



Drive Controller SD4x

EtherCAT-Anbindung

Das Einstiegshandbuch für den Anwender



Copyright

Originalbetriebsanleitung, Copyright © 2026 SIEB & MEYER AG

Alle Rechte vorbehalten.

Diese Anleitung darf nur mit einer ausdrücklichen schriftlichen Genehmigung der SIEB & MEYER AG kopiert werden. Das gilt auch für Auszüge.

Marken

Alle in dieser Anleitung aufgeführten Produkt-, Schrift- und Firmennamen und Logos sind gegebenenfalls Marken oder eingetragene Marken der jeweiligen Firmen.

SIEB & MEYER weltweit

Bei Fragen zu unseren Produkten oder technischen Rückfragen wenden Sie sich bitte an uns.

SIEB & MEYER AG
Auf dem Schmaarkamp 21
21339 Lüneburg
Deutschland

Tel.: +49 4131 203 0
info@sieb-meyer.de
<http://www.sieb-meyer.de>

SIEB & MEYER Shenzhen Trading Co. Ltd.
Room 316, Haixiang Plaza,
No. 1048 Nanhai Avenue,
Zhaoshang Subdistrict, Nanshan District,
Shenzhen City
China

Tel.: +86 755 2681 1417 / +86 755 2681 2487
info@sieb-meyer.cn
<http://www.sieb-meyer.cn>

SIEB & MEYER Asia Co. Ltd.
5 Fl, No. 578, Sec. 1
Min-Sheng N. Road
Kwei-Shan Hsiang
Guishan Dist., Taoyuan City 33393
Taiwan

Tel.: +886 3 311 5560
info@sieb-meyer.tw

1	Über dieses Handbuch.....	5
1.1	Abkürzungen.....	5
2	Antriebssteuerung (CiA 402).....	7
2.1	Servicedatenobjekte (SDO).....	7
2.2	Prozessdatenobjekte (PDO).....	8
2.3	Zustandsmaschine.....	9
2.3.1	Befehle.....	10
	Controlword (0x6040).....	10
2.3.2	Zustände.....	11
	Statusword (0x6041).....	11
2.3.3	Option Codes.....	12
	Abort Connection Option Code (0x6007).....	13
	Quick Stop Option Code (0x605A).....	13
	Shutdown Option Code (0x605B).....	13
	Disable Operation Option Code (0x605C).....	13
	Halt Option Code (0x605D).....	14
	Fault Reaction Option Code (0x605E).....	14
2.4	Betriebsarten.....	14
2.4.1	Cyclic Synchronous Velocity Mode (CSV).....	14
2.4.2	Profile Velocity Mode (PV).....	16
2.4.3	Velocity Mode (VL).....	17
2.4.3.1	Soll- und Istwerte im Velocity Mode.....	18
	vl Target Velocity (0x6042).....	18
	vl Velocity Demand (0x6043).....	18
	vl Velocity Actual Value (0x6044).....	18
	vl Velocity Min Max Amount (0x6046).....	18
	vl Velocity Acceleration (0x6048).....	19
	vl Velocity Deceleration (0x6049).....	20
	vl Velocity Quick Stop (0x604A).....	20
	vl Set-point Factor (0x604B).....	21
	vl Dimension Factor (0x604C).....	22
2.4.4	Torque Mode.....	23
2.4.5	Betriebsart wechseln.....	23
	Supported Drive Modes (0x6502).....	24
	Mode of Operation (0x6060).....	25
	Mode of Operation Display (0x6061).....	25
2.5	Soll- und Istwerte.....	25
2.5.1	Allgemeine Sollwerte.....	26
	Target Velocity (0x60FF).....	26
	Target Torque (0x6071).....	26
	Interpolation Time Period (0x60C2).....	26
	Motion Profile Type (0x6086).....	27
2.5.2	Allgemeine Istwerte.....	27
	Position Actual Value (0x6064).....	27
	Velocity Demand Value (0x606B).....	28
	Velocity Actual Value (0x606C).....	28
	Torque Demand (0x6074).....	28
	Torque Actual Value (0x6077).....	28
	Current Actual Value (0x6078).....	28
	DC Link Circuit Voltage (0x6079).....	29
2.5.3	Motorparameter.....	29
	Motor Rated Current (0x6075).....	29
	Motor Rated Torque (0x6076).....	29
	Motor Type (0x6402).....	29
2.5.4	SI-Einheiten.....	30
	Position (0x60A8).....	30
	Velocity (0x60A9).....	30
	Acceleration (0x60AA).....	31
	Jerk (0x60AB).....	31
2.5.5	Skalierung.....	31
	Polarity (0x607E).....	31
2.5.6	Offsets.....	32
	Velocity Offset (0x60B1).....	32
	Torque Offset (0x60B2).....	32

2.5.7	Begrenzungen.....	32
	Max Torque (0x6072).....	32
	Max Current (0x6073).....	33
	Max Profile Velocity (0x607F).....	33
	Max Motor Speed (0x6080).....	33
	Positive Torque Limit (0x60E0).....	33
	Negative Torque Limit (0x60E1).....	33
2.5.7.7	Beschleunigungs- und Verzögerungsbegrenzungen.....	34
	Max Acceleration (0x60C5).....	34
	Max Deceleration (0x60C6).....	34
	Profile Acceleration (0x6083).....	34
	Profile Deceleration (0x6084).....	34
	Quick Stop Deceleration (0x6085).....	34
	Torque Slope (0x6087).....	35
2.5.8	Sonstige.....	35
	Digital Inputs (0x60FD).....	35
	Digital Outputs (0x60FE).....	35
	Error Code (0x603F).....	36
2.6	Parametersatzumschaltung.....	37
2.6.1	Parametersatzumschaltung über Feldbus.....	38
3	Ansteuerung über EtherCAT.....	40
3.1	EtherCAT-Netzwerk.....	40
3.2	SPS-Anbindung (TwinCAT3).....	40
3.2.1	EtherCAT State Machine (ESM).....	46
3.2.2	SDO-Kommunikation.....	48
3.2.3	Adressierung und Identifikation.....	52
3.2.3.1	Auto-Inkrement-Adresse.....	52
3.2.3.2	EtherCAT-Adresse.....	52
	EtherCAT-Adresse im SubDevice speichern.....	53
3.2.3.3	Hot Connect.....	54
	Configured Station Alias (ADO 0x0012).....	55
	Explicit Device Identification (ADO 0x0134).....	56
3.2.4	Allgemeine Objekte.....	58
	Objekt 0x1000 – Device Type.....	58
	Objekt 0x1600 bis 0x1603 – RPDO Mapping Parameter.....	59
	Objekt 1A00h bis 1A03h – TPDO Mapping Parameter.....	60
	Objekt 0x1C00 – Sync Manager Communication Type.....	61
	Objekt 0x1C12 – RPDO Assignment.....	62
	Objekt 0x1C13 – TPDO Assignment.....	62
3.3	Unterstützte Synchronisationsarten.....	63
3.3.1	Free Run.....	64
3.3.2	DC-Synchron (Distributed Clocks).....	64
4	Antriebsparametrierung und Diagnose.....	66
4.1	Parametrierung in drivemaster4.....	66
4.2	Diagnose in drivemaster4.....	67
4.2.1	Telegramme.....	67
4.3	Problembehandlung: Zustandsmaschine ignoriert Steuerbefehle.....	68
5	Weiterführende Informationen.....	70
	Anhang.....	71
A	Präfixe (aus CiA 303-2).....	71
B	Einheiten (aus CiA 303-2 und CiA 402).....	72
	Index.....	74

1 Über dieses Handbuch

Das vorliegende Handbuch beschreibt die Ankoppelung eines Drive Controllers der SD4x-Gerätefamilie an eine übergeordnete Steuerung (MainDevice) über EtherCAT. Informationen und Sicherheitshinweise zu Ihrem SD4x-Antrieb finden Sie in der entsprechenden Hardware- bzw. Softwaredokumentation.

Eine grundlegende Beschreibung des EtherCAT-Netzwerks finden Sie auf der Website der EtherCAT Technology Group (www.ethercat.org).

Das implementierte Protokoll EtherCAT CoE basiert auf den CANopen-Profilen. Die CANopen-Profile beschreiben den Datenaustausch zwischen den Busteilnehmern. Der CANopen-Standard definiert das grundlegende Kommunikationsprofil und die Geräteprofile der angeschlossenen Busteilnehmer.

EtherCAT stellt mit dem CoE-Protokoll die gleichen Kommunikationsmechanismen wie der CANopen-Standard bereit:

- ▶ Objektverzeichnis
- ▶ Mapping von Prozessdatenobjekten (PDO)
- ▶ Servicedatenobjekte (SDO)
- ▶ Netzwerkmanagement (NMT)

Zudem können Anwender die CANopen-Geräteprofile nutzen, zum Beispiel das Antriebsprofil CiA 402. Hierfür sind die Normen der Nutzerorganisation CiA und der EtherCAT Technology Group ETG heranzuziehen, siehe [Abschnitt „Weitere Dokumente“, Seite 70](#).

1.1 Abkürzungen

CAN	C ontroller A rea N etwork
CiA	C AN in A utomation
CoE	C ANopen o ver E therCAT
CSV	C yclic S ynchronous V elocity Mode
DC	D istributed C locks (verteilte Uhren)
EDS	E lectronic D ata S heet (Gerätebeschreibungsdatei im ASCII-Format)
ESI	E therCAT S ubDevice I nformation
ESM	E therCAT S tate M achine
ETG	E therCAT T echnology G roup
EtherCAT	E thernet for C ontrol A utomation T echnology
MainDevice	M ain D evice (ehemals Master)
PDO	P rozess d aten o bjekt
PV	P rofile V elocity Mode
RO	R ead O nly (Objektzugriff: nur lesen)
RW	R ead W rite (Objektzugriff: lesen und schreiben)
Rx	Receive (empfangen)
SDO	S ervice d aten o bjekt
SM	S ync M anager
SoE	S ERCOS o ver E therCAT
SPS	s peicher p rogrammierbare S teuerung



- SU** Sync Unit (Synchronisationsmodul)
- SubDevice** SubordinateDevice (ehemals Slave)
- Tx** Transmit (senden)
- VL** Velocity Mode

2 Antriebssteuerung (CiA 402)

Die Drive Controller der Gerätefamilie SD4x unterstützen das Antriebsprofil CiA 402. Das bedeutet, dass die in diesem Kapitel beschriebene Antriebssteuerung nach dem DS301-Standard und dem DS402-Standard umgesetzt ist. Die entsprechenden Antriebsobjekte haben den Präfix „DS301“ bzw. „DS402“.

Das Antriebsprofil CiA 402 definiert Objekte, welche in [Prozessdatenobjekte \(PDO\)](#) und [Servicedatenobjekte \(SDO\)](#) gegliedert sind. PDOs können zyklisch (echtzeitfähig) und SDOs können azyklisch (auf Anfrage, nicht echtzeitfähig) über den Feldbus ausgetauscht werden.

Der Antrieb folgt einer vorgegebenen Zustandsmaschine. Diese wird mit Hilfe der Befehle aus dem [Controlword](#) gesteuert und gibt über das [Statusword](#) den aktuellen Zustand zurück. Über die [Option Codes](#) können Sie das Abschaltverhalten des Antriebs für die entsprechenden Abschaltbefehle verändern. Die verwendeten Soll- und Istwerte sind abhängig von der gewählten Betriebsart ([Cyclic Synchronous Velocity Mode](#), [Profile Velocity Mode](#), [Velocity Mode](#) oder [Torque Mode](#)). Die Umschaltung der Betriebsart ist unter [Betriebsart wechseln](#) beschrieben.

Die Fehlercodes sind hardwareabhängig. Eine detaillierte Beschreibung der Fehlercodes finden Sie in der Hardwaredokumentation zu Ihrem Gerät.

Objektdarstellung

Alle relevanten Objekte sind in Tabellenform mit folgenden Informationen dargestellt:

Kurzname des Objekts		Kurze Beschreibung des Objekts	
Eindeutiger Name des Objekts			Objektindex
Zugriff ⁽¹⁾	Speicher ⁽²⁾	Einheit	Datentyp

⁽¹⁾ Mögliche Werte:

RW = Read/Write (Lesen/Schreiben)
RO = Read only (nur Lesen)

⁽²⁾ Mögliche Werte:

Volatile = flüchtiger Datenspeicher (geht nach Stromabschaltung verloren)
Non-volatile = nichtflüchtiger Datenspeicher (bleibt nach Stromabschaltung erhalten)

2.1 Servicedatenobjekte (SDO)

Servicedatenobjekte sind für Daten vorgesehen, die nur selten und nicht in Echtzeit, azyklisch ausgetauscht werden. Während der Bootphase des Feldbusses wird zum Beispiel das PDO-Mapping zwischen der Feldbussteuerung (MDevice) und dem Feldbus Teilnehmer (SubDevice) über SDOs ausgehandelt.

SDO-Zugriffe werden immer vom MDevice ausgelöst, indem er eine Anfrage an das SubDevice sendet. Die Anfrage ist azyklisch und es kann mehrere Feldbuszyklen dauern, bis sie vom SubDevice beantwortet wird.

Die Feldbussteuerung behandelt SDO-Telegramme nachrangig. Sollte die gesamte Feldbuskapazität also bereits von PDO-Telegrammen belegt sein, kann keine SDO-Kommunikation mehr stattfinden.

Eine SDO-Anfrage beinhaltet 2 Adressen:

- ▶ SubDevice-Adresse (z. B. ADS-Adresse oder ID)
- ▶ Objektadresse (z. B. Objekt-Index und Objekt-Subindex)

Eine Anfrage kann einzelne Objekte oder auch ganze Objektblöcke adressieren. Der genaue Aufbau und Ablauf einer SDO-Anfrage ist feldbusspezifisch und kann variieren.

2.2 Prozessdatenobjekte (PDO)

Prozessdatenobjekte sind für die Nutzdaten vorgesehen, die in Echtzeit, zyklisch ausgetauscht werden.

Welche Prozessdaten ausgetauscht werden sollen, wird von der Feldbussteuerung (M-Device) während der Bootphase des Feldbusses überprüft oder ausgehandelt. Dies ist davon abhängig, ob ein Feldbusteilnehmer (SubDevice) fest vorgegebene oder veränderbare PDOs unterstützt. SD4x-Antriebe unterstützen veränderbare PDOs.

Nicht alle Datenobjekte eines SubDevices können in Echtzeit verarbeitet werden. Deshalb können auch nicht alle Daten über PDOs ausgetauscht werden. Welche Daten über PDOs ausgetauscht werden können, geht meist aus den feldbusspezifischen Beschreibungsdateien (z. B. ESI-, XML- oder EDS-Datei) hervor.

Hinweis

Die heutigen Feldbusse haben ausreichend Bandbreite, um viele Nutzdaten in hoher Geschwindigkeit übertragen zu können. Dennoch sollten Daten, die keine Echtzeitanforderung haben, über Servicedatenobjekte ausgetauscht werden.

2.3 Zustandsmaschine

Die Zustandsmaschine (im DS402-Standard: Finite State Automaton, FSA) steuert die Leistungselektronik des Antriebs. Das folgende Diagramm zeigt die Zustände, deren Bedeutung und den Ablauf der Zustandsmaschine.

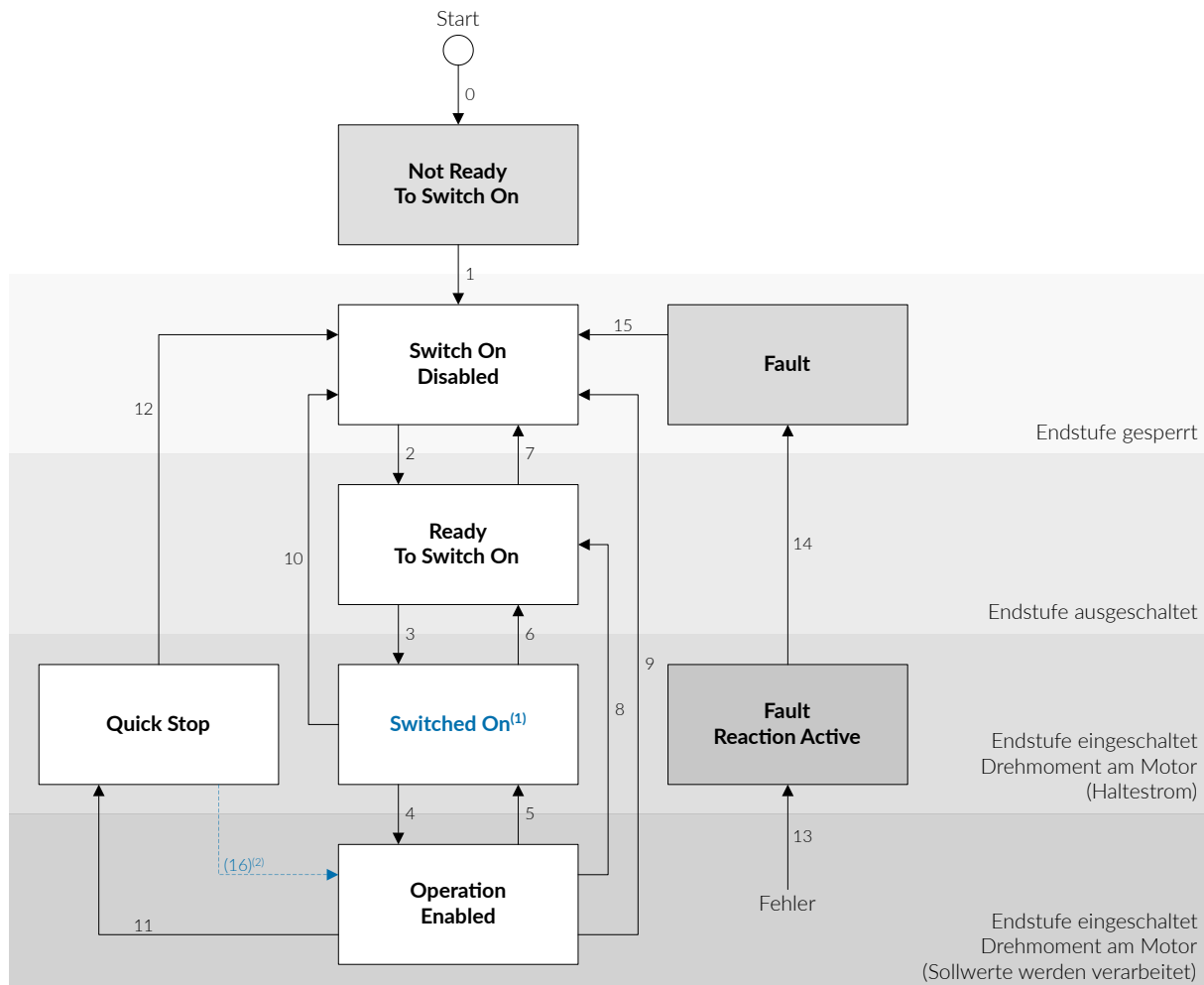


Abb. 1: Antriebszustände und Zustandsübergänge

- (1) Abweichend vom DS402-Standard wird bei SD4x-Antrieben im Zustand „Switched On“ die Endstufe bereits eingeschaltet, sodass Drehmoment am Motor anliegt. Die Sollwerte sind in diesem Zustand noch nicht frei gegeben und der Motor wird nur gehalten.
- (2) Die Transition 16 wird vom DS402-Standard nicht mehr empfohlen. Aus diesem Grund unterstützen SD4x-Antriebe die Transition 16 nicht.

Die Kästen entsprechen den Zuständen der Zustandsmaschine. Die entsprechenden Zustände der Leistungselektronik finden Sie auf der rechten Seite der Darstellung. Die grau gefüllten Kästen enthalten Zustände, die ausschließlich automatisch und nicht über Befehle erreicht werden. Die Nummern an den Übergängen sind Transitionen, die mithilfe der Befehle ausgelöst werden können. Welcher Befehl welchen Übergang auslöst, finden Sie im [Kapitel 2.3.1 „Befehle“, Seite 10](#).

Hinweis

Der Zustand der Zustandsmaschine kann zusätzlich durch äußere Ereignisse geändert werden, sodass eine Betriebsfreigabe über Feldbus nicht möglich ist. Weitere Informationen dazu finden Sie im [Kapitel 4.3 „Problembehandlung: Zustandsmaschine ignoriert Steuerbefehle“, Seite 68](#).

2.3.1 Befehle

Die Steuerbefehle sind vom aktiven Zustand abhängig. Sie werden durch Setzen der folgenden Bitmuster im Controlword ausgelöst:

Befehl	Bits im Controlword ⁽¹⁾					Transition
	Bit 7 Fehler zu- rücksetzen	Bit 3 Betrieb freigeben	Bit 2 Schnellhalt	Bit 1 Spannung freigeben	Bit 0 Einschalten	
Shutdown (Stillsetzen)	0	X	1	1	0	2, 6, 8
Switch On (Einschalten)	0	X ⁽²⁾	1	1	1	3
Disable Voltage (Spannung sperren)	0	X	X	0	X	7, 9, 10, 12
Quick Stop (Schnellhalt)	0	X	0	1	X	7, 10, 11
Disable Operation (Betrieb sperren)	0	0	1	1	1	5
Enable Operation (Betrieb freigeben)	0	1	1	1	1	4, (16) ⁽³⁾
Fault Reset (Fehler zurücksetzen)	 ⁽⁴⁾	X	X	X	X	15

⁽¹⁾ Die mit „X“ gekennzeichneten Bits haben an dieser Stelle keine Bedeutung.

⁽²⁾ Bei 1 wechselt die Zustandsmaschine nach Abarbeitung von „Switched On“ automatisch in den nächsten Zustand „Operation Enabled“.

⁽³⁾ Die Transition 16 wird vom DS402-Standard nicht mehr empfohlen. Aus diesem Grund unterstützen SD4x-Antriebe die Transition 16 nicht.

⁽⁴⁾ Positive Flanke (Wechsel von 0 auf 1, Impuls)

Nicht erwähnte Bits im Controlword sind für die Befehle irrelevant. Ist ein Befehl in einem Zustand unbekannt, wird er ignoriert. Nicht erwähnte Transitionen sind automatische Übergänge in der Zustandsmaschine, die nicht über die Befehle ausgelöst werden können.

Controlword (0x6040)

Das Controlword ist ein bitcodiertes WORD und dient zur Übermittlung von Steueranforderungen an den Antrieb.

Controlword		Steuerbefehle: Bit 0: Einschalten Bit 1: Spannung freigeben Bit 2: Schnellhalt Bit 3: Betrieb freigeben Bit 4: Betriebsartspezifisch 1 Bit 5: Betriebsartspezifisch 2 Bit 6: Betriebsartspezifisch 3 Bit 7: Fehlerreset Bit 8: Halt Bits 9 bis 15: Reserviert	
DS402_0x6040_CONTROLWORD			ID: 0x6040
RW	Volatile	Einheit: –	U16

Aus den Bits 0 (Einschalten), 1 (Spannung freigeben), 2 (Schnellhalt), 3 (Betrieb freigeben) und 7 (Fehlerreset) werden die Befehle zur Steuerung der Zustandsmaschine zusammengesetzt.

Die Funktion „Halt“ (Bit 8) ist nur im Zustand „Operation Enabled“ aktiv. Sie führt dazu, dass der Motor mit dem Halt-Option-Code gestoppt und anschließend gehalten wird. Dabei wird der Zustand „Operation Enabled“ nicht verlassen.

Die betriebsartspezifischen Bits 4 bis 7 werden in den entsprechenden Betriebsarten erläutert.

2.3.2 Zustände

Die Zustände der Zustandsmaschine sind nach dem folgenden Bitmuster im Statusword hinterlegt. Für die einzelnen Zustände sind die folgenden Bits gesetzt:

Zustand	Bits im Statusword ⁽¹⁾					
	Bit 6 Einschaltsperrere	Bit 5 Schnellhalt	Bit 3 Fehler	Bit 2 Betrieb freigegeben	Bit 1 Eingeschaltet	Bit 0 Einschaltbereit
Not Ready To Switch On (Firmware nicht bereit)	0	X	0	0	0	0
Switch On Disabled (Einschaltsperrere)	1	X	0	0	0	0
Ready To Switch On (Einschaltbereit)	0	1	0	0	0	1
Switched On (Eingeschaltet)	0	1	0	0	1	1
Operation Enabled (Betrieb freigegeben)	0	1	0	1	1	1
Quick Stop Active (Schnellhalt)	0	0	0	1	1	1
Fault Reaction Active (Fehlerreaktion aktiv)	0	X	1	1	1	1
Fault (Fehler)	0	X	1	X	0	X

⁽¹⁾ Die mit „X“ gekennzeichneten Bits haben an dieser Stelle keine Bedeutung.

Nicht erwähnte Bits im Statusword sind für die Zustände irrelevant.

Statusword (0x6041)

Das Statusword ist ein bitcodiertes WORD und enthält Statusinformationen des Antriebs.

Statusword	Gerätezustände: Bit 0: Einschaltbereit Bit 1: Eingeschaltet Bit 2: Betrieb freigegeben Bit 3: Fehler Bit 4: Spannung freigegeben Bit 5: Schnellhalt Bit 6: Einschaltsperrere Bit 7: Warnung Bit 8: <i>Drive Setup Tool (DRS) Active*</i> Bit 9: Remote Bit 10: Zielwert erreicht Bit 11: <i>Stromgrenze erreicht*</i> Bit 12: Betriebsartspezifisch 1 Bit 13: Betriebsartspezifisch 2 Bit 14: <i>Safe Torque Off (STO)*</i> Bit 15: <i>Initialisierung abgeschlossen*</i> <i>*Kursiv dargestellte Bitbelegungen sind SIEB & MEYER-spezifisch und können bei anderen Antriebsherstellern abweichen.</i>		
DS402_0x6041_STATUSWORD			
RO	Volatile	Einheit: –	U16
			ID: 0x6041

Die Bits 0 (Einschaltbereit), 1 (Eingeschaltet), 2 (Betrieb freigegeben), 3 (Fehler), 5 (Schnellhalt) und 6 (Einschaltsperrung) liefern über Bitmuster den aktuellen Zustand der Zustandsmaschine zurück. Liegt ein Fehler an, wird die Fehlerinformation im **Error Code** (ID 0x603F) zurückgegeben.

Bit 4 (Spannung freigegeben) wird aktiv, wenn die Spannung zur Speisung des Zwischenkreises anliegt.

Bit 7 (Warnung) wird aktiv, wenn eine Warnmeldung vorliegt. Informationen zur aktuellen Warnmeldung werden im **Error Code** (ID 0x603F) zurückgegeben.

Bit 8 (Drive Setup Tool (DRS) Active) ist SIEB & MEYER-spezifisch und wird aktiv, wenn der Antrieb von der *drivemaster4*-Software oder einem anderen Inbetriebnahmewerkzeug gesteuert wird.

Bit 9 (Remote) wird aktiv, wenn der Antrieb die Steuerbefehle aus dem Controlword zur Steuerung verwendet.

Bit 10 (Zielwert erreicht) wird aktiv, wenn der Antrieb den Zielwert erreicht hat. Dieses Bit ist von der Betriebsart, dem Halt-Bit im Controlword und dem aktuellen Zustand abhängig:

- ▶ Betriebsarten: Bit 10 wird gesetzt, wenn der eingestellte Zielwert für die Geschwindigkeit, die Position oder das Drehmoment erreicht wurde.
Beispiel für die Geschwindigkeitsbetriebsarten (CSV/PV/VL): Der Zielwert ist erreicht, wenn die Differenz der Istgeschwindigkeit zur Sollgeschwindigkeit länger als die Geschwindigkeitsfensterzeit innerhalb des Geschwindigkeitsfensters liegt. Das Geschwindigkeitsfenster stellen Sie in der *drivemaster4*-Software unter „Meldungen → Meldungen → Sollwert erreicht [M51]“ ein.
- ▶ Zustand „Quick Stop Active“: Wenn der Antrieb den Zustand „Quick Stop Active“ nicht selbstständig verlässt, gibt Bit 10 an, dass die Schnellhaltfunktion abgeschlossen wurde und der Motor gehalten wird.
- ▶ Halt-Bit im Controlword (Bit 8) ist aktiv: Bit 10 wird gesetzt, sobald der Antrieb steht.
- ▶ Umschalten der Betriebsart: Bit 10 wird gesetzt, wenn die Betriebsart erfolgreich umgeschaltet wurde.

Bit 11 (Stromgrenze erreicht) ist SIEB & MEYER-spezifisch und wird aktiv, wenn der Antrieb an der eingestellten Stromgrenze betrieben wird.

Die Bits 12 und 13 sind betriebsartspezifisch und werden in den Kapiteln über die Betriebsarten erläutert.

Bit 14 (Safe Torque Off (STO)) ist SIEB & MEYER-spezifisch und wird aktiv, wenn die Sicherheitsfunktion STO den Antrieb abgeschaltet hält. Dies ist zum Beispiel bei einem Nothalt der Fall, der mit den SAFE-Eingängen des Antriebs verbunden ist.

Bit 15 (Initialisierung abgeschlossen) ist SIEB & MEYER-spezifisch und wird aktiv, sobald der Antrieb erfolgreich initialisiert wurde.

2.3.3 Option Codes

Die Option Codes definieren das Verhalten des Antriebs bei Empfang eines Stopp-Befehls (z. B. Disable Operation) oder bei einem Abschaltereignis (z. B. Auftreten einer Störung oder Verbindungsverlust zur Steuerung).

Abort Connection Option Code (0x6007)

Dieses Objekt definiert die Reaktion des Antriebs auf einen Verbindungsabbruch durch eines der folgenden Ereignisse: Bus-Off-Zustand, Heartbeat-/Node-Guarding-Fehler, NMT-Stopped-Zustand, Zurücksetzen (Reset) der Anwendung oder der Konfiguration.

Abort Connection Option Code		Auswahl der Reaktion auf einen Feldbus-Kommunikationsfehler: 00: Keine Aktion 01: Fehlersignal 02: Befehl Spannung sperren 03: Schnellhaltbefehl	
Objekt: DS402_0X6007_ABORT_CONNECTION_OPTION_CODE			ID: 0x6007
RW	Volatile	Einheit: –	l16

Quick Stop Option Code (0x605A)

Dieses Objekt definiert die Reaktion des Antriebs auf einen Schnellhaltbefehl.

Quick Stop Option Code		Auswahl der Reaktion auf einen Schnellhaltbefehl: 00: Antriebsfunktion sperren 01: Abbremsen mit Bremsrampe und Wechsel in den Status 'Einschaltsperrung' 02: Abbremsen mit Schnellhalttrappe und Wechsel in den Status 'Einschaltsperrung' 03: Abbremsen mit Strombegrenzung und Wechsel in den Status 'Einschaltsperrung'	
DS402_0X605A_QUICK_STOP_OPTION_CODE			ID: 0x605A
RW	Volatile	Einheit: –	l16

Die Rampen sind in der Betriebsart definiert.

Shutdown Option Code (0x605B)

Dieses Objekt definiert die Reaktion des Antriebs auf einen Shutdown-Befehl (Stillsetzen), d. h. bei einem Wechsel vom Zustand „Operation Enabled“ (Betrieb freigegeben) in den Zustand „Ready To Switch On“ (Einschaltbereit).

Shutdown Option Code		Auswahl der Reaktion auf den Befehl „Shutdown“: –1: Abbremsen durch Kurzschließen der Motorphasen 00: Antriebsfunktion sperren (Leistungsendstufe des Antriebs ausschalten) 01: Abbremsen mit Bremsrampe, Antriebsfunktion sperren	
DS402_0X605B_SHUTDOWN_OPTION_CODE			ID: 0x605B
RW	Volatile	Einheit: –	l16

Die Bremsrampe ist in der Betriebsart definiert.

Disable Operation Option Code (0x605C)

Dieses Objekt definiert die Reaktion des Antriebs auf einen Disable-Operation-Befehl (Betrieb sperren), d. h. bei einem Wechsel vom Zustand „Operation Enabled“ (Betrieb freigegeben) in den Zustand „Switched On“ (Eingeschaltet).

Disable Operation Option Code		Auswahl der Reaktion auf den Befehl „Disable Operation“: 01: Abbremsen mit Bremsrampe, Antriebsfunktion sperren	
DS402_0X605C_DISABLE_OPERATION_OPTION_CODE			ID: 0x605C
RW	Volatile	Einheit: –	l16

Die Bremsrampe ist in der Betriebsart definiert.

Hinweis

Eine Änderung des Wertes ist nicht vorgesehen.

Halt Option Code (0x605D)

Dieses Objekt definiert die Reaktion des Antriebs auf einen Halte-Befehl.

Halt Option Code		Auswahl der Reaktion auf den Befehl „Halt“: 01: Abbremsen mit Bremsrampe und Status 'Betrieb freigegeben' beibehalten 02: Abbremsen mit Schnellhalterampe und Status 'Betrieb freigegeben' beibehalten 03: Abbremsen mit Strombegrenzung und Status 'Betrieb freigegeben' beibehalten	
DS402_0X605D_HALT_OPTION_CODE			ID: 0x605D
RW	Volatile	Einheit: –	l16

Die Rampen sind in der Betriebsart definiert.

Fault Reaction Option Code (0x605E)

Dieses Objekt definiert die Reaktion des Antriebs auf einen Fehler.

Fault Reaction Option Code		Auswahl der Reaktion auf einen Fehler: 00: Antriebsfunktion sperren, Motor dreht frei 01: Abbremsen mit Bremsrampe 02: Abbremsen mit Schnellhalterampe 03: Abbremsen mit Strombegrenzung 04: Abbremsen mit Spannungsbegrenzung	
DS402_0X605D_HALT_OPTION_CODE			ID: 0x605E
RW	Volatile	Einheit: –	l16

Die Rampen sind in der Betriebsart definiert.

2.4 Betriebsarten

Der Antrieb unterstützt verschiedenen Betriebsarten, die abhängig von der Gerätevariante, dem Motor, den angeschlossenen Sensoren und der eingestellten Antriebsfunktion sind.

Die folgenden Kapitel beschreiben die möglichen Betriebsarten und den Betriebsartwechsel.

2.4.1 Cyclic Synchronous Velocity Mode (CSV)

Die Betriebsart Cyclic Synchronous Velocity Mode (CSV Mode) ermöglicht eine externe Trajektorienberechnung oder Lageregelung durch eine übergeordneten Steuerung. Der Antrieb folgt dann den Sollwerten von der Steuerung. Damit die Sollwerte sauber interpoliert werden können, muss der Antrieb mit der Steuerung synchronisiert werden. Zudem muss die Interpolationszeit (Objekt **Interpolation Time Period**) auf die Feldbuszykluszeit gesetzt werden.

Für die Steuerung des Antriebs im CSV Mode sind mindestens die Objekte **Control-word**, **Statusword** und **Target Velocity** notwendig. Als Rückgabewert empfiehlt sich, **Velocity Actual Value** zu verwenden.

Das folgende Diagramm zeigt die verwendeten Objekte und beispielhaft deren Funktion:

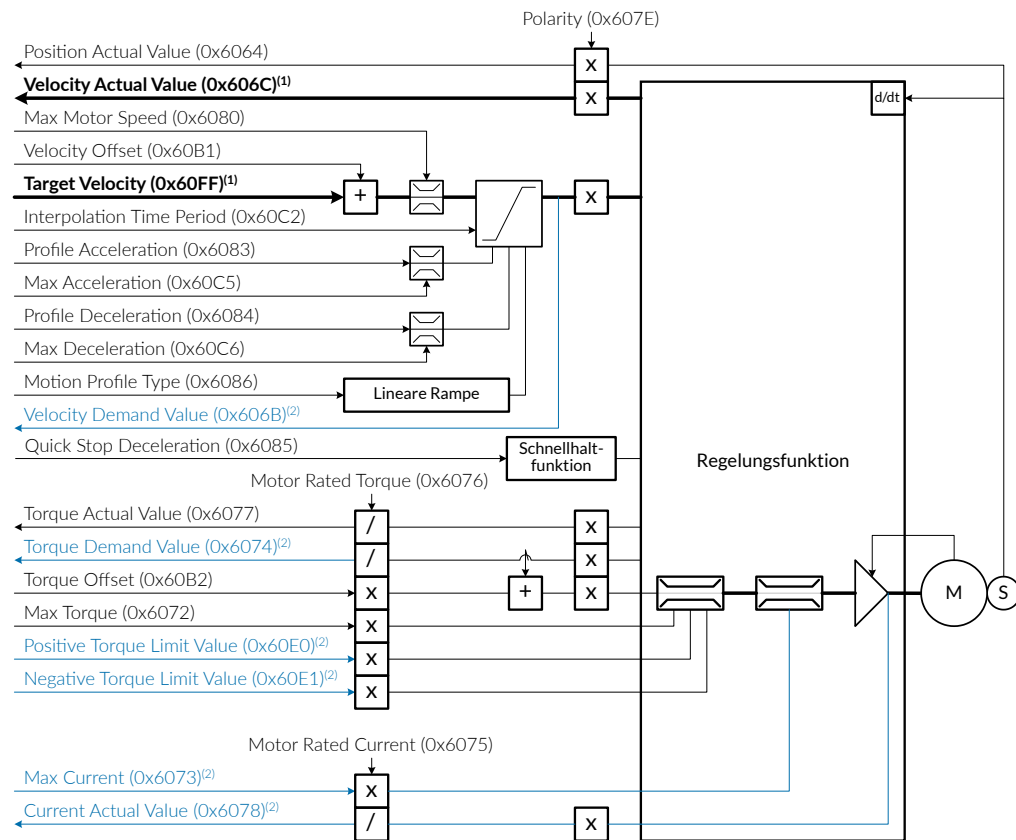


Abb. 2: Blockdiagramm für CSV Mode

- (1) Die fett markierten Objekte sind die notwendigen und empfohlenen Prozessdaten (PDO) für die Betriebsart CSV-Modus.
- (2) SD4x-Antriebe unterstützen diese (blau gekennzeichneten) Objekte zusätzlich zu den Standardobjekten im DS402-Antriebsprofil.

Die Objekte **Quick Stop Deceleration** oder **Profile Deceleration** werden verwendet, wenn das Anhalten per Schnellhaltrampe oder Bremsrampe in einem Option Code definiert ist.

Der **Torque Offset** kann zur Drehmomentvorsteuerung verwendet werden. Zusätzlich bleiben die eingestellten Drehmoment-, Strom- und Beschleunigungsgrenzen aktiv.

Aus den Objekten **Interpolation Time Period** und **Target Velocity** ergibt sich automatisch die Geschwindigkeitsänderung bis zur nächsten Stützstelle einer Trajektorienberechnung.

CSV-spezifische Bits im Statusword

Die betriebsartspezifischen Bits im Statusword sind folgendermaßen belegt:

- Bit 12 (Antrieb folgt Sollwert) ist aktiv, solange der Antrieb der Vorgabetrajektorie folgt.

2.4.2 Profile Velocity Mode (PV)

Für die Steuerung des Antriebs in der Betriebsart Profile Velocity Mode sind mindestens die Objekte **Controlword**, **Statusword** und **Target Velocity** notwendig. Als Rückgabewert empfiehlt sich, **Velocity Actual Value** zu verwenden. Wenn der Motor keinen Drehzahlsensor hat, errechnet der Antrieb die Istgeschwindigkeit aus dem aktuell an den Motor gegebenen Drehfeld.

Das folgende Diagramm zeigt die verwendeten Objekte und beispielhaft deren Funktion:

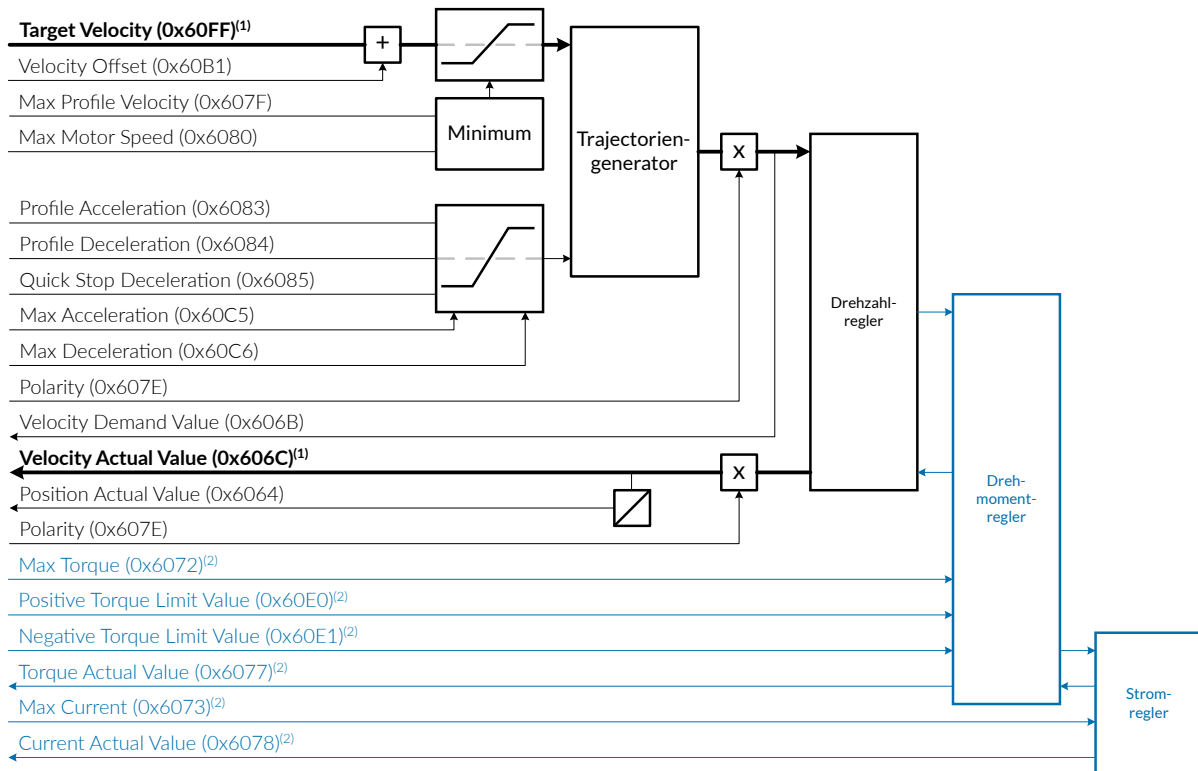


Abb. 3: Blockdiagramm für Profile Velocity Mode (Geschwindigkeitsmodus PV)

- (1) Die fett markierten Objekte sind die notwendigen und empfohlenen Prozessdaten (PDO) für die Betriebsart Profile Velocity Mode.
- (2) SD4x-Antriebe unterstützen diese (blau gekennzeichneten) Objekte zusätzlich zu den Standardobjekten im DS402-Antriebsprofil.

PV-spezifische Bits im Controlword und Statusword

Die betriebsartspezifischen Bits 4, 5 und 6 im Controlword sind optional und werden in dieser Betriebsart nicht verwendet.

Die betriebsartspezifischen Bits im Statusword sind folgendermaßen belegt:

- ▶ Bit 12 (Drehzahl Null) wird inaktiv, wenn der Geschwindigkeitswert die Geschwindigkeitsschwelle länger als die Geschwindigkeitsschwellenzeit überschreitet. Unterhalb dieser Schwelle ist das Bit aktiv und zeigt an, dass die Achse stillsteht. Das Geschwindigkeitsfenster stellen Sie in der *drivemaster4*-Software unter „Meldungen → Meldungen → Geschwindigkeit Null [M15]“ ein.
- ▶ Bit 13 (maximaler Schlupf) wird aktiv, wenn der Antrieb den parametrisierten maximalen Schlupf erreicht.

2.4.3 Velocity Mode (VL)

Der Velocity Mode verwendet eigene Objekte zur Skalierung der Soll- und Istwerte. In der Standardeinstellung werden Drehzahlen in *Umdrehungen pro Minute* [1/min] angegeben. Sie können die Einheit mit dem Skalierungsobjekt **vl Dimension Factor** ändern. Zusätzlich können die Soll- und Istgeschwindigkeit mit dem Objekt **vl Set-point Factor** skaliert werden. Die Auflösung der Soll- und Istgeschwindigkeit ist in dieser Betriebsart jedoch durch die Objektgröße auf 16 Bit begrenzt. Für Anwendungen, die eine höhere Auflösung benötigen, steht die Betriebsart [Profile Velocity Mode \(PV\), Seite 16](#) zur Verfügung. Diese bietet eine 32-Bit-Auflösung für Soll- und Istgeschwindigkeiten.

Für die Steuerung des Antriebs in der Betriebsart Velocity Mode sind mindestens die Objekte **Controlword**, **Statusword** und **vl Target Velocity** notwendig. Als Rückgabewert empfiehlt sich, **vl Velocity Actual Value** zu verwenden.

Das folgende Diagramm zeigt die verwendeten Objekte und beispielhaft deren Funktion:

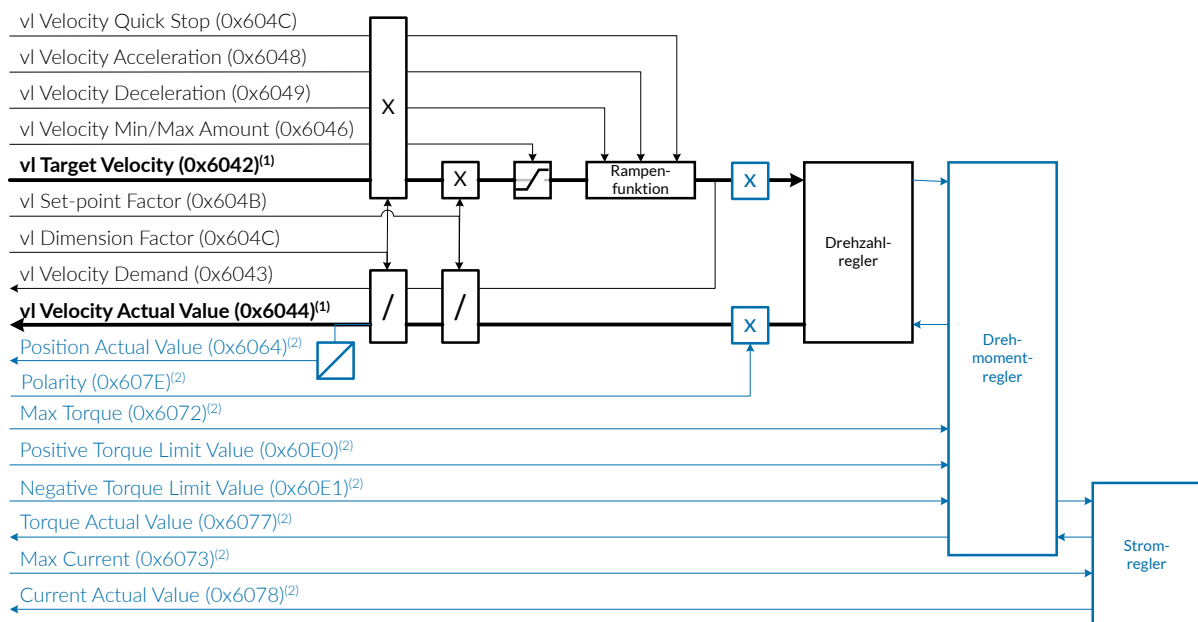


Abb. 4: Blockdiagramm für Velocity Mode (Geschwindigkeitsmodus VL)

- (1) Die fett markierten Objekte sind die notwendigen und empfohlenen Prozessdaten (PDO) für die Betriebsart Velocity Mode.
- (2) SD4x-Antriebe unterstützen diese (blau gekennzeichneten) Objekte zusätzlich zu den Standardobjekten im DS402-Antriebsprofil.

VL-spezifische Bits im Controlword und Statusword

Die betriebsartsspezifischen Bits 4, 5 und 6 im Controlword sind optional und werden von SD4x-Antrieben nicht unterstützt.

Die betriebsartsspezifischen Bits im Statusword sind laut DS402-Antriebsprofil im Velocity Mode reserviert. SD4x-Antriebe verwenden in dieser Betriebsart dennoch das Bit 12 wie folgt:

- Bit 12 (Drehzahl Null) wird inaktiv, wenn der Geschwindigkeitsistwert die Geschwindigkeitsschwelle länger als die Geschwindigkeitsschwellenzeit überschreitet. Unterhalb dieser Schwelle ist das Bit aktiv und zeigt an, dass die Achse stillsteht. Das Geschwindigkeitsfenster stellen Sie in der *drivemaster4*-Software unter „Meldungen → Meldungen → Geschwindigkeit Null [M15]“ ein.

2.4.3.1 Soll- und Istwerte im Velocity Mode

vl Target Velocity (0x6042)

Dieses Objekt enthält die Zieldrehzahl im Velocity Mode.

vl Target Velocity		Zieldrehzahl im Velocity Mode Wenn die Objekte vl Set-point Factor (0x604B) und vl Dimension Factor (0x604C) auf den Wert 1 (Default) gesetzt sind, gibt das Objekt den Drehzahlwert in 1/min an. Positive Werte führen zur Vorwärtsbewegung, negative Werte zur Rückwärtsbewegung.	
DS402_0x6042_VL_TARGET_VELOCITY			ID: 0x6042
RW	Volatile	Einheit: 1/min (Default)	l16

vl Velocity Demand (0x6043)

Dieses Objekt gibt den resultierenden Drehzahlsollwert im Velocity Mode nach Rampen- und Begrenzungsfunktion zurück.

vl Velocity Demand		Drehzahlsollwert im Velocity Mode Der Wert gibt den Drehzahlsollwert nach der Rampenfunktion und der Begrenzungsfunktion an. Einheit und Richtung entsprechen Objekt vl Target Velocity (0x6042).	
DS402_0x6043_VL_VELOCITY_DEMAND			ID: 0x6043
RO	Volatile	Einheit: 1/min (Default)	l16

vl Velocity Actual Value (0x6044)

Dieses Objekt gibt den Drehzahlwert im Velocity Mode zurück.

vl Velocity Actual Value		Drehzahlwert im Velocity Mode Einheit und Richtung entsprechen Objekt vl Target Velocity (0x6042).	
DS402_0x6044_VL_VELOCITY_ACTUAL_VALUE			ID: 0x6044
RO	Volatile	Einheit: 1/min (Default)	l16

Abhängig von der Applikation (mit oder ohne Sensor), wird der Drehzahlwert entweder von einem Sensor oder vom berechneten Drehfeld abgeleitet.

vl Velocity Min Max Amount (0x6046)

Über dieses Objekt kann eine Begrenzung der Drehzahl im Velocity Mode eingestellt werden.

vl Velocity Min Max Amount		Drehzahlbegrenzung im Velocity Mode Sub-IDs: ▶ 0x6046:00 = Highest subindex supported (2) ▶ 0x6046:01 = vl Velocity Min Amount ▶ 0x6046:02 = vl Velocity Max Amount Wenn das Objekt vl Dimension Factor (0x604C) auf den Wert 1 (Default) gesetzt ist, gibt das Objekt die Drehzahlbegrenzungen in 1/min an.	
DS402_0x6046_VL_VELOCITY_MIN_MAX_AMOUNT			ID: 0x6046
RW	Volatile	Einheit: 1/min (Default)	BYTEARRAY

Dieses Objekt enthält im Subindex 0 den letzten Subindex des Objekts, in diesem Fall 2.

Die folgende Darstellung zeigt die Begrenzung der minimalen und der maximalen Drehzahl über das Objekt **vl Velocity Min Max Amount**. Die punktierte Linie zeigt die Drehzahlvorgabe und die blaue, durchgezogene Linie zeigt die resultierende Solldrehzahl.

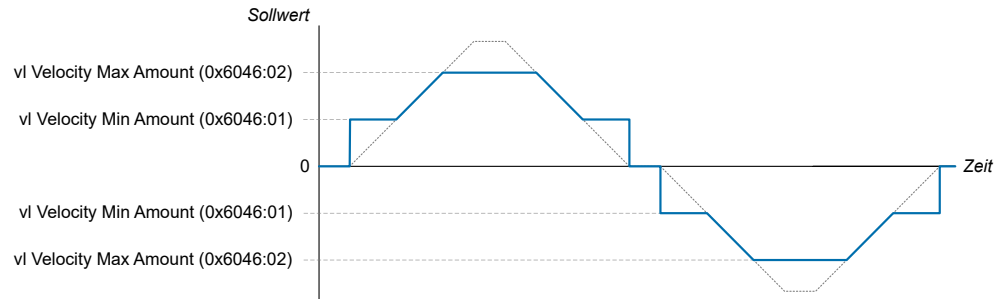


Abb. 5: Drehzahlbegrenzung im Velocity Mode

vl Velocity Acceleration (0x6048)

Über dieses Objekt kann eine Begrenzung der Beschleunigung im Velocity Mode eingestellt werden.

vl Velocity Acceleration		Beschleunigungsrampe im Velocity Mode Sub-IDs: ▶ 0x6048:00 = Highest subindex supported (2) ▶ 0x6048:01 = Delta Speed Wenn das Objekt vl Dimension Factor (0x604C) auf den Wert 1 (Default) gesetzt ist, gibt das Objekt die Drehzahl in 1/min an. ▶ 0x6048:02 = Delta Time	
DS402_0x6048_VL_VELOCITY_ACCELERATION			ID: 0x6048
RW	Volatile	Einheit: 1/min / s (Default)	DS402_ACCERLERATION_REC

Dieses Objekt enthält im Subindex 0 den letzten Subindex des Objekts, in diesem Fall 2.

Die folgende Darstellung zeigt die Begrenzung der Beschleunigung über das Objekt **vl Velocity Acceleration**. Die punktierte Linie zeigt die Drehzahlvorgabe und die blaue, durchgezogene Linie zeigt die resultierende Solldrehzahl.

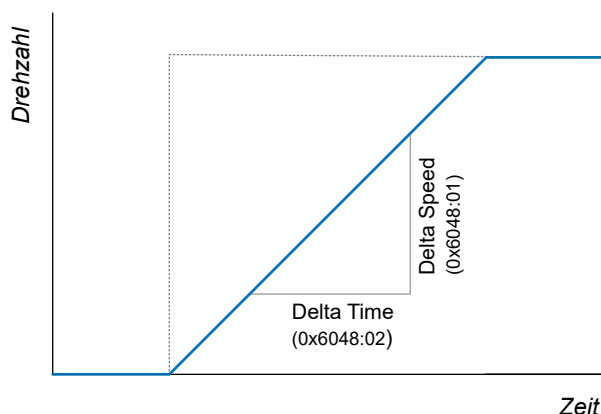


Abb. 6: Beschleunigungsbegrenzung im Velocity Mode

Somit wird die resultierende Beschleunigungsrampe in den Standardeinstellungen in **Umdrehungen pro Minute pro Sekunde** angegeben.

vl Velocity Deceleration (0x6049)

Über dieses Objekt kann eine Begrenzung der Bremsbeschleunigung im Velocity Mode eingestellt werden.

vl Velocity Deceleration		Bremsrampe im Velocity Mode Sub-IDs: ▶ 0x6049:00 = Highest subindex supported (2) ▶ 0x6049:01 = Delta Speed Wenn das Objekt vl Dimension Factor (0x604C) auf den Wert 1 (Default) gesetzt ist, gibt das Objekt die Drehzahl in 1/min an. ▶ 0x6049:02 = Delta Time Die Zeit wird in s angegeben. Der Wert 0 ist nicht erlaubt, in diesem Fall wird der Minimalwert (1 s) verwendet.	
DS402_0x6049_VL_VELOCITY_DECELERATION		ID: 0x6049	
RW	Volatile	Einheit: 1/min / s (Default)	DS402_ACCERLERATION_REC

Dieses Objekt enthält im Subindex 0 den letzten Subindex des Objekts, in diesem Fall 2.

Die folgende Darstellung zeigt die Begrenzung der Bremsbeschleunigung über das Objekt **vl Velocity Deceleration**. Die punktierte Linie zeigt die Drehzahlvorgabe und die blaue, durchgezogene Linie zeigt die resultierende Solldrehzahl.

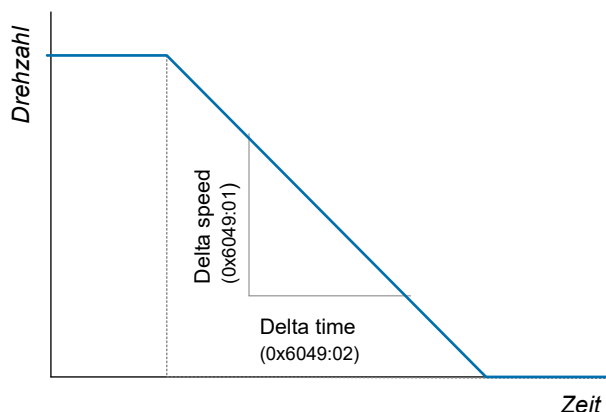


Abb. 7: Bremsbeschleunigungsbegrenzung im Velocity Mode

Somit wird die resultierende Verzögerungsrampe in den Standardeinstellungen in *Umdrehungen pro Minute pro Sekunde* angegeben.

vl Velocity Quick Stop (0x604A)

Über dieses Objekt kann eine Begrenzung der Bremsbeschleunigung bei einem Schnellhalt im Velocity Mode eingestellt werden.

vl Velocity Quick Stop		Schnellhalt rampe im Velocity Mode Sub-IDs: ▶ 0x604A:00 = Highest subindex supported (2) ▶ 0x604A:01 = Delta Speed Wenn das Objekt vl Dimension Factor (0x604C) auf den Wert 1 (Default) gesetzt ist, gibt das Objekt die Drehzahl in 1/min an. ▶ 0x604A:02 = Delta Time Die Zeit wird in s angegeben. Der Wert 0 ist nicht erlaubt, in diesem Fall wird der Minimalwert (1 s) verwendet.	
DS402_0x604A_VL_VELOCITY_QUICK_STOP		ID: 0x604A	
RW	Volatile	Einheit: 1/min / s (Default)	DS402_ACCERLERATION_REC

Dieses Objekt enthält im Subindex 0 den letzten Subindex des Objekts, in diesem Fall 2.

Die folgende Darstellung zeigt die Begrenzung der Bremsbeschleunigung bei einem Schnellhalt über das Objekt **vl Velocity Quick Stop**. Die punktierte Linie zeigt die Drehzahlvorgabe und die blaue, durchgezogene Linie zeigt die resultierende Solldrehzahl.

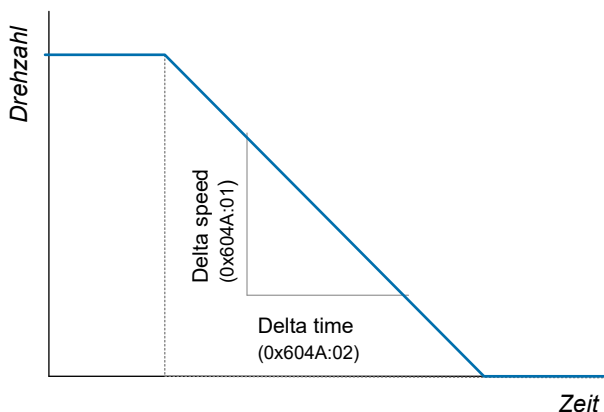


Abb. 8: Bremsbeschleunigungsbegrenzung bei Schnellhalt im Velocity Mode

Somit wird die resultierende Schnellhalttrampe in den Standardeinstellungen in *Umdrehungen pro Minute pro Sekunde* angegeben.

vl Set-point Factor (0x604B)

Über dieses Objekt kann ein zusätzlicher Skalierungsfaktor für die Drehzahlsollwerte und den Drehzahlwert im Velocity Mode eingestellt werden.

vl Set-point Factor		Skalierungsfaktor für Soll- und Istwert im Velocity Mode Sub-IDs: ▶ 0x604B:00 = Highest subindex supported (2) ▶ 0x604B:01 = Numerator Der Zähler hat keine Einheit. Der Wert 0 ist nicht erlaubt, in diesem Fall bleibt der letzte gültige Wert erhalten. ▶ 0x604B:02 = Denominator Der Nenner hat keine Einheit. Der Wert 0 ist nicht erlaubt, in diesem Fall bleibt der letzte gültige Wert erhalten.	
DS402_0X604B_VL_SET_POINT_FACTOR		ID: 0x604B	
RW	Volatile	Einheit: –	DS402_ACCERLERATION_REC

Dieses Objekt enthält im Subindex 0 den letzten Subindex des Objekts, in diesem Fall 2.

Diese Einheitenskalierung betrifft ausschließlich **vl Target Velocity** (0x6042), **vl Velocity Demand** (0x6043) und **vl Velocity Actual Value** (0x6044), also nicht die Rampen- und Begrenzungsfunktionen.

$$\text{Skalierungsfaktor der Sollwerte} = \frac{\text{Numerator (0x604B:01)}}{\text{Denominator (0x604B:02)}}$$

Anwendungsbeispiele

- Die Drehzahl soll in *Umdrehungen pro Minute* mit einer Nachkommastelle vorgegeben werden:
 - Numerator bleibt auf 1 gesetzt.
 - Denominator wird auf 10 gesetzt.

Die Werte in den Objekten **vl Target Velocity**, **vl Velocity Demand** und **vl Velocity Actual Value** werden sofort neu interpretiert: Die Vorgabe 101 bedeutet nun 10,1 1/min.

Hinweis

Mit dieser Einstellung sind nur Drehzahlvorgaben/-anzeigen zwischen –3276,8 und 3276,7 1/min möglich.

2. Es sollen Drehzahlen größer 32767 1/min vorgegeben werden:

- Numerator wird auf 10 gesetzt.
- Denominator bleibt auf 1 gesetzt.

Die Werte in den Objekten **vl Target Velocity**, **vl Velocity Demand** und **vl Velocity Actual Value** werden sofort neu interpretiert: Die Vorgabe 101 bedeutet nun 1010 1/min.

Hinweis

Mit dieser Einstellung sind Drehzahlvorgaben/-anzeigen zwischen –327680 und 327670 1/min möglich. Die Soll- und Istdrehzahlen können jedoch nur noch in 1/min-Zehnerschritten angegeben/dargestellt werden.

vl Dimension Factor (0x604C)

Über dieses Objekt werden alle Drehzahlwerte im Velocity Mode skaliert. Dies betrifft neben den Soll- und Istwerten auch die Rampen- und Begrenzungsfunktionen.

vl Dimension Factor		Skalierungsfaktor für die Drehzahleinheit im Velocity Mode Sub-IDs: ▶ 0x604C:00 = Highest subindex supported (2) ▶ 0x604C:01 = Numerator Der Zähler hat keine Einheit. Der Wert 0 ist nicht erlaubt, in diesem Fall bleibt der letzte gültige Wert erhalten. ▶ 0x604C:02 = Denominator Der Nenner hat keine Einheit. Der Wert 0 ist nicht erlaubt, in diesem Fall bleibt der letzte gültige Wert erhalten.	
DS402_0X604C_VL_DIMENSION_FACTOR		ID: 0x604C	
RW	Volatile	Einheit: –	DS402_ACCELERATION_REC

Dieses Objekt enthält im Subindex 0 den letzten Subindex des Objekts, in diesem Fall 2.

Ziel dieser Skalierung: Jede benutzerspezifische Geschwindigkeit besteht aus bestimmten Einheiten bezogen auf Zeiteinheiten (z. B. 1/s, Nocken/min, m/s usw.). Der Dimensionsfaktor rechnet diese Geschwindigkeitseinheiten in eine gemeinsame Basis um (z. B. in Umdrehungen pro Minute).

$$\text{Skalierungsfaktor zu 1/min} = \frac{\text{Numerator (0x604C:01)}}{\text{Denominator (0x604C:02)}}$$

Bei einer Änderung des Skalierungsfaktors, werden die Rampen- und Begrenzungsfunktionen automatisch auf die neue Einheit skaliert. Das betrifft folgende Objekte: vl Velocity Min Amount (0x6046:01), vl Velocity Max Amount (0x6046:02), vl Velocity Acceleration (0x6048:01), vl Velocity Deceleration (0x6049:01) und vl Velocity Quick Stop (0x604A:1). Wenn eines dieser Objekte als PDO mit einer übergeordneten Steuerung ausgetauscht wird, muss die Steuerung die Skalierungsarbeit für das entsprechende Objekt übernehmen.

Anwendungsbeispiel

Die Drehzahl soll in *Umdrehungen pro Sekunde* mit einer Nachkommastelle vorgegeben werden:

- Numerator wird auf 60 gesetzt.
- Denominator wird auf 10 gesetzt.

Die Werte in den Objekten **vi Target Velocity**, **vi Velocity Demand** und **vi Velocity Actual Value** werden sofort neu interpretiert: Die Vorgabe 32 bedeutet nun 3,2 1/s, also 192 1/min.

Hinweis

Mit dieser Einstellung sind Drehzahlvorgaben/-anzeigen zwischen -196608 und 196602 1/min möglich. Die Soll- und Ist-drehzahlen können jedoch nur noch in 1/min-Sechsserschritten angegeben/dargestellt werden.

2.4.4 Torque Mode

Für die Steuerung des Antriebs in der Betriebsart Torque Mode sind mindestens die Objekte **Controlword**, **Statusword** und **Target Torque** notwendig. Als Rückgabewert empfiehlt sich, **Torque Actual Value** zu verwenden.

Das folgende Diagramm zeigt die verwendeten Objekte und beispielhaft deren Funktion:

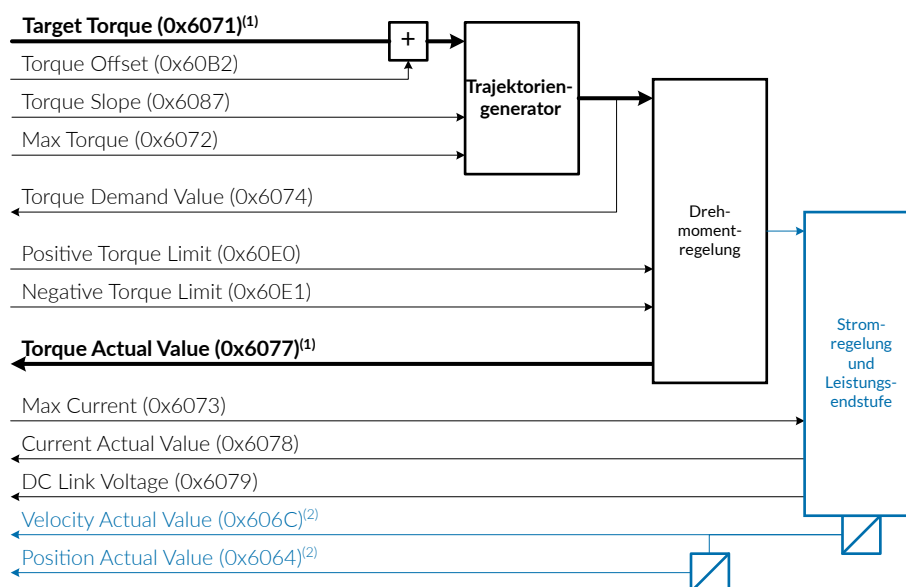


Abb. 9: Blockdiagramm für Torque Mode (Drehmomentbetrieb)

- (1) Die fett markierten Objekte sind die notwendigen und empfohlenen Prozessdaten (PDO) für die Betriebsart Profile Velocity Mode.
- (2) SD4x-Antriebe unterstützen diese (blau gekennzeichneten) Objekte zusätzlich zu den Standardobjekten im DS402-Antriebsprofil.

2.4.5 Betriebsart wechseln

Der Betriebsartwechsel erlaubt das einfache Umschalten zwischen den verfügbaren Betriebsarten des Antriebs.

Hinweis

Alternativ können Sie bei SD4x-Antrieben den kompletten Parametersatz wechseln (siehe [Kapitel 2.6 „Parametersatzumschaltung“, Seite 37](#)). Diese Option sollten Sie in Betracht ziehen, wenn mehrere Parameter neben der Betriebsart geändert werden sollen.

⚠️ WARNUNG



Verletzungsgefahr durch ungewollten Anlauf des Motors

→ Setzen Sie die Sollwerte vor dem Wechsel der Betriebsart auf 0, um ein ungewolltes Anlaufen des Motors zu verhindern.

Die verfügbaren Betriebsarten werden im Objekt **Supported Drive Modes** (0x6502) angezeigt. Sie wechseln die Betriebsart mithilfe des Objekts **Mode of Operation** (0x6060). Die aktive Betriebsart wird im Objekt **Mode of Operation Display** (0x6061) zurückgegeben.

Nach erfolgreichem Betriebsartwechsel wird im **Statusword** (0x6041) das Bit 10 (Zielwert erreicht) gesetzt.

Die neue Betriebsart wird erst im Zustand „Operation Enabled“ aktiv und übernommen. In diesem Zustand kann die Betriebsart nicht wieder gewechselt werden.

Eine neue Betriebsartvorgabe wird erst beim Wechsel in den Zustand „Operation Enabled“ übernommen. Befindet sich die Zustandsmaschine bereits in diesem Zustand, muss „Operation Enabled“ zunächst verlassen und dann wieder eingenommen werden.

Supported Drive Modes (0x6502)

Dieses Objekt enthält die aktuell verfügbaren Betriebsarten des Antriebs als Bitcode. Die gesetzten Bits in dem 32-Bit-Objekt zeigen an, dass die entsprechenden Betriebsarten verfügbar sind.

Supported Drive Modes		Verfügbare Betriebsarten: Bit 00: pp – Profile Position Mode Bit 01: vl – Velocity Mode Bit 02: pv – Profile Velocity Mode Bit 03: tq – Profile Torque Mode Bit 04: hmaf – Homing Mode with Additional Feedback Bit 05: hm – Homing Mode Bit 06: ip – Interpolated Position Mode Bit 07: csp – Cyclic Sync Position Mode Bit 08: csv – Cyclic Sync Velocity Mode Bit 09: cst – Cyclic Sync Torque Mode Bit 10: cstca – Cyclic Sync Torque Mode with Commutation Angle Bit 11: ppaf – Profile Position Mode with Additional Feedback Bit 12: pvaf – Profile Velocity Mode with Additional Feedback Bit 13: cspaf – Cyclic Sync Position Mode with Additional Feedback Bit 14: csvaf – Cyclic Sync Velocity Mode with Additional Feedback Bit 15: r – Reserviert	
DS402_0X6502_SUPPORTED_DRIVE_MODES		ID: 0x6502	
RO	Volatile	Einheit: –	U32

Mode of Operation (0x6060)

Über dieses Objekt kann die Betriebsart gewechselt werden.

Mode of Operation		Auswahl der Betriebsart: 01: Profile Position Mode 02: Velocity Mode 03: Profile Velocity Mode 04: Profile Torque Mode 06: Homing Mode 07: Interpolated Position Mode 08: Cyclic Sync Position Mode 09: Cyclic Sync Velocity Mode 10: Cyclic Sync Torque Mode 11: Cyclic Sync Torque Mode with Commutation Angle 12: Profile Position Mode with Additional Feedback 13: Profile Velocity Mode with Additional Feedback 14: Homing Mode with Additional Feedback 15: Cyclic Sync Position Mode with Additional Feedback 16: Cyclic Sync Velocity Mode with Additional Feedback	
DS402_0x6060_MODE_OF_OPERATION			ID: 0x6060
RW	Volatile	Einheit: –	U8

Nicht alle Betriebsarten stehen zur Verfügung. Standardmäßig unterstützt der SD4x-Antrieb die Betriebsarten 2 (Velocity Mode), 3 (Profile Velocity Mode) und 4 (Profile Torque Mode).

Mode of Operation Display (0x6061)

Dieses Objekt gibt die aktuell aktive Betriebsart zurück.

Mode of Operation Display		Aktive Betriebsart: 01: Profile Position Mode 02: Velocity Mode 03: Profile Velocity Mode 04: Profile Torque Mode 06: Homing Mode 07: Interpolated Position Mode 08: Cyclic Sync Position Mode 09: Cyclic Sync Velocity Mode 10: Cyclic Sync Torque Mode 11: Cyclic Sync Torque Mode with Commutation Angle 12: Profile Position Mode with Additional Feedback 13: Profile Velocity Mode with Additional Feedback 14: Homing Mode with Additional Feedback 15: Cyclic Sync Position Mode with Additional Feedback 16: Cyclic Sync Velocity Mode with Additional Feedback	
DS402_0x6061_MODE_OF_OPERATION_DISPLAY			ID: 0x6061
RO	Volatile	Einheit: –	U8

2.5 Soll- und Istwerte

Im Folgenden sind die Soll- und Istwerte beschrieben, die für das Antriebsprofil (DS402) relevant sind. Die Verwendung der Werte wird durch die gewählte Betriebsart definiert

Hinweis

Der Velocity Mode (VL) verwendet zum großen Teil eigene Soll-, Ist-, Skalierungs- und Begrenzungswerte. Diese sind direkt bei der Betriebsart beschrieben, siehe [Seite 18](#).

2.5.1 Allgemeine Sollwerte

Target Velocity (0x60FF)

Dieses Objekt gibt die Zielgeschwindigkeit an. Dieser Zielwert wird normalerweise am Ende der Beschleunigungsrampe während einer Profilbewegung erreicht. Bei Positionierungsbetriebsarten ist der Zielwert für beide Bewegungsrichtungen gültig.

Target Velocity		Zielgeschwindigkeit in Anwandereinheiten	
DS402_0x60FF_TARGET_VELOCITY			ID: 0x60FF
RW	Volatile	Einheit: –	I32

Der Wert wird in benutzerdefinierten Geschwindigkeitseinheiten angegeben. Die Standardeinstellung für SD4x-Antriebe ist 10^{-3} 1/min . Die Geschwindigkeitseinheit wird mit der Faktorengruppe **SI Unit Velocity** (0x60A9) skaliert. Außerdem kann der resultierende Sollwert durch das Objekt **Velocity Offset** (0x60B1) beeinflusst werden. Geschwindigkeits- und Beschleunigungsgrenzen können Sie zusätzlich festlegen.

Standardmäßig sollte das Antriebssystem so eingestellt werden, dass positive Vorgaben zu einer Vorwärtsbewegung und negative Vorgaben zu einer Rückwärtsbewegung führen.

Target Torque (0x6071)

Dieses Objekt gibt den Drehmomentsollwert an.

Target Torque		Drehmomentsollwert Der Wert wird in Promille (1000 = 100 %) des Nenndrehmoments, Objekt Motor Rated Torque (0x6076), angegeben.	
DS402_0x6071_TARGET_TORQUE			ID: 0x6071
RW	Volatile	Einheit: ‰	I16

Für den Drehmomentsollwert kann über das Objekt **Torque Offset** (0x60B2) ein Offset eingestellt werden. Begrenzungen werden über die Objekte **Torque Slope** (0x6087), **Positive Torque Limit** (0x60E0) und **Negative Torque Limit** (0x60E1) eingestellt.

Interpolation Time Period (0x60C2)

Dieses Objekt definiert den Zeitraum zwischen zwei Aktualisierungen des Sollwertes.

Interpolation Time Period		Interpolationszeitraum Sub-IDs: ▶ 0x60C2:00 [U8] = Highest Subindex supported (2) ▶ 0x60C2:01 [U8] = Interpolation Time Period Value (1) ▶ 0x60C2:02 [I8] = Interpolation Time Index (-3)	
DS402_0x60C2_INTERPOLATION_TIME_PERIOD			ID: 0x60C2
RW	Volatile	Einheit: –	BYTEARRAY

Dieses Objekt enthält im Subindex 0 den letzten Subindex des Objekts, in diesem Fall 2.

Der Wert für den Interpolationszeitraum <Subindex1> wird in der Einheit $10^{\text{hoch } \text{<Subindex2>}} \text{ Sekunden}$ vorgegeben (Default = $1 \times 10^{-3} \text{ s} = 1 \text{ ms}$). Ist die resultierende Zeit T_{itp} kleiner als die Buszykluszeit, wird der Sollwert zum Zeitpunkt T_{itp} erreicht (sofern nicht anders begrenzt). In Betriebsarten mit einem Sollwertpuffer, wird nach Ablauf der Interpolationszeit der nächste Wert aus dem Puffer verwendet.

Wenn eine zyklisch synchrone Betriebsart aktiv ist, sollten Sie dieses Objekt auf die aktuelle Feldbuszykluszeit einstellen. Anderenfalls weicht die Interpolationszeit von der Feldbuszykluszeit ab und es entstehen Sollwertsprünge. Wenn die Interpolation abgeschlossen ist bevor über den Feldbus ein neuer Sollwert zur Verfügung steht, wird der Endwert der Interpolation bis zur nächsten Aktualisierung gehalten (siehe Abbildung rechts).

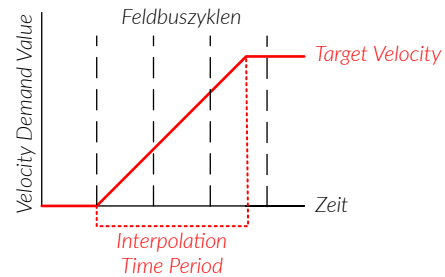


Abb. 10: Interpolation ≠ Feldbuszykluszeit

Die folgenden Abbildungen zeigen beispielhaft eine Interpolation (Velocity Demand Value in blau) eines Sollwertes (Target Value in rot):

Interpolation Time Period richtig eingestellt:

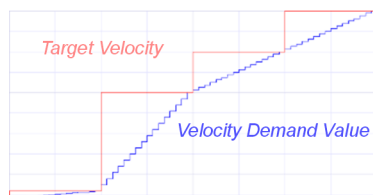


Abb. 11: Keine Sollwertsprünge

Interpolation Time Period falsch eingestellt:

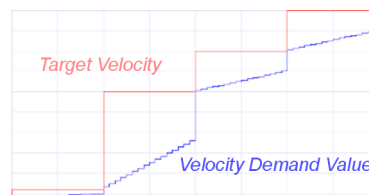


Abb. 12: Sollwertsprünge

Motion Profile Type (0x6086)

Dieses Objekt gibt die Beschleunigungs- und Bremsrampenform an.

Motion Profile Type		Bewegungsprofiltyp: 0: Lineare Rampe 1: Sin ² -Rampe 2: Ruckfreie Rampe 3: Ruckbegrenzte Rampe	
DS402_0X6086_MOTION_PROFILE_TYPE			ID: 0x6086
RW	Volatile	Einheit: –	I16

SD4x-Antriebe unterstützen zurzeit nur die lineare Rampenform. Somit ist dieses Objekt fest auf 0 eingestellt.

2.5.2 Allgemeine Istwerte

Position Actual Value (0x6064)

Dieses Objekt gibt die Istposition zurück, die vom Lagemesssystem (z. B. Encoder) ermittelt wurde.

Position Actual Value		Istposition in Anwandereinheiten	
DS402_0X6064_POSITION_ACTUAL_VALUE			ID: 0x6064
RO	Volatile	Einheit: –	I32

Standardmäßig wird der Wert in (Encoder-)Inkrementen angegeben, siehe Objekt **SI Unit Position** (0x60A8). Die Anzahl der Inkremente pro Umdrehung wird in der *drive-master4*-Software bei den Feedback-Einstellungen festgelegt.

Velocity Demand Value (0x606B)

Dieses Objekt gibt den resultierenden Drehzahlsollwert nach Rampen- und Begrenzungsfunktion zurück.

Velocity Demand Value		Drehzahlsollwert nach der Rampenfunktion und Begrenzungsfunktion Einheit und Richtung entsprechen Objekt Target Velocity (0x60FF).	
DS402_0x606B_VELOCITY_DEMAND_VALUE			ID: 0x606B
RO	Volatile	Einheit: –	I32

Velocity Actual Value (0x606C)

Dieses Objekt gibt die aktuelle Drehzahl zurück.

Velocity Actual Value		Drehzahlwert in Anwindereinheiten Einheit und Richtung entsprechen Objekt Target Velocity (0x60FF).	
DS402_0x606C_VELOCITY_ACTUAL_VALUE			ID: 0x606C
RO	Volatile	Einheit: –	I32

Abhängig von der Applikation (mit oder ohne Sensor), wird der Drehzahlwert entweder von einem Sensor oder vom berechneten Drehfeld abgeleitet.

Torque Demand (0x6074)

Dieses Objekt gibt den resultierenden Drehmomentsollwert nach Rampen- und Begrenzungsfunktion zurück.

Torque Demand		aktueller Drehmomentsollwert nach Rampen- und Begrenzungsfunktion Der Wert wird in Promille (1000 = 100 %) des Nenndrehmoments, Objekt Motor Rated Torque (0x6076), angegeben.	
DS402_0x6074_TORQUE_DEMAND			ID: 0x6074
RO	Volatile	Einheit: ‰	I16

Torque Actual Value (0x6077)

Dieses Objekt gibt das aktuelle Drehmoment zurück.

Torque Actual Value		Drehmomentwert Einheit und Richtung entsprechen Objekt Target Torque (0x6071).	
DS402_0x6077_TORQUE_ACTUAL_VALUE			ID: 0x6077
RO	Volatile	Einheit: ‰	I16

Current Actual Value (0x6078)

Dieses Objekt gibt den aktuellen Strom zurück.

Current Actual Value		Stromwert Der Wert wird in Promille (1000 = 100 %) des Nennstroms, Objekt Motor Rated Current (0x6075), angegeben.	
DS402_0x6078_CURRENT_ACTUAL_VALUE			ID: 0x6078
RO	Volatile	Einheit: ‰	I16

DC Link Circuit Voltage (0x6079)

Dieses Objekt gibt die aktuell anliegende Zwischenkreisspannung zurück.

DC Link Circuit Voltage		Zwischenkreisspannung	
DS402_0x6079_DC_LINK_CIRCUIT_VOLTAGE			ID: 0x6079
RO	Volatile	Einheit: mV	U32

2.5.3 Motorparameter

Motor Rated Current (0x6075)

Dieses Objekt enthält den konfigurierten Motornennstrom. Der Wert sollte der Angabe auf dem Motortypenschild entsprechen.

Motor Rated Current		Nennstrom des Motors Alle relativen Stromangaben beziehen sich auf diesen Wert.	
DS402_0x6075_MOTOR_RATED_CURRENT			ID: 0x6075
RW	Volatile	Einheit: mA	U32

Motor Rated Torque (0x6076)

Dieses Objekt enthält das konfigurierte Motornenndrehmoment. Der Wert sollte der Angabe auf dem Motortypenschild entsprechen.

Motor Rated Torque		Nennndrehmoment des Motors Alle relativen Drehmomentangaben beziehen sich auf diesen Wert.	
DS402_0x6076_MOTOR_RATED_TORQUE			ID: 0x6076
RW	Volatile	Einheit rotativer Motor: mNm Einheit Linearmotor: Nm	U32

Motor Type (0x6402)

Über dieses Objekt kann der Motortyp ausgewählt werden.

Motor Type		Auswahl des Motortyps: 0x0000: Non-standard Motor 0x0001: Phase Modulated DC Motor 0x0002: Frequency Controlled DC Motor 0x0003: PM Synchronous Motor 0x0004: AC Synchronous Sinewave Wound Field 0x0005: AC Synchronous Reluctance Switched 0x0006: AC Asynchronous Induction Polyphase Wound Rotor 0x0007: AC Asynchronous Induction Squirrel Cage 0x0008: AC Synchronous Step 0x0009: Micro-step Stepper Motor 0x000A: AC Synchronous Sinusoidal PM 0x000B: AC Synchronous Brushless PM Trapezoidal 0x000C: AC Synchronous Reluctance Sync 0x000D: DC Commutator PM 0x000E: DC Commutator Wound Field Series 0x000F: DC Commutator Wound Field Shunt 0x0010: DC Commutator Wound Field Compound 0x0011 bis 0x7FFF: No Motor Type Assigned	
DS402_0x6402_MOTORTYPE			ID: 0x6402
RO	Volatile	Einheit: Inkrement	U16

2.5.4 SI-Einheiten

SI (französisch *Système international d'unités*) ist das auf dem internationalen Größensystem basierende Einheitensystem. Die folgenden SI-Einheiten-Objekte enthalten die Grundlage für die Einheiten der Soll- und Istwerte in den Betriebsarten (ausgenommen Velocity Mode (VL), siehe [Soll- und Istwerte im Velocity Mode \(S. 18\)](#)).

Zurzeit stehen die SI-Einheiten-Objekte für SD4x-Antriebe nur lesend zur Verfügung.

Aufbau der SI-Einheiten-Objekte

Die Objekte für SI-Einheiten sind 32 Bit groß und in 4 Teile unterteilt:

Bit-Nummer									
31	...	24	23	...	16	15	...	8	7
Präfix			SI-Numerator (Zähler)			SI-Denominator (Nenner)			Reserviert (00)

Die Dokumente CiA 890 und CiA 402 beschreiben, welche Variablenwerte welchen SI-Einheiten oder Präfixen entsprechen. Eine zusammenfassende Tabelle über mögliche Präfixe und SI-Einheiten aus beiden Normen finden sie im [Anhang \(S. 71\)](#).

Position (0x60A8)

Dieses Objekt enthält die Positionseinheit.

Position		Positionseinheit	
DS402_0X60A8_SI_UNIT_POSITION			ID: 0x60A8
RW	Volatile	Einheit: –	U32

Standardmäßig wird die Position bei SD4x-Antrieben in Inkrementen angegeben. Dies entspricht folgendem Variableninhalt:

Byte	3	2	1	0
Wert	00	B5	00	00
Inhalt	10^0	Inkrement	1	–

Eine Positionsvorgabe von 1244 entspricht somit 1244 Inkrementen.

Velocity (0x60A9)

Dieses Objekt enthält die Geschwindigkeitseinheit.

Velocity		Geschwindigkeitseinheit	
Objekt: DS402_0X60A9_SI_UNIT_VELOCITY			ID: 0x60A9
RW	Volatile	Einheit: –	U32

Standardmäßig wird die Geschwindigkeit bei SD4x-Antrieben in $10^{-3} \times 1/\text{min}$ angegeben. Die Einheit 1/min steht für *Umdrehungen pro Minute* (engl. *revolutions per minute* – *rpm*). Dies entspricht folgendem Variableninhalt:

Byte	3	2	1	0
Wert	FD	B4	47	00
Inhalt	10^{-3}	Umdrehung	Minute	–

Eine Geschwindigkeitsvorgabe von 2500000 entspricht somit 2500 1/min.

Acceleration (0x60AA)

Dieses Objekt enthält die Einheit für die Beschleunigung bzw. Verzögerung.

Acceleration		Beschleunigung-/Verzögerungseinheit	
DS402_0X60AA_SI_UNIT_ACCELERATION			ID: 0x60AA
RW	Volatile	Einheit: –	U32

Standardmäßig wird die Beschleunigung/Verzögerung bei SD4x-Antrieben in $10^{-3} \times 1/s^2$ (*Umdrehungen pro Quadratsekunde*) angegeben. Dies entspricht folgendem Variableninhalt:

Byte	3	2	1	0
Wert	FD	B4	57	00
Inhalt	10^{-3}	Umdrehung	Quadratsekunde	–

Eine Beschleunigungsvorgabe von 300000 entspricht somit $300 \text{ } 1/s^2$.

Jerk (0x60AB)

Dieses Objekt enthält die Einheit für den Ruck.

Jerk		Ruckeinheit	
DS402_0X60AB_SI_UNIT_JERK			ID: 0x60AB
RW	Volatile	Einheit: –	U32

Standardmäßig wird der Ruck bei SD4x-Antrieben in $10^{-3} \times 1/s^3$ (*Umdrehungen pro Kubiksekunde*) angegeben. Dies entspricht folgendem Variableninhalt:

Byte	3	2	1	0
Wert	FD	B4	A0	00
Inhalt	10^{-3}	Umdrehung	Kubiksekunde	–

Eine Ruckvorgabe von 2000 entspricht somit $2 \text{ } 1/s^3$.

2.5.5 Skalierung

In SD4x-Antrieben sind außer **Polarity** keine weiteren Skalierungsobjekte implementiert.

Polarity (0x607E)

Dieses Objekt ermöglicht die Umkehrung der Bewegungsrichtung des Motors.

Polarity		Drehrichtung des Motors	
DS402_0X607E_POLARITY			ID: 0x607E
RW	Volatile	Einheit: –	U8

Ist der Polaritätsparameter nicht gesetzt, führt ein rechtsdrehender Drehmotor zu einer Bewegung in positiver Richtung, ein linksdrehender Drehmotor zu einer Bewegung in negativer Richtung. Bei eingestelltem Polaritätsparameter kehrt sich die Bewegungsrichtung entsprechend um.

Das 8-Bit-Objekt ist wie folgt unterteilt:

Bit-Nummer			
7	6	5 ... 1	0
Position Polarity	Velocity Polarity	Reserviert	Global Polarity

- ▶ Bit 7 gilt für positionsbezogene Betriebsarten (z. B. Profile Position Mode, Interpolated Position Mode, Cyclic Sync Position Mode).
- ▶ Bit 6 gilt für geschwindigkeitsbezogenen Betriebsarten (z. B. Profile Velocity Mode, Cyclic Sync Velocity Mode).
- ▶ Wenn das Bit 0 auf den Wert 1 gesetzt ist, gilt die Invertierung für alle Betriebsarten. In diesem Fall spielen die Werte von Bit 6 und Bit 7 keine Rolle und sollten nicht verwendet werden.

2.5.6 Offsets

Velocity Offset (0x60B1)

Dieses Objekt gibt den Offset für Sollgeschwindigkeiten an.

Velocity Offset		Offset für Sollgeschwindigkeiten Die Einheit entspricht der in Objekt Target Velocity (0x60FF).	
DS402_0x60B1_VELOCITY_OFFSET			ID: 0x60B1
RW	Volatile	Einheit: –	I32

In den positionsbezogenen Betriebsarten enthält das Objekt den Eingangswert für die Geschwindigkeitsvorsteuerung. In den Geschwindigkeitsbetriebsarten enthält es den Geschwindigkeitsoffset des Antriebs. In den Velocity-Betriebsarten (ausgenommen [Velocity Mode \(VL\)](#) (S. 17)) setzt sich die Zielgeschwindigkeit also aus der **Target Velocity** und dem **Velocity Offset** zusammen.

Torque Offset (0x60B2)

Dieses Objekt gibt den Offset für Solldrehmomentvorgaben an.

Torque Offset		Offset für Solldrehmoment Der Wert wird in Promille (1000 = 100 %) des Nenndrehmoments, Objekt Motor Rated Torque (0x6076), angegeben.	
DS402_0x60B2_TORQUE_OFFSET			ID: 0x60B2
RW	Volatile	Einheit: ‰	I16

In den positions- und geschwindigkeitsbezogenen Betriebsarten enthält das Objekt den Eingangswert für die Drehmomentvorsteuerung. In den Drehmomentbetriebsarten enthält es den Drehmomentoffset des Antriebs. In den Torque-Betriebsarten setzt sich das Zieldrehmoment also aus dem **Target Torque** und dem **Torque Offset** zusammen.

2.5.7 Begrenzungen

Max Torque (0x6072)

Dieses Objekt enthält den maximal erlaubten Wert des Drehmoments.

Max Torque		Drehmomentbegrenzung für das Drehmomentprofil Der Wert wird in Promille (1000 = 100 %) des Nenndrehmoments, Objekt Motor Rated Torque (0x6076), angegeben.	
DS402_0x6072_MAX_TORQUE			ID: 0x6072

RW	Volatile	Einheit: ‰	U16
----	----------	------------	-----

Max Current (0x6073)

Über dieses Objekt kann eine Begrenzung für den drehmomenterzeugenden Strom im Motor vorgegeben werden.

Max Current	Strombegrenzung des Wirkstroms Der Wert wird in Promille (1000 = 100 %) des Nennstroms, Objekt Motor Rated Current (0x6075), angegeben.		
DS402_0x6073_MAX_CURRENT			ID: 0x6073
RW	Volatile	Einheit: ‰	U16

Max Profile Velocity (0x607F)

Über dieses Objekt kann die maximale Geschwindigkeit vorgegeben werden.

Max Profile Velocity	Geschwindigkeitsbegrenzung für das Bewegungsprofil Der Wert gilt für beide Richtungen. Die Einheit entspricht der in Objekt Target Velocity (0x60FF).		
DS402_0x607F_MAX_PROFILE_VELOCITY			ID: 0x607F
RW	Volatile	Einheit: –	U32

Max Motor Speed (0x6080)

Über dieses Objekt kann die maximale Geschwindigkeit für den Motor vorgegeben werden. Der Wert wird dem Motordatenblatt entnommen und dient zum Schutz des Motors.

Max Motor Speed	Durch den Motor vorgegebene maximale Geschwindigkeit Der Wert gilt für beide Richtungen. Die Einheit entspricht der in Objekt Target Velocity (0x60FF).		
DS402_0x6080_MAX_MOTOR_SPEED			ID: 0x6080
RW	Volatile	Einheit: –	U32

Positive Torque Limit (0x60E0)

Über dieses Objekt kann das maximale positive Drehmoment im Motor vorgegeben werden. Das positive Drehmoment wirkt bei motorischem Betrieb mit positiver Geschwindigkeit und bei regenerativem Betrieb mit negativer Geschwindigkeit.

Positive Torque Limit	Obere Grenze des Drehmomentsollwerts Der Wert wird in Promille (1000 = 100 %) des Nenndrehmoments, Objekt Motor Rated Torque (0x6076), angegeben.		
DS402_0x60E0_POSITIVE_TORQUE_LIMIT_VALUE			ID: 0x60E0
RW	Volatile	Einheit: ‰	I16

Negative Torque Limit (0x60E1)

Über dieses Objekt kann das maximale negative Drehmoment im Motor vorgegeben werden. Das negative Drehmoment wirkt bei motorischem Betrieb mit negativer Geschwindigkeit und bei regenerativem Betrieb mit positiver Geschwindigkeit.

Negative Torque Limit	Untere Grenze des Drehmomentsollwerts Der Wert wird in Promille (1000 = 100 %) des Nenndrehmoments, Objekt Motor Rated Torque (0x6076), angegeben.		
DS402_0x60E1_NEGATIVE_TORQUE_LIMIT_VALUE			ID: 0x60E1
RW	Volatile	Einheit: ‰	I16

2.5.7.7 Beschleunigungs- und Verzögerungsbegrenzungen

Max Acceleration (0x60C5)

Über dieses Objekt kann die maximal zulässige Beschleunigung vorgegeben werden. Damit können Sie die Beschleunigung auf einen akzeptablen Wert begrenzen, um Schäden am Motor und den bewegten Teilen zu verhindern.

Max Acceleration		Parametrierte maximale Beschleunigung des Systems Die Einheit wird im Objekt Acceleration (0x60AA) vorgegeben.	
DS403_0X60C5_MAX_ACCELERATION			ID: 0x60C5
RW	Volatile	Einheit rotativer Motor: 1/s ² Einheit Linearmotor: mm/s ²	U32

Max Deceleration (0x60C6)

Über dieses Objekt kann die maximal zulässige Verzögerung vorgegeben werden. Damit können Sie die Verzögerung auf einen akzeptablen Wert begrenzen, um Schäden am Motor und den bewegten Teilen zu verhindern.

Max Deceleration		Parametrierte maximale Verzögerung des Systems Die Einheit wird im Objekt Acceleration (0x60AA) vorgegeben.	
DS403_0X60C6_MAX_DECELERATION			ID: 0x60C6
RW	Volatile	Einheit rotativer Motor: 1/s ² Einheit Linearmotor: mm/s ²	U32

Profile Acceleration (0x6083)

Über dieses Objekt kann die Beschleunigung vorgegeben werden.

Profile Acceleration		Beschleunigungsrampe Die Einheit wird im Objekt Acceleration (0x60AA) vorgegeben.	
DS402_0x6083_PROFILE_ACCELERATION			ID: 0x6083
RW	Volatile	Einheit rotativer Motor: 1/s ² Einheit Linearmotor: mm/s ²	U32

Profile Deceleration (0x6084)

Über dieses Objekt kann die Verzögerung vorgegeben werden.

Profile Deceleration		Verzögerungsrampe Die Einheit wird im Objekt Acceleration (0x60AA) vorgegeben.	
DS402_0x6084_PROFILE_DECELERATION			ID: 0x6084
RW	Volatile	Einheit rotativer Motor: 1/s ² Einheit Linearmotor: mm/s ²	U32

Quick Stop Deceleration (0x6085)

Über dieses Objekt kann die Verzögerung bei einem Schnellhalt vorgegeben werden. Wann eine Schnellhalt ausgelöst wird, können Sie über die [Option Codes \(S. 12\)](#) konfigurieren.

Quick Stop Deceleration		Schnellhalttrampe Die Einheit wird im Objekt Acceleration (0x60AA) vorgegeben.	
DS402_0x6085_QUICK_STOP_DECELERATION			ID: 0x6085
RW	Volatile	Einheit rotativer Motor: 1/s ² Einheit Linearmotor: mm/s ²	U32

Torque Slope (0x6087)

Über dieses Objekt kann die Änderungsrate des Drehmoments vorgegeben werden.

Torque Slope		Drehmomentänderungsrate Der Wert wird in Promille (1000 = 100 %) des Nenndrehmoments, Objekt Motor Rated Torque (0x6076), pro Sekunde angegeben.	
DS402_0x6087_TORQUE_SLOPE			ID: 0x6087
RW	Volatile	Einheit: ‰/s	U32

2.5.8

Sonstige

Digital Inputs (0x60FD)

Dieses Objekt gibt den Zustand der digitalen Eingänge des SD4x-Antriebs zurück. Das Objekt ist als 32-Bit-Variable angelegt. Wenn ein Bit logisch 1 ist, dann ist die auf diesem Bit belegte Funktion aktiv.

Digital Inputs		Zustand der digitalen Eingänge Bitbelegung: Bit 0: Negativer Endschalter Bit 1: Positiver Endschalter Bit 2: Home-Schalter Bit 3: Interlock aktiviert Bits 4 bis 15: Reserviert Bits 16 bis 31: Eingang 1 bis 16 - konfigurierbar	
DS402_0X60FD_DIGITAL_INPUTS			ID: 0x60FD
RO	Volatile	Einheit: –	U32

Bit 0 gibt den Zustand des negativen Endschalters zurück. Bei SD4x-Antrieben entspricht dieses Bit der Funktion „(16) Negativer Endschalter“ für digitale Eingänge.

Bit 1 gibt den Zustand des positiven Endschalters zurück. Bei SD4x-Antrieben entspricht dieses Bit der Funktion „(15) Positiver Endschalter“ für digitale Eingänge.

Bit 3 gibt den Zustand des Interlock-Eingangs zurück. Wenn dieses Eingangssignal deaktiviert wird, muss der SD4x-Antrieb in den Zustand „Einschaltsperrung“ oder „Fehlerreaktion aktiv“ wechseln. Dadurch wird die Leistungsstufe des Antriebs gesperrt und gegen Wiedereinschalten gesichert. Bei SD4x-Antrieben entspricht dieses Bit der invertierten Funktion „(01) Regler Ein“ für digitale Eingänge. Wenn zum Beispiel der digitale Eingang 2 des SD4x mit der Funktion „(01) Regler Ein“ konfiguriert wird und der physikalische Eingang 2 logisch 0 ist, dann ist Bit 3 im Objekt **Digital Inputs** logisch 1 und Bit 17 logisch 0. Ist der physikalische Eingang 2 logisch 1, dann ist Bit 3 logisch 0 und Bit 17 logisch 1.

Bit 16 bis 31 entsprechen den Eingängen des Geräts. Wie viele Bits genutzt werden, hängt von der tatsächlichen Anzahl der physischen Eingänge des SD4x-Antriebs ab.

Digital Outputs (0x60FE)

Über dieses Objekt kann der Zustand der digitalen Ausgänge des SD4x-Antriebs sowohl abgerufen als auch gesetzt werden.

Es enthält folgende Subobjekte: Zustandsvariable **Physical Outputs** (0x60FE:1) und Maskierungsvariable **Bit Mask** (0x60FE:2). Beide Subobjekte sind 32-Bit-Variablen und enthalten die Bitbelegung der digitalen Ausgänge. Die Maskierungsvariable erlaubt das Holen der Steuerhoheit über einzelne digitale Ausgänge des Geräts. Wenn die Maskierungsvariable für einen Ausgang logisch 0 ist, dann steuert das Gerät diesen Ausgang

und sein aktueller Zustand wird über die Zustandsvariable zurückgeliefert. Sobald ein Bit in der Maskierungsvariable logisch 1 gesetzt wird, übernimmt die Zustandsvariable die Steuerung des entsprechenden Ausganges.

Digital Outputs		Zugriff auf die digitalen Ausgänge Bitbelegung: Bit 0: Motorhaltebremse aktiv Bits 1 bis 15: Reserviert Bits 16 bis 31: Ausgang 1 bis 16 - konfigurierbar Sub-IDs: ▶ 0x60FE:0 = Highest subindex supported (2) ▶ 0x60FE:1 = Physical Outputs ▶ 0x60FE:2 = Bit Mask	
DS402_0X60FE_DIGITAL_OUTPUTS			ID: 0x60FE
RW	Volatile	Einheit: –	BYTEARRAY

Bit 0 liefert den Zustand bzw. den Steuerbefehl der Motorhaltebremse. Bei SD4x-Antrieben entspricht dieses Bit der Funktion „(05) Motor Haltebremse“ für digitale Ausgänge.

Bit 16 bis 31 entsprechen den Ausgängen des Geräts. Wie viele Bits genutzt werden, hängt von der tatsächlichen Anzahl der physischen Ausgänge des SD4x-Antriebs ab.

Highest subindex supported (0x60FE:0)

Dieses Objekt enthält im Subindex 0 den letzten Subindex des Objekts, in diesem Fall 2.

Physical Outputs (0x60FE:1)

Dieses Objekt enthält in Abhängigkeit von der Maskierungsvariable **Bit Mask** (0x60FE:2) entweder die aktuellen Zustände der digitalen Ausgänge oder steuert einzelne Ausgänge. Eine logische 1 für ein Bit bedeutet aktiv, eine logische 0 bedeutet nicht aktiv.

Bit Mask (0x60FE:2)

Dieses Objekt kann die Steuerhoheit über einzelne digitale Ausgänge des Geräts an das Objekt **Physical Outputs** (0x60FE:1) übergeben. Eine logische 1 für ein Bit bedeutet Steuerhoheit, eine logische 0 bedeutet, dass der Zustand des Ausganges gelesen wird.

Beispiel:

- ▶ Ausgang 1: parametrier mit Funktion „(01) Betriebsbereit“
- ▶ Ausgang 2: parametrier mit Funktion „(03) Eingeschaltet“

Der Zustand des physikalischen Ausganges 1 soll beobachtet werden, während der physikalische Ausgang 2 den logischen Zustand 1 (also aktiv) ausgeben soll.

Um dies zu erreichen, setzen Sie die Bits im Objekt **Bit Mask** wie folgt: Bit 16 auf logisch 0 und Bit 17 auf logisch 1. Der logische Zustand von Ausgang 2 wird nun durch Bit 17 im Objekt **Physical Outputs** definiert, Bit 16 entspricht weiterhin dem logischen Zustand des physikalischen Ausganges 1. Solange Bit 16 in der Maskierungsvariablen logisch 0 belassen wird, kann die Steuerung den Zustand von Ausgang 1 nicht ändern.

Error Code (0x603F)

Dieses Objekt gibt den Fehlercode des zuletzt aufgetretenen Fehlers zurück.

Error Code		Letzter gespeicherter Fehlercode nach Antriebsprofil DS402. Der Wert 0 bedeutet, dass kein Fehler im Gerät anliegt.	
DS402_0X603F_ERROR_CODE			ID: 0x603F
RO	Volatile	Einheit: –	U16

Die Fehlercodes sind hardwareabhängig. Eine detaillierte Beschreibung der Fehlercodes finden Sie in der Hardwarebeschreibung zu Ihrem Gerät.

2.6 Parametersatzumschaltung

Für einen SD4x-Antrieb können bis zu 64 Parametersätze angelegt sein. Das Umschalten auf einen anderen Parametersatz ist nur bei ausgeschalteter Endstufe möglich. Ein erfolgreicher Parametersatzwechsel führt zu einem Geräteneustart mit dem ausgewählten Parametersatz.

Hinweis

Wenn das PDO-Mapping über die Feldbussteuerung eingestellt wird, muss dies nach einer Parametersatzumschaltung erneut durchgeführt werden. Das bedeutet in der Regel, dass das Gerät noch einmal die Feldbus-Initialisierungsphase durchlaufen muss (z. B. den NMT-Zustandswechsel von Pre-Operational zu Operational).

Die Parametersatzumschaltung ist über folgende Steuerquellen implementiert: *drive-master4*-Software, digitale Eingänge oder Feldbus. Die Steuerquelle bestimmt auch, welcher Parametersatz beim Hochfahren des Geräts aktiviert wird.

Auswahl Quelle (0x2037)

Dieses Objekt bestimmt die Steuerquelle für die Auswahl und Umschaltung des Parametersatzes. Intern (innerhalb der *drivemaster4*-Software) finden Sie dieses Objekt unter dem Index 0x0037.

Folgende Optionen stehen zur Verfügung:

- ▶ **(00) Feste Auswahl (EEPROM):**
Diese Option ist die Werkseinstellung. Das Gerät wird mit dem zuletzt ausgewählten Parametersatz gestartet. Sie können den Parametersatz sowohl über die *drive-master4*-Software als auch über das Feldbussystem umschalten.
- ▶ **(01) Digitale Eingänge:**
Das Gerät wird mit dem Parametersatz 0 gestartet und schaltet direkt in den durch die digitalen Eingänge vorgegebenen Parametersatz. Sie können den Parametersatz ausschließlich über die digitalen Eingänge umschalten.
- ▶ **(02) Bussystem:**
Das Gerät wird mit dem Parametersatz 0 gestartet. Sie können den Parametersatz ausschließlich über das Feldbussystem umschalten.

Ablaufdiagramm

Das folgende Diagramm zeigt den zeitlichen Ablauf der Parametersatzumschaltung. Die eigentliche Umschaltung von einem Parametersatz auf einen anderen geschieht über das Deaktivieren und Aktivieren des Befehls „Regler Ein“:

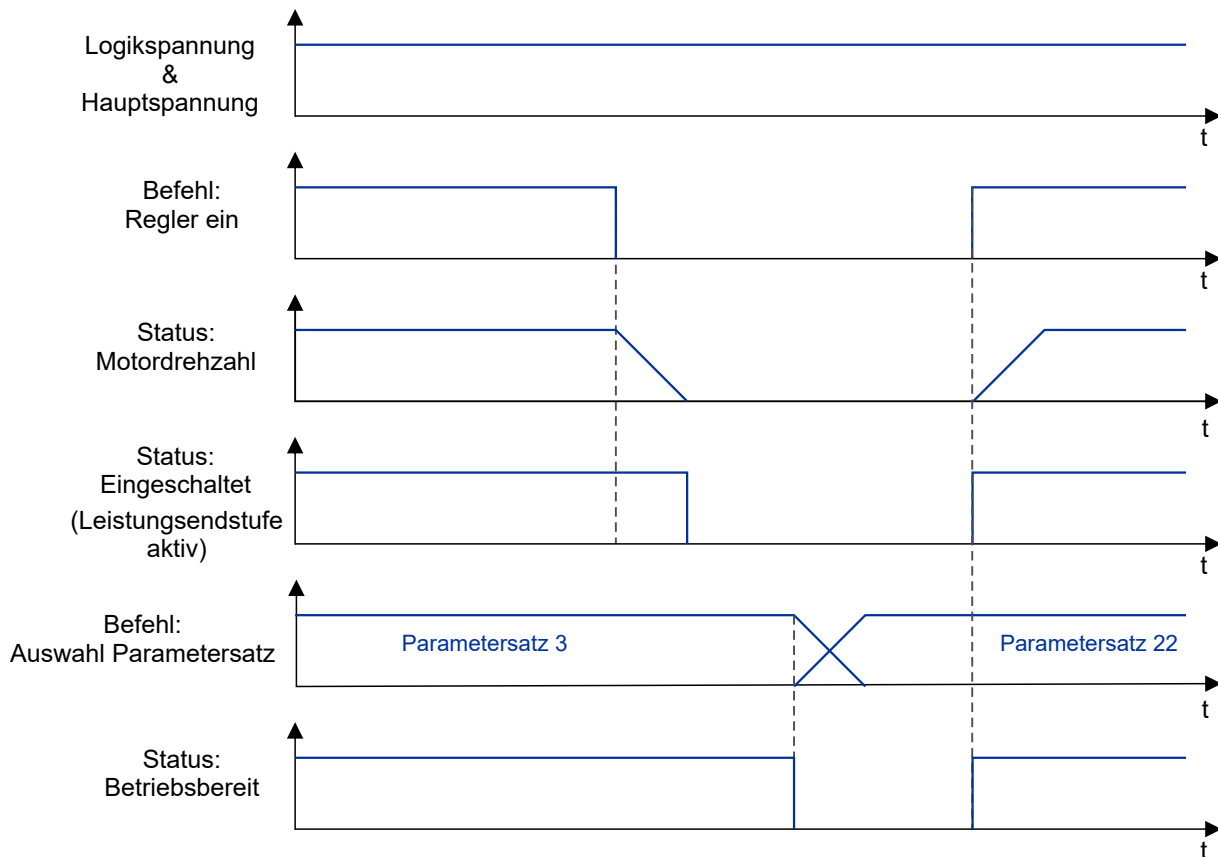


Abb. 13: Ablauf der Parametersatzumschaltung

Im Folgenden ist die Parametersatzumschaltung über Feldbus genauer beschrieben. Eine Beschreibung zur Parametersatzumschaltung über die *drivemaster4*-Software oder die digitalen Eingängen finden Sie im Dokument „Drive Controller SD4x – Funktionen und Parameter“.

2.6.1 Parametersatzumschaltung über Feldbus

Der Parametersatz wird über das Objekt 0x202E (Auswahl) umgeschaltet.

Folgende Voraussetzungen müssen für eine erfolgreiche Parametersatzumschaltung über das Bussystem erfüllt sein:

1. Das Objekt 0x2037 (Auswahl Quelle) ist auf **(02) Bussystem** oder **(00) Feste Auswahl (EEPROM)** gesetzt.
2. Die Endstufe ist ausgeschaltet.

Auswahl (0x202E)

Das Objekt enthält die Nummer des aktuell ausgewählten Parametersatzes. Intern (innerhalb der *drivemaster4*-Software) finden Sie dieses Objekt unter dem Index 0x002E.

Um über das Objekt 0x202E zwischen bestehenden Parametersätzen umzuschalten, muss nur die entsprechende Parametersatznummer in das Objekt geschrieben werden.

Eine Parametersatzumschaltung führt zu einem Geräteneustart und dadurch zu einem kurzzeitigen Verlust der Feldbusverbindung – also der Kontrolle über das Gerät.

3 Ansteuerung über EtherCAT

Im Folgenden wird die Antriebsadressierung im EtherCAT-Netzwerk und die Anbindung an die SPS am Beispiel der Beckhoff-Software TwinCAT3 beschrieben.

Die Feldbuskommunikation muss im SD4x-Antrieb bereits parametrierbar sein, damit er im EtherCAT-Netzwerk richtig erkannt wird. Weitere Informationen dazu finden Sie im [Kapitel 4.1 „Parametrierung in drivemaster4“, Seite 66](#).

3.1 EtherCAT-Netzwerk

EtherCAT ist ein leistungsstarker Feldbus, der speziell für die Automatisierungstechnik entwickelt wurde. Das Netzwerk basiert auf dem Ethernet-Protokoll und kann mit handelsüblichen CAT5-Netzwerkkabeln verbunden werden. Die Topologie und damit die Adressierung (OSI-Schicht 3) und der Telegramm-Transport (OSI-Schicht 4) weichen jedoch vom Standard-Ethernet (IP- oder TCP/UDP-basiert) ab. Ein normales, IP-basiertes Ethernet-Netzwerk kann nicht direkt mit EtherCAT verbunden werden, sondern muss mittels Ethernet over EtherCAT (EoE) über das EtherCAT-MDevice (z. B. SPS) angebunden werden.

Der EtherCAT wird in einer Daisy Chain aufgebaut und die Adressen werden vom MDevice vergeben, siehe [Abb. 14, Seite 40](#). Durch die Daisy-Chain-Topologie ergeben sich die Geräte-Adressen automatisch aus der Position des Teilnehmers (SubDevice) in der Kette. Sie können aber auch manuell durch den Bediener vergeben werden. Weitere Informationen zur Adressierung finden Sie im [Kapitel 3.2.3 „Adressierung und Identifikation“, Seite 52](#).

Nur das EtherCAT-MDevice darf Telegramme erzeugen, die in der Kette von einem SubDevice zum nächsten weitergeleitet werden. Dabei liest jedes SubDevice die Sollwerte aus dem Telegramm und schreibt gleichzeitig seine Istwerte hinein. Der letzte Teilnehmer in der Kette sendet das Telegramm zurück zum MDevice.

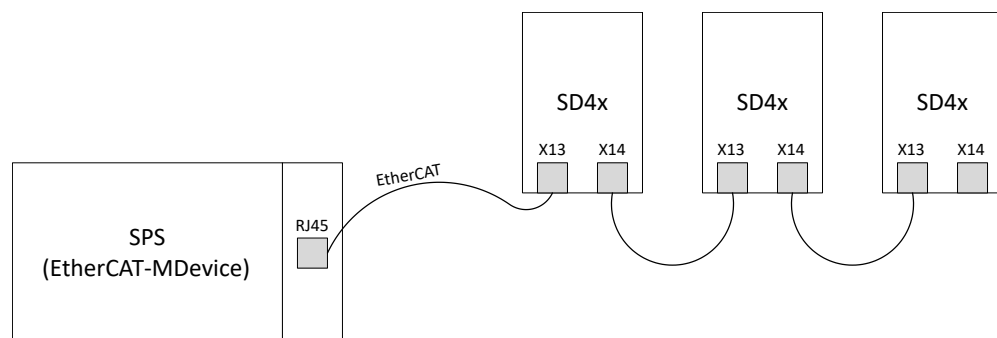


Abb. 14: EtherCAT-Netzwerk (Daisy Chain)

Die Anschlussbelegung der EtherCAT-Schnittstellen und die Bedeutung der entsprechenden Status-LEDs sind in der Hardwaredokumentation des jeweiligen SD4x-Antriebs beschrieben.

3.2 SPS-Anbindung (TwinCAT3)

Die Anbindung an die SPS wird nachfolgend am Beispiel der TwinCAT3 der Firma Beckhoff beschrieben.

1. Kopieren Sie die von SIEB & MEYER bereitgestellte XML-Datei in das Twin-CAT3-Verzeichnis ...\\TwinCAT\\3.1\\Config\\Io\\EtherCAT.

– z. B.: SIEB_MEYER_SD4x.xml

Diese XML-Datei enthält die EtherCAT SubDevice Information (ESI), d. h. sie beschreibt die Kommunikationsparameter und das Objektverzeichnis des Antriebs.

2. Starten Sie die TC3-Oberfläche und stellen Sie die Verbindung zum Target-System her.
3. Schalten Sie das Target-System in den Config-Modus und bestätigen Sie die Abfragen „Load I/O Devices?“ und „Activate Free Run?“ mit „Ja“.
4. Klicken Sie mit der rechten Maustaste auf „Devices“ und wählen Sie „Scan“ im Kontextmenü aus.

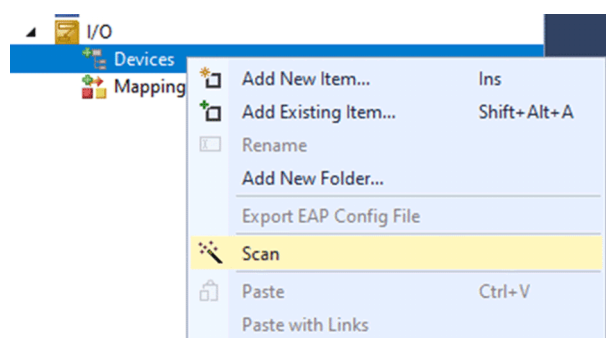


Abb. 15: Scan ausführen

5. Wählen Sie im folgenden Dialog das EtherCAT-Protokoll aus.

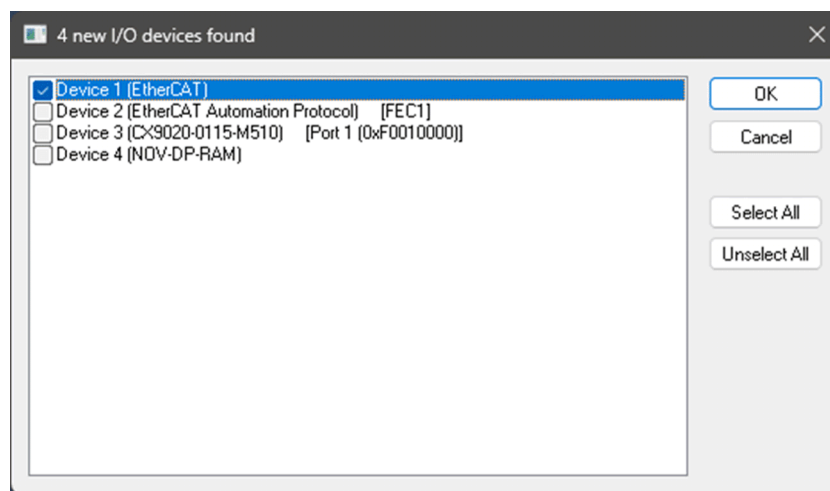


Abb. 16: EtherCAT auswählen

6. Bestätigen Sie die Abfrage „Scan for boxes“ mit „Ja“.

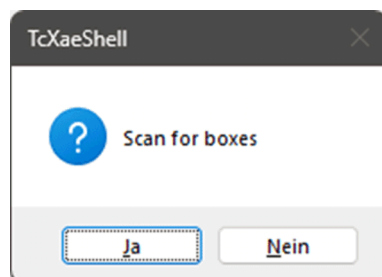


Abb. 17: Scan for boxes

7. Brechen Sie den Dialog „EtherCAT drive(s) added“ mit „Cancel“ ab. Anderenfalls legt Beckhoff automatisch das Motion-Control-Framework an und verwendet dies.

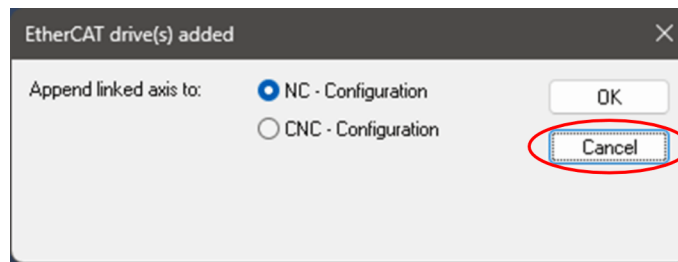


Abb. 18: EtherCAT drive(s) added abbrechen

- TwinCAT3 lädt nun automatisch alle gefundenen EtherCAT-Geräte in das Projekt wie im folgenden Screenshot gezeigt:

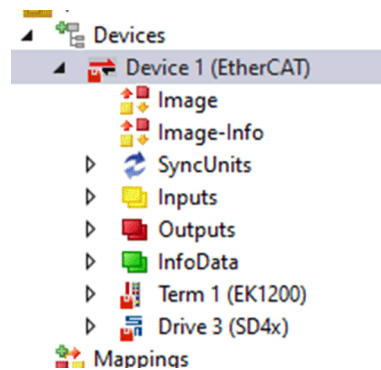


Abb. 19: SD4x in Projektstruktur

Wenn der SD4x-Antrieb nicht oder nicht korrekt erkannt wurde, ist die XML-Datei (ESI) wahrscheinlich nicht richtig installiert (siehe Schritt 1) oder TwinCAT3 wurde noch nicht neu gestartet.

8. Doppelklicken Sie in der Projektstruktur auf den SD4x-Antrieb, um dessen Einstellungen zu öffnen. Wechseln Sie auf den Reiter „Process Data“:

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	6	Outputs	
3	6	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	6.0	Status		3	0
0x1A03	13.0	Status			0
0x1600	6.0	Controls		2	0
0x1603	13.0	Controls			0

PDO Assignment (0x1C12):

☒ 0x1600
☐ 0x1603

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x6041:00	2.0	0.0	Statusword	UINT	
0x606C:00	4.0	2.0	Velocity Actual Value	DINT	
	6.0				

Download:

☒ PDO Assignment
☒ PDO Configuration

Predefined PDO Assignment: 'Custom'

☐ Load PDO info from device

☐ Sync Unit Assignment...

Name **Online** **Type** **Size** **>Add**

Statusword	37456	UINT	2.0	39.0
Velocity Actual ...	0	DINT	4.0	41.0
WcState	0	BIT	0.1	1522.1
InputToggle	0	BIT	0.1	1524.1
State	8	UINT	2.0	1550.0
.AdsAddr	5.146.241.177.2.1:1...	AMSADDR	8.0	1552.0
Chn0	0	USINT	1.0	1560.0
Controlword	0	UINT	2.0	39.0
Target Velocity	0	DINT	4.0	41.0

Abb. 20: Process Data

Hier können Sie die Prozessdaten erweitern oder austauschen.

Wenn im Bereich „Download“ die Checkboxes „PDO Assignment“ und „PDO Configuration“ aktiviert sind, wird dieses Prozessdatenmapping automatisch bei der Feldbusinitialisierung mit dem SD4x-Antrieb ausgetauscht. Dabei werden nur PDOs aktiv ausgetauscht, die in der Spalte „SM“ eine Nummer stehen haben (also einem SyncManager zugeordnet sind).

Zusätzlich können Sie eigene PDOs anlegen: Die Indizes 0x1A00 bis 0x1A03 werden für TxPDOs (Status-Daten) unterstützt und 0x1600 bis 0x1603 für RxPDOs (Control-Daten). Über „Predefined PDO Assignment“ können Sie auch vorgefertigte PDO-Sets auswählen.

Hinweis

SD4x-Antriebe mit einer Firmware kleiner Vx.8.x zeigen die Diagnose-Informationen unter Umständen nicht korrekt in der *drivemaster4*-Software an, wenn andere PDO Adressen als 0x1A00 und 0x1600 verwendet werden.

- Im Reiter „Startup“ können Sie zu den Mapping-Parametern noch eigene Parameter hinzufügen, die während der Feldbusinitialisierung geschrieben werden sollen. Dazu klicken Sie mit der rechten Maustaste in die Liste und wählen „Add New Item“ im Kontextmenü aus. Hier können Sie zum Beispiel das Objekt „Mode of operation“ vorbelegen wie in der folgenden Abbildung gezeigt:

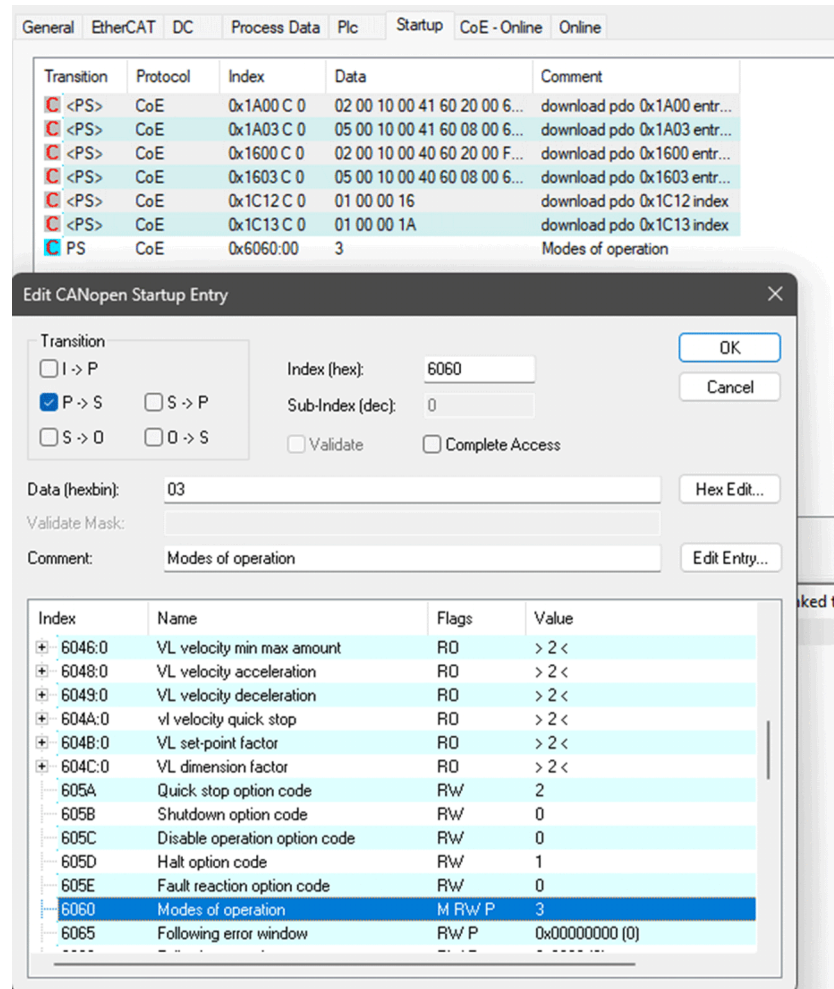


Abb. 21: Mode of operation einstellen

10. Wenn Sie im SPS-Programm bereits mappbare Variablen angelegt haben, können Sie diese nun den Prozessdaten zuweisen:

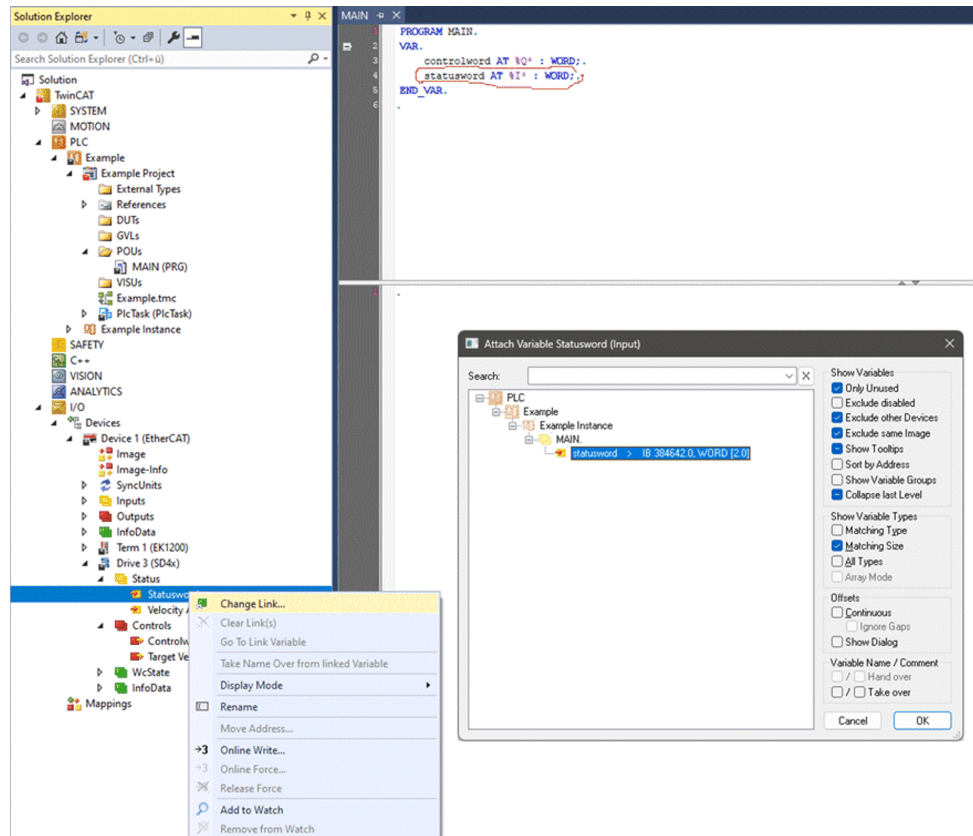


Abb. 22: Variablen verlinken

11. Laden und aktivieren Sie die Konfiguration in der SPS. Bestätigen Sie die Abfragen „Activate Configuration?“ und „Restart TwinCAT System in Run Mode?“ mit „OK“.

→ Die Konfiguration und das SPS-Programm werden in die SPS geladen und gestartet.

- ✓ Wechseln sie zum Reiter „Online“.

Die EtherCAT State Machine wechselt in den Zustand „Operational“ (OP). Die verlinkten Ein- und Ausgangsdaten sind markiert und die aktuellen Werte werden angezeigt.

Name	Online	Type	Size	> Addr...	In/Out	User ID	Linked to
Statusword	37456	UINT	2.0	39.0	Input	0	
Velocity Actual ...	0	DINT	4.0	41.0	Input	0	
WcState	0	BIT	0.1	1522.1	Input	0	
InputToggle	0	BIT	0.1	1524.1	Input	0	
State	8	UINT	2.0	1550.0	Input	0	
AdsAddr	5.146.241.177.2.1:1...	AMSADDR	8.0	1552.0	Input	0	
Chn0	0	USINT	1.0	1560.0	Input	0	
Controlword	0	UINT	2.0	39.0	Output	0	
Target Velocity	0	DINT	4.0	41.0	Output	0	

Abb. 23: Online-Anzeige

3.2.1 EtherCAT State Machine (ESM)

Die Kommunikation des Antriebs wird über die EtherCAT State Machine (ESM) von der SPS gesteuert. Die EtherCAT State Machine koordiniert MDevice- und SubDevice-Applikationen beim Hochfahren und während des Betriebs. Zustandsänderungen im Sub-Device werden vom MDevice angefordert.

Die EtherCAT State Machine kennt folgende Zustände:

Zustand	Wert
Init	1
Pre-Operational	2
Bootstrap	3
Safe-Operational	4
Operational	8

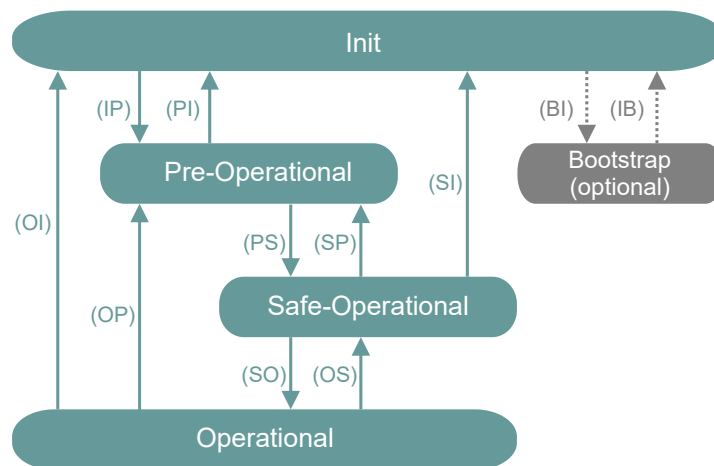


Abb. 24: Zustandsänderungen der EtherCAT State Machine

Die nachfolgenden Zustandsbeschreibungen sind der Onlinedokumentation der Firma Beckhoff entnommen, siehe [Beckhoff Information System](#).

Init	Im Zustand Init findet keine Kommunikation auf der Anwendungsebene statt. Das MDevice hat Zugriff auf die ihn informierenden Register der Data-Link-Layer.
Pre-Operational (Pre-OP)	Im Zustand Pre-Operational findet Mailboxkommunikation auf der Applikationsschicht statt, jedoch noch keine Prozessdatenkommunikation.
Safe-Operational (Safe-OP)	Im Zustand Safe-Operational findet Prozessdatenkommunikation statt. Eingänge werden gelesen, Ausgänge werden noch nicht geschrieben.
Operational (OP)	Im Zustand Operational haben die Ausgänge gültige Werte und die Eingänge werden wie schon im Zustand Safe-Operational weiterhin ausgewertet. Das Receive-PDO (RPDO) wird vom MDevice an den Antrieb gesendet. Das Transmit-PDO (TPDO) wird vom Antrieb an das MDevice gesendet.
Bootstrap	Der Zustand Bootstrap wird nur über den Zustand Init erreicht und auch nur zum Zustand Init wieder verlassen. Der Zustand Bootstrap wird von der EtherCAT-Organisation für Firmware-Updates über EoE-Protokoll (EoE = Ethernet over EtherCAT) empfohlen.

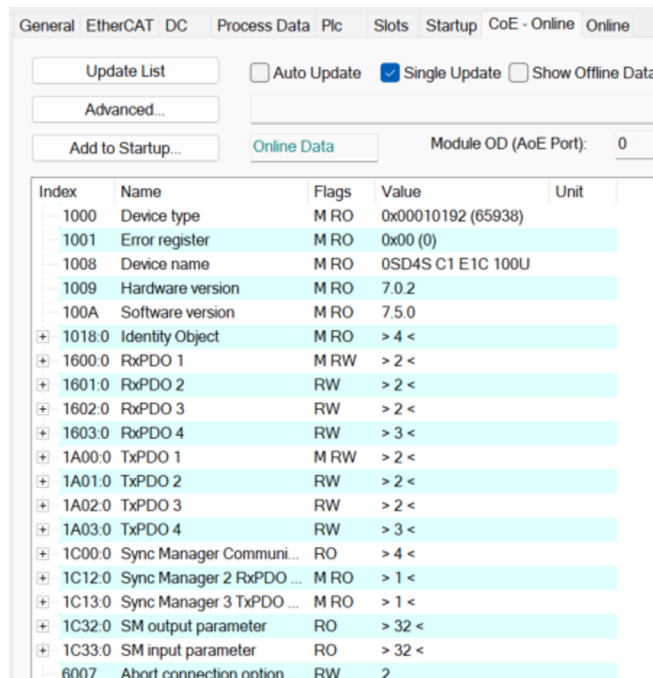
Hinweis

Diese Update-Funktion wird von den SD4x-Geräten nicht unterstützt. Im Zustand Bootstrap findet keine Prozessdatenkommunikation statt.

3.2.2 SDO-Kommunikation

Die SDO-Kommunikation läuft über die CoE-Schnittstelle (CoE = CANopen over Ethernet) des EtherCAT-MDevices. Während des Hochfahrens (Startup) sendet das EtherCAT-MDevice automatisch einige SDOs. An dieser Stelle können Sie auch weitere SDOs von Hand hinzufügen (siehe [Kapitel 3.2 „SPS-Anbindung \(TwinCAT3\)“](#), [Seite 40](#)).

Außerdem bietet TwinCAT3 die Diagnoseseite „CoE - Online“. Hier werden die aktuellen Werte der Objekte angezeigt, die in der Gerätebeschreibungsdatei (ESI/XML) enthalten sind. Zudem können Sie die Objekte an dieser Stelle ändern oder zu den Startup-SDOs hinzufügen:



Index	Name	Flags	Value	Unit
1000	Device type	M RO	0x00010192 (65938)	
1001	Error register	M RO	0x00 (0)	
1008	Device name	M RO	0SD4S C1 E1C 100U	
1009	Hardware version	M RO	7.0.2	
100A	Software version	M RO	7.5.0	
1018.0	Identity Object	M RO	> 4 <	
1600.0	RxPDO 1	M RW	> 2 <	
1601.0	RxPDO 2	RW	> 2 <	
1602.0	RxPDO 3	RW	> 2 <	
1603.0	RxPDO 4	RW	> 3 <	
1A00.0	TxPDO 1	M RW	> 2 <	
1A01.0	TxPDO 2	RW	> 2 <	
1A02.0	TxPDO 3	RW	> 2 <	
1A03.0	TxPDO 4	RW	> 3 <	
1C00.0	Sync Manager Communi...	RO	> 4 <	
1C12.0	Sync Manager 2 RxPDO ...	M RO	> 1 <	
1C13.0	Sync Manager 3 TxPDO ...	M RO	> 1 <	
1C32.0	SM output parameter	RO	> 32 <	
1C33.0	SM input parameter	RO	> 32 <	
6007	Abort connection option ...	RW	2	

Abb. 25: TwinCAT3-Software – CoE - Online

Die SPS-Hersteller bieten für die SDO-Kommunikation aus dem SPS-Anwenderprogramm fertige Funktionsbausteine an. Bei Beckhoff finden Sie diese Funktionsbausteine im Ordner „CoE Interface“ der Tc2_EtherCAT-Bibliothek.

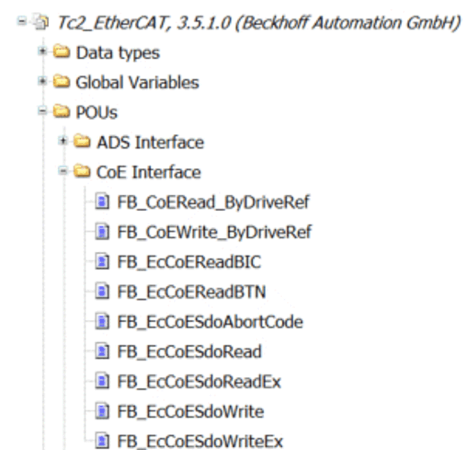


Abb. 26: Tc2_EtherCAT-Ordner

Mit FB_EcCoESdoRead können SDO-Objekte gelesen und mit FB_EcCoESdoWrite geschrieben werden:



Abb. 27: Funktionsbausteine zum Lesen und Schreiben der SDOs

Für die Eingänge sNetId (netId) und nSlaveAddr (port) können Sie die InfoData des EtherCAT-SubDevices übernehmen. Die netId aus den InfoData hat jedoch einen anderen Datentyp als die netId-Eingänge der CoE-Funktionen. Zur Konvertierung in den richtigen Datentyp, können Sie die Funktion F_CreateAmsNetId aus der Tc2_System-Bibliothek verwenden.

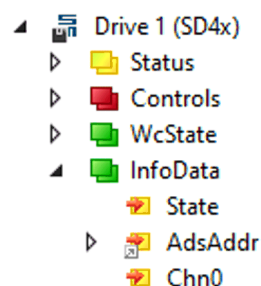


Abb. 28: SubDevice-InfoData

Der tTimeout-Eingang ist bereits vorinitialisiert und kann leer bleiben.

bExecute, bBusy und bError sind angelehnt an die Schnittstelle einer PLCopen-Execute-Funktion. Tritt ein Fehler während der Bearbeitung des SDO-Objekts auf, liefert der Ausgang nErrId den entsprechenden Fehlercode zurück. Dieser kann als ADS-Return-Code interpretiert werden. Hierfür können Sie zum Beispiel die ENUM E_AdsErr aus der Tc2_System-Bibliothek verwenden oder die Beckhoff-Dokumentationen zur Hilfe nehmen.

Die Eingänge nIndex, nSubIndex, cbBufLen und pDstBuf (bzw. pSrcBuf) definieren das SDO-Objekt. nIndex und nSubIndex enthalten die SDO-Adresse, cbBufLen die Objektgröße und pDstBuf (bzw. pSrcBuf) den Puffer, aus dem gelesen (bzw. geschrieben) werden soll. Optimalerweise zeigt der Pointer auf einen Speicherbereich in der SPS, in dem

eine Variable mit dem richtigen Datentyp und der richtigen Größe liegt. Informationen zu den SDO-Adressen und zum Datentyp finden Sie in diesem Dokument oder im Objektbrowser der *drivemaster4*-Software:

Name	Objekt	ID	Sub-ID	Wert	Einheit	Datentyp
> Gerät	DEVICE					
> Gerätesteuerung	DEVICE_CONTROL					
> Kommunikation	COMMUNICATION					
> Allgemein	COMMUNICATION_COMMON					
> TCP_IP	TCP_IP					
> DS301	DS301					
> DS402	DS402					
DS402 - Abort Connection Option Code	DS402_0x6007_ABORT_CONNECTION_OPTION_CODE	0x6007	(02) Befehl Spannung sperren			UNIT_I16
DS402 - Error Code	DS402_0x603F_ERROR_CODE	0x603f	0			UNIT_U16
DS402 - Controlword	DS402_0x6040_CONTROLWORD	0x6040	0			UNIT_U16
DS402 - Statusword	DS402_0x6041_STATUSWORD	0x6041	36913			UNIT_U16
DS402 - vl Target Velocity	DS402_0x6042_VL_TARGET_VELOCITY	0x6042	0			UNIT_I16
DS402 - vl Velocity Demand	DS402_0x6043_VL_VELOCITY_DEMAND	0x6043	0			UNIT_I16
DS402 - vl Velocity Actual Value	DS402_0x6044_VL_VELOCITY_ACTUAL_VALUE	0x6044	0			UNIT_I16
DS402 - vl Velocity Min/Max Amount	DS402_0x6046_VL_VELOCITY_MIN_MAX_AMOUNT	0x6046	02 00 00 00 00 60 ea 00 ...			UNIT_BYTEARRAY
DS402 - vl Velocity Acceleration	DS402_0x6048_VL_VELOCITY_ACCELERATION	0x6048	02504600000100			UNIT_DS402_ACCELERATION_REC
DS402 - vl Velocity Deceleration	DS402_0x6049_VL_VELOCITY_DECELERATION	0x6049	02504600000100			UNIT_DS402_ACCELERATION_REC
DS402 - vl Velocity Quick Stop	DS402_0x604A_VL_VELOCITY_QUICK_STOP	0x604a	02504600000100			UNIT_DS402_ACCELERATION_REC
DS402 - vl Set-point Factor	DS402_0x604B_VL_SET_POINT_FACTOR	0x604b	02 01 00 01 00			UNIT_BYTEARRAY
DS402 - vl Dimension Factor	DS402_0x604C_VL_DIMENSION_FACTOR	0x604c	02 01 00 00 01 00 00 ...			UNIT_BYTEARRAY
DS402 - Quick Stop Option Code	DS402_0x605A_QUICK_STOP_OPTION_CODE	0x605a	(02) Abbremsen mit Schnellhalte- und We...			UNIT_I16
DS402 - Shutdown Option Code	DS402_0x605B_SHUTDOWN_OPTION_CODE	0x605b	(00) Antriebsfunktion sperren (Leistungsst...			UNIT_I16
DS402 - Disable Operation Option Code	DS402_0x605C_DISABLE_OPERATION_OPTION_CODE	0x605c	(00) Antriebsfunktion sperren			UNIT_I16
DS402 - Halt Option Code	DS402_0x605D_HALT_OPTION_CODE	0x605d	(01) Abbremsen mit Bremsrampe und Status ...			UNIT_I16
DS402 - Fault Reaction Option Code	DS402_0x605E_FAULT_REACTION_OPTION_CODE	0x605e	(01) Abbremsen mit Bremsrampe			UNIT_I16
DS402 - Mode of Operation	DS402_0x6060_MODE_OF_OPERATION	0x6060	(03) Profile Velocity Mode			UNIT_U8
DS402 - Mode of Operation Display	DS402_0x6061_MODE_OF_OPERATION_DISPLAY	0x6061	(03) Profile Velocity Mode			UNIT_U8
DS402 - Position Actual Internal Value	DS402_0x6063_POSITION_ACTUAL_INTERNAL_VALUE	0x6063	0			inkrement
DS402 - Position Actual Value	DS402_0x6064_POSITION_ACTUAL_VALUE	0x6064	0			UNIT_I32
DS402 - Following Error Window	DS402_0x6065_FOLLOWING_ERROR_WINDOW	0x6065	0			UNIT_U32
DS402 - Following Error Time Out	DS402_0x6066_FOLLOWING_ERROR_TIME_OUT	0x6066	0			ms
DS402 - Velocity Sensor Actual Value	DS402_0x6069_VELOCITY_SENSOR_ACTUAL_VALUE	0x6069	0			inc/s
DS402 - Velocity Demand Value	DS402_0x606B_VELOCITY_DEMAND_VALUE	0x606b	0			UNIT_I32
DS402 - Velocity Actual Value	DS402_0x606C_VELOCITY_ACTUAL_VALUE	0x606c	0			UNIT_I32
DS402 - Target Torque	DS402_0x6071_TARGET_TORQUE	0x6071	0		%	UNIT_I16
DS402 - Max Torque	DS402_0x6072_MAX_TORQUE	0x6072	10000		%	UNIT_U16
DS402 - Max Current	DS402_0x6073_MAX_CURRENT	0x6073	1188		%	UNIT_U16
DS402 - Torque Demand	DS402_0x6074_TORQUE_DEMAND	0x6074	0		%	UNIT_I16
DS402 - Motor Rated Current	DS402_0x6075_MOTOR_RATED_CURRENT	0x6075	10100		mA	UNIT_U32
DS402 - Motor Rated Torque	DS402_0x6076_MOTOR_RATED_TORQUE	0x6076	1447		mNm	UNIT_U32
DS402 - Torque Actual Value	DS402_0x6077_TORQUE_ACTUAL_VALUE	0x6077	0		%	UNIT_I16
DS402 - Current Actual Value	DS402_0x6078_CURRENT_ACTUAL_VALUE	0x6078	0		%	UNIT_I16

Abb. 29: *drivemaster4*-Objektbrowser

Beispiel

Hinweis

Ein TwinCAT3-Beispielprojekt zur Anbindung eines SD4x-Antriebs über EtherCAT finden Sie auf der SIEB & MEYER-Website unter <https://www.sieb-meyer.de/downloads>.

Aufgabe: Das Objekt **DS402 - Motor Rated Torque** soll aus einem SD4x-Antrieb über die SDO-Kommunikation ausgelesen werden.

Vorgehen:

- Suchen Sie das Objekt **DS402 - Motor Rated Torque** im Objektbrowser der *drivemaster4*-Software (siehe Abb. 29, Seite 50).
 - Die angezeigte ID (0x6076) ist der Index. Die Sub-ID ist der Subindex. Da das Objekt keine Subobjekte hat, ist der Subindex „0“.
 - Der Datentyp des Objekts ist UNIT_U32, dies steht für einen 32 Bit unsigned Integer.
- Legen Sie in der SPS nun eine Variable „motorRatedTorque“ mit dem Datentyp UDINT im VAR-Bereich eines Bausteins oder Programms an.
- Im pDstBuf-Eingang definieren Sie mit Hilfe der ADR(motorRatedTorque)-Funktion einen Pointer auf den Speicherbereich der motorRatedTorque-Variable.
- Im cbBufLen-Eingang geben Sie die Größe der Variable in Bytes an (in diesem Fall „4“ oder Größe per SIZEOF(motorRatedTorque)-Funktion auslesen).
- Für die ADS-Adresse legen Sie eine Variable vom Typ AMSADDR an und mappen diese auf die InfoData.AdsAddr des SD4x-EtherCAT-SubDevices.
Darin hinterlegt das EtherCAT-MDevice dann automatisch die aktuelle ADS-Adresse des SD4x-Antriebs.

6. Zur Diagnose legen Sie eine Variable `adsReturnCode` vom Typ `Tc2_System.E_AdsErr` an.

Im SPS-Anwenderprogramm könnte die Definition etwa so aussehen:

```
VAR.
adsAddress AT %I* : AMSADDR; // EtherCAT -> InfoData -> AdsAddr.
sdoReader : Tc2_EtherCAT.FB_EcCoESdoRead;
motorRatedTorque : UDINT;
adsReturnCode : Tc2_System.E_AdsErr;
END_VAR.

sdoReader(
  sNetId:= Tc2_System.F_CreateAmsNetId(adsAddress.netId),
  nSlaveAddr:= adsAddress.port,
  nSubIndex:= 0,
  nIndex:= 16#6076,
  pDstBuf:= ADR(motorRatedTorque),
  cbBufLen:= SIZEOF(motorRatedTorque),
  bExecute:= ,
  tTimeout:= ,
  bBusy=> ,
  bError=> ,
  nErrId=> adsReturnCode,
  cbRead=> );
```

Abb. 30: Motor Rated Torque auslesen

Resultat: Der Funktionsbaustein muss zyklisch aufgerufen werden. Sobald am Eingang `bExecute` eine steigende Flanke anliegt, wird die Kommunikation einmal gestartet. Wenn `bBusy` „FALSE“ ist, steht in der Variable `motorRatedTorque` der ausgelesene Wert. Wenn `bError` aktiv ist, steht in `adsReturnCode` ein Fehlercode.

Nach einem erfolgreichen Kommunikationszyklus sieht die Onlinebeobachtung etwa so aus:

Expression	Type	Value	Prepared va...	Address	Comment
* <code>adsAddress</code>	AMSADDR	(netId=[5.90.167.232.2.1], port...		%I*	EtherCAT -> InfoData -> Ad...
= <code>sdoReader</code>	Tc2_EtherCAT.FB_EcCoES...				
* <code>sNetId</code>	T_AmsNetId	'5.90.167.232.2.1'			AmsNetId of the EtherCAT ...
* <code>nSlaveAddr</code>	UINT	1002			Address of the slave device.
* <code>nSubIndex</code>	BYTE	0			CANopen Sdo subindex.
* <code>nIndex</code>	WORD	24694			CANopen Sdo index.
* <code>pDstBuf</code>	PVOID	3645393508			Contains the address of the ...
* <code>cbBufLen</code>	UDINT	4			Contains the max. number of...
* <code>bExecute</code>	BOOL	TRUE			Function block execution is t...
* <code>tTimeout</code>	TIME	T#5s			States the time before the fu...
* <code>bBusy</code>	BOOL	FALSE			
* <code>bError</code>	BOOL	FALSE			
* <code>nErrId</code>	UDINT	0			
* <code>cbRead</code>	UDINT	4			
* <code>nOffset</code>	DWORD	1618345984			
* <code>fbAdsReadEx</code>	ADSREDEX				
* <code>RisingEdge</code>	R_TRIG				
* <code>bSet0</code>	BOOL	FALSE			
* <code>motorRatedTorque</code>	UDINT	1447			
* <code>adsReturnCode</code>	E_ADSERR	NOERR			

```
1 sdoReader(
2   sNetId:= 5.90.167.2 := Tc2_System.F_CreateAmsNetId(adsAddress.netId 5.90.167.232.2.1),
3   nSlaveAddr:= 1002 := adsAddress.port 1002,
4   nSubIndex:= 0 := 0,
5   nIndex:= 24694 := 16#6076,
6   pDstBuf:= 3645393508 := ADR(motorRatedTorque 1447),
7   cbBufLen:= 4 := SIZEOF(motorRatedTorque 1447),
8   bExecute:= ,
9   tTimeout:= ,
10  bBusy=> ,
11  bError=> ,
12  nErrId:= 0 => adsReturnCode NOERR,
13  cbRead=> );
```

Abb. 31: Rückgabe für `motorRatedTorque`-SDO

Der parametrisierte Motor in diesem Beispiel hat also ein Nenndrehmoment von 1447 mNm.

3.2.3 Adressierung und Identifikation

In einem EtherCAT-Netzwerk gibt es verschiedene Verfahren zur Adressierung und Identifizierung eines SubDevices. Wechseln Sie in der TwinCAT3-Software auf den Reiter „EtherCAT“. Dort stehen folgende Identifikationsverfahren für die SD4x-Antriebe zur Verfügung:

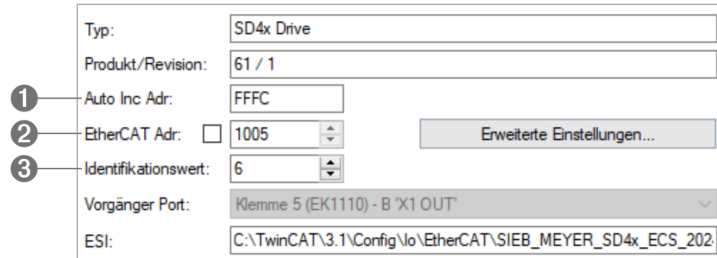


Abb. 32: TwinCAT3-Software – EtherCAT-Adressierung

- [1] Auto-Inkrement-Adresse
- [2] EtherCAT-Adresse
- [3] Identifikation über Hot Connect

Hinweis

Folgende Identifikationsverfahren werden von den Drive Controllern der Serie SD4x **nicht** unterstützt:

- ▶ Data Word
- ▶ Fast Hot Connect

3.2.3.1 Auto-Inkrement-Adresse

Die Auto-Inkrement-Adresse entspricht einer topologischen Adresse des EtherCAT-SubDevices. Dabei erhält das SubDevice seine Adresse anhand seiner physikalischen Position im Netzwerk.

Das EtherCAT-MDevice vergibt die Auto-Inkrement-Adresse während der Initialisierung der Buskommunikation. Das erste SubDevice im Netzwerk bekommt die Auto-Inkrement-Adresse 0 (0x0000). Für jedes weitere SubDevice wird der Wert dekrementiert, d. h. das zweite SubDevice hat die Adresse -1 (0xFFFF), das dritte SubDevice hat die Adresse -2 (0xFFFE) usw.

3.2.3.2 EtherCAT-Adresse

Die EtherCAT-Adresse entspricht der Feldbusadresse des SubDevices. Diese Adresse kann fest eingestellt sein oder automatisch während der Initialisierung der Buskommunikation vom MDevice vergeben werden.

Die folgende Abbildung zeigt die EtherCAT-Adressen bei automatischer Vergabe durch das MDevice. Das erste SubDevice erhält die Adresse 1001. Für jedes weitere SubDevice wird der Wert um 1 inkrementiert:

Nummer	Boxbezeichnung	Adresse
1	Klemme 6 (EK1200)	
2	Klemme 2 (EL1859)	1001
3	Klemme 8 (EL6731)	1002
4	Klemme 9 (EL6631)	1003
5	Klemme 5 (EK1110)	1004
6	Box 11 (SD4x Drive)	1005

Abb. 33: EtherCAT-Adressen bei automatischer Vergabe

In diesem Fall kann sich die EtherCAT-Adresse ändern, wenn ein neues SubDevice im EtherCAT-Netzwerk dazukommt.

Damit die Adresse unabhängig von der topologischen Position im Netzwerk ist, müssen Sie die Checkbox vor der EtherCAT-Adresse aktivieren:

EtherCAT Adr: ☒ 1003

Abb. 34: EtherCAT-Adresse festlegen

Dadurch erhält das SubDevice immer die gleiche Adresse. Für die nachfolgenden SubDevices wird diese Adresse inkrementiert, wenn für sie keine feste Adresse eingestellt ist:

Nummer	Boxbezeichnung	Adresse
1	Term 4 (EK1200)	
2	Term 2 (EK1110)	1001
3	Box 3 (SD4x Drive)	1003
4	Box 4 (SD4x Drive)	1004

Abb. 35: EtherCAT-Adressen bei fester Einstellung

EtherCAT-Adresse im SubDevice speichern

Sie können eine feste EtherCAT-Adresse für ein SubDevice im EEPROM speichern. Dafür muss bereits eine Verbindung zum SubDevice bestehen. Gehen Sie wie folgt vor:

1. Klicken Sie im EtherCAT-Reiter auf „Erweiterten Einstellungen“ und wechseln Sie im Dialog auf die Seite „Konfigurierte Station Alias“.

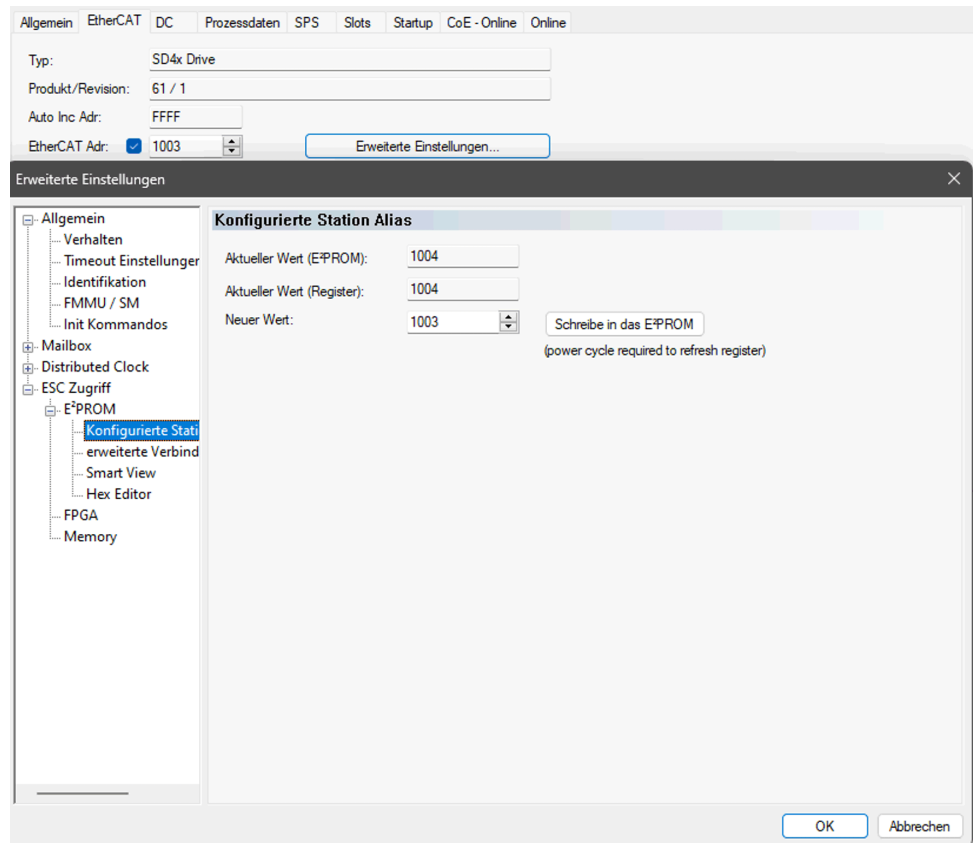


Abb. 36: TwinCAT3-Software – Erweiterte Einstellungen für EtherCAT

2. Tragen Sie die SubDevice-Adresse in „Neuer Wert“ ein und klicken Sie auf die Schaltfläche „Schreibe in das E²PROM“.
 3. Bestätigen Sie Ihre Eingabe mit „OK“.
- ✓ Das EtherCAT-SubDevice übernimmt beim nächsten Neustart den eingetragenen Wert in sein ESC-Register 0x0012.

3.2.3.3 Hot Connect

Hot Connect erlaubt eine von der Topologie unabhängige Zuordnung von EtherCAT-SubDevices anhand fest voreingestellter Adressen. Für die Verwendung von Hot Connect müssen Sie das SubDevice zuerst zu einer Hot-Connect-Gruppe hinzufügen:

- ⇒ Klicken Sie mit der rechten Maustaste auf das SubDevice und wählen Sie „Hot Connect Gruppe anfügen“ im Kontextmenü aus.

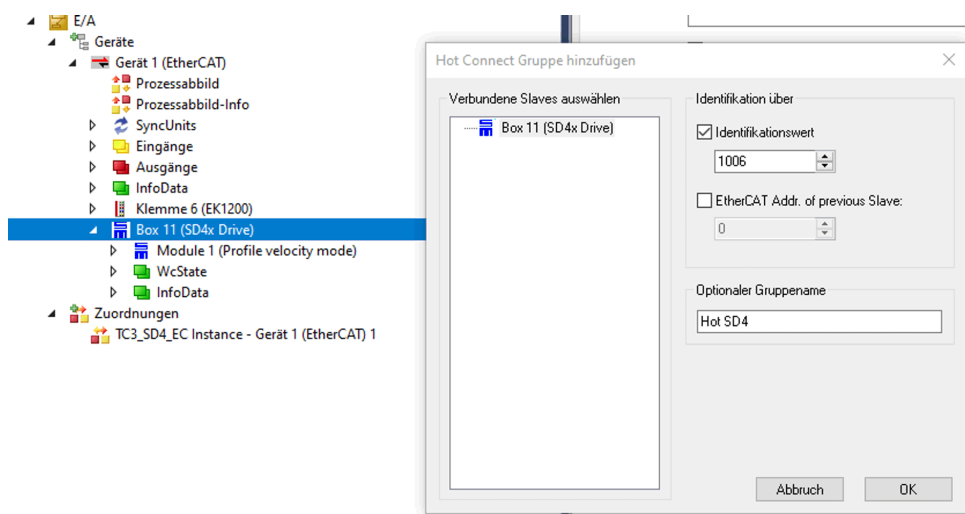


Abb. 37: TwinCAT3-Software – Hot-Connect-Gruppe hinzufügen

- Tragen Sie die Adresse der Hot-Connect-Gruppe im „Identifikationswert“ ein.
 - Wählen Sie ggf. einen Gruppennamen und bestätigen Sie mit „OK“.
- ✓ Das Gerät wurde zu der Hot-Connect-Gruppe hinzugefügt. Danach steht der Reiter „Hot Connect“ für das SubDevice zur Verfügung.

Um die SubDevices unabhängig von ihrer topologischer Position eindeutig zu identifizieren, müssen Sie ihnen eine festgelegte Adresse zuweisen. Dafür stehen die folgenden Identifikationsverfahren zu Verfügung:

- Configured Station Alias
- Explicit Device ID

Ist ein Identifikationsverfahren eingestellt, nimmt das MDevice das SubDevice nur in die zyklische Kommunikation auf, wenn dieser einen gültigen Identifikationswert hat. Auf diese Weise wird sichergestellt, dass bei identischen Geräten die richtige Konfiguration zum jeweiligen Gerät ausgewählt wird. Bei einem Gerätetausch muss der Identifikationswert im Ersatzgerät entsprechend eingestellt werden.

Configured Station Alias (ADO 0x0012):

Diese Adresse wird auch als „Second Station Address“ (SSA) bezeichnet.

1. Speichern Sie zunächst den gewünschten Identifikationswert im EEPROM des SubDevices wie im [Kapitel „EtherCAT-Adresse im SubDevice speichern“, Seite 53](#) beschrieben.

2. Klicken Sie dann im Reiter „Hot Connect“ auf die Schaltfläche „Konfigurieren“.

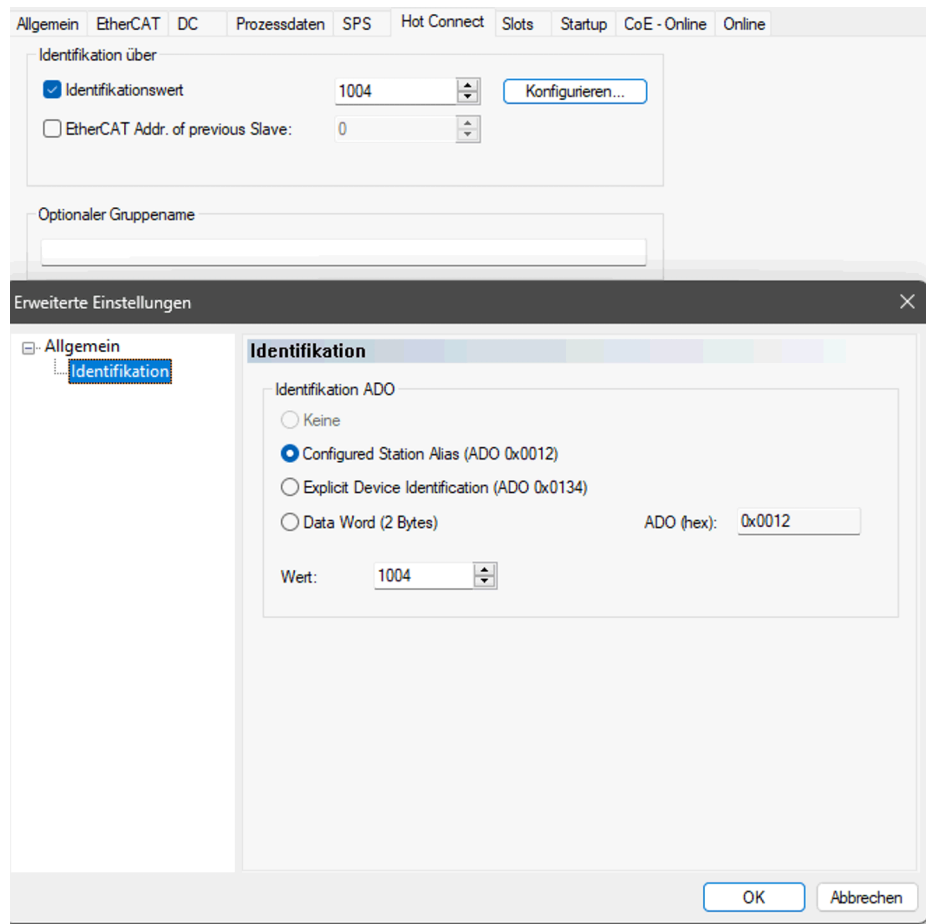


Abb. 38: TwinCAT3-Software – Configured Station Alias

- a. Wählen Sie unter Identifikation die Option „Configured Station-Alias (ADO 0x0012)“ und geben Sie den Identifikationswert des SubDevices ein.
 - b. Bestätigen Sie Ihre Eingaben mit OK.
- ✓ Das SubDevice hat nun einen gültigen Identifikationswert und wird vom EtherCAT-MDevice in die zyklische Kommunikation aufgenommen.

Explicit Device Identification (ADO 0x0134)

Bei diesem Verfahren muss der Identifikationswert im SubDevice mit dem Wert im EtherCAT-MDevice übereinstimmen.

Hinweis

SD4x-Antriebe unterstützen diese Funktion ab Parametersatzversion 1.1.12 (Objekt-ID 0x0029).

1. Stellen Sie zunächst die Identifikation im SD4x-Antrieb ein. Öffnen Sie dazu die *drivemaster4*-Software auf der Parameterseite „Feldbus“.

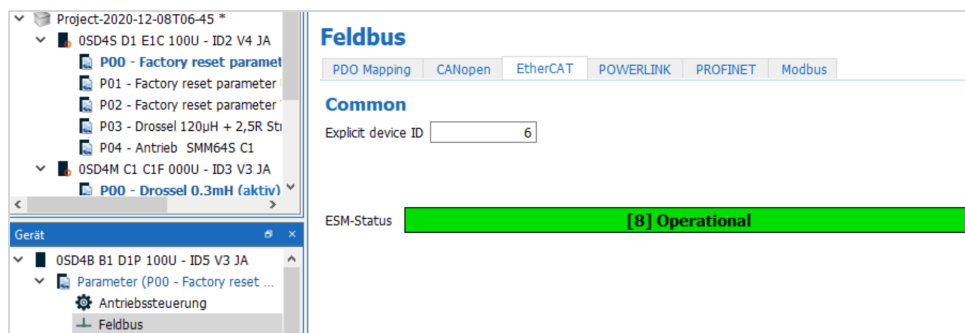


Abb. 39: drivemaster4-Software – EtherCAT-Parametrierung

- a. Geben Sie den gewünschten Identifikationswert im Feld „Explicit Device ID“ ein.

Der Wert 0 (Default) deaktiviert das Verfahren im Antrieb. Die Werte 1 bis 65535 aktivieren das Verfahren im Antrieb.

- b. Laden Sie die Parameter in den Antrieb.

2. Stellen Sie nun die Identifikation im EtherCAT-MDevice ein. Klicken Sie dazu im Reiter „Hot Connect“ auf die Schaltfläche „Konfigurieren“.

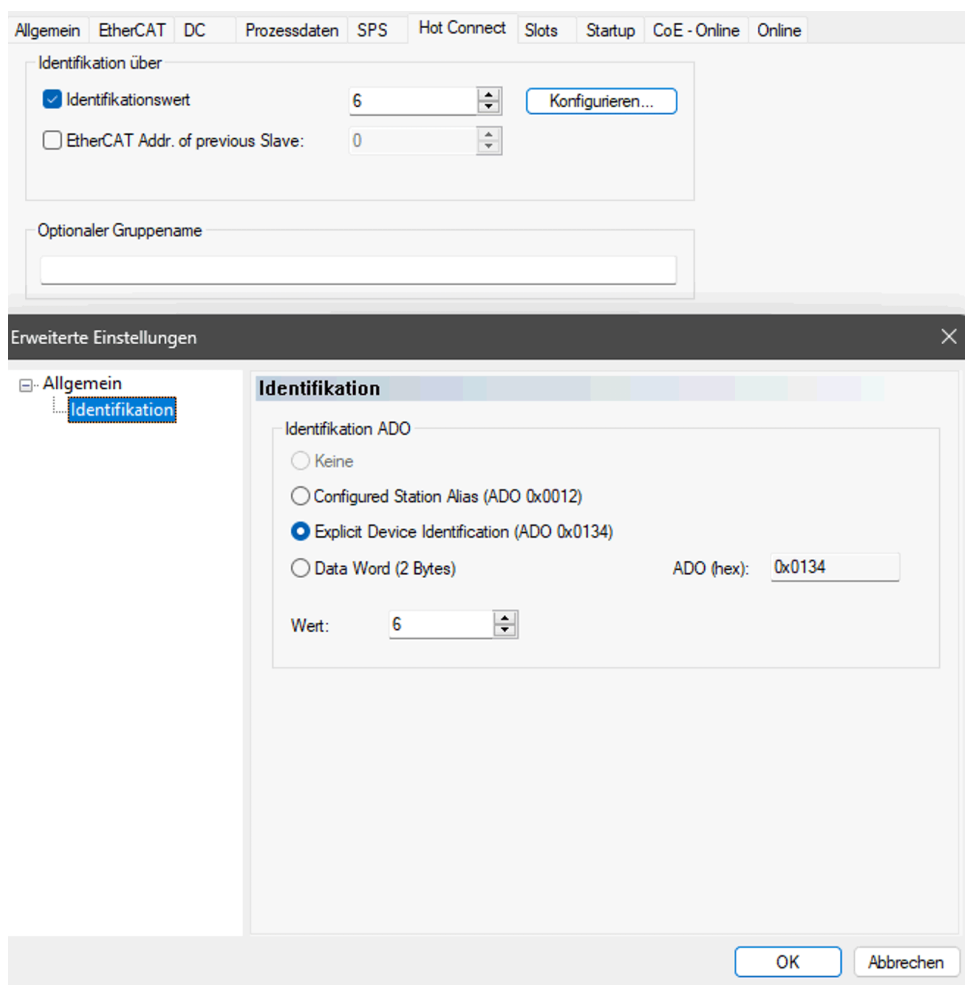


Abb. 40: TwinCAT3-Software – Explicit Device Identification

- a. Wählen Sie unter Identifikation die Option „Explicit Device Identification (ADO 0x0134)“.
 - b. Geben Sie im Feld „Wert“ die Identifikation des SubDevices ein und bestätigen Sie mit „OK“.
3. Wechseln Sie nun auf den Reiter „EtherCAT“ und klicken Sie auf „Erweiterten Einstellungen“. Wechseln Sie im Dialog auf die Seite „Verhalten“ und aktivieren Sie die Checkbox „Identifikation prüfen“..

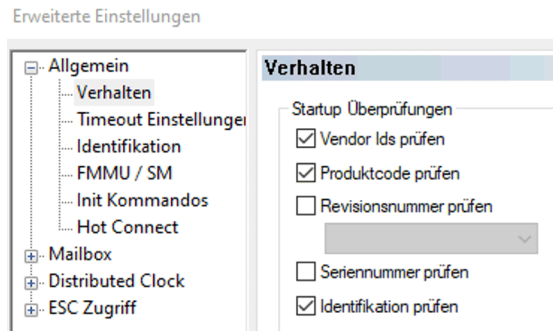


Abb. 41: TwinCAT3-Software – Verhalten einstellen

- ✓ Nun prüft das EtherCAT-MDevice beim Starten, ob der hier konfigurierte Identifikationswert mit dem im EtherCAT-SubDevice eingestellten Identifikationswert übereinstimmt. Wenn ja, nimmt das MDevice das SubDevice in die zyklische Kommunikation auf.

3.2.4 Allgemeine Objekte

Im Folgenden sind allgemeine Objekte aus dem DS301-Antriebsprofil und der EtherCAT-Spezifikation für die Adressierung und das PDO-Mapping beschrieben.

Objekt 0x1000 – Device Type

Dieses Objekt beschreibt die Art und die Funktion des Geräts.

Hinweis

Die SPS kann über die Objekte 0x1000 (Device Type) und 0x6502 (Supported Drive Modes) den Gerätetyp und die zur Verfügung stehenden Betriebsarten ermitteln. Über das Objekt 0x6060 (Mode of Operation) kann die SPS in die gewünschte Betriebsart wechseln, siehe [Kapitel 2.4.5 „Betriebsart wechseln“, Seite 23](#).

Index	0x1000
Name	Device Type
Objektcode	VAR
Datentyp	Unsigned 32

Zugriff	RO
PDO-Mapping	–
Einheit	–
Wertebereich	0 ... (2 ³² –1)
Defaultwert	0x00010192

Bit-Nummer									
31	...	24	23	...	16	15	...	0	
Zusätzliche Informationen						Geräteprofilnummer			
Modus-Bits			Typ						
Geräteprofilnummer:		0x0192			Geräteprofil DS402				
Zusätzliche Informationen:		0x01			Frequenzumrichter				

Objekt 0x1600 bis 0x1603 – RPDO Mapping Parameter

Diese Objekte enthalten die Mappingparameter für RPDO 0 bis RPDO 3 des Geräts.

Alle Werte müssen über das Bussystem parametrisiert werden.

Hinweis

Wenn ein PDO-Mapping für RPDO 0 in der *drivemaster4*-Software angelegt ist, wird dies von der SPS überschrieben.

Die Anzahl der Parameter wird im Subindex 0 angegeben. Die Subindexe 1 bis 8 enthalten entsprechend die Mappingparameter.

Index	0x1600 bis 0x1603
Name	RPDO Mapping Parameter
Objektcode	RECORD
Datentyp	PDO Mapping Parameter Record

Subindex	0
Name	Number of Mapped Objects
Datentyp	Unsigned 8
Zugriff	RW
PDO-Mapping	–
Einheit	–
Wertebereich	0 ... 8
Defaultwert (Profile Velocity Mode)	0x1600 (RPDO 0): 3 0x1601 (RPDO 1): 0 0x1602 (RPDO 2): 0 0x1603 (RPDO 3): 0
Beschreibung	Anzahl gemappter Objekte

Subindex	1 bis 8
Name	RPDO Mapping Parameter x
Datentyp	Unsigned 32
Zugriff	RW
PDO-Mapping	–
Einheit	–
Wertebereich	siehe Wertedefinition unten
Defaultwert (Profile Velocity Mode)	0x1600 (RPDO 0): 0x60400010, 0x60420010, 0x60720010 0x1601 (RPDO 1): 0, 0, 0, 0, 0, 0, 0, 0 0x1602 (RPDO 2): 0, 0, 0, 0, 0, 0, 0, 0 0x 1603 (RPDO 3): 0, 0, 0, 0, 0, 0, 0, 0
Beschreibung	Beschreibung des PDO (durch Index, Subindex und Länge)

Wertedefinition des Mapping-Parameters:

Bit-Nummer									
31	...	16	15	...	8	7	...	0	
Objektindex			Objekt-Subindex			Länge in Bits (8, 16, 32)			

Die folgende Abbildung zeigt das RPDO-0-Mapping (Objekt 0x1600) für den Profile Velocity Mode in der TwinCAT3-Software:

Abb. 42: TwinCAT3-Software – RPDO-0-Mapping

Objekt 1A00h bis 1A03h – TPDO Mapping Parameter

Diese Objekte enthalten die Mapping-Parameter für TPDO 0 bis TPDO 3 des Geräts.

Alle Werte müssen über das Bussystem parametrieren werden.

Hinweis

Wenn ein PDO-Mapping für TPDO 0 in der *drivemaster4*-Software angelegt ist, wird dies von der SPS überschrieben.

Die Anzahl der Parameter wird im Subindex 0 angegeben. Die Subindexe 1 bis 8 enthalten entsprechend die Mapping Parameter.

Index	1A00h bis 1A03h
Name	TPDO Mapping Parameter
Objektcode	RECORD
Datentyp	PDO Mapping Parameter Record

Subindex	0
Name	Number of Mapped Objects
Datentyp	Unsigned 8
Zugriff	RW
PDO-Mapping	–
Einheit	–
Wertebereich	0 ... 8
Defaultwert (Profile Velocity Mode)	1A00h (RPDO 0): 3 1A01h (RPDO 1): 0 1A02h (RPDO 2): 0 1A03h (RPDO 3): 0
Beschreibung	Anzahl gemappter Objekte

Subindex	1 bis 8
Name	TPDO Mapping Parameter x
Datentyp	Unsigned 8
Zugriff	RW
PDO-Mapping	–

Einheit	–
Wertebereich	siehe Wertedefinition unten
Defaultwert (Profile Velocity Mode)	1A00h (RPDO 0): 60400010h, 60420010h, 60720010h 1A01h (RPDO 1): 0, 0, 0, 0, 0, 0, 0, 0, 0 1A02h (RPDO 2): 0, 0, 0, 0, 0, 0, 0, 0, 0 1A03h (RPDO 3): 0, 0, 0, 0, 0, 0, 0, 0, 0
Beschreibung	Beschreibung des PDO (durch Index, Subindex und Länge)

Wertedefinition des Mapping-Parameters:

Bit-Nummer						
31	...	16	15	...	8	7
Objektindex		Objekt-Subindex			Länge in Bits (8, 16, 32)	

Die folgende Abbildung zeigt das TPDO-0-Mapping (Objekt 0x1A00) für den Profile Velocity Mode in der TwinCAT3-Software:

General EtherCAT DC Process Data Plc Slots Startup CoE - Online Online						
Sync Manager:				PDO List:		
SM	Size	Type	Flags	Index	Size	Name
0	128	MbxOut		0x1A00	12.0	TxPDO
1	128	MbxIn		0x1600	8.0	RxPDO
2	8	Outputs				
3	12	Inputs				
PDO Assignment (0x1C13):				PDO Content (0x1A00):		
☒ 0x1A00				Index	Size	Offs
				0x6041:00	2.0	0.0
				0x606C:00	4.0	2.0
				0x6064:00	4.0	6.0
				0x603F:00	2.0	10.0
						12.0

Abb. 43: TwinCAT3-Software – TPDO-0-Mapping

Objekt 0x1C00 – Sync Manager Communication Type

Index	0x1C00
Name	Sync Manager Communication Type
Objektcode	REC
Datentyp	siehe Subindex-Beschreibungen

Subindex	0
Name	Subindex 0
Zugriff	RO
PDO-Zuordnung	nicht möglich
Datentyp	Unsigned 8
Einheit	–
Wertebereich	4
Vorgabewert	4
Interne Objektnummer	–
Beschreibung	höchster unterstützter Subindex

Subindex	1 – 4
Name	PDO Mapping
Zugriff	RO
PDO-Zuordnung	nicht möglich

Datentyp	Unsigned 8
Einheit	–
Wertebereich	1 ... 4
Vorgabewert	siehe unten
Interne Objektnummer	–
Beschreibung	Art des Sync-Managers

Vorgabewerte der PDO-Mapping-Objekte

Objekt	Sync Manager Type (Objekt 0x1C00)	
Vorgabewert Subindex 0	4	Highest subindex supported (höchster unterstützter Sub-index)
Vorgabewert Subindex 1	1	Channel 1: Mailbox write (Kanal 1: Mailbox schreiben)
Vorgabewert Subindex 2	2	Channel 2: Mailbox read (Kanal 2: Mailbox lesen)
Vorgabewert Subindex 3	3	Channel 3: Process data write (Kanal 3: Prozessdaten schreiben)
Vorgabewert Subindex 4	4	Channel 4: Process Data read (Kanal 4: Prozessdaten lesen)

Objekt 0x1C12 – RPDO Assignment

Das Objekt beschreibt, welche PDOs als EtherCAT-Eingangsdaten übertragen werden.

Index	0x1C12
Name	RPDO Assignment
Objektcode	RECORD
Datentyp	siehe Subindex-Beschreibungen

Subindex	0
Name	Subindex 0
Zugriff	RO
PDO-Zuordnung	nicht möglich
Datentyp	Unsigned 8
Einheit	–
Wertebereich	1 ... 4
Vorgabewert	1
Interne Objektnummer	–
Beschreibung	höchster unterstützter Subindex

Subindex	1 bis 4
Name	Subindex x
Zugriff	RW
PDO-Zuordnung	nicht möglich
Datentyp	Unsigned 16
Einheit	–
Wertebereich	0x1600 ... 0x1603
Vorgabewert	0x1600
Interne Objektnummer	–
Beschreibung	PDO-Mapping für das n-te Mapping-Objekt

Objekt 0x1C13 – TPDO Assignment

Das Objekt beschreibt, welche PDOs als EtherCAT-Ausgangsdaten übertragen werden.

Index	0x1C13
Name	TPDO Assignment
Objektcode	RECORD

Datentyp	siehe Subindex-Beschreibungen
Subindex	0
Name	Subindex 0
Zugriff	RO
PDO-Zuordnung	nicht möglich
Datentyp	Unsigned 8
Einheit	–
Wertebereich	1 ... 4
Vorgabewert	1
Interne Objektnummer	–
Beschreibung	höchster unterstützter Subindex
Subindex	1 bis 4
Name	Subindex x
Zugriff	RW
PDO-Zuordnung	nicht möglich
Datentyp	Unsigned 16
Einheit	–
Wertebereich	0x1A00 ... 0x1A03
Vorgabewert	0x1A00
Interne Objektnummer	–
Beschreibung	PDO-Mapping für das n-te Mapping-Objekt

3.3 Unterstützte Synchronisationsarten

Der SD4x-Antrieb unterstützt folgende Synchronisationsarten:

- ▶ Free Run: Der Antrieb ist auf einen internen Takt synchronisiert und somit nicht synchron mit dem EtherCAT.
- ▶ DC-synchron (DC = Distributed Clocks): Der Antrieb wird basierend auf der DC-Einheit auf das Sync0-Ereignis synchronisiert. Dieses Signal ist auf wenige Nanosekunden taktgenau.

Die Synchronisationsart „SM-synchron“ (Synchron zu SyncManager-Ereignis) wird aufgrund der hohen Echtzeitanforderungen an die Motorregelung nicht unterstützt. In dieser Synchronisationsart wird der Antrieb auf das empfangende EtherCAT-Frame synchronisiert. Dieses Synchronisationssignal ist jedoch nur im Mikrosekundenbereich taktgenau (jittert).

Die im Antrieb verfügbaren Synchronisationsarten finden Sie in den SyncManager-Objekten 0x1C32 und 0x1C33, jeweils im Subobjekt „Synchronization Types supported“.

Die folgende Abbildung zeigt die Synchronisationsarten im Vergleich:

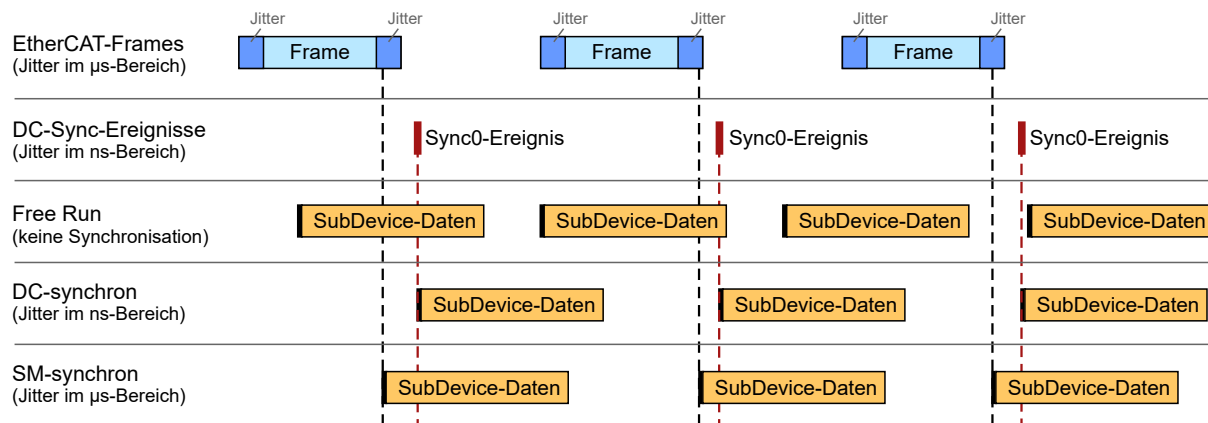


Abb. 44: EtherCAT-Synchronisationsarten

3.3.1 Free Run

Der Antrieb ist auf eine interne Uhr synchronisiert. Die EtherCAT-Prozessdaten werden beim Empfang eines EtherCAT-Telegramms zwischengespeichert und im nächsten internen Takt übernommen und verarbeitet.

Die Sollwertverarbeitung im Antrieb läuft mit einem festen Takt (standardmäßig 16 kHz) und ist somit nicht einstellbar.

Die momentan aktive Synchronisationsart finden Sie in den SyncManager-Objekten 0x1C32 und 0x1C33, jeweils im Subobjekt „Synchronization Type“. Außerdem gibt das Subobjekt „Cycle Time“ den aktuellen (internen) Takt als Zykluszeit in Nanosekunden zurück.

3.3.2 DC-Synchron (Distributed Clocks)

Der Antrieb und das EtherCAT-MDevice sind auf die Distributed Clocks (DC) synchronisiert. Sie laufen somit taktsynchron. Die Sollwertverarbeitung des Antriebs behält ihren eigenen Takt bei (standardmäßig 16 kHz). Jedoch werden die Sollwerte erst zum nächsten Sync0-Takt übernommen. Dadurch werden die Sollwerte immer genau zum gleichen Zeitpunkt von MDevice und SubDevice verarbeitet (Abweichungen nur im Nanosekundenbereich). Zudem ist die Verarbeitung synchronisiert mit allen anderen DC-synchronen EtherCAT-Teilnehmern.

Damit der Antriebstakt auf den Sync0-Takt synchronisiert werden kann, muss die Sync0-Zykluszeit ein Vielfaches der minimalen Zykluszeit für die Sollwertverarbeitung des Antriebs sein: $\text{Sync0-Zykluszeit} = 2^n \times \text{minimale Zykluszeit}$.

Bei der DC-Synchronisierung können zurzeit folgende Sync0-Takte verwendet werden:

Sync0-Takt	Resultierende Sync0-Zykluszeit
4 kHz	250 µs
2 kHz	500 µs
1 kHz	1 ms
500 Hz	2 ms
250 Hz	4 ms
125 Hz	8 ms

Die minimale Zykluszeit kann aus den SyncManager-Objekten 0x1C32 und 0x1C33 ausgelesen werden, jeweils aus dem Subobjekt „Minimum Cycle Time“. In den Zuständen Pre-Operational und Operational der EtherCAT State Machine (ESM) ist der Antrieb bereit zum Synchronisieren. Werden Sync0-Signale im richtigen Takt erkannt, läuft der Antrieb Sync0-synchron. Ist das nicht der Fall, bleibt der Antrieb im Free Run.

Die momentan aktive Synchronisationsart finden Sie in den SyncManager-Objekten 0x1C32 und 0x1C33, jeweils im Subobjekt „Synchronization Type“. Außerdem kann über dieses Subobjekt eine gewünschte Synchronisationsart angefordert werden. Sofern die Anforderung möglich ist, wird in die neue Synchronisationsart gewechselt. Die Synchronisationsart bleibt nach einem erfolgreichem Wechsel bis zur nächsten Anforderung oder dem nächsten automatischen Wechsel (ESM-Zustand wechselt) erhalten.

⚠️ WARNUNG



Änderung der Synchronisation im laufenden Betrieb

Der Antrieb benötigt eine gewisse Zeit (bis zu mehreren Sekunden), um sich auf einen Sync0-Takt zu synchronisieren. Deshalb kann ein Umschalten des Sync0-Taktes oder der Synchronisationsart im laufenden Betrieb zu schwerwiegenden Fehlern in der Antriebsregelung führen. Unter Umständen sind Personen- und Sachschäden die Folge.

→ Sperren Sie den Betrieb bevor Sie den Sync0-Takt oder die Synchronisationsart ändern.

Kann der Antrieb nicht auf das Sync0-Signal synchronisiert werden, wird in den SyncManager-Objekten das Subobjekt „Sync Error“ aktiv. Dieses Subobjekt darf als PDO gemappt werden.

Wenn der Antrieb erfolgreich synchronisiert wurde, steht der Sync0-Takt als Zykluszeit in den SyncManager-Objekten, jeweils im Subobjekt „Cycle Time“ in Nanosekunden.

4 Antriebsparametrierung und Diagnose

Im Folgenden wird die Parametrierung und Diagnose eines SD4x-Antriebs mit der Software *drivemaster4* beschrieben.

4.1 Parametrierung in *drivemaster4*

Bevor Sie den SD4x-Antrieb in einem EtherCAT-Netzwerk betreiben können, müssen Sie die EtherCAT-Kommunikation mit Hilfe der Parametriersoftware *drivemaster4* konfigurieren.

1. Öffnen Sie in der *drivemaster4*-Software im Gerätebaum unter „Parameter“ die Seite „Antriebssteuerung“.
2. Wählen Sie für den Steuerkanal und damit auch für den Sollwertkanal (Hauptsollwert) den Eintrag „EtherCAT CoE“ aus.



Abb. 45: EtherCAT-Parametrierung in *drivemaster4*

Die eingestellte Betriebsart „(03) Geschwindigkeitsmodus PV“ entspricht dem [Profile Velocity Mode \(S. 16\)](#). Für den [Velocity Mode \(S. 17\)](#) parametrieren Sie die Betriebsart „(02) Geschwindigkeitsmodus VL“.

- ✓ Der SD4x-Antrieb erwartet nun, dass er über EtherCAT zyklisch mit Sollwerten (PDOs – Prozessdatenobjekte) versorgt wird.

Die azyklische Kommunikation ist unabhängig vom Sollwertkanal und funktioniert immer.

Hinweis

Für EtherCAT sind keine Einstellungen auf der Parameterseite „Feldbus“ notwendig. Das PDO-Mapping wird vom EtherCAT-MDevice (SPS) durchgeführt und an den SD4x-Antrieb gesendet.

4.2 Diagnose in *drivemaster4*

In der *drivemaster4*-Oberfläche werden auf der Seite „Diagnose → Feldbus“ unter dem Reiter „EtherCAT“ die übertragenen Telegramme zwischen dem Antrieb und der SPS dargestellt.

4.2.1 Telegramme

Die folgende Abbildung zeigt die protokollierten Telegramme auf der EtherCAT-Diagnoseseite.

Feldbus							
CANopen EtherCAT POWERLINK PROFINET IO Modbus Diagnose							
Name	Zähler	Zeitstempel	Msgs/s	Dir	Cmd	Länge	Status
RPDO 0							
Data : 06 00 00 00 00 00 00 00							
Controlword			6		0006h		
Target Velocity			0		00000000h		
Max Torque			0		0000h		
TPDO 0	86403	136742.29	1000	Tx		12	
Data : 40 92 00 00 00 00 00 00 00 00 00 00							
Statusword		37440			9240h		
Velocity Actual Value		0			00000000h		
Position Actual Value		0			00000000h		
Error Code		0			0000h		
Link_Status_Change	2	5448.46	0	Rx	2f8ah	32	0h
Link_Status_Change	2	5448.51	0	Tx	2f8bh	0	0h
ESM_AlStatus_Changed	3	7125.46	0	Rx	19deh	6	0h
ESM_AlStatus_Changed	3	7125.47	0	Tx	19dfh	0	0h
ESM_AL_Control_Changed	3	7119.45	0	Rx	1b1ch	18	0h
ESM_AL_Control_Changed	3	7120.93	0	Tx	1b48h	4	0h
Write_Object	11	7089.45	0	Rx	6a02h	12	0h
Write_Object	11	7089.47	0	Tx	6a03h	3	0h
Write_Object_Validation_	7	7092.45	0	Rx	6a2ah	4	0h
Write_Object_Validation_	7	7092.46	0	Tx	6a2bh	3	0h
Get_Object_Access_Info	1	7086.45	0	Rx	6a16h	4	0h
Get_Object_Access_Info	1	7086.48	0	Tx	6a17h	10	0h

◀ Von der Steuerung
 ▶ Zur Steuerung

ESM Status [8] Operational

Abb. 46: EtherCAT-Busaktivitäten

Ausgabefenster

Die Daten der zyklischen Telegramme RPDO x und TPDO x können durch Aufklappen der Zeilen in ihre parametrisierten Objekte aufgeschlüsselt werden.

Die azyklischen Telegramme werden paarweise als Anfrage- und Antworttelegramm dargestellt. Die Nutzdaten sind nicht dargestellt, da hier zum Teil über 1000 Bytes übertragen werden und somit die Übersichtlichkeit verloren ginge.

Für die Telegramme werden die folgenden Informationen im Ausgabefenster angezeigt:

Name	Name des Telegramms
Zähler	Zähler der Häufigkeit des Telegramms
Zeitstempel	antriebsinterner Zeitstempel
Msgs/s	Anzahl der gesendeten Telegramme pro Sekunde

Dir	Richtung des Telegramms aus Sicht des Antriebs:
	▸ Rx: Telegramm wurde vom Antrieb empfangen. ▸ Tx: Telegramm wurde vom Antrieb gesendet.

Zusätzliche Informationen bei den azyklischen Telegrammen:

Cmd	Befehlsnummer des azyklischen Telegramms
Length	Länge des Telegramms in Byte
Status	Fehlerstatus für die Ausführung des Befehls

Löschen

Über die Schaltfläche „Löschen“ können Sie das Ausgabefenster leeren. Anschließend startet die Protokollierung der erkannten Nachrichten neu.

ESM Status

Dieses Feld zeigt den Status der EtherCAT State Machine an. Mögliche Zustände sind: Init, Pre-Operational, Bootstrap, Safe-Operational, Operational.

4.3 Problembehandlung: Zustandsmaschine ignoriert Steuerbefehle

Wenn der Zustand der Endstufe durch ein äußeres Ereignis geändert wird, werden die Befehle aus dem Controlword der Zustandsmaschine überschrieben, solange das Ereignis aktiv ist. Die folgenden Ereignisse sind die häufigsten Gründe solcher Zustandswechsel:

1. Sperrender digitaler Eingang
 - Die digitalen Eingangsfunktionen **(01) Regler Ein** und **(02) Regler Ein statisch** sind UND-verknüpft mit **Steuerbit 0 (Einschalten)** des Controlwords. Daher können sie das Einschalten über den Feldbus verhindern.
 - Die digitale Eingangsfunktion **(05) Schnellhalt** ist UND-verknüpft mit **Steuerbit 2 (Schnellhalt)** des Controlwords. Daher kann sie die Schnellhaltfunktion unabhängig vom Feldbus aktivieren.

Zur Fehlerbehebung können Sie den sperrenden Eingang in der Parametrierung abwählen, siehe Dokumentation „Drive Controller SD4x – Funktionen und Parameter“, Kapitel „Digitale Eingänge“.
2. Fehlende Leistungsversorgung

Wird die Einspeisung unterbrochen, reagiert der SD4x-Antrieb wie folgt:

 - Bei eingeschalteter Endstufe: Der Antrieb schaltet mit dem Fehler E48 ab.
 - Bei ausgeschalteter Endstufe: Der Antrieb wechselt selbstständig in den Zustand „Einschaltsperr“.
3. Sicherheitsfunktion STO (Safe Torque Off)

Wird die Sicherheitsfunktion STO ausgelöst, reagiert der SD4x-Antrieb wie folgt:

 - Bei eingeschalteter Endstufe: Der Antrieb schaltet mit dem Fehler E18 ab.
 - Bei ausgeschalteter Endstufe: Der Antrieb wechselt selbstständig in den Zustand „Einschaltsperr“.

Weitere Informationen zur Sicherheitsfunktion STO finden Sie in der Hardwarebeschreibung Ihres SD4x-Antriebs, Kapitel „Funktionale Sicherheit“.
4. Steuerung über *drivemaster4*-Inbetriebnahmewerkzeug

Wenn die Steuerung des SD4x-Antriebs über das *drivemaster4*-Inbetriebnahmewerkzeug aktiv ist, kann das Gerät nicht über Feldbus gesteuert werden. In diesem Fall ist das **Statusbit 8 (Drive Setup Tool (DRS) Active)** gesetzt.



Tipp

In der *drivemaster4*-Software werden unter „Tools/Diagnose → Antriebsistwerte → Freigeben/Sperren“ alle Hardware- und Softwarekomponenten angezeigt, die zur Betriebsfreigabe über Feldbus benötigt werden.

5 Weiterführende Informationen

Weitere Dokumente

Die folgenden Dokumente enthalten weitere Informationen zu diesem Thema:

Anbieter	Dokument
SIEB & MEYER AG	▸ drivemaster4 – Bedienen ▸ Antriebssystem SD4 – Ethernet-Konfiguration
EtherCAT Technology Group	▸ ETG.1000 1 - 6 EtherCAT Services and Protocols ▸ ETG.2000 EtherCAT SubDevice Information (ESI), Specification and Implementation
CiA e.V.	▸ CiA 301, Version 4.2.0. CANopen Application Layer and Communication Profile ▸ CiA 402, Version 5.0.0. CANopen Device Profile Drives and Motion Control ▸ CiA 890, Version 5.0.0. Presentation of SI Units and Prefixes

Websites

Die folgenden Websites enthalten weitere Informationen zu diesem Thema:

Anbieter	Dokument
SIEB & MEYER AG	www.sieb-meyer.de
EtherCAT Technology Group	www.ethercat.org
CiA e.V.	www.can-cia.org
Beckhoff Automation GmbH & Co. KG	www.beckhoff.com

Anhang

A Präfixe (aus CiA 303-2)

Index	Symbol	Faktor
0x13 ... 0x7F	Reserviert	
0x12	E	10^{18}
0x11	-	10^{17}
0x10	-	10^{16}
0x0F	P	10^{15}
0x0E	-	10^{14}
0x0D	-	10^{13}
0x0C	T	10^{12}
0x0B	-	10^{11}
0x0A	-	10^{10}
0x09	G	10^9
0x08	-	10^8
0x07	-	10^7
0x06	M	10^6
0x05	-	10^5
0x04	-	10^4
0x03	k	10^3
0x02	h	10^2
0x01	da	10^1
0x00	-	10^0
0xFF	d	10^{-1}
0xFE	c	10^{-2}
0xFD	m	10^{-3}
0xFC	-	10^{-4}
0xFB	-	10^{-5}
0xFA	μ	10^{-6}
0xF9	-	10^{-7}
0xF8	-	10^{-8}
0xF7	n	10^{-9}
0xF6	-	10^{-10}
0xF5	-	10^{-11}
0xF4	p	10^{-12}
0xF3	-	10^{-13}
0xF2	-	10^{-14}
0xF1	f	10^{-15}
0xF0	-	10^{-16}
0xEF	-	10^{-17}
0xEE	a	10^{-18}
0xED bis 0x80	Reserviert	

B Einheiten (aus CiA 303-2 und CiA 402)

Index	Symbol	Faktor
0x00	Dimensionslos	–
0x01	m	Meter
0x02	kg	Kilogramm
0x03	s	Sekunde
0x04	A	Ampere
0x05	K	Kelvin
0x06	mol	Mol
0x07	cd	Candela
0x10	rad	Radiant
0x11	sr	Steradian
0x20	Hz	Hertz
0x21	N	Newton
0x22	Pa	Pascal
0x23	J	Joule
0x24	W	Watt
0x25	C	Coulomb
0x26	V	Volt
0x27	F	Farad
0x28	Ω	Ohm
0x29	S	Siemens
0x2A	Wb	Weber
0x2B	T	Tesla
0x2C	H	Henry
0x2D	°C	Grad Celsius
0x2E	lm	Lumen
0x2F	lx	Lux
0x30	Bq	Becquerel
0x31	Gy	Gray
0x32	Sv	Sievert
0x33	kat	Katal
0x40	g	Gon
0x41	°	Grad
0x42	'	Bogenminute
0x43	"	Bogensekunde
0x44	l oder L	Liter
0x45	a	Ar
0x46	ha	Hektar
0x47	min	Minute
0x48	h	Stunde
0x49	d	Tag
0x4A	a	Jahr
0x4B	g	Gramm
0x4C	t	Tonne (metrisch)
0x4E	bar	Bar
0x4F	P	Poise
0x50	St	Stokes
0x51	eV	Elektronenvolt
0x52	u	atomare Masseneinheit
0x53	AU	astronomische Einheit
0x54	pc	Parsec
0x55	m/s^2	Meter pro Quadratsekunde
0x56	Nm	Newtonmeter

Index	Symbol	Faktor
0x57	s^2	Quadratsekunde
0x58	m^2	Quadratmeter
0x59	m^3	Kubikmeter
0x5A	$Pa \cdot s$	Pascalsekunde
0x5B	$J/(kg \cdot K)$	Joule pro Kilogramm mal Kelvin
0x5C	$W/(m \cdot K)$	Watt pro Meter mal Kelvin
0x5D	$J/(mol \cdot K)$	Joule pro Mol mal Kelvin
0x5E	$W/(m^2 \cdot sr)$	Watt pro Quadratmeter mal Steradian
0x5F	kat/m^3	Katal pro Kubikmeter
0xA0	s^3	Kubiksekunde
0xB4	–	Umdrehung
0xB5	–	Inkrement
0xAC	–	Schritt

Index

A

Abkürzungen [5](#)
Adressierung [40](#)
Adressierung (TC3) [52](#)
Auto-Inkrement-Adresse (TC3) [52](#)

B

Betriebsarten [14](#)
Betriebsartwechsel [23](#)

C

Configured Station-Alias (TC3) [55](#)
Controlword [10](#)
CSV Mode [14](#)
Cyclic Synchronous Velocity Mode [14](#)

D

DC-synchron [64](#)
Diagnose
 drivemaster4 [67](#)
Drehmomentbetrieb [23](#)

E

ESM-Zustände [46](#)
EtherCAT-Adresse (TC3) [52](#)
Explicit Device Identification [56](#)

F

Free Run [64](#)

G

Geschwindigkeitsmodus PV [16](#)
Geschwindigkeitsmodus VL [17](#)

H

Hot Connect (TC3) [54](#)

O

Objekte

Abort Connection Option Code [13](#)
Acceleration [31](#)
Controlword [10](#)
Current Actual Value [28](#)
DC Link Circuit Voltage [29](#)
Device Type [58](#)
Digital Inputs [35](#)
Digital Outputs [35](#)
Disable Operation Option Code [13](#)
Error Code [36](#)
Fault Reaction Option Code [14](#)
Halt Option Code [14](#)
Interpolation Time Period [26](#)
Jerk [31](#)
Max Acceleration [34](#)
Max Current [33](#)
Max Deceleration [34](#)
Max Motor Speed [33](#)
Max Profile Velocity [33](#)
Max Torque [32](#)
Mode of Operation [25](#)
Mode of Operation Display [25](#)
Motion Profile Type [27](#)
Motor Rated Current [29](#)
Motor Rated Torque [29](#)
Motor Type [29](#)
Negative Torque Limit [33](#)
Polarity [31](#)
Position [30](#)
Position Actual Value [27](#)
Positive Torque Limit [33](#)
Profile Acceleration [34](#)
Profile Deceleration [34](#)
Quick Stop Deceleration [34](#)
Quick Stop Option Code [13](#)
RPDO Mapping Parameter [59](#)
Shutdown Option Code [13](#)

Statusword [11](#)
Supported Drive Modes [24](#)
Target Torque [26](#)
Target Velocity [26](#)
Torque Actual Value [28](#)
Torque Demand [28](#)
Torque Offset [32](#)
Torque Slope [35](#)
TPDO Mapping Parameter [60](#)
Velocity [30](#)
Velocity Actual Value [28](#)
Velocity Demand Value [28](#)
Velocity Offset [32](#)
vI Dimension Factor [22](#)
vI Set-point Factor [21](#)
vI Target Velocity [18](#)
vI Velocity Acceleration [19](#)
vI Velocity Actual Value [18](#)
vI Velocity Deceleration [20](#)
vI Velocity Demand [18](#)
vI Velocity Min Max Amount [18](#)
vI Velocity Quick Stop [20](#)
Option Codes [12](#)

P

Parametersatzumschaltung [37](#)
PDO [8](#)
Profile Velocity Mode [16](#)

S

SDO [7](#)
SDO-Kommunikation (TC3) [48](#)
Statusword [11](#)
Steuerbefehle Antrieb [10](#)
Synchronisationsarten [63](#)

T

Torque Mode [23](#)
TwinCAT3-Anbindung [40](#)

V

Velocity Mode [17](#)

Z

Zustände [11](#)
Zustandsmaschine [9](#)
 Problembehandlung [68](#)
Zustandsübergänge des Antriebs [9](#)