

SIEMENS

Hans Berger

# Automatisieren mit SIMATIC S7-1500

Projektieren, Programmieren und Testen  
mit STEP 7 Professional

3. Auflage

Berger Automatisieren mit SIMATIC S7-1500



# Automatisieren mit SIMATIC S7-1500

Projektieren, Programmieren und Testen  
mit STEP 7 Professional

von Hans Berger

3., überarbeitete und erweiterte Auflage, 2019

Publicis Pixelpark

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Autor und Verlag haben alle Texte und Abbildungen in diesem Buch mit großer Sorgfalt erarbeitet. Dennoch können Fehler nicht ausgeschlossen werden. Eine Haftung des Verlags oder des Autors, gleich aus welchem Rechtsgrund, für durch die Verwendung der Programmierbeispiele verursachte Schäden ist ausgeschlossen.

[www.publicis.de/books](http://www.publicis.de/books)

**Print ISBN 978-3-89578-485-9**

**ePDF ISBN 978-3-89578-972-4**

3., überarbeitete und erweiterte Auflage, 2019

Herausgeber: Siemens Aktiengesellschaft, Berlin und München

Verlag: Publicis Pixelpark, Erlangen

© 2019 by Publicis Pixelpark Erlangen – eine Zweigniederlassung der Publicis Pixelpark GmbH

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwendung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen, Bearbeitungen sonstiger Art sowie für die Einspeicherung und Verarbeitung in elektronischen Systemen. Dies gilt auch für die Entnahme von einzelnen Abbildungen und bei auszugsweiser Verwertung von Texten.

Printed in Germany

# Vorwort

Das Automatisierungssystem SIMATIC vereinigt alle Teilbereiche einer Automatisierungslösung unter einer einheitlichen Systemarchitektur zu einem homogenen Gesamtsystem von der Feldebene bis zur Leittechnik.

Das Konzept *Totally Integrated Automation* (TIA) bedeutet, mit einer einzigen Systembasis und Werkzeugen mit einheitlichen Bedienoberflächen alle Automatisierungskomponenten einheitlich zu behandeln. Diesen Anforderungen wird das Automatisierungssystem SIMATIC gerecht mit Durchgängigkeit bei Projektierung, Programmierung, Datenhaltung und Kommunikation.

Das vorliegende Buch beschreibt das Automatisierungssystem SIMATIC S7-1500. Eine S7-1500-Steuerung ist kompakt aufgebaut und modular erweiterbar. Für die Kommunikation mit anderen Automatisierungssystemen bieten die CPU-Baugruppen integrierte Busschnittstellen für Industrial Ethernet und – abhängig vom Baugruppentyp – auch für PROFIBUS DP.

Die Engineeringsoftware STEP 7 Professional im TIA Portal erschließt die gesamte Funktionalität der S7-1500-Controller. STEP 7 Professional ist das gemeinsame Werkzeug für die Konfiguration des Hardware-Aufbaus und die Programmierung des Anwenderprogramms genauso wie für den Programmtest und die Diagnose.

Für die Gestaltung des Anwenderprogramms stellt STEP 7 Professional fünf Programmiersprachen zur Verfügung: Kontaktplan (KOP) mit einer Stromlaufplan-ähnlichen grafischen Darstellung, Funktionsplan (FUP) mit einer an elektronische Schaltkreissysteme angelehnten grafischen Darstellung, Structured Control Language (SCL) mit einer Pascal-ähnlichen Hochsprache, Anweisungsliste (AWL) mit der Formulierung der Steuerungsaufgabe als Befehlsauflistung und schließlich mit GRAPH eine Ablaufsteuerung mit sequenzieller Bearbeitung des Anwenderprogramms.

Das Testen des Anwenderprogramms erleichtert STEP 7 Professional mit Beobachtungstabellen für das Beobachten, Steuern und Forcen von Variablenwerten, mit der Darstellung des Programms mit den aktuellen Variablenwerten im laufenden Betrieb und mit einer Offline-Simulation des Automatisierungsgeräts.

Das vorliegende Buch beschreibt das Projektieren, Programmieren und Testen eines Automatisierungsgeräts CPU 1500 V2.6 mit der Engineeringsoftware STEP 7 Professional V15.1 und der Simulationssoftware S7-PLCSIM V15.1.

# Der Inhalt des Buchs auf einen Blick

---

## 1 Start

Das Automatisierungssystem SIMATIC S7-1500 in der Übersicht.  
Eine Einführung in die Engineeringsoftware SIMATIC STEP 7 Professional V15.1.  
Die Grundlage der Automatisierungslösung: Ein Projekt erstellen und bearbeiten.

---

## 2 Automatisierungssystem SIMATIC S7-1500

Die Baugruppen von SIMATIC S7-1500 in der Übersicht: Aufbau eines Automatisierungssystems, CPU-Baugruppen, Signal-, Technologie- und Kommunikationsbaugruppen.

---

## 3 Gerätekonfiguration

Eine Station konfigurieren, Baugruppen parametrieren und Stationen vernetzen.

---

## 4 Variablen, Adressierung und Datentypen

Die Eigenschaften der Operandenbereiche Eingänge, Ausgänge, Peripherie, Merker, Daten und temporäre Lokaldaten und wie sie adressiert werden: absolut, symbolisch und indirekt.  
Die Beschreibung der elementaren und strukturierten Datentypen, Datentypen für Bausteinparameter, Zeiger, Anwender- und Systemdatentypen.

---

## 5 Programmbearbeitung

Wie sich die CPU-Baugruppe in den Betriebszuständen ANLAUF, RUN und STOP verhält.  
Wie das Anwenderprogramm durch Bausteine strukturiert wird, welche Eigenschaften die Bausteine haben und wie sie aufgerufen werden.  
Wie das Anwenderprogramm bearbeitet wird: Anlaufverhalten, Hauptprogramm, Alarmbearbeitung, Fehlerbehandlung und Diagnose.

---

## 6 Der Programmierer

Arbeiten mit der PLC-Variablen-tabelle, Codebausteine und Datenbausteine erstellen und bearbeiten, Bausteine übersetzen und Programminformationen auswerten.

---

## 7 Die Programmiersprache Kontaktplan KOP

Das Charakteristische der KOP-Programmierung; Reihen- und Parallelschaltung von Kontakten; der Einsatz von Spulen, Standard-Boxen, Q-Boxen und EN/ENO-Boxen.

---

## 8 Die Programmiersprache Funktionsplan FUP

Das Charakteristische der FUP-Programmierung; Boxen für binäre Verknüpfungen; der Einsatz von Standard-Boxen, Q-Boxen und EN/ENO-Boxen.

---

## 9 Die Programmiersprache Structured Control Language SCL

Das Charakteristische der SCL-Programmierung; Operatoren und Ausdrücke, Arbeiten mit Binär- und Digitalfunktionen, Programmbearbeitung steuern mit Kontrollanweisungen.

---

---

## **10 Die Programmiersprache Anweisungsliste AWL**

Das Charakteristische der AWL-Programmierung; Programmierung von binären Verknüpfungen, Anwendung von Digitalfunktionen und Steuern der Programmbearbeitung.

---

## **11 Die Programmiersprache Ablaufsteuerung GRAPH**

Was eine Ablaufsteuerung ist und was ihre Elemente sind: Ablaufketten, Schritte, Transitionen und Verzweigungen. Wie eine Ablaufsteuerung mit GRAPH projiziert wird.

---

### **Die Beschreibung der Steuerungsfunktionen**

**12 Basisfunktionen:** Funktionen für Binärsignale: binäre Verknüpfungen, Speicherfunktionen, Flankenauswertungen, Zeit-/Zählfunktionen.

**13 Digitalfunktionen:** Funktionen für Digitalvariablen: Übertragungs-, Vergleichs-, Arithmetik-, Mathematik-, Konvertierungs-, Schiebe-, Logik- und Zeichenkettenfunktionen.

**14 Programmsteuerung:** Sprungfunktionen, Bausteine aufrufen und beenden, mit Bausteinparametern arbeiten, optimierter Bausteinzugriff, ARRAY- und CPU-Datenbausteine.

---

## **15 Online-Betrieb, Diagnose und Programmtest**

Ein Programmiergerät an die PLC-Station anschließen, den Online-Betrieb einschalten, die Projektdateien übertragen und das Anwenderprogramm schützen. Die Anwenderbausteine laden, ändern, löschen und vergleichen. Mit der Hardware-Diagnose arbeiten. Das Anwenderprogramm testen. Messwertaufzeichnung mit der Trace-Funktion.

---

## **16 Dezentrale Peripherie**

In der Übersicht: das dezentrale Peripheriesystem ET 200.

Wie ein PROFINET IO System projiziert wird und welche Eigenschaften es hat.

Wie ein PROFIBUS DP Mastersystem projiziert wird und welche Eigenschaften es hat.

---

## **17 Kommunikation**

Mit welchen Kommunikationsfunktionen die Open User Communication realisiert wird.

Welche Eigenschaften die S7-Kommunikation hat und mit welchen Kommunikationsfunktionen sie programmiert wird.

Wie die PtP-Kommunikation realisiert wird.

---

## **18 Anhang**

Wie ein mit STEP 7 V5.x erstelltes Projekt in das TIA Portal migriert wird.

Wie der Webserver in der CPU projiziert wird und welche Möglichkeiten er bietet.

Technologieobjekte für Zählen, Messen, Motion Control, PID Control.

Wie Daten protokolliert und Rezepturen übertragen werden.

Mit der Simulationssoftware S7-PLCSIM ein Anwenderprogramm offline testen.

Maschinen- und Anlagen-Diagnose mit ProDiag.

---



# Inhaltsverzeichnis

<b>1 Einführung</b>	23
1.1 Übersicht Automatisierungssystem S7-1500	23
1.1.1 Automatisierungsgerät SIMATIC S7-1500	24
1.1.2 Übersicht STEP 7 Professional	25
1.1.3 Verschiedene Programmiersprachen	27
1.1.4 Bearbeitung des Anwenderprogramms	29
1.1.5 Datenhaltung im SIMATIC-Automatisierungssystem	31
1.2 Einführung in STEP 7 Professional V15.1	33
1.2.1 STEP 7 installieren	33
1.2.2 Automation License Manager	34
1.2.3 STEP 7 Professional starten	34
1.2.4 Portalansicht	35
1.2.5 Die Fenster der Projektansicht	36
1.2.6 Informationssystem	38
1.2.7 Bedienoberfläche anpassen	39
1.3 Ein SIMATIC-Projekt bearbeiten	40
1.3.1 Strukturierte Darstellung der Projektdaten	40
1.3.2 Projektdaten und Editoren für eine PLC-Station	40
1.3.3 Mit Projekten arbeiten	44
1.3.4 Mit Referenzprojekten arbeiten	49
1.3.5 Mit Bibliotheken arbeiten	49
1.3.6 Mehrsprachige Projekte	53
<b>2 Automatisierungssystem SIMATIC S7-1500</b>	55
2.1 Komponenten einer S7-1500-Station	55
2.2 CPU-Baugruppen S7-1500	57
2.2.1 CPU-Varianten	57
2.2.2 Bedien- und Anzeigeelemente	62
2.2.3 SIMATIC Memory Card	63
2.2.4 Busschnittstellen	63
2.3 Signalbaugruppen	64
2.3.1 Digitaleingabebaugruppen	66
2.3.2 Digitalausgabebaugruppen	66
2.3.3 Digitalein-/ausgabebaugruppen	67
2.3.4 Analogeingabebaugruppen	68
2.3.5 Analogausgabebaugruppen	69
2.3.6 Analogein-/ausgabebaugruppen	69
2.4 Technologiebaugruppen	69

---

2.5	Kommunikationsbaugruppen	70
2.6	Weitere Baugruppen	72
2.6.1	Systemstromversorgungsbaugruppen	72
2.6.2	Laststromversorgungsbaugruppen	72
2.7	SIPLUS S7-1500	73
<b>3</b>	<b>Gerätekonfiguration</b>	<b>74</b>
3.1	Einführung	74
3.2	Eine Station konfigurieren	76
3.2.1	Eine PLC-Station hinzufügen	76
3.2.2	Eine Baugruppe hinzufügen	76
3.3	Baugruppen parametrieren	78
3.3.1	Die CPU-Eigenschaften parametrieren	78
3.3.2	Baugruppen adressieren	82
3.3.3	Signalbaugruppen parametrieren	84
3.3.4	Eine Konfigurationssteuerung projektieren	86
3.4	Eine Vernetzung projektieren	89
3.4.1	Einführung	89
3.4.2	Eine Station vernetzen	90
3.4.3	Teilnehmeradressen in einem Subnetz	92
3.4.4	Kommunikationsdienste und Verbindungstypen	93
3.4.5	Eine Verbindung projektieren	94
3.4.6	Ein PROFINET-Subnetz projektieren	96
3.4.7	Ein PROFIBUS-Subnetz projektieren	100
<b>4</b>	<b>Variablen, Adressierung und Datentypen</b>	<b>103</b>
4.1	Operanden und Variablen	103
4.1.1	Einführung, Übersicht	103
4.1.2	Operandenbereiche Eingänge und Ausgänge	105
4.1.3	Operandenbereich Merker	106
4.1.4	Operandenbereich Daten	108
4.1.5	Operandenbereich temporäre Lokaldaten	109
4.2	Adressierung von Variablen und Konstanten	110
4.2.1	Signalweg	110
4.2.2	Absolute Adressierung	111
4.2.3	Symbolische Adressierung	114
4.2.4	Adressierung eines Variablenteils	116
4.2.5	Adressierung einer Konstanten	116
4.2.6	Indirekte Adressierung	117
4.3	Adressierung von Hardware-Objekten	125
4.4	Allgemeines zu Datentypen	125
4.4.1	Übersicht der Datentypen	125
4.4.2	Implizite Datentypkonvertierung	127
4.4.3	Variablen überlagern (Datentypsichten)	130

4.5	Elementare Datentypen	132
4.5.1	Bitfolge-Datentypen BOOL, BYTE, WORD, DWORD und LWORD	132
4.5.2	BCD-Zahlen BCD16 und BCD32	134
4.5.3	Datentypen CHAR und WCHAR	135
4.5.4	Ganzzahl-Datentypen ohne Vorzeichen USINT, UINT, UDINT, ULINT	135
4.5.5	Ganzzahl-Datentypen mit Vorzeichen SINT, INT, DINT und LINT	136
4.5.6	Gleitpunkt-Datentypen REAL und LREAL	137
4.5.7	Datentypen für Zeitdauern	139
4.5.8	Datentypen für Zeitpunkte	141
4.6	Strukturierte Datentypen	142
4.6.1	Datum und Uhrzeit DATE_AND_TIME (DT)	142
4.6.2	Datum und Uhrzeit DATE_AND_LTIME (DTL)	144
4.6.3	Zeichenketten STRING und WSTRING	144
4.6.4	Datentyp ARRAY	145
4.6.5	Datentyp STRUCT	149
4.7	Parametertypen	151
4.7.1	Übersicht	151
4.7.2	Parametertypen BLOCK_FC und BLOCK_FB (AWL)	152
4.7.3	Parametertyp DB_ANY	152
4.7.4	Parametertyp VOID	153
4.7.5	Parametertypen POINTER, ANY, VARIANT und REFERENCE	153
4.7.6	Parametertyp ARRAY[*]	155
4.8	Zeiger	155
4.8.1	Einführung	155
4.8.2	Bereichszeiger	156
4.8.3	DB-Zeiger	156
4.8.4	ANY-Zeiger	158
4.9	PLC-Datentypen	158
4.9.1	Einen PLC-Datentyp programmieren	158
4.9.2	Einen PLC-Datentyp anwenden	159
4.9.3	PLC-Datentypen vergleichen	161
4.9.4	Einen PLC-Datentyp umnummerieren	162
4.10	Systemdatentypen	162
4.10.1	Systemdatentypen für Zeitfunktionen	162
4.10.2	Systemdatentypen für Zählfunktionen	164
4.10.3	Startinformation	165
4.11	Hardware-Datentypen	166
<b>5</b>	<b>Anwenderprogramm bearbeiten</b>	<b>167</b>
5.1	Betriebszustände	167
5.1.1	Betriebszustand STOP	168
5.1.2	Betriebszustand ANLAUF	169
5.1.3	Betriebszustand RUN	171
5.1.4	Betriebszustand HALT	172
5.1.5	Remanenzverhalten von Operanden	173

---

5.2 Anwenderprogramm erstellen	174
5.2.1 Programmbearbeitung	174
5.2.2 Strukturierung des Anwenderprogramms	176
5.2.3 Bausteinararten	177
5.2.4 Bausteineigenschaften	180
5.2.5 Bausteinschnittstelle	181
5.2.6 Einen Codebaustein aufrufen	187
5.2.7 Programmierempfehlungen	188
5.3 Anlaufprogramm	197
5.3.1 Organisationsbausteine für das Anlaufprogramm	197
5.3.2 Remanente Daten zurücksetzen	198
5.3.3 Baugruppenadresse ermitteln	198
5.3.4 Baugruppen parametrieren	201
5.4 Hauptprogramm	205
5.4.1 Organisationsbausteine für das Hauptprogramm	205
5.4.2 Prozessabbild-Aktualisierung	206
5.4.3 Zykluszeit	210
5.4.4 Reaktionszeit	212
5.4.5 Programm stoppen und verzögern	214
5.4.6 Uhrzeit	214
5.4.7 Systemzeit lesen	220
5.4.8 Betriebsstundenzähler	220
5.5 Alarmbearbeitung	221
5.5.1 Einführung zur Alarmbearbeitung	221
5.5.2 Uhrzeitalarme	226
5.5.3 Verzögerungsalarme	230
5.5.4 Weckalarme	234
5.5.5 Prozessalarme	238
5.5.6 Alarmer zur Laufzeit zuordnen	240
5.5.7 Alarmzusatzinformation lesen	241
5.6 Fehlerbehandlung, Diagnose	243
5.6.1 Fehlerursachen und Fehlerreaktionen	243
5.6.2 Lokale Fehlerbehandlung	243
5.6.3 Globale Fehlerbehandlung (Synchronfehler)	248
5.6.4 Synchronfehlerbearbeitung sperren und freigeben	250
5.6.5 Asynchronfehler	253
5.6.6 Alarmer und Asynchronfehler sperren, verzögern und freigeben	257
5.7 Diagnose im Anwenderprogramm	259
5.7.1 Diagnosealarm OB 82	259
5.7.2 Startinformation lesen	261
5.7.3 Laufzeit-Informationen lesen	262
5.7.4 Diagnosefunktionen im Anwenderprogramm	263
5.8 Meldungen projektieren	270
5.8.1 Einführung	270
5.8.2 Meldungen nach dem Meldenummerverfahren projektieren	274
5.8.3 Systembausteine für Meldungen	279

5.8.4 Anwender-Diagnosemeldung erzeugen .....	282
5.8.5 CPU-Meldungsanzeige .....	283
5.9 Software Units .....	285
<b>6 Programmator</b> .....	<b>290</b>
6.1 Einführung .....	290
6.2 PLC-Variablen-Tabelle .....	291
6.2.1 PLC-Variablen-Tabelle anlegen und bearbeiten .....	291
6.2.2 PLC-Variablen definieren und bearbeiten .....	291
6.2.3 PLC-Variablen-Tabellen vergleichen .....	294
6.2.4 PLC-Variablen-Tabelle exportieren und importieren .....	295
6.2.5 Konstanten-Tabellen .....	296
6.3 Einen Codebaustein programmieren .....	296
6.3.1 Einen neuen Codebaustein anlegen .....	296
6.3.2 Arbeitsbereich des Programmators für Codebausteine .....	298
6.3.3 Bausteineigenschaften für Codebausteine festlegen .....	300
6.3.4 Einen Baustein schützen .....	303
6.3.5 Bausteinschnittstelle programmieren .....	306
6.3.6 Allgemeines Vorgehen beim Programmieren der Steuerungsfunktion .....	309
6.3.7 Instanzdaten anlegen .....	311
6.4 Einen Datenbaustein programmieren .....	315
6.4.1 Einen neuen Datenbaustein anlegen .....	315
6.4.2 Arbeitsbereich des Programmators für Datenbausteine .....	316
6.4.3 Bausteineigenschaften für Datenbausteine festlegen .....	317
6.4.4 Datenvariablen deklarieren .....	320
6.4.5 Datenvariablen in Globaldatenbausteinen eingeben .....	321
6.5 Bausteine übersetzen .....	323
6.5.1 Übersetzung starten .....	323
6.5.2 SCL-Bausteine übersetzen .....	324
6.5.3 Fehler nach der Übersetzung beheben .....	325
6.6 Arbeiten mit Quelldateien .....	326
6.7 Programm-Informationen .....	329
6.7.1 Querverweisliste .....	330
6.7.2 Belegungsplan .....	333
6.7.3 Aufrufstruktur .....	334
6.7.4 Abhängigkeitsstruktur .....	335
6.7.5 Konsistenzprüfung .....	336
6.7.6 Speicherauslastung der CPU .....	337
<b>7 Kontaktplan KOP</b> .....	<b>338</b>
7.1 Einführung .....	338
7.1.1 Mit Kontaktplan programmieren .....	338
7.1.2 Programmelemente des Kontaktplans .....	344
7.2 Binäre Verknüpfungen mit KOP programmieren .....	344
7.2.1 Schließerkontakt und Öffnerkontakt .....	344

---

7.2.2 Reihen- und Parallelschaltung von Kontakten .....	347
7.2.3 T-Abzweig, offener Parallelzweig .....	348
7.2.4 Verknüpfungsergebnis negieren im Kontaktplan .....	348
7.2.5 Flankenauswertung einer Binärvariablen im Kontaktplan .....	349
7.2.6 Gültigkeitsprüfung einer Gleitpunktvariablen im Kontaktplan .....	350
7.2.7 Vergleich-Kontakte .....	351
7.3 Speicherfunktionen mit KOP programmieren .....	351
7.3.1 Einfache und negierende Spule .....	352
7.3.2 Setzen- und Rücksetzen-Spule .....	353
7.3.3 Speicherndes Verhalten durch Selbsthaltung .....	354
7.3.4 Flankenauswertung mit Impulsabgabe im Kontaktplan .....	354
7.3.5 Mehrfaches Setzen und Rücksetzen (Bitfeld füllen) im Kontaktplan ..	356
7.3.6 Spulen mit Zeitverhalten .....	356
7.4 Q-Boxen mit KOP programmieren .....	357
7.4.1 Speicher-Boxen im Kontaktplan .....	358
7.4.2 Flankenauswertung des Stromflusses .....	359
7.4.3 Zeitfunktionen im Kontaktplan .....	359
7.4.4 Zählfunktionen im Kontaktplan .....	360
7.5 EN/ENO-Boxen mit KOP programmieren .....	362
7.5.1 Flankenauswertung mit einer EN/ENO-Box .....	362
7.5.2 Übertragungsfunktionen im Kontaktplan .....	364
7.5.3 Arithmetische Funktionen im Kontaktplan .....	364
7.5.4 Mathematische Funktionen im Kontaktplan .....	366
7.5.5 Konvertierungsfunktionen im Kontaktplan .....	366
7.5.6 Schiebefunktionen im Kontaktplan .....	367
7.5.7 Logikfunktionen im Kontaktplan .....	368
7.5.8 Funktionen für Zeichenketten im Kontaktplan .....	369
7.5.9 Die CALCULATE-Box im Kontaktplan .....	370
7.6 VARIANT-Funktionen mit KOP programmieren .....	370
7.7 REFERENCE-Funktionen mit KOP programmieren .....	372
7.8 DB_ANY-Funktionen mit KOP programmieren .....	374
7.9 Programmsteuerung mit KOP .....	375
7.9.1 Sprungfunktionen im Kontaktplan .....	375
7.9.2 Bausteinende-Funktion im Kontaktplan .....	378
7.9.3 Bausteinaufruf-Funktionen im Kontaktplan .....	379
<b>8 Funktionsplan FUP .....</b>	<b>381</b>
8.1 Einführung .....	381
8.1.1 Mit Funktionsplan programmieren .....	381
8.1.2 Programmelemente des Funktionsplans .....	387
8.2 Binäre Verknüpfungen mit FUP programmieren .....	388
8.2.1 Abfrage auf Signalzustand „1“ und auf Signalzustand „0“ .....	388
8.2.2 Eine binäre Verknüpfung im Funktionsplan programmieren .....	388
8.2.3 Binäre Funktionen kombinieren .....	392
8.2.4 Verknüpfungsergebnis negieren im Funktionsplan .....	392
8.2.5 T-Abzweig im Funktionsplan .....	392

8.2.6	Flankenauswertung von Binärvariablen im Funktionsplan	393
8.2.7	Gültigkeitsprüfung von Gleitpunktzahlen im Funktionsplan	394
8.2.8	Vergleichsfunktionen im Funktionsplan	394
8.3	Standard-Boxen mit FUP programmieren	395
8.3.1	Zuweisung und negierende Zuweisung	396
8.3.2	Setzen- und Rücksetzen-Box	397
8.3.3	Flankenauswertung mit Impulsausgabe im Funktionsplan	398
8.3.4	Mehrfaches Setzen und Rücksetzen (Bitfeld füllen) im Funktionsplan	398
8.3.5	Standard-Boxen mit Zeitverhalten	399
8.4	Q-Boxen mit FUP programmieren	400
8.4.1	Speicher-Boxen im Funktionsplan	401
8.4.2	Flankenauswertung des Verknüpfungsergebnisses im Funktionsplan	402
8.4.3	Zeitfunktionen im Funktionsplan	403
8.4.4	Zählfunktionen im Funktionsplan	404
8.5	EN/ENO-Boxen mit FUP programmieren	405
8.5.1	Flankenauswertung mit einer EN/ENO-Box	405
8.5.2	Übertragungsfunktionen im Funktionsplan	407
8.5.3	Arithmetische Funktionen im Funktionsplan	408
8.5.4	Mathematische Funktionen im Funktionsplan	409
8.5.5	Konvertierungsfunktionen im Funktionsplan	410
8.5.6	Schiebefunktionen im Funktionsplan	410
8.5.7	Logikfunktionen im Funktionsplan	411
8.5.8	Funktionen für Zeichenketten im Funktionsplan	413
8.5.9	Die CALCULATE-Box im Funktionsplan	413
8.6	VARIANT-Funktionen mit FUP programmieren	413
8.7	REFERENCE-Funktionen mit FUP programmieren	415
8.8	DB_ANY-Funktionen mit FUP programmieren	418
8.9	Programmsteuerung mit FUP	419
8.9.1	Sprungfunktionen im Funktionsplan	419
8.9.2	Bausteinende-Funktion im Funktionsplan	422
8.9.3	Bausteinaufruf-Funktionen im Funktionsplan	423
<b>9</b>	<b>Structured Control Language SCL</b>	<b>425</b>
9.1	Einführung	425
9.1.1	Mit SCL programmieren	425
9.1.2	SCL-Anweisungen und Operatoren	430
9.2	Binäre Verknüpfungen mit SCL programmieren	432
9.2.1	Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	432
9.2.2	Eine Binäre Verknüpfung in SCL programmieren	433
9.2.3	Verknüpfungsergebnis in SCL negieren	435
9.3	Speicherfunktionen mit SCL programmieren	436
9.3.1	Wertzuweisung einer Binärvariablen	436
9.3.2	Setzen und Rücksetzen in SCL	437
9.3.3	Flankenauswertung in SCL	437

---

9.4 Zeit- und Zählfunktionen mit SCL programmieren	438
9.4.1 Zeitfunktionen in SCL	438
9.4.2 Zählfunktionen in SCL	439
9.5 Digitalfunktionen mit SCL programmieren	439
9.5.1 Übertragungsfunktion, Wertzuweisung einer Digitalvariablen	440
9.5.2 Vergleichsfunktionen in SCL	440
9.5.3 Arithmetische Funktionen in SCL	440
9.5.4 Mathematische Funktionen in SCL	442
9.5.5 Konvertierungsfunktionen in SCL	443
9.5.6 Schiebefunktionen in SCL	445
9.5.7 Wortverknüpfungen, logischer Ausdruck in SCL	445
9.5.8 Funktionen für Zeichenketten in SCL	446
9.6 VARIANT-Funktionen mit SCL programmieren	447
9.7 REFERENCE-Funktionen mit SCL programmieren	449
9.8 DB_ANY-Funktion mit SCL programmieren	451
9.9 Programmsteuerung mit SCL	452
9.9.1 Kontrollanweisungen	452
9.9.2 Bausteinendefunktion bei SCL	461
9.9.3 Aufruf einer Funktion (FC) bei SCL	461
9.9.4 Aufruf eines Funktionsbausteins (FB) bei SCL	462
9.9.5 Versorgung von Parametern	464
<b>10 Anweisungsliste AWL</b>	<b>465</b>
10.1 Einführung	465
10.1.1 Mit AWL programmieren	465
10.1.2 Adressierung von 64-Bit-Variablen	472
10.2 Binäre Verknüpfungen mit AWL programmieren	473
10.2.1 Bearbeitung einer binären Verknüpfung, Verknüpfungsschritt	473
10.2.2 Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	474
10.2.3 UND-Funktion in der Anweisungsliste	475
10.2.4 ODER-Funktion in der Anweisungsliste	476
10.2.5 Exklusiv-ODER-Funktion in der Anweisungsliste	476
10.2.6 Kombinierte binäre Verknüpfungen in der Anweisungsliste	476
10.2.7 Verknüpfungsergebnis steuern	479
10.3 Speicherfunktionen mit AWL programmieren	480
10.3.1 Zuweisung in der Anweisungsliste	480
10.3.2 Setzen und Rücksetzen in der Anweisungsliste	481
10.3.3 Flankenbewertung in der Anweisungsliste	481
10.4 Zeit- und Zählfunktionen mit AWL programmieren	482
10.4.1 Zeitfunktionen in der Anweisungsliste	482
10.4.2 Zählfunktionen in der Anweisungsliste	484
10.5 Digitalfunktionen mit AWL programmieren	485
10.5.1 Übertragungsfunktionen in der Anweisungsliste	485
10.5.2 Vergleichsfunktionen in der Anweisungsliste	489
10.5.3 Arithmetische Funktionen in der Anweisungsliste	492
10.5.4 Mathematische Funktionen in der Anweisungsliste	496



10.5.5	Konvertierungsfunktionen in der Anweisungsliste	498
10.5.6	Schiebefunktionen in der Anweisungsliste	500
10.5.7	Wortverknüpfungen in der Anweisungsliste	502
10.5.8	Funktionen für Zeichenketten in der Anweisungsliste	505
10.6	VARIANT-Funktionen mit AWL programmieren	506
10.7	REFERENCE-Funktionen mit AWL programmieren	509
10.8	DB_ANY-Funktionen mit AWL programmieren	509
10.9	Programmsteuerung mit AWL	512
10.9.1	Sprungfunktionen in der Anweisungsliste	512
10.9.2	Bausteinende-Funktionen in der Anweisungsliste	513
10.9.3	Bausteinaufruf-Funktion in der Anweisungsliste	513
10.10	Weitere AWL-Funktionen	517
10.10.1	Mit Statusbits arbeiten	518
10.10.2	Akkumulatorfunktionen	522
10.10.3	Arbeiten mit den Datenbausteinregistern	525
10.10.4	Teiladressierung von Datenoperanden	527
10.10.5	Absolute Adressierung von temporären Lokaldaten	529
10.10.6	Arbeiten mit den Adressregistern	529
10.10.7	Speicherindirekte Adressierung	534
10.10.8	Registerindirekte Adressierung	535
10.10.9	Direkter Zugriff auf komplexe Lokalvariablen	539
10.10.10	Nullanweisungen	541
<b>11</b>	<b>Ablaufsteuerung S7-GRAPH</b>	<b>542</b>
11.1	Einführung	542
11.1.1	Was ist eine Ablaufsteuerung?	542
11.1.2	Eigenschaften einer Ablaufsteuerung	542
11.2	Elemente einer Ablaufsteuerung	544
11.2.1	Schritte und Transitionen	544
11.2.2	Sprünge in einer Ablaufsteuerung	546
11.2.3	Verzweigungen einer Ablaufkette	546
11.2.4	Permanente Anweisungen	548
11.2.5	Schritt- und Transitionsfunktionen	548
11.2.6	Bearbeitung von Bedingungen	552
11.2.7	Bearbeitung von Aktionen	553
11.3	Eine Ablaufsteuerung projektieren	559
11.3.1	Allgemeines Vorgehen bei der Projektierung	559
11.3.2	Den GRAPH-Funktionsbaustein programmieren	560
11.3.3	Projektieren der Kettenstruktur	561
11.3.4	Schritte und Transitionen programmieren	564
11.3.5	Permanente Anweisungen programmieren	566
11.3.6	Meldungen projektieren	566
11.3.7	Attribute des GRAPH-Funktionsbausteins	567
11.3.8	Den GRAPH-Funktionsbaustein aufrufen	568
11.4	Ablaufsteuerung testen	569
11.4.1	GRAPH-Funktionsbaustein laden	569

---

11.4.2	Einstellungen zum Programmtest	570
11.4.3	Betriebsarten verwenden	571
11.4.4	Ablaufkette synchronisieren	571
11.4.5	Mit Programmstatus testen	573
<b>12</b>	<b>Basisfunktionen</b>	<b>575</b>
12.1	Binäre Verknüpfungen	575
12.1.1	Einführung	575
12.1.2	Arbeiten mit Binärsignalen	576
12.1.3	UND-Funktion, Reihenschaltung	580
12.1.4	ODER-Funktion, Parallelschaltung	580
12.1.5	Exklusiv-ODER-Funktion, Antivalenzfunktion	581
12.1.6	Verknüpfungsergebnis negieren, NOT-Kontakt	582
12.2	Speicherfunktionen	583
12.2.1	Einführung	583
12.2.2	Einfache und negierende Spule, Zuweisung	583
12.2.3	Einzelnes Setzen und Rücksetzen	584
12.2.4	Mehrfaches Setzen und Rücksetzen	585
12.2.5	Vorrangiges Setzen und Rücksetzen, Speicher-Boxen	586
12.3	Flankenbewertung	588
12.3.1	Funktionsweise einer Flankenbewertung	588
12.3.2	Flankenbewertung einer Binärvariablen (KOP, FUP)	590
12.3.3	Flankenbewertung mit Impulsausgabe (KOP, FUP)	591
12.3.4	Flankenbewertung mit einer Q-Box (KOP, FUP)	592
12.3.5	Flankenbewertung mit einer EN/ENO-Box (KOP, FUP)	593
12.3.6	Flankenbewertung bei SCL	594
12.3.7	Flankenbewertung bei AWL	596
12.4	Zeitfunktionen	597
12.4.1	Eine Zeitfunktion hinzufügen	597
12.4.2	Impulsbildung TP	598
12.4.3	Einschaltverzögerung TON	599
12.4.4	Ausschaltverzögerung TOF	600
12.4.5	Akkumulierende Einschaltverzögerung TONR	601
12.4.6	Eine Zeitfunktion mit einer Zeitdauer laden	602
12.4.7	Eine Zeitfunktion zurücksetzen	603
12.5	Zählfunktionen	603
12.5.1	Eine Zählfunktion einfügen	604
12.5.2	Vorwärtszähler CTU	604
12.5.3	Rückwärtszähler CTD	605
12.5.4	Vorwärts-Rückwärtszähler CTUD	606
<b>13</b>	<b>Digitalfunktionen</b>	<b>609</b>
13.1	Übertragungsfunktionen	610
13.1.1	Variable kopieren, MOVE- und S_MOVE-Box bei KOP und FUP	610
13.1.2	Wertzuweisungen bei SCL	612

13.1.3	Laden und Transferieren bei AWL	614
13.1.4	Variable kopieren, MOVE und S_MOVE bei AWL	616
13.1.5	VARIANT-Variable lesen und schreiben	617
13.1.6	Zuweisung und Zuweisungsversuch von REFERENCE-Zeigern	618
13.1.7	Datenbereich kopieren mit MOVE_BLK_VARIANT	620
13.1.8	Datenbereich kopieren mit MOVE_BLK und UMOVE_BLK	622
13.1.9	Datenbereich füllen mit FILL_BLK und UFILL_BLK	624
13.1.10	Variable von und zu einen BYTE-Feld übertragen	624
13.1.11	Variable von und zu einem Bitfeld übertragen	628
13.1.12	Lesen und Schreiben mit PEEK und POKE (SCL, AWL)	630
13.1.13	Byte-Reihenfolge ändern mit SWAP	632
13.1.14	Bereichsgrenzen ermitteln bei ARRAY[*]	633
13.2	Vergleichsfunktionen	633
13.2.1	Ausführung der Vergleichsfunktion	634
13.2.2	Bereichsvergleich bei KOP und FUP	636
13.2.3	Gleitpunkt-Variable testen, OK-Kontakt, OK-Box	636
13.2.4	VARIANT-Zeiger testen bei KOP, FUP und AWL	638
13.2.5	VARIANT-Zeiger testen bei SCL	640
13.2.6	REFERENCE-Zeiger testen	642
13.2.7	DB_ANY-Funktionen	643
13.3	Arithmetische Funktionen	644
13.3.1	Arithmetische Funktionen für Zahlenwerte	644
13.3.2	Arithmetische Funktionen für Zeitwerte	646
13.3.3	Dekrementieren und Inkrementieren	646
13.4	Mathematische Funktionen	648
13.4.1	Allgemeine Funktionsbeschreibung	648
13.4.2	Winkelfunktionen SIN, COS, TAN	650
13.4.3	Arcusfunktionen ASIN, ACOS, ATAN	650
13.4.4	Quadrat bilden und Quadratwurzel ziehen	650
13.4.5	Logarithmus und Potenz	651
13.4.6	Nachkommastellen extrahieren, Absolutwert und Negation bilden	653
13.4.7	Rechnen mit der CALCULATE-Box bei KOP und FUP	654
13.5	Konvertierungsfunktionen	656
13.5.1	Konvertierungsfunktionen CONV, S_CONV und T_CONV	657
13.5.2	Konvertierung von DB_ANY	662
13.5.3	Konvertierungsfunktionen für Gleitpunktzahlen	664
13.5.4	Konvertierungsfunktionen STRG_TO_CHARS und CHARS_TO_STRG	666
13.5.5	Konvertierungsfunktionen STRG_VAL und VAL_STRG	666
13.5.6	Konvertierungsfunktionen ATH und HTA	670
13.5.7	Konvertierungsfunktionen SCALE_X und NORM_X	672
13.6	Schiebefunktionen	672
13.6.1	Allgemeine Funktionsbeschreibung	672
13.6.2	Rechts schieben SHR	673
13.6.3	Links schieben SHL	675
13.6.4	Rechts rotieren ROR	675
13.6.5	Links rotieren ROL	675

---

13.7 Logikfunktionen	675
13.7.1 Wortverknüpfungen	676
13.7.2 Invertieren, Einerkomplement bilden	677
13.7.3 Codierfunktionen DECO und ENCO	678
13.7.4 Auswahlfunktionen SEL, MUX und DEMUX	679
13.7.5 Minimumauswahl MIN, Maximumauswahl MAX	682
13.7.6 Begrenzer LIMIT	682
13.8 Zeichenketten bearbeiten	683
13.9 Symbolnamen lesen	693
<b>14 Programmsteuerung</b>	<b>698</b>
14.1 Sprungfunktionen	699
14.1.1 Einführung	699
14.1.2 Absoluter Sprung	699
14.1.3 Bedingter Sprung	701
14.1.4 Sprungliste	703
14.1.5 Sprungverteiler	705
14.1.6 Schleifensprung	705
14.2 Bausteinende-Funktionen	707
14.2.1 Bausteinende-Funktion RET (KOP und FUP)	707
14.2.2 RETURN-Anweisung (SCL)	709
14.2.3 Bausteinende-Funktionen BEB, BEA und BE (AWL)	709
14.3 Aufruf von Codebausteinen	709
14.3.1 Einführung	709
14.3.2 Aufruf einer Funktion FC	710
14.3.3 Aufruf eines Funktionsbausteins FB	710
14.3.4 Asynchron arbeitende Systembausteine	713
14.3.5 EN/ENO-Mechanismus	713
14.4 Arbeiten mit Bausteinen	717
14.4.1 Bausteine mit optimiertem und Standardzugriff	717
14.4.2 Datentypen der lokalen Variablen	719
14.4.3 Bausteinparameter adressieren	721
14.4.4 Bausteinparameter versorgen	724
14.4.5 Übergabe von Bausteinparametern	726
14.5 Datenbausteinfunktionen	729
14.5.1 Datenbausteinattribute lesen	730
14.5.2 Ladespeicher lesen und schreiben	730
14.5.3 ARRAY-Datenbausteine	732
14.5.4 Systembausteine für den Zugriff auf ARRAY-Datenbausteine	733
14.5.5 CPU-Datenbausteine	735
<b>15 Online-Betrieb, Diagnose und Test</b>	<b>739</b>
15.1 PLC-Station online verbinden	740
15.1.1 Programmiergerät an die PLC-Station anschließen	740
15.1.2 Die CPU urlöschen	742

15.1.3	Den Auslieferungszustand wiederherstellen	743
15.2	Projektdateien übertragen	744
15.2.1	Die Projektdateien erstmalig laden	745
15.2.2	Die Projektdateien nachladen	747
15.2.3	Das Anwenderprogramm schützen	748
15.2.4	Mit Online-Projektdateien arbeiten	751
15.2.5	Mit der Memory Card arbeiten	754
15.3	Mit Bausteinen im Online-Betrieb arbeiten	755
15.3.1	Einführung	755
15.3.2	Einen Online-Baustein bearbeiten	756
15.3.3	Einen Baustein laden und zurückerladen	757
15.3.4	Ohne Reinitialisierung laden	758
15.3.5	Mit Momentaufnahmen arbeiten	760
15.3.6	Mit Einstellwerten arbeiten	761
15.3.7	Bausteine vergleichen	763
15.4	Hardware-Diagnose	766
15.4.1	Status-Anzeigen an den Baugruppen	766
15.4.2	Diagnosefenster	767
15.4.3	Diagnosepuffer	769
15.4.4	Online-Tools	769
15.4.5	Weitere Diagnose-Informationen über das Programmiergerät	771
15.5	Anwenderprogramm testen	772
15.5.1	Aufrufumgebung definieren	773
15.5.2	Testen mit Programmstatus	774
15.5.3	Testen im Einzelschrittmodus	778
15.5.4	PLC-Variablen beobachten	780
15.5.5	Datenvariablen beobachten und steuern	781
15.5.6	Testen mit Beobachtungstabellen	782
15.5.7	Testen mit der Force-Tabelle	787
15.6	Messwertaufzeichnung mit der Trace-Funktion	790
15.6.1	Einführung	790
15.6.2	Erstellen der Trace-Konfiguration	791
15.6.3	Messwerte aufzeichnen	792
15.6.4	Messungen auswerten	793
15.6.5	Messungen speichern	794
<b>16</b>	<b>Dezentrale Peripherie</b>	<b>796</b>
16.1	Einführung, Übersicht	796
16.2	Dezentrales Peripheriesystem ET 200	797
16.2.1	ET 200MP	797
16.2.2	ET 200M	798
16.2.3	ET 200SP	799
16.2.4	ET 200S	800
16.2.5	ET 200pro	800
16.2.6	ET 200AL	801
16.2.7	ET 200eco und ET200eco PN	802

---

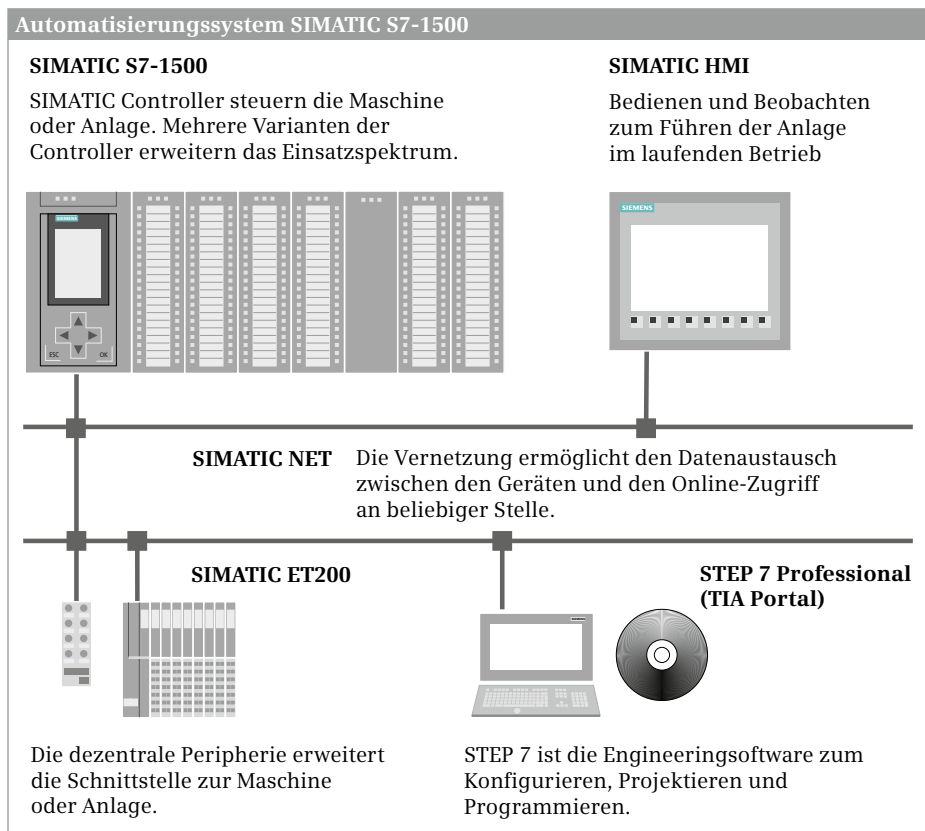
16.3 PROFINET IO .....	802
16.3.1 Komponenten von PROFINET IO .....	802
16.3.2 Adressen bei PROFINET IO .....	805
16.3.3 PROFINET IO projektieren .....	807
16.3.4 Kopplungsbaugruppen für PROFINET IO .....	813
16.3.5 Echtzeit-Kommunikation bei PROFINET .....	814
16.3.6 Spezielle PROFINET-Konfigurationen .....	818
16.4 PROFIBUS DP .....	822
16.4.1 Komponenten von PROFIBUS DP .....	822
16.4.2 Adressen bei PROFIBUS DP .....	825
16.4.3 PROFIBUS DP projektieren .....	827
16.4.4 Kopplungsbaugruppen für PROFIBUS DP .....	830
16.4.5 Spezielle PROFIBUS-Konfigurationen .....	833
16.5 Systembausteine für dezentrale Peripherie .....	836
16.5.1 Peripheriedaten lesen und schreiben .....	836
16.5.2 Diagnosedaten von einem DP-Normslave lesen .....	839
16.5.3 Datensatz empfangen und bereitstellen .....	840
16.5.4 Dezentrale Station aktivieren/deaktivieren .....	842
16.5.5 PROFINET-IO-System umkonfigurieren .....	843
16.6 DPV1-Alarme .....	843
16.7 Taktsynchronität .....	846
16.7.1 Einführung .....	846
16.7.2 Taktsynchronität bei PROFINET .....	846
16.7.3 Taktsynchronität bei PROFIBUS .....	849
16.7.4 Taktsynchronalarm .....	852
16.7.5 Prozessabbilder taktsynchron aktualisieren .....	854
<b>17 Kommunikation .....</b>	<b>855</b>
17.1 Übersicht .....	855
17.2 Open User Communication .....	858
17.2.1 Grundlagen .....	858
17.2.2 Datenstruktur der Open User Communication .....	859
17.2.3 Verbindung aufbauen und Daten senden mit TSEND_C .....	860
17.2.4 Verbindung aufbauen und Daten empfangen mit TRCV_C .....	861
17.2.5 Open User Communication projektieren .....	863
17.2.6 Weitere Funktionen für die Open User Communication .....	865
17.3 S7-Kommunikation .....	868
17.3.1 Grundlagen .....	868
17.3.2 Einseitiger Datenaustausch .....	868
17.3.3 Zweiseitiger Datenaustausch .....	870
17.3.4 S7-Kommunikation projektieren .....	873
17.4 Freeport-Kommunikation .....	874
17.4.1 Einführung in die Freeport-Kommunikation .....	874
17.4.2 Konfigurieren der Kommunikationsbaugruppe CM PtP .....	875
17.4.3 Freeport-Kommunikationsfunktionen .....	876

17.5 Weitere Kommunikationsfunktionen .....	880
17.5.1 USS-Protokoll für Antriebe .....	880
17.5.2 Modbus RTU .....	882
17.5.3 Modbus TCP .....	884
<b>18 Anhang .....</b>	<b>887</b>
18.1 Ein Projekt migrieren .....	887
18.2 Webserver .....	890
18.2.1 Webserver aktivieren .....	890
18.2.2 Standard-Webseiten .....	893
18.2.3 Basic-Webseiten .....	896
18.2.4 Service-Daten auslesen .....	896
18.2.5 Webserver initialisieren und Webseiten synchronisieren (WWW) ...	896
18.3 Technologieobjekte .....	897
18.3.1 Technologieobjekte für Motion Control .....	898
18.3.2 Technologieobjekte für PID Control .....	903
18.3.3 Technologieobjekte für Zählen und Messen .....	905
18.4 Daten protokollieren und Rezepturen übertragen .....	910
18.4.1 Einführung in die Datenprotokollierung .....	910
18.4.2 Datenprotokollierung anwenden .....	910
18.4.3 Funktionen für die Datenprotokollierung .....	910
18.4.4 Einführung zur Rezepturenübertragung .....	913
18.4.5 Funktionen für die Rezepturenübertragung .....	914
18.5 Simulation mit S7-PLCSIM .....	915
18.5.1 S7-PLCSIM allgemein .....	915
18.5.2 Die Bedienoberfläche von PLCSIM .....	917
18.5.3 In S7-PLCSIM mit STEP-7-Testfunktionen testen .....	918
18.5.4 Mit einem Simulationsprojekt arbeiten .....	920
18.5.5 Mit dem Adressbereich testen .....	921
18.5.6 Mit der SIM-Tabelle testen .....	921
18.5.7 Mit der Sequenztabelle testen .....	923
18.6 Maschinen- und Anlagendiagnose mit ProDiag .....	926
<b>Stichwortverzeichnis .....</b>	<b>930</b>

# 1 Einführung

## 1.1 Übersicht Automatisierungssystem S7-1500

SIMATIC S7-1500 ist das modulare Automatisierungssystem für den mittleren und oberen Leistungsbereich. Verschiedene Varianten der Controller passen die Leistungsfähigkeit an den jeweiligen Einsatzfall an. Je nach Bedarf kann das Automatisierungsgerät mit Ein-/Ausgabebaugruppen für Digital- und Analogsignale sowie Technologie- und Kommunikationsbaugruppen modular erweitert werden. Das Automatisierungssystem SIMATIC S7-1500 ist nahtlos eingebettet in die SIMATIC-Systemarchitektur (Bild 1.1).



**Bild 1.1** Bestandteile des Automatisierungssystems SIMATIC S7-1500



Die dezentrale Peripherie SIMATIC ET200 bietet eine zusätzliche Erweiterung mit Ein-/Ausgabebaugruppen, die mit PROFIBUS DP oder PROFINET IO mit dem zentralen Steuerungsgerät verbunden sind. Die dezentralen Stationen können in einem Schaltschrank oder – mit besonderen Bauformen für erhöhte mechanische Anforderung ausgestattet – auch vor Ort direkt an der Maschine oder Anlage installiert werden.

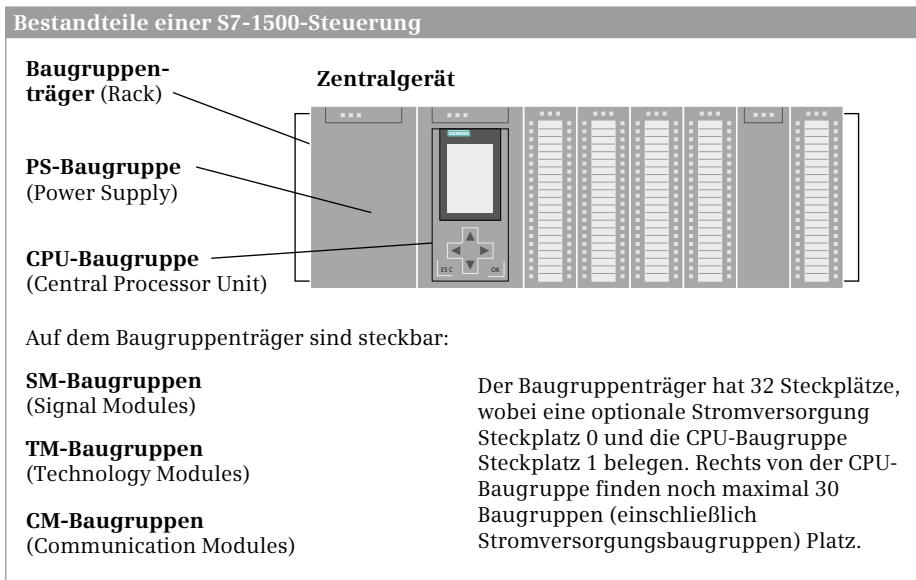
Mit SIMATIC HMI (HMI = Human Machine Interface) wird eine Maschine oder Anlage geführt und deren Funktion beobachtet. Die Geräte können – je nach Variante – über Prozessbilder Bedienfunktionen zur Verfügung stellen, Betriebs- und Störungsmeldungen anzeigen und die Automatisierungsdaten in Form von Rezepturen oder Messwertarchiven verwalten.

SIMATIC NET übernimmt den Datenaustausch über verschiedene Bussysteme zwischen den SIMATIC-Steuerungsgeräten, der dezentralen Peripherie, den Bediengeräten und dem Programmiergerät. Das Programmiergerät kann ein Personal Computer, ein Industrierechner oder ein Notebook mit dem Betriebssystem Microsoft Windows sein.

Mit der Engineering-Software STEP 7 werden die SIMATIC-Komponenten konfiguriert, projiziert, parametrisiert und programmiert. Das TIA Portal (TIA = Totally Integrated Automation) ist das zentrale Werkzeug zum Verwalten der Automatisierungsdaten und der dazugehörigen Editoren in Form eines hierarchisch gegliederten Projekts.

### 1.1.1 Automatisierungsgerät SIMATIC S7-1500

Die wesentlichen Bestandteile eines S7-1500-Automatisierungsgeräts sind im Bild 1.2 dargestellt.



**Bild 1.2** Bestandteile einer S7-1500-Steuerung mit Standard-Controller

Die **CPU-Baugruppe** (Zentralbaugruppe) enthält das Betriebssystem und das Anwenderprogramm. Das Anwenderprogramm steht spannungsausfallsicher auf der in der CPU-Baugruppe steckenden *SIMATIC Memory Card*. Die Bearbeitung des Anwenderprogramms findet im Arbeitsspeicher in der CPU statt. Die auf der CPU-Baugruppe vorhandenen Busschnittstellen stellen die Verbindung zu anderen Automatisierungsgeräten her.

Die Verbindung zur gesteuerten Maschine oder Anlage übernehmen **SM-Baugruppen** (Signalbaugruppen). Diese Ein- und Ausgabebaugruppen gibt es für Digital- und Analogsignale mit verschiedenen Spannungen und Strömen.

Die **TM-Baugruppen** (Technologiebaugruppen) sind signalvorverarbeitende, „intelligente“ Peripheriebaugruppen, die vom Prozess kommende Signale unabhängig von der CPU aufbereiten, verarbeiten und entweder wieder direkt an den Prozess zurückgeben oder an der internen Schnittstelle der CPU zur Verfügung stellen. TM-Baugruppen übernehmen Funktionen, die die CPU meist nicht schnell genug ausführen kann, wie z. B. Impulse zählen.

Die **CM-Baugruppen** (Kommunikationsbaugruppen) ermöglichen einen Datenverkehr, der die Funktionalität der Standard-Schnittstellen auf der CPU-Baugruppe bezüglich Protokolle und Kommunikationsfunktionen überschreitet.

**(System-)Stromversorgungsbaugruppen** sorgen für die benötigten internen Spannungen im Automatisierungsgerät. Bei Bedarf können bis zu drei System-Stromversorgungsbaugruppen im Automatisierungsgerät eingesetzt werden. Lastspannungen bzw. Lastströme werden über externe Laststromversorgungen (Power Modules, PM) bereitgestellt, die auch eine 24V-Primärspannung für System-Stromversorgungsbaugruppen liefern können.

### 1.1.2 Übersicht STEP 7 Professional

STEP 7 ist das zentrale Automatisierungswerkzeug für SIMATIC. STEP 7 benötigt zum Betrieb eine Autorisierung (Lizenz) und läuft auf den jeweils aktuellen Betriebssystemen unter Microsoft Windows. Die Projektierung eines S7-1500-Controllers geschieht mit zwei Ansichten: der Portalansicht und der Projektansicht.

Die **Portalansicht** ist aufgabenorientiert angelegt. Im *Startportal* öffnen Sie ein bestehendes Projekt, erstellen ein neues Projekt oder migrieren ein Projekt. Ein „Projekt“ ist eine Datenstruktur, die alle erforderlichen Programme und Daten für Ihr Automatisierungsvorhaben enthält. Von hier aus sind die wichtigsten Werkzeuge und Funktionen von STEP 7 über weitere Portale erreichbar: Das Portal *Geräte & Netze* zur Konfiguration der Hardware, das Portal *PLC-Programmierung* zum Bearbeiten des Anwenderprogramms, das Portal *Motion & Technology* zum Erstellen von Technologie-Objekten, das Portal *Visualisierung* zum Projektieren von HMI-Systemen und das Portal *Online & Diagnose* für den Online-Betrieb des Programmiergeräts (Bild 1.3).

Die **Projektansicht** bietet eine objektorientierte Ansicht mit mehreren Fenstern, deren Inhalt je nach ausgeführter Tätigkeit wechselt (Bild 1.4). In der *Gerätekonfiguration* steht der Arbeitsbereich mit dem zu konfigurierenden Gerät im Mittel-

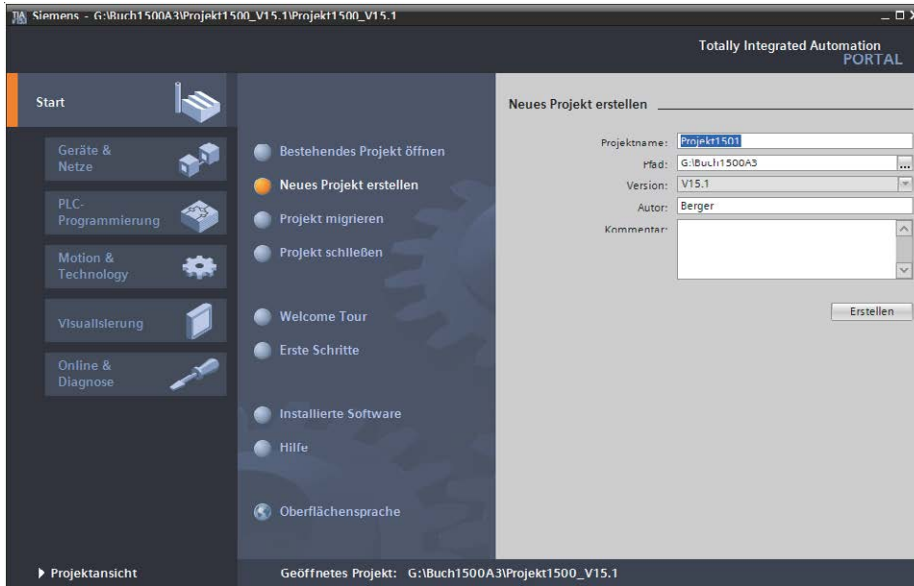


Bild 1.3 Werkzeug im Startportal von STEP 7 Professional V15.1

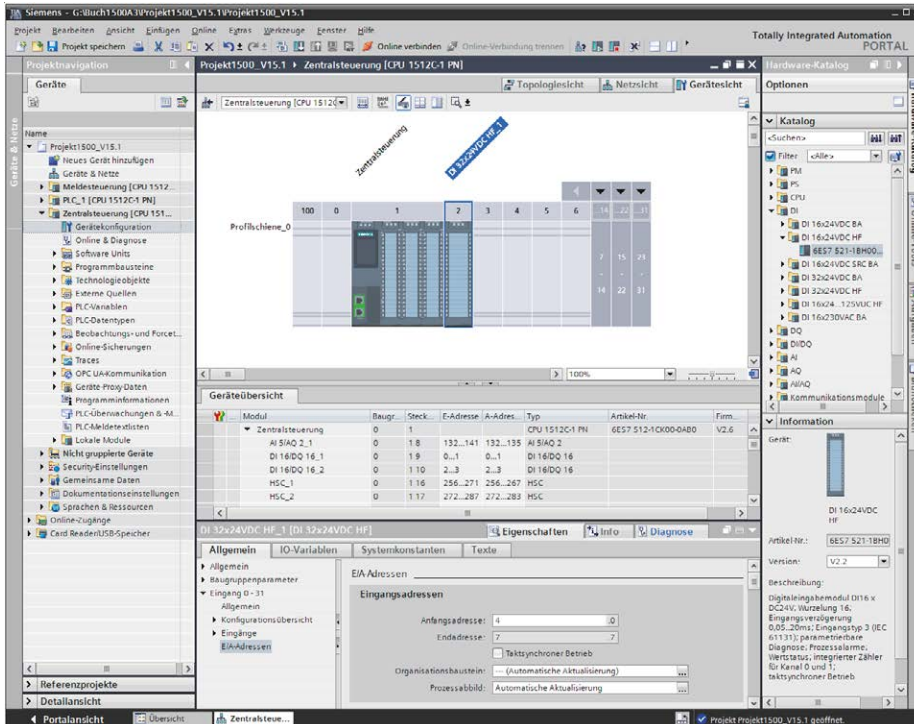


Bild 1.4 Beispiel für die Projektansicht: Arbeitsbereich der Gerätekonfiguration

punkt. In der Gerätesicht wird der Baugruppenträger mit den bereits platzierten Baugruppen gezeigt. Ein weiteres Fenster – das Inspektorfenster – enthält die Eigenschaften der selektierten Baugruppe, das Aufgaben-Fenster (Task Card) gibt Unterstützung durch den Hardware-Katalog mit den zur Verfügung stehenden Baugruppen. Die Netzsicht gestattet die Vernetzung zwischen PLC- und HMI-Stationen.

Bei der *PLC-Programmierung* bearbeiten Sie im Arbeitsbereich den ausgewählten Baustein. Wiederum sehen Sie im Inspektorfenster die Eigenschaften des markierten Objekts und können diese einstellen. Das Aufgabenfenster enthält diesmal den Programmelemente-Katalog mit den zur Verfügung stehenden Programmelementen und Anweisungen. Ähnliches gilt für die Bearbeitung der PLC-Variablen oder beim Online-Programmtest durch Beobachtungstabellen.

Und stets haben Sie die *Projektnavigation* im Blickfeld. Sie enthält alle Objekte des STEP-7-Projekts. So können Sie jederzeit ein Objekt, beispielsweise einen Programmbaustein oder eine Beobachtungstabelle, auswählen und dieses Objekt mit den entsprechenden Editoren, die beim Öffnen des Objekts automatisch starten, bearbeiten.

Zusätzlich gibt es eine **Bibliotheksansicht**, mit der die Elemente der Projektbibliothek und der geöffneten globalen Bibliotheken bearbeitet werden können.

### 1.1.3 Verschiedene Programmiersprachen

Als Programmiersprache für das Anwenderprogramm können Sie zwischen Kontaktplan (KOP), Funktionsplan (FUP), Structured Control Language (SCL), Anweisungsliste (AWL) und Ablaufsteuerung (GRAPH) wählen (Bild 1.5).

Mit **Kontaktplan** programmieren Sie die Steuerungsaufgabe angelehnt an den Stromlaufplan. Die Verknüpfung der binären Signalzustände wird durch die serielle oder parallele Anordnung von Kontakten und Spulen dargestellt. Komplexe Funktionen, wie beispielsweise die arithmetischen Funktionen, werden mit Boxen dargestellt, die Sie wie Kontakte oder Spulen im Kontaktplan anordnen.

Mit **Funktionsplan** programmieren Sie die Steuerungsaufgabe angelehnt an elektronische Schaltkreissysteme. Binäre Verknüpfungen werden durch Verschaltung von UND- und ODER-Funktionen realisiert und mit Speicher-Boxen abgeschlossen. Komplexe Boxen übernehmen die Verknüpfung digitaler Variablen, beispielsweise bei den arithmetischen Funktionen.

**Structured Control Language** eignet sich besonders für die Programmierung von komplexen Algorithmen oder für Aufgabenstellungen aus dem Bereich der Datenverwaltung. Das Programm besteht aus SCL-Anweisungen, die beispielsweise Wertzuweisungen, Vergleiche oder Kontrollanweisungen sein können.

Mit **Anweisungsliste** programmieren Sie die Steuerungsaufgabe durch eine Folge von Anweisungen. Jede AWL-Anweisung enthält eine Vorschrift, was zu tun ist, und eventuell einen Operand, mit dem die Operation ausgeführt wird. AWL ist gleichermaßen geeignet für binäre und digitale Verknüpfungen wie für die Programmierung komplexer Steuerungsaufgaben.

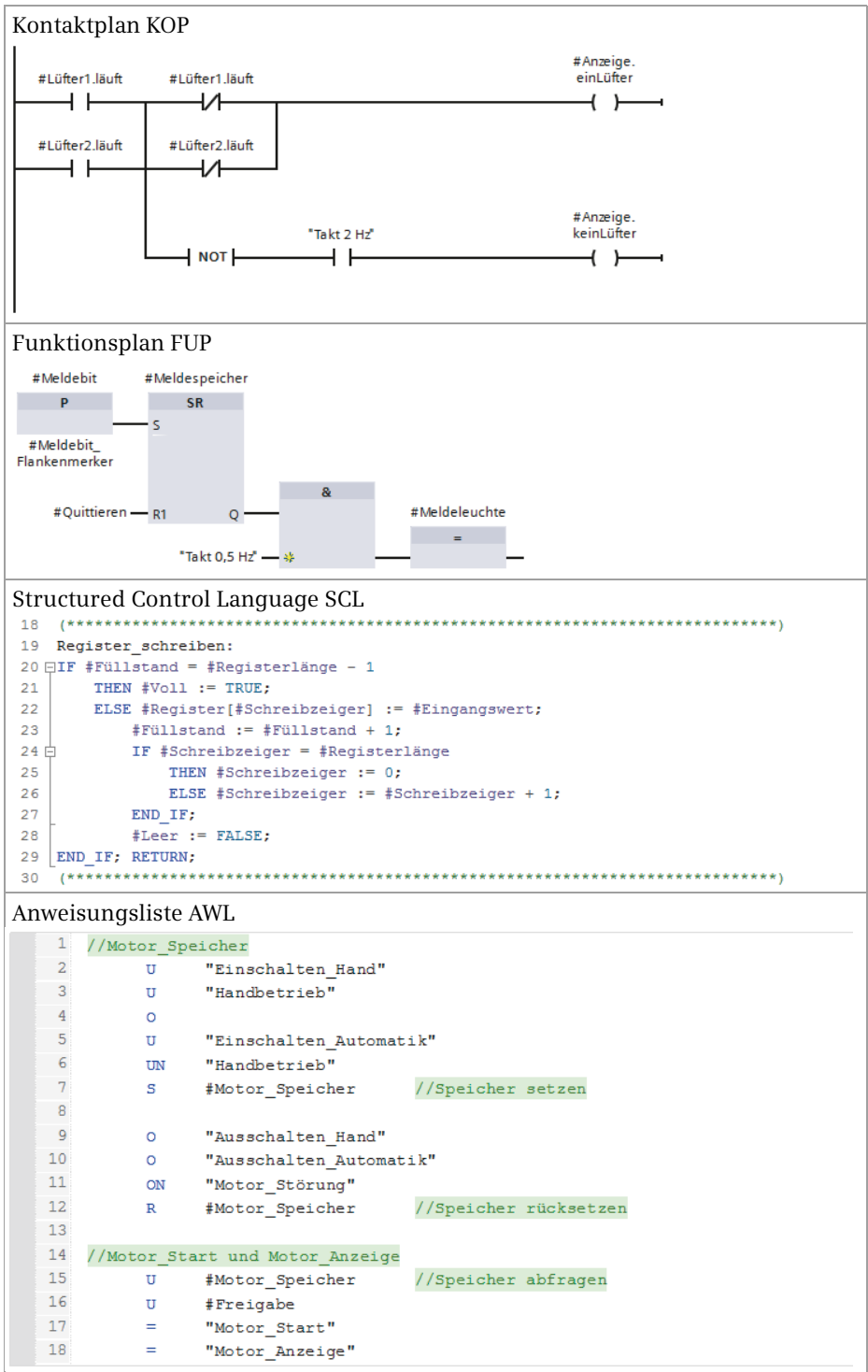
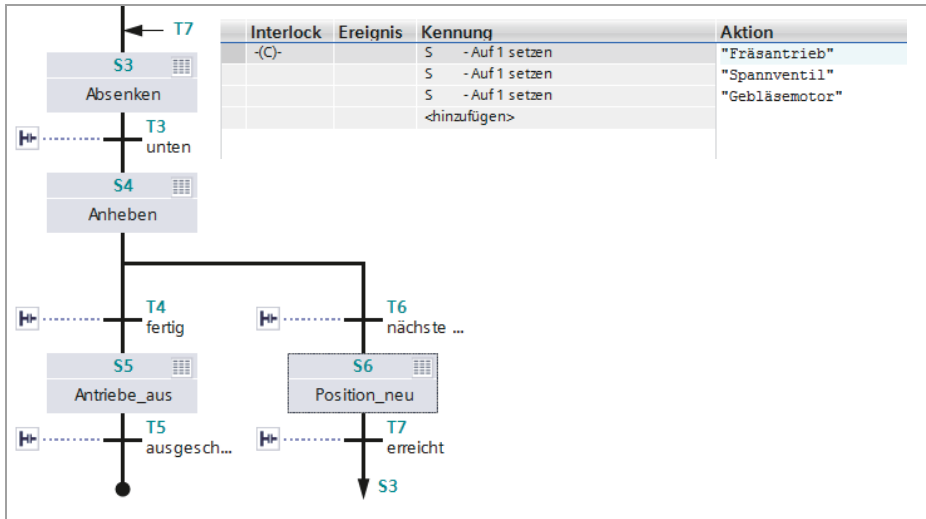


Bild 1.5 Beispiele für die Darstellung in KOP, FUP, SCL und AWL

Mit **GRAPH** programmieren Sie eine Steuerungsaufgabe als Ablaufsteuerung, in der die sequenzielle Folge von Aktionen vorherrscht. Die einzelnen Schritte und Verzweigungen werden durch Weiterschaltbedingungen freigegeben, die mit KOP oder FUP programmiert werden können (Bild 1.6).



**Bild 1.6** Beispiel für eine GRAPH-Ablaufkette und eine Schrittprojektierung

### 1.1.4 Bearbeitung des Anwenderprogramms

Nach dem Einschalten der Versorgungsspannung prüft der Steuerungsprozessor die vorhandene Hardware und parametrisiert die Baugruppen. Anschließend wird einmalig – falls vorhanden – ein Anlaufprogramm bearbeitet. Das Anlaufprogramm gehört zum Anwenderprogramm, das Sie programmieren. Im Anlaufprogramm können beispielsweise Baugruppen initialisiert werden.

Das Anwenderprogramm ist in der Regel in einzelne Abschnitte – „Bausteine“ genannt – eingeteilt. Die Organisationsbausteine (OB) sind die Schnittstelle zwischen Betriebssystem und Anwenderprogramm. Das Betriebssystem ruft zu bestimmten Ereignissen einen Organisationsbaustein auf, in dem dann das Anwenderprogramm bearbeitet wird (Bild 1.7).

Zur Strukturierung des Programms stehen Funktionsbausteine (FB) und Funktionen (FC) zur Verfügung. Funktionsbausteine haben einen Speicher, in dem lokale Variablen dauerhaft abgelegt sind. Funktionen haben diesen Speicher nicht.

Für den Aufruf (das Starten der Bearbeitung) von Funktionsbausteinen und Funktionen gibt es Programmanweisungen. Jeder Bausteinaufruf kann mit Ein- und Ausgängen – „Bausteinparameter“ genannt – versehen werden. Beim Aufruf können Variablen übergeben werden, mit denen das Programm im Baustein arbeiten soll. Auf diese Weise kann ein Baustein mit einer gewissen Funktion (z. B. der Aus-