

SIEMENS

Hans Berger

Automatisieren mit SIMATIC S7-1500

Projektieren, Programmieren und Testen
mit STEP 7 Professional



Berger Automatisieren mit SIMATIC S7-1500

Automatisieren mit SIMATIC S7-1500

Projektieren, Programmieren und Testen
mit STEP 7 Professional

von Hans Berger

Publicis Publishing

Bibliografische Information der Deutschen Nationalbibliothek

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

Print ISBN 978-3-89578-403-3

ePDF ISBN 978-3-89578-903-8

Herausgeber Siemens Aktiengesellschaft, Berlin und München

Verlag: Publicis Publishing, Erlangen

© 2014 by Publicis Erlangen, Zweigniederlassung der PWW 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 Teilsysteme 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 neu entwickelte 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 Auflistung von maschinennahen Befehlen 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 des Automatisierungssystems S7-1500 mit der Engineeringsoftware STEP 7 Professional in der Version V12 SP 1 in Verbindung mit einer CPU 1500 mit der Firmware Version V1.1 und der Simulationssoftware PLCSIM Version V12 SP 1.

Erlangen, im September 2013

Hans Berger

Der Inhalt des Buchs auf einen Blick

Start

Das Automatisierungssystem SIMATIC S7-1500 in der Übersicht.
Eine Einführung in die Engineeringsoftware SIMATIC STEP 7 Professional V12.
Die Grundlage der Automatisierungslösung: Ein Projekt erstellen und bearbeiten.

Automatisierungssystem SIMATIC S7-1500

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

Gerätekonfiguration

Eine Station konfigurieren, Baugruppen parametrieren und Stationen vernetzen.

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.

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.

Der Programmeditor

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

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.

Die Programmiersprache Funktionsplan FUP

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

Die Programmiersprache Structured Control Language SCL

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

Die Programmiersprache Anweisungsliste AWL

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

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

Basisfunktionen: Funktionen für Binärsignale: binäre Verknüpfungen, Speicherfunktionen, Flankenauswertungen, SIMATIC-Zeit-/Zählfunktionen, IEC-Zeit-/Zählfunktionen.

Digitalfunktionen: Funktionen für Digitalvariablen: Übertragungs-, Vergleichs-, Arithmetik-, Mathematik-, Konvertierungs-, Schiebe- und Logikfunktionen.

Programmsteuerung: Mit Sprungfunktionen im Programm verzweigen, Funktionen und Funktionsbausteine aufrufen und beenden, ARRAY- und CPU-Datenbausteine.

Online-Betrieb und Programmtest

Ein Programmiergerät an die PLC-Station anschließen, den Online-Betrieb einschalten, die Projektdaten übertragen und das Anwenderprogramm schützen.

Die Anwenderbausteine laden, ändern, löschen und vergleichen.

Mit der Hardware-Diagnose arbeiten und das Anwenderprogramm testen.

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.

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.

Anhang

Wie externe Quelldateien für AWL- und SCL-Bausteine erstellt und importiert werden.

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.

Technologiefunktionen: Zählen, Messen, Motion Control, PID Control

Wie mit der Simulationssoftware S7-PLCSIM ein Anwenderprogramm offline getestet wird.

Inhaltsverzeichnis

1 Einführung	24
1.1 Übersicht Automatisierungssystem S7-1500	24
1.1.1 Automatisierungsgerät SIMATIC S7-1500	25
1.1.2 Übersicht STEP 7 Professional V12	26
1.1.3 Verschiedene Programmiersprachen	28
1.1.4 Bearbeitung des Anwenderprogramms	30
1.1.5 Datenhaltung im SIMATIC-Automatisierungssystem	33
1.2 Einführung in STEP 7 Professional V12	34
1.2.1 STEP 7 installieren	34
1.2.2 Automation License Manager	35
1.2.3 STEP 7 Professional starten	35
1.2.4 Portalansicht	35
1.2.5 Die Fenster der Projektansicht	36
1.2.6 Hilfe-Informationssystem	39
1.2.7 Bedienoberfläche anpassen	40
1.3 Ein SIMATIC-Projekt bearbeiten	40
1.3.1 Strukturierte Darstellung der Projektdaten	41
1.3.2 Projektdaten und Editoren für eine PLC-Station	41
1.3.3 Ein Projekt erstellen und bearbeiten	45
1.3.4 Mit Referenzprojekten arbeiten	48
1.3.5 Bibliotheken anlegen und bearbeiten	49
2 Automatisierungssystem SIMATIC S7-1500	50
2.1 Komponenten einer S7-1500-Station	50
2.2 CPU-Baugruppen S7-1500	52
2.2.1 CPU-Varianten	52
2.2.2 Bedien- und Anzeigeelemente	52
2.2.3 SIMATIC Memory Card	54
2.2.4 Speicherbereiche in einer S7-1500-Station	55
2.2.5 Busschnittstellen	57
2.3 Signalbaugruppen	58
2.3.1 Digitaleingabebaugruppen	58
2.3.2 Digitalausgabebaugruppen	59
2.3.3 Analogeingabebaugruppen	60
2.3.4 Analogausgabebaugruppe	60

2.4 Technologiebaugruppen	61
2.5 Kommunikationsbaugruppen	62
2.6 Sonstige Baugruppen	63
2.6.1 Systemstromversorgungsbaugruppen	63
2.6.2 Laststromversorgungsbaugruppen	63
3 Gerätekonfiguration	64
3.1 Einführung	64
3.2 Eine Station konfigurieren	66
3.2.1 Eine PLC-Station hinzufügen	66
3.2.2 Eine Baugruppe hinzufügen	67
3.3 Baugruppen parametrieren	68
3.3.1 Die CPU-Eigenschaften parametrieren	68
3.3.2 Baugruppen adressieren	71
3.3.3 Signalbaugruppen parametrieren	73
3.4 Eine Vernetzung projektieren	77
3.4.1 Einführung, Übersicht	77
3.4.2 Eine Station vernetzen	78
3.4.3 Teilnehmeradressen in einem Subnetz	79
3.4.4 Kommunikationsdienste und Verbindungstypen	80
3.4.5 Eine Verbindung projektieren	82
3.4.6 Ein PROFINET-Subnetz projektieren	84
3.4.7 Ein PROFIBUS-Subnetz projektieren	88
4 Variablen, Adressierung und Datentypen	91
4.1 Operanden und Variablen	91
4.1.1 Einführung, Übersicht	91
4.1.2 Operandenbereiche Eingänge und Ausgänge	93
4.1.3 Operandenbereich Merker	95
4.1.4 Operandenbereich Daten	97
4.1.5 Operandenbereich temporäre Lokaldaten	98
4.2 Adressierung von Operanden und Variablen	99
4.2.1 Signalweg	99
4.2.2 Absolute Adressierung	100
4.2.3 Symbolische Adressierung	103
4.2.4 Adressierung eines Variablenbereichs	104
4.2.5 Adressierung einer Konstanten	105
4.3 Indirekte Adressierung	105
4.3.1 Übersicht	105
4.3.2 Indirekte Adressierung von ARRAY-Komponenten	106
4.3.3 Indirekte Adressierung einer Variablen in einem ARRAY-DB	107
4.3.4 Indirekte Adressierung eines Datenbausteins	108
4.3.5 Indirekte Adressierung mit einem ANY-Zeiger	108
4.3.6 Indirekte Adressierung mit PEEK und POKE (SCL)	110

4.4	Adressierung von Hardware-Objekten	113
4.5	Allgemeines zu Datentypen	113
4.5.1	Übersicht der Datentypen	113
4.5.2	Implizite Datentypkonvertierung	115
4.5.3	Variablen überlagern (Datentypsichten)	117
4.6	Elementare Datentypen	119
4.6.1	Bitfolge-Datentypen BOOL, BYTE, WORD, DWORD und LWORD	119
4.6.2	Datentyp CHAR	121
4.6.3	BCD-Zahlen BCD16 und BCD32	122
4.6.4	Festpunkt-Datentypen ohne Vorzeichen USINT, UINT, UDINT, ULINT	122
4.6.5	Festpunkt-Datentypen mit Vorzeichen SINT, INT, DINT und LINT	123
4.6.6	Gleitpunkt-Datentypen REAL und LREAL	124
4.6.7	Datentypen für Zeitdauern	126
4.6.8	Datentypen für Zeitpunkte	127
4.7	Strukturierte Datentypen	129
4.7.1	Datum und Uhrzeit DATE_AND_TIME (DT)	130
4.7.2	Datum und Uhrzeit DATE_AND_LTIME (DTL)	130
4.7.3	Zeichenkette STRING	131
4.7.4	Datenfeld ARRAY	133
4.7.5	Datenstruktur STRUCT	135
4.8	Parametertypen	137
4.8.1	Übersicht	137
4.8.2	Parametertypen TIMER und COUNTER	138
4.8.3	Parametertypen für IEC-Zeitfunktionen	138
4.8.4	Parametertypen für IEC-Zählfunktionen	138
4.8.5	Parametertypen BLOCK_FC und BLOCK_FB (AWL)	139
4.8.6	Parametertyp DB_ANY	139
4.8.7	Parametertyp VOID	140
4.8.8	Parametertypen POINTER, ANY und VARIANT	140
4.9	Zeiger	141
4.9.1	Einführung	141
4.9.2	Bereichszeiger	141
4.9.3	DB-Zeiger	143
4.9.4	ANY-Zeiger	143
4.10	PLC-Datentypen	144
4.10.1	Einen PLC-Datentyp programmieren	144
4.10.2	Einen PLC-Datentyp anwenden	145
4.10.3	PLC-Datentypen vergleichen	145
4.11	Systemdatentypen	146
4.11.1	Systemdatentypen für IEC-Zeitfunktionen	146
4.11.2	Systemdatentypen für IEC-Zählfunktionen	147
4.11.3	Datentyp ERROR_STRUCT	147
4.11.4	Startinformation	149
4.12	Hardware-Datentypen	150

5 Programmbearbeitung	152
5.1 Betriebszustände der CPU-Baugruppe	152
5.1.1 Betriebszustand STOP	153
5.1.2 Betriebszustand ANLAUF	154
5.1.3 Betriebszustand RUN	156
5.1.4 Remanenzverhalten von Operanden	156
5.2 Anwenderprogramm erstellen	157
5.2.1 Programmentwurf	157
5.2.2 Programmbearbeitung	161
5.3 Bausteine programmieren	163
5.3.1 Bausteinarten	163
5.3.2 Bausteineigenschaften	165
5.3.3 Bausteinschnittstelle	167
5.3.4 Bausteinparameter programmieren	170
5.4 Bausteine aufrufen	172
5.4.1 Allgemeines zu Aufrufen von Codebausteinen	172
5.4.2 Bausteinparameter versorgen	174
5.4.3 Aufruf einer Funktion (FC)	176
5.4.4 Aufruf eines Funktionsbausteins (FB)	176
5.4.5 „Weiterreichen“ von Bausteinparametern	178
5.5 Anlaufprogramm	178
5.5.1 Anlauf-Organisationsbausteine	178
5.5.2 Remanente Daten zurücksetzen	180
5.5.3 Baugruppenadresse ermitteln	180
5.5.4 Baugruppen parametrieren	183
5.6 Hauptprogramm	187
5.6.1 Hauptprogramm-Organisationsbausteine	187
5.6.2 Prozessabbild-Aktualisierung	189
5.6.3 Zykluszeit	192
5.6.4 Reaktionszeit	195
5.6.5 Programm stoppen und verzögern	197
5.6.6 Uhrzeit	198
5.6.7 Systemzeit lesen	202
5.6.8 Betriebsstundenzähler	202
5.7 Alarmbearbeitung	204
5.7.1 Einführung zur Alarmbearbeitung	204
5.7.2 Uhrzeitalarme	208
5.7.3 Verzögerungsalarme	212
5.7.4 Weckalarme	215
5.7.5 Prozessalarme	219
5.7.6 Alarme zur Laufzeit zuordnen	221
5.7.7 Alarmzusatzinformation lesen	222
5.8 Fehlerbehandlung	224
5.8.1 Fehlerursachen und Fehlerreaktion	224
5.8.2 Lokale Fehlerbehandlung	225

5.8.3 Globale Fehlerbehandlung (Synchronfehler)	228
5.8.4 Synchronfehlerbearbeitung sperren und freigeben	230
5.8.5 Asynchronfehler	233
5.8.6 Alarme und Asynchronfehler sperren, verzögern und freigeben	236
5.9 Diagnose im Anwenderprogramm	238
5.9.1 Diagnosealarm	238
5.9.2 Startinformation lesen	240
5.9.3 Diagnosefunktionen im Anwenderprogramm	240
5.10 Meldungen projektieren	248
5.10.1 Einführung	248
5.10.2 Meldungen nach dem Meldenummervverfahren projektieren	252
5.10.3 Bausteine für die Meldungsprogrammierung	256
5.10.4 CPU-Meldungsanzeige	257
6 Programmeditor	261
6.1 Einführung	261
6.2 PLC-Variablen-tabelle	262
6.2.1 PLC-Variablen-tabelle anlegen und bearbeiten	262
6.2.2 PLC-Variablen definieren und bearbeiten	262
6.2.3 PLC-Variablen-tabellen vergleichen	265
6.2.4 PLC-Variablen-tabelle exportieren und importieren	266
6.2.5 Konstanten-tabellen	267
6.3 Einen Codebaustein programmieren	268
6.3.1 Einen neuen Codebaustein anlegen	268
6.3.2 Arbeitsbereich des Programmeditors für Codebausteine	269
6.3.3 Bausteineigenschaften für Codebausteine festlegen	271
6.3.4 Bausteine schützen	274
6.3.5 Bausteinschnittstelle programmieren	276
6.3.6 Steuerungsfunktion programmieren	278
6.3.7 Variablen editieren	282
6.3.8 Mit Programmkommentaren arbeiten	285
6.4 Einen Datenbaustein programmieren	286
6.4.1 Einen neuen Datenbaustein anlegen	286
6.4.2 Arbeitsbereich des Programmeditors für Datenbausteine	287
6.4.3 Bausteineigenschaften für Datenbausteine festlegen	288
6.4.4 Datenvariablen deklarieren	290
6.4.5 Datenvariablen in Globaldatenbausteinen eingeben	292
6.5 Bausteine übersetzen	293
6.5.1 Übersetzung starten	293
6.5.2 SCL-Bausteine übersetzen	294
6.5.3 Fehler nach der Übersetzung beheben	295
6.6 Programminformationen	296
6.6.1 Querverweisliste	296
6.6.2 Belegungsplan	297
6.6.3 Aufrufstruktur	299
6.6.4 Abhängigkeitsstruktur	300

6.6.5 Konsistenzprüfung	301
6.6.6 Speicherauslastung der CPU	302
6.7 Sprachen-Einstellung	303
7 Kontaktplan KOP	305
7.1 Einführung	305
7.1.1 Programmieren mit KOP allgemein	305
7.1.2 Programmelemente des Kontaktplans	307
7.2 Binäre Verknüpfungen mit KOP programmieren	307
7.2.1 Schließerkontakt und Öffnerkontakt	308
7.2.2 Reihen- und Parallelschaltung von Kontakten	309
7.2.3 T-Abzweig, offener Parallelzweig	311
7.2.4 Verknüpfungsergebnis negieren im Kontaktplan	311
7.2.5 Flankenauswertung einer Binärvariablen im Kontaktplan	312
7.2.6 Gültigkeitsprüfung einer Gleitpunktvariablen im Kontaktplan	313
7.2.7 Vergleich-Kontakte	313
7.3 Speicherfunktionen mit KOP programmieren	314
7.3.1 Einfache und negierende Spule	314
7.3.2 Setzen- und Rücksetzen-Spule	316
7.3.3 Speicherndes Verhalten durch Selbsthaltung	317
7.3.4 Flankenauswertung mit Impulsausgabe im Kontaktplan	317
7.3.5 Mehrfaches Setzen und Rücksetzen (Bitfeld füllen) im Kontaktplan ..	319
7.3.6 Spulen mit Zeitverhalten	319
7.3.7 Spulen mit Zählverhalten	321
7.4 Q-Boxen mit KOP programmieren	322
7.4.1 Speicher-Boxen im Kontaktplan	323
7.4.2 Flankenauswertung des Stromflusses	323
7.4.3 SIMATIC-Zeitfunktionen im Kontaktplan	324
7.4.4 SIMATIC-Zählfunktionen im Kontaktplan	325
7.4.5 IEC-Zeitfunktionen im Kontaktplan	325
7.4.6 IEC-Zählfunktionen im Kontaktplan	326
7.5 EN/ENO-Boxen mit KOP programmieren	328
7.5.1 Flankenauswertung mit einer EN/ENO-Box	329
7.5.2 Übertragungsfunktionen im Kontaktplan	330
7.5.3 Arithmetische Funktionen im Kontaktplan	330
7.5.4 Mathematische Funktionen im Kontaktplan	331
7.5.5 Konvertierungsfunktionen im Kontaktplan	332
7.5.6 Schiebefunktionen im Kontaktplan	332
7.5.7 Logikfunktionen im Kontaktplan	333
7.5.8 Funktionen für Zeichenketten im Kontaktplan	335
7.6 Programmsteuerung mit KOP	335
7.6.1 Sprungfunktionen im Kontaktplan	335
7.6.2 Bausteinaufruf-Funktionen im Kontaktplan	338
7.6.3 Bausteinende-Funktion im Kontaktplan	338
7.6.4 EN/ENO-Mechanismus im Kontaktplan	339

8 Funktionsplan FUP	343
8.1 Einführung	343
8.1.1 Programmieren mit FUP allgemein	343
8.1.2 Programmelemente des Funktionsplans	345
8.2 Binäre Verknüpfungen mit FUP programmieren	345
8.2.1 Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	345
8.2.2 Eine binäre Verknüpfung im Funktionsplan programmieren	348
8.2.3 UND-Funktion im Funktionsplan	348
8.2.4 ODER-Funktion im Funktionsplan	348
8.2.5 Exklusiv-ODER-Funktion im Funktionsplan	349
8.2.6 Kombinierte binäre Verknüpfungen, Verknüpfungsergebnis negieren	349
8.2.7 T-Abzweig im Funktionsplan	350
8.2.8 Flankenauswertung von Binärvariablen im Funktionsplan	351
8.2.9 Gültigkeitsprüfung von Gleitpunktzahlen im Funktionsplan	351
8.2.10 Vergleichsfunktionen im Funktionsplan	352
8.3 Standard-Boxen mit FUP programmieren	353
8.3.1 Zuweisung und negierende Zuweisung	353
8.3.2 Setzen- und Rücksetzen-Box	355
8.3.3 Flankenauswertung mit Impulsausgabe im Funktionsplan	356
8.3.4 Mehrfaches Setzen und Rücksetzen (Bitfeld füllen) im Funktionsplan	356
8.3.5 Standard-Boxen mit Zeitverhalten	357
8.3.6 Standard-Boxen mit Zählerverhalten	359
8.4 Q-Boxen mit FUP programmieren	360
8.4.1 Speicher-Boxen im Funktionsplan	360
8.4.2 Flankenauswertung des Verknüpfungsergebnisses im Funktionsplan	360
8.4.3 SIMATIC-Zeitfunktionen im Funktionsplan	362
8.4.4 SIMATIC-Zählfunktionen im Funktionsplan	363
8.4.5 IEC-Zeitfunktionen im Funktionsplan	363
8.4.6 IEC-Zählfunktionen im Funktionsplan	364
8.5 EN/ENO-Boxen mit FUP programmieren	366
8.5.1 Flankenauswertung mit einer EN/ENO-Box	367
8.5.2 Übertragungsfunktionen im Funktionsplan	368
8.5.3 Arithmetische Funktionen im Funktionsplan	368
8.5.4 Mathematische Funktionen im Funktionsplan	369
8.5.5 Konvertierungsfunktionen im Funktionsplan	370
8.5.6 Schiebefunktionen im Funktionsplan	371
8.5.7 Logikfunktionen im Funktionsplan	372
8.5.8 Funktionen für Zeichenketten im Funktionsplan	372
8.6 Programmsteuerung mit FUP	373
8.6.1 Sprungfunktionen im Funktionsplan	374
8.6.2 Bausteinaufruf-Funktionen im Funktionsplan	376
8.6.3 Bausteinende-Funktion im Funktionsplan	377
8.6.4 EN/ENO-Mechanismus im Funktionsplan	378

9 Structured Control Language SCL	381
9.1 Einführung	381
9.1.1 Programmieren mit SCL allgemein	381
9.1.2 SCL-Anweisungen und Operatoren	383
9.2 Binäre Verknüpfungen mit SCL programmieren	385
9.2.1 Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	385
9.2.2 UND-Funktion in SCL	386
9.2.3 ODER-Funktion in SCL	387
9.2.4 Exklusiv-ODER-Funktion in SCL	387
9.2.5 Kombinierte binäre Verknüpfungen in SCL	387
9.2.6 Verknüpfungsergebnis negieren in SCL	388
9.3 Speicherfunktionen mit SCL programmieren	389
9.3.1 Wertzuweisung einer Binärvariablen	389
9.3.2 Setzen und Rücksetzen in SCL	389
9.3.3 Flankenauswertung in SCL	389
9.4 Zeit- und Zählfunktionen mit SCL programmieren	390
9.4.1 SIMATIC-Zeitfunktionen in SCL	390
9.4.2 SIMATIC-Zählfunktionen in SCL	392
9.4.3 IEC-Zeitfunktionen in SCL	393
9.4.4 IEC-Zählfunktionen in SCL	393
9.5 Digitalfunktionen mit SCL programmieren	394
9.5.1 Übertragungsfunktion, Wertzuweisung einer Digitalvariablen	394
9.5.2 Vergleichsfunktionen in SCL	395
9.5.3 Arithmetische Funktionen in SCL	396
9.5.4 Mathematische Funktionen in SCL	397
9.5.5 Konvertierungsfunktionen in SCL	398
9.5.6 Schiebefunktionen in SCL	400
9.5.7 Wortverknüpfungen, logischer Ausdruck in SCL	401
9.5.8 Funktionen für Zeichenketten in SCL	402
9.6 Programmsteuerung mit SCL	403
9.6.1 Mit der Variablen ENO arbeiten	403
9.6.2 EN/ENO-Mechanismus bei SCL	404
9.6.3 Kontrollanweisungen	406
9.6.4 Bausteinfunktionen	417
10 Anweisungsliste AWL	419
10.1 Einführung	419
10.1.1 Programmieren mit AWL allgemein	419
10.1.2 Aufbau einer AWL-Anweisung	421
10.1.3 Eine AWL-Anweisung eingeben	421
10.1.4 Adressierung von 64-Bit-Variablen	423
10.1.5 AWL-Netzwerke in KOP- und FUP-Bausteinen	423
10.2 Binäre Verknüpfungen mit AWL programmieren	424
10.2.1 Bearbeitung einer binären Verknüpfung, Verknüpfungsschritt	424
10.2.2 Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	426
10.2.3 UND-Funktion in der Anweisungsliste	427

10.2.4 ODER-Funktion in der Anweisungsliste	428
10.2.5 Exklusiv-ODER-Funktion in der Anweisungsliste	428
10.2.6 Kombinierte binäre Verknüpfungen in der Anweisungsliste	428
10.2.7 Verknüpfungsergebnis steuern	431
10.3 Speicherfunktionen mit AWL programmieren	432
10.3.1 Zuweisung in der Anweisungsliste	432
10.3.2 Setzen und Rücksetzen in der Anweisungsliste	432
10.3.3 Flankenbewertung in der Anweisungsliste	433
10.4 Zeit- und Zählfunktionen mit AWL programmieren	434
10.4.1 SIMATIC-Zeitfunktionen in der Anweisungsliste	434
10.4.2 SIMATIC-Zählfunktionen in der Anweisungsliste	436
10.4.3 IEC-Zeitfunktionen in der Anweisungsliste	437
10.4.4 IEC-Zählfunktionen in der Anweisungsliste	438
10.5 Digitalfunktionen mit AWL programmieren	439
10.5.1 Übertragungsfunktionen in der Anweisungsliste	441
10.5.2 Vergleichsfunktionen in der Anweisungsliste	443
10.5.3 Arithmetische Funktionen in der Anweisungsliste	447
10.5.4 Mathematische Funktionen in der Anweisungsliste	451
10.5.5 Konvertierungsfunktionen in der Anweisungsliste	453
10.5.6 Schiebefunktionen in der Anweisungsliste	455
10.5.7 Wortverknüpfungen in der Anweisungsliste	457
10.5.8 Funktionen für Zeichenketten in der Anweisungsliste	459
10.6 Programmsteuerung mit AWL	461
10.6.1 Sprungfunktionen in der Anweisungsliste	462
10.6.2 Bausteinaufruf-Funktion in der Anweisungsliste	463
10.6.3 Bausteinende-Funktionen in der Anweisungsliste	466
10.7 Weitere AWL-Funktionen	467
10.7.1 Mit Statusbits arbeiten	468
10.7.2 EN/ENO-Mechanismus in der Anweisungsliste	473
10.7.3 Akkumulatorfunktionen	475
10.7.4 Arbeiten mit den Datenbausteinregistern	477
10.7.5 Teiladressierung von Datenoperanden	479
10.7.6 Absolute Adressierung von temporären Lokaldaten	480
10.7.7 Arbeiten mit den Adressregistern	481
10.7.8 Speicherindirekte Adressierung	483
10.7.9 Registerindirekte Adressierung	488
10.7.10 Direkter Zugriff auf komplexe Lokalvariablen	489
10.7.11 Datenablage der Bausteinparameter einer Funktion (FC)	491
10.7.12 Datenablage der Bausteinparameter eines Funktionsbausteins (FB)	495
10.7.13 Datenablage einer Lokalinstanz in einer Multiinstanz	497
10.7.14 Nullanweisungen	498
11 Ablaufsteuerung S7-GRAPH	500
11.1 Einführung	500
11.1.1 Was ist eine Ablaufsteuerung?	500
11.1.2 Eigenschaften einer Ablaufsteuerung	501

11.1.3 Programm für eine Ablaufsteuerung, Mengengerüst	502
11.1.4 Betriebsarten	502
11.1.5 Vorgehen bei der Projektierung	503
11.2 Elemente einer Ablaufsteuerung	503
11.2.1 Schritte und Transitionen	503
11.2.2 Sprünge in einer Ablaufsteuerung	505
11.2.3 Verzweigungen einer Ablaufkette	505
11.2.4 GRAPH-spezifische Variablen	507
11.2.5 Permanente Anweisungen	508
11.2.6 Schritt- und Transitionsfunktionen	508
11.2.7 Bearbeitung von Aktionen	512
11.3 Eine Ablaufsteuerung projektieren	518
11.3.1 Den GRAPH-Funktionsbaustein programmieren	519
11.3.2 Projektieren der Kettenstruktur	520
11.3.3 Schritte und Transitionen programmieren	521
11.3.4 Permanente Anweisungen programmieren	523
11.3.5 Bausteinübergreifende Meldungen projektieren	523
11.3.6 Attribute des GRAPH-Funktionsbausteins	524
11.3.7 GRAPH-Funktionsbaustein anwenden	526
11.4 Ablaufsteuerung testen	527
11.4.1 GRAPH-Funktionsbaustein laden	527
11.4.2 Einstellungen zum Programmtest	528
11.4.3 Betriebsarten verwenden	529
11.4.4 Ablaufkette synchronisieren	529
11.4.5 Mit Programmstatus testen	530
12 Basisfunktionen	533
12.1 Binäre Verknüpfungen	533
12.1.1 Einführung	533
12.1.2 Arbeiten mit Binärsignalen	534
12.1.3 UND-Funktion, Reihenschaltung	537
12.1.4 ODER-Funktion, Parallelschaltung	538
12.1.5 Exklusiv-ODER-Funktion, Antivalenzfunktion	539
12.1.6 Verknüpfungsergebnis negieren, NOT-Kontakt	539
12.2 Speicherfunktionen	541
12.2.1 Einführung	541
12.2.2 Einfache und negierende Spule, Zuweisung	541
12.2.3 Einzelnes Setzen und Rücksetzen	542
12.2.4 Mehrfaches Setzen und Rücksetzen	543
12.2.5 Vorrangiges Setzen und Rücksetzen, Speicherfunktion	545
12.3 Flankenbewertung	546
12.3.1 Funktionsweise einer Flankenbewertung	546
12.3.2 Flankenbewertung einer Binärvariablen (KOP, FUP)	548
12.3.3 Flankenbewertung mit Impulsausgabe (KOP, FUP)	549
12.3.4 Flankenbewertung mit einer Q-Box (KOP, FUP)	550
12.3.5 Flankenbewertung mit einer EN/ENO-Box (KOP, FUP)	551

12.3.6	Flankenauswertung bei SCL	552
12.3.7	Flankenauswertung bei AWL	554
12.4	SIMATIC-Zeitfunktionen	555
12.4.1	Übersicht	555
12.4.2	Programmieren einer Zeitfunktion	557
12.4.3	Zeitverhalten als Impuls	561
12.4.4	Zeitverhalten als verlängerter Impuls	563
12.4.5	Zeitverhalten als Einschaltverzögerung	564
12.4.6	Zeitverhalten als speichernde Einschaltverzögerung	566
12.4.7	Zeitverhalten als Ausschaltverzögerung	568
12.5	IEC-Zeitfunktionen	570
12.5.1	Einführung	570
12.5.2	Impulsbildung TP	571
12.5.3	Einschaltverzögerung TON	572
12.5.4	Ausschaltverzögerung TOF	573
12.5.5	Akkumulierende Einschaltverzögerung TONR	574
12.5.6	Eine IEC-Zeitfunktion mit einer Zeitdauer laden	576
12.5.7	Eine IEC-Zeitfunktion zurücksetzen	576
12.6	SIMATIC-Zählfunktionen	577
12.6.1	Übersicht	577
12.6.2	Programmieren einer Zählfunktion	578
12.6.3	Funktionsweise einer Zählfunktion	582
12.6.4	Freigeben einer Zählfunktion bei AWL	583
12.7	IEC-Zählfunktionen	585
12.7.1	Einführung	585
12.7.2	Vorwärtszähler CTU	586
12.7.3	Rückwärtszähler CTD	587
12.7.4	Vorwärts/Rückwärtszähler CTUD	588
13	Digitalfunktionen	590
13.1	Allgemeines	590
13.2	Übertragungsfunktionen	591
13.2.1	Allgemeines zur „einfachen“ Übertragungsfunktion	591
13.2.2	Variable kopieren, MOVE-Box bei KOP und FUP	591
13.2.3	Zeichenkette kopieren, S_MOVE-Box bei KOP und FUP	592
13.2.4	Wertzuweisungen bei SCL	592
13.2.5	Laden und Transferieren bei AWL	595
13.2.6	Datenbereich kopieren (MOVE_BLK_VARIANT)	596
13.2.7	Datenbereich kopieren (MOVE_BLK, UMOVE_BLK)	598
13.2.8	Datenbereich füllen (FILL_BLK, UFILL_BLK)	598
13.2.9	Datenbereich kopieren und füllen (BLKMOV, UBLKMOV, FILL)	600
13.2.10	Bytes tauschen (SWAP)	602
13.3	Vergleichsfunktionen	602
13.3.1	Ausführung der „einfachen“ Vergleichsfunktion	603
13.3.2	Vergleichsfunktion T_COMP	605
13.3.3	Vergleichsfunktion S_COMP	605

13.3.4 Bereichsvergleich	606
13.4 Arithmetische Funktionen	607
13.4.1 Arithmetische Funktionen für Zahlenwerte	607
13.4.2 Arithmetische Funktionen für Datum und Uhrzeit	607
13.4.3 Dekrementieren und Inkrementieren	610
13.5 Mathematische Funktionen	611
13.5.1 Allgemeine Funktionsbeschreibung	611
13.5.2 Winkelfunktionen SIN, COS, TAN	612
13.5.3 Arcusfunktionen ASIN, ACOS, ATAN	612
13.5.4 Quadrat bilden und Quadratwurzel ziehen	613
13.5.5 Logarithmus und Potenz	613
13.5.6 Nachkommastellen extrahieren, Absolutwert und Negation bilden ..	615
13.5.7 Rechnen mit der CALCULATE-Box bei KOP und FUP	617
13.6 Konvertierungsfunktionen	619
13.6.1 Datentypwandlung mit der Konvertierungsfunktion CONVERT	619
13.6.2 Datentypwandlung mit ROUND, CEIL, FLOOR und TRUNC	624
13.6.3 Datentypwandlung mit T_CONV	625
13.6.4 Datentypwandlung mit S_CONV	626
13.6.5 Konvertierungsfunktionen STRG_TO_CHARS und CHARS_TO_STRG ..	626
13.6.6 Konvertierungsfunktionen STRG_VAL und VAL_STRG	627
13.6.7 Datentypwandlung von Hexadezimalzahlen	631
13.6.8 Skalieren und Normieren	631
13.7 Schiebefunktionen	636
13.7.1 Allgemeine Funktionsbeschreibung	636
13.7.2 Rechts schieben	636
13.7.3 Links schieben	638
13.7.4 Rechts rotieren	638
13.7.5 Links rotieren	639
13.8 Logikfunktionen	640
13.8.1 Wortverknüpfungen	640
13.8.2 Invertieren, Einerkomplement bilden	642
13.8.3 Codierfunktionen DECO und ENCO	643
13.8.4 Auswahlfunktionen SEL, MUX und DEMUX	644
13.8.5 Minimumauswahl MIN, Maximumauswahl MAX	645
13.8.6 Begrenzer LIMIT	645
13.9 Bearbeitung von Zeichenketten (Datentyp STRING)	648
13.9.1 Aktuelle Länge einer Zeichenkette ausgeben LEN	648
13.9.2 Maximale Länge einer Zeichenkette ausgeben MAX_LEN	648
13.9.3 Zeichenketten zusammenfassen CONCAT	649
13.9.4 Linke Teilzeichenkette ausgeben LEFT	650
13.9.5 Rechte Teilzeichenkette ausgeben RIGHT	650
13.9.6 Mittlere Teilzeichenkette ausgeben MID	650
13.9.7 Teilzeichenkette löschen DELETE	651
13.9.8 Zeichenkette einfügen INSERT	651
13.9.9 Teilzeichenkette ersetzen REPLACE	653
13.9.10 Teilzeichenkette finden FIND	653

14 Programmsteuerung	655
14.1 Sprungfunktionen	656
14.1.1 Einführung	656
14.1.2 Absoluter Sprung	656
14.1.3 Bedingte Sprungfunktionen	658
14.1.4 Sprungliste	660
14.1.5 Sprungverteiler	662
14.1.6 Schleifensprung	662
14.2 Aufruf von Codebausteinen	664
14.2.1 Allgemeines zu Bausteinaufrufen	664
14.2.2 Aufruf einer Funktion FC	665
14.2.3 Aufruf eines Funktionsbausteins FB	667
14.3 Bausteinende-Funktionen	669
14.3.1 Bausteinende-Funktion RET (KOP und FUP)	670
14.3.2 RETURN-Anweisung (SCL)	670
14.3.3 Bausteinende-Funktionen BEB, BEA und BE (AWL)	670
14.4 Datenbausteinfunktionen	672
14.4.1 Datenbausteinattribute lesen	672
14.4.2 Ladespeicher lesen und schreiben	673
14.4.3 ARRAY-Datenbausteine	675
14.4.4 Systembausteine für den Zugriff auf ARRAY-Datenbausteine	676
14.4.5 CPU-Datenbausteine	678
15 Online-Betrieb und Programmtest	681
15.1 Programmiergerät an die PLC-Station anschließen	682
15.1.1 IP-Adressen des Programmiergeräts	682
15.1.2 Das Programmiergerät mit der PLC-Station verbinden	683
15.1.3 Eine IP-Adresse der CPU-Baugruppe zuweisen	685
15.1.4 Den Online-Betrieb einschalten	685
15.1.5 Die CPU urlöschen	686
15.1.6 Den Auslieferungszustand wiederherstellen	687
15.2 Projektdaten übertragen	688
15.2.1 Die Projektdaten erstmalig laden	688
15.2.2 Die Projektdaten nachladen	691
15.2.3 Das Anwenderprogramm schützen	691
15.2.4 Mit Online-Projektdaten arbeiten	695
15.2.5 Mit der Memory Card arbeiten	696
15.3 Mit Bausteinen im Online-Betrieb arbeiten	697
15.3.1 Einführung	697
15.3.2 Einen Baustein ändern und laden	698
15.3.3 Laden ohne Reinitialisierung	701
15.3.4 Einen Baustein aus der CPU zurückladen	702
15.3.5 Mit Einstellwerten arbeiten	703
15.3.6 Bausteine vergleichen	705

15.4 Hardware-Diagnose	708
15.4.1 Status-Anzeigen an den Baugruppen	708
15.4.2 Diagnose-Informationen	709
15.4.3 Diagnosepuffer	710
15.4.4 Diagnose-Funktionen	711
15.4.5 Online-Tools	712
15.4.6 Weitere Diagnose-Informationen über das Programmiergerät	713
15.5 Anwenderprogramm testen	714
15.5.1 Aufrufumgebung definieren	714
15.5.2 Testen mit Programmstatus	715
15.5.3 PLC-Variablen beobachten	718
15.5.4 Datenvariablen beobachten	720
15.5.5 Testen mit Beobachtungstabellen	721
15.5.6 Testen mit der Force-Tabelle	725
15.6 Messwertaufzeichnung mit der Trace-Funktion	728
15.6.1 Einführung	728
15.6.2 Erstellen der Trace-Konfiguration	728
15.6.3 Trace laden und aufzeichnen	729
15.6.4 Messungen speichern und auswerten	730
16 Dezentrale Peripherie	733
16.1 Einführung, Übersicht	733
16.2 Dezentrales Peripheriesystem ET 200	734
16.2.1 ET 200MP	734
16.2.2 ET 200M	735
16.2.3 ET 200SP	735
16.2.4 ET 200S	736
16.2.5 ET 200pro	737
16.2.6 ET 200eco und ET200eco PN	738
16.3 PROFINET IO	738
16.3.1 Komponenten von PROFINET IO	738
16.3.2 Adressen bei PROFINET IO	740
16.3.3 PROFINET IO projektieren	743
16.3.4 Kopplungsbaugruppen für PROFINET IO	746
16.3.5 Echtzeit-Kommunikation bei PROFINET	748
16.3.6 Spezielle PROFINET-Konfigurationen	752
16.4 PROFIBUS DP	755
16.4.1 Komponenten von PROFIBUS DP	755
16.4.2 Adressen bei PROFIBUS DP	758
16.4.3 PROFIBUS DP projektieren	760
16.4.4 Kopplungsbaugruppen für PROFIBUS DP	764
16.4.5 Spezielle PROFIBUS-Konfigurationen	767
16.5 Systembausteine für dezentrale Peripherie	769
16.5.1 Nutzdaten lesen und schreiben	769
16.5.2 Diagnosedaten von einem DP-Normslave lesen	772

16.5.3 Datensatz empfangen und bereitstellen	773
16.5.4 Dezentrale Station aktivieren/deaktivieren	775
16.6 DPV1-Alarme	776
16.7 Taktsynchronität	778
16.7.1 Einführung	778
16.7.2 Taktsynchronität bei PROFINET	778
16.7.3 Taktsynchronität bei PROFIBUS	782
16.7.4 Taktsynchronalarm	784
16.7.5 Prozessabbilder taktsynchron aktualisieren	786
17 Kommunikation	788
17.1 Übersicht	788
17.2 Open User Communication	791
17.2.1 Grundlagen	791
17.2.2 Datenstruktur der Open User Communication	792
17.2.3 Verbindung aufbauen und Daten senden mit TSEND_C	793
17.2.4 Verbindung aufbauen und Daten empfangen mit TRCV_C	794
17.2.5 Open User Communication projektieren	796
17.2.6 Weitere Funktionen der Open User Communication	798
17.3 S7-Kommunikation	801
17.3.1 Grundlagen	801
17.3.2 Einseitiger Datenaustausch	802
17.3.3 Zweiseitiger Datenaustausch	804
17.3.4 S7-Kommunikation projektieren	807
17.4 Punkt-zu-Punkt-Kommunikation	808
17.4.1 Einführung in die Punkt-zu-Punkt-Kommunikation	808
17.4.2 Konfigurieren der Kommunikationsbaugruppe CM PtP	809
17.4.3 Punkt-zu-Punkt-Kommunikationsfunktionen	810
17.5 Weitere Kommunikationsfunktionen	814
17.5.1 USS-Protokoll für Antriebe	814
17.5.2 Modbus RTU	816
17.5.3 Modbus TCP	818
18 Anhang	821
18.1 Arbeiten mit Quelldateien	821
18.1.1 Allgemeines Vorgehen	821
18.1.2 Einen Codebaustein in der Quelldatei programmieren	824
18.1.3 Einen Datenbaustein in der Quelldatei programmieren	830
18.1.4 Einen PLC-Datentyp in der Quelldatei programmieren	832
18.2 Projekte migrieren und hochrüsten	834
18.2.1 Ein Projekt migrieren	834
18.2.2 Ein Projekt hochrüsten	837
18.3 Webserver	838
18.3.1 Webserver aktivieren	838
18.3.2 Web-Informationen auslesen	838

18.3.3 Standard-Webseiten	839
18.3.4 Service-Daten auslesen	842
18.3.5 Webserver initialisieren und Webseiten synchronisieren (WWW) ...	842
18.4 Technologiefunktionen	843
18.4.1 Technologiebaugruppen TM Count 2×24V und TM PosInput 2	843
18.4.2 Technologieobjekte für Zählen und Messen	846
18.4.3 Technologieobjekte für Motion Control	849
18.4.4 Technologieobjekte für PID Control	853
18.5 Daten protokollieren und Rezepturen übertragen	855
18.5.1 Einführung in die Datenprotokollierung	855
18.5.2 Datenprotokollierung anwenden	856
18.5.3 Funktionen für die Datenprotokollierung	856
18.5.4 Einführung zur Rezepturenübertragung	859
18.5.5 Funktionen für die Rezepturenübertragung	860
18.6 Simulation mit PLCSIM	861
18.6.1 Unterschiede zu einer realen CPU-Baugruppe	862
18.6.2 PLCSIM installieren	862
18.6.3 Simulation starten und speichern	863
18.6.4 Mit der SIM-Tabelle testen	865
18.6.5 Mit der Sequenztabelle testen	866
18.6.6 Die Testfunktionen von STEP 7 anwenden	868
Stichwortverzeichnis	870

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).

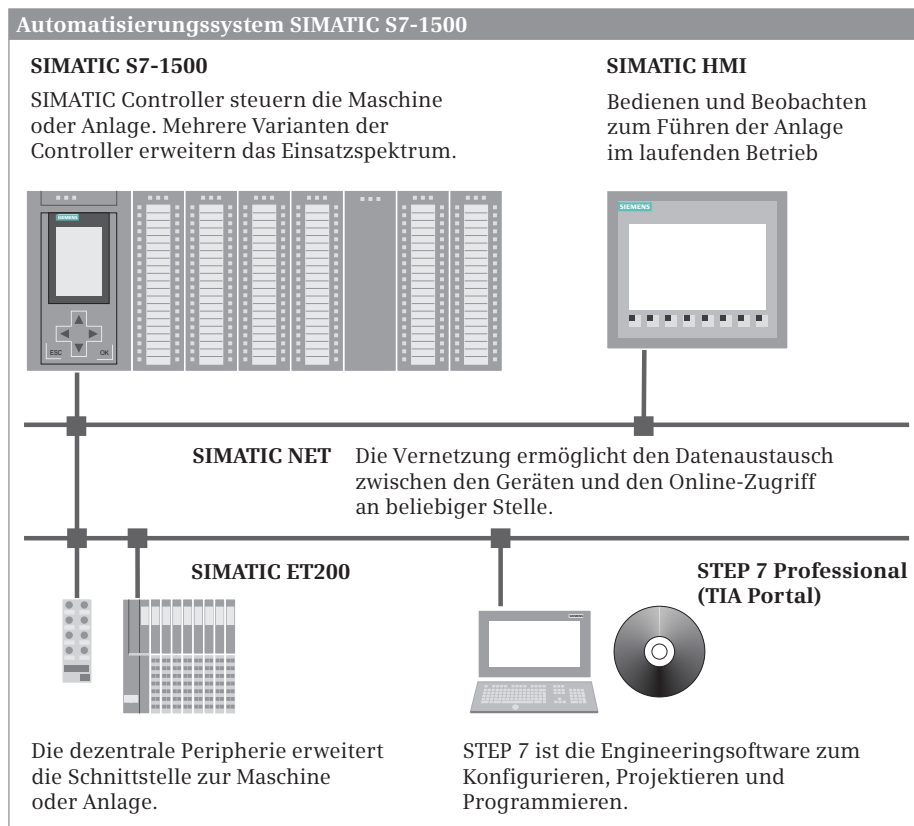


Bild 1.1 Bestandteile des Automatisierungssystems SIMATIC S7-1500

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.

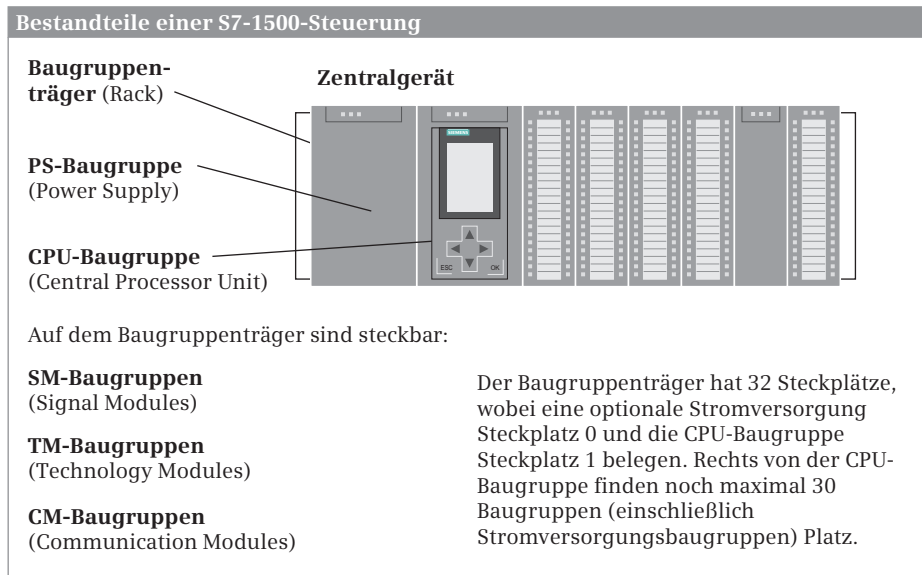


Bild 1.2 Bestandteile einer S7-1500-Steuerung

Die **CPU-Baugruppe** 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 V12

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-

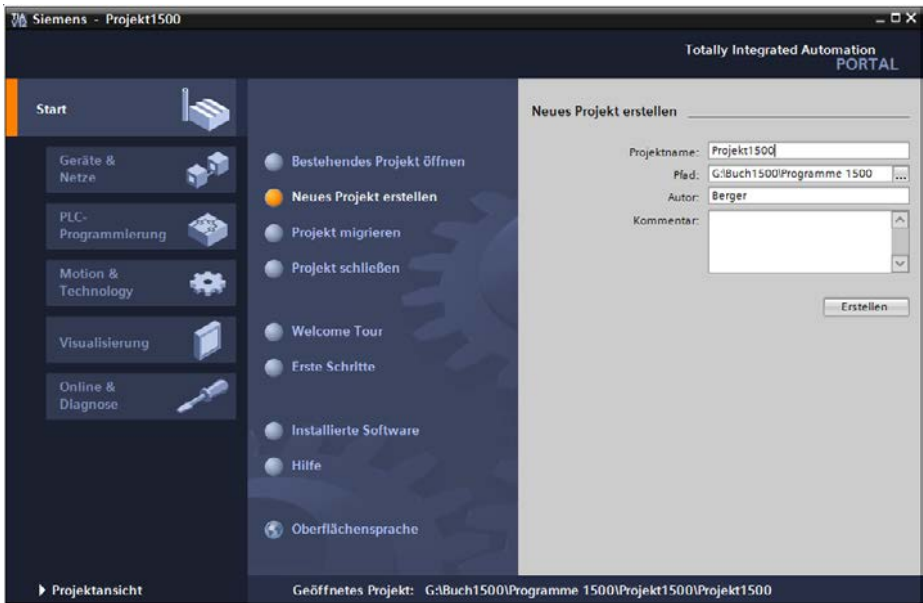


Bild 1.3 Werkzeuge im Startportal von STEP 7 Professional V12

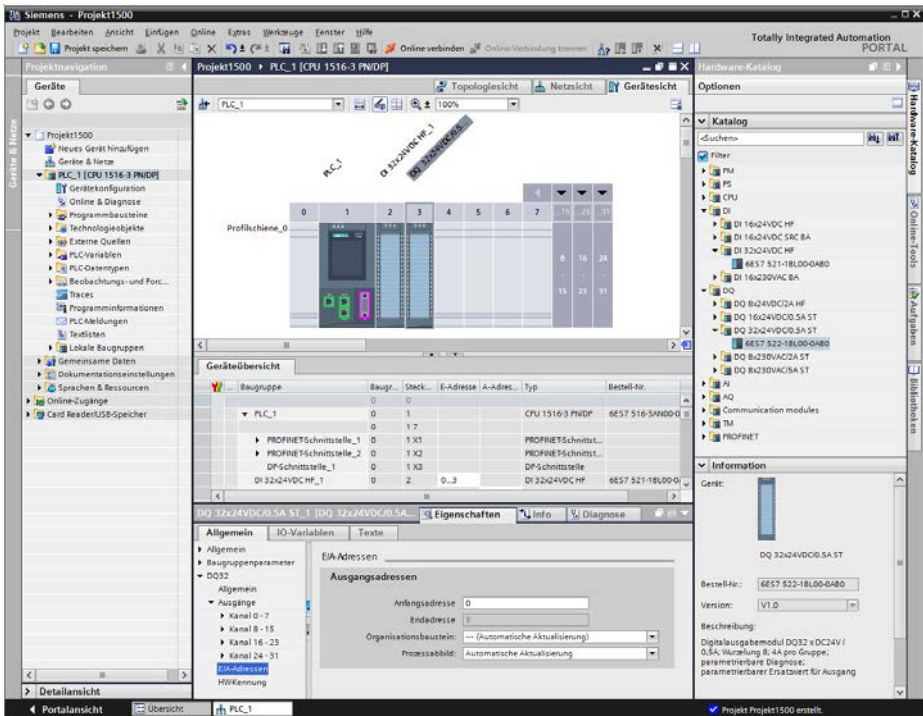


Bild 1.4 Beispiel für die Projektansicht: Arbeitsbereich der Gerätekonfiguration

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.

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.

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 (Bild 1.5). Komplexe Funktionen, wie beispielsweise die arithmetischen Funktionen, werden mit Boxen dargestellt, die Sie wie Kontakte oder Spulen im Kontaktplan anordnen.

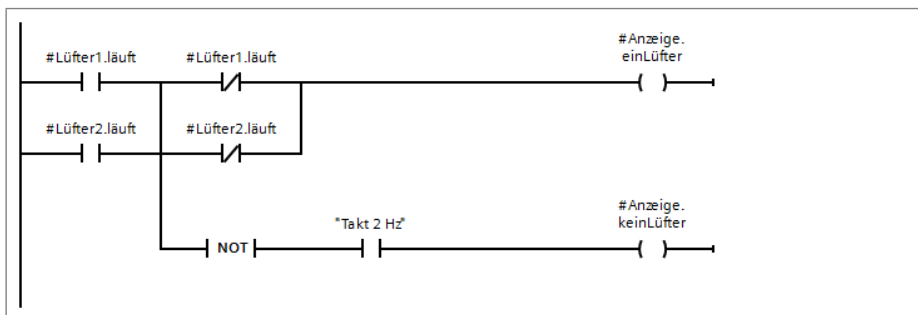


Bild 1.5 Beispiel für die Darstellung in Kontaktplan

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 (Bild 1.6). Komplexe Boxen übernehmen die Verknüpfung digitaler Variablen, beispielsweise bei den arithmetischen Funktionen.

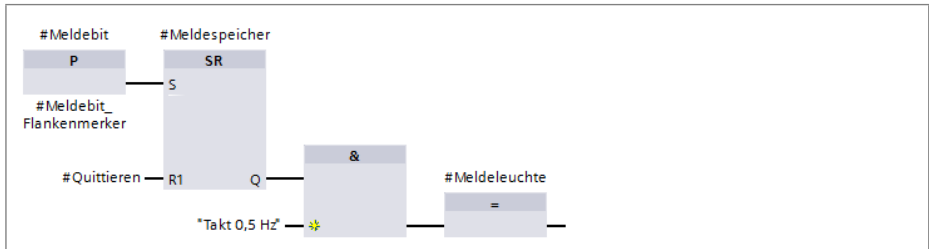


Bild 1.6 Beispiel für die Darstellung in Funktionsplan

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 (Bild 1.7).

```

18  (*****)
19  Register_schreiben:
20  IF #Füllstand = #Registerlänge - 1
21  THEN #Voll := TRUE;
22  ELSE #Register[#Schreibzeiger] := #Eingangswert;
23       #Füllstand := #Füllstand + 1;
24  IF #Schreibzeiger = #Registerlänge
25  THEN #Schreibzeiger := 0;
26  ELSE #Schreibzeiger := #Schreibzeiger + 1;
27  END IF;
28  #Leer := FALSE;
29  END_IF; RETURN;
30  (*****)

```

Bild 1.7 Beispiel für SCL-Anweisungen

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 1.8).

Mit **GRAPH** programmieren Sie eine Steuerungsaufgabe als Ablaufsteuerung, in der die sequenzielle Folge von Aktionen vorherrscht. Die einzelnen Schritte und Verzweigungen werden durch Weberschaltbedingungen freigegeben, die mit KOP oder FUP programmiert werden können (Bild 1.9).