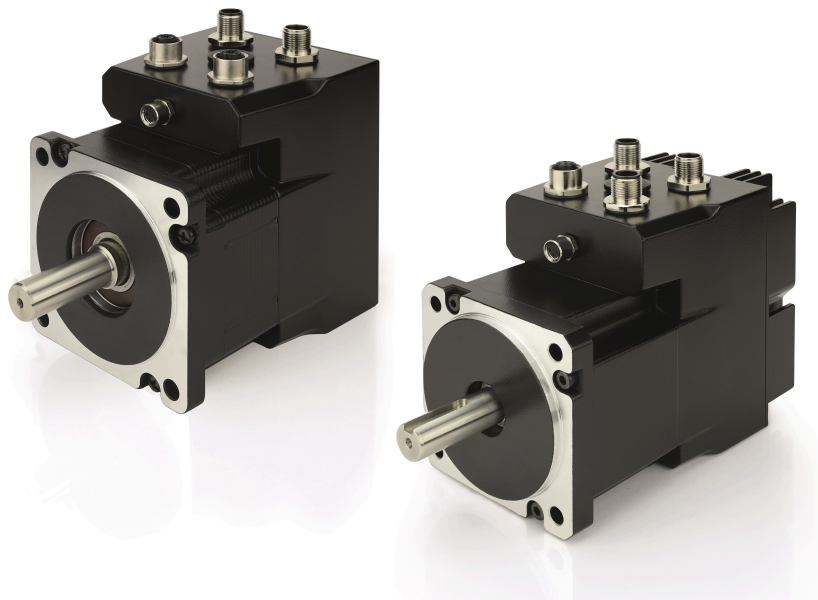


# Technisches Handbuch PD6-E

Feldbus: Modbus RTU

Zu nutzen mit folgenden Varianten:

PD6-E891S95-E-65-5, PD6-E891M95-E-65-5, PD6-E891L95-E-65-5, PD6-EB80SD-E-65-5, PD6-EB80MD-E-65-5, PD6-EB80LD-E-65-5, PD6-EB80CD-E-65-5



# Inhalt

<b>1</b>	<b>Einleitung.....</b>	<b>10</b>
1.1	Versionshinweise.....	10
1.2	Urheberrecht, Kennzeichnung und Kontakt.....	10
1.3	Bestimmungsgemäße Verwendung.....	10
1.4	Gewährleistung und Haftungsausschluss.....	11
1.5	Zielgruppe und Qualifikation.....	11
1.6	EU-Richtlinien zur Produktsicherheit.....	11
1.7	Mitgeltende Vorschriften.....	11
1.8	Verwendete Symbole.....	11
1.9	Hervorhebungen im Text.....	12
1.10	Zahlenwerte.....	12
1.11	Bits.....	12
1.12	Zählrichtung (Pfeile).....	13
<b>2</b>	<b>Sicherheits- und Warnhinweise.....</b>	<b>14</b>
<b>3</b>	<b>Technische Daten und Anschlussbelegung.....</b>	<b>15</b>
3.1	Umgebungsbedingungen.....	15
3.2	Maßzeichnungen.....	15
3.2.1	PD6-E891S95-E-65-5.....	15
3.2.2	PD6-E891M95-E-65-5.....	16
3.2.3	PD6-E891L95-E-65-5.....	16
3.2.4	PD6-EB80SD-E-65-5.....	16
3.2.5	PD6-EB80MD-E-65-5.....	17
3.2.6	PD6-EB80LD-E-65-5.....	17
3.2.7	PD6-EB80CD-E-65-5.....	17
3.3	Elektrische Eigenschaften und technische Daten.....	18
3.3.1	Elektrische Daten.....	18
3.3.2	Technische Daten.....	18
3.4	Übertemperaturschutz.....	19
3.5	LED-Signalisierung.....	23
3.5.1	Betriebs-LED.....	23
3.5.2	Modbus LEDs.....	24
3.6	Anschlussbelegung.....	24
3.6.1	Übersicht.....	24
3.6.2	X1 — Spannungsversorgung.....	24
3.6.3	X2 — Ein-/Ausgänge und Logikversorgung.....	26
3.6.4	X3 — Modbus RTU (RS-485) IN.....	28
3.6.5	X4 — Modbus RTU (RS-485) OUT.....	30
3.6.6	X5 — externe Haltebremse.....	31
3.6.7	X6 — Mini USB.....	32
3.6.8	Schalter S1 - Hex-Codierschalter für Slave-Adresse und Baudrate.....	32
3.6.9	Schalter S2 — 150 Ohm Terminierungswiderstand.....	32
<b>4</b>	<b>Inbetriebnahme.....</b>	<b>33</b>
4.1	Konfiguration über USB.....	34
4.1.1	Allgemeines.....	34
4.1.2	USB-Anschluss.....	34
4.1.3	Konfigurationsdatei.....	35

4.1.4 NanoJ-Programm.....	37
4.2 Konfiguration über Modbus RTU.....	38
4.2.1 Kommunikationseinstellungen.....	39
4.2.2 Kommunikation aufbauen.....	39
4.3 Konfigurieren der Sensoren.....	40
4.4 Auto-Setup.....	41
4.4.1 Parameter-Ermittlung.....	42
4.4.2 Durchführung.....	42
4.4.3 Parameterspeicherung.....	44
4.5 Spezielle Fahrmodi (Takt-Richtung und Analog-Drehzahl).....	44
4.5.1 Aktivierung.....	45
4.5.2 Takt-Richtung.....	45
4.5.3 Analog-Drehzahl.....	45
4.5.4 Automatische Fahrt mit 30 U/min (Testfahrt).....	47
<b>5 Generelle Konzepte.....</b>	<b>48</b>
5.1 Betriebsarten.....	48
5.1.1 Allgemein.....	48
5.1.2 Open Loop.....	49
5.1.3 Closed Loop.....	51
5.1.4 Slow Speed.....	57
5.2 CiA 402 Power State Machine.....	58
5.2.1 Zustandsmaschine.....	58
5.2.2 Verhalten beim Verlassen des Zustands <i>Operation enabled</i> .....	60
5.3 Benutzerdefinierte Einheiten.....	63
5.3.1 Einheiten.....	64
5.3.2 Encoderauflösung.....	65
5.3.3 Getriebeübersetzung.....	66
5.3.4 Vorschubkonstante.....	66
5.3.5 Berechnungsformeln für Benutzereinheiten.....	66
5.4 Begrenzung des Bewegungsbereichs.....	68
5.4.1 Verhalten beim Erreichen der Endschalter.....	68
5.4.2 Software-Endschalter.....	69
5.5 Zykluszeiten.....	69
<b>6 Betriebsmodi.....</b>	<b>70</b>
6.1 Profile Position.....	70
6.1.1 Übersicht.....	70
6.1.2 Setzen von Fahrbefehlen.....	71
6.1.3 Genauigkeitsverlust bei Relativbewegungen.....	75
6.1.4 Randbedingungen für eine Positionierfahrt.....	76
6.1.5 Ruck-begrenzter und nicht ruck-begrenzter Modus.....	77
6.2 Velocity.....	78
6.2.1 Beschreibung.....	78
6.2.2 Aktivierung.....	78
6.2.3 Controlword.....	78
6.2.4 Statusword.....	78
6.2.5 Objekteinträge.....	78
6.3 Profile Velocity.....	79
6.3.1 Beschreibung.....	79
6.3.2 Aktivierung.....	79
6.3.3 Controlword.....	80
6.3.4 Statusword.....	80
6.3.5 Objekteinträge.....	80
6.4 Profile Torque.....	82
6.4.1 Beschreibung.....	82
6.4.2 Aktivierung.....	83

6.4.3 Controlword.....	83
6.4.4 Statusword.....	83
6.4.5 Objekteinträge.....	83
6.5 Homing.....	84
6.5.1 Übersicht.....	84
6.5.2 Referenzfahrt-Methode.....	86
6.6 Interpolated Position Mode.....	92
6.6.1 Übersicht.....	92
6.6.2 Aktivierung.....	92
6.6.3 Controlword.....	92
6.6.4 Statusword.....	92
6.6.5 Benutzung.....	92
6.6.6 Setup.....	93
6.6.7 Operation.....	93
6.7 Cyclic Synchronous Position.....	93
6.7.1 Übersicht.....	93
6.7.2 Objekteinträge.....	94
6.8 Cyclic Synchronous Velocity.....	95
6.8.1 Übersicht.....	95
6.8.2 Objekteinträge.....	95
6.9 Cyclic Synchronous Torque.....	96
6.9.1 Übersicht.....	96
6.9.2 Objekteinträge.....	97
6.10 Takt-Richtungs-Modus.....	97
6.10.1 Beschreibung.....	97
6.10.2 Aktivierung.....	97
6.10.3 Generelles.....	97
6.10.4 Statusword.....	98
6.10.5 Unterarten des Takt-Richtungs-Modus.....	98
6.11 Auto-Setup.....	99
6.11.1 Beschreibung.....	99
6.11.2 Aktivierung.....	99
6.11.3 Controlword.....	99
6.11.4 Statusword.....	99
<b>7 Spezielle Funktionen.....</b>	<b>100</b>
7.1 Digitale Ein- und Ausgänge.....	100
7.1.1 Bitzuordnung.....	100
7.1.2 Digitale Eingänge.....	100
7.1.3 Digitale Ausgänge.....	105
7.2 Analoge Eingänge.....	109
7.2.1 Objekteinträge.....	109
7.2.2 Analogwert skalieren.....	109
7.3 Automatische Bremsensteuerung.....	110
7.3.1 Beschreibung.....	110
7.3.2 Aktivierung und Anschluss.....	110
7.3.3 Steuerung der Bremse.....	110
7.3.4 Bremsen-PWM.....	111
7.4 I <sup>2</sup> t Motor-Überlastungsschutz.....	112
7.4.1 Beschreibung.....	112
7.4.2 Objekteinträge.....	112
7.4.3 Aktivierung.....	113
7.4.4 Funktion von I <sup>2</sup> t.....	113
7.5 Objekte speichern.....	113
7.5.1 Allgemeines.....	114
7.5.2 Kategorie: Kommunikation.....	114
7.5.3 Kategorie: Applikation.....	114
7.5.4 Kategorie: Benutzer.....	116

7.5.5 Kategorie: Bewegung.....	116
7.5.6 Kategorie: Tuning.....	116
7.5.7 Kategorie: Modbus RTU.....	117
7.5.8 Speichervorgang starten.....	117
7.5.9 Speicherung verwerfen.....	118
7.5.10 Konfiguration verifizieren.....	118
<b>8 Modbus RTU.....</b>	<b>119</b>
8.1 Modbus Modicon-Notation bei SPS.....	119
8.2 Allgemeines.....	119
8.3 Kommunikationseinstellungen.....	120
8.3.1 Drehschalter.....	120
8.4 Funktionscodes.....	121
8.5 Funktioncode-Beschreibungen.....	121
8.5.1 FC 3 (03 <sub>h</sub> ) Read Input Registers / FC 4 (04 <sub>h</sub> ) Read Holding Registers.....	121
8.5.2 FC 6 (06 <sub>h</sub> ) Write Single Register.....	122
8.5.3 FC 16 (10 <sub>h</sub> ) Write Multiple Registers.....	123
8.5.4 FC 17 (11 <sub>h</sub> ) Report Server ID.....	124
8.5.5 FC 23 (17 <sub>h</sub> ) Read/Write Multiple registers.....	125
8.5.6 FC 8 (08 <sub>h</sub> ) Diagnostics.....	126
8.5.7 FC 43 (2B <sub>h</sub> ) Encapsulated Interface Transport.....	131
8.5.8 FC 101 (65 <sub>h</sub> ) Read complete object dictionary.....	138
8.5.9 FC 102 (66 <sub>h</sub> ) Read complete array or record.....	141
8.5.10 Ausnahmecodes.....	144
8.6 Prozessdatenobjekte (PDO).....	145
8.6.1 Konfiguration.....	145
8.6.2 Übertragung.....	145
8.7 NanoJ-Objekte.....	146
<b>9 Programmierung mit NanoJ.....</b>	<b>147</b>
9.1 NanoJ-Programm.....	147
9.2 Mapping im NanoJ-Programm.....	151
9.3 NanoJ-Funktionen im NanoJ-Programm.....	152
9.4 Einschränkungen und mögliche Probleme.....	154
<b>10 Objektverzeichnis Beschreibung.....</b>	<b>156</b>
10.1 Übersicht.....	156
10.2 Aufbau der Objektbeschreibung.....	156
10.3 Objektbeschreibung.....	156
10.4 Wertebeschreibung.....	157
10.5 Beschreibung.....	158
1000h Device Type.....	159
1001h Error Register.....	160
1003h Pre-defined Error Field.....	161
1008h Manufacturer Device Name.....	165
1009h Manufacturer Hardware Version.....	166
100Ah Manufacturer Software Version.....	166
1010h Store Parameters.....	167
1011h Restore Default Parameters.....	170
1018h Identity Object.....	174
1020h Verify Configuration.....	175
1F50h Program Data.....	177
1F51h Program Control.....	178
1F57h Program Status.....	179
2028h MODBUS Slave Address.....	180
202Ah MODBUS RTU Baudrate.....	180

202Ch MODBUS RTU Stop Bits.....	180
202Dh MODBUS RTU Parity.....	181
2030h Pole Pair Count.....	182
2031h Max Motor Current.....	182
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.....	185
2039h Motor Currents.....	187
203Ah Homing On Block Configuration.....	189
203Bh I2t Parameters.....	191
203Dh Torque Window.....	194
203Eh Torque Window Time Out.....	195
203Fh Max Slippage Time Out.....	195
2057h Clock Direction Multiplier.....	196
2058h Clock Direction Divider.....	196
205Ah Absolute Sensor Boot Value (in User Units).....	196
205Bh Clock Direction Or Clockwise/Counter Clockwise Mode.....	197
2084h Bootup Delay.....	197
2101h Fieldbus Module Availability.....	198
2102h Fieldbus Module Control.....	199
2103h Fieldbus Module Status.....	200
2290h PDI Control.....	202
2291h PDI Input.....	203
2292h PDI Output.....	204
2300h NanoJ Control.....	205
2301h NanoJ Status.....	206
2302h NanoJ Error Code.....	207
230Eh Timer.....	208
230Fh Uptime Seconds.....	209
2310h NanoJ Input Data Selection.....	210
2320h NanoJ Output Data Selection.....	211
2330h NanoJ In/output Data Selection.....	212
2400h NanoJ Inputs.....	214
2410h NanoJ Init Parameters.....	215
2500h NanoJ Outputs.....	216
2600h NanoJ Debug Output.....	216
2701h Customer Storage Area.....	217
2800h Bootloader And Reboot Settings.....	218
3202h Motor Drive Submode Select.....	220
3203h Feedback Selection.....	221
3204h Feedback Mapping.....	223
320Dh Torque Of Inertia Factor.....	224
3210h Motor Drive Parameter Set.....	226
3212h Motor Drive Flags.....	230
321Ah Current Controller Parameters.....	231
321Bh Velocity Controller Parameters.....	234
321Ch Position Controller Parameters.....	235
321Dh Pre-control.....	237
321Eh Voltage Limit.....	238
3220h Analog Input Digits.....	239
3221h Analog Inputs Control.....	240
3225h Analog Input Switches.....	240
3240h Digital Inputs Control.....	241
3242h Digital Input Routing.....	244
3243h Digital Input Homing Capture.....	245
3250h Digital Outputs Control.....	247
3252h Digital Output Routing.....	250

3320h Analog Input Values.....	251
3321h Analog Input Offsets.....	252
3322h Analog Input Numerators.....	253
3323h Analog Input Denominators.....	254
3380h Feedback Sensorless.....	255
33A0h Feedback Incremental A/B/I 1.....	256
33A1h Feedback Incremental A/B/I 2.....	258
3502h MODBUS Rx PDO Mapping.....	259
3602h MODBUS Tx PDO Mapping.....	263
3700h Deviation Error Option Code.....	267
3701h Limit Switch Error Option Code.....	267
4012h HW Information.....	268
4013h HW Configuration.....	269
4014h Operating Conditions.....	270
4015h Special Drive Modes.....	272
4016h Factory Settings.....	273
4021h Ballast Configuration.....	274
4040h Drive Serial Number.....	276
4041h Device Id.....	276
4042h Bootloader Infos.....	276
603Fh Error Code.....	278
6040h Controlword.....	278
6041h Statusword.....	279
6042h VI Target Velocity.....	281
6043h VI Velocity Demand.....	281
6044h VI Velocity Actual Value.....	282
6046h VI Velocity Min Max Amount.....	282
6048h VI Velocity Acceleration.....	283
6049h VI Velocity Deceleration.....	284
604Ah VI Velocity Quick Stop.....	285
604Ch VI Dimension Factor.....	287
605Ah Quick Stop Option Code.....	288
605Bh Shutdown Option Code.....	288
605Ch Disable Option Code.....	289
605Dh Halt Option Code.....	290
605Eh Fault Option Code.....	290
6060h Modes Of Operation.....	291
6061h Modes Of Operation Display.....	292
6062h Position Demand Value.....	292
6063h Position Actual Internal Value.....	293
6064h Position Actual Value.....	293
6065h Following Error Window.....	294
6066h Following Error Time Out.....	294
6067h Position Window.....	295
6068h Position Window Time.....	296
606Bh Velocity Demand Value.....	296
606Ch Velocity Actual Value.....	297
606Dh Velocity Window.....	297
606Eh Velocity Window Time.....	298
606Fh Velocity Threshold.....	298
6070h Velocity Threshold Time.....	299
6071h Target Torque.....	299
6072h Max Torque.....	300
6073h Max Current.....	301
6074h Torque Demand.....	301
6075h Motor Rated Current.....	302
6077h Torque Actual Value.....	302
607Ah Target Position.....	303
607Bh Position Range Limit.....	303

607Ch Home Offset.....	304
607Dh Software Position Limit.....	304
607Eh Polarity.....	305
607Fh Max Profile Velocity.....	306
6080h Max Motor Speed.....	307
6081h Profile Velocity.....	308
6082h End Velocity.....	308
6083h Profile Acceleration.....	309
6084h Profile Deceleration.....	309
6085h Quick Stop Deceleration.....	310
6086h Motion Profile Type.....	310
6087h Torque Slope.....	311
608Fh Position Encoder Resolution.....	311
6090h Velocity Encoder Resolution.....	312
6091h Gear Ratio.....	314
6092h Feed Constant.....	315
6096h Velocity Factor.....	316
6097h Acceleration Factor.....	317
6098h Homing Method.....	318
6099h Homing Speed.....	319
609Ah Homing Acceleration.....	320
60A2h Jerk Factor.....	320
60A4h Profile Jerk.....	321
60A8h SI Unit Position.....	323
60A9h SI Unit Velocity.....	324
60B0h Position Offset.....	324
60B1h Velocity Offset.....	325
60B2h Torque Offset.....	325
60C1h Interpolation Data Record.....	326
60C2h Interpolation Time Period.....	327
60C4h Interpolation Data Configuration.....	328
60C5h Max Acceleration.....	330
60C6h Max Deceleration.....	330
60E4h Additional Position Actual Value.....	331
60E5h Additional Velocity Actual Value.....	332
60E6h Additional Position Encoder Resolution - Encoder Increments.....	333
60E8h Additional Gear Ratio - Motor Shaft Revolutions.....	334
60E9h Additional Feed Constant - Feed.....	335
60EBh Additional Position Encoder Resolution - Motor Revolutions.....	336
60EDh Additional Gear Ratio - Driving Shaft Revolutions.....	337
60EEh Additional Feed Constant - Driving Shaft Revolutions.....	338
60F2h Positioning Option Code.....	339
60F4h Following Error Actual Value.....	341
60F8h Max Slippage.....	342
60FAh Control Effort.....	342
60FCh Position Demand Internal Value.....	343
60FDh Digital Inputs.....	343
60FEh Digital Outputs.....	344
60FFh Target Velocity.....	345
6502h Supported Drive Modes.....	346
6503h Drive Catalogue Number.....	347
6505h Http Drive Catalogue Address.....	347
<b>11 Copyrights.....</b>	<b>348</b>
11.1 Einführung.....	348
11.2 AES.....	348
11.3 MD5.....	348
11.4 uIP.....	349



11.5 DHCP.....	349
11.6 CMSIS DSP Software Library.....	349
11.7 FatFs.....	349
11.8 Protothreads.....	350
11.9 lwIP.....	350
11.10 littlefs.....	351

## 1 Einleitung

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

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

### 1.1 Versionshinweise

Version Handbuch	Datum	Änderungen	Version Firmware	Version Hardware
1.0.0	07/2022	Veröffentlichung	v2213	W003

### 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](http://www.nanotec.de)

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

### 1.3 Bestimmungsgemäße Verwendung

Der Motor mit integrierter Steuerung *PD6-E* findet Verwendung als Komponente von Antriebssystemen in vielfältigen Industrieanwendungen.

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

Unter keinen Umständen darf dieses Nanotec-Produkt als Sicherheitsbauteil in ein Produkt oder eine Anlage integriert werden. Alle Produkte, in denen eine von Nanotec hergestellte Komponente enthalten ist, müssen bei Übergabe an Endnutzer entsprechende Warnhinweise samt Anleitung für sichere Verwendung und sicheren Betrieb enthalten. Alle von Nanotec bereitgestellten Warnhinweise müssen unmittelbar an den Endnutzer weitergegeben werden.

## 1.4 Gewährleistung und Haftungsausschluss

Für Schäden und Betriebsstörungen, die durch Montagefehler, Nichtbeachtung dieses Handbuchs oder unsachgemäße Reparaturen entstehen, übernimmt Nanotec keine Haftung. Die Auswahl bzw. Verwendung von Nanotec-Produkten liegt im Verantwortungsbereich des Anlagenkonstruktors bzw. Endnutzers. Nanotec übernimmt keinerlei Verantwortung für die Integration des Produkts in das Endsystem.

Es gelten unsere Allgemeinen Geschäftsbedingungen auf [www.nanotec.com](http://www.nanotec.com).



### HINWEIS

Änderungen oder Umbauten des Produkts sind nicht zulässig.

## 1.5 Zielgruppe und Qualifikation

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

- Entwicklungsingenieure
- Anlagenkonstruktoren
- Monteure/Servicekräfte
- Applikationsingenieure

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

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

## 1.6 EU-Richtlinien zur Produktsicherheit

Folgende EU-Richtlinien wurden beachtet:

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

## 1.7 Mitgeltende Vorschriften

Neben diesem technischen Handbuch sind folgende Vorschriften zu beachten:

- Unfallverhütungsvorschriften
- örtliche Vorschriften zur Arbeitssicherheit

## 1.8 Verwendete Symbole

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

### VORSICHT



**Der Hinweis VORSICHT verweist auf eine möglicherweise gefährliche Situation.**

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

- ▶ Beschreibt, wie Sie die gefährliche Situation vermeiden.

### HINWEIS



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

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

► Beschreibt, wie Sie die Fehlbedienung vermeiden.



### TIPP

Zeigt einen Tipp zur Anwendung oder Aufgabe.

## 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`.

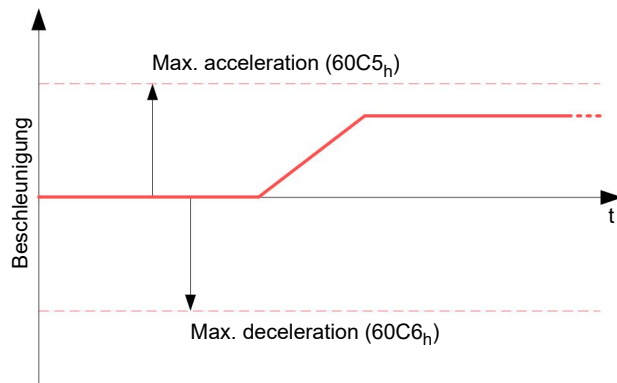
## 1.11 Bits

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

	MSB							LSB	
Bit Nummer	7	6	5	4	3	2	1	0	
Bits	0	1	0	1	0	1	0	1	≅ 55 <sub>hex</sub> ≅ 85 <sub>dec</sub>

## 1.12 Zählrichtung (Pfeile)

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



## 2 Sicherheits- und Warnhinweise

### VORSICHT

#### Überhitzungs- oder Brandgefahr bei unzureichender Kühlung!



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

- ▶ Achten Sie beim Einsatz darauf, dass die Kühlung und die Umgebungsbedingungen gewährleistet sind.
- ▶ Achten Sie bei der Einbaulage des Motors *PD6-EB...* darauf, dass die Kühlrippen die Wärmeabführung begünstigen.

### HINWEIS

#### Beschädigung der Steuerung durch Erregerspannung des Motors!



Während des Betriebs können Spannungsspitzen die Steuerung beschädigen.

- ▶ Verbauen Sie geeignete Schaltungen (z. B. Stützkondensator), die Spannungsspitzen abbauen.

### HINWEIS

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



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

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

### HINWEIS

#### Beschädigung der Elektronik durch verpolten Anschluss der Versorgungsspannung!



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

- ▶ Beachten Sie die Pin-Belegung.
- ▶ Installieren Sie eine Leitungsschutzeinrichtung (Sicherung) in der Zuleitung.

### 3 Technische Daten und Anschlussbelegung

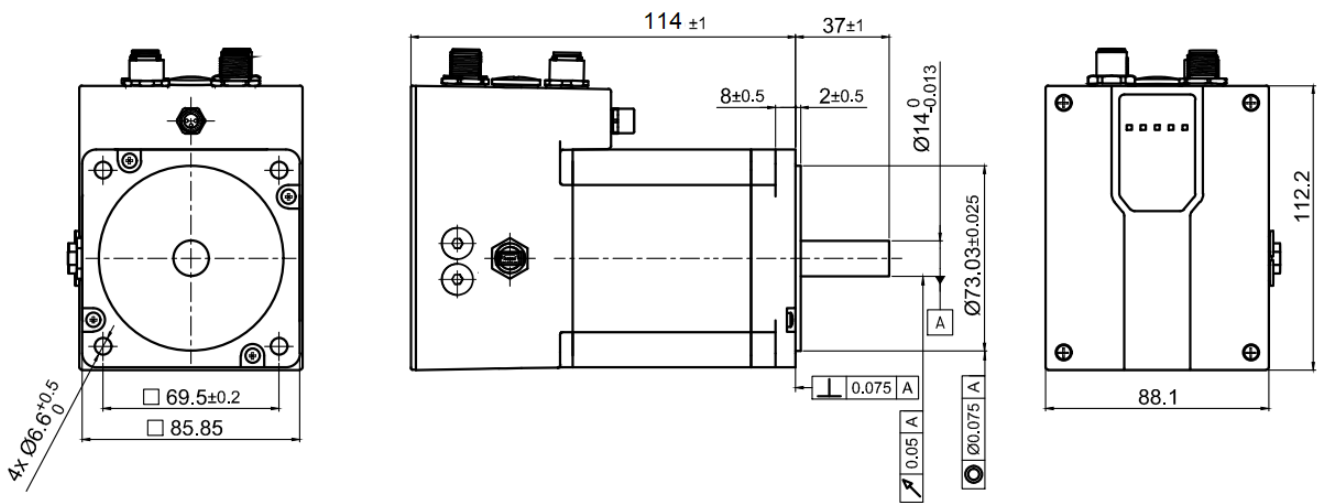
#### 3.1 Umgebungsbedingungen

Umgebungsbedingung	Wert
Schutzklasse	IP65 (IP54 am Wellenausgang)
Umgebungstemperatur (Betrieb)	-10 ... +40°C
Luftfeuchtigkeit (nicht kondensierend)	0 ... 85%
Max. Aufstellhöhe über <i>NN</i> (ohne Leistungsbeschränkung)	1500 m
Umgebungstemperatur (Lagerung)	-25 ... +85°C

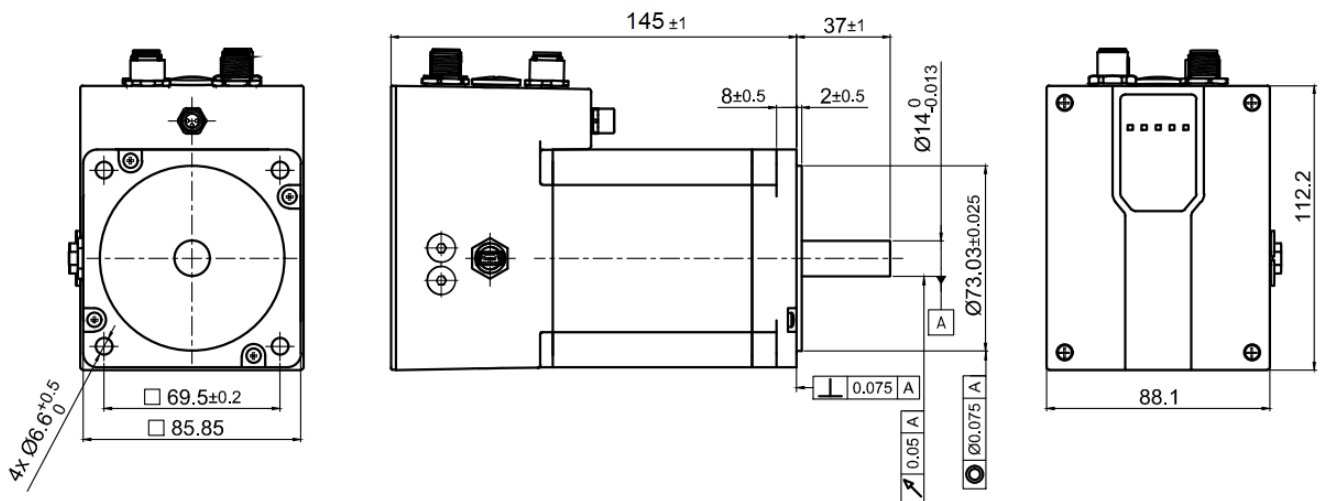
#### 3.2 Maßzeichnungen

Alle Maße sind in Millimetern.

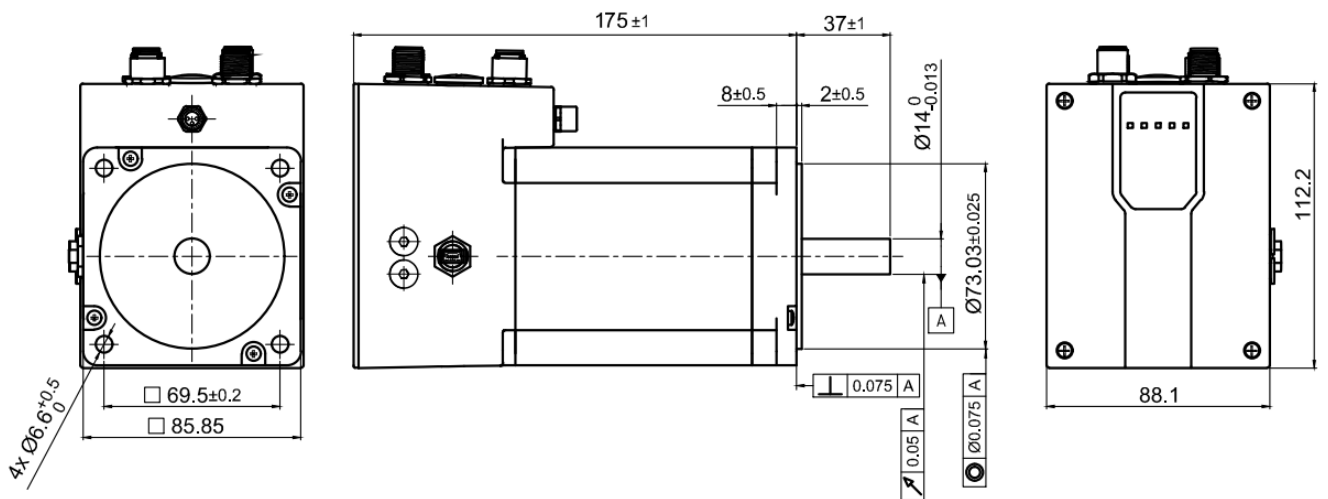
##### 3.2.1 PD6-E891S95-E-65-5



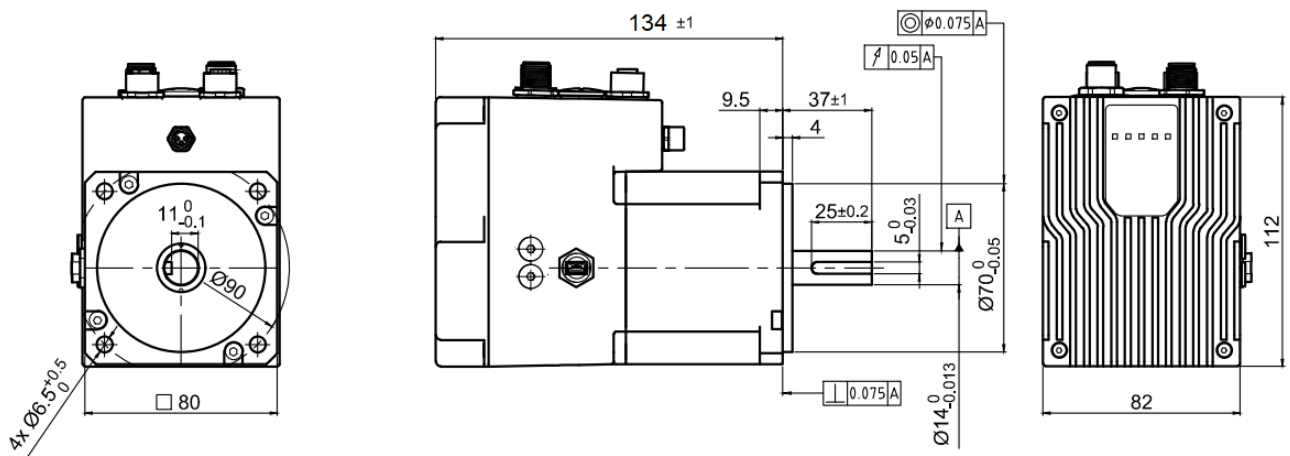
### 3.2.2 PD6-E891M95-E-65-5



### 3.2.3 PD6-E891L95-E-65-5

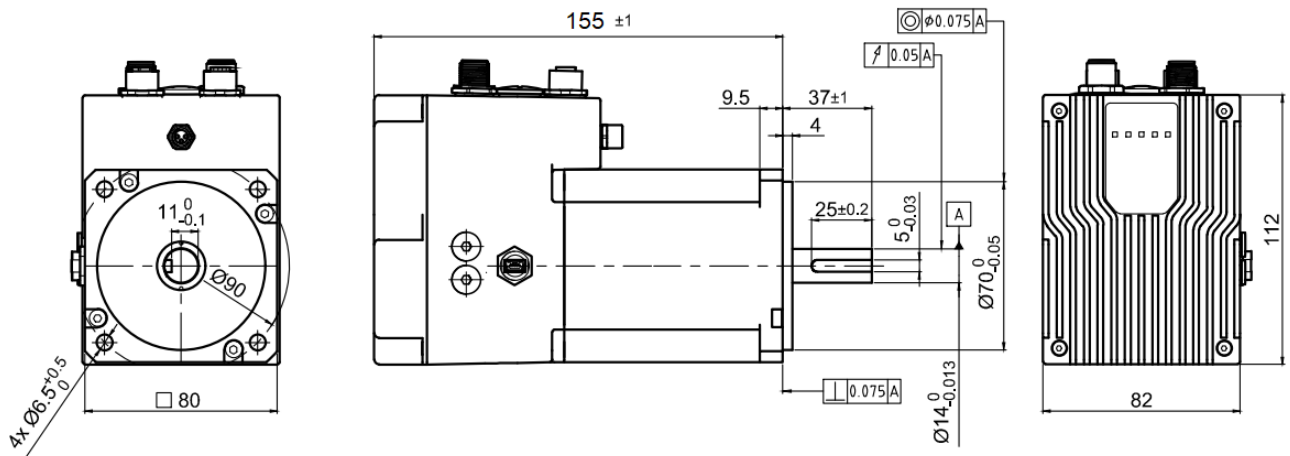


### 3.2.4 PD6-EB80SD-E-65-5

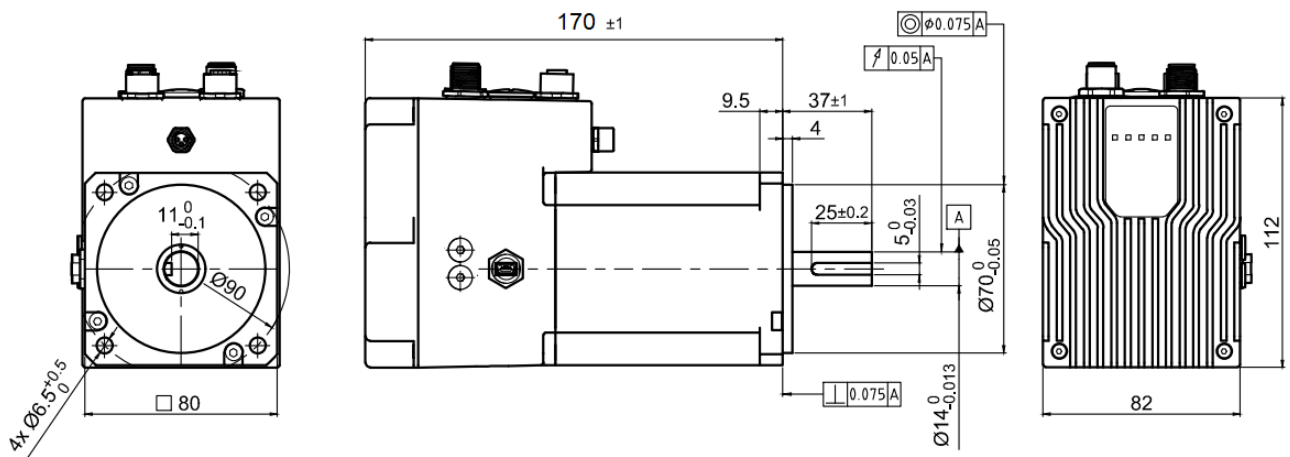




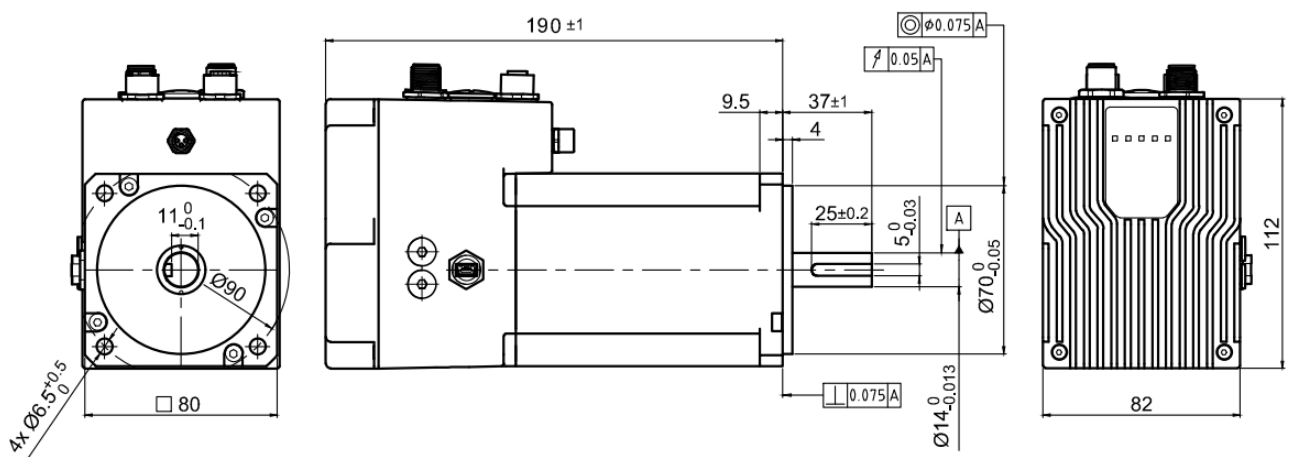
**3.2.5 PD6-EB80MD-E-65-5**



**3.2.6 PD6-EB80LD-E-65-5**



**3.2.7 PD6-EB80CD-E-65-5**



### 3.3 Elektrische Eigenschaften und technische Daten

#### 3.3.1 Elektrische Daten

Betriebsspannung (Nennwert)	48 V DC
Zulässiger Bereich der Betriebsspannung	12...57,6 V DC
Logikspannung (Nennwert)	24 V DC
Spannungsbereich der Logikversorgung	12...30 V DC
Nennstrom	<ul style="list-style-type: none"> <li>■ PD6-E891S95-... : 9,5 A eff.</li> <li>■ PD6-E8918M95-... : 9,5 A eff.</li> <li>■ PD6-E8918L95-... : 9,5 A eff.</li> <li>■ PD6-EB80SD -... : 6,9 A eff.</li> <li>■ PD6-EB80MD -... : 14 A eff.</li> <li>■ PD6-EB80LD -... : 18,75 A eff.</li> <li>■ PD6-EB80CD-... : 25 A eff.</li> </ul>
Spitzenstrom (für max. 5 s)	<ul style="list-style-type: none"> <li>■ PD6-E8918S95-... : 12,5 A eff.</li> <li>■ PD6-E8918M95-... : 12,5 A eff.</li> <li>■ PD6-E8918L95-... : 12,5 A eff.</li> <li>■ PD6-EB80SD-... : 20 A eff.</li> <li>■ PD6-EB80MD-... : 40 A eff.</li> <li>■ PD6-EB80LD-... : 65 A eff.</li> <li>■ PD6-EB80CD-... : 75 A eff.</li> </ul>

#### 3.3.2 Technische Daten

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-Richtungs-Modus</i>
Sollwertvorgabe/ Programmierung	<i>Modbus RTU, Analogeingang, NanoJ-Programm</i>
Eingänge	6 Digitaleingänge (+5 V/+24 V DC), einzeln per Software umschaltbar, Werkseinstellung: 5 V  1 Analogeingang 0...10 V oder 0...20 mA (per Software umschaltbar)
Ausgänge	2 digitale Ausgänge, <i>Open Drain</i> , max. 100 mA  1 Bremsenausgang für externe Haltebremse
Integrierter Encoder	magnetischer Singleturn-Absolut-Encoder, 1024 Impulse/Umdrehung (4096 nach Quadratur)
Schutzschaltung	Über- und Unterspannungsschutz, interne Ballast-Schaltung  Übertemperaturschutz (> 75°C auf der Leistungsplatine, entspricht ca. 65 - 72 °C außen am Deckel)

Verpolungsschutz: bei Verpolung Kurzschluss zwischen Versorgungsspannung und GND über Leistungsdiode, daher ist eine Leitungsschutzeinrichtung (Sicherung) in Zuleitung nötig. Die Werte der Sicherung sind abhängig von der Applikation und müssen

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

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

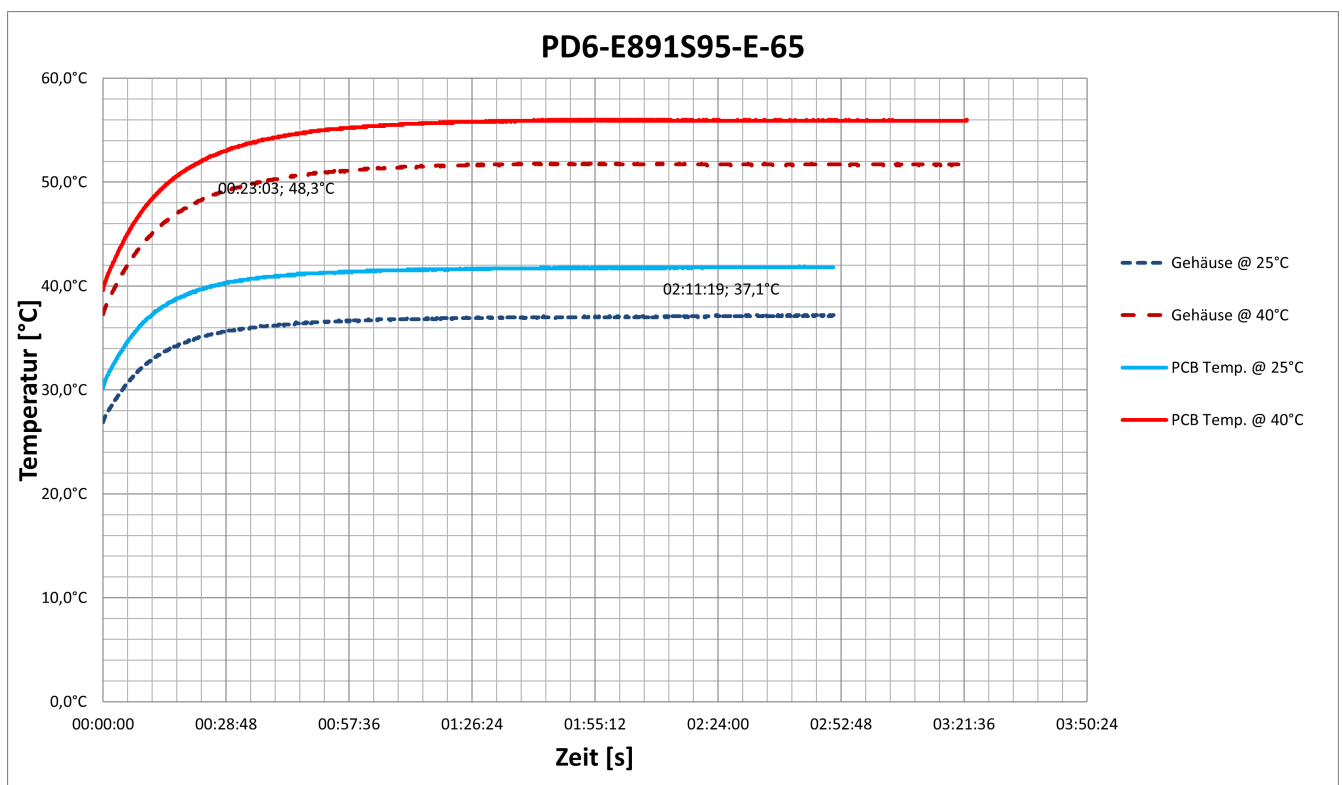
### 3.4 Übertemperaturschutz

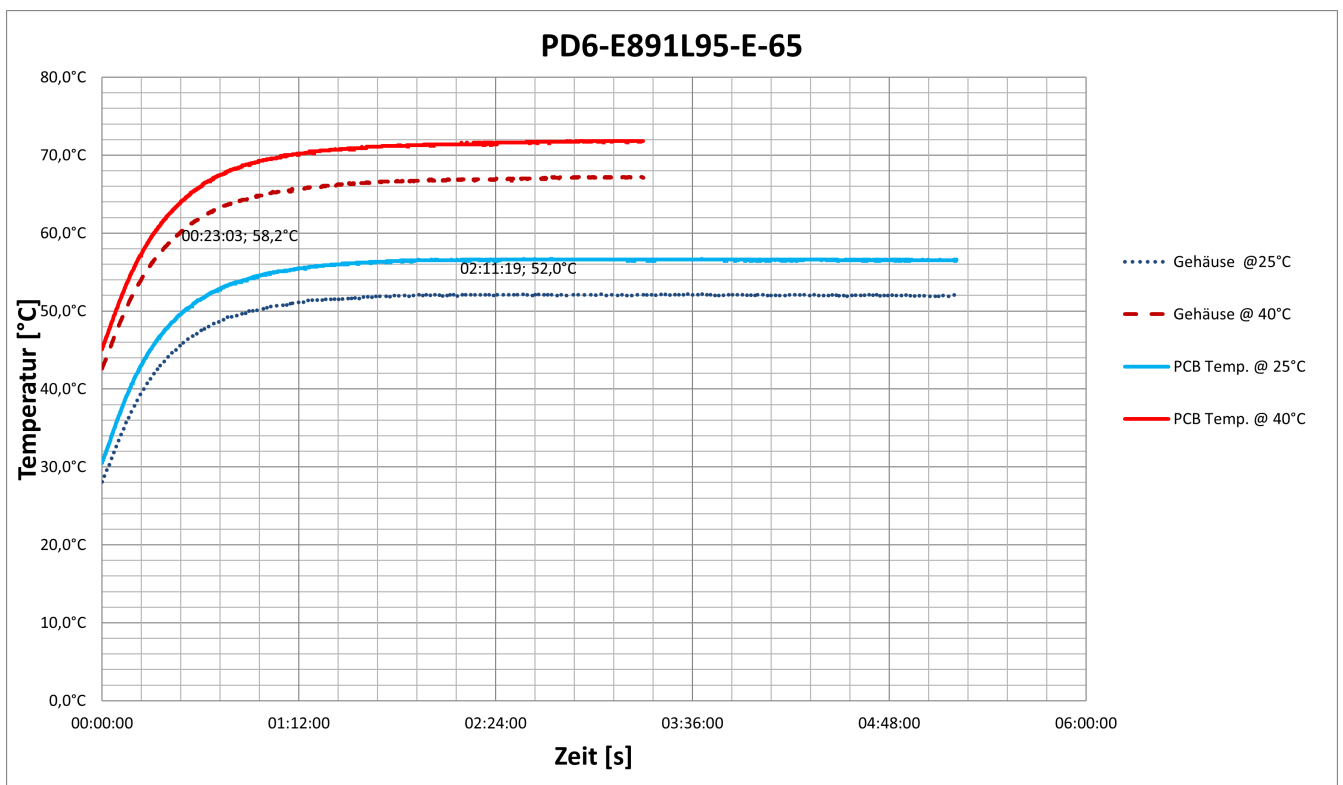
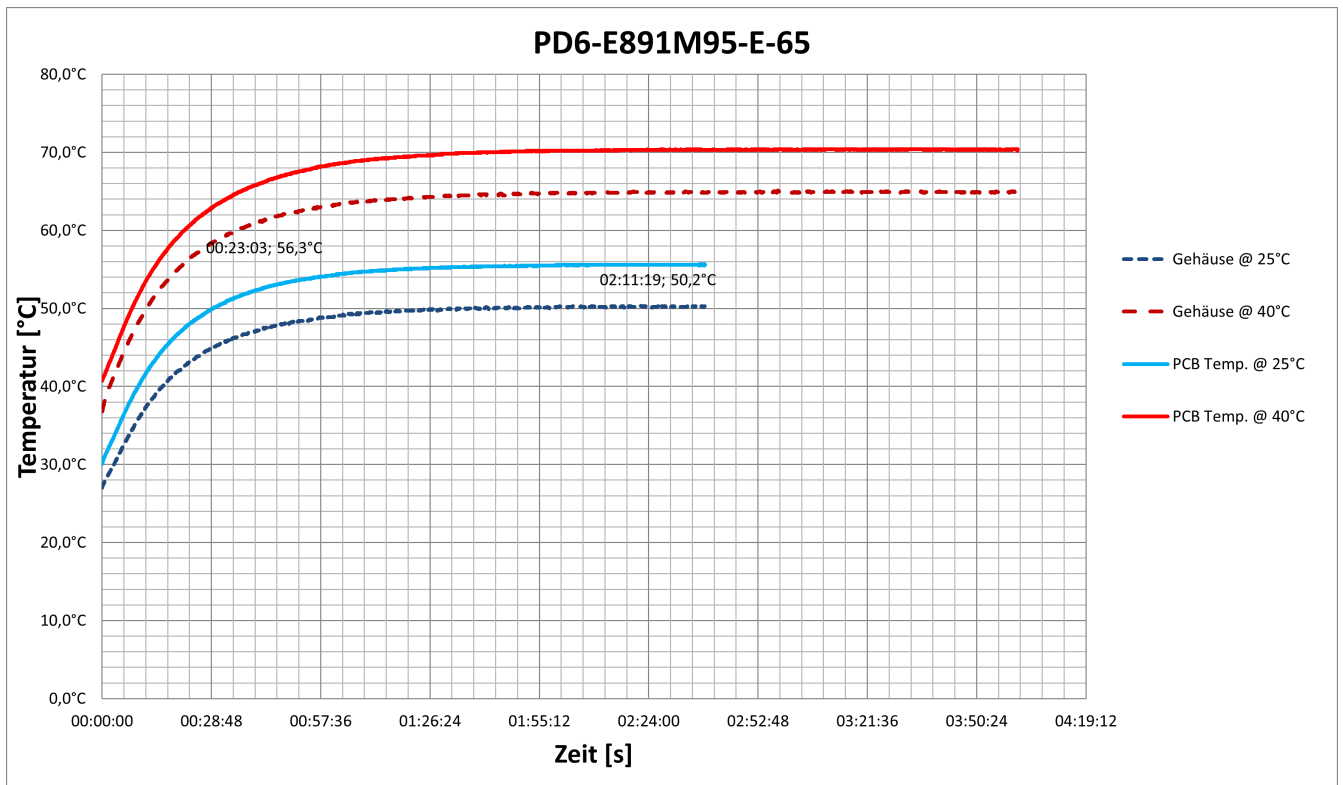
Ab einer Temperatur von ca. 75°C auf der Leistungsplatine (entspricht 65 - 72°C außen am Deckel) wird das Leistungsteil der Steuerung abgeschaltet und das Fehlerbit gesetzt (siehe Objekt 1001<sub>n</sub> und 1003<sub>n</sub>). Nach Abkühlung und dem Bestätigen des Fehlers (siehe Tabelle für das Contolword,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

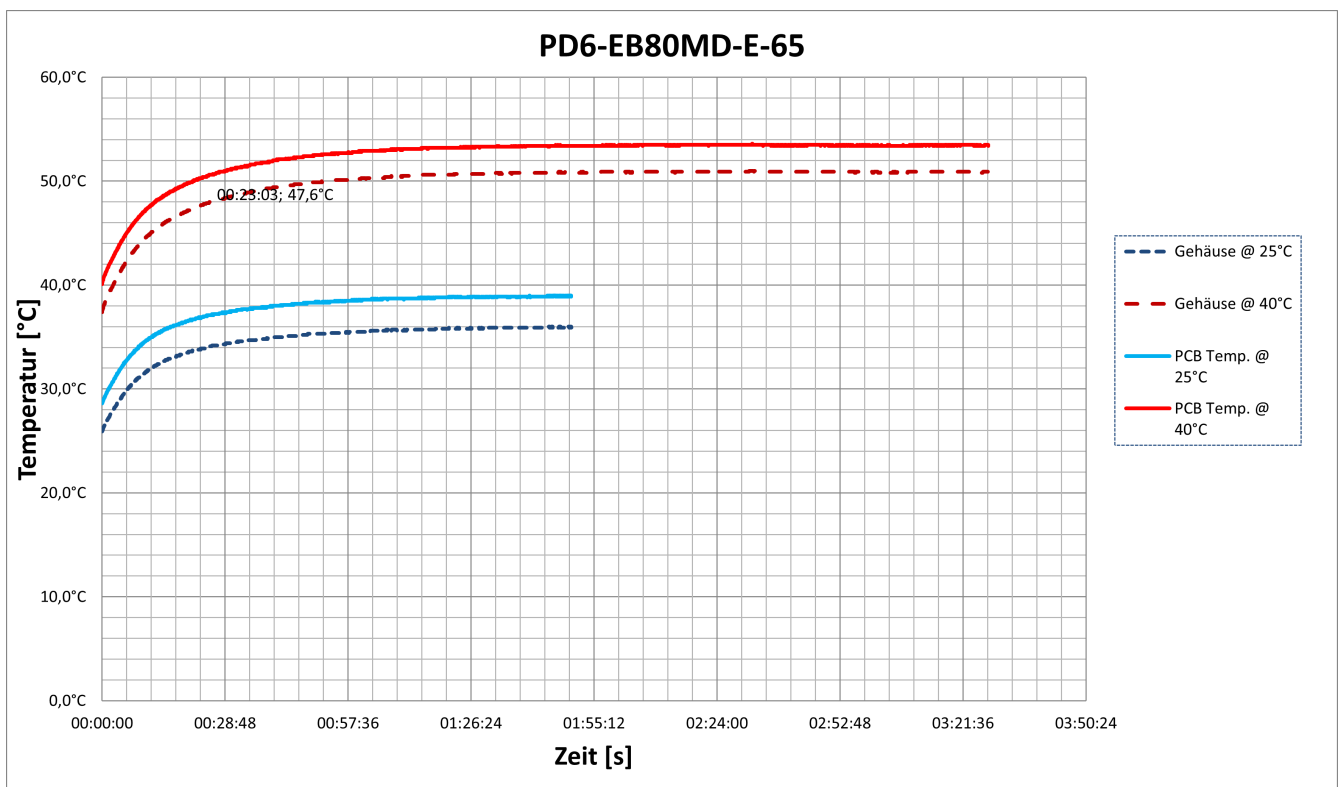
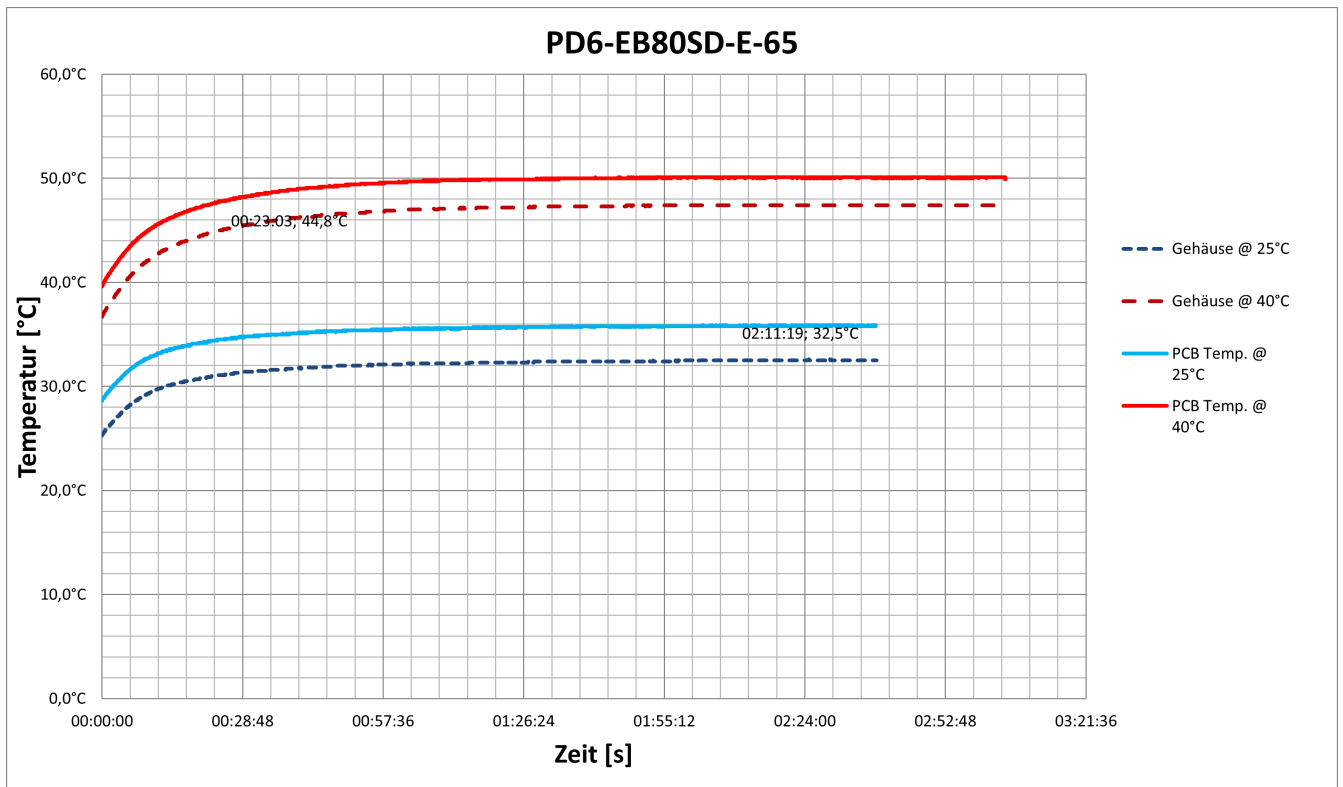
Es wurden Temperaturtests unter folgenden Bedingungen durchgeführt:

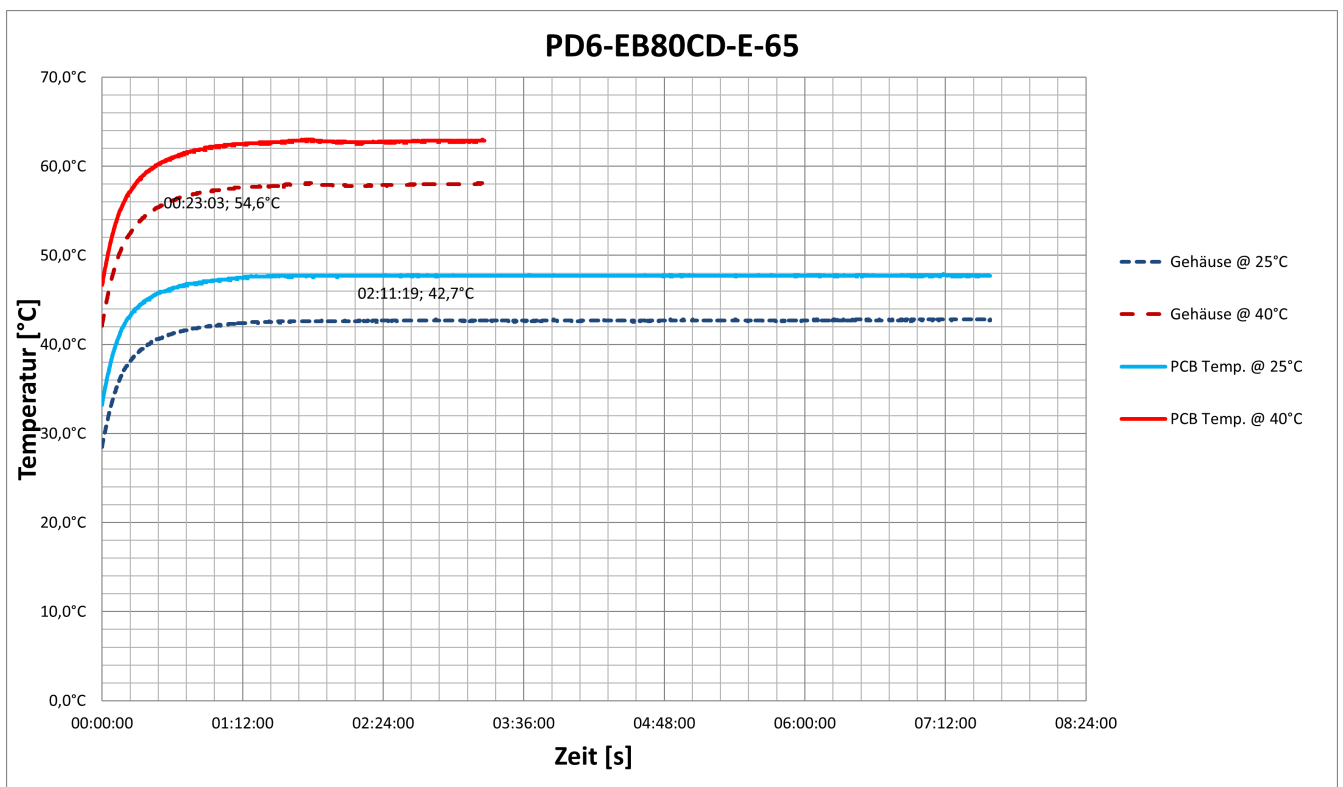
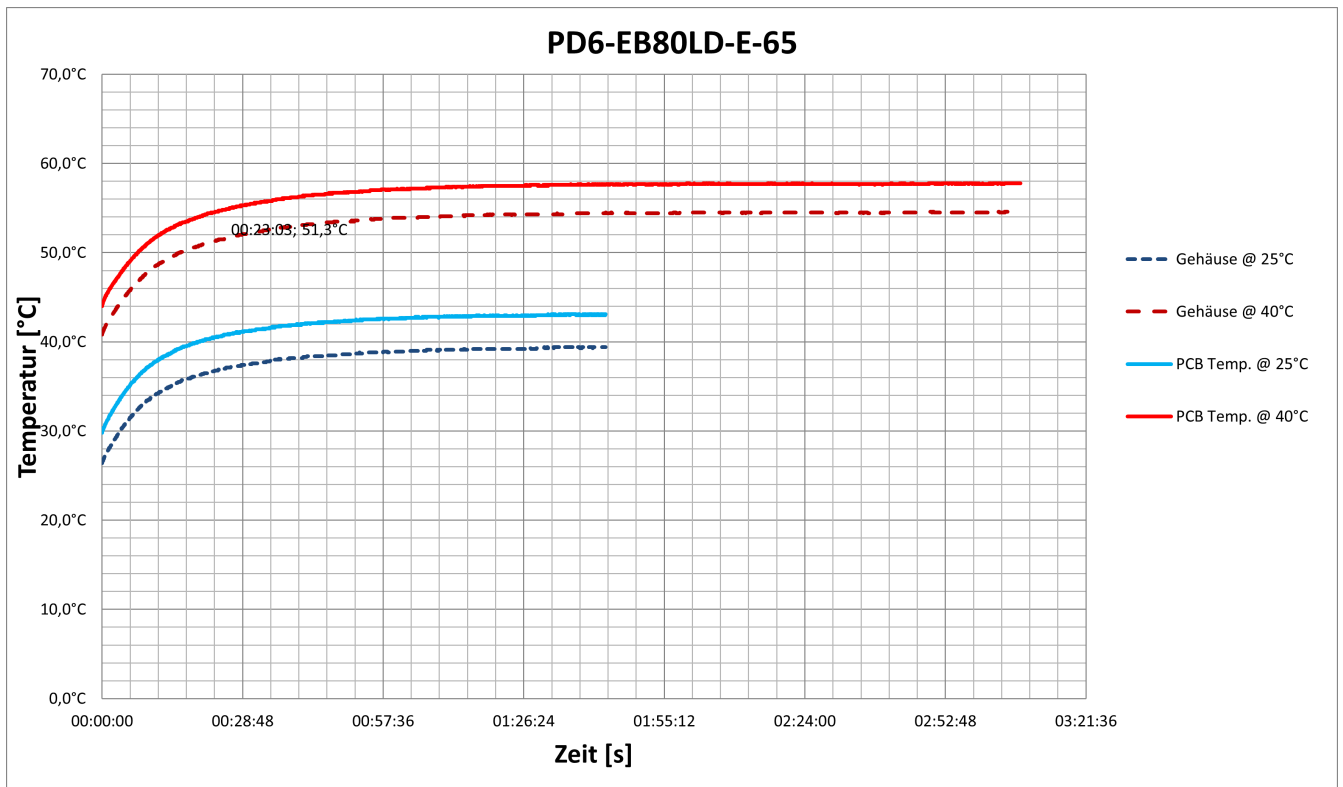
- Betriebsspannung / Logikspannung: 48/24 V DC
- Motorstrom: Nennstrom (siehe Elektrische Eigenschaften und technische Daten)
- Operationsmodus: *Open Loop*, Drehzahlmodus 100 U/min
- Umgebungstemperatur: 25°C / 40°C
- Aufstellhöhe: 500 m über NN
- keine externe Kühlung im Klimaschrank, z. B. über Lüfter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

Die folgenden Grafiken zeigen die Ergebnisse der Temperaturtests:









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

**HINWEIS**

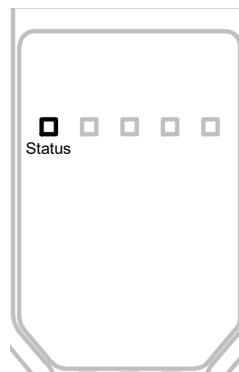


Da das genaue Temperaturverhalten außer vom Motor auch von der Anflanschung und dem dortigen Wärmeübergang sowie von der Konvektion in der Applikation abhängt, empfehlen wir bei Applikationen, die hinsichtlich Stromhöhe und Umgebungstemperatur problematisch sind, immer einen Dauertest in der realen Umgebung.

### 3.5 LED-Signalisierung

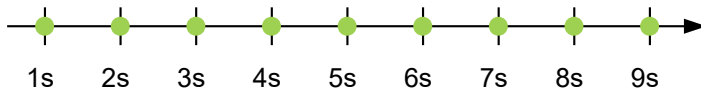
#### 3.5.1 Betriebs-LED

Die Betriebs-LED zeigt den aktuellen Status an.



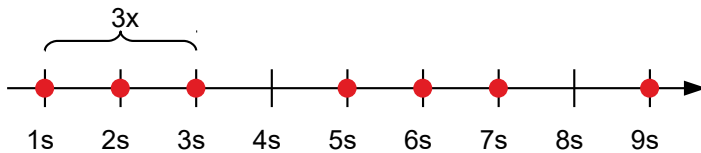
##### 3.5.1.1 Normaler Betrieb

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



##### 3.5.1.2 Fehlerfall

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



Folgende Tabelle zeigt die Bedeutung der Fehlernummern.

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


**HINWEIS**

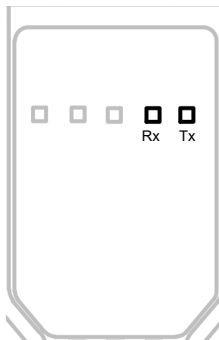
Für jeden aufgetretenen Fehler wird im Objekt `1003h` ein genauerer Fehlercode hinterlegt.


**TIPP**

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

### 3.5.2 Modbus LEDs

Die zwei Modbus LEDs blinken, wenn die Modbus-Kommunikation aktiv ist.



## 3.6 Anschlussbelegung

### 3.6.1 Übersicht

Stecker	Funktion
X1	Spannungsversorgung
X2	Ein-/Ausgänge und Logikversorgung
X3	Modbus RTU IN
X4	Modbus RTU OUT
X5	Anschluss für externe Haltebremse
X6	Mini-USB
S1	Hex-Codierschalter für Node-ID und Baudrate

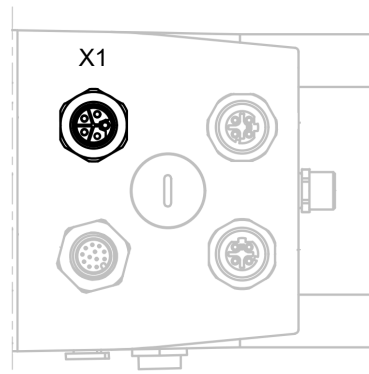

**HINWEIS**

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

### 3.6.2 X1 — Spannungsversorgung

Anschluss für die Hauptversorgung (Leistung). Stecker-Typ: M12, 4-polig, L-kodiert, male  
 Passendes Nanotec-Kabel: ZK-M12-4-...-...-LFF (nicht im Lieferumfang enthalten)





**HINWEIS**



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

### 3.6.2.1 Spannungsquelle

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

**HINWEIS**



**EMV: Bei einer DC-Stromversorgungsleitung mit einer Länge von >30 m oder Verwendung des Motors an einem DC-Bus sind zusätzliche Entstör- und Schutzmaßnahmen notwendig.**

- ▶ Ein EMI-Filter ist in die DC-Zuleitung mit möglichst geringem Abstand zur Steuerung/Motor einzufügen.
- ▶ Lange Daten- oder Versorgungsleitungen sind durch Ferrite zu führen.

### 3.6.2.2 Pin-Belegung

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

**HINWEIS**



**Beschädigung der Elektronik durch verpolten Anschluss der Versorgungsspannung!**

- ▶ Beachten Sie die Pin-Belegung.
- ▶ Installieren Sie eine Leitungsschutzeinrichtung (Sicherung) in der Zuleitung.

### 3.6.2.3 Zulässige Betriebsspannung

Die maximale Betriebsspannung beträgt 57,6 V DC. Steigt die Eingangsspannung der Steuerung über den in 2034<sub>h</sub> eingestellten Schwellwert, wird der Motor abgeschaltet und ein Fehler ausgelöst. Ab der in 4021<sub>h</sub>:02<sub>h</sub> eingestellten Ansprechschwelle wird die integrierte Ballast-Schaltung aktiviert (Drahtwiderstand CRS500JT-73-15RAA von VITROHM mit 5 W Dauerleistung).

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

An die Versorgungsspannung muss ein Ladekondensator von mindestens 4700 µF / 50 V (ca. 1000 µF pro Ampere Nennstrom) angeschlossen sein, um ein Überschreiten der zulässigen Betriebsspannung (z. B. beim Bremsvorgang) zu vermeiden.

#### HINWEIS

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

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

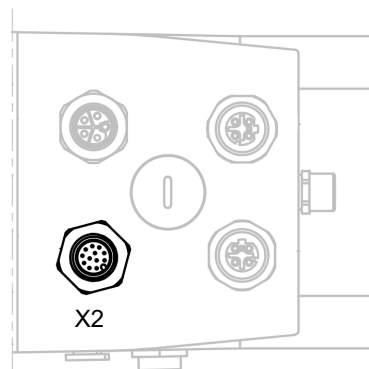


- ▶ Verbauen Sie geeignete Schaltungen (z. B. Ladekondensator), die Spannungsspitzen abbauen.
- ▶ Bei BLDC-Motoren (PD6-EB...): Wählen Sie eine Spannungsquelle, die der Nennspannung des jeweiligen Motors entspricht, wie im Motordatenblatt angegeben.
- ▶ Verwenden Sie ein Netzteil mit Schutzschaltung gegen Überspannung.

### 3.6.3 X2 — Ein-/Ausgänge und Logikversorgung

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

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



Pin	Funktion	Bemerkung
1	GND	
2	Digitaler Eingang 1	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240 <sub>h</sub> , max. 1 MHz
3	Digitaler Eingang 2	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240 <sub>h</sub> , max. 1 MHz
4	Digitaler Eingang 3	5 V / 24 V Pegel, umschaltbar per Software mit Objekt 3240 <sub>h</sub> , max. 1 MHz

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

**HINWEIS**



Die Logikversorgung versorgt die Elektronik, den Encoder und die Kommunikationsschnittstelle. Die Wicklungen des Motors werden nicht von der Logikversorgung versorgt.

Für Eingang 1 bis 6 gelten folgende Schaltschwellen:

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

**Alternative Funktion**

Sie können in 4015<sub>h</sub> die alternative Funktion der digitalen Eingänge aktivieren, die für die *speziellen Fahrmodi* verwendet wird. Siehe Kapitel *Spezielle Fahrmodi (Takt-Richtung und Analog-Drehzahl)*.

Wenn Sie das 3240<sub>h</sub>:07<sub>h</sub> auf den Wert "1" setzen, stehen Ihnen, anstatt sechs single-ended, drei differentielle Eingänge zur Verfügung.

Die folgende Tabelle zeigt alle möglichen Kombinationen:

Pin	Basisfunktion		Alternative Funktion	
	Single-ended	Differenziell	Single-ended	Differenziell
2	Eingang 1	- Eingang 1	Freigabe	-Freigabe
3	Eingang 2 / Richtungseingang im Takt-Richtungs-Modus	+ Eingang 1	Richtung	Freigabe
4	Eingang 3 / Takteingang im Takt- Richtungs-Modus	-Eingang 2/ Richtungseingang im Takt-Richtungs-Modus	Takt	-Richtung
5	Eingang 4	+ Eingang 2 / + Richtungseingang im Takt-Richtungs-Modus	Digitaler Eingang 4	Richtung

Pin	Basisfunktion		Alternative Funktion	
	Single-ended	Differenziell	Single-ended	Differenziell
6	Eingang 5	- Eingang 3 / - Takteingang im Takt-Richtungs-Modus	Digitaler Eingang 5	-Takt
7	Eingang 6	+Eingang 3 / + Takteingang im Takt-Richtungs-Modus	Digitaler Eingang 6	Takt

**Externen Encoder anschließen**

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

**HINWEIS**



Die durch die Encoder-Signale belegten Pins können nicht mehr als Takt-/Richtungs-Eingang verwendet werden.

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

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

**HINWEIS**

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



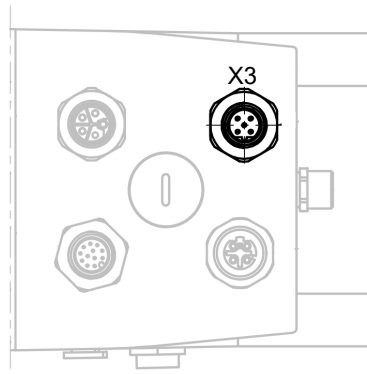
Für 24-V-Encoder müssen Sie:

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

**3.6.4 X3 — Modbus RTU (RS-485) IN**

Anschluss für Modbus RTU. Stecker-Typ: M12, 5-polig, B-kodiert, male

Passendes Nanotec-Kabel: ZK-M12-5-2M-1-B-S (nicht im Lieferumfang enthalten)



Pin	Funktion	Bemerkung
1	n.c.	Mit Pin 1 von X4 intern verbunden
2	RS-485-	
3	COMMON GND	galvanisch getrennt von dem GND der Hauptversorgung und der Ein-/Ausgänge
4	RS-485+	
5	n.c.	

### 3.6.4.1 Leitungspolarisation RS-485

#### HINWEIS



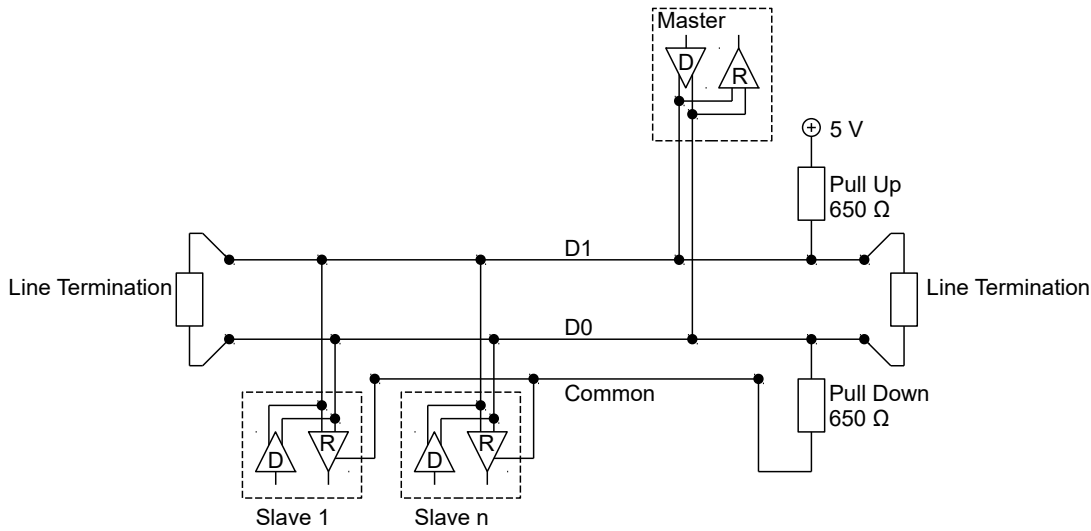
Die Steuerung ist **nicht** mit einer Leitungspolarisation ausgestattet und erwartet, dass das Master Gerät eine besitzt.

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

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

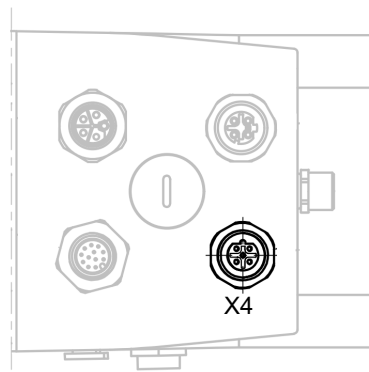
Der Wert dieser Widerstände muss zwischen 450 Ohm und 650 Ohm liegen. Ein 650 Ohm Widerstand erlaubt eine höhere Anzahl an Geräten am Bus.

In diesem Fall muss eine Leitungspolarisation an einer Stelle für den gesamten seriellen Bus angebracht werden. Generell sollte dieser Punkt an dem Master Gerät oder seinem Anschluss sein. Alle anderen Geräte müssen dann keine Leitungspolarisation mehr umsetzen.



### 3.6.5 X4 — Modbus RTU (RS-485) OUT

Anschluss für Modbus RTU. Stecker-Typ: M12, 5-polig, B-kodiert, female



Pin	Funktion	Bemerkung
1	n.c.	Mit Pin 1 von X3 intern verbunden
2	RS-485-	
3	COMMON GND	galvanisch getrennt von dem GND der Hauptversorgung und der Ein-/Ausgänge
4	RS-485+	
5	n.c.	

#### 3.6.5.1 Leitungspolarisation RS-485

**HINWEIS**



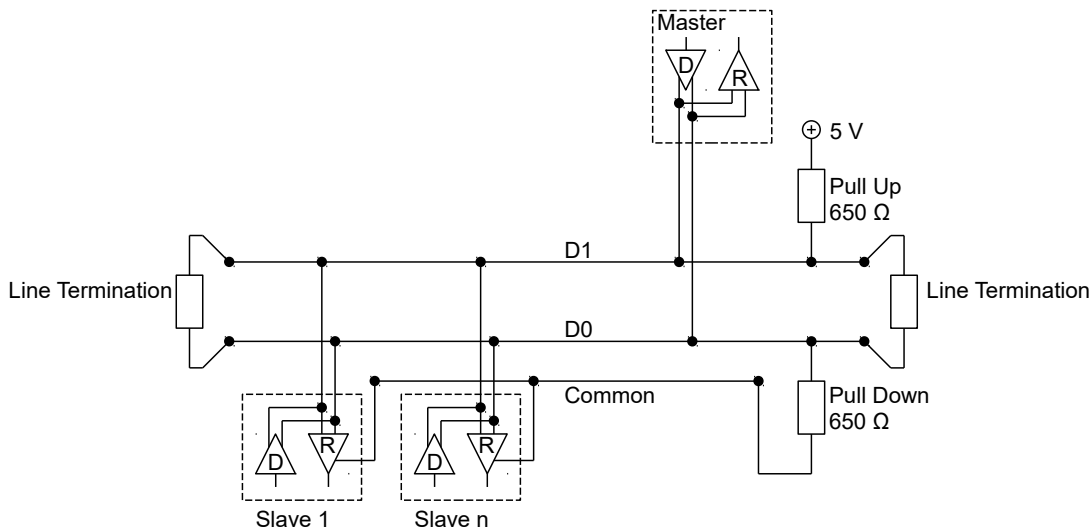
Die Steuerung ist **nicht** mit einer Leitungspolarisation ausgestattet und erwartet, dass das Master Gerät eine besitzt.

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

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

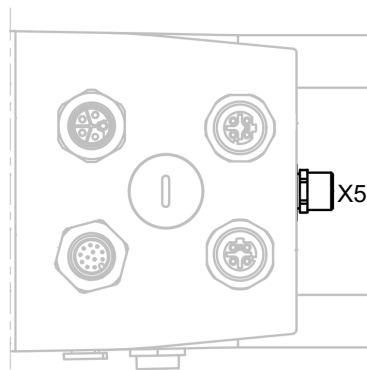
Der Wert dieser Widerstände muss zwischen 450 Ohm und 650 Ohm liegen. Ein 650 Ohm Widerstand erlaubt eine höhere Anzahl an Geräten am Bus.

In diesem Fall muss eine Leitungspolarisation an einer Stelle für den gesamten seriellen Bus angebracht werden. Generell sollte dieser Punkt an dem Master Gerät oder seinem Anschluss sein. Alle anderen Geräte müssen dann keine Leitungspolarisation mehr umsetzen.



### 3.6.6 X5 — externe Haltebremse

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

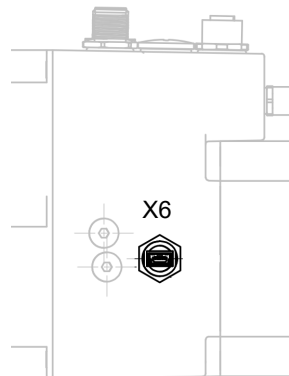


Pin	Funktion	Bemerkung
1	Brake	intern verbunden mit +UB
2	Brake GND	PWM-gesteuerter Open-Drain Ausgang, max 1,5 A
4	n.c.	nicht verbunden

Weitere Details zur Ansteuerung der Haltebremse finden Sie im Kapitel [Automatische Bremsensteuerung](#).

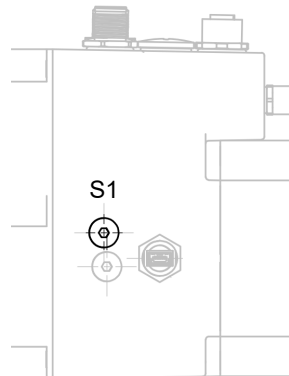
### 3.6.7 X6 – Mini USB

Mini-USB-Anschluss.



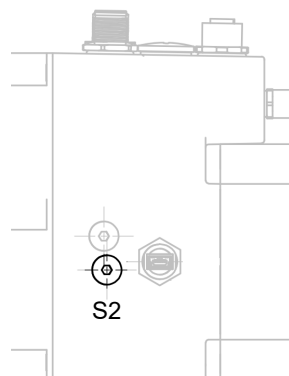
### 3.6.8 Schalter S1 - Hex-Codierschalter für Slave-Adresse und Baudrate

Über den Drehschalter S1 kann die Quelle für die Slave-Adresse und die Baudrate eingestellt werden. Siehe Kapitel [Kommunikationseinstellungen](#).



### 3.6.9 Schalter S2 – 150 Ohm Terminierungswiderstand

Der Schalter S2 schaltet die Terminierung von 120 Ohm zwischen RS485+ und RS485- zu (DIP-Schalter auf "ON", links) oder ab.





## 4 Inbetriebnahme

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

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

Beachten Sie folgende Hinweise:

### VORSICHT



#### **Bewegte Teile können zu Handverletzungen führen!**

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

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

### VORSICHT



#### **Motorbewegungen sind im freistehenden Betrieb unkontrolliert und können Verletzungen hervorrufen.**

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

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

### VORSICHT



#### **Bewegte Teile können Haare und lose Kleidung erfassen.**

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

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

### HINWEIS

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

Geeignete Maßnahmen können sein:



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

**HINWEIS****Störung des Motors und anderer Geräte durch elektromagnetische Wechselfelder!**

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



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

## 4.1 Konfiguration über USB

### 4.1.1 Allgemeines

Es gibt folgende Möglichkeiten, die Steuerung zu konfigurieren:

#### Konfigurationsdatei

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

#### NanoJ-Programm

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

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

1. Die Konfigurationsdatei wird ausgelesen und verarbeitet.
2. Der Hex-Codierschalter für die Auswahl der *speziellen Fahrmodi* wird ausgelesen und als Konfiguration angewendet. Siehe Kapitel [Spezielle Fahrmodi \(Takt-Richtung und Analog-Drehzahl\)](#).
3. Das NanoJ-Programm wird gestartet.

### 4.1.2 USB-Anschluss

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

Es werden drei Dateien angezeigt, die Konfigurationsdatei (`cfg.txt`), das *NanoJ-Programm* (`vmmcode.usr`) und die Informationsdatei (`info.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 Mini-USB-Kabel. Benutzen Sie keinesfalls USB-Kabel, die Hersteller von Mobiltelefonen ihren Produkten beilegen. Diese USB-Kabel können eine andere Steckerform oder Pin-Belegung aufweisen.
- Speichern Sie keine anderen Dateien auf der Steuerung als die nachfolgend aufgelisteten:



1. `cfg.txt`
2. `vmmcode.usr`
3. `info.bin`
4. `reset.txt`

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

### TIPP

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

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



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

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

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

## 4.1.3 Konfigurationsdatei

### 4.1.3.1 Allgemeines

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

### HINWEIS



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

### 4.1.3.2 Lesen und Schreiben der Datei

So erhalten Sie Zugriff auf die Datei:

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

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

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

#### TIPP



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

### 4.1.3.3 Aufbau der Konfigurationsdatei

#### Kommentare

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

#### Beispiel

```
; Dies ist eine Kommentarzeile
```

#### Zuweisungen

#### HINWEIS



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

Werte im Objektverzeichnis lassen sich mit folgender Syntax setzen:

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

#### <Index>

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

#### <Subindex>

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

#### <Wert>

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

Sie können auch einzelne Bits setzen:

#### Bit setzen

```
3202:00.03=1
```

**Bit zurücksetzen**

```
3202:00.03=0
```

**Bitweise OR**

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

**Bitweise AND**

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

**Beispiel**

Setzen des Objekts 203B<sub>h</sub>:01 (Nennstrom) auf den Wert "600" (mA):

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

Setzen des Objekts 3202<sub>h</sub>: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.

**4.1.4 NanoJ-Programm**

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

1. Schreiben und kompilieren Sie Ihr Programm, wie es in Kapitel Programmierung mit NanoJ beschrieben ist.
2. Schließen Sie die Spannungsversorgung an die Steuerung an und schalten Sie die Spannungsversorgung ein.
3. Verbinden Sie die Steuerung mit Ihrem PC über das USB-Kabel.
4. Nachdem der PC das Gerät als Wechseldatenträger erkannt hat, öffnen Sie einen Explorer und löschen Sie auf der Steuerung die Datei `vmmcode.usr`.

5. Navigieren Sie im Explorer in das Verzeichnis mit Ihrem Programm. Die compilierte Datei hat den gleichen Namen wie die Sourcecode-Datei, nur mit der Dateinamen-Endung `.usr`. Benennen Sie diese Datei in `vmmcode.usr` um.
6. Kopieren Sie die Datei `vmmcode.usr` auf die Steuerung.  
Um das *NanoJ-Programm* beim nächsten Neustart der Steuerung zu starten, fügen Sie folgende Zeile in die Konfigurationsdatei ein:

```
2300:00=1
```

7. Trennen Sie die Spannungsversorgung der Steuerung für ca. 1 Sekunde, bis die Betriebs-LED aufhört zu blinken.
8. Verbinden Sie die Spannungsversorgung wieder. Mit diesem Start der Steuerung wird das neue *NanoJ-Programm* eingelesen und gestartet.

#### TIPP



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

#### HINWEIS



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

#### TIPP

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

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

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



Also zum Beispiel:

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

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

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

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

## 4.2 Konfiguration über Modbus RTU

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

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

### 4.2.1 Kommunikationseinstellungen

Slave-Adresse, Baudrate und Parität ergeben sich abhängig von der Position des *Drehschalters* S1 und ggf. noch von den Objekten 2028<sub>h</sub>, 202A<sub>h</sub>, 202D<sub>h</sub>.

Konfiguration	Objekt	Wertebereich	Werkseinstellung
Slave Adresse	<u>2028<sub>h</sub></u>	1 bis 247	5
Baudrate	<u>202A<sub>h</sub></u>	7200 bis 256000	19200
Parity	<u>202D<sub>h</sub></u>	<ul style="list-style-type: none"> <li>■ None: 0x00</li> <li>■ Even: 0x04</li> <li>■ Odd: 0x06</li> </ul>	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

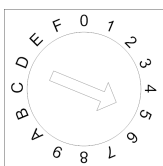
Unterstützt werden folgende Baudraten:

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

Sie müssen die Änderungen speichern, indem Sie den Wert "65766173<sub>h</sub>" in das Objekt 1010<sub>h</sub>:0B<sub>h</sub> schreiben. Die Änderungen werden erst nach einem Neustart der Steuerung übernommen.

#### 4.2.1.1 Drehschalter

Der *PD6-E* verfügt über einen Hex-Codierschalter - ähnlich wie in der nachfolgenden Abbildung.



Sie können damit die Quelle für die Slave-Adresse, die Baudrate und die Parität bestimmen, die nachfolgende Tabelle zeigt die möglichen Kombinationen.

Position des Drehschalters		Slave-Adresse	Baudrate und Parität
dec	hex		
0	0	Objekt <u>2028<sub>h</sub></u>	Objekt <u>202A<sub>h</sub></u> bzw. <u>202D<sub>h</sub></u>
1-14	1-E	Zahl des Drehschalters	Objekt <u>202A<sub>h</sub></u> bzw. <u>202D<sub>h</sub></u>
15	F	5	19200, even Parity

### 4.2.2 Kommunikation aufbauen

1. Verbinden Sie den *Modbus-Master* mit der Steuerung über die RS-485 + und RS-485- (siehe [X3 — Modbus RTU \(RS-485\) IN](#)) Leitungen.
2. Versorgen Sie die Steuerung mit Spannung.
3. Ändern Sie ggf. die Konfigurationswerte.  
Ab Werk (Drehschalter auf "1") ist die Steuerung ist auf Slave Address 1 , Baudrate 19200 Baud, even Parity, 1 Stop Bit eingestellt.
4. Zum Testen der Schnittstelle senden Sie die Bytes 01 65 55 00 2E 97 an die Steuerung (eine detaillierte Beschreibung der Modbus-Funktionscodes finden Sie im Kapitel [Modbus RTU](#)).  
Das Objektverzeichnis wird ausgelesen.

### 4.3 Konfigurieren der Sensoren

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

Objekt	Rückführung	Beschreibung
<a href="#">3380<sub>h</sub></a>	Sensorless	enthält Mess- und Konfigurations-Werte für die sensorlose Regelung
<a href="#">33A1<sub>h</sub></a>	Inkrementaler Encoder 2	enthält Konfigurationswerte für den zweiten ( <u>externen</u> ) inkrementalen Encoder

**HINWEIS**



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

In diesem Fall müssen Sie die Parameter in die entsprechenden Objekte (siehe [3204<sub>h</sub>](#), [60E6<sub>h</sub>](#) und [60EB<sub>h</sub>](#)) 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<sub>h</sub>](#) und [60ED<sub>h</sub>](#)) und/oder die Vorschubkonstante (Objekte [60E9<sub>h</sub>](#) und [60EE<sub>h</sub>](#)) einstellen und speichern (Kategorie *Applikation*).

**Beispiel**

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

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

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



Objekt	Wert
60E6h Additional Position Encoder Resolution - Encoder Increments	1256637
60EBh Additional Position Encoder Resolution - Motor Revolutions	20
60E8h Additional Gear Ratio - Motor Shaft Revolutions	4
60EDh Additional Gear Ratio - Driving Shaft Revolutions	1
60E9h Additional Feed Constant - Feed	2513274 Inkr. (entspricht 1256,637 mm )
60EEh Additional Feed Constant - Driving Shaft Revolutions	10

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

Im Objekt [3203<sub>h</sub>](#) 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<sub>h</sub>](#) 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<sub>h</sub>](#) ein Fehlercode hinterlegt.

## 4.4 Auto-Setup

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

#### TIPP



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

#### HINWEIS



**Beachten Sie die folgenden Voraussetzungen für das Durchführen des Auto-Setups:**

- ▶ Der Motor muss lastfrei sein.
- ▶ Der Motor darf nicht berührt werden.
- ▶ Der Motor muss sich frei in beliebige Richtungen drehen können.
- ▶ Es darf kein *NanoJ*-Programm laufen (Objekt [2300<sub>h</sub>](#):00<sub>h</sub> Bit 0 = "0", siehe [2300<sub>h</sub> 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.

### HINWEIS



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

## 4.4.1 Parameter-Ermittlung

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

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

### HINWEIS



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

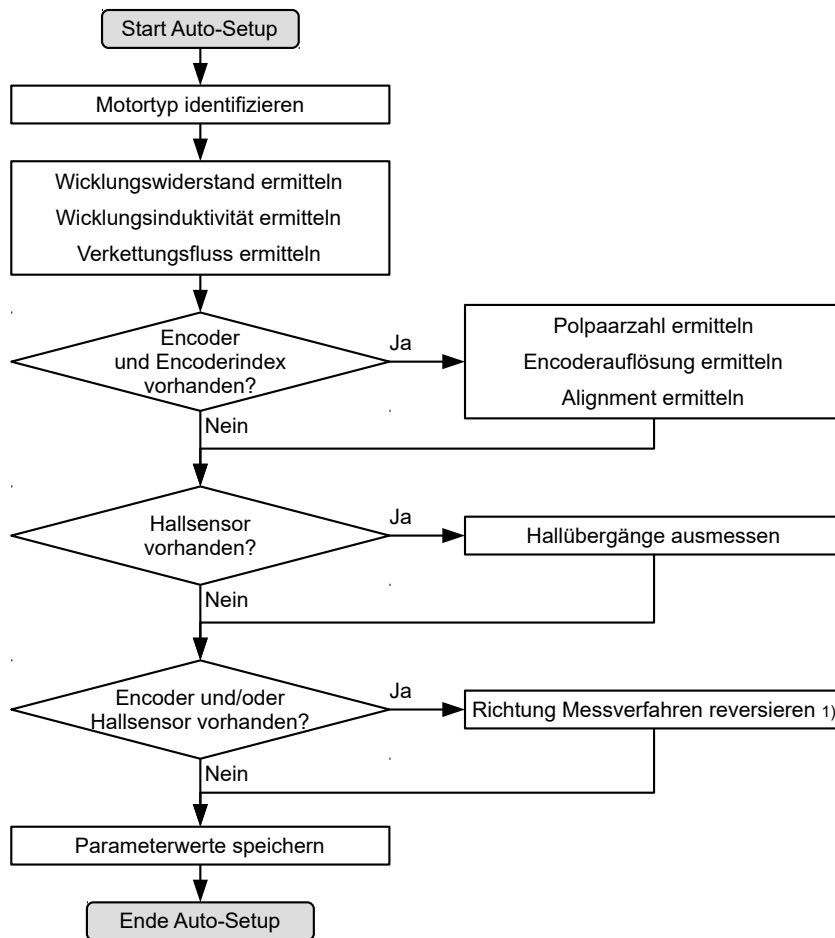
Parameter	Motor ohne Encoder	Motor mit Encoder und Index	Motor mit Encoder ohne Index
Encoderauflösung	-	✓	---
Alignment (Verschiebung des elektrischen Nullpunkts zum Index)	-	✓	---

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

## 4.4.2 Durchführung

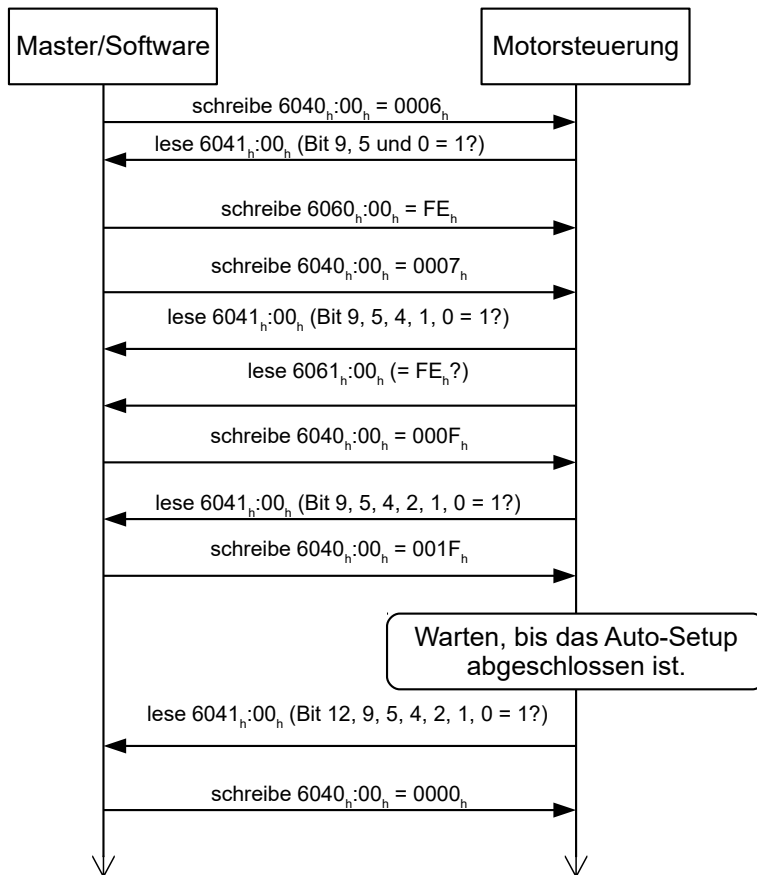
- Zum Vorwählen des Betriebsmodus *Auto-Setup* tragen Sie in das Objekt 6060<sub>h</sub>:00<sub>h</sub> den Wert "-2" (= "FE<sub>h</sub>") ein.  
Die *Power state machine* muss nun in den Zustand *Operation enabled* versetzt werden, siehe [CiA 402 Power State Machine](#).
- Starten Sie das *Auto-Setup* mit Setzen von Bit 4 *OMS* im Objekt 6040<sub>h</sub>:00<sub>h</sub> (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<sub>h</sub>:00<sub>h</sub> (Statusword) zeigt an, dass das Auto-Setup vollständig durchgeführt und beendet wurde. Zusätzlich kann über das Bit 10 TARG im Objekt 6041<sub>h</sub>:00<sub>h</sub> abgefragt werden, ob ein Encoder-Index gefunden wurde (= "1") oder nicht (= "0").



#### 4.4.3 Parameterspeicherung

Nach erfolgreichem *Auto-Setup* werden die ermittelten Parameterwerte automatisch in die zugehörigen Objekte übernommen und mit dem Speichermechanismus gespeichert, siehe [Objekte speichern](#) und [1010h Store Parameters](#). Benutzt werden die Kategorien *Drive* 1010<sub>h</sub>:05<sub>h</sub> und *Tuning* 1010<sub>h</sub>:06<sub>h</sub>.

### VORSICHT



#### Unkontrollierte Motorbewegungen!

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

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

#### 4.5 Spezielle Fahrmodi (Takt-Richtung und Analog-Drehzahl)

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

- [Takt-Richtung](#)
- [Analog-Drehzahl](#)
- [Automatische Fahrt mit 30 U/min \(Testfahrt\)](#)

Sie können ebenso die [Betriebsart](#), *Open Loop* oder *Closed Loop*, bestimmen.

Der digitale Eingang 1 dient dabei als Freigabe (siehe [X2 — Ein-/Ausgänge und Logikversorgung](#)).

### HINWEIS



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

#### 4.5.1 Aktivierung

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

Die folgende Tabelle listet alle möglichen Modi und den Wert für 4015:02<sub>h</sub> auf:

Wert	Modus			
00 <sub>h</sub> /01 <sub>h</sub>	Takt-Richtung	-	-	Open Loop
02 <sub>h</sub>	Takt-Richtung (Testfahrt)	Automatische Fahrt mit 30 U/min	Drehrichtung im Uhrzeigersinn	Open Loop
03 <sub>h</sub>	Takt-Richtung (Testfahrt)	Automatische Fahrt mit 30 U/min	Drehrichtung gegen den Uhrzeigersinn	Open Loop
04 <sub>h</sub>	Analog-Drehzahl	Richtung über "Richtungs"-Eingang	Maximale Drehzahl 1000 U/min	Open Loop
05 <sub>h</sub>	Analog-Drehzahl	Richtung über "Richtungs"-Eingang	Maximale Drehzahl 100 U/min	Open Loop
06 <sub>h</sub>	Analog-Drehzahl	Offset 5 V (Joystick-Modus)	Maximale Drehzahl 1000 U/min	Open Loop
07 <sub>h</sub>	Analog-Drehzahl	Offset 5 V (Joystick-Modus)	Maximale Drehzahl 100 U/min	Open Loop
08 <sub>h</sub> /09 <sub>h</sub>	Takt-Richtung	-	-	Closed Loop
0A <sub>h</sub>	Takt-Richtung (Testfahrt)	Automatische Fahrt mit 30 U/min	Drehrichtung im Uhrzeigersinn	Closed Loop
0B <sub>h</sub>	Takt-Richtung (Testfahrt)	Automatische Fahrt mit 30 U/min	Drehrichtung gegen den Uhrzeigersinn	Closed Loop
0C <sub>h</sub>	Analog-Drehzahl	Richtung über "Richtungs"-Eingang	Maximale Drehzahl 1000 U/min	Closed Loop
0D <sub>h</sub>	Analog-Drehzahl	Richtung über "Richtungs"-Eingang	Maximale Drehzahl 100 U/min	Closed Loop
0E <sub>h</sub>	Analog-Drehzahl	Offset 5 V (Joystick-Modus)	Maximale Drehzahl 1000 U/min	Closed Loop
0F <sub>h</sub>	Analog-Drehzahl	Offset 5 V (Joystick-Modus)	Maximale Drehzahl 100 U/min	Closed Loop

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

#### 4.5.2 Takt-Richtung

Die Steuerung setzt intern den Betriebsmodus auf *Takt-Richtung*. Sie müssen die Eingänge *Freigabe*, *Takt* und *Richtung* beschalten (siehe Kapitel *X2 — Ein-/Ausgänge und Logikversorgung*).

#### 4.5.3 Analog-Drehzahl

Die Steuerung setzt intern den Betriebsmodus auf *Velocity*. Zur Vorgabe der Drehzahl wird die Spannung am analogen Eingang benutzt und die entsprechende Zielgeschwindigkeit wird in 6042<sub>h</sub> geschrieben.

### 4.5.3.1 Maximale Drehzahl

Die maximale Drehzahl kann zwischen 100 U/min und 1000 U/min gewechselt werden, dabei passt die Steuerung die Skalierung in 604C<sub>h</sub> automatisch an.

#### HINWEIS



Sollten Sie danach in einen anderen Modus wechseln wollen, müssen Sie ggf. die Skalierung in 604C<sub>h</sub> anpassen bzw. zurücksetzen.

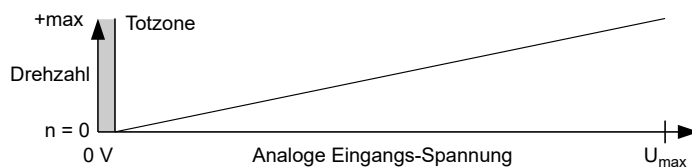
Ist eine andere Drehzahl notwendig, dann lässt sich diese über den Skalierungsfaktor für die Geschwindigkeit (Objekt 604C<sub>h</sub>) oder den Analogwert (siehe Analoge Eingänge) einstellen.

### 4.5.3.2 Verrechnung der Analogspannung

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

#### Normaler Modus

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



#### Joystick Modus

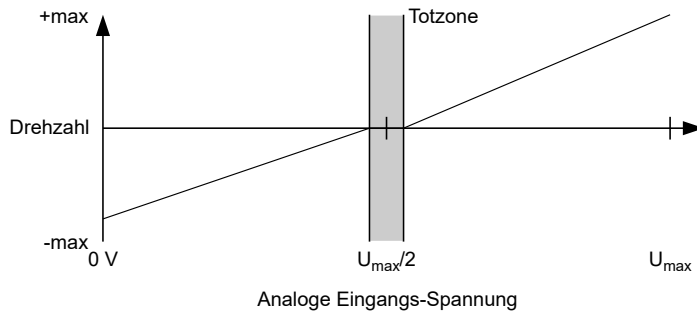
Sie müssen den Eingang *Freigabe* und den *Analogeingang* beschalten (siehe Kapitel X2 — Ein-/Ausgänge und Logikversorgung). Die Hälfte der maximalen, analogen Spannung entspricht der Drehzahl 0, dabei passt die Steuerung den Offset in 3321<sub>h</sub> automatisch an.

#### HINWEIS



Sollten Sie danach in einen anderen Modus wechseln wollen, müssen Sie ggf. den Offset in 3321<sub>h</sub> anpassen bzw. zurücksetzen.

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



#### 4.5.4 Automatische Fahrt mit 30 U/min (Testfahrt)

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

## 5 Generelle Konzepte

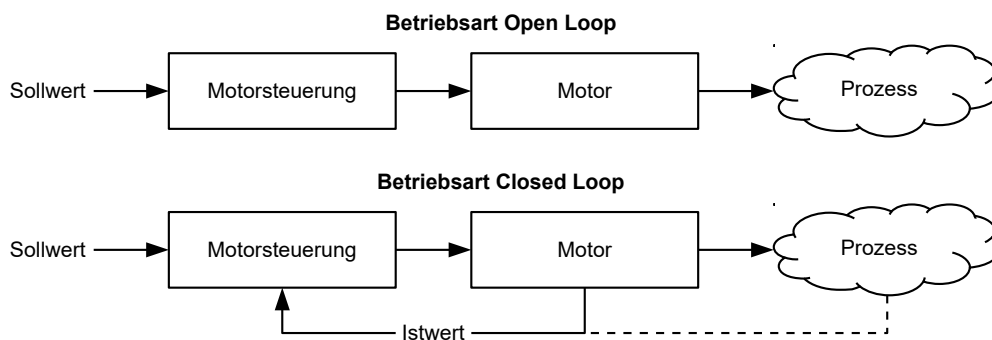
### 5.1 Betriebsarten

#### 5.1.1 Allgemein

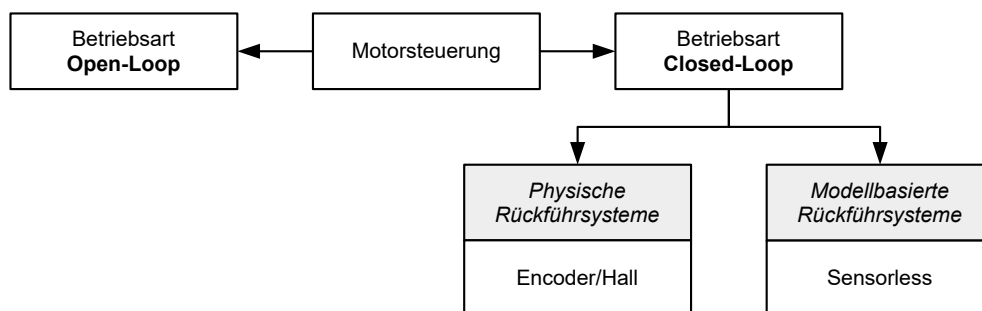
Die Betriebsart von Systemen ohne Rückführung wird als *Open Loop*, die mit Rückführung als *Closed Loop* bezeichnet. In der Betriebsart *Closed Loop* ist es zunächst unerheblich, ob die zurückgeführten Signale vom Motor selbst oder aus dem beeinflussten Prozess kommen.

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

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



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



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

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

Rückführung	Schrittmotor	BLDC-Motor
Hall	nein	ja



Rückführung	Schrittmotor	BLDC-Motor
Encoder	ja	ja
Sensorless	ja	ja

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

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

Betriebsmodus	Betriebsart		
	Open Loop	Closed Loop	Slow Speed
Profile Position	ja	ja	ja
Velocity	ja	ja	ja
Profile Velocity	ja	ja	ja
Profile Torque	nein <sup>1)</sup>	ja	nein
Homing	ja <sup>2)</sup>	ja	ja
Interpolated Position Mode	ja <sup>3)</sup>	ja	ja
Cyclic Synchronous Position	ja <sup>3)</sup>	ja	ja
Cyclic Synchronous Velocity	ja <sup>3)</sup>	ja	ja
Cyclic Synchronous Torque	nein <sup>1)</sup>	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.

## 5.1.2 Open Loop

### 5.1.2.1 Einführung

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

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

entgegen. Unter bestimmten Betriebsbedingungen neigt diese Kombination zu Resonanzen, vergleichbar einem Feder-Masse-System.

### 5.1.2.2 Inbetriebnahme

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

- Im Objekt 2030<sub>h</sub> (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<sub>h</sub>:00<sub>h</sub> den maximal zulässigen Motorstrom (Motorschutz) in mA eingeben (siehe Motordatenblatt)
- Im Objekt 6075<sub>h</sub>:00<sub>h</sub> den Nennstrom des Motors in mA (siehe Motordatenblatt) eingeben.
- Im Objekt 6073<sub>h</sub>:00<sub>h</sub>: 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<sub>h</sub> entspricht. Ein Wert größer "1000" wird intern auf "1000" limitiert.
- Im Objekt 3202<sub>h</sub> (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<sub>h</sub> (Motor Drive Submode Select) das Bit 3 (CurRed) auf "1" setzen.
- Im Objekt 2036<sub>h</sub> (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<sub>h</sub> (Open Loop Current Reduction Value/factor) wird der Effektivwert angegeben, auf den der Nennstrom reduziert werden soll, wenn die Stromabsenkung im *Open Loop* aktiviert wird und sich der Motor im Stillstand befindet.

### 5.1.2.3 Optimierungen

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

- Strom reduzieren oder erhöhen, siehe Objekt 6073<sub>h</sub> bzw. 6075<sub>h</sub>. 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<sub>h</sub>:09<sub>h</sub> (I\_P) und 3210<sub>h</sub>:0A<sub>h</sub> (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<sub>h</sub> (Profile Acceleration), 6084<sub>h</sub> (Profile Deceleration) und 6081<sub>h</sub> (Profile Velocity).

#### Betriebsmodus Velocity

Objekte 6048<sub>h</sub> (Velocity Acceleration), 6049<sub>h</sub> (Velocity Deceleration) und 6042<sub>h</sub> (Target Velocity).

#### Betriebsmodus Profile Velocity

Objekte 6083<sub>h</sub> (Profile Acceleration), 6084<sub>h</sub> (Profile Deceleration) und 6081<sub>h</sub> (Profile Velocity).

#### Betriebsmodus Homing

Objekte 609A<sub>h</sub> (Homing Acceleration), 6099<sub>h</sub>:01<sub>h</sub> (Speed During Search For Switch) und 6099<sub>h</sub>:02<sub>h</sub> (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

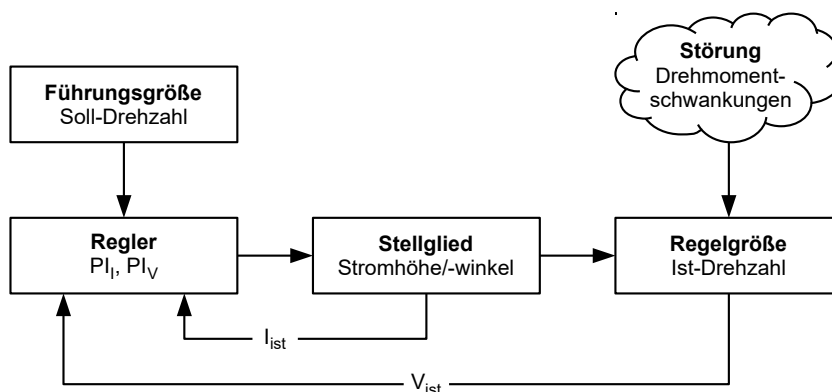
Änderung der Schrittauflösung über die Objekte 2057<sub>h</sub> (Clock Direction Multiplier) und 2058<sub>h</sub> (Clock Direction Divider). Beschleunigungs-/Verzögerungsrampen durch Anpassen der Impulsfrequenz optimieren, um den Resonanzbereich möglichst schnell zu durchlaufen.

## 5.1.3 Closed Loop

### 5.1.3.1 Einführung

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

*Closed Loop* am Beispiel einer Drehzahlregelung:



- PI<sub>I</sub> = Proportional-/Integralregler Stromregelkreis
- PI<sub>V</sub> = Proportional-/Integralregler Drehzahlregelkreis
- I<sub>ist</sub> = Aktueller Strom
- V<sub>ist</sub> = 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.

### 5.1.3.2 Reglerstruktur

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

Der Stromregler ist in allen Betriebsmodi aktiv. Der Geschwindigkeitsregler ebenso, mit der einzigen Ausnahme der "Real Torque"-Modi (Drehmomentmodus ohne Drehzahl-Begrenzung, wenn das Bit 5 in `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<sub>h</sub>:01<sub>h</sub></code>	Stromregler	[mV/A]	Proportional-Anteil momentbildende Komponente
	Proportional Gain $K_p$ for $I_q$		
<code>321A<sub>h</sub>:02<sub>h</sub></code>	Stromregler	[ $\mu$ s]	Integrierzeit momentbildenden Komponente
	Integrator Time $T_i$ for $I_q$		
<code>321A<sub>h</sub>:03<sub>h</sub></code>	Stromregler	[mV/A]	Proportional-Anteil feldbildende Komponente
	Proportional Gain $K_p$ for $I_d$		
<code>321A<sub>h</sub>:04<sub>h</sub></code>	Stromregler	[ $\mu$ s]	Integrierzeit feldbildende Komponente
	Integrator Time $T_i$ for $I_d$		
<code>321B<sub>h</sub>:01<sub>h</sub></code>	Geschwindigkeitsregler	[mA/Hz]	Proportional-Anteil
	Proportional Gain $K_p$		
<code>321B<sub>h</sub>:02<sub>h</sub></code>	Geschwindigkeitsregler	[ $\mu$ s]	Integrierzeit
	Integrator Time $T_i$		
<code>321C<sub>h</sub>:01<sub>h</sub></code>	Positionsregler	[Hz]	Proportional-Anteil
	Proportional Gain $K_p$		

Objekt	Name	Einheit	Beschreibung
		(Reglerabweichung in mech. Umdrehungen pro Sekunde)	
321C <sub>h</sub> :02 <sub>h</sub>	Positionsregler Integrator Time Ti	[μs]	Integrierzeit

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

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

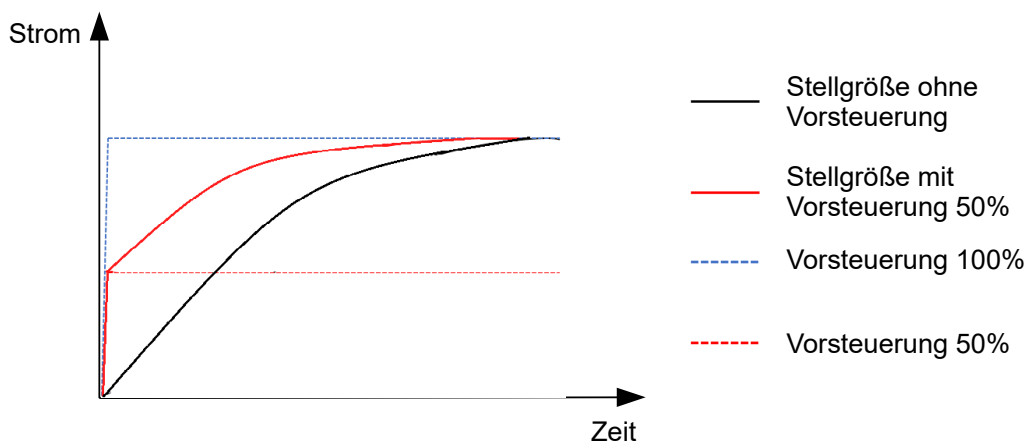
### 5.1.3.3 Vorsteuerung

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

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

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

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



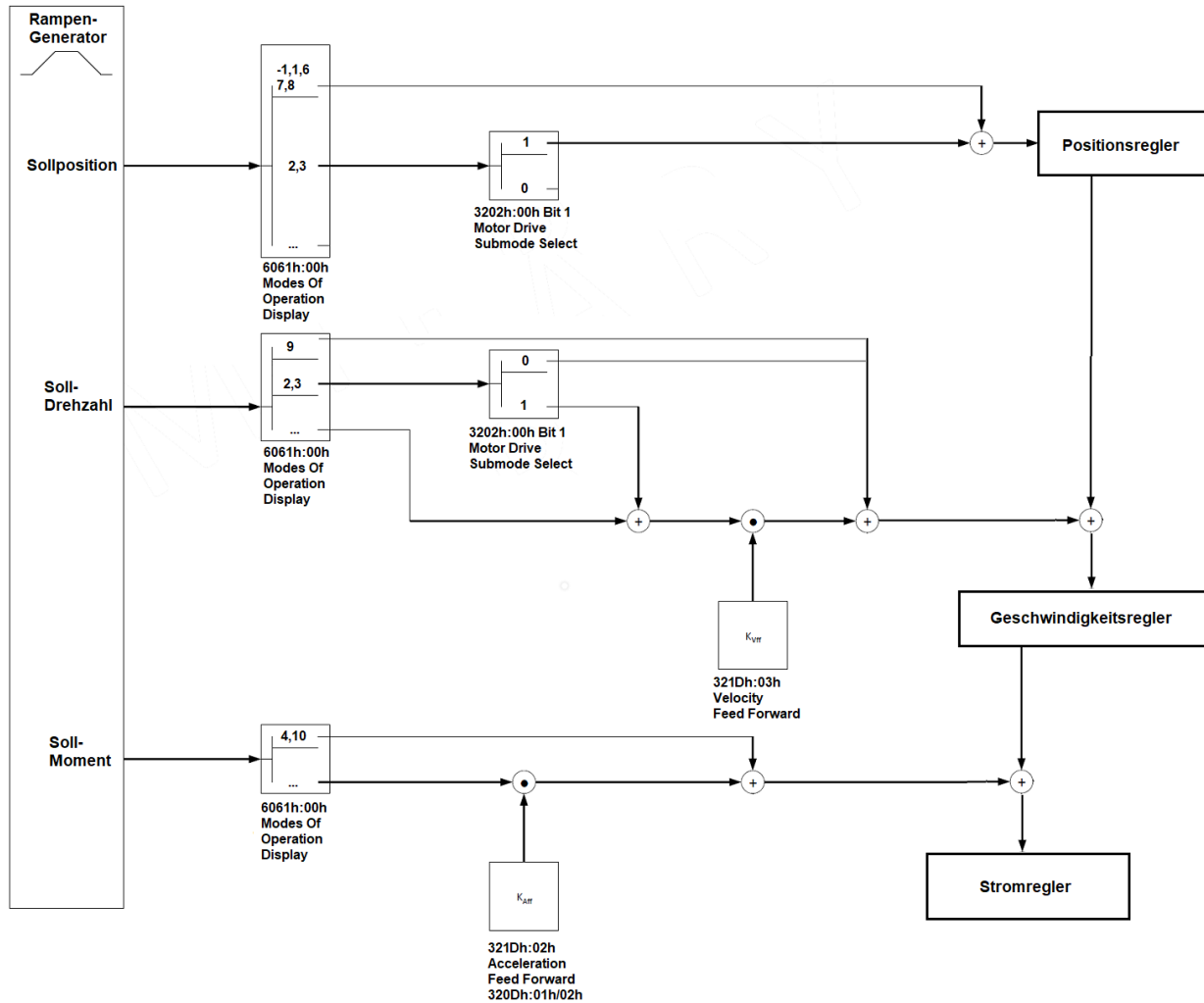
Der Faktor für die *Geschwindigkeitsvorsteuerung* wird im Objekt 321D<sub>h</sub>:03<sub>h</sub> in Promille des Ausgangs des Rampengenerators (606B<sub>h</sub>) eingestellt und vor dem Geschwindigkeitsregler zum Ausgang des Positionsreglers addiert. Die *Geschwindigkeitsvorsteuerung* ist in allen Modi mit Positionsregelkreis aktiv:

- Profile Position
- Homing
- Interpolated Position Mode
- Cyclic Synchronous Position

- Takt-Richtungs-Modus
- Velocity/Profile Velocity, wenn das Bit 1 in  $3202_h$  auf "1" steht

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

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



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

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

### 5.1.3.4 Zuordnung der Rückführungen zu den Regelkreisen

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

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

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

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

Subindex 01<sub>h</sub> 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 <sup>1</sup>
2	Kommutierung	0	1 <sup>2</sup>	1
<b>Index:Subindex</b>		3203 <sub>h</sub> :01 <sub>h</sub>	3203 <sub>h</sub> :02 <sub>h</sub>	3203 <sub>h</sub> :03 <sub>h</sub>

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

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

### Kommutierungshilfe

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

Als Hilfestellung können Sie einen zweiten Sensor als Kommutierungssensor einstellen (Bit 2 des entsprechenden Subindex in 3203<sub>h</sub>). 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<sub>h</sub>).

#### 5.1.3.5 Inbetriebnahme

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

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

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

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

### 5.1.3.6 Optimierungen

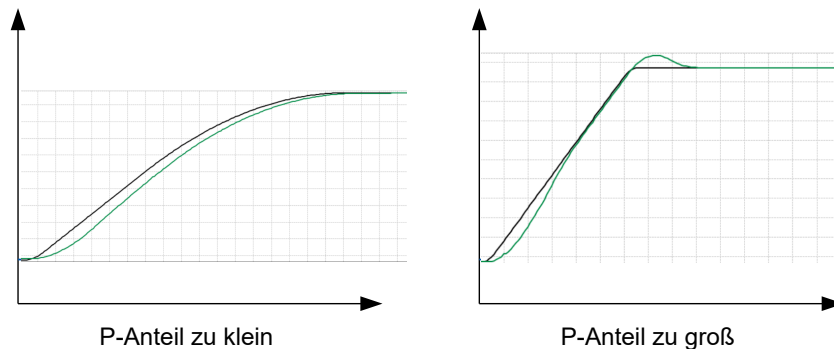
Im *Closed Loop* wird die gemessene Regelgröße (Istwert) permanent mit der Führungsgröße (Sollwert) verglichen. Bei Abweichungen zwischen diesen Größen regelt die Steuerung entsprechend den vorgegebenen Regelparametern nach.

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

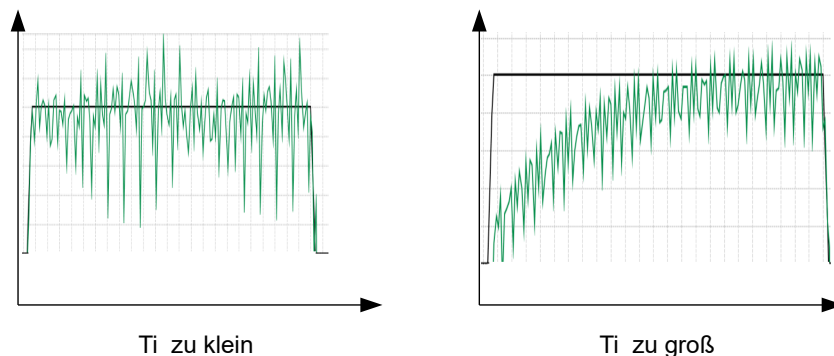
Es ist aufgrund der kaskadierten Reglerstruktur sinnvoll, mit der Optimierung des innersten Reglers (Stromreglers) zu beginnen, bevor der Geschwindigkeits- und ggf. der Positionsregler optimiert werden. Jeder der drei Regler besteht aus einem Proportional- und einem Integral-Anteil, die normalerweise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

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

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



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.

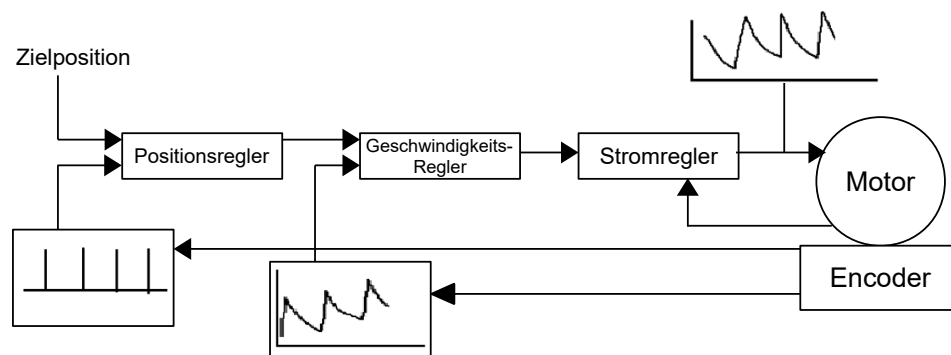
## 5.1.4 Slow Speed

### 5.1.4.1 Einführung

Die Betriebsart *Slow Speed* vereint die Vorteile der *Open Loop*- und *Closed Loop*-Technologie im niedrigen Drehzahlbereich und kann angewendet werden, wenn ein Encoder als Rückführung vorhanden ist. *Slow Speed* bietet eine Schleppfehlerüberwachung, weist aber einen laufruhigeren Betrieb auf, als im 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-Inkrememente 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 zuzufolge haben, wie die folgende Abbildung zeigt.



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

### 5.1.4.2 Aktivierung

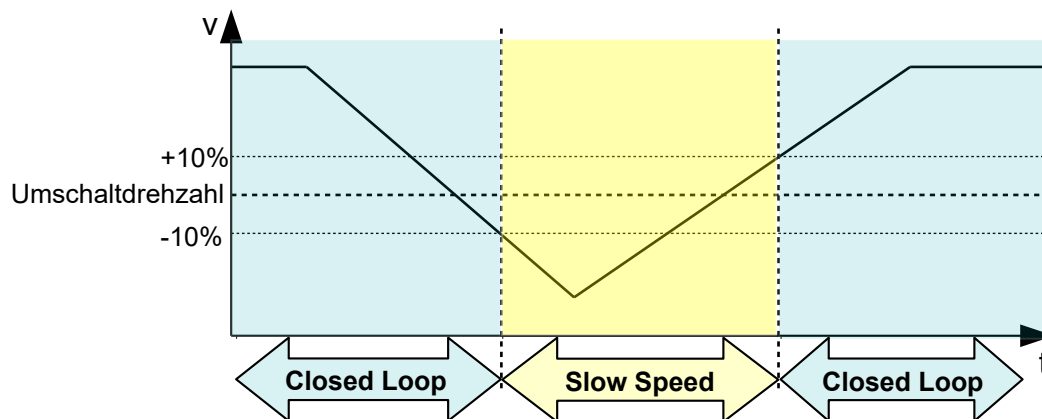
Um die Betriebsart *Slow Speed* zu aktivieren, müssen Sie:

1. den *Closed Loop* aktivieren,
2. das Bit 7 im Objekt `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 Umschaltdrehzahl wird in Umdrehungen pro Minute berechnet wie folgt:

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

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



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

### 5.1.4.3 Optimierungen

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

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

- den Motorstrom (Objekte [6075<sub>h</sub>](#), [6073<sub>h</sub>](#)) zu reduzieren, wenn von *Closed Loop* in *Slow Speed* gewechselt wird,
- unterschiedliche Reglerparameter (siehe [Reglerstruktur](#)) für jeden Drehzahlbereich zu ermitteln.

## 5.2 CiA 402 Power State Machine

### 5.2.1 Zustandsmaschine

#### 5.2.1.1 CiA 402

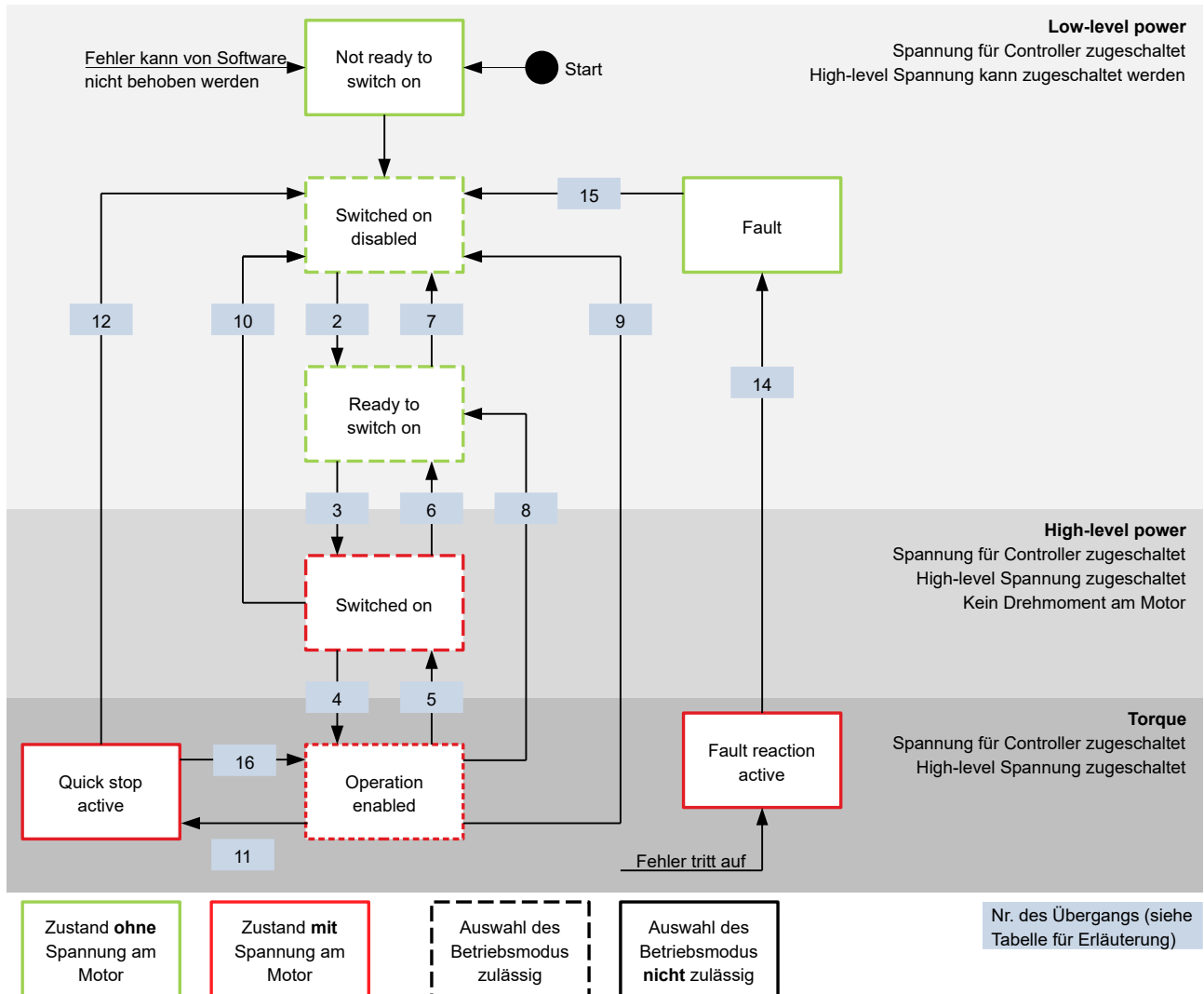
Um die Steuerung betriebsbereit zu schalten, ist es notwendig, eine Zustandsmaschine (*State Machine*) zu durchlaufen. Diese ist im *CANopen-Standard 402* definiert. Zustandsänderungen werden im Objekt [6040<sub>h</sub>](#) (Controlword) angefordert. Der tatsächliche Zustand der Zustandsmaschine lässt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

#### 5.2.1.2 Controlword

Zustandsänderungen werden über Objekt [6040<sub>h</sub>](#) (Controlword) angefordert.

### Zustandsübergänge

Das Diagramm zeigt die möglichen Zustandsübergänge.



In der nachfolgenden Tabelle sind die Bit-Kombinationen für das Controlword aufgelistet, die zu den entsprechenden Zustandsübergängen führen. Ein X entspricht dabei einem nicht weiter zu berücksichtigenden Bit-Zustand. Ausnahmen sind das Rücksetzen des Fehlers (Fault reset) und der Wechsel von *Quick Stop Active* nach *Operation Enabled*: Der Übergang wird nur durch steigende Flanke des Bits angefordert.

Kommando	Bit im Objekt 6040 <sub>h</sub>					Übergang
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Disable voltage	0	X	X	0	X	7, 10, 9, 12
Quick stop	0	X	0	1	X	11
Disable operation	0	0	1	1	1	5

Kommando	Bit im Objekt 6040 <sub>h</sub>					Übergang
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Enable operation	0	1	1	1	1	4
Enable operation after Quick stop	0	1		1	1	16
Fault / warning reset		X	X	X	X	15

### 5.2.1.3 Statusword

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

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

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

**HINWEIS**



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

### 5.2.1.4 Betriebsmodus

Der Betriebsmodus wird im Objekt 6060<sub>h</sub> eingestellt. Der tatsächlich aktive Betriebsmodus wird im 6061<sub>h</sub> angezeigt.

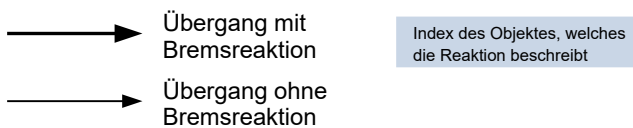
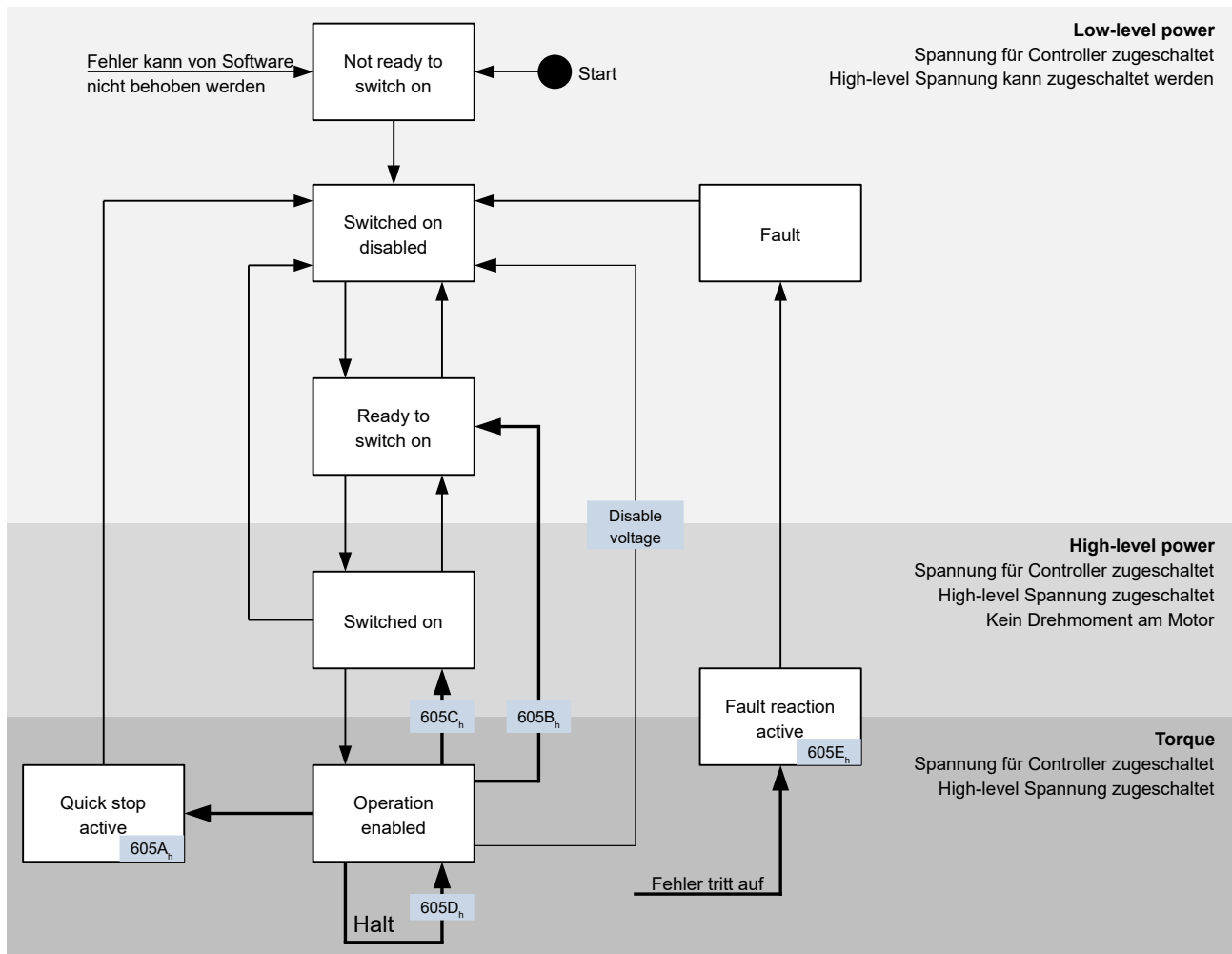
Die Einstellung oder Änderung des Betriebsmodus ist jederzeit möglich.

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

### 5.2.2.1 Bremsreaktionen

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

Die nachfolgende Grafik zeigt eine Übersicht der Bremsreaktionen.



### 5.2.2.2 Quick stop active

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

In diesem Fall wird die in Objekt 605A<sub>n</sub> hinterlegte Aktion ausgeführt (siehe nachfolgende Tabelle).

Wert in Objekt 605A <sub>n</sub>	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 <sub>n</sub> ) 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 <sub>h</sub>	Beschreibung
6	bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten. Abbremsen mit <i>quick stop ramp</i> (6085 <sub>h</sub> ) 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](#).

### 5.2.2.3 Ready to switch on

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

In diesem Fall wird die in Objekt 605B<sub>h</sub> hinterlegte Aktion ausgeführt (siehe nachfolgende Tabelle).

Wert in Objekt 605B <sub>h</sub>	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

### 5.2.2.4 Switched on

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

In diesem Fall wird die in Objekt 605C<sub>h</sub> hinterlegte Aktion ausgeführt (siehe nachfolgende Tabelle).

Wert in Objekt 605C <sub>h</sub>	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

### 5.2.2.5 Halt

Das Bit ist gültig in folgenden Modi:

- [Profile Position](#)
- [Velocity](#)
- [Profile Velocity](#)
- [Profile Torque](#)
- [Interpolated Position Mode](#)

Beim Setzen des Bit 8 in Objekt 6040<sub>h</sub> (Controlword) wird die in 605D<sub>h</sub> hinterlegte Aktion ausgeführt (siehe nachfolgende Tabelle):

Wert in Objekt 605D <sub>h</sub>	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 <sub>h</sub> )
3 bis 32767	Reserviert

### 5.2.2.6 Fault

Fehlerfall (fault):

Sollte ein Fehler auftreten, wird der Motor abgebremst, wie es in Objekt 605E<sub>h</sub> hinterlegt ist.

Wert in Objekt 605E <sub>h</sub>	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 <sub>h</sub> )
3 bis 32767	Reserviert

Für jeden aufgetretenen Fehler wird im Objekt 1003<sub>h</sub> ein genauerer Fehlercode hinterlegt.

### 5.2.2.7 Schlepp-/Schlupffehler

Sollte ein Schlepp- oder Schlupffehler auftreten, wird der Motor abgebremst, wie es in Objekt 3700<sub>h</sub> hinterlegt ist.

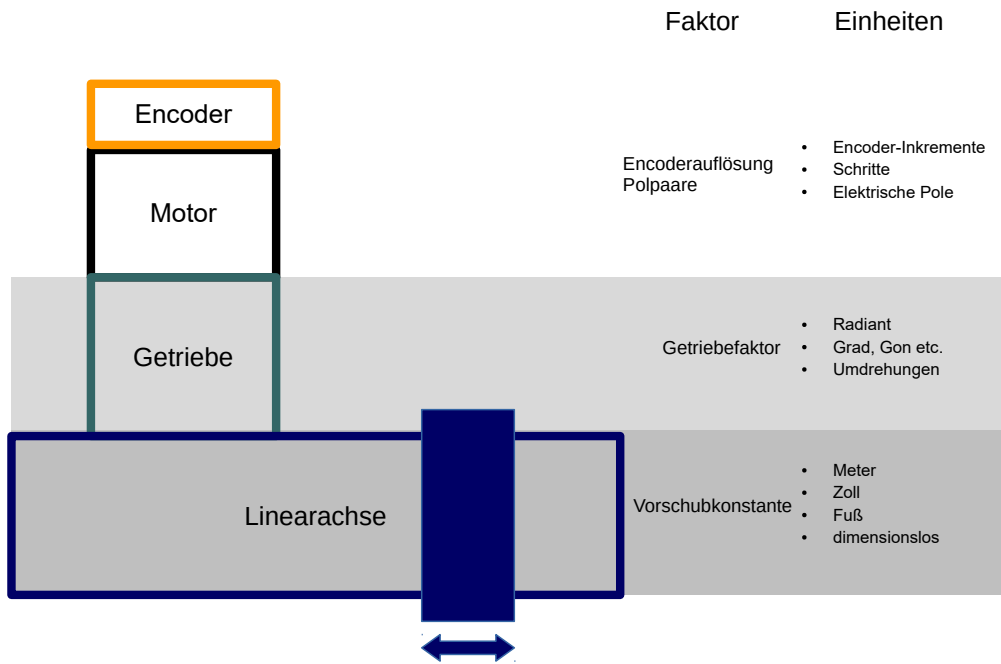
Wert	Beschreibung
-32768 bis -2	Reserviert
-1	keine Reaktion
0	Endstufe ohne Bremsrampe abschalten; Antriebsfunktion gesperrt — Motor kann sich frei drehen
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsmodus)
2	Abbremsen mit <i>quick stop ramp</i> (6085 <sub>h</sub> )
3 bis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

Sie können die Fehlerüberwachung deaktivieren, indem Sie das Objekt 6065<sub>h</sub> auf den Wert "-1" (FFFFFFFF<sub>h</sub>), bzw. das Objekt 60F8<sub>h</sub> auf den Wert "7FFFFFFFF<sub>h</sub>" setzen.

## 5.3 Benutzerdefinierte Einheiten

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

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



**HINWEIS**



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

### 5.3.1 Einheiten

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

In der nachfolgenden Tabelle sind alle unterstützten Einheiten für die Position und deren Werte für 60A8<sub>h</sub> (Positionseinheit) bzw. 60A9<sub>h</sub> (Geschwindigkeitseinheit) aufgelistet. Abhängig von der verwendeten Einheit wird die Vorschubkonstante (6092<sub>h</sub>) und/oder die Getriebeübersetzung (6091<sub>h</sub>) berücksichtigt.

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



Name	Einheitenzeichen	Wert	6091 <sub>h</sub>	6092 <sub>h</sub>	Beschreibung
encoder increment		B5 <sub>h</sub>	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 <sub>h</sub> ) multipliziert mit 65536 einer Motorumdrehung.
step		AC <sub>h</sub>	nein	nein	<i>Schritte</i> . Bei 2-phasigen Schrittmotoren entspricht die Anzahl der Polpaare (2030 <sub>h</sub> ) multipliziert mit 4 einer Umdrehung. Bei 3-phasigen BLDC-Motoren entspricht die Anzahl der Polpaare (2030 <sub>h</sub> ) multipliziert mit 6 einer Umdrehung.
electrical pole		C0 <sub>h</sub>	nein	nein	<i>Elektrische Pole</i> . Bei einem Schrittmotor, der z.B. 50 Polpaare (2030 <sub>h</sub> ) hat, entspricht die Einheit 1/50 einer Umdrehung.
dimensionless		00 <sub>h</sub>	ja	ja	<i>dimensionslose Längeneinheit</i>

In der nachfolgenden Tabelle sind alle unterstützten Einheiten für die Zeit und deren Werte für 60A9<sub>h</sub> (Geschwindigkeitseinheit) aufgelistet:

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

In der nachfolgenden Tabelle sind die möglichen Exponenten und deren Werte für 60A8<sub>h</sub> (Positionseinheit), bzw. 60A9<sub>h</sub> (Geschwindigkeitseinheit) aufgelistet:

Faktor	Exponent	Wert
10 <sup>6</sup>	6	06 <sub>h</sub>
10 <sup>5</sup>	5	05 <sub>h</sub>
...	...	...
10 <sup>1</sup>	1	01 <sub>h</sub>
10 <sup>0</sup>	0	00 <sub>h</sub>
10 <sup>-1</sup>	-1	FF <sub>h</sub>
...	..	...
10 <sup>-5</sup>	-5	FB <sub>h</sub>
10 <sup>-6</sup>	-6	FA <sub>h</sub>

### 5.3.2 Encoderauflösung

Die physikalische Auflösung des für die Positionsmessung verwendeten Encoders/Sensors berechnet sich aus den Encoder-Inkrementen (60E6<sub>h</sub> (Encoder Increments)) pro Motorumdrehungen (60EB<sub>h</sub> (Motor Revolutions)).

### 5.3.3 Getriebeübersetzung

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

### 5.3.4 Vorschubkonstante

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

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

### 5.3.5 Berechnungsformeln für Benutzereinheiten

#### 5.3.5.1 Positionseinheit

Das Objekt  $60A8_h$  enthält:

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

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

#### Beispiel

Wird  $60A8_h$  mit dem Wert "FF410000<sub>h</sub>" beschrieben (Bits 16-23=41<sub>h</sub> und Bits 24-31=FF<sub>h</sub>), 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<sub>h</sub>" beschrieben (Bits 16-23=01<sub>h</sub> und Bits 24-31=FD<sub>h</sub>(=-3)), wird die Einheit auf *Millimeter* eingestellt.

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

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

Im Kapitel [Zuordnung der Rückführungen zu den Regelkreisen](#) wird beschrieben, wie Sie festlegen, welcher Encoder/Sensor für die Positions-Regelung und -Messung verwendet werden soll. Wenn Sie keinen externen Encoder an X2 angeschlossen haben, wird der interne Absolut-Encoder verwendet.

#### 5.3.5.2 Geschwindigkeitseinheit

Das Objekt  $60A9_h$  enthält:

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

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

**Beispiel**

Wird  $60A9_h$  mit dem Wert "00B44700<sub>h</sub>" beschrieben (Bits 8-15=00<sub>h</sub>, Bits 16-23=B4<sub>h</sub> und Bits 24-31=47<sub>h</sub>), wird die Einheit auf *Umdrehungen pro Minute* eingestellt (Werkseinstellung).

**Beispiel**

Wird das  $60A9_h$  mit dem Wert "FD010300<sub>h</sub>" beschrieben (Bits 8-15=FD<sub>h</sub>(=-3), Bits 16-23=01<sub>h</sub> und Bis 24-31=03<sub>h</sub>), wird die Einheit auf *Millimeter pro Sekunde* eingestellt.

Im Kapitel Zuordnung der Rückführungen zu den Regelkreisen wird beschrieben, wie Sie festlegen, welcher Encoder/Sensor für die Geschwindigkeits-Regelung und -Messung verwendet werden soll. Wenn Sie keinen externen Encoder an X2 angeschlossen haben, wird der interne Absolut-Encoder verwendet.

**HINWEIS**



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

**Umrechnungsfaktor für die Geschwindigkeitseinheit**

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

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

**5.3.5.3 Beschleunigungseinheit**

Die Beschleunigungseinheit ist Geschwindigkeitseinheit pro Sekunde.

**Umrechnungsfaktor für die Beschleunigungseinheit**

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

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

**5.3.5.4 Ruckeinheit**

Die Ruckeinheit ist Beschleunigungseinheit pro Sekunde.

## Umrechnungsfaktor für den Ruck

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

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

## 5.4 Begrenzung des Bewegungsbereichs

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

### 5.4.1 Verhalten beim Erreichen der Endschalter

Wird ein Endschalter ausgelöst, so wird die Endschalterposition intern gespeichert, das Bit 7 (*Warning*) in  $6041_h$  (*Statusword*) gesetzt und die 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<sub>h</sub> kurzzeitig auf "-2".

Falls bei Verwendung der Werte "5" oder "6" in 3701<sub>h</sub>, 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<sub>h</sub> zurück in den Zustand *Operation Enabled*, ohne aber eine Fahrt zu starten (Bit 4 in 6040<sub>h</sub> auf 0 bzw. Zielgeschwindigkeit oder - Drehmoment auf "0" setzen).
2. Setzen Sie 3701<sub>h</sub> auf "-2" .
3. Geben Sie die Endschalter wieder frei.
4. Setzen Sie 3701<sub>h</sub> wieder auf "5" oder "6" zurück.

### 5.4.2 Software-Endschalter

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

## 5.5 Zykluszeiten

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

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

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

## 6 Betriebsmodi

### 6.1 Profile Position

#### 6.1.1 Übersicht

##### 6.1.1.1 Beschreibung

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

##### 6.1.1.2 Aktivierung

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

##### 6.1.1.3 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 <sub>h</sub>		
Bit 9	Bit 5	Definition
X	1	Die neue Zielposition wird sofort angefahren.
0	0	Das Positionieren wird erst vollständig abgeschlossen, bevor die nächste Zielposition mit den neuen Limitierungen angefahren wird.
1	0	Die momentane Zielposition wird nur durchfahren, danach wird die neue Zielposition mit den neuen Werten angefahren.

Siehe dazu das Bild in "[Setzen von Fahrbefehlen](#)".

#### HINWEIS



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

### 6.1.1.4 Statusword

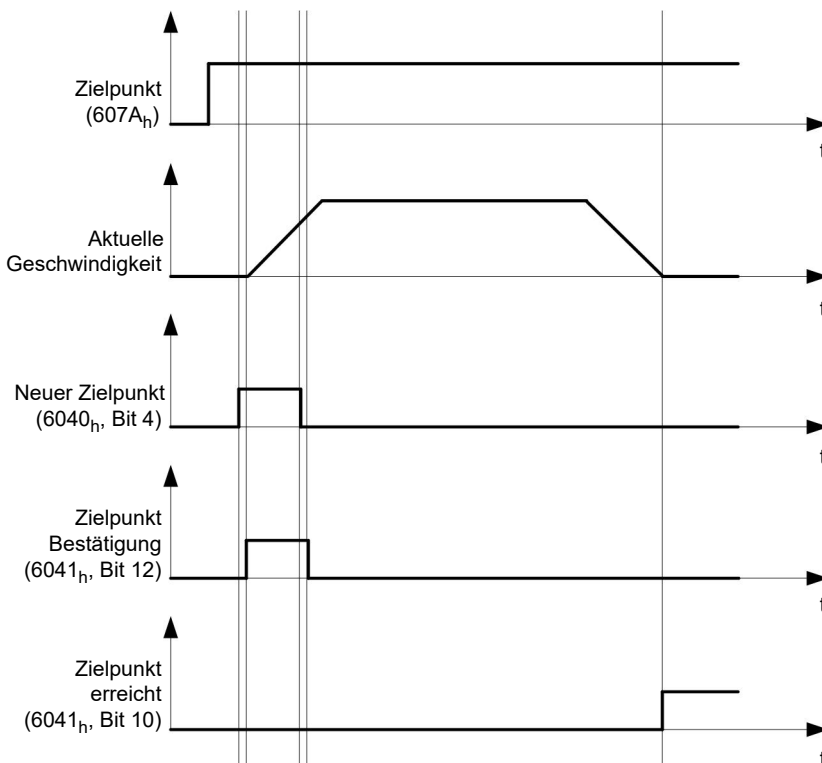
Folgende Bits im Objekt  $6041_h$  (Statusword) haben eine gesonderte Funktion:

- Bit 10 (Target Reached): Dieses Bit ist auf "1" gesetzt, wenn das letzte Ziel erreicht wurde und der Motor eine vorgegebene Zeit ( $6068_h$ ) innerhalb eines Toleranzfensters ( $6067_h$ ) steht. Das Bit wird auch auf "1" gesetzt, wenn das Halt-Bit (Bit 8) in  $6040_h$  gesetzt wurde und sobald der Motor sich im Stillstand befindet.
- Bit 11: Limit überschritten: Die Sollposition über- oder unterschreitet die in  $607D_h$  eingegebenen Grenzwerte.
- Bit 12 (Set-point acknowledge): Dieses Bit bestätigt den Erhalt eines neuen und gültigen Zielpunktes. Es wird synchron zu dem Bit "New set-point" im Controlword gesetzt und zurückgesetzt. Eine Ausnahme besteht, wenn eine neue Fahrt gestartet wird, während eine andere noch nicht abgeschlossen ist, und die nächste Fahrt erst nach dem Abschluss der ersten Fahrt ausgeführt werden soll. In diesem Fall wird das Bit erst zurückgesetzt, wenn der Befehl angenommen wurde und die Steuerung bereit ist, neue Fahrbefehle auszuführen. Wird ein neuer Fahrauftrag gesendet, obwohl dieses Bit noch gesetzt ist, wird der neueste Fahrauftrag ignoriert. Das Bit wird nicht gesetzt, wenn eine der folgenden Bedingungen erfüllt ist:
  - Die neue Zielposition kann unter Einhaltung aller Randbedingungen nicht mehr erreicht werden.
  - Es wird bereits eine Zielposition angefahren und zudem ist bereits eine Zielposition vorgegeben. Eine neue Zielposition lässt sich erst vorgeben, nachdem die aktuelle Positionierung abgeschlossen ist.
- Bit 13 (Following Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schleppfehler größer als die eingestellten Grenzen ist ( $6065_h$  (Following Error Window) und  $6066_h$  (Following Error Time Out)).

## 6.1.2 Setzen von Fahrbefehlen

### 6.1.2.1 Fahrbefehl

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



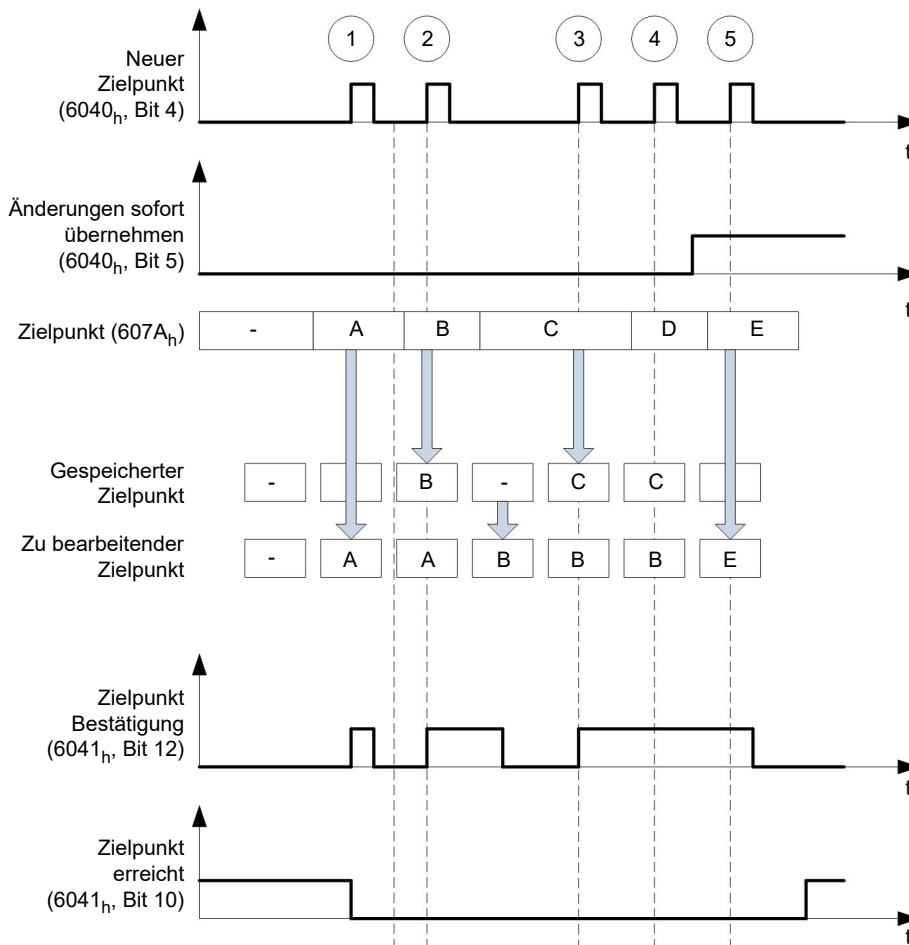
Die Steuerung kann das Bit 4 im Objekt  $6040_h$  (Controlword) auch selbstständig zurücksetzen. Das wird mit den Bits 4 und 5 des Objektes  $60F2_h$  eingestellt.

### 6.1.2.2 Weitere Fahrbefehle

Bit 12 im Objekt 6041<sub>h</sub> (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<sub>h</sub> (Controlword, Bit: "Change Set-Point Immediately") gesetzt, arbeitet die Steuerung ohne den Zwischenspeicher, neue Fahrbefehle werden direkt umgesetzt (Zeitpunkt 5).

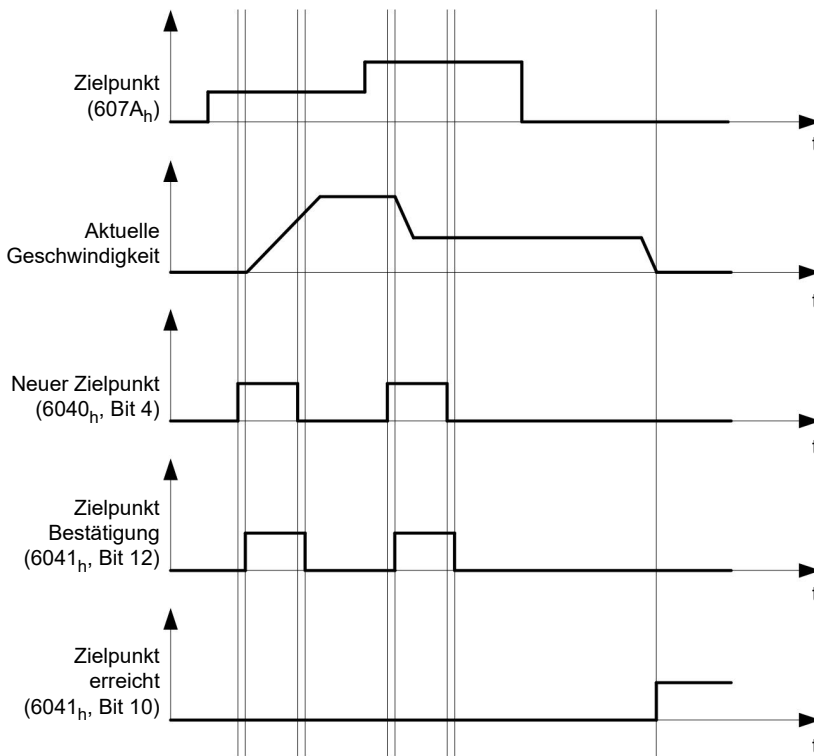
#### Zeitpunkte



#### Übergangsprozedur für zweite Zielposition

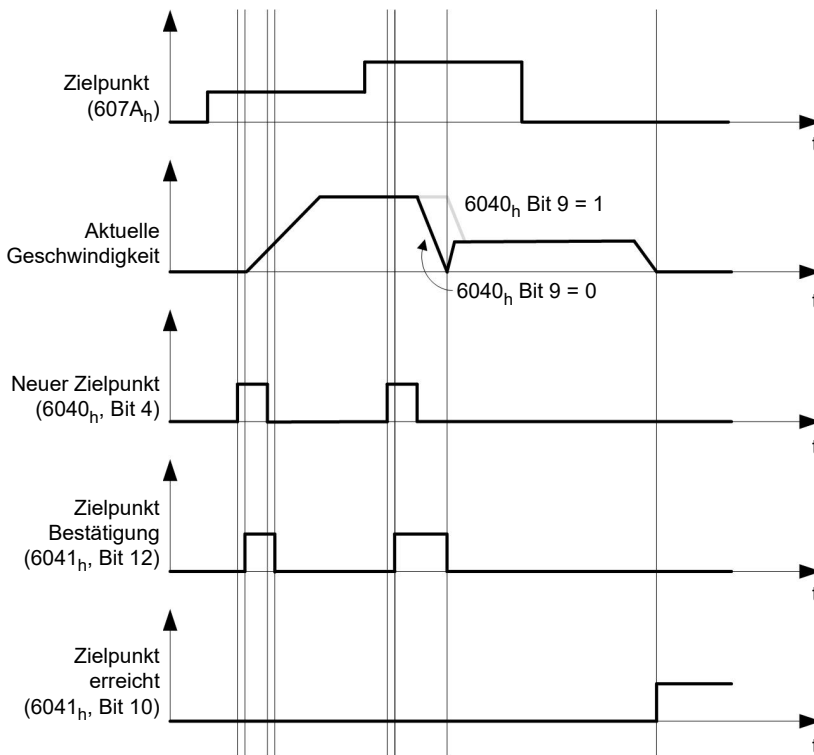
Die folgende Grafik zeigt die Übergangsprozedur für die zweite Zielposition, während die erste Zielposition angefahren wird. In dieser Abbildung ist Bit 5 von Objekt 6040<sub>h</sub> (Controlword) auf "1" gesetzt, der neue Zielwert wird demnach sofort übernommen.





### Möglichkeiten zum Anfahren einer Zielposition

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



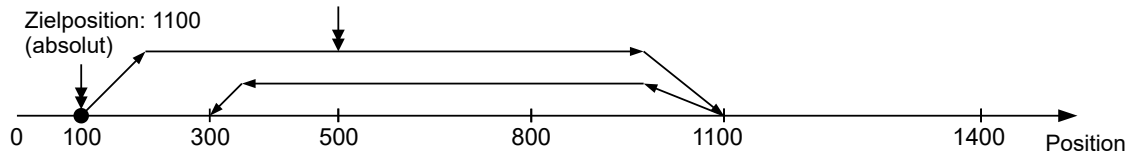
### Mögliche Kombinationen von Fahrbefehlen

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

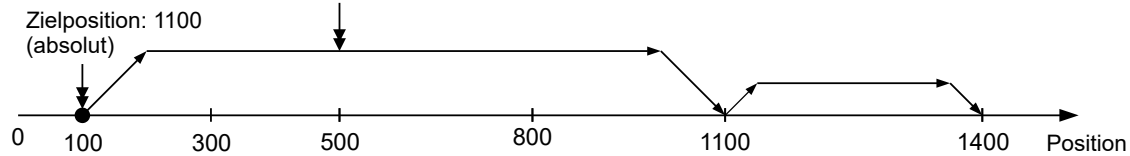
Die nachfolgenden Bilder setzen voraus:

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

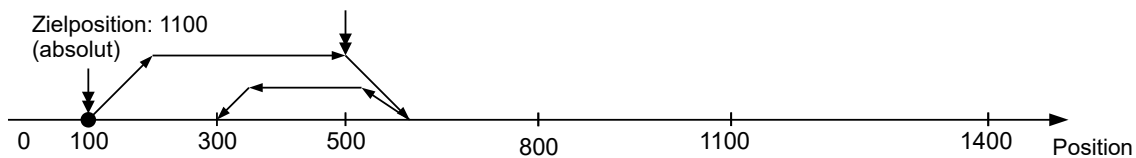
- Änderung im Zielpunkt übernehmen ( $6040_n:00$  Bit 5 = 0)
- Positionierung absolut ( $6040_n:00$  Bit 6 = 0)
- Zielposition: 300



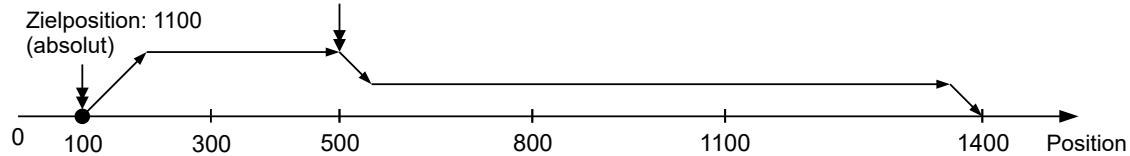
- Relativ zu der vorhergehenden Zielposition ( $60F2_n:00 = 0$ )
- Änderung im Zielpunkt übernehmen ( $6040_n:00$  Bit 5 = 0)
- Positionierung relativ ( $6040_n:00$  Bit 6 = 1)
- Zielposition: 300



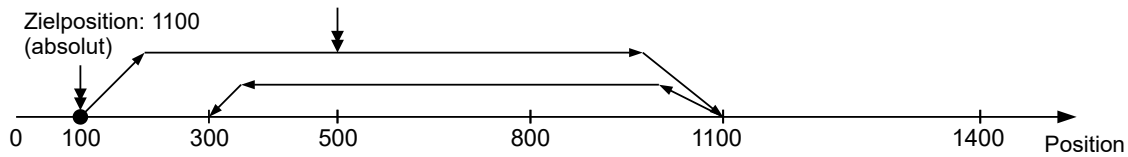
- Änderung sofort übernehmen ( $6040_n:00$  Bit 5 = 1)
- Positionierung absolut ( $6040_n:00$  Bit 6 = 0)
- Zielposition: 300



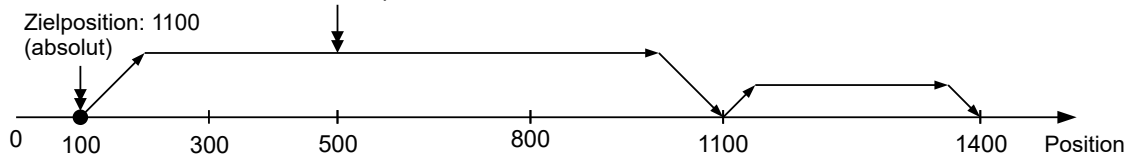
- Relativ zu der vorhergehenden Zielposition ( $60F2_n:00 = 0$ )
- Änderung sofort übernehmen ( $6040_n:00$  Bit 5 = 1)
- Positionierung relativ ( $6040_n:00$  Bit 6 = 1)
- Zielposition: 300



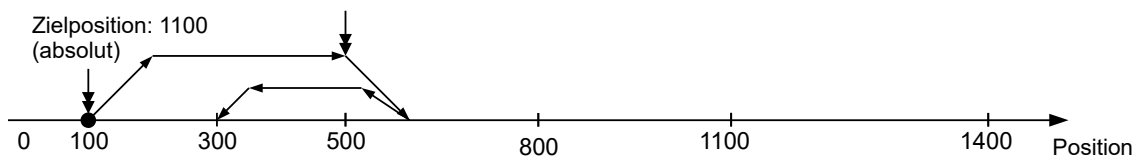
- Änderung im Zielpunkt übernehmen (6040<sub>n</sub>:00 Bit 5 = 0)
- Positionierung absolut (6040<sub>n</sub>:00 Bit 6 = 0)
- Zielposition: 300



- Relativ zu der aktuellen Position (60F2<sub>n</sub>:00 = 1)
- Änderung im Zielpunkt übernehmen (6040<sub>n</sub>:00 Bit 5 = 0)
- Positionierung relativ (6040<sub>n</sub>:00 Bit 6 = 1)
- Zielposition: 300

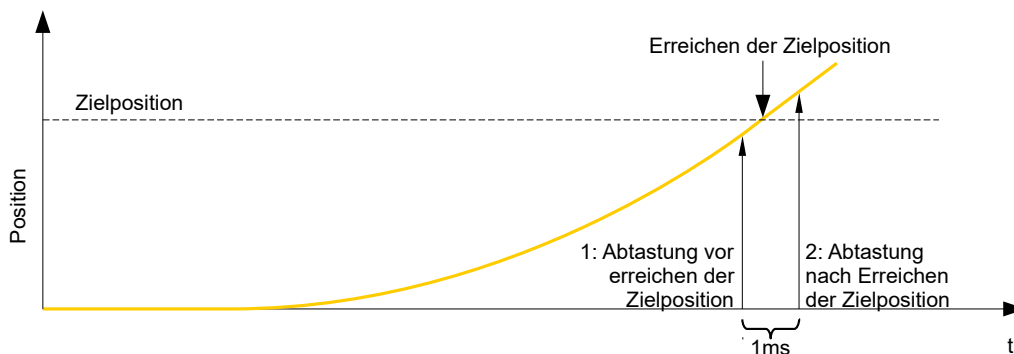


- Änderung sofort übernehmen (6040<sub>n</sub>:00 Bit 5 = 1)
- Positionierung absolut (6040<sub>n</sub>:00 Bit 6 = 0)
- Zielposition: 300



### 6.1.3 Genauigkeitsverlust bei Relativbewegungen

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



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

## 6.1.4 Randbedingungen für eine Positionierfahrt

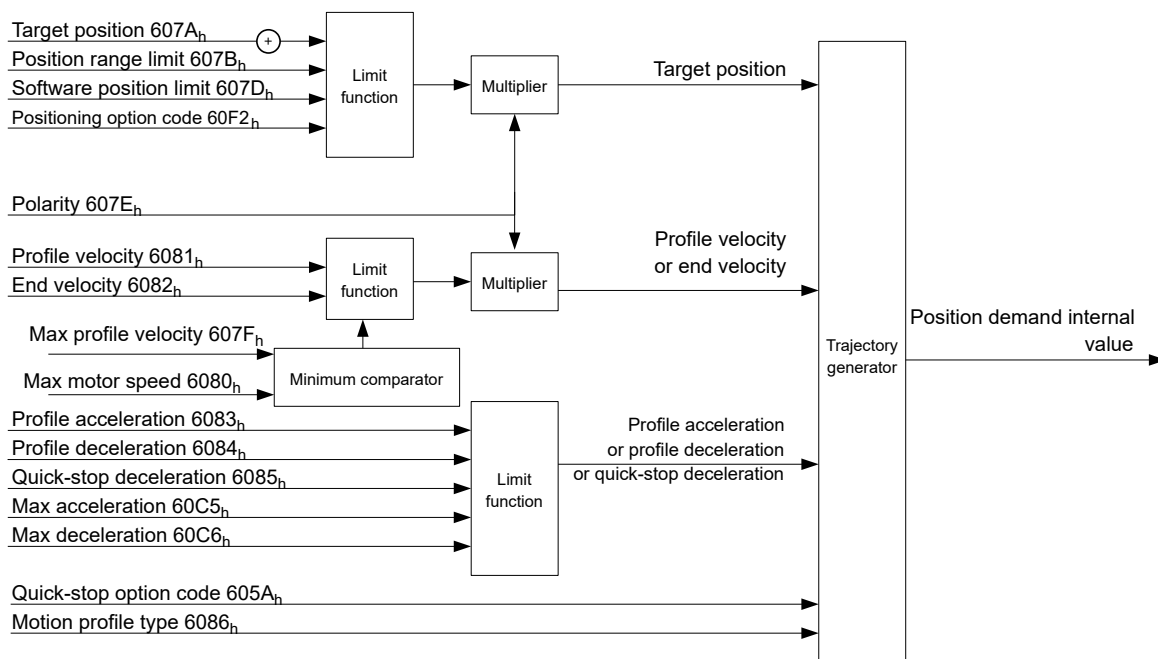
### 6.1.4.1 Objekteinträge

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

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

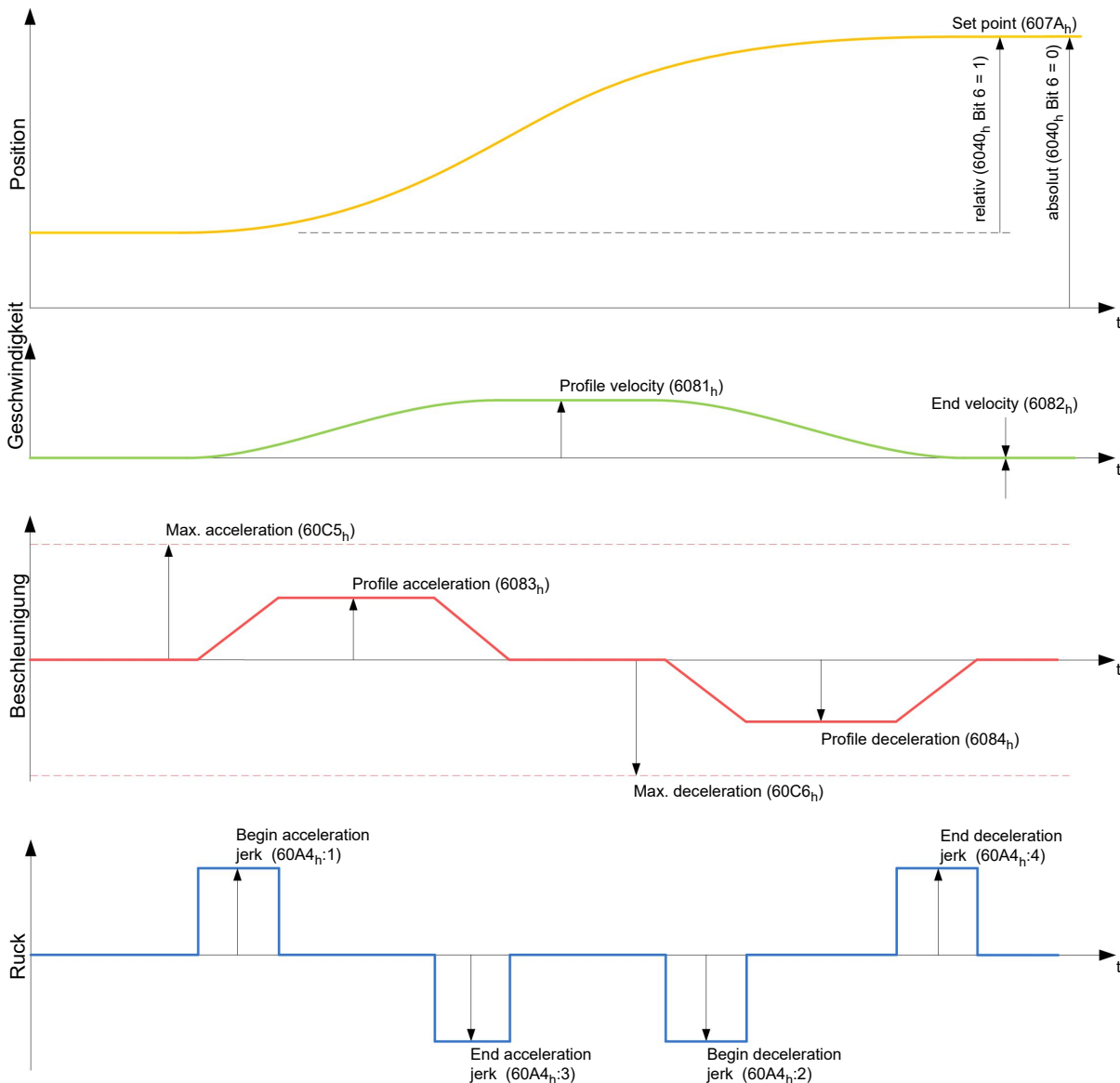
### 6.1.4.2 Objekte für die Positionierfahrt

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



### 6.1.4.3 Parameter für die Zielposition

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



## 6.1.5 Ruck-begrenzter und nicht ruck-begrenzter Modus

### 6.1.5.1 Beschreibung

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

### 6.1.5.2 Ruck-begrenzter Modus

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

### 6.1.5.3 Nicht ruck-begrenzter Modus

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

## 6.2 Velocity

### 6.2.1 Beschreibung

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

### 6.2.2 Aktivierung

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

### 6.2.3 Controlword

Folgende Bits im Objekt 6040<sub>h</sub> (Controlword) haben eine gesonderte Funktion:

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

### 6.2.4 Statusword

Folgende Bits im Objekt 6041<sub>h</sub> (Statusword) haben eine gesonderte Funktion:

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

### 6.2.5 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- 604C<sub>h</sub> (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<sub>h</sub>: Target Velocity.  
Hier wird die Zielgeschwindigkeit in benutzerdefinierten Einheiten eingestellt.
- 6048<sub>h</sub>: 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:

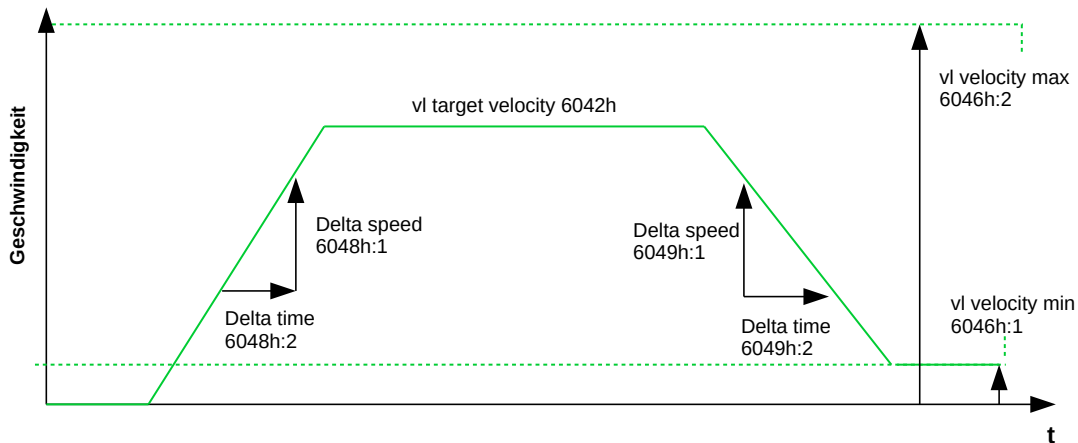
$$\text{VL velocity acceleration} = \frac{\text{Delta speed (6048}_{h}:1)}{\text{Delta time (6048}_{h}:2)}$$

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

Folgende Objekte können zur Kontrolle der Funktion genutzt werden:

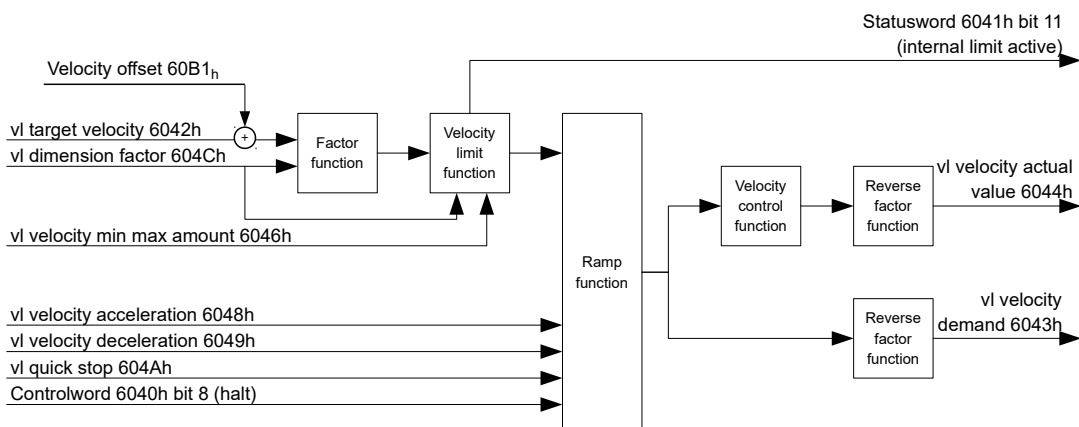
- 6043<sub>h</sub> (VI Velocity Demand)
- 6044<sub>h</sub> (VI Velocity Actual Value)

### 6.2.5.1 Geschwindigkeiten im Velocity Mode



### 6.2.5.2 Objekte für den Velocity Mode

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



## 6.3 Profile Velocity

### 6.3.1 Beschreibung

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

### 6.3.2 Aktivierung

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

### 6.3.3 Controlword

Folgende Bits im Objekt 6040<sub>h</sub> (Controlword) haben eine gesonderte Funktion:

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

### 6.3.4 Statusword

Folgende Bits im Objekt 6041<sub>h</sub> (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).

<b>6041<sub>h</sub> Bit 10</b>	<b>6040<sub>h</sub> Bit 8</b>	<b>Beschreibung</b>
0	0	Zielgeschwindigkeit nicht erreicht
0	1	Achse bremst
1	0	Zielgeschwindigkeit innerhalb Zielfenster (definiert in <u>606D<sub>h</sub></u> und <u>606E<sub>h</sub></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<sub>h</sub> (Velocity Threshold) für eine Zeit von 6070<sub>h</sub> (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<sub>h</sub> Max Slippage und 203F<sub>h</sub> Max Slippage Time Out).

### 6.3.5 Objekteinträge

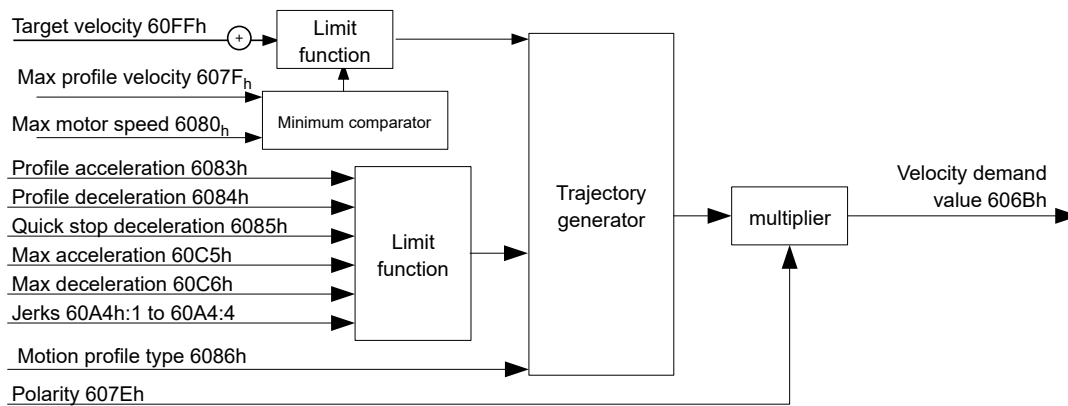
Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- 606B<sub>h</sub> (Velocity Demand Value): Dieses Objekt enthält die Ausgabe des Rampengenerators, die gleichzeitig der Vorgabewert für den Geschwindigkeitsregler ist.
- 606C<sub>h</sub> (Velocity Actual Value): Gibt die aktuelle Istgeschwindigkeit an.
- 606D<sub>h</sub> (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<sub>h</sub> (Statusword) auf "1" gesetzt ist.
- 606E<sub>h</sub> (Velocity Window Time): Dieses Objekt gibt an, wie lange die reale Geschwindigkeit und die Sollgeschwindigkeit nahe beieinander liegen müssen (siehe 606D<sub>h</sub> "Velocity Window"), damit Bit 10 "Zielgeschwindigkeit erreicht" im Objekt 6041<sub>h</sub> (Statusword) auf "1" gesetzt wird.
- 607E<sub>h</sub> (Polarity): Wird hier Bit 6 auf "1" gestellt, wird das Vorzeichen der Zielgeschwindigkeit umgekehrt.
- 6083<sub>h</sub> (Profile acceleration): Setzt den Wert für die Beschleunigungsrampe.
- 6084<sub>h</sub> (Profile Deceleration): Setzt den Wert für die Bremsrampe.
- 6085<sub>h</sub> (Quick Stop Deceleration): Setzt den Wert für die Bremsrampe für die Schnellbremsung.
- 6086<sub>h</sub> (Motion Profile Type): Hier kann der Rampentyp ausgewählt werden ("0" = Trapez-Rampe, "3" = ruck-begrenzte Rampe).
- 60FE<sub>h</sub> (Target Velocity): Gibt die zu erreichende Zielgeschwindigkeit an.



- Die Geschwindigkeit wird durch  $607F_h$  (Max Profile Velocity) und  $6080_h$  (Max Motor Speed) begrenzt, der kleinere Wert wird als Grenze herangezogen.

### 6.3.5.1 Objekte im Profile Velocity Mode

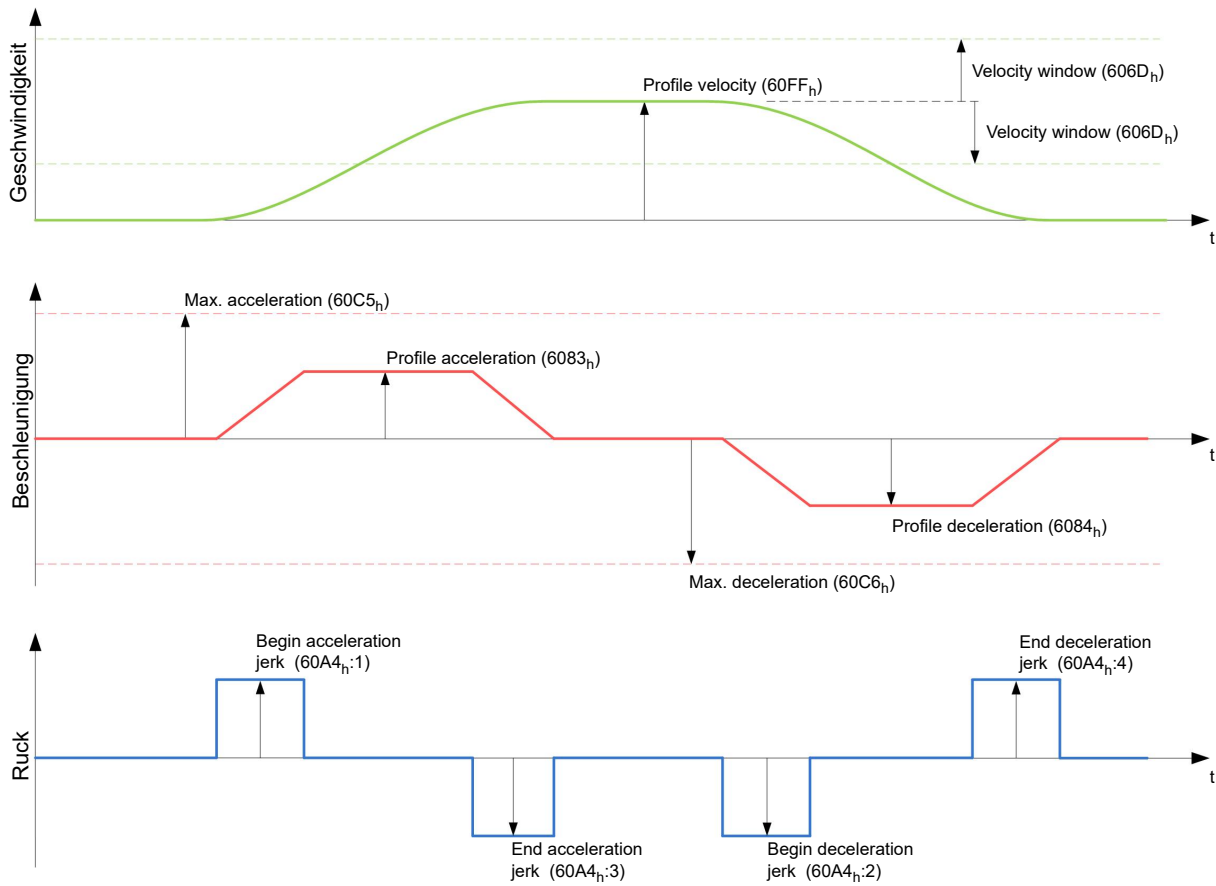


### 6.3.5.2 Aktivierung

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

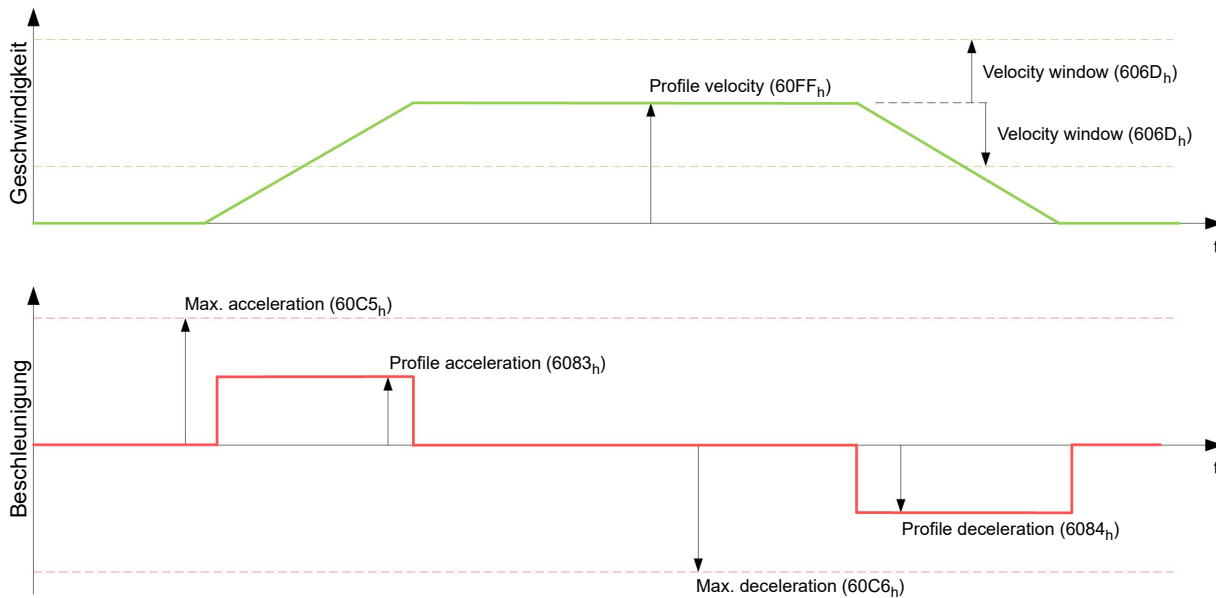
### 6.3.5.3 Limitierungen im ruck-limitierten Fall

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



### 6.3.5.4 Limitierungen im Trapez-Fall

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



## 6.4 Profile Torque

### 6.4.1 Beschreibung

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

### HINWEIS



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

## 6.4.2 Aktivierung

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

## 6.4.3 Controlword

Folgende Bits im Objekt 6040<sub>h</sub> (Controlword) haben eine gesonderte Funktion:

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

## 6.4.4 Statusword

Folgende Bits im Objekt 6041<sub>h</sub> (Statusword) haben eine gesonderte Funktion:

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

<u>6040<sub>h</sub></u> Bit 8	<u>6041<sub>h</sub></u> 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<sub>h</sub>) überschreitet das in 6072<sub>h</sub> eingegebene maximale Drehmoment.

## 6.4.5 Objekteinträge

Alle Werte der folgenden Einträge im Objektverzeichnis sind als Tausendstel des maximalen Drehmoments anzugeben, welches dem Nennstrom (203B<sub>h</sub>:01<sub>h</sub>) entspricht. Dazu zählen die Objekte:

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

### HINWEIS

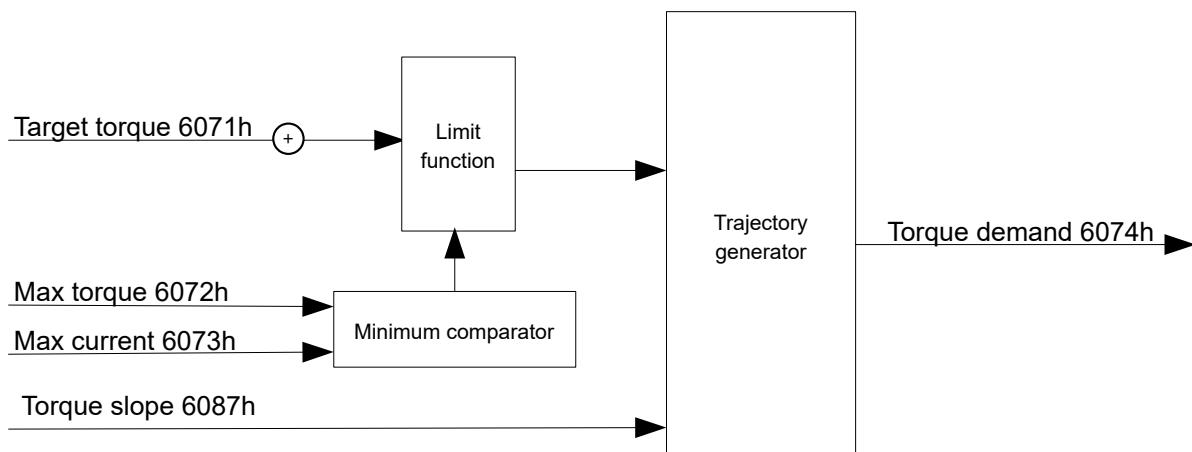


Diese Werte sind nicht limitiert auf 100% des Nennstroms ( $203B_h:01_h$ ). Drehmomentwerte höher als das Nenndrehmoment (generiert von dem Nennstrom) können erreicht werden, wenn die Maximaldauer ( $203B_h:02_h$ ) des maximalen Stroms ( $6073_h$ ) gesetzt wird (siehe I2t Motor-Überlastungsschutz). Alle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

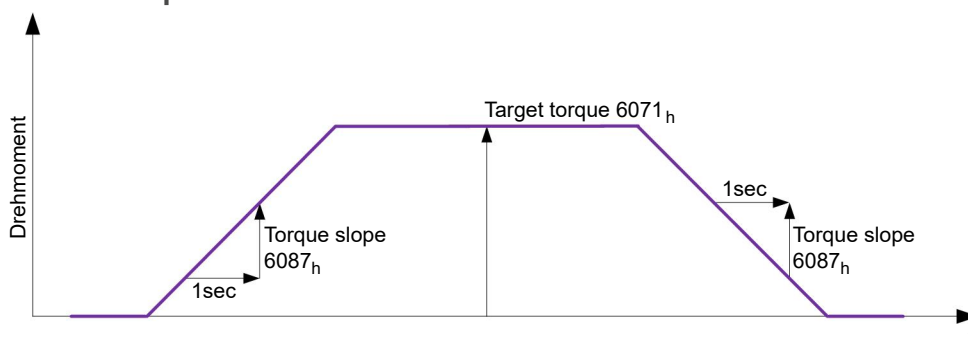
Die folgenden Objekte werden zudem für diesen Operationsmodus benötigt:

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

#### 6.4.5.1 Objekte des Rampengenerators



#### 6.4.5.2 Torque-Verlauf



## 6.5 Homing

### 6.5.1 Übersicht

#### 6.5.1.1 Beschreibung

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

### 6.5.1.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt `6060h` (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.

### 6.5.1.3 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.

### 6.5.1.4 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).

### 6.5.1.5 Objekteinträge

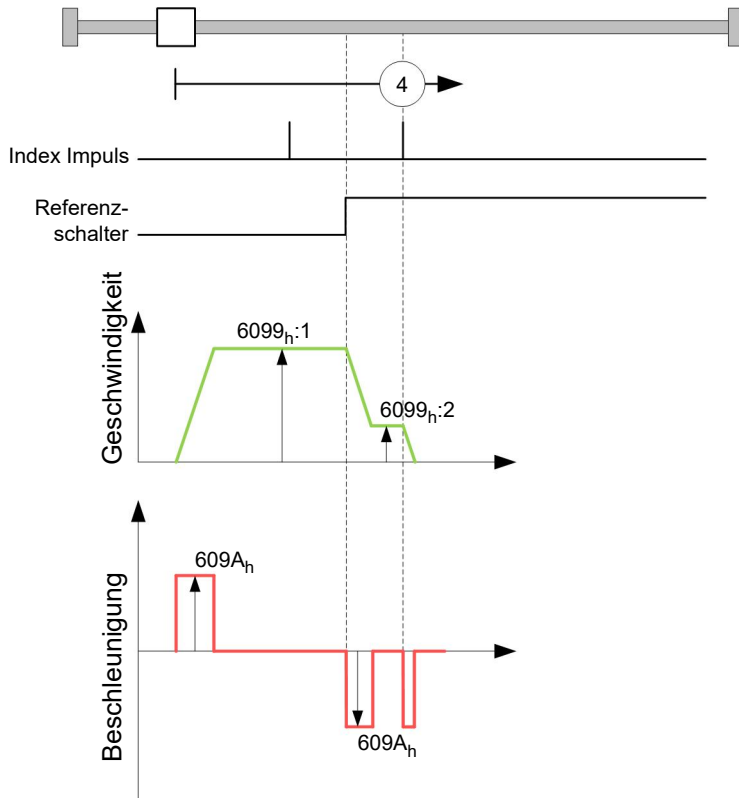
Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

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

## Geschwindigkeiten der Referenzfahrt

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



## 6.5.2 Referenzfahrt-Methode

### 6.5.2.1 Beschreibung

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

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

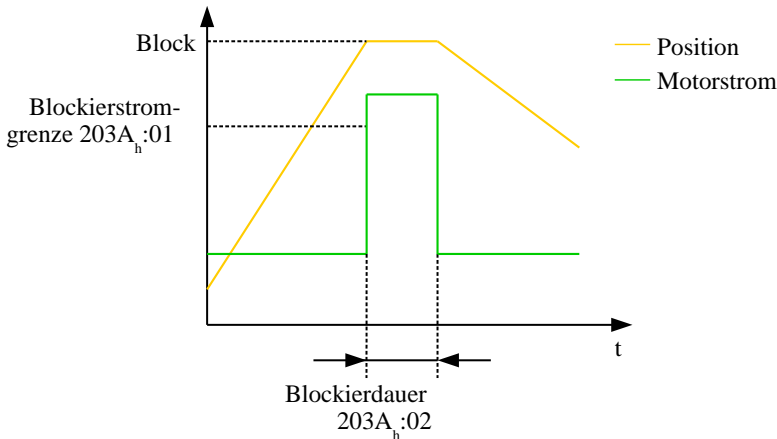
Bei Methoden, die Homing auf Block benutzen, gelten die gleichen Abbildungen wie für die Methoden mit Endschalter. Da sich außer den fehlenden Endschaltern nichts ändert, wurde auf neue Abbildungen verzichtet. Hier gilt für die Abbildungen, dass die Endschalter durch eine mechanische Blockierung ersetzt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

### 6.5.2.2 Homing auf Block

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

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

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



### 6.5.2.3 Methoden-Überblick

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

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

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

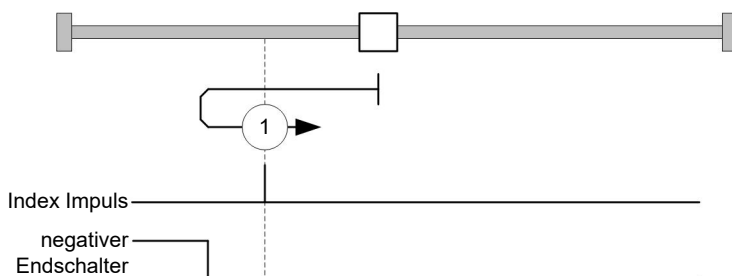
Folgende Methoden können für Homing auf Block benutzt werden:

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

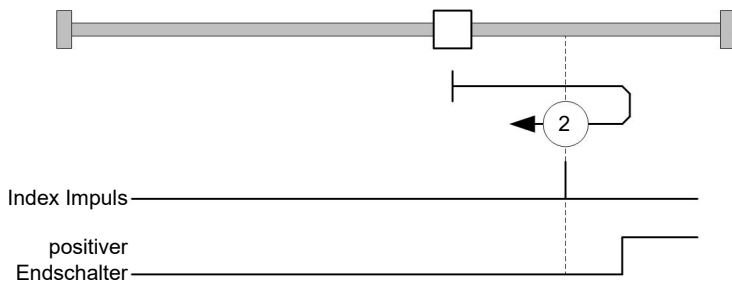
### 6.5.2.4 Methoden 1 und 2

Referenzieren auf Endschalter und Index-Impuls.

Methode 1 referenziert auf negativen Endschalter und Index-Impuls:



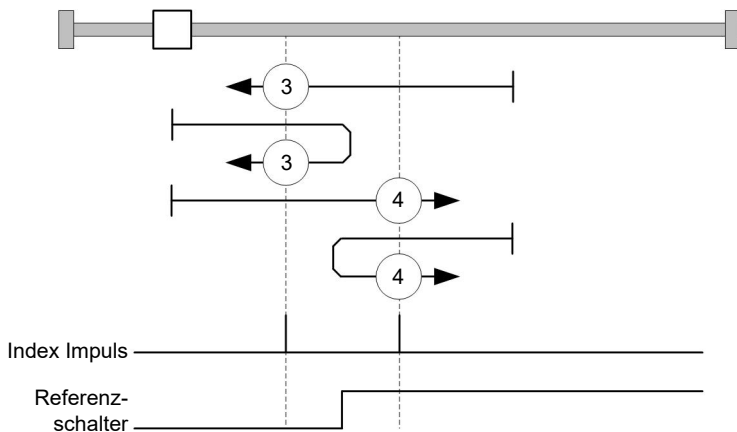
Methode 2 referenziert auf positiven Endschalter und Index-Impuls:



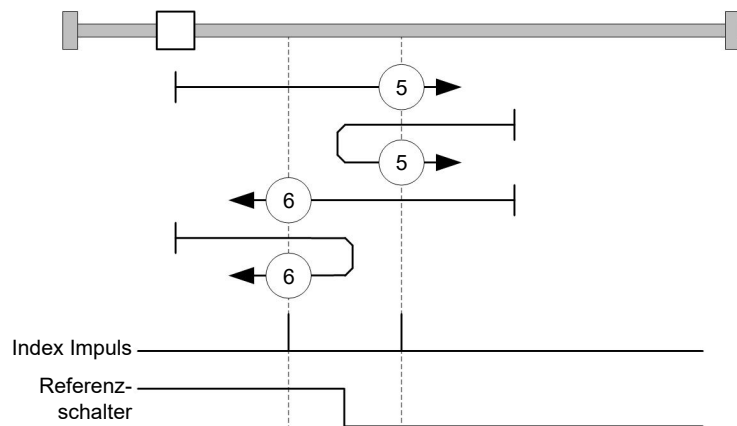
### 6.5.2.5 Methoden 3 bis 6

Referenzieren auf die Schaltflanke des Referenzschalters und Index-Impuls.

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



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



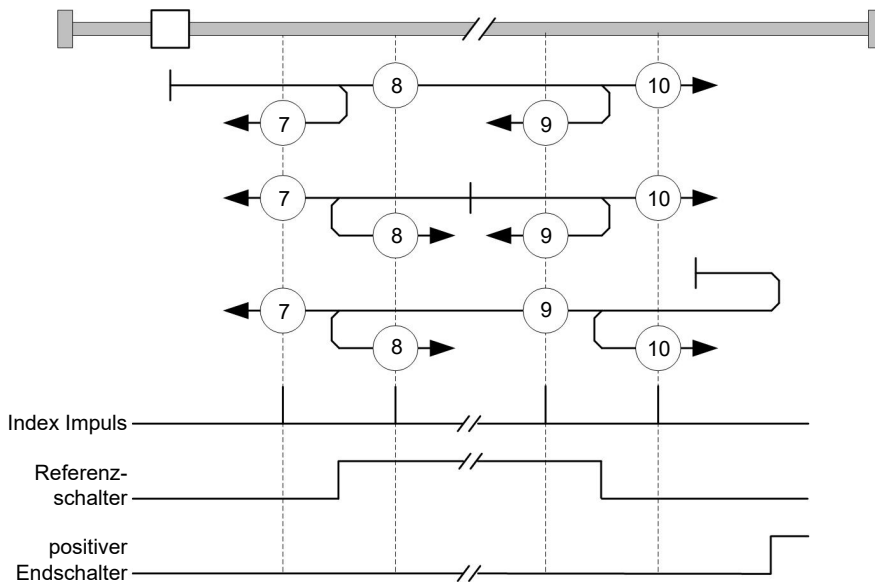
### 6.5.2.6 Methoden 7 bis 14

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

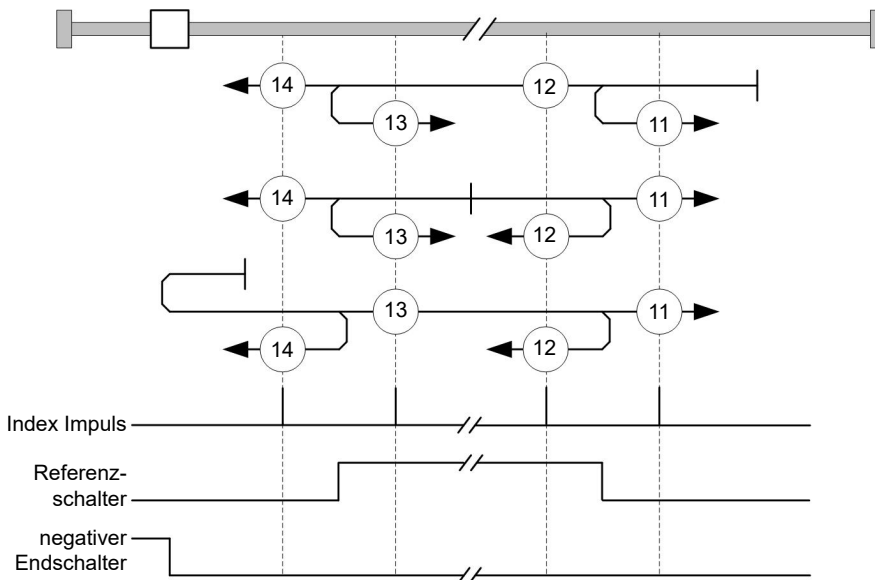
Bei diesen Methoden ist die derzeitige Position relativ zum Referenzschalter unwichtig. Mit der Methode 10 wird beispielsweise immer auf den Index-Impuls rechts neben der rechten Flanke des Referenzschalters referenziert.

Die Methoden 7 bis 10 berücksichtigen den positiven Endschalter:





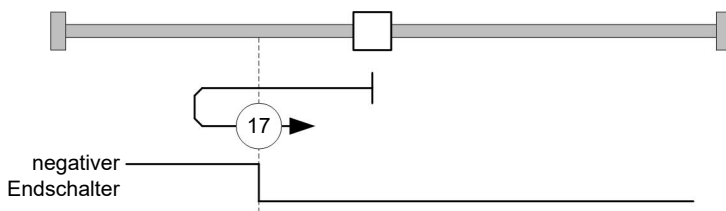
Die Methoden 11 bis 14 berücksichtigen den negativen Endschalter:



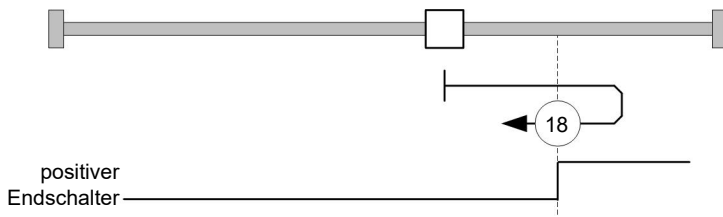
### 6.5.2.7 Methoden 17 und 18

Referenzieren auf den Endschalter ohne den Index-Impuls.

Methode 17 referenziert auf den negativen Endschalter:



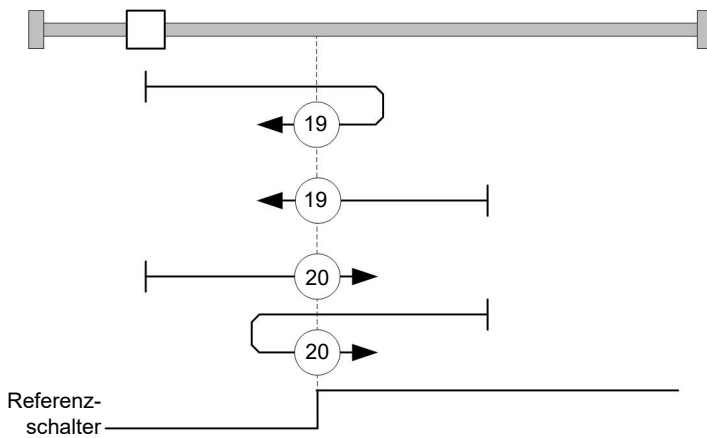
Methode 18 referenziert auf den positiven Endschalter:



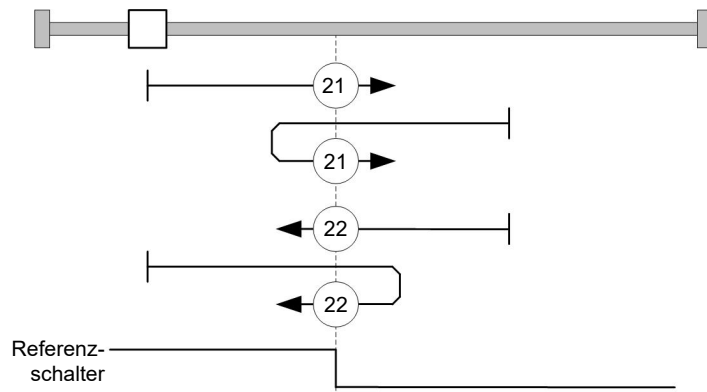
### 6.5.2.8 Methoden 19 bis 22

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

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



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

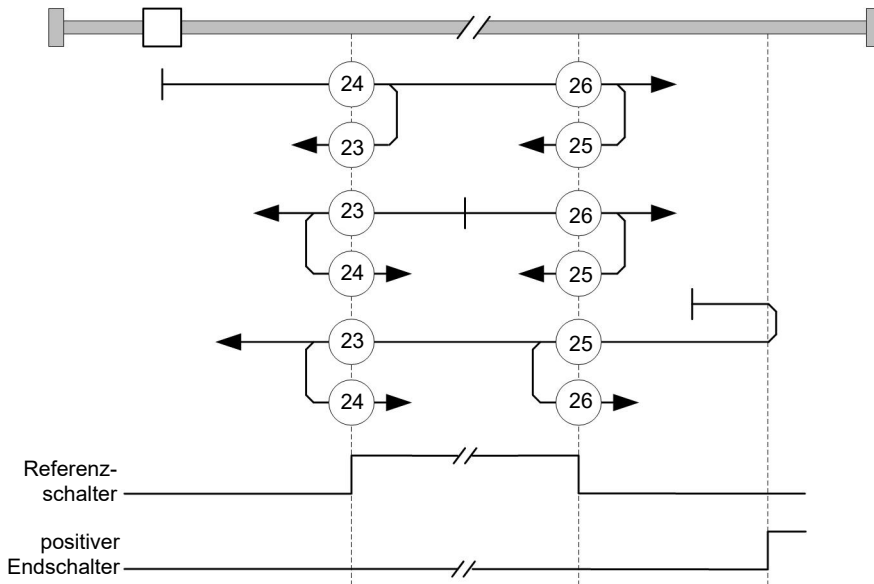


### 6.5.2.9 Methoden 23 bis 30

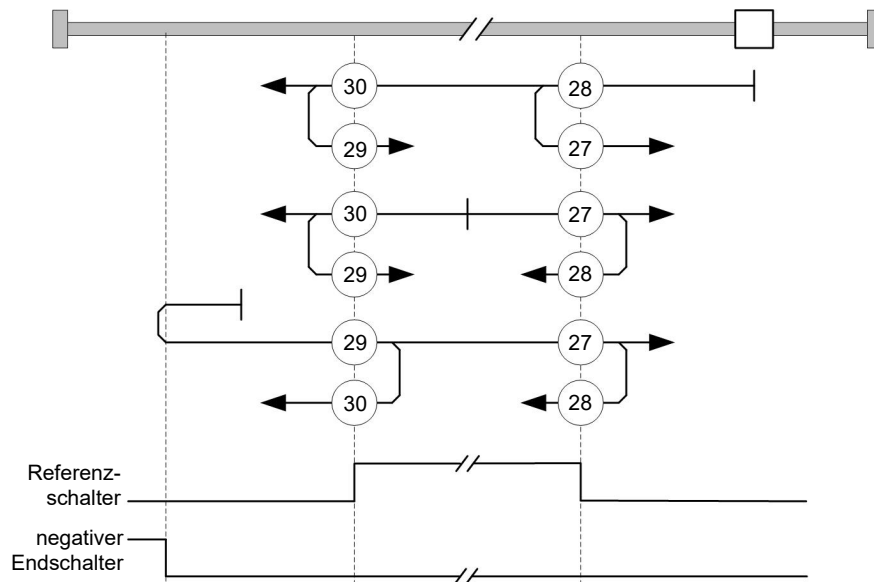
Referenzieren auf Referenzschalter ohne den Index-Impuls (mit Endschaltern).

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

Die Methoden 23 bis 26 berücksichtigen den positiven Referenzschalter:



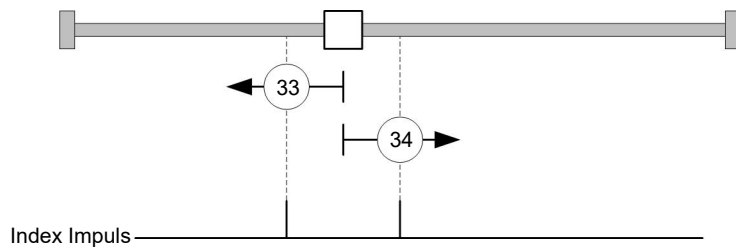
Die Methoden 27 bis 30 berücksichtigen den negativen Referenzschalter:



### 6.5.2.10 Methoden 33 und 34

Referenzieren auf den nächsten Index-Impuls.

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



### 6.5.2.11 Methode 35

Referenziert auf die aktuelle Position.

### HINWEIS



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

## 6.6 Interpolated Position Mode

### 6.6.1 Übersicht

#### 6.6.1.1 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.

#### 6.6.1.2 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.

### 6.6.2 Aktivierung

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

### 6.6.3 Controlword

Folgende Bits im Objekt 6040<sub>h</sub> (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<sub>h</sub>.

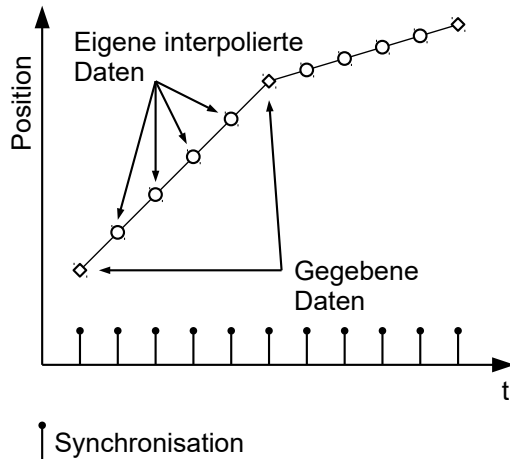
### 6.6.4 Statusword

Folgende Bits im Objekt 6041<sub>h</sub> (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<sub>h</sub> (Following Error Window) und 6066<sub>h</sub> (Following Error Time Out)).

### 6.6.5 Benutzung

Die Steuerung folgt einem linear interpolierten Pfad zwischen der aktuellen und der vorgegebenen Zielposition. Die (nächste) Zielposition muss in das Datensatz 60C1<sub>h</sub>:01<sub>h</sub> geschrieben werden.



In der derzeitigen Implementation wird nur

- lineare Interpolation
- und eine Zielposition

unterstützt.

### 6.6.6 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).

### 6.6.7 Operation

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

## 6.7 Cyclic Synchronous Position

### 6.7.1 Übersicht

#### 6.7.1.1 Beschreibung

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

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



#### HINWEIS

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

### 6.7.1.2 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.

### 6.7.1.3 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*").

### 6.7.1.4 Controlword

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

### 6.7.1.5 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.

## 6.7.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

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

- **605A<sub>h</sub>** (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<sub>h</sub>** (Max Motor Speed): maximale Geschwindigkeit
- **60C2<sub>h</sub>:01<sub>h</sub>** (Interpolation Time Period): Dieses Objekt gibt die Zeit eines *Zyklus* vor, in diesen Zeitabständen muss ein neuer Sollwert in das **607A<sub>h</sub>** geschrieben werden.  
Es gilt dabei: Zykluszeit = Wert des **60C2<sub>h</sub>:01<sub>h</sub>** \* 10<sup>Wert des **60C2:02**</sup> Sekunden.
- **60C2<sub>h</sub>:02<sub>h</sub>** (Interpolation Time Index): Dieses Objekt gibt die Zeitbasis der Zyklen an. Derzeit wird nur der Wert **60C2<sub>h</sub>:02<sub>h</sub>**=-3 unterstützt, das ergibt eine Zeitbasis von 1 Millisekunde.
- **60B0<sub>h</sub>** (Position Offset): Offset für den Positionssollwert in benutzerdefinierten Einheiten
- **60B1<sub>h</sub>** (Velocity Offset): Offset für den Geschwindigkeitssollwert in benutzerdefinierten Einheiten
- **60B2<sub>h</sub>** (Torque Offset): Offset für den Drehmomentsollwert in Promille

Folgende Objekte können in dem Modus ausgelesen werden:

- **6064<sub>h</sub>** (Position Actual Value)
- **606C<sub>h</sub>** (Velocity Actual Value)
- **60F4<sub>h</sub>** (Following Error Actual Value)

## 6.8 Cyclic Synchronous Velocity

### 6.8.1 Übersicht

#### 6.8.1.1 Beschreibung

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

#### 6.8.1.2 Aktivierung

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

#### 6.8.1.3 Controlword

In diesem Modus haben die Bits des Controlword **6040<sub>h</sub>** keine gesonderte Funktion.

#### 6.8.1.4 Statusword

Folgende Bits im Objekt **6041<sub>h</sub>** (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 <b>60FF<sub>h</sub></b> (Target Velocity) wird ignoriert
12	1	Steuerung folgt der Zielvorgabe, das Objekt <b>60FF<sub>h</sub></b> (Target Velocity) wird als Eingabe für die Positionsregelung genutzt.
13	0	Reserviert
13	1	Reserviert

### 6.8.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

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

Folgende Objekte können in dem Modus ausgelesen werden:

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

## 6.9 Cyclic Synchronous Torque

### 6.9.1 Übersicht

#### 6.9.1.1 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.

#### 6.9.1.2 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](#)").

#### 6.9.1.3 Controlword

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

#### 6.9.1.4 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

## 6.9.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **6071<sub>h</sub>** (Target Torque): Dieses Objekt muss zyklisch mit dem Drehmoment-Sollwert beschrieben werden und ist relativ zu **6072<sub>h</sub>** einzustellen.
- **6072<sub>h</sub>** (Max Torque): Beschreibt das maximal zulässige Drehmoment.
- **6073<sub>h</sub>** (Max Current): Maximaler Strom. Das Minimum von **6073<sub>h</sub>** und **6072<sub>h</sub>** wird als Limit für das Drehmoment in **6071<sub>h</sub>** verwendet.
- **6080<sub>h</sub>** (Max Motor Speed): maximale Geschwindigkeit
- **60C2<sub>h</sub>:01<sub>h</sub>** (Interpolation Time Period): Dieses Objekt gibt die Zeit eines *Zyklus* vor, in diesen Zeitabständen muss ein neuer Sollwert in das **6071<sub>h</sub>** geschrieben werden. Es gilt dabei:  $\text{Zykluszeit} = \text{Wert des } 60C2_{h:01_{h}} * 10^{\text{Wert des } 60C2_{h:02_{h}}} \text{ Sekunden}$ .
- **60C2<sub>h</sub>:02<sub>h</sub>** (Interpolation Time Index): Dieses Objekt gibt die Zeitbasis der Zyklen an. Derzeit wird nur der Wert **60C2<sub>h</sub>:02<sub>h</sub>=-3** unterstützt, das ergibt eine Zeitbasis von 1 Millisekunde.
- **60B2<sub>h</sub>** (Torque Offset): Offset für den Drehmomentsollwert in Promille

Folgende Objekte können in dem Modus ausgelesen werden:

- **606C<sub>h</sub>** (Velocity Actual Value)
- **6074<sub>h</sub>** (Torque Demand)

## 6.10 Takt-Richtungs-Modus

### 6.10.1 Beschreibung

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

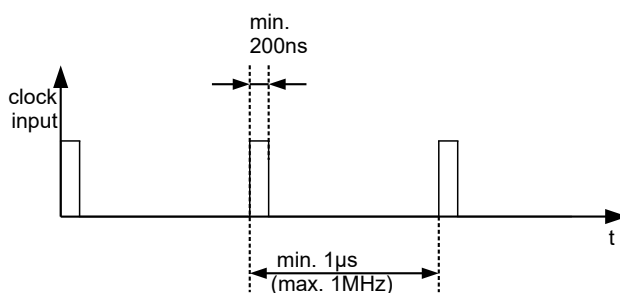
### 6.10.2 Aktivierung

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

### 6.10.3 Generelles

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

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



- Die 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.

### 6.10.4 Statusword

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

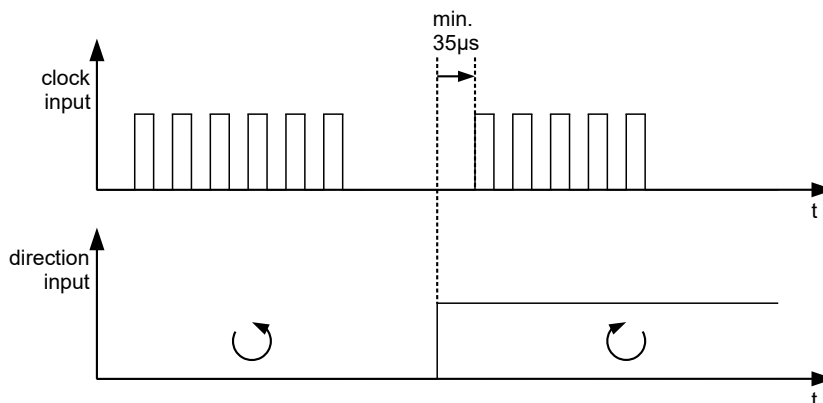
- Bit 13 (Following Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schleppfehler größer als die eingestellten Grenzen ist ( $6065_h$  (Following Error Window) und  $6066_h$  (Following Error Time Out)).

### 6.10.5 Unterarten des Takt-Richtungs-Modus

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

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

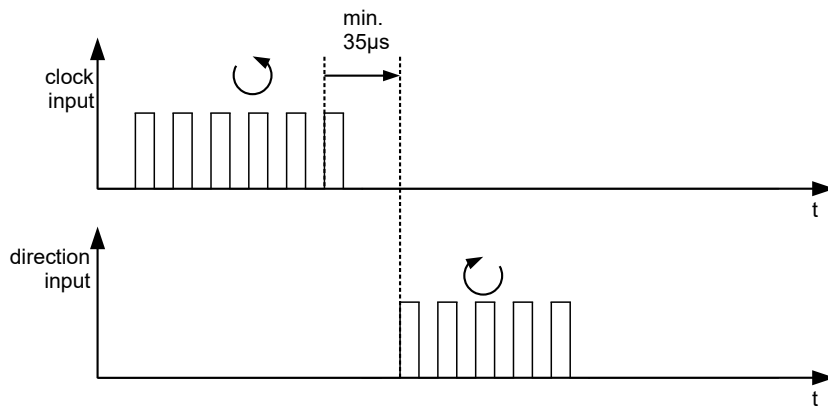
In diesem Modus müssen über den Takteingang die Pulse vorgegeben werden, das Signal des Richtungseingangs gibt dabei die Drehrichtung vor (siehe nachfolgende Grafik).



### 6.10.5.2 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).



## 6.11 Auto-Setup

### 6.11.1 Beschreibung

Um einige Parameter im Bezug zum Motor und den angeschlossenen Sensoren (Encoder/Hallsensoren) zu ermitteln, wird ein *Auto-Setup* durchgeführt. Der Closed Loop Betrieb setzt ein erfolgreich abgeschlossenes *Auto-Setup* voraus. Bei den Plug & Drive Motoren ist es nicht notwendig ein *Auto-Setup* auszuführen, da dieses bereits werksseitig durchgeführt wurde. Für Details siehe entsprechenden Abschnitt im Kapitel Inbetriebnahme.

### 6.11.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt `6060h` (Modes Of Operation) der Wert "-2" ("FE<sub>n</sub>") gesetzt werden (siehe CiA 402 Power State Machine).

### 6.11.3 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.

### 6.11.4 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

## 7 Spezielle Funktionen

### 7.1 Digitale Ein- und Ausgänge

Diese Steuerung verfügt über digitale Ein- und Ausgänge.

#### 7.1.1 Bitzuordnung

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

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

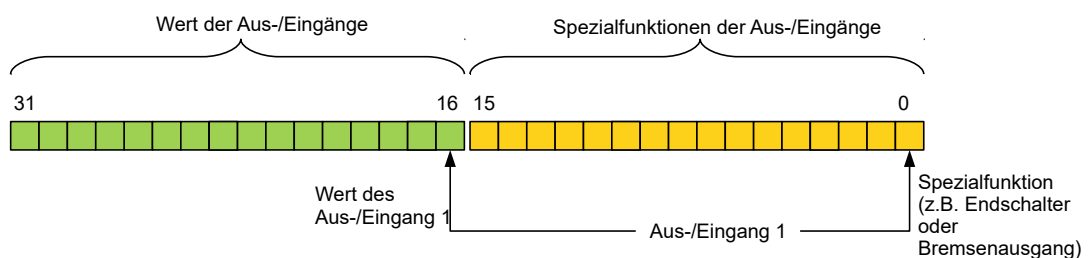
#### Beispiel

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

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

In der nachfolgenden Zeichnung ist diese Zuordnung graphisch dargestellt.

Bits eines beliebigen Objektes zur Steuerung eines Aus-/Eingangs



### 7.1.2 Digitale Eingänge

#### 7.1.2.1 Übersicht



#### HINWEIS

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



#### HINWEIS

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

Folgende Eingänge stehen zur Verfügung:

Eingang	Sonderfunktion	Schaltswelle umschaltbar	Differenziell / single-ended
1	Negativer Endschalter	ja, 5 V oder 24 V (siehe <a href="#">3240<sub>h</sub>:06<sub>h</sub></a> )	Die Eingänge sind nur gemeinsam umschaltbar. (siehe <a href="#">3240<sub>h</sub>:07<sub>h</sub></a> )
2	Positiver Endschalter / Richtungseingang im Takt-Richtungs-Modus	ja, 5 V oder 24 V (siehe <a href="#">3240<sub>h</sub>:06<sub>h</sub></a> )	
3	Referenzschalter / Takteingang im Takt-Richtungs-Modus	ja, 5 V oder 24 V (siehe <a href="#">3240<sub>h</sub>:06<sub>h</sub></a> )	
4	keine	ja, 5 V oder 24 V (siehe <a href="#">3240<sub>h</sub>:06<sub>h</sub></a> )	
5	keine	ja, 5 V oder 24 V (siehe <a href="#">3240<sub>h</sub>:06<sub>h</sub></a> )	
6	keine	ja, 5 V oder 24 V (siehe <a href="#">3240<sub>h</sub>:06<sub>h</sub></a> )	

Wenn Sie das [3240<sub>h</sub>:07<sub>h</sub>](#) auf den Wert "1" setzen, stehen Ihnen, anstatt sechs single-ended, drei differenzielle Eingänge zur Verfügung:

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

### 7.1.2.2 Objekteinträge

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

- [3240<sub>h</sub>:01<sub>h</sub>](#) (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<sub>h</sub>:01<sub>h</sub>](#) auf "1" gesetzt werden.

- [3240<sub>h</sub>:02<sub>h</sub>](#) (Function Inverted): Dieser Subindex wechselt von Schließer-Logik (ein logischer High-Pegel am Eingang ergibt den Wert "1" im Objekt [60FD<sub>h</sub>](#)) 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<sub>h</sub>:03<sub>h</sub>](#) (Force Enable): Dieser Subindex schaltet die Softwaresimulation von Eingangswerten ein, wenn das entsprechende Bit auf "1" gesetzt ist.

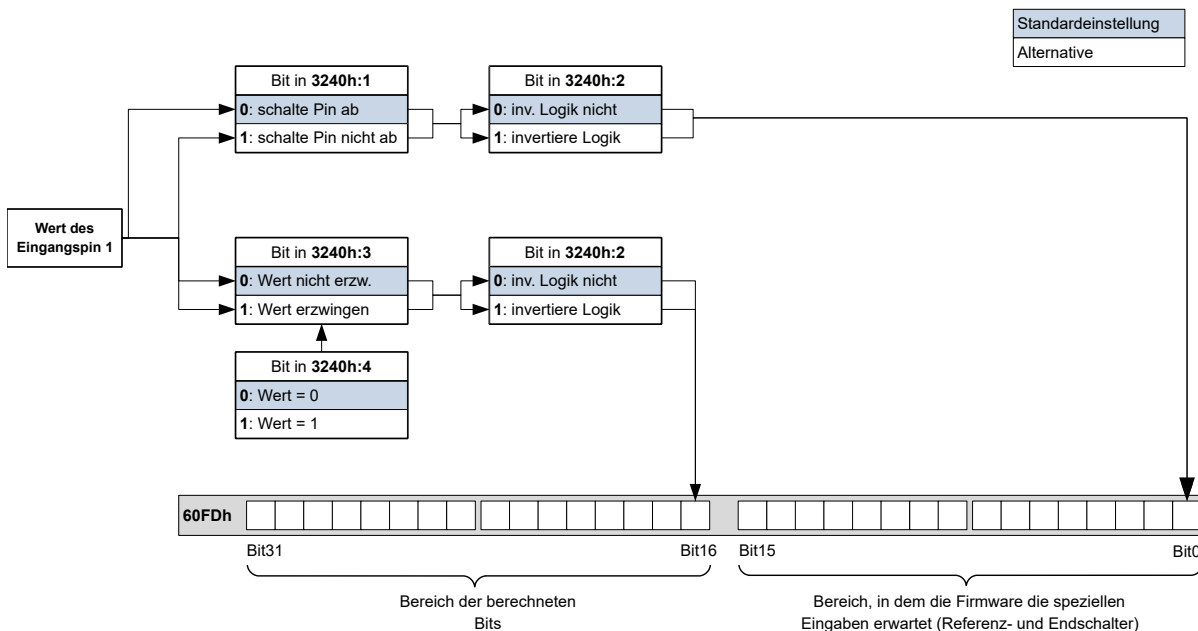
Dann werden nicht mehr die tatsächlichen, sondern die in Objekt  $3240_h:04_h$  eingestellten Werte für den jeweiligen Eingang verwendet. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw.

- $3240_h:04_h$  (Force Value): Dieses Bit gibt den Wert vor, der als Eingangswert eingelesen werden soll, wenn das gleiche Bit im Objekt  $3240_h:03_h$  gesetzt wurde.
- $3240_h:05_h$  (Raw Value): Dieses Objekt beinhaltet den unmodifizierten Eingabewert.
- $3240_h:06_h$  (Input Range Select): Damit können Eingänge - welche über diese Funktion verfügen - von der Schaltschwelle von 5 V (Bit auf "0") auf die Schaltschwelle 24 V (Bit auf "1") umgeschaltet werden. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw.
- $3240_h:07_h$  (Differential Select): Dieser Subindex schaltet bei den Eingängen zwischen "single-ended Eingang" (Wert "0" in dem Subindex) zu "Differentieller Eingang" (Wert "1" in dem Subindex) für alle Eingänge auf einmal um.
- $60FD_h$  (Digital Inputs): Dieses Objekt enthält eine Zusammenfassung der Eingänge und der Spezialfunktionen.

### 7.1.2.3 Verrechnung der Eingänge

Verrechnung des Eingangssignals am Beispiel von Eingang 1:

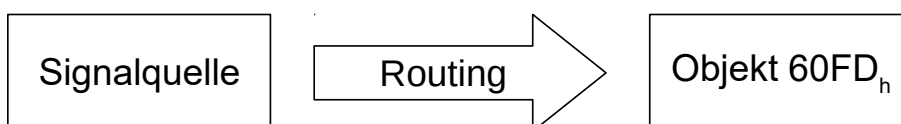
Der Wert an Bit 0 des Objekts  $60FD_h$  wird von der Firmware als negativer Endschalter interpretiert, das Ergebnis der vollständigen Verrechnung wird in Bit 16 abgelegt.



### 7.1.2.4 Input Routing

#### Prinzip

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



#### Aktivierung

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

**HINWEIS**


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

**HINWEIS**


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

**Routing**

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

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

Die nachfolgende Tabelle beschreibt die invertierten Signale der vorherigen Tabelle.

Nummer		Signalquelle
dec	hex	
128	80	Signal ist immer 1

Nummer		Signalquelle
dec	hex	
129	81	Invertierter Physikalischer Eingang 1
130	82	Invertierter Physikalischer Eingang 2
131	83	Invertierter Physikalischer Eingang 3
132	84	Invertierter Physikalischer Eingang 4
133	85	Invertierter Physikalischer Eingang 5
134	86	Invertierter Physikalischer Eingang 6
135	87	Invertierter Physikalischer Eingang 7
136	88	Invertierter Physikalischer Eingang 8
137	89	Invertierter Physikalischer Eingang 9
138	8A	Invertierter Physikalischer Eingang 10
139	8B	Invertierter Physikalischer Eingang 11
140	8C	Invertierter Physikalischer Eingang 12
141	8D	Invertierter Physikalischer Eingang 13
142	8E	Invertierter Physikalischer Eingang 14
143	8F	Invertierter Physikalischer Eingang 15
144	90	Invertierter Physikalischer Eingang 16
196	C4	Invertierter Encoder Eingang "A"
197	C5	Invertierter Encoder Eingang "B"
198	C6	Invertierter Encoder Eingang "Index"
199	C7	Invertiertes USB Power Signal

### Beispiel

Es soll der Eingang 1 auf Bit 16 des Objekts `60FDh` geroutet werden:

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

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

### 7.1.2.5 Interlock-Funktion

Bei der Interlock-Funktion handelt es sich um eine Freigabe, die Sie über das Bit 3 in `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.

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

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".
2. Um den Eingang 4 auf Bit 3 zu routen, setzen Sie das `3242h:04h` auf "4".



## 7.1.3 Digitale Ausgänge

### 7.1.3.1 Ausgänge

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

### 7.1.3.2 Beschaltung



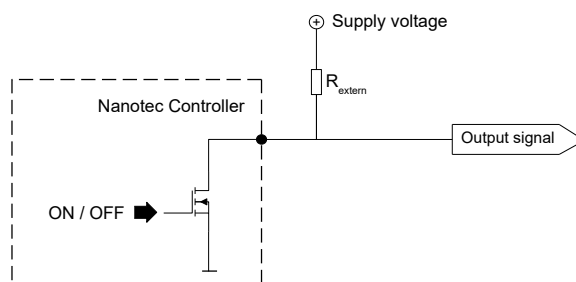
#### HINWEIS

Beachten Sie immer die maximale Belastbarkeit des Ausganges (siehe [Anschlussbelegung](#)).

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

#### Beispiel

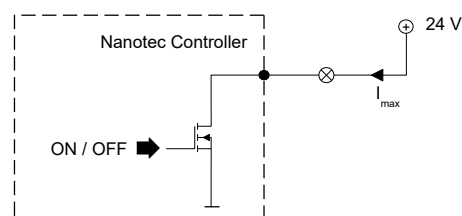
Es soll das digitale Ausgangssignal weiter verwendet werden. Dazu ist eine Beschaltung wie im nachfolgenden Bild zu realisieren.



Bei einer Versorgungsspannung von +24 V wird ein Widerstandswert  $R_{\text{extern}}$  von 10 k $\Omega$  empfohlen.

#### Beispiel

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



### 7.1.3.3 Objekteinträge

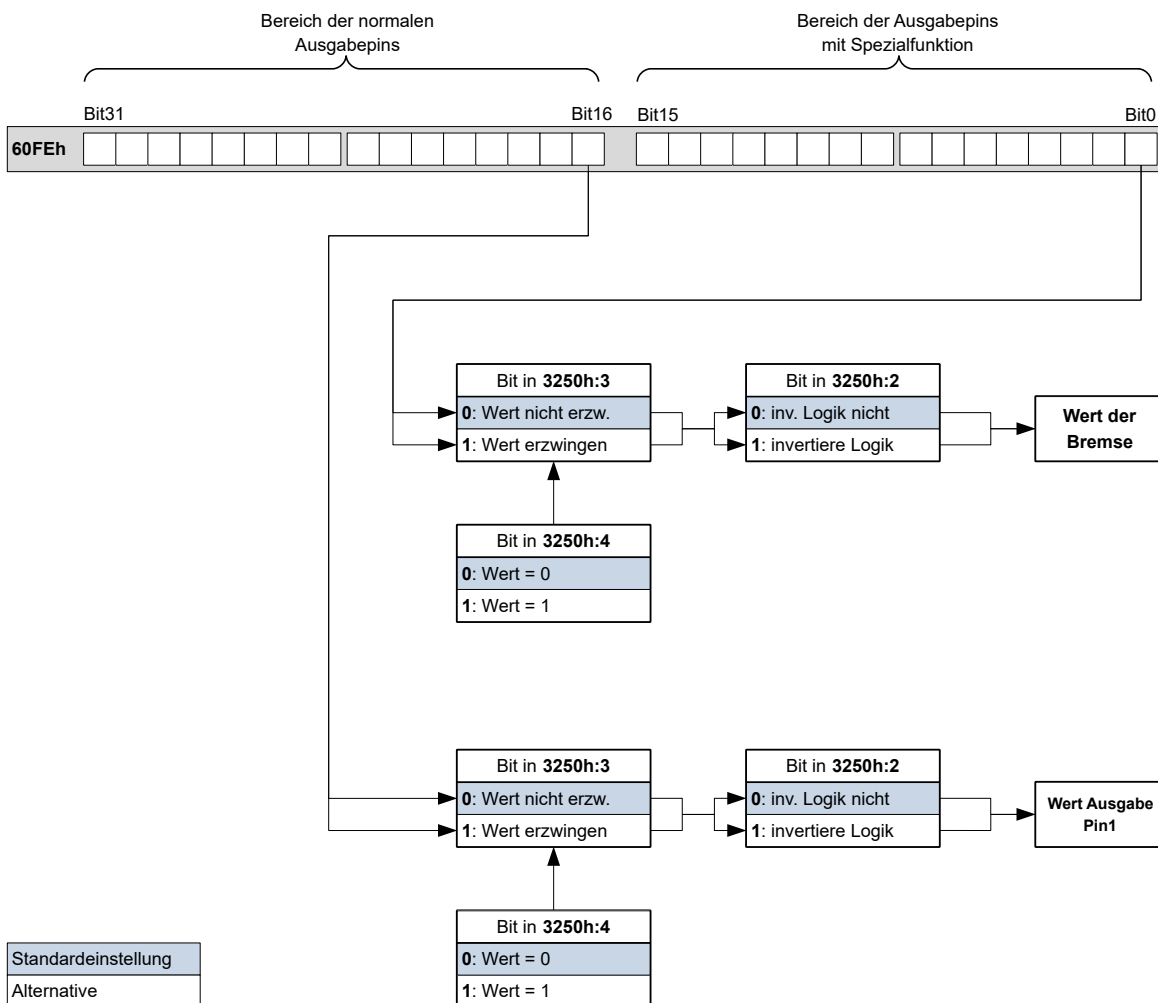
Es existieren zusätzliche OD-Einträge, um den Wert der Ausgänge zu manipulieren (siehe dazu das nachfolgende Beispiel). Ähnlich wie bei den Eingängen wirkt immer nur das Bit an der entsprechenden Stelle auf den jeweiligen Ausgang:

- $3250_h;01_h$ : Keine Funktion.

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

### 7.1.3.4 Verrechnung der Ausgänge

Beispiel für die Verrechnung der Bits für die Ausgänge:

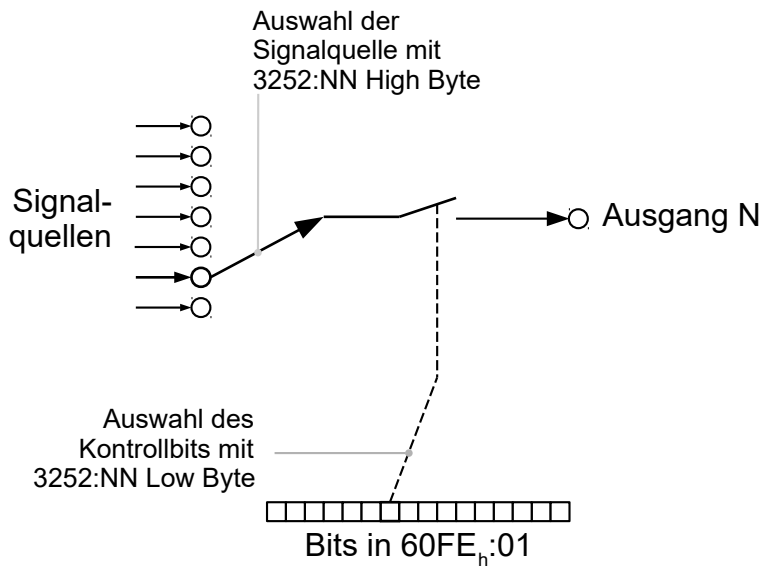


### 7.1.3.5 Output Routing

#### Prinzip

Der "Output Routing Mode" weist einem Ausgang eine Signalquelle zu, ein Kontrollbit im Objekt **60FE<sub>h</sub>:01<sub>h</sub>** schaltet das Signal ein oder aus.

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



### Aktivierung

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

**HINWEIS**



Die Einträge  $3250_h:01_h$  bis  $3250_h:04_h$  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 PWM-Ausgangs (Software-PWM)
$02_h$	Konfiguration des Ausgangs 1
$03_h$	Konfiguration des Ausgangs 2 (falls verfügbar)
...	...
$0n_h$	Konfiguration des Ausgangs n (falls verfügbar)

**HINWEIS**



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

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

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



#### TIPP

Um das Routing zu deaktivieren, tragen Sie den Wert  $FFFF_h$  ein.

#### Nummer in $3252:01$ bis $0n$

$00XX_h$	Ausgang ist immer "1"
$01XX_h$	Ausgang ist immer "0"
$02XX_h$	Encodersignal ( $6063_h$ ) mit Frequenzteiler 1
$03XX_h$	Encodersignal ( $6063_h$ ) mit Frequenzteiler 2
$04XX_h$	Encodersignal ( $6063_h$ ) mit Frequenzteiler 4
$05XX_h$	Encodersignal ( $6063_h$ ) mit Frequenzteiler 8
$06XX_h$	Encodersignal ( $6063_h$ ) mit Frequenzteiler 16
$07XX_h$	Encodersignal ( $6063_h$ ) mit Frequenzteiler 32
$08XX_h$	Encodersignal ( $6063_h$ ) mit Frequenzteiler 64
$09XX_h$	Position Actual Value ( $6064_h$ ) mit Frequenzteiler 1
$0AXX_h$	Position Actual Value ( $6064_h$ ) mit Frequenzteiler 2
$0BXX_h$	Position Actual Value ( $6064_h$ ) mit Frequenzteiler 4
$0CXX_h$	Position Actual Value ( $6064_h$ ) mit Frequenzteiler 8
$0DXX_h$	Position Actual Value ( $6064_h$ ) mit Frequenzteiler 16
$0EXX_h$	Position Actual Value ( $6064_h$ ) mit Frequenzteiler 32
$0FXX_h$	Position Actual Value ( $6064_h$ ) mit Frequenzteiler 64
$10XX_h$	PWM-Signal, das mit Objekt $2038_h:05_h$ und $06_h$ konfiguriert wird
$11XX_h$	Invertiertes PWM-Signal, das mit Objekt $2038_h:05_h$ und $06_h$ konfiguriert wird

#### HINWEIS



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

#### Beispiel

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

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

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

### Beispiel

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

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

## 7.2 Analoge Eingänge

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

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

### 7.2.1 Objekteinträge

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

- `3220h` (Analog Inputs):  
Dieses Objekt zeigt die Momentanwerte der Analogeingänge in *ADC Digits* an.
- `3221h` (Analogue Inputs Control):  
Mit diesem Objekt schalten Sie einen Analogeingang von Spannungs- auf Strommessung um.
- `3320h` (Read Analogue Input):  
Dieses Objekt zeigt die Momentanwerte der Analogeingänge in benutzerdefinierten Einheiten an.
- `3321h` (Analogue Input Offset):  
Dies ist der Offset, der zum eingelesenen Analogwert (`3220h`) addiert wird, bevor die Skalierung (Multiplikator aus dem Objekt `3322h` und Teiler aus dem Objekt `3323h`) vorgenommen wird.
- `3322h` (Analogue Input Factor Numerator):  
Dies ist der Wert, mit dem der eingelesene Analogwert (`3220h + 3321`) multipliziert wird, bevor er in das Objekt `3320h` geschrieben wird.
- `3323h` (Analogue Input Factor Denominator):  
Dies ist der Wert, mit dem der eingelesene Analogwert (`3220h + 3321h`) dividiert wird, bevor er in das Objekt `3320h` geschrieben wird.

### 7.2.2 Analogwert skalieren

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

Die benutzerdefinierten Einheiten setzen sich aus Offset (`3321h`) und Skalierungswert (`3322h / 3323h`) zusammen. Sind beide noch mit Default-Werten beschrieben, wird der Wert in `3320h` in der Einheit *ADC Digits* angegeben.

### Beispiel

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

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

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

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

$$1023 \text{ Digits} * 10000 \text{ mV} / 1023 = 10000 \text{ mV}$$

## 7.3 Automatische Bremsensteuerung

### 7.3.1 Beschreibung

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

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

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

### 7.3.2 Aktivierung und Anschluss

Die Bremse kann entweder automatisch oder manuell gesteuert werden:

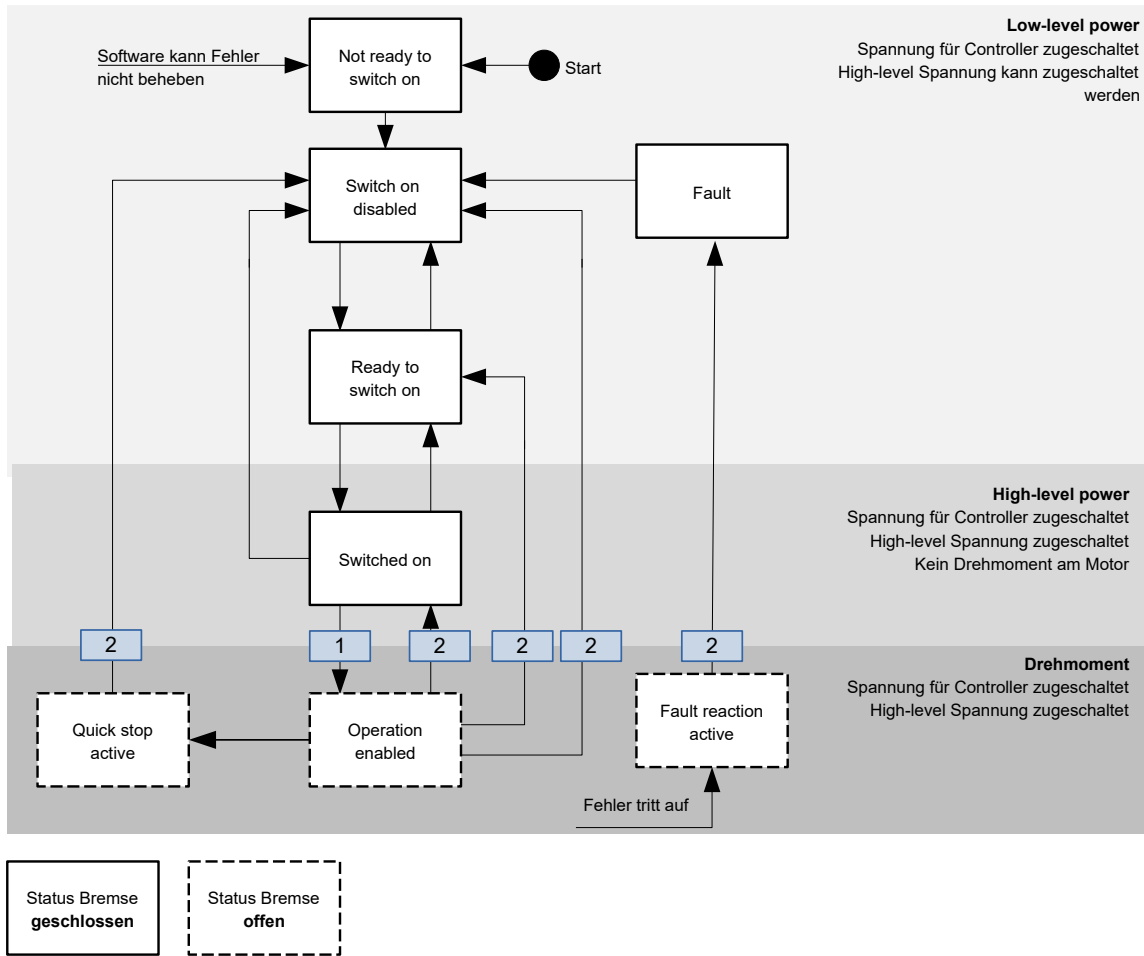
- Automatisch: Bit 2 des Objekts 3202<sub>h</sub> auf "1" setzen aktiviert die Bremsensteuerung.
- Manuell: Bit 2 des Objekts 3202<sub>h</sub> auf "0" setzen deaktiviert die Bremsensteuerung, die Bremse lässt sich jetzt mit dem Bit 0 im Objekt 60FE<sub>h</sub>:01<sub>h</sub> kontrollieren.

#### 7.3.2.1 Anschluss

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

### 7.3.3 Steuerung der Bremse

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



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

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

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

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

### 7.3.4 Bremsen-PWM

Die eingeschaltete Bremse erzeugt am Ausgang der Steuerung ein PWM-Signal, welches im Tastgrad und der Frequenz eingestellt werden kann. Sollte ein Ausgangspin ohne PWM benötigt werden, lässt sich ein Tastgrad von 100 Prozent einstellen.

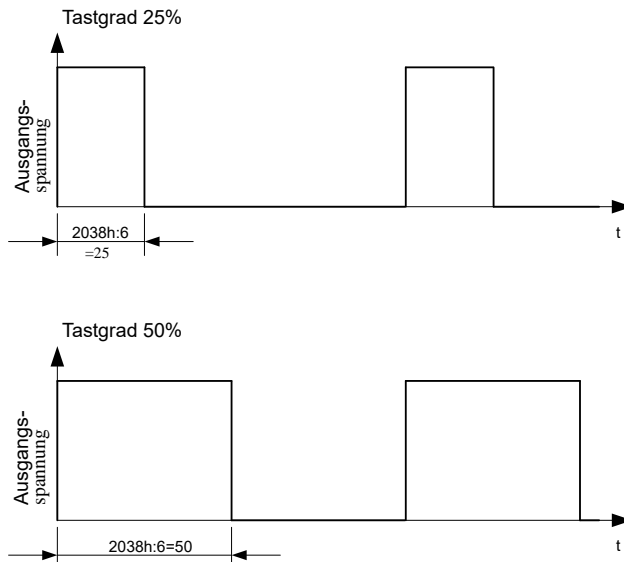
#### 7.3.4.1 Frequenz

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

### 7.3.4.2 Tastgrad

Der Tastgrad – das Verhältnis Impuls- zu Periodendauer – wird im  $2038h:6$  eingestellt. Der Wert wird als Prozentzahl angesehen und kann zwischen 2 und 100 gewählt werden. Bei einem Wert von 100 ist der Ausgangspin dauerhaft eingeschaltet.

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



## 7.4 I<sup>2</sup>t Motor-Überlastungsschutz

### 7.4.1 Beschreibung

#### HINWEIS



Für Schrittmotoren wird nur der Nennstrom und kein Maximalstrom angegeben. Daher erfolgt die Nutzung von I<sup>2</sup>t mit Schrittmotoren ohne Gewähr.

Das Ziel des I<sup>2</sup>t 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  $3202h$  muss auf "1" gesetzt sein).

### 7.4.2 Objekteinträge

Folgende Objekte haben Einfluss auf den I<sup>2</sup>t Motor-Überlastungsschutz:

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

Folgende Objekte zeigen den gegenwärtigen Zustand von I<sup>2</sup>t an:

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



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

### 7.4.3 Aktivierung

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

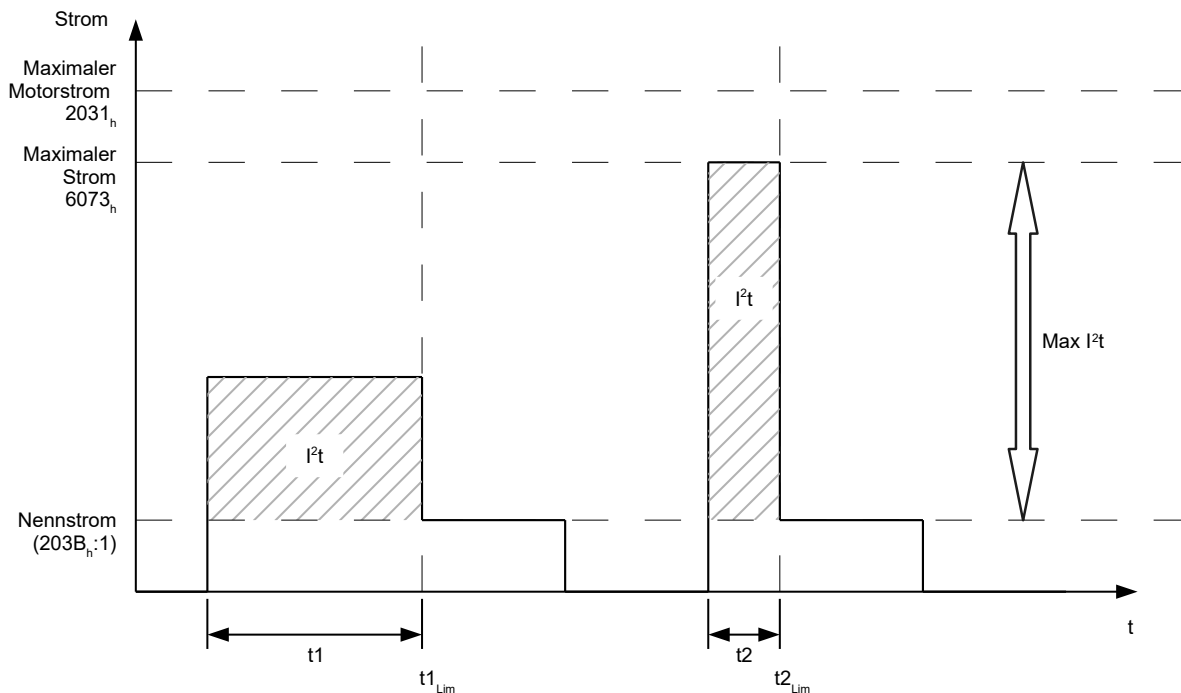
Zum Aktivieren des Modus müssen Sie die vier oben genannten Objekteinträge (`2031h`, `6073h`, `203Bh:1h`, `203Bh:2h`) sinnvoll beschreiben. Das bedeutet, dass der Maximalstrom größer als der Nennstrom sein muss und ein Zeitwert für die maximale Dauer des Maximalstroms eingetragen sein muss. Wenn diese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

### 7.4.4 Funktion von $I^2t$

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

Der Motor kann solange mit Maximalstrom laufen, bis das berechnete  $I^2t_{Lim}$  erreicht wird. Daraufgehend wird der Strom sofort auf Nennstrom gesenkt. Der Maximalstrom wird durch den maximalen Motorstrom (`2031h`) 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  $I^2t_{Lim}$  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  $I^2t_{Lim}$  schneller erreicht, als im Zeitraum  $t_1$ .

## 7.5 Objekte speichern

### HINWEIS



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

### HINWEIS



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

### HINWEIS



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

## 7.5.1 Allgemeines

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

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

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

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

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

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

## 7.5.2 Kategorie: Kommunikation

- 2102<sub>h</sub>: Fieldbus Module Control
- 3502<sub>h</sub>: MODBUS Rx PDO Mapping
- 3602<sub>h</sub>: MODBUS Tx PDO Mapping

## 7.5.3 Kategorie: Applikation

- 2034<sub>h</sub>: Upper Voltage Warning Level
- 2035<sub>h</sub>: Lower Voltage Warning Level
- 2036<sub>h</sub>: Open Loop Current Reduction Idle Time
- 2037<sub>h</sub>: Open Loop Current Reduction Value/factor
- 2038<sub>h</sub>: Brake Controller Timing
- 203A<sub>h</sub>: Homing On Block Configuration
- 203D<sub>h</sub>: Torque Window
- 203E<sub>h</sub>: Torque Window Time Out
- 203F<sub>h</sub>: Max Slippage Time Out
- 2057<sub>h</sub>: Clock Direction Multiplier

- 2058<sub>h</sub>: Clock Direction Divider
- 205B<sub>h</sub>: Clock Direction Or Clockwise/Counter Clockwise Mode
- 2084<sub>h</sub>: Bootup Delay
- 2290<sub>h</sub>: PDI Control
- 2300<sub>h</sub>: NanoJ Control
- 2410<sub>h</sub>: NanoJ Init Parameters
- 2800<sub>h</sub>: Bootloader And Reboot Settings
- 3210<sub>h</sub>: Motor Drive Parameter Set
- 3212<sub>h</sub>: Motor Drive Flags
- 321A<sub>h</sub>: Current Controller Parameters
- 321B<sub>h</sub>: Velocity Controller Parameters
- 321C<sub>h</sub>: Position Controller Parameters
- 321D<sub>h</sub>: Pre-control
- 321E<sub>h</sub>: Voltage Limit
- 3221<sub>h</sub>: Analog Inputs Control
- 3240<sub>h</sub>: Digital Inputs Control
- 3242<sub>h</sub>: Digital Input Routing
- 3243<sub>h</sub>: Digital Input Homing Capture
- 3250<sub>h</sub>: Digital Outputs Control
- 3252<sub>h</sub>: Digital Output Routing
- 3321<sub>h</sub>: Analog Input Offsets
- 3322<sub>h</sub>: Analog Input Numerators
- 3323<sub>h</sub>: Analog Input Denominators
- 3700<sub>h</sub>: Deviation Error Option Code
- 3701<sub>h</sub>: Limit Switch Error Option Code
- 4013<sub>h</sub>: HW Configuration
- 4015<sub>h</sub>: Special Drive Modes
- 6040<sub>h</sub>: Controlword
- 6042<sub>h</sub>: VI Target Velocity
- 6046<sub>h</sub>: VI Velocity Min Max Amount
- 6048<sub>h</sub>: VI Velocity Acceleration
- 6049<sub>h</sub>: VI Velocity Deceleration
- 604A<sub>h</sub>: VI Velocity Quick Stop
- 604C<sub>h</sub>: VI Dimension Factor
- 605A<sub>h</sub>: Quick Stop Option Code
- 605B<sub>h</sub>: Shutdown Option Code
- 605C<sub>h</sub>: Disable Option Code
- 605D<sub>h</sub>: Halt Option Code
- 605E<sub>h</sub>: Fault Option Code
- 6060<sub>h</sub>: Modes Of Operation
- 6065<sub>h</sub>: Following Error Window
- 6066<sub>h</sub>: Following Error Time Out
- 6067<sub>h</sub>: Position Window
- 6068<sub>h</sub>: Position Window Time
- 606D<sub>h</sub>: Velocity Window
- 606E<sub>h</sub>: Velocity Window Time
- 606F<sub>h</sub>: Velocity Threshold
- 6070<sub>h</sub>: Velocity Threshold Time
- 6071<sub>h</sub>: Target Torque
- 6072<sub>h</sub>: Max Torque
- 607A<sub>h</sub>: Target Position
- 607B<sub>h</sub>: Position Range Limit
- 607C<sub>h</sub>: Home Offset

- 607D<sub>h</sub>: Software Position Limit
- 607E<sub>h</sub>: Polarity
- 607F<sub>h</sub>: Max Profile Velocity
- 6081<sub>h</sub>: Profile Velocity
- 6082<sub>h</sub>: End Velocity
- 6083<sub>h</sub>: Profile Acceleration
- 6084<sub>h</sub>: Profile Deceleration
- 6085<sub>h</sub>: Quick Stop Deceleration
- 6086<sub>h</sub>: Motion Profile Type
- 6087<sub>h</sub>: Torque Slope
- 6091<sub>h</sub>: Gear Ratio
- 6092<sub>h</sub>: Feed Constant
- 6096<sub>h</sub>: Velocity Factor
- 6097<sub>h</sub>: Acceleration Factor
- 6098<sub>h</sub>: Homing Method
- 6099<sub>h</sub>: Homing Speed
- 609A<sub>h</sub>: Homing Acceleration
- 60A2<sub>h</sub>: Jerk Factor
- 60A4<sub>h</sub>: Profile Jerk
- 60A8<sub>h</sub>: SI Unit Position
- 60A9<sub>h</sub>: SI Unit Velocity
- 60B0<sub>h</sub>: Position Offset
- 60B1<sub>h</sub>: Velocity Offset
- 60B2<sub>h</sub>: Torque Offset
- 60C1<sub>h</sub>: Interpolation Data Record
- 60C2<sub>h</sub>: Interpolation Time Period
- 60C4<sub>h</sub>: Interpolation Data Configuration
- 60C5<sub>h</sub>: Max Acceleration
- 60C6<sub>h</sub>: Max Deceleration
- 60E8<sub>h</sub>: Additional Gear Ratio - Motor Shaft Revolutions
- 60E9<sub>h</sub>: Additional Feed Constant - Feed
- 60ED<sub>h</sub>: Additional Gear Ratio - Driving Shaft Revolutions
- 60EE<sub>h</sub>: Additional Feed Constant - Driving Shaft Revolutions
- 60F2<sub>h</sub>: Positioning Option Code
- 60F8<sub>h</sub>: Max Slippage
- 60FE<sub>h</sub>: Digital Outputs
- 60FF<sub>h</sub>: Target Velocity

### 7.5.4 Kategorie: Benutzer

- 2701<sub>h</sub>: Customer Storage Area

### 7.5.5 Kategorie: Bewegung

- 3202<sub>h</sub>: Motor Drive Submode Select
- 320D<sub>h</sub>: Torque Of Inertia Factor
- 6073<sub>h</sub>: Max Current
- 6080<sub>h</sub>: Max Motor Speed

### 7.5.6 Kategorie: Tuning

- 2030<sub>h</sub>: Pole Pair Count
- 2031<sub>h</sub>: Max Motor Current
- 203B<sub>h</sub>: I2t Parameters
- 3203<sub>h</sub>: Feedback Selection

- 3380<sub>h</sub>: Feedback Sensorless
- 33A0<sub>h</sub>: Feedback Incremental A/B/I 1
- 33A1<sub>h</sub>: Feedback Incremental A/B/I 2
- 4021<sub>h</sub>: Ballast Configuration
- 6075<sub>h</sub>: Motor Rated Current
- 608F<sub>h</sub>: Position Encoder Resolution
- 6090<sub>h</sub>: Velocity Encoder Resolution
- 60E6<sub>h</sub>: Additional Position Encoder Resolution - Encoder Increments
- 60EB<sub>h</sub>: Additional Position Encoder Resolution - Motor Revolutions

### 7.5.7 Kategorie: Modbus RTU

- 2028<sub>h</sub>: MODBUS Slave Address
- 202A<sub>h</sub>: MODBUS RTU Baudrate
- 202D<sub>h</sub>: MODBUS RTU Parity

### 7.5.8 Speichervorgang starten

**VORSICHT**



**Unkontrollierte Motorbewegungen!**

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

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

**HINWEIS**



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

Für jede *Kategorie* gibt es einen Subindex im Objekt 1010<sub>h</sub>. Um alle Objekte dieser *Kategorie* zu speichern, muss der Wert "65766173<sub>h</sub>" <sup>1</sup> 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<sub>h</sub> für welche *Kategorie* zuständig ist.

Subindex	Kategorie
01 <sub>h</sub>	Alle Kategorien mit der Ausnahme von 0B <sub>h</sub> (Modbus RTU)
02 <sub>h</sub>	Kommunikation
03 <sub>h</sub>	Applikation
04 <sub>h</sub>	Benutzer
05 <sub>h</sub>	Bewegung
06 <sub>h</sub>	Tuning
0B <sub>h</sub>	Modbus RTU

<sup>1</sup> Das entspricht dezimal der 1702257011<sub>d</sub> bzw. dem ASCII String `save`.

### 7.5.9 Speicherung verwerfen

Falls alle Objekte oder eine *Kategorie* an gespeicherten Objekten gelöscht werden sollen, muss in das Objekt 1011<sub>h</sub> der Wert "64616F6C<sub>h</sub>"<sup>2</sup> geschrieben werden. Folgende Subindizes entsprechen dabei einer *Kategorie*:

Subindex	Kategorie
01 <sub>h</sub>	Alle Kategorien (Zurücksetzen auf Werkseinstellung) mit der Ausnahme von 06 <sub>h</sub> (Tuning) und 0B <sub>h</sub> (Modbus RTU)
02 <sub>h</sub>	Kommunikation
03 <sub>h</sub>	Applikation
04 <sub>h</sub>	Benutzer
05 <sub>h</sub>	Bewegung
06 <sub>h</sub>	Tuning
0B <sub>h</sub>	Modbus RTU

Die gespeicherten Objekte werden daraufhin verworfen, die Änderung wirkt erst nach einem Neustart der Steuerung aus. Sie können die Steuerung neu starten, indem Sie den Wert "746F6F62<sub>h</sub>" in 2800<sub>h</sub>:01<sub>h</sub> eintragen.

#### HINWEIS



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

### 7.5.10 Konfiguration verifizieren

Das Objekt 1020<sub>h</sub> 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<sub>h</sub> können mit einem Datum und einer Uhrzeit beschrieben und anschließend über 1010<sub>h</sub>:01 zusammen mit allen anderen speicherbaren Objekten gespeichert werden.

Die Einträge von 1020<sub>h</sub> werden auf "0" zurückgesetzt, sobald ein beliebiges speicherbares Objekt (einschließlich 1010<sub>h</sub>:0x<sub>h</sub>, außer 1010<sub>h</sub>:01<sub>h</sub> und 1020<sub>h</sub>) 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<sub>h</sub>.
3. Das Tool oder der Master aktiviert das Speichern aller Objekte 1010<sub>h</sub>:01<sub>h</sub> = 65766173<sub>h</sub>. Das Datum und die Uhrzeit im Objekt 1020<sub>h</sub> werden ebenfalls abgespeichert.

Nach einem Neustart der Steuerung kann der Master den Wert in 1020<sub>h</sub>:01<sub>h</sub> und 1020:01<sub>h</sub> 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.

<sup>2</sup> Das entspricht dezimal der 1684107116<sub>d</sub> bzw. dem ASCII String load.

## 8 Modbus RTU

Modbus-Referenzen: [www.modbus.org](http://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.

### 8.1 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<sub>h</sub>) - 9998 (270E<sub>h</sub>) gemappt.
- Holding Register 40001- 49999 wird auf Modbus Telegram Adresse 0 (0<sub>h</sub>) - 9998 (270E<sub>h</sub>) gemappt.

#### HINWEIS



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

### 8.2 Allgemeines

Modbus ist generell Big-Endian basiert.

Die einzigen Ausnahmen bilden dabei die Kommandos mit den Funktionscodes 43 (2B<sub>h</sub>) , 101 (65<sub>h</sub>) und 102 (66<sub>h</sub>) 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<sub>h</sub>: Mit diesem Kommando wird der Wert 12345678<sub>h</sub> in das Objekt 0123<sub>h</sub> (existiert nicht) geschrieben:

#### FC

Funktionscode

#### Daten

Datenbereich, Decodierung ist abhängig vom benutzen Funktionscode

#### TIPP



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

## 8.3 Kommunikationseinstellungen

Slave-Adresse, Baudrate und Parität ergeben sich abhängig von der Position des *Drehschalters* S1 und ggf. noch von den Objekten 2028<sub>h</sub>, 202A<sub>h</sub>, 202D<sub>h</sub>.

Konfiguration	Objekt	Wertebereich	Werkseinstellung
Slave Adresse	<u>2028<sub>h</sub></u>	1 bis 247	5
Baudrate	<u>202A<sub>h</sub></u>	7200 bis 256000	19200
Parity	<u>202D<sub>h</sub></u>	<ul style="list-style-type: none"> <li>■ None: 0x00</li> <li>■ Even: 0x04</li> <li>■ Odd: 0x06</li> </ul>	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

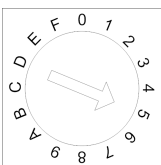
Unterstützt werden folgende Baudraten:

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

Sie müssen die Änderungen speichern, indem Sie den Wert "65766173<sub>h</sub>" in das Objekt 1010<sub>h</sub>:0B<sub>h</sub> schreiben. Die Änderungen werden erst nach einem Neustart der Steuerung übernommen.

### 8.3.1 Drehschalter

Der *PD6-E* verfügt über einen Hex-Codierschalter - ähnlich wie in der nachfolgenden Abbildung.



Sie können damit die Quelle für die Slave-Adresse, die Baudrate und die Parität bestimmen, die nachfolgende Tabelle zeigt die möglichen Kombinationen.

Position des Drehschalters		Slave-Adresse	Baudrate und Parität
dec	hex		
0	0	Objekt <u>2028<sub>h</sub></u>	Objekt <u>202A<sub>h</sub></u> bzw. <u>202D<sub>h</sub></u>
1-14	1-E	Zahl des Drehschalters	Objekt <u>202A<sub>h</sub></u> bzw. <u>202D<sub>h</sub></u>
15	F	5	19200, even Parity



## 8.4 Funktionscodes

Die folgenden "Funktionscodes" werden unterstützt:

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

## 8.5 Funktioncode-Beschreibungen

### 8.5.1 FC 3 (03<sub>h</sub>) Read Input Registers / FC 4 (04<sub>h</sub>) 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 <sub>h</sub> / 04 <sub>h</sub>
Startadresse	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
Anzahl der Register	2 Bytes	1 bis (7D <sub>h</sub> )
CRC	2 Bytes	

Response ("M" entspricht der Anzahl der zu lesenden Register)		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	03 <sub>h</sub> / 04 <sub>h</sub>

Response ("M" entspricht der Anzahl der zu lesenden Register)		
Name	Länge	Wert
Anzahl Bytes	1 Byte	2 * M
Registerwert	2 Bytes	
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	83 <sub>h</sub> / 84 <sub>h</sub>
Ausnahmecode (siehe <a href="#">Ausnahmecodes</a> )	1 Byte	01, 02, 03 oder 04
CRC	2 Bytes	

### Beispiel

Nachfolgend ein Beispiel eines Lese-Request und Response des Registers 5000 (1388<sub>h</sub>) 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

## 8.5.2 FC 6 (06<sub>h</sub>) 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 <sub>h</sub>
Registeradresse	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
Registerwert	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
CRC	2 Bytes	

Response		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	06 <sub>h</sub>

Response		
Name	Länge	Wert
Registeradresse	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
Registerwert	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	86 <sub>h</sub>
Ausnahmecode (siehe <a href="#">Ausnahmecodes</a> )	1 Byte	01, 02, 03 oder 04
CRC	2 Bytes	

### Beispiel

Nachfolgend ein Beispiel eines Write-Request und Response in das Register 6000 (1770<sub>h</sub>) mit dem Wert "0001<sub>h</sub>":

#### 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

### 8.5.3 FC 16 (10<sub>h</sub>) 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 <sub>h</sub>
Startadresse	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
Anzahl der Register	2 Bytes	0001 <sub>h</sub> bis 007B <sub>h</sub>
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 <sub>h</sub>
Startadresse	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
Anzahl der Register	2 Bytes	0001 <sub>h</sub> bis 007B <sub>h</sub>
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	90 <sub>h</sub>
Ausnahmecode (siehe <a href="#">Ausnahmecodes</a> )	1 Byte	01, 02, 03 oder 04
CRC	2 Bytes	

### Beispiel

Nachfolgend ein Beispiel eines Mehrfach-Schreibens der Werte "0102<sub>h</sub>" und "0304<sub>h</sub>" startend ab Registeradresse 6000 (1770<sub>h</sub>), 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

### 8.5.4 FC 17 (11<sub>h</sub>) Report Server ID

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

Request		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	11 <sub>h</sub>
CRC	2 Bytes	

Response		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	03 <sub>h</sub>

Response		
Name	Länge	Wert
Anzahl Bytes	1 Byte	01 <sub>h</sub>
Run Indicator Status	1 Byte	00 <sub>h</sub> = OFF, FF <sub>h</sub> = ON
Zusatzdaten		
CRC	2 Bytes	

Fehler		
Name	Länge	Wert
Slave-Adresse	1 Byte	
Fehlercode	1 Byte	91 <sub>h</sub>
Ausnahmecode (siehe <a href="#">Ausnahmecodes</a> )	1 Byte	01 oder 04
CRC	2 Bytes	

### Beispiel

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

#### Request

SA	FC	CRC
05	11	C2 EC

#### Response

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

### 8.5.5 FC 23 (17<sub>h</sub>) 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 <sub>h</sub>
Lesen: Startadresse	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
Lesen: Anzahl Register	2 Bytes	0001 <sub>h</sub> bis 0079 <sub>h</sub>
Schreiben: Startadresse	2 Bytes	0000 <sub>h</sub> bis FFFF <sub>h</sub>
Schreiben: Anzahl Register	2 Bytes	0001 <sub>h</sub> bis 0079 <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
Ausnahmecode (siehe <a href="#">Ausnahmecodes</a> )	1 Byte	01, 02, 03 oder 04
CRC	2 Bytes	

### Beispiel

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

#### 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

## 8.5.6 FC 8 (08<sub>h</sub>) 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 <sub>h</sub>
Unterfunktionscode	2 Bytes	
Data	N x 2 Bytes	

Response:

Name	Länge	Wert
Funktionscode	1 Byte	08 <sub>h</sub>
Unterfunktionscode	2 Bytes	
Data	N x 2 Bytes	

Fehler:

Name	Länge	Wert
Funktionscode	1 Byte	88 <sub>h</sub>
Ausnahmecode (siehe <a href="#">Ausnahmecodes</a> )	1 Bytes	01 oder 03 oder 04

### 8.5.6.1 FC 8.10 (08<sub>h</sub>.0A<sub>h</sub>) 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 <sub>h</sub> 0A <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	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

### 8.5.6.2 FC 8.11 (08<sub>h</sub>.0B<sub>h</sub>) 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 <sub>h</sub> 0B <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	Total Message Count

### 8.5.6.3 FC 8.12 (08<sub>h</sub>.0C<sub>h</sub>) 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 <sub>h</sub> 0C <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	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

### 8.5.6.4 FC 8.13 (08<sub>h</sub>.0D<sub>h</sub>) 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 <sub>h</sub> 0D <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	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

### 8.5.6.5 FC 8.14 (08<sub>h</sub>.0E<sub>h</sub>) 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 <sub>h</sub> 0E <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	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

**8.5.6.6 FC 8.15 (08<sub>h</sub>-0F<sub>h</sub>) 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 <sub>h</sub> 0F <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	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

**8.5.6.7 FC 8.16 (08<sub>h</sub>-10<sub>h</sub>) 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 <sub>h</sub> 10 <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	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

**8.5.6.8 FC 8.17 (08<sub>h</sub>-11<sub>h</sub>) 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 <sub>h</sub> 11 <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	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

**8.5.6.9 FC 8.18 (08<sub>h</sub>-12<sub>h</sub>) 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 <sub>h</sub> 12 <sub>h</sub>	00 <sub>h</sub> 00 <sub>h</sub>	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

**8.5.7 FC 43 (2B<sub>h</sub>) 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 <sub>h</sub> (43 <sub>d</sub> )
MEI type	1 Byte	0D <sub>h</sub> (13 <sub>d</sub> )
Protokolloptionen Bereich	2 bis 5 Byte	
Adressen- und Datenbereich	N Bytes	
CRC	2 Bytes	

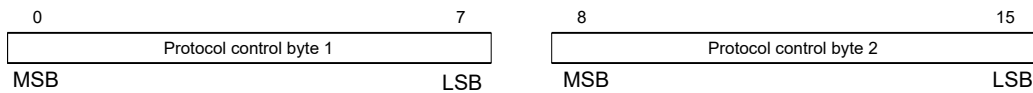
**Protokolloptionen Bereich**

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	

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

### Adressen- und Datenbereich

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

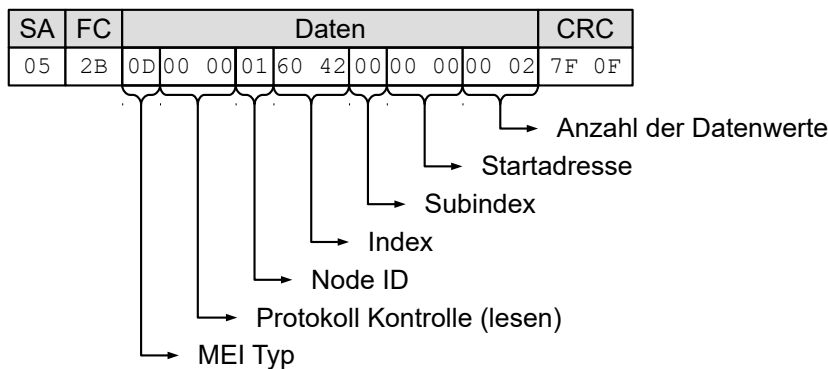
Name	Bytegröße und Bytereihenfolge	Beispiel / Bereich
Node-ID	1 Byte	01 <sub>h</sub> bis 7F <sub>h</sub>
Index	1 Byte, high	0000 <sub>h</sub> bis FFFF <sub>h</sub>
	1 Byte, low	
Subindex	1 Byte	00 <sub>h</sub> bis FF <sub>h</sub>
Startadresse	1 Byte, high	0000 <sub>h</sub> bis FFFF <sub>h</sub>
	1 Byte, low	

Name	Bytgröße und Bytereihenfolge	Beispiel / Bereich
Anzahl der Datenwerte	1 Byte, high	0000 <sub>h</sub> bis 00FD <sub>h</sub>
Schreib-/Lesedaten	1 Byte, low	Die Daten sind codiert wie in Kapitel <u>Allgemeines</u> beschrieben.

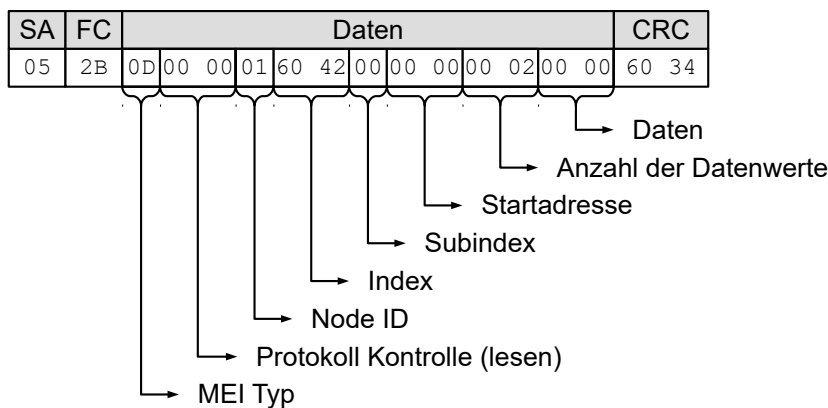
**Beispiel:**

Um das Objekt 6042<sub>h</sub>:00<sub>h</sub> 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<sub>h</sub>" (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	A9 89

**Setze 2031 = 03E8<sub>h</sub>" (1000 mA)****Request**

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

**Response**

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

**Setze 6040 = "00<sub>h</sub>"****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<sub>h</sub>"****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<sub>h</sub>"****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<sub>h</sub>"****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<sub>h</sub>"****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<sub>h</sub>:00<sub>h</sub>****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<sub>h</sub>:00<sub>h</sub>****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

### 8.5.7.1 Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

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

CANopen-Fehlercode	Beschreibung
FFFF0000 <sub>h</sub>	<i>Abort no error</i>
FFFF1003 <sub>h</sub>	Service wird nicht unterstützt
FFFF1004 <sub>h</sub>	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>
FFFF0003 <sub>h</sub>	Unbekanntes oder nicht valides Kommando
FFFF0008 <sub>h</sub>	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E <sub>h</sub>	Allgemeiner Fehler im Parameter
FFFF0011 <sub>h</sub>	Länge des Parameters falsch
FFFF0012 <sub>h</sub>	Länge des Parameters zu groß
FFFF0013 <sub>h</sub>	Länge des Parameters zu klein
FFFF0015 <sub>h</sub>	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 <sub>h</sub>	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 <sub>h</sub>	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 <sub>h</sub>	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 <sub>h</sub>	Allgemeiner Fehler
FFFF001E <sub>h</sub>	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 <sub>h</sub>	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 <sub>h</sub> + 80 <sub>h</sub> (171 <sub>d</sub> = 43 <sub>d</sub> + 128 <sub>d</sub> ) (zeigt Fehler an)
Modbus exception code	1 Byte	FF <sub>h</sub> ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D <sub>h</sub>
Exception code	1 Byte	AE <sub>h</sub>
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<sub>h</sub>: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

### 8.5.8 FC 101 (65<sub>h</sub>) Read complete object dictionary

Dieser Funktionscode wird zum Auslesen des gesamten Objektverzeichnisses verwendet.

Um das Auslesen des Objektverzeichnisses zu starten oder neu zu starten, muss der Unterfunktionscode 55<sub>h</sub> versendet werden. Dieser Code setzt das Auslesen des Objektverzeichnisses auf das Objekt 0000<sub>h</sub> zurück. Alle nachfolgenden Objektverzeichnis-Frames müssen dann den Unterfunktionscode AA<sub>h</sub> 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 <sub>h</sub>
Unterfunktionscode	1 Byte	55 <sub>h</sub> oder AA <sub>h</sub>
Länge der Daten	1 Byte	00 <sub>h</sub>
CRC	2 Bytes	

#### Response:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	65 <sub>h</sub>
Funktionscode	1 Byte	
Unterfunktionscode	1 Byte	
Länge der Daten	1 Byte	
n mal "Objektverzeichnis-Frame"	1 - 252 Bytes	
CRC	2 Bytes	

Ein Objektverzeichnis-Frame besteht aus den folgenden Bytes:

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

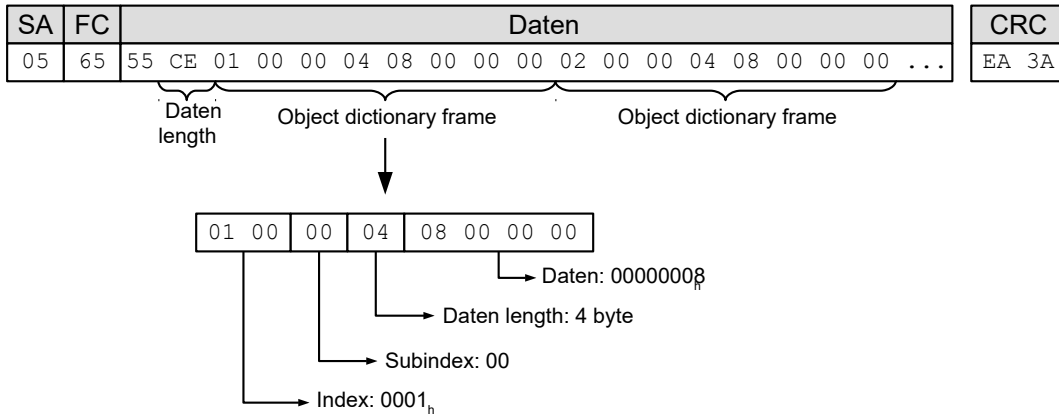
#### Beispiel

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

Start des Auslesens des Objektverzeichnisses mit dem Request:

SA	FC	Daten	CRC
05	65	55 00	2F A7

Die Response ist:



Den nächsten Teil des Objektverzeichnisses auslesen mit dem Request:

SA	FC	Daten	CRC
05	65	AA 00	6E 57

Die Response ist:

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

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

SA	FC	Daten	CRC
05	E5	0D	EA 94

### 8.5.8.1 Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

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

CANopen-Fehlercode	Beschreibung
FFFF0000 <sub>h</sub>	<i>Abort no error</i>
FFFF1003 <sub>h</sub>	Service wird nicht unterstützt
FFFF1004 <sub>h</sub>	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>

CANopen-Fehlercode	Beschreibung
FFFF0003 <sub>h</sub>	Unbekanntes oder nicht valides Kommando
FFFF0008 <sub>h</sub>	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E <sub>h</sub>	Allgemeiner Fehler im Parameter
FFFF0011 <sub>h</sub>	Länge des Parameters falsch
FFFF0012 <sub>h</sub>	Länge des Parameters zu groß
FFFF0013 <sub>h</sub>	Länge des Parameters zu klein
FFFF0015 <sub>h</sub>	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 <sub>h</sub>	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 <sub>h</sub>	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 <sub>h</sub>	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 <sub>h</sub>	Allgemeiner Fehler
FFFF001E <sub>h</sub>	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 <sub>h</sub>	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 <sub>h</sub> + 80 <sub>h</sub> (171 <sub>d</sub> = 43 <sub>d</sub> + 128 <sub>d</sub> ) (zeigt Fehler an)
Modbus exception code	1 Byte	FF <sub>h</sub> ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D <sub>h</sub>
Exception code	1 Byte	AE <sub>h</sub>
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

Bit	Name	Beschreibung
		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<sub>h</sub>: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

### 8.5.9 FC 102 (66<sub>h</sub>) 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<sub>h</sub> versendet werden. Dieser Code setzt das Auslesen auf das Objekt mit Subindex 00<sub>h</sub> zurück. Alle nachfolgenden Requests müssen dann den Unterfunktionscode AA<sub>h</sub> 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 <sub>h</sub>
Unterfunktionscode	1 Byte	55 <sub>h</sub> oder AA <sub>h</sub>
Länge der Daten	1 Byte	00 <sub>h</sub>
Index des zu lesenden Arrays	2 Bytes	
CRC	2 Bytes	

#### Response:

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	65 <sub>h</sub>
Funktionscode	1 Byte	
Unterfunktionscode	1 Byte	

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

Ein Objektverzeichnis-Frame besteht aus den folgenden Bytes:

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

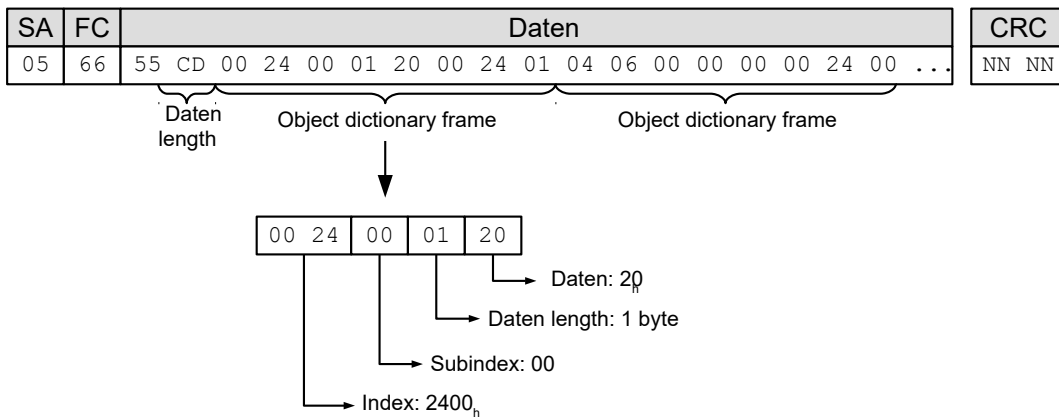
**Beispiel**

Alle folgenden Zahlenwerte sind in Hexadezimal notiert, der Index des zu lesenden Objektes ist 2400<sub>h</sub>. Die Adresse des Slaves ist "5"<sub>h</sub>.

Start des Auslesens des Arrays mit dem Request:

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

Die Response ist:



**8.5.9.1 Fehlerreaktion**

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	2B <sub>h</sub> + 80 <sub>h</sub> (171 <sub>d</sub> = 43 <sub>d</sub> + 128 <sub>d</sub> ) (zeigt Fehler an)
Modbus exception code	1 Byte	FF <sub>h</sub> ("extended exception")
Extended exception Länge	2 Bytes	6

Name	Länge	Beispielwert
MEI type	1 Byte	0D <sub>h</sub>
Exception code	1 Byte	CE <sub>h</sub>
Fehlercode	4 Bytes	CANopen-Fehlercode, siehe nachfolgende Tabelle
CRC	2 Bytes	

CANopen-Fehlercode	Beschreibung
FFFF0000 <sub>h</sub>	<i>Abort no error</i>
FFFF1003 <sub>h</sub>	Service wird nicht unterstützt
FFFF1004 <sub>h</sub>	Lücke im Counter-Byte des Felds <i>Protokoll-Kontrolle</i>
FFFF0003 <sub>h</sub>	Unbekanntes oder nicht valides Kommando
FFFF0008 <sub>h</sub>	Zugriff auf das Objekt wird nicht unterstützt
FFFF000E <sub>h</sub>	Allgemeiner Fehler im Parameter
FFFF0011 <sub>h</sub>	Länge des Parameters falsch
FFFF0012 <sub>h</sub>	Länge des Parameters zu groß
FFFF0013 <sub>h</sub>	Länge des Parameters zu klein
FFFF0015 <sub>h</sub>	Parameter-Daten außerhalb des gültigen Wertebereichs (für Write-Kommandos)
FFFF0016 <sub>h</sub>	Parameter-Daten überschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0017 <sub>h</sub>	Parameter-Daten unterschreiten den gültigen Wertebereich (für Write-Kommandos)
FFFF0018 <sub>h</sub>	Maximale eingegebene Werte kleiner als minimale Werte
FFFF0019 <sub>h</sub>	Allgemeiner Fehler
FFFF001E <sub>h</sub>	Angefordertes Objekt ist zu groß für einzelne Nachricht
FFFF1004 <sub>h</sub>	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 <sub>h</sub> + 80 <sub>h</sub> (171 <sub>d</sub> = 43 <sub>d</sub> + 128 <sub>d</sub> ) (zeigt Fehler an)
Modbus exception code	1 Byte	FF <sub>h</sub> ("extended exception")
Extended exception length	2 Bytes	2 + Länge von "Supported protocol control"
MEI type	1 Byte	0D <sub>h</sub>
Exception code	1 Byte	AE <sub>h</sub>
Supported protocol control	1 oder 2 Bytes	Siehe nachfolgende Tabelle
CRC	2 Bytes	

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

Das nachfolgende Beispiel zeigt einen Fehler im Falle eines fehlerhaften Request. Der Request liest das 6061<sub>h</sub>: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

### 8.5.10 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*.



## 8.6 Prozessdatenobjekte (PDO)

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

### 8.6.1 Konfiguration

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

- 3502<sub>h</sub> für das Modbus Rx (Master → Slave) PDO-Mapping
- 3602<sub>h</sub> 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<sub>h</sub> und 3602<sub>h</sub> lassen sich mit Nachrichten mit dem Modbus-Funktionscode 2B<sub>h</sub> beschreiben.

### 8.6.2 Übertragung

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

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

Index	Datentyp
0002 <sub>h</sub>	Vorzeichenbehaftete Ganzzahl (8 Bit)
0003 <sub>h</sub>	Vorzeichenbehaftete Ganzzahl (16 Bit)
0004 <sub>h</sub>	Vorzeichenbehaftete Ganzzahl (32 Bit)
0005 <sub>h</sub>	Vorzeichenlose Ganzzahl (8 Bit)
0006 <sub>h</sub>	Vorzeichenlose Ganzzahl (16 Bit)
0007 <sub>h</sub>	Vorzeichenlose Ganzzahl (32 Bit)

Das Mapping ist wie folgt:

- Das PDO RX Image fängt an der Modbus-Register-Adresse 6000<sub>d</sub> (1770<sub>h</sub>) an.
- Das PDO TX Image fängt an der Modbus-Register-Adresse 5000<sub>d</sub> (1388<sub>h</sub>) an.

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

#### HINWEIS



Um das Mapping ändern zu können, müssen Sie es zuerst deaktivieren, indem Sie den entsprechenden Subindex 0<sub>h</sub> auf "0" setzen.

Nach dem Schreiben der Objekte in die jeweiligen Subindizes, tragen Sie die Anzahl der gemappten Objekte in den Subindex 0<sub>h</sub> ein.

### Beispiel

In dem Mapping sollen folgende Objekte eingestellt werden:

- 3602<sub>h</sub>:00<sub>h</sub> = "0<sub>h</sub>" (Mapping wird deaktiviert)
- 3602<sub>h</sub>:01<sub>h</sub> = "60410010<sub>h</sub>" (das Objekt 6041<sub>h</sub>:00<sub>h</sub>, Länge 16 Bit wird gemappt)
- 3602<sub>h</sub>:02<sub>h</sub> = "00050008<sub>h</sub>" (das Dummy-Objekt 0005<sub>h</sub>:00<sub>h</sub>, Länge 8 Bit wird gemappt)
- 3602<sub>h</sub>:03<sub>h</sub> = "60610008<sub>h</sub>" (das Objekt 6061<sub>h</sub>:00<sub>h</sub>, 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:

SA	FC	Daten	CRC
05	04	13 88 00 07	34 E2

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

SA	FC	Daten												CRC				
05	04	0E	06	40	00	02	00	00	00	00	00	00	00	00	00	00	1C	98

### 8.7 NanoJ-Objekte

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

- $2500_h$  mit 32 x 32 Bit Werten wird auf die Modbus Register Adresse ab  $2000_d$  ( $7D0_h$ ) 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	00	00	00	00	00	00	00	00	50	9D

## 9 Programmierung mit NanoJ

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

### 9.1 NanoJ-Programm

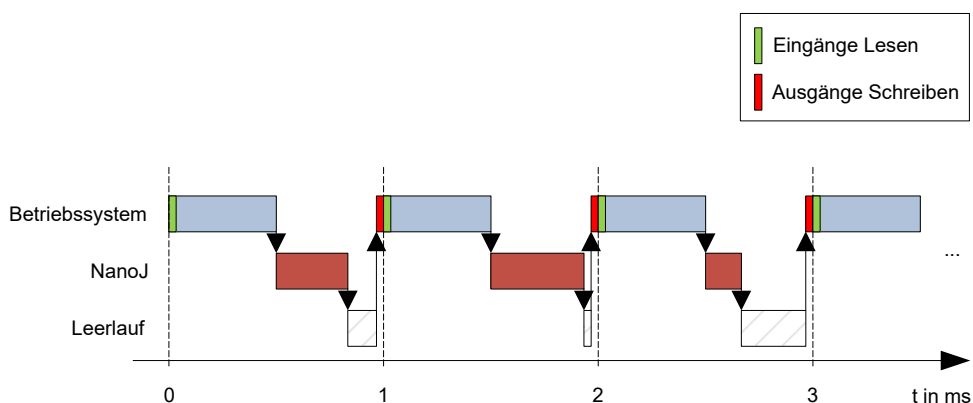
Ein *NanoJ-Programm* stellt eine geschützte Ausführungsumgebung innerhalb der Firmware zur Verfügung. In dieser kann der Anwender eigene Abläufe anlegen. Diese können dann Funktionen in der Steuerung auslösen, indem beispielsweise Einträge im Objektverzeichnis gelesen oder geschrieben werden.

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

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

#### 9.1.1 Verfügbare Rechenzeit

Ein *NanoJ-Programm* erhält zyklisch im 1 ms-Takt Rechenzeit (siehe folgende Abbildung). Da durch Interrupts und Systemfunktionen der Firmware Rechenzeit verloren geht, 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<sub>h</sub> die Ziffer 4 eingetragen, im Fehlerregister bei Objekt 2302<sub>h</sub> 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<sub>h</sub> schreiben. Im *AutoYield*-Modus ist aber das *NanoJ-Programm* nicht mehr echtzeitfähig und läuft nicht mehr im 1-Millisekunde-Takt.

## 9.1.2 Geschützte Ausführungsumgebung

Durch prozessorspezifische Eigenschaften wird eine sogenannte *Geschützte Ausführungsumgebung* generiert. Ein Benutzerprogramm in der geschützten Ausführungsumgebung hat nur die Möglichkeit, auf speziell zugewiesene Speicherbereiche und Systemressourcen zuzugreifen. Beispielsweise wird ein Versuch, auf ein Prozessor-IO-Register direkt zu schreiben, mit einem *MPU Fault* quittiert und das Benutzerprogramm wird mit dem entsprechenden Fehlercode im Objektverzeichnis abgebrochen.

## 9.1.3 NanoJ-Programm - Kommunikationsmöglichkeiten

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

- Lesen und Schreiben von OD-Werten per PDO-Mapping
- direktes Lesen und Schreiben von OD-Werten über NanoJ-Funktionen
- Aufruf sonstiger NanoJ-Funktionen (z. B. [Debug-Ausgabe](#) schreiben)

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

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

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

Die gesetzten Mappings können über die GUI bei den Objekten 2310<sub>h</sub>, 2320<sub>h</sub>, und 2330<sub>h</sub> 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<sub>h</sub>.
- [2500h NanoJ Outputs](#): Array mit zweiunddreißig S32-Werten, wo das *NanoJ-Programm* Werte ablegen kann, die über den Feldbus ausgelesen werden können

## 9.1.4 NanoJ-Programm ausführen

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

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

Die Konfiguration der Kopiervorgänge ist dem CANopen-Standard angelehnt.

Zusätzlich kann über NanoJ-Funktionen auf Werte des Objektverzeichnisses zugegriffen werden. Dies ist im Allgemeinen deutlich langsamer und daher sind Mappings vorzuziehen. Die Anzahl an Mappings ist begrenzt (jeweils 16 Einträge in In/Out/InOut).

#### TIPP



Nanotec empfiehlt: Häufig genutzte und veränderte OD-Einträge mappen und auf weniger häufig genutzte OD-Einträge per NanoJ-Funktion zuzugreifen.

Eine Liste verfügbarer NanoJ-Funktionen findet sich im Kapitel [NanoJ-Funktionen im NanoJ-Programm](#).

#### TIPP



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

### 9.1.5 NanoJ-Programm OD-Einträge

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

OD-Index	Name und Beschreibung
2300 <sub>h</sub>	<a href="#">2300h NanoJ Control</a>
2301 <sub>h</sub>	<a href="#">2301h NanoJ Status</a>
2302 <sub>h</sub>	<a href="#">2302h NanoJ Error Code</a>
2310 <sub>h</sub>	<a href="#">2310h NanoJ Input Data Selection</a>
2320 <sub>h</sub>	<a href="#">2320h NanoJ Output Data Selection</a>
2330 <sub>h</sub>	<a href="#">2330h NanoJ In/output Data Selection</a>

#### Beispiel:

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

- Überprüfen des Eintrags [2302<sub>h</sub>](#) auf Fehlercode.
- Wenn kein Fehler:  
*NanoJ-Programm* starten durch Beschreiben von Objekt [2300<sub>h</sub>](#), Bit 0 = "1" bzw. durch Neustarten der Steuerung.

#### HINWEIS



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

- Überprüfen des Eintrags [2302<sub>h</sub>](#) auf Fehlercode und des Objekts [2301<sub>h</sub>](#), Bit 0 = "1".

Um ein laufendes Programm anzuhalten: Beschreiben des Eintrags [2300<sub>h</sub>](#) mit dem Bit 0 Wert = "0".

### 9.1.6 Aufbau NanoJ-Programm

Ein Benutzerprogramm besteht aus mindestens zwei Anweisungen:

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

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

#### HINWEIS



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

#### HINWEIS



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

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

#### Beispiele:

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

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

Folgende Zuweisung führt zu einem Fehler beim Kompilieren:

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

### 9.1.7 NanoJ-Programmbeispiel

Das Beispiel zeigt das Programmieren eines Rechtecksignals in das Objekt `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](http://www.nanotec.de).

## 9.2 Mapping im NanoJ-Programm

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

### TIPP

Nanotec empfiehlt:



- Benutzen Sie das Mapping, falls Sie den Zugriff auf ein Objekt im Objektverzeichnis häufiger benötigen, z. B. das *Controlword* 6040<sub>h</sub> oder das *Statusword* 6041<sub>h</sub>.
- Für den einzelnen Zugriff auf Objekte bieten sich eher die Funktionen `od_write()` und `od_read()` an, siehe [Zugriff auf das Objektverzeichnis](#).

### 9.2.1 Deklaration des Mappings

Die Deklaration des Mappings gliedert sich dabei folgendermaßen:

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

Dabei gilt:

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

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

### HINWEIS



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

## 9.2.2 Beispiel eines Mappings

Beispiel eines Mappings und der zugehörigen Variablenzugriffe:

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

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

#include "wrapper.h"

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

## 9.2.3 Möglicher Fehler bei `od_write()`

Eine mögliche Fehlerquelle ist ein schreibender Zugriff mittels der Funktion `od_write()` (siehe [NanoJ-Funktionen im NanoJ-Programm](#)) auf ein Objekt im Objektverzeichnis, welches gleichzeitig als Mapping angelegt wurde. Nachfolgend aufgelisteter Code ist fehlerhaft:

```
map U16 controlWord as output 0x6040:00
#include " wrapper.h"
void user()
{
  [...]
  Out.controlWord = 1;
  [...]
  od_write(0x6040, 0x00, 5 ); // der Wert wird durch das Mapping überschrieben
  [...]
}
```

Die Zeile mit dem Befehl `od_write(0x6040, 0x00, 5 );` ist wirkungslos. Wie in der Einleitung beschrieben, werden alle Mappings am Ende jeder Millisekunde in das Objektverzeichnis kopiert.

Damit ergibt sich folgender Ablauf:

1. Die Funktion `od_write` schreibt den Wert 5 in das Objekt `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.

## 9.3 NanoJ-Funktionen im NanoJ-Programm

Mit NanoJ-Funktionen ist es möglich, in der Firmware eingebaute Funktionen direkt aus einem Benutzerprogramm aufzurufen. Eine direkte Code-Ausführung ist nur in dem geschützten Bereich der geschützten Ausführungsumgebung möglich und wird über sogenannte *Cortex-Supervisor-Calls* (Svc Calls) realisiert. Dabei wird mit dem Aufruf der Funktion ein Interrupt ausgelöst und die Firmware hat so die Möglichkeit, temporär eine Code-Ausführung außerhalb der geschützten Ausführungsumgebung zuzulassen. Der Entwickler des Benutzerprogramms muss sich jedoch um diesen Mechanismus nicht kümmern - für ihn sind die NanoJ-Funktionen wie ganz normale C-Funktionen aufrufbar. Lediglich die Datei *wrapper.h* muss - wie üblich - eingebunden werden.



### 9.3.1 Zugriff auf das Objektverzeichnis

void **od\_write** (U32 index, U32 subindex, U32 value)

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

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

#### HINWEIS



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

U32 **od\_read** (U32 index, U32 subindex)

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

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

#### HINWEIS



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

#### Beispiel

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

### 9.3.2 Prozesssteuerung

void **yield**()

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

void **sleep** (U32 ms)

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

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

### 9.3.3 Debug-Ausgabe

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

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

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

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

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

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

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

**HINWEIS**



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

Dieser OD-Eintrag hat den Index 2600<sub>h</sub> und ist 64 Zeichen lang, siehe [2600h NanoJ Debug Output](#). In Subindex 00 ist immer die Anzahl der bereits geschriebenen Zeichen enthalten.

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

**HINWEIS**



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

**HINWEIS**



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

### 9.4 Einschränkungen und mögliche Probleme

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

Einschränkung/Problem	Maßnahme
Wenn ein Objekt gemappt wird, z. B. 0x6040, wird das Objekt alle 1 ms auf seinen vorherigen Wert zurückgesetzt. Das macht die Steuerung dieses Objekts über den Feldbus oder das <i>Plug &amp; 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.	

## 10 Objektverzeichnis Beschreibung

### 10.1 Übersicht

In diesem Kapitel finden Sie eine Beschreibung aller Objekte.

Sie finden hier Angaben zu:

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

### 10.2 Aufbau der Objektbeschreibung

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

#### **Funktion**

In diesem Abschnitt wird kurz die Funktion des Objektverzeichnisses beschrieben.

#### **Objektbeschreibung**

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

#### **Wertebeschreibung**

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

#### **Beschreibung**

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

### 10.3 Objektbeschreibung

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

#### **Index**

Benennt den Index des Objekts in Hexadezimalschreibweise.

#### **Objektname**

Der Name des Objekts.

#### **Object Code**

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

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

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

### Datentyp

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

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

### Speicherbar

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

### Firmware Version

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

### Änderungshistorie (ChangeLog)

Hier werden eventuelle Änderungen an dem Objekt notiert.

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

### Zugriff

Hier wird die Zugriffsbeschränkung eingetragen. Folgende Beschränkungen gibt es:

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

### PDO-Mapping

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

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

## 10.4 Wertebeschreibung

### HINWEIS



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

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

**Subindex**

Nummer des aktuell beschriebenen Untereintrages.

**Name**

Der Name des Untereintrages.

**Datentyp**

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

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

**Zugriff**

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

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

**PDO-Mapping**

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

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

**Zulässige Werte**

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

**Vorgabewert**

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

**10.5 Beschreibung**

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

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

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

**Beispiel [4]**

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

**Beispiel [2]**

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

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

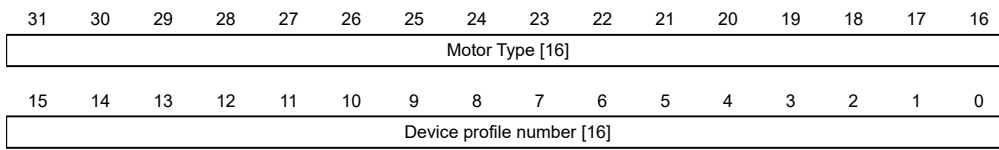
**1000h Device Type****Funktion**

Beschreibt den Steuerungstyp.

**Objektbeschreibung**

Index	1000 <sub>h</sub>
Objektname	Device Type
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00040192<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00040192<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 00020192<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 00020192<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00040192<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00040192<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00040192<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 00040192<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00020192<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00020192<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 00020192<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 00020192<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 00020192<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 00020192<sub>h</sub></li> </ul>
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<sub>h</sub> bzw. 0402<sub>d</sub> (Vorgabewert): Der CiA 402-Standard wird unterstützt.

## 1001h Error Register

### Funktion

Fehlerregister: Im Fehlerfall wird das entsprechende Fehlerbit gesetzt. Sollte der Fehler nicht mehr bestehen, wird es automatisch wieder gelöscht.



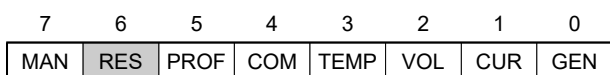
#### HINWEIS

Für jeden aufgetretenen Fehler wird im Objekt 1003<sub>h</sub> ein genauerer Fehlercode hinterlegt.

## Objektbeschreibung

Index	1001 <sub>h</sub>
Objektname	Error Register
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## Beschreibung





**GEN**

Genereller Fehler

**CUR**

Strom

**VOL**

Spannung

**TEMP**

Temperatur

**COM**

Kommunikation

**PROF**

Betrifft das Geräteprofil

**RES**

Reserviert, immer "0"

**MAN**

Hersteller-spezifisch

**1003h Pre-defined Error Field****Funktion**

Dieses Objekt beinhaltet einen Fehlerstapel mit bis zu acht Einträgen.

**Objektbeschreibung**

Index	1003 <sub>h</sub>
Objektname	Pre-defined Error Field
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	

**Wertebeschreibung**

Subindex	00 <sub>h</sub>
Name	Number Of Errors
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

Subindex	01 <sub>h</sub>
----------	-----------------

Name	1st Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	2nd Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	03 <sub>h</sub>
Name	3th Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	04 <sub>h</sub>
Name	4th Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	05 <sub>h</sub>
Name	5th Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	06 <sub>h</sub>
Name	6th Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	07 <sub>h</sub>
Name	7th Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	08 <sub>h</sub>
Name	8th Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 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<sub>h</sub>) beantwortet. Wird in den Subindex 0 eine "0" geschrieben, beginnt die Zählung von neuem.

### Bitbeschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Error Number [8]								Error Class [8]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Error Code [16]															

### Error Number [8]

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

Fehlernummer	Beschreibung
0	Watchdog-Reset
1	Eingangsspannung (+Ub) zu hoch
2	Ausgangsstrom zu hoch
3	Eingangsspannung (+Ub) zu niedrig

Fehlernummer	Beschreibung
4	Fehler am Feldbus
6	Nur CANopen: NMT-Master braucht zu lange, um Nodeguarding-Anforderung zu schicken
7	Sensor 1 (siehe 3204 <sub>h</sub> ): Fehler durch elektrische Störung oder defekte Hardware
8	Sensor 2 (siehe 3204 <sub>h</sub> ): Fehler durch elektrische Störung oder defekte Hardware
9	Sensor 3 (siehe 3204 <sub>h</sub> ): Fehler durch elektrische Störung oder defekte Hardware
10	Warnung: Positiver Endschalte überschritten
11	Warnung: Negativer Endschalte überschritten
12	Übertemperatur-Fehler
13	Die Werte des Objekts 6065 <sub>h</sub> (Following Error Window) und des Objekts 6066 <sub>h</sub> (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 <sub>h</sub> ), 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 <sub>h</sub> :01 <sub>h</sub> /6075 <sub>h</sub> )
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
40	Warnung: Ballast-Widerstand thermisch überlastet
43	Eingangsspannung (+Ub Logic) zu hoch
44	Eingangsspannung (+Ub Logic) zu niedrig
46	Interlock-Fehler: Bit 3 in 60FD <sub>h</sub> steht auf "0", der Motor darf nicht fahren (siehe Abschnitt <i>Interlock-Funktion</i> im Kapitel <i>Digitale Eingänge</i> )

### Error Class[8]

Dieses Byte ist identisch mit dem Objekt 1001<sub>h</sub>

## Error Code[16]

Die Bedeutung der beiden Bytes lässt sich aus der nachfolgenden Tabelle entnehmen.

Error Code	Beschreibung
1000 <sub>h</sub>	Allgemeiner Fehler
2300 <sub>h</sub>	Strom am Ausgang der Steuerung zu groß
3100 <sub>h</sub>	Über-/ Unterspannung am Eingang der Steuerung
4200 <sub>h</sub>	Temperaturfehler innerhalb der Steuerung
5440 <sub>h</sub>	Interlock-Fehler: Bit 3 in 60FD <sub>h</sub> steht auf "0", der Motor darf nicht fahren (siehe Abschnitt <i>Interlock-Funktion</i> im Kapitel <u>Digitale Eingänge</u> )
6010 <sub>h</sub>	Software-Reset (Watchdog)
6100 <sub>h</sub>	Interner Softwarefehler, generisch
6320 <sub>h</sub>	Nennstrom muss gesetzt werden (203B <sub>h</sub> :01 <sub>h</sub> /6075 <sub>h</sub> )
7113 <sub>h</sub>	Warnung: Ballast-Widerstand thermisch überlastet
7121 <sub>h</sub>	Motor blockiert
7200 <sub>h</sub>	Interner Fehler: Korrekturfaktor für Referenzspannung fehlt im OTP
7305 <sub>h</sub>	Sensor 1 (siehe 3204 <sub>h</sub> ) fehlerhaft
7306 <sub>h</sub>	Sensor 2 (siehe 3204 <sub>h</sub> ) fehlerhaft
7307 <sub>h</sub>	Sensor n (siehe 3204 <sub>h</sub> ), wo n größer 2
7600 <sub>h</sub>	Warnung: Nichtflüchtiger Speicher voll oder korrupt, Neustart der Steuerung für Aufräumarbeiten
8100 <sub>h</sub>	Fehler bei der Feldbusüberwachung
8130 <sub>h</sub>	Nur CANopen: "Life Guard"-Fehler oder "Heartbeat"-Fehler
8200 <sub>h</sub>	Nur CANopen: Slave brauchte zu lange um PDO Nachrichten zu Senden.
8210 <sub>h</sub>	Nur CANopen: PDO wurde nicht verarbeitet aufgrund eines Längen-Fehlers
8220 <sub>h</sub>	Nur CANopen: PDO Länge überschritten
8240 <sub>h</sub>	Nur CANopen: unerwartete Sync-Länge
8400 <sub>h</sub>	Fehler in der Drehzahlüberwachung: Schlupffehler zu groß
8611 <sub>h</sub>	Fehler in der Positionsüberwachung: Schleppfehler zu groß
8612 <sub>h</sub>	Fehler in der Positionsüberwachung: Endschalter überschritten

## 1008h Manufacturer Device Name

### Funktion

Enthält den Gerätenamen als Zeichenkette.

### Objektbeschreibung

Index	1008 <sub>h</sub>
Objektname	Manufacturer Device Name
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein

Zulässige Werte

Vorgabewert

- PD6-E891L95-E-65-5: PD6-E891L95-E-65-5
- PD6-E891L95-ES-65-5: PD6-E891L95-ES-65-5
- PD6-EB80CD-E-65-5: PD6-EB80CD-E-65-5
- PD6-EB80CD-ES-65-5: PD6-EB80CD-ES-65-5
- PD6-E891S95-E-65-5: PD6-E891S95-E-65-5
- PD6-E891S95-ES-65-5: PD6-E891S95-ES-65-5
- PD6-E891M95-E-65-5: PD6-E891M95-E-65-5
- PD6-E891M95-ES-65-5: PD6-E891M95-ES-65-5
- PD6-EB80SD-E-65-5: PD6-EB80SD-E-65-5
- PD6-EB80SD-ES-65-5: PD6-EB80SD-ES-65-5
- PD6-EB80MD-E-65-5: PD6-EB80MD-E-65-5
- PD6-EB80MD-ES-65-5: PD6-EB80MD-ES-65-5
- PD6-EB80LD-E-65-5: PD6-EB80LD-E-65-5
- PD6-EB80LD-ES-65-5: PD6-EB80LD-ES-65-5

Firmware Version

FIR-v1426

Änderungshistorie

## 1009h Manufacturer Hardware Version

### Funktion

Dieses Objekt enthält die Hardware-Version als Zeichenkette.

### Objektbeschreibung

Index	1009 <sub>h</sub>
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 <sub>h</sub>
Objektname	Manufacturer Software Version
Object Code	VARIABLE

Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FIR-v2213-B1029645
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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0D <sub>h</sub>

---

---

Subindex	01 <sub>h</sub>
Name	Save All Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---



---

Subindex	02 <sub>h</sub>
Name	Save Communication Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---



---

Subindex	03 <sub>h</sub>
Name	Save Application Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---



---

Subindex	04 <sub>h</sub>
Name	Save Customer Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---



---

Subindex	05 <sub>h</sub>
Name	Save Drive Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---



---

Subindex	06 <sub>h</sub>
Name	Save Tuning Parameters To Non-volatile Memory
Datentyp	UNSIGNED32

---



Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
<hr/>	
Subindex	07 <sub>h</sub>
Name	Save Miscellaneous Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
<hr/>	
Subindex	08 <sub>h</sub>
Name	Save Reserved1 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	09 <sub>h</sub>
Name	Save Reserved2 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	0A <sub>h</sub>
Name	Save CANopen Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
<hr/>	
Subindex	0B <sub>h</sub>
Name	Save Modbus RTU Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000001 <sub>h</sub>
Subindex	0C <sub>h</sub>
Name	Save Ethernet Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
Subindex	0D <sub>h</sub>
Name	Save Profibus Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## 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<sub>h</sub>" in den jeweiligen Subindex geschrieben werden. Das entspricht dezimal der 1702257011<sub>d</sub> 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 <sub>h</sub>
Objektname	Restore Default Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426

## Änderungshistorie

Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "Restore Default Parameter" auf "Restore Default Parameters".

Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 2 auf 4.

Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5.

Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Comm Default Parameters" auf "Restore Communication Default Parameters".

Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Application Default Parameters" auf "Restore Application Default Parameters".

Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7.

Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0D <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Restore All Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Restore Communication Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

Subindex	03 <sub>h</sub>
----------	-----------------

Name	Restore Application Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

Subindex	04 <sub>h</sub>
Name	Restore Customer Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

Subindex	05 <sub>h</sub>
Name	Restore Drive Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

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

---

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

---

Subindex	08 <sub>h</sub>
Name	Restore Reserved1 Configurations To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

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

## Beschreibung

Wird der Wert 64616F6C<sub>h</sub> (bzw. 1684107116<sub>d</sub> 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 <sub>h</sub>
Objektname	Identity Object
Object Code	RECORD
Datentyp	IDENTITY
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Vendor-ID
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000026C <sub>h</sub>

Subindex	02 <sub>h</sub>
----------	-----------------

Name	Product Code
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00000130<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00000139<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 00000142<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 0000014B<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00000154<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 0000015D<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00000166<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 0000016F<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00000178<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00000181<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 0000025C<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 00000265<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 0000026E<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 00000277<sub>h</sub></li> </ul>

---

Subindex	03 <sub>h</sub>
Name	Revision Number
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	08A50000 <sub>h</sub>

---

Subindex	04 <sub>h</sub>
Name	Serial Number
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Configuration Date
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Configuration Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

Subindex 01<sub>h</sub> (Konfigurationsdatum) soll die Anzahl der Tage seit dem 1. Januar 1984 enthalten.

Subindex 02<sub>h</sub> (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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Program Data Bootloader/firmware
Datentyp	DOMAIN
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0

Subindex	02 <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Program Control Bootloader/firmware
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Program Control NanoJ
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Program Status Bootloader/firmware
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Program Status NanoJ
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 2028h MODBUS Slave Address

### Funktion

Dieses Objekt enthält die Slave-Adresse für Modbus.

### Objektbeschreibung

Index	2028 <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
Objektname	MODBUS RTU Baudrate
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Modbus RTU
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00004B00 <sub>h</sub>
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 <sub>h</sub>
Objektname	MODBUS RTU Stop Bits
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>
Firmware Version	FIR-v1436
Änderungshistorie	<p>Firmware Version FIR-v1540: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "nein".</p> <p>Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".</p>

## Beschreibung

Die Anzahl der Stopbits ist abhängig von der Parity, welche im Objekt 202D<sub>h</sub> eingestellt werden kann.

### 202Dh MODBUS RTU Parity

## Funktion

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

## Objektbeschreibung

Index	202D <sub>h</sub>
Objektname	MODBUS RTU Parity
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Modbus RTU
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>
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 <sub>h</sub>
Objektname	Pole Pair Count
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00000032<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00000032<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 00000004<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 00000004<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00000032<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00000032<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00000032<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 00000032<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00000004<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00000004<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 00000004<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 00000004<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 00000004<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 00000004<sub>h</sub></li> </ul>
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 <sub>h</sub>
Objektname	Max Motor Current
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 0000251C<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 0000251C<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 000124F8<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 000124F8<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00002508<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00002508<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 000024FE<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 000024FE<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00004E20<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00004E20<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 00009C40<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 00009C40<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 0000FDE8<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 0000FDE8<sub>h</sub></li> </ul>
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 <sub>h</sub>
Objektname	Upper Voltage Warning Level
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	0000F5B3 <sub>h</sub>
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<sub>h</sub> minus 2 Volt) ist.

## 2035h Lower Voltage Warning Level

### Funktion

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

### Objektbeschreibung

---

Index	2035 <sub>h</sub>
Objektname	Lower Voltage Warning Level
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00002C52 <sub>h</sub>
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<sub>h</sub> 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 <sub>h</sub>
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 <sub>h</sub>
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<sub>h</sub> = "1") und sich der Motor im Stillstand befindet.

### Objektbeschreibung

---

Index	2037 <sub>h</sub>
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 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

---

### Beschreibung

#### Wert von 2037<sub>h</sub> größer/gleich 0 und kleiner als Wert 6075<sub>h</sub>

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

#### Wert von 2037<sub>h</sub> im Bereich von -1 bis -100

Der eingetragene Wert wird als eine Prozentzahl interpretiert und bestimmt die Reduktion des Nennstroms in 2037<sub>h</sub>. Für die Berechnung wird der Wert in 6075<sub>h</sub> herangezogen.

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

Die Angabe -100 in 2037<sub>h</sub> 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 <sub>h</sub>
Objektname	Brake Controller Timing
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Close Brake Idle Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Shutdown Power Idle Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>

Subindex	03 <sub>h</sub>
Name	Open Brake Delay Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>

Subindex	04 <sub>h</sub>
Name	Start Operation Delay Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	05 <sub>h</sub>
Name	PWM Frequency
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	zwischen 0 bzw. 50 (Bremsenausgang) und 2000 (7D0 <sub>h</sub> ) 20000 (4E20 <sub>h</sub> )
Vorgabewert	00000000 <sub>h</sub>

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

## Beschreibung

Die Subindizes haben folgende Funktionen:

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

## 2039h Motor Currents

### Funktion

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

### Objektbeschreibung

Index	2039 <sub>h</sub>
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<sub>h</sub>, "Actual Current" hinzugefügt. Phasenströme Ia und Ib in I<sub>α</sub> und I<sub>β</sub> geändert (Clarke-Transformation).</p>

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	05 <sub>h</sub>
Subindex	01 <sub>h</sub>
Name	Id
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	Iq
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	03 <sub>h</sub>
Name	I <sub>α</sub>
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	04 <sub>h</sub>
Name	I <sub>β</sub>
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	05 <sub>h</sub>
Name	Actual Current
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

- 01<sub>h</sub>: feldbildende Komponente des Stroms
- 02<sub>h</sub>: momentbildende Komponente des Stroms
- 03<sub>h</sub>: I<sub>α</sub>
- 04<sub>h</sub>: I<sub>β</sub>
- 05<sub>h</sub>: Gesamtstrom durch  $\sqrt{2}$ , also runtergerechnet auf eine Motorphase. Im *Closed Loop* wird zusätzlich das Vorzeichen von I<sub>q</sub> verwendet. Der Stromwert kann dann zum Vergleich mit dem Strom aus 6075<sub>h</sub>, 2031<sub>h</sub> und 203B<sub>h</sub>:05<sub>h</sub> 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<sub>d</sub> (Subindex 01<sub>h</sub>) und I<sub>q</sub> (Subindex 02<sub>h</sub>) 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Minimum Current For Block Detection
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00000947<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00000947<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 0000186A<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 0000186A<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00000942<sub>h</sub></li> </ul>

- PD6-E891S95-ES-65-5: 00000942<sub>h</sub>
- PD6-E891M95-E-65-5: 0000093F<sub>h</sub>
- PD6-E891M95-ES-65-5: 0000093F<sub>h</sub>
- PD6-EB80SD-E-65-5: 000006BD<sub>h</sub>
- PD6-EB80SD-ES-65-5: 000006BD<sub>h</sub>
- PD6-EB80MD-E-65-5: 00000DAC<sub>h</sub>
- PD6-EB80MD-ES-65-5: 00000DAC<sub>h</sub>
- PD6-EB80LD-E-65-5: 0000124F<sub>h</sub>
- PD6-EB80LD-ES-65-5: 0000124F<sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	Block Detection Time
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000000C8 <sub>h</sub>

---

## Beschreibung

Die Subindizes haben folgende Funktion:

- 01<sub>h</sub>: 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<sub>h</sub>. Beispiel: der Wert "1000" entspricht 1000 mA (=1 A), der Wert "-70" entspricht 70% von 2031<sub>h</sub>.
- 02<sub>h</sub>: Gibt die Zeit in ms an, die der Motor nach der Blockdetektion trotzdem noch gegen den Block fahren soll.

## 203Bh I2t Parameters

### Funktion

Dieses Objekt hält die Parameter für die I<sup>2</sup>t-Überwachung.

Die I<sup>2</sup>t-Überwachung wird aktiviert, in dem in 203B<sub>h</sub>:01 und 203B<sub>h</sub>:02 ein Wert größer 0 eingetragen wird und in 6073<sub>h</sub> ein Wert größer 1000 (siehe I2t Motor-Überlastungsschutz).

I<sup>2</sup>t kann nur für den *Closed Loop*-Betrieb verwendet werden, mit einer Ausnahme: Wenn I<sup>2</sup>t im *Open Loop*-Betrieb aktiviert ist, wird der Strom auf den kleineren der Werte von 203B<sub>h</sub>:01<sub>h</sub>, 6073<sub>h</sub> und 2031<sub>h</sub> begrenzt.

### Objektbeschreibung

---

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

Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Nominal Current" auf "Motor Rated Current".

Firmware Version FIR-v1825-B577172: Eintrag "Name" geändert von "Motor Rated Current" auf "Nominal Current".

Firmware Version FIR-v1825-B577172: Eintrag "Name" geändert von "Nominal Current" auf "Motor Rated Current".

Firmware Version FIR-v1825-B577172: Eintrag "Name" geändert von "Motor Rated Current" auf "Nominal Current".

Firmware Version FIR-v1825-B577172: Eintrag "Name" geändert von "Nominal Current" auf "Motor Rated Current".

Firmware Version FIR-v1825-B577172: Die Anzahl der Einträge haben sich geändert von 8 auf 7.

Firmware Version FIR-v1926-B648637: Eintrag "Name" geändert von "Maximum Duration Of Peak Current" auf "Maximum Duration Of Max Current".

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Motor Rated Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 0000251C<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 0000251C<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 000061A8<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 000061A8<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00002508<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00002508<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 000024FE<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 000024FE<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00001AF4<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00001AF4<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 000036B0<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 000036B0<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 0000493E<sub>h</sub></li> </ul>



---

 ■ PD6-EB80LD-ES-65-5: 0000493E<sub>h</sub>


---

Subindex	02 <sub>h</sub>
Name	Maximum Duration Of Max Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00000000<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00000000<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 00001388<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 00001388<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00000000<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00000000<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00000000<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 00000000<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00001388<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00001388<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 00001388<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 00001388<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 00001388<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 00001388<sub>h</sub></li> </ul>

---

Subindex	03 <sub>h</sub>
Name	Threshold
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	04 <sub>h</sub>
Name	CalcValue
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	05 <sub>h</sub>
Name	LimitedCurrent
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 <sub>h</sub>
Subindex	06 <sub>h</sub>
Name	Status
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

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

- 01<sub>h</sub>: Hier wird der im Motordatenblatt angegebene Nennstrom in mA eingetragen. Dieser muss kleiner als der eingegebene Strom in 2031<sub>h</sub> und 6073<sub>h</sub> sein, sonst wird die Überwachung nicht aktiviert. Der angegebene Wert wird als Effektivwert interpretiert.
- 02<sub>h</sub>: Gibt die maximale Dauer des Maximalstroms (6073<sub>h</sub>) in ms an.
- 03<sub>h</sub>: Threshold, gibt die Grenze in A<sup>2</sup>ms an, von der abhängt, ob auf Maximalstrom oder Nennstrom geschaltet wird.
- 04<sub>h</sub>: CalcValue, gibt den berechneten Wert in A<sup>2</sup>ms an, welcher mit Threshold verglichen wird, um den Strom einzustellen.
- 05<sub>h</sub>: LimitedCurrent, zeigt den gegenwärtigen Strom als Effektivwert an, der von I<sup>2</sup>t eingestellt wurde.
- 06<sub>h</sub>: aktueller Status. Ist der Subentry-Wert "0", ist I<sup>2</sup>t deaktiviert, ist der Wert "1", wird I<sup>2</sup>t 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"<sub>h</sub> gesetzt, wird die Überwachung abgeschaltet, das Bit "Target reached" im Objekt 6041<sub>h</sub> (Statusword) wird nie gesetzt.

### Objektbeschreibung

Index	203D <sub>h</sub>
Objektname	Torque Window
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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 Ist Drehmoment muss sich für diese Zeit (in Millisekunden) innerhalb des "Torque Window" (203D<sub>h</sub>) befinden, damit das Zieldrehmoment als erreicht gilt.

### Objektbeschreibung

Index	203E <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

### Beschreibung

Weicht die Istgeschwindigkeit von der Sollgeschwindigkeit so stark ab, dass der Wert (Absolutbetrag) des Objekts 60F8<sub>h</sub> (Max Slippage) überschritten wird, wird das Bit 13 im Objekt 6041<sub>h</sub> gesetzt. Die Abweichung muss länger andauern als die Zeit im Objekt 203E<sub>h</sub>.

Im Objekt 3700<sub>h</sub> kann eine Reaktion auf den Schlupffehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt 1003<sub>h</sub> 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 <sub>h</sub>
Objektname	Clock Direction Multiplier
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000080 <sub>h</sub>
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 <sub>h</sub>
Objektname	Clock Direction Divider
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## 205Ah Absolute Sensor Boot Value (in User Units)

### Funktion

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

## Objektbeschreibung

Index	205A <sub>h</sub>
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 <sub>h</sub>
Firmware Version	FIR-v1446
Änderungshistorie	<p>Firmware Version FIR-v1512: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Encoder Boot Value" auf "Absolute Sensor Boot Value (in User Units)".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".</p>

## 205Bh Clock Direction Or Clockwise/Counter Clockwise Mode

### Funktion

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

### Objektbeschreibung

Index	205B <sub>h</sub>
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 <sub>h</sub>
Firmware Version	FIR-v1504
Änderungshistorie	

## 2084h Bootup Delay

### Funktion

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

### Objektbeschreibung

---

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

---

### 2101h Fieldbus Module Availability

#### Funktion

Zeigt die verfügbaren Feldbusse an.

#### Objektbeschreibung

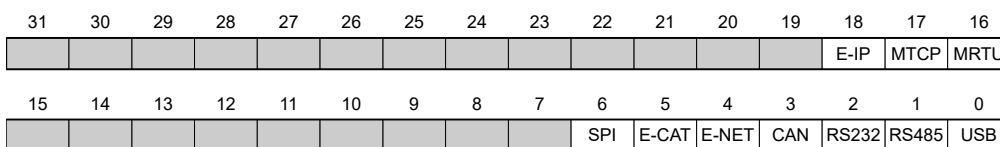
---

Index	2101 <sub>h</sub>
Objektname	Fieldbus Module Availability
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00190003 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Object Name" geändert von "Fieldbus Module" auf "Fieldbus Module Availability".

---

#### Beschreibung

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



#### USB

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

**RS-485**

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

**RS-232**

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

**CAN**

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

**E-NET**

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

**E-CAT**

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

**SPI**

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

**MRTU**

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

**MTCP**

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

**E-IP**

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

## 2102h Fieldbus Module Control

### Funktion

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

### Objektbeschreibung

Index	2102 <sub>h</sub>
Objektname	Fieldbus Module Control
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Kommunikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00190003 <sub>h</sub>
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Kommunikation".

## Beschreibung

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

Dabei gilt die folgende Verteilung der Bits:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													E-IP	MTCP	MRTU
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SPI	E-CAT	E-NET	CAN	RS232	RS485	USB

### USB

USB Schnittstelle

### RS-485

RS-485 Schnittstelle

### RS-232

RS-232 Schnittstelle

### CAN

CANopen Schnittstelle

### E-NET

EtherNET Schnittstelle

### E-CAT

EtherCAT Schnittstelle

### SPI

SPI Schnittstelle

### MRTU

Modbus RTU Protokoll

### MTCP

Modbus TCP Protokoll

### E-IP

EtherNet/IP™ Protokoll

## 2103h Fieldbus Module Status

### Funktion

Zeigt die aktiven Feldbusse an.

### Objektbeschreibung

Index	$2103_h$
Objektname	Fieldbus Module Status
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein



Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	

---

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	Fieldbus Module Disable Mask
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000002 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	Fieldbus Module Enabled
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00190003 <sub>h</sub>

---

## Beschreibung

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

Subindex 2 (Fieldbus Module Enabled): Dieser Subindex zeigt alle zur Zeit aktivierten physikalischen Schnittstellen und Protokolle an. Der Wert "1" bedeutet, dass der Feldbus aktiv ist.

Für Subindex 1 und 2 gilt folgende Verteilung der Bits:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													E-IP	MTCP	MRTU
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SPI	E-CAT	E-NET	CAN	RS232	RS485	USB

**USB**

USB Schnittstelle

**RS-485**

RS-485 Schnittstelle

**RS-232**

RS-232 Schnittstelle

**CAN**

CANopen Schnittstelle

**E-NET**

EtherNET Schnittstelle

**E-CAT**

EtherCAT Schnittstelle

**SPI**

SPI Schnittstelle

**MRTU**

Modbus RTU Protokoll

**MTCP**

Modbus TCP Protokoll

**E-IP**

EtherNet/IP™ Protokoll

**2290h PDI Control****Funktion**

Mit diesem Objekt können Sie das *Plug&Drive-Interface* aktivieren. Weitergehende Informationen finden Sie im Dokument *Funktionsbeschreibung Plug&Drive-Interface*.

**Objektbeschreibung**

Index	2290 <sub>h</sub>
Objektname	PDI Control
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	PDI Set Value 1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	

Vorgabewert	00000000 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	PDI Set Value 2
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
Subindex	03 <sub>h</sub>
Name	PDI Set Value 3
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>
Subindex	04 <sub>h</sub>
Name	PDI Command
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	PDI Status
Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	PDI Return Value
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

**2300h NanoJ Control**
**Funktion**

Steuert die Ausführung eines *NanoJ-Programms*.

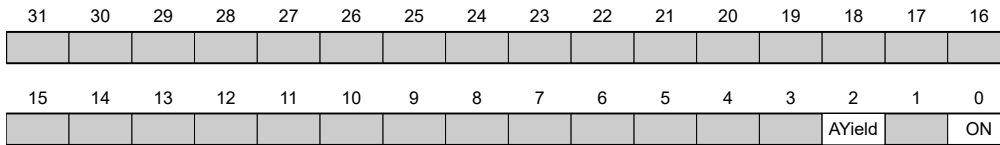
**Objektbeschreibung**

Index	2300 <sub>h</sub>
Objektname	NanoJ Control
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

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

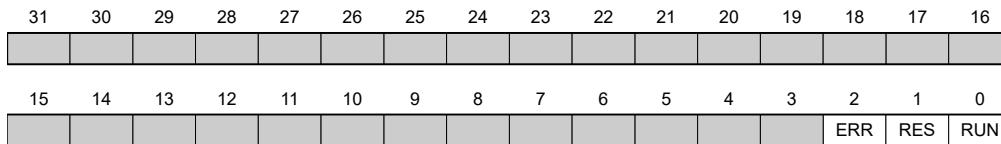
---

Index	2301 <sub>h</sub>
Objektname	NanoJ Status
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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<sub>h</sub> ausgelesen werden.

### 2302h NanoJ Error Code

#### Funktion

Zeigt an, welcher Fehler bei der Ausführung des Benutzerprogramms aufgetreten ist.

#### Objektbeschreibung

---

Index	2302 <sub>h</sub>
Objektname	NanoJ Error Code
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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 <sub>h</sub>	Firmware unterstützt verwendete Funktion nicht (z. B. <code>sin</code> , <code>cosin</code> etc.)
0005 <sub>h</sub>	Time Out: Code wird zu lange ohne <code>yield()</code> oder <code>sleep()</code> ausgeführt
0007 <sub>h</sub>	Zu viele Variablen auf dem Stack
0100 <sub>h</sub>	Ungültige NanoJ Programmdatei
0101 <sub>h</sub>	Ungültige NanoJ-Version der Programmdatei
0102 <sub>h</sub>	CRC-Fehler in der NanoJ-Programmdatei

Fehler bei dem Zugriff auf ein Objekt:

Nummer	Beschreibung
1xxxxyy <sub>h</sub>	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 <sub>h</sub>	Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs <code>input</code> deklariert (siehe <a href="#">2310h NanoJ Input Data Selection</a> )
3000000 <sub>h</sub>	Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs <code>output</code> deklariert (siehe <a href="#">2320h NanoJ Output Data Selection</a> )
4000000 <sub>h</sub>	Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs <code>inout</code> deklariert (siehe <a href="#">2330h NanoJ In/output Data Selection</a> )
1000 <sub>h</sub>	Zugriff auf ein nicht existierendes Objekt im Objektverzeichnis
1001 <sub>h</sub>	Schreibzugriff auf schreibgeschützten Eintrag im OD
1002 <sub>h</sub>	Es wurde versucht, einen zu niedrigen oder zu hohen Wert in ein Objekt zu schreiben.
1003 <sub>h</sub>	Es wurde versucht, ein Objekt auszulesen, das nur Schreibzugriff erlaubt.
1FFF <sub>h</sub>	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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	1ms Timer
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

## 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 <sub>h</sub>
Objektname	Uptime Seconds
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 <sub>h</sub>

---

Subindex	01 <sub>h</sub> - 10 <sub>h</sub>
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein

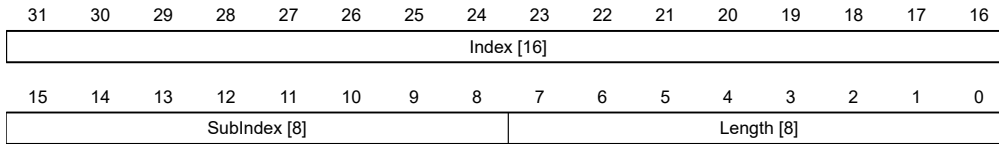
Zulässige Werte  
 Vorgabewert 00000000<sub>h</sub>

---

### 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 <sub>h</sub>
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

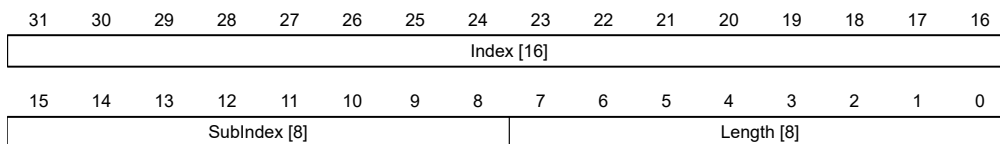
Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 <sub>h</sub>

Subindex	01 <sub>h</sub> - 10 <sub>h</sub>
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 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 <sub>h</sub>
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

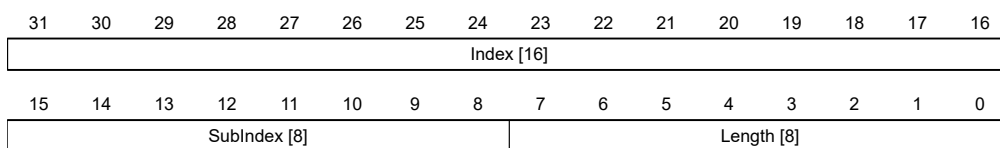
Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 <sub>h</sub>

Subindex	01 <sub>h</sub> - 10 <sub>h</sub>
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

### 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20 <sub>h</sub>
Subindex	01 <sub>h</sub> - 20 <sub>h</sub>
Name	NanoJ Input #1 - #32
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

Hier können dem *NanoJ-Programm* z. B. Vorgabewerte übergeben werden.

## 2410h NanoJ Init Parameters

### Funktion

Dieses Objekt funktioniert identisch dem Objekt 2400<sub>h</sub> mit dem Unterschied, dass dieses Objekt gespeichert werden kann.

### Objektbeschreibung

Index	2410 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20 <sub>h</sub>

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

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20 <sub>h</sub>

Subindex	01 <sub>h</sub> - 20 <sub>h</sub>
Name	NanoJ Output #1 - #32
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

### 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

Subindex	01 <sub>h</sub> - 40 <sub>h</sub>
Name	Value #1 - #64
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FE <sub>h</sub>

Subindex	01 <sub>h</sub> - FE <sub>h</sub>
Name	Storage #1 - #254
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Reboot Command
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Reboot Delay Time In Ms
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

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

## Beschreibung

Die Subindizes haben folgende Funktion:

- 01<sub>h</sub>: Wird hier der Wert "746F6F62<sub>h</sub>" eingetragen, wird die Firmware rebootet.
- 02<sub>h</sub>: Zeit in Millisekunden: verzögert den Reboot der Firmware um die jeweilige Zeit.
- 03<sub>h</sub>: 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 <sub>h</sub>
Objektname	Motor Drive Submode Select
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00000001<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00000001<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 00000041<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 00000041<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00000001<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00000001<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00000001<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 00000001<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00000041<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00000041<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 00000041<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 00000041<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 00000041<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 00000041<sub>h</sub></li> </ul>
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

### 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<sub>h</sub> wird also ignoriert, 3210<sub>h</sub>:3 und 3210<sub>h</sub>: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 <sub>h</sub>
Objektnamen	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 <sub>h</sub>
----------	-----------------

Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

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

Subindex	02 <sub>h</sub>
Name	2nd Feedback Interface
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

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

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>:  
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<sub>h</sub> 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.

#### HINWEIS



Wird das Bit 0 in 3202<sub>h</sub> 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Index Of 1st Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	3380 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Index Of 2nd Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	33A0 <sub>h</sub>

Subindex	03 <sub>h</sub>
Name	Index Of 3rd Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	33A1 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>:  
 Subindex n verweist auf den Index des zugehörigen Objekts für die Konfiguration der entsprechenden Rückführung.  
 Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel [Konfigurieren der Sensoren](#).

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

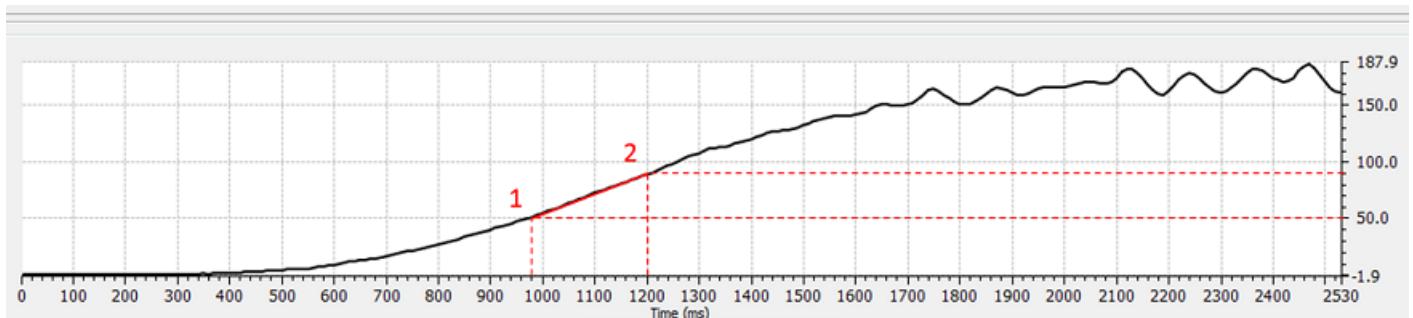
Subindex	01 <sub>h</sub>
Name	Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Acceleration
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 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<sub>h</sub>:01<sub>h</sub> ein.
3. Zeichnen Sie (z. B. im *Plug & Drive Studio*) die aktuelle Geschwindigkeit (Objekt 606C<sub>h</sub>) auf. Berechnen Sie die Beschleunigung in den eingestellten benutzerdefinierten Einheiten für den Drehzahlbereich, wo diese konstant ist. Tragen Sie den Wert in 320D<sub>h</sub>:02<sub>h</sub> ein.  
 Am Beispiel der Drehzahl-Kurve in der folgenden Abbildung:  
 $(90-50)/(1200-980)=182$  U/min pro Sekunde.



## 3210h Motor Drive Parameter Set

### Funktion

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

### Objektbeschreibung

Index	3210 <sub>h</sub>
Objektname	Motor Drive Parameter Set
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "S_P" auf "Position Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "S_I" auf "Position Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "V_P" auf "Velocity Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "V_I" auf "Velocity Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Id_P" auf "Flux Current Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Id_I" auf "Flux Current Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Iq_P" auf "Torque Current Loop, Proportional Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "Iq_I" auf "Torque Current Loop, Integral Gain (closed Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "I_P" auf "Torque Current Loop, Proportional Gain (dspDrive - Stepper Motor, Open Loop)".</p> <p>Firmware Version FIR-v1626: Eintrag "Name" geändert von "I_I" auf "Torque Current Loop, Integral Gain (dspDrive - Stepper Motor, Open Loop)".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Torque Current Loop, Proportional Gain (dspDrive - Stepper Motor, Open Loop)" auf "Torque Current Loop, Proportional Gain (open Loop)".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Torque Current Loop, Integral Gain (dspDrive - Stepper Motor, Open Loop)" auf "Torque Current Loop, Integral Gain (open Loop)".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Datentyp" geändert von "INTEGER32" auf "UNSIGNED32".</p>

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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0C <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Position Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000800 <sub>h</sub>

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

Subindex	03 <sub>h</sub>
Name	Velocity Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00002EE0 <sub>h</sub>

Subindex	04 <sub>h</sub>
Name	Velocity Loop, Integral Gain (closed Loop)

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000001E <sub>h</sub>
<hr/>	
Subindex	05 <sub>h</sub>
Name	Flux Current Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00881EE0 <sub>h</sub>
<hr/>	
Subindex	06 <sub>h</sub>
Name	Flux Current Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0007C740 <sub>h</sub>
<hr/>	
Subindex	07 <sub>h</sub>
Name	Torque Current Loop, Proportional Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00881EE0 <sub>h</sub>
<hr/>	
Subindex	08 <sub>h</sub>
Name	Torque Current Loop, Integral Gain (closed Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0007C740 <sub>h</sub>
<hr/>	
Subindex	09 <sub>h</sub>
Name	Torque Current Loop, Proportional Gain (open Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	004DC880 <sub>h</sub>
<hr/>	
Subindex	0A <sub>h</sub>
Name	Torque Current Loop, Integral Gain (open Loop)
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	001D2B30 <sub>h</sub>
<hr/>	
Subindex	0B <sub>h</sub>
Name	Velocity Feed Forward Factor In Per Mille
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>
<hr/>	
Subindex	0C <sub>h</sub>
Name	Acceleration Feed Forward Factor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

- Subindex 00<sub>h</sub>: Anzahl der Einträge
- Subindex 01<sub>h</sub>: Proportional-Anteil des S-Reglers (Position)
- Subindex 02<sub>h</sub>: Integral-Anteil des S-Reglers (Position)
- Subindex 03<sub>h</sub>: Proportional-Anteil des V-Reglers (Geschwindigkeit)
- Subindex 04<sub>h</sub>: Integral-Anteil des V-Reglers (Geschwindigkeit)
- Subindex 05<sub>h</sub>: (Closed Loop) Proportional-Anteil des Stromreglers der feldbildenden Komponente
- Subindex 06<sub>h</sub>: (Closed Loop) Integral-Anteil des Stromreglers der feldbildenden Komponente
- Subindex 07<sub>h</sub>: (Closed Loop) Proportional-Anteil des Stromreglers der momentbildenden Komponente
- Subindex 08<sub>h</sub>: (Closed Loop) Integral-Anteil des Stromreglers der momentbildenden Komponente
- Subindex 09<sub>h</sub>: (Open Loop) Proportional-Anteil des Stromreglers der feldbildenden Komponente
- Subindex 0A<sub>h</sub>: (Open Loop) Integral-Anteil des Stromreglers der feldbildenden Komponente
- Subindex 0B<sub>h</sub>: (Closed Loop) Geschwindigkeitsvorsteuerung in Promille. Default ist 1000 und damit ein Faktor von 1.
- Subindex 0C<sub>h</sub>: (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<sub>h</sub> werden erst nach einem Neustart der Steuerung aktiv. Das Auto-Setup muss danach erneut durchgeführt werden.

### Objektbeschreibung

Index	3212 <sub>h</sub>
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	<p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 2 auf 3.</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Name" geändert von "Enable Legacy Power Mode" auf "Reserved".</p>

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>
Subindex	01 <sub>h</sub>
Name	Reserved
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Override Field Inversion
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>
<hr/>	
Subindex	03 <sub>h</sub>
Name	Auto-setup With Current Controller Parameters From The OD
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>
<hr/>	
Subindex	04 <sub>h</sub>
Name	Use 321Ah, 321Bh, 321Ch, 321Dh, Instead Of 3210h
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

## Beschreibung

Für den Subindex 02<sub>h</sub> 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<sub>h</sub> gültige Werte:

- Wert = "0": Auto-Setup erkennt den Motortyp (Schrittmotor oder BLDC-Motor) und verwendet den entsprechenden vorkonfigurierten Parametersatz.
- Wert = "1": Auto-Setup mit den Werten für den Regler durchführen, die vor dem Auto-Setup im Objekt 3210<sub>h</sub> bzw. 320E<sub>h</sub> eingetragen wurden, die Werte in 3210<sub>h</sub> bzw. 320E<sub>h</sub> werden nicht geändert.

## 321Ah Current Controller Parameters

### Funktion

Enthält die Parameter für den Stromregler (Kommutierung). In der Regel sollen die Werte für I<sub>q</sub> (Subindex 01<sub>h</sub>/02<sub>h</sub>) und I<sub>d</sub> (Subindex 03<sub>h</sub>/04<sub>h</sub>) gleich sein. Siehe Kapitel Reglerstruktur.

### Objektbeschreibung

Index	321A <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>

---



---

Subindex	01 <sub>h</sub>
Name	Proportional Gain Kp For Iq [mV/A]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00002A30<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00002A30<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 00000258<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 00000258<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00000820<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00000820<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 000020D0<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 000020D0<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00000A28<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00000A28<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 000004B0<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 000004B0<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 00000320<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 00000320<sub>h</sub></li> </ul>

---



---

Subindex	02 <sub>h</sub>
Name	Integrator Time Ti For Iq [µs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein



## Zulässige Werte

## Vorgabewert

- PD6-E891L95-E-65-5: 00002891<sub>h</sub>
- PD6-E891L95-ES-65-5: 00002891<sub>h</sub>
- PD6-EB80CD-E-65-5: 00001207<sub>h</sub>
- PD6-EB80CD-ES-65-5: 00001207<sub>h</sub>
- PD6-E891S95-E-65-5: 00001450<sub>h</sub>
- PD6-E891S95-ES-65-5: 00001450<sub>h</sub>
- PD6-E891M95-E-65-5: 000023AA<sub>h</sub>
- PD6-E891M95-ES-65-5: 000023AA<sub>h</sub>
- PD6-EB80SD-E-65-5: 00001223<sub>h</sub>
- PD6-EB80SD-ES-65-5: 00001223<sub>h</sub>
- PD6-EB80MD-E-65-5: 00001770<sub>h</sub>
- PD6-EB80MD-ES-65-5: 00001770<sub>h</sub>
- PD6-EB80LD-E-65-5: 00001262<sub>h</sub>
- PD6-EB80LD-ES-65-5: 00001262<sub>h</sub>

## Subindex

03<sub>h</sub>

## Name

Proportional Gain Kp For Id [mV/A]

## Datentyp

UNSIGNED32

## Zugriff

lesen/schreiben

## PDO-Mapping

nein

## Zulässige Werte

## Vorgabewert

- PD6-E891L95-E-65-5: 00002A30<sub>h</sub>
- PD6-E891L95-ES-65-5: 00002A30<sub>h</sub>
- PD6-EB80CD-E-65-5: 00000258<sub>h</sub>
- PD6-EB80CD-ES-65-5: 00000258<sub>h</sub>
- PD6-E891S95-E-65-5: 00000820<sub>h</sub>
- PD6-E891S95-ES-65-5: 00000820<sub>h</sub>
- PD6-E891M95-E-65-5: 000020D0<sub>h</sub>
- PD6-E891M95-ES-65-5: 000020D0<sub>h</sub>
- PD6-EB80SD-E-65-5: 00000A28<sub>h</sub>
- PD6-EB80SD-ES-65-5: 00000A28<sub>h</sub>
- PD6-EB80MD-E-65-5: 000004B0<sub>h</sub>
- PD6-EB80MD-ES-65-5: 000004B0<sub>h</sub>
- PD6-EB80LD-E-65-5: 00000320<sub>h</sub>
- PD6-EB80LD-ES-65-5: 00000320<sub>h</sub>

## Subindex

04<sub>h</sub>

## Name

Integrator Time Ti For Id [μs]

## Datentyp

UNSIGNED32

## Zugriff

lesen/schreiben

## PDO-Mapping

nein

## Zulässige Werte

## Vorgabewert

- PD6-E891L95-E-65-5: 00002891<sub>h</sub>
- PD6-E891L95-ES-65-5: 00002891<sub>h</sub>
- PD6-EB80CD-E-65-5: 00001207<sub>h</sub>
- PD6-EB80CD-ES-65-5: 00001207<sub>h</sub>

- PD6-E891S95-E-65-5: 00001450<sub>h</sub>
- PD6-E891S95-ES-65-5: 00001450<sub>h</sub>
- PD6-E891M95-E-65-5: 000023AA<sub>h</sub>
- PD6-E891M95-ES-65-5: 000023AA<sub>h</sub>
- PD6-EB80SD-E-65-5: 00001223<sub>h</sub>
- PD6-EB80SD-ES-65-5: 00001223<sub>h</sub>
- PD6-EB80MD-E-65-5: 00001770<sub>h</sub>
- PD6-EB80MD-ES-65-5: 00001770<sub>h</sub>
- PD6-EB80LD-E-65-5: 00001262<sub>h</sub>
- PD6-EB80LD-ES-65-5: 00001262<sub>h</sub>

## 321Bh Velocity Controller Parameters

### Funktion

Enthält die Parameter für den Geschwindigkeitsregler. Siehe Kapitel [Reglerstruktur](#).

### Objektbeschreibung

Index	321B <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Proportional Gain Kp [mA/Hz]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00000140<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00000140<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 00000280<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 00000280<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00000140<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00000140<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00000140<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 00000140<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00000280<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00000280<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 00000280<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 00000280<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 00000280<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 00000280<sub>h</sub></li> </ul>
-------------	---

---

Subindex	02 <sub>h</sub>
Name	Integrator Time Ti [μs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00007A12<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00007A12<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 0000F424<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 0000F424<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00007A12<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00007A12<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00007A12<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 00007A12<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 0000F424<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 0000F424<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 0000F424<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 0000F424<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 0000F424<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 0000F424<sub>h</sub></li> </ul>

---

## 321Ch Position Controller Parameters

### Funktion

Enthält die Parameter für den Positionsregler. Siehe Kapitel [Reglerstruktur](#).

### Objektbeschreibung

---

Index	321C <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

---



---

Subindex	01 <sub>h</sub>
Name	Proportional Gain Kp [Hz]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00000166<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00000166<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 0000001D<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 0000001D<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00000166<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00000166<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00000166<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 00000166<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 0000001D<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 0000001D<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 0000001D<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 0000001D<sub>h</sub></li> <li>■ PD6-EB80LD-E-65-5: 0000001D<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 0000001D<sub>h</sub></li> </ul>

---



---

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

---

## 321Dh Pre-control

### Funktion

Enthält die Parameter für die Vorsteuerung. Siehe Kapitel [Reglerstruktur](#).

### Objektbeschreibung

Index	321D <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Voltage Pre-control [%]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Acceleration Pre-control [%]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>

Subindex	03 <sub>h</sub>
Name	Velocity Pre-control [% <sub>oo</sub> ]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>

## 321Eh Voltage Limit

### Funktion

Maximal zulässige PWM-Spannung (Tastverhältnis). Werte  $\leq 1000$  werden als Promille-Werte (der verfügbaren Spannung) interpretiert. Werte  $> 1000$  als Millivolt. Siehe auch Kapitel [Reglerstruktur](#).

### Objektbeschreibung

Index	321E <sub>h</sub>
Objektname	Voltage Limit
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000186A0 <sub>h</sub>
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 <sub>o_low</sub>	Keine; der Spannungsvektor beschreibt einen Kreis.
U <sub>o_low</sub> ...U <sub>o_high</sub>	Der Spannungsvektor beschreibt einen Kreis, der an vier/sechs Seiten immer mehr abgeflacht wird, proportional zum eingestellten Wert.
$\geq$ U <sub>o_high</sub>	Volle; Der Spannungsvektor beschreibt einen Quadrat bzw. ein Sechseck.

#### U<sub>o\_low</sub>

Die niedrigste Spannung, ab welcher eine Übermodulation stattfindet. Wird wie folgt berechnet:

Betriebsspannung\*0,9425

**U<sub>o\_high</sub>**

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<sub>h</sub> kann der jeweilige Analogeingang als Strom- oder Spannungseingang konfiguriert werden.

**Objektbeschreibung**

Index	3220 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Analog Input Digits
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Analog Input #1 Digit
Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

## Beschreibung

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

- Spannungseingang:  $x \text{ Digits} * 10 \text{ V} / 1023 \text{ Digits}$
- Stromeingang (falls konfigurierbar):  $x \text{ Digits} * 20 \text{ mA} / 1023 \text{ Digits}$

## 3221h Analog Inputs Control

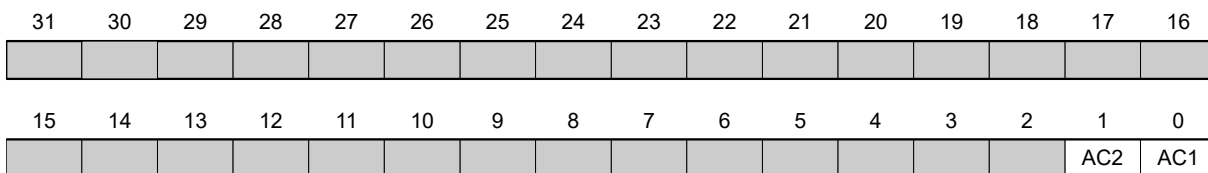
### Funktion

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

### Objektbeschreibung

Index	3221 <sub>h</sub>
Objektname	Analog Inputs Control
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Analog Input Switch #1
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

## Beschreibung

### 3240h Digital Inputs Control

#### Funktion

Mit diesem Objekt lassen sich digitale Eingänge manipulieren wie in Kapitel [Digitale Ein- und Ausgänge](#) beschrieben .

#### Objektbeschreibung

Index	3240 <sub>h</sub>
Objektname	Digital Inputs Control

Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1426: Subindex 01<sub>h</sub>: Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable"</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 8 auf 9.</p>

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	08 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Special Function Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	03 <sub>h</sub>
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	04 <sub>h</sub>
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	05 <sub>h</sub>
Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

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

---



---

Subindex	07 <sub>h</sub>
Name	Differential Select
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	08 <sub>h</sub>
Name	Routing Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

## Beschreibung

Die Subindizes haben folgende Funktion:

- **3240<sub>h</sub>:01<sub>h</sub>** (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<sub>h</sub>:01<sub>h</sub>** auf "1" gesetzt werden.
- **3240<sub>h</sub>:02<sub>h</sub>** (Function Inverted): Dieser Subindex wechselt von Schließer-Logik (ein logischer High-Pegel am Eingang ergibt den Wert "1" im Objekt **60FD<sub>h</sub>**) 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<sub>h</sub>:03<sub>h</sub>** (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<sub>h</sub>:04<sub>h</sub>** eingestellten Werte für den jeweiligen Eingang verwendet. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw..
- **3240<sub>h</sub>:04<sub>h</sub>** (Force Value): Dieses Bit gibt den Wert vor, der als Eingangswert eingelesen werden soll, wenn das gleiche Bit im Objekt **3240<sub>h</sub>:03<sub>h</sub>** gesetzt wurde.
- **3240<sub>h</sub>:05<sub>h</sub>** (Raw Value): Dieses Objekt beinhaltet den unmodifizierten Eingabewert.
- **3240<sub>h</sub>:07<sub>h</sub>** (Differential Select): Dieser Subindex schaltet bei den Eingängen zwischen "single-ended Eingang" (Wert "0" in dem Subindex) zu "Differentieller Eingang" (Wert "1" in dem Subindex) für alle Eingänge auf einmal um.
- **3240<sub>h</sub>:08<sub>h</sub>** (Routing Enable): Der Wert "1" in diesem Subindex aktiviert das Input Routing.

## 3242h Digital Input Routing

### Funktion

Dieses Objekt bestimmt die Quelle des Inputroutings, die im **60FD<sub>h</sub>** endet.

### Objektbeschreibung

Index	3242 <sub>h</sub>
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 <sub>h</sub>
----------	-----------------

Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	20 <sub>h</sub>

Subindex	01 <sub>h</sub> - 20 <sub>h</sub>
Name	Input Source For Bit #1 - #32
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8

Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	Control
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	Capture Count
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	03 <sub>h</sub>
Name	Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

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

## Beschreibung

- Subindex 01<sub>h</sub>: 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<sub>h</sub>: gibt die Anzahl der notierten Pegelwechsel seit dem letzten Start der Funktion wieder; wird auf 0 zurückgesetzt, wenn Subindex 01<sub>h</sub> auf 1,2 oder 3 gesetzt wird
- Subindex 03<sub>h</sub>: Encoder Position des Pegelwechsels (in absoluten Benutzereinheiten aus 6064<sub>h</sub>)
- Subindex 04<sub>h</sub>: 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 <sub>h</sub>
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<sub>h</sub>: 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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	09 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	No Function

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	03 <sub>h</sub>
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	04 <sub>h</sub>
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	05 <sub>h</sub>
Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	06 <sub>h</sub>
Name	Reserved1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO



Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	07 <sub>h</sub>
Name	Reserved2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	08 <sub>h</sub>
Name	Routing Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	09 <sub>h</sub>
Name	Enable Mask [Bit0=StatusLed, Bit1=ErrorLed]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	FFFFFFFF <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 01<sub>h</sub>: Ohne Funktion.
- 02<sub>h</sub>: Mit diesem Subindex wird die Logik invertiert (von Öffner-Logik auf Schließer-Logik).
- 03<sub>h</sub>: Mit diesem Subindex wird der Ausgangswert erzwungen, wenn das Bit den Wert "1" hat. Der Pegel des Ausgangs wird in Subindex 4<sub>h</sub> festgelegt.
- 04<sub>h</sub>: 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<sub>h</sub>: In diesen Subindex wird die an die Ausgänge gelegte Bitkombination abgelegt.
- 08<sub>h</sub>: Wird der Subindex auf "1" gesetzt, wird das *Output Routing* aktiviert.

### HINWEIS



Die Einträge 3250<sub>h</sub>:01<sub>h</sub> bis 3250<sub>h</sub>:04<sub>h</sub> haben dann **keine** Funktion mehr, bis das *Output Routing* wieder abgeschaltet wird.

- 09<sub>h</sub>: 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<sub>h</sub> kontrolliert werden kann. Details finden Sie im Kapitel *Output Routing*.

### Objektbeschreibung

Index	3252 <sub>h</sub>
Objektname	Digital Output Routing
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v2213-B1029645: Eintrag "Name" geändert von "Control Bit Of 60FE <sub>h</sub> :1h And Source For Brake Output" auf "Control Bit Of 60FE <sub>h</sub> :1h And Source For Brake Output".

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>
Subindex	01 <sub>h</sub>
Name	Control Bit Of 60FE <sub>h</sub> :1h And Source For Brake Output
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	1080 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	Control Bit Of 60FE <sub>h</sub> :1h And Source For Output #1
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	0090 <sub>h</sub>
<hr/>	
Subindex	03 <sub>h</sub>
Name	Control Bit Of 60FEh:1h And Source For Output #2
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0091 <sub>h</sub>

### 3320h Analog Input Values

#### Funktion

Dieses Objekt zeigt die Momentanwerte der Analogeingänge in benutzerdefinierten Einheiten an.

#### Objektbeschreibung

Index	3320 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Analog Input Values
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub>
Name	Analog Input #1 Value
Datentyp	INTEGER32
Zugriff	nur lesen

PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

## Beschreibung

Die benutzerdefinierten Einheiten setzen sich aus Offset (3321<sub>h</sub>) und Skalierungswert (3322<sub>h</sub>/ 3323<sub>h</sub>) zusammen. Sind beide noch mit Default-Werten beschrieben, wird der Wert in 3320<sub>h</sub> in der Einheit *ADC Digits* angegeben.

Formel zum Umrechnen von Digits in die jeweilige Einheit:

- Spannungseingang: x Digits \* 10 V / 1023 Digits
- Stromeingang (falls konfigurierbar): x Digits \* 20 mA / 1023 Digits

Für die Subeinträge gilt:

- Subindex 00<sub>h</sub>: Anzahl der Analogeingänge
- Subindex 01<sub>h</sub>: Analogwert 1
- Subindex 02<sub>h</sub>: Analogwert 2 (falls vorhanden)

## 3321h Analog Input Offsets

### Funktion

Offset, der zum eingelesenen Analogwert (3220<sub>h</sub>) addiert wird, bevor die Skalierung (Multiplikator aus dem Objekt 3322 und Teiler aus dem Objekt 3323<sub>h</sub>) vorgenommen wird.

### Objektbeschreibung

---

Index	3321 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Analog Input Offsets
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	Analog Input #1 Offset
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

## 3322h Analog Input Numerators

### Funktion

Wert, mit dem der eingelesene Analogwert (3220<sub>h</sub> + 3321<sub>h</sub>) multipliziert wird, bevor er in das Objekt 3320<sub>h</sub> geschrieben wird.

### Objektbeschreibung

Index	3322 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Analog Input Numerators
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Analog Input #1 Numerator
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>

## Beschreibung

Die Subindizes enthalten:

- Subindex 01<sub>h</sub>: Multiplikator für Analogeingang 1
- Subindex 02<sub>h</sub>: Multiplikator für Analogeingang 2 (falls vorhanden)

## 3323h Analog Input Denominators

### Funktion

Wert, mit dem der eingelesene Analogwert (3320<sub>h</sub>+ 3321<sub>h</sub>) dividiert wird, bevor er in das Objekt 3320<sub>h</sub> geschrieben wird.

### Objektbeschreibung

Index	3323 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Analog Input Denominators
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Analog Input #1 Denominator
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>

## Beschreibung

Die Subindizes enthalten:

- Subindex 01<sub>h</sub>: Teiler für Analogeingang 1
- Subindex 02<sub>h</sub>: 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	05 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Resistance [Ohm]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Inductance [H]
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	03 <sub>h</sub>
Name	Magnetic Flux [Vs]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	04 <sub>h</sub>
Name	Switch On Speed [rpm]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000078 <sub>h</sub>

---

Subindex	05 <sub>h</sub>
Name	Switch Off Speed [rpm]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000064 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 01<sub>h</sub>: Wicklungswiderstand. Float-Wert, hier als UNSIGNED32 dargestellt. Wird vom Auto-Setup ermittelt.
- 02<sub>h</sub>: Wicklungsinduktivität. Float-Wert, hier als UNSIGNED32 dargestellt. Wird vom Auto-Setup ermittelt.
- 03<sub>h</sub>: Verkettungsfluss. Float-Wert, hier als UNSIGNED32 dargestellt. Wird vom Auto-Setup ermittelt.
- 04<sub>h</sub>: Einschalt Drehzahl in U/min. Ab dieser Drehzahl wird der *Closed Loop* ( *Sensorless*) aktiviert, wenn vom Auto-Setup keine Sensoren erkannt wurden.
- 05<sub>h</sub>: Ausschalt Drehzahl in U/min. Unter dieser Drehzahl wird der *Closed Loop* ( *Sensorless*) deaktiviert, wenn vom Auto-Setup keine Sensoren erkannt wurden.

## 33A0h Feedback Incremental A/B/I 1

### Funktion

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



## Objektbeschreibung

Index	33A0 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Configuration
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	8001 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 01<sub>h</sub> (Configuration): folgende Bits haben eine Bedeutung:

- Bit 0: Wert = "0": Der Encoder verfügt nicht über einen Index. Wert = "1" : Encoder-Index ist vorhanden und soll verwendet werden.
- Bit 15: Wert = "1": der Encoder ist ein Singleturn-Absolut-Encoder.
- 02<sub>h</sub> (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.

## 33A1h Feedback Incremental A/B/I 2

### Funktion

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

### Objektbeschreibung

Index	33A1 <sub>h</sub>
Objektname	Feedback Incremental A/B/I 2
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B533384
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Configuration
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

## Beschreibung

### Beschreibung

Die Subindizes haben folgende Funktion:

- 01<sub>h</sub> (Configuration): folgende Bits haben eine Bedeutung:
  - Bit 0: Wert = "0": Der Encoder verfügt nicht über einen Index. Wert = "1" : Encoder-Index ist vorhanden und soll verwendet werden.
  - Bit 15: Wert = "1": der Encoder ist ein Singleturn-Absolut-Encoder.
- 02<sub>h</sub> (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.

## 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<sub>h</sub> auf "0" setzen.

Nach dem Schreiben der Objekte in die jeweiligen Subindizes, tragen Sie die Anzahl der gemappten Objekte in den Subindex 0<sub>h</sub> ein.

## Objektbeschreibung

Index	3502 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	07 <sub>h</sub>
Subindex	01 <sub>h</sub>
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60400010 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00050008 <sub>h</sub>
Subindex	03 <sub>h</sub>
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60600008 <sub>h</sub>
Subindex	04 <sub>h</sub>
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	607A0020 <sub>h</sub>
<hr/>	
Subindex	05 <sub>h</sub>
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60810020 <sub>h</sub>
<hr/>	
Subindex	06 <sub>h</sub>
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60420010 <sub>h</sub>
<hr/>	
Subindex	07 <sub>h</sub>
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60FE0120 <sub>h</sub>
<hr/>	
Subindex	08 <sub>h</sub>
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	09 <sub>h</sub>
Name	9th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	0A <sub>h</sub>
Name	10th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0B <sub>h</sub>
Name	11th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0C <sub>h</sub>
Name	12th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0D <sub>h</sub>
Name	13th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0E <sub>h</sub>
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0F <sub>h</sub>
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32

---

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	10 <sub>h</sub>
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 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<sub>h</sub> auf "0" setzen.

Nach dem Schreiben der Objekte in die jeweiligen Subindizes, tragen Sie die Anzahl der gemappten Objekte in den Subindex 0<sub>h</sub> ein.

### Objektbeschreibung

Index	3602 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries

Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub>
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60410010 <sub>h</sub>
<hr/>	
Subindex	02 <sub>h</sub>
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00050008 <sub>h</sub>
<hr/>	
Subindex	03 <sub>h</sub>
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60610008 <sub>h</sub>
<hr/>	
Subindex	04 <sub>h</sub>
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60640020 <sub>h</sub>
<hr/>	
Subindex	05 <sub>h</sub>
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein



Zulässige Werte	
Vorgabewert	60440010 <sub>h</sub>
<hr/>	
Subindex	06 <sub>h</sub>
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60FD0020 <sub>h</sub>
<hr/>	
Subindex	07 <sub>h</sub>
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	08 <sub>h</sub>
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	09 <sub>h</sub>
Name	9th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	0A <sub>h</sub>
Name	10th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	0B <sub>h</sub>
Name	11th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0C <sub>h</sub>
Name	12th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0D <sub>h</sub>
Name	13th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0E <sub>h</sub>
Name	14th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	0F <sub>h</sub>
Name	15th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---



---

Subindex	10 <sub>h</sub>
Name	16th Object To Be Mapped
Datentyp	UNSIGNED32

---

Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 3700h Deviation Error Option Code

### Funktion

Das Objekt enthält die auszuführende Aktion, wenn ein Schlepp- oder Schlupffehler ausgelöst wird.

### Objektbeschreibung

Index	3700 <sub>h</sub>
Objektname	Deviation Error Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FFFF <sub>h</sub>
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 <sub>h</sub> )
3 bis 32767	reserviert

## 3701h Limit Switch Error Option Code

### Funktion

Wird ein Endschalter überfahren, wird das Bit 7 ( *Warning*) in 6041<sub>h</sub> ( *Statusword*) gesetzt und die in diesem Objekt hinterlegte Aktion ausgeführt. Siehe Kapitel Begrenzung des Bewegungsbereichs.

## Objektbeschreibung

Index	3701 <sub>h</sub>
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 <sub>h</sub>
Firmware Version	FIR-v1748-B538662
Änderungshistorie	

## Beschreibung

Wert in Objekt 3701 <sub>h</sub>	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.

## 4012h HW Information

### Funktion

Dieses Objekt zeigt Informationen über die Hardware an.

### Objektbeschreibung

Index	4012 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	EEPROM Size In Bytes
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	HW Configuration #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

## Beschreibung

reserviert

## 4014h Operating Conditions

### Funktion

Dieses Objekt dient zum Auslesen aktueller Umgebungswerte der Steuerung.

### Objektbeschreibung

Index	4014 <sub>h</sub>
Objektname	Operating Conditions
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1540
Änderungshistorie	Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".

Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 02 geändert von "lesen/schreiben" auf "nur lesen".

Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Temperature PCB [d?C]" auf "Temperature PCB [Celsius \* 10]".

Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 03 geändert von "lesen/schreiben" auf "nur lesen".

Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 4 auf 6.

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	05 <sub>h</sub>
Subindex	01 <sub>h</sub>
Name	Voltage UB Power [mV]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	Voltage UB Logic [mV]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Subindex	03 <sub>h</sub>
Name	Temperature PCB [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	04 <sub>h</sub>
Name	Temperature Motor [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	05 <sub>h</sub>
Name	Temperature Microcontroller Chip [Celsius * 10]
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

Die Subindizes enthalten:

- 01<sub>h</sub>: aktuelle Versorgungsspannung in [mV]
- 02<sub>h</sub>: aktuelle Logikspannung in [mV]
- 03<sub>h</sub>: aktuelle Temperatur der Steuerungsplatine in [d°C] (Zehntelgrad)
- 04<sub>h</sub>: reserviert
- 05<sub>h</sub>: reserviert

## 4015h Special Drive Modes

### Funktion

Mit diesem Objekt können Sie die *speziellen Fahrmodi* aus- oder einschalten. Siehe Kapitel Spezielle Fahrmodi (Takt-Richtung und Analog-Drehzahl).

### Objektbeschreibung

Index	4015 <sub>h</sub>
Objektname	Special Drive Modes
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	



## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Special Drive Mode Configuration
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Virtual Config Switch Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktionen:

- 01<sub>h</sub>:
  - Wert="0"<sub>h</sub>: die *speziellen Fahrmodi* werden ausgeschaltet
  - Wert="2"<sub>h</sub>: die *speziellen Fahrmodi* werden eingeschaltet und der Modus wird im Subindex 02<sub>h</sub> eingestellt.
- 02<sub>h</sub>: Definiert den verwendeten Modus.

## 4016h Factory Settings

### Funktion

Dieses Objekt zeigt an, ob das Auto-Setup in der Produktion ausgeführt wurde.

### Objektbeschreibung

Index	4016 <sub>h</sub>
Objektname	Factory Settings
Object Code	ARRAY
Datentyp	UNSIGNED8

Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1738-B501312
Änderungshistorie	Firmware Version FIR-v2013-B726332: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Factory Autoseup Done
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

## Beschreibung

Für den Subindex 01<sub>h</sub> zulässige Werte:

- Wert = "0" : Das *Auto-Setup* wurde nicht ausgeführt.
- Wert = "1" : Das *Auto-Setup* wurde ausgeführt.

## 4021h Ballast Configuration

### Funktion

Mit diesem Objekt schalten Sie die Ballast-Schaltung ein oder aus und bestimmen deren Ansprechschwelle.

### Objektbeschreibung

Index	4021 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	Settings [Bit0: On/Off]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	UB Power Limit [mV]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000F3BF <sub>h</sub>

---

Subindex	03 <sub>h</sub>
Name	UB Power Hysteresis [mV]
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000001F4 <sub>h</sub>

---

## Beschreibung

Die Subindizes haben folgende Funktion:

- 01<sub>h</sub>:

- Bit 0: schaltet den Ballast ein (Wert = "1") oder aus (Wert = "0")
- 02<sub>h</sub>: Ansprechschwelle (Ein-/Ausschalten) der Ballast-Schaltung
- 03<sub>h</sub>: Hysterese für die Ansprechschwelle (Ein-/Ausschalten)

## 4040h Drive Serial Number

### Funktion

Dieses Objekt hält die Seriennummer der Steuerung.

### Objektbeschreibung

Index	4040 <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
-------	-------------------

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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	Bootloader Version
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	Bootloader Supported Fieldbus
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	03 <sub>h</sub>
Name	Bootloader Hw-group
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

## Beschreibung

Die Subindizes haben folgende Funktionen:

- 01<sub>h</sub>: Version des Bootloaders. Die 4 höchstwertigen Bytes erhalten die Hauptversionsnummer, die 4 niedrigwertigsten Bytes die Nebenversionsnummer. Beispiel für die Version 4.2: 00040002<sub>h</sub>
- 02<sub>h</sub>: 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<sub>h</sub>. Für die Beschreibung der Error-Codes schauen Sie unter Objekt 1003<sub>h</sub> nach.

### Objektbeschreibung

Index	603F <sub>h</sub>
Objektname	Error Code
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## Beschreibung

Bedeutung des Fehlers siehe Objekt 1003<sub>h</sub> (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 <sub>h</sub>
Objektname	Controlword
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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 <sub>h</sub>
Objektname	Statusword
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	nein

Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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<sub>h</sub>)

### VE (Voltage Enabled)

Spannung angelegt

### QS (Quick Stop)

Wert = "0": Steuerung befindet sich in dem Zustand "Quick Stop"

### SOD (Switched On Disabled)

Wert = "1": Steuerung befindet sich in dem Zustand "Switched On Disabled"

### WARN (Warning)

Wert = "1": Warnung

### SYNC (Synchronisation)

Wert = "1": Steuerung ist synchron zum Feldbus, Wert = "0": Steuerung ist nicht synchron zum Feldbus

### REM (Remote)

Remote (Wert des Bits immer "1" , außer wenn die speziellen Fahrmodi aktiviert sind )

### TARG

Zielvorgabe erreicht

### ILA (Internal Limit Active)

Limit überschritten

### OMS (Operation Mode Specific)

Bedeutung abhängig vom gewählten Betriebsmodus



**CLA (Closed Loop Active)**

Wert = "1": die Steuerung befindet sich im Status *Operation enabled* und der Closed Loop ist aktiviert.

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

Statusword (6041 <sub>h</sub> )	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 <sub>h</sub>
Objektname	VI Target Velocity
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00C8 <sub>h</sub>
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 <sub>h</sub>
Objektname	VI Velocity Demand
Object Code	VARIABLE
Datentyp	INTEGER16

Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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 <sub>h</sub>
Objektname	VI Velocity Actual Value
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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 <sub>h</sub>
Objektname	VI Velocity Min Max Amount
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

---

### Wertebeschreibung

---

Subindex	00 <sub>h</sub>
----------	-----------------

---

Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	MinAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	MaxAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00007530 <sub>h</sub>

## Beschreibung

Subindex 1 enthält die Minimalgeschwindigkeit.

Subindex 2 enthält die Maximalgeschwindigkeit.

Wird eine Zielgeschwindigkeit (Objekt 6042<sub>h</sub>) vom Betrag her kleiner als die Minimalgeschwindigkeit angegeben, gilt die Minimalgeschwindigkeit und das Bit 11 (Internal Limit Reached) in 6041<sub>h</sub> Statusword<sub>h</sub> wird gesetzt.

Eine Zielgeschwindigkeit größer als die Maximalgeschwindigkeit setzt die Geschwindigkeit auf die Maximalgeschwindigkeit und das Bit 11 (Internal Limit Reached) in 6041<sub>h</sub> Statusword<sub>h</sub> wird gesetzt.

## 6048h VI Velocity Acceleration

### Funktion

Setzt die Beschleunigungsrampe im Velocity Mode (siehe [Velocity](#)).

### Objektbeschreibung

---

Index	6048 <sub>h</sub>
Objektname	VI Velocity Acceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426

Änderungshistorie
 

---

**Wertebeschreibung**

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>

**Beschreibung**

Die Beschleunigung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01<sub>h</sub>: enthält die Geschwindigkeitsänderung.

Subindex 02<sub>h</sub>: enthält die Zeitänderung.

**6049h VI Velocity Deceleration**
**Funktion**

Setzt die Verzögerung (Bremsrampe) im Velocity Mode (siehe [Velocity](#)).

**Objektbeschreibung**

Index	6049 <sub>h</sub>
Objektname	VI Velocity Deceleration
Object Code	RECORD

Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

---

## Wertebeschreibung

---

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

---



---

Subindex	01 <sub>h</sub>
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 <sub>h</sub>

---



---

Subindex	02 <sub>h</sub>
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>

---

## Beschreibung

Die Verzögerung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01<sub>h</sub>: enthält die Geschwindigkeitsänderung.

Subindex 02<sub>h</sub>: 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 <sub>h</sub>
Objektname	VI Velocity Quick Stop
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>

## Beschreibung

Die Verzögerung wird als Bruch in benutzerdefinierten Einheiten angegeben:

Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01<sub>h</sub>: enthält die Geschwindigkeitsänderung.

Subindex 02<sub>h</sub>: 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 <sub>h</sub>
Objektname	VI Dimension Factor
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	VI Dimension Factor Numerator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	VI Dimension Factor Denominator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

### 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 <sub>h</sub>
Objektname	Quick Stop Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0002 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

### Beschreibung

Wert in Objekt 605A <sub>h</sub>	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 <sub>h</sub> ) 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 <sub>h</sub> ) 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 <sub>h</sub>
Objektname	Shutdown Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## Beschreibung

Wert in Objekt 605B <sub>h</sub>	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 <sub>h</sub>
Objektname	Disable Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## Beschreibung

Wert in Objekt 605C <sub>h</sub>	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<sub>h</sub> das Bit 8 (Halt) gesetzt wird.

### Objektbeschreibung

Index	605D <sub>h</sub>
Objektname	Halt Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## Beschreibung

Wert in Objekt 605D <sub>h</sub>	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> ( <u>6085<sub>h</sub></u> )
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 <sub>h</sub>
Objektname	Fault Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0002 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## Beschreibung

Wert in Objekt 605E <sub>h</sub>	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 <sub>h</sub> )
3 bis 32767	Reserviert

## 6060h Modes Of Operation

### Funktion

In dieses Objekt wird der gewünschte Betriebsmodus eingetragen.

### Objektbeschreibung

Index	6060 <sub>h</sub>
Objektname	Modes Of Operation
Object Code	VARIABLE
Datentyp	INTEGER8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

## Beschreibung

Modus	Beschreibung
-2	Auto-Setup
-1	Takt-Richtungsmodus
0	No mode change/no mode assigned
1	Profile Position Mode
2	Velocity Mode
3	Profile Velocity Mode
4	Profile Torque Mode
5	Reserved
6	Homing Mode
7	Interpolated Position Mode
8	Cyclic Synchronous Position Mode
9	Cyclic Synchronous Velocity Mode
10	Cyclic Synchronous Torque Mode

## 6061h Modes Of Operation Display

### Funktion

Zeigt den aktuellen Betriebsmodus. Siehe auch [6060h Modes Of Operation](#).

### Objektbeschreibung

Index	6061 <sub>h</sub>
Objektname	Modes Of Operation Display
Object Code	VARIABLE
Datentyp	INTEGER8
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## 6062h Position Demand Value

### Funktion

Gibt die aktuelle Sollposition in [benutzerdefinierten Einheiten](#) an.

### Objektbeschreibung

Index	6062 <sub>h</sub>
Objektname	Position Demand Value
Object Code	VARIABLE

Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

---

## 6063h Position Actual Internal Value

### Funktion

Enthält die aktuelle Istposition in Encoder-Inkrementen. Im Gegensatz zu den Objekten 6062<sub>h</sub> und 6064<sub>h</sub> 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<sub>h</sub> Null, sind die Zahlenwerte dieses Objekts ungültig.

### Objektbeschreibung

---

Index	6063 <sub>h</sub>
Objektname	Position Actual Internal Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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 <sub>h</sub>
Objektname	Position Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32

Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

---

## 6065h Following Error Window

### Funktion

Definiert den maximal erlaubten Schleppfehler in benutzerdefinierten Einheiten symmetrisch zur Sollposition.

### Objektbeschreibung

---

Index	6065 <sub>h</sub>
Objektname	Following Error Window
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000100 <sub>h</sub>
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<sub>h</sub> gesetzt. Die Abweichung muss länger andauern als die Zeit in dem Objekt 6066<sub>h</sub>.

Wird der Wert des "Following Error Window" auf "FFFFFFFF"<sub>h</sub> gesetzt, wird die Schleppfehler-Überwachung abgeschaltet.

In dem Objekt 3700<sub>h</sub> kann eine Reaktion auf den Schleppfehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt 1003<sub>h</sub> eingetragen.

## 6066h Following Error Time Out

### Funktion

Zeit in Millisekunden, bis ein zu großer Schleppfehler zu einer Fehlermeldung führt.

### Objektbeschreibung

---

Index	6066 <sub>h</sub>
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 <sub>h</sub>
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<sub>h</sub> überschritten wird, wird das Bit 13 im Objekt 6041<sub>h</sub> gesetzt. Die Abweichung muss länger als die Zeit in diesem Objekt anhalten.

In dem Objekt 3700<sub>h</sub> kann eine Reaktion auf den Schleppfehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt 1003<sub>h</sub> 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 <sub>h</sub>
Objektname	Position Window
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000000A <sub>h</sub>
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<sub>h</sub> gesetzt. Die Bedingung muss länger erfüllt sein als die im Objekt 6068<sub>h</sub> definierte Zeit.

Wird der Wert auf "FFFFFFFF"<sub>h</sub> gesetzt, wird die Überwachung abgeschaltet.

## 6068h Position Window Time

### Funktion

Die Istposition muss sich für diese Zeit in Millisekunden innerhalb des "Position Window" (6067<sub>h</sub>) befinden, damit die Zielposition als erreicht gilt in den Modi Profile Position und Interpolated Position Mode.

### Objektbeschreibung

Index	6068 <sub>h</sub>
Objektname	Position Window Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0064 <sub>h</sub>
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<sub>h</sub>, wird das Bit 10 im Objekt 6041<sub>h</sub> gesetzt. Die Bedingung muss länger erfüllt sein als die im Objekt 6068<sub>h</sub> definierte Zeit.

## 606Bh Velocity Demand Value

### Funktion

Vorgabegeschwindigkeit in benutzerdefinierten Einheiten für den Geschwindigkeitsregler.

### Objektbeschreibung

Index	606B <sub>h</sub>
Objektname	Velocity Demand Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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 <sub>h</sub>
Objektname	Velocity Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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 <sub>h</sub>
Objektname	Velocity Window
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	001E <sub>h</sub>
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<sub>h</sub> gesetzt. Die Bedingung muss länger erfüllt sein als die im Objekt 606E<sub>h</sub> 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<sub>h</sub>) befinden, damit das Ziel als erreicht gilt.

### Objektbeschreibung

---

Index	606E <sub>h</sub>
Objektname	Velocity Window Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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<sub>h</sub>, wird das Bit 10 im Objekt 6041<sub>h</sub> 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 <sub>h</sub>
Objektname	Velocity Threshold
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
Firmware Version	FIR-v2013-B726332
Änderungshistorie	

---

## Beschreibung

Ist die Istgeschwindigkeit größer als der Wert in 606F<sub>h</sub> (Velocity Threshold) für eine Zeit von 6070<sub>h</sub> (Velocity Threshold Time), hat das Bit 12 in 6041<sub>h</sub> (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<sub>h</sub> im Modus Profile Velocity als ungleich Null gilt.

### Objektbeschreibung

---

Index	6070 <sub>h</sub>
Objektname	Velocity Threshold Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
Firmware Version	FIR-v2013-B726332
Änderungshistorie	

---

## Beschreibung

Ist die Istgeschwindigkeit größer als der Wert in 606F<sub>h</sub> (Velocity Threshold) für eine Zeit von 6070<sub>h</sub> (Velocity Threshold Time), hat das Bit 12 in 6041<sub>h</sub> (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 <sub>h</sub>
Objektname	Target Torque
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben

PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".

---

## Beschreibung

Dieses Objekt wird als Tausendstel des Drehmoments gerechnet, z.B. der Wert "500" bedeutet "50%" des Nenndrehmoments, "1100" ist äquivalent zu 110%. Das Nenndrehmoment entspricht dem Nennstrom im Objekt 203B<sub>h</sub>:01.

Das Minimum von 6073<sub>h</sub> und 6072<sub>h</sub> wird als Limit für das Drehmoment in 6071<sub>h</sub> verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031<sub>h</sub>) nicht übersteigen.

## 6072h Max Torque

### Funktion

Das Objekt beschreibt das maximale Drehmoment für den Profile Torque und Cyclic Synchronous Torque Modus in Promille des Nenndrehmoments.

### Objektbeschreibung

---

Index	6072 <sub>h</sub>
Objektname	Max Torque
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0064 <sub>h</sub>
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<sub>h</sub>:01.

Das Minimum von 6073<sub>h</sub> und 6072<sub>h</sub> wird als Limit für das Drehmoment in 6071<sub>h</sub> verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031<sub>h</sub>) nicht übersteigen.

## 6073h Max Current

### Funktion

Enthält den Maximalstrom in Promille des eingestellten Nennstroms an. Wird durch den maximalen Motorstrom (2031<sub>h</sub>) begrenzt. Siehe auch [I2t Motor-Überlastungsschutz](#).

#### HINWEIS



Für Schrittmotoren wird nur der Nennstrom und kein Maximalstrom angegeben. Daher sollte der Wert von 6073<sub>h</sub> den Wert 1000 (100%) in der Regel nicht überschreiten.

### Objektbeschreibung

Index	6073 <sub>h</sub>
Objektname	Max Current
Object Code	VARIABLE
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03E8 <sub>h</sub>
Firmware Version	FIR-v1825-B577172
Änderungshistorie	

### Beschreibung

Der Maximalstrom wird in Promille des Nennstroms wie folgt berechnet:

$$(6073_{h} * 203B_{h}:01) / 1000$$

Der Maximalstrom bestimmt:

- den Maximalstrom für den [I2t Motor-Überlastungsschutz](#),
- den Sollstrom im *Open Loop*-Betrieb.

## 6074h Torque Demand

### Funktion

Momentaner vom Rampengenerator geforderter Drehmomentsollwert in Promille des Nenndrehmoments für den internen Regler.

### Objektbeschreibung

Index	6074 <sub>h</sub>
Objektname	Torque Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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<sub>h</sub>:01.

Das Minimum von 6073<sub>h</sub> und 6072<sub>h</sub> wird als Limit für das Drehmoment in 6071<sub>h</sub> verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031<sub>h</sub>) nicht übersteigen.

## 6075h Motor Rated Current

### Funktion

Enthält den in 203B<sub>h</sub>:01<sub>h</sub> 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 <sub>h</sub>
Objektname	Torque Actual Value
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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<sub>h</sub>:01.

Das Minimum von 6073<sub>h</sub> und 6072<sub>h</sub> wird als Limit für das Drehmoment in 6071<sub>h</sub> verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031<sub>h</sub>) 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 <sub>h</sub>
Objektname	Target Position
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000FA0 <sub>h</sub>
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 <sub>h</sub>
Objektname	Position Range Limit
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Min Position Range Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Max Position Range Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

Wird dieser Bereich über- oder unterschritten, erfolgt ein Überlauf. Um diesen Überlauf zu verhindern, können im Objekt 607D<sub>h</sub> ("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 <sub>h</sub>
Objektname	Home Offset
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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 <sub>h</sub>
Objektname	Software Position Limit
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Min Position Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Max Position Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

Die absolute Zielposition muss innerhalb der hier gesetzten Grenzen liegen. Der Home Offset (607C<sub>h</sub>) wird nicht berücksichtigt.

## 607Eh Polarity

### Funktion

Mit diesem Objekt lässt sich die Drehrichtung umkehren.

## Objektbeschreibung

Index	607E <sub>h</sub>
Objektname	Polarity
Object Code	VARIABLE
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "RX-PDO".

## Beschreibung

Generell gilt für die Richtungsumkehr: Ist ein Bit auf den Wert "1" gesetzt, ist die Umkehrung aktiviert. Ist der Wert "0", ist die Drehrichtung wie im jeweiligen Modus beschrieben.

7	6	5	4	3	2	1	0
POS	VEL						

### VEL (Velocity)

Umkehr der Drehrichtung in folgenden Modi:

- [Profile Velocity Mode](#)
- [Cyclic Synchronous Velocity Mode](#)

### POS (Position)

Umkehr der Drehrichtung in folgenden Modi:

- [Profile Position Mode](#)
- [Cyclic Synchronous Position Mode](#)

### TIPP



Sie können ein Invertieren des Drehfeldes erzwingen, dass alle Betriebsmodi betrifft. Siehe Objekt [3212<sub>h</sub>:02<sub>h</sub>](#).

## 607Fh Max Profile Velocity

### Funktion

Gibt die maximale Geschwindigkeit in benutzerdefinierten Einheiten an, für die Modi [Profile Position](#), [Interpolated Position Mode](#) (nur wenn der [Closed Loop](#) aktiviert ist) und [Profile Velocity](#).

### Objektbeschreibung

Index	607F <sub>h</sub>
-------	-------------------

Objektname	Max Profile Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00007530 <sub>h</sub>
Firmware Version	FIR-v1540
Änderungshistorie	<p>Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Max profile velocity" auf "Max Profile Velocity".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Datentyp" geändert von "INTEGER16" auf "UNSIGNED32".</p> <p>Firmware Version FIR-v1738-B501312: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "nur lesen" auf "lesen/schreiben".</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "TX-PDO" auf "RX-PDO".</p>

---

## 6080h Max Motor Speed

### Funktion

Enthält die maximal zulässige Geschwindigkeit des Motors in benutzerdefinierten Einheiten.

### Objektbeschreibung

---

Index	6080 <sub>h</sub>
Objektname	Max Motor Speed
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Bewegung
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	<ul style="list-style-type: none"> <li>■ PD6-E891L95-E-65-5: 00007530<sub>h</sub></li> <li>■ PD6-E891L95-ES-65-5: 00007530<sub>h</sub></li> <li>■ PD6-EB80CD-E-65-5: 0000109A<sub>h</sub></li> <li>■ PD6-EB80CD-ES-65-5: 0000109A<sub>h</sub></li> <li>■ PD6-E891S95-E-65-5: 00007530<sub>h</sub></li> <li>■ PD6-E891S95-ES-65-5: 00007530<sub>h</sub></li> <li>■ PD6-E891M95-E-65-5: 00007530<sub>h</sub></li> <li>■ PD6-E891M95-ES-65-5: 00007530<sub>h</sub></li> <li>■ PD6-EB80SD-E-65-5: 00001004<sub>h</sub></li> <li>■ PD6-EB80SD-ES-65-5: 00001004<sub>h</sub></li> <li>■ PD6-EB80MD-E-65-5: 0000109A<sub>h</sub></li> <li>■ PD6-EB80MD-ES-65-5: 0000109A<sub>h</sub></li> </ul>

	<ul style="list-style-type: none"> <li>■ PD6-EB80LD-E-65-5: 000010CC<sub>h</sub></li> <li>■ PD6-EB80LD-ES-65-5: 000010CC<sub>h</sub></li> </ul>
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 <sub>h</sub>
Objektname	Profile Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## 6082h End Velocity

### Funktion

Gibt die Geschwindigkeit am Ende der gefahrenen Rampe in benutzerdefinierten Einheiten an.

### Objektbeschreibung

Index	6082 <sub>h</sub>
Objektname	End Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32

Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

---

## 6083h Profile Acceleration

### Funktion

Gibt die maximale Beschleunigung in benutzerdefinierten Einheiten an.

### Objektbeschreibung

Index	6083 <sub>h</sub>
Objektname	Profile Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

---

## 6084h Profile Deceleration

### Funktion

Gibt die maximale Verzögerung (Bremsrampe) in benutzerdefinierten Einheiten an. Wird durch 60C6<sub>h</sub> limitiert.

### Objektbeschreibung

Index	6084 <sub>h</sub>
Objektname	Profile Deceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 <sub>h</sub>
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<sub>h</sub> (Max Deceleration) und ggf. 60A4<sub>h</sub> (Profile Jerk).

### Objektbeschreibung

Index	6085 <sub>h</sub>
Objektname	Quick Stop Deceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 <sub>h</sub>
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 <sub>h</sub>
Objektname	Motion Profile Type
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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 <sub>h</sub>
Objektname	Torque Slope
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000064 <sub>h</sub>
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<sub>h</sub>:01.

Das Minimum von 6073<sub>h</sub> und 6072<sub>h</sub> wird als Limit für das Drehmoment in 6071<sub>h</sub> verwendet.

Das Zieldrehmoment kann das Spitzendrehmoment (proportional zum maximalen Motorstrom in 2031<sub>h</sub>) nicht übersteigen.

## 608Fh Position Encoder Resolution

### Funktion

Enthält die physikalische Auflösung (siehe Objekte 60E6<sub>h</sub>/ 60EB<sub>h</sub>) des Encoders/Sensors, der für die Positionsregelung verwendet wird (siehe 3203h Feedback Selection).

### Objektbeschreibung

Index	608F <sub>h</sub>
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>

Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Encoder Increments
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Motor Revolutions
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Position Encoder Resolution = Encoder Increments (608F<sub>h</sub>:01<sub>h</sub>) / Motor Revolutions (608F<sub>h</sub>:02<sub>h</sub>)

## 6090h Velocity Encoder Resolution

### Funktion

Enthält die physikalische Auflösung (siehe Objekte 60E6<sub>h</sub>/ 60EB<sub>h</sub>) des Encoders/Sensors, der für die Drehzahlregelung verwendet wird (siehe 3203h Feedback Selection).

## Objektbeschreibung

Index	6090 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>
Subindex	01 <sub>h</sub>
Name	Encoder Increments Per Second
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	Motor Revolutions Per Second
Datentyp	INTEGER32

Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Velocity Encoder Resolution = Encoder Increments per second (6090<sub>h</sub>:01<sub>h</sub>) / Motor Revolutions per second (6090<sub>h</sub>:02<sub>h</sub>)

## 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Motor Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Shaft Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Gear Ratio = Motor Revolutions (6091<sub>h</sub>:01<sub>h</sub>) / Shaft Revolutions (6091<sub>h</sub>:02<sub>h</sub>)

## 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 <sub>h</sub>
Objektname	Feed Constant
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Feed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Shaft Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Feed Constant = Feed (6092<sub>h</sub>:01<sub>h</sub>) / Shaft Revolutions (6092<sub>h</sub>:02<sub>h</sub>)

## 6096h Velocity Factor

### Funktion

Dieses Objekt beinhaltet den Faktor, der zum Umrechnen von benutzerdefinierten Geschwindigkeitseinheiten verwendet wird. Siehe Kapitel [Benutzerdefinierte Einheiten](#).

### Objektbeschreibung

Index	6096 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Numerator
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	Divisor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

## Beschreibung

Die Subindizes haben folgende Funktionen:

- 01<sub>h</sub>: Zähler des Faktors
- 02<sub>h</sub>: 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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	

Vorgabewert	02 <sub>h</sub>
Subindex	01 <sub>h</sub>
Name	Numerator
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	Divisor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktionen:

- 01<sub>h</sub>: Zähler des Faktors
- 02<sub>h</sub>: Nenner des Faktors

## 6098h Homing Method

### Funktion

Dieses Objekt definiert die Referenzfahrt-Methode im Homing Mode.

### Objektbeschreibung

Index	6098 <sub>h</sub>
Objektname	Homing Method
Object Code	VARIABLE
Datentyp	INTEGER8
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	23 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## 6099h Homing Speed

### Funktion

Gibt die Geschwindigkeiten für den Homing Mode (6098<sub>h</sub>) in benutzerdefinierten Einheiten an.

### Objektbeschreibung

Index	6099 <sub>h</sub>
Objektname	Homing Speed
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Speed During Search For Switch
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000032 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Speed During Search For Zero
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000000A <sub>h</sub>

### 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 <sub>h</sub>
Objektname	Homing Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

---

Subindex	01 <sub>h</sub>
Name	Numerator
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

Subindex	02 <sub>h</sub>
Name	Divisor
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

## Beschreibung

Die Subindizes haben folgende Funktionen:

- 01<sub>h</sub>: Zähler des Faktors
- 02<sub>h</sub>: Nenner des Faktors

## 60A4h Profile Jerk

### Funktion

Im Falle einer Rampe mit begrenztem Ruck kann in diesem Objekt die Größe des Rucks eingetragen werden. Ein Eintrag mit dem Wert "0" bedeutet, dass der Ruck nicht begrenzt ist.

## Objektbeschreibung

Index	60A4 <sub>h</sub>
Objektname	Profile Jerk
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	<p>Firmware Version FIR-v1614: Eintrag "Name" geändert von "End Acceleration Jerk" auf "Begin Deceleration Jerk".</p> <p>Firmware Version FIR-v1614: Eintrag "Name" geändert von "Begin Deceleration Jerk" auf "End Acceleration Jerk".</p>

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>
Subindex	01 <sub>h</sub>
Name	Begin Acceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	Begin Deceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>
Subindex	03 <sub>h</sub>
Name	End Acceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>
<hr/>	
Subindex	04 <sub>h</sub>
Name	End Deceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 <sub>h</sub>

### Beschreibung

- Subindex 01<sub>h</sub> ( *Begin Acceleration Jerk*): Anfangsruck bei Beschleunigung
- Subindex 02<sub>h</sub> ( *Begin Deceleration Jerk*): Anfangsruck bei Bremsung
- Subindex 03<sub>h</sub> ( *End Acceleration Jerk*): Abschlussruck bei Beschleunigung
- Subindex 04<sub>h</sub> ( *End Deceleration Jerk*): Abschlussruck bei Bremsung

### 60A8h SI Unit Position

#### Funktion

Dieses Objekt beinhaltet die Positionseinheit. Siehe Kapitel [Benutzerdefinierte Einheiten](#).

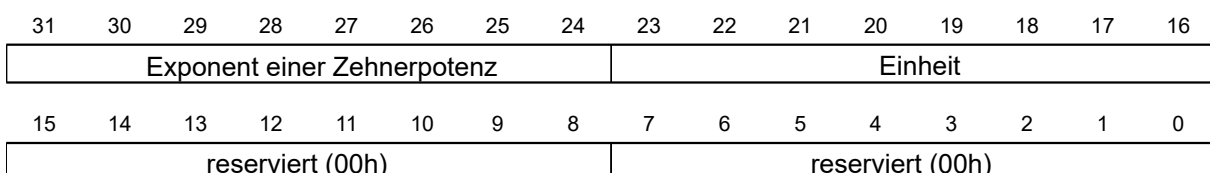
#### Objektbeschreibung

Index	60A8 <sub>h</sub>
Objektname	SI Unit Position
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FF410000 <sub>h</sub>
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

### Beschreibung

Das Objekt 60A8<sub>h</sub> enthält :

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



## 60A9h SI Unit Velocity

### Funktion

Dieses Objekt beinhaltet die Geschwindigkeitseinheit. Siehe Kapitel [Benutzerdefinierte Einheiten](#).

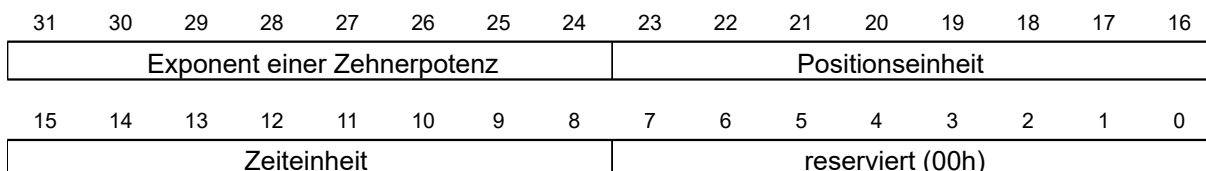
### Objektbeschreibung

Index	60A9 <sub>h</sub>
Objektname	SI Unit Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00B44700 <sub>h</sub>
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

### Beschreibung

Das Objekt 60A9<sub>h</sub> enthält :

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



## 60B0h Position Offset

### Funktion

Offset für den Positionssollwert in [benutzerdefinierten Einheiten](#). Wird im Modus [Cyclic Synchronous Position](#) berücksichtigt.

### Objektbeschreibung

Index	60B0 <sub>h</sub>
Objektname	Position Offset
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

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 <sub>h</sub>
Objektname	Velocity Offset
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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 <sub>h</sub>
Objektname	Torque Offset
Object Code	VARIABLE
Datentyp	INTEGER16
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	1st Set-point
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

### Beschreibung

Der Wert wird zum nächsten Synchronisationszeitpunkt übernommen.

## 60C2h Interpolation Time Period

### Funktion

Dieses Objekt enthält die Interpolationszeit.

### Objektbeschreibung

Index	60C2 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Interpolation Time Period Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Interpolation Time Index
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FD <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktionen:

- 01<sub>h</sub>: Interpolationszeit.
- 02<sub>h</sub>: Zehnerexponent der Interpolationszeit: muss den Wert -3 halten (entspricht der Zeitbasis in Millisekunden).

Es gilt dabei: Zykluszeit = Wert des 60C2<sub>h</sub>:01<sub>h</sub> \* 10<sup>Wert des 60C2:02</sup> 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 <sub>h</sub>
Objektname	Interpolation Data Configuration
Object Code	RECORD
Datentyp	INTERPOLATION_DATA_CONFIGURATION
Speicherbar	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1512
Änderungshistorie	<p>Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 05 geändert von "lesen/schreiben" auf "nur schreiben".</p> <p>Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 06 geändert von "lesen/schreiben" auf "nur schreiben".</p> <p>Firmware Version FIR-v1626: Eintrag "Speicherbar" geändert von "nein" auf "ja, Kategorie: Applikation".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p>

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 <sub>h</sub>



Subindex	01 <sub>h</sub>
Name	MaximumBufferSize
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	ActualBufferSize
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
Subindex	03 <sub>h</sub>
Name	BufferOrganization
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>
Subindex	04 <sub>h</sub>
Name	BufferPosition
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>
Subindex	05 <sub>h</sub>
Name	SizeOfDataRecord
Datentyp	UNSIGNED8
Zugriff	nur schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>
Subindex	06 <sub>h</sub>
Name	BufferClear
Datentyp	UNSIGNED8

Zugriff	nur schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

---

## Beschreibung

Der Wert des Subindex 01<sub>h</sub> enthält die maximal mögliche Anzahl der interpolierten Datensätze.

Der Wert des Subindex 02<sub>h</sub> enthält die momentane Anzahl der interpolierten Datensätze.

Wenn Subindex 03<sub>h</sub> "00<sub>h</sub>" ist, bedeutet das eine FIFO-Puffer-Organisation, wenn es "01<sub>h</sub>" ist, gibt es eine Ring-Puffer-Organisation an.

Der Wert des Subindex 04<sub>h</sub> ist ohne Einheit und gibt den nächsten freien Puffer-Einstiegspunkt an.

Der Wert des Subindex 05<sub>h</sub> wird in der Einheit "Byte" angegeben.

Wenn der Wert "00<sub>h</sub>" in den Subindex 06<sub>h</sub> geschrieben wird, löscht es die eingegangenen Daten im Puffer, deaktiviert den Zugriff und löscht alle Interpolierten Datensätze.

Wenn der Wert "01<sub>h</sub>" in den Subindex 06<sub>h</sub> 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 <sub>h</sub>
Objektname	Max Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 <sub>h</sub>
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 <sub>h</sub>
Objektname	Max Deceleration

Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

---

## 60E4h Additional Position Actual Value

### Funktion

Enthält die aktuelle Istposition aller vorhandenen Rückführungen in benutzerdefinierten Einheiten.

### Objektbeschreibung

---

Index	60E4 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

---

Subindex	01 <sub>h</sub> - 03 <sub>h</sub>
Name	Additional Position Actual Value #1 - #3
Datentyp	INTEGER32

Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>:  
 Subindex n enthält die aktuelle Istposition der entsprechenden Rückführung.  
 Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

## 60E5h Additional Velocity Actual Value

### Funktion

Enthält die aktuelle Istgeschwindigkeit aller vorhandenen Rückführungen in benutzerdefinierten Einheiten.

### Objektbeschreibung

Index	60E5 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

Subindex	01 <sub>h</sub> - 03 <sub>h</sub>
Name	Additional Velocity Actual Value #1 - #3
Datentyp	INTEGER32
Zugriff	nur lesen

PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

---

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>:  
 Subindex n enthält die aktuelle Istgeschwindigkeit der entsprechenden Rückführung.  
 Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

## 60E6h Additional Position Encoder Resolution - Encoder Increments

### Funktion

Mit diesem Objekt und mit 60EB<sub>h</sub> wird die Auflösung jeder vorhandenen Rückführung berechnet.

### Objektbeschreibung

---

Index	60E6 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

---



---

Subindex	01 <sub>h</sub> - 03 <sub>h</sub>
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #1 - #3
Datentyp	INTEGER32
Zugriff	lesen/schreiben

PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>:  
 Subindex n enthält die Anzahl der Inkremente der entsprechenden Rückführung.  
 Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

Die Auflösung der Rückführung "n" berechnet sich wie folgt:

Position Encoder Resolution = Encoder Increments (60E6<sub>h</sub>:01<sub>h</sub>) / Motor Revolutions (60EB<sub>h</sub>:02<sub>h</sub>)

### 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<sub>h</sub> können Sie die Getriebeübersetzung jeder vorhandenen Rückführung einstellen.

### Objektbeschreibung

Index	60E8 <sub>h</sub>
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 <sub>h</sub>
----------	-----------------

Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub> - 03 <sub>h</sub>
Name	Additional Gear Ratio - Motor Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>: Subindex "n" enthält die Anzahl der Motorumdrehungen für die entsprechende Rückführung. Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

Die Getriebeübersetzung der Rückführung "n" berechnet sich wie folgt:

$$\text{Gear Ratio} = \text{Motor Shaft Revolutions (60E8}_h:n_h) / \text{Driving Shaft Revolutions (60ED}_h:n_h)$$

## 60E9h Additional Feed Constant - Feed

### Funktion

In diesem Objekt und in 60EE<sub>h</sub> können Sie eine Vorschubkonstante für jede vorhandene Rückführung einstellen.

### Objektbeschreibung

Index	60E9 <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

Subindex	01 <sub>h</sub> - 03 <sub>h</sub>
Name	Additional Feed Constant - Feed Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>: Subindex "n" enthält den Vorschub in benutzerdefinierten Einheiten für die entsprechende Rückführung.

Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel Konfigurieren der Sensoren.

Die Vorschubkonstante der Rückführung "n" berechnet sich wie folgt:

Feed Constant = Feed (60E9<sub>h:n<sub>h</sub></sub>) / Driving Shaft Revolutions (60EE<sub>h:n<sub>h</sub></sub>)

## 60EB<sub>h</sub> Additional Position Encoder Resolution - Motor Revolutions

### Funktion

Mit diesem Objekt und mit 60E6<sub>h</sub> wird die Auflösung jeder vorhandenen Rückführung berechnet.

### Objektbeschreibung

Index	60EB <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

Subindex	01 <sub>h</sub> - 03 <sub>h</sub>
Name	Additional Position Encoder Resolution - Motor Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert="1" bis "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>:  
 Subindex n enthält die Anzahl der Motorumdrehungen der entsprechenden Rückführung.  
 Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel [Konfigurieren der Sensoren](#).

Die Auflösung der Rückführung "n" berechnet sich wie folgt:

Position Encoder Resolution = Encoder Increments (60E6<sub>h</sub>:n<sub>h</sub>) / Motor Revolutions (60EB<sub>h</sub>:n<sub>h</sub>)

## 60EDh Additional Gear Ratio - Driving Shaft Revolutions

### Funktion

In diesem Objekt und in 60E8<sub>h</sub> können Sie die Getriebeübersetzung jeder vorhandenen Rückführung einstellen.

### Objektbeschreibung

Index	60ED <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

Subindex	01 <sub>h</sub> - 03 <sub>h</sub>
Name	Additional Gear Ratio - Driving Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>: Subindex "n" enthält die Anzahl der Umdrehungen der Abtriebsachse für die entsprechende Rückführung.  
 Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel [Konfigurieren der Sensoren](#).

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<sub>h</sub>](#) können Sie eine Vorschubkonstante für jede vorhandene Rückführung einstellen.

### Objektbeschreibung

Index	60EE <sub>h</sub>
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 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

---

Subindex	01 <sub>h</sub> - 03 <sub>h</sub>
Name	Additional Feed Constant - Driving Shaft Revolutions Feedback Interface #1 - #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

---

## Beschreibung

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub>: Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- n<sub>h</sub>: Subindex "n" enthält die Anzahl der Umdrehungen der Abtriebsachse für die entsprechende Rückführung.  
Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*. Die Reihenfolge der restlichen Rückführungen entspricht der Tabelle im Kapitel [Konfigurieren der Sensoren](#).

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 <sub>h</sub>
Objektname	Positioning Option Code
Object Code	VARIABLE
Datentyp	UNSIGNED16

Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 <sub>h</sub>
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<sub>h</sub> = "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 <u>6064<sub>h</sub></u> ) ausgeführt.
1	1	Reserviert

#### RRO (Request-Response Option)

Diese Bits bestimmen das Verhalten bei der Übergabe des Controlwords 6040<sub>h</sub> 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<sub>h</sub> auf den Wert "0" gesetzt.



#### HINWEIS

Diese Optionen bringen die Steuerung dazu, das Objekt Controlword 6040<sub>h</sub> 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.

Bit 5	Bit 4	Definition
1	1	Reserviert

### RADO (Rotary Axis Direction Option)

Diese Bits bestimmen die Drehrichtung im "Profile Position" Modus.

Bit 7	Bit 6	Definition
0	0	Normale Positionierung ähnlich einer linearen Achse: Falls eines der "Position Range Limits" $607B_h:01_h$ und $02_h$ erreicht oder überschritten wird, wird der Vorgabewert automatisch an das andere Ende der Limits übertragen. Nur mit dieser Bitkombination ist eine Bewegung größer als der Modulo-Wert möglich.
0	1	Positionierung nur in negativer Richtung: falls die Zielposition größer als die aktuelle Position ist fährt die Achse über das "Min Position Range Limit" aus Objekt $607D_h:01_h$ zu der Zielposition.
1	0	Positionierung nur in positiver Richtung: falls die Zielposition kleiner als die aktuelle Position ist fährt die Achse über das "Max Position Range Limit" aus Objekt $607D_h:01_h$ zu der Zielposition.
1	1	Positionierung mit dem kürzesten Weg zur Zielposition. Falls die Differenz zwischen aktueller Position und Zielposition in einem $360^\circ$ System kleiner als $180^\circ$ 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 <sub>h</sub>
Objektname	Following Error Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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 <sub>h</sub>
Objektname	Max Slippage
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000190 <sub>h</sub>
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<sub>h</sub> gesetzt. Die Abweichung muss länger andauern als die Zeit im Objekt 203F<sub>h</sub>.

Wird der Wert des 60F8<sub>h</sub> auf "7FFFFFFF"<sub>h</sub> gesetzt, wird die Schlupffehler-Überwachung abgeschaltet.

Im Objekt 3700<sub>h</sub> kann eine Reaktion auf den Schlupffehler gesetzt werden. Wenn eine Reaktion definiert ist, wird auch ein Fehler im Objekt 1003<sub>h</sub> eingetragen.

## 60FAh Control Effort

### Funktion

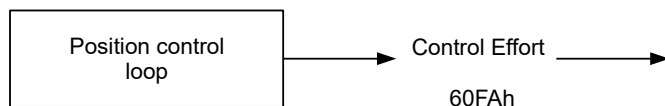
Dieses Objekt beinhaltet die Korrekturgeschwindigkeit (Stellgröße) in benutzerdefinierten Einheiten, die vom Positionsregler dem Geschwindigkeitsregler zugeführt wird.

### Objektbeschreibung

Index	60FA <sub>h</sub>
Objektname	Control Effort
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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.



## 60FC<sub>h</sub> Position Demand Internal Value

### Funktion

Zeigt den aktuellen Vorgabewert für den Positionsregler in Encoder-Inkrementen an (siehe Reglerstruktur).

### Objektbeschreibung

Index	60FC <sub>h</sub>
Objektname	Position Demand Internal Value
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Firmware Version	FIR-v1738-B501312
Änderungshistorie	

## 60FD<sub>h</sub> Digital Inputs

### Funktion

Mit diesem Objekt können die Digitalen Eingänge des Motors gelesen werden.

### Objektbeschreibung

Index	60FD <sub>h</sub>
Objektname	Digital Inputs
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	

Vorgabewert                    00000000<sub>h</sub>  
 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 <sub>h</sub>
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

---

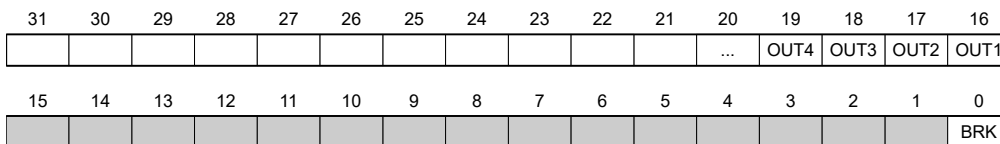
Subindex	00 <sub>h</sub>
Name	Number Of Entries
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein



Zulässige Werte	
Vorgabewert	01 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub>
Name	Physical Outputs
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>

### Beschreibung

Zum Schreiben der Ausgänge müssen noch die Einträge in Objekt 3250<sub>h</sub>, Subindex 02<sub>h</sub> bis 05<sub>h</sub> berücksichtigt werden.



#### BRK (Brake)

Bit für den Bremsenausgang (falls der Controller diese Funktion unterstützt):

Wert "1" heißt, dass die Bremse aktiviert wird (kein Strom kann zwischen den beiden Pins des Bremsen-Anschlusses fließen, die Bremse ist geschlossen).

#### OUT n (Output No n)

Bit für den jeweiligen digitalen Ausgang, die genaue Zahl der Digitalausgänge ist abhängig von der Steuerung.

### 60FFh Target Velocity

#### Funktion

In dieses Objekt wird die Zielgeschwindigkeit für den Profile Velocity und Cyclic Synchronous Velocity Mode in benutzerdefinierten Einheiten eingetragen.

#### Objektbeschreibung

Index	60FF <sub>h</sub>
Objektname	Target Velocity
Object Code	VARIABLE
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
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<sub>h</sub>.

### Objektbeschreibung

Index	6502 <sub>h</sub>
Objektname	Supported Drive Modes
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	000003EF <sub>h</sub>
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 <sub>h</sub>
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 <sub>h</sub>
Objektname	Http Drive Catalogue Address
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1426
Änderungshistorie	

---

## 11 Copyrights

### 11.1 Einführung

In der Nanotec Software sind Komponenten aus Produkten externer Software-Hersteller integriert. In diesem Kapitel finden Sie die Copyright-Informationen zu den verwendeten externen Software-Quellen.

### 11.2 AES

FIPS-197 compliant AES implementation

Based on XySSL: Copyright (C) 2006-2008 Christophe Devine

Copyright (C) 2009 Paul Bakker <polarssl\_maintainer at polarssl dot org>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution; or, the application vendor's website must provide a copy of this notice.
- Neither the names of PolarSSL or XySSL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The AES block cipher was designed by Vincent Rijmen and Joan Daemen.

<http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

### 11.3 MD5

MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

## 11.4 uIP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 11.5 DHCP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 11.6 CMSIS DSP Software Library

Copyright (C) 2010 ARM Limited. All rights reserved.

## 11.7 FatFs

FatFs - FAT file system module include file R0.08 (C)ChaN, 2010

FatFs module is a generic FAT file system module for small embedded systems.

This is a free software that opened for education, research and commercial developments under license policy of following 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.

## 11.8 Protothreads

Protothread class and macros for lightweight, stackless threads in C++.

This was "ported" to C++ from Adam Dunkels' protothreads C library at: <http://www.sics.se/~adam/pt/>

Originally ported for use by Hamilton Jet ([www.hamiltonjet.co.nz](http://www.hamiltonjet.co.nz)) by Ben Hoyt, but stripped down for public release. See his blog entry about it for more information: <http://blog.micropledge.com/2008/07/protothreads/>

Original BSD-style license

Copyright (c) 2004-2005, Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the Institute and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Institute or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

## 11.9 lwIP

Copyright (c) 2001-2004 Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO

EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This file is part of the lwIP TCP/IP stack.

Author: Adam Dunkels <adam@sics.se>

## 11.10 littlefs

```
/*
 * The little filesystem
 *
 * Copyright (c) 2017, Arm Limited. All rights reserved.
 * SPDX-License-Identifier: BSD-3-Clause
 */
```

Copyright (c) 2017, Arm Limited. All rights reserved.

- Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/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.