

Handbuch PD4-C

Feldbus: USB

Zur Benutzung mit folgenden Varianten:

- PD4-C5918M4204-E-01
- PD4-C6018L4204-E-01
- PD4-CB59M024035-E-01

Inhalt

1 Impressum.....	7
2 Sicherheits- und Warnhinweise.....	8
2.1 Wichtige Hinweise.....	8
2.2 Qualifikation des Personals.....	8
2.3 Gefahren- und Warnhinweise.....	8
2.4 Sonstige Hinweise.....	9
3 Zu diesem Handbuch.....	10
3.1 Einführung.....	10
3.2 Zahlenwerte.....	10
3.3 Bits.....	10
3.4 Zählrichtung (Pfeile).....	10
3.5 Versionshinweise.....	10
4 Technische Daten und Anschlussbelegung.....	12
4.1 Maßzeichnungen.....	12
4.2 Elektrische Eigenschaften.....	12
4.3 LED Signalisierung.....	13
4.4 Anschlussbelegung.....	14
5 Konfiguration.....	17
5.1 Allgemeines.....	17
5.2 DIP Schalter.....	17
5.3 USB Anschluss.....	18
5.4 Konfigurationsdatei.....	18
5.5 NanoJ-Programm.....	21
6 Inbetriebnahme.....	23
6.1 Sicherheitshinweise.....	23
6.2 Vorbereitung.....	23
7 Betriebsmodi.....	25
7.1 Profile Position.....	25
7.2 Velocity.....	33
7.3 Profile Velocity.....	35
7.4 Profile Torque.....	38
7.5 Homing.....	39
8 Generelle Konzepte.....	47
8.1 DS402 Power State machine.....	47
8.2 Benutzerdefinierte Einheiten.....	51
9 Spezielle Funktionen.....	54

9.1 Digitale Ein- und Ausgänge..... 54
9.2 I²t Motor-Überlastungsschutz..... 56

10 Programmierung mit NanoJ..... 59

10.1 Einleitung..... 59
10.2 Verfügbare Rechenzeit..... 59
10.3 Interaktion des Benutzerprogramms mit der Steuerung..... 59
10.4 OD-Einträge zur Steuerung und Konfiguration der VMM..... 60
10.5 NanoJEasyV2..... 61
10.6 Systemcalls..... 64

11 Objektverzeichnis Beschreibung..... 66

11.1 Übersicht..... 66
11.2 Aufbau der Objektbeschreibung..... 66
11.3 Objektbeschreibung..... 66
11.4 Wertebeschreibung..... 68
11.5 Beschreibung..... 69
1000h Device Type..... 69
1001h Error Register..... 70
1003h Pre-defined Error Field..... 71
1008h Manufacturer Device Name..... 74
1009h Manufacturer Hardware Version..... 75
100Ah Manufacturer Software Version..... 75
1010h Store Parameter..... 76
1011h Restore Default Parameter..... 77
1018h Identity Object..... 78
1600h Drive Control..... 79
1601h Positioning Control..... 82
1602h Velocity Control..... 84
1603h Output Control..... 86
1A00h Drive Status..... 89
1A01h Positioning Status..... 91
1A02h Velocity Status..... 94
1A03h Input Status..... 96
2010h IP-Configuration..... 99
2011h Static-IP-Address..... 100
2012h Static-IP-Subnet-Mask..... 101
2018h Current-IP-Address..... 102
2019h Current-IP-Subnet-Mask..... 103
2020h ApplInfo-Static-IP-Address..... 104
2021h ApplInfo-Static-IP-Subnet-Mask..... 105
2022h Drive Serial Number..... 106
2030h Pole Pair Count..... 106
2031h Peak Current..... 106
2032h Maximum Speed..... 107
2033h Plunger Block..... 107
2034h Upper Voltage Warning Level..... 108
2035h Lower Voltage Warning Level..... 108
2036h Open Loop Current Reduction Idle Time..... 109
2037h Open Loop Current Reduction Value/factor..... 109
2038h Brake Controller Timing..... 110
2039h Motor Currents..... 112
203Ah Homing On Block Configuration..... 113
203Bh I2t Parameters..... 115
2050h Encoder Alignment..... 117
2051h Encoder Optimization..... 117
2052h Encoder Resolution..... 118

2053h Index Polarity.....	119
2054h Index Width.....	119
2056h Limit Switch Tolerance Band.....	120
2057h Clock Direction Multiplier.....	120
2058h Clock Direction Divider.....	121
2059h Encoder Configuration.....	121
2060h Compensate Polepair Count.....	122
2061h Velocity Numerator.....	122
2062h Velocity Denominator.....	123
2063h Acceleration Numerator.....	123
2064h Acceleration Denominator.....	124
2065h Jerk Numerator.....	124
2066h Jerk Denominator.....	125
2084h Bootup Delay.....	125
2101h Fieldbus Module.....	126
2200h Sampler Control.....	127
2201h Sampler Status.....	127
2202h Sample Data Selection.....	128
2203h Sampler Buffer Information.....	130
2204h Sample Time In Ms.....	132
2300h VMM Control.....	132
2301h VMM Status.....	133
2302h VMM Error Code.....	133
2303h Number Of Active User Program.....	135
2304h Table Of Available User Programs.....	135
2310h VMM Input Data Selection.....	138
2320h VMM Output Data Selection.....	142
2330h VMM In/output Data Selection.....	146
2400h VMM Inputs.....	150
2500h VMM Outputs.....	157
2600h VMM Debug Output.....	164
2700h User Storage Area.....	177
2800h Bootloader Setting.....	181
3202h Motor Drive Submode Select.....	182
3208h Motor Drive Sensor Source Open Loop.....	183
3209h Motor Drive Sensor Source Closed Loop.....	185
320Ah Motor Drive Sensor Display Open Loop.....	186
320Bh Motor Drive Sensor Display Closed Loop.....	188
3210h Motor Drive Parameter Set.....	189
3220h Analog Inputs.....	192
3221h Analogue Inputs Control.....	193
3240h Digital Inputs Control.....	194
3250h Digital Outputs Control.....	196
3700h Following Error Option Code.....	198
603Fh Error Code.....	198
6040h Controlword.....	199
6041h Statusword.....	200
6042h VI Target Velocity.....	201
6043h VI Velocity Demand.....	201
6044h VI Velocity Actual Value.....	202
6046h VI Velocity Min Max Amount.....	202
6048h VI Velocity Acceleration.....	204
6049h VI Velocity Deceleration.....	205
604Ah VI Velocity Quick Stop.....	206
604Ch VI Dimension Factor.....	207
605Ah Quick Stop Option Code.....	208
605Bh Shutdown Option Code.....	208
605Ch Disable Option Code.....	209
605Dh Halt Option Code.....	210

605Eh Fault Option Code.....	210
6060h Modes Of Operation.....	211
6061h Modes Of Operation Display.....	212
6062h Position Demand Value.....	212
6063h Position Actual Internal Value.....	213
6064h Position Actual Value.....	213
6065h Following Error Window.....	214
6066h Following Error Time Out.....	214
6067h Position Window.....	215
6068h Position Window Time.....	215
606Bh Velocity Demand Value.....	216
606Ch Velocity Actual Value.....	216
606Dh Velocity Window.....	217
606Eh Velocity Window Time.....	217
6071h Target Torque.....	218
6072h Max Torque.....	218
6074h Torque Demand.....	218
607Ah Target Position.....	219
607Bh Position Range Limit.....	219
607Ch Home Offset.....	220
607Dh Software Position Limit.....	221
607Eh Polarity.....	222
6081h Profile Velocity.....	223
6082h End Velocity.....	223
6083h Profile Acceleration.....	223
6084h Profile Deceleration.....	224
6085h Quick Stop Deceleration.....	224
6086h Motion Profile Type.....	225
6087h Torque Slope.....	225
608Fh Position Encoder Resolution.....	226
6091h Gear Ratio.....	227
6092h Feed Constant.....	228
6098h Homing Method.....	229
6099h Homing Speed.....	229
609Ah Homing Acceleration.....	230
60A4h Profile Jerk.....	231
60C2h Interpolation Time Period.....	232
60C5h Max Acceleration.....	233
60C6h Max Deceleration.....	234
60F4h Following Error Actual Value.....	234
60FDh Digital Inputs.....	235
60FEh Digital Outputs.....	236
60FFh Target Velocity.....	237
6402h Read Analogue Input.....	237
6431h Analogue Input Offset.....	238
6432h Analogue Input Pre-scaling.....	239
6502h Supported Drive Modes.....	240
6505h Http Drive Catalogue Address.....	241

12 Copyrights..... 243

12.1 Einführung.....	243
12.2 AES.....	243
12.3 Arcfour (RC4).....	243
12.4 MD5.....	244
12.5 uIP.....	244
12.6 DHCP.....	244
12.7 CMSIS DSP Software Library.....	245
12.8 FatFs.....	245

12.9 Protothreads..... 245

1 Impressum

Copyright © 2014 Nanotec Electronic GmbH & Co. KG. All rights reserved.

Die Firmware unsere Controller kann Softwarebestandteile von Dritten enthalten. Die Lizenzbedingungen und Copyrights dieser Codebestandteile finden Sie im Kapitel **Copyrights**.

Nanotec[®] Electronic GmbH & Co. KG

Kapellenstraße 6

D-85622 Feldkirchen bei München

Tel.: +49 (0)89-900 686-0

Fax: +49 (0)89-900 686-50

Internet: www.nanotec.de

Alle Rechte vorbehalten!

MS-Windows 98/NT/ME/2000/XP/7 sind eingetragene Warenzeichen der Microsoft Corporation.

Original-Programmierhandbuch

2 Sicherheits- und Warnhinweise

2.1 Wichtige Hinweise

Vor der Installation und Inbetriebnahme der Steuerung ist dieses Technische Handbuch sorgfältig durchzulesen.

Nanotec[®] behält sich im Interesse seiner Kunden das Recht vor, technische Änderungen und Weiterentwicklungen von Hard- und Software zur Verbesserung der Funktionalität dieses Produktes ohne besondere Ankündigung vorzunehmen.

Dieses Handbuch wurde mit der gebotenen Sorgfalt zusammengestellt. Es dient ausschließlich der technischen Beschreibung des Produktes und der Anleitung zur Inbetriebnahme. Die Gewährleistung erstreckt sich gemäß unseren allgemeinen Geschäftsbedingungen ausschließlich auf Reparatur oder Umtausch defekter Geräte, eine Haftung für Folgeschäden und Folgefehler ist ausgeschlossen. Bei der Installation des Gerätes sind die gültigen Normen und Vorschriften zu beachten.

Für Kritik, Anregungen und Verbesserungsvorschläge wenden Sie sich an die oben angegebene Adresse oder per Email an: **info@nanotec.de**

2.2 Qualifikation des Personals

Arbeiten an und mit diesem Produkt dürfen nur von Fachkräften vorgenommen werden

- welche den Inhalt des Handbuches kennen und verstanden haben
- welche eine Ausbildung oder die entsprechende Erfahrung haben, eventuellen Gefahren abzuschätzen, hervorzusehen oder erkennen zu können, welche bei der Benutzung der Steuerung entstehen können
- welche alle geltenden Normen, Bestimmungen und Unfallverhütungsvorschriften kennen, welche beim Arbeiten am und mit dem Produkt beachtet werden müssen
- welche die Personensicherheit beim Einsatz der Steuerung in einem Gesamtsystem sicher stellen können

Der Betrieb darf nur mit den spezifizierten Kabeln und dem entsprechenden Zubehör erfolgen. Verwenden Sie nur Original-Zubehör und Original-Ersatzteile.

2.3 Gefahren- und Warnhinweise

Alle in dieser Dokumentation aufgelisteten Hinweise sind in einheitlicher Form abgedruckt. Je nach Grad der Gefährdung des Anwenders oder der Steuerung wird eine Gefährdungssituation in die nachfolgenden Klassen eingeteilt.

 GEFAHR
Der Hinweis mit GEFAHR verweist auf eine unmittelbar gefährliche Situation, die bei Missachtung des Hinweises unabwendbar zu einem schweren oder tödlichen Unfall führen wird.

 WARNUNG
Der Hinweis WARNUNG verweist auf eine eventuell gefährliche Situation, die bei Missachtung des Hinweises möglicherweise zu einem schweren oder tödlichen Unfall oder zu Beschädigungen an diesem Gerät oder anderen Geräten führen kann.

 **VORSICHT**

Der Hinweis VORSICHT verweist auf eine eventuell gefährliche Situation, die bei Missachtung des Hinweises **möglicherweise** zu einem Unfall oder zu Beschädigungen an diesem Gerät oder anderen Geräten führen kann.

VORSICHT

Der Hinweis VORSICHT ohne das Warnsymbol verweist auf eine möglicherweise gefährliche Situation, die bei Missachtung des Hinweises **möglicherweise** zu Beschädigungen an diesem Gerät oder anderen Geräten führen kann.

2.4 Sonstige Hinweise

In dieser Dokumentation werden noch folgende weitere Hinweiskategorien benutzt:

Tipp Dieses Feld weist auf eine Möglichkeit hin, Arbeiten zu vereinfachen.

Hinweis

Dieses Feld weist auf eventuelle Fehlerquellen oder Verwechslungsgefahren hin.

Beispiel

Dieses Feld enthält ein Beispiel.

3 Zu diesem Handbuch

3.1 Einführung

Dieses Handbuch richtet sich an Programmierer, die eine Motorsteuerung mit Hilfe der Steuerung von Nanotec® programmieren wollen.

3.2 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>

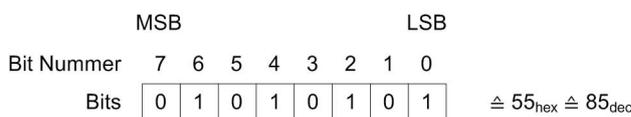
Sowohl der Index, als auch der Subindex werden in hexadezimaler Schreibweise angegeben. Sollte kein Subindex notiert sein, gilt der Subindex 0.

Beispiel: der Subindex 5 des Objektes 1003_h wird adressiert mit "1003_h:5_h", der Subindex 0 des Objektes 6040_h mit "6040_h".

Im letzten Teil des Handbuchs werden alle Objekte vollständig aufgelistet, die Referenzen im Fließtext oder in Tabellen werden im Schriftschnitt fett gesetzt, z. B. **6040_h**.

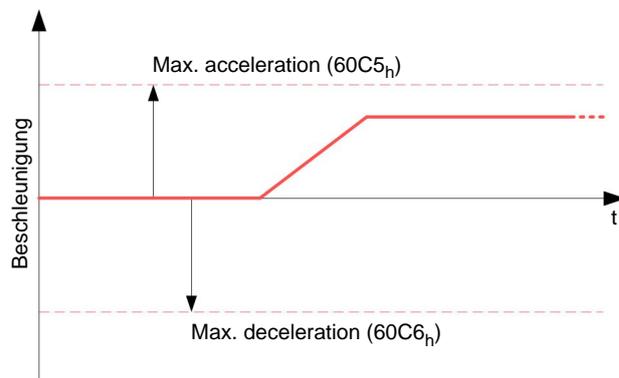
3.3 Bits

Einzelne Bits in einem Objekt beginnen bei der Nummerierung immer bei dem LSB mit 0. Siehe nachfolgende Abbildung am Beispiel eines Datentyps "UNSIGNED8".



3.4 Zählrichtung (Pfeile)

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



3.5 Versionshinweise

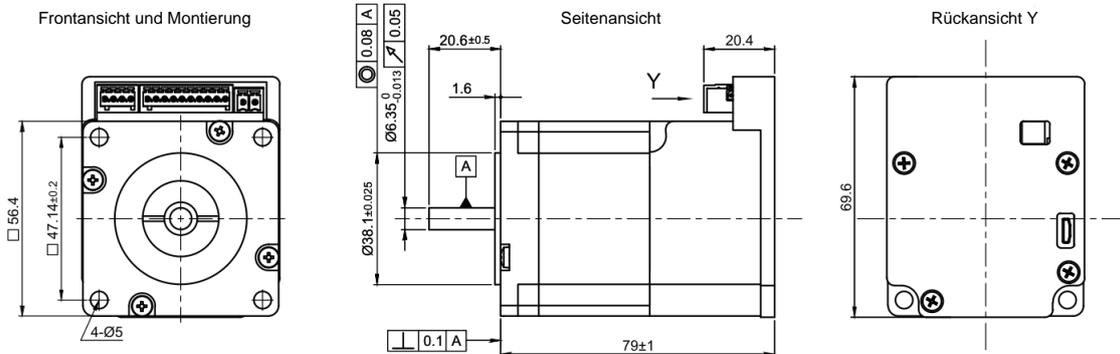
Version	Datum	Änderungen
1.0.0	03.03.2014	Veröffentlichung

Version	Datum	Änderungen
1.0.3	12.05.2014	Kleinere Verbesserungen und Korrekturen, Feld "Vorgabewert" nun belegt

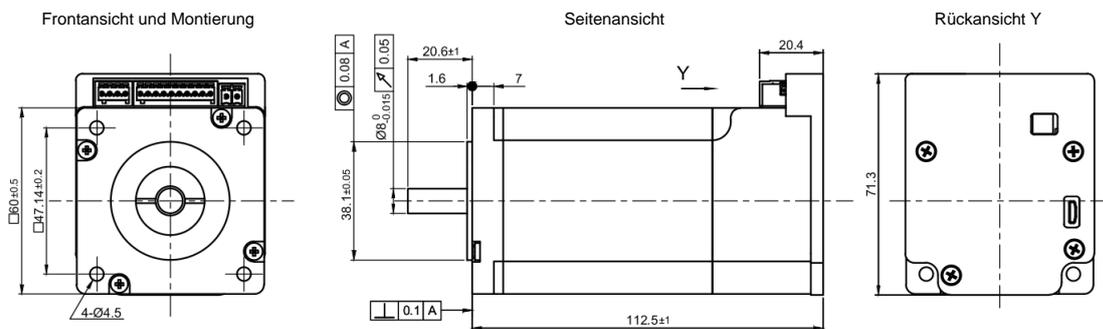
4 Technische Daten und Anschlussbelegung

4.1 Maßzeichnungen

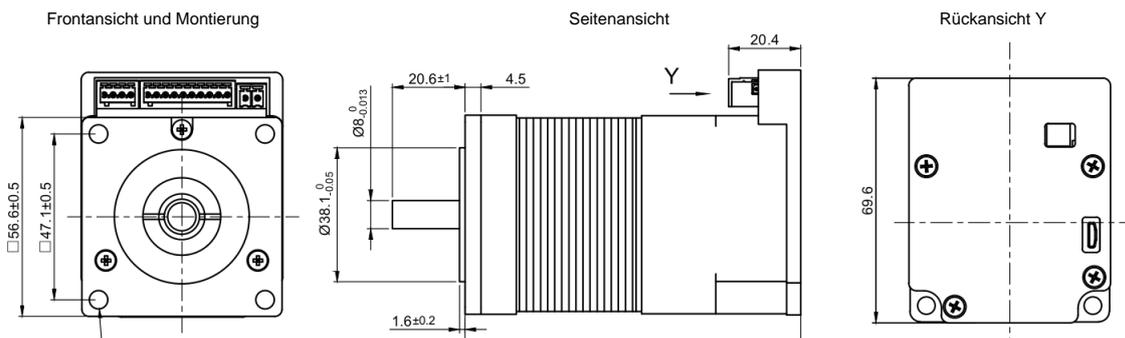
4.1.1 PD4-C5918M4204-E-01



4.1.2 PD4-C6018L4204-E-01



4.1.3 PD4-CB59M024035-E-01



4.2 Elektrische Eigenschaften

4.2.1 Technische Daten Motor

	PD4-C	PD4-CB
Art	Hochpoliger DC-Servo (Schrittmotor)	Niedrigpoliger DC-Servo (BLDC)
Betriebsspannung	12 V bis 48 V	12 V bis 24 V
Phasenstrom eff.	4,2 A	8 A

	PD4-C	PD4-CB
RMS für 1s	max. 6,3 A	max. 20 A

4.2.2 Technische Daten I/O

Ausführung	USB
Betriebsarten	Drehmoment, Drehzahl, Position, Homing
Sollwertvorgabe/ Programmierung	Takt-Richtung, Analogeingang, NanoJ V2, USB
Eingänge	Single/Differential Takt/Richtung/Enable (+5V / +24 V) 3 Digitaleingänge (+24 V) 1 Analogeingang (0 V bis 10 V)
Ausgänge	1 Ausgang, max. 0,5 A, Open Drain
Integrierter Encoder	Single Turn, magnetischer Absolut-Encoder, 1024 Impulse/U.

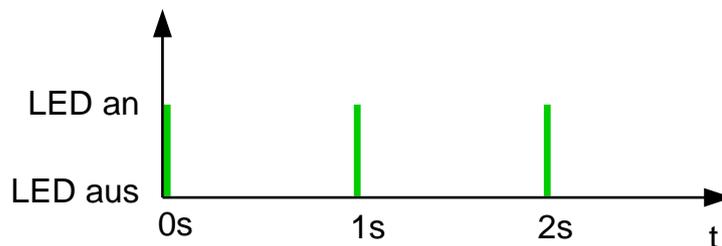
4.2.3 Leistungsgrößen

Typ	Haltemoment Ncm	Nenn-/ Spitzenmoment Ncm	Nenn-drehzahl (U/min)	Länge mm	Gewicht kg
PD4-C5918M4204- E-01/-08	110			79	0,8
PD4-C6018L4204- E-01/-08	350			112,5	1,5
PD4-CB59M024035- E-01	370	37 (110)	3500	95	0,9

4.3 LED Signalisierung

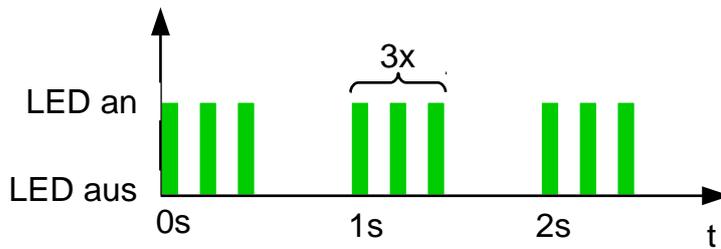
4.3.1 Normaler Betrieb

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



4.3.2 Fehlerfall

Sollte ein Fehler vorliegen, wird eine Fehlernummer innerhalb einer Sekunde mit der LED angezeigt. In der folgenden Darstellung wird der Fehler mit der Nummer 3 signalisiert.



Die Bedeutung der Fehlernummer ist in der nachfolgenden Tabelle abgedruckt.

Anzahl Blink	Fehler
1	Allgemein
2	Spannung
3	Temperatur
4	Überstrom
5	Regler

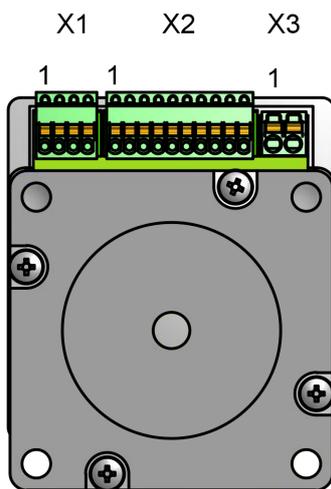
Hinweis

Für jeden aufgetretenen Fehler wird im Objekt **1003_n** ein wesentlich genauerer Fehlercode hinterlegt.

4.4 Anschlussbelegung

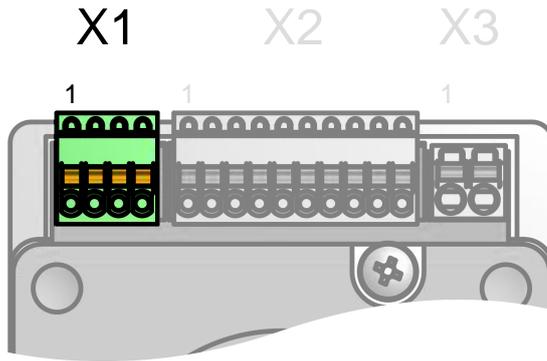
4.4.1 Übersicht

Die nachfolgende Abbildung zeigt die Steuerung mit Blick auf die Welle.



4.4.2 Analog-Eingang (Stecker X1)

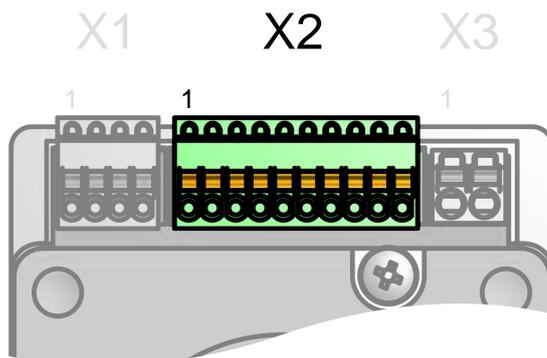
Anschlüsse für Analog-Modus



Pin	Beschreibung
1	GND
2	Analog input (0 V - 10 V)
3	Output (open drain)
4	+12 V (voltage output, max. 100 mA)

4.4.3 Takt-Richtungseingänge (Stecker X2)

Anschlüsse für die analogen Eingänge sowie Eingänge für die Takt-Richtungs-Steuerung.



Pin	Beschreibung
1	Input1 (+24 V)
2	Input2 (+24 V)
3	Input3 (+24 V)
4	-Enable +5 V / +24 V
5	Enable +5 V / +24 V
6	-Direction +5 V / +24 V
7	Direction +5 V / +24 V
8	-Clock +5 V / +24 V
9	Clock +5 V / +24 V
10	GND

4.4.4 Spannungsversorgung (Stecker X3)

Sicherheitshinweis

 **VORSICHT**

Gefahr vor elektrischer Überspannung!

- Eine Betriebsspannung, die höher ist, als die zulässige Betriebsspannung (siehe Kapitel "**Technische Daten Motor**") zerstört die Endstufe!
- Ein Vertauschen der Anschlüsse kann die Endstufe zerstören!
- Leitungen niemals unter Spannung verbinden oder trennen!
- Die Versorgungsspannung muss so gewählt werden, dass diese niemals die zulässige Betriebsspannung des Motors übersteigt. Speziell Störungen durch andere Verbraucher oder durch den Motor induzierte Spannungen sind hier in Betracht zu ziehen und es ist ggf. eine Spannung zu wählen, die eine ausreichend hohe Sicherheitsreserve bietet.

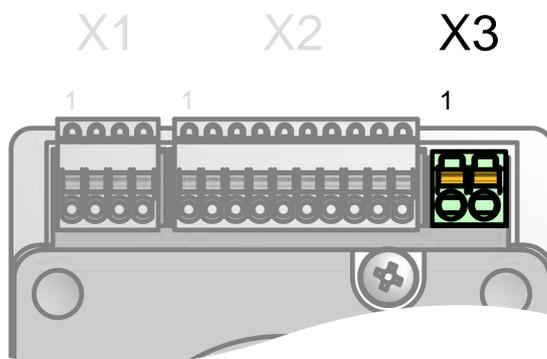
Spannungsquelle

Die Betriebs- oder Versorgungsspannung liefert eine Batterie (Kleinspannung 12 V – 24 V), ein Transformator mit Gleichrichtung und Siebung, oder besser ein Schaltnetzteil.

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	Beschreibung
1	+Vcc
2	GND

Zulässige Betriebsspannung

Die maximale Betriebsspannung beträgt nach Motortyp:

- 12 V bis 24 V für BLDC Motoren
- 12 V bis 48 V für Stepper Motoren

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

5 Konfiguration

5.1 Allgemeines

Es gibt folgende Möglichkeiten, die Steuerung zu konfigurieren:

DIP-Schalter

Auf der Rückseite sind vier DIP-Schalter angebracht. Näheres lesen Sie unter dem Kapitel "**DIP-Schalter**".

Konfigurationsdatei

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

VMM-Programm

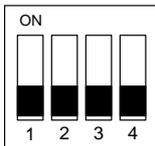
Dieses Programm lässt sich mit NanoJEasy programmieren, kompilieren und anschließend über USB auf die Steuerung übertragen. Lesen Sie dazu die Kapitel "**USB-Anschluss**" und "**VMM-Programm**".

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

1. Konfigurationsdatei wird ausgelesen und verarbeitet.
2. Die DIP-Schalter werden ausgelesen und als Konfiguration angewendet.
3. Das VMM Programm wird gestartet

5.2 DIP Schalter

Die Steuerung lässt sich mit DIP Schaltern auf der Rückseite konfigurieren. Die Grundeinstellung im Auslieferungszustand ist in der nachfolgenden Abbildung dargestellt.



Ein nach oben geschobener Schalter ist in der Position "Ein". Ein nach unten geschobener Schalter ist in der Position "Aus".

Schalter-Konfigurationen:

Schalter	Ein	Aus
1	Analog-Drehzahl-Modus	Takt/Richtungs-Modus; Vollschritte
2	<ul style="list-style-type: none"> • Im Takt/Richtungs-Modus: manuelle Fahrt in 30 U/min • Im Analog-Modus: Offset 5 V (Joystick Modus) 	<ul style="list-style-type: none"> • Im Takt/Richtungs-Modus: Standard Takt/Richtung Modus • Im Analog-Modus: Richtung über Direction Eingang
3	<ul style="list-style-type: none"> • Im Takt/Richtungs-Modus: manuelle Fahrt Richtung links • Im Analog-Modus: max-Drehzahl: 100 U/min 	<ul style="list-style-type: none"> • Im Takt/Richtungs-Modus: manuelle Fahrt Richtung rechts • Im Analog-Modus: max-Drehzahl 1000 U/min
4	Closed Loop-Modus	Open Loop-Modus

5.3 USB Anschluss

VORSICHT

- 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. pd4cfg.txt
 2. vmmcode.usr
 3. info.bin
 4. reset.txt
 5. firmware.bin

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

Hinweis

- Beim Anschließen des USB-Kabels wird der Motor zum Stillstand gebracht. Dazu wird der Modus "Swiched On" gesetzt (siehe dazu Kapitel "**DS402 Power State machine**")
- Die Spannungsversorgung der Steuerung muss beim USB-Betrieb ebenfalls angeschlossen sein.

Wird die Steuerung über ein USB-Kabel mit einem PC verbunden, verhält sich die Steuerung wie ein Wechseldatenträger. Sie können somit die Konfigurationsdatei oder das VMM-Programm auf die Steuerung speichern. Alle Änderungen an Dateien werden erst gültig, nachdem die Steuerung neu gestartet wurde (zum Beispiel mit kurzem Trennen von der Spannungsversorgung).

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

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

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

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

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

5.4 Konfigurationsdatei

5.4.1 Allgemeines

Lesen Sie vorab das Kapitel **USB-Anschluss**, falls noch nicht geschehen.

Die Konfigurationsdatei `pd4cfg.txt` dient dazu, Werte für das Objektverzeichnis beim Start auf einen bestimmten Wert vor zu belegen. Diese Datei ist in einer speziellen Syntax gehalten, um den Zugriff auf die Objekte des Objektverzeichnisses möglichst einfach zu gestalten. Die Steuerung wertet alle Zuweisungen in der Datei von oben nach unten aus.

Hinweis

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

5.4.2 Lesen und Schreiben der Datei

So erhalten Sie Zugriff auf die Datei:

1. Schließen Sie die Spannungsversorgung an den Anschluss X3 (siehe Kapitel "**Anschlussbelegung X3**") 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 oder vergleichbarem Editor das Verzeichnis der Steuerung an. Dort ist die Datei "`pd4cfg.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.
4. Verbinden Sie die Spannungsversorgung wieder. Mit diesem Start der Steuerung werden die neuen Werte der Konfigurationsdatei ausgelesen und wirksam.

Tipp Um die Steuerung neu zu starten, können Sie auch eine leere Datei `reset.txt` auf die Steuerung kopieren.

Damit startet die Steuerung neu. Die Datei `reset.txt` wird beim Neustart gelöscht.

5.4.3 Syntax

Kommentare

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

Beispiel

```
; Dies ist eine Kommentarzeile
```

Zuweisungen

VORSICHT

Informieren Sie sich **vor** dem Setzen eines Wertes über dessen Datentyp (siehe Kapitel **Objektverzeichnis**)! 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.

<Wert>

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

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.

Beispiel

Setzen des Objekts **6040_h:00** auf den Wert "6":

```
6040:00=0006
```

5.4.4 Bedingte Auswertung

Die DIP-Schalter können dazu benutzt werden, nur bestimmte Zuweisungen auszuführen. Folgende Syntax wird zum bedingten Ausführen benutzt:

```
#<Nr>:<Zuweisung>
```

<Nr>

Hier wird die Nummer des DIP-Schalters angegeben, wie sie auf den Schaltern aufgedruckt ist. Gültige Werte sind 1 bis 4

<Zuweisung>

Hier wird die Zuweisung angegeben, wie im Abschnitt "**Zuweisungen**" beschrieben.

Beispiel

Folgender Code setzt das Objekt **2057_h:00_h** "Clock Direction Multiplier" (Takt/Richtungs-Multiplikator):

- auf 1, wenn der DIP-Schalter 1 auf "Aus" geschaltet ist.

- auf 2, wenn der DIP-Schalter auf "Ein" geschaltet ist (der vorherige Wert wird überschrieben).

```
2057:00=00000001
#1:2057:00=00000002
```

5.5 NanoJ-Programm

Auf der Steuerung kann ein VMM Programm ausgeführt werden. Um ein Programm auf die Steuerung zu laden und zu starten gehen Sie dazu 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 den Anschluss X3 (siehe Kapitel "**Anschlussbelegung X3**") 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 Datei-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 nun die Datei "vmmcode.usr" auf die Steuerung.
7. Trennen Sie die Spannungsversorgung der Steuerung für ca. 1 Sekunde.
8. Verbinden Sie die Spannungsversorgung wieder. Mit diesem Start der Steuerung wird das neue VMM-Programm eingelesen und gestartet.

Tipp Um die Steuerung neu zu starten können Sie auch eine leere Datei `reset.txt` auf die Steuerung kopieren.

Damit startet die Steuerung neu. Die Datei `reset.txt` wird beim Neustart gelöscht.

Hinweis

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

Tipp Das Löschen des alten VMM-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 <ZIEL>/vmmcode.usr
```

6 Inbetriebnahme

6.1 Sicherheitshinweise

 **WARNUNG**

In einer Wohnumgebung kann dieses Produkt hochfrequente Störung verursachen, die Entstörmaßnahmen erforderlich machen können.

 **VORSICHT**

Elektromagnetische Wechselfelder!

Elektromagnetische Wechselfelder um die stromführenden Leitungen, insbesondere um die Versorgungs- und Motorleitungen, können den Motor und andere Geräte stören.

- Leitungen abschirmen. Den Anschluss des Schirms einseitig oder 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.

6.2 Vorbereitung

Zu Inbetriebnahme werden folgende Komponenten benötigt:

- Steuerung PD4C
- Spannungsversorgung entsprechend des Datenblattes

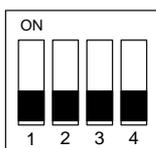
Entsprechend dem Modus der genutzt werden soll:

- Für den Analog-Modus: eine zusätzliche Spannungsquelle 0 V bis 10 V
- Für den Takt-Richtungs-Modus: Takt-Generator

6.2.1 Takt-Richtungs-Modus

Lesen Sie das Kapitel **Konfiguration** der Steuerung, falls noch nicht geschehen

1. Schalten Sie eine eventuell angeschlossene Spannungsversorgung ab.
2. Stellen Sie alle DIP-Schalter auf den Zustand "Aus" (entspricht der Grundeinstellung im Auslieferungszustand):



3. Schließen Sie die Spannungsversorgung an den Anschluss X3 der Steuerung an (siehe "**Spannungsversorgung (Stecker X3)**").
4. Schließen Sie den Taktgenerator an den Anschluss X2 an (siehe "**Takt-Richtungseingänge (Stecker X2)**").

Mit der Veränderung der Frequenz am Taktgenerator muss der Motor die Drehgeschwindigkeit ändern.

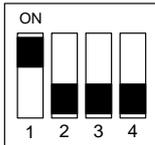
6.2.2 Analog-Modus

VORSICHT

Stellen Sie sicher, dass die Spannung am Analog-Eingang den Wert von 10 V nicht überschreitet.

Lesen Sie das Kapitel **Konfiguration** der Steuerung, falls noch nicht geschehen.

1. Schalten Sie eine eventuell angeschlossene Spannungsversorgung ab.
2. Die DIP-Schalter müssen - bis auf Schalter 1 - alle auf dem Zustand "Aus" stehen:



3. Schließen Sie die Spannungsversorgung an den Anschluss X3 der Steuerung an (siehe "**Spannungsversorgung (Stecker X3)**").
4. Schließen Sie die regelbare Spannungsquelle an den Anschluss X1 Pin 2 an (siehe "**Analog-Eingang (Stecker X1)**")

Mit Änderung der Spannung am Eingang für den Analogeingang muss der Motor die Drehzahl ändern.

7 Betriebsmodi

7.1 Profile Position

7.1.1 Besonderheit PD4C USB

Hinweis

Da diese Steuerung keinen Feldbus besitzt, ist der nachfolgende Betriebsmodus nur mit einem NanoJ Programm nutzbar.

Weitere Hinweise zur Programmierung und Benutzung eines NanoJ Programms findet sich im Kapitel **Programmierung mit NanoJ**.

7.1.2 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

Zum Aktivieren des Modus muss im Objekt **6060_h** (Modes Of Operation) der Wert "1" gesetzt werden (siehe "DS402 Power State machine").

Controlword

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

- Bit 4 startet einen Fahrauftrag. Dieser wird bei einem Übergang von "0" nach "1" ausgeführt.
- Bit 5: Ist dieses Bit auf "1" gesetzt, wird ein durch Bit 4 ausgelöster Fahrauftrag sofort ausgeführt. Ist es auf "0" gesetzt, wird der gerade ausgeführte Fahrauftrag zu Ende gefahren und erst im Anschluss der nächste Fahrauftrag gestartet.
- Bit 6: Bei "0" ist die Zielposition (**607A_h**) absolut und bei "1" ist die Zielposition relativ zur aktuellen Position.
- Bit 9: Ist dieses Bit gesetzt, so wird die Geschwindigkeit für erst beim Erreichen der ersten Zielposition geändert. Das bedeutet, dass vor Erreichen des ersten Ziels keine Bremsung durchgeführt wird, da der Motor auf dieser Position nicht stehen bleiben soll.

Controlword 6040 _h		
Bit 9	Bit 5	Definition
X	1	Die neue Zielposition wird sofort angefahren.
0	0	Das Positionieren wird erst vollständig abgeschlossen, bevor die nächste Zielposition mit den neuen Limitierungen angefahren wird.
1	0	Die aktuelle Geschwindigkeit wird bis zum Erreichen der momentanen Zielposition gehalten, erst dann wird die neue Zielposition mit den neuen Werten angefahren.

Siehe dazu das Bild in "**Setzen von Fahrbefehlen**".

Statusword

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

- Bit 10 (Target Reached): Dieses Bit ist auf "1" gesetzt, wenn das letzte Ziel erreicht wurde und der Motor eine vorgegebene Zeit (**6068_h**) innerhalb eines Toleranzfensters (**6067_h**) steht.
- 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 auftritt:

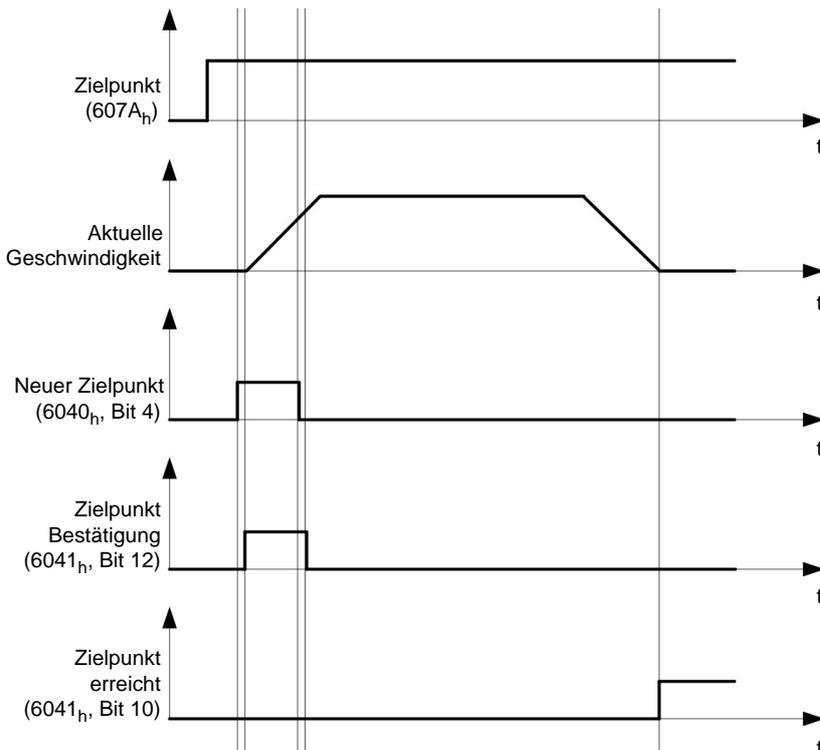
- Die neue Zielposition kann unter Einhaltung aller Randbedingungen nicht mehr erreicht werden.
- Es wird bereits eine Zielposition angefahren und zudem ist bereits eine Zielposition vorgegeben. Eine neue Zielposition lässt sich erst vorgeben, nachdem die aktuelle Positionierung abgeschlossen ist.
- Die neue Position ist außerhalb des gültigen Bereichs (**607D_h** (Software Position Limit)).
- Bit 13 (Following Error): Dieses Bit wird im Closed Loop-Betrieb gesetzt, wenn der Schleppfehler größer als die eingestellten Grenzen ist (**6065_h** (Following Error Window) und **6066_h** (Following Error Time Out)).

7.1.3 Setzen von Fahrbefehlen

Fahrbefehl

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

Profil des Fahrbefehls

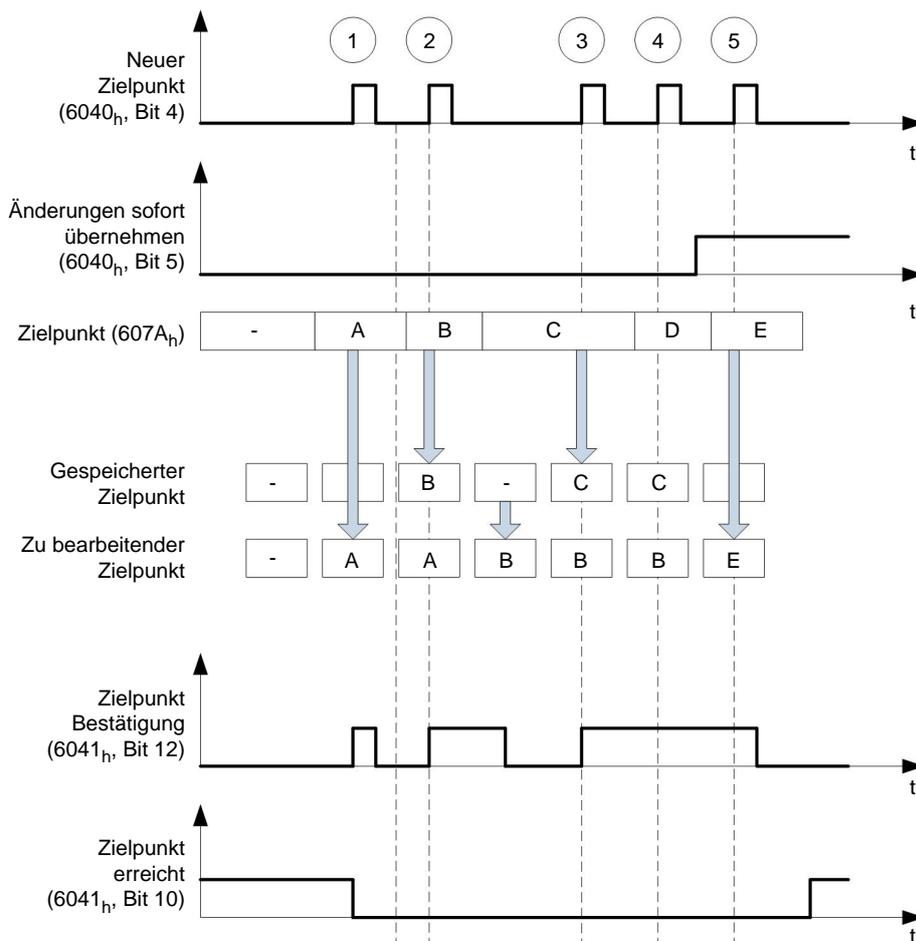


Weitere Fahrbefehle

Bit 12 im Objekt **6041_h** (Statusword, Set-point acknowledge) fällt auf "0", falls ein weiterer Fahrbefehl zwischengespeichert werden kann (siehe Zeitpunkt 1 im nachfolgenden Bild). Solange eine Zielposition angefahren wird, lässt sich eine zweite Zielposition vorbereitend an die Steuerung übergeben. Dabei können alle Parameter - wie Geschwindigkeit, Beschleunigung, Bremsbeschleunigung usw. - neu gesetzt werden (Zeitpunkt 2). Ist der Zwischenspeicher wieder leer, lässt sich der nächste Zeitpunkt einreihen (Zeitpunkt 3).

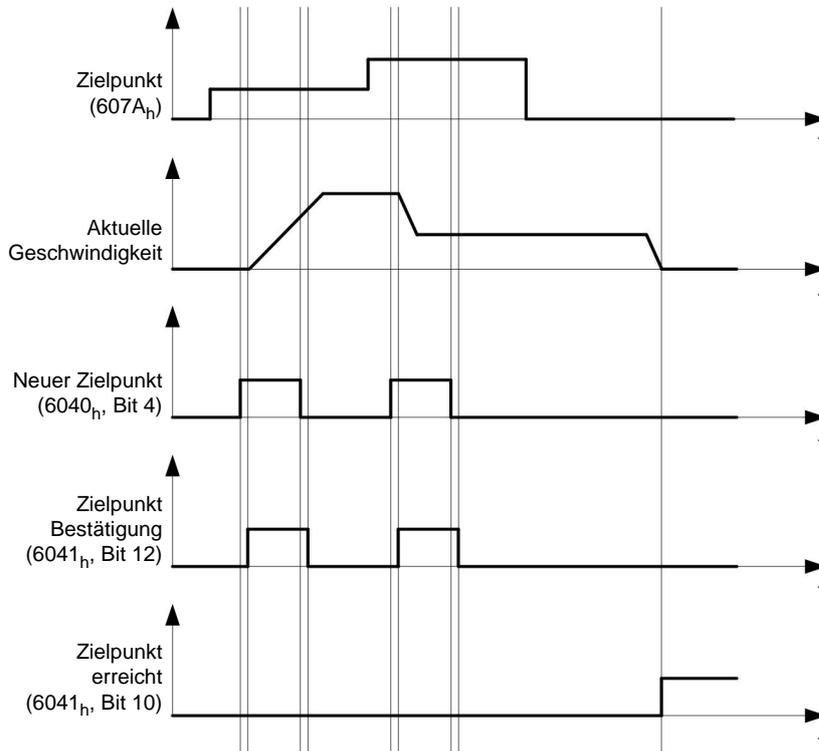
Sollte der Zwischenspeicher schon voll sein, wird ein neuer Zielpunkt ignoriert (Zeitpunkt 4). Wird Bit 5 im Objekt **6040_h** (Controlword, Bit: "Change Set-Point Immediately") gesetzt, arbeitet die Steuerung ohne den Zwischenspeicher, neue Fahrbefehle werden direkt umgesetzt (Zeitpunkt 5).

Zeitpunkte



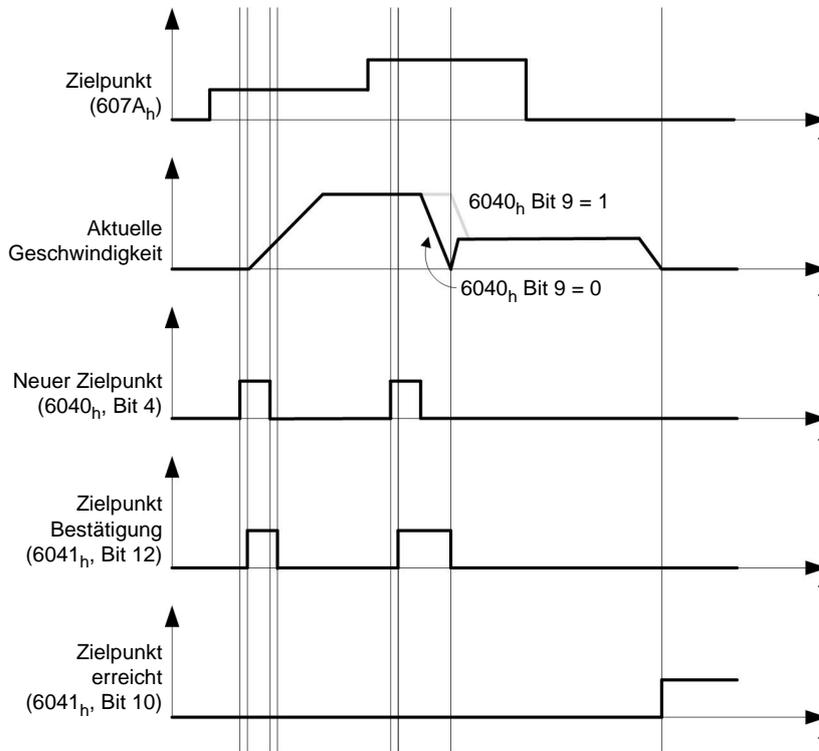
Übergangsprozedur für zweite Zielposition

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



Möglichkeiten zum Anfahren einer Zielposition

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



7.1.4 Randbedingungen für eine Positionierfahrt

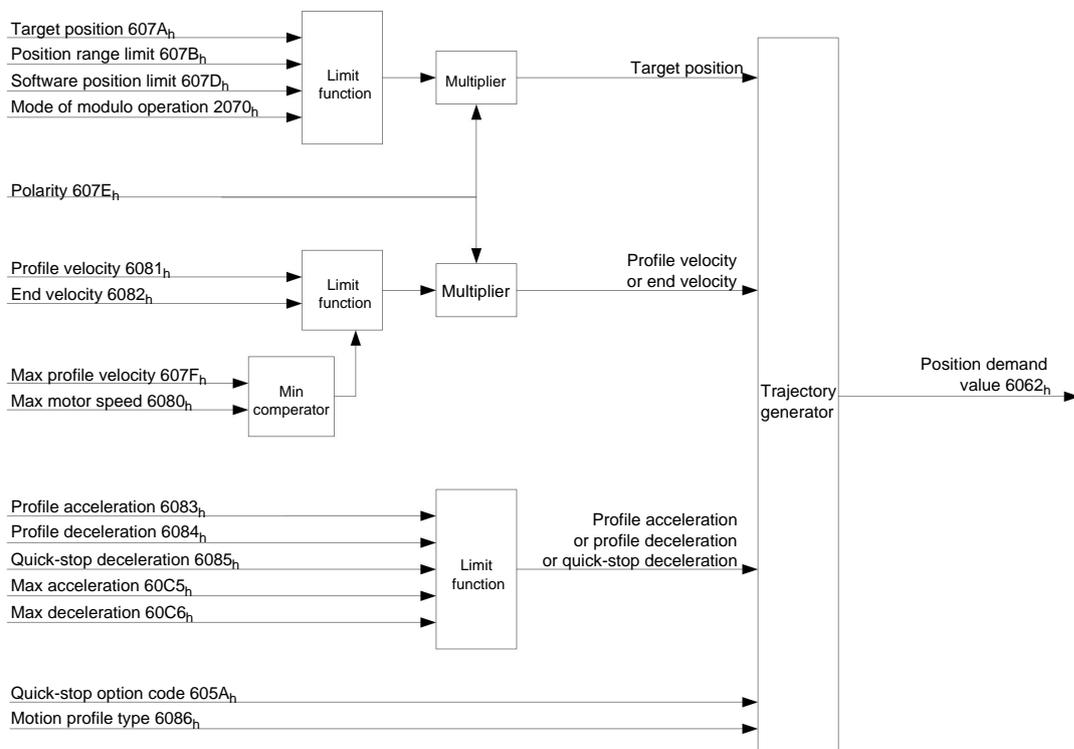
Objekteinträge

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

- **6064_h** (Position Actual Value): derzeitige Position des Motors
- **607A_h** (Target Position): vorgesehene Zielposition
- **607B_h** (Position Range Limit): Definition der Endanschläge (siehe Kapitel weiter unten)
- **607C_h** (Home Offset): Verschiebung des Maschinen-Nullpunkts (**siehe "Homing Mode"**)
- **607D_h** (Software Position Limit): Grenzen einer modulo-Operation zur Nachbildung einer endlosen Rotationsachse (**siehe "Achsenkalibrierung"**)
- **607E_h** (Polarity): Drehrichtung
- **6081_h** (Profile Velocity): maximale Geschwindigkeit, mit der die Position angefahren werden soll
- **6082_h** (End Velocity): Geschwindigkeit beim Erreichen der Zielposition
- **6083_h** (Profile Acceleration): gewünschte Anfahrbeschleunigung
- **6084_h** (Profile deceleration): gewünschte Bremsbeschleunigung
- **6085_h** (Quick Stop Deceleration): Nothalt-Bremsbeschleunigung im Falle des Zustandes "Quick stop active" der "DS402 Power State machine"
- **6086_h** (Motion Profile Type): Typ der zu fahrenden Rampe; ist der Wert "0", wird der Ruck nicht limitiert, ist der Wert "3", werden die Werte von 60A4_h:1_h- 4_h als Limitierungen des Rucks gesetzt.
- **60C5_h** (Max Acceleration): die maximale Beschleunigung, die beim Anfahren der Endposition nicht überschritten werden darf
- **60C6_h** (Max Deceleration): die maximale Bremsbeschleunigung, die beim Anfahren der Endposition nicht überschritten werden darf
- **60A4_h** (Profile Jerk), Subindex 01_h bis 04_h: Objekte zur Beschreibung der Grenzwerte für den Ruck

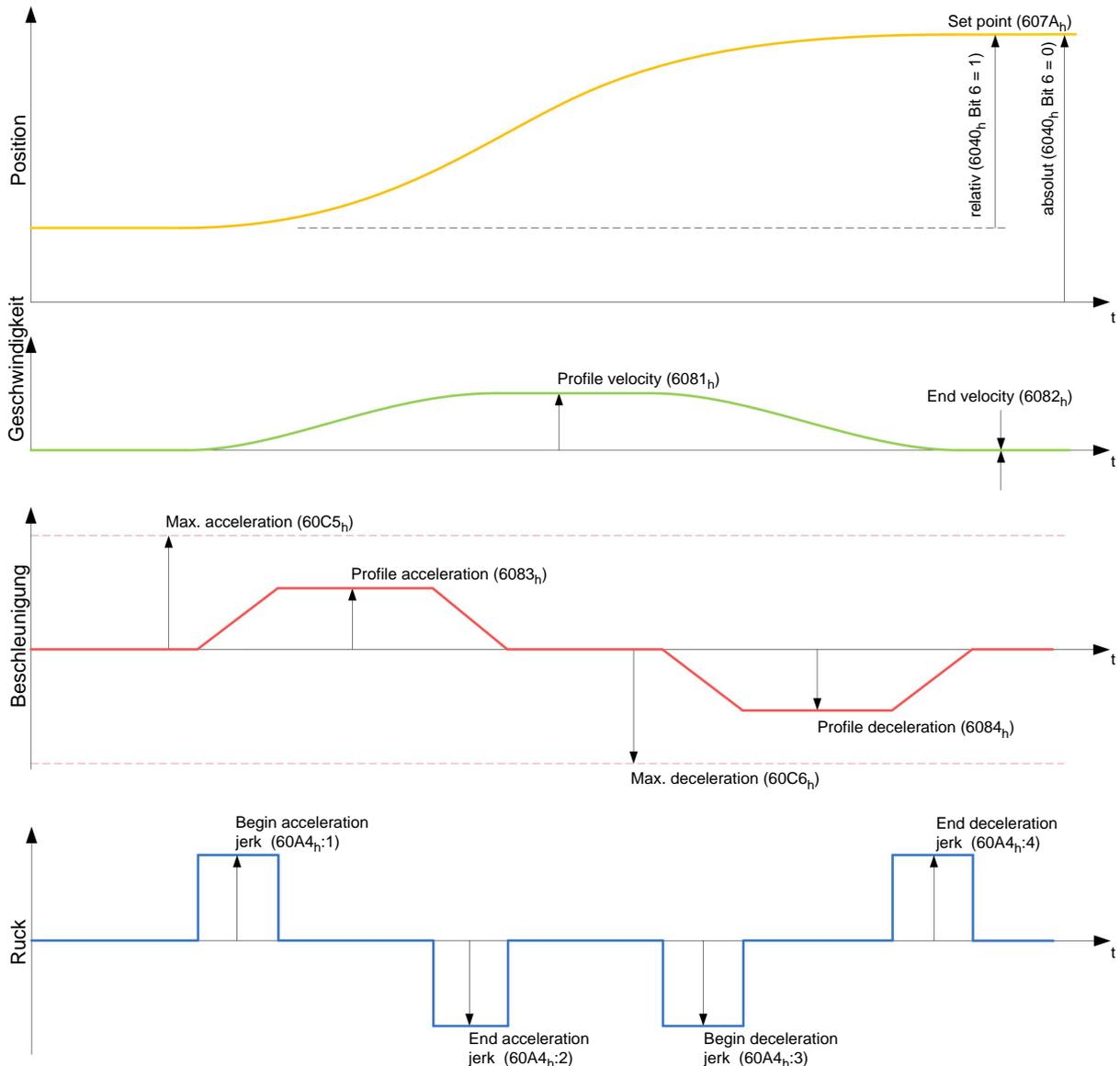
Objekte für die Positionierfahrt

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).



7.1.5 Ruck-begrenzter 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_h** auf "3" gesetzt wird. Damit werden die Einträge für die Rucke in Objekt **60A4_h:1_h - 4_h** gültig.

Nicht ruck-begrenzter Modus

Eine "0" in einem Eintrag bedeutet keine Ruck-Limitierung an der jeweiligen Stelle im Profil.

Sind alle vier Einträge von Objekt **60A4_h** auf "0" gesetzt, wird eine "nicht ruck-begrenzte" Rampe gefahren.

Eine "nicht ruck-begrenzte" Rampe wird gefahren, indem entweder alle Werte des Rucks in den Einträgen **60A4_h:1_h** bis **60A4_h:4_h** auf "0" und das Objekt **6086_h** auf "3" gesetzt werden oder der Eintrag im Objekt **6086_h** auf "0" gesetzt wird.

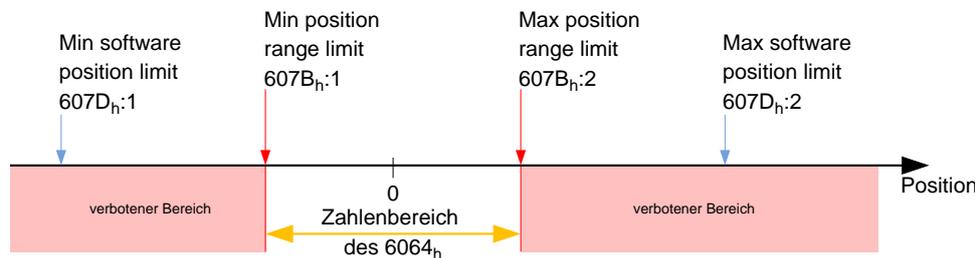
7.1.6 Achsenskalierung

Software- Endschalter

Die Steuerung berücksichtigt Software-Endschalter (**607D_h** (Software Position Limit)). Fahrbefehle über die Endschalter hinaus werden abgelehnt. Sollte sich der Motor beim Einrichten der Endschalter außerhalb des zulässigen Bereichs befinden, werden nur Fahrbefehle in Richtung des zulässigen Bereichs angenommen.

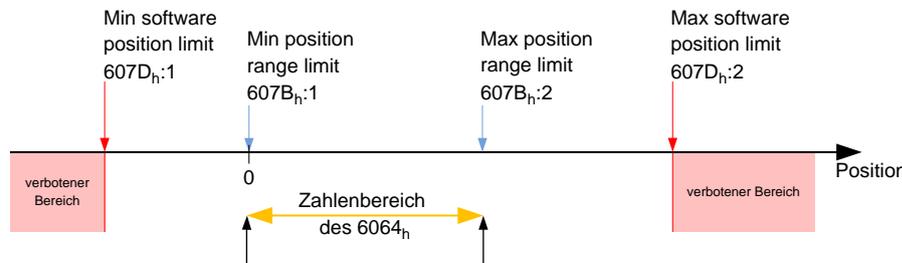
Linearachse

Für eine Linearachse müssen die Software-Endschalter **607D_h:1_h** und **607D_h:2_h** außerhalb des Wertebereichs der Endanschläge liegen.



Rotationsachse

Für eine Rotationsachse kann ein Software-Überlauf (bzw. Modulo-Operation) eingestellt werden. Damit beginnt die Angabe der Position im Objekt **6064_h** bei einem Über- oder Unterschreiten der Limits jeweils am gegenüberliegenden Ende des Bereichs. Die Angabe der Zielposition im Objekt **607A_h** kann dabei nach wie vor auf dem absoluten Zahlenstrahl für die Position erfolgen, nur die Angabe der Position im Objekt **6064_h** wird beeinflusst.



Drehrichtung

In diesem Modus ist die resultierende Drehrichtung nicht immer eindeutig. Daher kann über die Modulo-Operation **2070_h** (Mode Of Modulo Operation) spezifiziert werden, wie sich die Steuerung verhalten soll:

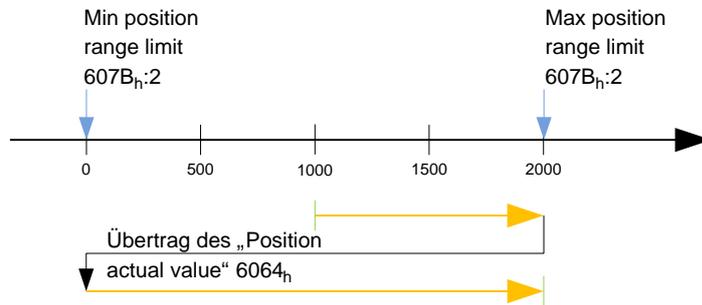
- 0: Überlauf mitzählen
- 1: immer in positive Richtung drehen
- 2: immer in negative Richtung drehen
- 3: immer den kürzeren Weg fahren (Standardeinstellung)
- 4: immer den längeren Weg fahren

Überlauf mitzählen

Im normalen Modulo-Betrieb ist es nicht möglich, eine - oder mehr als eine - volle Umdrehung zu fahren. Im Modus "Überlauf mitzählen" (Objekt **2070_h** auf "0" gesetzt) hingegen werden diese Überläufe mitgerechnet.

Beispiel:

Der Motor befindet sich derzeit auf der Position 1000, der Unterlauf ist auf 0 und der Überlauf auf 2000 eingestellt. Ein Fahrbefehl um eine volle Umdrehung auf die Position 3000 wird normalerweise ignoriert, denn $(3000 \bmod 2000) = 1000$ und auf dieser Position steht der Motor bereits. Ist im Objekt **2070_h** der Wert "0" eingetragen, werden Fahrbefehle absolut gerechnet, die Ausgabe des Objekts **6040_h** erfolgt aber nach wie vor mit Über- und Unterläufen.

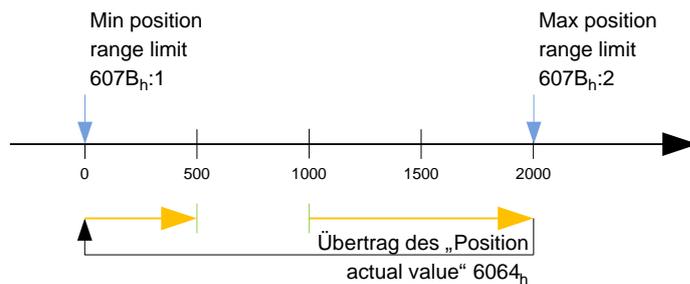


Positive Drehrichtung

Wird das Objekt **2070_h** auf "1" gesetzt, werden Überträge nicht mitgezählt und der Motor dreht immer in die positive Richtung - auch wenn das möglicherweise die weitere Fahrstrecke bedeutet.

Beispiel:

Ein absoluter Fahrbefehl auf 500 (oder entsprechend relativ um 500 Einheiten) von der Position 1000 aus führt dazu, dass der Motor den "Umweg" über den Übertrag nehmen wird, da dadurch die positive Drehrichtung realisiert wird.



Negative Drehrichtung

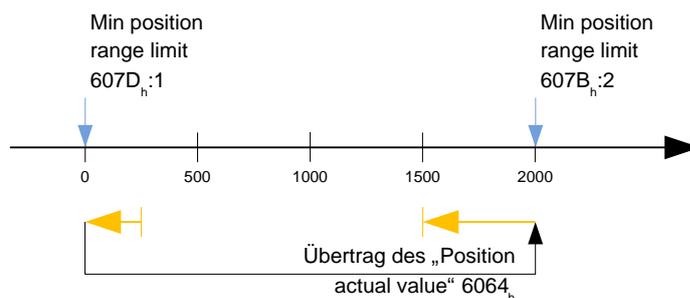
Dieser Modus entspricht dem Modus "Positive Drehrichtung" (siehe "**Positive Drehrichtung**"), nur dass die Drehrichtung entsprechend immer negativ ist (Objekt **2070_h** auf "2" gesetzt).

Kürzeste Fahrstrecke

Wird das Objekt **2070_h** auf "3" gesetzt, wird der Motor immer die Richtung der kürzesten Fahrstrecke einschlagen. Anfallende Überträge (siehe "**Überlauf mitzählen**") werden dabei nicht gerechnet.

Beispiel:

Ein absoluter Fahrbefehl auf 1500 von der Position 250 aus (oder entsprechend relativ um 1250 Einheiten) führt dazu, dass der Motor gegen den Uhrzeigersinn fahren wird. Das entspricht einer (kürzeren) Weglänge von 750 Einheiten, wohingegen die Strecke in Richtung mit dem Uhrzeigersinn 1250 Einheiten lang wäre.



Längste Fahrstrecke

Dieser Modus entspricht dem Modus "Kürzeste Fahrstrecke" (siehe "**Kürzeste Fahrstrecke**"), nur dass immer die Richtung der längeren Wegstrecke gewählt wird (Objekt **2070_h** auf "4" gesetzt).

7.2 Velocity

7.2.1 Besonderheit PD4C USB

Hinweis

Da diese Steuerung keinen Feldbus besitzt, ist der nachfolgende Betriebsmodus nur mit einem NanoJ Programm nutzbar.

Weitere Hinweise zur Programmierung und Benutzung eines NanoJ Programms findet sich im Kapitel **Programmierung mit NanoJ**.

7.2.2 Beschreibung

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

7.2.3 Aktivierung

Zum Aktivieren des Modus muss im Objekt **6060_h** (Modes Of Operation) der Wert "2" gesetzt werden (siehe "**DS402 Power State machine**").

7.2.4 Controlword

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

- Bit 2 dient zum Auslösen eines Nothalts. Wird es auf "0" gesetzt, führt der Motor eine Schnellbremsung mit der in Objekt **604A_h** eingestellten Schnell-Halt Rampe durch. Danach wechselt die Steuerung in den Zustand "Switch on disabled" (siehe **6040_h**).
- Bit 8 (Halt): Bei einem Übergang von "1" auf "0" beschleunigt der Motor mit der eingestellten Beschleunigungs-Rampe bis zur Zielgeschwindigkeit. Bei einem Übergang von "0" auf "1" bremst der Motor entsprechend der Bremsrampe ab und bleibt stehen.

7.2.5 Statusword

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

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

7.2.6 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **604C_h** (Dimension Factor):

Hier wird die Einheit der Geschwindigkeitsangaben für die nachfolgenden Objekte festgelegt.

Werden die Subindices 1 und 2 auf den Wert "1" eingestellt, erfolgt die Geschwindigkeitsangabe in Umdrehungen pro Minute.

Sonst enthält der Subindex 1 den Multiplikator und der Subindex 2 den Divisor, mit dem Geschwindigkeitsangaben verrechnet werden. Das Ergebnis wird als Umdrehungen pro Sekunde interpretiert, wobei über Objekt **2060_h** ausgewählt wird, ob es sich um elektrische (**2060_h = 0**) oder mechanische (**2060_h = 1**) Umdrehungen pro Sekunde handelt.

Hier wird die Zielgeschwindigkeit in Benutzereinheiten eingestellt.

- **6042_h**: Target Velocity
- **6048_h**: Velocity Acceleration

Dieses Objekt definiert die Startbeschleunigung. Der Subindex 1 enthält dabei die Geschwindigkeitsänderung, der Subindex 2 die zugehörige Zeit in Sekunden. Beides zusammen wird als Beschleunigung verrechnet:

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

- **6049_h** (Velocity Deceleration):

Dieses Objekt definiert die Bremsbeschleunigung. Die Subindizes sind dabei so aufgebaut, wie im Objekt **6048_h** beschrieben, die Geschwindigkeitsdifferenz ist mit positiven Vorzeichen anzugeben.

- **6085_h** (Quick Stop Deceleration):

Dieses Objekt definiert die Schnell-Halt Bremsbeschleunigung. Die Subindizes sind dabei so aufgebaut, wie im Objekt **6048_h** beschrieben, die Geschwindigkeitsdifferenz ist mit positiven Vorzeichen anzugeben.

- **6046_h** (Velocity Min Max Amount):

In diesem Objekt werden die Limitierungen der Zielgeschwindigkeiten angegeben.

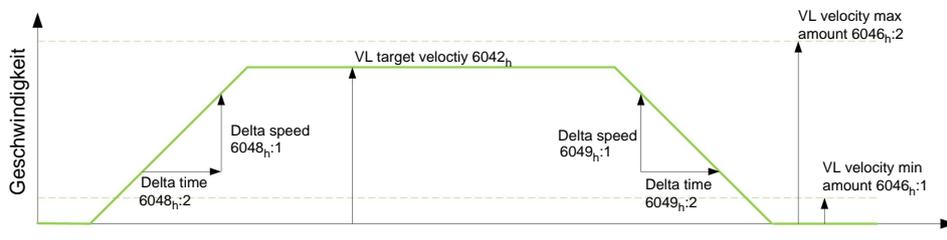
In **6046_h:1_h** wird die minimale Geschwindigkeit eingestellt. Unterschreitet die Zielgeschwindigkeit (**6042_h**) die Minimalgeschwindigkeit, wird der Wert auf die Minimalgeschwindigkeit **6046_h:1_h** begrenzt.

In **6046_h:2_h** wird die maximale Geschwindigkeit eingestellt. Überschreitet die Zielgeschwindigkeit (**6042_h**) die Maximalgeschwindigkeit, wird der Wert auf die Maximalgeschwindigkeit **6046_h:2_h** begrenzt.

- **604A_h** (Velocity Quick Stop):

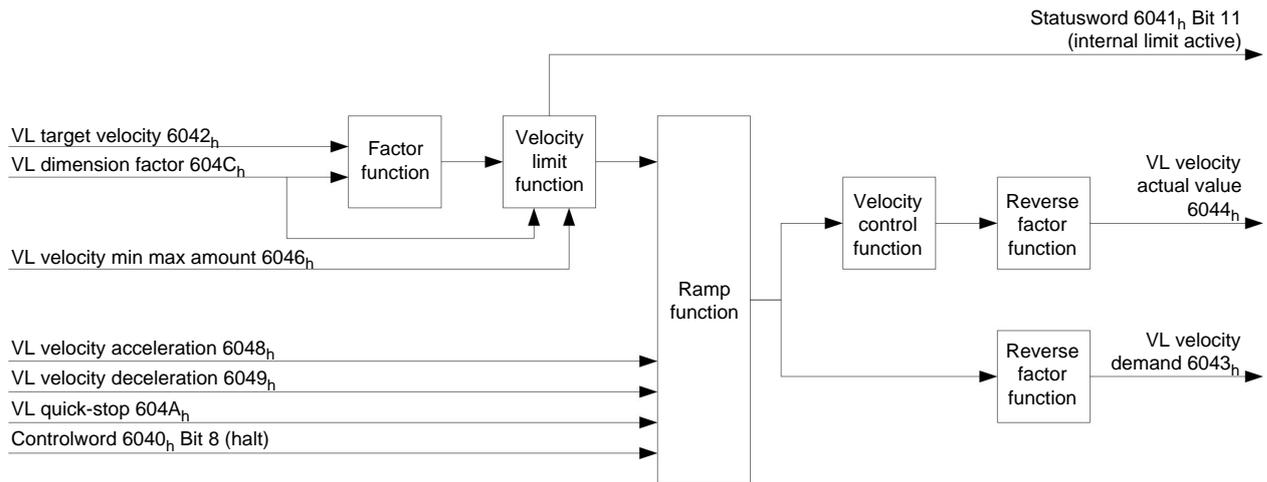
Mit diesem Objekt kann die Schnellstop-Rampe eingestellt werden. Die Subindizes 1 und 2 sind dabei identisch wie bei Objekt **6048_h** beschrieben.

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_h** gesetzt (internal limit active).



7.3 Profile Velocity

7.3.1 Besonderheit PD4C USB

Hinweis

Da diese Steuerung keinen Feldbus besitzt, ist der nachfolgende Betriebsmodus nur mit einem NanoJ Programm nutzbar.

Weitere Hinweise zur Programmierung und Benutzung eines NanoJ Programms findet sich im Kapitel **Programmierung mit NanoJ**.

7.3.2 Beschreibung

Dieser Modus betreibt den Motor im Velocity Mode mit erweiterten Rampen. Im Gegensatz zum Velocity Mode (**siehe "Velocity"**) kann bei diesem Modus über einen externen Encoder die momentane Geschwindigkeit überwacht werden.

7.3.3 Aktivierung

Zum Aktivieren des Modus muss im Objekt **6060_h** (Modes Of Operation) der Wert "3" gesetzt werden (**siehe "DS402 Power State machine"**).

7.3.4 Controlword

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

- Bit 2 dient zum Auslösen eines Nothalts. Wird es auf "0" gesetzt, führt der Motor eine Schnellbremsung mit der in Objekt **6085_h** eingestellten Rampe durch. Danach wechselt die Steuerung in den Zustand "Switch on disabled" (**6040_h**).
- Bit 8 (Halt): Bei einem Übergang von "1" auf "0" beschleunigt der Motor mit der eingestellten Startrampe bis zur Zielgeschwindigkeit. Bei einem Übergang von "0" auf "1" bremst der Motor ab und bleibt stehen.

7.3.5 Statusword

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

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

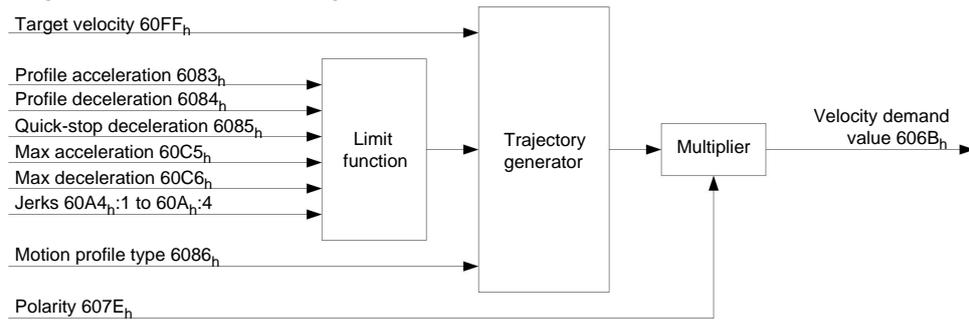
6041 _h Bit 10	6040 _h Bit 8	Beschreibung
0	0	Zielgeschwindigkeit nicht erreicht
0	1	Achse bremst
1	0	Zielgeschwindigkeit innerhalb Zielfenster (definiert in 606D_h und 606E_h)
1	1	Geschwindigkeit der Achse ist 0

7.3.6 Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **606B_h** (Velocity Demand Value):
Dieses Objekt enthält die Ausgabe des Rampengenerators, die gleichzeitig der Vorgabewert für den Geschwindigkeitsregler ist.
- **606C_h** (Velocity Actual Value):
Gibt die aktuelle Istgeschwindigkeit an.
- **606D_h** (Velocity Window):
Dieser Wert gibt an, wie stark die tatsächliche Geschwindigkeit von der Sollgeschwindigkeit abweichen darf, damit das Bit 10 (Zielgeschwindigkeit erreicht; Target Reached") im Objekt **6041_h** (Statusword) auf "1" gesetzt ist.
- **606E_h** (Velocity Window Time):
Dieses Objekt gibt an, wie lange die reale Geschwindigkeit und die Sollgeschwindigkeit nahe beieinander liegen müssen (siehe **606D_h** "Velocity Window"), damit Bit 10 "Zielgeschwindigkeit erreicht" im Objekt **6041_h** (Statusword) auf "1" gesetzt wird.
- **607E_h** (Polarity):
Wird hier Bit 6 auf "1" gestellt, wird das Vorzeichen der Zielgeschwindigkeit umgekehrt.
- **6083_h** (Profile acceleration):
Setzt den Wert für die Beschleunigungsrampe im Velocity Mode.
- **6084_h** (Profile Deceleration):
Setzt den Wert für die Bremsrampe im Velocity-Mode.
- **6085_h** (Quick Stop Deceleration):
Setzt den Wert für die Bremsrampe für die Schnellbremsung im Velocity Mode.
- **6086_h** (Motion Profile Type):
Hier kann der Rampentyp ausgewählt werden (0 = Trapez-Rampe, 3 = ruck-begrenzte Rampe).
- **604A_h** (Velocity Quick Stop), Subindex 01_h bis 04_h:
Hier werden die vier Ruck-Werte angegeben, falls eine ruck-begrenzte Rampe eingestellt ist.
- **60FF_h** (Target Velocity):
Gibt die zu erreichende Zielgeschwindigkeit an.
- **2031_h** (Peak Current):
Maximalstrom in mA

Objekte im Profile Velocity Mode

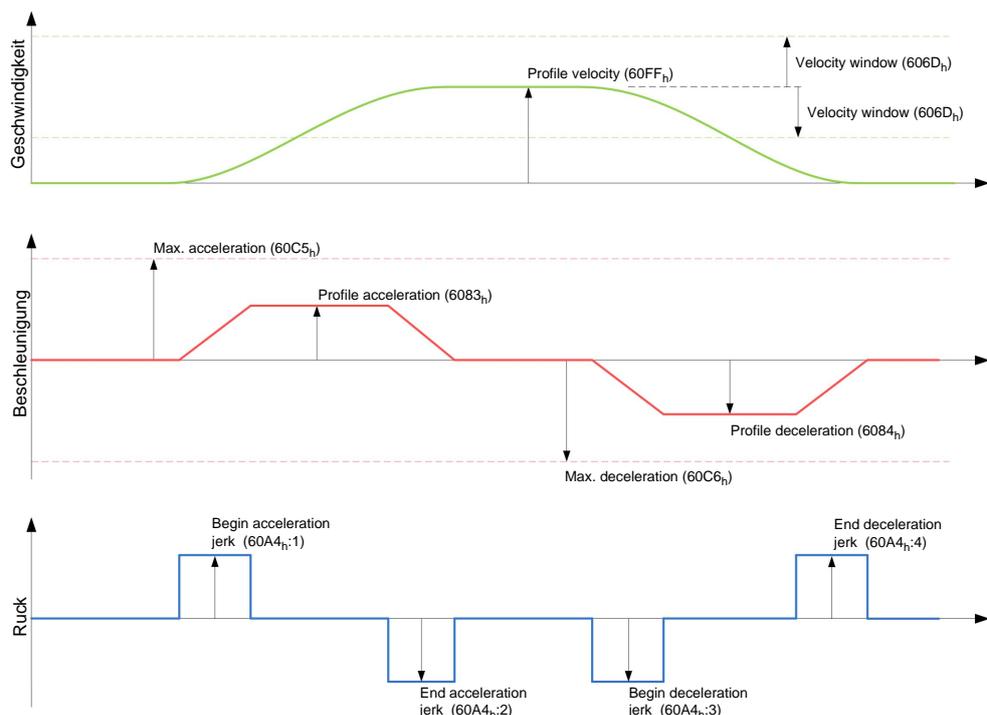


Aktivierung des Modus

Nachdem der Modus im Objekt **6060_h** (Modes Of Operation) ausgewählt wurde und die "Power State machine" (siehe "**DS402 Power State machine**") auf "Operation enabled" geschaltet wurde, wird der Motor auf die Zielgeschwindigkeit im Objekt **60FF_h** beschleunigt (siehe nachfolgende Bilder). Dabei werden die Geschwindigkeits-, Beschleunigungs- und bei ruck-begrenzten Rampen auch die Ruckgrenzwerte berücksichtigt.

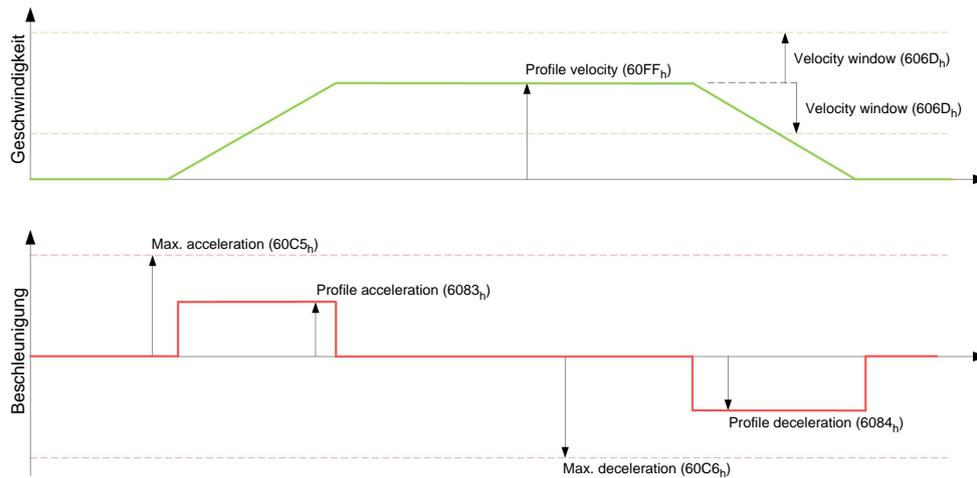
Limitierungen im ruck-limitierten Fall

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



Limitierungen im Trapezfall

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



7.4 Profile Torque

7.4.1 Besonderheit PD4C USB

Hinweis

Da diese Steuerung keinen Feldbus besitzt, ist der nachfolgende Betriebsmodus nur mit einem NanoJ Programm nutzbar.

Weitere Hinweise zur Programmierung und Benutzung eines NanoJ Programms findet sich im Kapitel **Programmierung mit NanoJ**.

7.4.2 Beschreibung

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

7.4.3 Aktivierung

Zum Aktivieren des Modus muss im Objekt **6060_h** (Modes Of Operation) der Wert "4" gesetzt werden (siehe "DS402 Power State machine").

7.4.4 Controlword

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

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

7.4.5 Statusword

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

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

6040_h	6041_h	Beschreibung
Bit 8	Bit 10	
0	0	Vorgegebenes Drehmoment nicht erreicht
0	1	Vorgegebenes Drehmoment erreicht
1	0	Achse beschleunigt

6040 _h	6041 _h	Beschreibung
Bit 8	Bit 10	
1	1	Geschwindigkeit der Achse ist 0

7.4.6 Objekteinträge

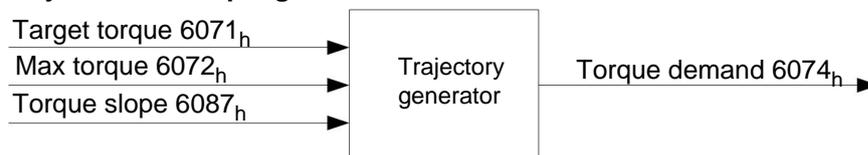
Alle Werte der folgenden Einträge im Objektverzeichnis sind als Tausendstel des maximalen Drehmoments anzugeben, welches dem Maximalstrom (2031_h) entspricht. Dazu zählen die Objekte:

- **6071_h** (Target Torque):
Zielvorgabe des Drehmomentes
- **6072_h** (Max Torque):
Maximales Drehmoment während der gesamten Rampe (Beschleunigen, Drehmoment halten, Abbremsen)
- **6074_h** (Torque Demand):
Momentaner Ausgabewert des Rampengenerators (Drehmoment) für den Regler
- **6087_h** (Torque Slope):
Max. Änderung des Drehmoments pro Sekunde
- **3202_h** Bit 5 (Motor Drive Submode Select):

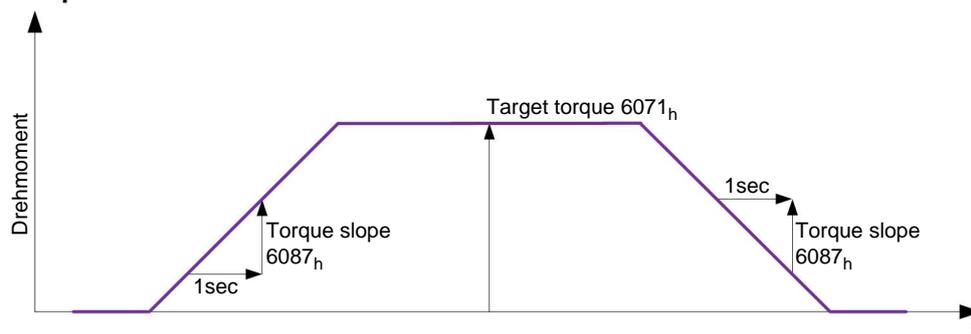
Ist dieses Bit auf "0" gesetzt, wird der Antriebsregler im Drehmoment-begrenzten Velocity Mode betrieben, d.h. die maximale Geschwindigkeit kann in Objekt 2032_h begrenzt werden und der Regler kann im Feldschwächebetrieb arbeiten.

Wird dieses Bit auf "1" gesetzt, arbeitet der Regler im Torque Mode, die maximale Geschwindigkeit kann hier nicht begrenzt werden und der Feldschwächebetrieb ist nicht möglich.

Objekte des Rampengenerators



Torque-Verlauf



7.5 Homing

7.5.1 Besonderheit PD4C USB

Hinweis

Da diese Steuerung keinen Feldbus besitzt, ist der nachfolgende Betriebsmodus nur mit einem NanoJ Programm nutzbar.

Hinweis

Weitere Hinweise zur Programmierung und Benutzung eines NanoJ Programms findet sich im Kapitel **Programmierung mit NanoJ**.

7.5.2 Übersicht

Beschreibung

Ziel der Referenzfahrt (Homing Method) ist es, die Steuerung auf den Encoder-Index des Motors oder Positionsschalter in einer Anlage zu synchronisieren.

Aktivierung

Zum Aktivieren des Modus muss im Objekt **6060_h** (Modes Of Operation) der Wert "6" gesetzt werden (**siehe "DS402 Power State machine"**).

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

Controlword

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

- Bit 2: dient zum Auslösen eines Nothalts. Wird es auf "0" gesetzt, führt der Motor eine Schnellbremsung mit der in Objekt **6085_h** eingestellten Rampe durch. Danach geht der Motor in den Zustand "Switch on disabled" (**siehe "DS402 Power State machine"**).
- 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_h** (Statusword) haben eine gesonderte Funktion:

Bit 13	Bit 12	Bit 10	Beschreibung
0	0	0	Referenzfahrt wird ausgeführt
0	0	1	Referenzfahrt ist unterbrochen oder nicht gestartet
0	1	0	Referenzfahrt gestartet, 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

Objekteinträge

Folgende Objekte sind zur Steuerung dieses Modus erforderlich:

- **6098_h** (Homing Method):
Methode, mit der referenziert werden soll (**siehe "Referenzfahrt-Methode"**)
- **6099_h:01_h** (Speed During Search For Switch):
Geschwindigkeit für die Suche nach dem Schalter
- **6099_h:02_h** (Speed During Search For Zero):
Geschwindigkeit für die Suche nach dem Index
- **609A_h** (Homing Acceleration):
Anfahr- und Bremsbeschleunigung für die Referenzfahrt
- **2056_h** (Limit Switch Tolerance Band):

Die Steuerung lässt nach dem Auffahren auf den positiven oder negativen Endschalter einen Toleranzbereich zu, den der Motor noch zusätzlich weiter fahren darf. Wird dieser Toleranzbereich überschritten, stoppt der Motor und die Steuerung wechselt in den Zustand "Fault". Falls während der Referenzfahrt Endschalter betätigt werden können, sollte der Toleranzbereich ausreichend gewählt werden, so dass der Motor beim Abbremsen den Toleranzbereich nicht verlässt. Andernfalls kann die Referenzfahrt nicht erfolgreich ausgeführt werden. Nach Abschluss der Referenzfahrt kann der Toleranzbereich, wenn dies die Anwendung erfordert, wieder auf "0" gesetzt werden.

- **203A_h:1_h** (Minimum Current For Block Detection):

Minimale Stromschwelle, durch deren Überschreiten, das Blockieren des Motors an einem Block erkannt werden soll.

- **203A_h:2_h** (Period Of Blocking):

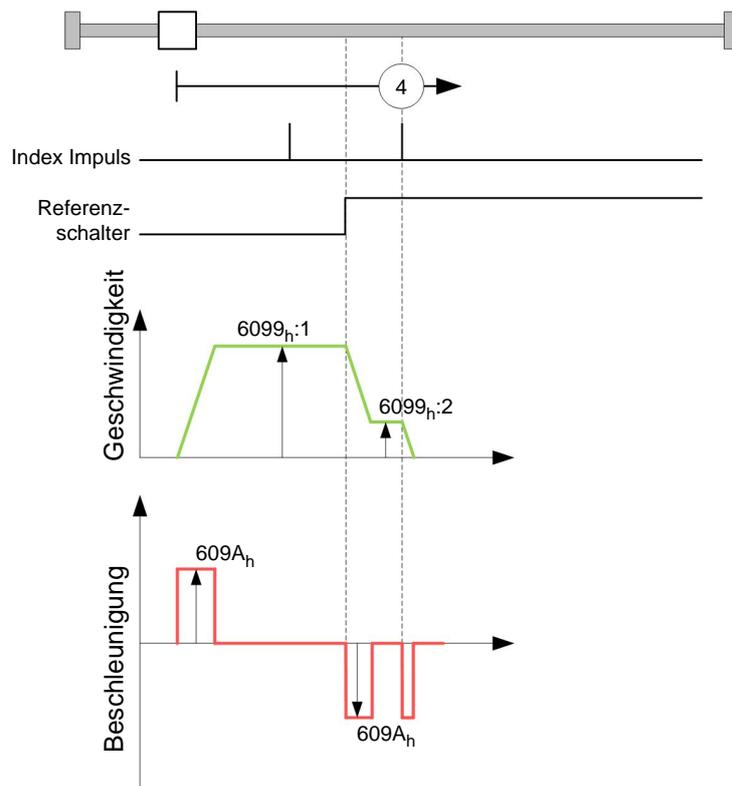
Gibt die Zeit in ms an, die der Motor nach der Blockdetektion trotzdem noch gegen den Block fahren soll.

- **203A_h:3_h** (Block Detection Time)

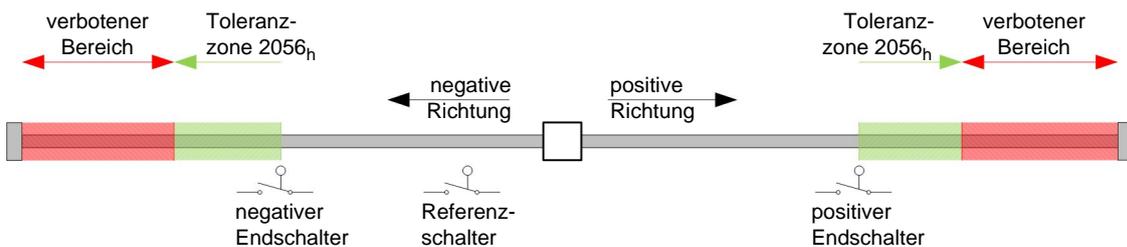
Gibt die Zeit in ms an, die der Strom mindestens oberhalb der minimalen Stromschwelle sein muss, um einen Block zu erkennen

Geschwindigkeiten der Referenzfahrt

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



Toleranzbänder der Endschalter



7.5.3 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 Index-Impuls referenziert wird oder in welche Richtung die Referenzfahrt startet. Methoden, die den Index-Impuls des Encoders benutzen, liegen im Zahlenbereich 1 bis 14, 33 und 34. Methoden, die auf einen Endschalter referenzieren, 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 Index-Impulse kommen vom Encoder, der mit der Welle des Motors verbunden und an die Steuerung angeschlossen ist.

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 einen mechanischen Block ersetzt werden müssen.

Homing auf Block

Homing auf Block funktioniert derzeit nur im Closed Loop-Betrieb einwandfrei. Auf die Feinheiten, die unter anderem bei Homing auf Block im Closed Loop-Betrieb geachtet werden müssen, wird im Kapitel über den Regler eingegangen.

Für bestimmte Anwendungen ist es zweckmäßig, nach der Detektion des Blocks, eine gewisse Zeit weiterhin gegen den Block zu fahren. Diese Zeit kann in Objekt **203A_h:2_h** in ms eingestellt werden.

Um eine sehr genaue Erkennung des Blocks zu gewährleisten, sollte man mit einer sehr niedrigen Geschwindigkeit (**6099_h:1_h**), hoher Stromgrenze (**203A_h:1_h**) und hoher Homing-Beschleunigung (**609A_h**) gegen den Block fahren. Zusätzlich kann noch über die Block-Detektionszeit (**203A_h:3_h**) die Erkennung verfeinert werden.

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 enthalten einen Index-Impuls
- Methoden 15 und 16 sind nicht vorhanden
- Methoden 17 bis 30 haben keinen Index-Impuls
- Methoden 31 und 32 sind nicht vorhanden
- Methoden 33 und 34 referenzieren nur auf den nächsten Index-Impuls
- Methode 35 referenziert auf die aktuelle Position

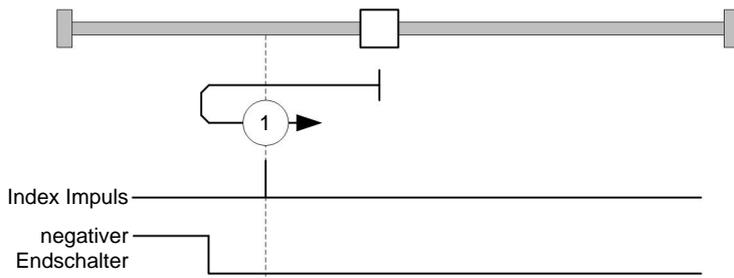
Folgende Methoden können für Homing auf Block benutzt werden:

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

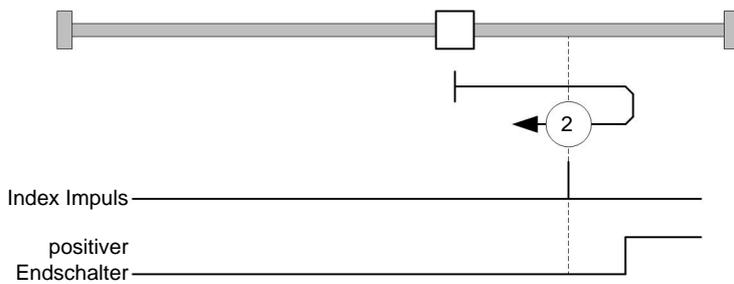
Methoden 1 und 2

Referenzieren auf Endschalter und Index-Impuls.

Methode 1 referenziert auf negativen Endschalter und Index-Impuls:



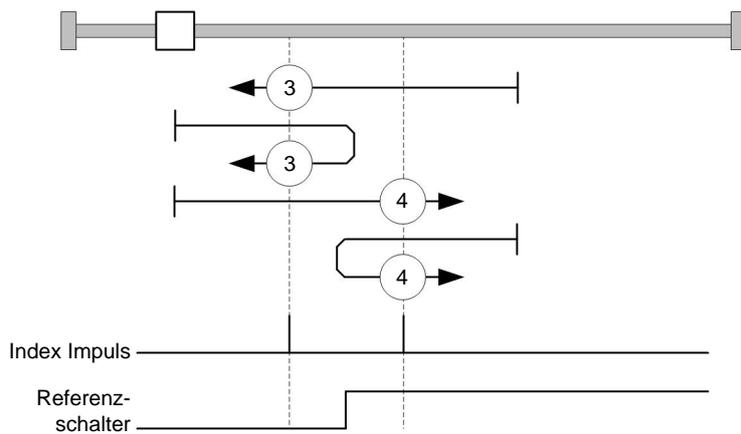
Methode 2 referenziert auf positiven Endschalter und Index-Impuls:



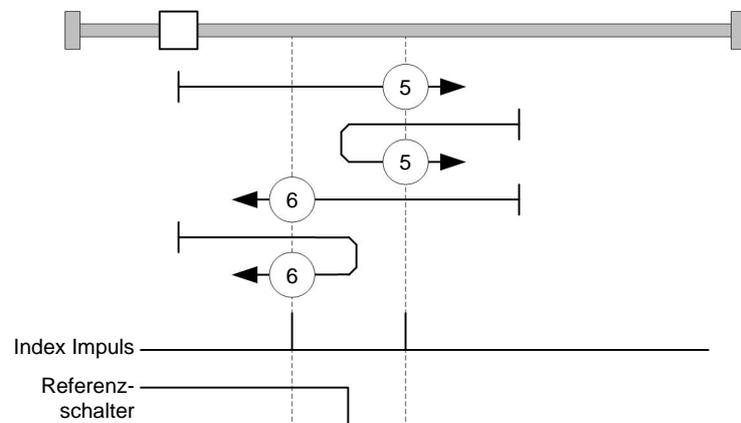
Methoden 3 bis 6

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:

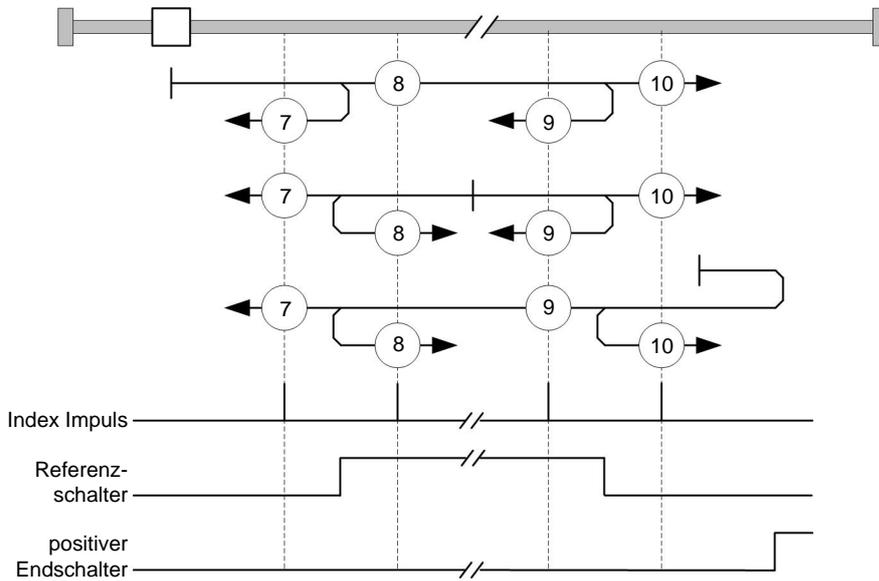


Methoden 7 bis 14

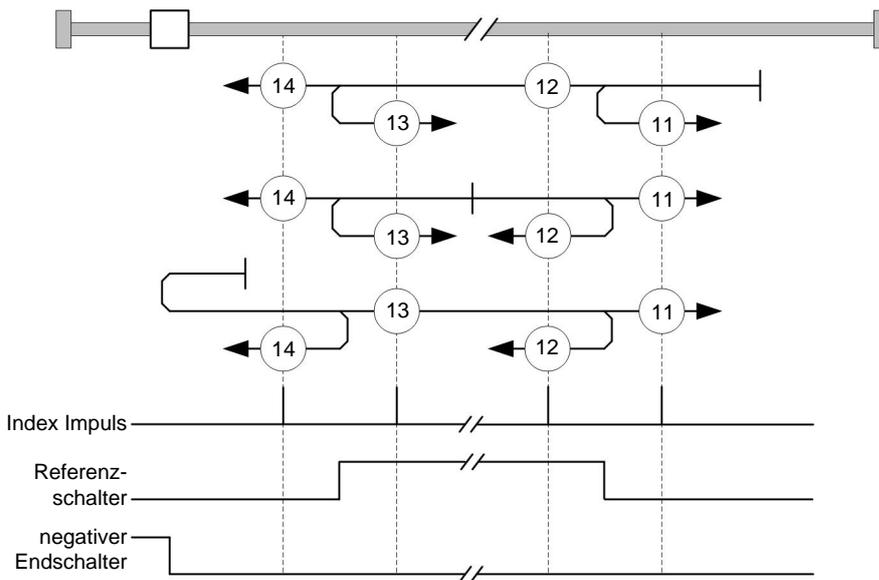
Referenzieren auf Referenzschalter und Index-Impuls (mit Endschaltern).

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:



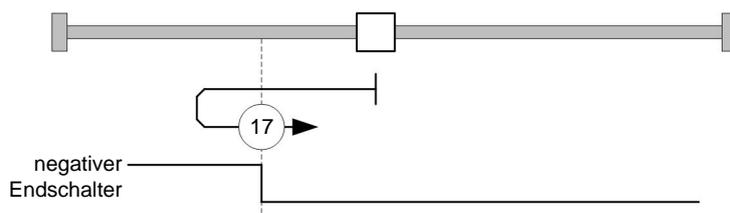
Die Methoden 11 bis 14 berücksichtigen den negativen Endschalter:



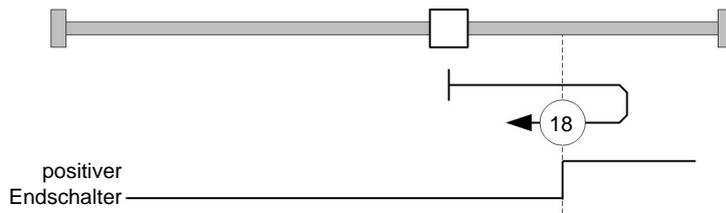
Methoden 17 und 18

Referenzieren auf den Endschalter ohne den Index-Impuls.

Methode 17 referenziert auf den negativen Endschalter:



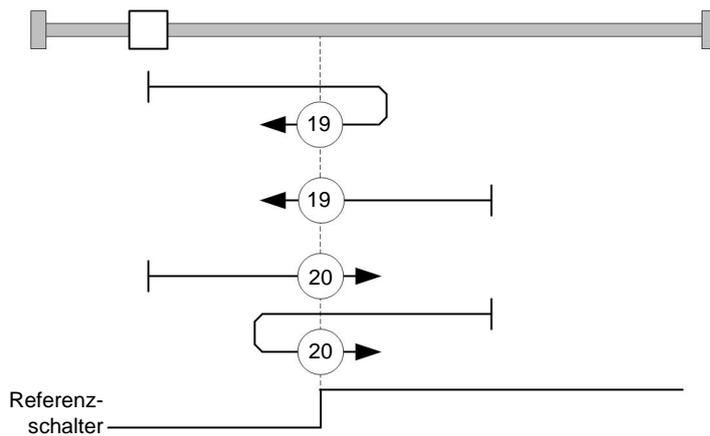
Methode 18 referenziert auf den positiven Endschalter:



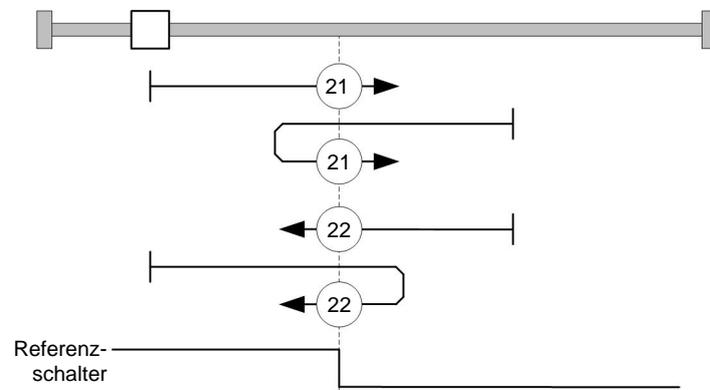
Methoden 19 bis 22

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:

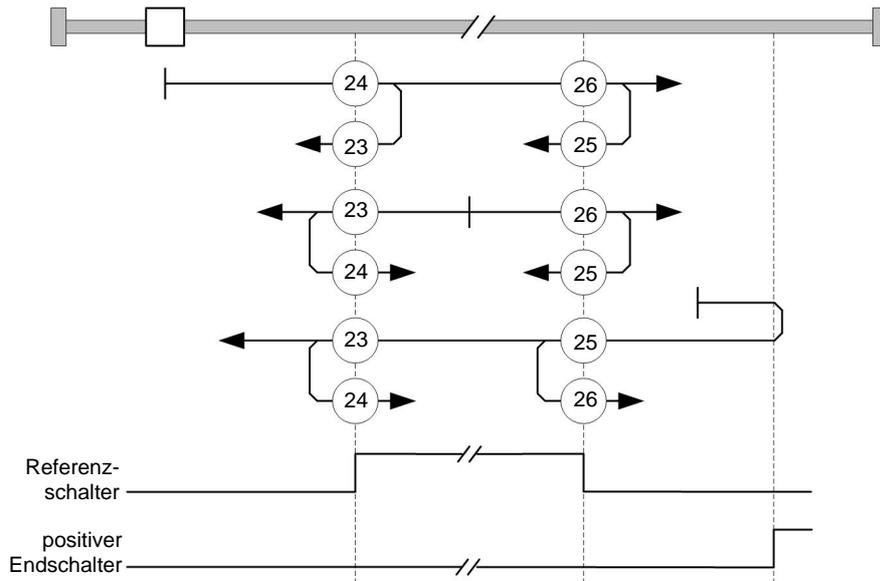


Methoden 23 bis 30

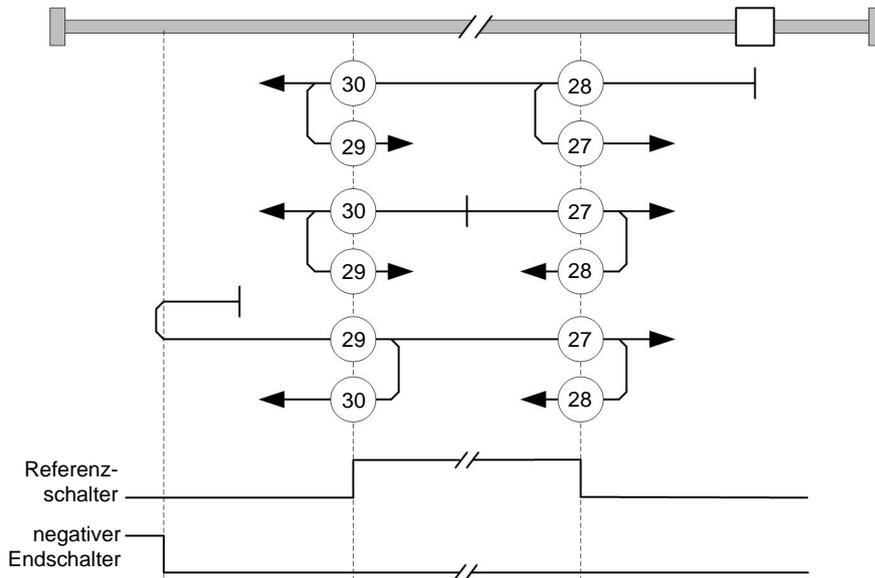
Referenzieren auf Referenzschalter ohne den Index-Impuls (mit Endschaltern).

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:



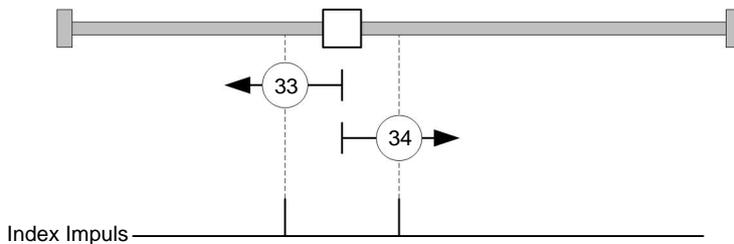
Die Methoden 27 bis 30 berücksichtigen den negativen Referenzschalter:



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.

8 Generelle Konzepte

8.1 DS402 Power State machine

8.1.1 Zustandsmaschine

CANopen DS402

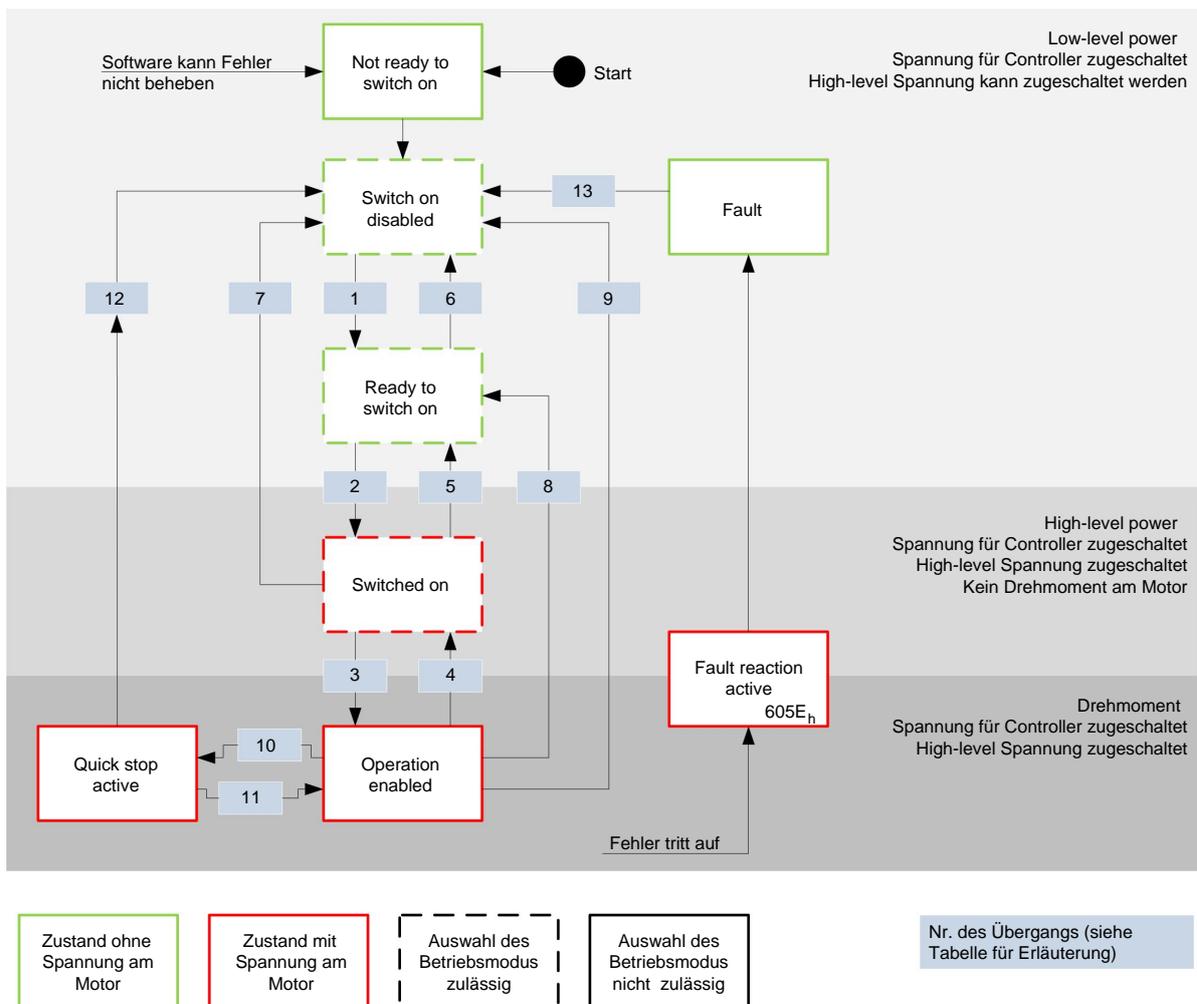
Um die Steuerung betriebsbereit zu schalten, ist es notwendig, eine Zustandsmaschine zu durchlaufen. Diese ist im CANopen-Standard DS402 definiert. Zustandsänderungen werden im Objekt **6040_h** (Controlword) angefordert. Der tatsächliche Zustand der Zustandsmaschine lässt sich aus dem Objekt **6041_h** (Statusword) entnehmen.

Controlword

Zustandsänderungen werden über Objekt **6040_h** (Controlword) angefordert. In der nachfolgenden **Tabelle** sind die Bitkombinationen aufgelistet, die zu den entsprechenden Zustandsübergängen führen.

Zustandsübergänge

Das Diagramm zeigt die möglichen Zustandsübergänge.



In der nachfolgenden **Tabelle** sind die Bitkombinationen für das Controlword aufgelistet, die zu den entsprechenden Zustandsübergängen führen. Ein X entspricht dabei einem nicht weiter zu

berücksichtigenden Bitzustand. Einzige Ausnahme ist das Rücksetzen des Fehlers (Fault reset): Der Übergang wird nur durch steigende Flanke des Bits angefordert.

Kommando	Bit im Objekt 6040 _h					Übergang
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	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, 11
Fault reset		X	X	X	X	13

Statusword

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

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

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

Betriebsmodus

Der eingestellte Betriebsmodus (6060_h) wird erst im Zustand "Operation enabled" aktiv. Die Einstellung oder Änderung des Betriebsmodus ist nur in folgenden Zuständen möglich (siehe gestrichelt umrahmte Zustände im Diagramm):

- Switch on disabled
- Ready to switch on
- Switched on

Im laufenden Betrieb ("Operation enabled") ist es nicht möglich, den Betriebsmodus zu wechseln. Der Zustand "Fault" wird verlassen, wenn das Bit 7 in Objekt 6040_h (Controlword) von "0" auf "1" gesetzt wird (steigende Flanke).

Hinweis: Tritt ein nicht behebbarer Fehler auf, wechselt die Steuerung in den Zustand "Not ready to switch on" und verbleibt dort. Zu diesen Fehlern zählen:

- Encoderfehler (z. B. durch fehlende Schirmung, Kabelbruch)

Außerdem kann dieser Zustand durch einen Busfehler mit dem Feldbustyp EtherCAT erreicht werden. In diesem Fall wird - nachdem der Busfehler behoben ist - automatisch wieder in den Zustand "Switch on disabled" gewechselt.

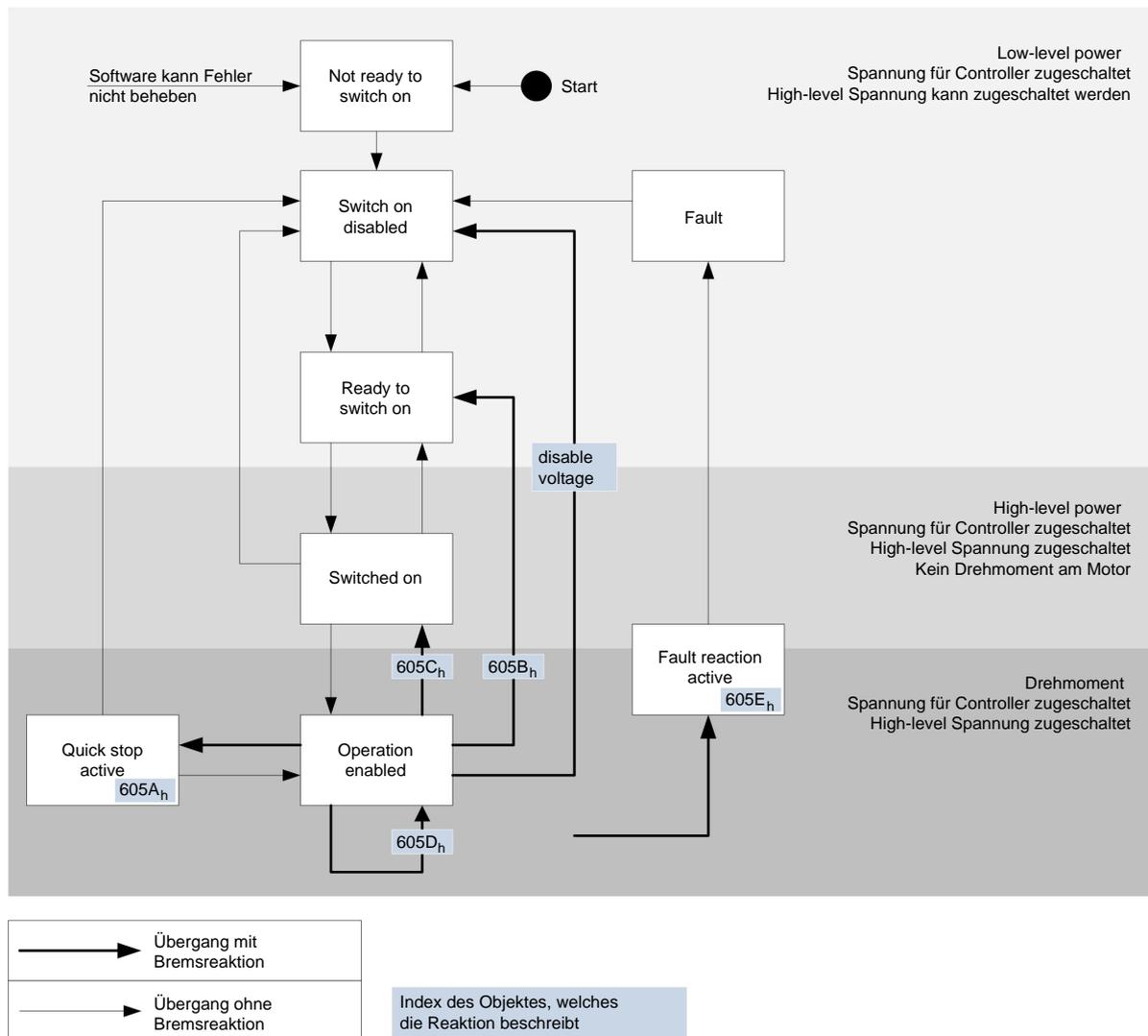
8.1.2 Verhalten beim Verlassen des Zustands "Operation enabled"

Bremsreaktionen

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

Dazu zählen die nachfolgend beschriebenen Übergänge.

Die nachfolgende Grafik zeigt eine Übersicht über die Bremsreaktionen.



Quick stop active

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

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

Wert in Objekt 605A _h	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt mit Kurzschlussbremsung

Wert in Objekt 605A _h	Beschreibung
1	Abbremsen mit "slow down ramp" (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in "Switch on disabled"
2	Abbremsen mit "quick stop ramp" und anschließendem Zustandswechsel in "Switch on disabled"
3 bis 32767	Reserviert

Ready to switch on

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

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

Wert in Objekt 605B _h	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt mit Kurzschlussbremsung
1	Abbremsen mit "slow down ramp" (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in "Switch on disabled"
2 bis 32767	Reserviert

Switched on

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

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

Wert in Objekt 605C _h	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt mit Kurzschlussbremsung
1	Abbremsen mit "slow down ramp" (Bremsbeschleunigung je nach Betriebsmodus) und anschließendem Zustandswechsel in "Switch on disabled"
2 bis 32767	Reserviert

Halt

Halt (halt):

Beim Setzen des Bit 8 in Objekt **6040_h** (Controlword) wird im Velocity Mode und im Profile Velocity Mode die in **605D_h** hinterlegte Reaktion ausgeführt (siehe nachfolgende Tabelle).

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

Fault

Fehlerfall (fault):

Sollte ein Fehler auftreten, wird der Motor abgebremst, wie es in Objekt **605E_h** hinterlegt ist.

Wert in Objekt 605E_h	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt mit Kurzschlussbremsung
1	Abbremsen mit "slow down ramp" (Bremsbeschleunigung je nach Betriebsart)
2	Abbremsen mit "quick stop ramp" (Bremsbeschleunigung je nach Betriebsart)
3 bis 32767	Reserviert

8.2 Benutzerdefinierte Einheiten

8.2.1 Übersicht

Einstellungen

Die Steuerung unterstützt die Möglichkeit, benutzerdefinierte Einheiten einzustellen. Damit lassen sich die entsprechenden Parameter z. B. direkt in Grad, mm, usw. setzen und auslesen.

Polpaarzahl- Kompensation

Unterschiede in den Polpaarzahlen von Motoren können kompensiert werden. Dazu ist der Wert im Objekt **2060_h** auf "1" zu setzen. Anschließend geht die Polpaarzahl automatisch in die nachfolgenden Berechnungen ein, so dass unterschiedliche Motoren an der Steuerung betrieben werden können, ohne dass eine Neukonfiguration erforderlich ist.

8.2.2 Berechnungsformeln für Benutzereinheiten

Getriebeübersetzung

Die Getriebeübersetzung berechnet sich aus Motorumdrehung (**6091_h:1_h** (Motor Revolutions)) pro Achsenumdrehung (**6091_h:2_h** (Shaft Revolutions)) wie folgt:

$$\text{Getriebeübersetzung} = \frac{\text{Motorumdrehung (6091}_{h:1})}{\text{Achsenumdrehung (6091}_{h:2})}$$

Sollten Objekt **6091_h:1_h** oder Objekt **6091_h:2_h** auf "0" gesetzt werden, setzt die Firmware den Wert auf "1".

Vorschubkonstante

Die Vorschubkonstante wird aus dem Vorschub (**6092_h:1_h** (Feed Constant) pro Umdrehung der Antriebsachse (**6092_h:2_h** (Shaft Revolutions)) wie folgt berechnet:

$$\text{Vorschubkonstante} = \frac{\text{Vorschub (6092}_{h:1})}{\text{Umdrehung der Antriebsachse (6092}_{h:2})}$$

Dies ist zur Angabe der Spindelsteigung bei einer Linearachse nützlich.

Sollte Objekt **6092_h:1_h** oder Objekt **6092_h:2_h** auf "0" gesetzt werden, setzt die Firmware den Wert auf "1".

Position

Die aktuelle Position in Benutzereinheiten (**6064_h**) berechnet sich wie folgt:

$$\text{tatsächliche Position} = \frac{\text{interne Position} \times \text{Vorschubkonstante}}{\text{Encoderauflösung} \times \text{Getriebeübersetzung}}$$

Geschwindigkeit

Die Geschwindigkeitsvorgaben der nachfolgenden Objekte können ebenfalls in Benutzereinheiten angegeben werden:

Objekt	Modus	Bedeutung
606B_h	Profile Velocity Mode	Ausgabewert des Rampengenerators
60FF_h	Profile Velocity Mode	Geschwindigkeitsvorgabe
6099_h	Homing Mode	Geschwindigkeit zum Suchen des Index / Schalters
6081_h	Profile Position Mode	Zielgeschwindigkeit
6082_h	Profile Position Mode	Endgeschwindigkeit

Dabei wird die interne Geschwindigkeit in mechanischen Umdrehungen pro Sekunde mit einem Faktor für Zähler (**2061_h**) und Nenner (**2062_h**) multipliziert. Die Geschwindigkeit in Benutzereinheiten berechnet sich aus

$$\text{Geschwindigkeit} = \frac{\text{interne Geschwindigkeit} \times \text{Faktor Zähler (2061}_{h})}{\text{Faktor Nenner (2062}_{h})}$$

Sollte Objekt **2061_h** oder Objekt **2062_h** auf "0" gesetzt werden, setzt die Firmware den Wert auf "1".

Beschleunigung

Die Beschleunigung kann ebenfalls in Benutzereinheiten angegeben werden:

Objekt	Modus	Bedeutung
609A_h	Homing Mode	Beschleunigung
6083_h	Profile Position Mode	Beschleunigung
6084_h	Profile Position Mode	Bremsbeschleunigung
60C5_h	Profile Velocity Mode	Beschleunigung
60C6_h	Profile Position Mode	Bremsbeschleunigung
6085_h	Zustand "Quick stop active" (DS402 Power State machine)	Bremsbeschleunigung

Dabei wird die interne Beschleunigung in mechanischen Umdrehungen pro Sekunde im Quadrat mit einem Faktor für Zähler (**2063_h**) und Nenner (**2064_h**) multipliziert.

$$\text{Beschleunigung} = \frac{\text{interne Beschleunigung} \times \text{Faktor Zähler (2063}_{h})}{\text{Faktor Nenner (2064}_{h})}$$

Sollte Objekt **2063_h** oder Objekt **2064_h** auf "0" gesetzt werden, setzt die Firmware den Wert auf "1".

Ruck

Für den Ruck lassen sich die Objekte **604A_h:1_h** bis **604A_h:4_h** in Benutzereinheiten angeben. Diese Objekte betreffen nur den Profile Position Mode und den Profile Velocity Mode.

Zur Verfügung stehen die Objekte **2065_h** für den Zähler und **2066_h** für den Nenner. Die Werte des Objekts **604A_h:1_h** bis **4_h** berechnen sich aus mechanischen Umdrehungen pro Sekunde zur dritten Potenz multipliziert mit Zähler und Nenner:

$$\text{Ruck} = \frac{\text{interner Wert} \times \text{Faktor Zähler (2065}_{h})}{\text{Faktor Nenner (2066}_{h})}$$

Sollte Objekt **2065_h** oder Objekt **2066_h** auf "0" gesetzt werden, setzt die Firmware den Wert auf "1".

Positionsangaben

Alle Positionswerte im Open und im Closed Loop-Betrieb werden in der Auflösung des virtuellen Positionencoders angegeben. Diese berechnet sich aus den Encoder-Inkrementen (**608F_h:1_h** (Encoder Increments)) pro Motorumdrehungen (**608F_h:2_h** (Motor Revolutions)) multipliziert mit der Polarität der Achse im Objekt **607E_h** Bit 0. Ist das Bit 0 in Objekt **607E_h** auf den Wert "1" gesetzt, entspricht das einer Polaritätsumkehr, bzw. dem Wert "-1" in der Formel:

$$\text{Auflösung des Positionencoders} = \text{Polarität (607E}_{h} \text{ Bit 0)} \times \frac{\text{Encoder-Inkmente (608F}_{h}:1)}{\text{Motorumdrehungen (608F}_{h}:2)}$$

Sollte der Wert **608F_h:1_h** oder der Wert **608F_h:2_h** auf "0" gesetzt werden, rechnet die Steuerung intern mit einer "1" weiter. Die Werkseinstellungen sind:

- Encoder-Inkmente **608F_h:1_h** = "2000"
- Motorumdrehungen **608F_h:2_h** = "1"
- Polarität **607E_h** Bit 0 = "0" (entspricht keiner Polaritätsumkehr)

Die Auflösung des angeschlossenen Positionencoders wird in Objekt **2052_h** eingestellt.

9 Spezielle Funktionen

9.1 Digitale Ein- und Ausgänge

Die Steuerung verfügt über digitale Ein- und Ausgänge.

9.1.1 Digitale Eingänge

Zwei Gruppen

Die Eingänge teilen sich in zwei Gruppen:

- Eingänge 1 bis 3 sind ausgelegt für Spannungen von 0 V bis 24 V mit den Schaltschwellen bei 11 V (ein) und 9,5 V (aus)
- Eingänge 4 und 6 sind Weitbereichseingänge für Spannungen von 0 V bis 5 V - 24 V. Die Schaltschwellen liegen bei 3,0 V (ein) und 2,5 V (aus).

Der Zustand eines Eingangs wird im Eintrag **60FD_h** als Bit-Wert abgelegt. Jedem Eingang ist dabei eine Bitposition zugewiesen: Eingang 1 entspricht dabei Bit 16 des Objekts **60FD_h**, Eingang 2 dem Bit 17 des Objekts **60FD_h**, usw. (siehe nachfolgende Tabelle).

Die Eingänge werden jede Millisekunde ausgewertet; Reaktionen auf Änderungen an den Eingängen können in < 2 ms erfolgen.

Objekt 60FD _h	Eingang / Sonderfunktion
Bit 0	Negativer Endschalter
Bit 1	Positiver Endschalter
Bit 2	Referenzschalter
Bit 16	Eingang 1
Bit 17	Eingang 2
Bit 18	Eingang 3
Bit 19	Eingang 4

Objekteinträge

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

- **3240_h:1_h**

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 bei einer Referenzfahrt (Homing Method) folgende Bits aus:

- Bit 0: Negativer Endschalter
- Bit 1: Positiver Endschalter
- Bit 2: Referenzschalter

- **3240_h:2_h**

Dieses Bit wechselt von Schließer-Logik (ein logischer High-Pegel am Pin ergibt den Wert "1" im Objekt **60FD_h**) auf Öffner-Logik (der logische High-Pegel am Pin ergibt den Wert "0"). Das gilt für die Sonderfunktionen (außer den Takt- und Richtungseingängen) und für die normalen Eingangspins.

- **3240_h:3_h**

Dieses Bit schaltet die Softwaresimulation von Eingangswerten ein, wenn es auf "1" gesetzt ist. Dann werden nicht mehr die tatsächlichen, sondern die in Objekt **3240_h:4_h** eingestellten Werte für den jeweiligen Eingang verwendet.

- **3240_h:4_h**

Dieses Bit gibt den Wert vor, der als Eingangswert eingelesen werden soll, wenn das gleiche Bit im Objekt **3240_h:3_h** gesetzt wurde.

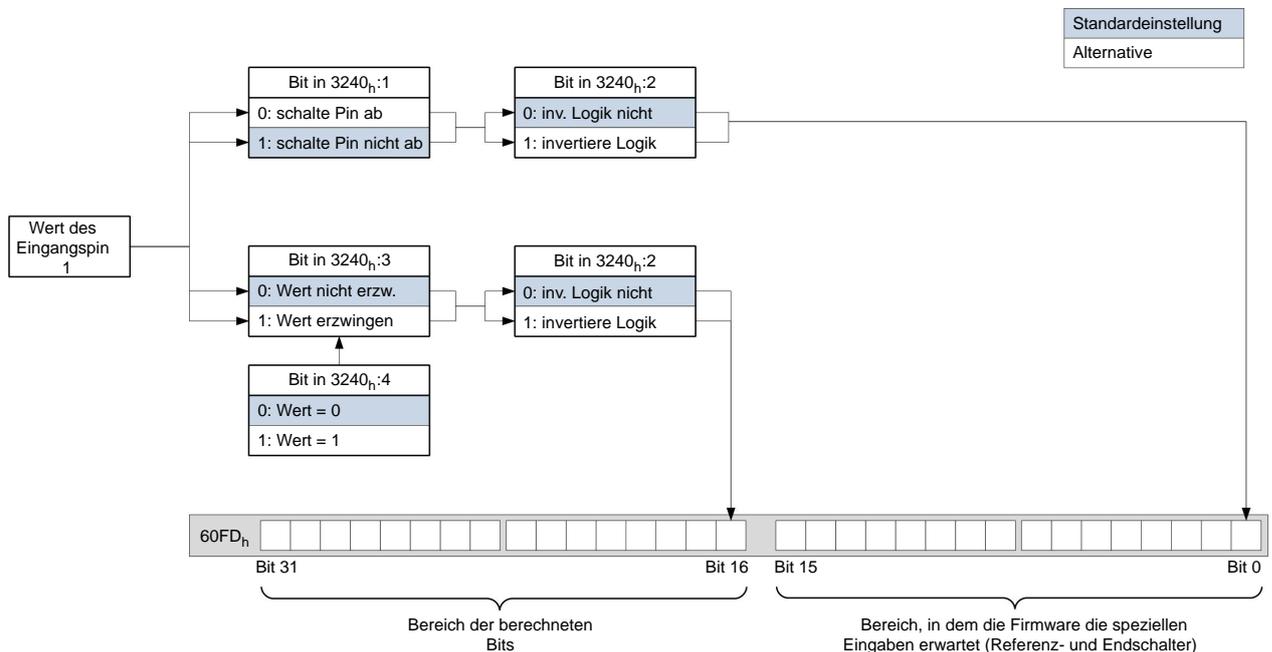
- **3240_h:5_h**

Dieses Objekt beinhaltet den unmodifizierten Eingabewert.

Verrechnung des Eingangspins

Verrechnung des Eingangspins am Beispiel von Pin 1:

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



9.1.2 Digitale Ausgänge

Ausgänge

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

Objekteinträge

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

- **3250_h:2_h**

Damit lässt sich die Logik von "Schließer" auf "Öffner" umstellen. Als "Schließer" konfiguriert, gibt der Pin einen logischen High-Pegel ab, sollte das Bit "1" sein. Bei der "Öffner"-Konfiguration wird bei einer "1" im Objekt **60FE_h** entsprechend ein logischer Low-Pegel ausgegeben.

- **3250_h:3_h**

Ist ein Bit in **3250_h** gesetzt, wird der Ausgang manuell gesteuert. Der Wert für den Ausgang steht dann in Objekt **3250_h:4_h**, dies ist auch für den Bremsenausgang möglich.

- **3250_h:4_h**

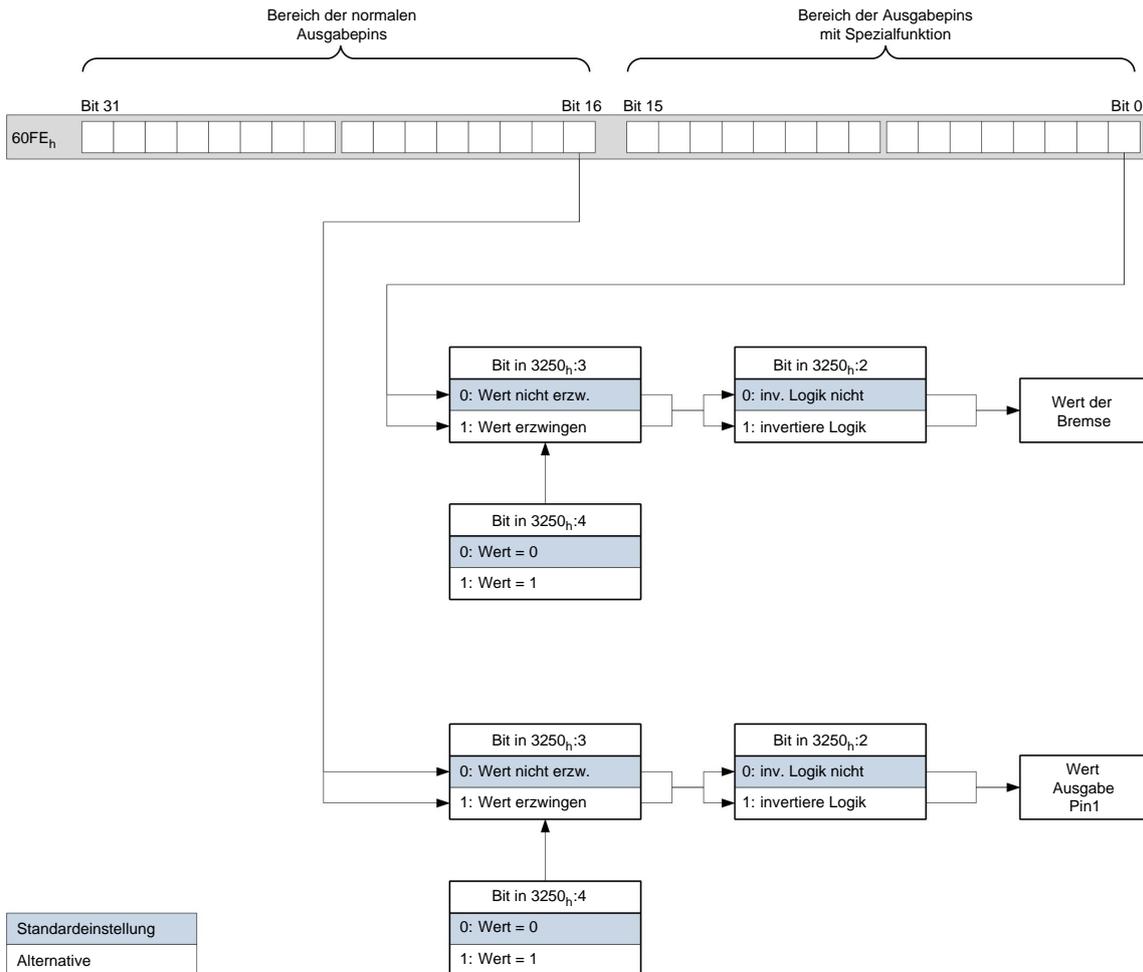
Die Bits in diesem Objekt geben den Ausgabewert vor, welcher am Ausgangspin angelegt sein soll, wenn die manuelle Steuerung des Ausgangspins über das Objekt **3250_h:3_h** aktiviert ist.

- **3250_h:5_h**

Dieses Objekt besitzt keine Funktion und ist aus Gründen der Kompatibilität enthalten.

Bits der Ausgabepins

Beispiel für die Verrechnung der Bits für die Ausgabepins:



9.2 I²t Motor-Überlastungsschutz

9.2.1 Beschreibung

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

Diese Funktion ist nur verfügbar, wenn sich die Steuerung in der Closed Loop-Betriebsart befindet (Bit 0 des Objekts **3202_h** auf "1" gesetzt) und sich der Motor **nicht** im Profile Torque Mode oder Cycle Synchrones Torque Mode befindet.

Es gibt eine Ausnahme: Sollte I²t im Open Loop-Betrieb aktiviert sein, wird der Strom auf den eingestellten Nominalstrom begrenzt, auch wenn der eingestellte Maximalstrom größer ist. Dieses Feature wurde aus Sicherheitsgründen implementiert, damit man auch aus dem Closed Loop-Betrieb

mit sehr hohem kurzzeitigem Maximalstrom in den Open Loop-Betrieb wechseln kann, ohne den Motor zu schädigen.

9.2.2 Objekteinträge

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

- **2031_h**: Peak Current - Gibt den Maximalstrom in mA an.
- **203B_h:1_h** Nominal Current - Gibt den Nennstrom in mA an.
- **203B_h:2_h** Maximum Duration Of Peak Current - Gibt die maximale Dauer des Maximalstroms in ms an.

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

- **203B_h:3_h** Threshold - Gibt die Grenze in mAs an, von der abhängt, ob auf Maximalstrom oder Nominalstrom geschaltet wird.
- **203B_h:4_h** CalcValue - Gibt den berechneten Wert an, welcher mit Threshold verglichen wird, um den Strom einzustellen.
- **203B_h:5_h** LimitedCurrent - Zeigt den gegenwärtigen Stromwert an, der von I^2t eingestellt wurde.
- **203B_h:6_h** Status:
 - Wert = "0": I^2t deaktiviert
 - Wert = "1": I^2t aktiviert

9.2.3 Aktivierung

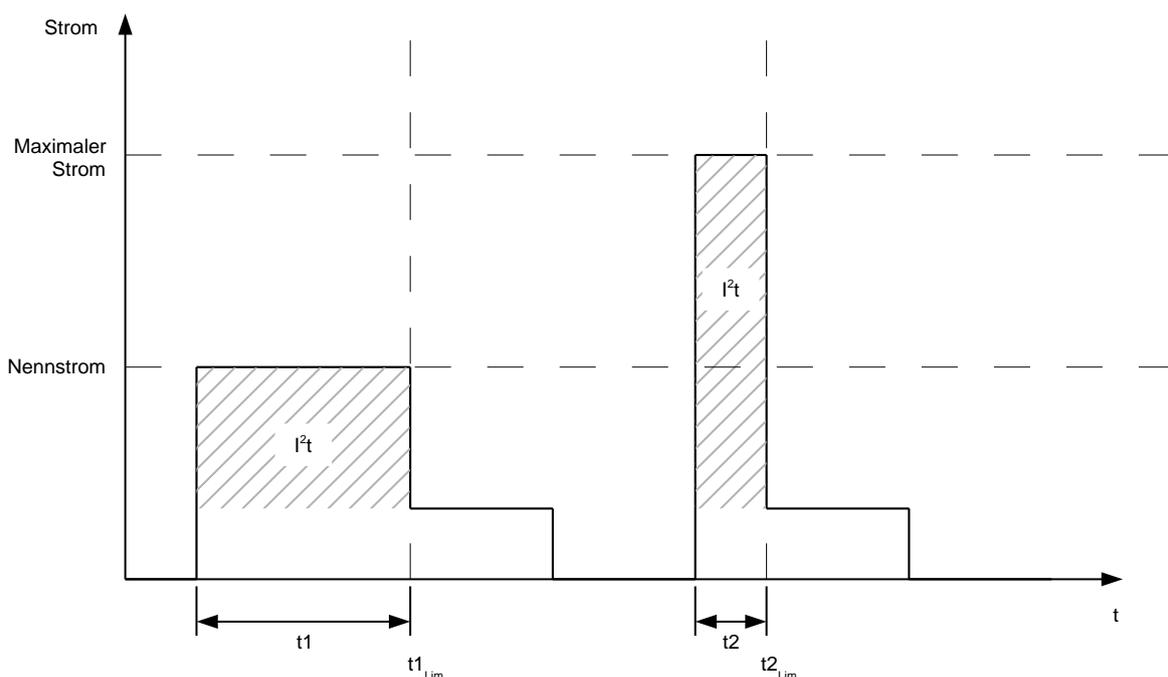
Zum Aktivieren des Modus müssen die drei oben genannten Objekteinträge sinnvoll beschrieben worden sein. Das bedeutet, dass der Maximalstrom größer als der Nennstrom sein muss und ein Zeitwert für die maximale Dauer des Maximalstroms eingetragen sein muss. Wenn diese Bedingungen nicht erfüllt sind, bleibt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

9.2.4 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.

Im folgenden Diagramm sind die Zusammenhänge noch einmal dargestellt.



Im ersten Abschnitt t_1 ist der Stromwert höher als der Nennstrom. Am Zeitpunkt $t_{1\text{Lim}}$ wird I_{Lim}^2 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_{Lim}^2 schneller erreicht, als im Zeitraum t_1 .

10 Programmierung mit NanoJ

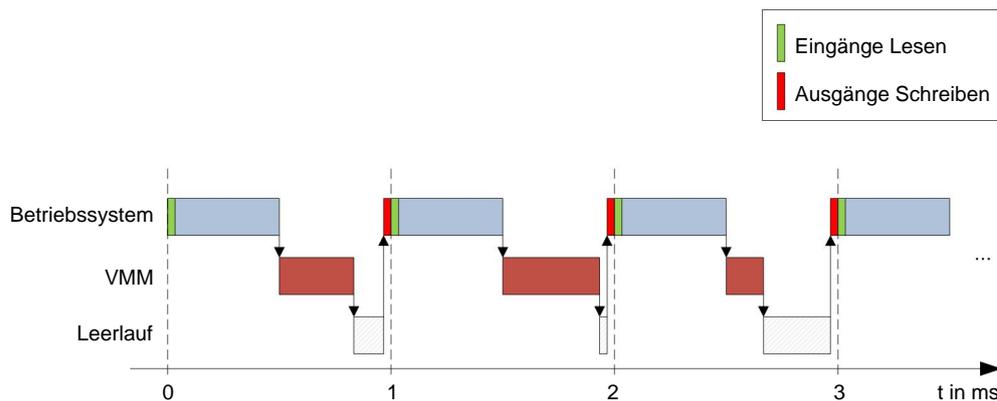
10.1 Einleitung

Der VMM (Virtual Machine Monitor) stellt eine geschützte Ausführungsumgebung innerhalb der Firmware zur Verfügung. In diese kann der Anwender eigene Programme (Benutzerprogramm, "User Program") laden. Diese können dann Funktionen in der Motorsteuerung auslösen, indem beispielsweise Einträge im Objektverzeichnis gelesen oder geschrieben werden.

Durch Verwendung von Schutzmechanismen wird verhindert, dass Benutzerprogramme die eigentliche Firmware zum Absturz bringen können. Im schlimmsten Fall wird lediglich die Ausführung des Benutzerprogramms mit einem im Objektverzeichnis hinterlegten Fehlercode abgebrochen.

10.2 Verfügbare Rechenzeit

Ein Benutzerprogramm erhält zyklisch im 1 ms Takt Rechenzeit (siehe auch nachfolgende Abbildung). Da durch Interrupts und Systemfunktionen der Firmware Rechenzeit verloren geht, stehen dem Benutzerprogramm (abhängig von Betriebsart und Anwendungsfall) nur ca. 30% - 50% dieser Zeit zur Verfügung. In dieser Zeit muss das Benutzerprogramm seine Arbeit erledigen und sich entweder beenden oder durch Aufruf der Funktion `yield()` die Rechenzeit abgeben haben. 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.



Sofern das System feststellt, dass das Benutzerprogramm mehr als die ihm zugeteilte Zeit benötigt, wird dieses beendet und im Objektverzeichnis ein Fehlercode gesetzt. Bei der Entwicklung von Benutzerprogrammen ist daher speziell bei zeitintensiveren Aufgaben eine sorgfältige Überprüfung des Laufzeitverhaltens durchzuführen. So empfiehlt sich daher beispielsweise die Verwendung von Tabellen, anstatt einen Sinuswert über eine `sin` Funktion zu berechnen.

Hinweis

Sollte das VMM-Programm zu lange die Rechenzeit nicht abgeben, wird es vom Betriebssystem beendet. In diesem Fall wird in das Statuswort bei Objekt `2301h` der VMM die Ziffer "4" eingetragen, im Fehlerregister der VMM bei Objekt `2302h` wird die Ziffer "5" (Timeout) notiert.

10.3 Interaktion des Benutzerprogramms mit der Steuerung

10.3.1 Kommunikationsmöglichkeiten

Ein Benutzerprogramm hat mehrere Möglichkeiten, mit der Motorsteuerung 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 seine 1 ms Zeitscheibe erhält, werden dazu von der Firmware die Werte aus dem OD in die Variablen des Benutzerprogramms übertragen. Sobald nun 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 3 Arten von Mappings definiert: Input, Output und Input/Output (In, Out, InOut). Input Mappings lassen sich nur lesen und werden nicht zurück ins OD übertragen. Output Mappings lassen sich nur schreiben. Input/Output Mappings erlauben hingegen Lesen und Schreiben.

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

Über die Angabe der Linker-Section wird in NanoJEasy gesteuert, ob eine Variable im Input-, Output- oder Datenbereich abgelegt wird.

10.3.2 Ausführung eines VMM-Zyklus

Zusammengefasst besteht der Ablauf bei der Ausführung eines VMM-Zyklus hinsichtlich des PDO-Mapping aus folgenden 3 einfachen 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 wieder zurück in das Objektverzeichnis kopieren.

Die Konfiguration der Kopiervorgänge lehnt sich an den CANopen Standard an.

Zusätzlich ist es auch möglich, über Systemcalls auf Werte des Objektverzeichnis zuzugreifen. Dies ist im Allgemeinen deutlich langsamer und daher sind Mappings vorzuziehen. Leider ist jedoch die Anzahl an Mappings begrenzt (jeweils 16 Einträge in In/Out/InOut). Es empfiehlt sich daher, häufig genutzte und veränderte OD-Werte zu mappen und auf weniger häufig genutzte OD-Einträge per Systemcall zuzugreifen. Eine Liste verfügbarer Systemcalls findet sich im Kapitel "**Systemcalls**".

Hinweis

Es wird dringend empfohlen, **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.

10.4 OD-Einträge zur Steuerung und Konfiguration der VMM

10.4.1 OD-Einträge

Der VMM wird durch OD-Einträge im Objekt-Bereich **2300_h** bis **2330_h** gesteuert und konfiguriert. Die Web-Oberfläche übernimmt dabei einen Großteil der Arbeit, so dass ein direkter Zugriff auf die Einträge durch den Anwender im Normalfall nicht notwendig ist.

OD-Index	Name
2300_h	VMM Control (Read/write)
2301_h	VMM Status (Read only)

OD-Index	Name
2302 _h	VMM Error Code (Read only)
2303 _h	Number Of Active User Program (Read/Write)
2304 _h	Table Of Available User Programs
2310 _h	VMM Input Data Selection
2320 _h	VMM Output Data Selection
2330 _h	VMM In/output Data Selection

10.4.2 Beispiel

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

- Kopieren der compilierten Datei über USB
- Schreiben des Wertes "54453554_h" in Objekt 2304_h:01_h.
- Schreiben des Wertes "31000000_h" in Objekt 2304_h:02_h.
- Schreiben des Wertes "1_h" in Objekt 2303_h.
- Überprüfen des Eintrags 2302_h auf Fehlercode.
- Wenn kein Fehler:
 - Starten des Programms durch Beschreiben von Objekt 2300_h, Bit 0 = "1".
- Überprüfen des Eintrags 2302_h auf Fehlercode und des Objekts 2301_h, Bit 0 = "1".

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

10.5 NanoJEasyV2

10.5.1 Installation und Benutzung

Einleitung

Als Alternative zum NanoIP lässt sich Programmierung, Upload und Steuerung eines Benutzerprogramms auch mit der Software "NanoJEasyV2" bewerkstelligen.

Installation

Gehen Sie zur Installation wie folgt vor:

1. Entpacken Sie die Datei "NanoJEasyV2.zip" in einen Ordner Ihrer Wahl.
2. Starten Sie das Programm über die Datei "NanoJEasy.exe".

10.5.2 Programmieren von Benutzerprogrammen

Aufbau Benutzerprogramm

Ein Benutzerprogramm besteht aus mindestens zwei Anweisungen:

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

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

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

Beispiel

Programmieren eines Rechtecksignals in das Objekt **2500_h:01_h**

1. Kopieren Sie folgenden Text in den Editor von NanoJ Easy und speichern Sie diese Datei unter dem Namen "main.cpp" ab.

```
// 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
```

2. Wenn das Programm fehlerfrei übersetzt wurde:

Benennen Sie die Ausgabedatei "main.usr" um in "vmmcode.usr".

3. Kopieren Sie die Datei mittels USB auf die Steuerung (siehe Kapitel "**USB-Anschluss**"). Zum Starten des Programms muss die Steuerung neu gestartet werden, lesen Sie dazu bitte das Kapitel "**NanoJ-Programm**" ab Schritt 2.

10.5.3 Aufbau eines Mappings

Einleitung

Mit dieser Methode lässt sich eine Variable im VMM-Programm direkt mit einem Eintrag im Objektverzeichnis verknüpfen. Das Anlegen des Mappings muss dabei am Anfang der Datei stehen - noch vor der #include "wrapper.h"-Anweisung. Lediglich ein Kommentar oberhalb des Mappings ist erlaubt.

Tipp Benutzen Sie das Mapping, falls Sie den Zugriff auf ein Objekt im Objektverzeichnis häufiger benötigen, wie beispielsweise das Controllword **6040_h** oder das Statusword **6041_h**.

Für den einzelnen Zugriff auf Objekte bieten sich eher die Funktionen `od_write()` und `od_read()` an (siehe Abschnitt "**Zugriff auf das Objektverzeichnis**")

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, also U32, U16, U08, S32, S16 oder S08.
- <NAME>
Der Name der Variable, wie sie später 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 Objektes im Objektverzeichnis.

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

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;
    [...]
}
```

Eventuelle Fehlerquelle

Eine mögliche Fehlerquelle ist ein schreibender Zugriff mittels der Funktion `od_write()` 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:

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

10.6 Systemcalls

10.6.1 Einleitung

Mit Systemcalls ist es möglich, in der Firmware eingebaute Funktionen direkt aus einem Benutzerprogramm aufzurufen. Da eine direkte Codeausführung nur in dem geschützten Bereich der Sandbox möglich ist, wird dies 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 Codeausführung 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.

10.6.2 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 Objektes im Objektverzeichnis
subindex	Subindex des zu schreibenden Objektes 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()` angehalten worden sein.

- void **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 Objektes im Objektverzeichnis
subindex	Subindex des zu lesenden Objektes im Objektverzeichnis
Rückgabewert	Inhalt des OD-Eintrags

Hinweis

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

Beispiel:

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

10.6.3 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
----	-----------------------------------

10.6.4 Debug-Ausgabe

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

- bool **VmmDebugOutputString** (const char *outstring)
- bool **VmmDebugOutputInt** (const U32 val)
- bool **VmmDebugOutputByte** (const U08 val)
- bool **VmmDebugOutputHalfWord** (const U16 val)
- bool **VmmDebugOutputWord** (const U32 val)
- bool **VmmDebugOutputFloat** (const Woat val)

Hinweis

Die Debug Ausgaben werden zunächst in einen eigenen Bereich des OD geschrieben und dann von dort von der Web-Oberfläche ausgelesen. Dieser OD-Eintrag hat den Index **2600_h** und ist 64 Zeichen lang. In Subindex 0 ist immer die Anzahl der bereits geschriebenen Zeichen enthalten.

Ist der Puffer vollgeschrieben, so schlägt `VmmDebugOutputxxx()` zunächst fehl, das Benutzerprogramm wird dann nicht weiter ausgeführt und hält an der Stelle der Debug Ausgabe an. Erst wenn die Web-Oberfläche den Puffer ausgelesen hat und danach Subindex 0 wieder zurückgesetzt hat, wird das Programm wieder fortgesetzt und `VmmDebugOutputxxx()` kehrt ins Benutzerprogramm zurück.

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

11 Objektverzeichnis Beschreibung

11.1 Übersicht

In diesem Kapitel des Handbuchs finden Sie eine Beschreibung aller Objekte.

Sie finden hier Angaben zu:

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

11.2 Aufbau der Objektbeschreibung

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

Funktion

In diesem Abschnitt wird kurz die Funktion des Objektverzeichnisses beschrieben.

Objektbeschreibung

Diese Tabelle gibt detailliert Auskunft über den Datentyp, Vorgabewerte und dergleichen. Eine genaue Beschreibung findet sich im Abschnitt "**Objektbeschreibung**".

Wertebeschreibung

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

Beschreibung

Hier werden genauere Angaben zu den einzelnen Bits eines Eintrags gemacht oder eventuelle Zusammensetzungen erläutert. Eine genauere Beschreibung findet sich im Abschnitt "**Beschreibung**".

11.3 Objektbeschreibung

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

Index

Benennt den Index des Objekts in Hexadezimalschreibweise.

Objektname

Der Name des Objekts.

Object Code

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

- VARIABLE: In dem Fall besteht das Objekt nur aus einer Variable, die mit dem Subindex 0 indiziert wird.
- ARRAY: Diese Objekte bestehen immer aus einem Subindex 0 - welcher die Menge der Untereinträge angibt - und den Untereinträgen selber ab dem Index 1. Der Datentyp innerhalb eines Arrays ändert sich nie, das heißt, Untereintrag 1 und alle folgenden Einträge haben immer den gleichen Datentyp.

- **RECORD:** Diese Objekte bestehen immer aus einem Untereintrag mit dem Subindex 0 - welcher die Menge der Untereinträge angibt - und den Untereinträgen selber ab dem Index 1. Im Gegensatz zu einem ARRAY kann der Datentyp der Subeinträge variieren, das bedeutet, dass beispielsweise Untereintrag 1 einen anderen Datentyp als Untereintrag 2 haben kann.
- **VISIBLE_STRING:** Das Objekt beschreibt eine in ASCII codierte Zeichenkette. 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 der Präfix "UNDSIGNED" benutzt.
- Die Größe der Variable in Bit wird an den Präfix angestellt und kann entweder 8, 16 oder 32 sein.

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 vor zu belegen. Der Wert, der beim Start der Steuerung in das Objekt geschrieben wird, wird in diesem Tabelleneintrag notiert.

11.4 Wertebeschreibung

Hinweis

Der Übersichtlichkeit halber werden einige Subentries zusammen gefasst, 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 der Präfix "UNDSIGNED" benutzt.
- Die Größe der Variable in Bit wird an den Präfix angestellt und kann entweder 8, 16 oder 32 sein.

Zugriff

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

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

PDO Mapping

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

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

Zulässige Werte

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

Vorgabewert

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

11.5 Beschreibung

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

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

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

Beispiel [4]

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

Beispiel [2]

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

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

B

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

A

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

1000h Device Type

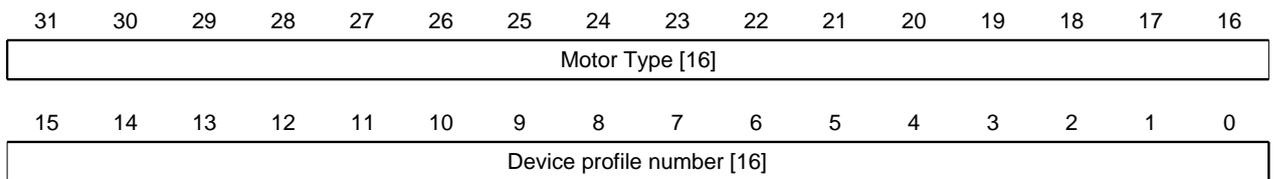
Funktion

Beschreibt den Steuerungstyp.

Objektbeschreibung

Index	1000 _h
Objektname	Device Type
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00040192 _h
Firmware Version	
Änderungshistorie	

Beschreibung



Motor Type[16]

Beschreibt den unterstützten Motor-Typ.

Device profile number[16]

Beschreibt den unterstützten CANopen Standard.

Werte:

0129_h (Vorgabewert): Der DS402 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 _h
Objektname	Error Register
Object Code	VARIABLE
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	
Änderungshistorie	

Beschreibung



GEN

Genereller Fehler

CUR

Strom

VOL

Spannung

TEMP

Temperatur

COM

Kommunikation

PROF

Betrifft das Geräteprofil

RES

Reserviert, immer "0"

1003h Pre-defined Error Field

Funktion

Dieses Objekt beinhaltet einen Fehlerstapel mit bis zu acht Einträgen.

Objektbeschreibung

Index	1003 _h
Objektname	Pre-defined Error Field
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Errors
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h -08 _h
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02
Name	Standard Error Field

Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	04
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	05
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	06
Name	Standard Error Field
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07
Name	Standard Error Field
Datentyp	UNSIGNED32

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

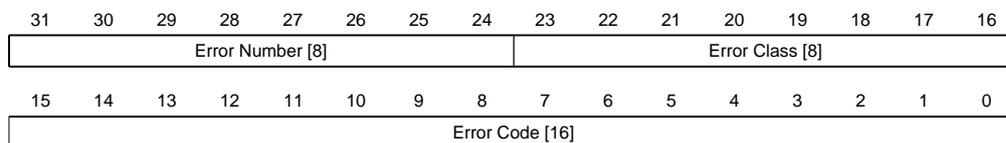
Beschreibung

Allgemeine Funktionsweise

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

Die Anzahl der bereits aufgetreten Fehler lässt sich aus dem Objekt mit dem Subindex 0 ablesen. Wird in dieses Objekt 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
1	Eingangsspannung zu hoch
2	Ausgangsstrom zu hoch
3	Eingangsspannung zu niedrig
4	Fehler am Feldbus
5	Motor dreht - trotz aktivierter Sperre - in die falsche Richtung
6	Nur CANopen: NMT-Master braucht zu lange, um Nodeguarding-Anforderung zu schicken
7	Encoderfehler durch elektrische Störung oder defekte Hardware
8	Encoderfehler; Index während des Autosetups nicht gefunden
9	Fehler in der AB-Spur
10	Positiver Endschalter und Toleranzzone überschritten
11	Negativer Endschalter und Toleranzzone überschritten

Fehlernummer	Beschreibung
12	Temperatur des Gerätes oberhalb 80 °C
13	Die Werte des Objekts 6065_h (Following Error Window) und des Objekts 6066_h (Following Error Time Out) wurden überschritten, es wurde ein Fault ausgelöst. Dieser Fehler muss mit dem Bit 7 im Objekt 3202_h aktiviert werden.

Error Class[8]

Dieses Byte ist identisch mit dem Objekt **1001_h**

Error Code[16]

Die Bedeutung der beiden Bytes lässt sich aus der nachfolgenden Tabelle zu entnehmen.

Error Code	Beschreibung
1000 _h	Allgemeiner Fehler
2310 _h	Strom am Ausgang der Steuerung zu groß
3210 _h	Über-/ Unterspannung am Eingang der Steuerung
4200 _h	Temperaturfehler innerhalb der Steuerung
7305 _h	Inkrementaler Sensor 1 fehlerhaft
8000 _h	Fehler bei der Feldbusüberwachung
8130 _h	Nur CANopen: "Life Guard" Fehler oder "Heartbeat" - Fehler
8611 _h	Fehler in der Positionsüberwachung: Folgefehler zu groß
8612 _h	Fehler in der Positionsüberwachung: Referenzlimit

1008h Manufacturer Device Name

Funktion

Enthält den Gerätenamen als Zeichenkette.

Objektbeschreibung

Index	1008 _h
Objektname	Manufacturer Device Name
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	
Änderungshistorie	

Beschreibung

In Subindex 0 dieses Objekts steht die Länge der Zeichenkette. Ab Subindex 1 sind die einzelnen Zeichen enthalten. Die Zeichenkette ist nicht per Null-Zeichen terminiert.

1009h Manufacturer Hardware Version

Funktion

Dieses Objekt enthält die Hardware Version als Zeichenkette.

Objektbeschreibung

Index	1009 _h
Objektname	Manufacturer Hardware Version
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	
Änderungshistorie	

Beschreibung

In Subindex 0 dieses Objekts steht die Länge der Zeichenkette. Ab Subindex 1 sind die einzelnen Zeichen enthalten. Die Zeichenkette ist nicht per Null-Zeichen terminiert.

100Ah Manufacturer Software Version

Funktion

Dieses Objekt enthält die Software Version als Zeichenkette.

Objektbeschreibung

Index	100A _h
Objektname	Manufacturer Software Version
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	DEV-B29286
Firmware Version	
Änderungshistorie	

Beschreibung

In Subindex 0 dieses Objekts steht die Länge der Zeichenkette. Ab Subindex 1 sind die einzelnen Zeichen enthalten. Die Zeichenkette ist nicht per Null-Zeichen terminiert.

1010h Store Parameter

Funktion

Speichert die NodeID aus dem Objekt **2009_h** und die Baudraten-Einstellung vom Objekt **2005_h** fest ein, damit die nach einem Neustart oder Spannungsausfall wieder verfügbar sind. Zum Speichern der Werte muss in den Subindex 1 der Zahlenwert **65766173_h** geschrieben werden.

Hinweis

Die Werte werden **nur gespeichert**. Gültig werden die gespeicherten Einstellungen erst nach einer der folgenden Bedingungen:

- Ab- und Anschalten der Spannungsversorgung
- Senden einer CANopen-Nachricht "Reset Communication"
- Senden einer CANopen-Nachricht "Reset Node"

Achtung:

Objektbeschreibung

Index	1010 _h
Objektname	Store Parameter
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	Save All The Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Save The Comm Parameters To Non-volatile Memory
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Dieses Objekt dient dazu, die Werte aus dem Objektverzeichnis für NodeID (Objekt **2005_h**) und Baudrate (Objekt **2009_h**) nach einer Änderung für CANopen zu speichern.

Um den Speichervorgang zu starten muss der Wert 65766173_h in den Subindex 2 geschrieben werden. Das entspricht dezimal der 1702257011_d bzw. dem ASCII String "SAVE".

1011h Restore Default Parameter

Funktion

Mit diesem Objekt kann das gesamte Objektverzeichnis auf die Defaultwerte zurückgesetzt werden.

Objektbeschreibung

Index	1011 _h
Objektname	Restore Default Parameter
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	Restore All Default Parameters
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	

Vorgabewert 00000000_h

Beschreibung

Wird der Wert 64616F6C_h in dieses Objekt geschrieben, wird das gesamte Objektverzeichnis auf die Defaultwerte zurückgesetzt.

Hinweis

Um das Zurücksetzen wirksam zu machen, bootet die Steuerung im Anschluss neu.

1018h Identity Object

Funktion

Das Objekt enthält Informationen zum Hersteller, den Produktcode und die Revisions- und Seriennummer.

Objektbeschreibung

Index	1018 _h
Objektname	Identity Object
Object Code	RECORD
Datentyp	IDENTITY
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h

Subindex	01 _h
Name	Vendor-ID
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000026C _h

Subindex	02 _h
Name	Product Code

Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000003 _h
<hr/>	
Subindex	03 _h
Name	Revision Number
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
<hr/>	
Subindex	04 _h
Name	Serial Number
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

1600h Drive Control

Funktion

Dieses Objekt enthält ein Input PDO Mapping des aktiven Feldbusses (EtherCAT oder CANopen).

Objektbeschreibung

Index	1600 _h
Objektname	Drive Control
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein

Zulässige Werte

Vorgabewert 00_h

Subindex 01_h

Name 1st Object To Be Mapped

Datentyp UNSIGNED32

Zugriff lesen/schreiben

PDO Mapping nein

Zulässige Werte

Vorgabewert 00000000_h

Subindex 02_h

Name 2nd Object To Be Mapped

Datentyp UNSIGNED32

Zugriff lesen/schreiben

PDO Mapping nein

Zulässige Werte

Vorgabewert 00000000_h

Subindex 03_h

Name 3rd Object To Be Mapped

Datentyp UNSIGNED32

Zugriff lesen/schreiben

PDO Mapping nein

Zulässige Werte

Vorgabewert 00000000_h

Subindex 04_h

Name 4th Object To Be Mapped

Datentyp UNSIGNED32

Zugriff lesen/schreiben

PDO Mapping nein

Zulässige Werte

Vorgabewert 00000000_h

Subindex 05_h

Name 5th Object To Be Mapped

Datentyp UNSIGNED32

Zugriff lesen/schreiben

PDO Mapping nein

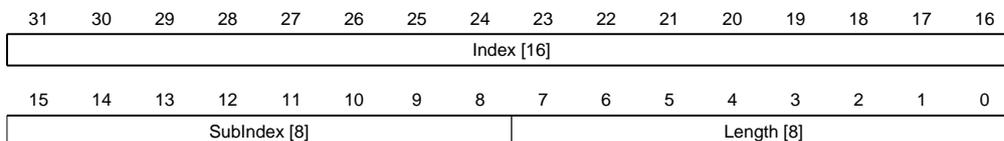
Zulässige Werte

Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

1601h Positioning Control

Funktion

Dieses Objekt enthält ein Input PDO Mapping des aktiven Feldbusses (EtherCAT oder CANopen).

Objektbeschreibung

Index	1601 _h
Objektname	Positioning Control
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32

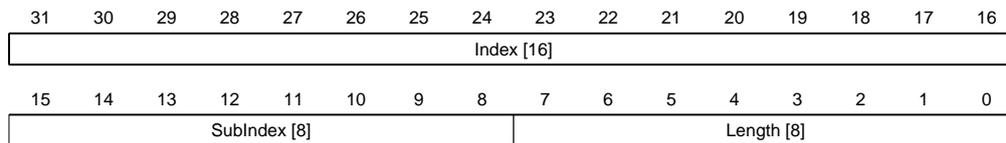
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

1602h Velocity Control

Funktion

Dieses Objekt enthält ein Input PDO Mapping des aktiven Feldbusses (EtherCAT oder CANopen).

Objektbeschreibung

Index	1602 _h
Objektname	Velocity Control
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h

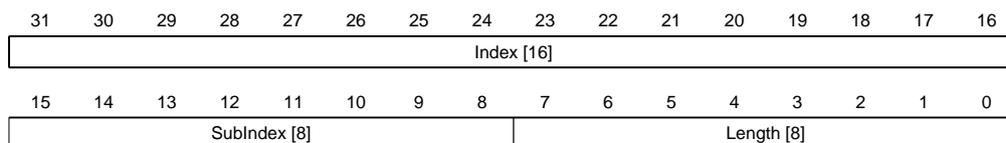
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

1603h Output Control

Funktion

Dieses Objekt enthält ein Input PDO Mapping des aktiven Feldbusses (EtherCAT oder CANopen).

Objektbeschreibung

Index	1603 _h
Objektname	Output Control
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

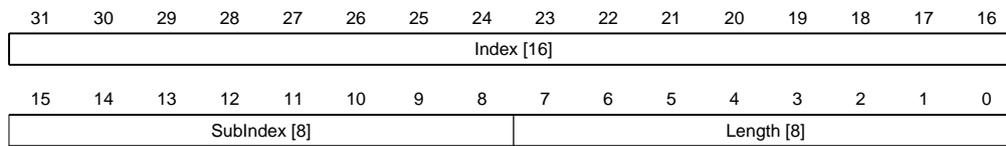
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

1A00h Drive Status

Funktion

Dieses Objekt beschreibt ein Output PDO Mapping des aktiven Feldbusses (EtherCAT oder CANopen).

Objektbeschreibung

Index	1A00 _h
Objektname	Drive Status
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h to 08 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

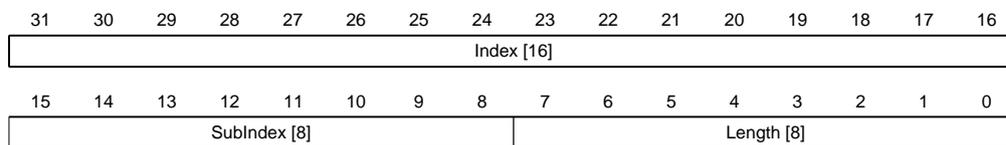
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein

Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

1A01h Positioning Status

Funktion

Dieses Objekt beschreibt ein Output PDO Mapping des aktiven Feldbusses (EtherCAT oder CANopen).

Objektbeschreibung

Index	1A01 _h
-------	-------------------

Objektname	Positioning Status
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	

Wertebeschreibung

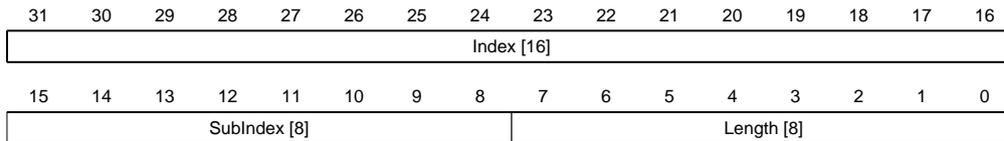
Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

1A02h Velocity Status

Funktion

Dieses Objekt beschreibt ein Output PDO Mapping des aktiven Feldbusses (EtherCAT oder CANopen).

Objektbeschreibung

Index	1A02 _h
Objektname	Velocity Status
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

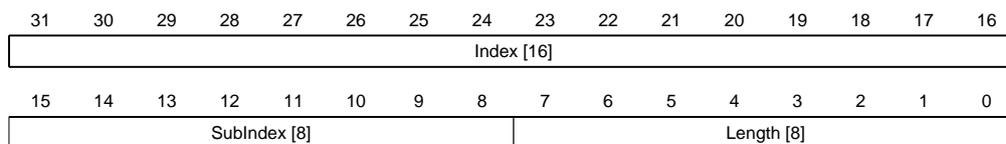
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h

Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

1A03h Input Status

Funktion

Dieses Objekt beschreibt ein Output PDO Mapping des aktiven Feldbusses (EtherCAT oder CANopen).

Objektbeschreibung

Index	1A03 _h
Objektname	Input Status
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	

Änderungshistorie

Wertebeschreibung

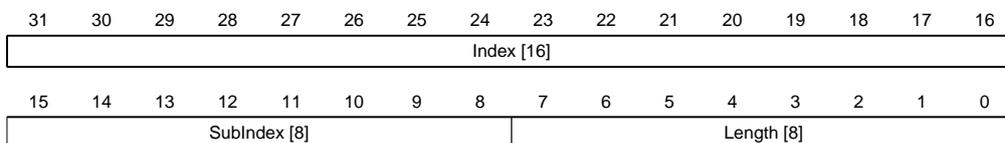
Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h
Name	1st Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	2nd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	3rd Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	4th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	05 _h
Name	5th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	6th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	7th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	8th Object To Be Mapped
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

2010h IP-Configuration

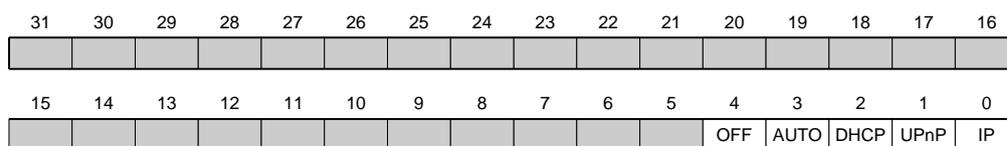
Funktion

Über dieses Objekt wird die Netzwerkkonfiguration eingestellt.

Objektbeschreibung

Index	2010 _h
Objektname	IP-Configuration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung



IP

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

OFF

Wert = "1": Die Netzwerkschnittstelle wird deaktiviert

2011h Static-IP-Address

Funktion

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

Objektbeschreibung

Index	2011 _h
Objektname	Static-IP-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

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

IP Address Part 1 [8]

Gibt den ersten Teil der IP-Adresse an

IP Address Part 2 [8]

Gibt den zweiten Teil der IP-Adresse an

IP Address Part 3 [8]

Gibt den dritten Teil der IP-Adresse an

IP Address Part 4 [8]

Gibt den vierten Teil der IP-Adresse an

Beispiel

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

192 => C0_h

168 => A8_h

2 => 02_h

0 => 0

Der zugehörige Einstellwert lautet dann C0A80200_h.

2012h Static-IP-Subnet-Mask

Funktion

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

Objektbeschreibung

Index	2012 _h
Objektname	Static-IP-Subnet-Mask
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

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 Maks Part 1 [8]

Gibt den ersten Teil der Subnetzmaske an

Subnet Maks Part 2 [8]

Gibt den zweiten Teil der Subnetzmaske an

Subnet Maks Part 3 [8]

Gibt den dritten Teil der Subnetzmaske an

Subnet Maks Part 4 [8]

Gibt den vierten Teil der Subnetzmaske an

Beispiel

Die Klasse-C Netzwerkmaske 255 . 255 . 255 . 0 wird zuerst in das Hexadezimalsystem umgewandelt und ergibt dann folgenden Konfigurationswert:

255 => FF_h

0 => 0

Der zugehörige Einstellwert lautet dann FFFFFFFF00_h.

2018h Current-IP-Address

Funktion

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

Objektbeschreibung

Index	2018 _h
Objektname	Current-IP-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

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

IP Address Part 1 [8]

Gibt den ersten Teil der IP-Adresse an

IP Address Part 2 [8]

Gibt den zweiten Teil der IP-Adresse an

IP Address Part 3 [8]

Gibt den dritten Teil der IP-Adresse an

IP Address Part 4 [8]

Gibt den vierten Teil der IP-Adresse an

Beispiel

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

192 => C0_h

168 => A8_h

2 => 02_h

0 => 0

Der zugehörige Einstellwert lautet dann C0A80200_h.

2019h Current-IP-Subnet-Mask

Funktion

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

Objektbeschreibung

Index	2019 _h
Objektname	Current-IP-Subnet-Mask
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

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 Maks Part 1 [8]

Gibt den ersten Teil der Subnetzmaske an

Subnet Maks Part 2 [8]

Gibt den zweiten Teil der Subnetzmaske an

Subnet Maks Part 3 [8]

Gibt den dritten Teil der Subnetzmaske an

Subnet Maks Part 4 [8]

Gibt den vierten Teil der Subnetzmaske an

Beispiel

Die Klasse-C Netzwerkmaske 255 . 255 . 255 . 0 wird zuerst in das Hexadezimalsystem umgewandelt und ergibt dann folgenden Konfigurationswert:

255 => FF_h

0 => 0

Der zugehörige Einstellwert lautet dann FFFFFFFF00_h.

2020h ApplInfo-Static-IP-Address

Funktion

Enthält die von Nanoflash vorgegebene IP-Adresse in Form eines 32-Bit Wortes.

Objektbeschreibung

Index	2020 _h
Objektname	ApplInfo-Static-IP-Address
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

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

IP Address Part 1 [8]

Gibt den ersten Teil der IP-Adresse an

IP Address Part 2 [8]

Gibt den zweiten Teil der IP-Adresse an

IP Address Part 3 [8]

Gibt den dritten Teil der IP-Adresse an

IP Address Part 4 [8]

Gibt den vierten Teil der IP-Adresse an

Beispiel

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

192 => C0_h

168 => A8_h

2 => 02_h

0 => 0

Der zugehörige Einstellwert lautet dann C0A80200_h.

2021h ApplInfo-Static-IP-Subnet-Mask

Funktion

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

Objektbeschreibung

Index	2021 _h
Objektname	ApplInfo-Static-IP-Subnet-Mask
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

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 Maks Part 1 [8]

Gibt den ersten Teil der Subnetzmaske an

Subnet Maks Part 2 [8]

Gibt den zweiten Teil der Subnetzmaske an

Subnet Maks Part 3 [8]

Gibt den dritten Teil der Subnetzmaske an

Subnet Maks Part 4 [8]

Gibt den vierten Teil der Subnetzmaske an

Beispiel

Die Klasse-C Netzwerkmaske 255 . 255 . 255 . 0 wird zuerst in das Hexadezimalsystem umgewandelt und ergibt dann folgenden Konfigurationswert:

255 => FF_h

0 => 0

Der zugehörige Einstellwert lautet dann FFFFFFFF00_h.

2022h Drive Serial Number

Funktion

Dieses Objekt hält die Seriennummer der Steuerung.

Objektbeschreibung

Index	2022 _h
Objektname	Drive Serial Number
Object Code	VARIABLE
Datentyp	VISIBLE_STRING
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	
Firmware Version	
Änderungshistorie	Version : Eintrag Name geändert von "Drive serial number" auf "Drive Serial Number"

2030h Pole Pair Count

Funktion

Enthält die Polpaarzahl des angeschlossenen Motors.

Objektbeschreibung

Index	2030 _h
Objektname	Pole Pair Count
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000032 _h
Firmware Version	
Änderungshistorie	Version : Eintrag Name geändert von "Pole pair count" auf "Pole Pair Count"

2031h Peak Current

Funktion

Gibt den Maximalstrom in mA an.

Objektbeschreibung

Index	2031 _h
Objektname	Peak Current
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000009C4 _h
Firmware Version	
Änderungshistorie	

2032h Maximum Speed

Funktion

Gibt die maximal zulässige Geschwindigkeit des V-Reglers in U/s oder U/min an.

Objektbeschreibung

Index	2032 _h
Objektname	Maximum Speed
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00030D40 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Die Umrechnung erfolgt anhand der in Objekt **604C_h** festgelegten Zähler und Nenner.

2033h Plunger Block

Funktion

Gibt die Positionsänderung in Benutzereinheiten (entsprechend Target Position **607A_h**) an, die maximal in die entsprechende Richtung erlaubt ist.

Objektbeschreibung

Index	2033 _h
Objektname	Plunger Block
Object Code	VARIABLE

Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Damit wird ein elektronischer Sperr-Riegel realisiert.

Der Wert 0 schaltet die Überwachung ab.

Der Wert 100 bedeutet beispielsweise, dass sich der Antrieb beliebig weit in die negative Richtung drehen darf, sobald er sich jedoch um mehr als 100 Schritte in die positive Richtung bewegt, wird der Motor sofort gestoppt und ein Fehler ausgelöst.

Dadurch kann z. B. beim Aufwickeln von Fäden ein versehentliches Abwickeln unterbunden werden.

2034h Upper Voltage Warning Level

Funktion

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

Objektbeschreibung

Index	2034 _h
Objektname	Upper Voltage Warning Level
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000C92C _h
Firmware Version	
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 **2036_h** minus 2 Volt) ist.

2035h Lower Voltage Warning Level

Funktion

Dieses Objekt enthält den Schwellwert für den Fehler "Unterspannung" in Volt.

Objektbeschreibung

Index	2035 _h
Objektname	Lower Voltage Warning Level
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00004650 _h
Firmware Version	
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 (Spannung des Objekts 2035_h plus 2 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 Stromreduktion aktiviert wird.

Objektbeschreibung

Index	2036 _h
Objektname	Open Loop Current Reduction Idle Time
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Firmware Version	
Änderungshistorie	

2037h Open Loop Current Reduction Value/factor

Funktion

Dieses Objekt beschreibt den Wert, auf den der Strom reduziert werden soll, wenn die Stromabsenkung im Open Loop aktiviert wird.

Objektbeschreibung

Index	2037 _h
-------	-------------------

Objektname	Open Loop Current Reduction Value/factor
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	FFFFFFCE _h
Firmware Version	
Änderungshistorie	

Beschreibung

Ist der Wert negativ zwischen -100 und -1, so gilt das als prozentualer Reduktionsfaktor bezogen auf den Maximalstrom (**2031_h**). Der Wert -100 entspricht dabei 100% des Wertes im Objekt **2031_h**, der Wert -50 wird als 50% des Objekts **2031_h** interpretiert, usw.

Ist der Wert positiv, wird der Strom auf den im Objekt **2037_h** eingetragenen Wert in mA reduziert.

2038h Brake Controller Timing

Funktion

Dieses Objekt enthält die Zeiten für die Bremsensteuerung in Millisekunden sowie die PWM-Freugenz und Tastgrad.

Objektbeschreibung

Index	2038 _h
Objektname	Brake Controller Timing
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h

Subindex	01 _h
Name	Close Brake Idle Time
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	02 _h
Name	Shutdown Power Idle Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	03 _h
Name	Open Brake Delay Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	04 _h
Name	Start Operation Delay Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	05 _h
Name	PWM Frequency
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	in between 1 and 2000 (7D0 _h)
Vorgabewert	00000000 _h
Subindex	06 _h
Name	PWM Duty Cycle
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO Mapping	nein
Zulässige Werte	in between 2 and 100
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Zeit zwischen dem Zeitpunkt des Stillstands des Motors und dem Schließen der Bremse.
- 02_h: Zeit zwischen dem Zeitpunkt des Schließens der Bremse und dem Absenken des Stromes.
- 03_h: Zeit zwischen dem Zeitpunkt des Setzens eines neuen Fahrbefehls und dem Öffnen der Bremse.
- 04_h: Zeit zwischen dem Öffnen der Bremse und dem Anlaufen des Motors.
- 05_h: Frequenz der Bremsen-PWM in Herz.
- 06_h: Tastgrad der Bremsen-PWM in Prozent.

2039h Motor Currents

Funktion

Dieses Objekt enthält die gemessenen Motorströme in mA.

Objektbeschreibung

Index	2039 _h
Objektname	Motor Currents
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h

Subindex	01 _h
Name	I_d
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	I_q
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	I_a
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	I_b
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

203Ah Homing On Block Configuration

Funktion

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

Objektbeschreibung

Index	203A _h
Objektname	Homing On Block Configuration
Object Code	ARRAY
Datentyp	UNSIGNED32
Zugriff	
PDO Mapping	
Zulässige Werte	
Vorgabewert	
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	03 _h
Subindex	01 _h
Name	Minimum Current For Block Detection
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000004EC _h
Subindex	02 _h
Name	Period Of Blocking
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03
Name	Block Detection Time
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000005 _h

Beschreibung

Die Subentries haben folgende Funktion:

- 01_h: Gibt den Stromgrenzwert an, ab dem ein Blockieren detektiert werden soll.
- 02_h: Gibt die Zeit in ms an, die der Motor nach der Blockdetektion trotzdem noch gegen den Block fahren soll.
- 03_h: Gibt die Zeit in ms an, die der Strom mindestens über der angegebenen Stromschwelle sein muss, um ein Blockieren zu detektieren.

203Bh I2t Parameters

Funktion

Dieses Objekt hält die Parameter für die I²t-Überwachung.

Die I²t-Überwachung wird aktiviert, in dem in 203B_h:2 ein Wert größer 0 eingetragen wird (siehe **Kapitel Motor-Überlastungsschutz**).

I²t kann nur für den Closed Loop-Betrieb verwendet werden, mit einer Ausnahme: Wenn I²t im Open Loop-Betrieb aktiviert ist, wird der Strom auf den eingestellten Nominalstrom begrenzt, auch wenn der eingestellte Maximalstrom größer ist. Dieses Feature wurde aus Sicherheitsgründen implementiert, damit man auch aus dem Closed Loop-Betrieb mit sehr hohem kurzzeitigem Maximalstrom in den Open Loop-Betrieb wechseln kann, ohne den Motor zu schädigen.

Objektbeschreibung

Index	203B _h
Objektname	I2t Parameters
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	06 _h
Subindex	01 _h
Name	Nominal Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Maximum Duration Of Peak Current
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	03
Name	Threshold
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04
Name	CalcValue
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05
Name	LimitedCurrent
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06
Name	Status
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subentries haben folgende Funktion:

- 01_h: Gibt den Nennstrom in mA an, muss kleiner als der Maximalstrom **2031**_h sein, sonst wird die Überwachung nicht aktiviert.
- 02_h: Gibt die maximale Dauer des Spitzenstroms in ms an.
- 03_h: Threshold, gibt die Grenze in mAs an, von der abhängt, ob auf Maximalstrom oder Nominalstrom geschaltet wird.
- 04_h: CalcValue, gibt den berechneten Wert an, welcher mit Threshold verglichen wird, um den Strom einzustellen.

- 05_h: LimitedCurrent, zeigt den gegenwärtigen Stromwert an, der von I²_t eingestellt wurde.
- 06_h: aktueller Status. Ist der Subentry-Wert "0", ist I²_t deaktiviert, ist der Wert "1", wird I²_t aktiviert

2050h Encoder Alignment

Funktion

Dieser Wert gibt den Winkelversatz zwischen Rotor und elektrischem Feld an.

Objektbeschreibung

Index	2050 _h
Objektnamen	Encoder Alignment
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Die exakte Bestimmung ist nur über das Auto-Setup möglich. Das Vorhandensein dieses Wertes ist für den Closed Loop-Betrieb erforderlich.

2051h Encoder Optimization

Funktion

Enthält Kompensationswerte, um einen besseren Rundlauf im Closed Loop-Betrieb zu erreichen.

Objektbeschreibung

Index	2051 _h
Objektnamen	Encoder Optimization
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein

Zulässige Werte	
Vorgabewert	03 _h
Subindex	
Subindex	01 _h
Name	Parameter 1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	
Subindex	02 _h
Name	Parameter 2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	
Subindex	03 _h
Name	Parameter 3
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die exakte Bestimmung ist nur über das Auto-Setup möglich.

2052h Encoder Resolution

Funktion

Beinhaltet die Auflösung des Encoders, der zur elektrischen Kommutierung verwendet wird.

Objektbeschreibung

Index	2052 _h
Objektname	Encoder Resolution
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein

Zulässige Werte	
Vorgabewert	00001000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Ein negativer Wert bedeutet, dass der Encoder gegensinnig zum Motor angetrieben wird. Dies lässt sich durch Umpolen einer Motorwicklung korrigieren.

2053h Index Polarity

Funktion

Gibt die Indexpolarität an.

Objektbeschreibung

Index	2053 _h
Objektname	Index Polarity
Object Code	VARIABLE
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Der Wert 0 bedeutet, dass der Index nicht invertiert ist.

Der Wert 1 bedeutet, der Index ist invertiert angeschlossen und wird in der Firmware invertiert.

2054h Index Width

Funktion

Gibt die Indexbreite in einer internen Rechengröße an.

Objektbeschreibung

Index	2054 _h
Objektname	Index Width
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	

Vorgabewert	FFFFFFFF _h
Firmware Version	
Änderungshistorie	

Beschreibung

Ist dieser Wert ungleich 0, wird der Encoder auf Fehler überwacht.

Der Wert -1 (FFFFFFFF_h) deaktiviert die Encoderüberwachung.

2056h Limit Switch Tolerance Band

Funktion

Gibt an, wie weit positive oder negative Endschalter überfahren werden dürfen, bis die Steuerung einen Fehler auslöst.

Dieses Toleranzband ist beispielweise erforderlich, um Referenzfahrten - bei denen Endschalter betätigt werden können - fehlerfrei abschließen zu können.

Objektbeschreibung

Index	2056 _h
Objektname	Limit Switch Tolerance Band
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h
Firmware Version	
Änderungshistorie	

2057h Clock Direction Multiplier

Funktion

Mit diesem Wert wird der Takt-Zählwert im Takt-/Richtungsmodus multipliziert, bevor er weiterverarbeitet wird.

Objektbeschreibung

Index	2057 _h
Objektname	Clock Direction Multiplier
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000080 _h

Firmware Version
Änderungshistorie

2058h Clock Direction Divider

Funktion

Durch diesen Wert wird der Takt-Zählwert im Takt-/Richtungsmodus dividiert, bevor er weiterverarbeitet wird.

Objektbeschreibung

Index	2058 _h
Objektname	Clock Direction Divider
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Firmware Version	
Änderungshistorie	

2059h Encoder Configuration

Funktion

DESCRIPTION!

Objektbeschreibung

Index	2059 _h
Objektname	Encoder Configuration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

2060h Compensate Polepair Count

Funktion

Ermöglicht, motorunabhängig Fahrsätze zu beauftragen.

Objektbeschreibung

Index	2060 _h
Objektname	Compensate Polepair Count
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Wird dieser Eintrag auf 1 gesetzt, wird die Polpaarzahl automatisch bei allen Positions-, Geschwindigkeits-, Beschleunigungs- und Jerk-Parametern eingerechnet.

Ist der Wert 0, geht die Polpaarzahl, wie bei herkömmlichen Schrittmotorsteuerungen, in Vorgabewerte mit ein und muss bei einem Motorwechsel berücksichtigt werden.

2061h Velocity Numerator

Funktion

Beinhaltet den Zähler, der zum Umrechnen der Geschwindigkeitsvorgaben im Profile Position Mode verwendet wird.

Objektbeschreibung

Index	2061 _h
Objektname	Velocity Numerator
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Die interne Rechengröße bezieht sich auf ganze mechanische (**2060_h=1**) oder elektrische (**2060_h=0**) Umdrehungen pro Sekunde.

So kann z. B. durch Setzen von Objekt **2061_h=1** und Objekt **2062_h=60** die Geschwindigkeit im Profile Position Mode in Umdrehung/min angegeben werden.

2062h Velocity Denominator

Funktion

Beinhaltet den Nenner, der zum Umrechnen der Geschwindigkeitsvorgaben im Profile Position Mode verwendet wird.

Objektbeschreibung

Index	2062 _h
Objektname	Velocity Denominator
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000003C _h
Firmware Version	
Änderungshistorie	

Beschreibung

Die interne Rechengröße bezieht sich auf ganze mechanische (**2060_h=1**) oder elektrische (**2060_h=0**) Umdrehungen pro Sekunde.

So kann z. B. durch Setzen von Objekt **2061_h=1** und Objekt **2062_h=60** die Geschwindigkeit im Profile Position Mode in Umdrehungen pro Minute angegeben werden.

2063h Acceleration Numerator

Funktion

Beinhaltet den Zähler, der zum Umrechnen der Beschleunigungsvorgaben im Profile Position Mode verwendet wird.

Objektbeschreibung

Index	2063 _h
Objektname	Acceleration Numerator
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	

Vorgabewert	00000001 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Die interne Rechengröße bezieht sich auf ganze mechanische (**2060_h=1**) oder elektrische (**2060_h=0**) Umdrehungen pro Sekunde.

So kann z. B. durch Setzen von Objekt **2063_h=1** und Objekt **2064_h=60** die Beschleunigung im Profile Position Mode in (Umdrehungen/min)/s² angegeben werden.

2064h Acceleration Denominator

Funktion

Beinhaltet den Nenner, der zum Umrechnen der Beschleunigungsvorgaben im Profile Position Mode verwendet wird.

Objektbeschreibung

Index	2064 _h
Objektname	Acceleration Denominator
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000003C _h
Firmware Version	
Änderungshistorie	

Beschreibung

Die interne Rechengröße bezieht sich auf ganze mechanische (**2060_h=1**) oder elektrische (**2060_h=0**) Umdrehungen pro Sekunde.

So kann z. B. durch Setzen von Objekt **2063_h=1** und Objekt **2064_h=60** die Beschleunigung im Profile Position Mode in (Umdrehungen/min)/s² angegeben werden.

2065h Jerk Numerator

Funktion

Beinhaltet den Zähler, der zum Umrechnen der Ruckvorgaben im Profile Position Mode verwendet wird.

Objektbeschreibung

Index	2065 _h
Objektname	Jerk Numerator
Object Code	VARIABLE

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Die interne Rechengröße bezieht sich auf ganze mechanische (**2060_h=1**) oder elektrische (**2060_h=0**) Umdrehungen pro Sekunde in der dritten Potenz.

So kann z. B. durch Setzen von Objekt **2065_h=1** und Objekt **2066_h=60** der Ruck im Profile Position Mode in (Umdrehungen/min)/s² angegeben werden.

2066h Jerk Denominator

Funktion

Beinhaltet den Nenner, der zum Umrechnen der Ruckvorgaben im Profile Position Mode verwendet wird.

Objektbeschreibung

Index	2066 _h
Objektname	Jerk Denominator
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000003C _h
Firmware Version	
Änderungshistorie	

Beschreibung

Die interne Rechengröße bezieht sich auf ganze mechanische (**2060_h=1**) oder elektrische (**2060_h=0**) Umdrehungen pro Sekunde.

So kann z. B. durch Setzen von Objekt **2065_h=1** und Objekt **2066_h=60** die Beschleunigung im Profile Position Mode in (Umdrehungen/min)/s² angegeben werden.

2084h Bootup Delay

Funktion

Mittels diesem Objekts lässt sich der Zeitraum zwischen Anlegen der Versorgungsspannung an die Steuerung und dem Bereitstellen der Funktionalität der Steuerung in Millisekunden angeben.

Objektbeschreibung

Index	2084 _h
Objektname	Bootup Delay
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

2101h Fieldbus Module

Funktion

Zeigt den Typ des aufgesteckten Feldbusmoduls an.

Objektbeschreibung

Index	2101 _h
Objektname	Fieldbus Module
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													RS485	CAN	ECAT

ECAT

Wert = "1": Der Feldbus EtherCAT ist verfügbar

CAN

Wert = "1": Der Feldbus CANopen ist verfügbar

RS485

Wert = "1": Eine RS485 Schnittstelle ist verfügbar

2200h Sampler Control

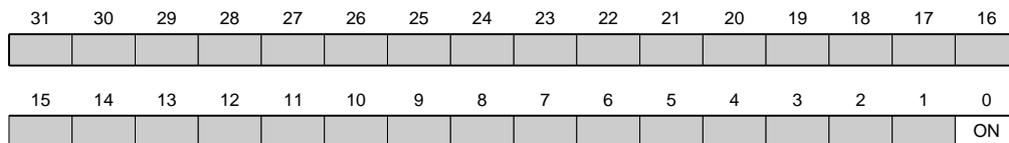
Funktion

Steuert den eingebauten Sampler, der dazu dient, zyklisch beliebige Werte aus dem Object Dictionary aufzuzeichnen.

Objektbeschreibung

Index	2200 _h
Objektname	Sampler Control
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung



ON

Wert = "1": Der Sampler wird aktiviert

2201h Sampler Status

Funktion

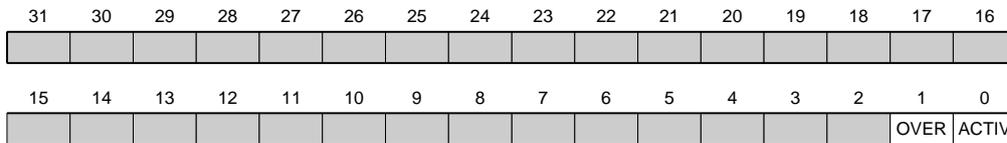
Zeigt den Betriebszustand des eingebauten Samplers an.

Objektbeschreibung

Index	2201 _h
Objektname	Sampler Status
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	

Änderungshistorie

Beschreibung



ACTIV

Wert = "1": Sampler ist aktiv und zeichnet Daten auf.

OVER

Wert = "1": Der Aufzeichnungspuffer wurde nicht schnell genug ausgelesen und Daten sind verloren gegangen. Daraufhin wird der Sampler gestoppt und muss erneut durch eine steigende Flanke im Objekt **2200_h** Bit 0 gestartet werden.

2202h Sample Data Selection

Funktion

Hier kann, analog zu Objekt **1600_h** ff. gesteuert werden, welche Daten pro Abtastung gemeinsam erfasst werden. In der aktuellen Firmware beträgt die Größe des Sampler-Puffers 12.000 Bytes.

Objektbeschreibung

Index	2202 _h
Objektname	Sample Data Selection
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Sample Value #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein

Zulässige Werte

Vorgabewert 60430010_h

Subindex 02_h
Name Sample Value #2
Datentyp UNSIGNED32
Zugriff lesen/schreiben
PDO Mapping nein

Zulässige Werte

Vorgabewert 22030220_h

Subindex 03_h
Name Sample Value #3
Datentyp UNSIGNED32
Zugriff lesen/schreiben
PDO Mapping nein

Zulässige Werte

Vorgabewert 00000000_h

Subindex 04_h
Name Sample Value #4
Datentyp UNSIGNED32
Zugriff lesen/schreiben
PDO Mapping nein

Zulässige Werte

Vorgabewert 00000000_h

Subindex 05_h
Name Sample Value #5
Datentyp UNSIGNED32
Zugriff lesen/schreiben
PDO Mapping nein

Zulässige Werte

Vorgabewert 00000000_h

Subindex 06_h
Name Sample Value #6
Datentyp UNSIGNED32
Zugriff lesen/schreiben
PDO Mapping nein

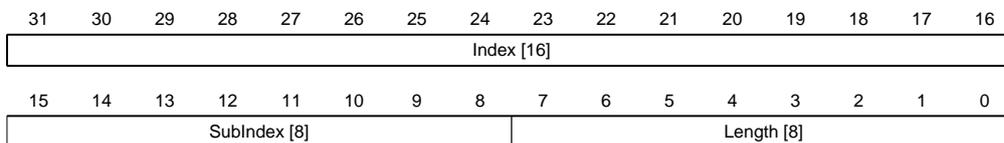
Zulässige Werte

Vorgabewert	00000000 _h
Subindex	07 _h
Name	Sample Value #7
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	Sample Value #8
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Jeder Subindex (1-8) beschreibt jeweils ein gemapptes Objekt.

Ein Mapping Eintrag besteht aus vier 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.

2203h Sampler Buffer Information

Funktion

Dieses Objekt stellt erweiterte Informationen zum Sampler bereit.

Objektbeschreibung

Index	2203 _h
Objektname	Sampler Buffer Information
Object Code	ARRAY

Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	03 _h
Subindex	01 _h
Name	Sample Buffer Size
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Sample Buffer Watermark
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	Sample Tick
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h beschreibt die maximale Größe des Sampler-Puffers in Bytes.
- 02_h enthält den momentanen Füllstand des Sampler-Puffers in Bytes.
- 03_h hält einen Zähler, der bei jeder Abtastung um eins erhöht wird.

2204h Sample Time In Ms

Funktion

Dieses Objekt enthält das Abtastintervall in Millisekunden des Samplers.

Objektbeschreibung

Index	2204 _h
Objektname	Sample Time In Ms
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h
Firmware Version	
Änderungshistorie	

2300h VMM Control

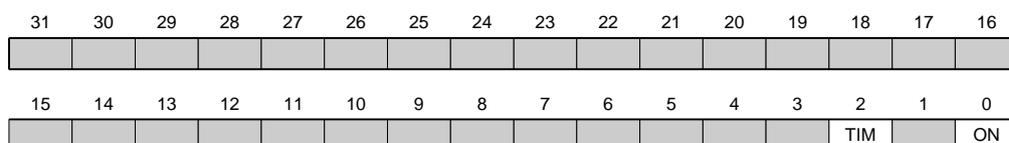
Funktion

Steuert die Ausführung eines Benutzerprogramms.

Objektbeschreibung

Index	2300 _h
Objektname	VMM Control
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung



ON

Schaltet die VMM ein (Wert = "1") oder aus (Wert = "0").

Bei einer steigenden Flanke in Bit 0 wird das Programm zuvor neu geladen und der Variablenbereich zurückgesetzt.

TIM

Schaltet die Zeitüberwachung ab (Wert = "1") oder an (Wert = "0").

2301h VMM Status

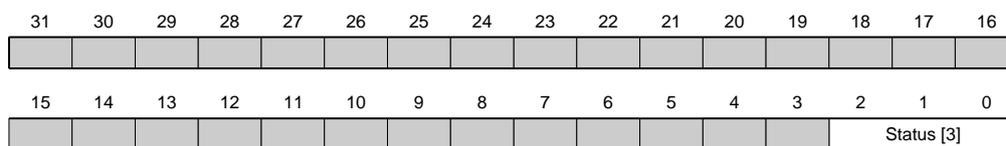
Funktion

Zeigt den Betriebszustand des Benutzerprogramms an.

Objektbeschreibung

Index	2301 _h
Objektname	VMM Status
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung



Status [3]

Beschreibt den aktuellen Status der VMM.

- Wert = "0": Programm ist angehalten
- Wert = "1": Programm läuft gerade
- Wert = "4": Programm wurde mit Fehler beendet. Fehlerursache kann in Objekt **2302_h** ausgelesen werden.

2302h VMM Error Code

Funktion

Zeigt an, welcher Fehler bei der Ausführung des Benutzerprogramms aufgetreten ist.

Objektbeschreibung

Index	2302 _h
-------	-------------------

Objektname	VMM Error Code
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Bechreibung

Fehlercodes bei Programmausführung:

Nummer	Beschreibung
0x0000	Kein Fehler
0x0001	Ungültiger Service Call (Cortex Svc)
0x0002	Speicherschutz-Verletzung (Cortex MPU Fault)
0x0003	Invalid Usage (Cortex Fehler z. B. durch einen im Usermode nicht zulässigen Assembler Befehl)
0x0004	Hardfault (Cortex Fehler)
0x0005	Timeout, Zeitüberschreitung des 1 ms Zyklus
0x0006	Busfault (Cortex Fehler)
0x0007	Invalid SP, Ungültiger Stackpointer
0x0100	Bad File, Ungültige Programmdatei

Dateisystem Fehlercodes beim Laden des Benutzerprogramms:

Nummer	Beschreibung
0x10001	Harter Fehler im Disktreiber
0x10002	Interner Dateisystemfehler
0x10003	Speichermedium nicht bereit
0x10004	Datei nicht gefunden
0x10005	Ordner nicht gefunden
0x10006	Ungültiger Dateiname/Ordnername
0x10008	Zugriff auf Datei nicht möglich
0x10009	Datei/Verzeichnis Objekt ist ungültig
0x1000A	Speicherrmedium ist schreibgeschützt
0x1000B	Laufwerksnummer ist ungültig
0x1000C	Arbeitsbereich des Laufwerks ist ungültig
0x1000D	Kein gültiges Dateisystem auf dem Laufwerk
0x1000E	Erstellung des Dateisystems ist fehlgeschlagen

Nummer	Beschreibung
0x1000F	Zugriff innerhalb der geforderten Zeit nicht möglich
0x10010	Zugriff wurde zurückgewiesen

2303h Number Of Active User Program

Funktion

Wählt eines von vier möglichen Benutzerprogrammen aus, deren Dateinamen zuvor in Objekt **2304_h** hinterlegt worden sind.

Objektbeschreibung

Index	2303 _h
Objektname	Number Of Active User Program
Object Code	VARIABLE
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Eine Änderung des Eintrags, während ein Benutzerprogramm ausgeführt wird, führt zu folgendem Ablauf:

- Das aktuelle Programm wird gestoppt.
- Das neu angewählte Programm wird geladen.
- Das neu geladene Programm wird gestartet.

2304h Table Of Available User Programs

Funktion

Hier werden die Dateinamen der verfügbaren Benutzerprogramme hinterlegt.

Objektbeschreibung

Index	2304 _h
Objektname	Table Of Available User Programs
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	08 _h
Subindex	01 _h
Name	Name Of User Program 1 UB
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Name Of User Program 1 LB
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	Name Of User Program 2 UB
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	Name Of User Program 2 LB
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	05 _h
Name	Name Of User Program 3 UB
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06 _h
Name	Name Of User Program 3 LB
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07 _h
Name	Name Of User Program 4 UB
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08 _h
Name	Name Of User Program 4 LB
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

In jeweils zwei aufeinander folgenden Subindices steht der Name eines Benutzerprogramms im ASCII-Zeichensatz kodiert.

Programm 1: Subindex 1 und 2

Programm 2: Subindex 3 und 4

Programm 3: Subindex 5 und 6

Programm 4: Subindex 7 und 8

Beispiel: So wird das Programm 1 mit der Bezeichnung " test . usr" wie folgt kodiert:

t = 74_h

e = 65_h

s = 73_h

Damit ergeben sich die beiden Einträge an Subindex 1 und 2 zu:

74657374_h, 00000000_h

In jeweils zwei aufeinander folgenden Subindices steht der Name eines Benutzerprogramms im ASCII-Zeichensatz kodiert. Der Subindex mit der Bezeichnung UB (Upper Byte) enthält dabei die ersten vier Buchstaben des Namens, der Subindex mit LB (Lower Byte) die letzten vier Buchstaben. Sollte der Name weniger als acht Buchstaben haben, müssen die fehlenden Buchstaben mit Nullen aufgefüllt werden.

2310h VMM Input Data Selection

Funktion

Beschreibt die Object Dictionary-Einträge, die in das Input PDO-Mapping des VMM-Programms kopiert werden.

Objektbeschreibung

Index	2310 _h
Objektname	VMM Input Data Selection
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 17

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h -10 _h
Name	Mapping #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02
Name	Mapping #2

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03
Name	Mapping #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04
Name	Mapping #4
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05
Name	Mapping #5
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06
Name	Mapping #6
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07
Name	Mapping #7
Datentyp	UNSIGNED32

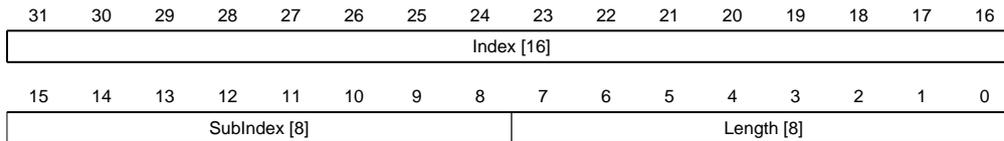
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08
Name	Mapping #8
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	09
Name	Mapping #9
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0A
Name	Mapping #10
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0B
Name	Mapping #11
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0C
Name	Mapping #12
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0D
Name	Mapping #13
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0E
Name	Mapping #14
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0F
Name	Mapping #15
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	10
Name	Mapping #16
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

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

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

2320h VMM Output Data Selection

Funktion

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

Objektbeschreibung

Index	2320 _h
Objektname	VMM Output Data Selection
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 17

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h -10 _h
Name	Mapping #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02
Name	Mapping #2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03
Name	Mapping #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04
Name	Mapping #4
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05
Name	Mapping #5
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06
Name	Mapping #6
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07

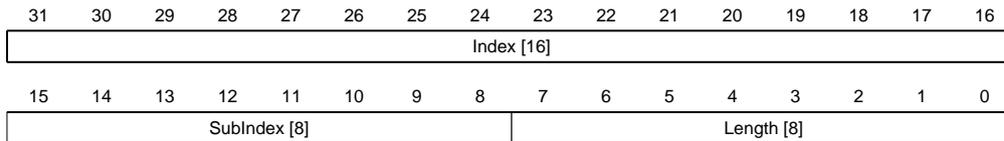
Name	Mapping #7
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08
Name	Mapping #8
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	09
Name	Mapping #9
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0A
Name	Mapping #10
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0B
Name	Mapping #11
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0C
Name	Mapping #12

Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0D
Name	Mapping #13
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0E
Name	Mapping #14
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0F
Name	Mapping #15
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	10
Name	Mapping #16
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

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

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

2330h VMM In/output Data Selection

Funktion

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

Objektbeschreibung

Index	2330 _h
Objektname	VMM In/output Data Selection
Object Code	RECORD
Datentyp	PDO_MAPPING
Firmware Version	
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 17

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
<hr/>	
Subindex	01 _h -10 _h
Name	Mapping #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	02
Name	Mapping #2
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03
Name	Mapping #3
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04
Name	Mapping #4
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05
Name	Mapping #5
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06
Name	Mapping #6
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

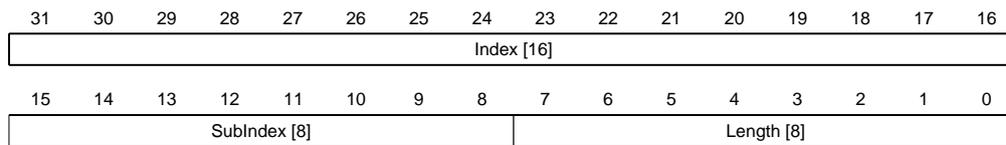
Subindex	07
Name	Mapping #7
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08
Name	Mapping #8
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	09
Name	Mapping #9
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0A
Name	Mapping #10
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0B
Name	Mapping #11
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0C

Name	Mapping #12
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0D
Name	Mapping #13
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0E
Name	Mapping #14
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0F
Name	Mapping #15
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	10
Name	Mapping #16
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

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

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



Index [16]

Darin ist der Index des zu mappenden Objektes enthalten

SubIndex [8]

Darin ist der Subindex des zu mappenden Objektes enthalten

Length [8]

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

2400h VMM Inputs

Funktion

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

Objektbeschreibung

Index	2400 _h
Objektnamen	VMM Inputs
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 33

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	20 _h
Subindex	01 _h -20 _h
Name	VMM Input 1#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	

Vorgabewert	00000000 _h
Subindex	02
Name	VMM Input 2#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03
Name	VMM Input 3#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04
Name	VMM Input 4#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05
Name	VMM Input 5#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06
Name	VMM Input 6#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	07
Name	VMM Input 7#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	08
Name	VMM Input 8#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	09
Name	VMM Input 9#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0A
Name	VMM Input 10#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0B
Name	VMM Input 11#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0C

Name	VMM Input 12#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	0D
Name	VMM Input 13#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	0E
Name	VMM Input 14#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	0F
Name	VMM Input 15#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	10
Name	VMM Input 16#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	11
Name	VMM Input 17#

Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	12
Name	VMM Input 18#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	13
Name	VMM Input 19#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	14
Name	VMM Input 20#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	15
Name	VMM Input 21#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	16
Name	VMM Input 22#
Datentyp	INTEGER32

Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	17
Name	VMM Input 23#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	18
Name	VMM Input 24#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	19
Name	VMM Input 25#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1A
Name	VMM Input 26#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1B
Name	VMM Input 27#
Datentyp	INTEGER32
Zugriff	lesen/schreiben

PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1C
Name	VMM Input 28#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1D
Name	VMM Input 29#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1E
Name	VMM Input 30#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1F
Name	VMM Input 31#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	20
Name	VMM Input 32#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Hier können dem VMM-Programm z. B. Vorgabewerte übergeben werden.

2500h VMM Outputs

Funktion

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

Objektbeschreibung

Index	2500 _h
Objektname	VMM Outputs
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 33

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	20 _h

Subindex	01 _h -20 _h
Name	VMM Output 1#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02
Name	VMM Output 2#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO

Zulässige Werte

Vorgabewert 00000000_h

Subindex 03
Name VMM Output 3#
Datentyp INTEGER32
Zugriff lesen/schreiben
PDO Mapping TX-PDO

Zulässige Werte

Vorgabewert 00000000_h

Subindex 04
Name VMM Output 4#
Datentyp INTEGER32
Zugriff lesen/schreiben
PDO Mapping TX-PDO

Zulässige Werte

Vorgabewert 00000000_h

Subindex 05
Name VMM Output 5#
Datentyp INTEGER32
Zugriff lesen/schreiben
PDO Mapping TX-PDO

Zulässige Werte

Vorgabewert 00000000_h

Subindex 06
Name VMM Output 6#
Datentyp INTEGER32
Zugriff lesen/schreiben
PDO Mapping TX-PDO

Zulässige Werte

Vorgabewert 00000000_h

Subindex 07
Name VMM Output 7#
Datentyp INTEGER32
Zugriff lesen/schreiben
PDO Mapping TX-PDO

Zulässige Werte

Vorgabewert	00000000 _h
Subindex	08
Name	VMM Output 8#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	09
Name	VMM Output 9#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0A
Name	VMM Output 10#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0B
Name	VMM Output 11#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0C
Name	VMM Output 12#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	0D
Name	VMM Output 13#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0E
Name	VMM Output 14#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	0F
Name	VMM Output 15#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	10
Name	VMM Output 16#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	11
Name	VMM Output 17#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	12

Name	VMM Output 18#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	13
Name	VMM Output 19#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	14
Name	VMM Output 20#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	15
Name	VMM Output 21#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	16
Name	VMM Output 22#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	17
Name	VMM Output 23#

Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	18
Name	VMM Output 24#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	19
Name	VMM Output 25#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	1A
Name	VMM Output 26#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	1B
Name	VMM Output 27#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	1C
Name	VMM Output 28#
Datentyp	INTEGER32

Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1D
Name	VMM Output 29#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1E
Name	VMM Output 30#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	1F
Name	VMM Output 31#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	20
Name	VMM Output 32#
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Hier kann das VMM-Programm Ergebnisse ablegen, die dann über den Feldbus ausgelesen werden können.

2600h VMM Debug Output

Funktion

Dieses Objekt enthält Debug-Ausgaben eines Benutzerprogramms.

Objektbeschreibung

Index	2600 _h
Objektname	VMM Debug Output
Object Code	ARRAY
Datentyp	UNSIGNED8
Firmware Version	
Änderungshistorie	Die Anzahl der Einträge haben sich geändert von 2 auf 65

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01-40 _h
Name	Value #1
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	02
Name	Value #2
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	03
Name	Value #3
Datentyp	UNSIGNED8

Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	04
Name	Value #4
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	05
Name	Value #5
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	06
Name	Value #6
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	07
Name	Value #7
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	08
Name	Value #8
Datentyp	UNSIGNED8
Zugriff	nur lesen

PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	09
Name	Value #9
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	0A
Name	Value #10
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	0B
Name	Value #11
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	0C
Name	Value #12
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	0D
Name	Value #13
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein

Zulässige Werte	
Vorgabewert	00 _h
Subindex	0E
Name	Value #14
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	0F
Name	Value #15
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	10
Name	Value #16
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	11
Name	Value #17
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	12
Name	Value #18
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	

Vorgabewert	00 _h
Subindex	13
Name	Value #19
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	14
Name	Value #20
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	15
Name	Value #21
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	16
Name	Value #22
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	17
Name	Value #23
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	18
Name	Value #24
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	19
Name	Value #25
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	1A
Name	Value #26
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	1B
Name	Value #27
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	1C
Name	Value #28
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	1D

Name	Value #29
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	1E
Name	Value #30
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	1F
Name	Value #31
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	20
Name	Value #32
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	21
Name	Value #33
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	22
Name	Value #34

Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	23
Name	Value #35
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	24
Name	Value #36
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	25
Name	Value #37
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	26
Name	Value #38
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	27
Name	Value #39
Datentyp	UNSIGNED8

Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	28
Name	Value #40
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	29
Name	Value #41
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	2A
Name	Value #42
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	2B
Name	Value #43
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	2C
Name	Value #44
Datentyp	UNSIGNED8
Zugriff	nur lesen

PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	2D
Name	Value #45
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	2E
Name	Value #46
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	2F
Name	Value #47
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	30
Name	Value #48
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	31
Name	Value #49
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein

Zulässige Werte	
Vorgabewert	00 _h
Subindex	32
Name	Value #50
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	33
Name	Value #51
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	34
Name	Value #52
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	35
Name	Value #53
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	36
Name	Value #54
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	

Vorgabewert	00 _h
Subindex	37
Name	Value #55
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	38
Name	Value #56
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	39
Name	Value #57
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	3A
Name	Value #58
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	3B
Name	Value #59
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	3C
Name	Value #60
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	3D
Name	Value #61
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	3E
Name	Value #62
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	3F
Name	Value #63
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	40
Name	Value #64
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Hier legt das VMM-Programm die Debug-Ausgaben ab, welche mit der Funktion `VmmDebugOutputString()`, `VmmDebugOutputInt()` und dergleichen aufgerufen wurden. Eine genaue Beschreibung der Debug-Ausgabe kann im Unterkapitel **Debug-Ausgabe** des Kapitels **Programmierung mit NanoJ** nachgelesen werden.

2700h User Storage Area

Funktion

DESCRIPTION!

Objektbeschreibung

Index	2700 _h
Objektname	User Storage Area
Object Code	RECORD
Datentyp	USER_STORAGE_AREA
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Storage Control Word
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	01 _h
Name	Storage 1#
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Subindex	02 _h
Name	Storage 2#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	

Vorgabewert	0000 _h
Subindex	03 _h
Name	Storage 3#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	04 _h
Name	Storage 4#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	05 _h
Name	Storage 5#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	06 _h
Name	Storage 6#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	07 _h
Name	Storage 7#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	08 _h
Name	Storage 8#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	09 _h
Name	Storage 9#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	0A _h
Name	Storage 10#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	0B _h
Name	Storage 11#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	0C _h
Name	Storage 12#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
Subindex	0D _h

Name	Storage 13#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	0E _h
Name	Storage 14#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	0F _h
Name	Storage 15#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	10 _h
Name	Storage 16#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	11 _h
Name	Storage 17#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	12 _h
Name	Storage 18#

Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	13 _h
Name	Storage 19#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	14 _h
Name	Storage 20#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h
<hr/>	
Subindex	15 _h
Name	Storage 21#
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0000 _h

Beschreibung

2800h Bootloader Setting

Funktion

DESCRIPTION!

Objektbeschreibung

Index	2800 _h
Objektname	Bootloader Setting
Object Code	ARRAY
Datentyp	UNSIGNED32

Firmware Version
Änderungshistorie

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Subindex	01 _h
Name	Bootloader Password
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

3202h Motor Drive Submode Select

Funktion

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

Objektbeschreibung

Index	3202 _h
Objektname	Motor Drive Submode Select
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								Ferr	BLDC	Torque		CurRed	Brake	VoS	CL/OL

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

Brake

Wert = "1": Einschalten der Bremsensteuerung

CurRed (Current Reduction)

Wert = "1": Stromabsenkung im Open Loop aktiviert

Torque

nur im **Profile Torque Mode** aktiv

Wert = "1": M-Regler ist aktiv, andernfalls ist ein V-Regler überlagert

BLDC

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

Ferr (Following Error)

Wert = "1": ein "Following Error" löst einen Fault mit zugehöriger Reaktion aus (siehe Objekt **605E_h**)

3208h Motor Drive Sensor Source Open Loop

Funktion

DESCRIPTION!

Objektbeschreibung

Index	3208 _h
Objektname	Motor Drive Sensor Source Open Loop
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

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

Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h
Subindex	01 _h
Name	Commutation
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Torque
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	Velocity
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	Position
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

3209h Motor Drive Sensor Source Closed Loop

Funktion

DESCRIPTION!

Objektbeschreibung

Index	3209 _h
Objektname	Motor Drive Sensor Source Closed Loop
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h

Subindex	01 _h
Name	Commutation
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Torque
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	Velocity
Datentyp	INTEGER32

Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	Position
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

320Ah Motor Drive Sensor Display Open Loop

Funktion

Damit kann die Quelle für die Objekte **6044_h** und **6064_h** im Modus "Open Loop" geändert werden.

Objektbeschreibung

Index	320A _h
Objektname	Motor Drive Sensor Display Open Loop
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h
Subindex	01 _h
Name	Commutation
Datentyp	INTEGER32
Zugriff	lesen/schreiben

PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	02 _h
Name	Torque
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
<hr/>	
Subindex	03 _h
Name	Velocity
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	FFFFFFFF _h
<hr/>	
Subindex	04 _h
Name	Position
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	FFFFFFFF _h

Beschreibung

Folgende Subindizes haben eine Bedeutung:

- 01_h: Ungenutzt
- 02_h: Ungenutzt
- 03_h: Verändert die Quelle des Objekts **6044_h**:
 - Wert = "-1": der intern berechnete Wert wird in das Objekt **6044_h** eingetragen
 - Wert = "0": der Wert wird auf 0 gehalten
 - Wert = "1": der Encoder-Wert wird in das Objekt **6044_h** eingetragen
- 04_h: Verändert die Quelle des **6064_h**:
 - Wert = "-1": der intern berechnete Wert wird in das Objekt **6064_h** eingetragen
 - Wert = "0": der Wert wird auf 0 gehalten
 - Wert = "1": der Encoder-Wert wird in das Objekt **6064_h** eingetragen

320Bh Motor Drive Sensor Display Closed Loop

Funktion

Damit kann die Quelle für die Objekte **6044_h** und **6064_h** im Modus "Closed Loop" geändert werden.

Objektbeschreibung

Index	320B _h
Objektname	Motor Drive Sensor Display Closed Loop
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h

Subindex	01 _h
Name	Commutation
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	02 _h
Name	Torque
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Subindex	03 _h
Name	Velocity
Datentyp	INTEGER32

Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	04 _h
Name	Position
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Folgende Subindizes haben eine Bedeutung:

- 01_h: Ungenutzt
- 02_h: Ungenutzt
- 03_h: Verändert die Quelle des Objekts **6044_h**:
 - Wert = "-1": der intern berechnete Wert wird in das Objekt **6044_h** eingetragen
 - Wert = "0": der Wert wird auf 0 gehalten
 - Wert = "1": der Encoder-Wert wird in das Objekt **6044_h** eingetragen
- 04_h: Verändert die Quelle des Objekts **6064_h**:
 - Wert = "-1": der intern berechnete Wert wird in das Objekt **6064_h** eingetragen
 - Wert = "0": der Wert wird auf 0 gehalten
 - Wert = "1": der Encoder-Wert wird in das Objekt **6064_h** eingetragen

3210h Motor Drive Parameter Set

Funktion

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

Objektbeschreibung

Index	3210 _h
Objektname	Motor Drive Parameter Set
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8

Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0A _h
Subindex	01 _h
Name	S_P
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000800 _h
Subindex	02 _h
Name	S_I
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	V_P
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00001B58 _h
Subindex	04 _h
Name	V_I
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000004 _h
Subindex	05 _h
Name	Id_P
Datentyp	INTEGER32
Zugriff	lesen/schreiben

PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000668A0 _h

Subindex	06 _h
Name	Id_I
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00002EE0 _h

Subindex	07 _h
Name	Iq_P
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000668A0 _h

Subindex	08 _h
Name	Iq_I
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00002EE0 _h

Subindex	09
Name	I_P
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00027100 _h

Subindex	0A
Name	I_I
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein

Zulässige Werte

Vorgabewert 000055F0_h

Beschreibung

- Subindex 00_h: Anzahl der Einträge
- Subindex 01_h: Proportionalanteil des S-Reglers
- Subindex 02_h: Integralanteil des S-Reglers
- Subindex 03_h: Proportionalanteil des V-Reglers
- Subindex 04_h: Integralanteil des V-Reglers
- Subindex 05_h: (Closed Loop) Proportionalanteil des Stromreglers der feldbildenden Komponente
- Subindex 06_h: (Closed Loop) Integralanteil des Stromreglers der feldbildenden Komponente
- Subindex 07_h: (Closed Loop) Proportionalanteil des Stromreglers der momentbildenden Komponente
- Subindex 08_h: (Closed Loop) Integralanteil des Stromreglers der momentbildenden Komponente
- Subindex 09_h: (Open Loop) Proportionalanteil des Stromreglers der momentbildenden Komponente
- Subindex 0A_h: (Open Loop) Integralanteil des Stromreglers der momentbildenden Komponente

3220h Analog Inputs

Funktion

Zeigt die Momentanwerte der Analogeingänge in [digits] an.

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

Objektbeschreibung

Index	3220 _h
Objektname	Analog Inputs
Object Code	ARRAY
Datentyp	INTEGER16
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	Analogue Input 1
Datentyp	INTEGER16
Zugriff	nur lesen

PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Subindex	02 _h
Name	Analogue Input 2
Datentyp	INTEGER16
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h

Beschreibung

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

- Spannungseingang: $(x \text{ digits} - 512 \text{ digits}) * 20 \text{ V} / 1024 \text{ digits}$
- Stromeingang: $x \text{ digits} * 20 \text{ mA} / 1024 \text{ digits}$

3221h Analogue Inputs Control

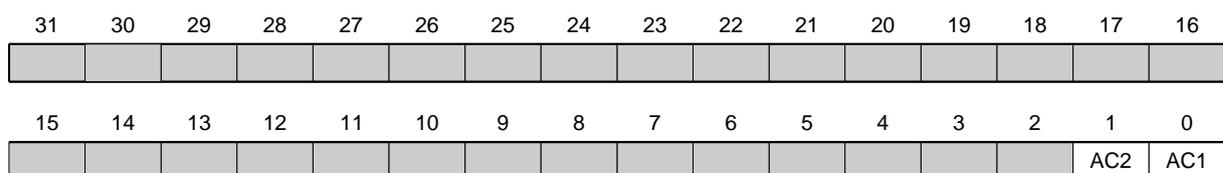
Funktion

Damit lässt sich ein Analog-Eingang von Spannungsmessung auf Strommessung

Objektbeschreibung

Index	3221 _h
Objektname	Analogue Inputs Control
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
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

3240h Digital Inputs Control

Funktion

Mit diesem Objekt lassen sich digitale Eingänge manipulieren wie in Kapitel **Digitale Ein- und Ausgänge** beschrieben. Dabei gilt für alle folgenden Subindizes, dass Bit 0 den digitalen Eingang 1 betrifft, Bit 1 den Eingang 2, usw.

Objektbeschreibung

Index	3240 _h
Objektname	Digital Inputs Control
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	Version 1.0.3: Subindex 01 _h : Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable"

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	07 _h
Subindex	01 _h
Name	Special Function Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	06
Name	Input Range Select
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	07
Name	Differential Select
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subentries haben folgende Funktion:

- 01_h: Mit diesem Subindex werden die Spezialfunktionen der jeweiligen Eingänge eingeschaltet wenn das Bit den Wert "1" hat.
- 02_h: Mit diesem Subindex wird die Logik eines Eingangs invertiert wenn das Bit des jeweiligen Eingangs den Wert "1" hat.
- 03_h: Mit diesem Subindex wird ein Eingangswert erzwungen, wenn das Bit den Wert "1" hat. Ein Eingang, dessen Wert erzwungen wird, ist damit unabhängig vom angelegten Spannungspegel immer auf dem Wert, welcher im Subindex 4_h eingetragen ist.
- 04_h: Mit diesem Subindex wird der zu erzwingende Eingangswert festgelegt.
- 05_h: Dieser Subindex enthält immer den gelesenen, unmodifizierten Eingangswert.

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 folgenden Subindizes, dass Bit 0 den Digitalausgang 1 betrifft, Bit 1 den Ausgang 2, usw.

Objektbeschreibung

Index	3250 _h
Objektname	Digital Outputs Control
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	Version 1.0.3: Subindex 01 _h : Eintrag "Name" geändert von "Special Function Disable" auf "Special Function Enable"

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	05 _h
Subindex	01 _h
Name	Special Function Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben

PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000F0001 _h
Subindex	02 _h
Name	Function Inverted
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	03 _h
Name	Force Enable
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	04 _h
Name	Force Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	05 _h
Name	Raw Value
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die Subentries haben folgende Funktion:

- 01_h: Ohne Funktion.
- 02_h: Mit diesem Subindex wird die Logik invertiert (von Öffner-Logik auf Schließer-Logik)
- 03_h: Mit diesem Subindex wird der Ausgangswert erzwungen wenn das Bit den Wert "1" hat. Der Pegel des Ausganges wird in Subindex 4_h festgelegt.

- 04_h: Mit diesem Subindex wird der am Ausgang anzulegende Pegel definiert. Der Wert "0" liefert am digitalen Ausgang einen logischen Low-Pegel, der Wert "1" entsprechend einen logischen High-Pegel.
- 05_h: In diesem dem Subindex wird die an die Ausgänge gelegte Bitkombination abgelegt.

3700h Following Error Option Code

Funktion

Das Objekt enthält die auszuführende Aktion, wenn ein zu Schleppfehler ausgelöst wird.

Objektbeschreibung

Index	3700 _h
Objektname	Following Error Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	FFFF _h
Firmware Version	
Änderungshistorie	

Beschreibung

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

603Fh Error Code

Funktion

Enthält den letzten aufgetretenen Fehler.

Objektbeschreibung

Index	603F _h
Objektname	Error Code
Object Code	VARIABLE
Datentyp	UNSIGNED16
Zugriff	nur lesen

PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Bedeutung des Fehlers siehe Objekt **1003_h** (Pre-defined Error Field).

6040h Controlword

Funktion

Mit diesem Objekt wird der Motor eingeschaltet und es können Fahrbefehle ausgeführt werden.

Objektbeschreibung

Index	6040 _h
Objektname	Controlword
Object Code	VARIABLE
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Dieses Objekt steuert die **DS402 Power State machine**. 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 [3], OMS

Bedeutung abhängig vom gewählten Betriebsmodus

FR (Fault Reset)

Setzt einen Fehler zurück (falls möglich)

HALT

Wert = "1": Löst einen Halt aus

6041h Statusword

Funktion

Mit diesem Objekt wird abgefragt, ob der mit dem Objekt **6040_h** (Controlword) kommandierte Zustand erreicht wurde.

Objektbeschreibung

Index	6041 _h
Objektname	Statusword
Object Code	VARIABLE
Datentyp	UNSIGNED16
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Dieses Objekt steuert die **DS402 Power State machine**. 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
CLA		OMS [2]	ILA	TARG	REM		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 (Operational Enabled)

Wert = "1": Steuerung befindet sich in dem Zustand "Operational Enabled"

FAULT

Fehler vorgefallen

VE (Voltage Enabled)

Spannung angelegt

QS (Quick Stop)

Wert = "1": 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

REM (Remote)

Remote (Wert des Bits immer "1")

TARG

Zielvorgabe erreicht

ILA (Internal Limit Reached)

Limit überschritten

OMS (Operation Mode Specific)

Bedeutung abhängig vom gewählten Betriebsmodus

CLA (Closed Loop Available)

Wert = "1": AutoSetup erfolgreich und Closed Loop möglich

6042h VI Target Velocity

Funktion

Gibt die Zielgeschwindigkeit in Benutzereinheiten an.

Objektbeschreibung

Index	6042 _h
Objektname	VI Target Velocity
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00C8 _h
Firmware Version	
Änderungshistorie	

6043h VI Velocity Demand

Funktion

Gibt die aktuelle Zielgeschwindigkeit in Benutzereinheiten an.

Objektbeschreibung

Index	6043 _h
Objektname	VI Velocity Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

6044h VI Velocity Actual Value

Funktion

Gibt die aktuelle Istgeschwindigkeit in Benutzereinheiten an.

Die Quelle dieses Objekts kann im Open Loop-Modus mit dem Objekt **320A_h:03_h** entweder auf den internen, berechneten Wert oder auf den Encoder gestellt werden.

Die Quelle dieses Objekts kann im Closed Loop-Modus mit dem Objekt **320B_h:03_h** entweder auf den internen, berechneten Wert oder auf den Encoder gestellt werden.

Objektbeschreibung

Index	6044 _h
Objektname	VI Velocity Actual Value
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

6046h VI Velocity Min Max Amount

Funktion

Mit diesem Objekt können Minimalgeschwindigkeit und Maximalgeschwindigkeit in Benutzereinheiten eingestellt werden.

Objektbeschreibung

Index	6046 _h
Objektname	VI Velocity Min Max Amount

Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	MinAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	MaxAmount
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00004E20 _h

Beschreibung

Subindex 1 enthält die Minimalgeschwindigkeit.

Subindex 2 enthält die Maximalgeschwindigkeit.

Hinweis

Wird eine Zielgeschwindigkeit (Objekt **6042_h**) vom Betrag her kleiner als die Minimalgeschwindigkeit angegeben, gilt die Minimalgeschwindigkeit. Ist die Zielgeschwindigkeit 0, hält der Motor an.

Eine Zielgeschwindigkeit größer als die Maximalgeschwindigkeit setzt die Geschwindigkeit auf die Maximalgeschwindigkeit und setzt das Bit 11 "Limit überschritten" im Objekt **6041_h** (Statusword).

6048h VI Velocity Acceleration

Funktion

Setzt die Beschleunigungsrampe im Velocity Mode (**siehe "Velocity"**).

Objektbeschreibung

Index	6048 _h
Objektname	VI Velocity Acceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h

Subindex	02 _h
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 _h

Beschreibung

Die Beschleunigung wird als Bruch angegeben:
Geschwindigkeitsänderung pro Zeitänderung.

Subindex 01_h: enthält die Geschwindigkeitsänderung in Schritten pro Sekunde (U32).

Subindex 02_h: enthält die Zeitänderung in Sekunden (U16).

6049h VI Velocity Deceleration

Funktion

Setzt die Bremsrampe im Velocity Mode (**siehe "Velocity"**).

Objektbeschreibung

Index	6049 _h
Objektname	VI Velocity Deceleration
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h

Subindex	02 _h
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 _h

604Ah VI Velocity Quick Stop

Funktion

Dieses Objekt definiert die Bremsbeschleunigung, wenn im Velocity Mode der Quick Stop-Zustand eingeleitet wird.

Objektbeschreibung

Index	604A _h
Objektname	VI Velocity Quick Stop
Object Code	RECORD
Datentyp	VELOCITY_ACCELERATION_DECELERATION
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	DeltaSpeed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 _h

Subindex	02 _h
Name	DeltaTime
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0001 _h

Beschreibung

Der Subindex 1 enthält dabei die Geschwindigkeitsänderung, der Subindex 2 die zugehörige Zeit in Sekunden.

Beides zusammen wird als Beschleunigung verrechnet:

Velocity Quick Stop = DeltaSpeed (604A_h:01_h) / DeltaTime (604A_h:02_h)

604Ch VI Dimension Factor

Funktion

Hier wird die Einheit der Geschwindigkeitsangaben für die Objekte festgelegt, welche den Velocity Mode betreffen.

Objektbeschreibung

Index	604C _h
Objektname	VI Dimension Factor
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	VI Dimension Factor Numerator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	02 _h
Name	VI Dimension Factor Denominator
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000003C _h

Beschreibung

Werden die Subindizes 1 und 2 auf den Wert 1 eingestellt, erfolgt die Geschwindigkeitsangabe in Umdrehungen pro Minute.

Sonst enthält der Subindex 1 den Nenner (Multiplikator) und der Subindex 2 den Zähler (Divisor), mit dem Geschwindigkeitsangaben verrechnet werden.

Das Ergebnis wird als Umdrehungen pro Sekunde interpretiert, wobei über **2060_h** ausgewählt wird, ob es sich um elektrische (**2060_h = 0**) oder mechanische (**2060_h = 1**) Umdrehungen pro Sekunde handelt.

605Ah Quick Stop Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der **DS402 Power State machine** in den Quick Stop-Zustand.

Objektbeschreibung

Index	605A _h
Objektname	Quick Stop Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Wert	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt mit Kurzschlussbremsung
1	Abbremsen mit "slow down ramp" (Bremsbeschleunigung je nach Betriebsart) und anschließendem Zustandswechsel zu "Switch on disabled"
2	Abbremsen mit "quick stop ramp" und anschließendem Zustandswechsel zu "Switch on disabled"
3 bis 32767	Reserviert

605Bh Shutdown Option Code

Funktion

Das Objekt enthält die auszuführende Aktion bei einem Übergang der **DS402 Power State machine** vom Zustand "Operation enabled" in den Zustand "Ready to switch on".

Objektbeschreibung

Index	605B _h
Objektname	Shutdown Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Wert	Beschreibung
-32768 bis -1	Reserviert
0	Soforthalt mit Kurzschlussbremsung
1	Abbremsen mit "slow down ramp" (Bremsbeschleunigung 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 **DS402 Power State machine** vom Zustand "Operation enabled" in den Zustand "Switched on".

Objektbeschreibung

Index	605C _h
Objektname	Disable Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Wert	Beschreibung
-32768 bis -1	Reserviert

Wert	Beschreibung
0	Soforthalt mit Kurzschlussbremsung
1	Abbremsen mit "slow down ramp" (Bremsbeschleunigung 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_h** das Halt-Bit 8 gesetzt wird.

Objektbeschreibung

Index	605D _h
Objektname	Halt Option Code
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0001 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Wert	Beschreibung
-32768 bis 0	Reserviert
1	Abbremsen mit "slow down ramp" (Bremsbeschleunigung je nach Betriebsart)
2	Abbremsen mit "quick stop ramp" (Bremsbeschleunigung 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 _h
Objektname	Fault Option Code
Object Code	VARIABLE

Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	0002 _h
Firmware Version	
Änderungshistorie	

Beschreibung

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

6060h Modes Of Operation

Funktion

In dieses Objekt wird der gewünschte Betriebsmodus eingetragen.

Objektbeschreibung

Index	6060 _h
Objektname	Modes Of Operation
Object Code	VARIABLE
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Modus	Beschreibung
-128 to -2	Manufacturer-specific operation modes
-1	Takt/Richtungsmodus
0	No mode change/no mode assigned
1	Profile Position Mode

Modus	Beschreibung
2	Velocity Mode
3	Profile Velocity Mode
4	Profile Torque Mode
5	Reserved
6	Homing Mode
7	nicht belegt
8	Cyclic Synchronous Position Mode
9	Cyclic Synchronous Velocity Mode
10	Cyclic Sync Torque Mode
11 bis 127	Reserved

6061h Modes Of Operation Display

Funktion

Enthält den aktuellen Betriebsmodus, der in Objekt **6060_h** ("Modes Of Operation") eingestellt ist.

Objektbeschreibung

Index	6061 _h
Objektnamen	Modes Of Operation Display
Object Code	VARIABLE
Datentyp	INTEGER8
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	
Änderungshistorie	

6062h Position Demand Value

Funktion

Gibt die aktuelle Sollposition in Benutzereinheiten an.

Objektbeschreibung

Index	6062 _h
Objektnamen	Position Demand Value
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	TX-PDO

Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

6063h Position Actual Internal Value

Funktion

Enthält die aktuelle Drehgeberposition in Inkrementen seit Einschalten des Antriebs.

Objektbeschreibung

Index	6063 _h
Objektname	Position Actual Internal Value
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

6064h Position Actual Value

Funktion

Enthält die aktuelle Istposition (Drehgeberposition umgerechnet laut Feed Constant (**6092**) und Gear Ratio (**6091**, sowie Referenzposition)

Die Quelle dieses Objekts kann im Open Loop-Modus mit dem Objekt **320A_h:04_h** entweder auf den internen, berechneten Wert oder auf den Encoder gestellt werden.

Die Quelle dieses Objekts kann im Closed Loop-Modus mit dem Objekt **320B_h:04_h** entweder auf den internen, berechneten Wert oder auf den Encoder gestellt werden.

Objektbeschreibung

Index	6064 _h
Objektname	Position Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	

Änderungshistorie

6065h Following Error Window

Funktion

Gibt den maximalen Schleppfehler symmetrisch zur Sollposition an.

Objektbeschreibung

Index	6065 _h
Objektname	Following Error Window
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000100 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Weicht die Istposition von der Sollposition so stark ab, dass der Wert dieses Objekts überschritten wird, wird das Bit 11 für "Limit überschritten" im Objekt **6041_h** (Statusword) gesetzt. Die Abweichung muss länger als die Zeit im Objekt **6066_h** anhalten.

Um eine automatische Reaktion auf den Fehler zu erhalten, muss das Bit 7 im Objekt **3202_h** aktiviert werden. Damit wird ein "Fault" erzeugt wenn der "Following Error" entsteht - und entsprechend darauf reagiert (**6041_h** Bit 3 "Fehler vorgefallen").

6066h Following Error Time Out

Funktion

Zeit in Millisekunden bis ein zu großer Schleppfehler zu einer Fehlermeldung führt.

Objektbeschreibung

Index	6066 _h
Objektname	Following Error Time Out
Object Code	VARIABLE
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0064 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Weicht die Istposition von der Sollposition so stark ab, dass der Wert des Objekts **6065_h** überschritten wird, wird das Bit 11 für "Limit überschritten" im **6041_h** (Statusword) gesetzt. Die Abweichung muss länger als die Zeit in diesem Objekt anhalten.

Um eine automatische Reaktion auf den Fehler zu erhalten, muss das Bit 7 im Objekt **3202_h** aktiviert werden. Damit wird ein "Fault" erzeugt wenn der "Following Error" entsteht - und entsprechend darauf reagiert (**6041_h** Bit 3 "Fehler vorgefallen").

6067h Position Window

Funktion

Gibt relativ zur Zielposition einen symmetrischen Bereich an, innerhalb dem das Ziel als erreicht gilt.

Objektbeschreibung

Index	6067 _h
Objektname	Position Window
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000000A _h
Firmware Version	
Änderungshistorie	

6068h Position Window Time

Funktion

Die Istposition muss sich für diese Zeit (in Millisekunden) innerhalb des "Position Window" (**6067**) befinden, damit die Zielposition als erreicht gilt.

Objektbeschreibung

Index	6068 _h
Objektname	Position Window Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0064 _h
Firmware Version	
Änderungshistorie	

606Bh Velocity Demand Value

Funktion

Vorgabegeschwindigkeit für den Regler im Profile Velocity Mode.

Dieses Objekt wird mit den benutzerdefinierten Einheiten verrechnet (siehe auch **Benutzerdefinierte Einheiten**). Im Auslieferungszustand ist die Steuerung auf Umdrehungen/Minute eingestellt.

Objektbeschreibung

Index	606B _h
Objektname	Velocity Demand Value
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Dieses Objekt enthält die Ausgabe des Rampengenerators, die gleichzeitig der Vorgabewert für den Geschwindigkeitsregler ist.

606Ch Velocity Actual Value

Funktion

Aktuelle Istgeschwindigkeit im Profile Velocity Mode.

Objektbeschreibung

Index	606C _h
Objektname	Velocity Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

606Dh Velocity Window

Funktion

Geschwindigkeitsfenster für den Profile Velocity Mode.

Objektbeschreibung

Index	606D _h
Objektname	Velocity Window
Object Code	VARIABLE
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Dieser Wert gibt an, wie stark die reale Geschwindigkeit von der Sollgeschwindigkeit abweichen darf, damit das Bit 10 "Zielvorgabe erreicht" im Statusword (**6041_h**) auf "1" gesetzt wird.

606Eh Velocity Window Time

Funktion

Zeitfenster für den Profile Velocity Mode.

Objektbeschreibung

Index	606E _h
Objektname	Velocity Window Time
Object Code	VARIABLE
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Dieses Objekt gibt an, wie lange die reale Geschwindigkeit und die Sollgeschwindigkeit nahe beieinander liegen müssen (siehe **606D_h**), damit Bit 10 "Zielvorgabe erreicht" im Statuswords (**6041_h**) auf "1" gesetzt wird.

6071h Target Torque

Funktion

Dieses Objekt enthält das Zieldrehmoment für den Profile Torque Mode.

Objektbeschreibung

Index	6071 _h
Objektname	Target Torque
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

6072h Max Torque

Funktion

Das Objekt beschreibt das maximale Drehmoment.

Objektbeschreibung

Index	6072 _h
Objektname	Max Torque
Object Code	VARIABLE
Datentyp	UNSIGNED16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

6074h Torque Demand

Funktion

Momentaner Ausgabewert des Rampengenerators (Drehmoment) für den internen Regler.

Objektbeschreibung

Index	6074 _h
-------	-------------------

Objektname	Torque Demand
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

607Ah Target Position

Funktion

Dieses Objekt gibt die Zielposition an.

Objektbeschreibung

Index	607A _h
Objektname	Target Position
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000FA0 _h
Firmware Version	
Änderungshistorie	

607Bh Position Range Limit

Funktion

Enthält die Minimal- und Maximalposition.

Objektbeschreibung

Index	607B _h
Objektname	Position Range Limit
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	Min Position Range Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	80000001 _h
Subindex	02 _h
Name	Max Position Range Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	7FFFFFFE _h

Beschreibung

Wird dieser Bereich über- oder unterschritten, erfolgt ein Überlauf. Um diesen Überlauf zu verhindern, können im Objekt **607D_h** ("Software Position Limit") Grenzwerte für die Zielposition eingestellt werden.

607Ch Home Offset

Funktion

Gibt die Differenz zwischen Null-Position der Applikation und dem Referenzpunkt der Maschine an. Dieses Objekt wird in der gleichen Einheit gerechnet, die bei der Berechnung für Objekt **607A_h** verwendet wird (siehe **Benutzerdefinierte Einheiten**).

Objektbeschreibung

Index	607C _h
Objektname	Home Offset
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

607Dh Software Position Limit

Funktion

Grenzwerte der Zielposition.

Objektbeschreibung

Index	607D _h
Objektname	Software Position Limit
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Min Position Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	80000000 _h

Subindex	02 _h
Name	Max Position Limit
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	

Vorgabewert 7FFFFFFF_h

Beschreibung

Die Zielposition muss innerhalb der hier gesetzten Grenzen liegen. Vor der Überprüfung wird jeweils der Home Offset (**607C_h**) abgezogen:

corrected Min Position Limit = Min Position Limit - Home Offset

corrected Max Position Limit = Max Position Limit - Home Offset.

607Eh Polarity

Funktion

Mit diesem Objekt lässt sich die Drehrichtung umkehren.

Objektbeschreibung

Index	607E _h
Objektname	Polarity
Object Code	VARIABLE
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h
Firmware Version	
Änderungshistorie	

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

6081h Profile Velocity

Funktion

Gibt die maximale Fahrgeschwindigkeit in Umdrehungen pro Sekunde an.

Dieses Objekt wird mit den benutzerdefinierten Einheiten verrechnet (siehe **Benutzerdefinierte Einheiten**). Im Auslieferungszustand ist die Steuerung auf Umdrehungen pro Minute eingestellt.

Objektbeschreibung

Index	6081 _h
Objektname	Profile Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h
Firmware Version	
Änderungshistorie	

6082h End Velocity

Funktion

Gibt die Geschwindigkeit am Ende der gefahrenen Rampe an.

Dieses Objekt wird mit den benutzerdefinierten Einheiten verrechnet (siehe **Benutzerdefinierte Einheiten**). Im Auslieferungszustand ist die Steuerung auf Umdrehungen pro Minute eingestellt.

Objektbeschreibung

Index	6082 _h
Objektname	End Velocity
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

6083h Profile Acceleration

Funktion

Gibt die maximale Beschleunigung in Umdrehungen/s² an.

Objektbeschreibung

Index	6083 _h
Objektname	Profile Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h
Firmware Version	
Änderungshistorie	

6084h Profile Deceleration

Funktion

Gibt die maximale Bremsbeschleunigung in Umdrehungen/s² an.

Objektbeschreibung

Index	6084 _h
Objektname	Profile Deceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	000001F4 _h
Firmware Version	
Änderungshistorie	

6085h Quick Stop Deceleration

Funktion

Gibt die maximale Quick Stop-Bremsbeschleunigung in Umdrehungen/s² an.

Objektbeschreibung

Index	6085 _h
Objektname	Quick Stop Deceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	00001388 _h
Firmware Version	
Änderungshistorie	

6086h Motion Profile Type

Funktion

Gibt den Rampentyp an.

Objektbeschreibung

Index	6086 _h
Objektname	Motion Profile Type
Object Code	VARIABLE
Datentyp	INTEGER16
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	0000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Wert = "0": = Trapez-Rampe

Wert = "3": ruck-begrenzte Rampe

6087h Torque Slope

Funktion

Dieses Objekt enthält die Steigung des Drehmoments im Torque Mode.

Objektbeschreibung

Index	6087 _h
Objektname	Torque Slope
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

608Fh Position Encoder Resolution

Funktion

Encoder-Inkrement pro Umdrehung.

Objektbeschreibung

Index	608F _h
Objektname	Position Encoder Resolution
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Encoder Increments
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000007D0 _h

Subindex	02 _h
Name	Motor Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Position Encoder Resolution = Encoder Increments (608F_h:01_h) / Motor Revolutions (608F_h:02_h)

6091h Gear Ratio

Funktion

Anzahl der Motorumdrehungen pro Umdrehung der Abtriebsachse.

Objektbeschreibung

Index	6091 _h
Objektname	Gear Ratio
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Motor Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	02 _h
Name	Shaft Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Gear Ratio = Motor Revolutions (6091_h:01_h) / Shaft Revolutions (6091_h:02_h)

6092h Feed Constant

Funktion

Vorschub pro Umdrehung im Falle eines Linearantriebs.

Objektbeschreibung

Index	6092 _h
Objektname	Feed Constant
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h

Subindex	01 _h
Name	Feed
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Subindex	02 _h
Name	Shaft Revolutions
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Feed Constant = Feed (6092_h:01_h) / Shaft Revolutions (6092_h:02_h)

6098h Homing Method

Funktion

Dieses Objekt wählt den Homing Mode aus.

Objektbeschreibung

Index	6098 _h
Objektname	Homing Method
Object Code	VARIABLE
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	23 _h
Firmware Version	
Änderungshistorie	

6099h Homing Speed

Funktion

Gibt die Geschwindigkeiten für den Homing Mode (**6098_h**) in Umdrehungen pro Sekunde an.

Dieses Objekt wird mit den benutzerdefinierten Einheiten verrechnet (siehe **Benutzerdefinierte Einheiten**). Im Auslieferungszustand ist die Steuerung auf Umdrehungen pro Minute eingestellt.

Objektbeschreibung

Index	6099 _h
Objektname	Homing Speed
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h

Name	Speed During Search For Switch
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000032 _h
Subindex	02 _h
Name	Speed During Search For Zero
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

Dieser Wert wird mit dem Zähler in Objekt **2061_h** und dem Nenner in Objekt **2062_h** verrechnet.

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 Schritten/s² an.

Objektbeschreibung

Index	609A _h
Objektname	Homing Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	

Vorgabewert	000001F4 _h
Firmware Version	
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.

60A4h Profile Jerk

Funktion

Im Falle einer ruck-begrenzten Rampe können in diesem Objekt die Größe der Rucks eingetragen werden. Ein Eintrag mit dem Wert "0" bedeutet, dass der Ruck nicht begrenzt ist.

Objektbeschreibung

Index	60A4 _h
Objektname	Profile Jerk
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	04 _h

Subindex	01 _h
Name	Begin Acceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h

Subindex	02 _h
Name	End Acceleration Jerk
Datentyp	UNSIGNED32

Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	03 _h
Name	Begin Deceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h
Subindex	04 _h
Name	End Deceleration Jerk
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	000003E8 _h

60C2h Interpolation Time Period

Funktion

Dieses Objekt enthält die Interpolationszeit in Millisekunden in 2er Potenzen.

Objektbeschreibung

Index	60C2 _h
Objektnamen	Interpolation Time Period
Object Code	RECORD
Datentyp	INTERPOLATION_TIME_PERIOD
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	h
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	h
Name	Highest Sub-index Supported

Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
<hr/>	
Subindex	h
Name	Interpolation Time Period Value
Datentyp	UNSIGNED8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h
<hr/>	
Subindex	h
Name	Interpolation Time Index
Datentyp	INTEGER8
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00 _h

Beschreibung

Die Subindizes haben folgende Funktionen:

- 01_h: Interpolations Zeit, Einheiten: Gibt die Interpolationszeit in 2er Potenzen an, Einheit in ms. Wird also eine "0" eingetragen, entspricht die Interpolationszeit dem Wert $2^0=1$ ms, der Wert "1" dementsprechend $2^1=2$ ms und so weiter.
- 02_h: Interpolations Zeit, Index: muss den Wert -3 halten.

60C5h Max Acceleration

Funktion

Dieses Objekt enthält die maximal zulässige Beschleunigungsrampe.

Für die Bremsrampe: siehe Objekt **60C6_h** "Max Deceleration".

Objektbeschreibung

Index	60C5 _h
Objektname	Max Acceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO

Zulässige Werte	
Vorgabewert	FFFFFFFFD _h
Firmware Version	
Änderungshistorie	

60C6h Max Deceleration

Funktion

Dieses Objekt enthält die maximal zulässige Bremsrampe.

Für die Beschleunigungsrampe : siehe Objekt **60C5_h** "Max Acceleration".

Objektbeschreibung

Index	60C6 _h
Objektname	Max Deceleration
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00001388 _h
Firmware Version	
Änderungshistorie	

60F4h Following Error Actual Value

Funktion

Dieses Objekt enthält den aktuellen Schleppfehler.

Objektbeschreibung

Index	60F4 _h
Objektname	Following Error Actual Value
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

Beschreibung

Dieses Objekt wird in den benutzerdefinierten Einheiten berechnet (siehe **Benutzerdefinierte Einheiten**).

60FDh Digital Inputs

Funktion

Mit diesem Objekt können die Digitaleingänge des Motors gelesen werden.

Objektbeschreibung

Index	60FD _h
Objektname	Digital Inputs
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
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
											DIR	CLK	HS	PLS	NLS

NLS (Negative Limit Switch)

negativer Endschalter

PLS (Positive Limit Switch)

positiver Endschalter

HS (Home Switch)

Referenzschalter

CLK (Clock)

Takteingang

DIR (Direction)

Richtungseingang

IN n (Input n)

Eingang n - die Anzahl der verwendeten Bits ist abhängig von der jeweiligen Steuerung.

60FEh Digital Outputs

Funktion

Mit diesem Objekt können die Digitalausgänge des Motors geschrieben werden.

Objektbeschreibung

Index	60FE _h
Objektname	Digital Outputs
Object Code	ARRAY
Datentyp	UNSIGNED32
Firmware Version	
Änderungshistorie	

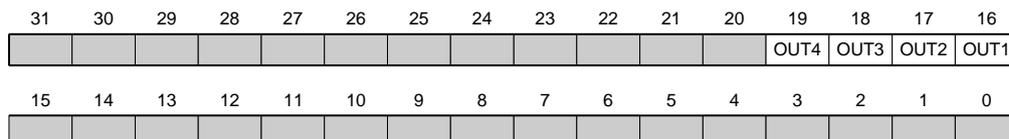
Wertebeschreibung

Subindex	00 _h
Name	Highest Sub-index Supported
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	01 _h

Subindex	01 _h
Name	Digital Outputs #1
Datentyp	UNSIGNED32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Zum Schreiben der Ausgänge müssen noch die Einträge in Objekt **3250_h**, Subindex 02_h bis 05_h berücksichtigt werden.



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 Mode eingetragen.

Dieses Objekt wird mit den benutzerdefinierten Einheiten verrechnet (siehe **Benutzerdefinierte Einheiten**). Im Auslieferungszustand ist die Steuerung auf Umdrehungen pro Minute eingestellt.

Objektbeschreibung

Index	60FF _h
Objektname	Target Velocity
Object Code	VARIABLE
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	RX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Firmware Version	
Änderungshistorie	

6402h Read Analogue Input

Funktion

Zeigt die Momentanwerte der Analogeingänge in User-Einheiten an.

Objektbeschreibung

Index	6402 _h
Objektname	Read Analogue Input
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

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

Subindex	01 _h
----------	-----------------

Name	Analogue Input 1
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Analogue Input 2
Datentyp	INTEGER32
Zugriff	nur lesen
PDO Mapping	TX-PDO
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

Die User-Einheiten setzen sich aus Offset (**6431_h**) und Pre-scaling Wert (**6432_h**) zusammen. Sind beide Objekteinträge noch mit Default-Werten beschrieben, wird der Wert in **6402_h** in der Einheit "ADC digits" angegeben.

Formeln zum Umrechnen von digits in die jeweilige Einheit:

Spannungseingang: $x \text{ digits} * 10 \text{ V} / 1024 \text{ digits}$

Stromeingang: $x \text{ digits} * 20 \text{ mA} / 1024 \text{ digits}$

Für die Subeinträge gilt:

- Subindex 00_h: Anzahl der Analogeingänge
- Subindex 01_h: Analogwert 1
- Subindex 02_h: Analogwert 2

6431h Analogue Input Offset

Funktion

Offset, der zum eingelesenen Analogwert (**6402_h**) addiert wird, bevor die Teilung mit dem Teiler aus dem Objekt **6432_h** vorgenommen wird.

Objektbeschreibung

Index	6431 _h
Objektname	Analogue Input Offset
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

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

Name	Number Of Analogue Inputs
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	Analogue Input 1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h
Subindex	02 _h
Name	Analogue Input 2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000000 _h

Beschreibung

- Subindex 00_h: Anzahl der Offsets
- Subindex 01_h: Offset für Analogeingang 1
- Subindex 02_h: Offset für Analogeingang 2

6432h Analogue Input Pre-scaling

Funktion

Wert, mit dem der eingelesene Analogwert (**6402_h**, **6431_h**) dividiert wird, bevor er in das Objekt **6402_h** geschrieben wird.

Objektbeschreibung

Index	6432 _h
Objektname	Analogue Input Pre-scaling
Object Code	ARRAY
Datentyp	INTEGER32
Firmware Version	
Änderungshistorie	

Wertebeschreibung

Subindex	00 _h
Name	Number Of Analogue Inputs
Datentyp	UNSIGNED8
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	02 _h
Subindex	01 _h
Name	Analogue Input 1
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h
Subindex	02 _h
Name	Analogue Input 2
Datentyp	INTEGER32
Zugriff	lesen/schreiben
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	00000001 _h

Beschreibung

- Subindex 00_h: Anzahl der Teiler
- Subindex 01_h: Teiler für Analogeingang 1
- Subindex 02_h: Teiler für Analogeingang 2

6502h Supported Drive Modes

Funktion

Das Objekt beschreibt die unterstützten Drive Modi.

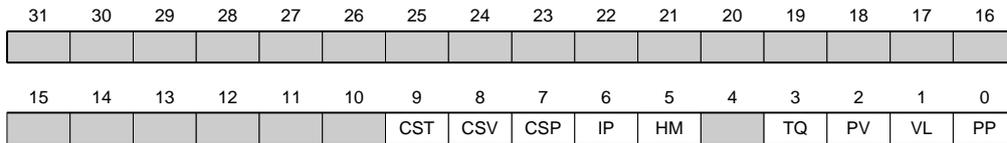
Objektbeschreibung

Index	6502 _h
Objektname	Supported Drive Modes
Object Code	VARIABLE
Datentyp	UNSIGNED32
Zugriff	nur lesen
PDO Mapping	TX-PDO

Zulässige Werte
 Vorgabewert 000000AF_h
 Firmware Version
 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 (Drehmoment) Modus

HM

Homing (Referenzfahrt) Modus

IP

Interpolated Position Modus

CSP

Cyclic Synchronous Position Modus

CSV

Cyclic Synchronous Velocity Modus

CST

Cyclic Sync Torque Modus

6505h Http Drive Catalogue Address

Funktion

Dieses Objekt enthält die Web-Adresse des Herstellers als Zeichenkette.

Objektbeschreibung

Index	6505 _h
Objektname	Http Drive Catalogue Address
Object Code	VARIABLE

Datentyp	VISIBLE_STRING
Zugriff	nur lesen
PDO Mapping	nein
Zulässige Werte	
Vorgabewert	http://www.nanotec.de
Firmware Version	
Änderungshistorie	

12 Copyrights

12.1 Einführung

In der Nanotec Software sind Komponenten aus Produkten externer Software-Hersteller integriert. In diesem Kapitel finden Sie die Copyright-Informationen zu den verwendeten externen Software-Quellen.

12.2 AES

FIPS-197 compliant AES implementation

Based on XySSL: Copyright (C) 2006-2008 Christophe Devine

Copyright (C) 2009 Paul Bakker <polarssl_maintainer at polarssl dot org>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution; or, the application vendor's website must provide a copy of this notice.
- Neither the names of PolarSSL or XySSL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The AES block cipher was designed by Vincent Rijmen and Joan Daemen.

<http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>

<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>

12.3 Arcfour (RC4)

Copyright (c) April 29, 1997 Kalle Kaukonen.

All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that this copyright notice and disclaimer are retained.

THIS SOFTWARE IS PROVIDED BY KALLE KAUKONEN 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 KALLE KAUKONEN OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12.4 MD5

MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

12.5 uIP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12.6 DHCP

Copyright (c) 2005, Swedish Institute of Computer Science

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

12.7 CMSIS DSP Software Library

Copyright (C) 2010 ARM Limited. All rights reserved.

12.8 FatFs

FatFs - FAT file system module include file R0.08 (C)ChaN, 2010

FatFs module is a generic FAT file system module for small embedded systems.

This is a free software that opened for education, research and commercial developments under license policy of following terms.

Copyright (C) 2010, ChaN, all right reserved.

The FatFs module is a free software and there is NO WARRANTY.

No restriction on use. You can use, modify and redistribute it for personal, non-profit or commercial product UNDER YOUR RESPONSIBILITY.

Redistributions of source code must retain the above copyright notice.

12.9 Protothreads

Protothread class and macros for lightweight, stackless threads in C++.

This was "ported" to C++ from Adam Dunkels' protothreads C library at: <http://www.sics.se/~adam/pt/>

Originally ported for use by Hamilton Jet (www.hamiltonjet.co.nz) by Ben Hoyt, but stripped down for public release. See his blog entry about it for more information: <http://blog.micropledge.com/2008/07/protothreads/>

Original BSD-style license

Copyright (c) 2004-2005, Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Institute nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the Institute and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Institute or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.