

Drive Controller SD4x

PROFINET IO-Anbindung

Das Einstiegshandbuch für den Anwender





Copyright

Originalbetriebsanleitung, Copyright © 2024 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
Fax: +49 4131 203 2000
info@sieb-meyer.de
<http://www.sieb-meyer.de>

SIEB & MEYER Shenzhen Trading Co. Ltd.
Room A208 2/F,
Internet Innovation and Creation Services Base Building (2),
No.126, Wanxia road, Shekou, Nanshan district,
Shenzhen City, 518067
P.R. China

Tel.: +86 755 2681 1417 / +86 755 2681 2487
Fax: +86 755 2681 2967
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
Fax: +886 3 322 1224
info@sieb-meyer.tw

1	Über dieses Handbuch.....	5
1.1	Abkürzungen.....	5
2	Parametrierung in drivemaster4.....	6
3	Ansteuerung über PROFINET IO.....	8
3.1	PROFINET IO-Netzwerk.....	8
3.2	Software-Anbindung an die SPS.....	8
3.3	PROFINET IO-Netzwerkmanagement (NMT).....	15
3.4	Zyklische Prozessdaten (PDO).....	15
3.4.1	Sollwerttelegramm.....	16
3.4.2	Istwerttelegramm.....	16
3.5	Azyklische Servicedaten (SDO).....	17
3.5.1	Schreib- und Lesevorgänge.....	19
3.5.2	Programmbeispiel für Siemens-SPS.....	20
4	Antriebssteuerung.....	25
4.1	Zustandsmaschine.....	26
4.1.1	Befehle.....	27
	Controlword (0x6040).....	27
4.1.2	Zustände.....	28
	Statusword (0x6041).....	28
4.1.3	Option Codes.....	29
	Abort Connection Option Code (0x6007).....	29
	Quick Stop Option Code (0x605A).....	30
	Shutdown Option Code (0x605B).....	30
	Disable Operation Option Code (0x605C).....	30
	Halt Option Code (0x605D).....	30
	Fault Reaction Option Code (0x605E).....	31
4.2	Betriebsarten.....	31
4.2.1	Profile Velocity Mode (PV).....	32
4.2.2	Velocity Mode (VL).....	33
4.2.2.1	Soll- und Istwerte im Velocity Mode.....	34
	vl Target Velocity (0x6042).....	34
	vl Velocity Demand (0x6043).....	34
	vl Velocity Actual Value (0x6044).....	34
	vl Velocity Min Max Amount (0x6046).....	34
	vl Velocity Acceleration (0x6048).....	35
	vl Velocity Deceleration (0x6049).....	36
	vl Velocity Quick Stop (0x604A).....	36
	vl Set-point Factor (0x604B).....	37
	vl Dimension Factor (0x604C).....	38
4.2.3	Torque Mode.....	39
4.2.4	Betriebsart wechseln.....	39
	Supported Drive Modes (0x6502).....	40
	Mode of Operation (0x6060).....	40
	Mode of Operation Display (0x6061).....	41
4.3	Soll- und Istwerte.....	41
4.3.1	Allgemeine Sollwerte.....	41
	Target Velocity (0x60FF).....	41
	Target Torque (0x6071).....	42
4.3.2	Allgemeine Istwerte.....	42
	Position Actual Value (0x6064).....	42
	Velocity Demand Value (0x606B).....	42
	Velocity Actual Value (0x606C).....	42
	Torque Demand (0x6074).....	43
	Torque Actual Value (0x6077).....	43
	Current Actual Value (0x6078).....	43
	DC Link Circuit Voltage (0x6079).....	43
4.3.3	Motorparameter.....	43
	Motor Rated Current (0x6075).....	43
	Motor Rated Torque (0x6076).....	43
	Motor Type (0x6402).....	44
4.3.4	SI-Einheiten.....	44



	Position (0x60A8).....	44
	Velocity (0x60A9).....	45
	Acceleration (0x60AA).....	45
	Jerk (0x60AB).....	45
4.3.5	Skalierung.....	46
	Polarity (0x607E).....	46
4.3.6	Offsets.....	46
	Velocity Offset (0x60B1).....	46
	Torque Offset (0x60B2).....	47
4.3.7	Begrenzungen.....	47
	Max Torque (0x6072).....	47
	Max Current (0x6073).....	47
	Max Profile Velocity (0x607F).....	47
	Max Motor Speed (0x6080).....	47
	Positive Torque Limit (0x60E0).....	48
	Negative Torque Limit (0x60E1).....	48
4.3.7.1	Beschleunigungs- und Verzögerungsbegrenzungen.....	48
	Max Acceleration (0x60C5).....	48
	Max Deceleration (0x60C5).....	48
	Profile Acceleration (0x6083).....	49
	Profile Deceleration (0x6084).....	49
	Quick Stop Deceleration (0x6085).....	49
	Torque Slope (0x6087).....	49
4.3.8	Sonstige.....	49
	Digital Inputs (0x60FD).....	49
	Digital Outputs (0x60FE).....	50
	Error Code (0x603F).....	51
4.4	Parametersatzumschaltung.....	51
4.4.1	Parametersatzumschaltung über Feldbus.....	52
5	Diagnose.....	53
5.1	Diagnose in drivemaster4.....	53
5.1.1	Telegramme.....	53
5.1.2	Diagnosedaten.....	55
5.2	Diagnose in der SPS.....	56
6	Weiterführende Informationen.....	57
7	Index.....	58

1 Über dieses Handbuch

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

Eine grundlegende Beschreibung des PROFINET IO-Netzwerks finden Sie auf der Website der PROFIBUS Nutzerorganisation e.V. (PNO) (www.profibus.com/).

Das implementierte Protokoll 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.

Die hier beschriebene PROFINET IO-Anbindung stellt folgende Kommunikationsmechanismen 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, siehe [Abschnitt „Weitere Dokumente“, Seite 57](#).

1.1 Abkürzungen

CAN	C ontroller A rea N etwork
CiA	C AN in A utomation
GSDML	G eneral S tation D escription X ML File (Gerätebeschreibungsdatei im XML-Format)
NMT	N etzwerk m anagement
PDO	P rozess d aten o bjekt
PNS	P ROFINET S tation
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 ervicedaten o bjekt
SPS	s peicher p rogrammierbare S teuerung
Tx	Transmit (senden)
VL	V elocity Mode

2 Parametrierung in *drivemaster4*

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

Antriebssteuerung

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 „PROFINET IO“ aus.

Abb. 1: PROFINET IO-Parametrierung in *drivemaster4*

Die eingestellte Betriebsart „(03) Geschwindigkeitsmodus PV“ entspricht dem Profile Velocity Mode. Für den Velocity Mode parametrieren Sie die Betriebsart „(02) Geschwindigkeitsmodus VL“.

Feldbus

3. Öffnen Sie die Parameterseite „Feldbus“ im Gerätebaum der *drivemaster4*-Software.
4. Öffnen Sie den Reiter „PDO Mapping“ und erstellen Sie das Mapping entsprechend der gewählten Betriebsart.

→ Die folgende Abbildung zeigt das Mapping für den Profile Velocity Mode:

Feldbus

PDO Mapping
CANopen
EtherCAT
POWERLINK
Profinet
Modbus

RPDO 0 Mapping

	Index	Größe [Byte]	Offset [Byte]	Name	Datentyp
Objekt 1	0x6040:00	2	0	Controlword	UNIT_U16
Objekt 2	0x60ff:00	4	2	Target Velocity	UNIT_I32
Objekt 3	0x6072:00	2	6	Max Torque	UNIT_U16
			8		

TPDO 0 Mapping

	Index	Größe [Byte]	Offset [Byte]	Name	Datentyp
Objekt 1	0x6041:00	2	0	Statusword	UNIT_U16
Objekt 2	0x606c:00	4	2	Velocity Actual Value	UNIT_I32
Objekt 3	0x6064:00	4	6	Position Actual Value	UNIT_I32
			10		

Abb. 2: Beispiel-Mapping Profile Velocity Mode

Die folgende Abbildung zeigt das Mapping für den Velocity Mode:

Feldbus

PDO Mapping
CANopen
EtherCAT
POWERLINK
Profinet
Modbus

RPDO 0 Mapping

	Index	Größe [Byte]	Offset [Byte]	Name	Datentyp
Objekt 1	0x6040:00	2	0	Controlword	UNIT_U16
Objekt 2	0x6042:00	2	2	vl Target Velocity	UNIT_I16
Objekt 3	0x6072:00	2	4	Max Torque	UNIT_U16
			6		

TPDO 0 Mapping

	Index	Größe [Byte]	Offset [Byte]	Name	Datentyp
Objekt 1	0x6041:00	2	0	Statusword	UNIT_U16
Objekt 2	0x6044:00	2	2	vl Velocity Actual Value	UNIT_I16
Objekt 3	0x6077:00	2	4	Torque Actual Value	UNIT_I16
Objekt 4	0x603f:00	2	6	Error Code	UNIT_U16
			8		

Abb. 3: Beispiel-Mapping Velocity Mode

- Speichern Sie die Parameter im Antrieb über die Schaltfläche  in der Symbolleiste und führen Sie einen Neustart durch.
- ✓ Der SD4x-Antrieb erwartet nun, dass er über PROFINET IO zyklisch mit Sollwerten (PDOs – Prozessdatenobjekte) versorgt wird.

3 Ansteuerung über PROFINET IO

Im Folgenden wird die Antriebsadressierung im PROFINET IO-Netzwerk und die Anbindung an die SPS am Beispiel der Siemens-Software TIA PORTAL beschrieben.

3.1 PROFINET IO-Netzwerk

Beim PROFINET IO-Netzwerk handelt es sich um eine ringförmige Netzstruktur. Der Bus basiert auf dem Ethernet-Protokoll und wird mit handelsüblichen Cat-5-Netzwerk-kabeln verbunden.

Die Adressierung der SD4x-Antriebe innerhalb des PROFINET IO-Netzwerks findet implizit durch die Vergabe von Stationsnamen im PROFINET IO-Controller statt.

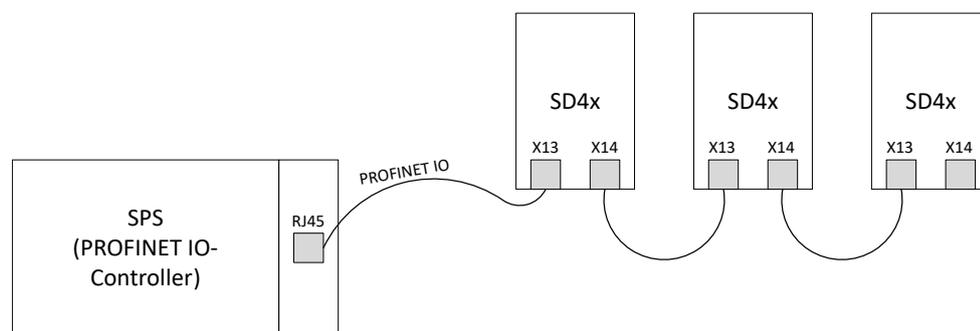


Abb. 4: PROFINET IO-Netzwerk

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

3.2 Software-Anbindung an die SPS

Die Anbindung an die SPS wird nachfolgend am Beispiel der Entwicklungsumgebung TIA PORTAL der Firma SIEMENS beschrieben.

1. Kopieren Sie die von SIEB & MEYER bereitgestellte GSDML-Datei sowie die BMP-Datei in das Projektverzeichnis
`{LW}\TIA\workspace\{projectname}\AdditionalFiles\GSD\` .
 - Beispiel GSDML: `gsdml-v2.35-sieb_meyer-0400-sd4x-20230510.xml`
 - Beispiel BMP: `gsdml-0400-0101-sm0sd4x.bmp`
2. Starten Sie die Entwicklungsumgebung TIA PORTAL und laden Sie das Projekt.

3. Zum Installieren der GSDML-Datei wählen Sie im Menü „Extras“ den Punkt „Gerätebeschreibungsdateien (GSD) verwalten“.

Das folgende Auswahlfenster wird geöffnet.

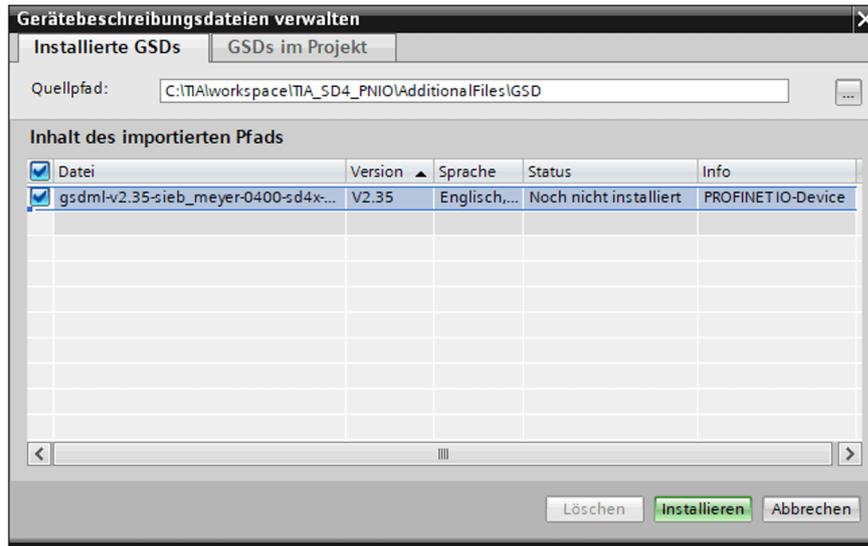


Abb. 5: GSDs verwalten

- a. Wählen Sie oben den Quellpfad, in dem sich Ihre GSDML-Datei befindet.
 - b. Wählen Sie die gewünschte Datei über die Checkbox links aus und klicken Sie auf „Installieren“.
 - Die GSDML-Datei wird nun in das Projekt eingebunden.
 - c. Schließen Sie das Fenster nach erfolgreicher Installation.
 - Der Hardware-Katalog wird nun aktualisiert.
4. Wechseln Sie in die Netzansicht.

- Wählen Sie den SD4x-Antrieb im Hardware-Katalog aus (Weitere Feldgeräte → PROFINET IO → I/O → SIEB & MEYER AG → pns → sd4x) und ziehen Sie ihn in das Netzwerk-Fenster.

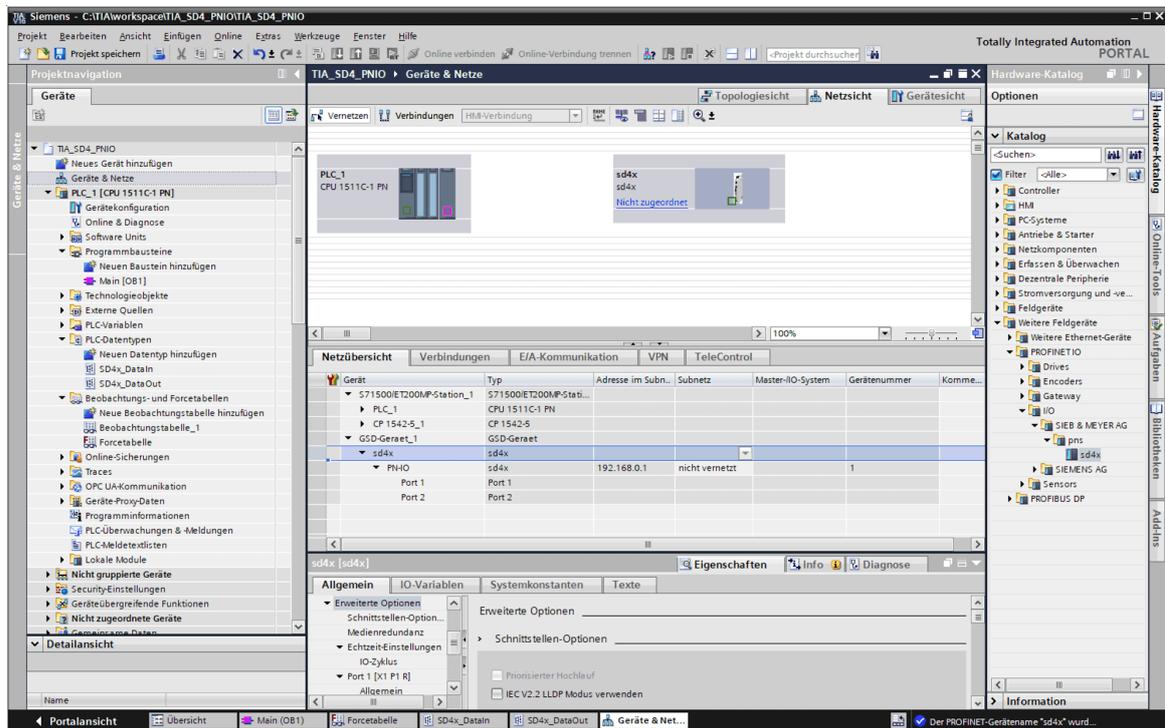


Abb. 6: Netzansicht

- Klicken Sie mit der rechten Maustaste auf den Text „Nicht zugeordnet“ und wählen Sie den Punkt „Neuem IO-Controller zuweisen“ im Kontextmenü aus.

Das folgende Fenster wird geöffnet:

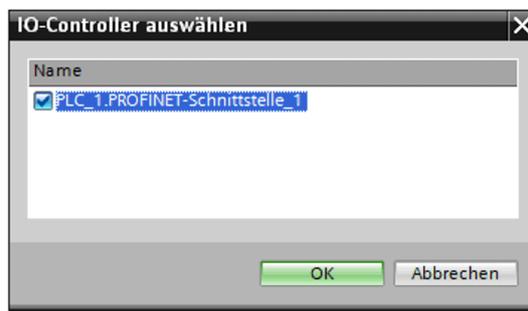


Abb. 7: IO-Controller auswählen

- Wählen Sie in dem Fenster die PROFINET-Schnittstelle aus und klicken Sie auf „OK“.
→ Der Antrieb wird nun mit dem IO-Controller verbunden.

7. Wechseln Sie nun in die Geräteansicht.

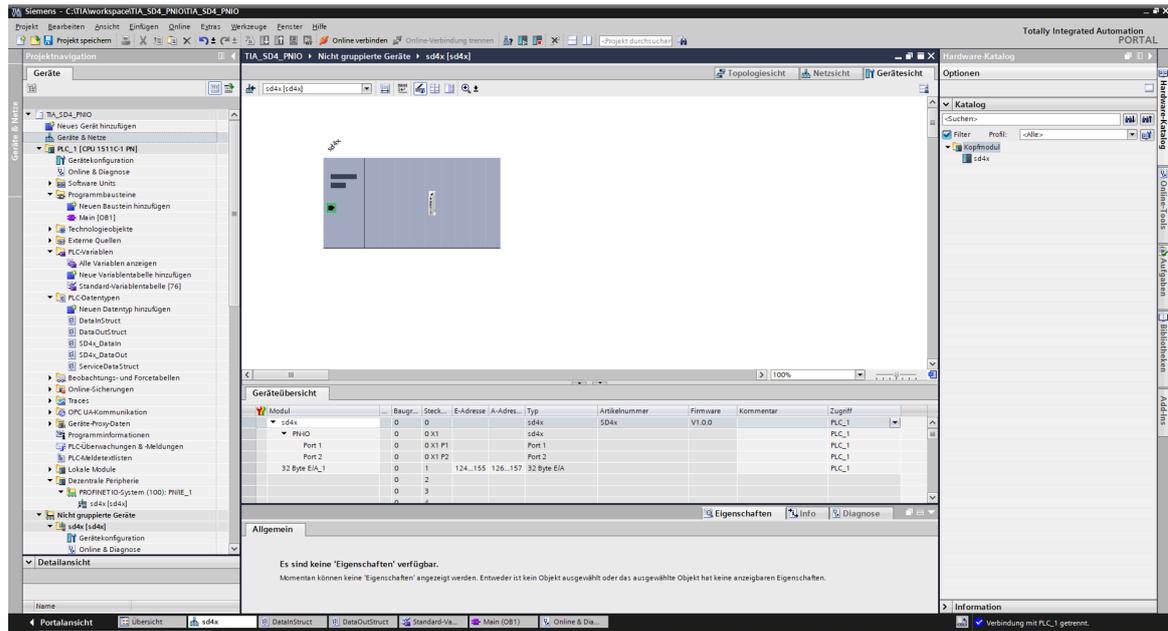


Abb. 8: Geräteansicht

8. Um dem Antrieb einen neuen Namen zu geben, gehen Sie wie folgt vor:
 - a. Klicken Sie den Modulnamen „sd4x“ an und geben Sie den gewünschten Namen ein.
 - b. Weisen Sie diesen Gerätenamen nun über das Icon „Name“ dem Antrieb zu. Das Fenster „PROFINET-Gerätenamen vergeben“ öffnet sich.

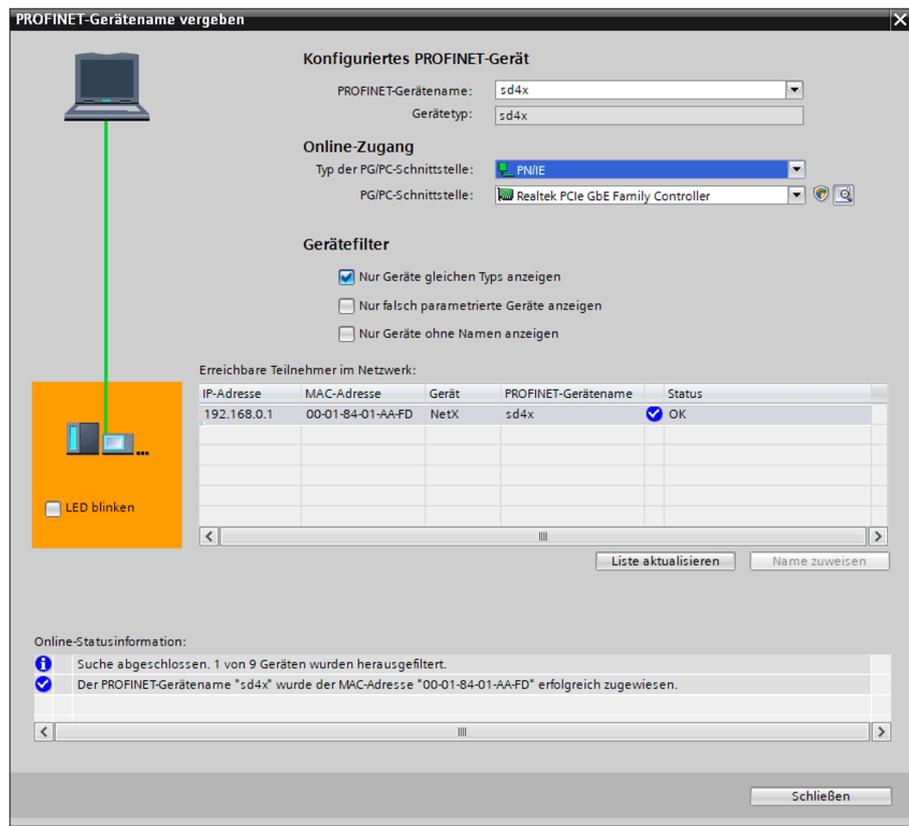


Abb. 9: Gerätenamen vergeben

- Erstellen Sie neue PLC-Datentypen für die Eingangs- und Ausgangsdaten mit einem „Array[0..31] of Byte“, z. B. DataInStruct und DataOutStruct.

DataInStruct:

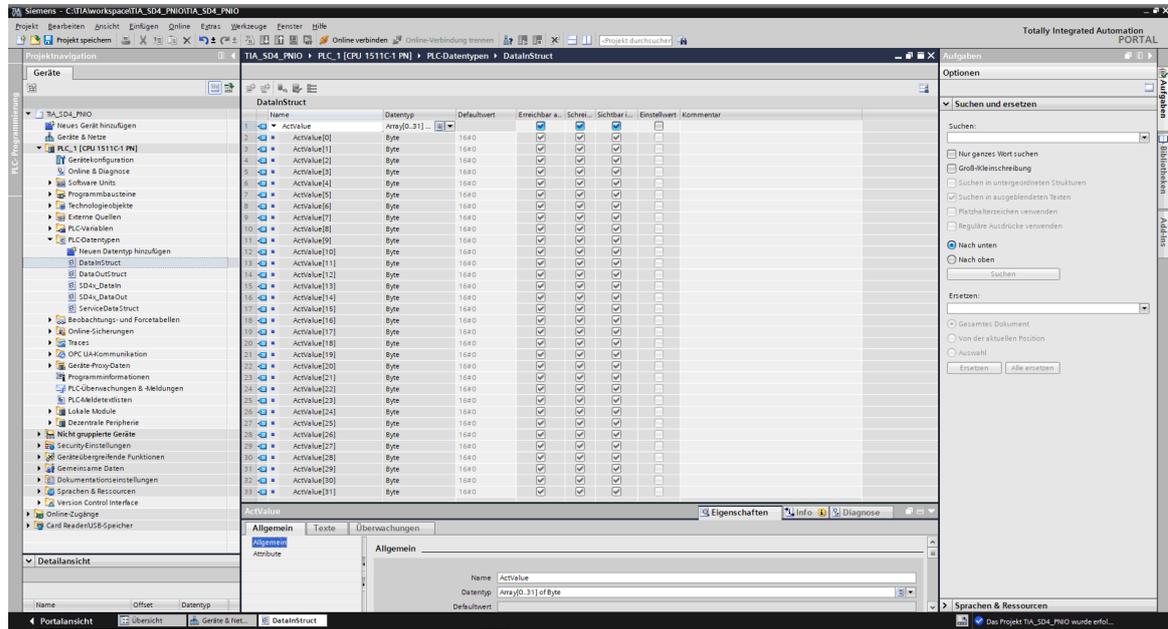


Abb. 11: PLC-Datentypen für die Eingangsdaten

DataOutStruct:

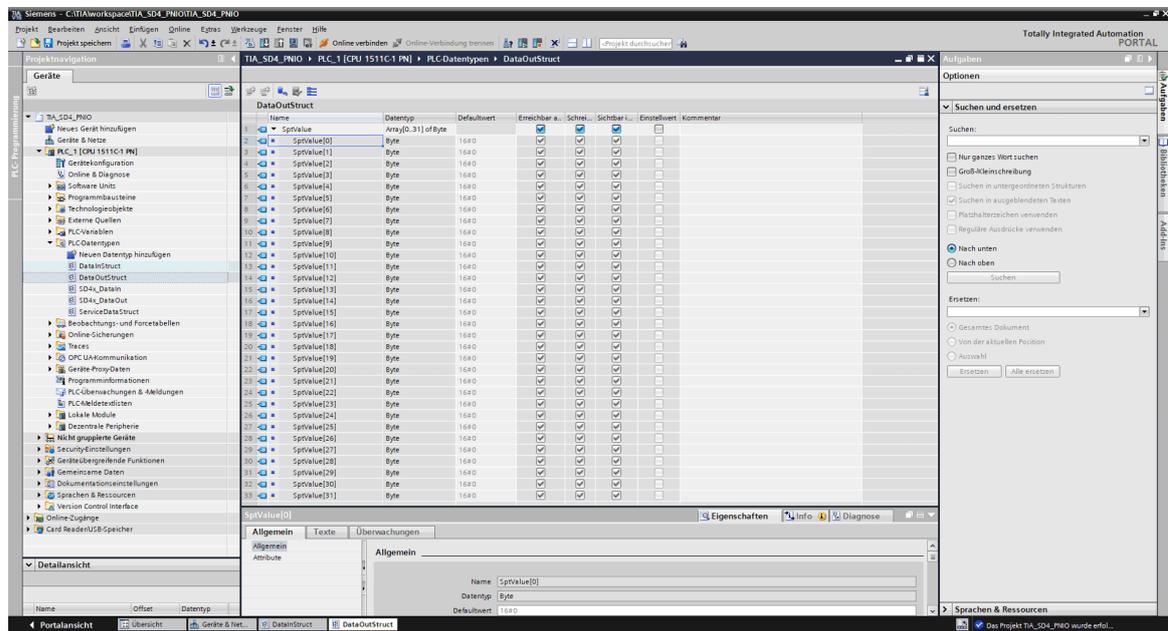


Abb. 12: PLC-Datentypen für die Ausgangsdaten

- Legen Sie neue PLC-Variablen in der Variablen-tabelle an und verknüpfen diese mit den Eingangs- und Ausgangsdatenstrukturen.

Beispiel:

InData_0 DataInStruct %I124.0

OutData_0 DataOutStruct %Q126.0

13. Erstellen Sie nun ihr **Main-Programm**.

Wie Sie die Programmvariablen den Prozessdaten des Antriebs zuweisen, ist im [Kapitel 3.4 „Zyklische Prozessdaten \(PDO\)“](#), Seite 15 beschrieben.

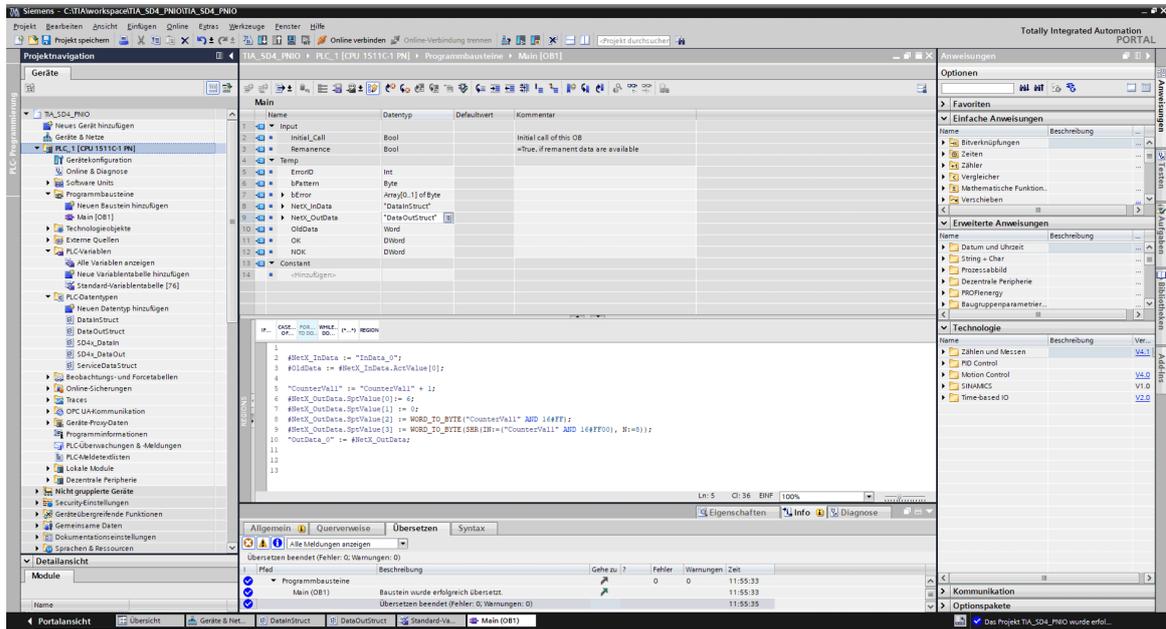


Abb. 13: Main-Programm erstellen

14. Übersetzen Sie Hardware und Software.

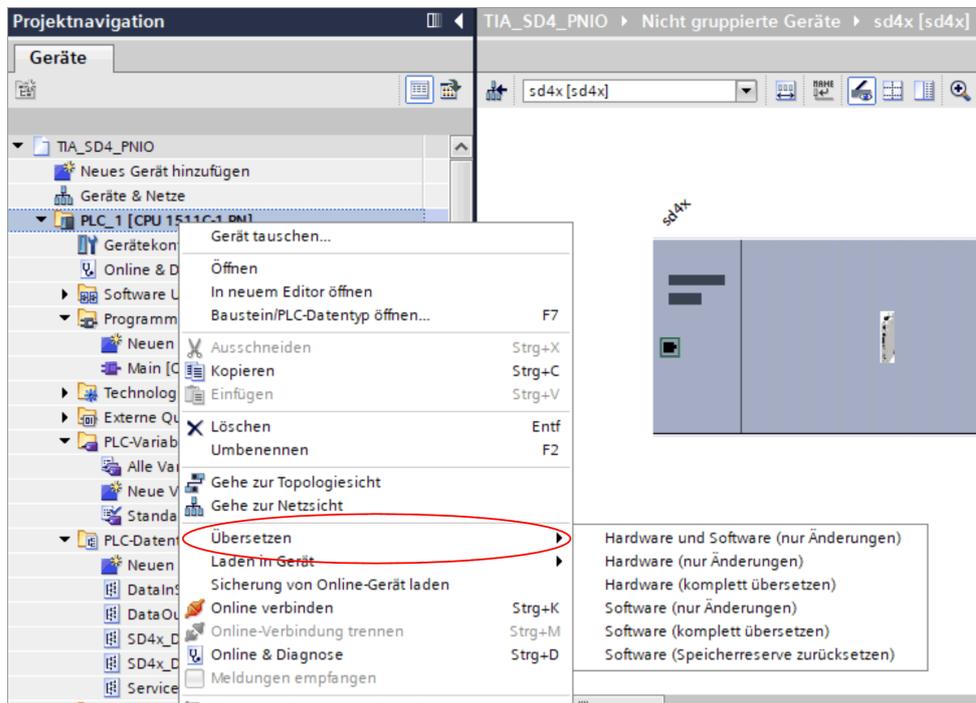


Abb. 14: Hardware und Software übersetzen

15. Laden Sie Hardware und Software in Ihre SPS.

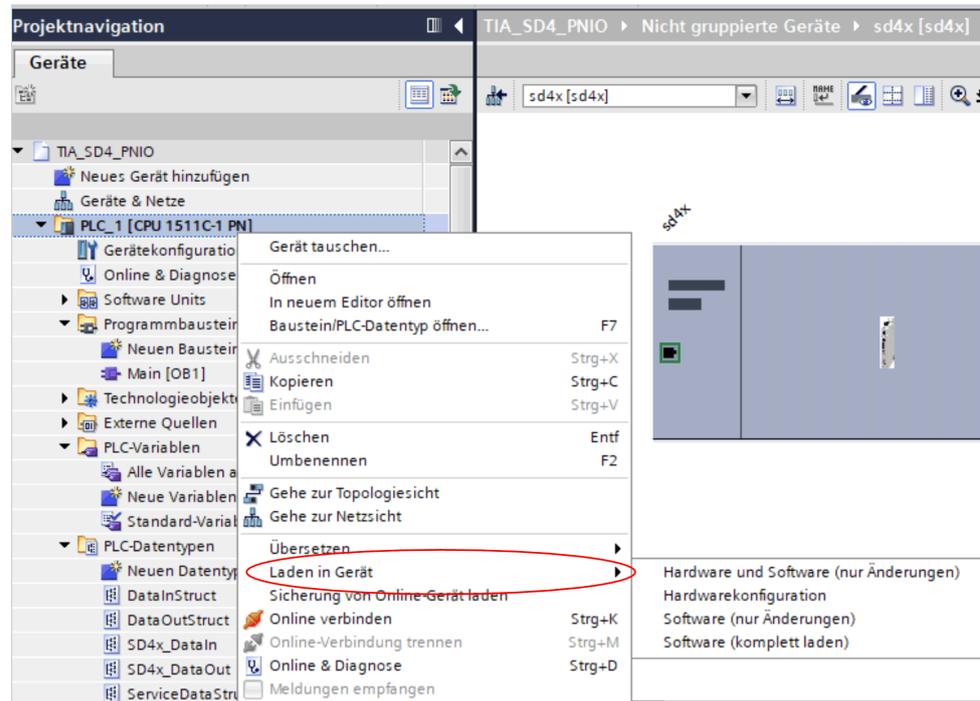


Abb. 15: Hardware und Software laden

16. Starten Sie ihre SPS im RUN-Mode.
17. Prüfen Sie mit der *drivemaster4*-Software, ob Ihre Daten korrekt zum Antrieb übertragen werden.
18. Prüfen Sie mit der SPS im Online-Mode, ob die Daten vom Antrieb korrekt empfangen werden.

3.3 PROFINET IO-Netzwerkmanagement (NMT)

Die Kommunikation des Antriebs wird über das PROFINET IO-Netzwerkmanagement von der SPS gesteuert. Das PROFINET IO-Netzwerkmanagement koordiniert die Controller- und Device-Applikationen beim Hochfahren und während des Betriebs.

Das PROFINET IO-Netzwerkmanagement kennt folgende Zustände:

Zustand	Wert	Beschreibung
UNKNOWN	0	Unbekannter Zustand
OFFLINE	1	Der Antrieb hat keine gültige Konfiguration.
STOP	2	Der Antrieb hat keine Kommunikation zum Controller. Es findet kein Verbindungsaufbau statt. Der Buszustand des Antriebs kann auf Ein oder Aus gesetzt werden.
IDLE	3	Die Kommunikation wird aufgebaut.
OPERATE	4	Die E/A-Verbindung ist hergestellt. Controller und Antrieb tauschen gültige E/A-Daten aus.

3.4 Zyklische Prozessdaten (PDO)

Der Prozessdatenkanal ermöglicht einen zyklischen Datenaustausch zwischen der SPS als *Controller* und dem Antrieb als *Device*.



Die Bedeutung der Daten im Antrieb wird mittels der Parametrierung „Feldbus → PDO Mapping“ in der *drivemaster4*-Software festgelegt.

3.4.1 Sollwerttelegramm

Das Sollwerttelegramm zur Ansteuerung des Antriebs besteht aus 32 Byte Nutzdaten. Die Sollwerte des SPS-Programms müssen diesen Bytes zugewiesen werden.

Beispiel für die Sollwerte im Profile Velocity Mode:

Byte	Name	Bit	Beschreibung
0	Controlword (Low Byte) Bits codiert nach DS402- Standard (Objekt 6040h)	0	Einschalten
		1	Spannung freigeben
		2	Schnellhalt
		3	Betrieb freigeben
		4	Betriebsartabhängig
		5	
		6	
7	Fehler zurücksetzen		
1	Controlword (High Byte) Bits codiert nach DS402- Standard (Objekt 6040h)	8	Halt
		9	Betriebsartabhängig
		10	Reserviert
		11	Herstellerspezifisch
		12	
		13	
14			
15			
2	Target Velocity (Objekt 60FFh)	Target Velocity Byte 0	
3		Target Velocity Byte 1	
4		Target Velocity Byte 2	
5		Target Velocity Byte 3	
6	Max Torque (Objekt 6072h)	Max Torque Byte 0	
7		Max Torque Byte 1	
8 ... 31	ungenutzt	= 0	

Wenn ein Objekt mit einer Größe von 2 Byte gesendet werden soll, muss das Objekt auf 2 Bytes des Sollwerttelegramms aufgeteilt werden.

Beispiel: DS402 Max Torque (Objekt 6072h)

```
MaxTorqueByte0 = WORD_TO_BYTE (MaxTorque AND 16#00FF)
MaxTorqueByte1 = WORD_TO_BYTE (SHR (IN := (MaxTorque AND 16#FF00), N := 8))
```

Wenn ein Objekt mit einer Größe von 4 Byte gesendet werden soll, muss das Objekt auf 4 Bytes des Sollwerttelegramms aufgeteilt werden.

Beispiel: DS402 Target Velocity (Objekt 60FFh)

```
TargetVelocityByte0 = DWORD_TO_BYTE (TargetVelocity AND 16#000000FF);
TargetVelocityByte1 = DWORD_TO_BYTE (SHR (IN := (TargetVelocity AND 16#0000FF00), N := 8));
TargetVelocityByte2 = DWORD_TO_BYTE (SHR (IN := (TargetVelocity AND 16#00FF0000), N := 16));
TargetVelocityByte3 = DWORD_TO_BYTE (SHR (IN := (TargetVelocity AND 16#FF000000), N := 24));
```

3.4.2 Istwerttelegramm

Das Istwerttelegramm zur Ansteuerung des Antriebs besteht aus 32 Byte Nutzdaten. Die Istwerte des SPS-Programms müssen diesen Bytes zugewiesen werden.

Beispiel für die Istwerte im Profile Velocity Mode:

Byte	Name	Bit	Beschreibung
0	Statusword (Low Byte) Bits codiert nach DS402- Standard (Objekt 6041h)	0	Einschaltbereit
		1	Eingeschaltet
		2	Betrieb freigegeben
		3	Fehler
		4	Spannung freigegeben
		5	Schnellhalt
		6	Einschaltsperr
		7	Warnung
1	Statusword (High Byte) Bits codiert nach DS402- Standard (Objekt 6041h)	8	Herstellerspezifisch: Inbetriebnahme-Modus aktiv
		9	Fernbetrieb
		10	Sollwert erreicht
		11	Interner Grenzwert erreicht
		12	Betriebsartabhängig: reserviert
		13	
		14	Herstellerspezifisch: Status Safe Torque Off (STO)
		15	Herstellerspezifisch: Initialisierung abgeschlossen
2 3 4 5	Velocity Actual Value (Objekt 606Ch)		Velocity Actual Value Byte 0 Velocity Actual Value Byte 1 Velocity Actual Value Byte 2 Velocity Actual Value Byte 3
6 7 8 9	Position Actual Value (Objekt 6064h)		Position Actual Value Byte 0 Position Actual Value Byte 1 Position Actual Value Byte 2 Position Actual Value Byte 3
10 11	Error Code (Objekt 603Fh)		Error Code Byte 0 Error Code Byte 1

Wenn ein Objekt mit einer Größe von 2 Byte gelesen werden soll, muss das Objekt auf 2 Bytes des Istwerttelegramms aufgeteilt werden.

Beispiel: DS402 Error Code (Objekt 603Fh)

```
ErrorCode = BYTE_TO_DWORD (ErrorCodeByte0) OR
SHL (IN := BYTE_TO_DWORD (ErrorCodeByte1), N := 8);
```

Wenn ein Objekt mit einer Größe von 4 Byte gelesen werden soll, muss das Objekt auf 4 Bytes des Istwerttelegramms aufgeteilt werden.

Beispiel: DS402 Position Actual Value (Objekt 6064h)

```
PositionActualValue = BYTE_TO_DWORD (PositionActualValueByte0) OR
SHL (IN := BYTE_TO_DWORD (PositionActualValueByte1), N := 8) OR
SHL (IN := BYTE_TO_DWORD (PositionActualValueByte2), N := 16) OR
SHL (IN := BYTE_TO_DWORD (PositionActualValueByte3), N := 24);
```

3.5 Azyklische Servicedaten (SDO)

Der Servicedatenkanal ermöglicht einen azyklischen Datenaustausch zwischen der SPS als *Controller* und dem Antrieb als *Device*. Über diesen Kanal kann die SPS einzelne Objekte azyklisch vom Antrieb lesen oder in den Antrieb schreiben.

Für die Übertragung werden die Kommunikationsfunktionsblöcke RDREC und WRREC in der SPS genutzt. Informationen zur Funktionsweise finden Sie im Dokument „PROFIBUS and PROFINET Guideline, Communication Function Blocks on PROFIBUS DP and PROFINET IO“.



Gemäß der Norm ist der Index auf 16 Bit begrenzt. Wenn der SPS-Hersteller einen 32-Bit-Wert für den Index verwendet, können davon nur die unteren 16 Bit genutzt werden.

Es können maximal 1024 Byte mit einem Aufruf gelesen oder geschrieben werden.

Mit dem 16-Bit-Index kann die SPS jedes Objekt im Objektverzeichnis des Antriebs ansprechen. Einfache Objekte bis zu einer Größe von 4 Byte können direkt gelesen oder beschrieben werden. Soll ein Objekt mit Subindex oder ein Array-Objekt angesprochen werden, muss der Subindex oder Array-Offset vorab übertragen werden. Hierzu steht das Objekt 158 (0x009E) „PROFINET IO – Subindex und Offset“ zur Verfügung. In diesem Objekt wird das unterste Byte für den Subindex und die oberen 3 Bytes für den Offset genutzt:

Byte 3	Byte 2	Byte 1	Byte 0
Offset			Subindex

Folgenden Daten gelten für Offset und Subindex im Objekt 158 „PROFINET IO – Subindex und Offset“:

Objekt 158	Offset	Subindex
Wertebereich	0 bis 16777215 (0x00 bis 0xFFFFF)	0 bis 255 (0x00 bis 0xFF)
Defaultwert	0	0

Nach jedem Schreib- oder Lesevorgang, außer bei Objekt 158, werden Subindex und Offset intern im Antrieb auf Null zurückgesetzt.

3.5.1 Schreib- und Lesevorgänge

Die folgenden Abbildungen zeigen die prinzipiellen Abläufe beim Schreiben und Lesen der azyklischen Servicedaten.

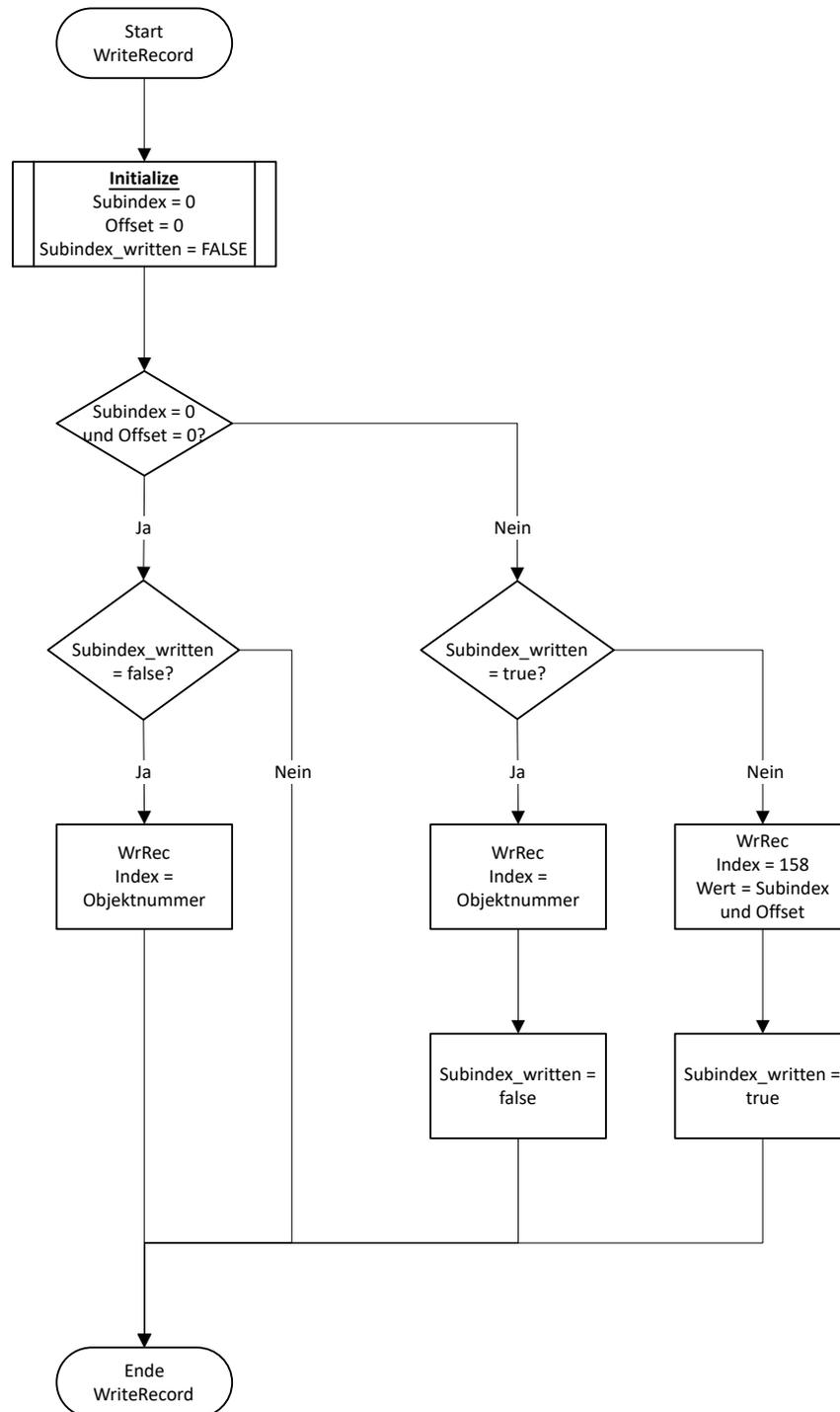


Abb. 16: Prinzipieller Schreibvorgang

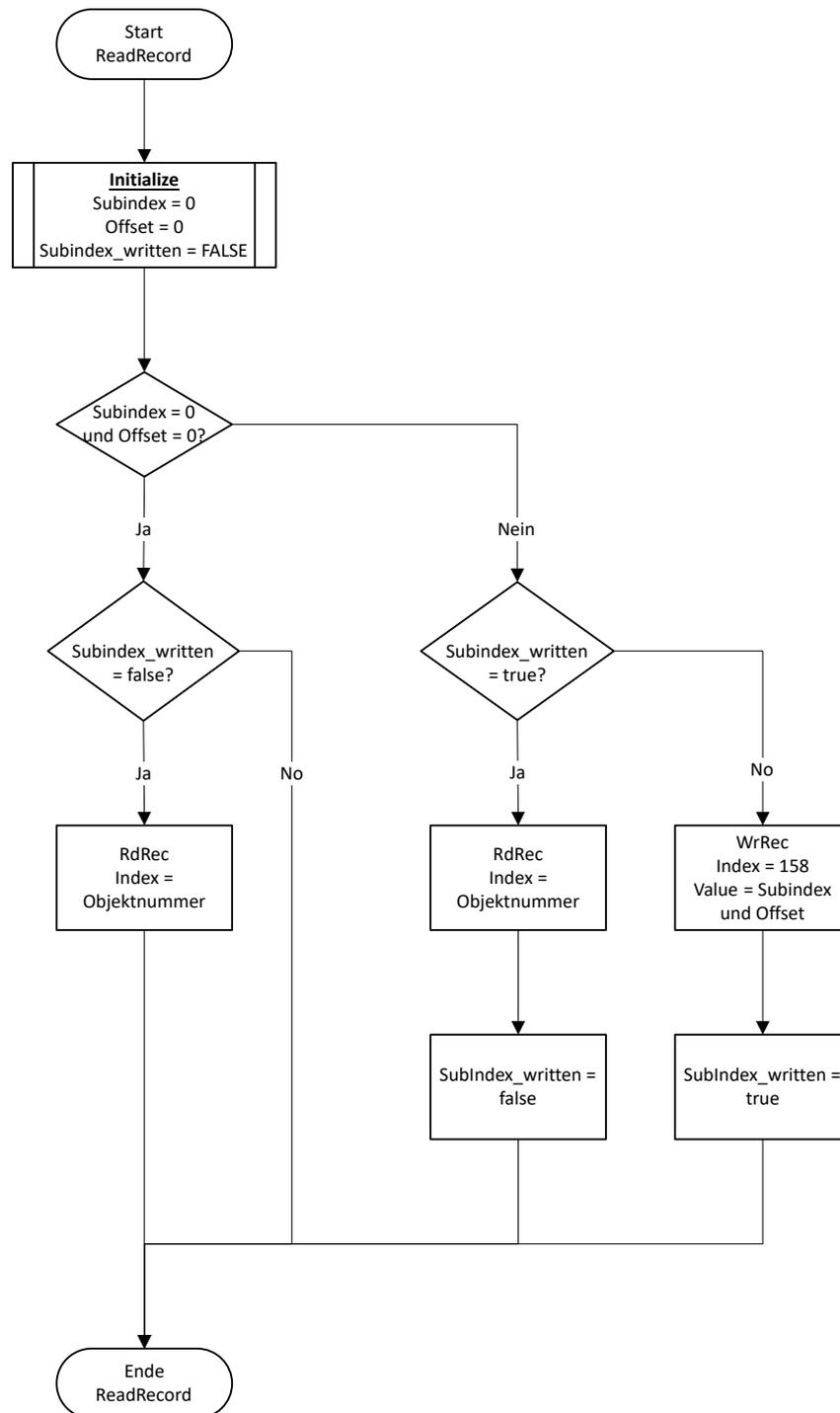


Abb. 17: Prinzipieller Lesevorgang

3.5.2 Programmbeispiel für Siemens-SPS

Das folgenden Programmbeispiel wurde mit dem TIA PORTAL für eine Siemens-SPS S7-1500 erstellt.

1. Fügen Sie unter „Programmbausteine“ einen weiteren Baustein hinzu.
 - a. Wählen Sie als Typ „Global-DB“.
 - b. Fügen Sie bei Bedarf Titel und Kommentar hinzu.

- c. Bestätigen Sie Ihre Eingaben mit „OK“.
 - Der neue Baustein wird unter „Programmbausteine“ angelegt.
2. Wählen Sie den Baustein aus und fügen Sie folgende Variablen hinzu:

Name	Typ
ReadRequest	BOOL
WriteRequest	BOOL
ReadRecord	ARRAY[0..1200] OF BYTE
WriteRecord	ARRAY[0..1200] OF BYTE

Nun müssen Sie das Programmbeispiel „Main (OB1)“ um die Schreib- und Lesevorgänge mittels der Instanzen der Funktionen für WRREC und RDREC ergänzen.

3. Ergänzen Sie **WriteRecord** im Programmbeispiel „Main (OB1)“ wie folgt:
 - a. Tippen Sie „WRR“ in Main ein.
 - Eine Auswahlliste wird geöffnet.
 - b. Wählen Sie den Eintrag „WRREC“ in der Auswahlliste aus.
 - Das Dialogfenster „Aufrufoptionen“ wird geöffnet.
 - c. Geben Sie im Fenster „Aufrufoptionen“ den gewünschten Namen für den Datenbaustein ein, z. B. „WRREC_DB_9“. Bestätigen Sie die Eingabe mit „OK“.

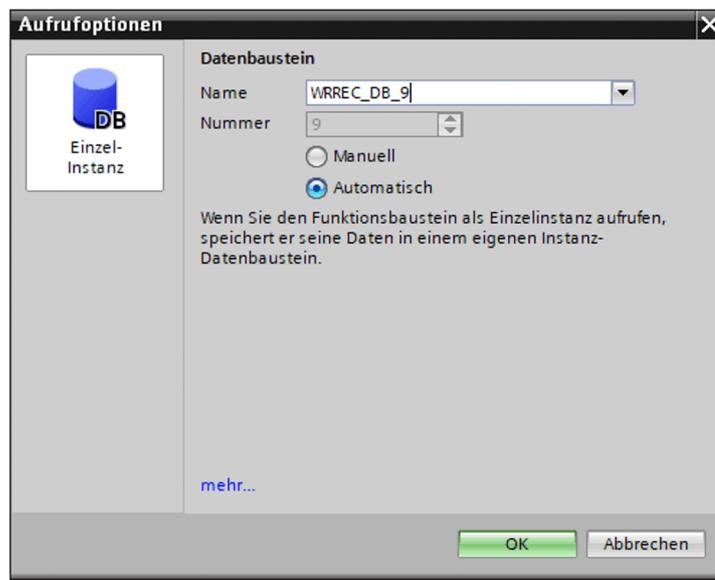


Abb. 18: Aufrufoptionen – WRREC

- Der Baustein wird nun unter „Programmbausteine → Systembausteine → Systemressourcen“ angelegt.

Gleichzeitig wird der Aufruf im Main-Programm hinzugefügt:

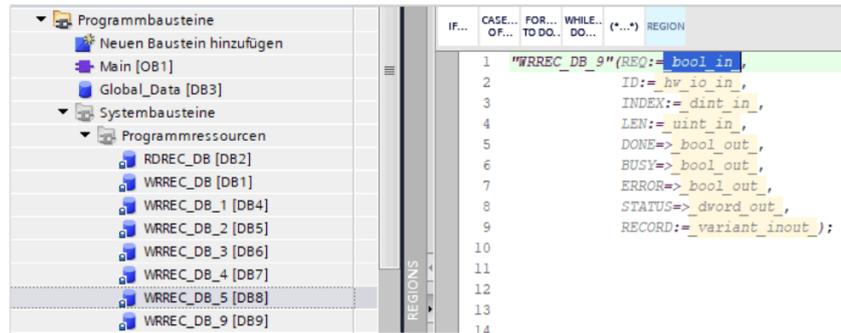


Abb. 19: Main – WRREC

- d. Sie können nun die Aufrufparameter für Ihre Applikation anpassen.

Beispiel:

```
"WRREC_DB_9"(REQ := "Global_Data".WriteRequest,
             ID := "sd4x~Head",
             INDEX := 16#60C5,
             LEN:=4,
             RECORD:="Global_Data".WriteRecord[0]);
```

Prinzipieller Programmcode:

```
// set acyclic data
"Global_Data".WriteSubIndexOffset := 16#02000000;

IF "Global_Data".WriteSubIndexOffset = 0 THEN
  IF NOT "Global_Data".ReadSIOWritten THEN
    "WRREC_DB_1"(REQ := "Global_Data".WriteRequest,
               ID := "sd4x~Head",
               INDEX := 16#6073,
               LEN := 2,
               RECORD := "Global_Data".WriteCurrent);
  END_IF;
ELSE
  IF "Global_Data".WriteSIOWritten THEN
    "WRREC_DB_2"(REQ := "Global_Data".WriteRequest,
               ID := "sd4x~Head",
               INDEX := 16#604A,
               LEN := 2,
               RECORD := "Global_Data".WriteRecord);
    IF "WRREC_DB_2".DONE THEN
      "Global_Data".WriteSIOWritten := FALSE;
    END_IF;
  ELSE
    "WRREC_DB_3"(REQ := "Global_Data".WriteRequest,
               ID := "sd4x~Head",
               INDEX := 158,
               LEN := 4,
               RECORD := "Global_Data".WriteSubIndexOffset);
    IF "WRREC_DB_3".DONE THEN
      "Global_Data".WriteSIOWritten := TRUE;
    END_IF;
  END_IF;
END_IF;
```

- 4. Ergänzen Sie **ReadRecord** im Programmbeispiel „Main (OB1)“ wie folgt:
 - a. Tippen Sie „RDR“ in Main ein.
 - Eine Auswahlliste wird geöffnet.
 - b. Wählen Sie den Eintrag „RDREC“ in der Auswahlliste aus.
 - Das Dialogfenster „Aufrufoptionen“ wird geöffnet.

- c. Geben Sie im Fenster „Aufrufoptionen“ den gewünschten Namen für den Datenbaustein ein, z. B. „RDREC_DB_10“. Bestätigen Sie die Eingabe mit „OK“.

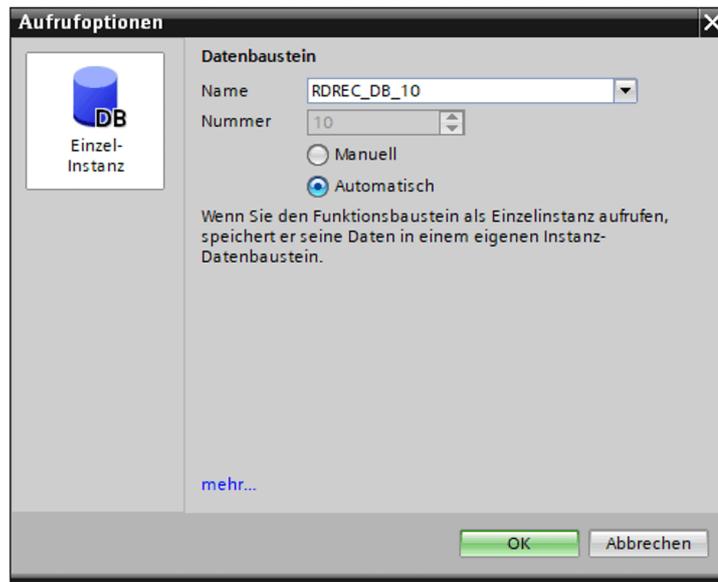


Abb. 20: Aufrufoptionen – RDREC

- Der Baustein wird nun unter „Programmbausteine → Systembausteine → Systemressourcen“ angelegt.

Gleichzeitig wird der Aufruf im Main-Programm hinzugefügt:

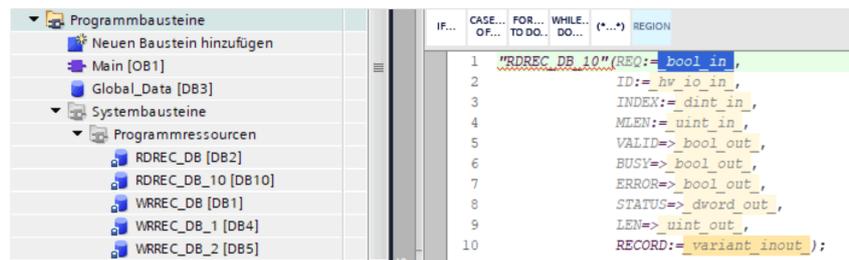


Abb. 21: Main – RDREC

- d. Sie können nun die Aufrufparameter für Ihre Applikation anpassen.

Beispiel:

```
"RDREC_DB_10" (REQ:="Global_Data".ReadRequest,
              ID:="sd4x~Head",
              INDEX:=16#604A,
              MLEN:=4,
              RECORD:="Global_Data".ReadRecord[0]);
```

Prinzipieller Programmcode:

```
// get acyclic data
"Global_Data".ReadSubIndexOffset := 16#01000000;

IF "Global_Data".ReadSubIndexOffset = 0 THEN
  IF NOT "Global_Data".ReadSIOWritten THEN
    "RDREC_DB"(REQ := "Global_Data".ReadRequest,
              ID := "sd4x~Head",
              INDEX := 16#6073,
              MLEN := 2,
              RECORD := "Global_Data".ReadRecord);
  END_IF;
ELSE
  IF "Global_Data".ReadSIOWritten THEN
    "RDREC_DB"(REQ := "Global_Data".ReadRequest,
              ID := "sd4x~Head",
              INDEX := 16#604A,
```

```
                MLEN := 4,
                RECORD := "Global_Data".ReadRecord);
IF "RDREC_DB".VALID THEN
    "Global_Data".ReadSIOWritten := FALSE;
END_IF;
ELSE
    "WRREC_DB" (REQ := "Global_Data".ReadRequest,
                ID := "sd4x~Head",
                INDEX := 158,
                LEN := 4,
                RECORD := "Global_Data".ReadSubIndexOffset);
IF "WRREC_DB".DONE THEN
    "Global_Data".ReadSIOWritten := TRUE;
END_IF;
END_IF;
END_IF;
```

5. Ergänzen Sie Ihr Programm um die oben beschriebenen Abläufe für die Schreib- und Lesevorgänge. Füllen Sie dazu `Global_Data.WriteRecord` mit den zu schreibenden Daten. Steuern Sie über `Global_Data.WrRequest` und `Global_Data.RdRequest` das Schreiben und Lesen der Objekte.

4 Antriebssteuerung

Die Antriebssteuerung in den Drive Controllern der Gerätefamilie SD4x ist nach dem DS402-Standard realisiert.

Dieser Standard definiert Objekte, welche in Prozessdatenobjekte (PDO) und Service-datenobjekte (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 ([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⁾ Möglicher Werte:

RW = Read/Write (Lesen/Schreiben)

RO = Read only (nur Lesen)

⁽²⁾ Möglicher Werte:

Volatile = flüchtiger Datenspeicher (geht nach Stromabschaltung verloren)

Non-volatile = nichtflüchtiger Datenspeicher (bleibt nach Stromabschaltung erhalten)

4.1 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.

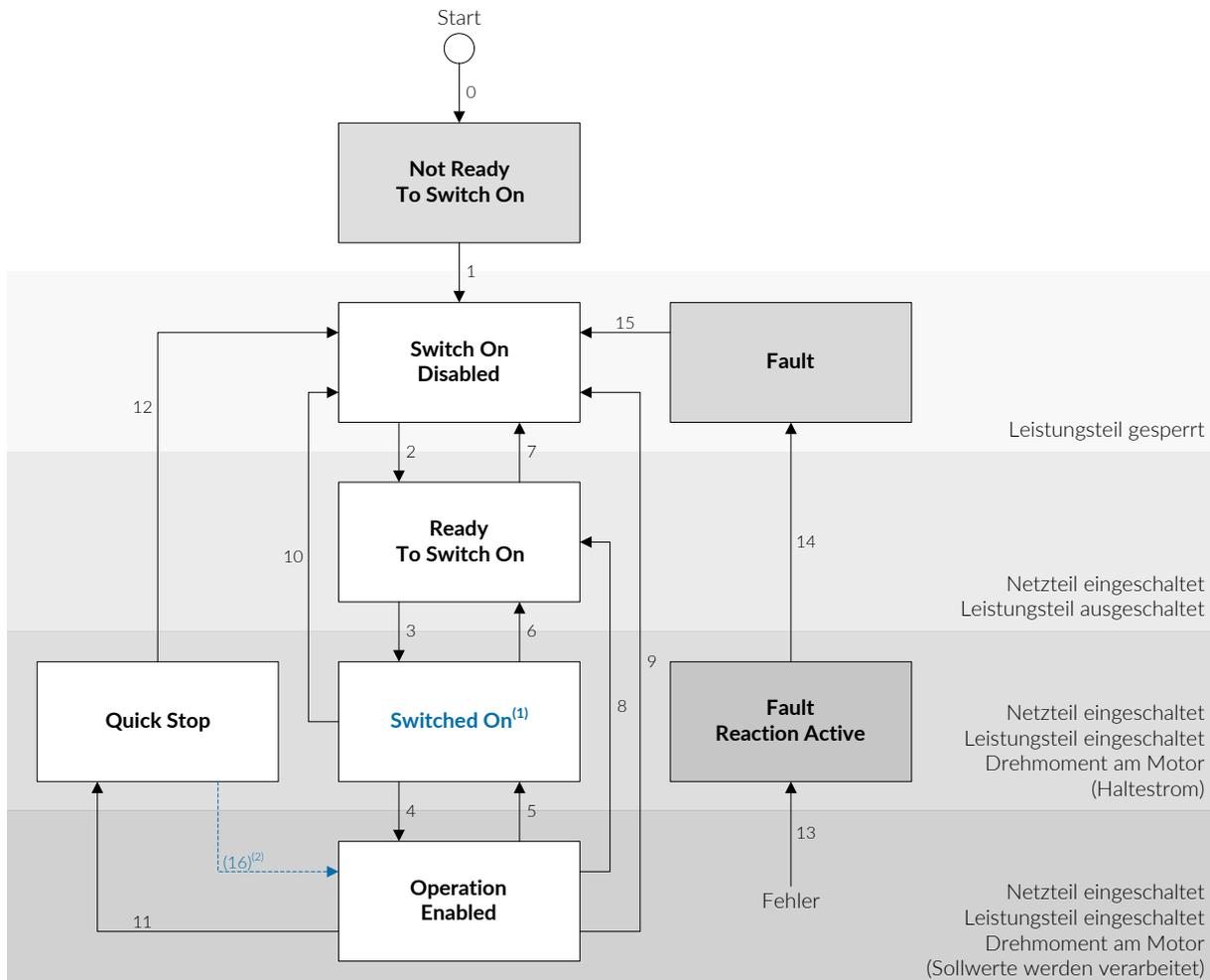


Abb. 22: Antriebszustände und Zustandsübergänge

- (1) Abweichend vom DS402-Standard wird bei SD4x-Antrieben im Zustand „Switched On“ die Endstufe bereits eingeschaltet, so dass Drehmoment am Motor anliegt. Die Sollwerte sind in diesem Zustand noch nicht freigegeben 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 Nummern an den Übergängen sind Transitionen, die mit Hilfe der Befehle ausgelöst werden können. Welcher Befehl welchen Übergang auslöst, finden Sie im [Kapitel 4.1.1 „Befehle“, Seite 27](#). 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 automatisch und nicht über Befehle erreicht werden.

4.1.1 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.

4.1.2 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 Einschaltsperrre	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 (Einschaltsperrre)	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: Einschaltsperrre 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> *Kursiv dargestellte Bitbelegungen sind SIEB & MEYER-spezifisch und können bei anderen Antriebsherstellern abweichen.	
DS402_0x6041_STATUSWORD		ID: 0x6041
RO	Volatile	Einheit: – U16

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 und dem aktuellen Zustand abhängig:

- ▶ Geschwindigkeitsbetriebsarten VL/PV: Bit 10 wird gesetzt, wenn die eingestellte Zielgeschwindigkeit erreicht wurde.
- ▶ Zustand „Quick Stop Active“: Wenn der Antrieb den Zustand „Quick Stop Active“ nicht selbständig verlässt, gibt Bit 10 an, dass die Schnellhaltfunktion abgeschlossen wurde und der Motor gehalten wird.
- ▶ Umschalten der Betriebsart: Das Bit 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.

4.1.3 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: –	116

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 'Einschaltsperr' 02: Abbremsen mit Schnellhalterampe und Wechsel in den Status 'Einschaltsperr' 03: Abbremsen mit Strombegrenzung und Wechsel in den Status 'Einschaltsperr'		
DS402_0X605A_QUICK_STOP_OPTION_CODE		ID: 0x605A	
RW	Volatile	Einheit: –	116

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: –	116

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: –	116

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 Schnellhaltrampe 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.

4.2 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.

4.2.1 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:

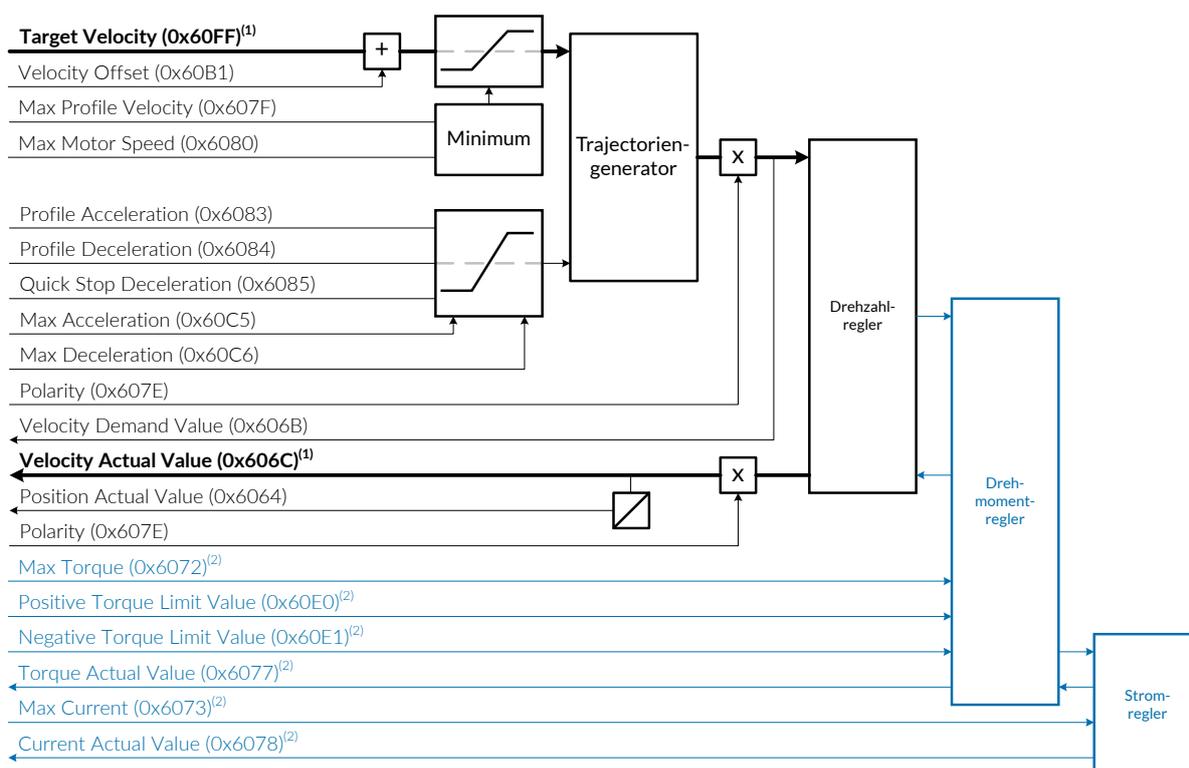


Abb. 23: Blockdiagramm für Profile Velocity Mode (Geschwindigkeitsmodus PV)

- (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 10 (Zielwert erreicht) wird aktiv, 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.
- ▶ 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.

4.2.2 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 32](#) 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:

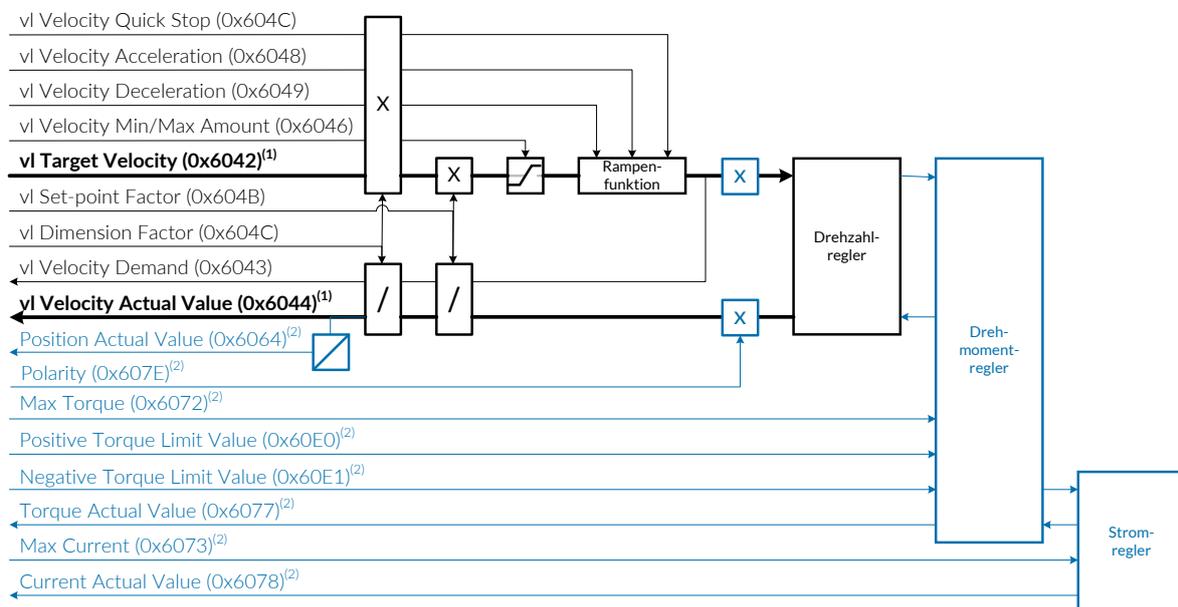


Abb. 24: Blockdiagramm für Velocity Mode (Geschwindigkeitsmodus VL)

- (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 die Bits 10 und 12 wie folgt:

- ▶ Bit 10 (Zielwert erreicht) wird aktiv, 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.
- ▶ 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.

4.2.2.1 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)	I16

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)	I16

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)	I16

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: <ul style="list-style-type: none"> ▶ 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.

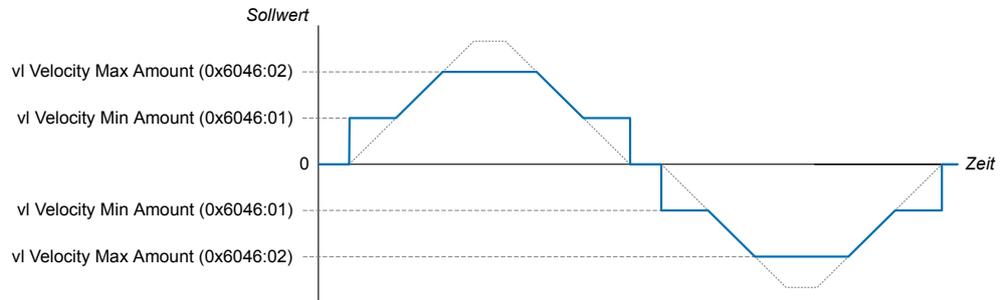


Abb. 25: Drehzahlbegrenzung im Velocity Mode

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: <ul style="list-style-type: none"> ▶ 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_ACCELERATION_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.

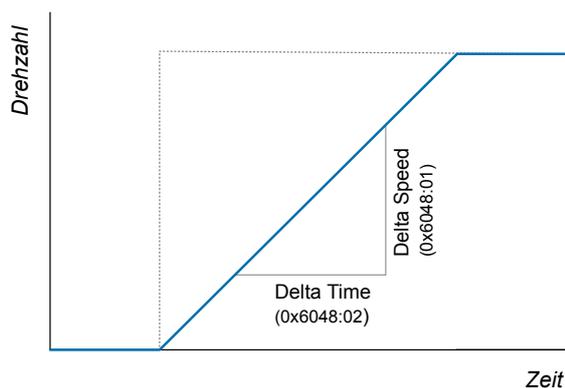


Abb. 26: Beschleunigungsbegrenzung im Velocity Mode

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: <ul style="list-style-type: none"> ▶ 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.

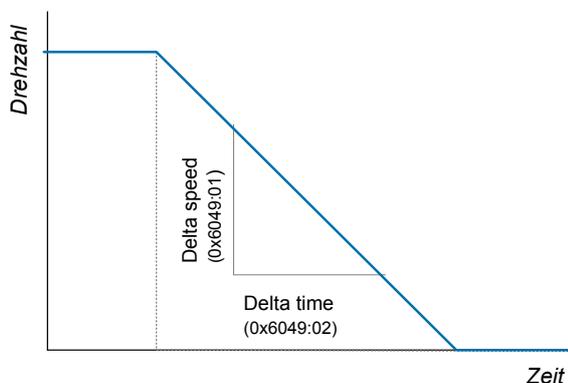


Abb. 27: Bremsbeschleunigungsbegrenzung im Velocity Mode

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		Schnellhaltrampe im Velocity Mode Sub-IDs: <ul style="list-style-type: none"> ▶ 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.

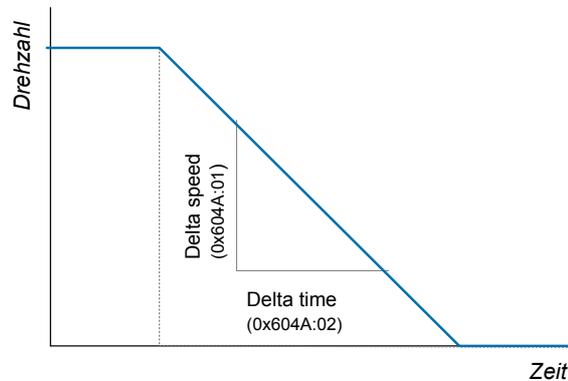


Abb. 28: Bremsbeschleunigungsbegrenzung bei Schnellhalt im Velocity Mode

Somit wird die resultierende Schnellhaltrampe 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: <ul style="list-style-type: none"> ▶ 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

1. 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: <ul style="list-style-type: none"> ▶ 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_ACCERLERATION_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 **vl Target Velocity**, **vl Velocity Demand** und **vl 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.

4.2.3 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:

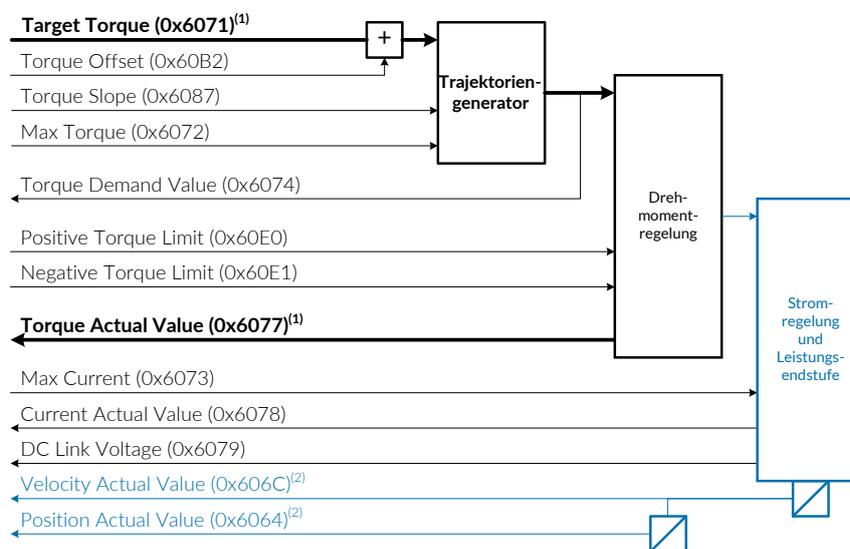


Abb. 29: Blockdiagramm für Torque Mode (Drehmomentbetrieb)

- (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.

4.2.4 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 4.4 „Parametersatzumschaltung“](#), Seite 51). 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

4.3 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 34](#).

4.3.1 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 Anwindereinheiten	
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.

4.3.2 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 Anwendereinheiten	
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 Anwendereinheiten 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		Drehmomentsollwert nach Rampen- und Begrenzungsfunktion Einheit und Richtung entsprechen Objekt Target Torque (0x6071).	
DS402_0x6074_TORQUE_DEMAND			ID: 0x6074
RO	Volatile	Einheit: ‰	I16

Torque Actual Value (0x6077)

Dieses Objekt gibt das aktuelle Drehmoment zurück.

Torque Actual Value		Drehmomentistwert 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		Stromistwert 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

4.3.3 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		Nenn Drehmoment 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

4.3.4 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 [Kapitel „Soll- und Istwerte im Velocity Mode“](#), Seite 34).

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	...	0
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.

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	Inkmente	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 $10^{-3} \times 1/\text{s}^2$ (*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 $1/\text{s}^2$.

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 $10^{-3} \times 1/\text{s}^3$ (*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 1/s^3 .

4.3.5 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.

4.3.6 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 [Kapitel „Velocity Mode \(VL\)“, Seite 33](#)) setzt sich die Zielgeschwindigkeit also aus der **Target Velocity** und dem **Velocity Offset** zusammen.

Torque Offset (0x60B2)

Dieses Objekt gibt den Offset für Soll Drehmomentvorgaben an.

Torque Offset		Offset für Soll Drehmoment 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.

4.3.7 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

4.3.7.1 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 \(S. 29\)](#) konfigurieren.

Quick Stop Deceleration		Schnellhaltrampe 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

4.3.8 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 „Einschaltsperr“ 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 entsprechende Ausgänge.

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 Ausgangs gelesen wird.

Beispiel:

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

Der Zustand des physikalischen Ausgangs 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 Ausgangs 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.

4.4 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.

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 *drive-master4*-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 *drivemaster4*-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.

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“.

4.4.1 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 Gerätereustart und dadurch zu einem kurzzeitigen Verlust der Feldbusverbindung – also der Kontrolle über das Gerät.

5 Diagnose

5.1 Diagnose in *drivemaster4*

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

Zusätzlich finden Sie die IP-Konfiguration sowie Diagnosedaten rechts auf der Seite.

5.1.1 Telegramme

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

Name	Zähler	Zeitstempel	Msgs/s	Dir	Cmd	Länge	Status
RPDO 0	23693	43254.13	1100	Rx		8	
Data : 06 00 53 e6 00 00 00 00							
Controlword		6			0006h		
Target Velocity		58963			0000e653h		
Max Torque		0			0000h		
TPDO 0	23693	43254.23	1100	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		
Unsupported	5	10950.49	0	Rx	1f2eh	8	0h
Unsupported	5	10950.50	0	Tx	1f2fh	0	c0000202h
Link_Status_Change	2	4953.46	0	Rx	2f8ah	32	0h
Link_Status_Change	2	4953.54	0	Tx	2f8bh	0	0h
Parameter_End	1	10938.50	0	Rx	1f0eh	12	0h
Parameter_End	1	10938.52	0	Tx	1f0fh	8	0h
Save_IP_Addr	1	7926.53	0	Rx	1fb8h	13	0h
Save_IP_Addr	1	7926.59	0	Tx	1fb9h	0	0h

Communication state: **[4] Operate**

Abb. 30: PROFINET IO-Busaktivitäten

Station Name

Die obere Zeile gibt den PROFINET IO-Stationsnamen an. Der Stationsname ist im folgenden Objekt hinterlegt:

Objekt 155 (0x009B) Station Name

Der Stationsname setzt sich wie folgt zusammen:

- ▶ Längenangabe des aktuellen Stationsnamens: 2 Byte
- ▶ Stationsname im ASCII-Format: max. 240 Byte

Beispiel:

- ▶ Datelänge des Stationsnamens: **4 Byte**
- ▶ Stationsname: **sd4x**

Darstellung im Objektbrowser der *drivemaster4*-Software:

Objekt	PROFINET IO - Stationsname								
	00	01	02	03	04	05	06	07	08
0000	04	00	73	64	34	78	00	00	00

Abb. 31: Stationsname

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: <ul style="list-style-type: none"> ▸ 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.

Communication state

Dieses Feld zeigt den Kommunikationsstatus des Bussystems an. Mögliche Zustände sind: Unknown, Offline, Stop, Idle und Operate.

5.1.2 Diagnosedaten

Die folgende Abbildung zeigt die IP-Konfiguration für die Ethernet-Kommunikation sowie die Diagnosedaten rechts auf der PROFINET IO-Diagnoseseite. Die angegebenen Werte sind typische Werte im Status OPERATE.

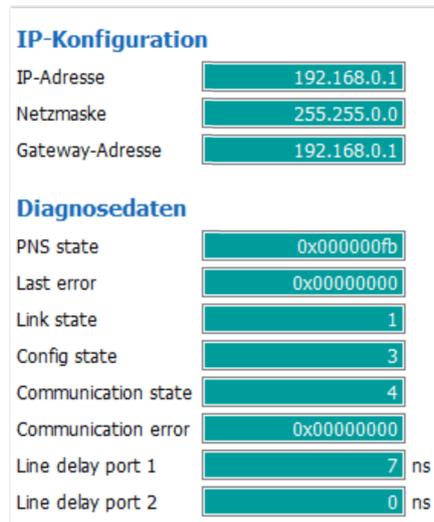


Abb. 32: IP-Konfiguration und Diagnosedaten

Die Daten sind in den folgenden Objekten hinterlegt.

Objekt 154 (0x009A) Diagnosedaten

Die Diagnosedaten setzen sich aus 8 Werten mit je 4 Byte zusammen:

Variable	Wert	Beschreibung	Beispielwert
PNS state	Bit	Status des Protokollstacks im NetX90	0x000000FB
	0	Geräteinformationen sind eingestellt	1
	1	PROFINET-Stack ist gestartet	1
	2	Reserviert	0
	3	Mindestens eine API ist präsent	1
	4	Modul 0 ist vorhanden	1
	5	Modul 0 und Submodul 1 sind vorhanden	1
	6	Netzwerkcommunication ist erlaubt	1
	7	Netzwerkcommunication ist aktiviert	1
	8	Konfiguration ist gelockt	0
	9	Fataler Fehler ist aufgetreten	0
	10	Diagnosedaten mit schwerwiegendem Fehler vorhanden	0
	11	Wartungserforderlicher Datensatz existiert	0
12	Wartungsanforderungsdatsatz existiert	0	
Last error		Anzeige des letzten Fehlers	0x00000000
Link state	0	Keine Information vorhanden	0x00000001
	1	Physikalischer Link arbeitet korrekt	
	2	Physikalischer Link arbeitet mit niedriger Geschwindigkeit	
	3	Kein Physikalischer Link vorhanden	

Variable	Wert	Beschreibung	Beispielwert
Config state	0	Nicht konfiguriert	0x00000003
	1	Konfiguriert über DBM-Dateien	
	2	Fehler während Konfiguration über DBM-Dateien	
	3	Konfiguriert über Applikation	
	4	Konfiguration über Applikation läuft	
	5	Fehler während Konfiguration über Applikation	
	6	Konfiguriert über Warmstart-Parameter	
	7	Konfiguration über Warmstart-Parameter läuft	
Communication state	0	UNKNOWN (schwarz)	0x00000004
	1	OFFLINE (grau)	
	2	STOP (rot)	
	3	IDLE (gelb)	
	4	OPERATE (grün)	
Communication error		Anzeige des Fehlerstatus im Kommunikationskanal	0x00000000
Line delay port 1		Anzeige der Signallaufzeit [ns] für Port 1	0x00000007
Line delay port 2		Anzeige der Signallaufzeit [ns] für Port 2	0x00000000

Darstellung im Objektbrowser der *drivemaster4*-Software:

Objekt PROFINET IO - Diagnosedaten																
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
0000	fb	00	00	00	00	00	00	00	01	00	00	00	03	00	00	00
0010	04	00	00	00	00	00	00	00	07	00	00	00	00	00	00	00

Abb. 33: Diagnosedaten

Objekt 156 (0x009C) IP-Adresse

Die IP-Adresse setzt sich aus 3 Werten mit je 4 Byte zusammen:

Beschreibung	Beispielwert
IP-Adresse	C0:A8:00:01
Netzmaske	FF:FF:00:00
Gateway	C0:A8:00:01

Darstellung im Objektbrowser der *drivemaster4*-Software:

Objekt PROFINET IO - IP-Adresse												
	00	01	02	03	04	05	06	07	08	09	0a	0b
0000	01	00	a8	c0	00	00	ff	ff	01	00	a8	c0

Abb. 34: IP-Adresse

5.2 Diagnose in der SPS

Keine Verbindung zum Antrieb

Sollte zu einem Gerät keine Verbindung aufgebaut werden, prüfen Sie, ob der Stationsname, die IP-Adresse und die MAC-Adresse mit den Daten im Antrieb übereinstimmen. Setzen Sie gegebenenfalls die gewünschten Daten in der *drivemaster4*-Software und übertragen Sie diese in den Antrieb.

6 Weiterführende Informationen

Weitere Dokumente

Die folgenden Dokumente enthalten weitere Informationen zu diesem Thema:

Anbieter	Dokument
SIEB & MEYER AG	<ul style="list-style-type: none"> ▶ drivemaster4 – Bedienen ▶ Antriebssystem SD4 – Ethernet-Konfiguration
PROFIBUS Nutzerorganisation e.V. (P-NO)	Dokumente zum PROFINET können als registrierter Benutzer von der Website der Organisation heruntergeladen werden.
CiA e.V.	<ul style="list-style-type: none"> ▶ 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
PROFIBUS Nutzerorganisation e.V. (P-NO)	www.profibus.com
CiA e.V.	www.can-cia.org
Siemens AG	www.siemens.com

7 Index

A

- Abkürzungen [5](#)
- Azyklische Servicedaten [17](#)
 - Programmbeispiel [20](#)

B

- Betriebsarten [31](#)
- Betriebsartwechsel [39](#)

C

- Controlword [27](#)

D

- Diagnose
 - drivemaster4 [53](#)
 - SPS [56](#)
- Drehmomentbetrieb [39](#)

G

- Geschwindigkeitsmodus PV [32](#)
- Geschwindigkeitsmodus VL [33](#)

K

- Kommunikationsstatus [15](#)

N

- Netzwerk [8](#)
- NMT-Status [15](#)

O

- Objekte
 - Abort Connection Option Code [29](#)
 - Acceleration [45](#)
 - Controlword [27](#)
 - Current Actual Value [43](#)
 - DC Link Circuit Voltage [43](#)
 - Digital Inputs [49](#)
 - Digital Outputs [50](#)

- Disable Operation Option Code [30](#)
- Error Code [51](#)
- Fault Reaction Option Code [31](#)
- Halt Option Code [30](#)
- Jerk [45](#)
- Max Acceleration [48](#)
- Max Current [47](#)
- Max Deceleration [48](#)
- Max Motor Speed [47](#)
- Max Profile Velocity [47](#)
- Max Torque [47](#)
- Mode of Operation [40](#)
- Mode of Operation Display [41](#)
- Motor Rated Current [43](#)
- Motor Rated Torque [43](#)
- Motor Type [44](#)
- Negative Torque Limit [48](#)
- Polarity [46](#)
- Position [44](#)
- Position Actual Value [42](#)
- Positive Torque Limit [48](#)
- Profile Acceleration [49](#)
- Profile Deceleration [49](#)
- Quick Stop Deceleration [49](#)
- Quick Stop Option Code [30](#)
- Shutdown Option Code [30](#)
- Statusword [28](#)
- Supported Drive Modes [40](#)
- Target Torque [42](#)
- Target Velocity [41](#)
- Torque Actual Value [43](#)
- Torque Demand [43](#)
- Torque Offset [47](#)
- Torque Slope [49](#)
- Velocity [45](#)
- Velocity Actual Value [42](#)
- Velocity Demand Value [42](#)
- Velocity Offset [46](#)
- vl Dimension Factor [38](#)

- vl Set-point Factor [37](#)
- vl Target Velocity [34](#)
- vl Velocity Acceleration [35](#)
- vl Velocity Actual Value [34](#)
- vl Velocity Deceleration [36](#)
- vl Velocity Demand [34](#)
- vl Velocity Min Max Amount [34](#)
- vl Velocity Quick Stop [36](#)

Option Codes [29](#)

P

Parametersatzumschaltung [51](#)

Profile Velocity Mode [32](#)

S

SPS-Anbindung [8](#)

Statusword [28](#)

Steuerbefehle Antrieb [27](#)

T

TIA PORTAL [8](#)

Torque Mode [39](#)

V

Velocity Mode [33](#)

Z

Zustände [28](#)

Zustandsmaschine [26](#)

Zustandsübergänge des Antriebs [26](#)