

SIEMENS

Hans Berger

Automatisieren mit SIMATIC S7-1500

Projektieren, Programmieren und Testen
mit STEP 7 Professional

2. Auflage

Berger Automatisieren mit SIMATIC S7-1500

Automatisieren mit SIMATIC S7-1500

Projektieren, Programmieren und Testen
mit STEP 7 Professional

von Hans Berger

2., wesentlich überarbeitete
und erweiterte Auflage, 2017

Publicis Publishing

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-books.de

Lektorat: Dr. Gerhard Seifudem, gerhard.seifudem@publicispixelpark.de

Print ISBN 978-3-89578-451-4

ePDF ISBN 978-3-89578-939-7

2., wesentlich überarbeitete und erweiterte Auflage, 2017

Herausgeber Siemens Aktiengesellschaft, Berlin und München

Verlag: Publicis Publishing, Erlangen

© 2017 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 V2.0 mit der Engineeringsoftware STEP 7 Professional V14 und der Simulationssoftware S7-PLCSIM V14.

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 V14.
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

1 Einführung	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 V14	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	48
1.3.5 Mit Bibliotheken arbeiten	48
1.3.6 Mehrsprachige Projekte	52
2 Automatisierungssystem SIMATIC S7-1500	54
2.1 Komponenten einer S7-1500-Station	54
2.2 CPU-Baugruppen S7-1500	56
2.2.1 CPU-Varianten	56
2.2.2 Bedien- und Anzeigeelemente	60
2.2.3 SIMATIC Memory Card	61
2.2.4 Busschnittstellen	62
2.3 Signalbaugruppen	63
2.3.1 Digitaleingabebaugruppen	63
2.3.2 Digitalausgabebaugruppen	65
2.3.3 Digitalein-/ausgabebaugruppen	66
2.3.4 Analogeingabebaugruppen	66
2.3.5 Analogausgabebaugruppe	66
2.3.6 Analogein-/ausgabebaugruppen	67
2.4 Technologiebaugruppen	68

2.5 Kommunikationsbaugruppen	69
2.6 Weitere Baugruppen	70
2.6.1 Systemstromversorgungsbaugruppen	70
2.6.2 Laststromversorgungsbaugruppen	70
2.7 SIPLUS S7-1500	71
3 Gerätekonfiguration	72
3.1 Einführung	72
3.2 Eine Station konfigurieren	74
3.2.1 Eine PLC-Station hinzufügen	74
3.2.2 Eine Baugruppe hinzufügen	74
3.3 Baugruppen parametrieren	76
3.3.1 Die CPU-Eigenschaften parametrieren	76
3.3.2 Baugruppen adressieren	80
3.3.3 Signalbaugruppen parametrieren	82
3.3.4 Eine Konfigurationssteuerung projektieren	84
3.4 Eine Vernetzung projektieren	87
3.4.1 Einführung	87
3.4.2 Eine Station vernetzen	88
3.4.3 Teilnehmeradressen in einem Subnetz	90
3.4.4 Kommunikationsdienste und Verbindungstypen	91
3.4.5 Eine Verbindung projektieren	92
3.4.6 Ein PROFINET-Subnetz projektieren	94
3.4.7 Ein PROFIBUS-Subnetz projektieren	98
4 Variablen, Adressierung und Datentypen	101
4.1 Operanden und Variablen	101
4.1.1 Einführung, Übersicht	101
4.1.2 Operandenbereiche Eingänge und Ausgänge	103
4.1.3 Operandenbereich Merker	104
4.1.4 Operandenbereich Daten	106
4.1.5 Operandenbereich temporäre Lokaldaten	107
4.2 Adressierung von Variablen und Konstanten	108
4.2.1 Signalweg	108
4.2.2 Absolute Adressierung	109
4.2.3 Symbolische Adressierung	112
4.2.4 Adressierung eines Variablenteils	114
4.2.5 Adressierung einer Konstanten	114
4.2.6 Indirekte Adressierung	115
4.3 Adressierung von Hardware-Objekten	123
4.4 Allgemeines zu Datentypen	123
4.4.1 Übersicht der Datentypen	123
4.4.2 Implizite Datentypkonvertierung	125
4.4.3 Variablen überlagern (Datentypsichten)	128
4.5 Elementare Datentypen	130

4.5.1	Bitfolge-Datentypen BOOL, BYTE, WORD, DWORD und LWORD	130
4.5.2	BCD-Zahlen BCD16 und BCD32	132
4.5.3	Datentypen CHAR und WCHAR	133
4.5.4	Festpunkt-Datentypen ohne Vorzeichen USINT, UINT, UDINT, ULINT	133
4.5.5	Festpunkt-Datentypen mit Vorzeichen SINT, INT, DINT und LINT	134
4.5.6	Gleitpunkt-Datentypen REAL und LREAL	135
4.5.7	Datentypen für Zeitdauern	137
4.5.8	Datentypen für Zeitpunkte	139
4.6	Strukturierte Datentypen	140
4.6.1	Datum und Uhrzeit DATE_AND_TIME (DT)	140
4.6.2	Datum und Uhrzeit DATE_AND_LTIME (DTL)	142
4.6.3	Zeichenketten STRING und WSTRING	142
4.6.4	Datentyp ARRAY	143
4.6.5	Datentyp STRUCT	147
4.7	Parametertypen	149
4.7.1	Übersicht	149
4.7.2	Parametertypen BLOCK_FC und BLOCK_FB (AWL)	150
4.7.3	Parametertyp DB_ANY	150
4.7.4	Parametertyp VOID	151
4.7.5	Parametertypen POINTER, ANY und VARIANT	151
4.7.6	Parametertyp ARRAY[*]	152
4.8	Zeiger	152
4.8.1	Einführung	152
4.8.2	Bereichszeiger	153
4.8.3	DB-Zeiger	153
4.8.4	ANY-Zeiger	155
4.9	PLC-Datentypen	155
4.9.1	Einen PLC-Datentyp programmieren	155
4.9.2	Einen PLC-Datentyp anwenden	156
4.9.3	PLC-Datentypen vergleichen	158
4.9.4	Einen PLC-Datentyp umnummerieren	159
4.10	Systemdatentypen	159
4.10.1	Systemdatentypen für Zeitfunktionen	159
4.10.2	Systemdatentypen für Zählfunktionen	161
4.10.3	Startinformation	162
4.11	Hardware-Datentypen	163
5	Anwenderprogramm bearbeiten	164
5.1	Betriebszustände	164
5.1.1	Betriebszustand STOP	165
5.1.2	Betriebszustand ANLAUF	166
5.1.3	Betriebszustand RUN	168
5.1.4	Remanenzverhalten von Operanden	169
5.2	Anwenderprogramm erstellen	170
5.2.1	Programmbearbeitung	170
5.2.2	Strukturierung des Anwenderprogramms	172

5.2.3 Bausteinararten	175
5.2.4 Bausteineigenschaften	177
5.2.5 Bausteinschnittstelle	177
5.2.6 Einen Codebaustein aufrufen	183
5.2.7 Programmierempfehlungen	186
5.3 Anlaufprogramm	193
5.3.1 Organisationsbausteine für das Anlaufprogramm	193
5.3.2 Remanente Daten zurücksetzen	195
5.3.3 Baugruppenadresse ermitteln	195
5.3.4 Baugruppen parametrieren	198
5.4 Hauptprogramm	202
5.4.1 Organisationsbausteine für das Hauptprogramm	202
5.4.2 Prozessabbild-Aktualisierung	203
5.4.3 Zykluszeit	207
5.4.4 Reaktionszeit	210
5.4.5 Programm stoppen und verzögern	211
5.4.6 Uhrzeit	212
5.4.7 Systemzeit lesen	217
5.4.8 Betriebsstundenzähler	217
5.5 Alarmbearbeitung	219
5.5.1 Einführung zur Alarmbearbeitung	219
5.5.2 Uhrzeitalarme	223
5.5.3 Verzögerungsalarme	227
5.5.4 Weckalarme	231
5.5.5 Prozessalarme	235
5.5.6 Alarmer zur Laufzeit zuordnen	237
5.5.7 Alarmzusatzinformation lesen	238
5.6 Fehlerbehandlung, Diagnose	240
5.6.1 Fehlerursachen und Fehlerreaktionen	240
5.6.2 Lokale Fehlerbehandlung	240
5.6.3 Globale Fehlerbehandlung (Synchronfehler)	245
5.6.4 Synchronfehlerbearbeitung sperren und freigeben	247
5.6.5 Asynchronfehler	250
5.6.6 Alarmer und Asynchronfehler sperren, verzögern und freigeben	254
5.7 Diagnose im Anwenderprogramm	256
5.7.1 Diagnosealarm	256
5.7.2 Startinformation lesen	258
5.7.3 Laufzeit-Informationen lesen	259
5.7.4 Diagnosefunktionen im Anwenderprogramm	260
5.8 Meldungen projektieren	267
5.8.1 Einführung	267
5.8.2 Meldungen nach dem Meldenummerverfahren projektieren	271
5.8.3 Systembausteine für Meldungen	276
5.8.4 Anwender-Diagnosemeldung erzeugen	279
5.8.5 CPU-Meldungsanzeige	280

6 Programmeditor	283
6.1 Einführung	283
6.2 PLC-Variablen-tabelle	284
6.2.1 PLC-Variablen-tabelle anlegen und bearbeiten	284
6.2.2 PLC-Variablen definieren und bearbeiten	284
6.2.3 PLC-Variablen-tabellen vergleichen	287
6.2.4 PLC-Variablen-tabelle exportieren und importieren	288
6.2.5 Konstanten-tabellen	289
6.3 Einen Codebaustein programmieren	289
6.3.1 Einen neuen Codebaustein anlegen	289
6.3.2 Arbeitsbereich des Programmeditors für Codebausteine	291
6.3.3 Bausteineigenschaften für Codebausteine festlegen	293
6.3.4 Einen Baustein schützen	296
6.3.5 Bausteinschnittstelle programmieren	297
6.3.6 Allgemeines Vorgehen beim Programmieren der Steuerungsfunktion	300
6.3.7 Steuerungsfunktion mit KOP und FUP programmieren	304
6.3.8 Steuerungsfunktion mit SCL programmieren	308
6.3.9 Steuerungsfunktion mit AWL programmieren	312
6.3.10 Steuerungsfunktion mit GRAPH programmieren	314
6.4 Einen Datenbaustein programmieren	315
6.4.1 Einen neuen Datenbaustein anlegen	315
6.4.2 Arbeitsbereich des Programmeditors für Datenbausteine	316
6.4.3 Bausteineigenschaften für Datenbausteine festlegen	317
6.4.4 Datenvariablen deklarieren	320
6.4.5 Datenvariablen in Globaldatenbausteinen eingeben	322
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	327
6.7 Programminformationen	330
6.7.1 Querverweisliste	330
6.7.2 Belegungsplan	331
6.7.3 Aufrufstruktur	333
6.7.4 Abhängigkeitsstruktur	334
6.7.5 Konsistenzprüfung	335
6.7.6 Speicherauslastung der CPU	336
7 Kontaktplan KOP	338
7.1 Einführung	338
7.1.1 Programmieren mit KOP allgemein	338
7.1.2 Programmelemente des Kontaktplans	340
7.2 Binäre Verknüpfungen mit KOP programmieren	340
7.2.1 Schließerkontakt und Öffnerkontakt	341
7.2.2 Reihen- und Parallelschaltung von Kontakten	343

7.2.3	T-Abzweig, offener Parallelzweig	344
7.2.4	Verknüpfungsergebnis negieren im Kontaktplan	345
7.2.5	Flankenauswertung einer Binärvariablen im Kontaktplan	345
7.2.6	Gültigkeitsprüfung einer Gleitpunktvariablen im Kontaktplan	346
7.2.7	Vergleich-Kontakte	347
7.3	Speicherfunktionen mit KOP programmieren	347
7.3.1	Einfache und negierende Spule	348
7.3.2	Setzen- und Rücksetzen-Spule	349
7.3.3	Speicherndes Verhalten durch Selbsthaltung	350
7.3.4	Flankenauswertung mit Impulsausgabe im Kontaktplan	350
7.3.5	Mehrfaches Setzen und Rücksetzen (Bitfeld füllen) im Kontaktplan	351
7.3.6	Spulen mit Zeitverhalten	352
7.4	Q-Boxen mit KOP programmieren	353
7.4.1	Speicher-Boxen im Kontaktplan	354
7.4.2	Flankenauswertung des Stromflusses	354
7.4.3	Zeitfunktionen im Kontaktplan	355
7.4.4	Zählfunktionen im Kontaktplan	357
7.5	EN/ENO-Boxen mit KOP programmieren	358
7.5.1	Flankenauswertung mit einer EN/ENO-Box	358
7.5.2	Übertragungsfunktionen im Kontaktplan	358
7.5.3	Arithmetische Funktionen im Kontaktplan	360
7.5.4	Mathematische Funktionen im Kontaktplan	361
7.5.5	Konvertierungsfunktionen im Kontaktplan	362
7.5.6	Schiebefunktionen im Kontaktplan	363
7.5.7	Logikfunktionen im Kontaktplan	363
7.5.8	Funktionen für Zeichenketten im Kontaktplan	364
7.6	VARIANT-Funktionen mit KOP programmieren	365
7.7	Programmsteuerung mit KOP	367
7.7.1	Sprungfunktionen im Kontaktplan	367
7.7.2	Bausteinende-Funktion im Kontaktplan	370
7.7.3	Bausteinaufruf-Funktionen im Kontaktplan	371
8	Funktionsplan FUP	373
8.1	Einführung	373
8.1.1	Programmieren mit FUP allgemein	373
8.1.2	Programmelemente des Funktionsplans	375
8.2	Binäre Verknüpfungen mit FUP programmieren	375
8.2.1	Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	376
8.2.2	Eine binäre Verknüpfung im Funktionsplan programmieren	378
8.2.3	UND-Funktion im Funktionsplan	379
8.2.4	ODER-Funktion im Funktionsplan	379
8.2.5	Exklusiv-ODER-Funktion im Funktionsplan	379
8.2.6	Kombinierte binäre Verknüpfungen, Verknüpfungsergebnis negieren	380
8.2.7	T-Abzweig im Funktionsplan	381
8.2.8	Flankenauswertung von Binärvariablen im Funktionsplan	381
8.2.9	Gültigkeitsprüfung von Gleitpunktzahlen im Funktionsplan	382

8.2.10 Vergleichsfunktionen im Funktionsplan	383
8.3 Standard-Boxen mit FUP programmieren	384
8.3.1 Zuweisung und negierende Zuweisung	384
8.3.2 Setzen- und Rücksetzen-Box	385
8.3.3 Flankenbewertung mit Impulsausgabe im Funktionsplan	386
8.3.4 Mehrfaches Setzen und Rücksetzen (Bitfeld füllen) im Funktionsplan	387
8.3.5 Standard-Boxen mit Zeitverhalten	387
8.4 Q-Boxen mit FUP programmieren	389
8.4.1 Speicher-Boxen im Funktionsplan	389
8.4.2 Flankenbewertung des Verknüpfungsergebnisses im Funktionsplan	390
8.4.3 Zeitfunktionen im Funktionsplan	391
8.4.4 Zählfunktionen im Funktionsplan	392
8.5 EN/ENO-Boxen mit FUP programmieren	393
8.5.1 Flankenbewertung mit einer EN/ENO-Box	393
8.5.2 Übertragungsfunktionen im Funktionsplan	395
8.5.3 Arithmetische Funktionen im Funktionsplan	395
8.5.4 Mathematische Funktionen im Funktionsplan	396
8.5.5 Konvertierungsfunktionen im Funktionsplan	397
8.5.6 Schiebefunktionen im Funktionsplan	398
8.5.7 Logikfunktionen im Funktionsplan	399
8.5.8 Funktionen für Zeichenketten im Funktionsplan	400
8.6 VARIANT-Funktionen mit FUP programmieren	401
8.7 Programmsteuerung mit FUP	402
8.7.1 Sprungfunktionen im Funktionsplan	403
8.7.2 Bausteinende-Funktion im Funktionsplan	406
8.7.3 Bausteinaufruf-Funktionen im Funktionsplan	407
9 Structured Control Language SCL	409
9.1 Einführung	409
9.1.1 Programmieren mit SCL allgemein	409
9.1.2 SCL-Anweisungen und Operatoren	411
9.2 Binäre Verknüpfungen mit SCL programmieren	413
9.2.1 Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	413
9.2.2 UND-Funktion in SCL	414
9.2.3 ODER-Funktion in SCL	414
9.2.4 Exklusiv-ODER-Funktion in SCL	415
9.2.5 Kombinierte binäre Verknüpfungen in SCL	415
9.2.6 Verknüpfungsergebnis negieren in SCL	416
9.3 Speicherfunktionen mit SCL programmieren	417
9.3.1 Wertzuweisung einer Binärvariablen	417
9.3.2 Setzen und Rücksetzen in SCL	417
9.3.3 Flankenbewertung in SCL	418
9.4 Zeit- und Zählfunktionen mit SCL programmieren	418
9.4.1 Zeitfunktionen in SCL	418
9.4.2 Zählfunktionen in SCL	419
9.5 Digitalfunktionen mit SCL programmieren	420

9.5.1 Übertragungsfunktion, Wertzuweisung einer Digitalvariablen	420
9.5.2 Vergleichsfunktionen in SCL	421
9.5.3 Arithmetische Funktionen in SCL	422
9.5.4 Mathematische Funktionen in SCL	423
9.5.5 Konvertierungsfunktionen in SCL	424
9.5.6 Schiebefunktionen in SCL	425
9.5.7 Wortverknüpfungen, logischer Ausdruck in SCL	426
9.5.8 Funktionen für Zeichenketten in SCL	427
9.6 VARIANT-Funktionen mit SCL programmieren	427
9.7 Programmsteuerung mit SCL	430
9.7.1 Kontrollanweisungen	430
9.7.2 Bausteinendefinition bei SCL	439
9.7.3 Aufruf einer Funktion (FC) bei SCL	439
9.7.4 Aufruf eines Funktionsbausteins (FB) bei SCL	440
9.7.5 Versorgung von Parametern	442
10 Anweisungsliste AWL	443
10.1 Einführung	443
10.1.1 Programmieren mit AWL allgemein	443
10.1.2 Aufbau einer AWL-Anweisung	445
10.1.3 Eine AWL-Anweisung eingeben	445
10.1.4 Adressierung von 64-Bit-Variablen	447
10.1.5 AWL-Netzwerke in KOP- und FUP-Bausteinen	447
10.2 Binäre Verknüpfungen mit AWL programmieren	448
10.2.1 Bearbeitung einer binären Verknüpfung, Verknüpfungsschritt	449
10.2.2 Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	450
10.2.3 UND-Funktion in der Anweisungsliste	451
10.2.4 ODER-Funktion in der Anweisungsliste	451
10.2.5 Exklusiv-ODER-Funktion in der Anweisungsliste	451
10.2.6 Kombinierte binäre Verknüpfungen in der Anweisungsliste	452
10.2.7 Verknüpfungsergebnis steuern	455
10.3 Speicherfunktionen mit AWL programmieren	455
10.3.1 Zuweisung in der Anweisungsliste	456
10.3.2 Setzen und Rücksetzen in der Anweisungsliste	456
10.3.3 Flankenbewertung in der Anweisungsliste	457
10.4 Zeit- und Zählfunktionen mit AWL programmieren	458
10.4.1 Zeitfunktionen in der Anweisungsliste	458
10.4.2 Zählfunktionen in der Anweisungsliste	459
10.5 Digitalfunktionen mit AWL programmieren	460
10.5.1 Übertragungsfunktionen in der Anweisungsliste	460
10.5.2 Vergleichsfunktionen in der Anweisungsliste	464
10.5.3 Arithmetische Funktionen in der Anweisungsliste	467
10.5.4 Mathematische Funktionen in der Anweisungsliste	471
10.5.5 Konvertierungsfunktionen in der Anweisungsliste	472
10.5.6 Schiebefunktionen in der Anweisungsliste	474
10.5.7 Wortverknüpfungen in der Anweisungsliste	477

10.5.8 Funktionen für Zeichenketten in der Anweisungsliste	480
10.6 VARIANT-Funktionen mit AWL programmieren	481
10.7 Programmsteuerung mit AWL	484
10.7.1 Sprungfunktionen in der Anweisungsliste	484
10.7.2 Bausteinende-Funktionen in der Anweisungsliste	486
10.7.3 Bausteinaufruf-Funktion in der Anweisungsliste	486
10.8 Weitere AWL-Funktionen	489
10.8.1 Mit Statusbits arbeiten	490
10.8.2 Akkumulatorfunktionen	495
10.8.3 Arbeiten mit den Datenbausteinregistern	497
10.8.4 Teiladressierung von Datenoperanden	499
10.8.5 Absolute Adressierung von temporären Lokaldaten	500
10.8.6 Arbeiten mit den Adressregistern	501
10.8.7 Speicherindirekte Adressierung	503
10.8.8 Registerindirekte Adressierung	506
10.8.9 Direkter Zugriff auf komplexe Lokalvariablen	510
10.8.10 Nullanweisungen	512
11 Ablaufsteuerung S7-GRAPH	513
11.1 Einführung	513
11.1.1 Was ist eine Ablaufsteuerung?	513
11.1.2 Eigenschaften einer Ablaufsteuerung	513
11.2 Elemente einer Ablaufsteuerung	515
11.2.1 Schritte und Transitionen	515
11.2.2 Sprünge in einer Ablaufsteuerung	517
11.2.3 Verzweigungen einer Ablaufkette	517
11.2.4 Permanente Anweisungen	519
11.2.5 Schritt- und Transitionsfunktionen	519
11.2.6 Bearbeitung von Bedingungen	523
11.2.7 Bearbeitung von Aktionen	524
11.3 Eine Ablaufsteuerung projektieren	530
11.3.1 Allgemeines Vorgehen bei der Projektierung	530
11.3.2 Den GRAPH-Funktionsbaustein programmieren	531
11.3.3 Projektieren der Kettenstruktur	532
11.3.4 Schritte und Transitionen programmieren	535
11.3.5 Permanente Anweisungen programmieren	537
11.3.6 Meldungen projektieren	537
11.3.7 Attribute des GRAPH-Funktionsbausteins	537
11.3.8 Den GRAPH-Funktionsbaustein aufrufen	539
11.4 Ablaufsteuerung testen	540
11.4.1 GRAPH-Funktionsbaustein laden	540
11.4.2 Einstellungen zum Programmtest	541
11.4.3 Betriebsarten verwenden	542
11.4.4 Ablaufkette synchronisieren	542
11.4.5 Mit Programmstatus testen	543

12 Basisfunktionen	546
12.1 Binäre Verknüpfungen	546
12.1.1 Einführung	546
12.1.2 Arbeiten mit Binärsignalen	547
12.1.3 UND-Funktion, Reihenschaltung	551
12.1.4 ODER-Funktion, Parallelschaltung	551
12.1.5 Exklusiv-ODER-Funktion, Antivalenzfunktion	552
12.1.6 Verknüpfungsergebnis negieren, NOT-Kontakt	553
12.2 Speicherfunktionen	554
12.2.1 Einführung	554
12.2.2 Einfache und negierende Spule, Zuweisung	554
12.2.3 Einzelnes Setzen und Rücksetzen	555
12.2.4 Mehrfaches Setzen und Rücksetzen	556
12.2.5 Vorrangiges Setzen und Rücksetzen, Speicher-Boxen	557
12.3 Flankenbewertung	559
12.3.1 Funktionsweise einer Flankenbewertung	559
12.3.2 Flankenbewertung einer Binärvariablen (KOP, FUP)	561
12.3.3 Flankenbewertung mit Impulsausgabe (KOP, FUP)	561
12.3.4 Flankenbewertung mit einer Q-Box (KOP, FUP)	562
12.3.5 Flankenbewertung mit einer EN/ENO-Box (KOP, FUP)	563
12.3.6 Flankenbewertung bei SCL	564
12.3.7 Flankenbewertung bei AWL	567
12.4 Zeitfunktionen	568
12.4.1 Eine Zeitfunktion hinzufügen	568
12.4.2 Impulsbildung TP	569
12.4.3 Einschaltverzögerung TON	570
12.4.4 Ausschaltverzögerung TOF	571
12.4.5 Akkumulierende Einschaltverzögerung TONR	572
12.4.6 Eine Zeitfunktion mit einer Zeitdauer laden	573
12.4.7 Eine Zeitfunktion zurücksetzen	574
12.5 Zählfunktionen	574
12.5.1 Eine Zählfunktion einfügen	575
12.5.2 Vorwärtszähler CTU	575
12.5.3 Rückwärtszähler CTD	576
12.5.4 Vorwärts-Rückwärtszähler CTUD	577
13 Digitalfunktionen	580
13.1 Übertragungsfunktionen	581
13.1.1 Variable kopieren, MOVE -und S_MOVE-Box bei KOP und FUP	581
13.1.2 Wertzuweisungen bei SCL	583
13.1.3 Laden und Transferieren bei AWL	586
13.1.4 Variable kopieren, MOVE und S_MOVE bei AWL	587
13.1.5 VARIANT-Variable lesen und schreiben	588
13.1.6 Datenbereich kopieren mit MOVE_BLK_VARIANT	589
13.1.7 Datenbereich kopieren mit MOVE_BLK und UMOVE_BLK	591
13.1.8 Datenbereich füllen mit FILL_BLK und UFILL_BLK	593

13.1.9	Variable von und zu einen BYTE-Feld übertragen	593
13.1.10	Lesen und Schreiben mit PEEK und POKE (SCL, AWL)	597
13.1.11	Byte-Reihenfolge ändern mit SWAP	597
13.1.12	Bereichsgrenzen ermitteln	599
13.2	Vergleichsfunktionen	599
13.2.1	Ausführung der Vergleichsfunktion	600
13.2.2	Bereichsvergleich bei KOP und FUP	603
13.2.3	Gleitpunkt-Variable testen, OK-Kontakt, OK-Box	603
13.2.4	VARIANT-Zeiger testen bei KOP, FUP und AWL	605
13.2.5	VARIANT-Zeiger testen bei SCL	606
13.3	Arithmetische Funktionen	609
13.3.1	Arithmetische Funktionen für Zahlenwerte	609
13.3.2	Arithmetische Funktionen für Zeitwerte	611
13.3.3	Dekrementieren und Inkrementieren	613
13.4	Mathematische Funktionen	614
13.4.1	Allgemeine Funktionsbeschreibung	614
13.4.2	Winkelfunktionen SIN, COS, TAN	615
13.4.3	Arcusfunktionen ASIN, ACOS, ATAN	615
13.4.4	Quadrat bilden und Quadratwurzel ziehen	616
13.4.5	Logarithmus und Potenz	616
13.4.6	Nachkommastellen extrahieren, Absolutwert und Negation bilden	618
13.4.7	Rechnen mit der CALCULATE-Box bei KOP und FUP	619
13.5	Konvertierungsfunktionen	621
13.5.1	Konvertierungsfunktionen CONV, S_CONV und T_CONV	622
13.5.2	Konvertierung von DB_ANY	628
13.5.3	Konvertierungsfunktionen für Gleitpunktzahlen	629
13.5.4	Konvertierungsfunktionen STRG_TO_CHARS und CHARS_TO_STRG	630
13.5.5	Konvertierungsfunktionen STRG_VAL und VAL_STRG	633
13.5.6	Konvertierungsfunktionen ATH und HTA	635
13.5.7	Konvertierungsfunktionen SCALE_X und NORM_X	637
13.6	Schiebefunktionen	639
13.6.1	Allgemeine Funktionsbeschreibung	639
13.6.2	Rechts schieben SHR	640
13.6.3	Links schieben SHL	640
13.6.4	Rechts rotieren ROR	640
13.6.5	Links rotieren ROL	640
13.7	Logikfunktionen	641
13.7.1	Wortverknüpfungen	641
13.7.2	Invertieren, Einerkomplement bilden	643
13.7.3	Codierfunktionen DECO und ENCO	644
13.7.4	Auswahlfunktionen SEL, MUX und DEMUX	645
13.7.5	Minimumauswahl MIN, Maximumauswahl MAX	646
13.7.6	Begrenzer LIMIT	646
13.8	Zeichenketten bearbeiten	649
13.9	Symbolnamen lesen	658

14 Programmsteuerung	663
14.1 Sprungfunktionen	664
14.1.1 Einführung	664
14.1.2 Absoluter Sprung	664
14.1.3 Bedingter Sprung	666
14.1.4 Sprungliste	668
14.1.5 Sprungverteiler	670
14.1.6 Schleifensprung	670
14.2 Bausteinende-Funktionen	672
14.2.1 Bausteinende-Funktion RET (KOP und FUP)	672
14.2.2 RETURN-Anweisung (SCL)	674
14.2.3 Bausteinende-Funktionen BEB, BEA und BE (AWL)	674
14.3 Aufruf von Codebausteinen	674
14.3.1 Einführung	674
14.3.2 Aufruf einer Funktion FC	675
14.3.3 Aufruf eines Funktionsbausteins FB	675
14.3.4 Asynchron arbeitende Systembausteine	678
14.3.5 EN/ENO-Mechanismus	678
14.4 Arbeiten mit Bausteinen	682
14.4.1 Bausteine mit optimiertem und Standardzugriff	682
14.4.2 Datentypen der lokalen Variablen	684
14.4.3 Bausteinparameter adressieren	686
14.4.4 Bausteinparameter versorgen	689
14.4.5 Übergabe von Bausteinparametern	691
14.5 Datenbausteinfunktionen	694
14.5.1 Datenbausteinattribute lesen	694
14.5.2 Ladespeicher lesen und schreiben	695
14.5.3 ARRAY-Datenbausteine	697
14.5.4 Systembausteine für den Zugriff auf ARRAY-Datenbausteine	698
14.5.5 CPU-Datenbausteine	700
15 Online-Betrieb, Diagnose und Test	704
15.1 PLC-Station online verbinden	705
15.1.1 Programmiergerät an die PLC-Station anschließen	705
15.1.2 Die CPU urlöschen	707
15.1.3 Den Auslieferungszustand wiederherstellen	708
15.2 Projektdaten übertragen	709
15.2.1 Die Projektdaten erstmalig laden	710
15.2.2 Die Projektdaten nachladen	712
15.2.3 Das Anwenderprogramm schützen	713
15.2.4 Mit Online-Projektdaten arbeiten	716
15.2.5 Mit der Memory Card arbeiten	719
15.3 Mit Bausteinen im Online-Betrieb arbeiten	720
15.3.1 Einführung	720
15.3.2 Einen Online-Baustein bearbeiten	721
15.3.3 Einen Baustein laden und zurückladen	722

15.3.4	Ohne Reinitialisierung laden	723
15.3.5	Mit Momentaufnahmen arbeiten	725
15.3.6	Mit Einstellwerten arbeiten	726
15.3.7	Bausteine vergleichen	728
15.4	Hardware-Diagnose	731
15.4.1	Status-Anzeigen an den Baugruppen	731
15.4.2	Diagnosefenster	732
15.4.3	Diagnosepuffer	734
15.4.4	Online-Tools	734
15.4.5	Weitere Diagnose-Informationen über das Programmiergerät	736
15.5	Anwenderprogramm testen	737
15.5.1	Aufrufumgebung definieren	738
15.5.2	Testen mit Programmstatus	739
15.5.3	PLC-Variablen beobachten	743
15.5.4	Datenvariablen beobachten und steuern	744
15.5.5	Testen mit Beobachtungstabellen	745
15.5.6	Testen mit der Force-Tabelle	750
15.6	Messwertaufzeichnung mit der Trace-Funktion	752
15.6.1	Einführung	752
15.6.2	Erstellen der Trace-Konfiguration	753
15.6.3	Messwerte aufzeichnen	754
16	Dezentrale Peripherie	758
16.1	Einführung, Übersicht	758
16.2	Dezentrales Peripheriesystem ET 200	759
16.2.1	ET 200MP	759
16.2.2	ET 200M	760
16.2.3	ET 200SP	761
16.2.4	ET 200S	762
16.2.5	ET 200pro	762
16.2.6	ET 200AL	763
16.2.7	ET 200eco und ET200eco PN	764
16.3	PROFINET IO	764
16.3.1	Komponenten von PROFINET IO	764
16.3.2	Adressen bei PROFINET IO	767
16.3.3	PROFINET IO projektieren	769
16.3.4	Kopplungsbaugruppen für PROFINET IO	775
16.3.5	Echtzeit-Kommunikation bei PROFINET	776
16.3.6	Spezielle PROFINET-Konfigurationen	780
16.4	PROFIBUS DP	784
16.4.1	Komponenten von PROFIBUS DP	784
16.4.2	Adressen bei PROFIBUS DP	787
16.4.3	PROFIBUS DP projektieren	789
16.4.4	Kopplungsbaugruppen für PROFIBUS DP	792
16.4.5	Spezielle PROFIBUS-Konfigurationen	795
16.5	Systembausteine für dezentrale Peripherie	798

16.5.1 Peripheriedaten lesen und schreiben	798
16.5.2 Diagnosedaten von einem DP-Normslave lesen	801
16.5.3 Datensatz empfangen und bereitstellen	802
16.5.4 Dezentrale Station aktivieren/deaktivieren	804
16.5.5 PROFINET-IO-System umkonfigurieren	805
16.6 DPV1-Alarmer	805
16.7 Taktsynchronität	808
16.7.1 Einführung	808
16.7.2 Taktsynchronität bei PROFINET	808
16.7.3 Taktsynchronität bei PROFIBUS	811
16.7.4 Taktsynchronalarm	813
16.7.5 Prozessabbilder taktsynchron aktualisieren	816
17 Kommunikation	817
17.1 Übersicht	817
17.2 Open User Communication	820
17.2.1 Grundlagen	820
17.2.2 Datenstruktur der Open User Communication	821
17.2.3 Verbindung aufbauen und Daten senden mit TSEND_C	822
17.2.4 Verbindung aufbauen und Daten empfangen mit TRCV_C	823
17.2.5 Open User Communication projektieren	825
17.2.6 Weitere Funktionen für die Open User Communication	827
17.3 S7-Kommunikation	830
17.3.1 Grundlagen	830
17.3.2 Einseitiger Datenaustausch	830
17.3.3 Zweiseitiger Datenaustausch	832
17.3.4 S7-Kommunikation projektieren	835
17.4 Freeport-Kommunikation	836
17.4.1 Einführung in die Freeport-Kommunikation	836
17.4.2 Konfigurieren der Kommunikationsbaugruppe CM PtP	837
17.4.3 Freeport-Kommunikationsfunktionen	838
17.5 Weitere Kommunikationsfunktionen	842
17.5.1 USS-Protokoll für Antriebe	842
17.5.2 Modbus RTU	844
17.5.3 Modbus TCP	846
18 Anhang	849
18.1 Ein Projekt migrieren	849
18.2 Webserver	852
18.2.1 Webserver aktivieren	852
18.2.2 Standard-Webseiten	855
18.2.3 Basic-Webseiten	858
18.2.4 Service-Daten auslesen	858
18.2.5 Webserver initialisieren und Webseiten synchronisieren (WWW) ...	859
18.3 Technologieobjekte	859

18.3.1 Technologieobjekte für Motion Control	859
18.3.2 Technologieobjekte für PID Control	866
18.3.3 Technologieobjekte für Zählen und Messen	867
18.4 Daten protokollieren und Rezepturen übertragen	872
18.4.1 Einführung in die Datenprotokollierung	872
18.4.2 Datenprotokollierung anwenden	872
18.4.3 Funktionen für die Datenprotokollierung	873
18.4.4 Einführung zur Rezepturenübertragung	875
18.4.5 Funktionen für die Rezepturenübertragung	877
18.5 Simulation mit S7-PLCSIM	878
18.5.1 S7-PLCSIM allgemein	878
18.5.2 Die Bedienoberfläche von PLCSIM	879
18.5.3 In S7-PLCSIM mit STEP-7-Testfunktionen testen	880
18.5.4 Mit einem Simulationsprojekt arbeiten	882
18.5.5 Mit dem Adressbereich testen	883
18.5.6 Mit der SIM-Tabelle testen	883
18.5.7 Mit der Sequenztabelle testen	885
18.6 Maschinen- und Anlagendiagnose mit ProDiag	888
Stichwortverzeichnis	892

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).

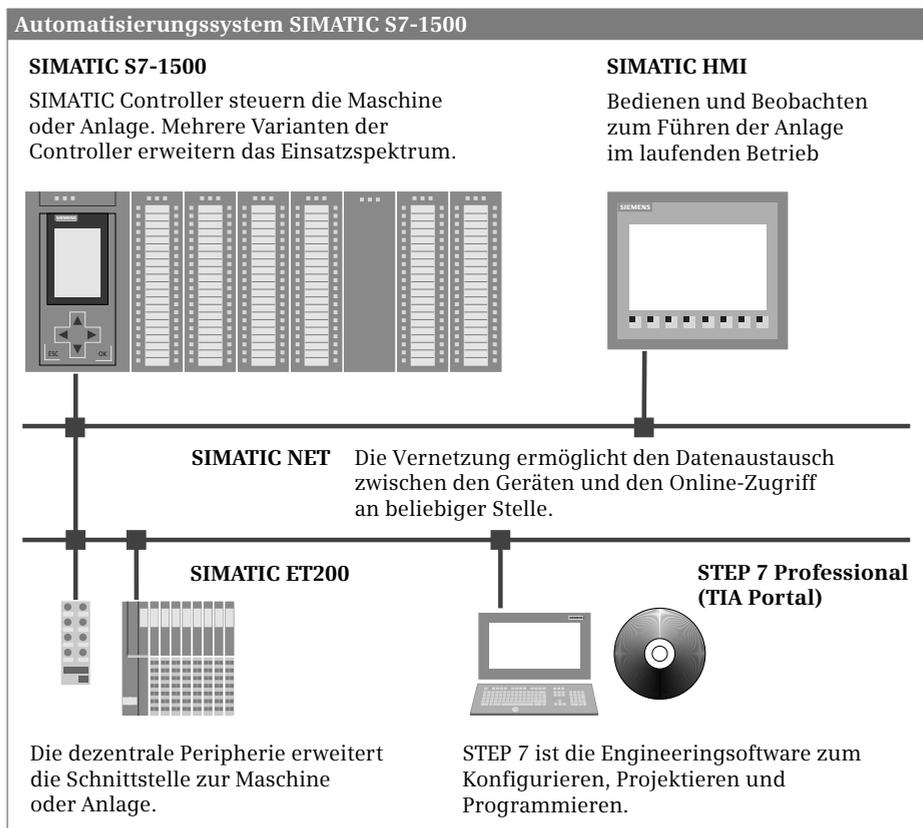


Bild 1.1 Bestandteile des Automatisierungssystems SIMATIC S7-1500

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.

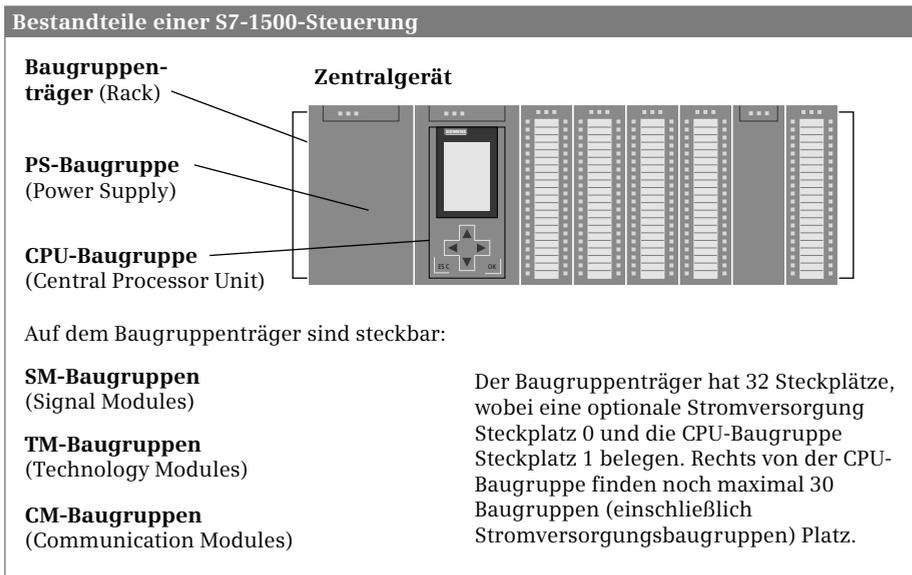


Bild 1.2 Bestandteile einer S7-1500-Steuerung mit Standard-Controller

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-

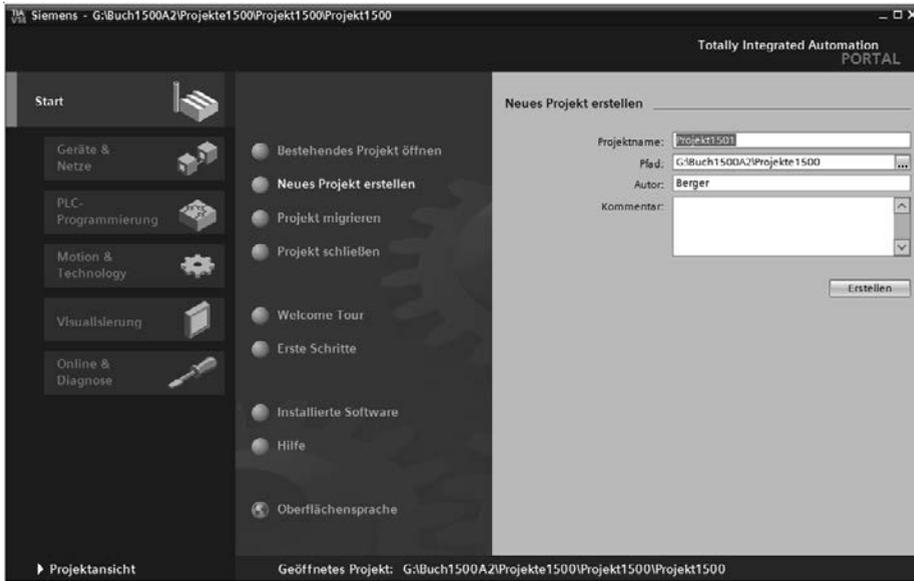


Bild 1.3 Werkzeug im Startportal von STEP 7 Professional V14

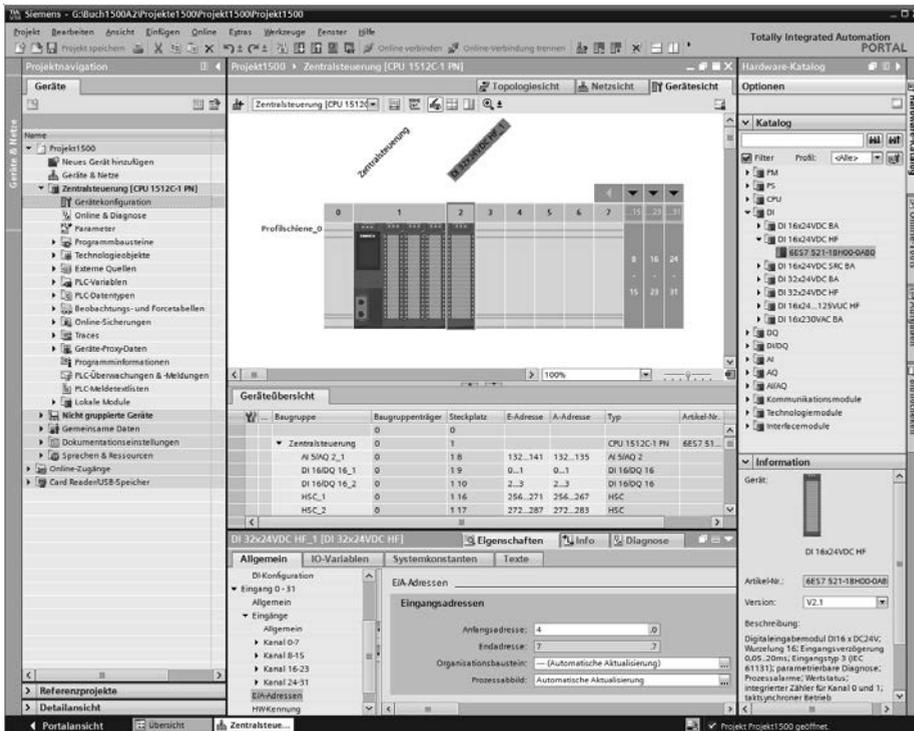


Bild 1.4 Beispiel für die Projektansicht: Arbeitsbereich der Gerätekonfiguration

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.

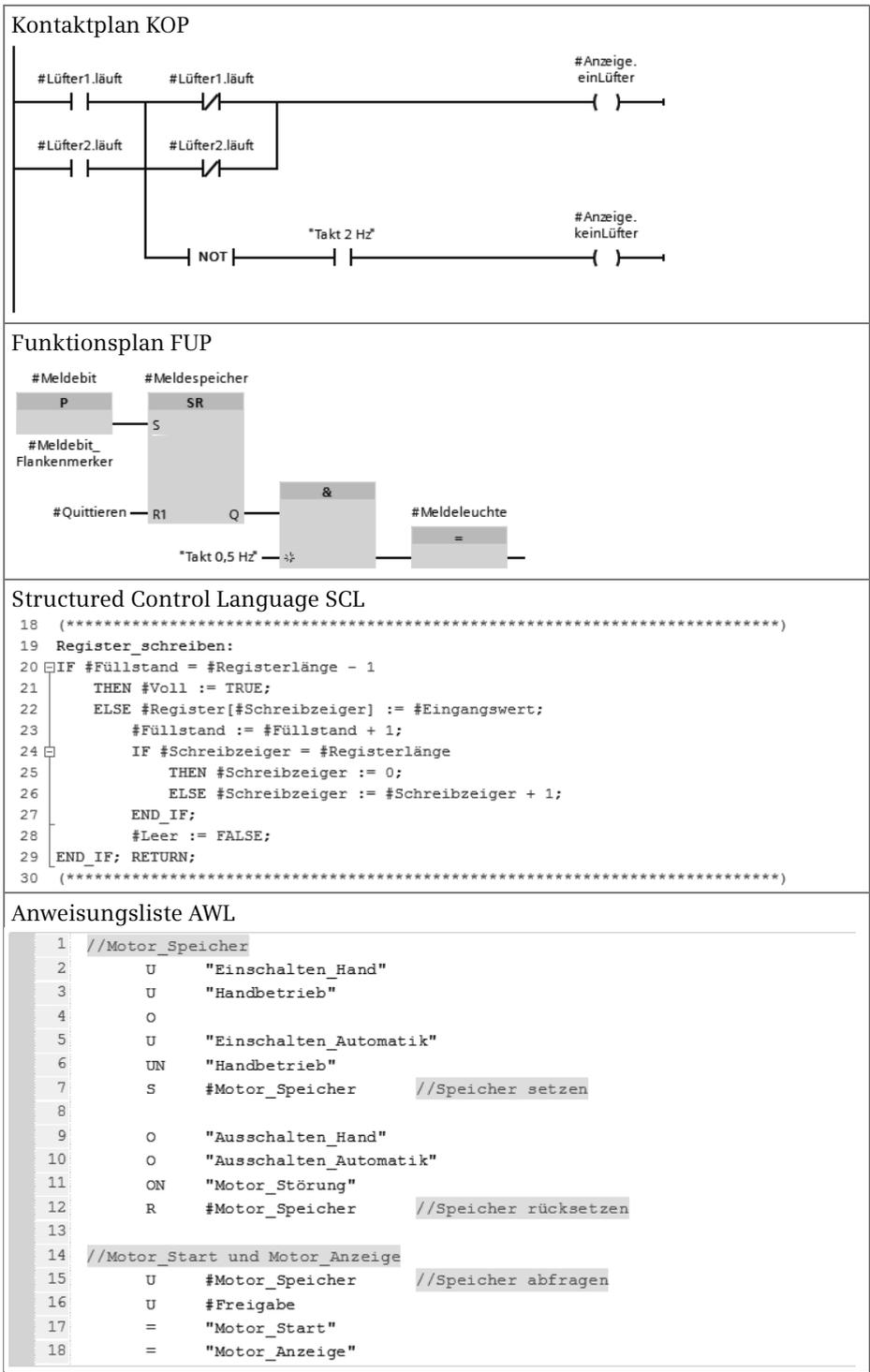


Bild 1.5 Beispiele für die Darstellung in KOP, FUP, SCL und AWL

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).

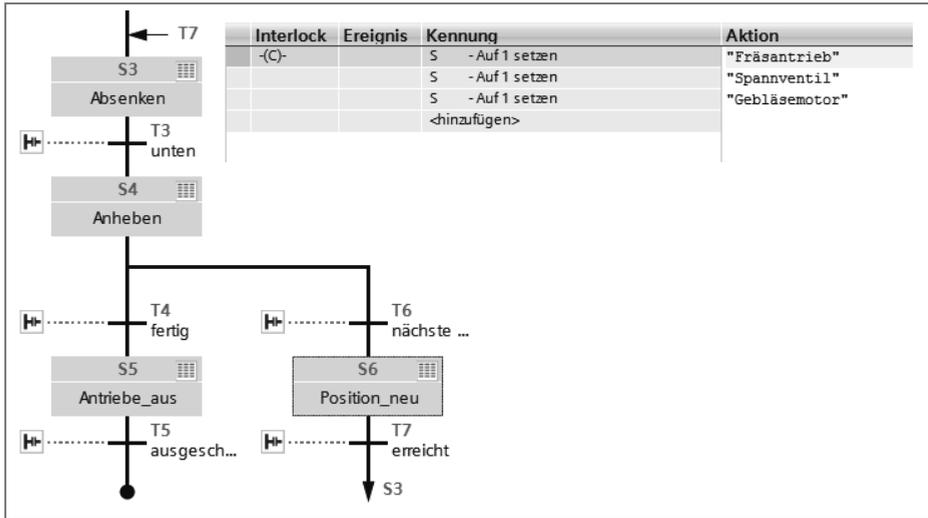


Bild 1.6 Beispiel für eine GRAPH-Ablaufkette und eine Schrittprojektierung

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-