



Antriebssystem SD2

SERVOLINK 4-Bussystembeschreibung



1	SERVOLINK 4-Bussystem.....	3
1.1	Struktur.....	4
1.2	Übertragungsformat.....	5
1.3	Interrupt.....	5
1.4	FPGA-Schnittstelle.....	5
1.5	RAM-Schnittstelle.....	6
2	Telegramme.....	8
2.1	Geschwindigkeitsmodus 1 (Mode 2).....	8
2.1.1	Sollwert-Telegramm.....	10
2.1.2	Istwert-Telegramm.....	11
2.2	Profile Velocity Mode (PV, Mode 3).....	11
2.2.1	Sollwert-Telegramm.....	13
2.2.2	Istwert-Telegramm.....	14
2.3	Interpolierende Lageregelung (Mode 7).....	14
2.3.1	Sollwert-Telegramm.....	16
2.3.2	Istwert-Telegramm.....	17
3	Servicedatenkanal.....	18
3.1	Abbildung im Telegramm.....	18
3.1.1	Sollwert-Telegramm.....	18
3.2	Istwert-Telegramm.....	19
3.3	Zugriff auf die Antriebsobjekte.....	20
3.3.1	Azyklische Servicedaten.....	20
3.3.2	Übersicht Zugriffsfunktionen.....	21
3.3.3	Lesen und Schreiben von einfachen Objekten.....	22
3.3.4	Lesen und Schreiben von Array-Objekten.....	23
3.3.5	Zugriff auf Array-/Stringobjekte.....	23
3.3.6	Fehlercodes des Servicedatenkanals.....	24

1 **SERVOLINK 4-Bussystem**

Der SERVOLINK 4 ist ein Bussystem für die Vernetzung von Antriebsverstärkern mit einem Steuerungsrechner. Die Struktur ist als Ring ausgelegt. Die Daten werden seriell und zyklisch übertragen. Als Medium stehen Lichtwellenleiter (TOSLINK) und RS422-Schnittstellen zur Verfügung. Es können auch beide Medien gleichzeitig eingesetzt werden. Zur Ankopplung stehen für die Master- und die Slave-Seite FPGA-Lösungen zur Verfügung, für die Masterseite gibt es zusätzlich das Gateway 036215Xy. Die Schnittstelle zum Controller verhält sich wie ein Dual-Port-Memory (DPRAM). Für jeden Antrieb stehen 16 Bytes Solldaten und 16 Bytes Istdaten zur Verfügung. Die Inhalte sind betriebsartenabhängig. Die Datenblöcke sind mit einem CRC-Check (zyklische Redundanzüberprüfung) versehen. Jeder Antrieb ersetzt die empfangenen Solldaten durch seine Istdaten, wodurch gewährleistet ist, dass es zu keiner Übertragungszeitverlängerung durch die Übertragung der Istwerte kommt.

1.1 Struktur

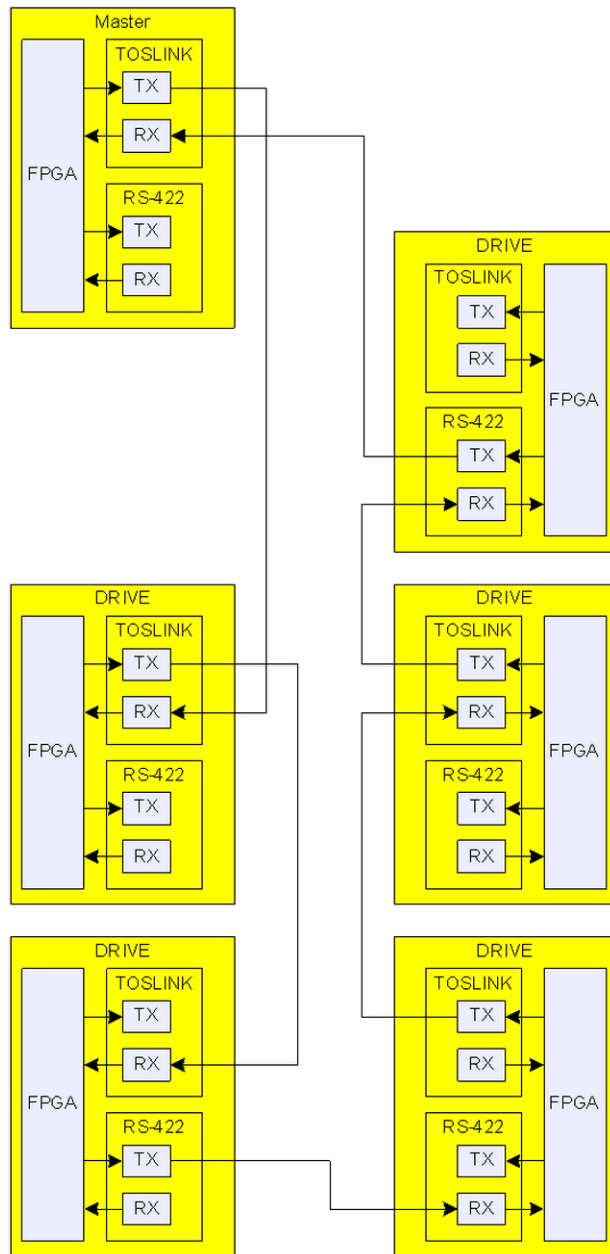


Abb. 1: Toslink und RS422 Mischbetrieb

1.2 Übertragungsformat

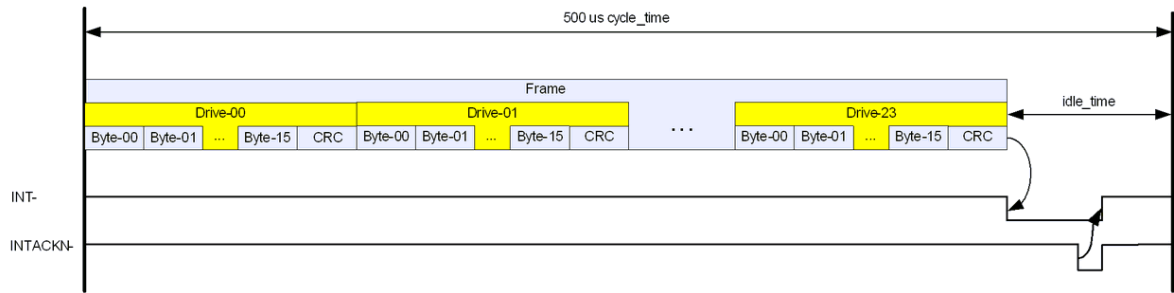


Abb. 2: SERVOLINK 4-Übertragungsformat

Übertragungszeit:

$$t_{\text{bit}} = 84 \text{ ns}$$

$$t_{\text{frame}} = t_{\text{bit}} \times n_{\text{drives}} ((n_{\text{byte}} + n_{\text{delay}} + 1_{\text{CRC}}) \times 10_{\text{bits}}) = 84 \text{ ns} \times 24 \times ((16 + 2 + 1) \times 10) = 380 \mu\text{s}$$

$$t_{\text{idle}} = t_{\text{cycle}} - t_{\text{frame}} = 500 \mu\text{s} - 380 \mu\text{s} = 120 \mu\text{s}$$

CRC:

$$1 + x^1 + x^2 + x^8$$

1.3 Interrupt

Das FPGA löst standardmäßig am Ende des Datenframes einen Interrupt aus. In der Interruptroutine müssen die Istwerte ausgelesen und die neuen Sollwerte in das FPGA geschrieben werden. Erfolgt dieses bis zum nächsten Zyklusstart nicht, werden automatisch Null-Daten an die Antriebe übertragen. Der Interrupt wird durch das Lesen einer Adresse im FPGA zurückgenommen.

1.4 FPGA-Schnittstelle

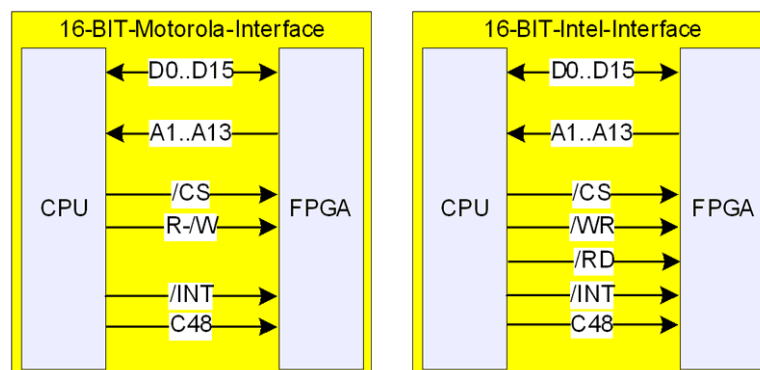


Abb. 3: FPGA-Schnittstelle

1.5 RAM-Schnittstelle

Adresse	Name	Bedeutung
0x0000 ... 0x017f	ACTUALVALUES	Antriebsistdaten
0x0200 ... 0x037f	SETPOINTVALUES	Antriebssolldaten
0x0400 ... 0x0402	CRCSTATE	CRC-Status, 0xfffff → OK
0x0404	DATAEXCHANGE	Bit = 0 → Leerlauf Bit = 1 → Datenaustausch
0x0406	INTACKN	Interrupt-Quittierung
0x0408	CONFIG	Konfiguration Bit-0 = 1 → Interrupt ON
0x040a	SELECT	Auswahl Medium Bit-0 = 0 → RS422 Bit-0 = 1 → TOSLINK
0x040c	DRIVEIN	Empfangsposition
0x040e	DRIVEOUT	Sendeposition
0x0410	INTPOSITION	Interrupt-Position
0x0412	ZEROCNT	16 Bit Nulldaten-Zähler
0x1000	VERSION	Versionscode
0x1002	TESTWORD	Testwort (0xaaaa)

ACTUAL VALUES

Antriebsistdaten: 24 Antriebe × 16 Byte pro Antrieb

SETPOINT VALUES

Antriebssolldaten: 24 Antriebe × 16 Byte pro Antrieb

CRC-STATE

Je Antrieb ein CRC-Bit. Bit-0 für Antrieb-0, Bit-1 für Antrieb-1, u.s.w.

Bit = 0 → Daten ungültig, Bit = 1 → Daten gültig

Beim Start eines neuen Zyklus werden alle Bits auf 0 gesetzt. Das CRC-Bit wird nach einem fehlerfreien Empfang des Datenpakets gesetzt.

DATA EXCHANGE

Bit-0 = 0 → SERVOLINK 4 ist in der Leerlauf-Phase. Bit-0 = 1 → SERVOLINK 4 ist in der Übertragungsphase.

INTACKN

Ein Zugriff auf dieses Register quittiert den Interrupt. Wird der Interrupt bis zum nächsten Zyklus nicht quittiert, werden automatisch Null-Daten gesendet.

CONFIG

Bit-0 = 0 → Interrupt gesperrt, Bit-0 = 1 → Interrupt freigegeben

SELECT

Bit-0 = 0 → RS422-Schnittstelle, Bit-0 = 1 → TOSLINK-Schnittstelle.

DRIVE-IN POSITION

Das Register gibt die Antriebsnummer des Datenpakets an, das gerade empfangen wird.

DRIVE-OUT POSITION

Das Register gibt die Antriebsnummer des Datenpakets an, das gerade gesendet wird.

INTPOSITION (Programmierbare Interrupt-Position)

Über das Register lässt sich die Interrupt-Position einstellen. Die Position wird in Bit-Zeiten angegeben. Die Position kann zwischen 160 und 5999 liegen. Beachten Sie hierbei, dass es pro Modul zu einer Verzögerung von 20 µs kommt.

ZEROCNT

Der Zähler wird bei jedem Sendevorgang eines Null-Datenzyklus erhöht.

VERSION

Versionskode des VHDL-Programms, z. B. 0x10001

TESTWORD

Testregister. Wert ist 0xaaaa.

2 Telegramme

Die SERVOLINK 4-Telegramme lassen sich in einen Prozessdaten- und einen Service-datenkanal aufteilen. Die Daten im Prozessdatenkanal werden in jedem Zyklus aktua-lisiert und übertragen. Entsprechende Daten sind das Steuer- und Statuswort. Der zu-sätzliche Bereich wird in den verschiedenen Betriebsarten unterschiedlich genutzt und in den entsprechenden Kapiteln separat beschrieben.

Der Servicedatenkanal dient zur Übertragung von nicht zeitkritischen oder größeren Da-tenbereichen. Die Übertragung erfolgt seriell.

2.1 Geschwindigkeitsmodus 1 (Mode 2)

Mit der Betriebsart „Geschwindigkeitsmodus 1“ stehen dem Benutzer die folgenden Möglichkeiten zur Verfügung:

- ▶ Einfache Vorgabe des Geschwindigkeitssollwertes
- ▶ Einfache Vorgabe der Strom-/Maximalmomentbegrenzung
- ▶ Interne Rampengeneratoren für
 - Geschwindigkeitssollwert mit linearer Beschleunigungs- und Bremsrampe
 - Geschwindigkeitssollwert mit verschliffener Beschleunigungs- und Bremsrampe
 - Begrenzung von Strom-/Maximalmoment
- ▶ Statusfunktionen für
 - Drehzahl Null
 - Drehzahlrampe hat den Zielwert erreicht
 - Drehzahlregler ist in der Strom bzw. Momentbegrenzung
 - Strom bzw. Momentbegrenzungsrampe hat den Zielwert erreicht
 - Gerätezustand
- ▶ Parametrierbare Schnellhaltfunktion
 - Bremsen mit maximalem Moment/Strom
 - Bremsen mit parametrierbarer Bremsrampe
 - Bremsen mit definiertem Herunterrampen des Moments bzw. des Stroms
- ▶ Gerätesteuerung nach Antriebsprofil DS402
- ▶ Drehzahlsteuerung und Drehmomentsteuerung möglich

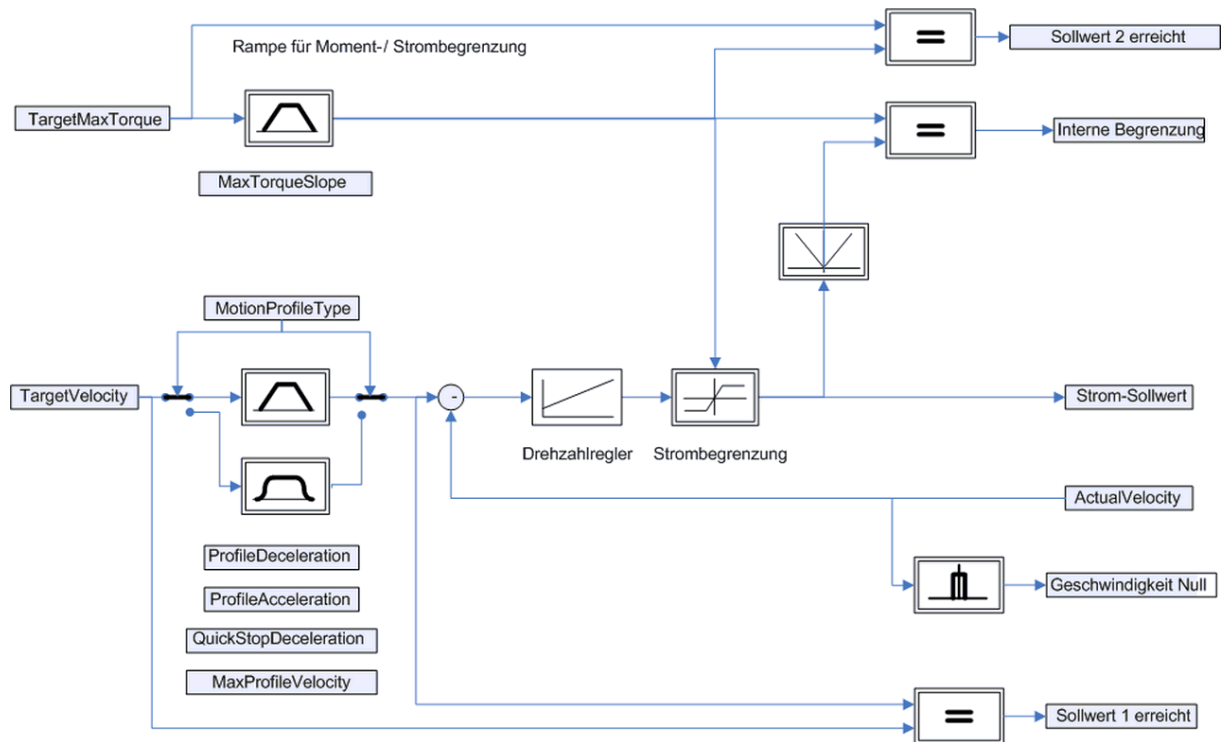


Abb. 4: Betriebsart „Geschwindigkeitsmodus 1“

2.1.1 Sollwert-Telegramm

Das Sollwert-Telegramm zur Ansteuerung des Antriebs besteht aus Nutzdaten von 16 Byte, die wie folgt aufgeteilt sind:

Byte	Größe	Name	Bit	Beschreibung
0	2	ControlWord	0	Einschalten ⁽¹⁾
			1	Spannungsfreigabe ⁽¹⁾
			2	Schnellhalt ⁽¹⁾
			3	Betrieb freigeben ⁽¹⁾
			4	Reserviert
			5	Reserviert
			6	Reserviert
			7	Fehler zurücksetzen ⁽¹⁾
			8	Reserviert
			9	Reserviert
			10	Reserviert
			11	Reserviert
			12	Reserviert
			13	Reserviert
			14	Kennung Sollwert-Telegramm / 0 ⁽²⁾
			15	Kennung Sollwert-Telegramm / 1 ⁽²⁾
2	2	Res.		Reserviert
4	2	SPG_TARGET_VELOCITY_VL		Zielgeschwindigkeit VL 16 Bit 0x3fff = 100 % = Vscaling
6	2	SPG_TARGET_TORQUE_LIMIT		Drehmoment / Strombegrenzung 0x3fff = Spitzenstrom des Antriebsverstärkers NUR in der Antriebsfunktion SERVO, sonst reserviert
8	1	Res.		Reserviert
9	7	ServiceChannel		Siehe Seite 18

⁽¹⁾ Die Funktionsweise der einzelnen Bits ist in der Anleitung „Antriebssystem SD2 – Gerätesteuerung“ beschrieben.

⁽²⁾ Die Telegrammkennung ist für SERVOLINK 4 fest vorgegeben: Bit 14 ist immer 0 und Bit 15 immer 1.

2.1.2 Istwert-Telegramm

Das Istwert-Telegramm des Antriebs besteht aus Nutzdaten von 16 Byte, die wie folgt aufgeteilt sind:

Byte	Größe	Name	Bit	Beschreibung
0	2	StatusWord	0	Einschaltbereit ⁽¹⁾
			1	Eingeschaltet ⁽¹⁾
			2	Betrieb freigegeben ⁽¹⁾
			3	Fehler ⁽¹⁾
			4	Spannungsfreigabe ⁽¹⁾
			5	Schnellhalt ⁽¹⁾
			6	Einschaltsperr ⁽¹⁾
			7	Warnung ⁽¹⁾
			8	Status des digitalen Eingangs 4 / Messsystemkontakt ⁽²⁾
			9	Remote ⁽¹⁾
			10	Sollwert erreicht ⁽¹⁾
			11	Interner Grenzwert aktiv ⁽¹⁾
			12	Drehzahl 0 ⁽¹⁾
			13	SERVOLINK Toggle-Bit ⁽³⁾
			14	Kennung Istwert-Telegramm / 1 ⁽⁴⁾
			15	Kennung Istwert-Telegramm / 0 ⁽⁴⁾
2	4	PCTRL_POSITION_ACTUAL_VALUE	Aktuelle Position Lageregler 32 Bit, Auflösung 0,1 µm	
2 ⁽⁵⁾	2	PCTRL_POSITION_ACTUAL_VALUE	Aktuelle Position Lageregler 16 Bit, Auflösung 0,1 µm	
4 ⁽⁵⁾	2	ICTR_ACTIVE_CURRENT	Wirkstrom (0x3FFF ± Endstufen-Spitzenstrom)	
6	2	VCTRL_VELOCITY_ACTUAL_VALUE	Aktuelle Istgeschwindigkeit 0x3fff = 100% = Vscaling	
8	2	ICTRL_IQ_REFERENCE	Aktueller Stromsollwert 0x3fff = Spitzenstrom des Antriebsverstärkers	
10	6	ServiceData Channel	Siehe Seite 18	

⁽¹⁾ Die Funktionsweise der einzelnen Bits ist in der Anleitung „Antriebssystem SD2 – Gerätesteuerung“ beschrieben.

⁽²⁾ Die Funktionsweise des einzelnen Bits ist in der Anleitung „TID_SD2_Geraetesteuerung_Tiefenmesssystem.pdf“ beschrieben.

⁽³⁾ Das Bit wird bei jedem Aktualisieren der Istwerte getoggelt.

⁽⁴⁾ Die Telegrammkennung ist für SERVOLINK 4 fest vorgegeben: Bit 14 ist immer 1 und Bit 15 immer 0.

⁽⁵⁾ In der Antriebsfunktion UF/HSPAM wird anstelle des 32-Bit-Positionsistwertes die Position in 16 Bit und der aktuelle Wirkstrom übertragen.

2.2 Profile Velocity Mode (PV, Mode 3)

Mit der Betriebsart „Profile Velocity Mode“ stehen dem Benutzer folgende Möglichkeiten zur Verfügung:

- Geschwindigkeitsvorgabe über SERVOLINK 4

Dabei muss das Bewegungsprofil in der übergeordneten Steuerung erzeugt werden.

- ▶ SERVOLINK 4-Übertragungsrate: 2 kHz (default) oder 4 kHz
- ▶ Die Datenrate der Geschwindigkeitssollwerte kann 250 Hz, 500 Hz, 1000 Hz, 2000 Hz oder 4000 Hz betragen.
- ▶ Interne Feininterpolation der Geschwindigkeitssollwerte zum Reduzieren von Sollwertsprüngen, die aus der Datenrate der Vorgabewerte resultieren
- ▶ Direkte Vorgabe einer Stromvorsteuerung im zyklischen Sollwerttelegramm
- ▶ Automatische Umrechnung bzw. Skalierung der Positionsmesssysteme in 100 nm
- ▶ Berechnung einer Stromvorsteuerung mit 4 kHz im Antriebsverstärker
- ▶ Überwachung des Geschwindigkeitsfehlers im Antriebsverstärker
- ▶ Überwachung des vorgegebenen Bewegungsprofils bezüglich maximaler Geschwindigkeit und maximaler Beschleunigung
- ▶ Gerätesteuerung nach Antriebsprofil DS402
- ▶ Zusätzliche Drehmoment- bzw. Kraftbegrenzung

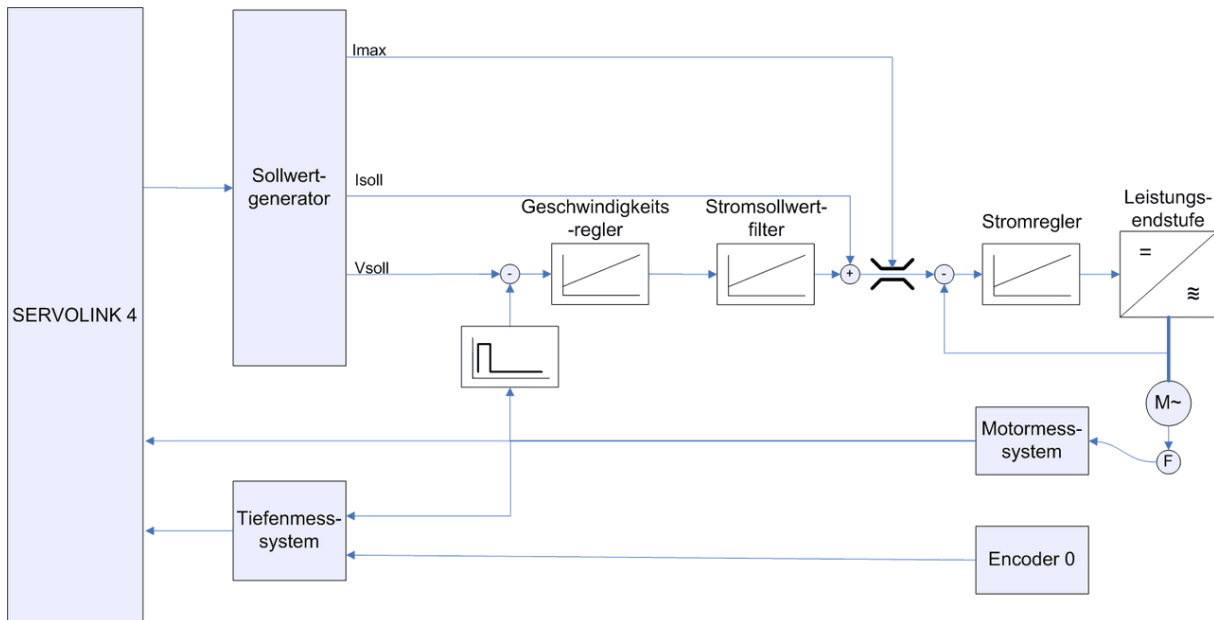


Abb. 5: Betriebsart „Profile Velocity Mode“

2.2.1 Sollwert-Telegramm

Das Sollwert-Telegramm zur Ansteuerung des Antriebs besteht aus Nutzdaten von 16 Byte, die wie folgt aufgeteilt sind:

Byte	Größe	Name	Bit	Beschreibung
0	2	ControlWord	0	Einschalten ⁽¹⁾
			1	Spannungsfreigabe ⁽¹⁾
			2	Schnellhalt ⁽¹⁾
			3	Betrieb freigeben ⁽¹⁾
			4	Reserviert
			5	Reserviert
			6	Reserviert
			7	Fehler zurücksetzen ⁽¹⁾
			8	Reserviert
			9	Reserviert
			10	Reserviert
			11	Reserviert
			12	Delta-Z-Funktion zurücksetzen ⁽²⁾
			13	Reserviert
			14	Kennung Sollwert-Telegramm / 0 ⁽³⁾
			15	Kennung Sollwert-Telegramm / 1 ⁽³⁾
2	2	Res.		Reserviert
4	2	SPG_TARGET_VELOCITY_VL		Zielgeschwindigkeit 32 Bit [0.1u/500µs*65536]
6	2	SPG_TARGET_TORQUE_LIMIT		Stromvorsteuerung 0x3fff = Spitzenstrom des Antriebsverstärkers
8	1	Res.		Reserviert
9	7	ServiceChannel		Siehe Seite 18

⁽¹⁾ Die Funktionsweise der einzelnen Bits ist in der Anleitung „Antriebssystem SD2 – Gerätesteuerung“ beschrieben.

⁽²⁾ Die Funktionsweise des einzelnen Bits ist in der Anleitung „TID_SD2_Geraetesteuerung_Tiefenmesssystem.pdf“ beschrieben.

⁽³⁾ Die Telegrammkennung ist für SERVOLINK 4 fest vorgegeben: Bit 14 ist immer 0 und Bit 15 immer 1.

2.2.2 Istwert-Telegramm

Das Istwert-Telegramm des Antriebs besteht aus Nutzdaten von 16 Byte, die wie folgt aufgeteilt sind:

Byte	Größe	Name	Bit	Beschreibung
0	2	StatusWord	0	Einschaltbereit ⁽¹⁾
			1	Eingeschaltet ⁽¹⁾
			2	Betrieb freigegeben ⁽¹⁾
			3	Fehler ⁽¹⁾
			4	Spannungsfreigabe ⁽¹⁾
			5	Schnellhalt ⁽¹⁾
			6	Einschaltsperr ⁽¹⁾
			7	Warnung ⁽¹⁾
			8	Status des digitalen Eingangs 4 / Messsystemkontakt ⁽²⁾
			9	Remote ⁽¹⁾
			10	Sollwert erreicht ⁽¹⁾
			11	Interner Grenzwert aktiv ⁽¹⁾
			12	Drehzahl 0 ⁽¹⁾
			13	SERVOLINK Toggle-Bit ⁽³⁾
			14	Kennung Istwert-Telegramm / 1 ⁽⁴⁾
			15	Kennung Istwert-Telegramm / 0 ⁽⁴⁾
2	4	PCTRL_POSITION_ACTUAL_VALUE	Aktuelle Position Lageregler 32 Bit, Auflösung 0,1 µm	
6	2	DELTAZ_POSITION_ACTUAL_VALUE	Aktuelle Delta Z Position, 16 Bit, Auflösung 0,1 µm	
8	2	ICTRL_IQ_REFERENCE	Aktueller Strom 0x3fff = Spitzenstrom des Antriebsverstärkers	
10	6	ServiceData Channel	Siehe Seite 18	

⁽¹⁾ Die Funktionsweise der einzelnen Bits ist in der Anleitung „Antriebssystem SD2 – Gerätesteuerung“ beschrieben.

⁽²⁾ Die Funktionsweise des einzelnen Bits ist in der Anleitung „TID_SD2_Geraetesteuerung_Tiefenmesssystem.pdf“ beschrieben.

⁽³⁾ Das Bit wird bei jedem Aktualisieren der Istwerte getoggelt.

⁽⁴⁾ Die Telegrammkennung ist für SERVOLINK 4 fest vorgegeben: Bit 14 ist immer 1 und Bit 15 immer 0.

2.3 Interpolierende Lageregelung (Mode 7)

Mit der Betriebsart „Interpolierende Lageregelung“ stehen dem Benutzer die folgenden Möglichkeiten zur Verfügung:

- ▶ Zyklische Positionssollwertvorgabe über SERVOLINK 4
Dabei muss das Bewegungsprofil in der übergeordneten Steuerung erzeugt werden.
- ▶ SERVOLINK 4-Übertragungsrate: 2 kHz (default) oder 4 kHz
- ▶ Die Datenrate der Geschwindigkeitssollwerte kann ein ganzzahliges Vielfaches von 4 kHz sein.
- ▶ Interne Berechnung der Geschwindigkeitssollwerte aus den Positionssollwerten mit Feininterpolation

- ▶ Lageregelung mit 4 kHz im Antriebsverstärker
- ▶ Automatische Umrechnung bzw. Skalierung der Positionsmesssysteme in 100 nm
- ▶ Berechnung einer Stromvorsteuerung mit 4 kHz im Antriebsverstärker
- ▶ Überwachung des vorgegebenen Bewegungsprofils bezüglich maximaler Geschwindigkeit und maximaler Beschleunigung
- ▶ Schleppfehlerüberwachung im Antriebsverstärker
- ▶ Gerätesteuerung nach Antriebsprofil DS402
- ▶ Zusätzliche Drehmoment- bzw. Kraftbegrenzung

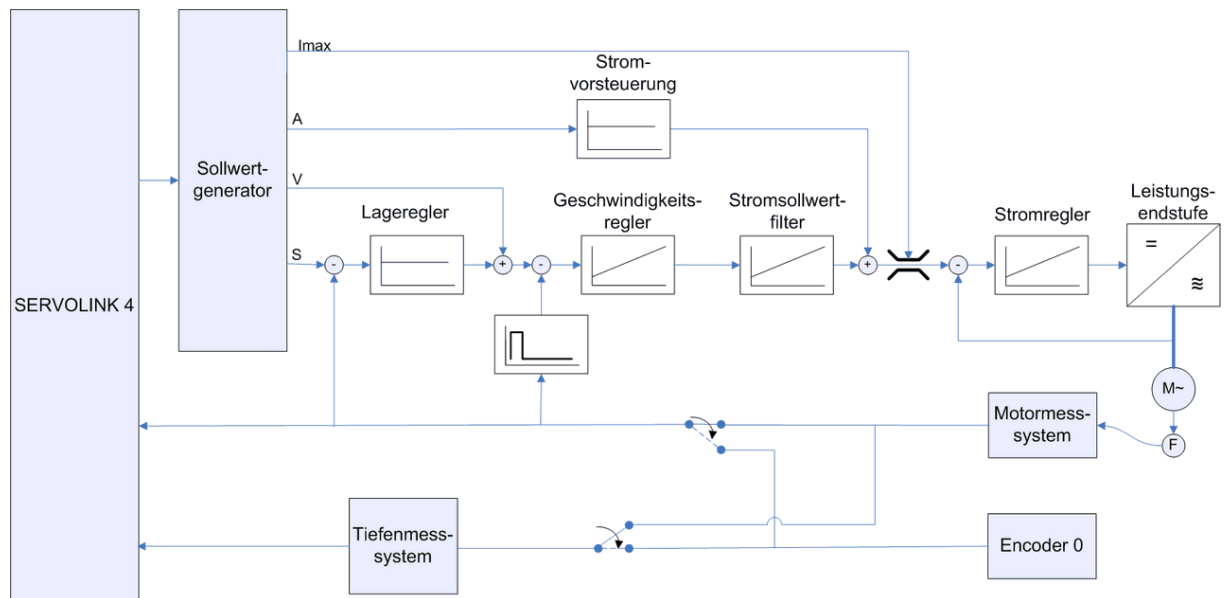


Abb. 6: Betriebsart „Interpolierende Lageregelung“

2.3.1 Sollwert-Telegramm

Das Sollwert-Telegramm zur Ansteuerung des Antriebs besteht aus Nutzdaten von 16 Byte, die wie folgt aufgeteilt sind:

Byte	Größe	Name	Bit	Beschreibung
0	2	ControlWord	0	Einschalten ⁽¹⁾
			1	Spannungsfreigabe ⁽¹⁾
			2	Schnellhalt ⁽¹⁾
			3	Betrieb freigeben ⁽¹⁾
			4	IP-Mode freigeben ⁽²⁾
			5	Reserviert ⁽²⁾
			6	Reserviert ⁽²⁾
			7	Fehler zurücksetzen ⁽¹⁾
			8	Halt-Bit (immer 1) ⁽²⁾
			9	interpolierende Lageregelung ⁽³⁾
			10	Geschwindigkeitsmodus 1 ⁽³⁾
			11	Profile Velocity Mode (PV) ⁽³⁾
			12	Delta Z Funktion zurücksetzen ⁽⁴⁾
			13	Reserviert
			14	Kennung Sollwert-Telegramm / 0 ⁽⁵⁾
			15	Kennung Sollwert-Telegramm / 1 ⁽⁵⁾
2	6	INTERPOLATION_DATA	Zielposition (Format 0.1μ*65536)	
8	1	INTERPOLATION_CONTROLWORD	Siehe Anleitung „Antriebssystem SD2 – Gerätesteuerung“	
9	7	ServiceChannel	Siehe Seite 18	

⁽¹⁾ Die Funktionsweise der einzelnen Bits ist in der Anleitung „Antriebssystem SD2 – Gerätesteuerung“ beschrieben.

⁽²⁾ Die Bits 4, 5, 6 und 8 sind betriebsartenspezifische Steuerbits.

⁽³⁾ Die Bits 9, 10 und 11 bestimmen die Betriebsart.

⁽⁴⁾ Die Funktionsweise des einzelnen Bits ist in der Anleitung „TID_SD2_Geraetesteuerung_Tiefenmesssystem.pdf“ beschrieben.

⁽⁵⁾ Die Telegrammkennung ist für SERVOLINK 4 fest vorgegeben: Bit 14 ist immer 0 und Bit 15 immer 1.

2.3.2 Istwert-Telegramm

Das Istwert-Telegramm des Antriebs besteht aus Nutzdaten von 16 Byte, die wie folgt aufgeteilt sind:

Byte	Größe	Name	Bit	Beschreibung
0	2	StatusWord	0	Einschaltbereit ⁽¹⁾
			1	Eingeschaltet ⁽¹⁾
			2	Betrieb freigegeben ⁽¹⁾
			3	Fehler ⁽¹⁾
			4	Spannungsfreigabe ⁽¹⁾
			5	Schnellhalt ⁽¹⁾
			6	Einschaltsperr ⁽¹⁾
			7	Warnung ⁽¹⁾
			8	Status des digitalen Eingangs 4 / Messsystemkontakt ⁽²⁾
			9	Remote ⁽¹⁾
			10	Sollwert erreicht ⁽¹⁾
			11	Interner Grenzwert aktiv ⁽¹⁾
			12	Interpolierung aktiv ⁽¹⁾
			13	SERVOLINK Toggle-Bit ⁽³⁾
			14	Kennung Istwert-Telegramm / 1 ⁽⁴⁾
			15	Kennung Istwert-Telegramm / 0 ⁽⁴⁾
2	4	PCTRL_POSITION__ACTUAL_VALUE	Aktuelle Position Lageregler 32 Bit, Auflösung 0,1 µm	
6	2	DELTAZ_POSITION_ACTUAL_VALUE	Aktuelle Delta Z Position, 16 Bit, Auflösung 0,1 µm	
8	2	ICTRL_IQ_REFERENCE	Aktueller Stromsollwert 0x3fff = Spitzenstrom des Antriebsverstärkers	
10	6	ServiceData Channel	Siehe Seite 18	

⁽¹⁾ Die Funktionsweise der einzelnen Bits ist in der Anleitung „Antriebssystem SD2 – Gerätesteuerung“ beschrieben.

⁽²⁾ Die Funktionsweise des einzelnen Bits ist in der Anleitung „TID_SD2_Geraetesteuerung_Tiefenmesssystem.pdf“ beschrieben.

⁽³⁾ Das Bit wird bei jedem Aktualisieren der Istwerte getoggelt.

⁽⁴⁾ Die Telegrammkennung ist für SERVOLINK 4 fest vorgegeben: Bit 14 ist immer 1 und Bit 15 immer 0.

3 Servicedatenkanal

Der Servicedatenkanal ermöglicht, ein beliebiges Datenobjekt zu lesen und zu schreiben. Die Inhalte werden dabei sequentiell über den zyklischen Datenkanal übertragen. Über den zyklischen Datenkanal können auch große Datenstrukturen und ganze Programme übertragen werden. Dies erfolgt im Gegensatz zu den Objekten im Prozessdatenkanal nicht zyklisch, sondern nur im Bedarfsfall.

3.1 Abbildung im Telegramm

3.1.1 Sollwert-Telegramm

Das Sollwert-Telegramm zur Ansteuerung des Antriebs besteht aus Nutzdaten von 16 Bytes. Dabei werden die Bytes 9 bis 15 für den Servicekanal genutzt:

Byte	Größe	Name	Bit	Beschreibung
0	9	CyclicDataChannel		Prozessdatenkanal
9	1	ServiceControl	0	ServiceValidToggle
			1	ServiceFunction Bit 0
			2	ServiceFunction Bit 1
			3	ServiceLastValidByteIndex Bit 0
			4	ServiceLastValidByteIndex Bit 1
			5	Reserviert S&M
			6	Reserviert S&M
			7	Reserviert S&M
10	2	ServiceIndex		Objektnummer
12	4	ServiceValue		Objektwert

Steuerbits „ValidToggle“

Durch einen Vergleich der Bits „ServiceValidToggle“ des Sollwert-Telegramms und „ServiceDoneToggle“ des Istwert-Telegramms kann ermittelt werden, ob eine Serviceanfrage noch aktiv ist oder ob bereits ein Ergebnis vorliegt. Haben die Bits den gleichen Wert, so kann das Ergebnis im Istwert-Telegramm gelesen werden. Sind die Werte unterschiedlich, ist eine Anfrage noch nicht beendet. Der Master toggelt das Bit „ServiceValidToggle“ bei jeder Serviceanfrage. Der Slave zieht nach Abarbeitung das „ServiceDoneToggle“ nach. Das Bit „ServiceValidToggle“ darf nur getoggelt werden, wenn es den gleichen Wert wie das Bit „ServiceDoneToggle“ hat. Der Ablauf wird in der [Abbildung 7 „Ablauf: Lesen und Schreiben von Objekten“, Seite 22](#) dargestellt.

Steuerbits „ServiceFunction0/1“

Durch die Bits „ServiceFunction0“ und „ServiceFunction1“ wird die gewünschte Funktion ausgewählt:

Wert	Bedeutung
0	Objekt lesen
1	Array-Index einstellen
2	Objekt schreiben
3	Reserviert

Steuerbits „ServiceLastValidByteIndex0/1“

Die Bits „ServiceLastValidByteIndex0/1“ geben die Anzahl der gültigen Datenbytes im letzten Datenwort an:

Wert	Bedeutung
0	1 Byte gültig
1	2 Bytes gültig
2	3 Bytes gültig
3	4 Bytes gültig

Index

Der Index beschreibt die ausgewählte Objektnummer.

Value (Schreiben)

In den Bytes 12 bis 15 wird der Objektwert übertragen.

3.2 Istwert-Telegramm

Das Istwert-Telegramm zur Ansteuerung des Antriebs besteht aus Nutzdaten von 16 Bytes. Dabei werden die Bytes 10 bis 13 und 15 für den Servicekanal genutzt:

Byte	Größe	Name	Bit	Beschreibung
0	10	CyclicDataChannel		Prozessdatenkanal
10	4	ServiceValue		Rückgabedaten
14	1	Reserved		Reserviert
15	1	ServiceState	0	ServiceDoneToggle
			1	ServiceFault
			2	Reserviert S&M
			3	Reserviert S&M
			4	Reserviert S&M
			5	Reserviert S&M
			6	Reserviert S&M
			7	Reserviert S&M

Statusbit „ServiceDoneToggle“

Das Bit „ServiceDoneToggle“ quittiert die Übernahme eines Datenwortes (siehe [Abbildung 7 „Ablauf: Lesen und Schreiben von Objekten“, Seite 22](#)).

Statusbit „ServiceFault“

Das Bit „ServiceFault“ zeigt die erfolgreiche Übertragung der Objektdaten an (0 = erfolgreich, 1 = nicht erfolgreich). Der Zustand des Bits ist gültig, sobald „ServiceValidToggle“ und „ServiceDoneToggle“ den gleichen Wert haben. Während einer laufenden Serviceanfrage (ServiceValidToggle ≠ ServiceDoneToggle) ist der Zustand undefiniert. Sollte eine folgende Serviceanfrage erfolgreich abgearbeitet werden, so wird dieses Bit entsprechend wieder gelöscht.

Value (Lesen)

In den Bytes 10 bis 13 wird der Objektwert übertragen.

3.3 Zugriff auf die Antriebsobjekte

3.3.1 Azyklische Servicedaten

Der Master steuert den Zugriff über den Servicedatenkanal über das Signal ServiceValidToggle. Der Antrieb meldet den Status der Bearbeitung über das Signal „ServiceDoneToggle“. Wird kein Servicekommando bearbeitet, haben die beiden Signale „ServiceValidToggle“ und „ServiceDoneToggle“ den gleichen Pegel. Ein neues Servicekommando wird vom Master gestartet, indem das Signal „ServiceValidToggle“ getoggelt wird. Ist die Bearbeitung des Servicekommandos durch den Antrieb beendet, wird dieses der Steuerung über die Nachführung des Signals „ServiceDoneToggle“ mitgeteilt.

Folgende Zugriffe sind möglich:

- ▶ Lesen eines Objektes:
Ein Lesezugriff wird im Sollwert-Telegramm des Servicedatenkanals über ‚ServiceFunction = 0‘ (Byte 17) gesteuert. Dabei wird über „ServiceIndex“ (low/high) das entsprechende Objekt ausgewählt. Das Ergebnis des Lesezugriffs wird dann im Istwert-Telegramm des Servicedatenkanals über „ServiceReturn“ zurückgeliefert.
Über Array-Index 0...3 kann ein Subindex übertragen werden.
- ▶ Setzen eines Array-Index:
Das Setzen eines Array-Index wird im Sollwert-Telegramm des Servicedatenkanals über ‚ServiceFunction = 1‘ (Byte 17) gesteuert. Dabei wird über „ServiceIndex“ (low/high) das entsprechende Objekt ausgewählt. Über Array-Index 0...3 wird der Array-Index übertragen.
- ▶ Schreiben eines Objektes:
Ein Schreibzugriff wird im Sollwert-Telegramm des Servicedatenkanals über ‚ServiceFunction = 2‘ gesteuert. Dabei wird über „ServiceIndex“ (low/high) das entsprechende Objekt ausgewählt, während „ServiceData“ den entsprechenden Wert enthält. Mittels „ServiceLastValidByteIndex“ muss die Anzahl der gültigen Bytes (minus 1) in diesem Telegramm festgelegt werden.

Beschreiben von Arrays

Im azyklischen Servicedatenkanal stehen maximal 32 Bit Nutzdaten zur Verfügung. Daher können auch nur Objekte mit einer maximalen Größe von 32 Bit geschrieben werden.

Sollen größere Objekte wie z. B. Arrays geschrieben werden, ist ein zusätzlicher Mechanismus notwendig. Zum Schreiben von Arrays muss zuerst der Array-Index, in den geschrieben werden soll, adressiert werden. Dies geschieht im Sollwert-Telegramm des Servicedatenkanals über ‚ServiceFunction = 1‘ (Set Array Index). Der Antrieb quittiert das Telegramm mit einem ServiceToggle und indem er den empfangenen Array-Index im Istwert-Telegramm zurücksendet. Im folgenden Sollwert-Telegramm können dann 1 bis 4 Bytes über das Kommando ‚ServiceFunction = 2‘ (Write Object) in das Array geschrieben werden. Der im Gerät gespeicherte Array-Index wird automatisch um die Anzahl der geschriebenen Bytes erhöht. Damit ist es möglich, in einem der folgenden Telegramme das Array mit einem erneuten Kommando ‚ServiceFunction = 2‘ (Write Object) fortlaufend zu beschreiben. Da im Antrieb nur eine Variable zur Verfügung steht, um den aktuellen Array-Index zu speichern, ist es nicht möglich, verschachtelt auf zwei Arrays zuzugreifen z. B. Schreibe Array A, Schreibe Array B, Schreibe Array A usw.

Lesen von Arrays

Beim Lesen von Array-Objekten wird in jedem Sollwert-Telegramm der Array-Index übertragen. Das Setzen eines Array-Index ist daher nicht nötig. Die im Antrieb gespeicherte Variable wird beim Lesezugriff nicht erhöht. Daher unterbricht ein Lesezugriff auf ein Array das inkrementelle Schreiben auf ein Array nicht.

Fehler beim Objektzugriff

Wenn der Zugriff auf ein Antriebsobjekt fehlerhaft ist (Index ist nicht gültig, Wertebereich wird überschritten, Objekt ist nicht beschreibbar usw.), wird dieser Fehler im Istwert-Telegramm über das Signal „ServiceFault“ angezeigt. Zusätzlich wird über „ServiceReturn“ die detaillierte Fehlernummer zurückgegeben.

3.3.2 Übersicht Zugriffsfunktionen

Sollwert-Telegramm	Byte	Objekt lesen (0)	Objekt schreiben (2)	Array-Index setzen (1)
ServiceControl	9	Servicesteuerung	Servicesteuerung	Servicesteuerung
ServiceIndex	10	Index LOW-Byte	Index LOW-Byte	Nicht belegt
	11	Index HIGH-Byte	Index HIGH-Byte	Nicht belegt
ServiceValue	12	Array-Index 0	Byte 0	Array-Index 0
	13	Array-Index 1	Byte 1	Array-Index 1
	14	Array-Index 2	Byte 2	Array-Index 2
	15	Array-Index 3	Byte 3	Array-Index 3

Istwert-Telegramm	Byte	Objekt lesen (0)	Objekt schreiben (2)	Array-Index setzen (1)
ServiceReturn	10	Byte 0 / Fehlercode	Array-Index 0 / Fehlercode	Array-Index 0
	11	Byte 1 / Fehlercode	Array-Index 1 / Fehlercode	Array-Index 1
	12	Byte 2	Array-Index 2	Array-Index 2
	13	Byte 3	Array-Index 3	Array-Index 3
ServiceValue	14	-	-	-
	15	Servicestatus	Servicestatus	Servicestatus

3.3.3 Lesen und Schreiben von einfachen Objekten

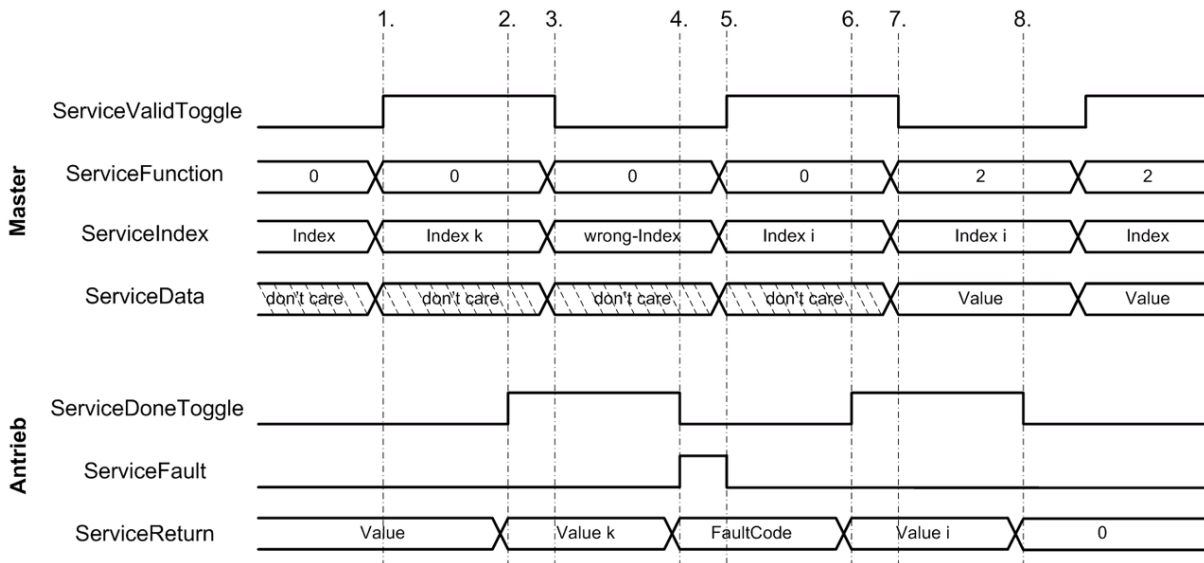


Abb. 7: Ablauf: Lesen und Schreiben von Objekten

1. Anstarten des Auslesens des Objektes mit dem Index ,k'.
2. Auslesen des Objektes mit dem Index ,k' ist fehlerfrei abgeschlossen.
3. Anstarten des Auslesens eines Objektes mit falschem Index.
4. Auslesen des Objektes mit falschem Index ist mit Fehler abgeschlossen.
5. Anstarten des Auslesens des Objektes mit dem Index ,i'.
6. Auslesen des Objektes mit dem Index ,i' ist fehlerfrei abgeschlossen.
7. Anstarten des Schreibens des Objektes mit dem Index ,i'.
8. Schreiben des Objektes mit dem Index ,i' ist fehlerfrei abgeschlossen.

3.3.4 Lesen und Schreiben von Array-Objekten

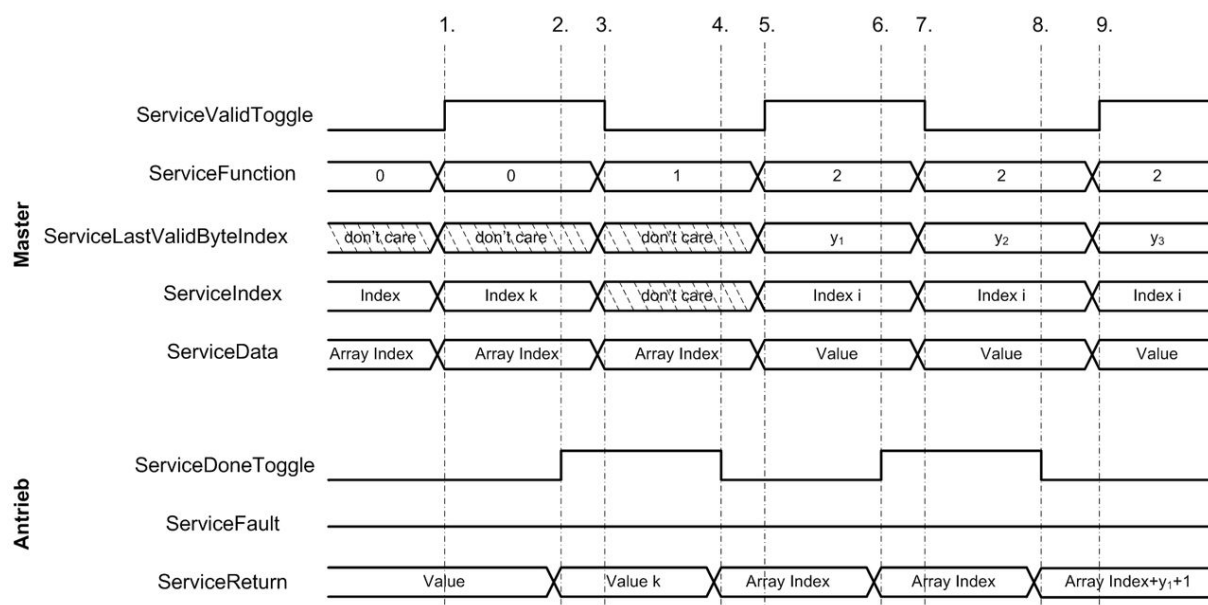


Abb. 8: Ablauf: Lesen und Schreiben von Arrays

1. Anstarten des Auslesens des Array-Objektes ‚k‘ an der Arrayposition „Array Index“.
2. Auslesen des Array-Objektes ‚k‘ ist fehlerfrei abgeschlossen.
3. Setzen der Array-Position für das darauffolgende Schreiben eines Arrays.
4. Quittierung vom Antrieb mit der Rückgabe des Array Index zur Kontrolle.
5. y_1+1 Bytes an der Arrayposition „Array Index“ in das Array ‚i‘ schreiben.
6. Quittierung vom Antrieb mit der Rückgabe des aktuellen Array Index zur Kontrolle.
7. Autoinkremental y_2+1 Bytes in das Array ‚i‘ schreiben.
8. Quittierung vom Antrieb mit der Rückgabe des aktuellen Array Index zur Kontrolle.
9. Autoinkremental y_3+1 Bytes in das Array ‚i‘ schreiben.

3.3.5 Zugriff auf Array-/Stringobjekte

Sollwertindex	Byte	Array-Objekt lesen	Array-Objekt schreiben	Array-Objekt Inkrementalschreibung
ServiceIndex	0	Index LOW-Byte	Index LOW-Byte	Index LOW-Byte
	1	Index HIGH-Byte	Index HIGH-Byte	Index HIGH-Byte
ServiceValue	0	Array-Index LOW-Byte	Array-Index LOW-Byte	Daten
	1	Array-Index HIGH-Byte	Array-Index HIGH-Byte	Daten
	2	Nicht belegt	Daten	Daten
	3	Nicht belegt	Daten	Daten

Istwertindex	Byte	Array-Objekt lesen	Array-Objekt schreiben	Array-Objekt Inkrementalschreibung
ServiceReturn	0	Daten / Fehlercode	Fehlercode	Fehlercode
	1	Daten / Fehlercode	Fehlercode	Fehlercode
	2	Daten	Nicht belegt	Array-Index LOW-Byte
	3	Daten	Nicht belegt	Array-Index HIGH-Byte

Um ein Array über den Servicedatenkanal zu lesen, wird im Sollwert-Telegramm die Bedeutung des ServiceValue angepasst, indem die 2 höherwertigen Bytes als Indexzeiger auf ein Wort des Arrays verwendet werden. Die grundsätzliche Datenübertragung funktioniert nach dem gleichen Prinzip wie bei einfachen Objekten. Als Ergebnis erhält man im Istwert-Telegramm 4 Bytes beginnend mit dem übermittelten Array-Index. Auf diese Weise kann schrittweise ein Array ausgelesen werden. Sollte bei dem Zugriff ein Fehler auftreten, so wird das ServiceFault-Bit gesetzt und der Fehlercode in den 2 höherwertigen Bytes des Istwert-Telegramms übermittelt.

Der Schreib- und Lesevorgang eines Arrays funktionieren analog. Auch hier werden die 2 höherwertigen Bytes des ServiceValue als Indexzeiger genutzt. Die 2 niederwertigen Bytes enthalten die zu schreibenden Daten ab dem angegebenen Index. In den höherwertigen Bytes des Istwert-Telegramms wird bei ServiceFault ein Fehlercode übermittelt. Da die Bandbreite des Schreibkanals nur 16 Bits je Telegramm beträgt, ist die Verwendung eines Auto-Inkrement-Modus möglich. Hierzu muss das Service-AutoIncrement-Bit gesetzt werden. Ist das Bit gesetzt, können über ServiceValue 4 Byte übertragen werden.

3.3.6 Fehlercodes des Servicedatenkanals

Fehlercode	Beschreibung
0x00	Kein Fehler
0x81 oder 0x01	Toggle-Bit nicht nachgezogen
0x86 oder 0x06	CRC Fehler
0x87 oder 0x07	Kein freier Speicher
0x88 oder 0x08	Nicht erlaubter Zugriff auf ein Objekt
0x89 oder 0x09	Versucht ein Write-Only-Objekt zu lesen
0x8A oder 0x0A	Versucht ein Read-Only-Objekt zu beschreiben
0x8B oder 0x0B	Objekt nicht in der Objekt-Bibliothek vorhanden!
0x8C oder 0x0C	Reserviert
0x8D oder 0x0D	Reserviert
0x8E oder 0x0E	Allgemeiner Parameter-Inkompatibilitätsgrund
0x8F oder 0x0F	Allgemeine interne Inkompatibilität im Gerät
0x90 oder 0x10	Zugriff verweigert aufgrund eines Hardwarefehlers
0x91 oder 0x11	Datentyp nicht korrekt, Länge des Service-Parameters nicht korrekt
0x92 oder 0x12	Datentyp nicht korrekt, Länge des Service-Parameters zu hoch
0x93 oder 0x13	Datentyp nicht korrekt, Länge des Service-Parameters zu niedrig
0x94 oder 0x14	Subindex nicht vorhanden
0x95 oder 0x15	Wertebereich des Parameters überschritten (nur bei Schreibzugriff)
0x96 oder 0x16	Wert des geschriebenen Parameters zu hoch
0x97 oder 0x17	Wert des geschriebenen Parameters zu niedrig
0x98 oder 0x18	Maximalwert niedriger als Minimalwert
0x99 oder 0x19	Allgemeiner Fehler
0x9A oder 0x1A	Daten können nicht übertragen oder in der Applikation gespeichert werden.
0x9B oder 0x1B	Daten können aufgrund des Zustands der Steuerung nicht übertragen oder in der Applikation gespeichert werden.
0x9C oder 0x1C	Daten können aufgrund des zurückgesetzten Gerätes nicht übertragen oder in der Applikation gespeichert werden.
0x9D oder 0x1D	Dynamische Erzeugung des Objektverzeichnisses nicht möglich oder kein Objektverzeichnis vorhanden.
0x9E oder 0x1E	Lesezugriff verweigert
0x9F oder 0x1F	Schreibzugriff verweigert