

# Antriebssystem SD2x

## TIA-SD2-SERVOLINK 4- Antriebsprotokoll

Beschreibung der Bussystemanwendung zur Handhabung  
des Datenaustauschs im SD2x





## **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](mailto: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](mailto: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](mailto:info@sieb-meyer.tw)

<b>1</b>	<b>Lizenzhinweise.....</b>	<b>7</b>
<b>2</b>	<b>Programmbeschreibung.....</b>	<b>9</b>
2.1	Systemaufbau.....	9
2.2	Verfügbare Beispielprogramme.....	9
2.3	Installation.....	9
<b>3</b>	<b>Programminhalt.....</b>	<b>11</b>
3.1	Organisationsbaustein.....	11
3.2	Funktionsbausteine.....	11
3.2.1	Funktionsbausteine allgemein.....	11
3.2.2	Funktionsbausteine zur Ansteuerung eines Antriebs.....	11
3.2.3	Funktionsbausteine des Servicekanals.....	11
3.2.4	Funktionsbausteine für die SC-Objekte 467 bis 470.....	11
3.2.5	Funktionsbausteine des TIA Portal-Systems.....	12
3.3	Funktionen.....	12
3.4	Instanzdatenbausteine.....	12
3.5	Globale Datenbausteine.....	12
3.6	Anwenderdefinierte Datentypen.....	12
<b>4</b>	<b>Programmsteuerung.....</b>	<b>13</b>
4.1	Programmorganisationsbaustein.....	13
4.1.1	Main.....	13
4.1.1.1	Zuweisung der Feldbusdaten.....	13
4.1.1.2	Auswertung des digitalen Eingangs I_Speed_0.....	13
4.1.1.3	Erweiterung im Baustein Main bei mehreren Antrieben.....	13
4.1.1.4	Safety Control-Funktionen.....	14
	Übertragungsablauf.....	15
4.2	Funktionsbausteine.....	17
4.2.1	Ablaufsteuerung für diverse Funktionsbausteine.....	17
4.2.2	Funktionsbaustein FB_Init_FBs.....	18
4.2.2.1	Allgemeine Beschreibung.....	18
4.2.2.2	Aufruf des Funktionsbausteins.....	18
4.2.2.3	Ein- und Ausgänge.....	19
4.2.3	Funktionsbaustein FB_InitData.....	19
4.2.3.1	Allgemeine Beschreibung.....	19
4.2.3.2	Aufruf des Funktionsbausteins.....	19
4.2.3.3	Ein- und Ausgänge.....	19
4.2.4	Funktionsbaustein FB_ReadData.....	19
4.2.4.1	Allgemeine Beschreibung.....	19
4.2.4.2	Aufruf des Funktionsbausteins.....	19
4.2.4.3	Ein- und Ausgänge.....	19
4.2.5	Funktionsbaustein FB_WriteData.....	20
4.2.5.1	Allgemeine Beschreibung.....	20
4.2.5.2	Aufruf des Funktionsbausteins.....	20
4.2.5.3	Ein- und Ausgänge.....	20
4.2.6	Funktionsbaustein FB_ReadObject.....	20
4.2.6.1	Allgemeine Beschreibung.....	20
4.2.6.2	Aufruf des Funktionsbausteins.....	20
4.2.6.3	Ein- und Ausgänge.....	20
4.2.7	Funktionsbaustein FB_WriteObject.....	21
4.2.7.1	Allgemeine Beschreibung.....	21
4.2.7.2	Aufruf des Funktionsbausteins.....	21
4.2.7.3	Ein- und Ausgänge.....	21
4.2.8	Funktionsbaustein FB_SetArrayIndex.....	21
4.2.8.1	Allgemeine Beschreibung.....	21
4.2.8.2	Aufruf des Funktionsbausteins.....	21
4.2.8.3	Ein- und Ausgänge.....	21
4.2.9	Funktionsbaustein FB_SC_Handle.....	21
4.2.9.1	Allgemeine Beschreibung.....	21
4.2.9.2	Aufruf des Funktionsbausteins.....	22
4.2.9.3	Ein- und Ausgänge.....	22
4.2.9.4	Ablauf des Funktionsbausteins.....	23



4.2.10	Funktionsbaustein FB_SC_FillObject.....	24
4.2.10.1	Allgemeine Beschreibung.....	24
4.2.10.2	Aufruf des Funktionsbausteins.....	24
4.2.10.3	Ein- und Ausgänge.....	24
4.2.11	Funktionsbaustein FB_SC_HandleObject.....	24
4.2.11.1	Allgemeine Beschreibung.....	24
4.2.11.2	Aufruf des Funktionsbausteins.....	25
4.2.11.3	Ein- und Ausgänge.....	25
4.2.11.4	Ablauf des Funktionsbausteins.....	26
4.2.12	Funktionsbaustein FB_SC_CheckObject.....	27
4.2.12.1	Allgemeine Beschreibung.....	27
4.2.12.2	Aufruf des Funktionsbausteins.....	27
4.2.12.3	Ein- und Ausgänge.....	27
4.2.13	Funktionsbaustein FB_SwapINT.....	27
4.2.13.1	Allgemeine Beschreibung.....	27
4.2.13.2	Aufruf des Funktionsbausteins.....	27
4.2.13.3	Ein- und Ausgänge.....	27
4.2.14	Funktionsbaustein FB_SwapDINT.....	28
4.2.14.1	Allgemeine Beschreibung.....	28
4.2.14.2	Aufruf des Funktionsbausteins.....	28
4.2.14.3	Ein- und Ausgänge.....	28
<b>4.3</b>	<b>Funktionen.....</b>	<b>28</b>
4.3.1	Funktion FC_GetStatus.....	28
4.3.1.1	Allgemeine Beschreibung.....	28
4.3.1.2	Aufruf des Funktionsbausteins.....	28
4.3.1.3	Ein- und Ausgänge.....	28
<b>4.4</b>	<b>Globale Datenbausteine.....</b>	<b>29</b>
4.4.1	Datenbaustein DB_DRIVESTATE.....	29
4.4.2	Datenbaustein DB_ERRORCODE.....	29
4.4.3	Datenbaustein DB_GLOBALDATA.....	29
4.4.4	Datenbaustein DB_SC_PROCESSDATA.....	30
<b>4.5</b>	<b>SPS-Variablen.....</b>	<b>31</b>
4.5.1	Standardvariablen.....	31
<b>4.6</b>	<b>Anwenderdefinierte Datentypen.....</b>	<b>31</b>
4.6.1	Anwenderdefinierter Datentyp TS_DRIVE_STATUSWORD.....	31
4.6.2	Anwenderdefinierter Datentyp TS_DRIVE_DATA_IN.....	32
4.6.3	Anwenderdefinierter Datentyp TS_DRIVE_DATA_OUT.....	32
4.6.4	Anwenderdefinierter Datentyp TS_DRIVE_DATA.....	33
4.6.5	Anwenderdefinierter Datentyp TS_DRIVE_SETPOINTS.....	33
4.6.6	Anwenderdefinierter Datentyp TS_DRIVE_ACTUALVALUES.....	34
4.6.7	Anwenderdefinierter Datentyp TS_OBJECTDATA_SC.....	34
4.6.8	Anwenderdefinierter Datentyp TS_SC_PROCESSDATA.....	34
4.6.9	Anwenderdefinierter Datentyp TS_OBJECTDATA.....	34
4.6.10	Anwenderdefinierter Datentyp TS_HANDLE_DATA.....	35
<b>4.7</b>	<b>Checksumme bilden.....</b>	<b>35</b>
<b>4.8</b>	<b>Systemfunktionen und Systemfunktionsbausteine.....</b>	<b>35</b>
<b>5</b>	<b>Hardwarekonfiguration.....</b>	<b>37</b>
5.1	Einbinden der CPU.....	37
5.2	Einbinden der Gerätedateien.....	38
5.3	Einbinden eines PROFINET IO-Gateways 0362157(A).....	38
5.4	Einbinden eines PROFIBUS DP-Gateways 0362151(A).....	40
<b>6</b>	<b>Laden des Programmbeispiel.....</b>	<b>43</b>
6.1	Projekt übersetzen.....	43
6.2	Programmbeispiel laden.....	43
<b>7</b>	<b>Fehlererkennung.....</b>	<b>45</b>
7.1	Fehler in der S7-SPS.....	45
7.2	Fehler in der Antriebssteuerung SD2x.....	45
<b>8</b>	<b>SERVOLINK 4-Gateway-Testaufbau.....</b>	<b>47</b>

<b>8.1</b>	<b>Überprüfung an der S7-SPS.....</b>	<b>47</b>
<b>8.2</b>	<b>Überprüfung im Antrieb.....</b>	<b>47</b>
8.2.1	Motoransteuerung.....	47
<b>9</b>	<b>Umstellung des Feldbussystems.....</b>	<b>49</b>
<b>10</b>	<b>Weiterführende Dokumente.....</b>	<b>51</b>



# 1 Lizenzhinweise

Die in diesem Dokument beschriebenen SIEB & MEYER-Beispielprogramme unterliegen den folgenden Lizenzbedingungen.

Copyright © 2017, SIEB & MEYER AG. Alle Rechte vorbehalten.

Weiterverbreitung und Verwendung der Software im Quellformat oder in Binärform, mit oder ohne Veränderung, sind unter den folgenden Bedingungen zulässig:

- ▶ Weiterverbreitungen des Quellcodes müssen den obigen Urheberrechtsvermerk, diese Liste der Bedingungen und den folgenden Haftungsausschluss enthalten.
- ▶ Weiterverbreitete Binärform-Exemplare müssen den obigen Urheberrechtsvermerk, diese Liste der Bedingungen und den folgenden Haftungsausschluss in der Dokumentation und/oder anderen Materialien, die mit dem Exemplar verbreitet werden, enthalten.
- ▶ Alle Abweichungen von diesen Bedingungen erfordern die vorherige, schriftliche Genehmigung des Urheberrechtinhabers.

## Haftungsausschluss

DIESE SOFTWARE WIRD VON DEN URHEBERRECHTSINHABERN, DEN AUTOREN UND DEN BEITRAGSLEISTENDEN AUF DER BASIS „AS IS“ (SO WIE SIE IST) BEREITGESTELLT UND JEDLICHE AUSDRÜCKLICHE ODER STILLSCHWEIGENDE GEWÄHRLEISTUNG WIRD ABGELEHNT, DARUNTER INSBESONDERE DIE STILLSCHWEIGENDEN GEWÄHRLEISTUNGEN DER ALLGEMEINEN GEBRAUCHSTAUGLICHKEIT UND DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. UNTER KEINEN UMSTÄNDEN SIND DIE URHEBERRECHTSINHABER ODER DIE AUTOREN ODER DIE BEITRAGSLEISTENDEN HAFTBAR FÜR WIE AUCH IMMER ENTSTANDENE DIREKTE UND INDIREKTE SCHÄDEN, FOLGE- UND NEBENKOSTEN, KONKRETE SCHÄDEN, VERSCHÄRFTE SCHADENERSATZFORDERUNGEN ODER MANGELFOLGESCHÄDEN (DARUNTER INSBESONDERE DIE BESCHAFFUNG VON ERSATZPRODUKTEN ODER -DIENSTLEISTUNGEN, NUTZUNGS- ODER GEWINNAUSFÄLLE, DATENVERLUSTE UND GESCHÄFTSUNTERBRECHUNGEN); UNABHÄNGIG DAVON, AUF WELCHE HAFTUNGSTHEORIE DIESE GESTÜTZT WERDEN, GLEICHGÜLTIG, OB VERTRAGSHAFTUNG, VERSCHULDENSUNABHÄNGIGE HAFTUNG ODER DELIKTSHAFTUNG (INKLUSIVE FAHRLÄSSIGKEIT ODER SONSTIGES), UND DIE IN IRGEND EINER ART UND WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN SIE AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDEN.



## 2 Programmbeschreibung

Dieses Dokument beschreibt die Handhabung der zyklischen Prozessdaten und des Servicekanals zwischen SPS und SD2x über ein Gateway sowie den Zugriff auf die optionalen Sicherheitsfunktionen im SD2x als Erweiterung.

Hinweise zu den beschriebenen Programmbeispielen:

- ▶ Die hier beschriebenen Programmbeispiele werden für die verschiedenen Bussysteme benutzt.
- ▶ Der Programmablauf wird über die Aufrufe im Organisationsbaustein OB1 (Main) gesteuert.
- ▶ Um die Programmbeispiele übersichtlich zu halten, wurde auf die Programmierung der Fehlerauswertung verzichtet. Bei Bedarf müssen diese zusätzlichen Auswertungen vom Anwender programmiert werden.

### 2.1 Systemaufbau

Das SPS-Programm wurde mit dem Engineering-Tool TIA Portal von der Firma Siemens erstellt. Das Programm umfasst alle für die Kommunikation benötigten Funktionsbausteine und Funktionen.

Um das Programm für die Kommunikation zwischen der SPS und den SD2x-Antrieben von SIEB & MEYER einsetzen zu können, sind folgende Hardware- und Software-Komponenten erforderlich:

- ▶ TIA Portal V15.1
- ▶ Siemens SPS S7-1500
  - CPU 1511 C-1 PN
  - AI5 / AQ2\_1
  - DI16 / DQ16\_1
  - CP1542-5\_1
- ▶ SIEB&MEYER SERVOLINK 4-Gateway
  - Gateway 0362151(A) für PROFIBUS DP  
(Bis zu 12 Antriebe können an das Gateway angeschlossen werden.)
  - Gateway 0362157(A) für PROFINET IO  
(Bis zu 24 Antriebe können an das Gateway 0362157 angeschlossen werden und bis zu 16 Antriebe an 0362157A.)
- ▶ SIEB & MEYER SD2x

Die eingesetzte SPS kommuniziert über das entsprechende Feldbussystem mit dem Gateway. Das Gateway leitet die Informationen über weiter an die SD2x-Antriebe.

### 2.2 Verfügbare Beispielprogramme

Das folgende Beispielprogramm steht zur Verfügung:

- ▶ TIA\_SD2\_GATEWAY für PROFIBUS DP und PROFINET IO

### 2.3 Installation

1. Erstellen Sie zunächst ein Arbeitsverzeichnis für die Anwendungsprogramme. Beispiel: C:\SM\_AG\workspace



2. Kopieren Sie die ZAP-Datei des gewünschten Programms (z. B. „TIA\_SD2\_GATEWAY\_20190418\_0825.zap15\_1“) in das erstellte Verzeichnis.
  3. Starten Sie die TIA Portal-Oberfläche und wechseln Sie in die Projektansicht.
  4. Wählen Sie das Menü „Projekt → Dearchivieren ...“ aus.
  5. Wechseln Sie in das erstellte Verzeichnis und wählen Sie die ZAP-Datei aus. Bestätigen Sie Ihre Auswahl mit „Öffnen“. Nun wird ein Fenster zur Auswahl des Zielverzeichnisses geöffnet.
  6. Wählen Sie das gewünschte Projektverzeichnis (z. B. C:\SM\_AG\workspace) aus und klicken Sie auf „Ordner auswählen“. Die ZAP-Datei wird nun dearchiviert.
- ✓ Das Projekt befindet sich in einem Unterverzeichnis des Arbeitsverzeichnisses (z. B. C:\SM\_AG\workspace\TIA\_SD2\_GATEWAY). Anschließend steht das Projekt auch in der Oberfläche zur Verfügung.

## 3 Programminhalt

Das SPS-Programmbeispiel beinhaltet die folgenden Programmbausteine als SCL-Quellen.

### 3.1 Organisationsbaustein

Symbolischer Name	Nummer	Beschreibung
Main	OB1	Hauptprogramm

### 3.2 Funktionsbausteine

#### 3.2.1 Funktionsbausteine allgemein

Symbolischer Name	Nummer	Beschreibung
FB_Init_FBs	FB6	initialisiert die weiteren Funktionsbausteine
FB_SwapDINT	FB10	tauscht die Bytereihenfolge eines 4-Byte-Integerwerts
FB_SwapINT	FB11	tauscht die Bytereihenfolge eines 2-Byte-Integerwerts

#### 3.2.2 Funktionsbausteine zur Ansteuerung eines Antriebs

Symbolischer Name	Nummer	Beschreibung
FB_InitData	FB5	initialisiert die Feldbusdaten
FB_ReadData	FB7	wertet die zyklischen Feldbusdaten zur Antriebsstruktur aus
FB_WriteData	FB12	stellt die zyklischen Feldbusdaten aus der Antriebsstruktur zusammen

#### 3.2.3 Funktionsbausteine des Servicekanals

Symbolischer Name	Nummer	Beschreibung
FB_ReadObject	FB8	liest ein Objekt vom Antrieb
FB_SetArrayIndex	FB9	setzt den Array-Index-Zähler im Antrieb
FB_WriteObject	FB13	beschreibt ein Objekt im Antrieb

#### 3.2.4 Funktionsbausteine für die SC-Objekte 467 bis 470

Symbolischer Name	Nummer	Beschreibung
FB_SC_CheckObject	FB1	prüft die eingehenden Daten des Objekts
FB_SC_FillObject	FB2	füllt das Objekt mit den vorgegebenen Daten
FB_SC_HandleObject	FB4	steuert die Objektzugriffe
FB_SC_Handle	FB3	steuert den Zugriff auf einen Antrieb



### 3.2.5 Funktionsbausteine des TIA Portal-Systems

Funktionsbaustein	Nummer	Beschreibung
TON	IEC_Timer_0_DB	Einschaltverzögerung erzeugen; wird für die Zeitüberwachung genutzt

### 3.3 Funktionen

Symbolischer Name	Nummer	Beschreibung
FC_GetStatus	FC1	ermittelt den Gerätezustand aus dem Statuswort des Antriebs

### 3.4 Instanzdatenbausteine

Symbolischer Name	Nummer	Beschreibung
FB_SC_Handle_DB	DB7	Instanzdatenbaustein für FB_SC_Handle
FB_InitFBs_DB	DB5	Instanzdatenbaustein für FB_Init_FBs
FB_ReadData_DB	DB8	Instanzdatenbaustein für FB_ReadData
FB_WriteData_DB	DB9	Instanzdatenbaustein für FB_WriteData

### 3.5 Globale Datenbausteine

Symbolischer Name	Nummer	Beschreibung
DB_DRIVESTATE	DB1	Konstantenliste für den Status des Antriebs gemäß Antriebsprotokoll DS402
DB_ERRORCODE	DB2	Konstantenliste für die Fehlercodes
DB_GLOBALDATA	DB4	global genutzte Variablen
DB_SC_PROCESSDATA	DB3	Ein- und Ausgangsdaten für FB_SC_HandleObject

### 3.6 Anwenderdefinierte Datentypen

Symbolischer Name	Beschreibung
TS_DRIVE_STATUSWORD	Datenstruktur des Statusworts des Antriebs
TS_DRIVE_DATA_IN	Datenstruktur der Empfangsdaten vom Feldbussystem
TS_DRIVE_DATA_OUT	Datenstruktur der Sendedaten zum Feldbussystem
TS_DRIVE_DATA	Datenstruktur der Antriebsdaten
TS_DRIVE_ACTUALVALUES	Datenstruktur der Istwerte
TS_DRIVE_SETPOINTS	Datenstruktur der Sollwerte
TS_OBJECTDATA	Datenstruktur für den Objektzugriff
TS_HANDLE_DATA	Datenstruktur für die Steuerung eines Funktionsbausteins
TS_OBJECTDATA_SC	Struktur des Datenarrays für die SC-Objekte
TS_SC_PROCESSDATA	Datenstruktur für den Zugriff auf die SC-Objekte

# 4 Programmsteuerung

In diesem Kapitel werden die einzelnen Bausteine und ihre Funktionen beschrieben.

## 4.1 Programmorganisationsbaustein

### 4.1.1 Main

Das Programm wird durch den Organisationsbaustein Main gesteuert. Hier werden die Daten vom Feldbus gelesen, den Antrieben zugeordnet, bearbeitet und wieder auf den Feldbus geschrieben. Die einzelnen Programmschritte werden durch die Funktionsbausteine erledigt.

Vor der eigentlichen Ablaufsteuerung müssen die Daten einmal beim Starten des Programms oder nach einer Störung initialisiert werden.

#### 4.1.1.1 Zuweisung der Feldbusdaten

Die in den SPS-Variablen definierten Feldbusdaten können im Ablaufprogramm den Programmvariablen zugewiesen werden.

Beispiel für die Daten des Antriebs 0:

```
// assigning the drive number
"DB_GLOBALDATA".DriveNumber := 0;
// reading the drive actual values
"DB_GLOBALDATA".SD_Drive["DB_GLOBALDATA".DriveNumber].InData := "InData 0";
. . .
// output of drive reference values
"Outdata_0" := "DB_GLOBALDATA".SD_Drive["DB_GLOBALDATA".DriveNumber].OutData;
```

#### 4.1.1.2 Auswertung des digitalen Eingangs I\_Speed\_0

Über den Eingang I\_Speed\_0 wird die Geschwindigkeitsvorgabe festgelegt. Das Verfahren ist im [Kapitel 8.2.1 „Motoransteuerung“, Seite 47](#) beschrieben.

#### 4.1.1.3 Erweiterung im Baustein Main bei mehreren Antrieben

Für weitere Antriebe muss der durch „Drive 0“ gekennzeichnete Programmblock kopiert und angepasst werden. Für jeden neuen Programmblock sind folgende Anpassungen notwendig:

- ▶ Variable DrNo (Antriebsnummer) um 1 erhöhen
- ▶ weitere Ein- und Ausgänge für die Antriebssteuerung definieren
- ▶ Sollwerte für Geschwindigkeit und Strom anpassen

#### Beispiel

DrNo	Antrieb	SD2x-Adresse	TIA-Hardware-Konfiguration
0	Einzel	0	Servo drive 16 in / 16 out
1	Einzel	1	Servo drive 16 in / 16 out



DrNo	Antrieb	SD2x-Adresse	TIA-Hardware-Konfiguration
2	Doppel A	2	Servo drive 16 in / 16 out
3	Doppel B		Servo drive 16 in / 16 out
4	Doppel A	3	Servo drive 16 in / 16 out
5	Doppel B		Servo drive 16 in / 16 out
6	Einzel	4	Servo drive 16 in / 16 out
⋮	⋮	⋮	⋮

**Hinweis**

Achten Sie darauf, dass ein Doppelantrieb immer bei einer geraden DrNo beginnt.

### 4.1.1.4 Safety Control-Funktionen

Mit diesem Programmteil werden die Daten über den Funktionsbaustein FB\_SC\_Handle an die Safety Control-Funktionen der Antriebe geleitet.

**Hinweis**

Wenn Sie die Safety Control-Funktionen zur Unterstützung der SFM- und SLOF-Sicherheitsfunktionen in den SD2x-Antrieben nicht benötigen, kann dieser Programmteil entfallen und die hierfür benötigten Funktionsbausteine gelöscht werden.

Für die vollständige Datenübertragung und Auswertung müssen stets beide Safety Control-Funktionen (SC0 und SC1) des Antriebs angesprochen werden. Hierzu werden die Daten schrittweise für jeden Antrieb abgearbeitet:

- 0 Initialisierung der Funktionsbausteine
- 1 Vorgabewert „Grenze Drehzahl Null“ schreiben, SC0
- 2 Vorgabewert „Grenze Drehzahl Null“ schreiben, SC1
- 3 Vorgabewert „sicher begrenztes Drehfeld“ schreiben, SC0
- 4 Vorgabewert „sicher begrenztes Drehfeld“ schreiben, SC1
- 5 Istwert „sicher begrenztes Drehfeld“ lesen, SC0
- 6 Istwert „sicher begrenztes Drehfeld“ lesen, SC1
- 7 Status lesen, SC0
- 8 Status lesen, SC1
- 9 Statusauswertung
- 10 Kundenspezifischer Programmablauf

Danach werden die Punkte 5 bis 10 wiederholt. Durch Veränderung der Übergabeparameter an den FB\_SC\_Handle können weitere Befehle an die Safety Control-Funktionen gesendet werden.

## Übertragungsablauf

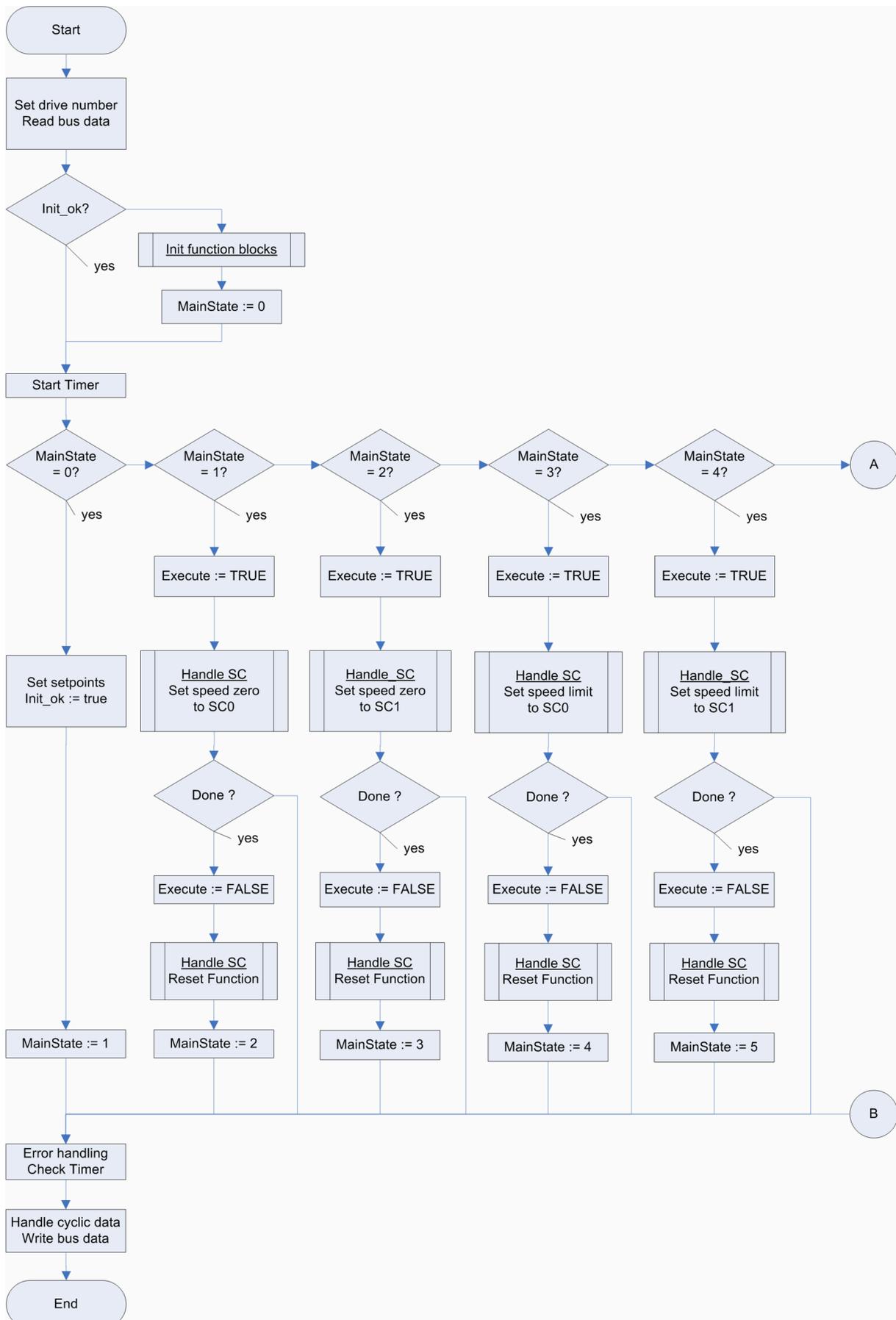


Abb. 1: Safety Control-Funktionen: Daten senden

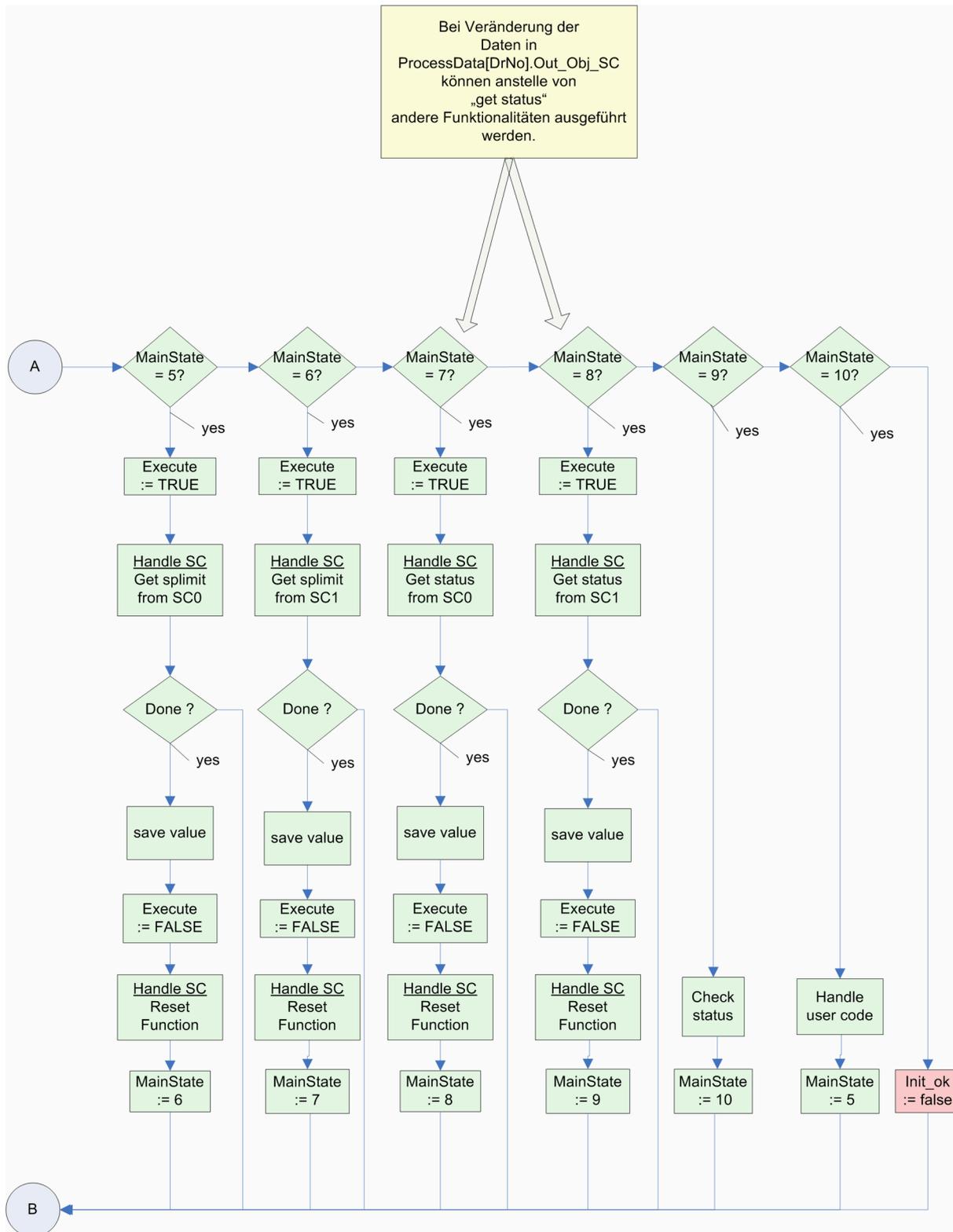


Abb. 2: Safety Control-Funktionen: Daten lesen



## Tipp

In den Parametersätzen des SD2x können Defaultwerte für „Drehzahl Null“ und „sicher begrenztes Drehfeld“ festgelegt werden. Diese Werte werden beim Starten der SD2x-Firmware an die Safety Control-Funktionen gesendet. Wenn Sie diese Defaultwerte nutzen, kann das Schreiben der Werte durch die SPS (Punkt 1 bis 4) entfallen. Stattdessen können gleich die Istwerte von den Safety Control-Funktionen gelesen werden, wenn sie in der SPS benötigt werden.

---

0	Initialisierung der Funktionsbausteine
5	Istwert „sicher begrenztes Drehfeld“ lesen, SC0
6	Istwert „sicher begrenztes Drehfeld“ lesen, SC1
7	Status lesen, SC0
8	Status lesen, SC1
9	Statusauswertung
10	Kundenspezifischer Programmablauf

---

## 4.2 Funktionsbausteine

### 4.2.1 Ablaufsteuerung für diverse Funktionsbausteine

Einige Funktionsbausteine müssen mit jedem SPS-Zyklus wiederholt aufgerufen werden, bis die Funktion vollständig abgearbeitet ist. Hierzu gehören insbesondere die Funktionsbausteine zum Lesen und Schreiben der SD2x-Objekte. Für die Ablaufsteuerung dieser Funktionsbausteine werden die Variablen des anwenderdefinierten Datentyps TS\_HANDLE\_DATA verwendet. Sie sind in den Eingangs- und Ausgangsdatenstrukturen hinterlegt. Vor der eigentlichen Nutzung der Funktionsbausteine sollten diese einmalig mit EXECUTE = FALSE im Funktionsbaustein FB\_Init\_FBs aufgerufen werden.

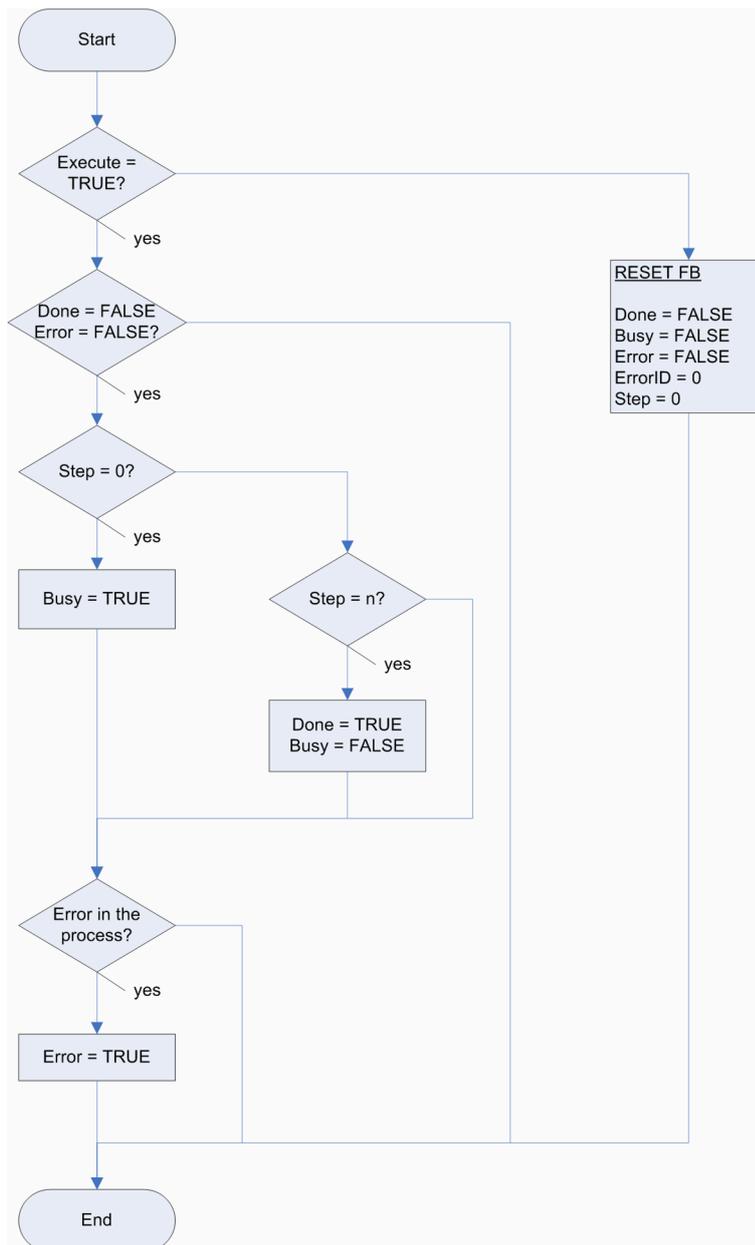


Abb. 3: Ablaufsteuerung der Funktionsbausteine

## 4.2.2 Funktionsbaustein FB\_Init\_FBs

### 4.2.2.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_Init\_FBs werden die weiteren Funktionsbausteine initialisiert. Nach erfolgreicher Durchführung der Initialisierung wird init\_ok des entsprechenden Antriebs auf TRUE gesetzt.

### 4.2.2.2 Aufruf des Funktionsbausteins

```
"FB_Init_FBs"(DrNo := "DB_GLOBALDATA".DriveNumber);
```

### 4.2.2.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer

## 4.2.3 Funktionsbaustein FB\_InitData

### 4.2.3.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_InitData werden die Befehlsbits des zyklischen Protokolls initialisiert und in der Antriebsstruktur abgelegt. Die Sollwerte werden dabei mit Null vorbelegt.

Danach müssen die Initialisierungsdaten einmal an den Antrieb gesendet werden.

### 4.2.3.2 Aufruf des Funktionsbausteins

```
#FB_InitData(DrNo := #DrNo);
```

### 4.2.3.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer

## 4.2.4 Funktionsbaustein FB\_ReadData

### 4.2.4.1 Allgemeine Beschreibung

Der Funktionsbaustein FB\_ReadData verarbeitet die zyklischen Daten des SD2x-Antriebs.

Die Daten des SD2x-Antriebs werden vom Funktionsbaustein zu einzelnen Informationen herausgefiltert. Die Informationen werden in der Antriebsstruktur abgelegt und stehen damit global im gesamten Anwenderprogramm zur Verfügung.

Am Ausgang des Bausteins steht eine Fehlerinformation zur Verfügung.

Dieser Funktionsbaustein muss immer vor der Verarbeitung der zyklischen Daten aufgerufen werden.

### 4.2.4.2 Aufruf des Funktionsbausteins

```
"FB_ReadData_DB"(DrNo := "DB_GLOBALDATA".DriveNumber);
```

### 4.2.4.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In/out	INT	Antriebsnummer
ErrorID	Out	WORD	Fehlernummer



## 4.2.5 Funktionsbaustein FB\_WriteData

### 4.2.5.1 Allgemeine Beschreibung

Der Funktionsbaustein FB\_WriteData stellt aus den Daten der Antriebsstruktur ein Sollwerttelegramm zusammen und schreibt die neuen zyklischen Parameter und Steuerbits in die Antriebsstruktur. Diese neuen Daten werden dann an den Antrieb gesendet. Am Ausgang des Bausteins steht dann eine Fehlerinformation an.

Dieser Funktionsbaustein muss immer nach der Verarbeitung der zyklischen Daten aufgerufen werden.

### 4.2.5.2 Aufruf des Funktionsbausteins

```
"FB_WriteData_DB"(DrNo := "DB_GLOBALDATA".DriveNumber);
```

### 4.2.5.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer
ErrorID	Out	WORD	Fehlernummer

## 4.2.6 Funktionsbaustein FB\_ReadObject

### 4.2.6.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_ReadObject kann lesend auf die Objektdaten des Antriebs über den Servicekanal zugegriffen werden. Das Ergebnis steht dann in der globalen Datenstruktur ObjectRdData zur Verfügung. Der funktionale Ablauf des Zugriffs ist im Handbuch „Antriebssystem SD2 – SERVOLINK 4-Bussystembeschreibung“ beschrieben.

### 4.2.6.2 Aufruf des Funktionsbausteins

```
#FB_ReadObject(DrNo := "DB_GLOBALDATA".DriveNumber);
```

### 4.2.6.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer

## 4.2.7 Funktionsbaustein FB\_WriteObject

### 4.2.7.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_WriteObject kann schreibend auf die Objektdaten des Antriebs über den Servicekanal zugegriffen werden. Die Vorgabewerte müssen in der globalen Datenstruktur ObjectWrData eingetragen sein. Der funktionale Ablauf des Zugriffs ist im Dokument „Antriebssystem SD2 – SERVOLINK 4-Bussystembeschreibung“ beschrieben.

### 4.2.7.2 Aufruf des Funktionsbausteins

```
#FB_WriteObject(DrNo := "DB_GLOBALDATA".DriveNumber);
```

### 4.2.7.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer

## 4.2.8 Funktionsbaustein FB\_SetArrayIndex

### 4.2.8.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_SetArrayIndex wird der Subindex eines Array-Objektes gesetzt. Danach wird beim Schreiben des Objektes der Subindex automatisch erhöht. Der funktionale Ablauf des Zugriffs ist im Handbuch „Antriebssystem SD2 – SERVOLINK 4-Bussystembeschreibung“ beschrieben.

Um den Index auf das erste Element zu setzen, muss der Aufrufparameter den Wert 4 erhalten. Der Wert muss in der globalen Datenstruktur ObjectAiData eingetragen sein.

### 4.2.8.2 Aufruf des Funktionsbausteins

```
#FB_SetArrayIndex (DrNo := "DB_GLOBALDATA".DriveNumber);
```

### 4.2.8.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
Drive	In	INT	Antriebsnummer

## 4.2.9 Funktionsbaustein FB\_SC\_Handle

### 4.2.9.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_SC\_Handle werden die Befehle mittels der Funktionsbausteine FB\_SC\_FillObject und FB\_SC\_HandleObject in die SC-Funktionen des Antriebs geschrieben. Durch den wiederholten Aufruf des Funktionsbausteins können schrittweise nacheinander Befehle abgearbeitet werden.



### 4.2.9.2 Aufruf des Funktionsbausteins

```
// handle communication for safety control 0, setting speed zero
"DB GLOBALDATA".SC HandleData["DB GLOBALDATA".DriveNumber].Execute := TRUE;
"FB SC Handle DB"(DrNo:= "DB GLOBALDATA".DriveNumber,
    SC:="DB SC PROCESSDATA".SC0,
    SubIndex:="DB SC PROCESSDATA".PAR SPEED ZERO,
    RdWr:="DB SC PROCESSDATA".WRITE PAR,
    Pattern:=#bPattern,
    Par:="DB_GLOBALDATA".SD_Drive["DB_GLOBALDATA".DriveNumber].Setpoints.
SpeedZero);
IF "DB GLOBALDATA".SC HandleData["DB GLOBALDATA".DriveNumber].Done THEN
    // reset function block
    "DB_GLOBALDATA".SC_HandleData["DB_GLOBALDATA".DriveNumber].Execute :=
FALSE;
    "FB SC Handle DB"();
    "DB GLOBALDATA".MainState["DB GLOBALDATA".DriveNumber] := 2;
    "DB GLOBALDATA".Pattern["DB GLOBALDATA".DriveNumber] :=
    "DB GLOBALDATA".Pattern["DB GLOBALDATA".DriveNumber] + 1;
    "IEC Timer 0 DB".TON(IN := FALSE, PT := T#0s);
END_IF;
```

### 4.2.9.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer
SC	In	BYTE	Nummer der Safety Control-Funktion
SubIndex	In	BYTE	Befehlsnummer
RdWr	In	BYTE	Lese- oder Schreibbefehl
Pattern	In	BYTE	Prüfmuster
Par	In	DWORD	Vorgabewert

### 4.2.9.4 Ablauf des Funktionsbausteins

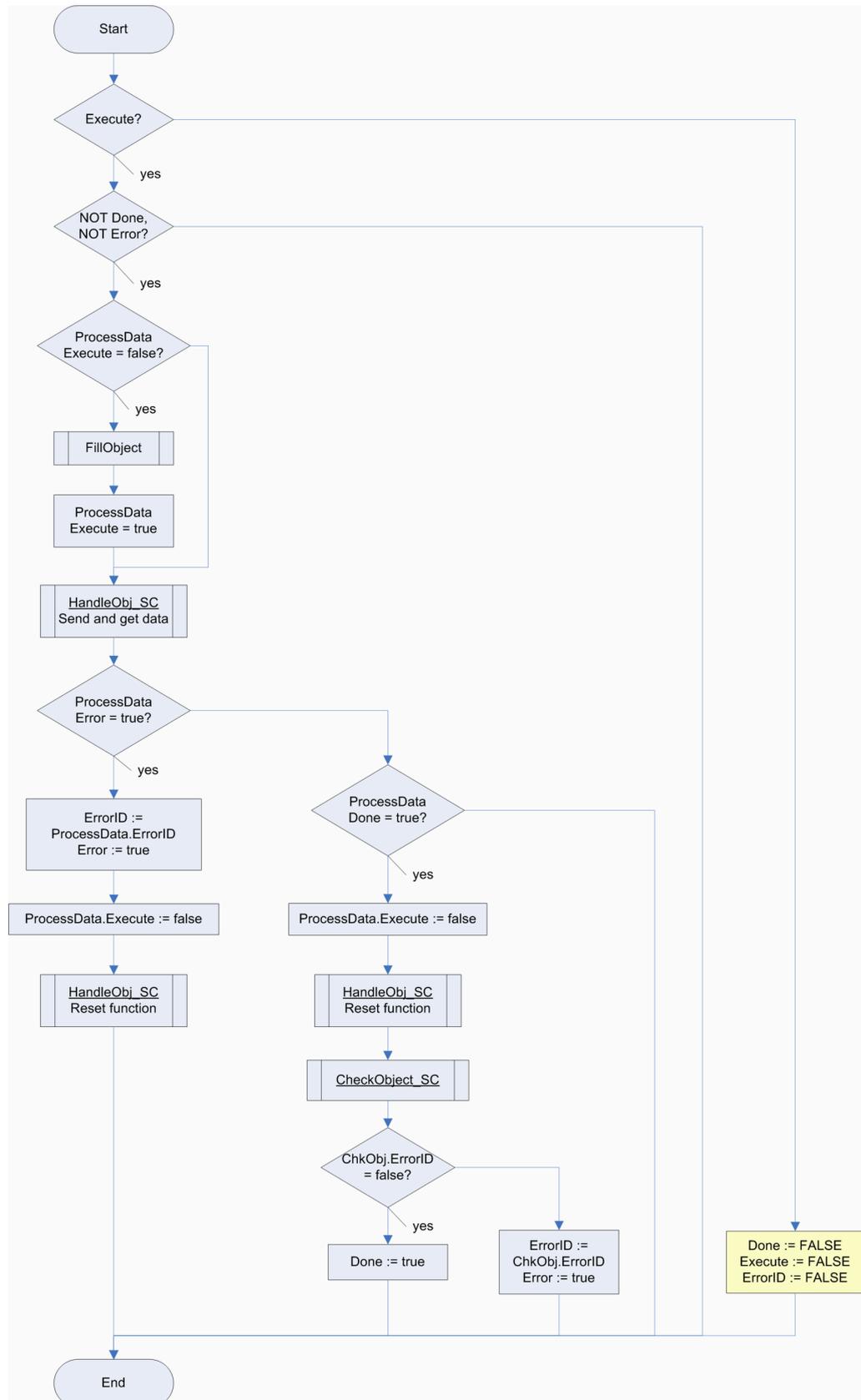


Abb. 4: FB\_SC\_Handle

## 4.2.10 Funktionsbaustein FB\_SC\_FillObject

### 4.2.10.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_SC\_FillObject werden die Eingangsparameter des Bausteins der Ausgangsstruktur zum Senden zugeordnet. Zusätzlich wird die Bildung der Checksumme aufgerufen und das Ergebnis in die Struktur eingetragen.

#### *Hinweis*

Weitere Informationen zu den Parametern finden Sie im Handbuch „Antriebssystem SD2 – Sicherheitsfunktionen SFM / SLOF“ (Kapitel „Feldbuskommunikation“).

### 4.2.10.2 Aufruf des Funktionsbausteins

```
#FB SC FillObject (DrNo := #DrNo,
    SC := #SC,
    SubIndex := #SubIndex,
    RdWr := #RdWr,
    Data := #Par,
    Pattern := #Pattern);
```

### 4.2.10.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer
SC	In	BYTE	Nummer der Safety Control-Funktion
SubIndex	In	BYTE	Befehlsnummer
RdWr	In	BYTE	Lese- oder Schreibfunktion
Data	In	DWORD	Vorgabewert
Pattern	In	BYTE	Prüfmuster

## 4.2.11 Funktionsbaustein FB\_SC\_HandleObject

### 4.2.11.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_SC\_HandleObject werden die Safety-Parameter über die Objekte 467 bis 468 in die Safety Control-Funktionen des Antriebs geschrieben und über die Objekte 469 bis 470 ausgelesen.

- ▶ Objekt 467: SAFE\_SPEED\_MONITOR\_DEMAND\_DATA\_OBJECT
- ▶ Objekt 468: SAFE\_SPEED\_MONITOR\_DEMAND\_CTRL\_OBJECT
- ▶ Objekt 469: SAFE\_SPEED\_MONITOR\_REPLY\_DATA\_OBJECT
- ▶ Objekt 470: SAFE\_SPEED\_MONITOR\_REPLY\_STATUS\_OBJECT

Dabei werden nacheinander folgende Befehle gesendet:

- ▶ Objekt schreiben: 4 Bytes (Sollwert) in Objekt 467 (über FB\_WriteObject)
- ▶ Objekt schreiben: 4 Bytes (Steuerdaten) in Objekt 468 (über FB\_WriteObject)
- ▶ Objekt lesen: 4 Bytes (Istwert) aus Objekt 469 (über FB\_ReadObject)
- ▶ Objekt lesen: 4 Bytes (Statusdaten) aus Objekt 470 (über FB\_ReadObject)

Mit den beiden Schreibbefehlen wird das Aufforderungstelegramm an die entsprechende Safety Control-Funktion des Antriebs gesendet. Die Daten hierfür stehen in den globalen Prozessdaten Out\_SC\_Obj (TS\_SC\_PROCESSDATA). Mit den beiden Lesebefehlen wird das Antworttelegramm der entsprechenden Safety Control-Funktion gelesen. Die gelesenen Daten werden in die globalen Prozessdaten In\_SC\_Obj (TS\_SC\_PROCESSDATA) eingetragen und können dann ausgewertet werden.

### 4.2.11.2 Aufruf des Funktionsbausteins

```
#FB_SC_HandleObject (DrNo := "DB_GLOBALDATA".DriveNumber);
```

### 4.2.11.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer

### 4.2.11.4 Ablauf des Funktionsbausteins

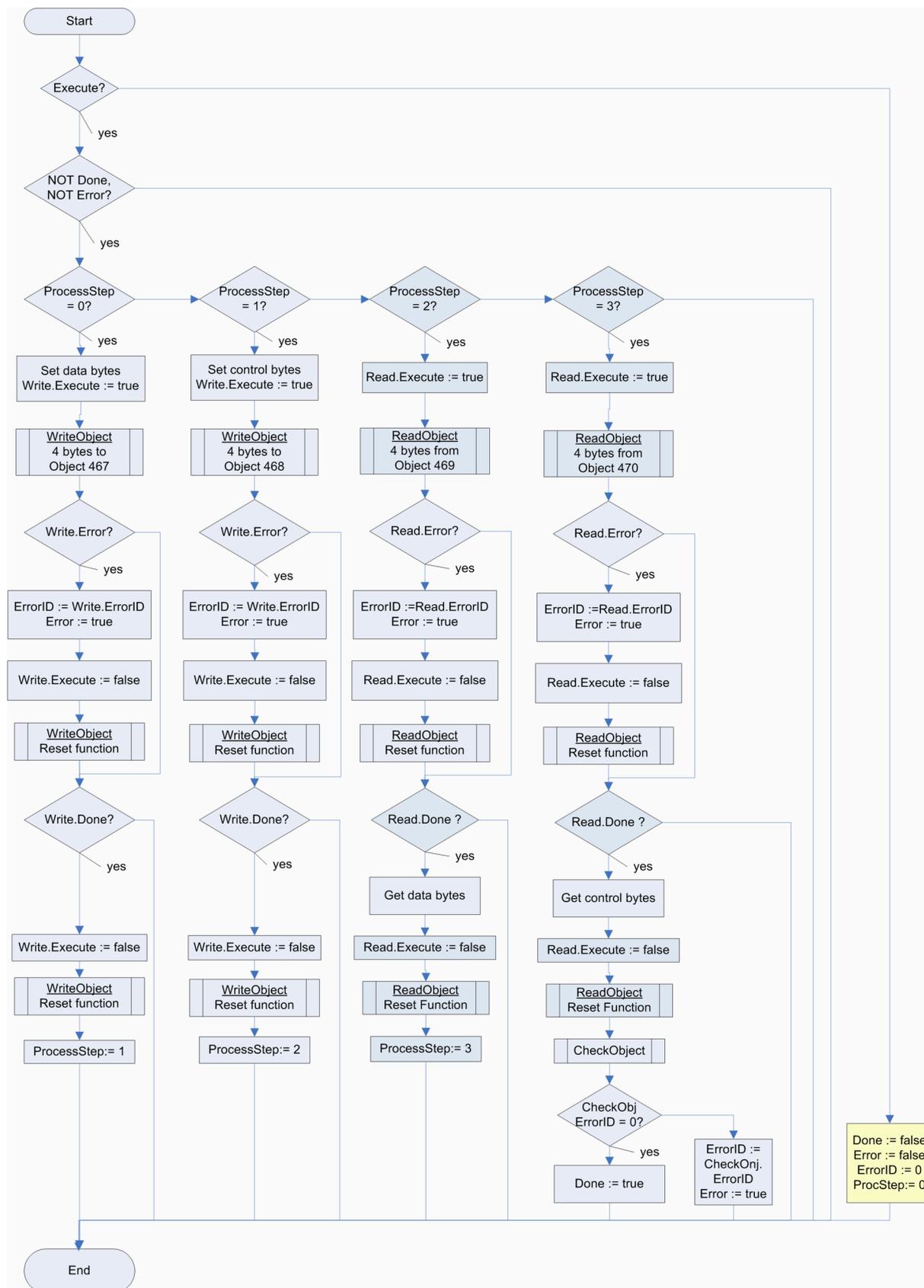


Abb. 5: FB\_SC\_HandleObject

## 4.2.12 Funktionsbaustein FB\_SC\_CheckObject

### 4.2.12.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_SC\_CheckObject werden die Adressen, die Prüfmuster und die Checksummen der Eingangs- und Ausgangsdaten des Telegrammaustauschs geprüft.

Hierzu wird die Checksumme über die Eingangsdaten gebildet und mit der übertragenen Checksumme verglichen. Tritt ein Fehler auf, wird der entsprechende Fehlercode zurückgegeben. Danach wird die Adresse geprüft. Die empfangene Adresse muss gleich der gesendeten Adresse sein. Bei Ungleichheit wird der entsprechende Fehlercode zurückgegeben. Zuletzt werden die Prüfmuster miteinander verglichen. Dazu wird das Ausgangsmuster um 1 erhöht und mit dem Eingangsmuster verglichen. Tritt ein Fehler auf, wird der entsprechende Fehlercode zurückgegeben.

### 4.2.12.2 Aufruf des Funktionsbausteins

```
#FB_SC_CheckObject (DrNo := "DB_GLOBALDATA".DriveNumber);
```

### 4.2.12.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
DrNo	In	INT	Antriebsnummer
ErrorID	Out	WORD	Fehlercode

## 4.2.13 Funktionsbaustein FB\_SwapINT

### 4.2.13.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_SwapINT wird die Bytereihenfolge eines 2-Byte-Integerwerts getauscht. Dieser Tausch wird benötigt, um die Formate Big-Endian und Little-Endian der Geräte zu berücksichtigen.

Bytereihenfolge vorher: Byte1, Byte2

Bytereihenfolge nachher: Byte2, Byte1

### 4.2.13.2 Aufruf des Funktionsbausteins

```
#FB_SwapINT (value := "DB_GLOBALDATA".SD_Drive[#DrNo].OutData.ServiceIndex);
```

### 4.2.13.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
value	In/Out	INT	Integerwert



## 4.2.14 Funktionsbaustein FB\_SwapDINT

### 4.2.14.1 Allgemeine Beschreibung

Mit dem Funktionsbaustein FB\_SwapDINT wird die Bytereihenfolge eines 4-Byte-Integerwerts getauscht. Dieser Tausch wird benötigt, um die Formate Big-Endian und Little-Endian der Geräte zu berücksichtigen.

Bytereihenfolge vorher: Byte1, Byte2, Byte3, Byte4

Bytereihenfolge nachher: Byte4, Byte3, Byte2, Byte1

### 4.2.14.2 Aufruf des Funktionsbausteins

```
#FB_SwapDINT (value := "DB_GLOBALDATA".SD_Drive[#DrNo].OutData.ServiceValue);
```

### 4.2.14.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
value	In/Out	DINT	Doppelintegerwert

## 4.3 Funktionen

### 4.3.1 Funktion FC\_GetStatus

#### 4.3.1.1 Allgemeine Beschreibung

Die Funktion FC\_GetStatus ermittelt aus dem Statuswort des Antriebs den Gerätezustand gemäß der Gerätezustandsmaschine (siehe Handbuch „Antriebssystem SD2 – Gerätesteuerung“). Dazu wird das Statuswort der Funktion übergeben. Die Funktion liefert als Rückgabewert einen Aufzählungswert von DB\_DRIVESTATE.

#### 4.3.1.2 Aufruf des Funktionsbausteins

```
"DB_GLOBALDATA".SD_Drive[#DrNo].ActValues.Status :=
    "FC_GetStatus"(Status :=
        "DB_GLOBALDATA".SD_Drive[#DrNo].InData.StatusWord);
```

#### 4.3.1.3 Ein- und Ausgänge

Parameter	In / Out	Typ	Beschreibung
Status	In	TS_DRIVE_STATUSWORD	Statuswort
ReturnValue	Return	WORD	Gerätezustand

## 4.4 Globale Datenbausteine

### 4.4.1 Datenbaustein DB\_DRIVESTATE

Der globale Datenbaustein DB\_DRIVESTATE enthält eine Aufzählung der Antriebsstatus gemäß dem Antriebsprotokoll DS402.

Zustand	Wert	Bedeutung
STATE_Start	0	Start, Initialisierung
STATE_NotReadyToSwitchOn	1	nicht einschaltbereit
STATE_SwitchOnDisabled	2	Einschaltsperr
STATE_ReadyToSwitchOn	3	einschaltbereit
STATE_SwitchedOn	4	eingeschaltet
STATE_OperationEnabled	5	Betrieb freigegeben
STATE_QuickStopActive	6	Schnellhalt
STATE_FaultReactActive	7	Störungsreaktion
STATE_Fault	8	Störung

### 4.4.2 Datenbaustein DB\_ERRORCODE

Der globale Datenbaustein DB\_ERRORCODE enthält die Fehlercodes, die von den Funktionsbausteinen zurückgeliefert werden.

Die Fehlercodes haben folgende Bedeutung:

Fehlercode	Wert	Bedeutung
ERR_None	0	kein Fehler
ERR_WrongState	1	Für den anliegenden Befehl befindet sich der Antrieb im falschen Gerätestatus.
ERR_Parameter	2	Parameterfehler
ERR_Pattern	3	falsches Prüfmuster
ERR_Checksum	4	falsche Checksumme
ERR_Address	5	falsche Adresse
ERR_SERVOLINK_Fault	6	SERVOLINK-Fehler
ERR_SERVOLINK_Slot	7	SERVOLINK-Slot-Fehler
ERR_TimeOut	8	Zeitüberschreitung

#### *Hinweis*

In den Fehlervariablen können weitere Fehlercodes stehen. Diese ergeben sich aus den Return-Codes der Bausteine des Servicekanals und sind im Handbuch „Antriebssystem SD2 – SERVOLINK 4-Bussystembeschreibung“ aufgelistet.

### 4.4.3 Datenbaustein DB\_GLOBALDATA

Der globale Datenbaustein DB\_GLOBALDATA enthält die programmrelevanten Daten, die an das jeweilige Projekt angepasst werden müssen.

Variable	Typ	Bedeutung
SD_Drive	ARRAY[0..15] OF "TS_DRIVE_DATA"	Struktur der Antriebsdaten
ObjectRdData	ARRAY[0..15] OF "TS_OBJECTDATA"	Ein-/Ausgangsdaten vom Funktionsbaustein
ObjectWrData	ARRAY[0..15] OF "TS_OBJECTDATA"	Ein-/Ausgangsdaten vom Funktionsbaustein



Variable	Typ	Bedeutung
ObjectAiData	ARRAY[0..15] OF "TS_OBJECTDATA"	Ein-/Ausgangsdaten vom Funktionsbaustein
ReadValue	ARRAY[0..15] OF DINT	gelesener Rückgabewert
ProcessData_SC	ARRAY[0..15] OF "TS_SC_PROCESSDATA"	Prozessdaten der Safety-Objekte
HandleData_SC	ARRAY[0..15] OF "TS_HANDLE_DATA"	Steuerungsparameter für den Funktionsbaustein
MainState	ARRAY[0..15] OF INT	Schrittzeiger im Main-Programm
ErrorID	ARRAY[0..15] OF WORD	Globaler Fehlercode
StandStill	ARRAY[0..15] OF BOOL	Status: Drehzahl Null
SpeedLimit	ARRAY[0..15] OF BOOL	Status: Drehfeldbegrenzung
Pattern	ARRAY[0..15] OF INT	Pattern-Feld (wird nach jeder Übertragung eines Befehls erhöht)
DriveNumber	INT	Antriebsnummer
ChkDataIn	ARRAY[0..7] OF BYTE	Array der Eingangsdaten zur Bildung der Checksumme
tChecksumIn	BYTE	Checksumme der Eingangsdaten
ChkDataOut	ARRAY[0..7] OF BYTE	Array der Ausgangsdaten zur Bildung der Checksumme
tChecksumOut	BYTE	Checksumme der Ausgangsdaten
oldstate	ARRAY[0..3] OF BYTE	alter Statuswert
ErrCounter	ARRAY[0..3] OF INT	Fehlerzähler
testNo	ARRAY[0..9] OF INT	Debugvariable
testIn	ARRAY[0..9] OF BYTE	Debugvariable
testOut	ARRAY[0..9] OF BYTE	Debugvariable
testi	INT	Debugvariable

#### 4.4.4 Datenbaustein DB\_SC\_PROCESSDATA

Der globale Datenbaustein DB\_SC\_PROCESSDATA enthält Konstanten für das Datenarray des Safety-Telegramms.

Parameter	Wert	Bedeutung
SC0	0x00	Adresswert für die Safety Control-Funktion 0
SC1	0x80	Adresswert für die Safety Control-Funktion 1
PAR_REPEAT	0	Subindex für „Wiederholung“
PAR_SPEED_ZERO	1	Subindex für „Drehzahl Null“
PAR_SPEED_LIMIT	2	Subindex für „sicher begrenztes Drehfeld“
PAR_STATUS	3	Subindex für „Status“
PAR_VERSION	4	Subindex für „Version“
PAR_COBID	5	Subindex für „COB-ID“
PAR_ACTIV_LIMIT	6	Subindex für „activate limit“
PAR_VERSIONIFO	7	Subindex für „version info“
READ_PAR	0x00	Kennung für die Funktion „Lesen“
WRITE_PAR	0x80	Kennung für die Funktion „Schreiben“

## 4.5 SPS-Variablen

### 4.5.1 Standardvariablen

Die Standardvariablen-tabelle enthält die Variablen, die auf Ein- oder Ausgänge des SPS-Systems gelegt sind.

Name	Datentyp	Adresse	Bedeutung
I_SwOn_0	BOOL	%I10.0	Eingang „Regler ein“
I_EnOp_0	BOOL	%I10.1	Eingang „Betrieb freigeben“
I_QStop_0	BOOL	%I10.2	Eingang „Schnellhalt“
I_ResFault_0	BOOL	%I10.3	Eingang „Fehlerreset“
I_ResFault_SC	BOOL	%I10.4	Eingang „Fehlerreset Safety“
I_Speed_0	BYTE	%IB11	Eingänge „Geschwindigkeitsvorgabe“
InData_0	TS_DRIVE_DATA_IN	%I124.0	Feldbuseingangsdaten Antrieb 0
InData_1	TS_DRIVE_DATA_IN	%I140.0	Feldbuseingangsdaten Antrieb 1
OutData_0	TS_DRIVE_DATA_OUT	%Q126.0	Feldbusausgangsdaten Antrieb 0
OutData_1	TS_DRIVE_DATA_OUT	%Q142.0	Feldbusausgangsdaten Antrieb 1

## 4.6 Anwenderdefinierte Datentypen

Die anwenderdefinierten Datentypen beschreiben den Aufbau der Datenstrukturen für die Antriebe. Hierzu gehören die Eingangs- und Ausgangswerte der Feldbusdaten und die aufbereiteten Soll- und Istwerte.

Auf die Daten kann über die globalen Variablen zugegriffen werden.

Die Struktur von TS\_DRIVE\_DATA ist an die Vorgabe der PLCopen-Gruppe angelehnt. Die Strukturen der Feldbusparameter sind an das CANopen-Protokoll angelehnt. Eine Beschreibung des Telegrammaufbaus ist den Handbüchern „Antriebssystem SD2–CAN-Bus-Anbindung“ und „Antriebssystem SD2–Gerätesteuerung“ zu finden.

### 4.6.1 Anwenderdefinierter Datentyp TS\_DRIVE\_STATUSWORD

Der anwenderdefinierte Datentyp TS\_DRIVE\_STATUSWORD enthält die Datenstruktur des Statusworts eines Antriebs.

Name	Typ	Bedeutung
<b>Statuswort Byte 0..1</b>		
ReadyToSwitchOn	BOOL	einschaltbereit
SwitchedOn	BOOL	eingeschaltet
OperationEnabled	BOOL	Betrieb freigegeben
Fault	BOOL	Störung
VoltageEnabled	BOOL	Spannung freigegeben
QuickStop	BOOL	Schnellhalt
SwitchOnDisabled	BOOL	Einschaltsperr
Warning	BOOL	Warnung
smres010	BOOL	reserviert
Remote	BOOL	Remote Mode
TargetReached	BOOL	Sollwert erreicht
InternalLimitActive	BOOL	interner Grenzwert erreicht
OperationModeSpecific0	BOOL	betriebsartenabhängig

Name	Typ	Bedeutung
OperationModeSpecific1	BOOL	betriebsartenabhängig
IstwertTelegrammKennung1	BOOL	Istwerttelegramm-Kennung, wird vom Antrieb auf 1 gesetzt
IstwertTelegrammKennung0	BOOL	Istwerttelegramm-Kennung, wird vom Antrieb auf 0 gesetzt

## 4.6.2 Anwenderdefinierter Datentyp TS\_DRIVE\_DATA\_IN

Der anwenderdefinierte Datentyp TS\_DRIVE\_DATA\_IN enthält die Datenstruktur der Empfangsdaten eines Antriebs vom Feldbussystem.

Name	Typ	Bedeutung
<b>Statuswort Byte 0..1</b>		
StatusWord	TS_DRIVE_STATUSWORD	aktuelles Statuswort
<b>Istwerte Byte 2..9</b>		
Act_Position	DINT	aktuelle Position
Act_Velocity	INT	aktuelle Geschwindigkeit
Act_Current	INT	aktueller Strom
<b>Service-Kanal Byte 10..15</b>		
ServiceReturn	DINT	Rückgabeparameter / Fehlercode
res14	BYTE	reserviert
ServiceDoneToggle	BOOL	Rückgabe Toggle Bit
ServiceFault	BOOL	Fehler
smres152	BOOL	reserviert
smres153	BOOL	reserviert
smres154	BOOL	reserviert
smres155	BOOL	reserviert
smres156	BOOL	reserviert
smres157	BOOL	reserviert

## 4.6.3 Anwenderdefinierter Datentyp TS\_DRIVE\_DATA\_OUT

Der anwenderdefinierte Datentyp TS\_DRIVE\_DATA\_OUT enthält die Datenstruktur der Sendedaten eines Antriebs zum Feldbussystem.

Name	Typ	Bedeutung
<b>Steuerwort Byte 0..1</b>		
SwitchOn	BOOL	Leistungsteil einschalten
EnableVoltage	BOOL	Spannung am Leistungsteil freigeben
QuickStop	BOOL	Schnellhalt
EnableOperation	BOOL	Betrieb freigeben
Mode0	BOOL	Operation mode Bit 0
Mode1	BOOL	Operation mode Bit 1
Mode2	BOOL	Operation mode Bit 2
FaultReset	BOOL	Fehlerreset
Halt	BOOL	Halt
res011	BOOL	reserviert
res012	BOOL	reserviert
smres013	BOOL	reserviert
smres014	BOOL	reserviert

Name	Typ	Bedeutung
smres015	BOOL	reserviert
SollwertTelegrammID0	BOOL	Sollwerttelegramm-Kennung, muss von der SPS auf 0 gesetzt werden
SollwertTelegrammID1	BOOL	Sollwerttelegramm-Kennung, muss von der SPS auf 1 gesetzt werden
<b>Sollwerte Byte 2..7</b>		
res02	INT	reserviert
Spt_Velocity	INT	Sollgeschwindigkeit
Spt_Current	INT	maximaler Sollstrom
<b>Byte 8</b>		
res08	BYTE	reserviert
<b>Servicekanal Byte 9..15</b>		
ServiceValidToggle	BOOL	neuer Service Befehl
ServiceFunction0	BOOL	Service-Funktion
ServiceFunction1	BOOL	
ServiceByteIndex0	BOOL	Anzahl Bytes
ServiceByteIndex1	BOOL	
smres095 bis smres097	3 * BOOL	reserviert
ServiceIndex	INT	Objektnummer
ServiceValue	DINT	Objektwert / Arrayindex

#### 4.6.4 Anwenderdefinierter Datentyp TS\_DRIVE\_DATA

Der anwenderdefinierte Datentyp TS\_DRIVE\_DATA enthält die Datenstruktur der Antriebsdaten basierend auf dem PLCopen-Standard.

Name	Typ	Bedeutung
init_ok	BOOL	Initialisierungsstatus
ActValues	TS_DRIVE_ACTUALVALUES	Istwerte des Antriebs
Setpoints	TS_DRIVE_SETPOINTS	Sollwerte für den Antrieb
InData	TS_DRIVE_DATA_IN	Datenblock (Bussystem->SPS)
OutData	TS_DRIVE_DATA_OUT	Datenblock (SPS->Bussystem)

#### 4.6.5 Anwenderdefinierter Datentyp TS\_DRIVE\_SETPOINTS

Der anwenderdefinierte Datentyp TS\_DRIVE\_SETPOINTS enthält die Datenstruktur der Sollwerte für den Antrieb.

Name	Typ	Bedeutung
FaultReset	BOOL	TRUE = Fehler zurücksetzen
SwitchOn	BOOL	TRUE = Endstufe einschalten
EnableOperation	BOOL	TRUE = Betrieb freigeben
EnableVoltage	BOOL	TRUE = Spannung freischalten
QuickStop	BOOL	FALSE = Schnellhalt auslösen
Velocity	INT	Sollwert Geschwindigkeit, max. 0x3FFF
Current	INT	Sollwert Strom, max. 0x3FFF
SpeedZero	DWORD	Sollwert Drehzahl Null [Hz]
SpeedLimit	DWORD	Sollwert Drehfeldbegrenzung [Hz]



### 4.6.6 Anwenderdefinierter Datentyp TS\_DRIVE\_ACTUALVALUES

Der anwenderdefinierte Datentyp TS\_DRIVE\_ACTUALVALUES enthält die Datenstruktur der aufbereiteten Istwerte des Antriebs.

Name	Typ	Bedeutung
Status	WORD	Gerätestatus des Antriebs
StatusWord	TS_DRIVE_STATUSWORD	Statuswort des Antriebs
Act_Position	DINT	aktuelle Position
Act_Velocity	INT	aktuelle Geschwindigkeit
Act_Current	INT	aktueller Strom
StatusSC0	DWORD	Status der Safety Control-Funktion SC0
StatusSC1	DWORD	Status der Safety Control-Funktion SC1
SpeedLimitSC0	DWORD	aktueller Vorgabewert „Drehzahlbegrenzung“ in SC0
SpeedLimitSC1	DWORD	aktueller Vorgabewert „Drehzahlbegrenzung“ in SC1

### 4.6.7 Anwenderdefinierter Datentyp TS\_OBJECTDATA\_SC

Der anwenderdefinierte Datentyp TS\_OBJECTDATA\_SC beschreibt die Struktur der Sende- und Empfangsdaten der Safety-Objekte.

Name	Typ	Bedeutung
Data	DWORD	Wert für die Funktionalität
Address	BYTE	Antriebsnummer mit Nummer der Safety Control-Funktion
Index	BYTE	Subindex für die Funktionalität mit Lese- oder Schreibfunktion
Pattern	BYTE	Prüfmuster
Checksum	BYTE	Checksumme
Value0	DINT	Rückgabewert vom Objektzugriff
Value1	DINT	Rückgabewert vom Objektzugriff

### 4.6.8 Anwenderdefinierter Datentyp TS\_SC\_PROCESSDATA

Der anwenderdefinierte Datentyp TS\_SC\_PROCESSDATA beschreibt die Struktur zur Steuerung der Zugriffe auf die Safety-Objekte.

Name	Typ	Bedeutung
Handle	TS_HANDLE_DATA	Struktur zur Steuerung des Funktionsbausteins
In_SC_Obj	TS_OBJECTDATA_SC	von SC erhaltene Objektdaten
Out_SC_Obj	TS_OBJECTDATA_SC	zur SC zu sendende Objektdaten

### 4.6.9 Anwenderdefinierter Datentyp TS\_OBJECTDATA

Der anwenderdefinierte Datentyp TS\_OBJECTDATA beschreibt die Struktur der Ein- und Ausgangsdaten der Funktionsbausteine FB\_WriteObject, FB\_ReadObject und FB\_SetArrayIndex.

Name	Typ	Bedeutung
Handle	TS_HANDLE_DATA	Struktur zur Steuerung des Funktionsbausteins
ObjectIndex	INT	Objekt-Index
ObjectSubindex	DINT	Objekt-SubIndex
ObjectValue	DINT	Vorgabewert für das Objekt
ObjectNoOfBytes	INT	Anzahl der gültigen Bytes (1..4)
ArrayIndex	DINT	Vorgabewert des Arrayindex, beginnend bei 0 oder 4
ReturnValue	DINT	Fehlercode (2 Byte) oder Rückgabewert (4 Byte)

## 4.6.10 Anwenderdefinierter Datentyp TS\_HANDLE\_DATA

Der anwenderdefinierte Datentyp TS\_HANDLE\_DATA beschreibt die Struktur zur Steuerung des Ablaufs eines Funktionsbausteins.

Name	Typ	Bedeutung
Execute	BOOL	TRUE = Freigabe der Funktionsdurchführung FALSE = Zurücksetzen des Funktionsablaufs
Busy	BOOL	TRUE = Funktionsdurchführung ist aktiv
Done	BOOL	TRUE = Funktion erfolgreich durchgeführt
Error	BOOL	TRUE = Funktion mit Fehler beendet
ErrorID	INT	Fehlercode
Step	INT	Nummer zum Ablauf

## 4.7 Checksumme bilden

Für die Überprüfung der Daten der Safety-Objekte wird eine Checksumme gebildet. Dazu werden die einzelnen Elemente der Struktur in ein Byte-Array geschrieben. Über dieses Byte-Array werden dann die Daten byteweise addiert und von dem Wert 0xFF subtrahiert.

$$Check = 0xFF - \sum_{i=0}^6 Byte\ i$$

### Beispiel: Checksummenbildung

Fehler zurücksetzen im SC1 für Antrieb B, Pattern = AA<sub>h</sub>

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Summe	Checksumme = 0xFF - Summe
0x01	0x00	0x00	0x00	0x81	0x83	0xAA	0xAF	0xFF - 0xAF = <b>0x50</b>

Für die Daten zum SD2x wird die Checksumme im Funktionsbaustein FB\_SC\_FillObject überprüft. Für die Daten vom SD2x wird die Checksumme im Funktionsbaustein FB\_SC\_CheckObject überprüft.

## 4.8 Systemfunktionen und Systemfunktionsbausteine

Der Systemfunktionsbaustein TON wird für die Zeitüberwachung genutzt.



Die Bausteine der TIA-Systemfunktionen sind in den entsprechenden Hilfen von Siemens beschrieben.

# 5 Hardwarekonfiguration

Das Beispielprojekt enthält folgende Hardwarekonfiguration:

- ▶ SPS S7-1500
- ▶ PROFINET IO-Controller – Gateway – SD2S
- ▶ PROFIBUS DP-Master – Gateway – SD2S

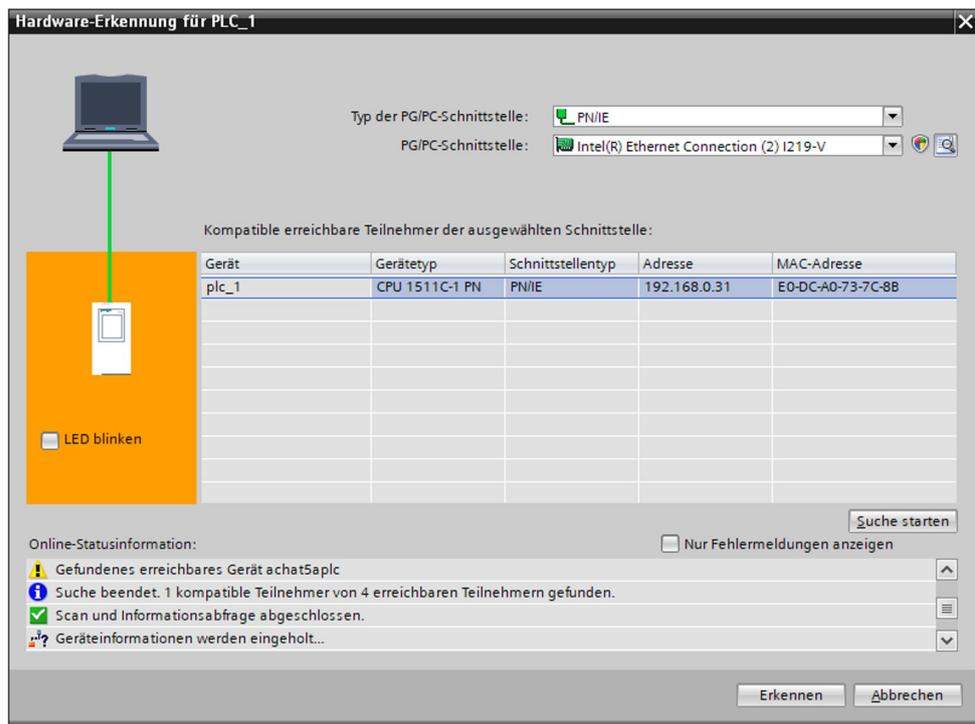
Sollten Sie eine andere Hardwarekonfiguration benötigen, gehen Sie im TIA Portal wie folgt vor:

1. Wählen Sie unter den Punkten „Geräte & Netze“ und „Neues Gerät hinzufügen“ die CPU aus. (Beispiel: Controller → Simatic S7-1500 → CPU → Nicht spezifizierte CPU 1500 → 6ES7 5XX-XXXXX-XXXX → Hinzufügen)
2. Zum Einbinden der CPU, der Gerätedaten und der Gateways gehen Sie vor wie in den folgenden Kapiteln beschrieben.

## 5.1 Einbinden der CPU

Die nicht spezifizierte CPU wird eingeblendet.

1. Klicken Sie auf „Ermitteln“, um das folgende Suchfenster zur Ermittlung der Konfiguration des angeschlossenen Geräts zu öffnen.

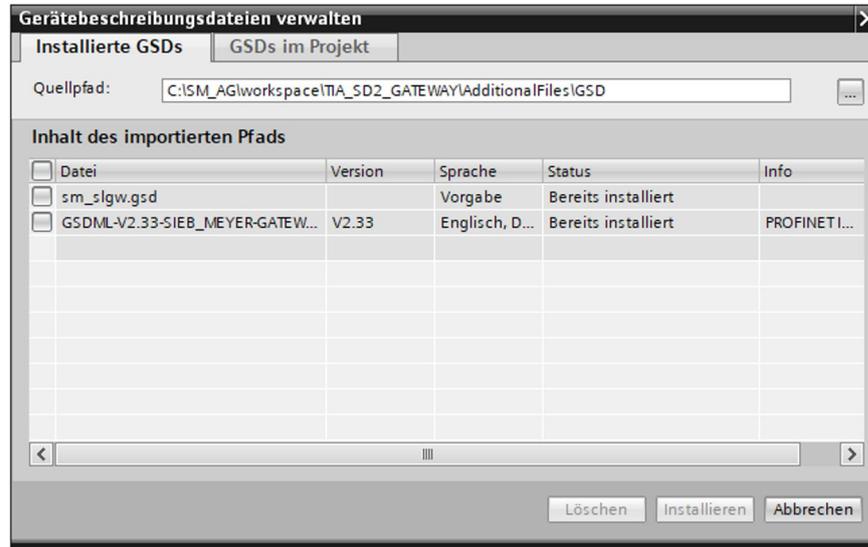


2. Stellen Sie über die Auswahllisten oben die Schnittstelle ein und klicken Sie auf „Suche starten“. Es werden alle erreichbaren Teilnehmer angezeigt.
  3. Wählen Sie die gewünschte CPU in der Geräteliste aus und klicken Sie auf „Erkennen“.
- ✓ Die CPU wird nun mit ihren Komponenten in die Geräteansicht eingeblendet.

## 5.2 Einbinden der Gerätedateien

Die aktuellen GSD-Dateien können Sie im Downloadbereich der SIEB & MEYER-Website herunterladen. Zum Einbinden der GSD-Dateien gehen Sie wie folgt vor:

1. Kopieren Sie die ZIP-Dateien in ein eigenes Verzeichnis und entpacken Sie die Dateien.
2. Wählen Sie im Menü „Extras“ den Punkt „Gerätebeschreibungsdateien (GSD) verwalten“. Das folgende Auswahlfenster wird geöffnet.



3. Wählen Sie oben den Quellpfad, in dem sich Ihre GSD-Dateien befinden.
  4. Wählen Sie die gewünschten Dateien über die Kontrollkästchen links aus und klicken Sie auf „Installieren“.
- ✓ Die GSD-Dateien sind nun im Projekt eingebunden.

## 5.3 Einbinden eines PROFINET IO-Gateways 0362157(A)

Zur Einbindung eines PROFINET IO-Gateways 0362157(A) gehen Sie wie folgt vor:

### Hinweis

Im TIA Portal V15.1 kann das Gateway nicht über das Menü „Online → Hardware-Erkennung → PROFINET-Device aus Netzwerk...“ hinzugefügt werden. Es erscheint die folgende Fehlermeldung:

*Das IO-Device „device name“ konnte nicht dem Projekt hinzugefügt werden, da die Artikelnummer des Online-Geräts nicht identifiziert werden konnte.*

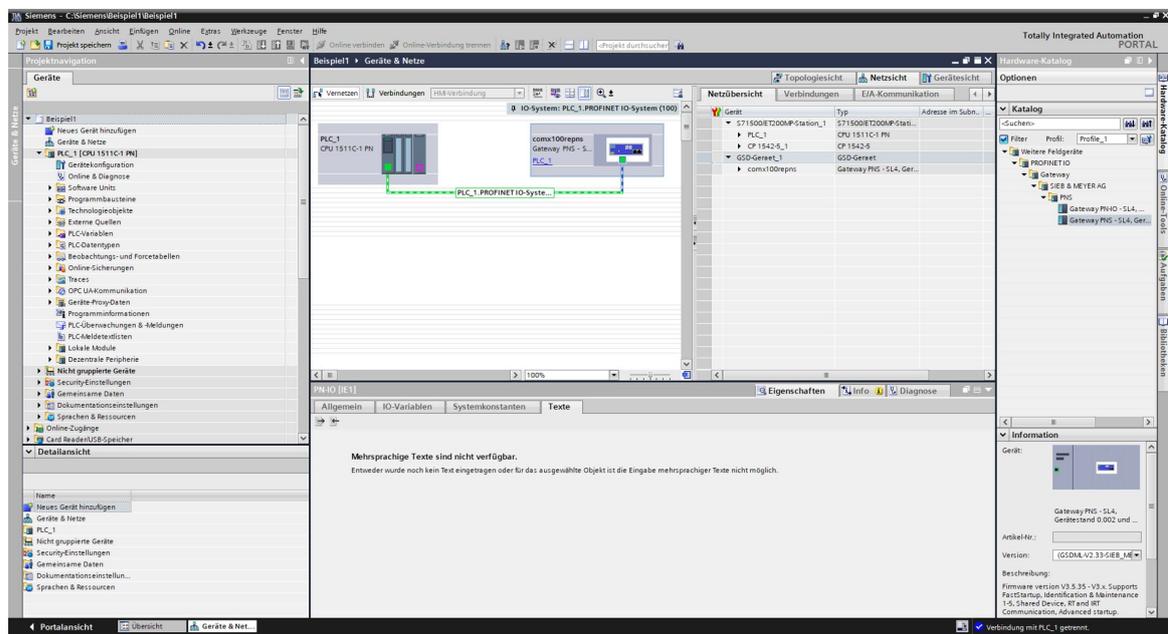
Da das Gateway kein Siemens-Produkt ist und somit keine Siemens-Artikelnummer enthält, kann das TIA Portal das Gerät nicht identifizieren und einfügen. Diese Komfortfunktion ist nur mit Siemens-Geräten möglich. Alle anderen Geräte müssen manuell über den Hardware-Katalog eingebunden werden.

1. Wechseln Sie in die Netzansicht.
2. Suchen Sie im Hardware-Katalog das passende Gateway aus: Weitere Feldgeräte → PROFINET IO → Gateway → SIEB & MEYER AG → PNS → Gateway PNS-SL4, Gerätestand 0.002 oder neuer.

3. Wählen Sie das Gerät aus und ziehen Sie es in das Netzwerk-Fenster.
4. 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:

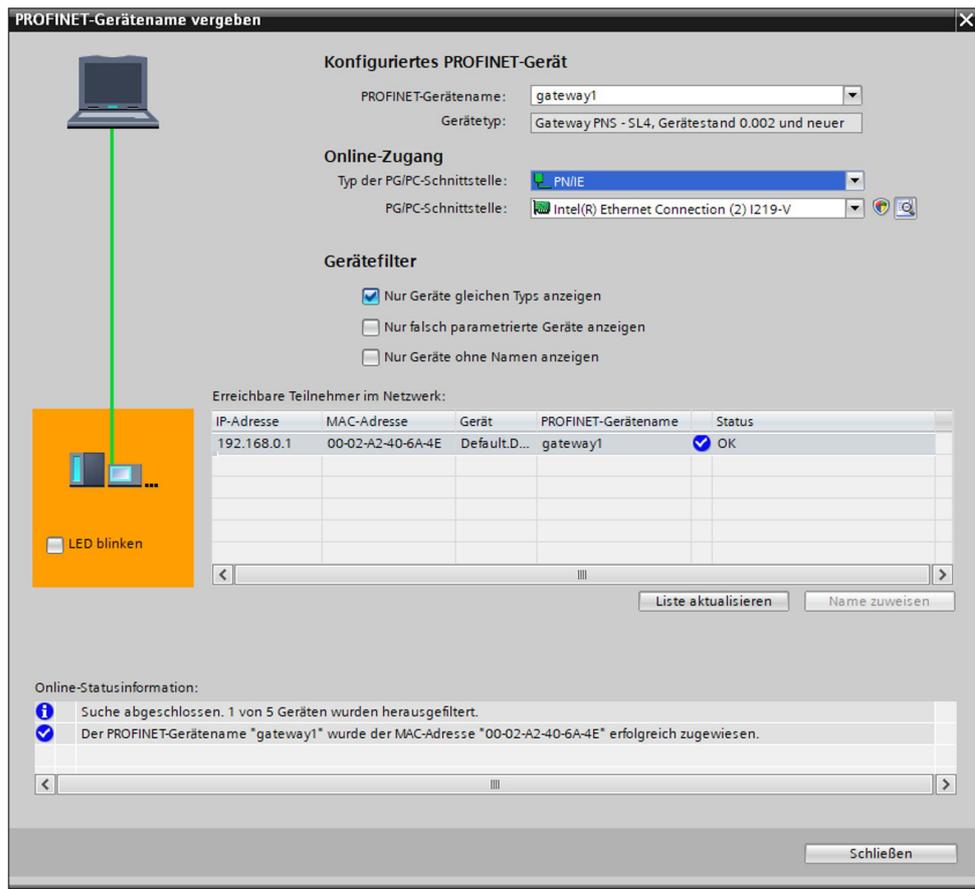


5. Wählen Sie in dem Fenster die PROFINET-Schnittstelle aus und klicken Sie auf „O-K“. Das Gateway wird nun mit dem IO-Controller verbunden.



6. Wechseln Sie nun in die Geräteansicht.
7. Um dem Gateway einen neuen Namen zu geben, klicken Sie den Modulnamen „comx100repns“ an und geben Sie den gewünschten Namen ein.

8. Weisen Sie diesen Gerätenamen nun über das Icon „Name“ dem Gateway zu. Das Fenster „PROFINET-Gerätenamen vergeben“ öffnet sich.



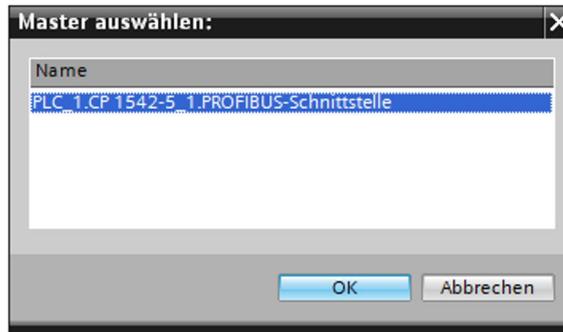
9. Wählen Sie im Hardware-Katalog die Servoantriebe aus: Modul → Ein-/Ausgangsmodule → Servoantrieb 16 Byte Ein-/Ausgang. Wiederholen Sie diesen Schritt für jeden Servoantrieb. Bei einem Doppelservoantrieb müssen zwei Antriebe ausgewählt werden.
  10. Klicken Sie auf „Liste aktualisieren“. Nun werden alle erreichbaren Geräte angezeigt.
  11. Wählen Sie das Gateway aus und klicken Sie auf „Name zuweisen“. Schließen Sie dann das Fenster.
- ✓ Das Gateway ist nun mit allen Servoantrieben im Projekt eingebunden. Bei weiteren Gateways muss dieser Vorgang für jedes Gateway wiederholt werden.

## 5.4 Einbinden eines PROFIBUS DP-Gateways 0362151(A)

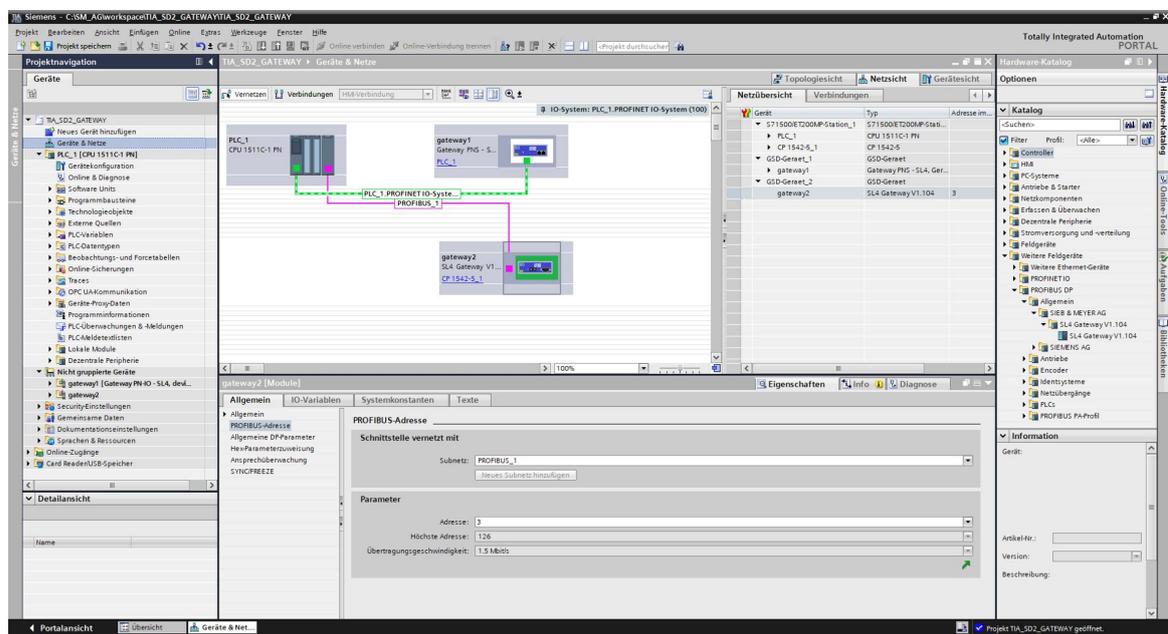
Zur Einbindung eines PROFIBUS DP-Gateways 0362151(A) gehen Sie wie folgt vor:

1. Wechseln Sie in die Netzansicht.
2. Suchen Sie im Hardware-Katalog das passende Gateway aus: Weitere Feldgeräte → PROFIBUS DP → Allgemein → SIEB & MEYER AG → SL4-Gateway V1.104 → SL4-Gateway V1.104.
3. Wählen Sie das Gerät aus und ziehen Sie es in das Netzwerk-Fenster.

4. Klicken Sie mit der rechten Maustaste auf den Text „Nicht zugeordnet“ und wählen Sie den Punkt „Neuem Master zuweisen“ im Kontextmenü aus. Das folgende Fenster wird geöffnet:



5. Wählen Sie in dem Fenster die PROFIBUS-Schnittstelle aus und klicken Sie auf „OK“. Das Gateway wird nun mit dem Master verbunden.



6. Wechseln Sie nun in die Geräteansicht.
  7. Um dem Gateway einen neuen Namen zu geben, klicken Sie den Modulnamen „Slave\_1“ an und geben Sie den gewünschten Namen ein.
  8. Wählen Sie im Hardware-Katalog die Servoantriebe aus: DS402 Single Axis 16o/16i oder DS402 Double Axis 16o/16i. Wiederholen Sie diesen Schritt für jeden Servoantrieb.
- ✓ Das Gateway ist nun mit allen Servoantrieben im Projekt eingebunden. Bei weiteren Gateways muss dieser Vorgang für jedes Gateway wiederholt werden.



# 6 Laden des Programmbeispiel

## 6.1 Projekt übersetzen

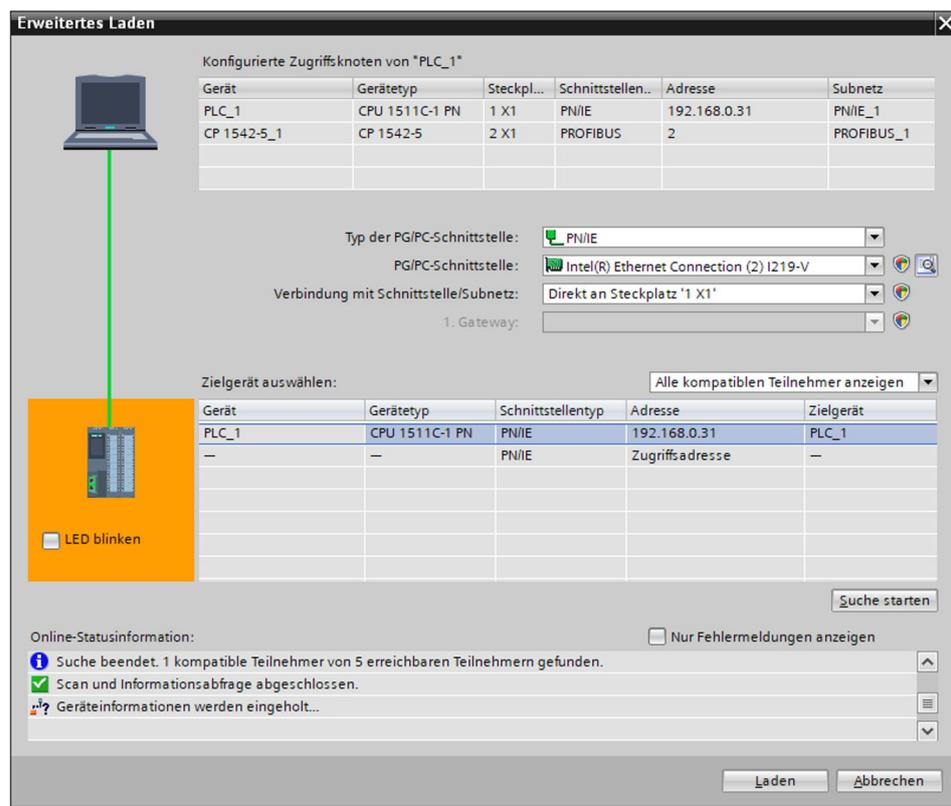
Bevor das Programmbeispiel geladen werden kann, muss es zunächst übersetzt werden:

1. Wählen Sie die CPU „PLC\_1 [CPU 1511C-1 PN]“ im Projektbaum aus.
  2. Klicken Sie auf das Icon für „Übersetzen“ in der Symbolleiste.
- ✓ Das Projekt wird nun übersetzt. Die Ergebnisse werden im Meldungsfenster angezeigt.

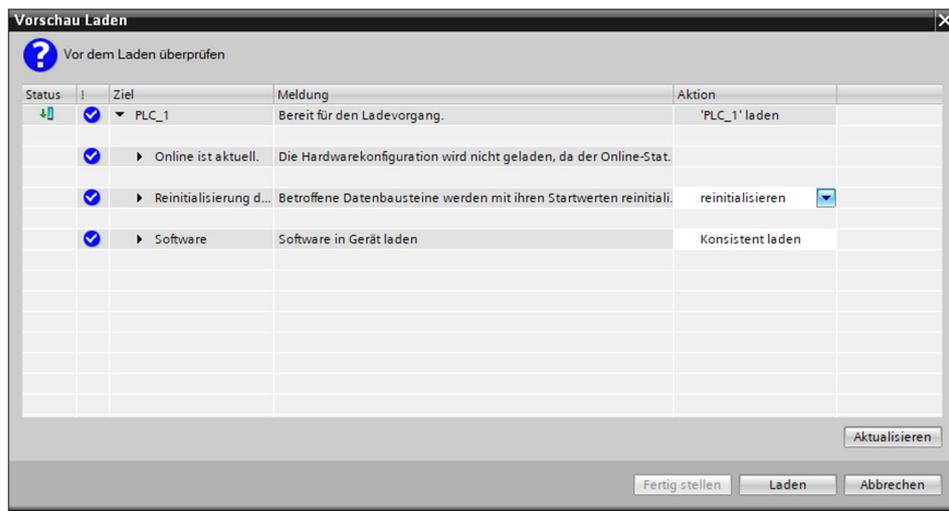
## 6.2 Programmbeispiel laden

Nun können Sie das Programmbeispiel wie folgt in das Gerät laden.

1. Klicken Sie auf das Icon für „Laden in Gerät“. Das Fenster „Erweitertes Laden“ wird geöffnet.
2. Wählen Sie ihre PG-Schnittstelle aus und klicken Sie auf „Suche starten“. Nach dem Suchdurchlauf werden die erreichbaren Zielgeräte angezeigt.



3. Wählen Sie die gewünschte SPS aus und klicken Sie auf „Laden“. Das Fenster „Vorschau Laden“ wird geöffnet.



4. Aktivieren Sie gegebenenfalls die aufgelisteten Aktionen (je nach Zustand ihrer SPS). Klicken Sie dann auf „Laden“. Der Fortschritt des Ladevorgangs wird angezeigt.
5. Nach dem Laden klicken Sie auf „Fertig stellen“.
- ✓ Das Programmbeispiel ist nun geladen. Die gesamte Verbindungskette muss sich im Status „OK“ befinden.

# 7 Fehlererkennung

Nachfolgend werden einige Fehlerbilder und deren Erkennung bei der Umsetzung Feldbus zu beschrieben.

## 7.1 Fehler in der S7-SPS

### **Ausfall der SPS**

Das Gateway erkennt einen Busfehler und setzt die Ausgangsdaten zum SERVOLINK 4 auf Null. Der Antrieb SD2x schaltet ab, da das Steuerwort Null ist.

Das Gateway meldet den Fehler E16.

### **Halt / Stopp der SPS**

Die S7-SPS setzt die Ausgangsdaten auf Null. Das Gateway leitet diese Daten weiter zum SERVOLINK 4. Der Antrieb SD2x schaltet ab, da das Steuerwort Null ist.

Das Gateway meldet den Fehler E10.

### **Austausch der Feldbusdaten**

Mit jedem Programmdurchlauf müssen die Prozessdaten mit den Feldbusdaten ausgetauscht werden. Findet kein wiederholter Datenaustausch statt, bleibt das bisherige Datenmuster erhalten und wird immer wieder gesendet. Das Gateway leitet diese Daten über SERVOLINK 4 weiter. Der Antrieb kann nicht unterscheiden, ob dauerhaft die gleichen Daten gesendet werden sollen oder der Datenaustausch nicht aufgerufen wird.

## 7.2 Fehler in der Antriebssteuerung SD2x

Ein Ausfall des Antriebs SD2x führt dazu, dass ein SERVOLINK 4-Telegramm nicht oder fehlerhaft beantwortet wird.

Das Gateway erkennt somit einen Fehler. Dabei wird zwischen einem generellen Fehler im SERVOLINK 4 und einem Slot-Fehler eines einzelnen Teilnehmers unterschieden. Wird ein genereller Fehler erkannt, werden die Ausgangsdaten zum Feldbus auf Null gesetzt. Wird ein Slot-Fehler erkannt, werden die Ausgangsdaten auf 0FFh gesetzt. Entsprechend kann der Status in dem SPS-Programm gesetzt und in eine Fehleroutine gesprungen werden.



## 8 SERVOLINK 4-Gateway-Testaufbau

Der Testaufbau erfolgt wie in der Hardwarebeschreibung dargestellt. Ist das System aufgebaut und einsatzbereit, kann mit der Überprüfung begonnen werden.

### 8.1 Überprüfung an der S7-SPS

#### CPU

Auf der Frontplatte befinden sich folgende LEDs am oberen Rand, die den Zustand der CPU und des PROFINET IO-Controllers anzeigen:

- ▶ RUN/STOP-LED (grün/gelb)
- ▶ ERROR-LED (rot)
- ▶ MAINT-LED (gelb)

Im Normalbetrieb leuchtet nur die RUN/STOP-LED grün. Tritt ein Busfehler auf, blinkt zusätzlich die ERROR-LED rot. Weitere Meldungen der LEDs können der Betriebsanleitung der SPS entnommen werden.

Zur Fehlerbehebung überprüfen Sie den korrekten Anschluss des Gateways. Prüfen Sie außerdem, ob die richtige Anzahl von Antrieben angeschlossen ist.

#### PROFIBUS DP-Master

Dieses Kommunikationsmodul hat eigene LEDs, die den Zustand anzeigen:

- ▶ RUN/STOP-LED (grün/gelb)
- ▶ ERROR-LED (rot)
- ▶ MAINT-LED (gelb)

Im Normalbetrieb leuchtet nur die RUN/STOP-LED grün. Tritt ein Busfehler auf, blinkt zusätzlich die ERROR-LED rot. Weitere Meldungen der LEDs können der Betriebsanleitung der SPS entnommen werden.

Zur Fehlerbehebung überprüfen Sie den korrekten Anschluss des Gateways. Prüfen Sie außerdem, ob die richtige Anzahl von Antrieben angeschlossen ist.

### 8.2 Überprüfung im Antrieb

Mit Hilfe des Busmonitors in der *drivemaster2*-Softwareoberfläche können die empfangenen SERVOLINK 4-Daten im Antrieb überprüft werden.

#### 8.2.1 Motoransteuerung

Wenn ein Motor am Antrieb angeschlossen ist, kann dieser über die Eingänge der SPS angesteuert werden.

- ▶ Mit dem Eingang I\_SwOn\_0 wird die Endstufe eingeschaltet. Der Antrieb wechselt in den Zustand „eingeschaltet“.
- ▶ Mit dem Eingang I\_EnOp\_0 wird der Betrieb freigegeben und der Motor kann drehen. Der Antrieb wechselt in den Zustand „Betrieb freigegeben“.
- ▶ Mit dem Eingang I\_QStop\_0 wird die Schnellhalt-Funktion ausgelöst. Bei nicht gesetztem Eingang wechselt der Antrieb in den Zustand „Schnellhalt“.



- ▶ Mit dem Eingang I\_ResFault\_0 wird ein Antriebsfehler im Zustand „Störung“ zurückgesetzt. Danach wechselt der Antrieb in den Zustand „einschaltbereit“.
- ▶ Mit dem Eingang I\_ResFault\_SC wird ein Fehler in der Sicherheitsfunktion SFM oder SLOF zurückgesetzt.

Mit den Eingängen von I\_Speed\_0 wird der Geschwindigkeitssollwert über die Werte 0 bis 10 eingestellt, das entspricht 0 bis 100 % der maximalen Geschwindigkeit des Motors. Im Demo-Programm wird der eingestellte Wert mit 1638 multipliziert und als Geschwindigkeitssollwert übertragen. Der Sollwert für den Strom ist im Demo-Programm fest auf 100 % eingestellt.

Die Maximalwerte (= 100 %) für die Geschwindigkeit und den Strom betragen  $16380_{10}$  bzw.  $3FFC_{16}$ .

Eingänge	Prozentwert [%]	Sollwert	
		dezimal	hexadezimal
0	0	0	0x0000
1	10	1638	0x0666
2	20	3276	0x0CCC
3	30	4914	0x1332
4	40	6552	0x1998
5	50	8190	0x1FFE
6	60	9828	0x2664
7	70	11466	0x2CCA
8	80	13104	0x3330
9	90	14742	0x3996
10	100	16380	0x3FFC

#### **Hinweis**

Für weitere Antriebe müssen die weiteren Eingänge entsprechend beschaltet werden.

## 9 Umstellung des Feldbussystems

Soll ein vorhandenes Projekt auf ein anderes Feldbussystem umgestellt werden, kann der Softwareteil beibehalten werden. Es müssen nur die Hardware-Konfiguration und die Verlinkungen zu den Variablen angepasst werden.

Die Anbindung der Gateway-Konfigurationen an die SPS-Software ist in den jeweiligen Handbüchern der Gateways beispielhaft für die TwinCAT 3-Software beschrieben. Für andere SPS-Systeme kann diese Beschreibung grundlegend verwendet werden.



# 10 Weiterführende Dokumente

Die folgenden Dokumente enthalten weitere Informationen zu diesem Thema:

Anbieter	Dokumentname
SIEB & MEYER AG	Gateway 0362151 – PROFIBUS-SERVOLINK 4-Verbindung
	Gateway 0362151A – PROFIBUS-SERVOLINK 4-Safety
	Gateway 0362157 – PROFINET IO-SERVOLINK 4-Verbindung
	Gateway 0362157A – PROFINET IO-SERVOLINK 4-Safety
	Antriebssystem SD2 – Sicherheitsfunktionen SFM / SLOF
	Antriebssystem SD2 – SERVOLINK 4-Bussystembeschreibung
	Antriebssystem SD2 – Gerätesteuerung
Siemens AG	profinet_step7_v15_function_manual_de-DE-de-DE.pdf
	81318674_Programming_guideline_DOKU_v13_de.pdf
	SCE_DE_012-100 Unspezifische Hardwarekonfiguration S7-1500_R1703.pdf
	Summary_SCE_Training_Curriculum_S7-1500_DE.pdf