
16 _h	Konfiguration des Ausgangs 22 (Pin 48)
17 _h	Konfiguration des Ausgangs 23 (Pin 50)
18 _h	Konfiguration des Ausgangs 24 (Pin 56)

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 konfiguriert wird (Bremsenausgang)
11XX _h	invertiertes PWM-Signal, das mit Objekt 2038 _h konfiguriert wird (Bremsenausgang)
12XX _h	PWM-Signal, das mit Objekt 3260 _h konfiguriert wird
13XX _h	PWM-Signal, das mit Objekt 3261 _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 $60FE:01$

8.1.5 Virtueller Encoderausgang

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

Funktion der Pins aktivieren

Um die Funktion zu aktivieren, setzen Sie $2372_h:0E_h$ und $2372_h:10_h$ auf "1152".

Siehe auch Ein- und Ausgangsbelegung festlegen).

Typ der Ausgangssignale wählen

Sie können im Objekt $205C_h:02_h$ einen der folgenden Typen wählen :

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

Quelle der Positionsdaten wählen

Über den Encoderausgang werden die Positionsdaten einer der vorhandenen Rückführungen wiedergeben.

Um die Quelle zu wählen, setzen Sie das Bit 3 im entsprechenden Subindex des Objekts 3203_h auf "1". Wenn Sie kein Bit setzen, wird der Wert aus 205C:01_h verwendet.

Auflösung der Ausgangssignale einstellen

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

8.2 Automatische Bremsensteuerung

8.2.1 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 Motorstopverhalten, lesen Sie auch das Kapitel Power State machine - Bremsreaktionen.

8.2.2 Aktivierung und Anschluss

Die Bremse kann entweder automatisch oder manuell gesteuert werden:

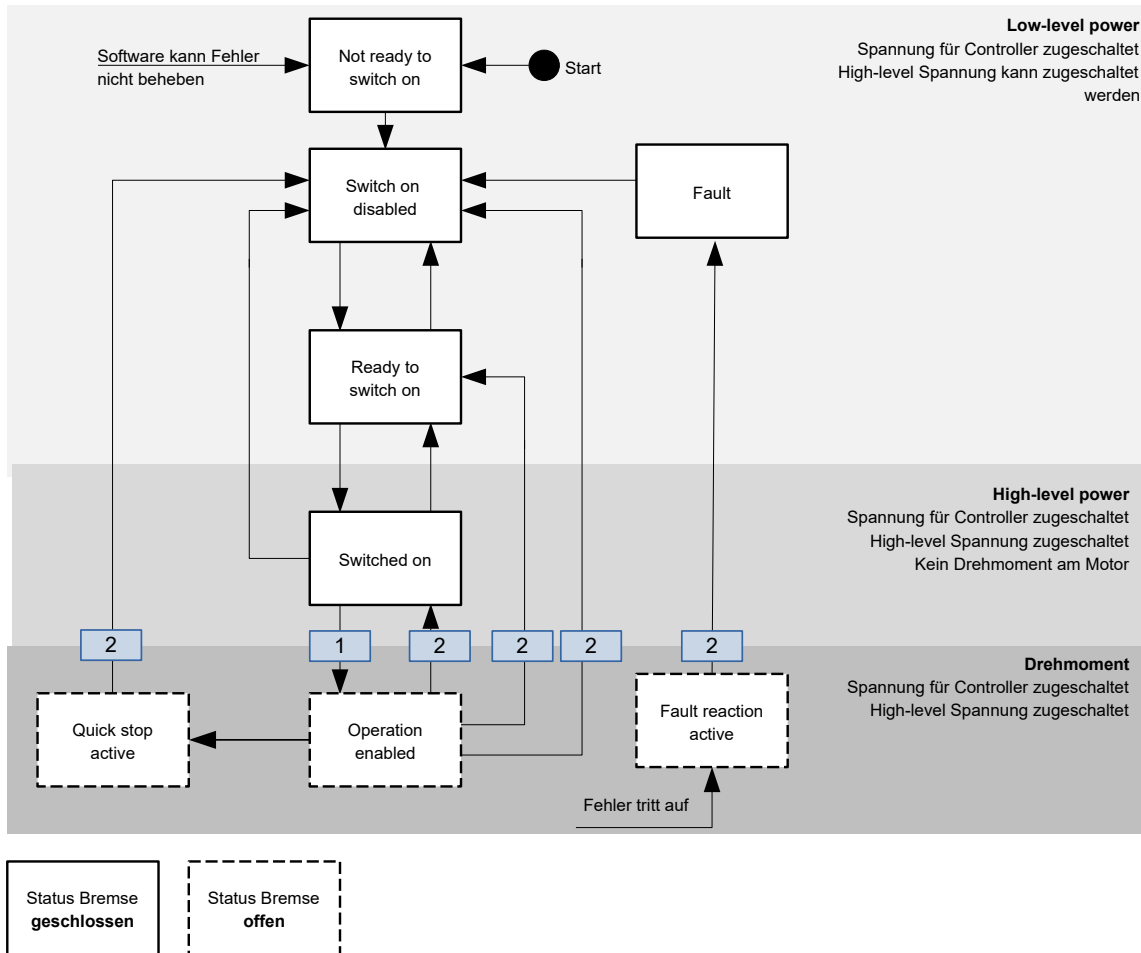
- Automatisch: Bit 2 des Objekts 3202_h auf "1" setzen aktiviert die Bremsensteuerung.
- Manuell: Bit 2 des Objekts 3202_h auf "0" setzen deaktiviert die Bremsensteuerung, die Bremse lässt sich jetzt mit dem Bit 0 im Objekt 60FE_h:01_h kontrollieren.

8.2.2.1 Anschluss

Der Bremsenausgang befindet sich am Pin 48 (siehe Anschlussbelegung). Um den Bremsenausgang zu benutzen, müssen Sie den Pin entsprechend konfigurieren (im Auslieferungszustand der Fall), siehe Digitale Ein- und Ausgänge).

8.2.3 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.

8.2.4 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.

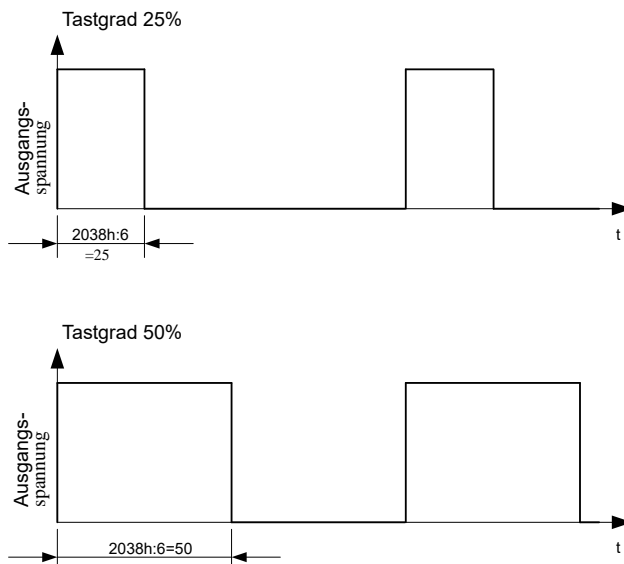
8.2.4.1 Frequenz

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.

8.2.4.2 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.



8.3 Externe Ballast-Schaltung

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

Um eine Beschädigung der Steuerung durch Überspannung zu verhindern, kann es je nach Höhe der Bremsleistung erforderlich sein, überschüssige Energie in Form von Wärme abzuführen. Die Steuerung stellt hierzu an Pin (Pin 52) einen Ausgang zur Ansteuerung einer externen Ballast-Schaltung bereit, bestehend aus einem Treiber, einem MOSFET als Schalter und einem ausreichend dimensionierten Ballast-Widerstand.

8.3.1 Steuerung des Ballast-Widerstands

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

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

In den folgenden Kapiteln werden die zu konfigurierenden Parameter beschrieben.

8.3.2 Ballast aktivieren

Um den Ballast zu aktivieren, setzen Sie das Bit 0 in $4021_h:01_h$ auf "1". Wenn Sie die Polarität des Pins zur Ansteuerung der externen Ballast-Schaltung (Pin 52, im Auslieferungszustand *active high*) invertieren möchten, setzen Sie das Bit 1 in $4021_h:01_h$ auf "1".

Die Ansprechschwelle in Millivolt, sowie die Hysterese beim Ein-/Ausschalten, tragen Sie in $4021_h:02_h$ bzw. $4021_h:03_h$ ein.

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

8.3.3 Ballast-Überwachung

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

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

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

Nominal Resistance $R_{Ballast}$ [mOhm]

Nennwert des Ballast-Widerstands

Cooling Power $P_{Stat_TA_Max}$ [mW]

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

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

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

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

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

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

Der Wert wird bei Lastwiderständen typischerweise im Datenblatt angegeben.

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

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

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

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

Long Term Reference Time t_{LT_Ref} [ms]

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

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

8.4 I²t Motor-Überlastungsschutz

8.4.1 Beschreibung



HINWEIS

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

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

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

8.4.2 Objekteinträge

Folgende Objekte haben Einfluss auf den I²t Motor-Überlastungsschutz:

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

Folgende Objekte zeigen den gegenwärtigen Zustand von I²t an:

- 203B_h:3_h Threshold - Gibt die Grenze in A²ms an, von der abhängt, ob auf Maximalstrom oder Nennstrom geschaltet wird.
- 203B_h:4_h CalcValue - Gibt den berechneten Wert in A²ms an, welcher mit Threshold verglichen wird, um den Strom einzustellen.
- 203B_h:5_h LimitedCurrent - Zeigt den gegenwärtigen Stromwert in mA an, der von I²t eingestellt wurde.
- 203B_h:6_h Status:
 - Wert = "0": I²t deaktiviert
 - Wert = "1": I²t aktiviert

8.4.3 Aktivierung

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

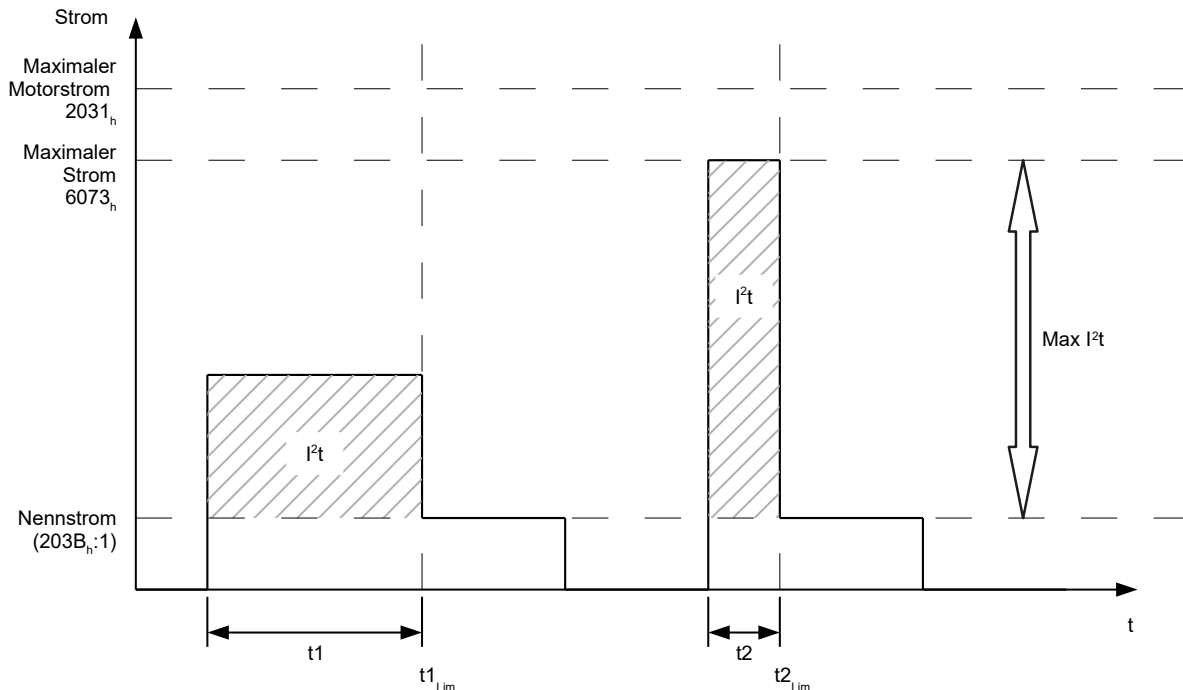
Zum Aktivieren des Modus müssen Sie die vier oben genannten Objekteinträge (2031_h, 6073_h, 203B_h:1_h, 203B_h:2_h) sinnvoll beschreiben. Das bedeutet, dass der Maximalstrom größer als der Nennstrom sein muss und ein Zeitwert für die maximale Dauer des Maximalstroms eingetragen sein muss. Wenn diese Bedingungen nicht erfüllt sind, bleibt die I²t Funktionalität deaktiviert.

8.4.4 Funktion von I²t

Durch die Angabe von Nennstrom, Maximalstrom und maximaler Dauer des Maximalstromes wird ein I²T_{Lim} berechnet.

Der Motor kann solange mit Maximalstrom laufen, bis das berechnete I²T_{Lim} erreicht wird. Darauf folgend wird der Strom sofort auf Nennstrom gesenkt. Der Maximalstrom wird durch den maximalen Motorstrom (2031_h) begrenzt.

In den folgenden Diagrammen sind die Zusammenhänge noch einmal dargestellt.



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

8.5 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 *i* (siehe [Konfiguration über USB](#)) 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.

8.5.1 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.
- Modbus RTU: Parameter mit Bezug auf die Modbus RTU-Kommunikation

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

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

8.5.2 Kategorie: Kommunikation

- 2102_h: Fieldbus Module Control
- 2104_h: Additional Fieldbus Configuration
- 3502_h: MODBUS Rx PDO Mapping
- 3602_h: MODBUS Tx PDO Mapping

8.5.3 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
- 2057_h: Clock Direction Multiplier
- 2058_h: Clock Direction Divider
- 205B_h: Clock Direction Or Clockwise/Counter Clockwise Mode
- 205C_h: Virtual Encoder Configuration
- 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
- 3241_h: Digital Input Capture
- 3242_h: Digital Input Routing
- 3243_h: Digital Input Homing Capture
- 3250_h: Digital Outputs Control
- 3252_h: Digital Output Routing
- 3260_h: Pwm Output 0
- 3261_h: Pwm Output 1
- 3273_h: Generic SPI Hardware Configuration

- 3274_h: Generic SPI Mosi Data
- 3321_h: Analog Input Offsets
- 3322_h: Analog Input Numerators
- 3323_h: Analog Input Denominators
- 3700_h: Deviation Error Option Code
- 3701_h: Limit Switch Error Option Code
- 4013_h: HW Configuration
- 6040_h: Controlword
- 6042_h: VI Target Velocity
- 6046_h: VI Velocity Min Max Amount
- 6048_h: VI Velocity Acceleration
- 6049_h: VI Velocity Deceleration
- 604A_h: VI Velocity Quick Stop
- 604C_h: VI Dimension Factor
- 605A_h: Quick Stop Option Code
- 605B_h: Shutdown Option Code
- 605C_h: Disable Option Code
- 605D_h: Halt Option Code
- 605E_h: Fault Option Code
- 6060_h: Modes Of Operation
- 6065_h: Following Error Window
- 6066_h: Following Error Time Out
- 6067_h: Position Window
- 6068_h: Position Window Time
- 606D_h: Velocity Window
- 606E_h: Velocity Window Time
- 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

8.5.4 Kategorie: Benutzer

- 2701_h: Customer Storage Area

8.5.5 Kategorie: Bewegung

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

8.5.6 Kategorie: Tuning

- 2030_h: Pole Pair Count
- 2031_h: Max Motor Current
- 203B_h: I2t Parameters
- 3203_h: Feedback Selection
- 3380_h: Feedback Sensorless
- 3390_h: Feedback Hall
- 33A0_h: Feedback Incremental A/B/I 1
- 33B0_h: Feedback SSI 1
- 33B1_h: Feedback SSI 2
- 4021_h: Ballast Configuration
- 6075_h: Motor Rated Current
- 608E_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

8.5.7 Kategorie: Modbus RTU

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

8.5.8 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 0B _h (Modbus RTU)
02 _h	Kommunikation
03 _h	Applikation
04 _h	Benutzer
05 _h	Bewegung
06 _h	Tuning
0B _h	Modbus RTU

8.5.9 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) und 0B _h (Modbus RTU)
02 _h	Kommunikation
03 _h	Applikation
04 _h	Benutzer
05 _h	Bewegung

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

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

Subindex	Kategorie
06 _h	Tuning
0B _h	Modbus RTU

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

HINWEIS



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

8.5.10 Konfiguration verifizieren

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

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

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

Die folgende Reihenfolge macht die Verifikation möglich:

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

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

9 Modbus RTU

Modbus-Referenzen: www.modbus.org.

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

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

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

9.1 Modbus Modicon-Notation bei SPS

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

Folgende Adress-Notation ist bei Nanotec Steuerungen relevant:

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

HINWEIS



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

9.2 Allgemeines

Modbus ist generell Big-Endian basiert.

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

Beispiel

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

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

SA
Slave-Adresse

FC
Funktionscode

Daten

Datenbereich, Decodierung ist abhängig vom benutzen Funktionscode

CRC

Cyclic redundancy check

HINWEIS

Wenn sich mehr als einer Modbus-Slave im Netzwerk befinden, muss der Modbus-Master mindestens 3 ms nach dem Empfang einer Antwort warten, bevor er die nächste Nachricht schickt.

TIPP

Um eine Broadcast-Nachricht an alle Teilnehmer zu senden, nutzen Sie die Slave-Adresse "0". Die Steuerung antwortet in dem Fall nicht.

9.3 Funktionscodes

Die folgenden "Funktionscodes" werden unterstützt:

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

9.4 Funktioncode-Beschreibungen

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

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

Request		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	03 _h / 04 _h
Startadresse	2 Bytes	0000 _h bis FFFF _h
Anzahl der Register	2 Bytes	1 bis (7D _h)
CRC	2 Bytes	

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

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

Beispiel

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

Request

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

Response

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

9.4.2 FC 6 (06_h) Write Single Register

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

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

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

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

Beispiel

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

Request

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

Response

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

9.4.3 FC 16 (10_h) Write Multiple Registers

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

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

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

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

Beispiel

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

Request

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

Response

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

9.4.4 FC 17 (11_h) Report Server ID

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

Request		
Name	Länge	Wert
Funktionscode	1 Byte	11 _h

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

Fehler		
Name	Länge	Wert
Fehlercode	1 Byte	91 _h
Ausnahmecode (siehe Ausnahmecodes)	1 Byte	01 oder 04

Beispiel

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

Request

SA	FC	CRC
05	11	C2 EC

Response

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

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

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

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

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

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

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

Beispiel

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

Request

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

Response

SA	FC	Daten	CRC
05	17	04 02 40 00 00	0F EC

9.4.6 FC 8 (08_h) Diagnostics

Der Modbus-Funktionscode FC08 bietet eine Menge an Tests zum Überprüfen des Kommunikationssystems zwischen Client und Server oder zum Überprüfen verschiedener interner Fehlerzustände innerhalb des Servers.

Diese Funktion verwendet einen zwei Byte großen Unterfunktionscode im Request, um den Typen des Tests zu definieren. Der Server wiederholt in einer normalen Response beides, den Funktions- und den Unterfunktionscode. Einige der Diagnosen enthalten Daten des Gerätes im Datenfeld der normalen Antwort.

Request:

Name	Länge	Wert
Funktionscode	1 Byte	08 _h
Unterfunktionscode	2 Bytes	
Data	N x 2 Bytes	

Response:

Name	Länge	Wert
Funktionscode	1 Byte	08 _h
Unterfunktionscode	2 Bytes	
Data	N x 2 Bytes	

Fehler:

Name	Länge	Wert
Funktionscode	1 Byte	88 _h
Ausnahmecode (siehe Ausnahmecodes)	1 Bytes	01 oder 03 oder 04

9.4.6.1 FC 8.10 (08_h.0A_h) Clear Counters and Diagnostic Register

Das Ziel dieser Anfrage ist, alle Zähler und Diagnose-Register zurückzusetzen. Zähler werden auch beim Einschalten der Steuerung zurückgesetzt.

Unterfunktion	Datenbereich
Request	Resonse
00 _h 0A _h	00 _h 00 _h Echo der Anfragedaten

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 0A 00 00	56 6A

Response

SA	FC	Daten	CRC
05	08	00 0A 00 00	C1 8D

9.4.6.2 FC 8.11 (08_h.0B_h) Return Bus Message Count

Der Datenbereich der Antwort gibt die Anzahl der Nachrichten zurück, welche seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung am Kommunikationssystem erkannt worden sind.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0B _h	00 _h 00 _h	Total Message Count

9.4.6.3 FC 8.12 (08_h.0C_h) Return Bus Communication Error Count

Der Datenbereich der Antwort gibt die Anzahl der CRC Fehler seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung zurück.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0C _h	00 _h 00 _h	CRC Error Count

Beispiel

Request

SA	FC	Daten	CRC
05	08	00 0C 00 00	21 8C

Response

SA	FC	Daten	CRC
05	08	00 0C 00 00	21 8C

9.4.6.4 FC 8.13 (08_h.0D_h) Return Bus Exception Error Count

Der Datenbereich der Antwort gibt die Anzahl der Modbus Ausnahmen seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung zurück.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0D _h	00 _h 00 _h	Exception Error Count

Beispiel**Request**

SA	FC	Daten	CRC
05	08	00 0D 00 00	70 4C

Response

SA	FC	Daten	CRC
05	08	00 0D 00 00	70 4C

9.4.6.5 FC 8.14 (08_h.0E_h) Return Server Message Count

Der Datenbereich der Antwort gibt die Anzahl der an das Gerät gerichteten und Broadcast-Nachrichten zurück, die von der Steuerung verarbeitet wurden. Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0E _h	00 _h 00 _h	Server Message Count

Beispiel**Request**

SA	FC	Daten	CRC
05	08	00 0E 00 00	80 4C

Response

SA	FC	Daten	CRC
05	08	00 0E 00 00	80 4C

9.4.6.6 FC 8.15 (08_h.0F_h) Return Server No Response Count

Der Datenbereich der Antwort gibt die Anzahl der an die Steuerung gerichteten Nachrichten zurück, für die keine Antwort zurückgesendet wurde (weder normale Antwort noch Ausnahme-Antwort). Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 0F _h	00 _h 00 _h	No Response Count

Beispiel**Request**

SA	FC	Daten	CRC
05	08	00 0F 00 00	D1 8C

Response

SA	FC	Daten	CRC
05	08	00 0F 00 00	D1 8C

9.4.6.7 FC 8.16 (08_h.10_h) Return Server NAK Count

Der Datenbereich der Antwort gibt die Anzahl der Nachrichten zurück, für die eine "Negative Acknowledge (NAK)"-Ausnahme-Antwort zurückgesendet wurde. Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 10 _h	00 _h 00 _h	Server NAK Count

Beispiel**Request**

SA	FC	Daten	CRC
05	08	00 10 00 00	E0 4A

Response

SA	FC	Daten	CRC
05	08	00 10 00 00	E0 4A

9.4.6.8 FC 8.17 (08_h.11_h) Return Server Busy Count

Der Datenbereich der Antwort gibt die Anzahl der Nachrichten zurück, für die eine "Server Device Busy"-Ausnahme-Antwort zurückgesendet wurde. Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung.

Unterfunktion	Datenbereich	
	Request	Response
00 _h 11 _h	00 _h 00 _h	Server NAK Count

Beispiel**Request**

SA	FC	Daten	CRC
05	08	00 11 00 00	B1 8A

Response

SA	FC	Daten	CRC
05	08	00 11 00 00	B1 8A

9.4.6.9 FC 8.18 (08_h.12_h) Return Bus Character Overrun Count

Der Datenbereich der Antwort gibt die Anzahl der an die Steuerung gerichteten Nachrichten zurück, die aufgrund einem Zeichenüberlauf nicht verarbeitet werden konnten. Gezählt werden die Nachrichten seit dem letzten Neustart, "Clear Counters and Diagnostic Register"-Request oder Einschalten der Steuerung. Ein Zeichenüberlauf entsteht dadurch, dass Zeichen schneller an der Steuerung ankommen, als sie gespeichert werden können, oder durch den Verlust eines Zeichens aufgrund eines Hardwarefehlers.

Unterfunktion	Datenbereich
Request	
Response	
00 _h 12 _h	00 _h 00 _h Server Character Overrun Count

Beispiel**Request**

SA	FC	Daten	CRC
05	08	00 12 00 00	41 8A

Response

SA	FC	Daten	CRC
05	08	00 12 00 00	41 8A

9.4.7 FC 43 (2B_h) Encapsulated Interface Transport

Diese Funktion ermöglicht einen einfachen Zugriff auf das CANopen-Objektverzeichnis. Weitere Details können in den folgenden Dokumentationen entnommen werden:

1. *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3*, Date: 26.04.2014, Version: 1.1b3
2. *CiA 309 Draft Standard Proposal - Access from other networks - Part 2: Modbus/TCP mapping V1.3*, Date: 30.07.2015, Version: 1.3

HINWEIS

Für die Nachrichten des Encapsulated Interface-Transport gilt zum Teil eine andere Byte-Reihenfolge, siehe Kapitel *Allgemeines*.

Definition des Request und Response:

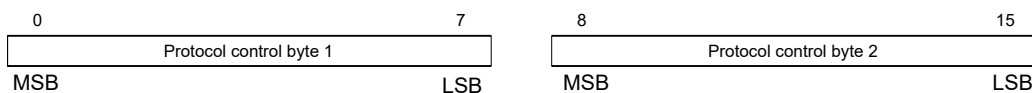
Name	Länge	Beispiel/Zahlenbereich
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h (43 _d)
MEI type	1 Byte	0D _h (13 _d)
Protokolloptionen Bereich	2 bis 5 Byte	
Adressen- und Datenbereich	N Bytes	
CRC	2 Bytes	

Protokolloptionen Bereich

Name	Länge	Beispiel/Zahlenbereich
Protokoll-Kontrolle	1 bis 2 Bytes	Siehe Beschreibung
Reserviert	1 Byte	Immer 0
(Optional) Zählerbyte	1 Byte	
(Optional) Netzwerk ID	1 Byte	
(Optional) Encodierte Daten	1 Byte	

Protokoll-Kontrolle:

Das Feld "Protokoll-Kontrolle" enthält die Merker, welche für die Kontrolle der Nachrichtenprotokolle benötigt werden. Die Bytes des Feldes "Protokoll Kontrolle" sind folgendermaßen definiert, falls der Merker "Verlängerung" gesetzt wurde (andernfalls entfällt das zweite Byte):



Das höchstwertige Bit (MSB) ist Bit 0 für "Protokoll-Kontrolle" Byte 1, und Bit 8 für "Protokoll Kontrolle" Byte 2. Das niedrigstwertige Bit (LSB) ist Bit 7 für "Protokoll Kontrolle" Byte 1, und Bit 15 für "Protokoll Kontrolle" Byte 2.

Bit	Name	Beschreibung
0	Merker "Verlängerung"	Dieses Bit wird genutzt, wenn das Objektverzeichnis Datenset größer ist, als in ein Modbus-Kommando passen würde. Das Datenset wird dann über mehrere Modbus-Nachrichten gestreckt, jede Nachricht enthält einen Teil des Datensets. "0" = Keine mehrfache Nachrichtentransaktion ("multiple message transaction") oder das Ende der mehrfachen Nachrichtentransaktion. "1" = Teil einer mehrfachen Nachrichtentransaktion.
1	Erweiterte Protokoll-Kontrolle	Länge der Protokoll-Kontrolle, der Wert "0" zeigt eine Länge von 1 Byte an, der Wert "1" eine Länge von 2 Byte.
2	Zähler Byte Option	Dieses Bit ist auf "1" um anzuzeigen, dass das Feld "counter byte" in dieser Nachricht genutzt wird. Ist dieses Bit auf "0" gesetzt, existiert das Feld "counter byte" nicht in dieser Nachricht.
3 und 4	Reserviert	0
5	Network ID Option	Nicht unterstützt, muss "0" sein.

Bit	Name	Beschreibung
6	Encodierte Datenoption	Nicht unterstützt, muss "0" sein.
7	Zugriffsmerker	Dieses Bit zeigt die Zugriffsmethode des angeforderten Kommandos an. "0" = lesen, "1" = schreiben.
8 to 15	Reserviert	0

Adressen- und Datenbereich

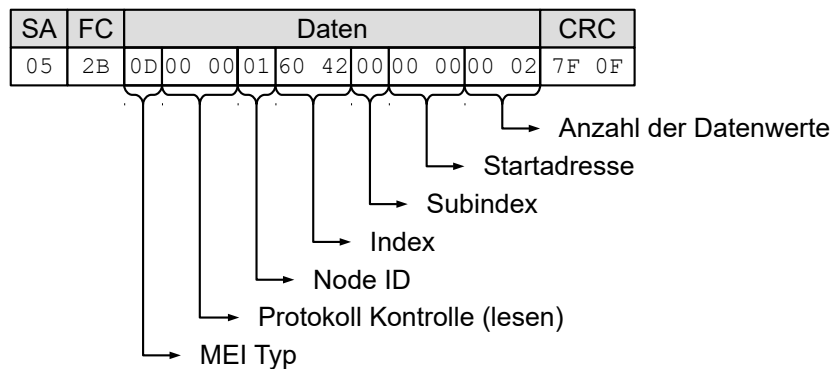
Der Adressen- und Datenbereich ist in der folgenden Tabelle definiert:

Name	Bytgröße und Bytereihenfolge	Beispiel / Bereich
Node-ID	1 Byte	01 _h bis 7F _h
Index	1 Byte, high	0000 _h bis FFFF _h
	1 Byte, low	
Subindex	1 Byte	00 _h bis FF _h
Startadresse	1 Byte, high	0000 _h bis FFFF _h
	1 Byte, low	
Anzahl der Datenwerte	1 Byte, high	0000 _h bis 00FD _h
	1 Byte, low	
Schreib-/Lesedaten	n Byte	Die Daten sind codiert wie in Kapitel <u>Allgemeines</u> beschrieben.

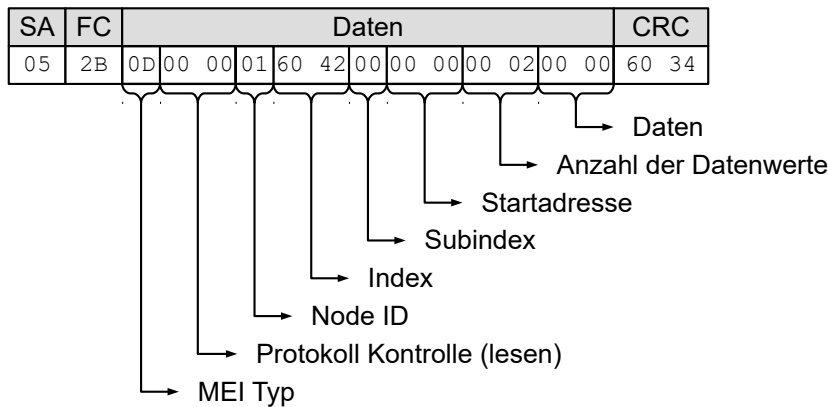
Beispiel:

Um das Objekt 6042_h:00_h auszulesen (16 Bit-Wert), muss folgende Nachricht vom Master verschickt werden (alle Werte sind in hexadezimaler Notation, die Slave-Id der Steuerung ist "5").

Request



Response



Als zusätzliches Beispiel nachfolgend eine Sequenz an Modbus-Nachrichten vom Master zum Slave, um den Motor im "Velocity" Modus sich drehen zu lassen:

Setze 6060 = "02_h" (velocity mode)

Request

SA	FC	Daten										CRC			
05	2B	0D	01	00	01	60	60	00	00	00	00	01	02	C9	2F

Response

SA	FC	Daten										CRC			
05	2B	0D	01	00	01	60	60	00	00	00	00	00	00	A9	89

Setze 2031 = 03E8_h" (1000 mA)

Request

SA	FC	Daten										CRC						
05	2B	0D	01	00	01	20	31	00	00	00	00	04	E8	03	00	00	C3	53

Response

SA	FC	Daten										CRC			
05	2B	0D	01	00	01	20	31	00	00	00	00	00	00	E5	CC

Setze 6040 = "00_h"

Request

SA	FC	Daten										CRC				
05	2B	0D	01	00	01	60	40	00	00	00	00	02	00	00	1C	2E

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00	AE E9

Setze 6040 = "80_h"**Request**

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 80 00	7D EE

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00	AE E9

Setze 6040 = "06_h"**Request**

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 06 00	1F 8E

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00	AE E9

Setze 6040 = "07_h"**Request**

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 07 00	1E 1E

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00	AE E9

Setze 6040 = "0F_h"**Request**

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 02 0F 00	19 DE

Response

SA	FC	Daten	CRC
05	2B	0D 01 00 01 60 40 00 00 00 00 00	AE E9

Nachfolgend zwei Beispiele zum Lesen eines Objektes:

Lese 6041_h:00_h

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 41 00 00 00 00 02	7F 3C

Response

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 41 00 00 00 00 02 37 06	B6 13

Lese 6061_h:00_h

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 61 00 00 00 00 01	38 5D

Response

SI	FC	Daten	CRC
05	2B	0D 00 00 01 60 61 00 00 00 00 01 00	5C D2

9.4.7.1 Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h + 80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception Länge	2 Bytes	6
MEI type	1 Byte	0D _h
Exception code	1 Byte	CE _h
Fehlercode	4 Bytes	CANopen-Fehlercode, siehe nachfolgende Tabelle
CRC	2 Bytes	

CANopen-Fehlercode	Beschreibung
FFFF0000 _h	Abort no error

CANopen-Fehlercode	Beschreibung
FFFF1003 _h	Service wird nicht unterstützt
FFFF1004 _h	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>
FFFF0003 _h	Unbekanntes oder nicht valides Kommando
FFFF0008 _h	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E _h	Allgemeiner Fehler im Parameter
FFFF0011 _h	Länge des Parameters falsch
FFFF0012 _h	Länge des Parameters zu groß
FFFF0013 _h	Länge des Parameters zu klein
FFFF0015 _h	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 _h	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 _h	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 _h	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 _h	Allgemeiner Fehler
FFFF001E _h	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 _h	Ungültige Sequenz der Nachrichten (z. B. wenn der Wert des <i>Counter-Bytes</i> entsprechend dem vorherigen Request oder Response nicht stimmt)

Im Falle, dass das nicht unterstützte Kontrolloptions-Bit gesetzt ist, wird folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h + 80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D _h
Exception code	1 Byte	AE _h
Supported protocol control	1 oder 2 Bytes	Siehe nachfolgende Tabelle
CRC	2 Bytes	

Bit	Name	Beschreibung
0	Merker "Verlängerung"	Dieses Bit wird genutzt, wenn das Objektverzeichnis Datenset größer ist, als in ein Modbus-Kommando passen würde. Das Datenset wird dann über mehrere Modbus-Nachrichten gestreckt, jede Nachricht enthält einen Teil des Datensets. "0" = Keine mehrfache Nachrichtentransaktion ("multiple message transaction") oder das Ende der mehrfachen Nachrichtentransaktion. "1" = Teil einer mehrfachen Nachrichtentransaktion.
1	Erweiterte Protokoll-Kontrolle	Länge der Protokoll-Kontrolle, der Wert "0" zeigt eine Länge von 1 Byte an, der Wert "1" eine Länge von 2 Byte.

Bit	Name	Beschreibung
2	Zähler Byte Option	Dieses Bit ist auf "1" um anzuzeigen, dass das Feld "counter byte" in dieser Nachricht genutzt wird. Ist dieses Bit auf "0" gesetzt, existiert das Feld "counter byte" nicht in dieser Nachricht.
3 und 4	Reserviert	0
5	Network ID Option	Nicht unterstützt, muss "0" sein.
6	Encodierte Datenoption	Nicht unterstützt, muss "0" sein.
7	Zugriffsmerker	Dieses Bit zeigt die Zugriffsmethode des angeforderten Kommandos an. "0" = lesen, "1" = schreiben.
8 to 15	Reserviert	0

Das nachfolgende Beispiel zeigt einen Fehler im Falle eines fehlerhaften Request. Der Request liest das 6061_h:00 mit der Länge von 2 Byte, das Objekt ist aber nur 1 Byte groß:

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 60 00 00 00 00 02	79 8D

Response

SA	FC	Daten	CRC
05	2B	FF 00 06 0D CE 12 00 07 06	AC 3C

9.4.8 FC 101 (65_h) Read complete object dictionary

Dieser Funktionscode wird zum Auslesen des gesamten Objektverzeichnis verwendet.

Um das Auslesen des Objektverzeichnis zu starten oder neu zu starten, muss der Unterfunktionscode 55_h versendet werden. Dieser Code setzt das Auslesen des Objektverzeichnis auf das Objekt 0000_h zurück. Alle nachfolgenden Objektverzeichnis-Frames müssen dann den Unterfunktionscode AA_h enthalten. Zum Ende, wenn alle Objekte ausgelesen wurden, wird eine "Error Response" generiert mit dem Abort-Code "No data available".

Das Format jedes "Objekt lesen" ist folgendermaßen:

Request:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	65 _h
Unterfunktionscode	1 Byte	55 _h oder AA _h
Länge der Daten	1 Byte	00 _h
CRC	2 Bytes	

Response:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	65 _h
Funktionscode	1 Byte	

Name	Länge	Wert / Bemerkung
Unterfunktionscode	1 Byte	
Länge der Daten	1 Byte	
n mal "Objektverzeichnis-Frame"	1 - 252 Bytes	
CRC	2 Bytes	

Ein Objektverzeichnis-Frame besteht aus den folgenden Bytes:

Name	Wert / Bemerkung
Index Low Byte	1 Byte
Index High Byte	1 Byte
Subindex	1 Byte
Anzahl der Bytes	1 Byte Anzahl m der validen Daten im Datenfeld
Daten Byte	m-1 Byte

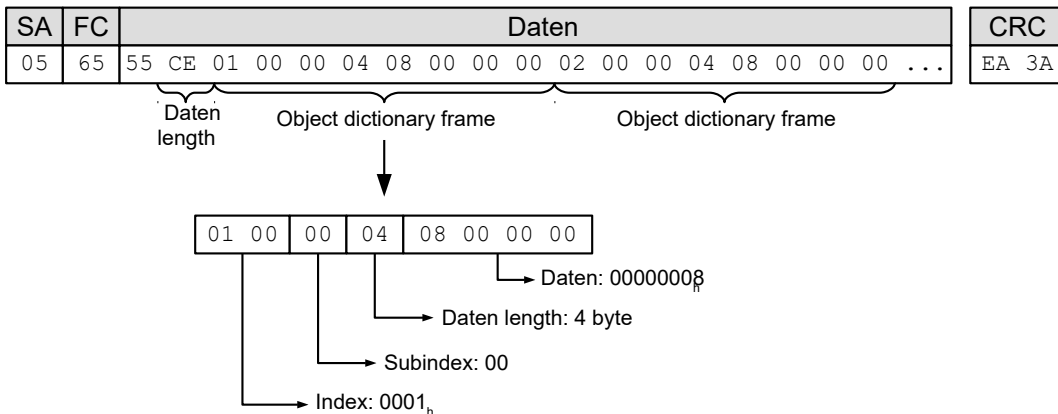
Beispiel

Alle folgenden Zahlenwerte sind in Hexadezimal notiert. Die Adresse des Slaves ist "5".

Start des Auslesens des Objektverzeichnisses mit dem Request:

SA	FC	Daten	CRC
05	65	55 00	2F A7

Die Response ist:



Den nächsten Teil des Objektverzeichnisses auslesen mit dem Request:

SA	FC	Daten	CRC
05	65	AA 00	6E 57

Die Response ist:

SA	FC	Daten	CRC
05	65	AA CD 21 00 0A 02 07 00 21 00 0B 02 07 00 21 00 0C 02 ...	NN NN

Wiederholen des Auslesens des Objektverzeichnis mit dem vorherigen Request, bis die Response ein Fehler ist:

SA	FC	Daten	CRC
05	E5	0D	EA 94

9.4.8.1 Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h + 80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception Länge	2 Bytes	6
MEI type	1 Byte	0D _h
Exception code	1 Byte	CE _h
Fehlercode	4 Bytes	CANopen-Fehlercode, siehe nachfolgende Tabelle
CRC	2 Bytes	

CANopen-Fehlercode	Beschreibung
FFFF0000 _h	<i>Abort no error</i>
FFFF1003 _h	Service wird nicht unterstützt
FFFF1004 _h	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>
FFFF0003 _h	Unbekanntes oder nicht valides Kommando
FFFF0008 _h	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E _h	Allgemeiner Fehler im Parameter
FFFF0011 _h	Länge des Parameters falsch
FFFF0012 _h	Länge des Parameters zu groß
FFFF0013 _h	Länge des Parameters zu klein
FFFF0015 _h	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 _h	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 _h	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 _h	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 _h	Allgemeiner Fehler
FFFF001E _h	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 _h	Ungültige Sequenz der Nachrichten (z. B. wenn der Wert des <i>Counter-Bytes</i> entsprechend dem vorherigen Request oder Response nicht stimmt)

Im Falle, dass das nicht unterstützte Kontrolloptions-Bit gesetzt ist, wird folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h + 80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D _h
Exception code	1 Byte	AE _h
Supported protocol control	1 oder 2 Bytes	Siehe nachfolgende Tabelle
CRC	2 Bytes	

Bit	Name	Beschreibung
0	Merker "Verlängerung"	Dieses Bit wird genutzt, wenn das Objektverzeichnis Datenset größer ist, als in ein Modbus-Kommando passen würde. Das Datenset wird dann über mehrere Modbus-Nachrichten gestreckt, jede Nachricht enthält einen Teil des Datensets. "0" = Keine mehrfache Nachrichtentransaktion ("multiple message transaction") oder das Ende der mehrfachen Nachrichtentransaktion. "1" = Teil einer mehrfachen Nachrichtentransaktion.
1	Erweiterte Protokoll-Kontrolle	Länge der Protokoll-Kontrolle, der Wert "0" zeigt eine Länge von 1 Byte an, der Wert "1" eine Länge von 2 Byte.
2	Zähler Byte Option	Dieses Bit ist auf "1" um anzuzeigen, dass das Feld "counter byte" in dieser Nachricht genutzt wird. Ist dieses Bit auf "0" gesetzt, existiert das Feld "counter byte" nicht in dieser Nachricht.
3 und 4	Reserviert	0
5	Network ID Option	Nicht unterstützt, muss "0" sein.
6	Encodierte Datenoption	Nicht unterstützt, muss "0" sein.
7	Zugriffsmerker	Dieses Bit zeigt die Zugriffsmethode des angeforderten Kommandos an. "0" = lesen, "1" = schreiben.
8 to 15	Reserviert	0

Das nachfolgende Beispiel zeigt einen Fehler im Falle eines fehlerhaften Request. Der Request liest das 6061_h:00 mit der Länge von 2 Byte, das Objekt ist aber nur 1 Byte groß:

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 60 00 00 00 00 02	79 8D

Response

SA	FC	Daten	CRC
05	2B	FF 00 06 0D CE 12 00 07 06	AC 3C

9.4.9 FC 102 (66_h) Read complete array or record

Dieser Funktionscode wird zum Auslesen eines gesamten Arrays oder Records vom Objektverzeichnis verwendet.

Um das Auslesen des Arrays zu starten oder neu zu starten, muss der Unterfunktionscode 55_h versendet werden. Dieser Code setzt das Auslesen auf das Objekt mit Subindex 00_h zurück. Alle nachfolgenden Requests müssen dann den Unterfunktionscode AA_h enthalten. Zum Ende, wenn alle Objekte ausgelesen wurden, wird eine "Error Response" generiert.

Das Format jedes "Objekt lesen" ist folgendermaßen:

Request:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	66 _h
Unterfunktionscode	1 Byte	55 _h oder AA _h
Länge der Daten	1 Byte	00 _h
Index des zu lesenden Arrays	2 Bytes	
CRC	2 Bytes	

Response:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	65 _h
Funktionscode	1 Byte	
Unterfunktionscode	1 Byte	
Länge der Daten	1 Byte	
n mal Objektverzeichnis-Frame	1 - 252 Bytes	
CRC	2 Bytes	

Ein Objektverzeichnis-Frame besteht aus den folgenden Bytes:

Name	Wert / Bemerkung
Index Low Byte	1 Byte
Index High Byte	1 Byte
Subindex	1 Byte
Anzahl der Bytes	1 Byte Anzahl m der validen Daten im Datenfeld
Daten Byte	m-1 Byte

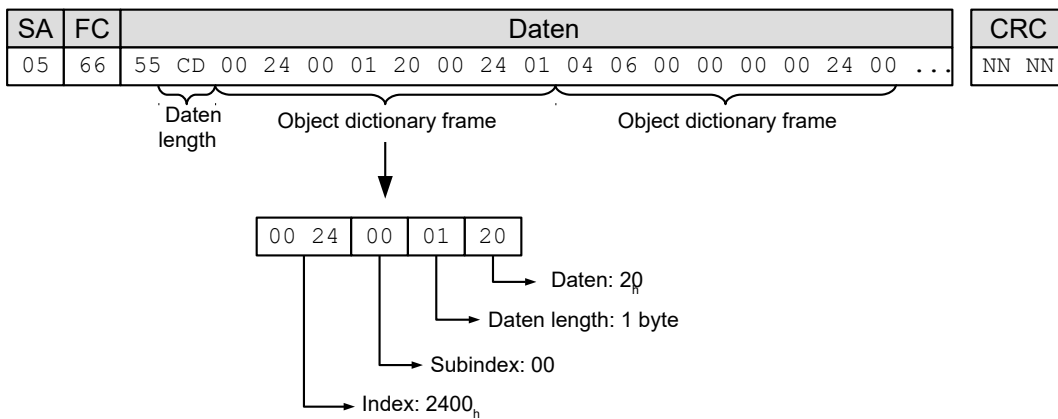
Beispiel

Alle folgenden Zahlenwerte sind in Hexadezimal notiert, der Index des zu lesenden Objektes ist 2400_h. Die Adresse des Slaves ist "5"_h.

Start des Auslesens des Arrays mit dem Request:

SA	FC	Daten	CRC
05	66	55 00 24 00	02 8A

Die Response ist:



9.4.9.1 Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h + 80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception Länge	2 Bytes	6
MEI type	1 Byte	0D _h
Exception code	1 Byte	CE _h
Fehlercode	4 Bytes	CANopen-Fehlercode, siehe nachfolgende Tabelle
CRC	2 Bytes	

CANopen-Fehlercode	Beschreibung
FFFF0000 _h	<i>Abort no error</i>
FFFF1003 _h	Service wird nicht unterstützt
FFFF1004 _h	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>
FFFF0003 _h	Unbekanntes oder nicht valides Kommando
FFFF0008 _h	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E _h	Allgemeiner Fehler im Parameter
FFFF0011 _h	Länge des Parameters falsch
FFFF0012 _h	Länge des Parameters zu groß
FFFF0013 _h	Länge des Parameters zu klein
FFFF0015 _h	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 _h	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 _h	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 _h	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 _h	Allgemeiner Fehler

CANopen-Fehlercode	Beschreibung
FFFF001E _h	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 _h	Ungültige Sequenz der Nachrichten (z. B. wenn der Wert des <i>Counter-Bytes</i> entsprechend dem vorherigen Request oder Response nicht stimmt)

Im Falle, dass das nicht unterstützte Kontrolloptions-Bit gesetzt ist, wird folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B _h + 80 _h (171 _d = 43 _d + 128 _d) (zeigt Fehler an)
Modbus exception code	1 Byte	FF _h ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D _h
Exception code	1 Byte	AE _h
Supported protocol control	1 oder 2 Bytes	Siehe nachfolgende Tabelle
CRC	2 Bytes	

Bit	Name	Beschreibung
0	Merker "Verlängerung"	Dieses Bit wird genutzt, wenn das Objektverzeichnis Datenset größer ist, als in ein Modbus-Kommando passen würde. Das Datenset wird dann über mehrere Modbus-Nachrichten gestreckt, jede Nachricht enthält einen Teil des Datensets. "0" = Keine mehrfache Nachrichtentransaktion ("multiple message transaction") oder das Ende der mehrfachen Nachrichtentransaktion. "1" = Teil einer mehrfachen Nachrichtentransaktion.
1	Erweiterte Protokoll-Kontrolle	Länge der Protokoll-Kontrolle, der Wert "0" zeigt eine Länge von 1 Byte an, der Wert "1" eine Länge von 2 Byte.
2	Zähler Byte Option	Dieses Bit ist auf "1" um anzuzeigen, dass das Feld "counter byte" in dieser Nachricht genutzt wird. Ist dieses Bit auf "0" gesetzt, existiert das Feld "counter byte" nicht in dieser Nachricht.
3 und 4	Reserviert	0
5	Network ID Option	Nicht unterstützt, muss "0" sein.
6	Encodierte Datenoption	Nicht unterstützt, muss "0" sein.
7	Zugriffsmerker	Dieses Bit zeigt die Zugriffsmethode des angeforderten Kommandos an. "0" = lesen, "1" = schreiben.
8 to 15	Reserviert	0

Das nachfolgende Beispiel zeigt einen Fehler im Falle eines fehlerhaften Request. Der Request liest das 6061_h:00 mit der Länge von 2 Byte, das Objekt ist aber nur 1 Byte groß:

Request

SA	FC	Daten	CRC
05	2B	0D 00 00 01 60 60 00 00 00 00 02	79 8D

Response

SA	FC	Daten	CRC
05	2B	FF 00 06 0D CE 12 00 07 06	AC 3C

9.4.10 Ausnahmecodes

Im Fehlerfall können abhängig vom Funktionscode folgende Ausnahmecodes in der Response enthalten sein:

Code	Name	Beschreibung
01	Illegal Function	Funktionscode nicht erkannt/erlaubt
02	Illegal Data Address	Register-Adresse nicht gültig oder existiert nicht
03	Illegal Data Value	Wert nicht gültig
04	Device Failure	nicht behebbarer Fehler

Weitere Details entnehmen Sie der Modbus-Spezifikation *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3*.

9.5 Prozessdatenobjekte (PDO)

Wie bei CANopen kann bei Modbus ein Prozessimage für Eingangs- und Ausgangsgrößen konfiguriert werden. Dieses Image beinhaltet nur noch Datenwerte einer oder mehrerer Objekte ohne Zusatzinformation wie Länge, Index oder Subindex. Damit lassen sich mittels einer Nachricht gleich mehrere Objekte lesen oder schreiben.

9.5.1 Konfiguration

Die Konfiguration des Image wird als "Mapping" bezeichnet und in folgenden Objekten geschrieben:

- 3502_h für das Modbus Rx (Master → Slave) PDO-Mapping
- 3602_h für das Modbus Tx (Slave → Master) PDO-Mapping

Beide Objekte beinhalten einen Array mit jeweils 16 Einträge. Der Subindex 00 gibt dabei die Anzahl der gültigen Einträge an.

Die Objekte 3502_h und 3602_h lassen sich mit Nachrichten mit dem Modbus-Funktionscode 2B_h beschreiben.

9.5.2 Übertragung

Die Daten werden aufeinander folgend ohne Lücke und Ausrichtung in die Nachricht geschrieben.

Wird ein Alignment (z.B. 16-Bit-Alignment) benötigt, kann man zusätzliche "Dummy-Objekte" mit in die Nachricht einbauen. Dummy-Objekte werden immer mit den Datenwert "0" übertragen. Diese Objekte sind in der nachfolgenden Tabelle abgedruckt.

Index	Datentyp
0002 _h	Vorzeichenbehaftete Ganzzahl (8 Bit)
0003 _h	Vorzeichenbehaftete Ganzzahl (16 Bit)

Index	Datentyp
0004 _h	Vorzeichenbehaftete Ganzzahl (32 Bit)
0005 _h	Vorzeichenlose Ganzzahl (8 Bit)
0006 _h	Vorzeichenlose Ganzzahl (16 Bit)
0007 _h	Vorzeichenlose Ganzzahl (32 Bit)

Das Mapping ist wie folgt:

- Das PDO RX Image fängt an der Modbus-Register-Adresse 6000_d (1770_h) an.
- Das PDO TX Image fängt an der Modbus-Register-Adresse 5000_d (1388_h) an.

Der Zugriff kann mit Funktionscode 17_h lesend/schreibend gleichzeitig erfolgen oder mit den Kommandos 03_h, 04_h, 06_h, 10_h auf die jeweiligen RX/TX Images.

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.

Beispiel

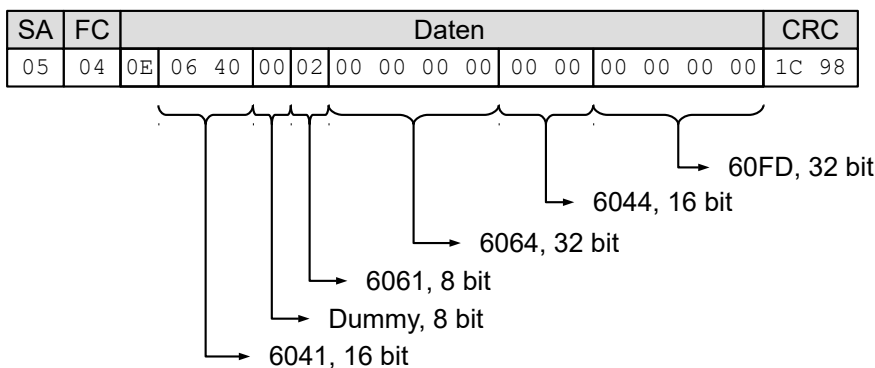
In dem Mapping sollen folgende Objekte eingestellt werden:

- 3602_h:00_h = "0_h" (Mapping wird deaktiviert)
- 3602_h:01_h = "60410010_h" (das Objekt 6041_h:00_h, Länge 16 Bit wird gemappt)
- 3602_h:02_h = "00050008_h" (das Dummy-Objekt 0005_h:00_h, Länge 8 Bit wird gemappt)
- 3602_h:03_h = "60610008_h" (das Objekt 6061_h:00_h, Länge 8 Bit wird gemappt)
- 3602_h:04_h = "60640020_h" (das Objekt 6064_h:00_h, Länge 32 Bit wird gemappt)
- 3602_h:05_h = "60440010_h" (das Objekt 6044_h:00_h, Länge 16 Bit wird gemappt)
- 3602_h:06_h = "60FD0020_h" (das Objekt 60FD_h:00_h, Länge 32 Bit wird gemappt)
- 3602_h:00_h = "6_h" (6 Werte werden gemappt)

Nach dem Mapping für das Objekt 6061_h:00_h wird ein Dummy-Objekt eingefügt, damit das nachfolgende Objekt 6064_h:00_h auf 32 Bit ausgerichtet wird.

RX Nachricht: Der Master schickt an den Slave folgende Nachricht:

TX Nachricht: Der Slave schickt an den Master folgende Antwort:



9.6 NanoJ-Objekte

Die NanoJ-Objekte 2400_h NanoJ Input und 2500_h (NanoJ Output) werden wie das Prozessimage auf Modbus-Register gemappt:

- 2500_h mit 32 x 32 Bit Werten wird auf die Modbus Register Adresse ab 2000_d (7D0_h) gemappt und kann auf diese Weise nur gelesen werden.
- 2400_h mit 32 x 32 Bit Werten wird auf die Modbus Register Adresse ab 3000_d (BB8_h) gemappt und kann auf diese Weise nur beschrieben werden.

Für den Zugriff können die Kommandos mit Funktionscode 03_h, 04_h, 10_h und 17_h verwendet werden. Es gilt die Einschränkung, dass die Adresse auf 32 Bit ausgerichtet (aligned) sein muss und auch bei einem Schreibvorgang immer mindestens 32 Bit geschrieben werden muss, damit die Daten konsistent sind.

Beispiel

Request: Der Master schickt an den Slave folgende Nachricht:

SA	FC	Daten	CRC
05	17	07 D0 00 08 0B B8 00 08 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	41 21

Reply: Der Slave schickt an den Master folgende Antwort:

SA	FC	Daten	CRC
05	17	10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	50 9D

10 Programmierung mit NanoJ

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

10.1 NanoJ-Programm

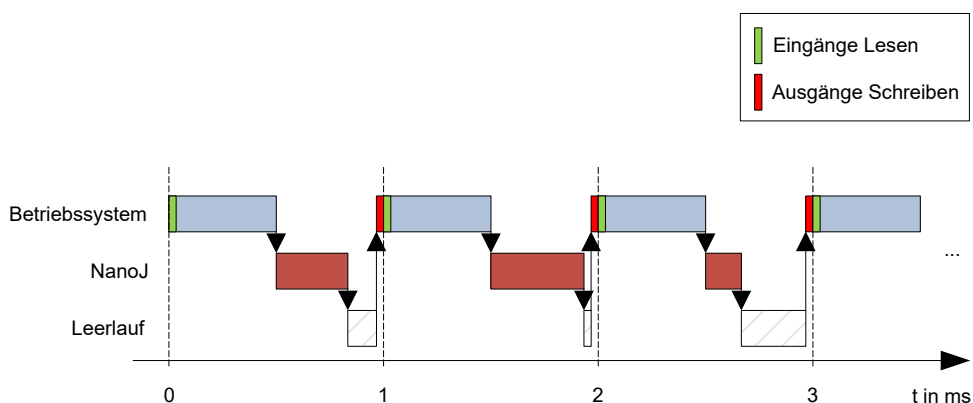
Ein *NanoJ-Programm* stellt eine geschützte Ausführungsumgebung innerhalb der Firmware zur Verfügung. In dieser kann der Anwender eigene Abläufe anlegen. Diese können dann Funktionen in der Steuerung auslösen, indem beispielsweise Einträge im Objektverzeichnis gelesen oder geschrieben werden.

Durch Verwendung von Schutzmechanismen wird verhindert, dass ein *NanoJ-Programm* die Firmware zum Absturz bringt. Im schlimmsten Fall wird die Ausführung mit einem im Objektverzeichnis hinterlegten Fehlercode abgebrochen.

Wenn das *NanoJ-Programm* auf die Steuerung geladen wurde, wird es nach dem Einschalten oder Neustarten der Steuerung automatisch ausgeführt, sofern Sie Bit 0 im Objekt `2300n` nicht auf "0" setzen.

10.1.1 Verfügbare Rechenzeit

Ein *NanoJ-Programm* erhält zyklisch im 1 ms-Takt Rechenzeit (siehe folgende Abbildung). Da durch Interrupts und Systemfunktionen der Firmware Rechenzeit verloren geht, stehen dem Benutzerprogramm (abhängig von Betriebsart und Anwendungsfall) nur ca. 30% ... 50% Rechenzeit zur Verfügung. In dieser Zeit muss das Benutzerprogramm den Zyklus durchlaufen und entweder beenden oder durch Aufruf der Funktion `yield()` die Rechenzeit abgeben. Bei Ersterem wird das Benutzerprogramm mit dem Beginn des nächsten 1 ms-Zyklus wieder neu gestartet, letzteres bewirkt eine Fortsetzung des Programms an dem der Funktion `yield()` nachfolgenden Befehl beim nächsten 1 ms-Zyklus.



Falls das *NanoJ-Programm* mehr als die ihm zugeteilte Zeit benötigt, wird es beendet und im Objektverzeichnis ein Fehlercode gesetzt.

TIPP



Bei der Entwicklung von Benutzerprogrammen ist speziell bei zeitintensiveren Aufgaben eine sorgfältige Überprüfung des Laufzeitverhaltens durchzuführen. So empfiehlt sich beispielsweise die Verwendung von Tabellen, anstatt einen Sinuswert über eine `sin` Funktion zu berechnen.

HINWEIS



Sollte das *NanoJ-Programm* zu lange die Rechenzeit nicht abgeben, wird es vom Betriebssystem beendet. In diesem Fall wird in das Statusword bei Objekt 2301_h die Ziffer 4 eingetragen, im Fehlerregister bei Objekt 2302_h wird die Ziffer 5 (Timeout) notiert, siehe [2301h NanoJ Status](#) und [2302h NanoJ Error Code](#).

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.

10.1.2 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.

10.1.3 NanoJ-Programm - Kommunikationsmöglichkeiten

Ein *NanoJ-Programm* hat mehrere Möglichkeiten, mit der Steuerung zu kommunizieren:

- Lesen und Schreiben von OD-Werten per PDO-Mapping
- direktes Lesen und Schreiben von OD-Werten über 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

10.1.4 NanoJ-Programm ausführen

Zusammengefasst besteht das *NanoJ-Programm* bei der Ausführung eines Zyklus hinsichtlich des PDO-Mappings aus folgenden drei Schritten:

1. Werte aus dem Objektverzeichnis lesen und in die Bereiche Inputs und Outputs kopieren
2. Benutzerprogramm ausführen
3. Werte aus den Bereichen Outputs und Inputs zurück in das Objektverzeichnis kopieren

Die Konfiguration der Kopiervorgänge ist dem CANopen-Standard angelehnt.

Zusätzlich kann über 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.

10.1.5 NanoJ-Programm OD-Einträge

Das *NanoJ-Programm* wird durch OD-Einträge im Objekt-Bereich 2300_h bis 2330_h gesteuert und konfiguriert (siehe [2300h NanoJ Control](#)).

OD-Index	Name und Beschreibung
2300 _h	2300h NanoJ Control
2301 _h	2301h NanoJ Status
2302 _h	2302h NanoJ Error Code
2310 _h	2310h NanoJ Input Data Selection
2320 _h	2320h NanoJ Output Data Selection
2330 _h	2330h NanoJ In/output Data Selection

Beispiel:

Um das Benutzerprogramm *TEST1.USR* zu starten, kann z. B. folgende Sequenz benutzt werden:

- Überprüfen des Eintrags [2302_h](#) auf Fehlercode.
- Wenn kein Fehler:
NanoJ-Programm starten durch Beschreiben von Objekt [2300_h](#), Bit 0 = "1" 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".

10.1.6 Aufbau NanoJ-Programm

Ein Benutzerprogramm besteht aus mindestens zwei Anweisungen:

- der Präprozessoranweisung `#include "wrapper.h"`
- der Funktion `void user() {}`

In der Funktion `void user()` lässt sich der auszuführende Code hinterlegen.

HINWEIS



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

HINWEIS



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

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

Beispiele:

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

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

Folgende Zuweisung führt zu einem Fehler beim Kompilieren:

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

10.1.7 NanoJ-Programmbeispiel

Das Beispiel zeigt das Programmieren eines Rechtecksignals in das Objekt `2500h:01h`.

```
// file main.cpp
map S32 outputReg1 as inout 0x2500:1
#include "wrapper.h"

// user program
void user()
{
  U16 counter = 0;
  while( 1 )
  {
    ++counter;

    if( counter < 100 )
      InOut.outputReg1 = 0;
    else if( counter < 200 )
      InOut.outputReg1 = 1;
  }
}
```

```

else
  counter = 0;

  // yield() 5 times (delay 5ms)
  for(U08 i = 0; i < 5; ++i )
    yield();
}
} // eof

```

Weitere Beispiele finden Sie auf www.nanotec.de.

10.2 Mapping im NanoJ-Programm

Mit dieser Methode wird eine Variable im *NanoJ-Programm* direkt mit einem Eintrag im Objektverzeichnis verknüpft. Das Anlegen des Mappings muss dabei am Anfang der Datei stehen - noch vor der `#include "wrapper.h"`-Anweisung.

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](#).

10.2.1 Deklaration des Mappings

Die Deklaration des Mappings gliedert sich dabei folgendermaßen:

```
map <TYPE> <NAME> as <input|output|inout> <INDEX>:<SUBINDEX>
```

Dabei gilt:

- `<TYPE>`
Der Datentyp der Variable; U32, U16, U08, S32, S16 oder S08.
- `<NAME>`
Der Name der Variable; wie sie im Benutzerprogramm verwendet wird.
- `<input|output|inout>`
Die Schreib- und Leseberechtigung einer Variable: Eine Variable kann entweder als `input`, `output` oder `inout` deklariert werden. Damit wird festgelegt, ob eine Variable lesbar (`input`), schreibbar (`output`) oder beides ist (`inout`) und über welche Struktur sie im Programm angesprochen werden muss.
- `<INDEX>:<SUBINDEX>`
Index und Subindex des zu mappenden Objekts im Objektverzeichnis.

Jede deklarierte Variable wird im Benutzerprogramm über eine der drei Strukturen *In*, *Out* oder *InOut* angesprochen, je nach definierter Schreib- und Leserichtung.

HINWEIS



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

10.2.2 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;
  [...]
}
```

10.2.3 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 `6040h:00h`.
2. Am Ende des 1 ms-Zyklus wird das Mapping geschrieben, welches ebenfalls das Objekt `6040h:00h` beschreibt, allerdings mit dem Wert 1.
3. Somit wird - aus Sicht des Benutzers - der `od_write`-Befehl wirkungslos.

10.3 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.

10.3.1 Zugriff auf das Objektverzeichnis

void **od_write** (U32 index, U32 subindex, U32 value)

Diese Funktion schreibt den übergebenen Wert an die angegebene Stelle in das Objektverzeichnis.

index	Index des zu schreibenden Objekts im Objektverzeichnis
subindex	Subindex des zu schreibenden Objekts im Objektverzeichnis
value	zu schreibender Wert

HINWEIS



Es wird dringend empfohlen, nach dem Aufruf eines `od_write()` die Prozessorzeit mit `yield()` abzugeben. Der Wert wird zwar sofort ins OD geschrieben. Damit die Firmware jedoch davon abhängige Aktionen auslösen kann, muss diese Rechenzeit erhalten und somit das Benutzerprogramm beendet oder mit `yield()` unterbrochen worden sein.

U32 **od_read** (U32 index, U32 subindex)

Diese Funktion liest den Wert an der angegebenen Stelle aus dem Objektverzeichnis und gibt ihn zurück.

index	Index des zu lesenden Objekts im Objektverzeichnis
subindex	Subindex des zu lesenden Objekts im Objektverzeichnis
Rückgabewert	Inhalt des OD-Eintrags

HINWEIS



Aktives Warten auf einen Wert im Objektverzeichnis sollte immer mit einem `yield()` verbunden werden.

Beispiel

```
while (od_read(2400,2) != 0) // wait until 2400:2 is set
{ yield(); }
```

10.3.2 Prozesssteuerung

void **yield**()

Diese Funktion gibt die Prozessorzeit wieder an das Betriebssystem ab. Das Programm wird in der nächsten Zeitscheibe wieder an der Stelle nach dem Aufruf fortgesetzt.

void **sleep** (U32 ms)

Diese Funktion gibt die Prozessorzeit für die angegebene Zahl an Millisekunden an das Betriebssystem ab. Das Benutzerprogramm wird anschließend an der Stelle nach dem Aufruf fortgesetzt.

ms	Zu wartende Zeit in Millisekunden
----	-----------------------------------

10.3.3 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 Objektverzeichnis geschrieben und dann von dort von *Plug & Drive Studio* ausgelesen.

Dieser OD-Eintrag hat den Index 2600_n 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](#)).

10.4 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 <i>Plug & Drive Studio</i> unmöglich.	Greifen Sie stattdessen mit <code>od_read/od_write</code> auf das Objekt zu.

Einschränkung/Problem	Maßnahme
Wenn ein Objekt als Output gemappt wurde und der Wert des Objekts niemals vor dem Start des <i>NanoJ-Programms</i> 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 <code>constant array</code> .
Zu viele lokale Variablen und Arrays innerhalb von Funktionen können zu einem Stacküberlauf führen.	Deklarieren Sie die Variablen 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.
<code>float</code> darf nicht mit Vergleichsoperatoren verwendet werden.	Verwenden Sie stattdessen <code>int</code> .
<code>double</code> 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.	
<code>math</code> oder <code>cmath</code> können nicht einbezogen werden.	

11 Objektverzeichnis Beschreibung

11.1 Übersicht

In diesem Kapitel finden Sie eine Beschreibung aller Objekte.

Sie finden hier Angaben zu:

- Funktionen
- Objektbeschreibungen ("Index")
- Wertebeschreibungen ("Subindices")
- Beschreibungen von Bits
- Beschreibung des Objekts

11.2 Aufbau der Objektbeschreibung

Die Beschreibung der Objekteinträge ist immer gleich aufgebaut und besteht im Normalfall aus folgenden Abschnitten:

Funktion

In diesem Abschnitt wird kurz die Funktion des Objektverzeichnisses beschrieben.

Objektbeschreibung

Diese Tabelle gibt detailliert Auskunft über den Datentyp, Vorgabewerte und dergleichen. Eine genaue Beschreibung findet sich im Abschnitt "[Objektbeschreibung](#)"

Wertebeschreibung

Diese Tabelle ist nur bei dem Datentyp "Array" oder "Record" verfügbar und gibt genaue Auskunft über die Untereinträge. Eine genauere Beschreibung der Einträge findet sich im Abschnitt "[Wertebeschreibung](#)"

Beschreibung

Hier werden genauere Angaben zu den einzelnen Bits eines Eintrags gemacht oder eventuelle Zusammensetzungen erläutert. Eine genauere Beschreibung findet sich im Abschnitt "[Beschreibung](#)"

11.3 Objektbeschreibung

Die Objektbeschreibung besteht aus einer Tabelle, welche folgende Einträge enthält:

Index

Benennt den Index des Objekts in Hexadezimalschreibweise.

Objektname

Der Name des Objekts.

Object Code

Der Typ des Objekts. Das kann einer der folgenden Einträge sein:

- VARIABLE: In dem Fall besteht das Objekt nur aus einer Variable, die mit dem Subindex 0 indiziert wird.
- ARRAY: Diese Objekte bestehen immer aus einem Subindex 0 - welcher die Menge der Untereinträge angibt - und den Untereinträgen selber ab dem Index 1. Der Datentyp innerhalb eines Arrays ändert sich nie, das heißt, Untereintrag 1 und alle folgenden Einträge haben immer den gleichen Datentyp.
- RECORD: Diese Objekte bestehen immer aus einem Untereintrag mit dem Subindex 0 - welcher die Menge der Untereinträge angibt - und den Untereinträgen selber ab dem Index 1. Im Gegensatz zu einem ARRAY kann der Datentyp der Subeinträge variieren, das bedeutet, dass beispielsweise Untereintrag 1 einen anderen Datentyp als Untereintrag 2 haben kann.

- **VISIBLE_STRING**: Das Objekt beschreibt eine in ASCII codierte Zeichenkette. Die Länge des Strings wird in Subindex 0 angegeben, die einzelnen Zeichen sind ab Subindex 1 gespeichert. Diese Zeichenketten sind **nicht** durch ein Null-Zeichen terminiert.

Datentyp

Hier wird die Größe und die Interpretation des Objekts angegeben. Für den Object Code "VARIABLE" gilt folgende Schreibweise:

- Es wird unterschieden zwischen Einträgen die vorzeichenbehaftet sind, das wird mit dem Präfix "SIGNED" bezeichnet. Für die vorzeichenunbehafteten Einträge wird das Präfix "UNSIGNED" benutzt.
- Die Größe der Variable in Bit wird an das Präfix angestellt und kann entweder 8, 16 oder 32 sein.

Speicherbar

Hier wird beschrieben ob dieses Objekt speicherbar ist und wenn ja, unter welcher Kategorie.

Firmware Version

Hier ist die Firmwareversion eingetragen, ab der das Objekt verfügbar ist.

Änderungshistorie (ChangeLog)

Hier werden eventuelle Änderungen an dem Objekt notiert.

Zudem gibt es noch die Einträge für den Datentyp "VARIABLE" folgende Tabelleneinträge:

Zugriff

Hier wird die Zugriffsbeschränkung eingetragen. Folgende Beschränkungen gibt es:

- "lesen/schreiben": Das Objekt kann sowohl gelesen, als auch geschrieben werden
- "nur lesen": Das Objekt kann nur aus dem Objektverzeichnis gelesen werden. Setzen eines Werte ist nicht möglich.

PDO-Mapping

Einige Bussysteme, wie CANopen oder EtherCAT unterstützen ein PDO-Mapping. In diesem Tabelleneintrag wird beschrieben, ob das Objekt in ein Mapping eingefügt werden darf und in welches. Dabei gibt es folgende Bezeichnungen:

- "no": Das Objekt darf in kein Mapping eingetragen werden.
- "TX-PDO": Das Objekt darf in ein RX Mapping eingetragen werden.
- "RX-PDO": Das Objekt darf in ein TX Mapping eingetragen werden.

Zulässige Werte

In einigen Fällen ist es nur erlaubt, bestimmte Werte in das Objekt zu schreiben. Sollte das der Fall sein, werden diese Werte hier aufgelistet. Besteht keine Beschränkung bleibt das Feld leer.

Vorgabewert

Um die Steuerung beim Einschalten in einen gesicherten Zustand zu bringen ist es nötig, einige Objekte mit Werten vorzubelegen. Der Wert, der beim Start der Steuerung in das Objekt geschrieben wird, wird in diesem Tabelleneintrag notiert.

11.4 Wertebeschreibung

HINWEIS



Der Übersichtlichkeit halber werden einige Subindizes zusammengefasst, wenn die Einträge alle den gleichen Namen haben.

In der Tabelle mit der Überschrift "Wertebeschreibung" werden alle Daten für Untereinträge mit Subindex 1 oder höher aufgelistet. Die Tabelle beinhaltet folgende Einträge:

Subindex

Nummer des aktuell beschriebenen Untereintrages.

Name

Der Name des Untereintrages.

Datentyp

Hier wird die Größe und die Interpretation des Untereintrages angegeben. Hier gilt immer folgende Schreibweise:

- Es wird unterschieden zwischen Einträgen die vorzeichenbehaftet sind, das wird mit dem Präfix "SIGNED" bezeichnet. Für die vorzeichenunbehafteten Einträge wird das Präfix "UNSIGNED" benutzt.
- Die Größe der Variable in Bit wird an das Präfix angestellt und kann entweder 8, 16 oder 32 sein.

Zugriff

Hier wird die Zugriffsbeschränkung für den Untereintrag eingetragen. Folgende Beschränkungen gibt es:

- "lesen/schreiben": Das Objekt kann sowohl gelesen, als auch geschrieben werden
- "nur lesen": Das Objekt kann nur aus dem Objektverzeichnis gelesen werden. Setzen eines Wertes ist nicht möglich.

PDO-Mapping

Einige Bussysteme, wie CANopen oder EtherCAT unterstützen ein PDO-Mapping. In diesem Tabelleneintrag wird beschrieben, ob der Untereintrag in ein Mapping eingefügt werden darf und in welches. Dabei gibt es folgende Bezeichnungen:

- "no": Das Objekt darf in kein Mapping eingetragen werden.
- "TX-PDO": Das Objekt darf in ein RX Mapping eingetragen werden.
- "RX-PDO": Das Objekt darf in ein TX Mapping eingetragen werden.

Zulässige Werte

In einigen Fällen ist es nur erlaubt, bestimmte Werte in den Untereintrag zu schreiben. Sollte das der Fall sein, werden diese Werte hier aufgelistet. Besteht keine Beschränkung, bleibt das Feld leer.

Vorgabewert

Um die Steuerung beim Einschalten in einen gesicherten Zustand zu bringen ist es nötig, einige Untereinträge mit Werten vor zu belegen. Der Wert, welcher beim Start der Steuerung in den Untereintrag geschrieben wird, wird in diesem Tabelleneintrag notiert.

11.5 Beschreibung

Dieser Abschnitt kann vorhanden sein, wenn die Benutzung zusätzliche Information verlangt. Sollten einzelne Bits eines Objekts oder Untereintrags unterschiedliche Bedeutung haben, so werden Diagramme wie im nachfolgenden Beispiel verwendet.

Beispiel: Das Objekt ist 8 Bit groß, Bit 0 und 1 haben separat eine Funktion. Bit 2 und 3 sind zu einer Funktion zusammengefasst, für Bit 4 bis 7 gilt das gleiche.

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

Beispiel [4]

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

Beispiel [2]

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

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

B

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

A

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

1000h Device Type

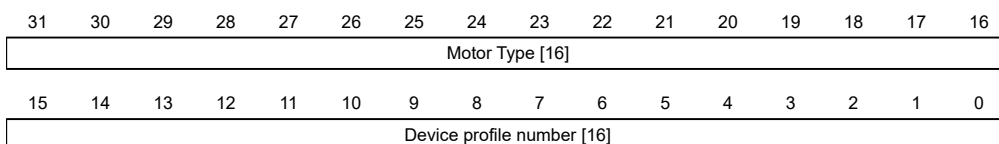
Funktion

Beschreibt den Steuerungstyp.

Objektbeschreibung

Index	1000 _h
Objektname	Device Type
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00060192 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung



Motor Type[16]

Beschreibt den unterstützten Motor-Typ. Die folgenden Werte sind möglich:

- Bit 23 bis Bit 16: Wert "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
<hr/>	
Subindex	08 _h
Name	8th Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

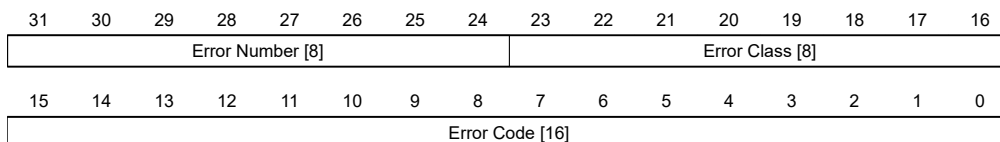
Beschreibung

Allgemeine Funktionsweise

Tritt ein neuer Fehler auf, wird dieser in Subindex 1 eingetragen. Die bereits vorhandenen Einträge in den Subindizes 1 bis 7 werden um eine Stelle nach hinten verschoben. Der Fehler auf Subindex 7 wird dabei entfernt.

Die Anzahl der bereits aufgetreten Fehler lässt sich aus dem Objekt mit dem Subindex 0 ablesen. Ist im Fehlerstapel zur Zeit kein Fehler eingetragen, dann ist das Auslesen eines der acht Subindizes 1-8 nicht möglich und wird mit einem Fehler (Abort-Code=08000024_h) beantwortet. Wird in den Subindex 0 eine "0" geschrieben, beginnt die Zählung von neuem.

Bitbeschreibung



Error Number [8]

Damit lässt sich der Grund des Fehlers genau eingrenzen. Die Bedeutung der Zahl lässt sich aus nachfolgender Tabelle entnehmen.

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

Fehlernummer	Beschreibung
12	Übertemperatur-Fehler
13	Die Werte des Objekts 6065 _h (Following Error Window) und des Objekts 6066 _h (Following Error Time Out) wurden überschritten, es wurde ein Fault ausgelöst.
14	Warnung: nichtflüchtiger Speicher voll. Der aktuelle Speichervorgang konnte nicht abgeschlossen werden, Teile der Daten des Speichervorgangs sind verloren. Neustart der Steuerung erforderlich für Aufräumarbeiten.
15	Motor blockiert
16	Warnung: nichtflüchtiger Speicher beschädigt, Neustart der Steuerung erforderlich für Aufräumarbeiten (alle gespeicherten Objekte werden auf Default zurückgesetzt).
17	Nur CANopen: Slave brauchte zu lange um PDO-Nachrichten zu Senden.
18	Sensor n (siehe 3204 _h), wo n größer 3: Fehler durch elektrische Störung oder defekte Hardware
19	Nur CANopen: PDO aufgrund eines Längenfehlers nicht verarbeitet
20	Nur CANopen: PDO Länge überschritten
21	Warnung: Starten Sie die Steuerung neu, um zukünftige Fehler beim Speichern (nichtflüchtiger Speicher voll/korrupt) zu vermeiden.
22	Nennstrom muss gesetzt werden (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
39	Fehler in der Ballast-Konfiguration: nicht gültige/realistische Parameter eingetragen (siehe <u>Ballast-Überwachung</u>)
40	Warnung: Ballast-Widerstand thermisch überlastet
46	Interlock-Fehler: Bit 3 in 60FD _h steht auf "0", der Motor darf nicht fahren (siehe Abschnitt <i>Interlock-Funktion</i> im Kapitel <u>Digitale Eingänge</u>)

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 <i>Interlock-Funktion</i> im Kapitel <u>Digitale Eingänge</u>)
6010 _h	Software-Reset (Watchdog)
6100 _h	Interner Softwarefehler, generisch

Error Code	Beschreibung
6320 _h	Nennstrom muss gesetzt werden (203B _h :01 _h /6075 _h)
7110 _h	Fehler in der Ballast-Konfiguration: nicht gültige/realistische Parameter eingetragen (siehe Ballast-Überwachung)
7113 _h	Warnung: Ballast-Widerstand thermisch überlastet
7121 _h	Motor blockiert
7200 _h	Interner Fehler: Korrekturfaktor für Referenzspannung fehlt im OTP
7305 _h	Sensor 1 (siehe 3204_h) fehlerhaft
7306 _h	Sensor 2 (siehe 3204_h) fehlerhaft
7307 _h	Sensor n (siehe 3204_h), wo n größer 2
7600 _h	Warnung: Nichtflüchtiger Speicher voll oder korrupt, Neustart der Steuerung für Aufräumarbeiten
8100 _h	Fehler bei der Feldbusüberwachung
8130 _h	Nur CANopen: "Life Guard"-Fehler oder "Heartbeat"-Fehler
8200 _h	Nur CANopen: Slave brauchte zu lange um PDO Nachrichten zu Senden.
8210 _h	Nur CANopen: PDO wurde nicht verarbeitet aufgrund eines Längen-Fehlers
8220 _h	Nur CANopen: PDO Länge überschritten
8240 _h	Nur CANopen: unerwartete Sync-Länge
8400 _h	Fehler in der Drehzahlüberwachung: Schlupffehler zu groß
8611 _h	Fehler in der Positionsüberwachung: Schleppfehler zu groß
8612 _h	Fehler in der Positionsüberwachung: Endschalter überschritten

1008h Manufacturer Device Name

Funktion

Enthält den Gerätenamen als Zeichenkette.

Objektbeschreibung

Index	1008 _h
Objektname	Manufacturer Device Name
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> ■ CPB6-1-5: CPB6-1-5 ■ CPB6-2-5: CPB6-2-5 ■ CPB3-1-5: CPB3-1-5 ■ CPB3-2-5: CPB3-2-5 ■ CPB15-5: CPB15-5 ■ CPB15-S-5: CPB15-S-5
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-v2213-B1030801
Firmware Version	FIR-v1426
Änderungshistorie	

1010h Store Parameters

Funktion

Mit diesem Objekt lässt sich das Speichern von Objekten starten. Siehe Kapitel [Objekte speichern](#).

Objektbeschreibung

Index	1010 _h
-------	-------------------

Objektname	Store Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1436: Eintrag "Objektname" geändert von "Store Parameter" auf "Store Parameters".</p> <p>Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 3 auf 4.</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5.</p> <p>Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7.</p> <p>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.</p>

Wertebeschreibung

Subindex	00 _h
Name	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
<hr/>	
Subindex	04 _h
Name	Save Customer Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	05 _h
Name	Save Drive Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	06 _h
Name	Save Tuning Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	07 _h
Name	Save Miscellaneous Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	08 _h
Name	Save Reserved1 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	09 _h
Name	Save Reserved2 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0A _h
Name	Save CANopen Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
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
<hr/>	
Subindex	0C _h
Name	Save Ethernet Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	0D _h
Name	Save Profibus Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000001 _h
-------------	-----------------------

Beschreibung

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

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

Für eine detaillierte Beschreibung siehe Kapitel [Objekte speichern](#).

1011h Restore Default Parameters

Funktion

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

Objektbeschreibung

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

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
<hr/>	
Subindex	06 _h
Name	Restore Tuning Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	07 _h
Name	Restore Miscellaneous Configurations
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	08 _h
Name	Restore Reserved1 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	09 _h
Name	Restore Reserved2 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0A _h
Name	Restore CANopen Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	0B _h
Name	Restore Modbus RTU Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	0C _h
Name	Restore Ethernet Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	0D _h
Name	Restore Profibus Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Wird der Wert 64616F6C_h (bzw. 1684107116_d oder ASCII `load`) in dieses Objekt geschrieben, werden Teile oder das gesamte Objektverzeichnis auf die Defaultwerte zurückgesetzt. Der verwendete Subindex entscheidet darüber, welcher Bereich zurück gesetzt wird.

Für eine detaillierte Beschreibung siehe Kapitel [Speicherung verwerfen](#).

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	<ul style="list-style-type: none"> ■ CPB6-1-5: 00000194_h ■ CPB6-2-5: 000001B2_h ■ CPB3-1-5: 000001C6_h ■ CPB3-2-5: 000001DA_h ■ CPB15-5: 0000020E_h ■ CPB15-S-5: 00000218_h

Subindex	03 _h
Name	Revision Number
Datentyp	UNSIGNED32

Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	08A50000 _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
<hr/>	
Subindex	01 _h
Name	Program Status Bootloader/firmware
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	Program Status NanoJ
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

2028h MODBUS Slave Address

Funktion

Dieses Objekt enthält die Slave-Adresse für Modbus.

Objektbeschreibung

Index	2028 _h
Objektname	MODBUS Slave Address
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Modbus RTU
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	1-247
Vorgabewert	05 _h
Firmware Version	FIR-v1436
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Modbus RTU".

202Ah MODBUS RTU Baudrate

Funktion

Dieses Objekt enthält die Baudrate des Modbus in Bd.

Objektbeschreibung

Index	202A _h
Objektname	MODBUS RTU Baudrate
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Modbus RTU
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00004B00 _h
Firmware Version	FIR-v1436
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Modbus RTU".

202Ch MODBUS RTU Stop Bits

Funktion

Dieses Objekt enthält die Anzahl der Stop-Bits des Modbus.

Objektbeschreibung

Index	202C _h
Objektname	MODBUS RTU Stop Bits
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h
Firmware Version	FIR-v1436
Änderungshistorie	Firmware Version FIR-v1540: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "nein". Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".

Beschreibung

Die Anzahl der Stopbits ist abhängig von der Parity, welche im Objekt 202D_h eingestellt werden kann.

202Dh MODBUS RTU Parity

Funktion

Dieses Objekt stellt bei Modbus RTU die Anzahl der Paritybits und Stopbits ein.

Objektbeschreibung

Index	202D _h
Objektname	MODBUS RTU Parity
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Modbus RTU
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1748-B531667: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Modbus RTU".

Beschreibung

Folgende Werte gelten:

- Wert "0x00": Parity None, Stop Bits 2
- Wert "0x04": Parity Even, Stop Bits 1
- Wert "0x06": Parity Odd, Stop Bits 1

2030h Pole Pair Count

Funktion

Enthält die Polpaarzahl des angeschlossenen Motors.

Objektbeschreibung

Index	2030 _h
Objektname	Pole Pair Count
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000032 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "nein" auf "ja, Kategorie: Tuning".

2031h Max Motor Current

Funktion

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

Steuerungsintern wird der eingegebene Wert immer als Effektivwert interpretiert.

Objektbeschreibung

Index	2031 _h
Objektname	Max Motor Current
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000258 _h
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".</p> <p>Firmware Version FIR-v1614: Eintrag "Object Name" geändert von "Peak Current" auf "Max Current".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Object Name" geändert von "Maximum Current" auf "Max Motor Current".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Object Name" geändert von "Max Motor Current" auf "Maximum Current".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Object Name" geändert von "Maximum Current" auf "Max Motor Current".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Object Name" geändert von "Max Motor Current" auf "Maximum Current".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Object Name" geändert von "Maximum Current" auf "Max Motor Current".</p>

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	<ul style="list-style-type: none"> ■ CPB6-1-5: 0000EE48_h ■ CPB6-2-5: 00011365_h ■ CPB3-1-5: 0000EE48_h ■ CPB3-2-5: 0000EE48_h ■ CPB15-5: 0000EE48_h ■ CPB15-S-5: 0000EE48_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	00002D78 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

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

2036h Open Loop Current Reduction Idle Time

Funktion

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

Objektbeschreibung

Index	2036 _h
Objektname	Open Loop Current Reduction Idle Time
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Firmware Version	FIR-v1426
Änderungshistorie	

2037h Open Loop Current Reduction Value/factor

Funktion

Dieses Objekt beschreibt den Effektivstrom, auf den der Motorstrom reduziert werden soll, wenn die Stromabsenkung im Open Loop aktiviert wird (Bit 3 in 3202_h = "1") und sich der Motor im Stillstand befindet.

Objektbeschreibung

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

Beschreibung

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

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

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

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

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

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

2038h Brake Controller Timing

Funktion

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

Objektbeschreibung

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

Wertebeschreibung

Subindex	00 _h
Name	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 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 in Hertz.
- 06_h: Tastgrad des PWM-Signals in Prozent.

2039h Motor Currents

Funktion

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

Objektbeschreibung

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

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	I _q
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	I _α
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	I _β
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: I_α
- 04_h: I_β
- 05_h: Gesamtstrom durch $\sqrt{2}$, also runtergerechnet auf eine Motorphase. Im *Closed Loop* wird zusätzlich das Vorzeichen von I_q 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: $I = \text{sgn}(I_q) * \sqrt{(I_{\alpha}^2 + I_{\beta}^2)} / \sqrt{2}$

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

Wertebeschreibung

Subindex	00 _h
Name	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	000005DC _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 I²t-Überwachung.

Die I²t-Überwachung wird aktiviert, in dem in 203B_h:01 und 203B_h:02 ein Wert größer 0 eingetragen wird und in 6073_h ein Wert größer 1000 (siehe [I2t Motor-Überlastungsschutz](#)).

I²t kann nur für den *Closed Loop*-Betrieb verwendet werden, mit einer Ausnahme: Wenn I²t im *Open Loop*-Betrieb aktiviert ist, wird der Strom auf den kleineren der Werte von 203B_h:01_h, 6073_h und 2031_h begrenzt.

Objektbeschreibung

Index	203B _h
Objektname	I2t Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1512: Eintrag "Savable" geändert von "nein" auf "ja, Kategorie: Applikation".</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 7 auf 8.</p> <p>Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".</p>

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	00000258 _h

Subindex	02 _h
Name	Maximum Duration Of Max Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _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 A²ms an, von der abhängt, ob auf Maximalstrom oder Nennstrom geschaltet wird.
- 04_h: CalcValue, gibt den berechneten Wert in A²ms an, welcher mit Threshold verglichen wird, um den Strom einzustellen.
- 05_h: LimitedCurrent, zeigt den gegenwärtigen Strom als Effektivwert an, der von I²t eingestellt wurde.
- 06_h: aktueller Status. Ist der Subentry-Wert "0", ist I²t deaktiviert, ist der Wert "1", wird I²t aktiviert.

203Dh Torque Window

Funktion

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

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

Objektbeschreibung

Index	203D _h
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.

2057h Clock Direction Multiplier

Funktion

Mit diesem Wert wird der Takt-Zählwert im Takt-Richtungs-Modus multipliziert, bevor er weiterverarbeitet wird.

Objektbeschreibung

Index	2057 _h
Objektname	Clock Direction Multiplier
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000080 _h
Firmware Version	FIR-v1426
Änderungshistorie	

2058h Clock Direction Divider

Funktion

Durch diesen Wert wird der Takt-Zählwert im Takt-Richtungs-Modus dividiert, bevor er weiterverarbeitet wird.

Objektbeschreibung

Index	2058 _h
Objektname	Clock Direction Divider
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Firmware Version	FIR-v1426
Änderungshistorie	

205Ah Absolute Sensor Boot Value (in User Units)

Funktion



TIPP

Dieses Objekt hat nur bei Verwendung eines Absolut-Encoders eine Funktion. Wird kein Absolut-Encoder verwendet, ist der Wert immer 0.

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

Objektbeschreibung

Index	205A _h
Objektname	Absolute Sensor Boot Value (in User Units)
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1446
Änderungshistorie	<p>Firmware Version FIR-v1512: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Encoder Boot Value" auf "Absolute Sensor Boot Value (in User Units)".</p>

Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".

205Bh Clock Direction Or Clockwise/Counter Clockwise Mode

Funktion

Mit diesem Objekt lässt sich der Takt-Richtungs-Modus (Wert = "0") auf den Rechts-/Linkslauf-Modus (Wert = "1") umschalten.

Objektbeschreibung

Index	205B _h
Objektname	Clock Direction Or Clockwise/Counter Clockwise Mode
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1504
Änderungshistorie	

205Ch Virtual Encoder Configuration

Funktion

Mit diesem Objekt konfigurieren Sie den virtuellen Encoderausgang. Details finden Sie im Kapitel Virtueller Encoderausgang.

Objektbeschreibung

Index	205C _h
Objektname	Virtual Encoder Configuration
Object Code	ARRAY
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2115-B1016293
Änderungshistorie	Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Number Of Increments To Be Sent When No Sensor In 0x3203 Is Chosen" auf "Number Of Increments To Be Sent When No Sensor In 0x3203 Is Chosen".

Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Output Signal Mode: 0=AB, 1=CkDir, 2=CW/CCW" auf "Output Signal Mode: 0=AB, 1=CkDir, 2=CW/CCW".

Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Numerator For Conversion Of Sensor Increments To Virtual Encoder Increments" auf "Numerator For Conversion Of Sensor Increments To Virtual Encoder Increments".

Firmware Version FIR-v2115-B1016293: Eintrag "Name" geändert von "Denominator For Conversion Of Sensor Increments To Virtual Encoder Increments" auf "Denominator For Conversion Of Sensor Increments To Virtual Encoder Increments".

Wertebeschreibung

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

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

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

Subindex	03 _h
Name	Numerator For Conversion Of Sensor Increments To Virtual Encoder Increments
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	

Vorgabewert	0000 _h
Subindex	04 _h
Name	Denominator For Conversion Of Sensor Increments To Virtual Encoder Increments
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 _h

2084h Bootup Delay

Funktion

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

Objektbeschreibung

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

2101h Fieldbus Module Availability

Funktion

Zeigt die verfügbaren Feldbusse an.

Objektbeschreibung

Index	2101 _h
Objektnamen	Fieldbus Module Availability
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00190003 _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.

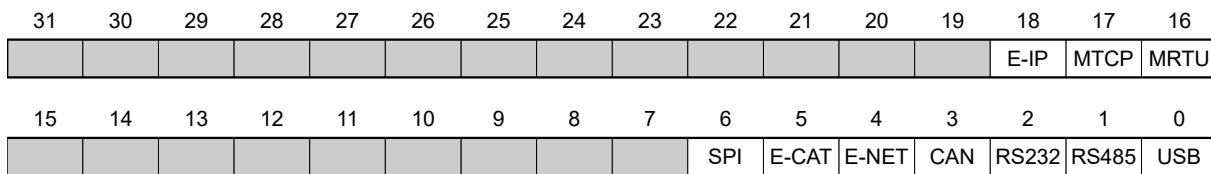
Objektbeschreibung

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	00190003 _h
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Kommunikation".

Beschreibung

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

Dabei gilt die folgende Verteilung der Bits:



USB

USB Schnittstelle

RS-485

RS-485 Schnittstelle

RS-232

RS-232 Schnittstelle

CAN

CANopen Schnittstelle

E-NET

EtherNET Schnittstelle

E-CAT

EtherCAT Schnittstelle

SPI

SPI Schnittstelle

MRTU

Modbus RTU Protokoll

MTCP

Modbus TCP Protokoll

E-IP

EtherNet/IP™ Protokoll

2103h Fieldbus Module Status**Funktion**

Zeigt die aktiven Feldbusse an.

Objektbeschreibung

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

Wertebeschreibung

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

Subindex	01 _h
Name	Fieldbus Module Disable Mask
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000002 _h

Subindex	02 _h
Name	Fieldbus Module Enabled
Datentyp	UNSIGNED32

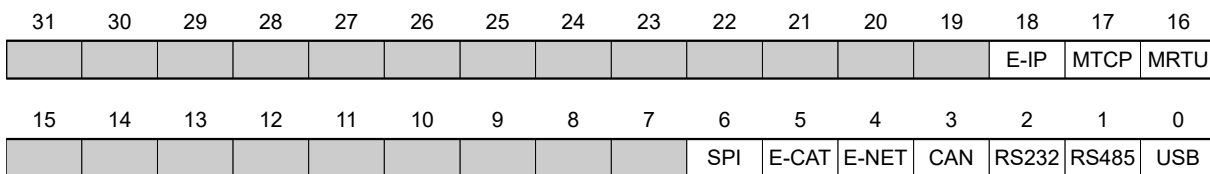
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00190003 _h

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 Fieldbus 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 Fieldbus aktiv ist.

Für Subindex 1 und 2 gilt folgende Verteilung der Bits:



USB

USB Schnittstelle

RS-485

RS-485 Schnittstelle

RS-232

RS-232 Schnittstelle

CAN

CANopen Schnittstelle

E-NET

EtherNET Schnittstelle

E-CAT

EtherCAT Schnittstelle

SPI

SPI Schnittstelle

MRTU

Modbus RTU Protokoll

MTCP

Modbus TCP Protokoll

E-IP

EtherNet/IP™ Protokoll

2104h Additional Fieldbus Configuration

Funktion

In diesem Objekt stellen Sie fest, wie die Kommunikationsparameter eingestellt werden.

Siehe auch [Konfiguration über Modbus RTU](#).

Objektbeschreibung

Index	2104 _h
Objektname	Additional Fieldbus Configuration
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2139-B1019507
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Fieldbus Configuration Source
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	02 _h
Name	Configuration Switch Type
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	1
Vorgabewert	01 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h (Fieldbus Configuration Source):
 - Wert 0: Die Parameter werden über die entsprechenden Objekte eingestellt. Siehe Konfiguration über Modbus RTU.
 - Wert 1: Die Steuerung prüft, ob Drehschalter an den dafür vorgesehenen Pins angeschlossen sind (siehe Anschließen der Steuerung), Sind diese nicht angeschlossen, werden die Parameter über die Objekte eingestellt.
- 02_h (Configuration Switch Type):
 - Wert 1: Es werden zwei Drehschalter verwendet, wie im Kapitel Anschließen der Steuerung beschrieben.

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".

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	

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

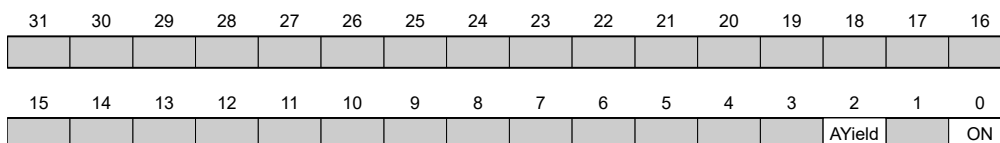
Funktion

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



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-Millisekunde-Takt (siehe Verfügbare Rechenzeit).



HINWEIS

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

2301h NanoJ Status

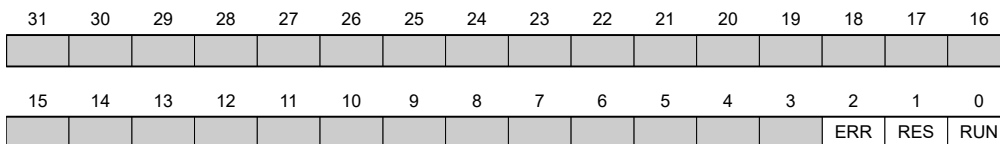
Funktion

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



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. <code>sin</code> , <code>cosin</code> etc.)
0005 _h	Time Out: Code wird zu lange ohne <code>yield()</code> oder <code>sleep()</code> 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	Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs <code>input</code> deklariert (siehe 2310h NanoJ Input Data Selection)
3000000 _h	Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs <code>output</code> deklariert (siehe 2320h NanoJ Output Data Selection)
4000000 _h	Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs <code>inout</code> 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

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

Wertebeschreibung

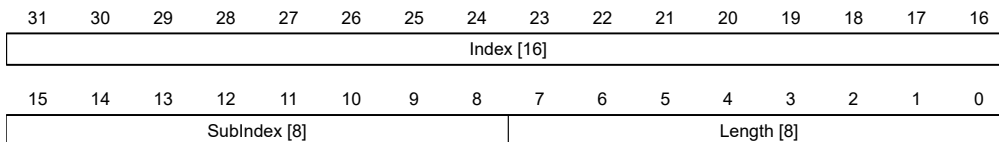
Subindex	00 _h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 _h

Subindex	01 _h - 10 _h
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-16) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammen setzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

2320h NanoJ Output Data Selection

Funktion

Beschreibt die Object Dictionary-Einträge, die in das Output PDO-Mapping des *NanoJ-Programms* kopiert werden, nachdem es ausgeführt worden ist.

Objektbeschreibung

Index	2320 _h
Objektname	NanoJ Output Data Selection
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1650-B472161
Änderungshistorie	<p>Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Output Data Selection" auf "NanoJ Output Data Selection".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p>

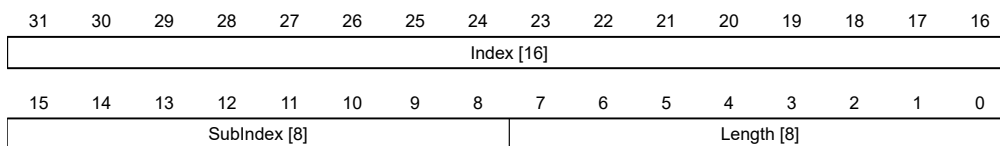
Wertebeschreibung

Subindex	00 _h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 _h
Subindex	01 _h - 10 _h
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-16) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping Eintrag besteht aus vier Byte welche sich nach nachfolgender Grafik zusammen setzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

2330h NanoJ In/output Data Selection

Funktion

Beschreibt die Object Dictionary-Einträge, die zunächst in das Input PDO-Mapping des NanoJ-Programms kopiert und nach dessen Ausführung wieder in das Output PDO-Mapping zurückkopiert werden.

Objektbeschreibung

Index	2330 _h
Objektname	NanoJ In/output Data Selection
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1650-B472161
Änderungshistorie	<p>Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM In/output Data Selection" auf "NanoJ In/output Data Selection".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p>

Wertebeschreibung

Subindex	00 _h
----------	-----------------

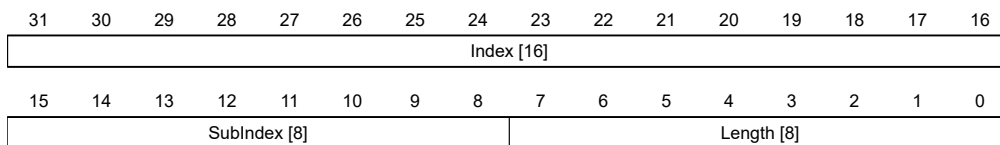
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 _h

Subindex	01 _h - 10 _h
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-16) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping-Eintrag besteht aus vier Bytes, die sich nach folgender Grafik zusammen setzen.



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

Subindex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

2400h NanoJ Inputs

Funktion

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

Objektbeschreibung

Index	2400 _h
Objektname	NanoJ Inputs
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 33

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

Beschreibung

Die Subindizes haben folgende Funktion:

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

3202h Motor Drive Submode Select

Funktion

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

Objektbeschreibung

Index	3202 _h
Objektname	Motor Drive Submode Select
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Fahrt".</p> <p>Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Fahrt" auf "ja, Kategorie: Bewegung".</p>

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	AutoAI	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)

Wert = "1": Stromabsenkung im *Open Loop* aktiviert

AutoAI (Auto Alignment)

Für den Fall, dass ein Betrieb im *Closed Loop* ist gefordert (Bit 0 in 3202_h ist gesetzt).

Wert = "1": das *Auto-Alignment*-Verfahren wird aktiviert; direkt nach dem Einschalten wird im *Open Loop* ein Alignment ermittelt und es wird gleich in die Betriebsart *Closed Loop* gewechselt, ohne dass der Encoder-Index gesehen wurde.

Der Rotor wird dabei ein wenig bewegt.

Wert = "0": kein *Auto-Alignment*, der Motor fährt im *Open Loop*, bis der Encoder-Index gesehen wird (maximal eine Umdrehung der Motorwelle).

Verfügt der zum Kommutieren verwendete inkrementale Encoder über keinen Index (Bit 0 in 33A0_h ist "0"), wird immer ein *Auto-Alignment* ermittelt.

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	05 _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
<hr/>	
Subindex	03 _h
Name	3rd Feedback Interface
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	04 _h
Name	4th Feedback Interface
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	05 _h
Name	5th 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 Rückführung 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	05 _h

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

Subindex	02 _h
Name	Index Of 2nd Feedback Interface

Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	3390 _h

Subindex	03 _h
Name	Index Of 3rd Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	33A0 _h

Subindex	04 _h
Name	Index Of 4th Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	33B0 _h

Subindex	05 _h
Name	Index Of 5th Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	33B1 _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

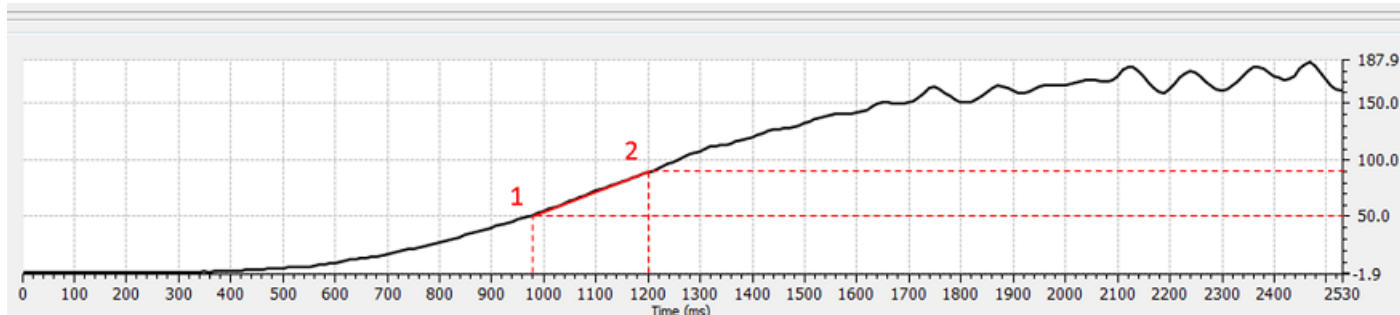
Subindex	02 _h
Name	Acceleration
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

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:
 $(90-50)/(1200-980)=182 \text{ U/min pro Sekunde}$.



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	<p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "S_P" auf "Position Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "S_I" auf "Position Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "V_P" auf "Velocity Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "V_I" auf "Velocity Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Id_P" auf "Flux Current Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Id_I" auf "Flux Current Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Iq_P" auf "Torque Current Loop, Proportional Gain (closed Loop)".</p>

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
<hr/>	
Subindex	03 _h
Name	Velocity Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00002EE0 _h
<hr/>	
Subindex	04 _h
Name	Velocity Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000001E _h
<hr/>	
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
<hr/>	
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
<hr/>	
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-Anteil 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
<hr/>	
Subindex	01 _h
Name	Reserved
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	02 _h
Name	Override Field Inversion
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	03 _h
Name	Auto-setup With Current Controller Parameters From The OD
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
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 I_q (Subindex 01_h/02_h) und I_d (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 K _p For I _q [mV/A]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000027E4 _h

Subindex	02 _h
Name	Integrator Time T _i For I _q [μs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000446 _h

Subindex	03 _h
Name	Proportional Gain K _p For I _d [mV/A]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000027E4 _h

Subindex	04 _h
Name	Integrator Time T _i For I _d [μs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000446 _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 K _p [mA/Hz]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000180 _h

Subindex	02 _h
Name	Integrator Time T _i [μs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000186A0 _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 K _p [Hz]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000032 _h

Subindex	02 _h
Name	Integrator Time T _i [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	02 _h

Subindex	01 _h - 02 _h
Name	Analog Input #1 - #2
Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Beschreibung

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

- Spannungseingang: $x \text{ Digits} * 3,3 \text{ V} / 1023 \text{ Digits}$
- Stromeingang (falls konfigurierbar): $x \text{ Digits} * 20 \text{ mA} / 1023 \text{ Digits}$

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).

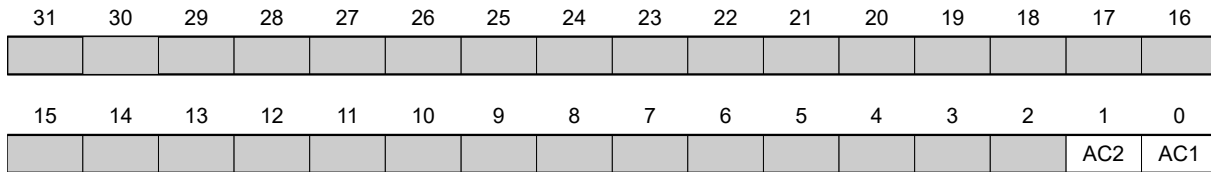
Siehe Kapitel [Analog Input Control](#)

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



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

3225h Analog Input Switches

Funktion

Dieses Objekt enthält den Wert des Drehschalters, der für die Konfiguration der Adresse der Steuerung verwendet wird. Die Schalter-Position wird nur beim Neustart einmalig ausgelesen.

Objektbeschreibung

Index	3225 _h
Objektname	Analog Input Switches
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1436
Änderungshistorie	Firmware Version FIR-v1436: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "RX-PDO" auf "TX-PDO".

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	Analog Input Switch #1
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Beschreibung

Verfügt die Steuerung über einen Drehschalter, wird im Subindex 01_h der Wert des Drehschalters angezeigt. Verfügt die Steuerung über 2 Drehschalter, wird in dem Subindex 01_h der Wert des Drehschalters angezeigt, der sich aus Schalter 1 und 2 zusammensetzt. Siehe Kapitel [Schalter für Kommunikationseinstellungen anschließen](#).

3240h Digital Inputs Control

Funktion

Mit diesem Objekt lassen sich digitale Eingänge manipulieren wie in Kapitel [Digitale Ein- und Ausgänge](#) beschrieben .

Objektbeschreibung

Index	3240 _h
Objektname	Digital Inputs Control
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1426: Subindex 01 _h : Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable" Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 8 auf 9.

Wertebeschreibung

Subindex	00 _h
Name	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	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.
- **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) um, falls der Eingang diese Funktion unterstützt.
- 3240_h:08_h (Routing Enable): Der Wert "1" in diesem Subindex aktiviert das Input Routing.

3241h Digital Input Capture

Funktion

Mit diesem Objekt kann automatisch die Encoderposition notiert werden, wenn am digitalen Eingang ein Pegelwechsel stattfindet. Siehe Kapitel Capture Input.

Objektbeschreibung

Index	3241 _h
Objektname	Digital Input Capture
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1446
Änderungshistorie	<p>Firmware Version FIR-v1446: Eintrag "Data type" geändert von "UNSIGNED32" auf "UNSIGNED8".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Name" geändert von "Encoder Raw Value" auf "Sensor Raw Value".</p> <p>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "TX-PDO".</p> <p>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "RX-PDO" auf "TX-PDO".</p> <p>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "RX-PDO" auf "TX-PDO".</p> <p>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 03 geändert von "RX-PDO" auf "TX-PDO".</p> <p>Firmware Version FIR-v1748-B531667: Tabellen-Eintrag "PDO-Mapping" bei Subindex 04 geändert von "RX-PDO" auf "TX-PDO".</p>

Wertebeschreibung

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

Vorgabewert	08 _h
Subindex	01 _h
Name	Control For Capture Input 1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Capture Count For Capture Input 1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	Value For Capture Input 1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	Sensor Raw Value For Capture Input 1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	Control For Capture Input 2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h

Name	Capture Count For Capture Input 2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	07 _h
Name	Value For Capture Input 2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	Sensor Raw Value For Capture Input 2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	TX-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

3242h Digital Input Routing

Funktion

Dieses Objekt bestimmt die Quelle des Inputroutings, die im 60FD_h endet.

Objektbeschreibung

Index	3242 _h
Objektname	Digital Input Routing
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Applikation

Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1504
Änderungshistorie	

Wertebeschreibung

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

Subindex	01 _h
Name	Input Source For Bit #1 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h

Subindex	02 _h
Name	Input Source For Bit #2 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h

Subindex	03 _h
Name	Input Source For Bit #3 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h

Subindex	04 _h
Name	Input Source For Bit #4 In 60FDh

Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	05 _h
Name	Input Source For Bit #5 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	06 _h
Name	Input Source For Bit #6 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	07 _h
Name	Input Source For Bit #7 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	08 _h
Name	Input Source For Bit #8 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	09 _h
Name	Input Source For Bit #9 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	0A _h
Name	Input Source For Bit #10 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	0B _h
Name	Input Source For Bit #11 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	0C _h
Name	Input Source For Bit #12 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	0D _h
Name	Input Source For Bit #13 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	0E _h
Name	Input Source For Bit #14 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	

Subindex	0F _h
Name	Input Source For Bit #15 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Subindex	10 _h
Name	Input Source For Bit #16 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Subindex	11 _h
Name	Input Source For Bit #17 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0E _h
Subindex	12 _h
Name	Input Source For Bit #18 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	10 _h
Subindex	13 _h
Name	Input Source For Bit #19 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0F _h
Subindex	14 _h
Name	Input Source For Bit #20 In 60FDh
Datentyp	UNSIGNED8

Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	12 _h
<hr/>	
Subindex	15 _h
Name	Input Source For Bit #21 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	11 _h
<hr/>	
Subindex	16 _h
Name	Input Source For Bit #22 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	14 _h
<hr/>	
Subindex	17 _h
Name	Input Source For Bit #23 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	13 _h
<hr/>	
Subindex	18 _h
Name	Input Source For Bit #24 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	16 _h
<hr/>	
Subindex	19 _h
Name	Input Source For Bit #25 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	

Vorgabewert	15 _h
Subindex	1A _h
Name	Input Source For Bit #26 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	17 _h
Subindex	1B _h
Name	Input Source For Bit #27 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	04 _h
Subindex	1C _h
Name	Input Source For Bit #28 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	18 _h
Subindex	1D _h
Name	Input Source For Bit #29 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Subindex	1E _h
Name	Input Source For Bit #30 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Subindex	1F _h

Name	Input Source For Bit #31 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h

Subindex	20 _h
Name	Input Source For Bit #32 In 60FDh
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h

3243h Digital Input Homing 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	Digital Input Homing Capture
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	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	<p>Firmware Version FIR-v1426: Subindex 01_h: Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable"</p> <p>Firmware Version FIR-v1446: Eintrag "Name" geändert von "Special Function Enable" auf "No Function".</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 6 auf 9.</p> <p>Firmware Version FIR-v2039: Subindex 09 hinzugefügt</p>

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
<hr/>	
Subindex	07 _h
Name	Reserved2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	Routing Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
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 Ausgangs wird in Subindex 4_h festgelegt.
- 04_h: Mit diesem Subindex wird der am Ausgang anzulegende Pegel definiert. Der Wert "0" liefert am digitalen Ausgang einen logischen Low-Pegel, der Wert "1" entsprechend einen logischen High-Pegel.
- 05_h: In 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_h:04_h haben dann **keine** Funktion mehr, bis das *Output Routing* wieder abgeschaltet wird.

- 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.

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	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	18 _h

Subindex	01 _h
Name	Control Bit Of 60FE _h :1h And Source For Output #1
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h

Subindex	02 _h
Name	Control Bit Of 60FE _h :1h And Source For Output #2
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h

Subindex	03 _h
Name	Control Bit Of 60FEh:1h And Source For Output #3
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h
Subindex	04 _h
Name	Control Bit Of 60FEh:1h And Source For Output #4
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	001A _h
Subindex	05 _h
Name	Control Bit Of 60FEh:1h And Source For Output #5
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h
Subindex	06 _h
Name	Control Bit Of 60FEh:1h And Source For Output #6
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	001C _h
Subindex	07 _h
Name	Control Bit Of 60FEh:1h And Source For Output #7
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h
Subindex	08 _h
Name	Control Bit Of 60FEh:1h And Source For Output #8
Datentyp	UNSIGNED16

Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h

Subindex	09 _h
Name	Control Bit Of 60FEh:1h And Source For Output #9
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h

Subindex	0A _h
Name	Control Bit Of 60FEh:1h And Source For Output #10
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h

Subindex	0B _h
Name	Control Bit Of 60FEh:1h And Source For Output #11
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h

Subindex	0C _h
Name	Control Bit Of 60FEh:1h And Source For Output #12
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	FFFF _h

Subindex	0D _h
Name	Control Bit Of 60FEh:1h And Source For Output #13
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	

Vorgabewert	001D _h
Subindex	0E _h
Name	Control Bit Of 60FEh:1h And Source For Output #14
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0010 _h
Subindex	0F _h
Name	Control Bit Of 60FEh:1h And Source For Output #15
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0012 _h
Subindex	10 _h
Name	Control Bit Of 60FEh:1h And Source For Output #16
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0011 _h
Subindex	11 _h
Name	Control Bit Of 60FEh:1h And Source For Output #17
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0014 _h
Subindex	12 _h
Name	Control Bit Of 60FEh:1h And Source For Output #18
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0013 _h
Subindex	13 _h

Name	Control Bit Of 60FEh:1h And Source For Output #19
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0016 _h
<hr/>	
Subindex	14 _h
Name	Control Bit Of 60FEh:1h And Source For Output #20
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0015 _h
<hr/>	
Subindex	15 _h
Name	Control Bit Of 60FEh:1h And Source For Output #21
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0018 _h
<hr/>	
Subindex	16 _h
Name	Control Bit Of 60FEh:1h And Source For Output #22
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	1080 _h
<hr/>	
Subindex	17 _h
Name	Control Bit Of 60FEh:1h And Source For Output #23
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0019 _h
<hr/>	
Subindex	18 _h
Name	Control Bit Of 60FEh:1h And Source For Output #24
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben

PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	001B _h

3260h Pwm Output 0

Funktion

Mit diesem Objekt konfigurieren Sie den ersten PWM-Ausgang. Den Ausgang müssen Sie als PWM-Ausgang mittels *Output Routing* festlegen.

Objektbeschreibung

Index	3260 _h
Objektname	Pwm Output 0
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1939-B682906
Änderungshistorie	

Wertebeschreibung

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

Subindex	01 _h
Name	Pwm Frequency
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00002710 _h

Subindex	02 _h
Name	Pwm Duty Cycle
Datentyp	UNSIGNED32

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

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Frequenz des PWM-Signals in Hertz. 50...20000
- 02_h: Tastgrad (Duty-Cycle) des PWM-Signals: 0...100

3261h Pwm Output 1

Funktion

Mit diesem Objekt konfigurieren Sie den zweiten PWM-Ausgang. Den Ausgang müssen Sie als PWM-Ausgang mittels *Output Routing* festlegen.

Objektbeschreibung

Index	3261 _h
Objektname	Pwm Output 1
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1939-B682906
Änderungshistorie	

Wertebeschreibung

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

Subindex	01 _h
Name	Pwm Frequency
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00002710 _h
<hr/>	
Subindex	02 _h
Name	Pwm Duty Cycle
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h: Frequenz des PWM-Signals in Hertz. 50...20000
- 02_h: Tastgrad (Duty-Cycle) des PWM-Signals: 0...100

3273h Generic SPI Hardware Configuration

Funktion

Siehe Kapitel [Generic SPI](#).

Objektbeschreibung

Index	3273 _h
Objektname	Generic SPI Hardware Configuration
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2213-B1029645
Änderungshistorie	

Wertebeschreibung

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

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

3274h Generic SPI Mosi Data

Funktion

Siehe Kapitel [Generic SPI](#).

Objektbeschreibung

Index	3274 _h
Objektname	Generic SPI Mosi Data
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2213-B1029645
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Length Of SPI Message To Be Sent
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h - 1F _h
Name	Generic SPI Mosi Data Byte #1 - #31
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

3275h Generic SPI Miso Data

Funktion

Siehe Kapitel [Generic SPI](#).

Objektbeschreibung

Index	3275 _h
Objektname	Generic SPI Miso Data
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2213-B1029645
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Length Of Received SPI Message
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h - 1F _h
Name	Generic SPI Miso Data Byte #1 - #31
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

3320h Analog Input Values

Funktion

Dieses Objekt zeigt die Momentanwerte der Analogeingänge in benutzerdefinierten Einheiten an.

Objektbeschreibung

Index	3320 _h
-------	-------------------

Objektname	Analog Input Values
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Analog Input Values
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h - 02 _h
Name	Analog Input #1 - #2
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

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

Formel zum Umrechnen von Digits in die jeweilige Einheit:

- Spannungseingang: $x \text{ Digits} * 3,3 \text{ V} / 1023 \text{ Digits}$
- Stromeingang (falls konfigurierbar): $x \text{ Digits} * 20 \text{ mA} / 1023 \text{ Digits}$

Für die Subeinträge gilt:

- Subindex 00_h: Anzahl der Analogeingänge
- Subindex 01_h: Analogwert 1
- Subindex 02_h: Analogwert 2 (falls vorhanden)

3321h Analog Input Offsets

Funktion

Offset, der zum eingelesenen Analogwert (3220_h) addiert wird, bevor die Skalierung (Multiplikator aus dem Objekt 3322 und Teiler aus dem Objekt 3323_h) vorgenommen wird.

Objektbeschreibung

Index	3321 _h
Objektname	Analog Input Offsets
Object Code	ARRAY
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2139-B1022383
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Analog Input Offsets
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h - 02 _h
Name	Analog Input #1 - #2
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h

3322h Analog Input Numerators

Funktion

Wert, mit dem der eingelesene Analogwert (3220_h + 3321_h) multipliziert wird, bevor er in das Objekt 3320_h geschrieben wird.

Objektbeschreibung

Index	3322 _h
Objektname	Analog Input Numerators
Object Code	ARRAY
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Analog Input Numerators
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h - 02 _h
Name	Analog Input #1 - #2
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	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 (3320_h+ 3321_h) dividiert wird, bevor er in das Objekt 3320_h geschrieben wird.

Objektbeschreibung

Index	3323 _h
Objektname	Analog Input Denominators
Object Code	ARRAY
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1926-B648637
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Analog Input Denominators
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h - 02 _h
Name	Analog Input #1 - #2
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	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: Einschaldrehzahl in U/min. Ab dieser Drehzahl wird der *Closed Loop* (*Sensorless*) aktiviert, wenn vom Auto-Setup keine Sensoren erkannt wurden.
- 05_h: Ausschaldrehzahl in U/min. Unter dieser Drehzahl wird der *Closed Loop* (*Sensorless*) deaktiviert, wenn vom Auto-Setup keine Sensoren erkannt wurden.

3390h Feedback Hall

Funktion

Enthält Konfigurationswerte für die Hall-Sensoren. Die Werte werden vom Auto-Setup ermittelt.

Objektbeschreibung

Index	3390 _h
Objektname	Feedback Hall
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	

Vorgabewert
 Firmware Version FIR-v1748-B531667
 Änderungshistorie

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	1st Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	02 _h
Name	2nd Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	03 _h
Name	3rd Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	04 _h
Name	4th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	05 _h
Name	5th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	06 _h
Name	6th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	07 _h
Name	7th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	08 _h
Name	8th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	09 _h
Name	9th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	

Subindex	0A _h
Name	10th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	0B _h
Name	11th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	0C _h
Name	12th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

33A0h Feedback Incremental A/B/I 1

Funktion

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

Objektbeschreibung

Index	33A0 _h
Objektname	Feedback Incremental A/B/I 1
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	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.
- 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	0C _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
<hr/>	
Subindex	0A _h
Name	Status Bitmask High
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0B _h
Name	Status Value Low
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0C _h
Name	Status Value High
Datentyp	UNSIGNED32
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 der Encoder-Daten wird von der Steuerung als Fehler interpretiert.

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
L								POS	POS	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	E	S	S	S

Sie müssen in die Subindizes folgende Werte eintragen:

- 05_h (Number Of Bits For Transfer): 20_h
- 07_h(Position Bitmask Low) 00FFFFFF_h
- 09_h (Status Bitmask Low): 8000 000F_h
- 0B_h (Status Value Low): 8000 000F_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".

33B1h Feedback SSI 2

Funktion

Enthält Konfigurationswerte für den zweiten externen SSI-Encoder.

Objektbeschreibung

Index	33B1 _h
Objektname	Feedback SSI 2
Object Code	RECORD
Datentyp	SSI ENCODER
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2139-B1019507
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0C _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
<hr/>	
Subindex	04 _h
Name	Home Position High
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05 _h
Name	Number Of Bits For Transfer
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	15 _h
<hr/>	
Subindex	06 _h
Name	Baud Rate
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00280DE8 _h
<hr/>	
Subindex	07 _h
Name	Position Bitmask Low
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	001FFFFE _h
<hr/>	
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

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 der Encoder-Daten wird von der Steuerung als Fehler interpretiert.

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
L								POS	POS	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	E	S	S	S

Sie müssen in die Subindizes folgende Werte eintragen:

- 05_h (Number Of Bits For Transfer): 20_h
- 07_h(Position Bitmask Low) 00FFFFFF_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".

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
<hr/>	
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60600008 _h
<hr/>	
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	607A0020 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60810020 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60420010 _h
<hr/>	
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
<hr/>	
Subindex	0E _h
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	0F _h
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	10 _h
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

3602h MODBUS Tx PDO Mapping

Funktion

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

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
<hr/>	
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60640020 _h
<hr/>	
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60440010 _h
<hr/>	
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60FD0020 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

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 <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit <i>quick stop ramp</i> (6085 _h)
3 bis 32767	reserviert

3701h Limit Switch Error Option Code

Funktion

Wird ein Endschalter überfahren, wird das Bit 7 (*Warning*) in 6041_h (*Statusword*) gesetzt und die in diesem Objekt hinterlegte Aktion ausgeführt. 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 <i>Switch on disabled</i>)
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt

Wert in Objekt 3701 _h	Beschreibung
6	bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten. Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

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	<p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 02 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Temperature PCB [d?C]" auf "Temperature PCB [Celsius * 10]".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 03 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 4 auf 6.</p>

Wertebeschreibung

Subindex	00 _h
Name	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: reserviert

4021h Ballast Configuration

Funktion

Mit diesem Objekt schalten Sie die Ballast-Schaltung ein oder aus und bestimmen Sie deren Ansprechschwelle. Ferner konfigurieren Sie die Ballast-Überwachung. Details finden Sie im Kapitel [Externe Ballast-Schaltung](#).

Objektbeschreibung

Index	4021 _h
Objektname	Ballast Configuration
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: 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	08 _h

Subindex	01 _h
Name	Settings [Bit0: On/Off, Bit1: Polarity]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	UB Power Limit [mV]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	<ul style="list-style-type: none"> ■ CPB6-1-5: 0000EC54_h ■ CPB6-2-5: 00011172_h ■ CPB3-1-5: 0000EC54_h ■ CPB3-2-5: 0000EC54_h ■ CPB15-5: 0000EC54_h ■ CPB15-S-5: 0000EC54_h
-------------	--

Subindex	03 _h
Name	UB Power Hysteresis [mV]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000001F4 _h

Subindex	04 _h
Name	Nominal Resistance [mOhm]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	05 _h
Name	Long Term Energy Limit [mWs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	06 _h
Name	Long Term Reference Time [ms]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	07 _h
Name	Short Term Energy Limit [mWs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	08 _h
Name	Cooling Power [mW]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 01_h:
 - Bit 0: schaltet den Ballast ein (Wert = "1") oder aus (Wert = "0")
 - Bit 1: invertiert (Wert = "1") die Polarität des Pins zur Ansteuerung der externen Ballast-Schaltung (im Auslieferungszustand *active high*)
- 02_h: Ansprechschwelle (Ein-/Ausschalten) der Ballast-Schaltung
- 03_h: Hysterese für die Ansprechschwelle (Ein-/Ausschalten)
- 04_h: Nennwert des Ballast-Widerstands
- 05_h: Energiemenge, die dem Widerstand innerhalb der *Long Term Reference Time* (Subindex 06_h) zugeführt werden darf, ohne ihn zu überlasten.
- 06_h: die Bezugszeit für das *Long Term Energy Limit* (Subindex 05_h) (typischerweise zwischen 1 und 5 Sekunden)
- 07_h: Energiemenge, die dem Widerstand innerhalb kurzer Lastpulse (<1 Sekunde) zugeführt werden darf, ohne ihn zu überlasten.
- 08_h: Die Wärmemenge, die der Widerstand durch Konvektion kontinuierlich an seine Umgebung abgeben kann bzw. darf.

4040h Drive Serial Number

Funktion

Dieses Objekt hält die Seriennummer der Steuerung.

Objektbeschreibung

Index	4040 _h
Objektname	Drive Serial Number
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1450

Änderungshistorie

4041h Device Id

Funktion

Dieses Objekt hält die ID des Geräts.

Objektbeschreibung

Index	4041 _h
Objektname	Device Id
Object Code	VARIABLE
Datentyp	OCTET_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1540
Änderungshistorie	

4042h Bootloader Infos

Objektbeschreibung

Index	4042 _h
Objektname	Bootloader Infos
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v2013-B726332
Änderungshistorie	

Wertebeschreibung

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

Vorgabewert	03 _h
Subindex	01 _h
Name	Bootloader Version
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Bootloader Supported Fieldbus
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	Bootloader Hw-group
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Version des Bootloaders. Die 4 höchstwertigen Bytes erhalten die Hauptversionsnummer, die 4 niedrigwertigsten Bytes die Nebenversionsnummer. Beispiel für die Version 4.2: 00040002_h
- 02_h: Vom Bootloader unterstützte 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")

TARG

Zielvorgabe erreicht

ILA (Internal Limit Active)

Limit überschritten

OMS (Operation Mode Specific)

Bedeutung abhängig vom gewählten Betriebsmodus

CLA (Closed Loop Active)

Wert = "1": die Steuerung befindet sich im Status *Operation enabled* und der Closed Loop ist aktiviert.

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

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

6042h VI Target Velocity

Funktion

Gibt die Zielgeschwindigkeit für den Velocity Modus in benutzerdefinierten Einheiten an.

Objektbeschreibung

Index	6042 _h
Objektname	VI Target Velocity
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00C8 _h
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

6043h VI Velocity Demand

Funktion

Vorgabegeschwindigkeit in benutzerdefinierten Einheiten für den Regler im Velocity Mode.

Objektbeschreibung

Index	6043 _h
Objektname	VI Velocity Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6044h VI Velocity Actual Value

Funktion

Gibt die aktuelle Istgeschwindigkeit im Velocity Modus in benutzerdefinierten Einheiten an.

Objektbeschreibung

Index	6044 _h
Objektname	VI Velocity Actual Value
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

6046h VI Velocity Min Max Amount

Funktion

Mit diesem Objekt können Minimalgeschwindigkeit und Maximalgeschwindigkeit in benutzerdefinierten Einheiten eingestellt werden.

Objektbeschreibung

Index	6046 _h
Objektname	VI Velocity Min Max Amount
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	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
<hr/>	
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 6041_h Statusword_h wird gesetzt.

Eine Zielgeschwindigkeit größer als die Maximalgeschwindigkeit setzt die Geschwindigkeit auf die Maximalgeschwindigkeit und das Bit 11 (Internal Limit Reached) in 6041_h Statusword_h wird gesetzt.

6048h VI Velocity Acceleration

Funktion

Setzt die Beschleunigungsrampe im Velocity Mode (siehe [Velocity](#)).

Objektbeschreibung

Index	6048 _h
Objektname	VI Velocity Acceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	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 VI Velocity Deceleration

Funktion

Setzt die Verzögerung (Bremsrampe) im Velocity Mode (siehe [Velocity](#)).

Objektbeschreibung

Index	6049 _h
Objektname	VI Velocity Deceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	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 VI Velocity Quick Stop

Funktion

Dieses Objekt definiert die Verzögerung (Bremsrampe), wenn im Velocity Mode der Quick Stop-Zustand eingeleitet wird.

Objektbeschreibung

Index	604A _h
Objektname	VI Velocity Quick Stop
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	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	VI Dimension Factor Numerator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	02 _h
Name	VI Dimension Factor Denominator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Der Subindex 1 enthält den Zähler (Multiplikator) und der Subindex 2 den Nenner (Divisor), mit dem interne Geschwindigkeitsangaben in Umdrehungen pro Minute verrechnet werden. Wird z.B. Subindex 1 auf den Wert "60" und Subindex 2 auf den Wert "1" eingestellt, erfolgt die Geschwindigkeitsangabe in Umdrehungen pro Sekunde (60 Umdrehungen pro 1 Minute).

605Ah Quick Stop Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der CiA 402 Power State Machine in den Zustand *Quick Stop active*.

Objektbeschreibung

Index	605A _h
Objektname	Quick Stop Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0002 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert in Objekt 605A _h	Beschreibung
0	Soforthalt mit anschließendem Zustandswechsel in <i>Switch on disabled</i>
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> (6085 _h) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.
6	Abbremsen mit <i>quick stop ramp</i> (6085 _h) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

605Bh Shutdown Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der CiA 402 Power State Machine vom Zustand *Operation enabled* in den Zustand *Ready to switch on*.

Objektbeschreibung

Index	605B _h
Objektname	Shutdown Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Wert in Objekt 605B _h	Beschreibung
-32768 bis -1	Reserviert
0	Sperren der Antriebsfunktion — Motor kann sich frei drehen
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Ready to switch on</i>
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	Sperren der Antriebsfunktion — Motor kann sich frei drehen
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switched on</i>
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 <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit <i>quick stop ramp</i> (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 <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit <i>quick stop ramp</i> (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
8	Cyclic Synchronous Position Mode
9	Cyclic Synchronous Velocity Mode

Modus	Beschreibung
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 Geschwindigkeitsregler.

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 Nenn Drehmoments.

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 Nenn Drehmoments, "1100" ist äquivalent zu 110%. Das Nenn Drehmoment 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 Nenn Drehmoments.

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 Nenn Drehmoments, "1100" ist äquivalent zu 110%. Das Nenn Drehmoment 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 I_{2t} Motor-Überlastungsschutz,
- den Sollstrom im *Open Loop*-Betrieb.

6074h Torque Demand

Funktion

Momentaner vom Rampengenerator geforderter Drehmomentsollwert in Promille des Nenndrehmoments für den internen Regler.

Objektbeschreibung

Index	6074 _h
Objektname	Torque Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt 203B_h:01.

Das Minimum von 6073_h und 6072_h wird als Limit für das Drehmoment in 6071_h verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031_h) nicht übersteigen.

6075h Motor Rated Current

Funktion

Enthält den in 203B_h:01_h eingetragenen Nennstrom in mA.

6077h Torque Actual Value

Funktion

Dieses Objekt zeigt den aktuellen Drehmomentwert in Promille des Nenndrehmoments für den internen Regler.

Objektbeschreibung

Index	6077 _h
Objektname	Torque Actual Value
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	FIR-v1540
Änderungshistorie	

Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt 203B_h:01.

Das Minimum von 6073_h und 6072_h wird als Limit für das Drehmoment in 6071_h verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031_h) nicht übersteigen.

607Ah Target Position

Funktion

Dieses Objekt gibt die Zielposition in benutzerdefinierten Einheiten für den Profile Position und Cyclic Synchronous Position Modus an.

Objektbeschreibung

Index	607A _h
Objektname	Target Position
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000FA0 _h
Firmware Version	FIR-v1426

Änderungshistorie

Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

607Bh Position Range Limit

Funktion

 Enthält die Minimal- und Maximalposition in benutzerdefinierten Einheiten.

Objektbeschreibung

Index	607B _h
Objektname	Position Range Limit
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	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 `3212h:02h`.

607Fh Max Profile Velocity

Funktion

Gibt die maximale Geschwindigkeit in benutzerdefinierten Einheiten an, für die Mod i [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	<p>Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Max profile velocity" auf "Max Profile Velocity".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "INTEGER16" auf "UNSIGNED32".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".</p>

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	00007530 _h
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Maximum Speed" auf "Max Motor Speed".</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "RX-PDO".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Bewegung".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Speicherbar" geändert von "ja, Kategorie: Bewegung" auf "ja, Kategorie: Tuning".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Bewegung".</p>

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 Nenn Drehmoments, "1100" ist äquivalent zu 110%. Das Nenn Drehmoment 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 [60E6h](#)/[60EBh](#)) 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	<p>Firmware Version FIR-v1738-B501312: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "nein" auf "RX-PDO".</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "RX-PDO".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".</p>

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	<p>Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Datentyp" geändert von "INTEGER32" auf "UNSIGNED32".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED32".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED32".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p> <p>Firmware Version FIR-v1825-B577172: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".</p>

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
Objektnamen	Homing Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h
Firmware Version	FIR-v1426

Änderungshistorie

Beschreibung

Die Rampe wird nur beim Losfahren verwendet. Beim Erreichen des Schalters wird sofort auf die niedrigere Geschwindigkeit umgeschaltet und beim Erreichen der Endposition wird sofort gestoppt.

60A2h Jerk Factor

Funktion

Dieses Objekt beinhaltet den Faktor, der zum Umrechnen von benutzerdefinierten Ruckeinheiten verwendet wird. Siehe Kapitel [Benutzerdefinierte Einheiten](#).

Objektbeschreibung

Index	60A2 _h
Objektname	Jerk Factor
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	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
Objektnamen	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
Objektnamen	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
Objektnamen	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 60C2_h:01_h * 10^{Wert des 60C2:02} Sekunden.

60C4h Interpolation Data Configuration

Funktion

Dieses Objekt bietet die maximale Puffergröße, gibt die konfigurierte Puffer-Organisation der interpolierten Daten an und bietet Objekte zur Definition der Größe des Datensatzes und zum Löschen des Puffers.

Es wird zudem verwendet, um die Position weiterer Datenpunkte zu speichern.

Objektbeschreibung

Index	60C4 _h
Objektname	Interpolation Data Configuration
Object Code	RECORD
Datentyp	INTERPOLATION_DATA_CONFIGURATION
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1512
Änderungshistorie	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	05 _h

Subindex	01 _h - 05 _h
Name	Additional Position Actual Value #1 - #5
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	05 _h

Subindex	01 _h - 05 _h
Name	Additional Velocity Actual Value #1 - #5
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	05 _h

Subindex	01 _h - 05 _h
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #1 - #5
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	05 _h

Subindex	01 _h - 05 _h
Name	Additional Gear Ratio - Motor Shaft Revolutions Feedback Interface #1 - #5
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h: Subindex "n" enthält die Anzahl der Motorumdrehungen für die entsprechende Rückführung. Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Getriebeübersetzung der Rückführung "n" berechnet sich wie folgt:

$$\text{Gear Ratio} = \text{Motor Shaft Revolutions (60E8}_{h:n_h}) / \text{Driving Shaft Revolutions (60ED}_{h:n_h})$$

60E9h Additional Feed Constant - Feed

Funktion

In diesem Objekt und in 60EEh 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	05 _h

Subindex	01 _h - 05 _h
Name	Additional Feed Constant - Feed Feedback Interface #1 - #5
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n_h: Subindex "n" enthält den Vorschub in benutzerdefinierten Einheiten für die entsprechende Rückführung.
 Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Vorschubkonstante der Rückführung "n" berechnet sich wie folgt:

Feed Constant = Feed (60E9_{h:n_h}) / Driving Shaft Revolutions (60EE_{h:n_h})

60EB_h Additional Position Encoder Resolution - Motor Revolutions

Funktion

Mit diesem Objekt und mit 60E6_h wird die Auflösung jeder vorhandenen Rückführung berechnet.

Objektbeschreibung

Index	60EB _h
Objektname	Additional Position Encoder Resolution - Motor Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

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

Subindex	01 _h - 05 _h
Name	Additional Position Encoder Resolution - Motor Revolutions Feedback Interface #1 - #5
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)

60ED_h Additional Gear Ratio - Driving Shaft Revolutions

Funktion

In diesem Objekt und in 60E8_h können Sie die Getriebeübersetzung jeder vorhandenen Rückführung einstellen.

Objektbeschreibung

Index	60ED _h
Objektname	Additional Gear Ratio - Driving Shaft Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

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

Subindex	01 _h - 05 _h
Name	Additional Gear Ratio - Driving Shaft Revolutions Feedback Interface #1 - #5
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Die Subindizes haben folgende Funktion:

- 00_h: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.

- n_h : Subindex "n" enthält die Anzahl der Umdrehungen der Abtriebsachse für die entsprechende Rückführung.
 Subindex 01_h entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Getriebeübersetzung der Rückführung "n" berechnet sich wie folgt:

$$\text{Gear Ratio} = \text{Motor Shaft Revolutions (60E8}_{h:n_h}) / \text{Driving Shaft Revolutions (60ED}_{h:n_h})$$

60EEh Additional Feed Constant - Driving Shaft Revolutions

Funktion

In diesem Objekt und in 60E9_h können Sie eine Vorschubkonstante für jede vorhandene Rückführung einstellen.

Objektbeschreibung

Index	60EE _h
Objektname	Additional Feed Constant - Driving Shaft Revolutions
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

Wertebeschreibung

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

Subindex	01 _h - 05 _h
Name	Additional Feed Constant - Driving Shaft Revolutions Feedback Interface #1 - #5
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:

$$\text{Feed Constant} = \text{Feed} (60E9_h:n_h) / \text{Driving Shaft Revolutions} (60EE_h:n_h)$$

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 Zielposition voran gegangen ist)
0	1	Positionsbewegungen werden relativ zum Vorgabewert (bzw. Ausgang) des Rampengenerators ausgeführt.
1	0	Positionsbewegungen werden relativ zur Istposition (Objekt 6064_h) ausgeführt.

Bit 1	Bit 0	Definition
1	1	Reserviert

RRO (Request-Response Option)

Diese Bits bestimmen das Verhalten bei der Übergabe des Controlwords 6040_h Bit 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 <u>Setzen von Fahrbefehlen</u> 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" <u>607B_h:01_h</u> und <u>02_h</u> 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 <u>607D_h:01_h</u> 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 <u>607D_h:01_h</u> 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

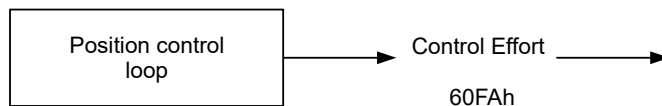
Dieses Objekt beinhaltet die Korrekturgeschwindigkeit (Stellgröße) in benutzerdefinierten Einheiten, die vom Positionsregler dem Geschwindigkeitsregler zugeführt wird.

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 Inkrementen des für die Position gewählten Sensors 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
							...	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	000003EF _h
Firmware Version	FIR-v1426
Änderungshistorie	

Beschreibung

Ein gesetztes Bit gibt an, ob der jeweilige Modus unterstützt wird. Ist der Wert des Bits "0", wird der Modus nicht unterstützt.

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	

12 Copyrights

12.1 Einführung

In der Nanotec Software sind Komponenten aus Produkten externer Software-Hersteller integriert. In diesem Kapitel finden Sie die Copyright-Informationen zu den verwendeten externen Software-Quellen.

12.2 AES

FIPS-197 compliant AES implementation

Based on XySSL: Copyright (C) 2006-2008 Christophe Devine

Copyright (C) 2009 Paul Bakker <polarssl_maintainer at polarssl dot org>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution; or, the application vendor's website must provide a copy of this notice.
- Neither the names of PolarSSL or XySSL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The AES block cipher was designed by Vincent Rijmen and Joan Daemen.

<http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

12.3 MD5

MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

12.4 uIP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12.5 DHCP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12.6 CMSIS DSP Software Library

Copyright (C) 2010 ARM Limited. All rights reserved.

12.7 FatFs

FatFs - FAT file system module include file R0.08 (C)ChaN, 2010

FatFs module is a generic FAT file system module for small embedded systems.

This is a free software that opened for education, research and commercial developments under license policy of following terms.

Copyright (C) 2010, ChaN, all right reserved.

The FatFs module is a free software and there is NO WARRANTY.

No restriction on use. You can use, modify and redistribute it for personal, non-profit or commercial product UNDER YOUR RESPONSIBILITY.

Redistributions of source code must retain the above copyright notice.

12.8 Protothreads

Protothread class and macros for lightweight, stackless threads in C++.

This was "ported" to C++ from Adam Dunkels' protothreads C library at: <http://www.sics.se/~adam/pt/>

Originally ported for use by Hamilton Jet (www.hamiltonjet.co.nz) by Ben Hoyt, but stripped down for public release. See his blog entry about it for more information: <http://blog.micropledge.com/2008/07/protothreads/>

Original BSD-style license

Copyright (c) 2004-2005, Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the Institute and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Institute or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

12.9 lwIP

Copyright (c) 2001-2004 Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO

EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This file is part of the lwIP TCP/IP stack.

Author: Adam Dunkels <adam@sics.se>

12.10 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/or other 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.