

# Technisches Handbuch C5-E

Feldbus: Modbus TCP, USB

**Zur Benutzung mit folgenden Varianten:**

C5-E-1-81, C5-E-2-81



Abbildung ähnlich / similar image

# Inhalt

<b>1</b>	<b>Einleitung.....</b>	<b>7</b>
1.1	Versionshinweise.....	7
1.2	Urheberrecht, Kennzeichnung und Kontakt.....	7
1.3	Bestimmungsgemäßer Gebrauch.....	7
1.4	Gewährleistung und Haftungsausschluss.....	8
1.5	Fachkräfte.....	8
1.6	EU-Richtlinien zur Produktsicherheit.....	8
1.7	Mitgeltende Vorschriften.....	8
1.8	Verwendete Symbole.....	8
1.9	Hervorhebungen im Text.....	9
1.10	Zahlenwerte.....	9
1.11	Bits.....	9
1.12	Zählrichtung (Pfeile).....	9
<b>2</b>	<b>Sicherheits- und Warnhinweise.....</b>	<b>11</b>
<b>3</b>	<b>Technische Daten und Anschlussbelegung.....</b>	<b>12</b>
3.1	Umgebungsbedingungen.....	12
3.2	Maßzeichnungen.....	13
3.3	Elektrische Eigenschaften und technische Daten.....	14
3.4	Übertemperaturschutz.....	15
3.5	LED-Signalisierung.....	16
3.6	Anschlussbelegung.....	17
<b>4</b>	<b>Inbetriebnahme.....</b>	<b>23</b>
4.1	Konfiguration über USB.....	23
4.2	Konfiguration der Ethernet-Schnittstelle.....	27
4.3	Konfiguration über Modbus TCP.....	30
4.4	Motordaten einstellen.....	31
4.5	Motor anschließen.....	31
4.6	Auto-Setup.....	32
<b>5</b>	<b>Generelle Konzepte.....</b>	<b>35</b>
5.1	Betriebsarten.....	35
5.2	CiA 402 Power State Machine.....	39
5.3	Benutzerdefinierte Einheiten.....	44
5.4	Begrenzung des Bewegungsbereichs.....	49
5.5	Zykluszeiten.....	50
<b>6</b>	<b>Betriebsmodi.....</b>	<b>51</b>
6.1	Profile Position.....	51
6.2	Velocity.....	59
6.3	Profile Velocity.....	60
6.4	Profile Torque.....	63
6.5	Homing.....	65
6.6	Interpolated Position Mode.....	73
6.7	Cyclic Synchronous Position.....	74

6.8 Cyclic Synchronous Velocity.....	76
6.9 Cyclic Synchronous Torque.....	77
6.10 Takt-Richtungs-Modus.....	78
6.11 Auto-Setup.....	80
<b>7 Spezielle Funktionen.....</b>	<b>81</b>
7.1 Digitale Ein- und Ausgänge.....	81
7.2 Automatische Bremsensteuerung.....	90
7.3 I <sup>2</sup> t Motor-Überlastungsschutz.....	92
7.4 Objekte speichern.....	94
<b>8 Modbus TCP.....</b>	<b>100</b>
8.1 Allgemeines.....	100
8.2 MBAP Header.....	100
8.3 Funktionscodes.....	101
8.4 Funktionscode-Beschreibungen.....	101
8.5 Prozessdatenobjekte (PDO).....	120
8.6 NanoJ-Objekte.....	122
<b>9 Programmierung mit NanoJ.....</b>	<b>123</b>
9.1 NanoJ-Programm.....	123
9.2 Mapping im NanoJ-Programm.....	127
9.3 Systemcalls im NanoJ-Programm.....	128
<b>10 Objektverzeichnis Beschreibung.....</b>	<b>130</b>
10.1 Übersicht.....	130
10.2 Aufbau der Objektbeschreibung.....	130
10.3 Objektbeschreibung.....	130
10.4 Wertebeschreibung.....	132
10.5 Beschreibung.....	133
1000h Device Type.....	133
1001h Error Register.....	134
1003h Pre-defined Error Field.....	135
1008h Manufacturer Device Name.....	139
1009h Manufacturer Hardware Version.....	139
100Ah Manufacturer Software Version.....	140
1010h Store Parameters.....	140
1011h Restore Default Parameters.....	144
1018h Identity Object.....	147
1020h Verify Configuration.....	149
1F50h Program Data.....	150
1F51h Program Control.....	151
1F57h Program Status.....	152
200Fh IEEE 802 MAC Address.....	153
2010h IP-Configuration.....	154
2011h Static-IPv4-Address.....	155
2012h Static-IPv4-Subnet-Mask.....	156
2013h Static-IPv4-Gateway-Address.....	157
2014h Current-IPv4-Address.....	158
2015h Current-IPv4-Subnet-Mask.....	159
2016h Current-IPv4-Gateway-Address.....	160
2028h MODBUS Slave Address.....	161
202Ah MODBUS RTU Baudrate.....	161
202Ch MODBUS RTU Stop Bits.....	162
202Dh MODBUS RTU Parity.....	162

2030h Pole Pair Count.....	163
2031h Max Motor Current.....	163
2034h Upper Voltage Warning Level.....	164
2035h Lower Voltage Warning Level.....	165
2036h Open Loop Current Reduction Idle Time.....	165
2037h Open Loop Current Reduction Value/factor.....	166
2038h Brake Controller Timing.....	166
2039h Motor Currents.....	168
203Ah Homing On Block Configuration.....	170
203Bh I2t Parameters.....	171
203Dh Torque Window.....	174
203Eh Torque Window Time Out.....	174
203Fh Max Slippage Time Out.....	175
2057h Clock Direction Multiplier.....	175
2058h Clock Direction Divider.....	176
2059h Encoder Configuration.....	176
205Ah Absolute Sensor Boot Value (in User Units).....	177
205Bh Clock Direction Or Clockwise/Counter Clockwise Mode.....	178
2084h Bootup Delay.....	178
2101h Fieldbus Module Availability.....	179
2102h Fieldbus Module Control.....	180
2103h Fieldbus Module Status.....	181
2290h PDI Control.....	183
2291h PDI Input.....	183
2292h PDI Output.....	185
2300h NanoJ Control.....	186
2301h NanoJ Status.....	187
2302h NanoJ Error Code.....	187
230Fh Uptime Seconds.....	189
2310h NanoJ Input Data Selection.....	189
2320h NanoJ Output Data Selection.....	190
2330h NanoJ In/output Data Selection.....	192
2400h NanoJ Inputs.....	193
2410h NanoJ Init Parameters.....	194
2500h NanoJ Outputs.....	195
2600h NanoJ Debug Output.....	196
2701h Customer Storage Area.....	197
2800h Bootloader And Reboot Settings.....	198
3202h Motor Drive Submode Select.....	199
3203h Feedback Selection.....	200
3204h Feedback Mapping.....	202
3210h Motor Drive Parameter Set.....	203
3212h Motor Drive Flags.....	207
3220h Analog Inputs.....	209
3221h Analogue Inputs Control.....	210
3225h Analogue Inputs Switches.....	211
3240h Digital Inputs Control.....	212
3242h Digital Input Routing.....	214
3243h Digital Input Homing Capture.....	217
3250h Digital Outputs Control.....	218
3252h Digital Output Routing.....	221
3320h Read Analogue Input.....	222
3321h Analogue Input Offset.....	224
3322h Analogue Input Pre-scaling.....	225
3390h Feedback Hall.....	226
33A0h Feedback Incremental A/B/I 1.....	229
3502h MODBUS Rx PDO Mapping.....	230
3602h MODBUS Tx PDO Mapping.....	234
3700h Deviation Error Option Code.....	238

3701h Limit Switch Error Option Code.....	239
4012h HW Information.....	240
4013h HW Configuration.....	241
4014h Operating Conditions.....	242
4040h Drive Serial Number.....	244
4041h Device Id.....	244
603Fh Error Code.....	245
6040h Controlword.....	245
6041h Statusword.....	246
6042h VI Target Velocity.....	248
6043h VI Velocity Demand.....	248
6044h VI Velocity Actual Value.....	249
6046h VI Velocity Min Max Amount.....	249
6048h VI Velocity Acceleration.....	250
6049h VI Velocity Deceleration.....	251
604Ah VI Velocity Quick Stop.....	252
604Ch VI Dimension Factor.....	254
605Ah Quick Stop Option Code.....	255
605Bh Shutdown Option Code.....	256
605Ch Disable Option Code.....	256
605Dh Halt Option Code.....	257
605Eh Fault Option Code.....	258
6060h Modes Of Operation.....	258
6061h Modes Of Operation Display.....	259
6062h Position Demand Value.....	260
6063h Position Actual Internal Value.....	260
6064h Position Actual Value.....	261
6065h Following Error Window.....	261
6066h Following Error Time Out.....	262
6067h Position Window.....	262
6068h Position Window Time.....	263
606Bh Velocity Demand Value.....	263
606Ch Velocity Actual Value.....	264
606Dh Velocity Window.....	264
606Eh Velocity Window Time.....	265
6071h Target Torque.....	265
6072h Max Torque.....	266
6073h Max Current.....	267
6074h Torque Demand.....	267
6075h Motor Rated Current.....	268
6077h Torque Actual Value.....	268
607Ah Target Position.....	269
607Bh Position Range Limit.....	269
607Ch Home Offset.....	270
607Dh Software Position Limit.....	271
607Eh Polarity.....	272
607Fh Max Profile Velocity.....	273
6080h Max Motor Speed.....	273
6081h Profile Velocity.....	274
6082h End Velocity.....	274
6083h Profile Acceleration.....	275
6084h Profile Deceleration.....	275
6085h Quick Stop Deceleration.....	276
6086h Motion Profile Type.....	276
6087h Torque Slope.....	277
608Fh Position Encoder Resolution.....	277
6090h Velocity Encoder Resolution.....	278
6091h Gear Ratio.....	280
6092h Feed Constant.....	281

6096h Velocity Factor.....	282
6097h Acceleration Factor.....	283
6098h Homing Method.....	284
6099h Homing Speed.....	284
609Ah Homing Acceleration.....	286
60A2h Jerk Factor.....	286
60A4h Profile Jerk.....	287
60A8h SI Unit Position.....	289
60A9h SI Unit Velocity.....	290
60B0h Position Offset.....	290
60B1h Velocity Offset.....	291
60B2h Torque Offset.....	291
60C1h Interpolation Data Record.....	292
60C2h Interpolation Time Period.....	293
60C4h Interpolation Data Configuration.....	294
60C5h Max Acceleration.....	296
60C6h Max Deceleration.....	296
60E4h Additional Position Actual Value.....	297
60E5h Additional Velocity Actual Value.....	298
60E6h Additional Position Encoder Resolution - Encoder Increments.....	299
60E8h Additional Gear Ratio - Motor Shaft Revolutions.....	300
60E9h Additional Feed Constant - Feed.....	301
60EBh Additional Position Encoder Resolution - Motor Revolutions.....	303
60EDh Additional Gear Ratio - Driving Shaft Revolutions.....	304
60EEh Additional Feed Constant - Driving Shaft Revolutions.....	305
60F2h Positioning Option Code.....	306
60F4h Following Error Actual Value.....	308
60F8h Max Slippage.....	308
60FAh Control Effort.....	309
60FCh Position Demand Internal Value.....	309
60FDh Digital Inputs.....	310
60FEh Digital Outputs.....	310
60FFh Target Velocity.....	312
6502h Supported Drive Modes.....	312
6503h Drive Catalogue Number.....	313
6505h Http Drive Catalogue Address.....	314

## **11 Copyrights..... 315**

11.1 Einführung.....	315
11.2 AES.....	315
11.3 MD5.....	315
11.4 uIP.....	316
11.5 DHCP.....	316
11.6 CMSIS DSP Software Library.....	316
11.7 FatFs.....	316
11.8 Protothreads.....	317
11.9 lwIP.....	317

## 1 Einleitung

Die *C5-E* ist eine Steuerung für den *Open Loop*- oder *Closed Loop*-Betrieb von Schrittmotoren und den *Closed Loop*-Betrieb von BLDC- Motoren.

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

Weitere Informationen zum Gerät finden Sie auf der Nanotec Homepage [www.nanotec.de](http://www.nanotec.de).

### 1.1 Versionshinweise

Version Handbuch	Datum	Änderungen	Version Firmware	Version Hardware
1.0.0	06/2018	erste Veröffentlichung	FIR-v1748	W002
1.1.0	11/2018	Änderungen in <b>Begrenzung des Bewegungsbereichs</b> und <b>Motordaten einstellen</b>	FIR-v1825	W002

### 1.2 Urheberrecht, Kennzeichnung und Kontakt

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



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

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

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

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

### 1.3 Bestimmungsgemäßer Gebrauch

Die *C5-E Steuerung* dient der Steuerung von Schritt- und BLDC-Motoren und ist für den Einsatz unter den freigegebenen **Umgebungsbedingungen** konzipiert.

Ein anderer Gebrauch gilt als nicht bestimmungsgemäß.



#### Hinweis

Änderungen oder Umbauten der Steuerung sind nicht zulässig.

## 1.4 Gewährleistung und Haftungsausschluss

Nanotec produziert Komponententeile, die ihren Einsatz in vielfältigen Industrieanwendungen finden. Die Auswahl und Anwendung von Nanotec-Produkten liegt im Verantwortungsbereich des Anlagenkonstruktors bzw. Endnutzers. Nanotec übernimmt keinerlei Verantwortung für die Integration der Produkte in das Endsystem.

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

Es gelten unsere Allgemeinen Geschäftsbedingungen: [de.nanotec.com/service/agb/](https://de.nanotec.com/service/agb/).

## 1.5 Fachkräfte

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

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

## 1.6 EU-Richtlinien zur Produktsicherheit

Folgende EU-Richtlinien wurden beachtet:

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

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





### Tip

Zeigt einen Tipp zur Anwendung oder Aufgabe.

## 1.9 Hervorhebungen im Text

Im Dokument gelten folgende Konventionen:

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

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

Ein *kursiv* hervorgehobener Text markiert benannte Objekte:

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

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

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

Ein Text in "Anführungszeichen" markiert Benutzereingaben:

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

## 1.10 Zahlenwerte

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

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

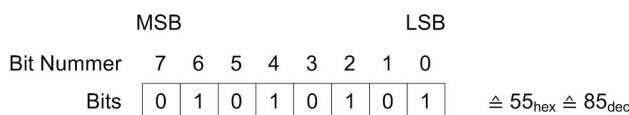
<Index>:<Subindex>

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

Beispiel: Der Subindex 5 des Objekts 1003<sub>h</sub> wird adressiert mit 1003<sub>h</sub>:05<sub>h</sub>, der Subindex 00 des Objekts 6040<sub>h</sub> mit 6040<sub>h</sub>.

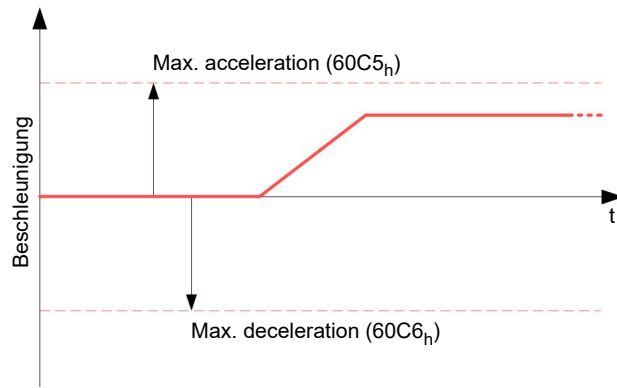
## 1.11 Bits

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



## 1.12 Zählrichtung (Pfeile)

In Abbildungen gilt die Zählrichtung immer in Richtung eines Pfeiles. Die in der nachfolgenden Abbildung beispielhaft dargestellten Objekte 60C5<sub>h</sub> und 60C6<sub>h</sub> werden beide positiv angegeben.



## 2 Sicherheits- und Warnhinweise



### Hinweis

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



### Hinweis

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



### Hinweis

- Ein Verpolungsschutz ist nicht gegeben.
- Bei Verpolung entsteht ein Kurzschluss zwischen Versorgungsspannung und GND (Masse) über die Leistungsdiode.
- Installieren Sie eine Leitungsschutzeinrichtung (Sicherung) in der Zuleitung.



### Hinweis

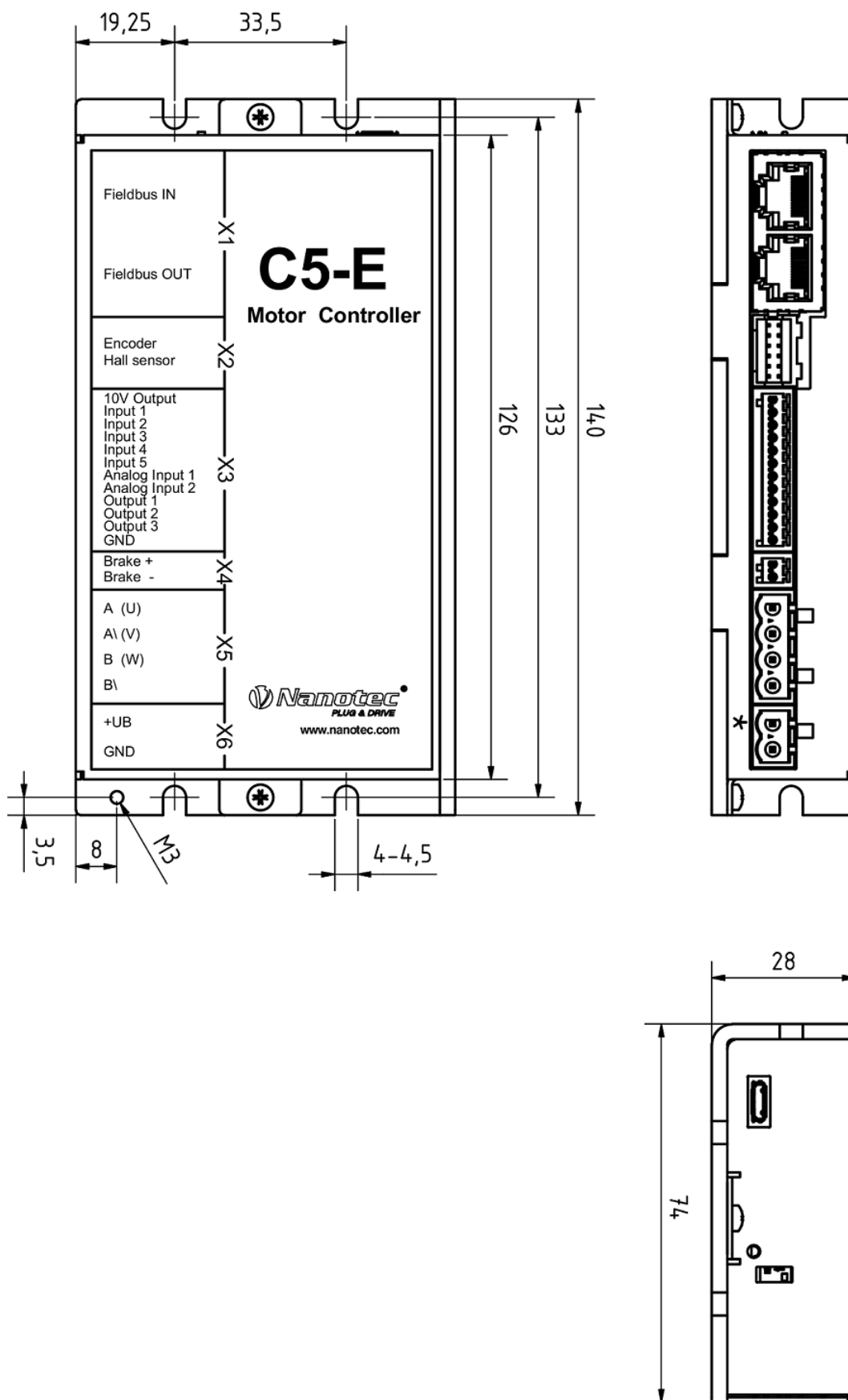
- Das Gerät enthält Bauteile, die empfindlich gegen elektrostatische Entladung sind.
- Unsachgemäßer Umgang kann das Gerät beschädigen.
- Beachten Sie die Grundprinzipien des ESD-Schutzes beim Umgang mit dem Gerät.

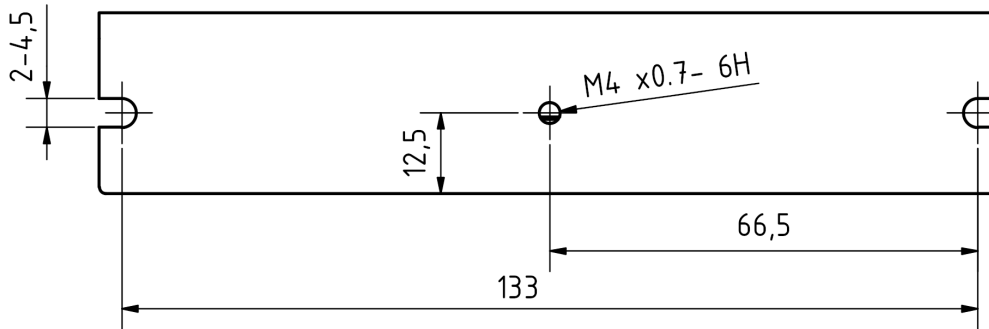
## 3 Technische Daten und Anschlussbelegung

### 3.1 Umgebungsbedingungen

Umgebungsbedingung	Wert
Schutzklasse	IP20
Umgebungstemperatur (Betrieb)	-10 ... +40°C
Luftfeuchtigkeit (nicht kondensierend)	0 ... 95 %
Aufstellhöhe über <i>NN</i> (ohne Leistungsbeschränkung)	1500 m
Umgebungstemperatur (Lagerung)	-25 ... +85°C

### 3.2 Maßzeichnungen





### 3.3 Elektrische Eigenschaften und technische Daten

Eigenschaft	Beschreibung/Wert
Betriebsspannung	12 V DC bis 48 V DC +/-5%
Nennstrom	C5-E-1-81 ( <i>low current</i> ): 6 A <sub>eff</sub> C5-E-2-81 ( <i>high current</i> ): 10 A <sub>eff</sub>
Spitzenstrom	C5-E-1-81 ( <i>low current</i> ): 6 A <sub>eff</sub> C5-E-2-81 ( <i>high current</i> ): 30 A <sub>eff</sub> für 5 Sekunden
Kommutierung	Schrittmotor Open Loop, Schrittmotor Closed Loop mit Encoder, BLDC-Motor Closed Loop mit Hall Sensor und BLDC-Motor Closed Loop mit Encoder
Betriebsmodi	<i>Profile Position Mode, Profile Velocity Mode, Profile Torque Mode, Velocity Mode, Homing Mode, Interpolated Position Mode, Cyclic Sync Position Mode, Cyclic Sync Velocity Mode, Cyclic Synchronous Torque Mode, Takt-Richtung-Modus</i>
Sollwertvorgabe/ Programmierung	<i>Takt-Richtung, Analog, NanoJ-Programm</i>
Schnittstellen	USB, Modbus TCP
Eingänge	<ul style="list-style-type: none"> <li>• 5 Eingänge 24 V (Eingang 1 bis 5) einzeln schaltbar zwischen 5 und 24 V, Werkseinstellung 5 V</li> <li>• 1 analoger Eingang, 10 Bit, schaltbar 0-10 V oder 0-20 mA, Werkseinstellung 0-10 V</li> <li>• 1 analoger Eingang, 10 Bit, 0-10 V</li> </ul>
Ausgänge	3 Ausgänge, (Open Drain, 0 schaltend, max. 24 V und 100 mA)
Schutzschaltung	<p>Über- und Unterpannungsschutz</p> <p>Übertemperaturschutz (&gt; 75° Celsius auf der Leistungsplatine)</p> <p>Verpolungsschutz: bei Verpolung Kurzschluss zwischen Versorgungsspannung und GND über Leistungsdiode, daher ist eine Leitungsschutzeinrichtung (Sicherung) in Zuleitung nötig. Die Werte der Sicherung ist abhängig von der Applikation und muss</p> <ul style="list-style-type: none"> <li>• größer als die maximale Stromaufnahme der Steuerung</li> <li>• kleiner als der maximale Strom der Spannungsversorgung ausgelegt werden.</li> </ul>

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

### 3.4 Übertemperaturschutz

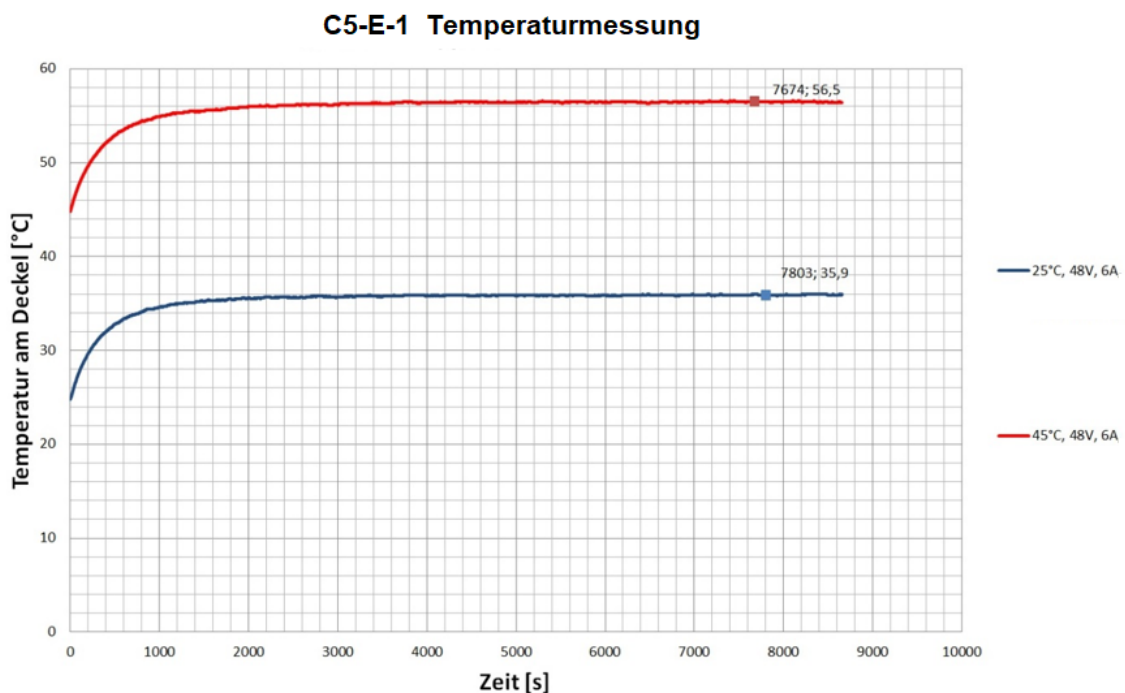
Ab einer Temperatur von ca. 75 °C auf der Leistungsplatine (entspricht 65 - 72 °C außen am Deckel) wird das Leistungsteil der Steuerung abgeschaltet und das Fehlerbit gesetzt (siehe Objekt **1001<sub>h</sub>** und **1003<sub>h</sub>**). Nach Abkühlung und dem Bestätigen des Fehlers (siehe **Tabelle für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

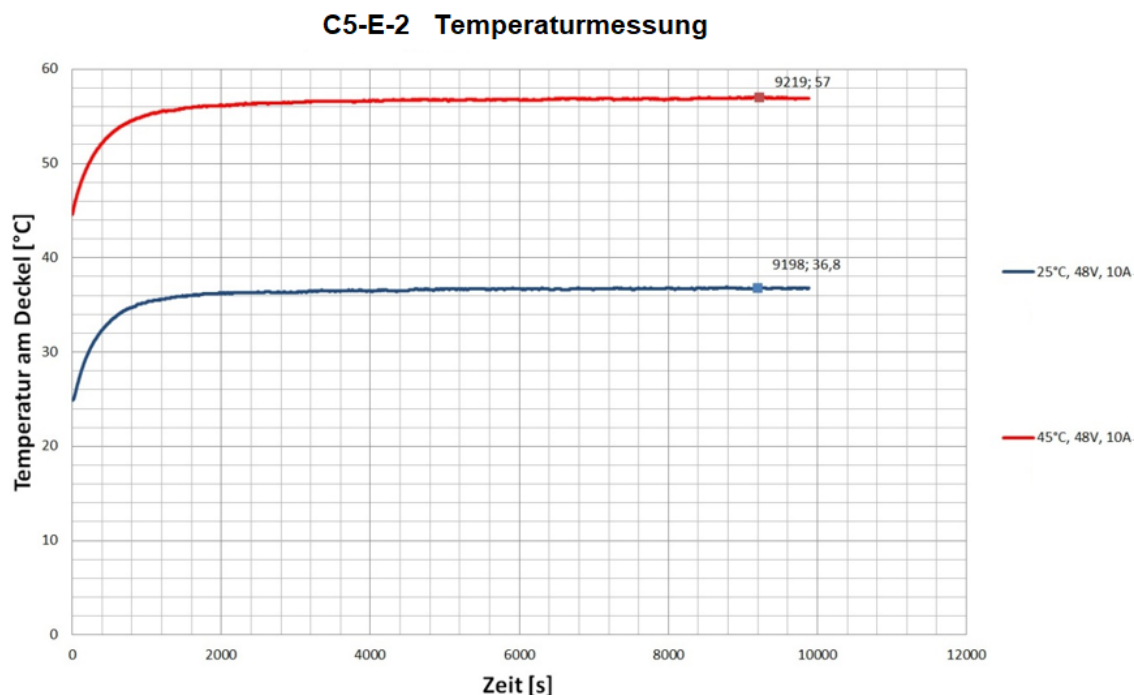
Die folgenden Ergebnisse von Temperaturtests geben einen Hinweis auf das Temperaturverhalten dieser Steuerung.

Es wurden Temperaturtests unter folgenden Bedingungen durchgeführt:

- Betriebsspannung: 48 V DC
- Motorstrom: 6 A (C5-E-1 *low current*)/10 A (C5-E-2 *high current*) effektiv
- Operationsmodus: Drehzahlmodus Vollschrift, 30 U/min
- Umgebungstemperatur: 25 °C / 45 °C
- Aufstellhöhe: 500 m über NN

Die folgende Grafik zeigt die Ergebnisse der Temperaturtests:





#### Zusammenfassung:

Bei 25 °C (+48 V, 6/10 A effektiv, Drehzahlmodus 30 U/min) ist die Steuerung länger als 3 Stunden in Betrieb gewesen ohne Abschaltung. Die Temperatur am Deckel war stabil bei ca. 37 °C.

Bei 45°C (+48V, 6/10 A effektiv, Drehzahlmodus 30 U/min) ist die Steuerung länger als 3 Stunden in Betrieb gewesen ohne Abschaltung. Die Temperatur am Deckel war stabil bei ca. 57 °C.

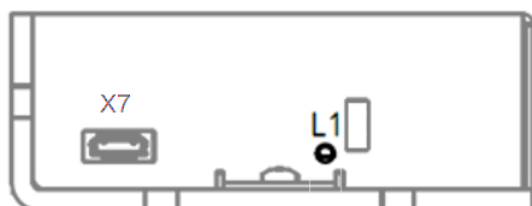


#### Hinweis

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

## 3.5 LED-Signalisierung

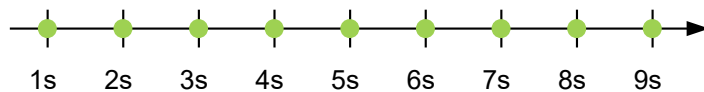
### 3.5.1 Betriebs-LED





### Normaler Betrieb

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



### Fehlerfall

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



Folgende Tabelle zeigt die Bedeutung der Fehlernummern.

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



#### Hinweis

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

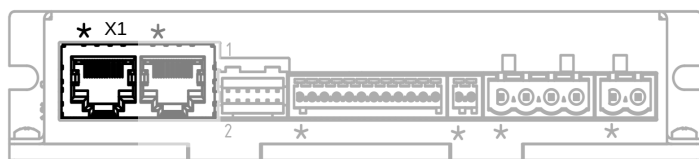
## 3.6 Anschlussbelegung

### 3.6.1 Übersicht

Stecker	Funktion
X1	Modbus TCP
X2	Encoder und Hall-Sensor Anschluss
X3	Digitale/analoge Ein- und Ausgänge
X4	Bremsen-Anschluss
X5	Motoranschluss
X6	Spannungsversorgung
X7	Micro USB-Anschluss
L1	Betriebs-LED

### 3.6.2 Stecker X1 - Modbus TCP

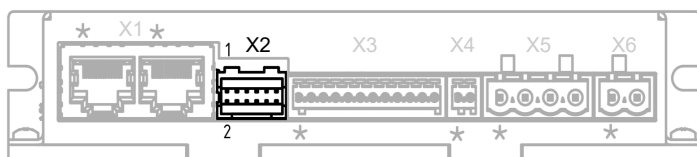
Pin 1 ist mit einem Stern "\*" markiert.



PIN	Funktion	Bemerkung
1	Tx+	
2	Tx-	
3	Rx+	
4	n.c.	
5	n.c.	
6	Rx-	
7	n.c.	
8	n.c.	

### 3.6.3 Stecker X2 - Encoder/Hall Sensor

Pin 1 und Pin 2 sind im Bild markiert.

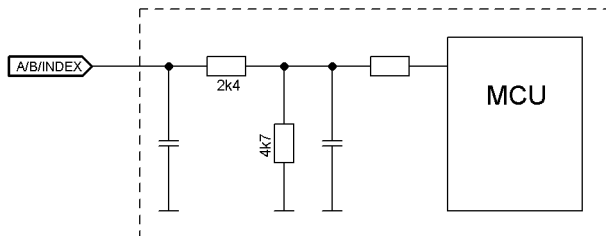


PIN	Funktion	Bemerkung
1	GND	
2	Vcc	5 V DC, Ausgangs- und Versorgungsspannung für Encoder / Hall Sensor; max. 200 mA
3	A	5 V Signal, max. 1 MHz
4	B	5 V Signal, max. 1 MHz
5	A\	5 V Signal, max. 1 MHz
6	B\	5 V Signal, max. 1 MHz
7	I	5 V Signal, max. 1 MHz
8	I\	5 V Signal, max. 1 MHz
9	Hall 1	5 V Signal
10	Hall 2	5 V Signal
11	Hall 3	5 V Signal
12	Shielding	Schirmung

Es gelten folgende Schaltschwellen für die Encoder-Eingänge:

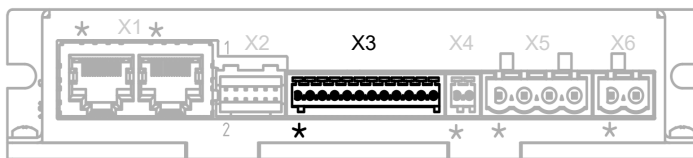
Typ	Schaltschwellen	
	Ein	Aus
Single	> 3,8 V	< 0,26 V
Differenz	> 3,8 V	< 0,26 V

Die interne Beschaltung der Encoder-Eingänge ist nachfolgend dargestellt.



### 3.6.4 Stecker X3 - Ein- und Ausgänge

Pin 1 ist mit einem Stern "\*" markiert.



PIN	Funktion	Bemerkung
1	+10 V DC	Ausgangsspannung, max. 200 mA
2	Digitaler Eingang 1	5 V / 24 V Signal, umschaltbar per Software mit Objekt <b>3240<sub>h</sub></b> , max. 1 MHz; Takteingang in Takt-Richtungs-Modus
3	Digitaler Eingang 2	5 V / 24 V Signal, umschaltbar per Software mit Objekt <b>3240<sub>h</sub></b> , max. 1 MHz; Richtungseingang in Takt-Richtungs-Modus
4	Digitaler Eingang 3	5 V / 24 V Signal, umschaltbar per Software mit Objekt <b>3240<sub>h</sub></b>
5	Digitaler Eingang 4	5 V / 24 V Signal, umschaltbar per Software mit Objekt <b>3240<sub>h</sub></b>
6	Digitaler Eingang 5	5 V / 24 V Signal, umschaltbar per Software mit Objekt <b>3240<sub>h</sub></b>
7	Analoger Eingang 1	10 Bit, 0-10 V oder 0-20 mA, umschaltbar per Software mit Objekt <b>3221<sub>h</sub></b>
8	Analoger Eingang 2	10 Bit, 0-10 V, nicht umschaltbar per Software
9	Digitaler Ausgang 1	Open Drain, maximal 24 V / 100 mA
10	Digitaler Ausgang 2	Open Drain, maximal 24 V / 100 mA
11	Digitaler Ausgang 3	Open Drain, maximal 24 V / 100 mA
12	GND	

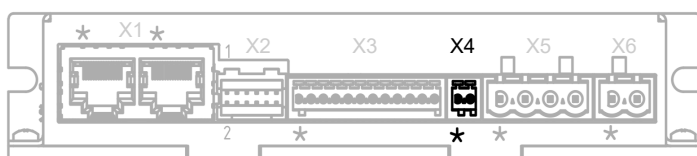
Für Eingang 1 bis 5 gelten folgende Schaltschwellen:

Max. Spannung	Schaltschwellen	
	Ein	Aus
5 V	> 3,8 V	< 0,26 V
24 V	> 14,42 V	< 4,16 V

Anschlussdaten	min	max
Leiterquerschnitt starr min	0,14 mm <sup>2</sup>	0,5 mm <sup>2</sup>
Leiterquerschnitt flexibel min.	0,14 mm <sup>2</sup>	0,5 mm <sup>2</sup>
Leiterquerschnitt flexibel m. Aderendhülse ohne Kunststoffhülse min	0,25 mm <sup>2</sup>	0,5 mm <sup>2</sup>
Leiterquerschnitt AWG min	26	20
AWG nach UL/CUL min	28	20

### 3.6.5 Stecker X4 - Bremsen-Anschluss

Pin 1 ist mit einem Stern "\*" markiert.

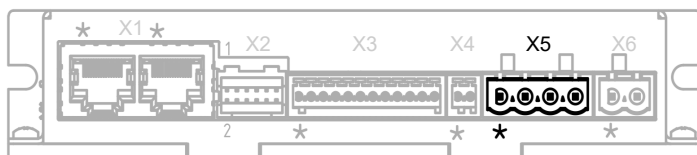


PIN	Funktion	Bemerkung
1	Bremse +	Intern verbunden mit +UB
2	Bremse -	PWM-gesteuerter Open-Drain Ausgang, max 1,5 A

Anschlussdaten	min	max
Leiterquerschnitt starr min	0,14 mm <sup>2</sup>	0,5 mm <sup>2</sup>
Leiterquerschnitt flexibel min.	0,14 mm <sup>2</sup>	0,5 mm <sup>2</sup>
Leiterquerschnitt flexibel m. Aderendhülse ohne Kunststoffhülse min	0,25 mm <sup>2</sup>	0,5 mm <sup>2</sup>
Leiterquerschnitt AWG min	26	20
AWG nach UL/CUL min	28	20

### 3.6.6 Stecker X5 -Motoranschluss

Pin 1 ist mit einem Stern "\*" markiert.



PIN	Funktion (Stepper)	Funktion (BLDC)
1	A	U
2	A\	V
3	B	W
4	B\	nicht benutzt

Anschlussdaten	min	max
Leiterquerschnitt starr min	0,2 mm <sup>2</sup>	1,5 mm <sup>2</sup>
Leiterquerschnitt flexibel min.	0,2 mm <sup>2</sup>	1,5 mm <sup>2</sup>
Leiterquerschnitt flexibel m. Aderendhülse ohne Kunststoffhülse min	0,25 mm <sup>2</sup>	1,5 mm <sup>2</sup>
Leiterquerschnitt flexibel m. Aderendhülse m. Kunststoffhülse min	0,25 mm <sup>2</sup>	0,75 mm <sup>2</sup>
Leiterquerschnitt AWG min	24	16
AWG nach UL/CUL min	24	16

### 3.6.7 Stecker X6 - Spannungsversorgung

#### Spannungsquelle

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

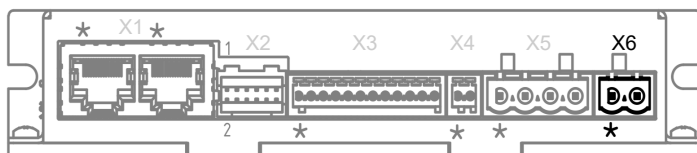


#### Hinweis

- 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.

#### Anschlüsse

Pin 1 ist mit einem Stern "\*" markiert.



PIN	Funktion	Bemerkung
1	+UB	12-48 V DC, ±5%
2	GND	

Anschlussdaten	min	max
Leiterquerschnitt starr min	0,2 mm <sup>2</sup>	1,5 mm <sup>2</sup>
Leiterquerschnitt flexibel min.	0,2 mm <sup>2</sup>	1,5 mm <sup>2</sup>
Leiterquerschnitt flexibel m. Aderendhülse ohne Kunststoffhülse min	0,25 mm <sup>2</sup>	1,5 mm <sup>2</sup>
Leiterquerschnitt flexibel m. Aderendhülse m. Kunststoffhülse min	0,25 mm <sup>2</sup>	0,75 mm <sup>2</sup>
Leiterquerschnitt AWG min	24	16
AWG nach UL/CUL min	24	16

### Zulässige Betriebsspannung

Die maximale Betriebsspannung beträgt 51,5 V DC. Steigt die Eingangsspannung der Steuerung über diesen Schwellwert, wird der Motor abgeschaltet und ein Fehler ausgelöst. Ab 50,5 V wird die integrierte Ballast-Schaltung (5W Leistung) aktiviert.

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

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

### 3.6.8 Micro USB (Stecker X7)

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



## 4 Inbetriebnahme

In diesem Kapitel wird beschrieben, wie Sie die Kommunikation zur Steuerung aufbauen und die notwendigen Parameter einstellen, damit der Motor betriebsbereit ist. Sie können die Steuerung über USB oder über Modbus TCP konfigurieren.

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

Beachten Sie folgenden Hinweis:



### Hinweis

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

## 4.1 Konfiguration über USB

### 4.1.1 Allgemeines

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

#### Konfigurationsdatei

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

#### NanoJ-Programm

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

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

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

### 4.1.2 USB-Anschluss

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

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

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



### Hinweis

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

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



### Tipp

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

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

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

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

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

## 4.1.3 Konfigurationsdatei

### Allgemeines

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



### Hinweis

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

### Lesen und Schreiben der Datei

So erhalten Sie Zugriff auf die Datei:

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



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

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



#### Tip

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

## Aufbau der Konfigurationsdatei

### Kommentare

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

### Beispiel

```
; Dies ist eine Kommentarzeile
```

### Zuweisungen



#### Hinweis

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

Werte im Objektverzeichnis lassen sich mit folgender Syntax setzen:

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

#### <Index>

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

#### <Subindex>

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

#### <Wert>

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

Sie können auch einzelne Bits setzen:

### Bit setzen

```
3202:00.3=1
```

### Bit zurücksetzen

```
3202:00.3=0
```

### Bitweise OR

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

### Bitweise AND

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

### Beispiel

Setzen des Objekts 203B<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

```
3202:00.3=1
```



#### Hinweis

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

## 4.1.4 NanoJ-Programm

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

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

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



#### Tip

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



#### Hinweis

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



#### Tip

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

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

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

Also zum Beispiel:

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

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

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

## 4.2 Konfiguration der Ethernet-Schnittstelle

### 4.2.1 Übersicht

#### Hardware-Adresse

Die Steuerung hat zunächst noch keine IP-Adresse, sondern wird über die aufgedruckte Hardware-Adresse (MAC-Adresse) angesprochen. Diese Adresse besteht aus 6 Hexadezimal-Zahlen in der Form 44-AA-E8-xx-xx-xx.

Die Hardware-Adresse ist eindeutig und unveränderbar und wird bei der Produktion vergeben. Im Regelfall wird diese nur beim Firmware-Update benötigt. Sobald sich der Bootloader beendet hat und die eigentliche Firmware in Betrieb geht, erfolgt die weitere Kommunikation über das Protokoll TCP/IP.

### IP-Adresse

Die Steuerung benötigt eine gültige IP-Adresse. Diese kann über folgende Wege bezogen werden:

- DHCP: Ein DHCP-Server vergibt die IP-Adresse an die Steuerung (Standardeinstellung).
- AutoIP: Die Steuerung ermittelt selbstständig eine geeignete IP-Adresse. Dies setzt voraus, dass sich der Kommunikationspartner im selben physikalischen Subnetz befindet und ebenfalls AutoIP verwendet.
- Statische IP-Adresse: Diese wird vom Benutzer festgelegt.

Welche Methode zum Einsatz kommt, ist von der Netzwerkumgebung abhängig und wird vom Netzwerkbetreuer festgelegt.

## 4.2.2 Einstellen der IP-Adresse

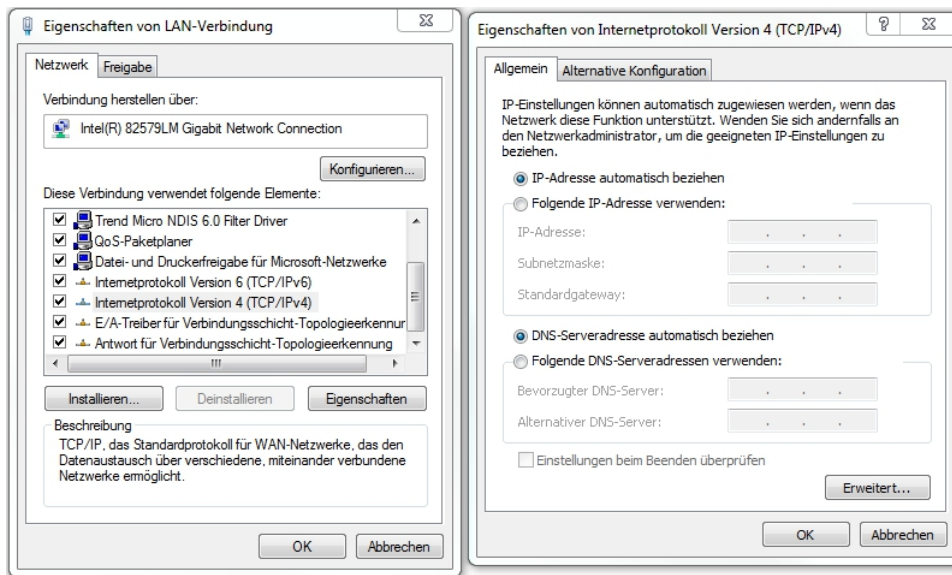
Die angeschlossenen Geräte (Steuerung und Kommunikationspartner) in einem Ethernet-Netzwerk oder bei einer Ethernet-Punkt-zu-Punkt-Verbindung benötigen jeweils eine eindeutige IP-Adresse. Diese kann entweder automatisch bezogen (DHCP) bzw. generiert (Auto-IP) oder statisch vorgegeben werden. Im weiteren Verlauf wird unter "Kommunikationspartner" ein PC oder Laptop verstanden.

Sie können die Steuerung in ein bestehendes Ethernet-Netzwerk integrieren. Dazu ist lediglich die physikalische Verbindung per Standard-Ethernetkabel herzustellen. Sofern DHCP auf der Steuerung aktiviert ist (werksseitig voreingestellt), wird die Steuerung auch automatisch im Netzwerk erkannt und kann sofort über einen im Netzwerk befindlichen PC bedient werden.

## 4.2.3 Einstellen DHCP/Auto-IP

IP-Adressen können in einem Netzwerk dynamisch von einem DHCP-Server bezogen werden oder beispielsweise bei einer PC-Direktverbindung ohne DHCP-Server automatisch durch die beiden kommunizierenden Geräte (z.B. PC und Steuerung) selbst generiert werden. In der Steuerung ist bereits werksseitig DHCP für den automatischen Bezug einer IP-Adresse von einem DHCP-Server oder der automatischen IP-Adressgenerierung voreingestellt. Es sind lediglich seitens des Kommunikationspartners (z.B. PC oder Laptop) eventuell einige Einstellungen für die Herstellung der Verbindung zur Steuerung notwendig. Einstellungen als Beispiel beim Betriebssystem Windows 7:

1. Windows-Start-Button drücken und *Systemsteuerung* auswählen.
2. *Netzwerk- und Freigabecenter* auswählen.
3. *Adaptoreinstellungen ändern* auswählen.
4. Es wird die Liste der verfügbaren Netzwerkadapter dargestellt. Am Adapter, mit welchem die Steuerung verbunden ist, die Eigenschaften öffnen (beispielsweise mit einem Klick mit der rechten Maustaste).
5. *Internetprotokoll Version 4 (TCP/IPv4)* anwählen und die Schaltfläche *Eigenschaften* drücken.
6. Option *IP-Adresse automatisch beziehen* auswählen.
7. Übernahme der Eingaben mit der Schaltfläche *OK* bestätigen.



#### 4.2.4 Einstellen einer statischen IP-Adresse

Sollen an die Steuerung und den Kommunikationspartner statische IP-Adressen vergeben werden, sind nur wenige Einstellungen seitens der Steuerung und des Kommunikationspartners durchzuführen.

Der Steuerung kann durch OD-Einträge eine statische IP-Adresse und Netzwerkmaske (jeweils IPv4) gegeben werden. Im Objektverzeichnis sind folgende Einträge maßgeblich:

Index	Beschreibung
<b>2010<sub>h</sub></b>	IP-Configuration, Bitmaske mit folgender Bedeutung: Bit 0: Eine statische IP-Adresse aus dem Objekt <b>2011<sub>h</sub></b> und die Netzwerkmaske aus dem Objekt <b>2012<sub>h</sub></b> wird genutzt.
<b>2011<sub>h</sub></b>	Statische IP-Adresse, 4 Bytes in Hex-Codierung
<b>2012<sub>h</sub></b>	Statische IP-Subnetzmaske, 4 Bytes in Hex-Codierung
<b>2013<sub>h</sub></b>	Gateway Adresse
<b>2014<sub>h</sub></b>	Aktive IP-Adresse, 4 Bytes in Hex-Codierung
<b>2015<sub>h</sub></b>	Aktive IP-Subnetzmaske, 4 Bytes in Hex-Codierung
<b>2016<sub>h</sub></b>	Momentan benutzte Gateway Adresse
<b>200F<sub>h</sub></b>	MAC-Adresse

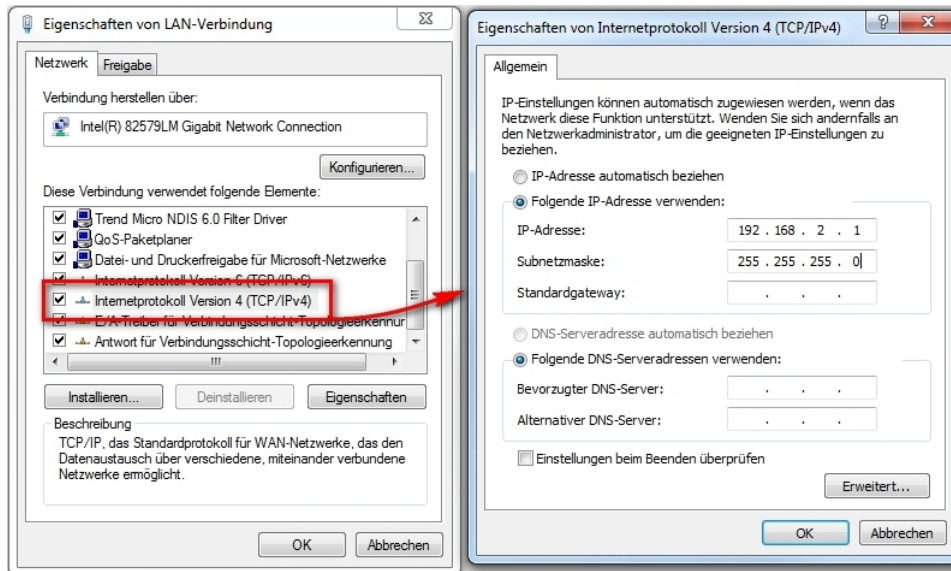
Anmerkungen:

- Wurde DHCP aktiviert und Auto-IP ist nicht aktiv, so benutzt die Steuerung die eingestellte statische IP-Adresse, falls über DHCP keine Adresse zugeteilt werden konnte (z. B. weil der DHCP-Server temporär nicht verfügbar ist).
- Wenn beide Objekte **2010<sub>h</sub>** und **2011<sub>h</sub>** auf den Wert "0" gesetzt werden, wird von einer falschen Konfiguration ausgegangen und DHCP und Auto-IP angeschaltet.
- Wenn im Objekt **2010<sub>h</sub>** Bit 0 gesetzt ist, wird die statische IP-Adresse benutzt. DHCP wird in diesem Fall nicht genutzt.
- Wenn nur DHCP angeschaltet ist und eine IP-Adressvergabe nicht funktioniert hat, wird unabhängig von Bit 0 versucht, sich mit der eingetragenen statischen IP-Adresse sich zu verbinden.

Dem Kommunikationspartner wird ebenfalls eine statische IP-Adresse gegeben. Einstellungen als Beispiel beim Betriebssystem Windows 7:

1. Windows-Start-Button drücken und *Systemsteuerung* auswählen.

2. *Netzwerk- und Freigabecenter* auswählen.
3. *Adaptoreinstellungen ändern* auswählen.
4. Es wird die Liste der verfügbaren Netzwerkadapter dargestellt. Am Adapter, mit welchem die Steuerung verbunden ist, die Eigenschaften öffnen (z.B. rechter Mausklick und *Eigenschaften* auswählen).
5. *Internetprotokoll Version 4 (TCP/IPv4)* anwählen und die Schaltfläche *Eigenschaften* drücken.
6. Option *Folgende IP-Adresse verwenden*: auswählen und im Feld *IP-Adresse* die gewünschte IP-Adresse und Netzwerkmaske eintragen.
7. Übernahme der Eingaben mit der Schaltfläche OK bestätigen.



## 4.3 Konfiguration über Modbus TCP

Diese Steuerung ist mit einer Modbus TCP-Schnittstelle ausgestattet. Die Nachrichten werden alle über TCP an den Port 502 der Steuerung geschickt, es wird nur eine Verbindung unterstützt. Eine CRC (wie es bei Modbus RTU benutzt wird) entfällt.

Die I/O-Daten mit den ggf. vorkonfigurierten Antriebsgrößen (siehe **Prozessdatenobjekte (PDO)**) können mit den Standard Modbus-Funktionscodes gesendet werden. Um aber eigene I/O-Daten zu konfigurieren, muss der Funktionscode 2Bh (CAN Encapsulation) vom Modbus-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.

Lesen Sie das Kapitel **Modbus TCP** für weitere Details.

### 4.3.1 Kommunikation aufbauen

Vor der Inbetriebnahme wird empfohlen, das Kapitel durchzulesen.

1. Schließen Sie die Versorgungsspannung an den Stecker X6 (siehe Kapitel **Stecker X6 - Spannungsversorgung**) an.
2. Verbinden Sie den *Modbus-Master* mit dem Anschluss X1 der Steuerung (siehe Kapitel **Stecker X1 - Modbus TCP**).
3. Zum Testen der Schnittstelle senden Sie die Bytes 00 00 00 00 00 0D 00 2B 0D 00 00 01 60 41 00 00 00 00 02 an die Steuerung (eine detaillierte Beschreibung der Modbus-Funktionscodes finden Sie im Kapitel **Modbus TCP**).

Das Statusword (6041<sub>h</sub>) wurde ausgelesen, Sie erhalten diese Antwort: 00 00 00 00 0F 00 2B  
0D 00 00 01 60 41 00 00 00 00 XX XX 06.

## 4.4 Motordaten einstellen

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

- Polpaarzahl: Objekt **2030**<sub>h</sub>:00<sub>h</sub> (Pole pair count) Hier ist die Anzahl der Motorpolpaare einzutragen. Bei einem Schrittmotor wird die Polpaarzahl über den Schrittwinkel berechnet, z.B.  $1,8^\circ = 50$  Polpaare,  $0,9^\circ = 100$  Polpaare (siehe Schrittwinkel im Motordatenblatt). Bei BLDC-Motoren ist die Polpaarzahl direkt im Motordatenblatt angegeben.
- Objekt **2031**<sub>h</sub>:00<sub>h</sub>: Maximal zulässiger Motorstrom (Motorschutz) in mA (siehe Motordatenblatt)
- Objekt **6075**<sub>h</sub>:00<sub>h</sub> Nennstrom des Motors in mA (siehe Motordatenblatt)
- Objekt **6073**<sub>h</sub>:00<sub>h</sub>: Maximaler Strom (bei einem Schrittmotor entspricht in der Regel dem Nennstrom, Bipolar) in Promille des eingestellten Nennstroms (siehe Motordatenblatt). Werkseinstellung: "1000", was 100% entspricht.
- Objekt **203B**<sub>h</sub>:02<sub>h</sub> Maximale Dauer des maximalen Stroms (**6073**<sub>h</sub>) in ms (für die Erstinbetriebnahme empfiehlt Nanotec einen Wert von 100 Millisekunden; Dieser Wert ist später an die konkrete Applikation anzupassen).
- Motortyp einstellen:
  - Schrittmotor:
    - Objekt **3202**<sub>h</sub>:00<sub>h</sub> (Motor Drive Submode Select): Definiert den Motortyp Schrittmotor, aktiviert die Stromabsenkung bei Stillstand des Motors: 0000008h. Siehe auch Kapitel **Inbetriebnahme Open Loop**.
  - BLDC-Motor:
    - Objekt **3202**<sub>h</sub>:00<sub>h</sub> (Motor Drive Submode Select): Definiert den Motortyp BLDC: 00000041h
- Motor mit Encoder: Objekt **2059**<sub>h</sub>:00<sub>h</sub> (Encoder Configuration): Je nach Encoderausführung ist einer der folgenden Werte einzutragen (siehe Motordatenblatt):
  - Versorgungsspannung 5V, differentiell: 00000000h
  - Versorgungsspannung 5V, single-ended: 00000002h
- Motor mit Bremse: Objekt **3202**<sub>h</sub>:00<sub>h</sub> (Motor Drive Submode Select): Für die Erstinbetriebnahme wird die Bremsensteuerung aktiviert. Abhängig von der konkreten Applikation kann diese Konfiguration bei Bedarf später wieder deaktiviert werden. Je nach Motortyp ist eines der folgenden Werte einzutragen:
  - Schrittmotor, Bremsensteuerung (und **Stromabsenkung**) aktiviert: 0000000Ch
  - BLDC-Motor, Bremsensteuerung aktiviert: 00000044h

## 4.5 Motor anschließen

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

- Motor anschließen:
  - an den Anschluss X5, siehe **Stecker X5 - Motoranschluss**
- Encoder/Hallsensoren anschließen:
  - an den Anschluss X2, siehe **Stecker X2 - Encoder/Hall Sensor**
- Bremse anschließen:
  - an den Anschluss X4, siehe **Stecker X4 - Bremsen-Anschluss**

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

## 4.6 Auto-Setup

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



### Tipp

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



### Hinweis

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



### Tipp

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

### 4.6.1 Parameter-Ermittlung

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

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

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

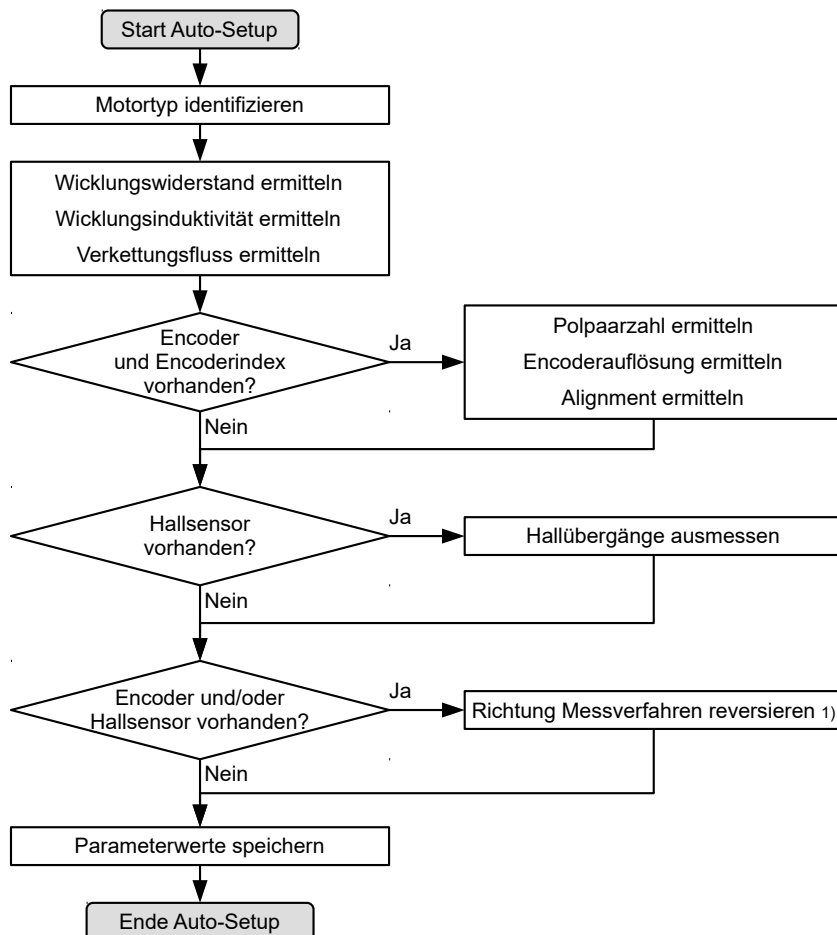
Parameter	Motor ohne Hallsensor	Motor mit Hallsensor
Hallübergänge	-	X



## 4.6.2 Durchführung

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

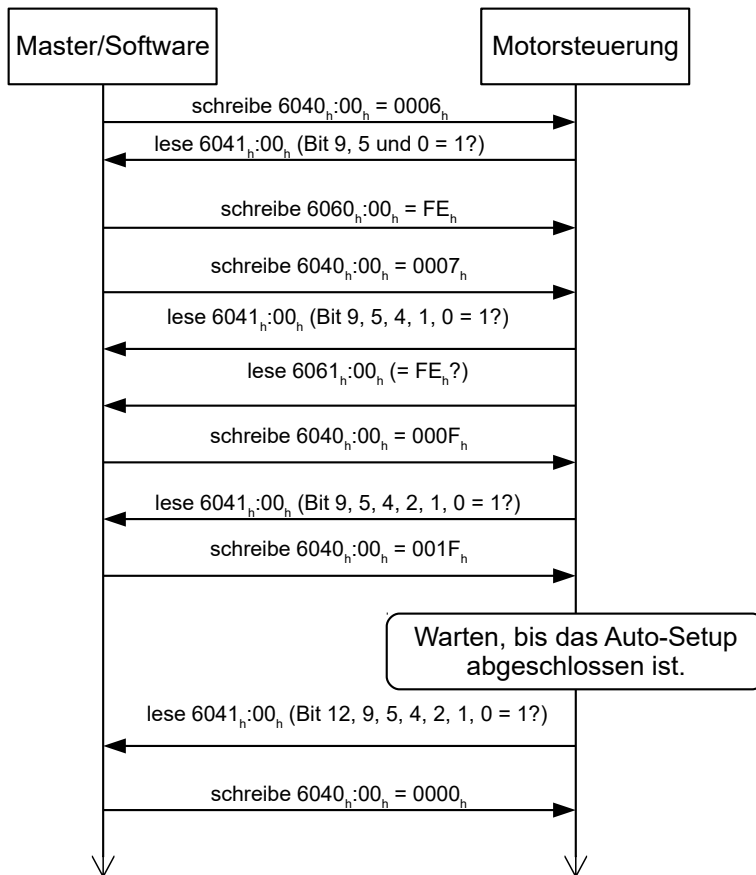
1. Zum Vorwählen des Betriebsmodus *Auto-Setup* tragen Sie in das Objekt 6060<sub>h</sub>:00<sub>h</sub> den Wert "-2" (= "FE<sub>h</sub>") ein.  
Die *Power state machine* muss nun in den Zustand *Operation enabled* versetzt werden, siehe **CiA 402 Power State Machine**.
2. Starten Sie das *Auto-Setup* 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").



### 4.6.3 Parameterspeicherung

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

## 5 Generelle Konzepte

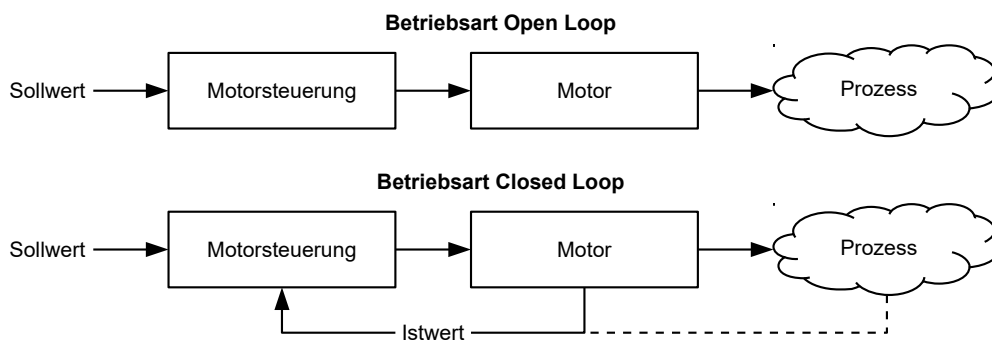
### 5.1 Betriebsarten

#### 5.1.1 Allgemein

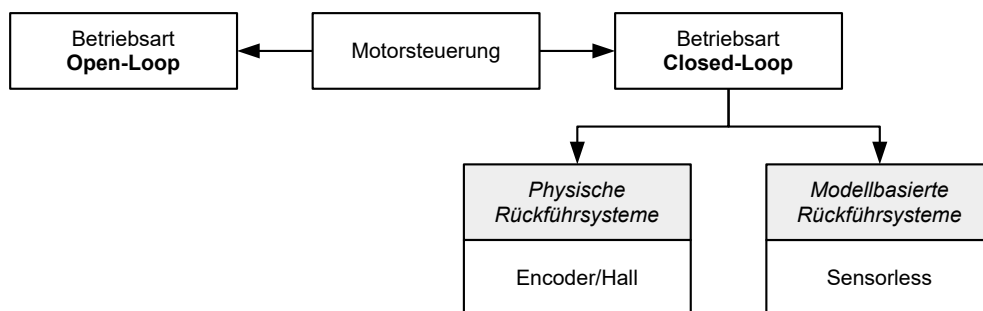
Die Betriebsart von Systemen ohne Rückführung wird als *Open Loop*, die mit Rückführung als *Closed Loop* bezeichnet. In der Betriebsart *Closed Loop* ist es zunächst unerheblich, ob die zurückgeführten Signale vom Motor selbst oder aus dem beeinflussten Prozess kommen.

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

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



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



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

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

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

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

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

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

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

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

## 5.1.2 Open Loop

### Einführung

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

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

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

### Inbetriebnahme

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

- Im Objekt **2030<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 (bei einem Schrittmotor entspricht in der Regel dem Nennstrom, Bipolar) in Promille des eingestellten Nennstroms eingeben (siehe Motordatenblatt). Werkseinstellung: "1000", was 100% entspricht.
- Im Objekt **3202<sub>h</sub>** (Motor Drive Submode Select) das Bit 0 (CL/OL) mit dem Wert "0" belegen.
- Soll der Takt-Richtungs-Modus angewendet werden, dann Kapitel **Takt-Richtungs-Modus** berücksichtigen.

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

- Im Objekt **3202<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 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.

### Optimierungen

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

- Strom reduzieren oder erhöhen, siehe Objekte **6073<sub>h</sub>** bzw. **203B<sub>h</sub>**. 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\_L) optimieren.
- 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 Cycle Synchronous Position

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

### Betriebsmodus Cycle Synchronous Velocity

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

### Betriebsmodus Takt-Richtung

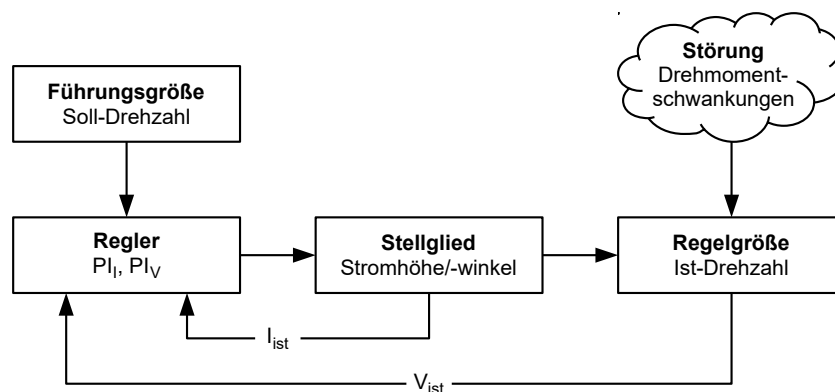
Änderung der Schrittauflösung über die Objekte **2057<sub>h</sub>** (Clock Direction Multiplier) und **2058<sub>h</sub>** (Clock Direction Divider). Beschleunigungs-/Verzögerungsrampen durch Anpassen der Impulsfrequenz optimieren, um den Resonanzbereich möglichst schnell zu durchlaufen.

## 5.1.3 Closed Loop

### Einführung

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

*Closed Loop* am Beispiel einer Drehzahlregelung:



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

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

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

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

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

### Inbetriebnahme

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

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

Das Bit 0 im **3202<sub>h</sub>** muss gesetzt sein.

## 5.2 CiA 402 Power State Machine

### 5.2.1 Zustandsmaschine

#### CiA 402

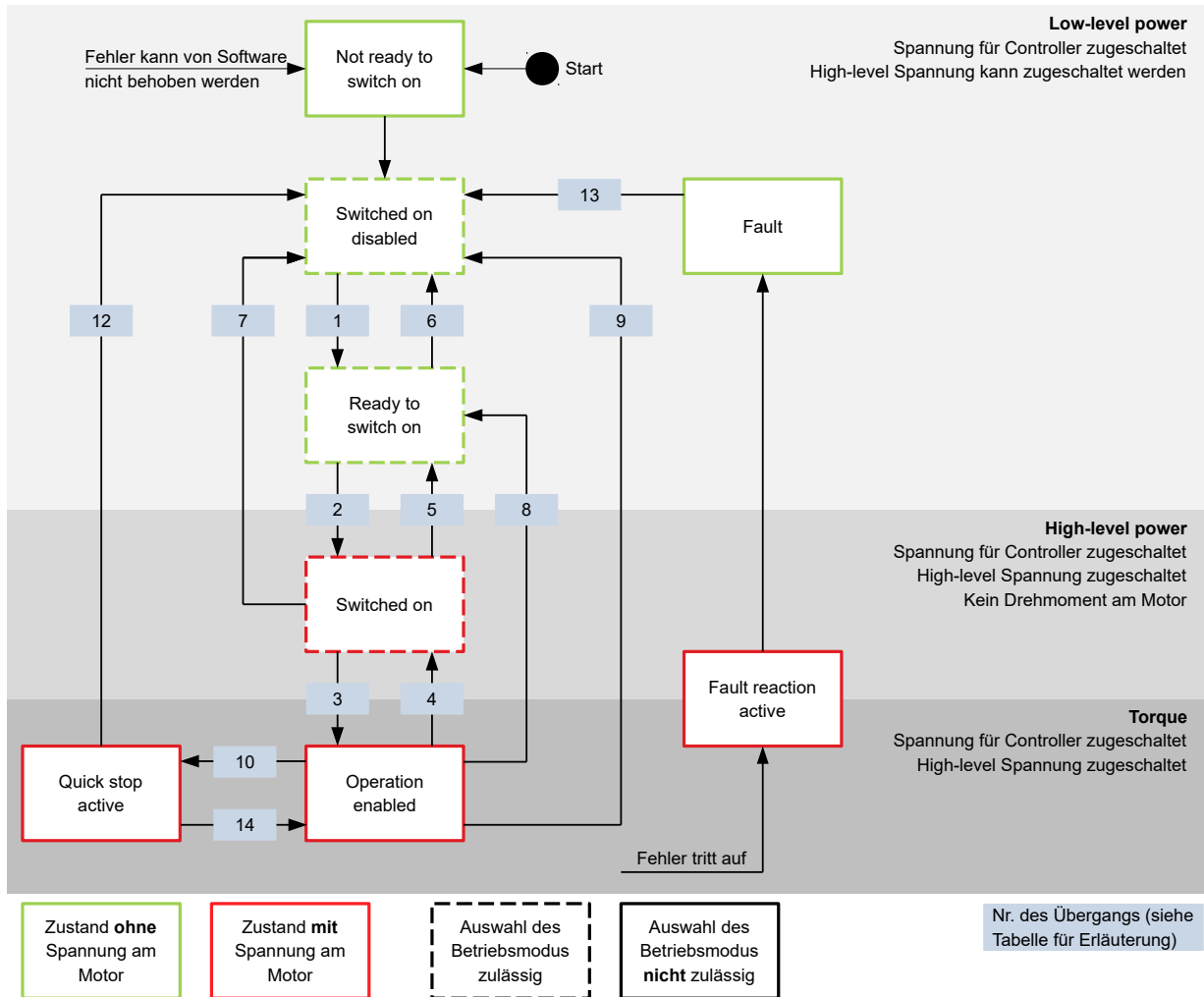
Um die Steuerung betriebsbereit zu schalten, ist es notwendig, eine Zustandsmaschine (*State Machine*) zu durchlaufen. Diese ist im *CANopen-Standard 402* definiert. Zustandsänderungen werden im Objekt **6040<sub>h</sub>** (Controlword) angefordert. Der tatsächliche Zustand der Zustandsmaschine lässt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

#### 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. Einzige Ausnahme ist das Rücksetzen des Fehlers (Fault reset): 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	1, 5, 8
Switch on	0	0	1	1	1	2
Disable voltage	0	X	X	0	X	6, 7, 9, 12
Quick stop	0	X	0	1	X	10
Disable operation	0	0	1	1	1	4
Enable operation	0	1	1	1	1	3
Enable operation after Quick stop	0	1	1	1	1	14



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

### Statusword

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

Statusword (6041 <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.

### Betriebsmodus

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

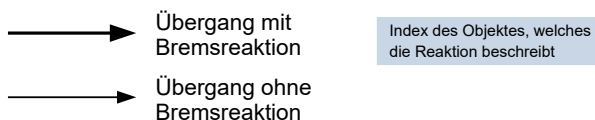
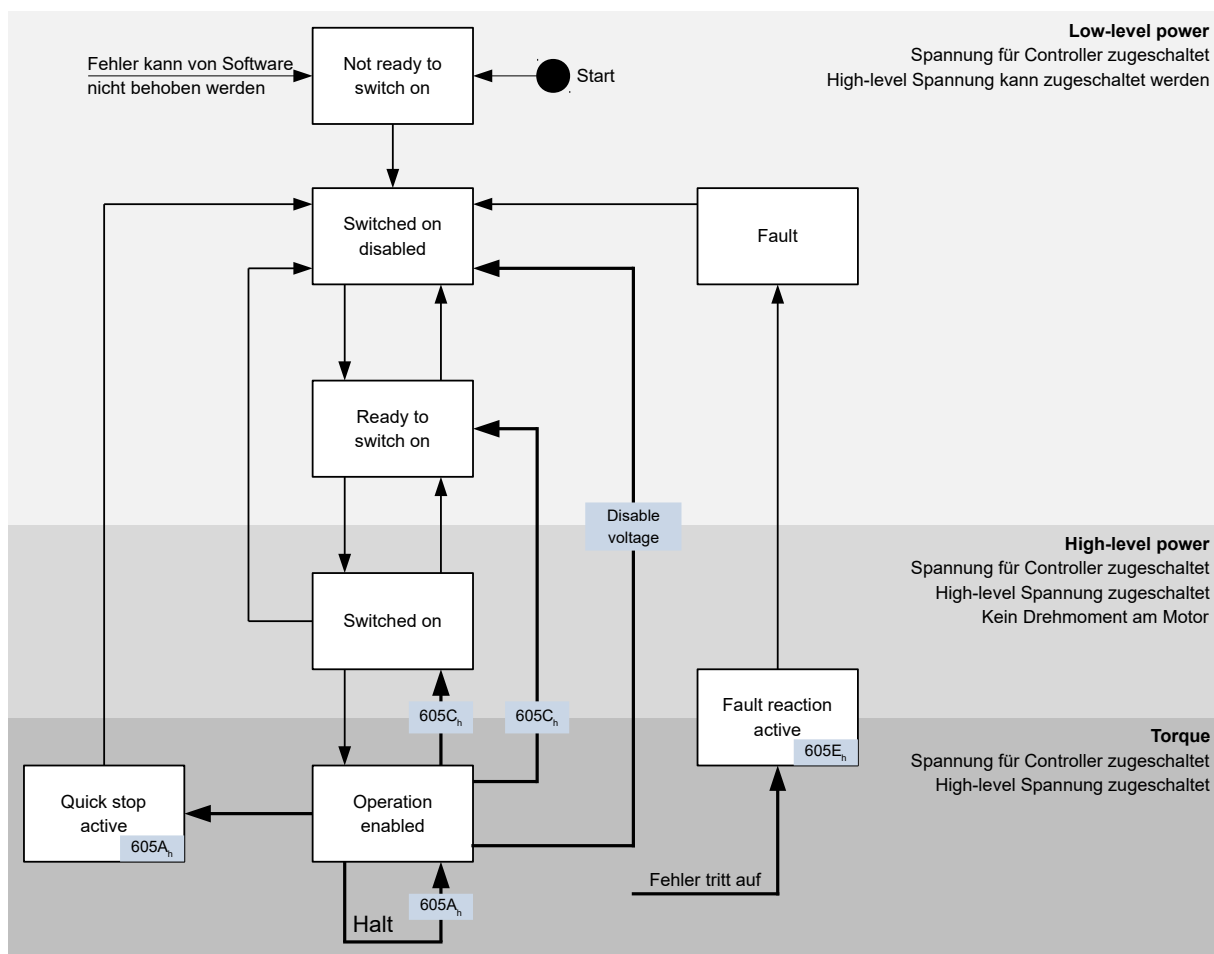
Die Einstellung oder Änderung des Betriebsmodus ist jederzeit möglich.

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

### Bremsreaktionen

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

Die nachfolgende Grafik zeigt eine Übersicht der Bremsreaktionen.



### Quick stop active

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

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

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> 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> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

### Ready to switch on

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

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

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

### Switched on

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

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

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

### Halt

Das Bit ist gültig in folgenden Modi:

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

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

Wert in Objekt 605D <sub>h</sub>	Beschreibung
-32768 bis 0	Reserviert

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

### Fault

Fehlerfall (fault):

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

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

### Schlepp-/Schlupffehler

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

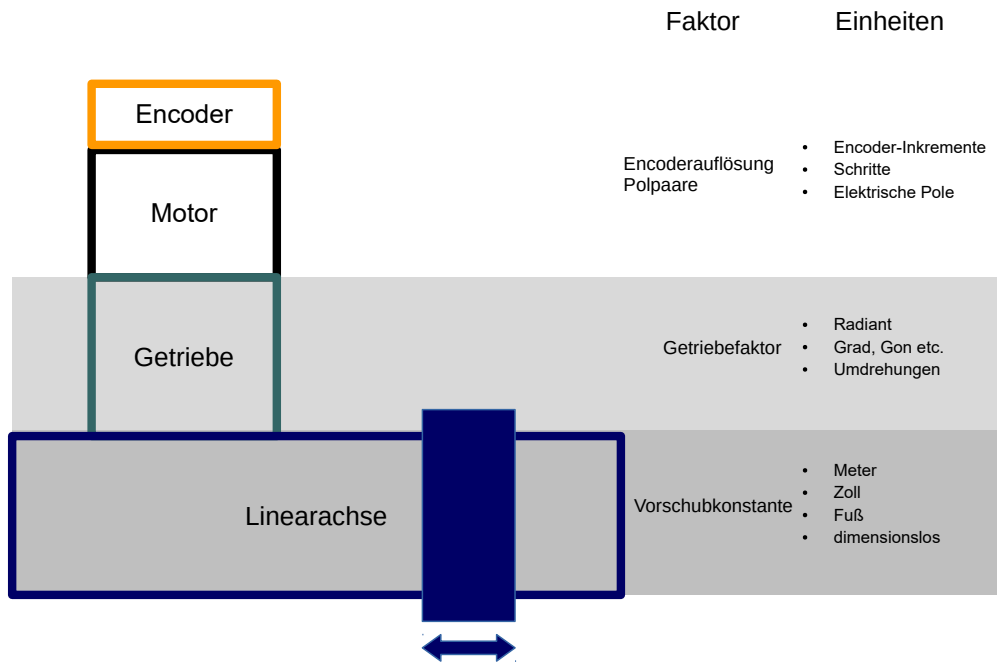
Wert	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit <i>slow down ramp</i> (Bremsbeschleunigung je nach Betriebsart)
2	Abbremsen mit <i>quick stop ramp</i> (Bremsbeschleunigung je nach Betriebsart)
3 bis 32767	Reserviert

Sie können die Fehlerüberwachung deaktivieren, indem Sie das Objekt **6065<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 (*S*) als auch einige spezifische Einheiten unterstützt. Ebenfalls möglich ist die Angabe einer Zehnerpotenz als Faktor.

In der nachfolgenden Tabelle sind alle unterstützten Einheiten für die Position und deren Werte für **60A8<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> (Winkleinheit, 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')

Name	Einheitenzeichen	Wert	6091 <sub>h</sub>	6092 <sub>h</sub>	Beschreibung
mechanical revolution		B4 <sub>h</sub>	ja	nein	<i>Umdrehung</i>
encoder increment		B5 <sub>h</sub>	nein	nein	<i>Encoder-Inkmente.</i> Abhängig vom verwendeten Sensor (Encoder/ Hall-Sensor) und <b>Betriebsart</b> . Im <i>Open Loop</i> - und <i>Sensorless</i> -Betrieb entsprechen 32000 <sub>h</sub> Inkmente einer Motorumdrehung.
step		AC <sub>h</sub>	nein	nein	<i>Schritte.</i> Bei 2-phasigen Schrittmotoren entspricht die Anzahl der Polpaare ( <b>2030<sub>h</sub></b> ) multipliziert mit 4 einer Umdrehung. Bei 3-phasigen BLDC-Motoren entspricht die Anzahl der Polpaare ( <b>2030<sub>h</sub></b> ) 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 ( <b>2030<sub>h</sub></b> ) 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 verwendeten Encoders/Sensors berechnet sich aus den Encoder-Inkmenten (**608F<sub>h</sub>:1<sub>h</sub>** (Encoder Increments)) pro Motorumdrehungen (**608F<sub>h</sub>:2<sub>h</sub>** (Motor Revolutions)):

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

### 5.3.3 Getriebeübersetzung

Die Getriebeübersetzung berechnet sich aus Motorumdrehungen (**6091<sub>h</sub>:1** (Motor Revolutions)) pro Achsendrehung (**6091<sub>h</sub>:2** (Shaft Revolutions)) wie folgt:

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

### 5.3.4 Vorschubkonstante

Die Vorschubkonstante berechnet sich aus dem Vorschub (**6092<sub>h</sub>:1** (Feed) pro Umdrehung der Abtriebsachse (**6092<sub>h</sub>:2** (Shaft Revolutions)) wie folgt:

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

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

### 5.3.5 Berechnungsformeln für Benutzereinheiten

#### Positionseinheit

Das Objekt **60A8<sub>h</sub>** 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<sub>h</sub>** 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<sub>h</sub>**) 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<sub>h</sub>** 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<sub>h</sub>**) von 1 fährt der Motor genau eine mechanische Umdrehung (wenn die **Getriebeübersetzung** und **Vorschubkonstante** 1:1 sind).

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

### Geschwindigkeitseinheit

Das Objekt **60A9<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**)

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<sub>h</sub>** 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<sub>h</sub>** 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.



### 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<sub>h</sub>:01<sub>h</sub>**) geteilt durch Faktor für Nenner (**6096<sub>h</sub>:02<sub>h</sub>**).

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

### Beschleunigungseinheit

Die Beschleunigungseinheit ist **Geschwindigkeitseinheit** pro Sekunde.

### Umrechnungsfaktor für die Beschleunigungseinheit

Der Faktor n für die Beschleunigungseinheit errechnet sich aus Zähler (**6097<sub>h</sub>:01<sub>h</sub>**) geteilt durch Nenner (**6097<sub>h</sub>:02<sub>h</sub>**).



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

### Ruckeinheit

Die Ruckeinheit ist **Beschleunigungseinheit** pro Sekunde.

#### Umrechnungsfaktor für den Ruck

Der Faktor n für den Ruck errechnet sich aus Zähler (**60A2<sub>h</sub>:01<sub>h</sub>**) geteilt durch Nenner (**60A2<sub>h</sub>:02<sub>h</sub>**).

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

## 5.4 Begrenzung des Bewegungsbereichs

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

### 5.4.1 Verhalten beim Erreichen der Endschalter

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

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

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

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

## 5.4.2 Software-Endschalter

Die Steuerung berücksichtigt Software-Endschalter (**607D<sub>n</sub>** (Software Position Limit)). Zielpositionen (**607A<sub>n</sub>**) werden durch **607D<sub>n</sub>** limitiert, die absolute Zielposition darf nicht größer sein als die Grenzen in **607D<sub>n</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 von 1 ms. Das bedeutet, dass Daten jeweils alle 1 ms verarbeitet werden, mehrfache Änderungen eines Wertes (z.B. Wert eines Objektes oder Pegel an einem digitalen Eingang) innerhalb einer ms können nicht erfasst werden.

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

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

## 6 Betriebsmodi

### 6.1 Profile Position

#### 6.1.1 Übersicht

##### Beschreibung

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

##### Aktivierung

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

##### Controlword

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

- Bit 4 startet einen Fahrauftrag. Dieser wird bei einem Übergang von "0" nach "1" übernommen. Eine Ausnahme besteht, wenn es von einem anderen Betriebsmodus nach *Profile Position* gewechselt wird: Ist das Bit 4 bereits gesetzt, muss es nicht auf "0" und wieder auf "1" gesetzt werden, damit der Fahrauftrag gestartet wird.
- Bit 5: Ist dieses Bit auf "1" gesetzt, wird ein durch Bit 4 ausgelöster Fahrauftrag sofort ausgeführt. Ist es auf "0" gesetzt, wird der gerade ausgeführte Fahrauftrag zu Ende gefahren und erst im Anschluss der nächste Fahrauftrag gestartet.
- Bit 6: Bei "0" ist die Zielposition (**607A<sub>h</sub>**) absolut und bei "1" ist die Zielposition relativ. Die Referenzposition ist abhängig von den Bits 0 und 1 des Objekts **60F2<sub>h</sub>**.
- 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>**.
- 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.

## Statusword

Folgende Bits im Objekt **6041<sub>h</sub>** (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<sub>h</sub>**) innerhalb eines Toleranzfensters (**6067<sub>h</sub>**) steht.
- Bit 11: Limit überschritten: Die Sollposition über- oder unterschreitet die in **607D<sub>h</sub>** 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.

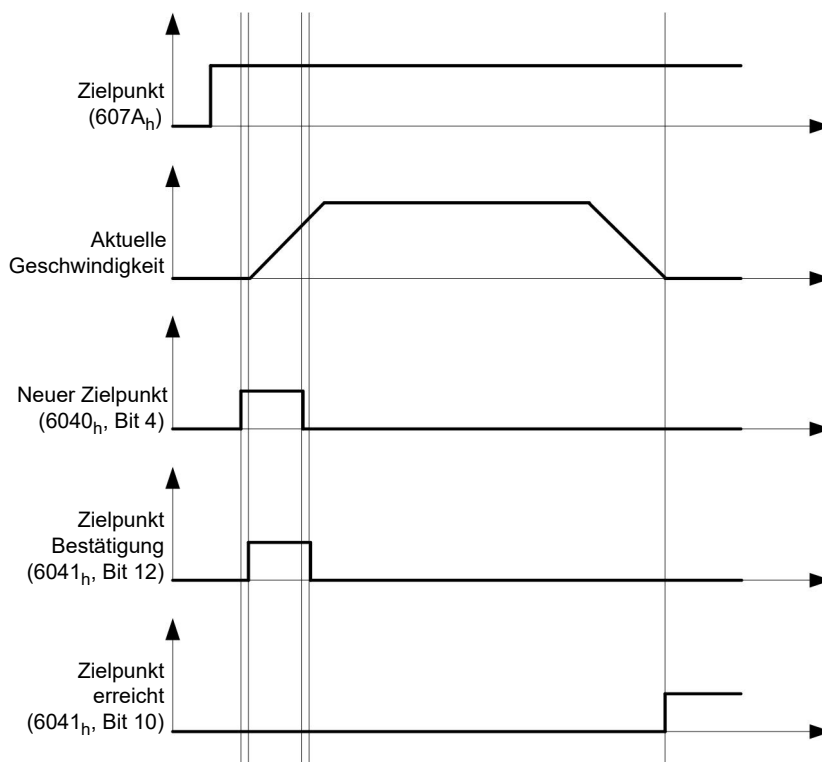
Das Bit wird nicht gesetzt, wenn eine der folgenden Bedingungen erfüllt ist:

- 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<sub>h</sub>** (Following Error Window) und **6066<sub>h</sub>** (Following Error Time Out)).

## 6.1.2 Setzen von Fahrbefehlen

### Fahrbefehl

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



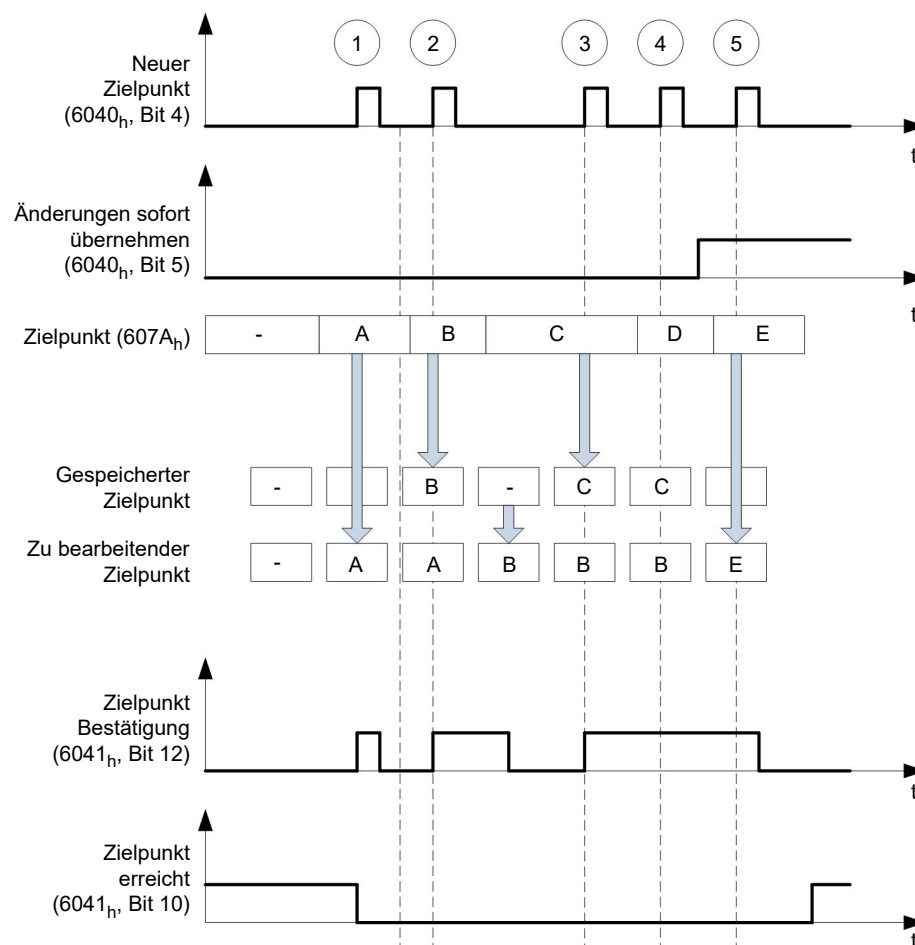
Die Steuerung kann das Bit 4 im Objekt **6040<sub>h</sub>** (Controlword) auch selbstständig zurücksetzen. Das wird mit den Bits 4 und 5 des Objektes **60F2<sub>h</sub>** eingestellt.

### 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).

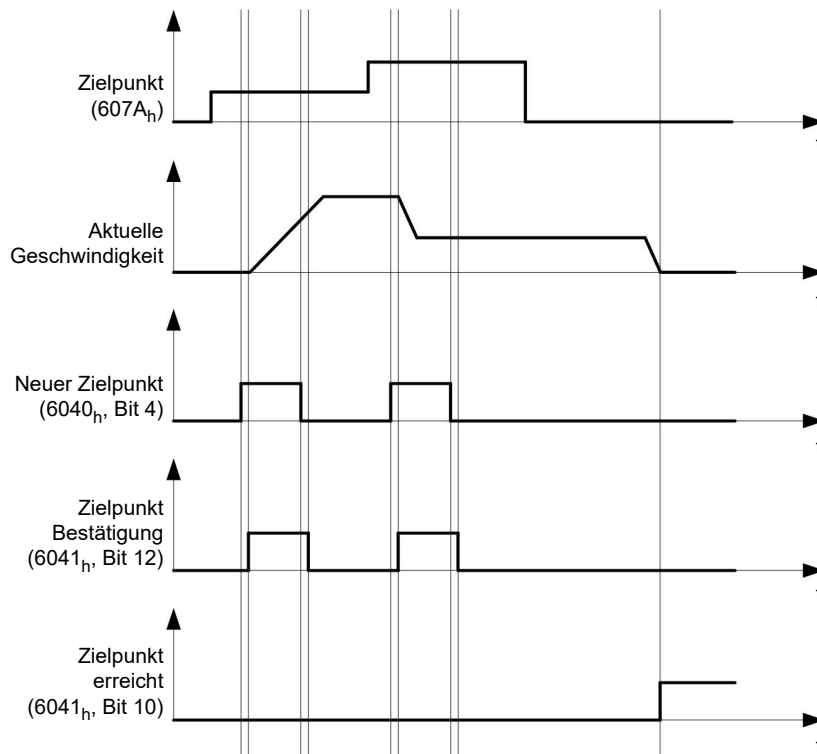
Sollte der Zwischenspeicher schon voll sein, wird ein neuer Zielpunkt ignoriert (Zeitpunkt 4). Wird Bit 5 im Objekt **6040<sub>h</sub>** (Controlword, Bit: "Change Set-Point Immediately") gesetzt, arbeitet die Steuerung ohne den Zwischenspeicher, neue Fahrbefehle werden direkt umgesetzt (Zeitpunkt 5).

### Zeitpunkte



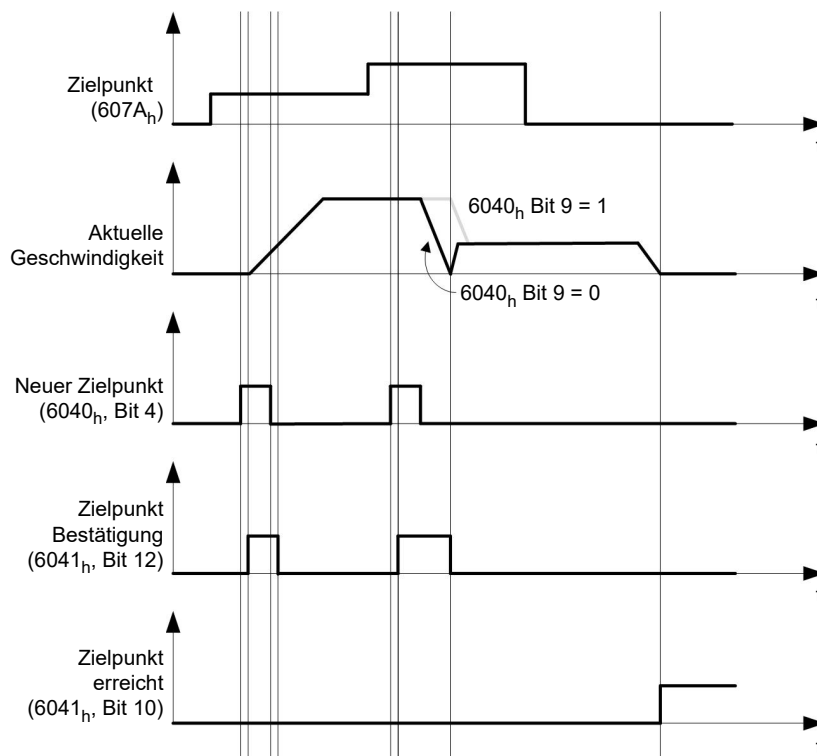
### Übergangsprozedur für zweite Zielposition

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.



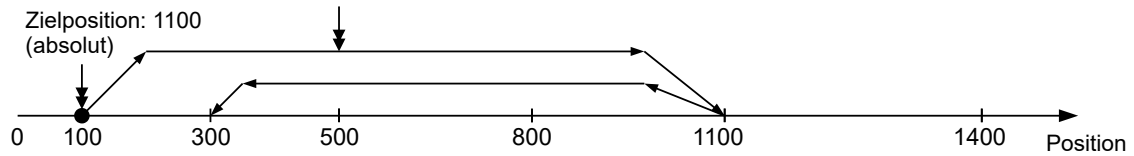
#### Mögliche Kombinationen von Fahrbefehlen

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

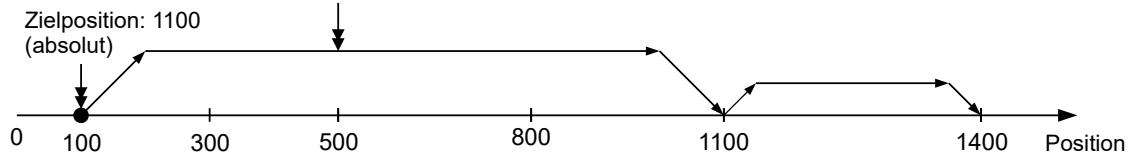
Die nachfolgenden Bilder setzen voraus:

- 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.

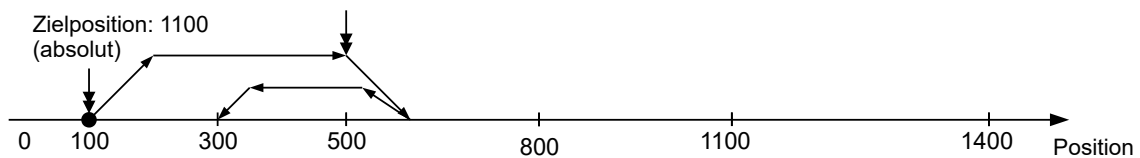
- Änderung im Zielpunkt übernehmen (6040<sub>h</sub>:00 Bit 5 = 0)
- Positionierung absolut (6040<sub>h</sub>:00 Bit 6 = 0)
- Zielposition: 300



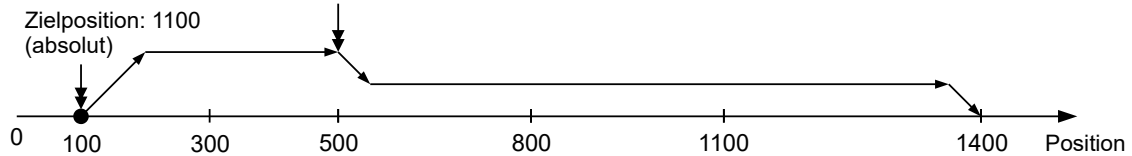
- Relativ zu der vorhergehenden Zielposition (60F2<sub>h</sub>:00 = 0)
- Änderung im Zielpunkt übernehmen (6040<sub>h</sub>:00 Bit 5 = 0)
- Positionierung relativ (6040<sub>h</sub>:00 Bit 6 = 1)
- Zielposition: 300



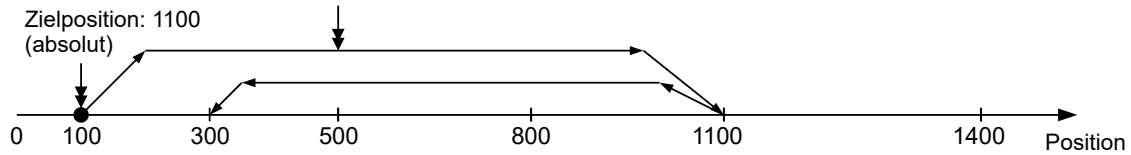
- Änderung sofort übernehmen (6040<sub>h</sub>:00 Bit 5 = 1)
- Positionierung absolut (6040<sub>h</sub>:00 Bit 6 = 0)
- Zielposition: 300



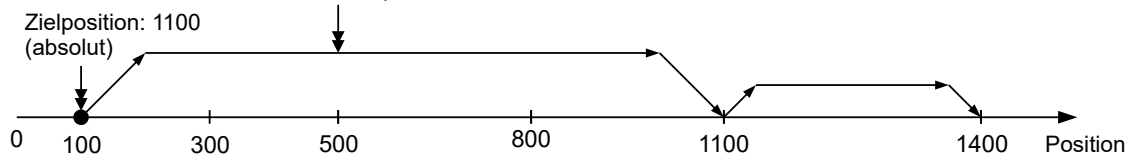
- Relativ zu der vorhergehenden Zielposition (60F2<sub>h</sub>:00 = 0)
- Änderung sofort übernehmen (6040<sub>h</sub>:00 Bit 5 = 1)
- Positionierung relativ (6040<sub>h</sub>:00 Bit 6 = 1)
- Zielposition: 300



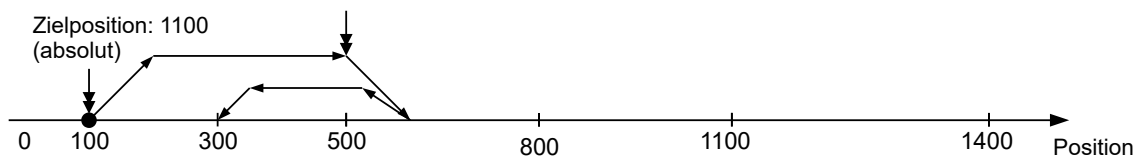
- Änderung im Zielpunkt übernehmen (6040<sub>h</sub>:00 Bit 5 = 0)
- Positionierung absolut (6040<sub>h</sub>:00 Bit 6 = 0)
- Zielposition: 300



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

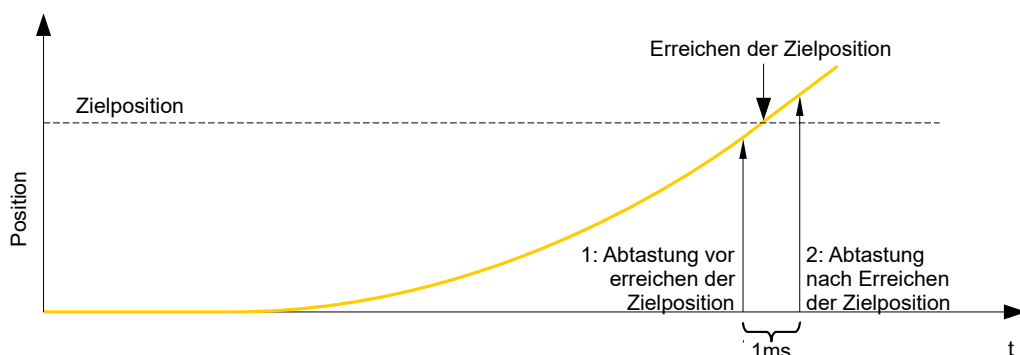


- Änderung sofort übernehmen (6040<sub>h</sub>:00 Bit 5 = 1)
- Positionierung absolut (6040<sub>h</sub>:00 Bit 6 = 0)
- Zielposition: 300



### 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

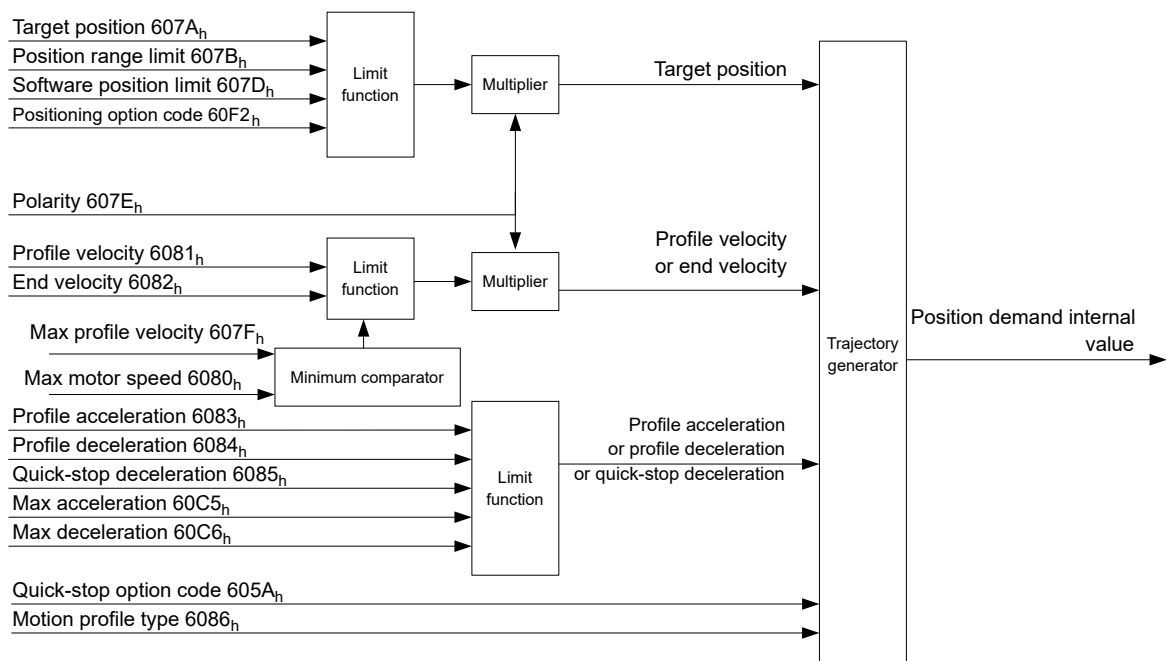
### Objekteinträge

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

- **607A<sub>h</sub>** (Target Position): vorgesehene Zielposition
- **607D<sub>h</sub>** (Software Position Limit): Definition der Endanschläge (siehe Kapitel **Software-Endschalter**)
- **607C<sub>h</sub>** (Home Offset): Gibt die Differenz zwischen Null-Position der Steuerung und dem Referenzpunkt der Maschine in **benutzerdefinierten Einheiten** an. (siehe "**Homing**")
- **607B<sub>h</sub>** (Position Range Limit): Grenzen einer Modulo-Operation zur Nachbildung einer endlosen Rotationsachse
- **607E<sub>h</sub>** (Polarity): Drehrichtung
- **6081<sub>h</sub>** (Profile Velocity): maximale Geschwindigkeit, mit der die Position angefahren werden soll
- **6082<sub>h</sub>** (End Velocity): Geschwindigkeit beim Erreichen der Zielposition
- **6083<sub>h</sub>** (Profile Acceleration): gewünschte Anfahrbeschleunigung
- **6084<sub>h</sub>** (Profile deceleration): gewünschte Bremsbeschleunigung
- **6085<sub>h</sub>** (Quick Stop Deceleration): Nothalt-Bremsbeschleunigung im Falle des Zustandes "Quick stop active" der "CiA 402 Power State machine"
- **6086<sub>h</sub>** (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<sub>h</sub>:1<sub>h</sub>- 4<sub>h</sub> als Limitierungen des Rucks gesetzt.
- **60C5<sub>h</sub>** (Max Acceleration): die maximale Beschleunigung, die beim Anfahren der Endposition nicht überschritten werden darf
- **60C6<sub>h</sub>** (Max Deceleration): die maximale Bremsbeschleunigung, die beim Anfahren der Endposition nicht überschritten werden darf
- **60A4<sub>h</sub>** (Profile Jerk), Subindex 01<sub>h</sub> bis 04<sub>h</sub>: Objekte zur Beschreibung der Grenzwerte für den Ruck.
- Die Geschwindigkeit wird durch **607F<sub>h</sub>** (Max Profile Velocity) und **6080<sub>h</sub>** (Max Motor Speed) begrenzt, der kleinere Wert wird als Grenze herangezogen.
- **60F2<sub>h</sub>** (Positioning Option Code): definiert das Positionierverhalten

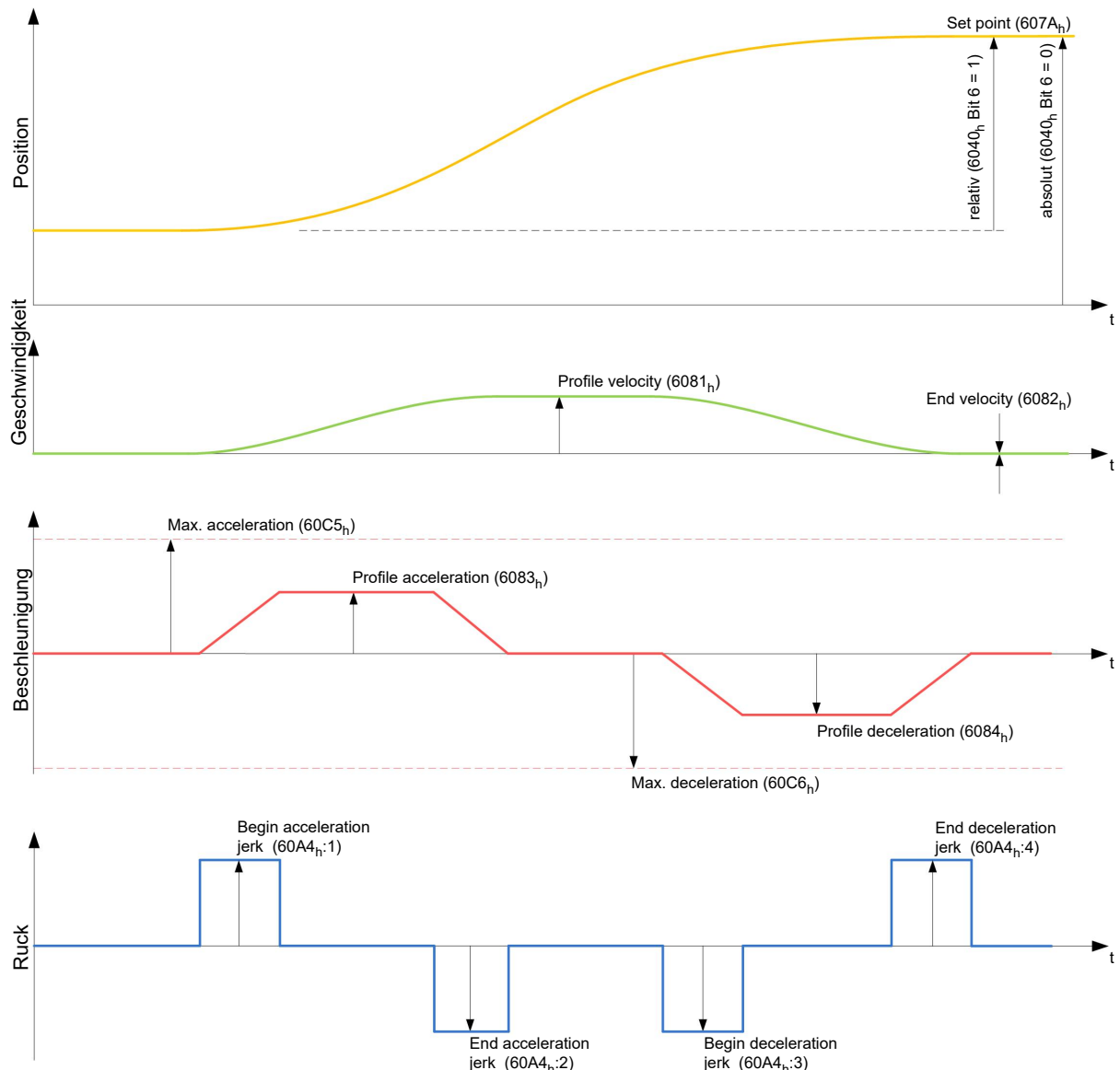
### Objekte für die Positionierfahrt

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



## Parameter für die Zielposition

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



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

### Beschreibung

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

### Ruck-begrenzter Modus

Eine ruck-begrenzte Positionierung lässt sich erreichen, indem das Objekt **6086<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.

### 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, ruckbegrenzte 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:

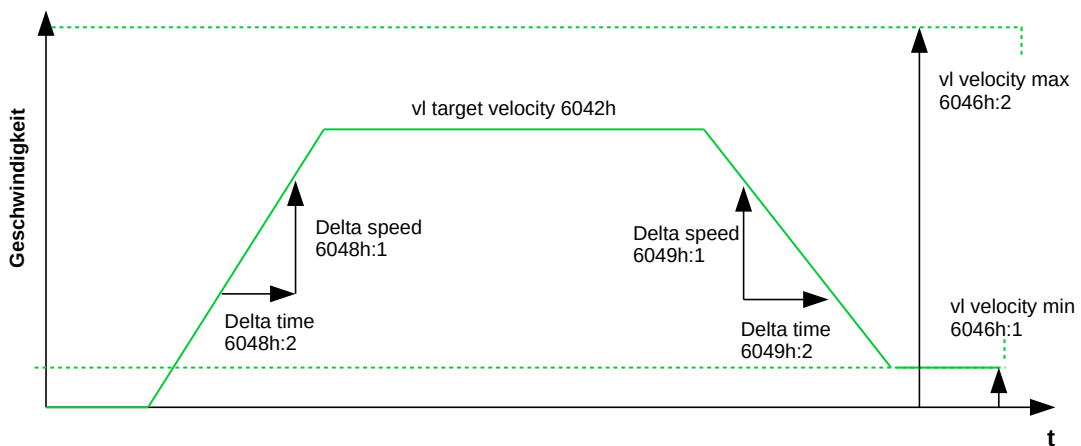
- **604C<sub>h</sub>** (Dimension Factor):  
Hier wird die Einheit der Geschwindigkeitsangaben für die nachfolgenden Objekte festgelegt. Der Subindex 1 enthält den Nenner (Multiplikator) und der Subindex 2 den Zähler (Divisor), mit dem interne Geschwindigkeitsangaben in Umdrehungen pro Minute verrechnet 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 Schnellstop-Rampe eingestellt werden. Die Subindizes 1 und 2 sind dabei identisch wie bei Objekt **6048<sub>h</sub>** beschrieben.
- **6080<sub>h</sub>** (Max Motor Speed): maximale Geschwindigkeit

Folgende Objekte können zur Kontrolle der Funktion genutzt werden:

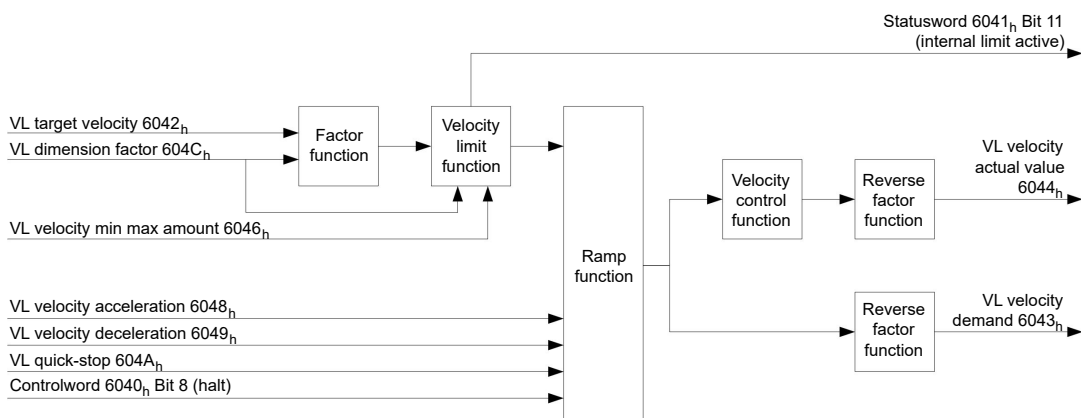
- **6043<sub>h</sub>** (VI Velocity Demand)
- **6044<sub>h</sub>** (VI Velocity Actual Value)

### Geschwindigkeiten im Velocity Mode



### Objekte für den Velocity Mode

Der Rampengenerator folgt der Zielgeschwindigkeit unter Einhaltung der eingestellten Geschwindigkeits- und Beschleunigungsgrenzen. Solange eine Begrenzung aktiv ist, wird das Bit 11 im Objekt **6041<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 <b>606D<sub>h</sub></b> und <b>606E<sub>h</sub></b> )
1	1	Geschwindigkeit der Achse ist 0

- Bit 13 (Deviation Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schlupffehler größer als die eingestellten Grenzen ist (**60F8<sub>h</sub> Max Slippage** und **203F<sub>h</sub> Max Slippage Time Out**).

## 6.3.5 Objekteinträge

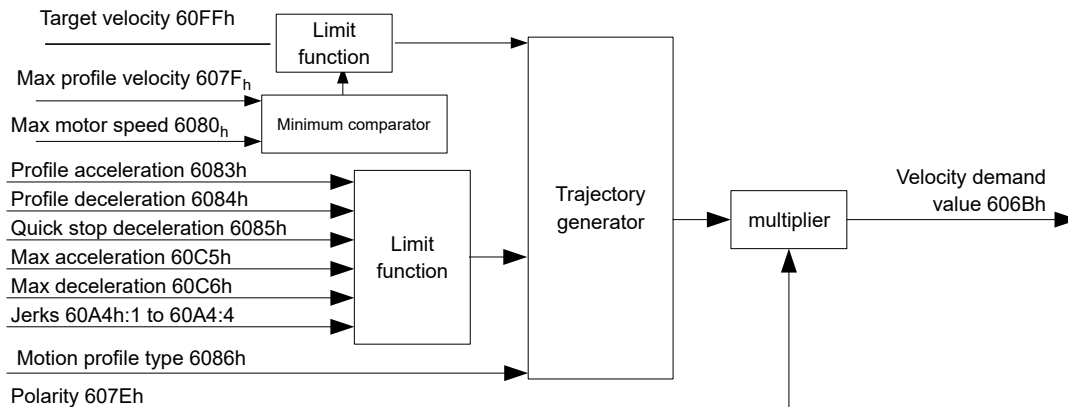
Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **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 im Velocity Mode.
- **6084<sub>h</sub>** (Profile Deceleration):  
Setzt den Wert für die Bremsrampe im Velocity-Mode.
- **6085<sub>h</sub>** (Quick Stop Deceleration):  
Setzt den Wert für die Bremsrampe für die Schnellbremsung im Velocity Mode.
- **6086<sub>h</sub>** (Motion Profile Type):

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

- **60FF<sub>h</sub>** (Target Velocity):  
Gibt die zu erreichende Zielgeschwindigkeit an.
- Die Geschwindigkeit wird durch **607F<sub>h</sub>** (Max Profile Velocity) und **6080<sub>h</sub>** (Max Motor Speed) begrenzt, der kleinere Wert wird als Grenze herangezogen.

### Objekte im Profile Velocity Mode

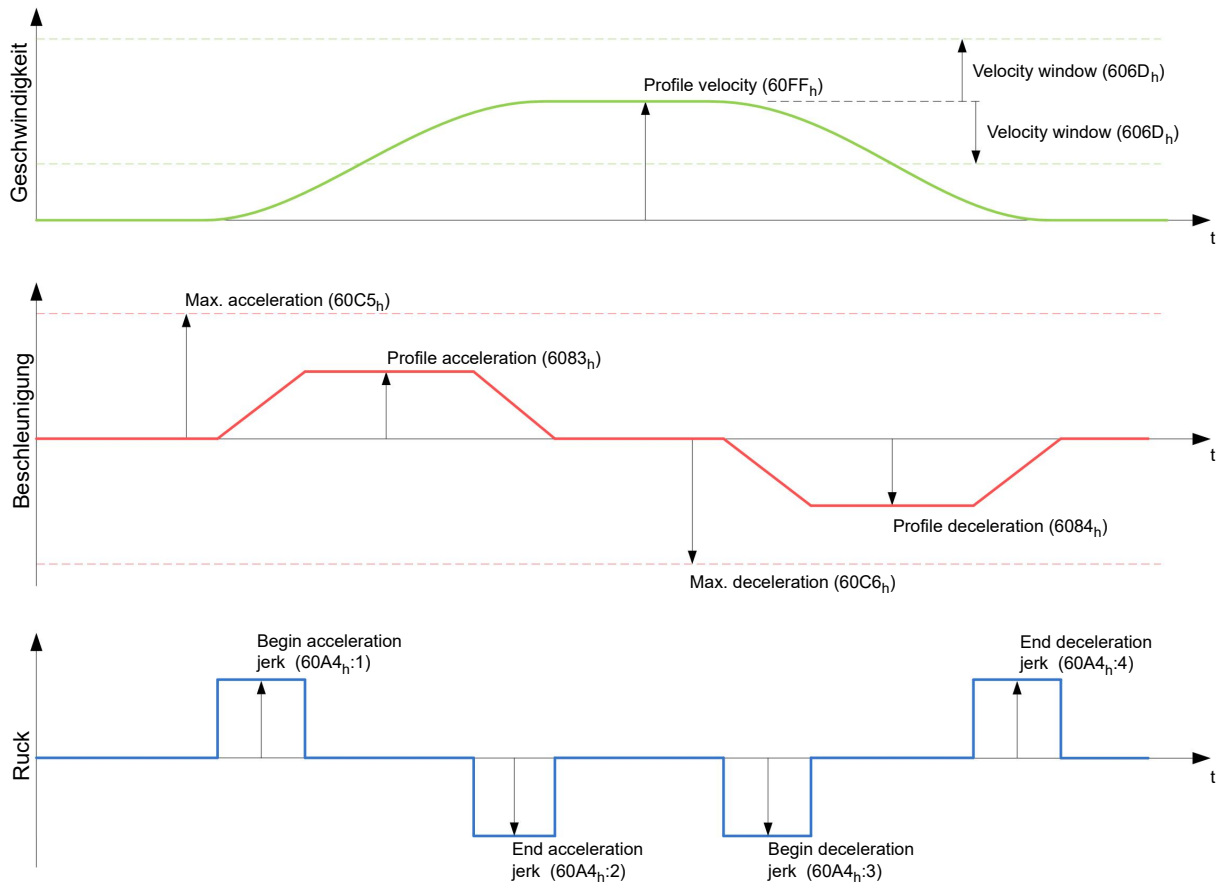


### Aktivierung

Nachdem der Modus im Objekt **6060<sub>h</sub>** (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<sub>h</sub>** beschleunigt (siehe nachfolgende Bilder). Dabei werden die Geschwindigkeits-, Beschleunigungs- und bei ruck-begrenzten Rampen auch die Ruckgrenzwerte berücksichtigt.

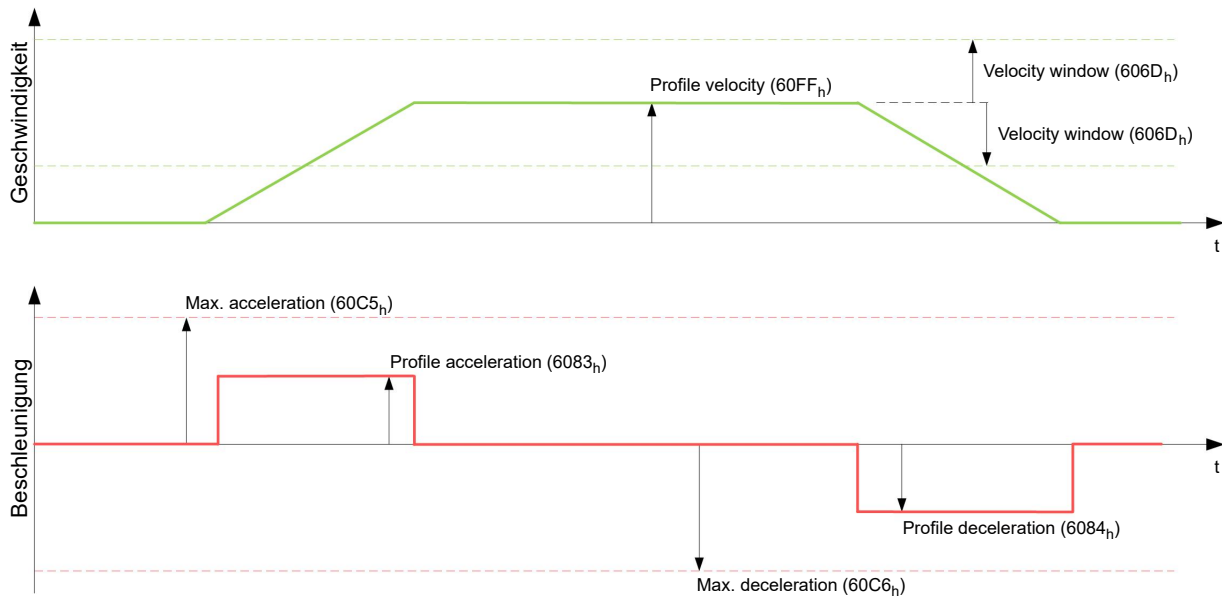
### Limitierungen im ruck-limitierten Fall

Das folgende Bild zeigt die einstellbaren Limitierungen im ruck-limitierten Fall (**6086<sub>h</sub> = 3**).



### Limitierungen im Trapez-Fall

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



## 6.4 Profile Torque

### 6.4.1 Beschreibung

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



### Hinweis

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

## 6.4.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060<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.

6040 <sub>h</sub> Bit 8	6041 <sub>h</sub> Bit 10	Beschreibung
0	0	Vorgegebenes Drehmoment nicht erreicht
0	1	Vorgegebenes Drehmoment erreicht
1	0	Achse beschleunigt
1	1	Geschwindigkeit der Achse ist 0

- Bit 11: Limit überschritten: Das Zieldrehmoment (**6071<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





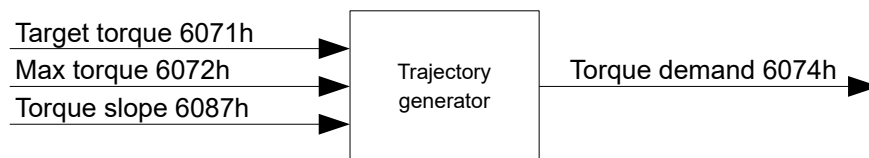
### Hinweis

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

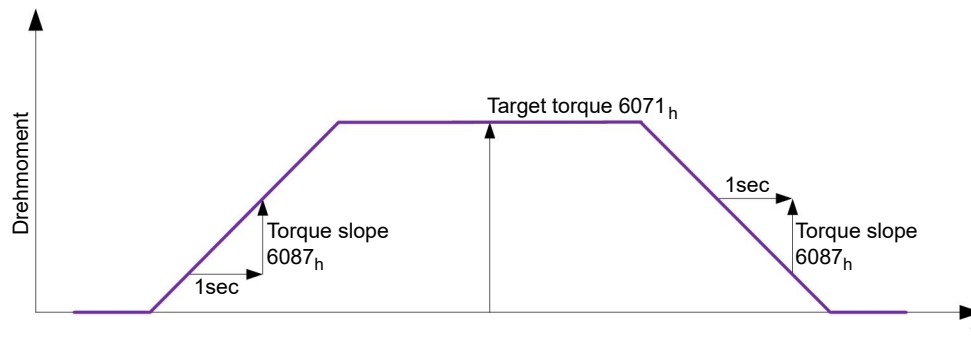
Die folgenden Objekte werden zudem für diesen Operationsmodus benötigt:

- **3202<sub>h</sub>** 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<sub>h</sub>** begrenzt werden und der Regler kann im Feldschwächebetrieb arbeiten.  
Wird dieses Bit auf "1" gesetzt, arbeitet der Regler im ("Real") Torque Mode, die maximale Geschwindigkeit kann hier nicht begrenzt werden und der Feldschwächebetrieb ist nicht möglich.

### Objekte des Rampengenerators



### Torque-Verlauf



## 6.5 Homing

### 6.5.1 Übersicht

#### Beschreibung

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

#### Aktivierung

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



### Tip

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

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

### Controlword

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

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

### Statusword

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

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



### Hinweis

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

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

### Objekteinträge

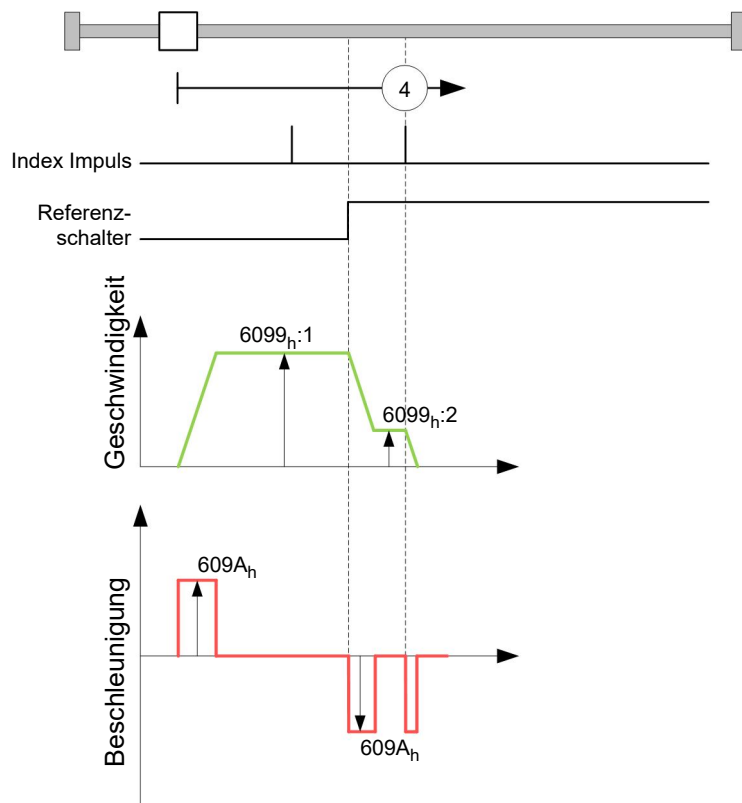
Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

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

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

### Geschwindigkeiten der Referenzfahrt

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



## 6.5.2 Referenzfahrt-Methode

### Beschreibung

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

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

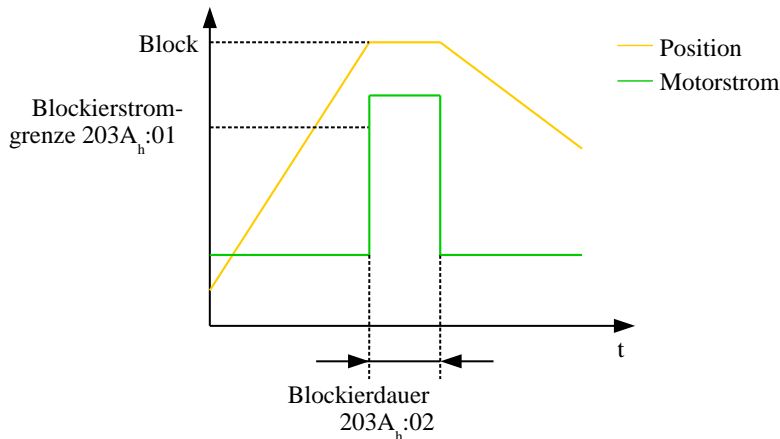
Bei Methoden, die Homing auf Block benutzen, gelten die gleichen Abbildungen wie für die Methoden mit Endschalter. Da sich außer den fehlenden Endschaltern nichts ändert, wurde auf neue Abbildungen verzichtet. Hier gilt für die Abbildungen, dass die Endschalter durch eine mechanische Blockierung ersetzt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

### Homing auf Block

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

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

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



### Methoden-Überblick

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

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

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

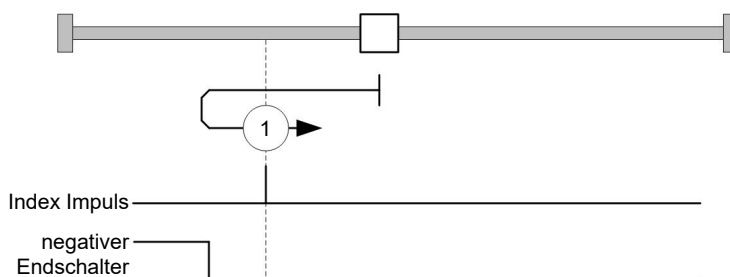
Folgende Methoden können für Homing auf Block benutzt werden:

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

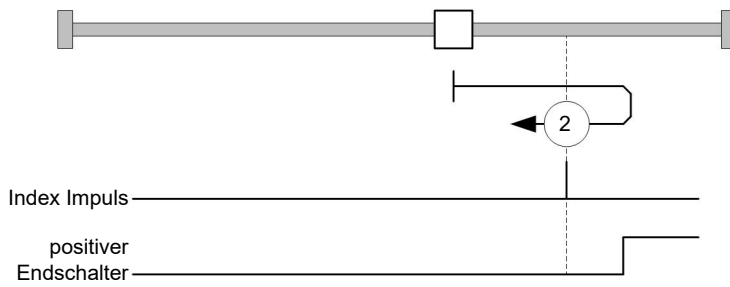
### Methoden 1 und 2

Referenzieren auf Endschalter und Index-Impuls.

Methode 1 referenziert auf negativen Endschalter und Index-Impuls:



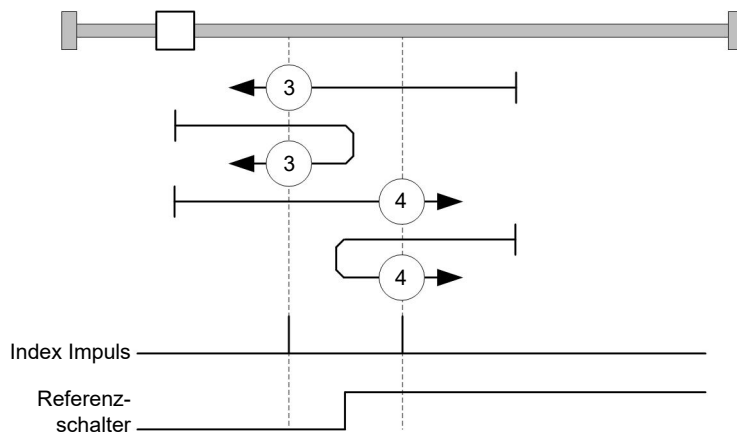
Methode 2 referenziert auf positiven Endschalter und Index-Impuls:



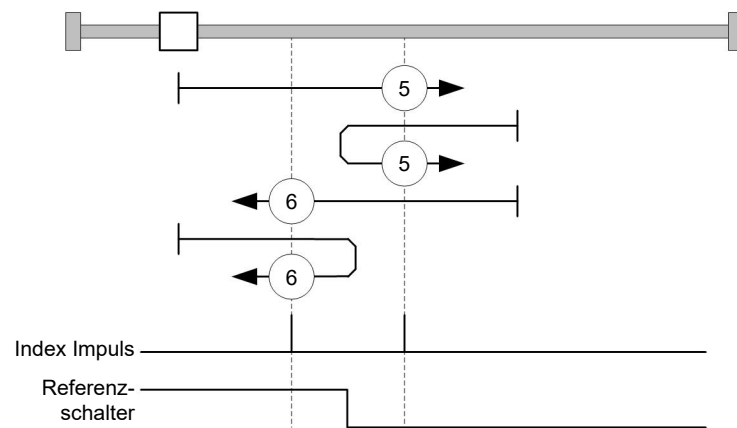
### Methoden 3 bis 6

Referenzieren auf die Schaltflanke des Referenzschalters und Index-Impuls.

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



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

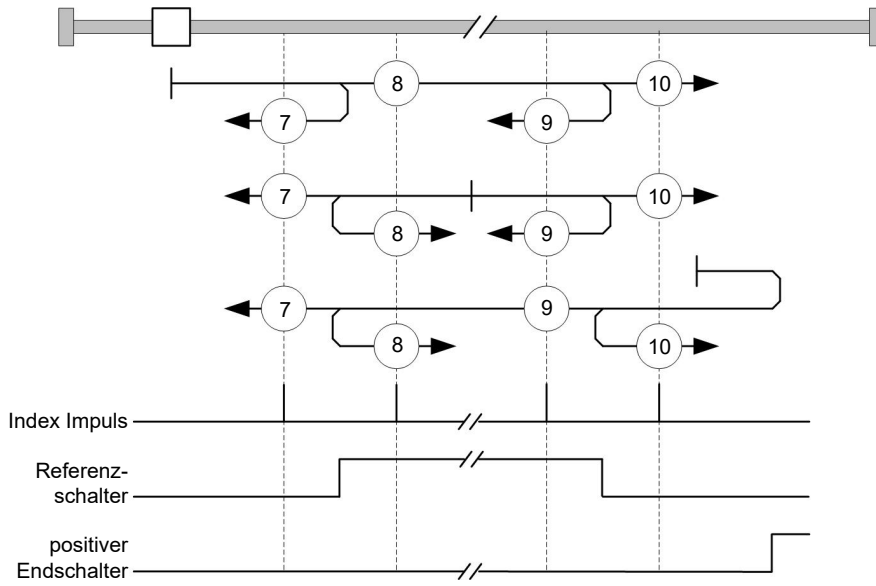


### Methoden 7 bis 14

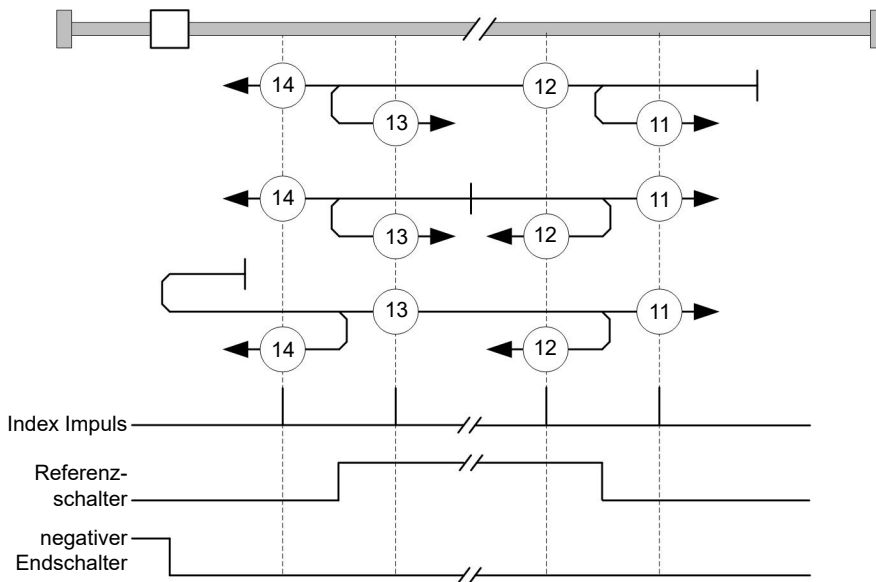
Referenzieren auf Referenzschalter und Index-Impuls (mit Endschaltern).

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

Die Methoden 7 bis 10 berücksichtigen den positiven Endschalter:



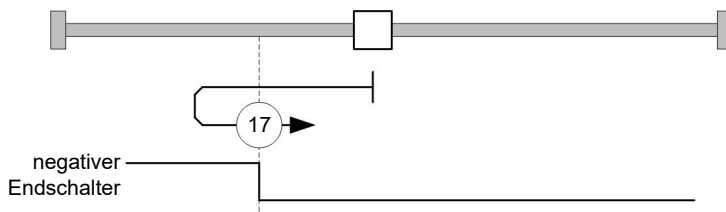
Die Methoden 11 bis 14 berücksichtigen den negativen Endschalter:



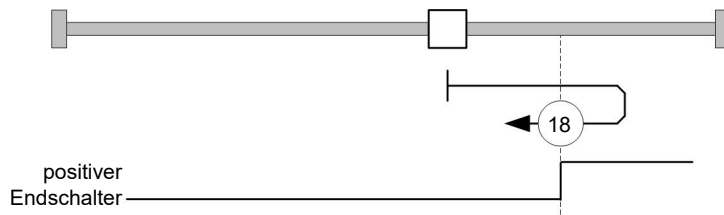
### Methoden 17 und 18

Referenzieren auf den Endschalter ohne den Index-Impuls.

Methode 17 referenziert auf den negativen Endschalter:



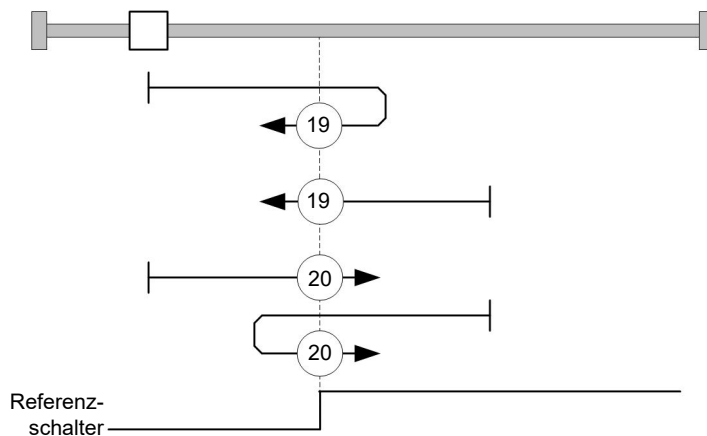
Methode 18 referenziert auf den positiven Endschalter:



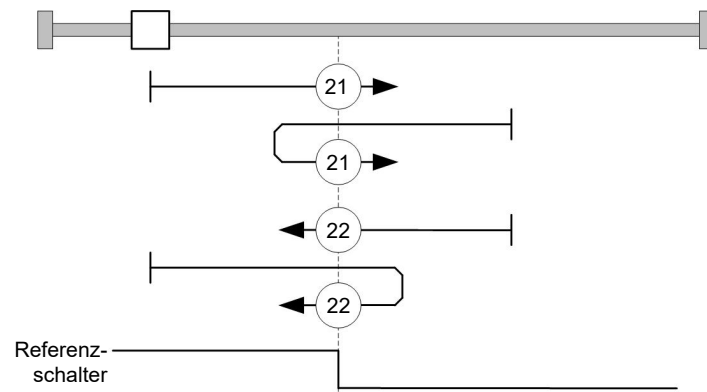
### Methoden 19 bis 22

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

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



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

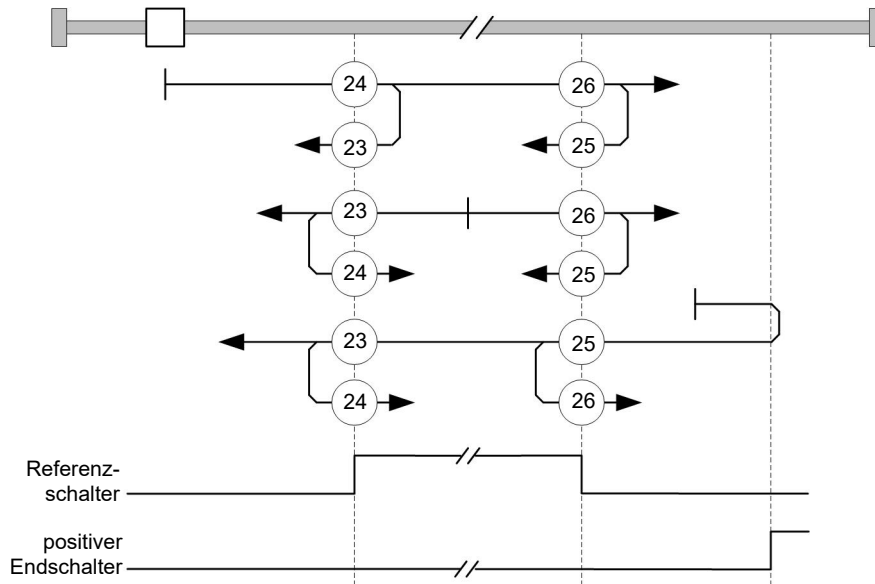


### Methoden 23 bis 30

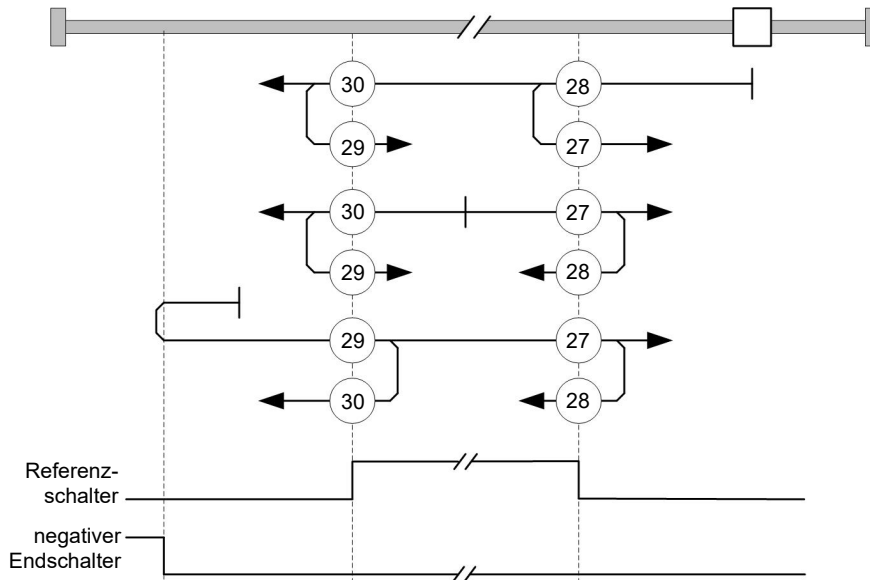
Referenzieren auf Referenzschalter ohne den Index-Impuls (mit Endschaltern).

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

Die Methoden 23 bis 26 berücksichtigen den positiven Referenzschalter:



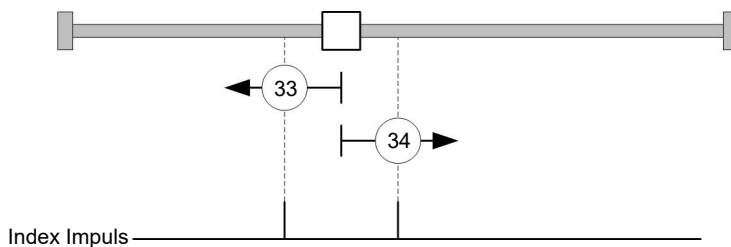
Die Methoden 27 bis 30 berücksichtigen den negativen Referenzschalter:



### Methoden 33 und 34

Referenzieren auf den nächsten Index-Impuls.

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



### Methode 35

Referenziert auf die aktuelle Position.





#### Hinweis

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

## 6.6 Interpolated Position Mode

### 6.6.1 Übersicht

#### Beschreibung

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

#### Synchronisierung zum SYNC-Objekt

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



#### Hinweis

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

### 6.6.2 Aktivierung

Um den Modus zu aktivieren, muss im Objekt **6060<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>**.

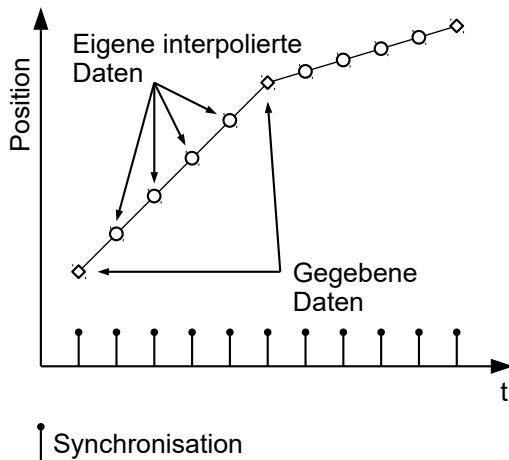
### 6.6.4 Statusword

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:

- **60C2<sub>h</sub>:01<sub>h</sub>**: Zeit zwischen zwei übergebenen Zielpositionen in ms.
- **60C4<sub>h</sub>:06<sub>h</sub>**: dieses Objekt ist auf "1" zu setzen um die Zielposition im Objekt **60C1<sub>h</sub>:01<sub>h</sub>** modifizieren zu dürfen.
- **6081<sub>h</sub>** (Profile Velocity): maximale Geschwindigkeit, mit der die Position angefahren werden soll
- Die Geschwindigkeit wird durch **607F<sub>h</sub>** (Max Profile Velocity) und **6080<sub>h</sub>** (Max Motor Speed) begrenzt, der kleinere Wert wird als Grenze herangezogen.
- Um den Motor drehen zu können, ist die *Power state machine* auf den Status *Operation enabled* zu setzen (siehe **CiA 402 Power State Machine**)

## 6.6.7 Operation

Nach dem Setup ist die Aufgabe der übergerodeten Steuerung, die Zielpositionen rechtzeitig in das Objekt **60C1<sub>h</sub>:01<sub>h</sub>** zu schreiben.

## 6.7 Cyclic Synchronous Position

### 6.7.1 Übersicht

#### Beschreibung

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

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



### Hinweis

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

### Aktivierung

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

### Controlword

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

### 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>607A<sub>h</sub></b> (Target Position) wird ignoriert
12	1	Steuerung folgt der Zielvorgabe, das Objekt <b>607A<sub>h</sub></b> (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<sub>h</sub>** eingegebenen Grenzwerte.

## 6.7.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **607A<sub>h</sub>** (Target Position): Dieses Objekt muss zyklisch mit dem Positions-Sollwert beschrieben werden.
- **607B<sub>h</sub>** (Position Range Limit): Dieses Objekt enthält die Vorgabe für einen Über- oder Unterlauf der Positionsangabe.
- **607D<sub>h</sub>** (Software Position Limit): Dieses Objekt legt die Limitierungen fest, innerhalb deren sich die Positionsvorgabe (**607A<sub>h</sub>**) befinden muss.
- **6065<sub>h</sub>** (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<sub>h</sub>**) außerhalb dieses Korridors, wird ein Schleppfehler gemeldet.
- **6066<sub>h</sub>** (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<sub>h</sub>**) befinden, wird ein Schleppfehler ausgelöst.
- **6085<sub>h</sub>** (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.
- **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:  $\text{Zykluszeit} = \text{Wert des } 60C2_{h:01_{h}} * 10^{\text{Wert des } 60C2_{h:02_{h}}} \text{ Sekunden.}$
- **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.

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

#### Beschreibung

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

#### Aktivierung

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

#### Controlword

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

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

- **605A<sub>h</sub>** (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<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 **60FF<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.

Folgende Objekte können in dem Modus ausgelesen werden:

- **606C<sub>h</sub>** (Velocity Actual Value)
- **607E<sub>h</sub>** (Polarity)

## 6.9 Cyclic Synchronous Torque

### 6.9.1 Übersicht

#### Beschreibung

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



#### Hinweis

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

#### Aktivierung

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

#### Controlword

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

#### 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>6071<sub>h</sub></b> (Target Torque) wird ignoriert
12	1	Steuerung folgt der Zielvorgabe, das Objekt <b>6071<sub>h</sub></b> (Target Torque) wird als Eingabe für die Positionsregelung genutzt.
13	0	Reserviert
13	1	Reserviert

## 6.9.2 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **6071<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.

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.

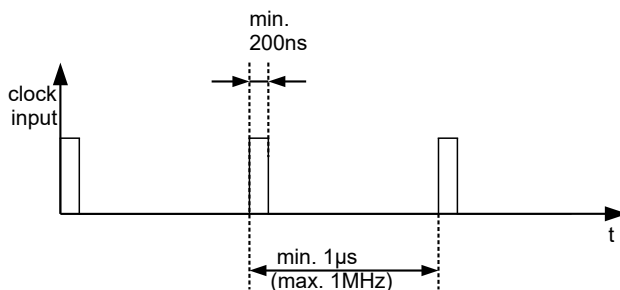
### 6.10.2 Aktivierung

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 Skalierung der Schritte erfolgt über die Objekte **2057<sub>h</sub>** und **2058<sub>h</sub>**. Dabei gilt die folgende Formel:

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

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



#### Hinweis

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

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



#### Hinweis

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

### 6.10.4 Statusword

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

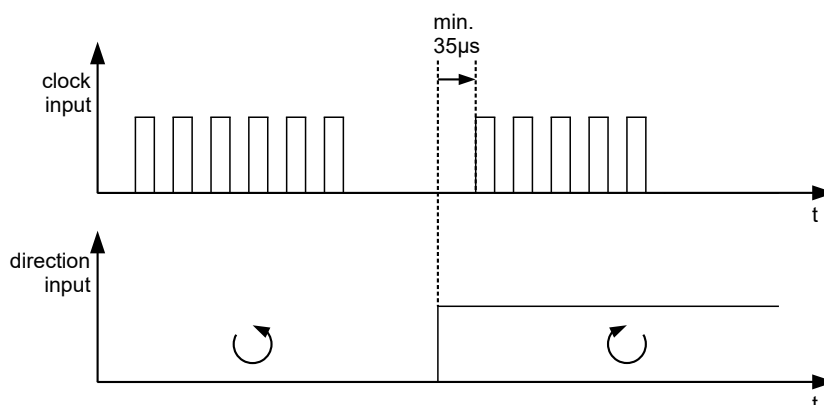
- Bit 13 (Following Error): Dieses Bit wird im *Closed Loop*-Betrieb gesetzt, wenn der Schlepfehler größer als die eingestellten Grenzen ist (**6065<sub>h</sub>** (Following Error Window) und **6066<sub>h</sub>** (Following Error Time Out)).

### 6.10.5 Unterarten des Takt-Richtungs-Modus

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

Um den Modus zu aktivieren muss das Objekt **205B<sub>h</sub>** auf den Wert "0" gesetzt sein (Werkseinstellung).

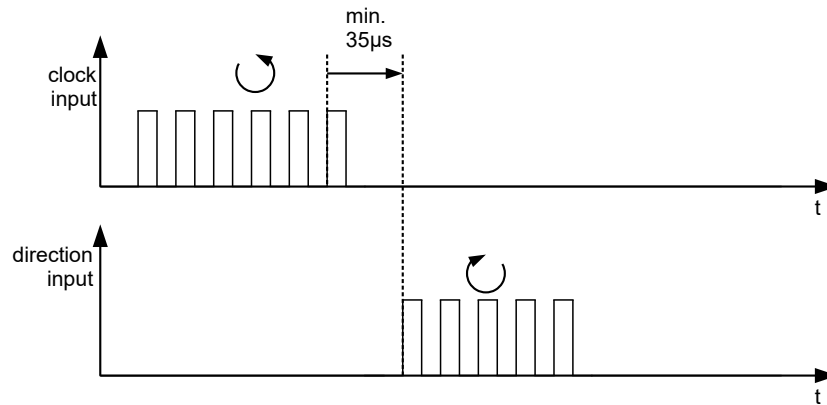
In diesem Modus müssen über den Takteingang die Pulse vorgegeben werden, das Signal des Richtungseingangs gibt dabei die Drehrichtung vor (siehe nachfolgende Grafik).



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

Um den Modus zu aktivieren muss das Objekt **205B<sub>h</sub>** auf den Wert "1" gesetzt sein.

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



## 6.11 Auto-Setup

### 6.11.1 Beschreibung

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

### 6.11.2 Aktivierung

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

### 6.11.3 Controlword

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

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

### 6.11.4 Statusword

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

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



## 7 Spezielle Funktionen

### 7.1 Digitale Ein- und Ausgänge

Diese Steuerung verfügt über digitale Ein- und Ausgänge.

#### 7.1.1 Bitzuordnung

Die Software der Steuerung ordnet jedem Eingang und Ausgang zwei Bits im jeweiligen Objekt (z.B. **60FD<sub>h</sub> Digital Inputs** bzw. **60FE<sub>h</sub> Digital Outputs**) zu:

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

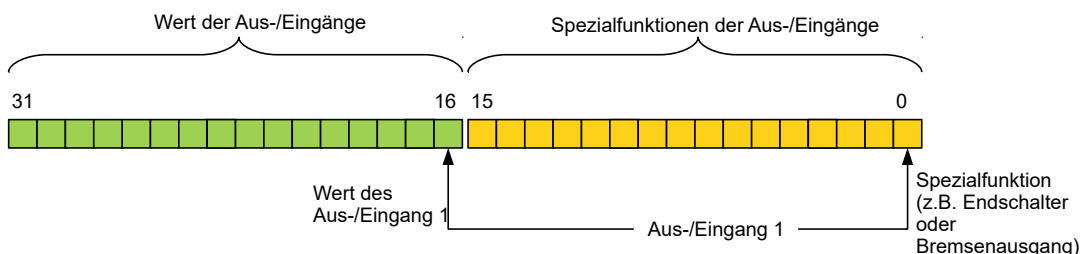
#### Beispiel

Um den Wert des Ausgangs 2 zu manipulieren, ist immer Bit 17 in **60FE<sub>h</sub>** zu benutzen.

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

In der nachfolgenden Zeichnung ist diese Zuordnung graphisch dargestellt.

Bits eines beliebigen Objektes zur Steuerung eines Aus-/Eingangs



#### 7.1.2 Digitale Eingänge

##### Übersicht



##### Hinweis

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



##### Hinweis

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

Folgende Eingänge stehen zur Verfügung:

Eingan	Sonderfunktion	Schaltswelle umschaltbar	Differenziell / single-ended
1	Negativer Endschalter/ Taktingang im Takt- Richtungs-Modus	ja, 5 V oder 24 V	single-ended
2	Positiver Endschalter/ Richtungseingang im Takt- Richtungs-Modus	ja, 5 V oder 24 V	single-ended
3	Referenzschalter	ja, 5 V oder 24 V	single-ended
4	keine	ja, 5 V oder 24 V	single-ended
5	keine	ja, 5 V oder 24 V	single-ended

## Objekteinträge

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

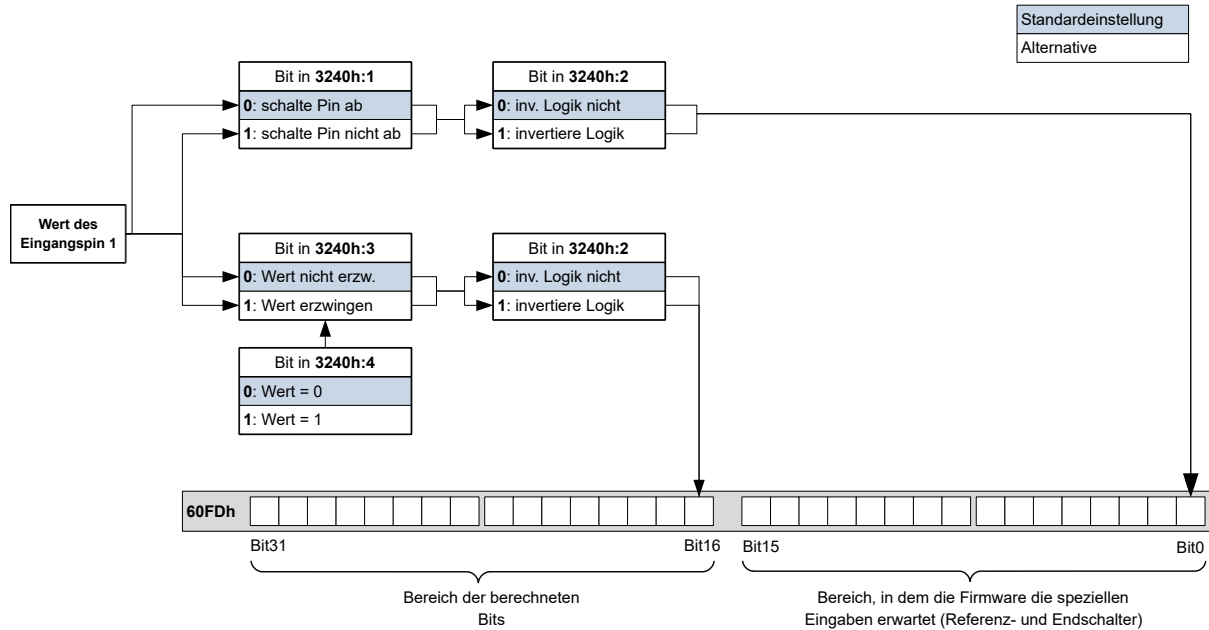
- **3240<sub>h</sub>:01<sub>h</sub>** (Special Function Enable): Dieses Bit erlaubt Sonderfunktionen eines Eingangs aus- (Wert "0") oder einzuschalten (Wert "1"). Soll Eingang 1 z.B. nicht als negativer Endschalter verwendet werden, so muss die Sonderfunktion abgeschaltet werden, damit nicht fälschlicherweise 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

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>:06<sub>h</sub>** (Input Range Select): Damit können Eingänge - welche über diese Funktion verfügen - von der Schaltswelle von 5 V (Bit auf "0") auf die Schaltswelle 24 V (Bit auf "1") umgeschaltet werden. Bit 0 entspricht dabei dem Eingang 1, Bit 1 dem Eingang 2 usw..
- **60FD<sub>h</sub>** (Digital Inputs): Dieses Objekt enthält eine Zusammenfassung der Eingänge und der Spezialfunktionen.

## Verrechnung der Eingänge

Verrechnung des Eingangssignals am Beispiel von Eingang 1:

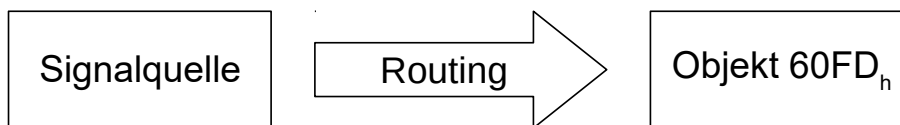
Der Wert an Bit 0 des Objekts **60FD<sub>h</sub>** wird von der Firmware als negativer Endschalter interpretiert, das Ergebnis der vollständigen Verrechnung wird in Bit 16 abgelegt.



## Input Routing

### Prinzip

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



### Aktivierung

Dieser Modus wird aktiviert, indem das Objekt 3240<sub>h</sub>:08<sub>h</sub> (Routing Enable) auf 1 gesetzt wird.



#### Hinweis

Die Einträge 3240<sub>h</sub>:01<sub>h</sub> bis 3240<sub>h</sub>:04<sub>h</sub> haben dann **keine** Funktion mehr, bis das Eingangsrouting wieder abgeschaltet wird.



#### Hinweis

Wird das *Input Routing* eingeschaltet, werden initial die Werte des 3242<sub>h</sub> 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<sub>h</sub> bestimmt, welche Signalquelle auf welches Bit des 60FD<sub>h</sub> geroutet wird. Der Subindex 01<sub>h</sub> des 3242<sub>h</sub> bestimmt Bit 0, Subindex 02<sub>h</sub> das Bit 1, und so weiter. Die Signalquellen und deren Nummern finden Sie in den nachfolgenden Listen.

Nummer		
dec	hex	Signalquelle
00	00	Signal ist immer 0
01	01	Physikalischer Eingang 1
02	02	Physikalischer Eingang 2
03	03	Physikalischer Eingang 3
04	04	Physikalischer Eingang 4
05	05	Physikalischer Eingang 5
06	06	Physikalischer Eingang 6
07	07	Physikalischer Eingang 7
08	08	Physikalischer Eingang 8
09	09	Physikalischer Eingang 9
10	0A	Physikalischer Eingang 10
11	0B	Physikalischer Eingang 11
12	0C	Physikalischer Eingang 12
13	0D	Physikalischer Eingang 13
14	0E	Physikalischer Eingang 14
15	0F	Physikalischer Eingang 15
16	10	Physikalischer Eingang 16
65	41	Hall Eingang "U"
66	42	Hall Eingang "V"
67	43	Hall Eingang "W"
68	44	Encoder Eingang "A"
69	45	Encoder Eingang "B"
70	46	Encoder Eingang "Index"
71	47	USB Power Signal
72	48	Status "Ethernet aktiv"

Die nachfolgende Tabelle beschreibt die invertierten Signale der vorherigen Tabelle.

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

Nummer		
dec	hex	Signalquelle
142	8E	Invertierter physikalischer Eingang 14
143	8F	Invertierter physikalischer Eingang 15
144	90	Invertierter physikalischer Eingang 16
193	C1	Invertierter Hall Eingang "U"
194	C2	Invertierter Hall Eingang "V"
195	C3	Invertierter Hall Eingang "W"
196	C4	Invertierter Encoder Eingang "A"
197	C5	Invertierter Encoder Eingang "B"
198	C6	Invertierter Encoder Eingang "Index"
199	C7	Invertiertes USB Power Signal
200	C8	Invertierter Status "Ethernet aktiv"

### Beispiel

Es soll der Eingang 1 auf Bit 16 des Objekts **60FD<sub>h</sub>** geroutet werden:

Die Nummer der Signalquelle für Eingang 1 ist die "1". Das Routing für Bit 16 wird in das **3242<sub>h</sub>:11<sub>h</sub>** geschrieben.

Demnach muss das Objekt **3242<sub>h</sub>:11<sub>h</sub>** auf den Wert "1" gesetzt werden.

## 7.1.3 Digitale Ausgänge

### Ausgänge

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

### Beschaltung



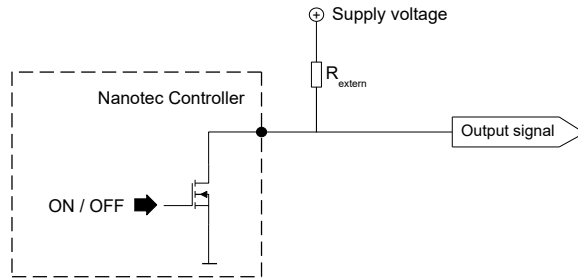
#### Hinweis

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

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

### Beispiel

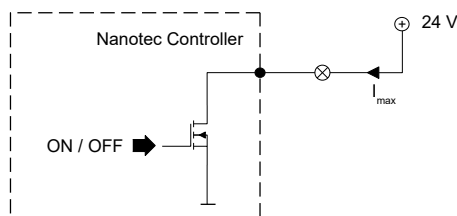
Es soll das digitale Ausgangssignal weiter verwendet werden. Dazu ist eine Beschaltung wie im nachfolgenden Bild zu realisieren.



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.



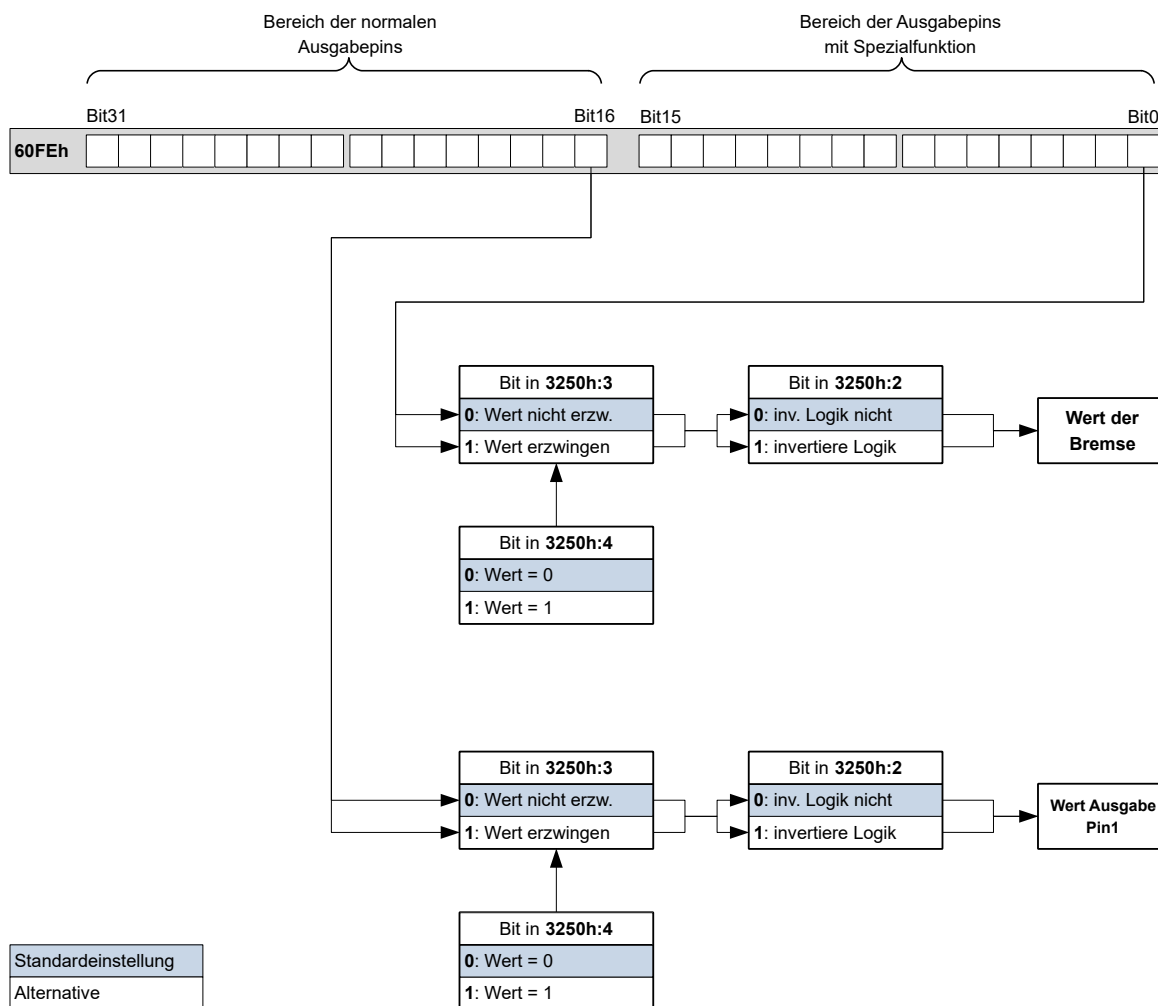
### Objekteinträge

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<sub>h</sub>:01<sub>h</sub>**: 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>**: Dieses Objekt besitzt keine Funktion und ist aus Gründen der Kompatibilität enthalten.

### Verrechnung der Ausgänge

Beispiel für die Verrechnung der Bits für die Ausgänge:

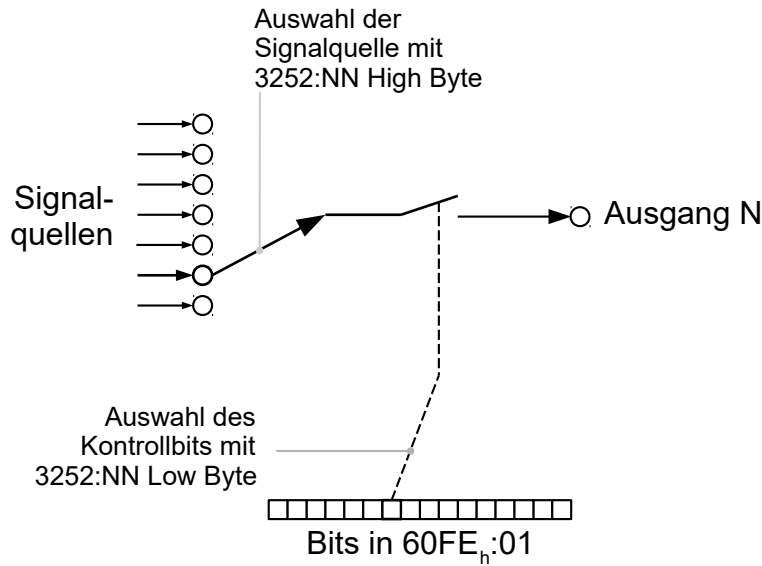


## Output Routing

### Prinzip

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

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



### Aktivierung

Dieser Modus wird aktiviert, indem das Objekt **3250<sub>h</sub>:08<sub>h</sub>** (Routing Enable) auf 1 gesetzt wird.



#### 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 "Ausgangsrouting" wieder abgeschaltet wird.

### Routing

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

Subindex 3252 <sub>h</sub>	Output Pin
01 <sub>h</sub>	Konfiguration des PWM-Ausgangs (falls verfügbar)
02 <sub>h</sub>	Konfiguration des Ausgangs 1
03 <sub>h</sub>	Konfiguration des Ausgangs 2 (falls verfügbar)
04 <sub>h</sub>	Konfiguration des Ausgangs 3 (falls verfügbar)
05 <sub>h</sub>	Konfiguration des Ausgangs 4 (falls verfügbar)



#### Hinweis

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

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

Bit 7 von **3252<sub>h</sub>:01<sub>h</sub>** bis **05** invertiert die Steuerung aus dem Objekt **60FE<sub>h</sub>:01**. Normalerweise schaltet der Wert "1" im Objekt **60FE<sub>h</sub>:01** das Signal "ein", ist das Bit 7 gesetzt, schaltet der Wert "0" das Signal ein.



**Nummer in 3252:01 bis 05**

00XX <sub>h</sub>	Ausgang ist immer "1"
01XX <sub>h</sub>	Ausgang ist immer "0"
02XX <sub>h</sub>	Encodersignal ( <b>6063<sub>h</sub></b> ) mit Frequenzteiler 1
03XX <sub>h</sub>	Encodersignal ( <b>6063<sub>h</sub></b> ) mit Frequenzteiler 2
04XX <sub>h</sub>	Encodersignal ( <b>6063<sub>h</sub></b> ) mit Frequenzteiler 4
05XX <sub>h</sub>	Encodersignal ( <b>6063<sub>h</sub></b> ) mit Frequenzteiler 8
06XX <sub>h</sub>	Encodersignal ( <b>6063<sub>h</sub></b> ) mit Frequenzteiler 16
07XX <sub>h</sub>	Encodersignal ( <b>6063<sub>h</sub></b> ) mit Frequenzteiler 32
08XX <sub>h</sub>	Encodersignal ( <b>6063<sub>h</sub></b> ) mit Frequenzteiler 64
09XX <sub>h</sub>	Position Actual Value ( <b>6064<sub>h</sub></b> ) mit Frequenzteiler 1
0AXX <sub>h</sub>	Position Actual Value ( <b>6064<sub>h</sub></b> ) mit Frequenzteiler 2
0BXX <sub>h</sub>	Position Actual Value ( <b>6064<sub>h</sub></b> ) mit Frequenzteiler 4
0CXX <sub>h</sub>	Position Actual Value ( <b>6064<sub>h</sub></b> ) mit Frequenzteiler 8
0DXX <sub>h</sub>	Position Actual Value ( <b>6064<sub>h</sub></b> ) mit Frequenzteiler 16
0EXX <sub>h</sub>	Position Actual Value ( <b>6064<sub>h</sub></b> ) mit Frequenzteiler 32
0FXX <sub>h</sub>	Position Actual Value ( <b>6064<sub>h</sub></b> ) mit Frequenzteiler 64
10XX <sub>h</sub>	PWM-Signal, das mit Objekt <b>2038<sub>h</sub>:05<sub>h</sub></b> und <b>06<sub>h</sub></b> konfiguriert wird
11XX <sub>h</sub>	Invertiertes PWM-Signal, das mit Objekt <b>2038<sub>h</sub>:05<sub>h</sub></b> und <b>06<sub>h</sub></b> konfiguriert wird

**Beispiel**

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

- **3250<sub>h</sub>:08<sub>h</sub>** = 1 (Routing aktivieren)
- **3252<sub>h</sub>:02<sub>h</sub>** = 0405<sub>h</sub> (04XX<sub>h</sub> + 0005<sub>h</sub>) Dabei ist:
  - 04XX<sub>h</sub>: Encodersignal mit Frequenzteiler 4
  - 0005<sub>h</sub>: Auswahl von Bit 5 des **60FE:01**

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

**Beispiel**

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

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

## 7.2 Automatische Bremsensteuerung

### 7.2.1 Beschreibung

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

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

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

### 7.2.2 Aktivierung und Anschluss

Die Bremse kann entweder automatisch oder manuell gesteuert werden:

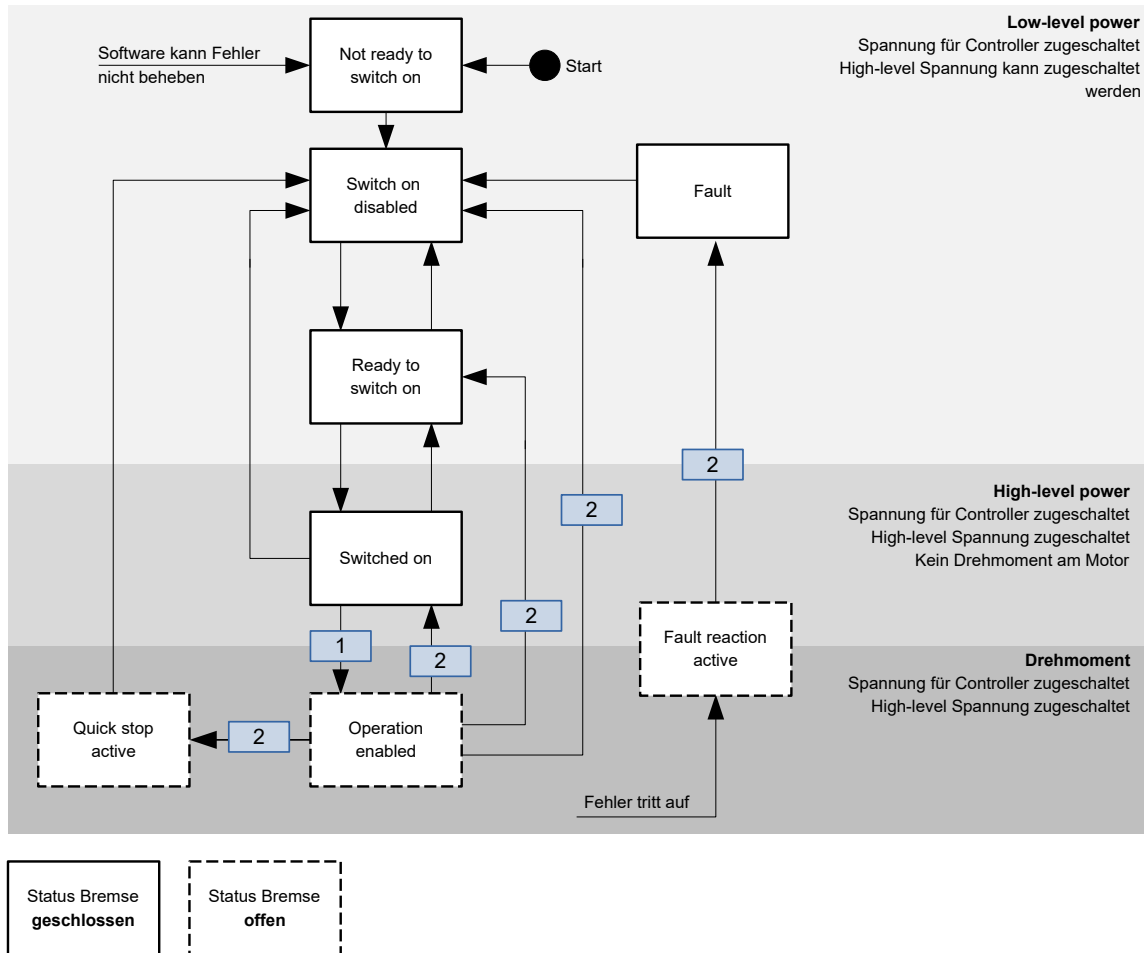
- Automatisch: Bit 2 des Objekts **3202<sub>h</sub>** auf "1" setzen aktiviert die Bremsensteuerung.
- Manuell: Bit 2 des Objekts **3202<sub>h</sub>** auf "0" setzen deaktiviert die Bremsensteuerung, die Bremse lässt sich jetzt mit dem Bit 0 im Objekt **60FE<sub>h</sub>:01<sub>h</sub>** kontrollieren.

#### Anschluss

Der Bremsenausgang befindet sich am Stecker X4 (siehe Kapitel "**Stecker X4 - Bremsen-Anschluss**")

### 7.2.3 Steuerung der Bremse

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



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

1. Der Motorstrom wird eingeschaltet.
2. Die Zeit, welche in **2038<sub>h</sub>:3<sub>h</sub>** hinterlegt wird, wird abgewartet.
3. Die Bremse löst sich.
4. Die Zeit, welche in **2038<sub>h</sub>:4<sub>h</sub>** 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<sub>h</sub>:1<sub>h</sub>** hinterlegt wird, wird abgewartet.
3. Die Bremse wird aktiviert.
4. Die Zeit, welche in **2038<sub>h</sub>:2<sub>h</sub>** hinterlegt wird, wird abgewartet.
5. Der Motorstrom wird abgeschaltet.

## 7.2.4 Bremsen-PWM

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

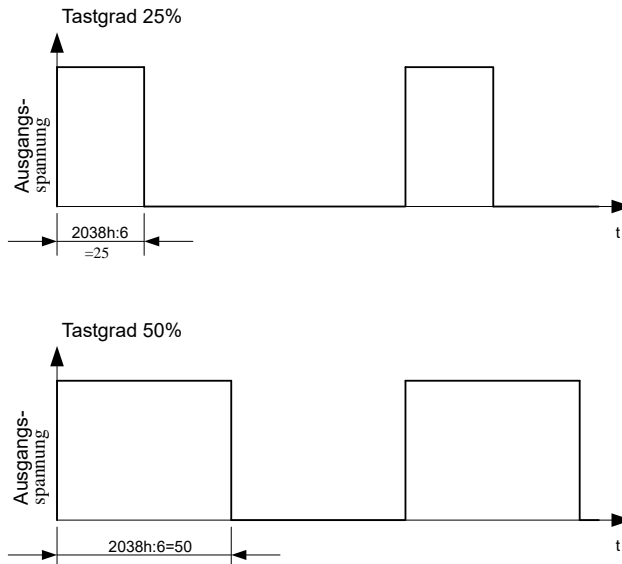
### Frequenz

Die Frequenz der Bremsen-PWM kann im Objekt **2038<sub>h</sub>:5<sub>h</sub>** eingestellt werden. Die Einheit ist Hertz, ein Wert größer 20000 ist nicht möglich.

## Tastgrad

Der Tastgrad - das Verhältnis Impuls- zu Periodendauer - wird im **2038<sub>h</sub>:6<sub>h</sub>** 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.



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

### 7.3.1 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 **3202<sub>h</sub>** muss auf "1" gesetzt sein).

### 7.3.2 Objekteinträge

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

- **2031<sub>h</sub>**: Max Motor Current - Gibt den maximal zulässigen Motorstrom in mA an.
- **203B<sub>h</sub>:1<sub>h</sub>** Motor Rated Current - Gibt den Nennstrom in mA an.
- **6073<sub>h</sub>** Max Current - Gibt den Maximalstrom in Promille des eingestellten Nennstroms an.
- **203B<sub>h</sub>:2<sub>h</sub>** 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:

- **203B<sub>h</sub>:3<sub>h</sub>** Threshold - Gibt die Grenze in mAs an, von der abhängt, ob auf Maximalstrom oder Nennstrom geschaltet wird.
- **203B<sub>h</sub>:4<sub>h</sub>** CalcValue - Gibt den berechneten Wert an, welcher mit Threshold verglichen wird, um den Strom einzustellen.

- **203B<sub>h</sub>:5<sub>h</sub> LimitedCurrent** - Zeigt den gegenwärtigen Stromwert an, der von  $I^2t$  eingestellt wurde.
- **203B<sub>h</sub>:6<sub>h</sub> Status**:
  - Wert = "0":  $I^2t$  deaktiviert
  - Wert = "1":  $I^2t$  aktiviert

### 7.3.3 Aktivierung

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

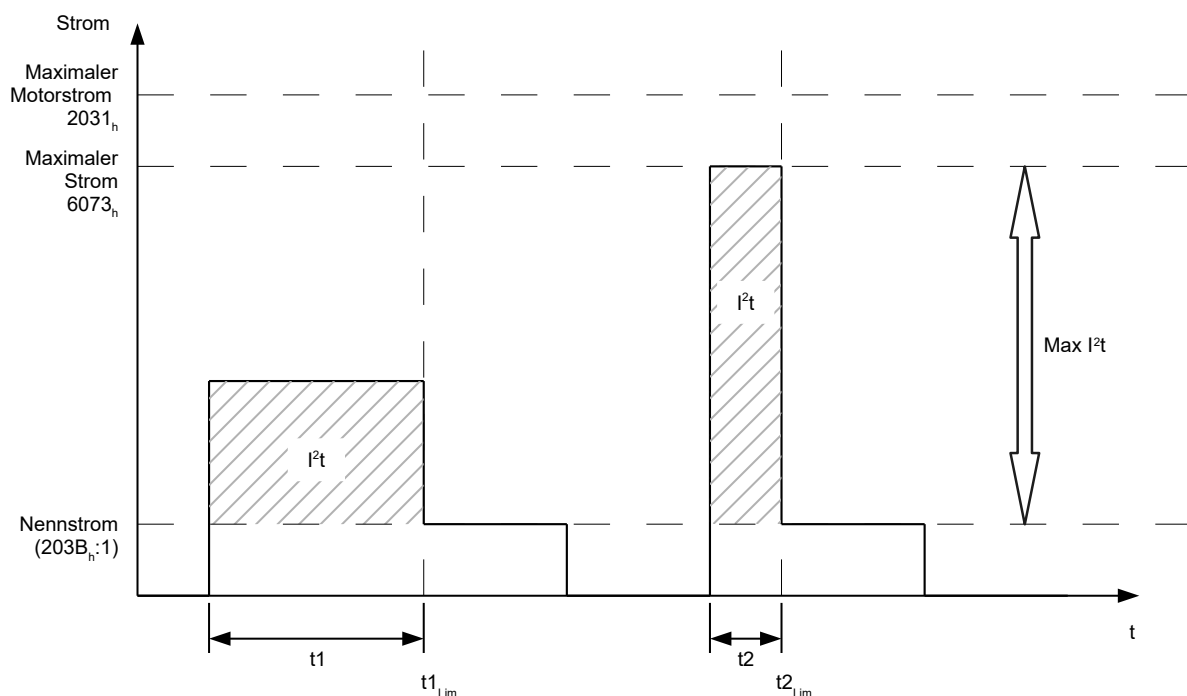
Zum Aktivieren des Modus müssen Sie die vier oben genannten Objekteinträge (**2031<sub>h</sub>**, **6073<sub>h</sub>**, **203B<sub>h</sub>:1<sub>h</sub>**, **203B<sub>h</sub>:2<sub>h</sub>**) sinnvoll beschreiben. Das bedeutet, dass der Maximalstrom größer als der Nennstrom sein muss und ein Zeitwert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

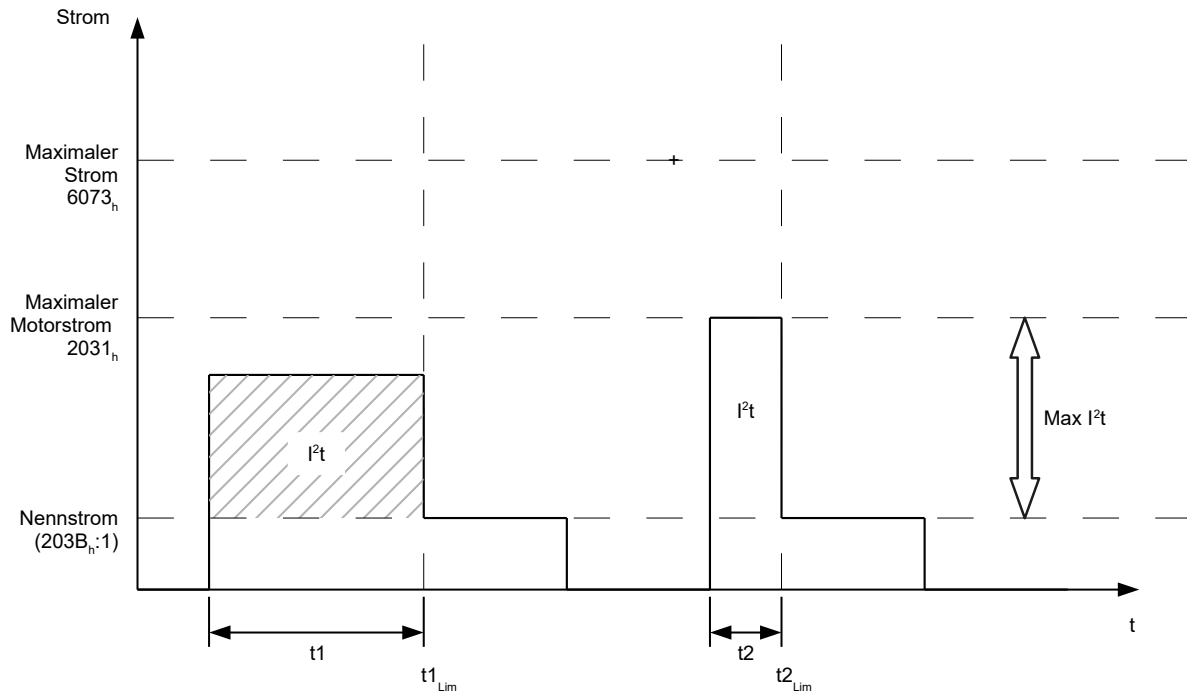
### 7.3.4 Funktion von $I^2t$

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

Der Motor kann solange mit Maximalstrom laufen, bis das berechnete  $I^2T_{Lim}$  erreicht wird. Darauf folgend wird der Strom sofort auf Nennstrom gesenkt. Der Maximalstrom wird durch den maximalen Motorstrom (**2031<sub>h</sub>**) 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  $t_1$ .

## 7.4 Objekte speichern



### Hinweis

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



### Hinweis

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

### 7.4.1 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.
- Ethernet: Parameter mit Bezug auf die Ethernet-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.

### 7.4.2 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

### 7.4.3 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
- **2291<sub>h</sub>**: PDI Input
- **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
- **3221<sub>h</sub>**: Analogue 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>**: Analogue Input Offset
- **3322<sub>h</sub>**: Analogue Input Pre-scaling
- **3700<sub>h</sub>**: Deviation Error Option Code
- **3701<sub>h</sub>**: Limit Switch Error Option Code
- **4013<sub>h</sub>**: HW Configuration
- **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
- **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

#### 7.4.4 Kategorie: Benutzer

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

#### 7.4.5 Kategorie: Bewegung

- **3202<sub>h</sub>**: Motor Drive Submode Select
- **3203<sub>h</sub>**: Feedback Selection
- **6073<sub>h</sub>**: Max Current
- **6080<sub>h</sub>**: Max Motor Speed

#### 7.4.6 Kategorie: Tuning

- **2030<sub>h</sub>**: Pole Pair Count
- **2031<sub>h</sub>**: Max Motor Current
- **203B<sub>h</sub>**: I2t Parameters
- **2059<sub>h</sub>**: Encoder Configuration
- **3390<sub>h</sub>**: Feedback Hall
- **33A0<sub>h</sub>**: Feedback Incremental A/B/I 1
- **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

#### 7.4.7 Kategorie: Modbus RTU

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

#### 7.4.8 Kategorie: Ethernet

- **2010<sub>h</sub>**: IP-Configuration
- **2011<sub>h</sub>**: Static-IPv4-Address
- **2012<sub>h</sub>**: Static-IPv4-Subnet-Mask
- **2013<sub>h</sub>**: Static-IPv4-Gateway-Address

#### 7.4.9 Speichervorgang starten



#### VORSICHT

##### **Unkontrollierte Motorbewegungen!**

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

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



### Hinweis

- Das Speichern kann einige Sekunden dauern. Unterbrechen Sie während des Speicherns keinesfalls die Spannungsversorgung. Andernfalls ist der Stand der gespeicherten Objekte undefiniert.
- Warten Sie immer, dass die Steuerung das erfolgreiche Speichern mit dem Wert "1" in dem entsprechenden Subindex im Objekt **1010<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 06 <sub>h</sub> (Tuning) und 0C <sub>h</sub> (Ethernet)
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
0C <sub>h</sub>	Ethernet

## 7.4.10 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 0C <sub>h</sub> (Ethernet)
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
0C <sub>h</sub>	Ethernet

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

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

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



#### 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 eine erneutes Auto-Setup nicht notwendig wird). Sie können diese Objekte mit Subindex 06<sub>h</sub> zurücksetzen.
- Die Objekte der *Kategorie* 0C<sub>h</sub> (Ethernet) werden mittels Subindex 01<sub>h</sub> nicht zurückgesetzt.

### 7.4.11 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.

## 8 Modbus TCP

Die Steuerung lässt sich mittels Modbus TCP ansteuern. In diesem Kapitel werden die Funktionscodes der Modbus-Kommunikationsstruktur beschrieben.

Modbus-Referenzen: **www.modbus.org**.

- *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3*, Date: 26.04.2014, Version: 1.1b3
- *MODBUS Messaging on TCP/IP Implementation Guide V1.0b*, Date: 24.10.2006, Version: 1.0b

Die Nachrichten werden alle über TCP an den Port 502 der Steuerung geschickt, es wird nur eine Verbindung unterstützt. Eine CRC (wie es bei Modbus RTU benutzt wird) entfällt.

Die I/O-Daten mit den ggf. vorkonfigurierten Antriebsgrößen (siehe **Prozessdatenobjekte (PDO)**) können mit den Standard Modbus-Funktionscodes gesendet werden. Um aber eigene I/O-Daten zu konfigurieren, muss der Funktionscode 2Bh (CAN Encapsulation) vom Modbus-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 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:

MBAP	FC	Daten
00 00 00 00 00 11 00	2B	0D 01 00 01 23 01 00 00 00 00 04 78 56 34 12

#### MBAP

Modbus Application Protocol Header (siehe **MBAP Header** für Details)

#### FC

Funktionscode

#### Daten

Datenbereich, Decodierung ist abhängig vom benutzen Funktionscode

### 8.2 MBAP Header

Bei Modbus TCP wird ein *Modbus Application Protocol Header* (kurz *MBAP Header*) vor die eigentliche Nachricht gestellt.

MBAP Header	Function code	Daten
-------------	---------------	-------

Dieser Header besteht aus folgenden Teilen:

Name	Länge	Wert
Transaction Identifier	2 Byte	
Protocol Identifier	2 Bytes	0000 <sub>h</sub> (Modbus)
Länge	2 Byte	
Unit Identifier	1 Bytes	00 <sub>h</sub>

Der MBAP Header ist 7 Byte lang:

#### Transaction Identifier

Wird benutzt für eine Transaktionspaarung, der Server (die Steuerung) kopiert den Wert aus der Anfrage (Request) des Clients in die Antwort (Response). Wenn der Client die Nummer bei jeder Anfrage erhöht, kann die Antwort eindeutig der Anfrage zugewiesen werden.

#### Protocol Identifier

Da ein Modbus Protokoll benutzt wird, hat das Feld immer den Wert 0.

#### Length

Die Länge der Daten inklusive des Felds *Unit Identifier* (1 Bytes), *Funktionscode* (1 Byte) und der Daten.

#### Unit Identifier

Dieses Feld wird zum internen Systemrouting benutzt. Da die Steuerung kein Routing unterstützt, besitzt das Feld immer den Wert 0.

## 8.3 Funktionscodes

Die folgenden "Funktionscodes" werden unterstützt:

	Name	Funktionscode	Unterfunktionscode
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	22 (16 <sub>h</sub> )	
	Read/Write Multiple Registers	23 (17 <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> )

## 8.4 Funktionscode-Beschreibungen

### 8.4.1 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
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0006 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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> )

Response ("M" entspricht der Anzahl der zu lesenden Register)		
Name	Länge	Wert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0003 <sub>h</sub> + 2*M
Unit Identifier	1 Byte	00 <sub>h</sub>
Funktionscode	1 Byte	03 <sub>h</sub> / 04 <sub>h</sub>
Anzahl Bytes	1 Byte	2 * M
Registerwert	2 Bytes	

Fehler		
Name	Länge	Wert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0003 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
Fehlercode	1 Byte	83 <sub>h</sub> / 84 <sub>h</sub>
Ausnahmecode	1 Byte	01, 02, 03 oder 04

### Beispiel

Nachfolgend ein Beispiel eines Lese-Request und Response des Registers 5000 (1388<sub>h</sub>) und des folgenden Registers (2 Register):

#### Request

MBAP	FC	Daten
00 00 00 00 00 06 00	03	13 88 00 02

#### Response

MBAP	FC	Daten
00 00 00 00 00 07 00	03	04 02 40 00 00

## 8.4.2 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
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0006 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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>

Response		
Name	Länge	Wert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0006 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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>

Fehler		
Name	Länge	Wert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0003 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
Fehlercode	1 Byte	86 <sub>h</sub>
Ausnahmecode	1 Byte	01, 02, 03 oder 04

### 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

MBAP	FC	Daten
00 00 00 00 00 06 00	06	17 70 00 01

#### Response

MBAP	FC	Daten
00 00 00 00 00 06 00	06	17 70 00 01

### 8.4.3 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
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0007 <sub>h</sub> + N * 2
Unit Identifier	1 Byte	00 <sub>h</sub>
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	

Response		
Name	Länge	Wert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0006 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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>

Fehler		
Name	Länge	Wert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>



Fehler		
Name	Länge	Wert
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0003 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
Fehlercode	1 Byte	90 <sub>h</sub>
Ausnahmecode	1 Byte	01, 02, 03 oder 04

### 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

MBAP	FC	Daten
00 00 00 00 00 0B 00	10	17 70 00 02 04 01 02 03 04

#### Response

MBAP	FC	Daten
00 00 00 00 00 06 00	10	17 70 00 02

## 8.4.4 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
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	000B <sub>h</sub> + 2 * N
Unit Identifier	1 Byte	00 <sub>h</sub>
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	

Response ("M" entspricht der Anzahl der zu schreibenden Bytes):		
Name	Länge	Wert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>

Response ("M" entspricht der Anzahl der zu schreibenden Bytes):		
Name	Länge	Wert
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0003 <sub>h</sub> + 2 * M
Unit Identifier	1 Byte	00 <sub>h</sub>
Funktionscode	1 Byte	17 <sub>h</sub>
Anzahl Bytes	1 Byte	2 * M
Gelesene Register	M * 2 Bytes	

Fehler		
Name	Länge	Wert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0003 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
Fehlercode	1 Byte	97 <sub>h</sub>
Ausnahmecode	1 Byte	01, 02, 03 oder 04

### 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

MBAP	FC	Daten
00 00 00 00 00 0F 00	17	13 88 00 02 17 70 00 02 04 01 02 03 04

#### Response

MBAP	FC	Daten
00 00 00 00 00 07 00	17	04 02 40 00 00

## 8.4.5 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:

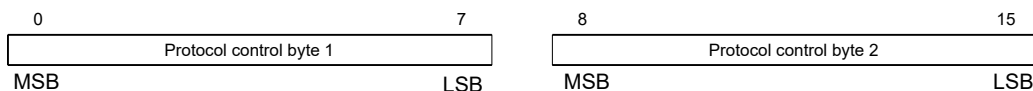
Name	Länge	Beispiel/Zahlenbereich
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	00NN <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
Funktionscode	1 Byte	2B <sub>h</sub> (43 <sub>d</sub> )
MEI type	1 Byte	0D <sub>h</sub> (13 <sub>d</sub> )
Protokolloptionen Bereich	2 bis 5 Byte	
Adressen- und Datenbereich	N Bytes	

### Protokolloptionen Bereich

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

### Protokoll-Kontrolle:

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



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

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

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

### Adressen- und Datenbereich

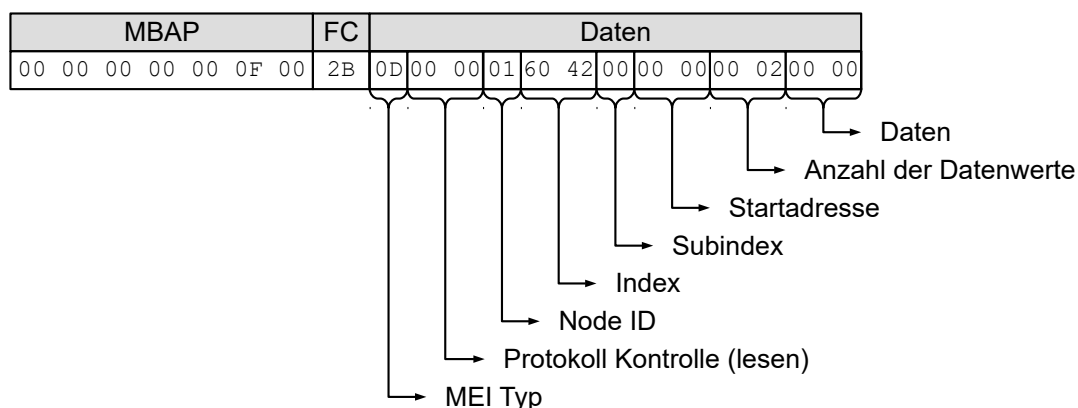
Der Adressen- und Datenbereich ist in der folgenden Tabelle definiert:

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	
Anzahl der Datenwerte	1 Byte, high	0000 <sub>h</sub> bis 00FD <sub>h</sub>
	1 Byte, low	
Schreib-/Lesedaten	n Byte	Die Daten sind codiert wie in Kapitel <b>Allgemeines</b> beschrieben.

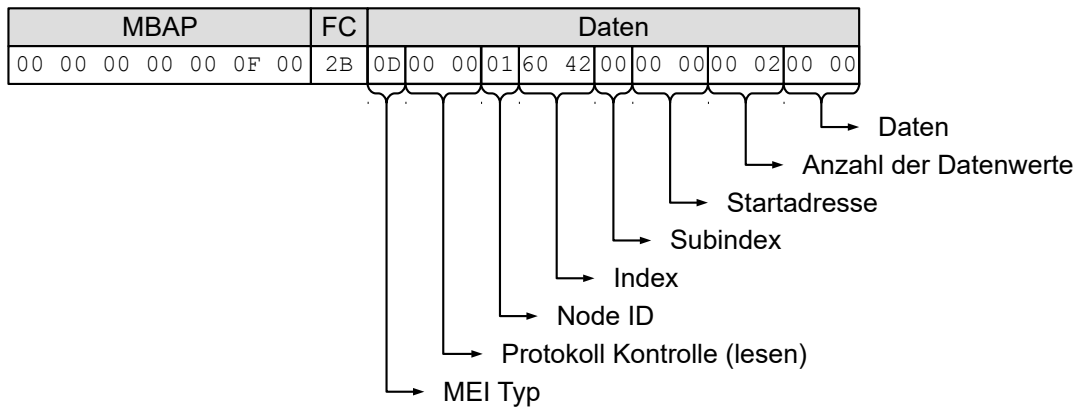
### Beispiel:

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).

### 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**

MBAP	FC	Daten
00 00 00 00 00 0E 00	2B	0D 01 00 01 60 60 00 00 00 00 01 02

**Response**

MBAP	FC	Daten
00 00 00 00 00 0D 00	2B	0D 01 00 01 60 60 00 00 00 00 00

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

**Request**

MBAP	FC	Daten
00 01 00 00 00 12 00	2B	0D 01 00 01 20 31 00 00 00 00 04 E8 03 00 00

**Response**

MBAP	FC	Daten
00 01 00 00 00 0D 00	2B	0D 01 00 01 20 31 00 00 00 00 00

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

**Request**

MBAP	FC	Daten
00 02 00 00 00 0F 00	2B	0D 01 00 01 60 40 00 00 00 00 02 00 00

**Response**

MBAP	FC	Daten
00 02 00 00 00 0D 00	2B	0D 01 00 01 60 40 00 00 00 00 00

**Setze 6040 = "80<sub>h</sub>"**

**Request**

MBAP	FC	Daten
00 03 00 00 00 0F 00	2B	0D 01 00 01 60 40 00 00 00 00 02 80 00

**Response**

MBAP	FC	Daten
00 03 00 00 00 0D 00	2B	0D 01 00 01 60 40 00 00 00 00 00

**Setze 6040 = "06<sub>h</sub>"**

**Request**

MBAP	FC	Daten
00 04 00 00 00 0F 00	2B	0D 01 00 01 60 40 00 00 00 00 02 06 00

**Response**

MBAP	FC	Daten
00 04 00 00 00 0D 00	2B	0D 01 00 01 60 40 00 00 00 00 00

**Setze 6040 = "07<sub>h</sub>"**

**Request**

MBAP	FC	Daten
00 05 00 00 00 0F 00	2B	0D 01 00 01 60 40 00 00 00 00 02 07 00

**Response**

MBAP	FC	Daten
00 05 00 00 00 0D 00	2B	0D 01 00 01 60 40 00 00 00 00 00

**Setze 6040 = "0F<sub>h</sub>"**

**Request**

MBAP	FC	Daten
00 06 00 00 00 0F 00	2B	0D 01 00 01 60 40 00 00 00 00 02 0F 00

### Response

MBAP	FC	Daten
00 06 00 00 00 0D 00	2B	0D 01 00 01 60 40 00 00 00 00 00

Nachfolgend zwei Beispiele zum Lesen eines Objektes:

**Lese 6041<sub>h</sub>:00<sub>h</sub>**

#### Request

MBAP	FC	Daten
00 00 00 00 00 0D 00	2B	0D 00 00 01 60 41 00 00 00 00 02

### Response

**Lese 6061<sub>h</sub>:00<sub>h</sub>**

#### Request

MBAP	FC	Daten
00 00 00 00 00 0D 00	2B	0D 00 00 01 60 61 00 00 00 00 01

### Response

MBAP	FC	Daten
00 00 00 00 00 0D 00	2B	0D 00 00 01 60 61 00 00 00 00 01 00

## Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlermeldung gesendet:

Name	Länge	Beispielwert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	000B <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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

CANopen-Fehlercode	Beschreibung
FFFF0000 <sub>h</sub>	Abort no error

CANopen-Fehlercode	Beschreibung
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
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0008/0009 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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

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.



Bit	Name	Beschreibung
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  $6061_{\text{h}}:00$  mit der Länge von 2 Byte, das Objekt ist aber nur 1 Byte groß:

**Request**

MBAP	FC	Daten
00 00 00 00 00 0D 00	2B	0D 00 00 01 60 60 00 00 00 00 02

**Response**

MBAP	FC	Daten
00 00 00 00 00 0B 00	2B	FF 00 06 0D CE 12 00 07 06

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

Dieser Funktionscode wird zum Auslesen des gesamten Objektverzeichnis verwendet.

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

Das Format jedes "Objekt lesen" ist folgendermaßen:

**Request:**

Name	Länge	Wert / Bemerkung
Slave-Adresse	1 Byte	
Funktionscode	1 Byte	$65_{\text{h}}$
Unterfunktionscode	1 Byte	$55_{\text{h}}$ oder $AA_{\text{h}}$
Länge der Daten	1 Byte	$00_{\text{h}}$
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	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

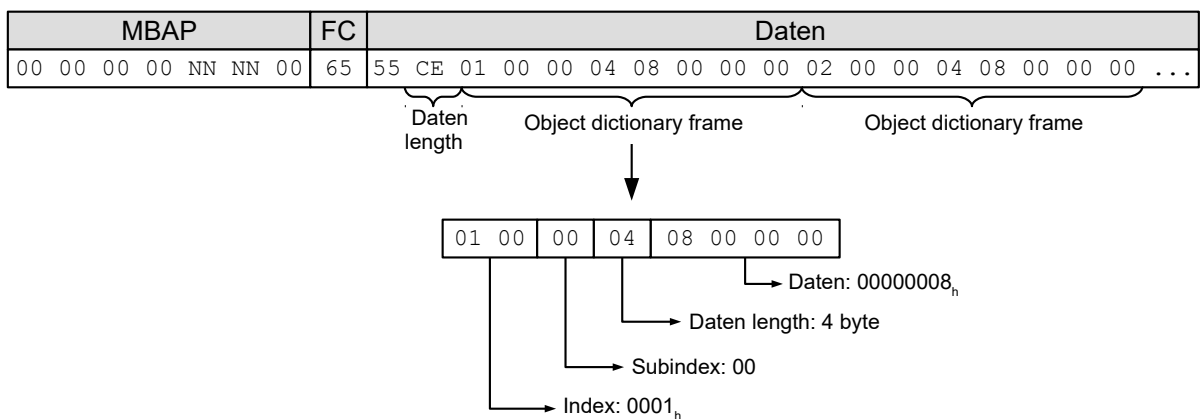
### Beispiel

Alle folgenden Zahlenwerte sind in Hexadezimal notiert.

Start des Auslesens des Objektverzeichnisses mit dem Request:

MBAP	FC	Daten
00 00 00 00 00 04 00	65	55 00

Die Response ist:



Den nächsten Teil des Objektverzeichnisses auslesen mit dem Request:

MBAP	FC	Daten
00 01 00 00 00 04 00	65	AA 00

Die Response ist:

MBAP	FC	Daten
00 01 00 00 NN NN 00	65	AA CD 21 00 0A 02 07 00 21 00 0B 02 07 00 21 00 0C 02 ...

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

MBAP	FC	Daten
NN NN 00 00 00 03 00	E5	0D

### Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	000B <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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

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

CANopen-Fehlercode	Beschreibung
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
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0008/0009 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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

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

MBAP	FC	Daten
00 00 00 00 00 0D 00	2B	0D 00 00 01 60 60 00 00 00 00 02

### Response

MBAP	FC	Daten
00 00 00 00 00 0B 00	2B	FF 00 06 0D CE 12 00 07 06

## 8.4.7 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
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0007 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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	

#### Response:

Name	Länge	Wert / Bemerkung
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0004 <sub>h</sub> +n
Unit Identifier	1 Byte	00 <sub>h</sub>
Funktionscode	1 Byte	
Unterfunktionscode	1 Byte	
Länge der Daten	1 Byte	
n mal Objektverzeichnis-Frame	1 - 252 Bytes	

Ein Objektverzeichnis-Frame besteht aus den folgenden Bytes:

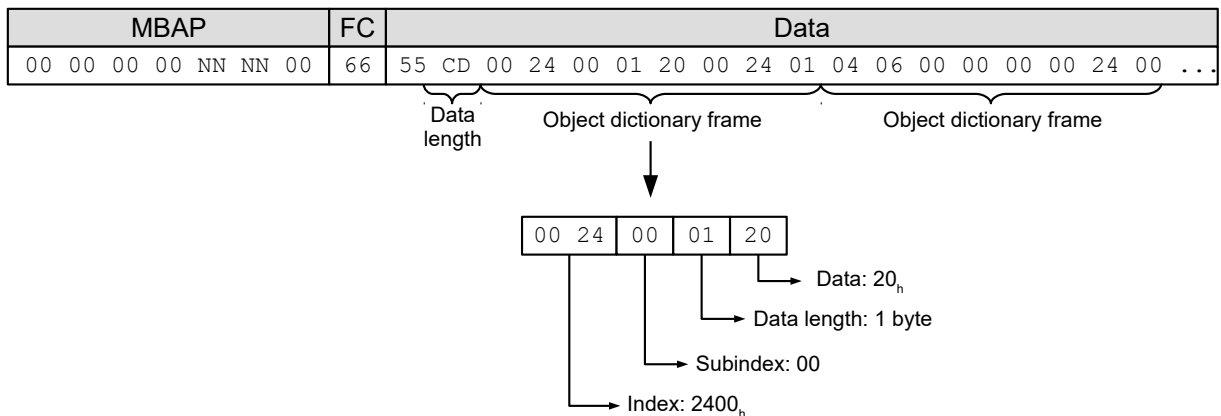
Name	Wert / Bemerkung	
Index Low Byte	1 Byte	
Index High Byte	1 Byte	
Subindex	1 Byte	
Anzahl der Bytes	1 Byte	Anzahl m der validen Daten im Datenfeld
Daten Byte	m-1 Byte	

### Beispiel

Alle folgenden Zahlenwerte sind in Hexadezimal notiert, der Index des zu lesenden Objektes ist 2400<sub>h</sub>.  
Start des Auslesens des Arrays mit dem Request:

MBAP	FC	Data
00 00 00 00 06 00	66	55 00 24 00

Die Response ist:



### Fehlerreaktion

Im Falle eines Fehlers wird die folgende Fehlernachricht gesendet:

Name	Länge	Beispielwert
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	000B <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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

CANopen-Fehlercode	Beschreibung
FFFF0000 <sub>h</sub>	Abort no error
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
Transaction Identifier	2 Bytes	0000 <sub>h</sub>
Protocol Identifier	2 Bytes	0000 <sub>h</sub>
Länge	2 Bytes	0008/0009 <sub>h</sub>
Unit Identifier	1 Byte	00 <sub>h</sub>
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

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

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

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

MBAP	FC	Daten
00 00 00 00 00 0D 00	2B	0D 00 00 01 60 60 00 00 00 00 02

#### Response

MBAP	FC	Daten
00 00 00 00 00 0B 00	2B	FF 00 06 0D CE 12 00 07 06

## 8.5 Prozessdatenobjekte (PDO)

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

### 8.5.1 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.

### 8.5.2 Übertragung

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



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

Index	Datentyp
0002 <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.

### Beispiel

In dem Mapping sollen folgende Objekte eingestellt werden:

- **3602<sub>h</sub>:00<sub>h</sub>** = "6<sub>h</sub>" (6 Werte werden gemappt)
- **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<sub>h</sub>:04<sub>h</sub>** = "60640020<sub>h</sub>" (das Objekt **6064<sub>h</sub>:00<sub>h</sub>**, Länge 32 Bit wird gemappt)
- **3602<sub>h</sub>:05<sub>h</sub>** = "60440010<sub>h</sub>" (das Objekt **6044<sub>h</sub>:00<sub>h</sub>**, Länge 16 Bit wird gemappt)
- **3602<sub>h</sub>:06<sub>h</sub>** = "60FD0020<sub>h</sub>" (das Objekt **60FD<sub>h</sub>:00<sub>h</sub>**, Länge 32 Bit wird gemappt)

Nach dem Mapping für das Objekt **6061<sub>h</sub>:00<sub>h</sub>** wird ein Dummy-Objekt eingefügt, damit das nachfolgende Objekt **6064<sub>h</sub>:00<sub>h</sub>** auf 32 Bit ausgerichtet wird.

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

MBAP	FC	Daten
00 00 00 00 00 06 00	04	13 88 00 07

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

MBAP	FC	Daten
00 00 00 00 00 12 00	04	0E 06 40 00 02 00 00 00 00 00 00 00 00 00

## 8.6 NanoJ-Objekte

Die NanoJ-Objekte **2400<sub>h</sub>** NanoJ Input und **2500<sub>h</sub>** (NanoJ Output) werden wie das Prozessimage auf Modbus-Register gemappt:

- **2500<sub>h</sub>** mit 32 x 32 Bit Werten wird auf die Modbus Register Adresse ab 2000<sub>d</sub> (BB8<sub>h</sub>) gemappt und kann auf diese Weise nur gelesen werden.
- **2400<sub>h</sub>** mit 32 x 32 Bit Werten wird auf die Modbus Register Adresse ab 3000<sub>d</sub> (7D0<sub>h</sub>) gemappt und kann auf diese Weise nur beschrieben werden.

Für den Zugriff können die Kommandos mit Funktionscode 03<sub>h</sub>, 04<sub>h</sub>, 10<sub>h</sub> und 17<sub>h</sub> 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:

MBAP	FC	Daten
00 00 00 00 00 1B 00	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

**Reply:** Der Slave schickt an den Master folgende Antwort:

MBAP	FC	Daten
00 00 00 00 00 13 00	17	10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

## 9 Programmierung mit NanoJ

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

### 9.1 NanoJ-Programm

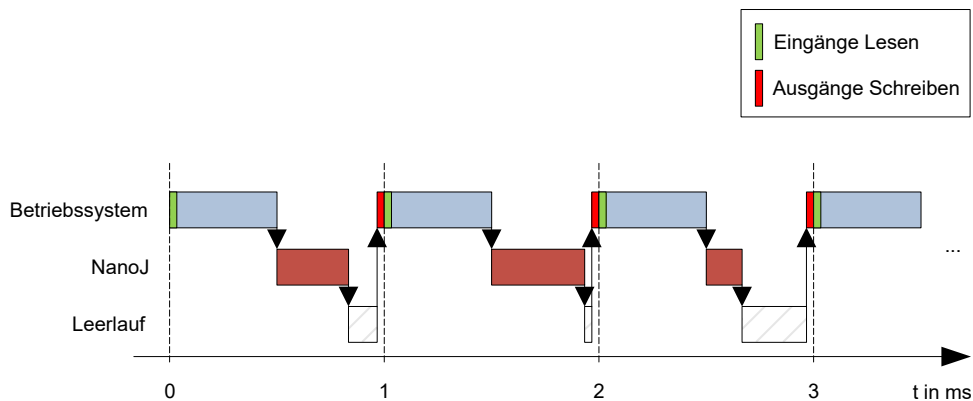
Ein *NanoJ-Programm* stellt eine geschützte Ausführungsumgebung innerhalb der Firmware zur Verfügung. In dieser kann der Anwender eigene Abläufe anlegen. Diese können dann Funktionen in der Steuerung auslösen, indem beispielsweise Einträge im Objektverzeichnis gelesen oder geschrieben werden.

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.

#### 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**.

## 9.1.2 Sandbox

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

## 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 Systemcalls
- Aufruf sonstiger Systemcalls (z. B. Debug-Ausgabe schreiben)

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

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

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

Die gesetzten Mappings können über die GUI bei den Objekten 2310<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 *NanoJEasy* gesteuert, ob eine Variable im Input-, Output- oder Datenbereich abgelegt wird.

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



### Tipp

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

Eine Liste verfügbarer Systemcalls findet sich im Kapitel **Systemcalls im NanoJ-Programm**.



### Tipp

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

## 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>	<b>2300h NanoJ Control</b>
2301 <sub>h</sub>	<b>2301h NanoJ Status</b>
2302 <sub>h</sub>	<b>2302h NanoJ Error Code</b>
2310 <sub>h</sub>	<b>2310h NanoJ Input Data Selection</b>
2320 <sub>h</sub>	<b>2320h NanoJ Output Data Selection</b>
2330 <sub>h</sub>	<b>2330h NanoJ In/output Data Selection</b>

### 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".



### 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".
- Umbenennen der Datei *TEST1.USR* in *vmmcode.usr*.
- Kopieren der Datei *vmmcode.usr* über USB auf die Steuerung.
- NanoJ-Programm starten durch Beschreiben von Objekt **2300<sub>h</sub>**, Bit 0 = "1" oder Neustart der Steuerung.
- Überprüfen des Eintrags **2302<sub>h</sub>** auf Fehlercode und des Objekts **2301<sub>h</sub>**, Bit 0 = "1" (NanoJ-Programm läuft).



### Hinweis

Aufgrund Limitierungen in der USB Implementation wird nach einem Neustart der Steuerung die Datei "VMMCODE.USR" auf eine Größe von 16kB gesetzt und das Erstelldatum auf den 13.03.2012 gestellt.

Um ein laufendes Programm anzuhalten: Beschreiben des Eintrags **2300<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 ist nicht korrekt :

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

## 9.1.7 NanoJ-Programmbeispiel

Das Beispiel zeigt das Programmieren eines Rechtecksignals in das Objekt **2500<sub>h</sub>:01<sub>h</sub>**.

```
// 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. Ein Kommentar oberhalb des Mappings ist erlaubt.



### 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.

## 9.2.2 Beispiel eines Mappings

Beispiel eines Mappings und der zugehörigen Variablenzugriffe:

```
map U16 controlWord as output 0x6040:00
map U08 statusWord as input 0x6041:00
map U08 modeOfOperation as inout 0x6060:00

#include "wrapper.h"

void user()
{
    [...]
    Out.controlWord = 1;
    U08 tmpVar = In.statusWord;
    InOut.modeOfOperation = tmpVar;
    [...]
}
```

## 9.2.3 Möglicher Fehler bei `od_write()`

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

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

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

Damit ergibt sich folgender Ablauf:

1. Die Funktion `od_write` schreibt den Wert 5 in das Objekt `6040h:00h`.
2. Am Ende des 1 ms-Zyklus wird das Mapping geschrieben, welches ebenfalls das Objekt `6040h:00h` beschreibt, allerdings mit dem Wert 1.
3. Somit wird - aus Sicht des Benutzers - der `od_write`-Befehl wirkungslos.

## 9.3 Systemcalls im NanoJ-Programm

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

### 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
----	-----------------------------------

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

### Firmware Version

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

### Änderungshistorie (ChangeLog)

Hier werden eventuelle Änderungen an dem Objekt notiert.

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

### Zugriff

Hier wird die Zugriffsbeschränkung eingetragen. Folgende Beschränkungen gibt es:

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

### PDO-Mapping

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

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

### Zulässige Werte

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

### Vorgabewert

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

## 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

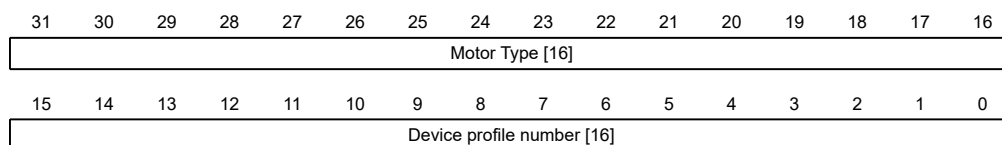
### Funktion

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	00060192 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

## Beschreibung



### Motor Type[16]

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

- Bit 23 bis Bit 16: Wert "1": Servoantrieb
- Bit 23 bis Bit 16: Wert "2": Schrittmotor

### Device profile number[16]

Beschreibt den unterstützten CANopen-Standard.

Werte:

0192<sub>h</sub> bzw. 0402<sub>d</sub> (Vorgabewert): Der CiA 402-Standard wird unterstützt.

## 1001h Error Register

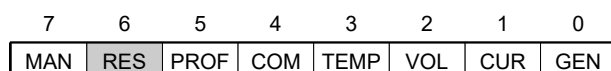
### Funktion

Fehlerregister: Im Fehlerfall wird das entsprechende Fehlerbit gesetzt. Sollte der Fehler nicht mehr bestehen, wird es automatisch wieder gelöscht.

### 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: Der Motor drehte sich in die falsche Richtung.

## 1003h Pre-defined Error Field

### Funktion

Dieses Objekt beinhaltet einen Fehlerstapel mit bis zu acht Einträgen.

### Objektbeschreibung

---

Index	1003 <sub>h</sub>
Objektnamen	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	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein

---

Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	02 <sub>h</sub>
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	03 <sub>h</sub>
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	04 <sub>h</sub>
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	05 <sub>h</sub>
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	06 <sub>h</sub>
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>



Subindex	07 <sub>h</sub>
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	08 <sub>h</sub>
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

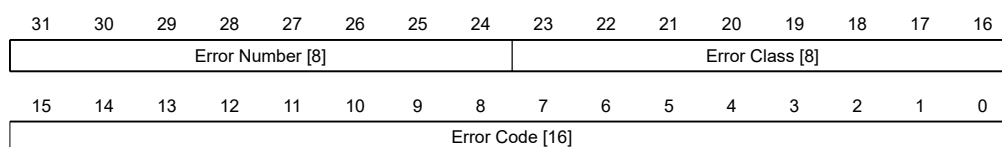
## Beschreibung

### 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



### Error Number [8]

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

Fehlernummer	Beschreibung
0	Watchdog-Reset
1	Eingangsspannung zu hoch
2	Ausgangsstrom zu hoch
3	Eingangsspannung zu niedrig
4	Fehler am Feldbus
6	Nur CANopen: NMT-Master braucht zu lange, um Nodeguarding-Anforderung zu schicken
7	Encoderfehler durch elektrische Störung oder defekte Hardware
9	Fehler in der A/B-Spur des Encoders

Fehlernummer	Beschreibung
10	Warnung: Positiver Endschalter überschritten
11	Warnung: Negativer Endschalter überschritten
12	Übertemperatur-Fehler
13	Die Werte des Objekts <b>6065<sub>h</sub></b> (Following Error Window) und des Objekts <b>6066<sub>h</sub></b> (Following Error Time Out) wurden überschritten, es wurde ein Fault ausgelöst.
14	Warnung: nichtflüchtiger Speicher voll. Der aktuelle Speichervorgang konnte nicht abgeschlossen werden, Teile der Daten des Speichervorgangs sind verloren. Neustart der Steuerung erforderlich für Aufräumarbeiten.
15	Motor blockiert
16	Warnung: nichtflüchtiger Speicher beschädigt, Neustart der Steuerung erforderlich für Aufräumarbeiten (alle gespeicherten Objekte werden auf Default zurückgesetzt).
17	Nur CANopen: Slave brauchte zu lange um PDO-Nachrichten zu Senden.
18	Hall-Sensor fehlerhaft
19	Nur CANopen: PDO aufgrund eines Längenfehlers nicht verarbeitet
20	Nur CANopen: PDO Länge überschritten
21	Warnung: Starten Sie die Steuerung neu, um zukünftige Fehler beim Speichern (nichtflüchtiger Speicher voll/korrupt) zu vermeiden.
22	Nennstrom muss gesetzt werden (203B <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

### 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
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> )
7121 <sub>h</sub>	Motor blockiert
7305 <sub>h</sub>	Inkrementaler oder Hall-Sensor fehlerhaft
7600 <sub>h</sub>	Warnung: Nichtflüchtiger Speicher voll oder korrupt, Neustart der Steuerung für Aufräumarbeiten

Error Code	Beschreibung
8000 <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	<ul style="list-style-type: none"> <li>• C5-E-1-81: C5-E-1-81</li> <li>• C5-E-2-81: C5-E-2-81</li> </ul>
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-v1825-B577172
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	Firmware Version FIR-v1436: Eintrag "Objektname" geändert von "Store Parameter" auf "Store Parameters".  Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 3 auf 4.

Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5.

Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7.

Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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	<p>Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "Restore Default Parameter" auf "Restore Default Parameters".</p> <p>Firmware Version FIR-v1436: Die Anzahl der Einträge haben sich geändert von 2 auf 4.</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 4 auf 5.</p> <p>Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Comm Default Parameters" auf "Restore Communication Default Parameters".</p> <p>Firmware Version FIR-v1512: Eintrag "Name" geändert von "Restore The Application Default Parameters" auf "Restore Application Default Parameters".</p> <p>Firmware Version FIR-v1540: Die Anzahl der Einträge haben sich geändert von 5 auf 7.</p> <p>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 7 auf 14.</p>

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
Subindex	07 <sub>h</sub>
Name	Restore Miscellaneous Configurations
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 <sub>h</sub>
<hr/>	
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>

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>

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.



#### Tip

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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub>
Name	Vendor-ID
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000026C <sub>h</sub>
<hr/>	
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>• C5-E-1-81: 00000057<sub>h</sub></li> <li>• C5-E-2-81: 00000058<sub>h</sub></li> </ul>
<hr/>	
Subindex	03 <sub>h</sub>
Name	Revision Number
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	07210000 <sub>h</sub>
<hr/>	
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	Highest Sub-index Supported
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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <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

Subindex	03 <sub>h</sub>
Name	Program Data DataFlash
Datentyp	DOMAIN
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0

## 1F51h Program Control

### Funktion

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

### Objektbeschreibung

Index	1F51 <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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <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>
Subindex	03 <sub>h</sub>
Name	Program Control DataFlash
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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <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>

Subindex	03 <sub>h</sub>
Name	Program Status DataFlash
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 200Fh IEEE 802 MAC Address

### Funktion

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

### Objektbeschreibung

Index	200F <sub>h</sub>
Objektnamen	IEEE 802 MAC Address
Object Code	VARIABLE

Datentyp	VISIBLE_STRING
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0
Firmware Version	FIR-v1748-B533384
Änderungshistorie	

## 2010h IP-Configuration

### Funktion

Über dieses Objekt wird die Ethernet-Schnittstelle konfiguriert.

### Objektbeschreibung

Index	2010 <sub>h</sub>
Objektname	IP-Configuration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Ethernet
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	0000006C <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1748-B533384: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Ethernet".

### Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									LLMNR	NBIO		AUTO	DHCP		IP

#### IP

Wert = "1": Eine statische IP-Adresse aus dem Objekt **2011<sub>h</sub>** wird genutzt und die Netzwerkmaske aus dem Objekt **2012<sub>h</sub>** wird genutzt.

#### UPnP

Wert = "1": Die UPnP (Universal Plug and Play) Benachrichtigungen werden aktiviert

#### DHCP

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

**AUTO**

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

**EXT**

Wert = "1": Die IP Adresse wurde von extern durch NanoFlash gesetzt und gilt nur bis zum nächsten Neustart der Steuerung.

**NBIOS**

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

**LLMNR**

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

## 2011h Static-IPv4-Address

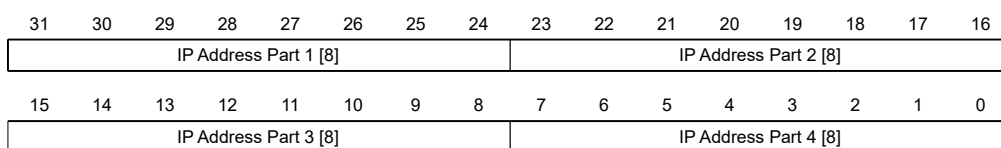
### Funktion

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

### Objektbeschreibung

Index	2011 <sub>h</sub>
Objektname	Static-IPv4-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Ethernet
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	C0A80792 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1450: Eintrag "Object Name" geändert von "Static-IP-Address" auf "Static-IPv4-Address".  Firmware Version FIR-v1748-B533384: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Ethernet".

### Beschreibung



**IP Address Part 1 [8]**

Gibt den ersten Teil der IP-Adresse an

### IP Address Part 2 [8]

Gibt den zweiten Teil der IP-Adresse an

### IP Address Part 3 [8]

Gibt den dritten Teil der IP-Adresse an

### IP Address Part 4 [8]

Gibt den vierten Teil der IP-Adresse an

### Beispiel

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

192 => C0<sub>h</sub>

168 => A8<sub>h</sub>

2 => 02<sub>h</sub>

0 => 0

Der zugehörige Einstellwert lautet dann C0A80200<sub>h</sub>.

## 2012h Static-IPv4-Subnet-Mask

### Funktion

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

### Objektbeschreibung

---

Index	2012 <sub>h</sub>
Objektnamen	Static-IPv4-Subnet-Mask
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Ethernet
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	FFFFFF00 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1450: Eintrag "Object Name" geändert von "Static-IP-Subnet-Mask" auf "Static-IPv4-Subnet-Mask".  Firmware Version FIR-v1748-B533384: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Ethernet".

---

## Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Subnet Mask Part 1 [8]								Subnet Mask Part 2 [8]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Subnet Mask Part 3 [8]								Subnet Mask Part 4 [8]							

### Subnet Mask Part 1 [8]

Gibt den ersten Teil der Subnetzmaske an

### Subnet Mask Part 2 [8]

Gibt den zweiten Teil der Subnetzmaske an

### Subnet Mask Part 3 [8]

Gibt den dritten Teil der Subnetzmaske an

### Subnet Mask Part 4 [8]

Gibt den vierten Teil der Subnetzmaske an

### Beispiel

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

255 => FF<sub>h</sub>

0 => 0

Der zugehörige Einstellwert lautet dann FFFFFFFF00<sub>h</sub>.

## 2013h Static-IPv4-Gateway-Address

### Funktion

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

### Objektbeschreibung

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

Firmware Version FIR-v1748-B533384: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "ja, Kategorie: Ethernet".

## Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP-Gateway-Address Part 1 [8]								IP-Gateway-Address Part 2 [8]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP-Gateway-Address Part 3 [8]								IP-Gateway-Address Part 4 [8]							

### IP-Gateway-Adress Part 1 [8]

Gibt den ersten Teil der IP-Gateway-Adresse an

### IP-Gateway-Adress Part 2 [8]

Gibt den zweiten Teil der IP-Gateway-Adresse an

### IP-Gateway-Adress 3 [8]

Gibt den dritten Teil der IP-Gateway-Adresse an

### IP-Gateway-Adress 4 [8]

Gibt den vierten Teil der IP-Gateway-Adresse an

## Beispiel

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

192 => C0<sub>h</sub>

168 => A8<sub>h</sub>

2 => 02<sub>h</sub>

0 => 0

Der zugehörige Einstellwert lautet dann C0A80200<sub>h</sub>.

## 2014h Current-IPv4-Address

### Funktion

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

### Objektbeschreibung

Index	2014 <sub>h</sub>
Objektname	Current-IPv4-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	nein

Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1450: Eintrag "Object Name" geändert von "Current-IP-Address" auf "Current-IPv4-Address".

## Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP Address Part 1 [8]								IP Address Part 2 [8]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP Address Part 3 [8]								IP Address Part 4 [8]							

### IP Address Part 1 [8]

Gibt den ersten Teil der IP-Adresse an

### IP Address Part 2 [8]

Gibt den zweiten Teil der IP-Adresse an

### IP Address Part 3 [8]

Gibt den dritten Teil der IP-Adresse an

### IP Address Part 4 [8]

Gibt den vierten Teil der IP-Adresse an

### Beispiel

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

192 => C0<sub>h</sub>

168 => A8<sub>h</sub>

2 => 02<sub>h</sub>

0 => 0

Der zugehörige Einstellwert lautet dann C0A80200<sub>h</sub>.

## 2015h Current-IPv4-Subnet-Mask

### Funktion

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

### Objektbeschreibung

Index	2015 <sub>h</sub>
Objektname	Current-IPv4-Subnet-Mask
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein

Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1450: Eintrag "Object Name" geändert von "Current-IP-Subnet-Mask" auf "Current-IPv4-Subnet-Mask".

## Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Subnet Mask Part 1 [8]								Subnet Mask Part 2 [8]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Subnet Mask Part 3 [8]								Subnet Mask Part 4 [8]							

### Subnet Mask Part 1 [8]

Gibt den ersten Teil der Subnetzmaske an

### Subnet Mask Part 2 [8]

Gibt den zweiten Teil der Subnetzmaske an

### Subnet Mask Part 3 [8]

Gibt den dritten Teil der Subnetzmaske an

### Subnet Mask Part 4 [8]

Gibt den vierten Teil der Subnetzmaske an

### Beispiel

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

255 => FF<sub>h</sub>

0 => 0

Der zugehörige Einstellwert lautet dann FFFFFFFF00<sub>h</sub>.

## 2016h Current-IPv4-Gateway-Address

### Funktion

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

### Objektbeschreibung

Index	2016 <sub>h</sub>
Objektname	Current-IPv4-Gateway-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein



---

Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
Firmware Version	FIR-v1540
Änderungshistorie	

---

## 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	Firmware Version FIR-v1540: Eintrag "Speicherbar" geändert von "ja, Kategorie: Kommunikation" auf "nein".  Firmware Version FIR-v1540: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".

### Beschreibung

Die Anzahl der Stopbits ist abhängig von der Parity welche im Objekt **202D<sub>h</sub>** eingestellt werden kann.

Anzahl der Stopbits	Wert in Objekt 202C <sub>h</sub>
1	0
2	2

## 202Dh MODBUS RTU Parity

### Funktion

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

### Objektbeschreibung

Index	202D <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>
Objektnamen	Pole Pair Count
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000032 <sub>h</sub>
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>• C5-E-1-81: 00000258<sub>h</sub></li><li>• C5-E-2-81: 000003E8<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>

---

## 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	0000C92C <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	00002710 <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<sub>h</sub>** \* (**2037<sub>h</sub>** + 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	Highest Sub-index Supported
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>
<hr/>	
Subindex	05 <sub>h</sub>
Name	PWM Frequency
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	zwischen 0 und 20000 (4E20 <sub>h</sub> )
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
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 der Bremsen-PWM in Hertz.
- 06<sub>h</sub>: Tastgrad der Bremsen-PWM in Prozent.

## 2039h Motor Currents

### Funktion

Dieses Objekt enthält die gemessenen Motorströme in mA.

### Objektbeschreibung

Index	2039 <sub>h</sub>
Objektnamen	Motor Currents
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "nein" auf "TX-PDO".



Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "TX-PDO".

Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 03 geändert von "nein" auf "TX-PDO".

Firmware Version FIR-v1504: Tabellen-Eintrag "PDO-Mapping" bei Subindex 04 geändert von "nein" auf "TX-PDO".

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	04 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	I_d
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	I_q
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	03 <sub>h</sub>
Name	I_a
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	04 <sub>h</sub>
Name	I_b

Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## Beschreibung



### 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	Highest Sub-index Supported
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	FFFFFFBA <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>:01<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 I<sup>2</sup>t 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 **I<sup>2</sup>t 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	<p>Firmware Version FIR-v1512: Eintrag "Savable" geändert von "nein" auf "ja, Kategorie: Applikation".</p> <p>Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 7 auf 8.</p> <p>Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Nominal Current" auf "Motor Rated Current".</p>

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	07 <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>• C5-E-1-81: 00000258<sub>h</sub></li> <li>• C5-E-2-81: 000003E8<sub>h</sub></li> </ul>

Subindex	02 <sub>h</sub>
Name	Maximum Duration Of Peak Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

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>
Subindex	07 <sub>h</sub>
Name	ActualResistance
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 mA an, von der abhängt, ob auf Maximalstrom oder Nennstrom geschaltet wird.
- 04<sub>h</sub>: CalcValue, gibt den berechneten Wert 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 Istdrehmoment 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) 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>**.

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>
-------	-------------------

---

---

Objektnamen	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>
Objektnamen	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	

---

## 2059h Encoder Configuration

### Funktion

Mit diesem Objekt kann die Versorgungsspannung und der Typ des Encoders umgeschaltet werden.

### Objektbeschreibung

---

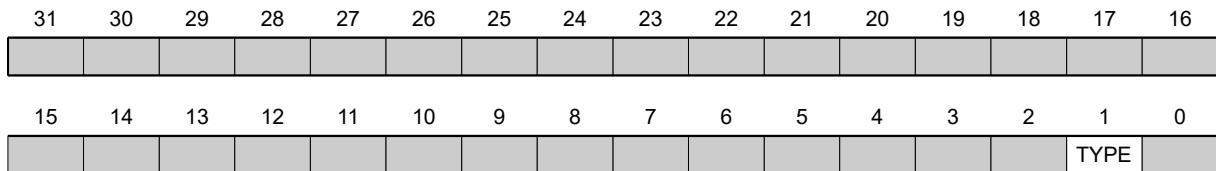
Index	2059 <sub>h</sub>
Objektnamen	Encoder Configuration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Tuning
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	

---



Vorgabewert	00000000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".

## Beschreibung



### TYPE

Legt den Typ des Encoders fest. Das Bit muss den Wert "0" bei einem differentiellen Encoder haben. Für einen single-ended Encoder muss das Bit auf "1" gesetzt werden.

## 205Ah Absolute Sensor Boot Value (in User Units)

### Funktion



#### Tipp

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

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

### Objektbeschreibung

Index	205A <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	Firmware Version FIR-v1512: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".  Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Encoder Boot Value" auf "Absolute Sensor Boot Value (in User Units)".

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

## 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

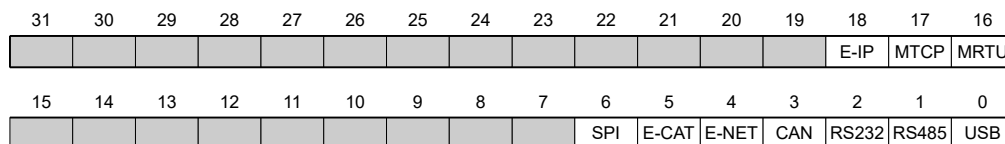
Zeigt die verfügbaren Feldbusse an.

### 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	001B0011 <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	001B0011 <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<sub>h</sub>:1<sub>h</sub>** werden alle physikalischen Schnittstellen/Protokolle angezeigt, welche aktiviert/deaktiviert werden können. Diese können in diesem Objekt (2102<sub>h</sub>) geschaltet werden. Der gegenwärtige Status der aktivierten Feldbusse steht im Objekt **2103<sub>h</sub>:2<sub>h</sub>**.

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 <sub>h</sub>
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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein

---

Zulässige Werte	
Vorgabewert	02 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub>
Name	Fieldbus Module Disable Mask
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	02 <sub>h</sub>
Name	Fieldbus Module Enabled
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	001B0011 <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>
Objektnamen	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	ja, Kategorie: Applikation
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B531667
Änderungshistorie	

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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>
<hr/>	
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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub>
Name	PDI Status

Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
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

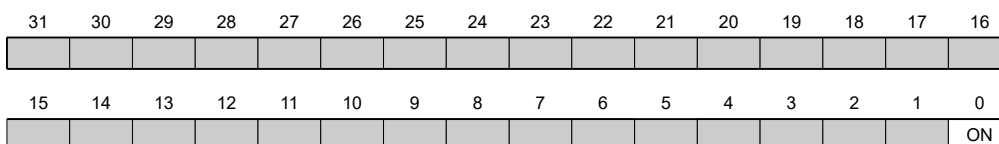
### Funktion

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.

## 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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ERR	RES	RUN

#### RUN

Wert = "0": Programm ist angehalten, Wert = "1": NanoJ-Programm läuft.

#### RES

Reserviert.

#### ERR

Programm wurde mit Fehler beendet. Fehlerursache kann aus dem Objekt **2302<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. sin, cosin 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 <b>2310h NanoJ Input Data Selection</b> )
3000000 <sub>h</sub>	Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs <code>output</code> deklariert (siehe <b>2320h NanoJ Output Data Selection</b> )
4000000 <sub>h</sub>	Ungültiges Mapping in der NanoJ-Programmdatei: es wurden zu viele Variablen des Typs <code>inout</code> deklariert (siehe <b>2330h NanoJ In/output Data Selection</b> )
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.

## 230Fh Uptime Seconds

### Funktion

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



#### Hinweis

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

### Objektbeschreibung

Index	230F <sub>h</sub>
Objektnamen	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>
Objektnamen	NanoJ Input Data Selection
Object Code	ARRAY
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1650-B472161
Änderungshistorie	Firmware Version FIR-v1436: Eintrag "Object Name" geändert von "VMM Input Data Selection" auf "NanoJ Input Data Selection". Firmware Version FIR-v1650-B472161: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "nein".

Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 00 geändert von "lesen/schreiben" auf "nur lesen".

Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".

## Wertebeschreibung

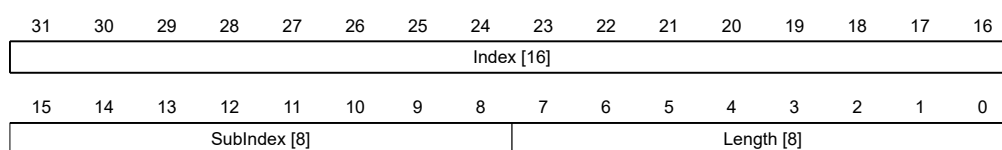
Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 <sub>h</sub>

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>

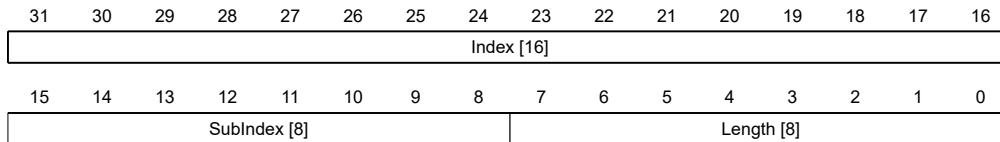
## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen

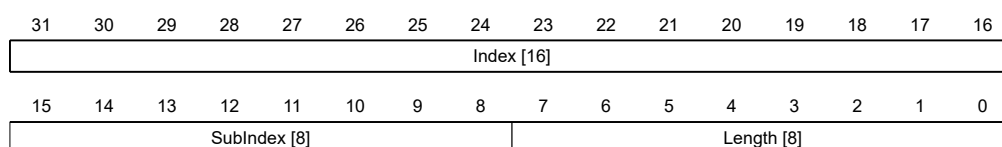


PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	10 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub> - 10 <sub>h</sub>
Name	Mapping #1 - #16
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

## 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	Highest Sub-index Supported
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	Highest Sub-index Supported
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	Highest Sub-index Supported
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	Highest Sub-index Supported
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()`, `VmmDebugOutputInt()` und dergleichen 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	Highest Sub-index Supported
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	Highest Sub-index Supported
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>
<hr/>	
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

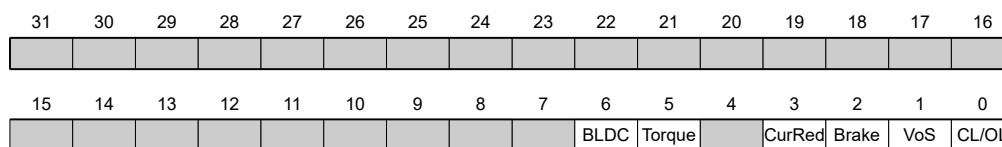
### Funktion

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	00000000 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Fahrt".  Firmware Version FIR-v1540: Eintrag "Saveable" geändert von "ja, Kategorie: Fahrt" auf "ja, Kategorie: Bewegung".

## Beschreibung



### CL/OL

Umschaltung zwischen *Open Loop* und *Closed Loop*

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

### VoS

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

### Brake

Wert = "1": Einschalten der automatischen Bremsensteuerung.

### CurRed (Current Reduction)

Wert = "1": Stromabsenkung im *Open Loop* aktiviert

### Torque

nur in den Betriebsmodi **Profile Torque** und **Cyclic Synchronous Torque** aktiv

Wert = "1": M-Regler ist aktiv, andernfalls ist ein V-Regler überlagert: in den Torque-Modi wird kein V-Regler zur Geschwindigkeitsbegrenzung verwendet, das Objekt **6080<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)

## 3203h Feedback Selection

### Funktion

In diesem Objekt werden die Quellen der Vorgaben für die Kommutierung, Geschwindigkeits- und Positionsregelung festgelegt.

### Objektbeschreibung

Index	3203 <sub>h</sub>
Objektnamen	Feedback Selection
Object Code	ARRAY
Datentyp	UNSIGNED8
Speicherbar	ja, Kategorie: Bewegung
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B538662



Änderungshistorie	Firmware Version FIR-v1748-B538662: Die Anzahl der Einträge haben sich geändert von 5 auf 4.
-------------------	--

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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 die Rückführung n für die Positionsregelung verwendet.
- Bit 1: wird das Bit auf "1" gesetzt, wird die Rückführung n für die Geschwindigkeitsregelung verwendet.
- Bit 2: wird das Bit auf "1" gesetzt, wird die Rückführung n für die Kommutierung im **Closed Loop** verwendet.

Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Welche Rückführung die Steuerung für die einzelnen Regler (Kommutierung, Geschwindigkeit, Position) berücksichtigt, ist implizit durch die Reihenfolge der Rückführungen vorgegeben.

Das Aufsuchen beginnt immer mit Rückführung 2 und setzt sich aufsteigend fort, bis alle produktspezifisch vorhandenen Rückführungen abgefragt wurden. Wird eine Rückführung gefunden deren Parametrierung gesetzt ist, dann wird diese dem entsprechenden Regler zugeordnet und die Suche abgebrochen.



#### Hinweis

Wird das Bit 0 in 3202<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	Firmware Version FIR-v1748-B538662: Die Anzahl der Einträge haben sich geändert von 5 auf 4.

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	03 <sub>h</sub>
<hr/>	
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>
<hr/>	
Subindex	02 <sub>h</sub>
Name	Index Of 2nd Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	3390 <sub>h</sub>
<hr/>	
Subindex	03 <sub>h</sub>
Name	Index Of 3rd Feedback Interface
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	33A0 <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*.

## 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> <p>Firmware Version FIR-v1650-B472161: Eintrag "Data type" geändert von "INTEGER32" auf "UNSIGNED32".</p> <p>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 11 auf 13.</p> <p>Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 bis 0A geändert von "nein" auf "RX-PDO".</p>

---

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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>

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	000668A0 <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	00002EE0 <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	000668A0 <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	00002EE0 <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	0003A980 <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	0000AFC8 <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>: Proportionalanteil des S-Reglers (Position)
- Subindex 02<sub>h</sub>: Integralanteil des S-Reglers (Position)
- Subindex 03<sub>h</sub>: Proportionalanteil des V-Reglers (Geschwindigkeit)
- Subindex 04<sub>h</sub>: Integralanteil des V-Reglers (Geschwindigkeit)
- Subindex 05<sub>h</sub>: (Closed Loop) Proportionalanteil des Stromreglers der feldbildenden Komponente
- Subindex 06<sub>h</sub>: (Closed Loop) Integralanteil des Stromreglers der feldbildenden Komponente
- Subindex 07<sub>h</sub>: (Closed Loop) Proportionalanteil des Stromreglers der momentbildenden Komponente
- Subindex 08<sub>h</sub>: (Closed Loop) Integralanteil des Stromreglers der momentbildenden Komponente
- Subindex 09<sub>h</sub>: (Open Loop) Proportionalanteil des Stromreglers der feldbildenden Komponente
- Subindex 0A<sub>h</sub>: (Open Loop) Integralanteil 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	Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 2 auf 3.  Firmware Version FIR-v1738-B501312: Eintrag "Name" geändert von "Enable Legacy Power Mode" auf "Reserved".

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	03 <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	Do Not Touch Controller Settings
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00 <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> eingetragen wurden, die Werte in 3210<sub>h</sub> werden nicht geändert.

## 3220h Analog Inputs

### Funktion

Zeigt die Momentanwerte der Analogeingänge in 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 Inputs
Object Code	ARRAY
Datentyp	INTEGER16
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub>
Name	Analogue Input 1
Datentyp	INTEGER16
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	02 <sub>h</sub>
Name	Analogue Input 2
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:  $x \text{ digits} * 10 \text{ V} / 1024 \text{ digits}$
- Stromeingang:  $x \text{ digits} * 20 \text{ mA} / 1024 \text{ digits}$

## 3221h Analogue Inputs Control

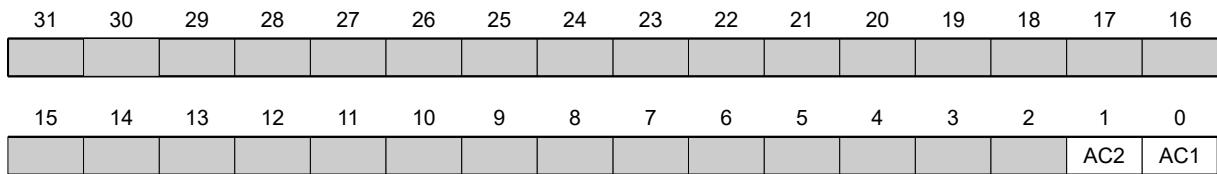
### Funktion

Mit diesem Objekt lässt sich ein Analog-Eingang von Spannungs- auf Strommessung umschalten.

### Objektbeschreibung

Index	3221 <sub>h</sub>
Objektname	Analogue 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 Analogue Inputs Switches

### Funktion

Dieses Objekt enthält . Die Schalter-Position wird nur beim Neustart einmalig ausgelesen.

### Objektbeschreibung

Index	3225 <sub>h</sub>
Objektname	Analogue Inputs 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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	01 <sub>h</sub>
----------	-----------------

Name	Analogue Input Switch1
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	Firmware Version FIR-v1426: Subindex 01 <sub>h</sub> : Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable"  Firmware Version FIR-v1512: Die Anzahl der Einträge haben sich geändert von 8 auf 9.

#### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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>
<hr/>	
Subindex	02 <sub>h</sub>
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	03 <sub>h</sub>
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	04 <sub>h</sub>
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	05 <sub>h</sub>
Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
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>
<hr/>	
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

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>:06<sub>h</sub>** (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..

## 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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	24 <sub>h</sub>

Subindex	01 <sub>h</sub> - 24 <sub>h</sub>
Name	Input Source #1 - #36
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 <sub>h</sub>

## Beschreibung

Der Subindex 01<sub>h</sub> enthält die Quelle für das Bit 0 des Objekts **60FD**. Der Subindex 02<sub>h</sub> enthält die Quelle für das Bit 1 des Objekts **60FD** und so weiter.

Die Nummer, die in eine Subindex geschrieben wird, bestimmt die Quelle für das zugehörige Bit. Die folgende Tabelle listet alle möglichen Signalquellen auf.

Nummer		
dec	hex	Signalquelle
00	00	Signal ist immer 0
01	01	Physikalischer Eingang 1
02	02	Physikalischer Eingang 2
03	03	Physikalischer Eingang 3
04	04	Physikalischer Eingang 4

Nummer		
dec	hex	Signalquelle
05	05	Physikalischer Eingang 5
06	06	Physikalischer Eingang 6
07	07	Physikalischer Eingang 7
08	08	Physikalischer Eingang 8
09	09	Physikalischer Eingang 9
10	0A	Physikalischer Eingang 10
11	0B	Physikalischer Eingang 11
12	0C	Physikalischer Eingang 12
13	0D	Physikalischer Eingang 13
14	0E	Physikalischer Eingang 14
15	0F	Physikalischer Eingang 15
16	10	Physikalischer Eingang 16
65	41	Hall Eingang "U"
66	42	Hall Eingang "V"
67	43	Hall Eingang "W"
68	44	Encoder Eingang "A"
69	45	Encoder Eingang "B"
70	46	Encoder Eingang "Index"
71	47	USB Power Signal
72	48	Status "Ethernet aktiv"
128	80	Signal ist immer 1
129	81	Invertierter physikalischer Eingang 1
130	82	Invertierter physikalischer Eingang 2
131	83	Invertierter physikalischer Eingang 3
132	84	Invertierter physikalischer Eingang 4
133	85	Invertierter physikalischer Eingang 5
134	86	Invertierter physikalischer Eingang 6
135	87	Invertierter physikalischer Eingang 7
136	88	Invertierter physikalischer Eingang 8
137	89	Invertierter physikalischer Eingang 9
138	8A	Invertierter physikalischer Eingang 10
139	8B	Invertierter physikalischer Eingang 11
140	8C	Invertierter physikalischer Eingang 12
141	8D	Invertierter physikalischer Eingang 13
142	8E	Invertierter physikalischer Eingang 14
143	8F	Invertierter physikalischer Eingang 15
144	90	Invertierter physikalischer Eingang 16
193	C1	Invertierter Hall Eingang "U"
194	C2	Invertierter Hall Eingang "V"
195	C3	Invertierter Hall Eingang "W"
196	C4	Invertierter Encoder Eingang "A"
197	C5	Invertierter Encoder Eingang "B"
198	C6	Invertierter Encoder Eingang "Index"
199	C7	Invertiertes USB Power Signal
200	C8	Invertierter Status "Ethernet aktiv"



## 3243h Digital Input Homing Capture

### Funktion

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



#### Hinweis

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

### Objektbeschreibung

Index	3243 <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	Highest Sub-index Supported
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>
<hr/>	
Subindex	03 <sub>h</sub>
Name	Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
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
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>

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	08 <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>
Subindex	07 <sub>h</sub>
Name	Reserved2
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:

- 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 diesem dem Subindex wird die an die Ausgänge gelegte Bitkombination abgelegt.

## 3252h Digital Output Routing

### Funktion

Dieses Objekt weist einem Ausgang eine Signalquelle zu, die mit dem **60FE<sub>h</sub>** kontrolliert werden kann.

### 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-v1748-B538662
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	05 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Output Control #1
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	

Vorgabewert	1080 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	Output Control #2
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0090 <sub>h</sub>
Subindex	03 <sub>h</sub>
Name	Output Control #3
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0091 <sub>h</sub>
Subindex	04 <sub>h</sub>
Name	Output Control #4
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0092 <sub>h</sub>
Subindex	05 <sub>h</sub>
Name	Output Control #5
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0093 <sub>h</sub>

## 3320h Read Analogue Input

### Funktion

Zeigt die Momentanwerte der Analogeingänge in benutzerdefinierten Einheiten an.

### Objektbeschreibung

Index	3320 <sub>h</sub>
Objektname	Read Analogue Input

Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	nein
Firmware Version	FIR-v1426
Änderungshistorie	

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Analogue Inputs
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Analogue Input 1
Datentyp	INTEGER32
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Analogue Input 2
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 Pre-scaling Wert (**3322<sub>h</sub>**) zusammen. Sind beide Objekteinträge 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:  $x \text{ digits} * 10 \text{ V} / 1024 \text{ digits}$
- Stromeingang:  $x \text{ digits} * 20 \text{ mA} / 1024 \text{ digits}$

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

## 3321h Analogue Input Offset

### Funktion

Offset, der zum eingelesenen Analogwert (**3320<sub>h</sub>**) addiert wird, bevor die Teilung mit dem Teiler aus dem Objekt **3322<sub>h</sub>** vorgenommen wird.

### Objektbeschreibung

Index	3321 <sub>h</sub>
Objektname	Analogue Input Offset
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Analogue Inputs
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Analogue Input 1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Analogue Input 2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>

### Beschreibung

- Subindex 00<sub>h</sub>: Anzahl der Offsets



- Subindex 01<sub>h</sub>: Offset für Analogeingang 1
- Subindex 02<sub>h</sub>: Offset für Analogeingang 2

## 3322h Analogue Input Pre-scaling

### 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	3322 <sub>h</sub>
Objektname	Analogue Input Pre-scaling
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Number Of Analogue Inputs
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	02 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Analogue Input 1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	alle Werte zulässig außer 0
Vorgabewert	00000001 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	Analogue Input 2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	alle Werte zulässig außer 0
Vorgabewert	00000001 <sub>h</sub>

## Beschreibung

Die Subindizes enthalten:

- Subindex 00<sub>h</sub>: Anzahl der Teiler
- Subindex 01<sub>h</sub>: Teiler für Analogeingang 1
- Subindex 02<sub>h</sub>: Teiler für Analogeingang 2

## 3390h Feedback Hall

### Funktion

Enthält Konfigurationswerte für die Hall-Sensoren. Die Werte werden vom **Auto-Setup** ermittelt.

### Objektbeschreibung

Index	3390 <sub>h</sub>
Objektname	Feedback Hall
Object Code	ARRAY
Datentyp	UNSIGNED16
Speicherbar	ja, Kategorie: Tuning
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	
Firmware Version	FIR-v1748-B531667
Änderungshistorie	

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0C <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	1st Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

Subindex	02 <sub>h</sub>
Name	2nd Alignment

Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	03 <sub>h</sub>
Name	3rd Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	04 <sub>h</sub>
Name	4th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	05 <sub>h</sub>
Name	5th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	06 <sub>h</sub>
Name	6th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	07 <sub>h</sub>
Name	7th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	08 <sub>h</sub>
Name	8th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	09 <sub>h</sub>
Name	9th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	0A <sub>h</sub>
Name	10th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	0B <sub>h</sub>
Name	11th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>
<hr/>	
Subindex	0C <sub>h</sub>
Name	12th Alignment
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 <sub>h</sub>

## 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	Highest Sub-index Supported
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

Die Subindizes haben folgende Funktion:

- 00<sub>h</sub> (Configuration): folgende Bits haben eine Bedeutung:
  - Bit 0: Wert = "0": Der Encoder verfügt nicht über einen Index. Wert = "1" : Encoder-Index gefunden und soll verwendet werden.
  - Bit 15: Wert = "1": der Encoder ist ein Singleturn-Absolut-Encoder.
- 01<sub>h</sub> (Alignment): Dieser Wert gibt den Versatz zwischen dem Index des Encoders und dem elektrischen Feld an.  
Die exakte Bestimmung ist nur über das **Auto-Setup** möglich. Das Vorhandensein dieses Wertes ist für den *Closed Loop*-Betrieb mit Encoder erforderlich.

## 3502h MODBUS Rx PDO Mapping

### Funktion

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

### Objektbeschreibung

Index	3502 <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	<p>Firmware Version FIR-v1738-B505321: Eintrag "Object Name" geändert von "MODBUS Rx PDO-Mapping" auf "MODBUS Rx PDO Mapping".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #1" auf "1st Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #2" auf "2nd Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #3" auf "3rd Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #4" auf "4th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #5" auf "5th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #6" auf "6th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #7" auf "7th Object To Be Mapped".</p> <p>Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #8" auf "8th Object To Be Mapped".</p>

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #9" auf "9th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #10" auf "10th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #11" auf "11th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #12" auf "12th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #13" auf "13th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #14" auf "14th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #15" auf "15th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #16" auf "16th Object To Be Mapped".

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	08 <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	32020020 <sub>h</sub>

Subindex	05 <sub>h</sub>
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	607A0020 <sub>h</sub>

Subindex	06 <sub>h</sub>
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60810020 <sub>h</sub>

Subindex	07 <sub>h</sub>
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60420010 <sub>h</sub>

Subindex	08 <sub>h</sub>
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben



PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	60FE0120 <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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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.

### 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".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #1" auf "1st Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #2" auf "2nd Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #3" auf "3rd Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #4" auf "4th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #5" auf "5th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #6" auf "6th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #7" auf "7th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #8" auf "8th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #9" auf "9th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #10" auf "10th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #11" auf "11th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #12" auf "12th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #13" auf "13th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #14" auf "14th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #15" auf "15th Object To Be Mapped".

Firmware Version FIR-v1748-B538662: Eintrag "Name" geändert von "Value #16" auf "16th Object To Be Mapped".

## Wertebeschreibung

---

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	06 <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	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>
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>
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>
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>

## 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	Soforthalt
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
2	Abbremsen mit "quick stop ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
3 bis 32767	Reserviert

## 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.

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

## 4012h HW Information

### Funktion

Dieses Objekt zeigt Informationen über die Hardware an.

### Objektbeschreibung

Index	4012 <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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein



Zulässige Werte	
Vorgabewert	01 <sub>h</sub>
<hr/>	
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	Highest Sub-index Supported
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

Bit 0 : 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	<p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 01 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 02 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1650-B472161: Eintrag "Name" geändert von "Temperature PCB [d?C]" auf "Temperature PCB [Celsius * 10]".</p> <p>Firmware Version FIR-v1650-B472161: Tabellen-Eintrag "Zugriff" bei Subindex 03 geändert von "lesen/schreiben" auf "nur lesen".</p> <p>Firmware Version FIR-v1738-B501312: Die Anzahl der Einträge haben sich geändert von 4 auf 6.</p>

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen

PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	05 <sub>h</sub>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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>
<hr/>	
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

## 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	

---

## 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).

## 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 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>
Objektnamen	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

### VE (Voltage Enabled)

Spannung angelegt

### QS (Quick Stop)

Wert = "0": Steuerung befindet sich in dem Zustand "Quick Stop"

### SOD (Switched On Disabled)

Wert = "1": Steuerung befindet sich in dem Zustand "Switched On Disabled"

### WARN (Warning)

Wert = "1": Warnung

### SYNC (Synchronisation)

Wert = "1": Steuerung ist synchron zum Feldbus, Wert = "0": Steuerung ist nicht synchron zum Feldbus

### REM (Remote)

Remote (Wert des Bits immer "1")

### TARG

Zielvorgabe erreicht

### ILA (Internal Limit Active)

Limit überschritten

### OMS (Operation Mode Specific)

Bedeutung abhängig vom gewählten Betriebsmodus

### CLA (Closed Loop Active)

Wert = "1": die Steuerung befindet sich im Status *Operation enabled* und der **Closed Loop** ist aktiviert.

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

Statusword (6041 <sub>h</sub> )	Zustand
xxxx xxxx x0xx 0000	Not ready to switch on

Statusword (6041 <sub>h</sub> )	Zustand
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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein

---

Zulässige Werte	
Vorgabewert	02 <sub>h</sub>
<hr/>	
Subindex	01 <sub>h</sub>
Name	MinAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <sub>h</sub>
<hr/>	
Subindex	02 <sub>h</sub>
Name	MaxAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00004E20 <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	Highest Sub-index Supported
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	Highest Sub-index Supported
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	Highest Sub-index Supported
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	Highest Sub-index Supported
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> und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.
6	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

## 605Bh Shutdown Option Code

### Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der **CiA 402 Power State Machine** vom Zustand *Operation enabled* in den Zustand *Ready to switch on*.

### Objektbeschreibung

Index	605B <sub>h</sub>
Objektnamen	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	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart) und anschließendem Zustandswechsel zu "Switch on disabled"
2 bis 32767	Reserviert

## 605Ch Disable Option Code

### Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der **CiA 402 Power State Machine** vom Zustand *Operation enabled* in den Zustand *Switched on*.

### Objektbeschreibung

Index	605C <sub>h</sub>
Objektnamen	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	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart) und anschließendem Zustandswechsel zu "Switch on disabled"
2 bis 32767	Reserviert

## 605Dh Halt Option Code

### Funktion

Das Objekt enthält die auszuführende Aktion, wenn im Controlword **6040<sub>h</sub>** das Bit 8 (Halt) gesetzt wird.

### Objektbeschreibung

---

Index	605D <sub>h</sub>
Objektnamen	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	Beschreibung
-32768 bis 0	Reserviert
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
2	Abbremsen mit "quick stop ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
3 bis 32767	Reserviert

## 605Eh Fault Option Code

### Funktion

Das Objekt enthält die auszuführende Aktion, wie der Motor im Fehlerfall zum Stillstand gebracht werden soll.

### Objektbeschreibung

Index	605E <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	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt
1	Abbremsen mit "slow down ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
2	Abbremsen mit "quick stop ramp" (Verzögerung (Bremsrampe) je nach Betriebsart)
3 bis 32767	Reserviert

## 6060h Modes Of Operation

### Funktion

In dieses Objekt wird der gewünschte Betriebsmodus eingetragen.

### Objektbeschreibung

Index	6060 <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 Drehgeberposition in 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.



#### Hinweis

Ist die Encoderauflösung im Objekt **608F<sub>h</sub>** = 0, 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**.

### 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 **6066<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 **6066<sub>h</sub>** definierte Zeit.

## 606Bh Velocity Demand Value

### Funktion

Vorgabegeschwindigkeit in **benutzerdefinierten Einheiten** für den Regler im **Profile Velocity Mode**.

### 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>
Objektnamen	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>
Objektnamen	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 **6066<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 **6066** definierte Zeit (siehe auch **Statusword im Modus Profile Velocity**).

## 6071h Target Torque

### Funktion

Dieses Objekt enthält das Zieldrehmoment für den **Profile Torque** und **Cyclic Synchronous Torque** Modus in Promille des 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

Gibt 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%) 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>
Objektnamen	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>
Objektnamen	Position Range Limit
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Applikation
Firmware Version	FIR-v1426
Änderungshistorie	

---

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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	Highest Sub-index Supported
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**
- **Velocity Mode**

### POS (Position)

Umkehr der Drehrichtung in folgenden Modi:

- **Profile Position Mode**
- **Cyclic Synchronous Position Mode**





### Tip

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 für den Modus **Profile Position**, **Interpolated Position Mode** und **Profile Velocity** in **benutzerdefinierten Einheiten** an.

### Objektbeschreibung

Index	607F <sub>h</sub>
Objektnamen	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

Gibt die maximal zulässige Geschwindigkeit des Motors in **benutzerdefinierten Einheiten** an.

### Objektbeschreibung

Index	6080 <sub>h</sub>
Objektnamen	Max Motor Speed
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	ja, Kategorie: Bewegung

---

Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00007530 <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1614: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning".  Firmware Version FIR-v1738-B501312: Eintrag "Object Name" geändert von "Maximum Speed" auf "Max Motor Speed".  Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 00 geändert von "nein" auf "RX-PDO".  Firmware Version FIR-v1748-B538662: Eintrag "Speicherbar" geändert von "ja, Kategorie: Tuning" auf "ja, Kategorie: Bewegung".

---

## 6081h Profile Velocity

### Funktion

Gibt die maximale Fahrgeschwindigkeit in **benutzerdefinierten Einheiten** an.

### Objektbeschreibung

---

Index	6081 <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>
Objektnamen	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.

### Objektbeschreibung

Index	6084 <sub>h</sub>
Objektnamen	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.

### 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	00000000 <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.

## 608Fh Position Encoder Resolution

### Funktion

Enthält die physikalische Auflösung des Encoders/Sensors, der für die Positionsregelung verwendet wird.

### Objektbeschreibung

Index	608F <sub>h</sub>
Objektname	Position Encoder Resolution
Object Code	ARRAY
Datentyp	INTEGER32
Speicherbar	ja, Kategorie: Tuning
Firmware Version	FIR-v1426
Änderungshistorie	Firmware Version FIR-v1738-B501312: Eintrag "Speicherbar" geändert von "ja, Kategorie: Applikation" auf "ja, Kategorie: Tuning". Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 01 geändert von "nein" auf "RX-PDO".

Firmware Version FIR-v1738-B501312: Tabellen-Eintrag "PDO-Mapping" bei Subindex 02 geändert von "nein" auf "RX-PDO".

Firmware Version FIR-v1748-B538662: Eintrag "Datentyp" geändert von "UNSIGNED32" auf "INTEGER32".

Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".

Firmware Version FIR-v1748-B538662: Eintrag "Data type" geändert von "UNSIGNED32" auf "INTEGER32".

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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	000007D0 <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 des Encoders/Sensors, der für die Drehzahlregelung verwendet wird.

## 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>

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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	00000000 <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

Anzahl der Motorumdrehungen pro Umdrehung der Abtriebsachse.

### 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	Highest Sub-index Supported
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

Vorschub im Falle eines Linearantriebs, in **benutzerdefinierten Einheiten** pro Umdrehungen der Abtriebsachse.

### 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	Highest Sub-index Supported
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	Highest Sub-index Supported
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	Highest Sub-index Supported
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	Highest Sub-index Supported
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. Die Geschwindigkeit in Subindex 2 soll daher unter 1000 Schritten pro Sekunde sein.
- Die Geschwindigkeit in Subindex 1 muss größer sein als die Geschwindigkeit in Subindex 2.

## 609Ah Homing Acceleration

### Funktion

Gibt die Beschleunigungsrampe für den Homing Mode in **benutzerdefinierten Einheiten** an.

### Objektbeschreibung

Index	609A <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	Highest Sub-index Supported
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	Firmware Version FIR-v1614: Eintrag "Name" geändert von "End Acceleration Jerk" auf "Begin Deceleration Jerk".  Firmware Version FIR-v1614: Eintrag "Name" geändert von "Begin Deceleration Jerk" auf "End Acceleration Jerk".

## Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
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>



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**)

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

## 60A9h SI Unit Velocity

### Funktion

Dieses Objekt beinhaltet die Geschwindigkeitseinheit. Siehe Kapitel **Benutzerdefinierte Einheiten**.

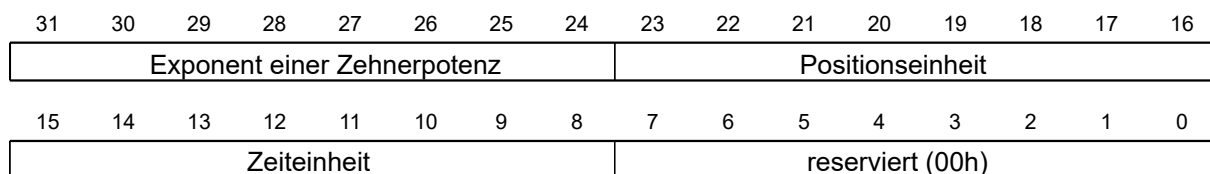
### Objektbeschreibung

Index	60A9 <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 in den Modi **Cyclic Synchronous Position**, und **Takt-Richtungs-Modus** 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	Highest Sub-index Supported
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	Highest Sub-index Supported
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	Highest Sub-index Supported
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 maximale 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 den Modus **Profile Position** und **Profile Velocity**.

### 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	Highest Sub-index Supported
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*.

## 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	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	Highest Sub-index Supported
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*.

## 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	Firmware Version FIR-v1748-B538662: Die Anzahl der Einträge haben sich geändert von 5 auf 4.

### Wertebeschreibung

Subindex	00 <sub>h</sub>
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	03 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00320000 <sub>h</sub>
Subindex	02 <sub>h</sub>
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000000C8 <sub>h</sub>
Subindex	03 <sub>h</sub>
Name	Additional Position Encoder Resolution - Encoder Increments Feedback Interface #3
Datentyp	INTEGER32
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 Inkremente der entsprechenden Rückführung.  
Subindex 01<sub>h</sub> entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

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>)

## 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	Highest Sub-index Supported
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 - 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 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	Highest Sub-index Supported
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 Vorschubkonstante der Rückführung "n" berechnet sich wie folgt:

Feed Constant = Feed (60E9<sub>h</sub>:n<sub>h</sub>) / Driving Shaft Revolutions (60EE<sub>h</sub>:n<sub>h</sub>)

## 60EBh 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	Highest Sub-index Supported
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 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>)

## 60ED<sub>h</sub> 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	Highest Sub-index Supported
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.



- $n_h$ : 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 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	Highest Sub-index Supported
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:

- $00_{h_i}$ : Wert= "n", wo "n" die Anzahl der vorhandenen Rückführungen.
- $n_{h_i}$ : Subindex "n" enthält die Anzahl der Umdrehungen der Abtriebsachse für die entsprechende Rückführung.  
Subindex  $01_{h_i}$  entspricht immer der ersten (und immer vorhandenen) Rückführung *Sensorless*.

Die Vorschubkonstante der Rückführung "n" berechnet sich wie folgt:

$$\text{Feed Constant} = \text{Feed} (60E9_{h_i:n_{h_i}}) / \text{Driving Shaft Revolutions} (60EE_{h_i:n_{h_i}})$$

## 60F2h Positioning Option Code

### Funktion

Das Objekt beschreibt das Positionierverhalten im **Profile Position** Modus.

### Objektbeschreibung

Index	60F2 <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 <b>6064<sub>h</sub></b> ) ausgeführt.

Bit 1	Bit 0	Definition
1	1	Reserviert

### RRO (Request-Response Option)

Diese Bits bestimmen das Verhalten bei der Übergabe des Controlwords **6040<sub>h</sub>** Bit 5 ("new setpoint") - die Steuerung übernimmt in diesem Fall die Freigabe des Bits selbständig. Damit fällt die Notwendigkeit weg, das Bit anschließend extern wieder auf "0" zu setzen. Nachdem das Bit von der Steuerung aus auf den Wert "0" gesetzt wurde, wird auch das Bit 12 ("setpoint acknowledgement") im Statusword **6041<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 <b>Setzen von Fahrbefehlen</b> beschrieben.
0	1	Die Steuerung wird das Bit "new setpoint" frei geben, sobald die momentane Zielfahrt ihr Ziel erreicht hat.
1	0	Die Steuerung wird das Bit "new setpoint" frei geben, sobald es der Steuerung möglich ist.
1	1	Reserviert

### RADO (Rotary Axis Direction Option)

Diese Bits bestimmen die Drehrichtung im "Profile Position" Modus.

Bit 7	Bit 6	Definition
0	0	Normale Positionierung ähnlich einer linearen Achse: Falls eines der "Position Range Limits" <b>607B<sub>h</sub>:01<sub>h</sub></b> und <b>02<sub>h</sub></b> 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 <b>607D<sub>h</sub>:01<sub>h</sub></b> 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 <b>607D<sub>h</sub>:01<sub>h</sub></b> zu der Zielposition.
1	1	Positionierung mit dem kürzesten Weg zur Zielposition. Falls die Differenz zwischen aktueller Position und Zielposition in einem 360° System kleiner als 180° ist, fährt die Achse in positiver Richtung.

## 60F4h Following Error Actual Value

### Funktion

Dieses Objekt enthält den aktuellen Schleppfehler in **benutzerdefinierten Einheiten**.

### Objektbeschreibung

---

Index	60F4 <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

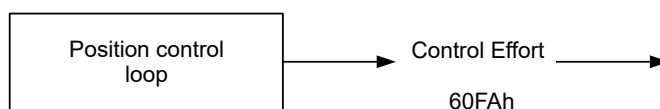
Dieses Objekt beinhaltet die Korrekturgeschwindigkeit in **benutzerdefinierten Einheiten**, die vom Positionsregler dem Geschwindigkeitsregler zugeführt wird.

### Objektbeschreibung

Index	60FA <sub>h</sub>
Objektnamen	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 (**3210<sub>h</sub>:01<sub>h</sub>**) und Integralanteil (**3210<sub>h</sub>:02<sub>h</sub>**) des Positionsreglers ab. Siehe auch Kapitel **Closed Loop**.



## 60FCh Position Demand Internal Value

### Funktion

Gibt die aktuelle Sollposition in Inkrementen an.

### Objektbeschreibung

Index	60FC <sub>h</sub>
Objektnamen	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	

## 60FDh 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
													HS	PLS	NLS

#### NLS (Negative Limit Switch)

negativer Endschalter

#### PLS (Positive Limit Switch)

positiver Endschalter

#### HS (Home Switch)

Referenzschalter

#### IN n (Input n)

Eingang n - die Anzahl der verwendeten Bits ist abhängig von der jeweiligen Steuerung.

## 60FEh Digital Outputs

### Funktion

Mit diesem Objekt können die **Digitalausgänge** des Motors geschrieben werden.

## Objektbeschreibung

Index	60FE <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	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO-Mapping	nein
Zulässige Werte	
Vorgabewert	01 <sub>h</sub>

Subindex	01 <sub>h</sub>
Name	Digital Outputs #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO-Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 <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.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												OUT4	OUT3	OUT2	OUT1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															BRK

### BRK (Brake)

Bit für den Bremsenausgang (falls der Controller diese Funktion unterstützt).

### OUT n (Output No n)

Bit für den jeweiligen digitalen Ausgang, die genaue Zahl der Digitalausgänge ist abhängig von der Steuerung.

## 60FFh Target Velocity

### Funktion

In dieses Objekt wird die Zielgeschwindigkeit für den **Profile Velocity** und **Cyclic Synchronous VelocityMode** in **benutzerdefinierten Einheiten** eingetragen.

### Objektbeschreibung

---

Index	60FF <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

---

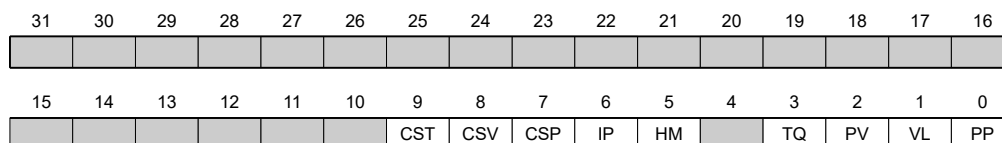
Index	6502 <sub>h</sub>
Objektname	Supported Drive Modes
Object Code	VARIABLE
Datentyp	UNSIGNED32
Speicherbar	nein
Zugriff	nur lesen
PDO-Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	000003EF <sub>h</sub>
Firmware Version	FIR-v1426
Änderungshistorie	

---

### Beschreibung

Ein gesetztes Bit gibt an, ob der jeweilige Modus unterstützt wird. Ist der Wert des Bits "0", wird der Modus nicht unterstützt.





**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	<a href="http://www.nanotec.de">http://www.nanotec.de</a>
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>