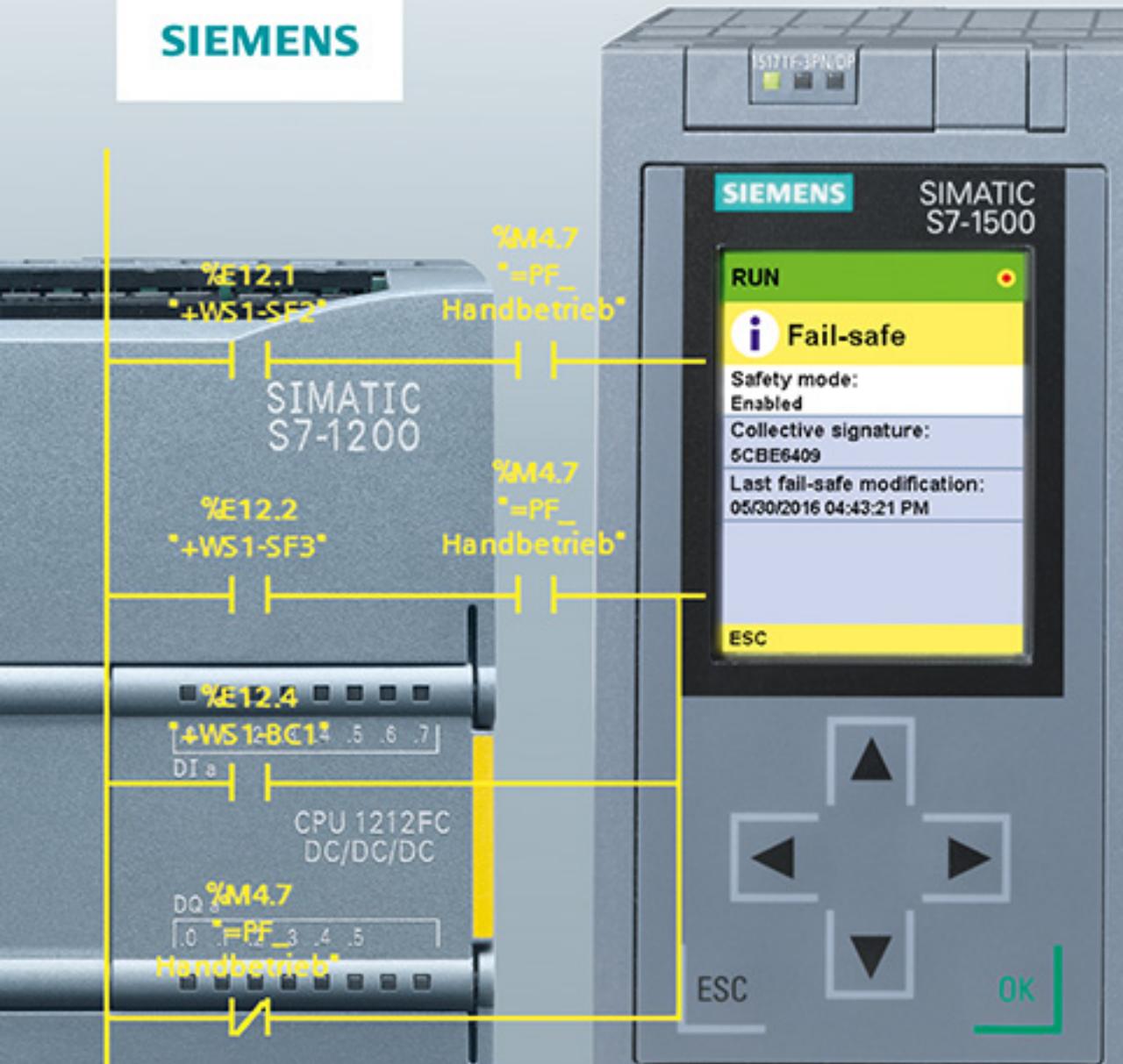


SIEMENS



Hans Berger

Automatisieren mit KOP im TIA Portal

Programmieren und Testen mit STEP 7
für SIMATIC S7-1200 und S7-1500

%M8.1
*=-WS1_MA1
Komm*

Berger Automatisieren mit KOP im TIA Portal

Automatisieren mit KOP im TIA Portal

Programme gestalten mit STEP 7
für SIMATIC S7-1200 und S7-1500

von Hans Berger

Publicis Pixelpark

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

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

Lektorat: Dr. Gerhard Seifudem, gerhard.seifudem@publicispixelpark.de
www.publicis.de/books

Print ISBN 978-3-89578-486-6

ePDF ISBN 978-3-89578-973-1

EPUB ISBN 978-3-89578-739-3

Herausgeber: Siemens Aktiengesellschaft, Berlin und München

Verlag: Publicis Pixelpark, Erlangen

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

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

Printed in Germany

Vorwort

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

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

Das vorliegende Buch beschreibt das Programmieren mit Kontaktplan, wobei die Gestaltung des Anwenderprogramms im Vordergrund steht. Für die auszuführenden Funktionen (die Steuerung der Aktoren) sind einfache, leicht verständliche Beispiele gewählt worden. Es wird gezeigt, wie einfache, „kleine“ Programme in linearer Struktur erstellt werden können, in denen nur Eingänge, Ausgänge und Merker benötigt werden. Und wie dann – darauf aufbauend – die Steuerungsfunktionen immer komplexer werden und das Programm entsprechend mit Globaldaten, Zeit- und Zählfunktionen erweitert wird. Mit der Erstellung eigener Funktionen zur mehrfachen Verwendung ist die lineare Programmstruktur ausgereizt.

Die modulare Programmstruktur basiert auf Funktionsbausteinen, die in Form eines „Unterprogramms“ abgegrenzte Teile des Anwenderprogramms enthalten. Die Funktionsbausteine sind parametrierbar und somit flexibel und mehrfach einsetzbar. Man kann einzelnen Anlagenobjekten Funktionsbausteine zuordnen, diese „bündeln“ und so mit übergeordneten Funktionsbausteinen ganze Gruppen von Anlagenteilen steuern. Für die Datenübergabe an Funktionsbausteine gibt es mächtige Datentypen, wie beispielsweise Zeiger für unterschiedliche Datentypen, Felder mit variablen Grenzen und die Beschaltung mit Instanzdaten.

Im nächsten Teil des Buchs werden alle Datentypen und die zu ihrer Bearbeitung verfügbaren Funktionen ausführlich beschrieben. Eine Einführung in den Online-Betrieb, die Behandlung der Projektdaten in einer PLC-Station und das Testen des Anwenderprogramms mit STEP 7 und S7-PLCSIM runden das Buch ab.

Die Beispielprogramme sind mit STEP 7 Professional Version 15.1 (TIA Portal) in Kontaktplan (KOP) erstellt und für die Controller S7-1200 (Firmware Version 4.2) und S7-1500 (Firmware Version 2.6) ausgelegt.

Der Inhalt des Buchs auf einen Blick

Hardware und Software

Das Buch beschreibt die Programmierung mit Kontaktplan (KOP) für die Controller S7-1200 und S7-1500 des Automatisierungssystems SIMATIC. Die Anwenderprogramme und Variablenbereiche dieser Controller sind weitgehend identisch. Die Programmierung erfolgt mit der Engineering-Software STEP 7 im TIA Portal mit umfangreicher Unterstützung für die Hardware-Projektierung und Programmerstellung. Die Hardware- und Software-Voraussetzungen für die Erstellung von Anwenderprogrammen mit Kontaktplan erläutert der erste Teil des Buchs.

- 1 Controller SIMATIC S7-1200/1500** – Stellt die relevanten technischen Spezifikationen der Controller vor und gibt einen Überblick über Operandenbereiche, Adressierung und Datentypen.
- 2 Anwenderprogramm** – Erläutert die Programmbearbeitung, Programmstruktur und verwendete Bausteine und zeigt, mit welchen Alarm- und Fehlerprogrammen der Controller arbeitet.
- 3 Einführung in STEP 7 (TIA Portal)** – Beschreibt die Benutzeroberfläche von STEP 7, bietet eine Einführung in die Hardware-Konfiguration und stellt die Schritte für die Erstellung eines Projekts vor. Zur Vertiefung wird das Beispielprojekt „1Projekt“ angelegt.
- 4 Programminformationen** – STEP 7 bietet umfangreiche Unterstützung durch Vergleichseditoren, Querverweislisten, Belegungsplan, Darstellung von Aufruf- und Abhängigkeitsstruktur sowie der Speicherauslastung.

Struktur von KOP-Programmen

Die Controller sind beinahe universell einsetzbar, von der kleinsten Anwendung bis zur Steuerung komplexer Maschinen. Das zeigt sich in der Struktur der Anwenderprogramme: Die Möglichkeiten reichen von einfachen Programmen mit linearer Struktur bis zu modular aufgebauten Programmen mit gekapselten, wiederverwendbaren Programmteilen. Der zweite Teil des Buchs zeigt, wie ein einfaches Programm mit linearer Struktur schrittweise erweitert werden kann bis zur Verwendung von Software Units.

- 5 Kontaktplan** – Zeigt das Arbeiten mit Netzwerken, die Platzierung und Beschriftung von Programmelementen sowie das Editieren der Variablen und erläutert die Aufgabenstellung und die Voraussetzungen der Beispielprogramme. Den Einstieg bildet das Programmbeispiel „2Grundlagen“, in dem bei linearer Programmstruktur nur Eingänge, Ausgänge und Merker verwendet werden. Auf den Seiten 126 und 127 befindet sich eine Übersicht der auf die Beispielprogramme bezogenen Bilder und Tabellen.
- 6 Globaldaten** – Globaldaten sind im gesamten Anwenderprogramm gültig und können mit jedem Datentyp, beispielsweise als Datenfeld ARRAY, deklariert werden. Sie sind in Datenbausteinen zusammengefasst, die im Arbeitsspeicher oder im Ladespeicher liegen. Im Programmbeispiel „3Globaldaten“, das auf das Beispiel „2Grundlagen“ aufbaut, werden Globaldaten als Instanzdaten für Zeit-/Zählfunktionen und als Statuswörter verwendet.
- 7 Funktionen** – Beschreibt die Mehrfachverwendung von Programmteilen als Funktionen (FC), die Deklaration von Bausteinschnittstellen, die Übergabe von Bausteinparametern, Fehlerbehandlung mit dem EN/ENO-Mechanismus, implizite und explizite Datenkonvertierung sowie die Arbeit mit Programm- und Sprungfunktionen. Im Beispielprogramm „4Funktionen“ werden einige neu hinzugekommene Funktionen als FC-Bausteine realisiert. Damit ist die lineare Programmstruktur ausgereizt.

- 8 Funktionsbausteine** – Funktionsbausteine sind bestens geeignet, ein Programm modular und übersichtlich aufzubauen. Sie haben einen eigenen Variablenpeicher, in dem die für jeden Aufruf relevanten Variablenwerte – die Instanzdaten – dauerhaft abgelegt sind. Das Beispielprogramm „5Bausteine“ zeigt bei gleicher Funktionalität wie „4Funktionen“ die modulare Programmstruktur mithilfe von Funktionsbausteinen. Software Units (für S7-1500) sind eigenständige, abgegrenzte Programmeinheiten, die voneinander unabhängig bearbeitet werden können. Das Beispielprogramm „6SoftwareUnits“ zeigt, wie Software Units zur Programmstrukturierung verwendet werden.
- 9 PLC-Datentypen und Parametertypen** – Eine modulare Programmstruktur erfordert einen ungehinderten, jedoch kontrollierten Transfer von Variablenwerten zwischen den einzelnen Bausteinen und Programmteilen. Vom Anwender selbst definierte PLC-Datentypen und Parametertypen wie DB_ANY, ARRAY[*], VARIANT und REFERENCE (S7-1500) ermöglichen einen freizügigen Datenaustausch, wie die Beispiele in diesem Kapitel zeigen.

Datentypen und Programmelemente

Der dritte Teil des Buchs widmet sich den Programmelementen des Kontaktplans, geordnet nach Datentypen. Es wird gezeigt, mit welchen Programmelementen (Programmfunktionen) die Variablen eines bestimmten Datentyps bearbeitet werden können. Beispiele, teilweise auch aus den Beispielprogrammen des zweiten Teils, zeigen die Anwendung.

- 10 Binärsignale** – Programmelemente für die Verknüpfung von Signalen mit den Signalzuständen „0“ und „1“ sind die Kontakte und Spulen in Reihen- und Parallelschaltung, Speicherfunktionen und Flankenbewertungen.
- 11 Bitfolgen** – In Bitfolge-Variablen sind mehrere Bits ohne Wertigkeit zusammengefasst, die mit Wortverknüpfungen, Schiebe- und Codierfunktionen bearbeitet werden. Das Kapitel geht auch auf die Byte-Reihenfolge im Speicher und die Slice-Adressierung ein.
- 12 Ganzzahlen und Gleitpunktzahlen** – Für das Rechnen mit Zahlenwerten stehen arithmetische und mathematische Funktionen zur Verfügung. Zahlenwert-Variablen können übertragen, verglichen und umgewandelt werden. Das Beispiel zur Berechnung des Median zeigt die Behandlung von Datenfeldern bei KOP.
- 13 Zeit- und Zählfunktionen** – Zeigt, welche Funktionen und Datentypen für Zeiten und Zähler zur Verfügung stehen, und informiert über die Uhrzeitfunktionen der CPU-Baugruppe.
- 14 Zeichen und Zeichenketten** – Beschreibt die Datentypen für Zeichen und Zeichenketten und wie Variablen mit diesen Datentypen bearbeitet und umgewandelt werden. Ein Beispiel zeigt, wie die Aufrufumgebung eines Funktionsbausteins ermittelt werden kann.

Online-Betrieb und Programmtest

Projektieren und Programmieren erfolgen offline ohne Controller-Hardware. Erst bei der Inbetriebnahme ist das „reale“ Zielgerät erforderlich. Der vierte Teil des Buchs beschreibt die Behandlung von Offline- und Online-Projektdateien und das Testen des Anwenderprogramms sowohl an einem „realen“ Controller, als auch mit der Simulationssoftware S7-PLCSIM.

- 15 Online-Betrieb** – Zeigt, wie ein Programmiergerät angeschlossen und Projektdateien übertragen werden. Erläutert den Umgang mit Aktualwerten, Einstellwerten und Momentaufnahmen. Beschreibt die Anwendung der Online-Tools und den Diagnosepuffer.
- 16 Programmtest** – Zeigt, wie man den Programmablauf und Variablen beobachtet und steuert – mittels Programmstatus, Beobachtungstabellen, Force-Tabelle sowie einem simulierten Automatisierungsgerät (PLCSIM).

Inhaltsverzeichnis

1 Controller SIMATIC S7-1200/1500	12
1.1 Übersicht Automatisierungssystem	12
1.2 Basic Controller SIMATIC S7-1200C	14
1.3 Advanced Controller SIMATIC S7-1500	16
1.4 Anwenderspeicher und Memory Card	18
1.5 Signalbaugruppen	20
1.6 Eingänge, Ausgänge, Merker	21
1.7 Übersicht Adressierung	25
1.8 Übersicht Datentypen	27
2 Anwenderprogramm	29
2.1 Programmbearbeitung	29
2.2 Anlauf- und Hauptprogramm	38
2.3 Alarmprogramm	40
2.4 Asynchronfehlerprogramm	47
2.5 Synchronfehlerprogramm	50
3 Einführung in STEP 7 (TIA Portal)	54
3.1 STEP 7 installieren und starten	54
3.2 Benutzeroberfläche	55
3.3 Ein Projekt bearbeiten	61
3.4 Ein Gerät konfigurieren	65
3.5 Die PLC-Variablen-tabelle ausfüllen	69
3.6 Das Anwenderprogramm erstellen	72
3.7 Beispielprogramm „1Projekt“	82
4 Programminformationen	87
4.1 Projektdaten vergleichen	87
4.2 Querverweisliste	92
4.3 Belegungsplan	95
4.4 Aufrufstruktur	96
4.5 Abhängigkeitsstruktur	97
4.6 Speicherauslastung der CPU	99

5 Kontaktplan	101
5.1 KOP-Programmierung allgemein	101
5.2 Programmelemente	104
5.3 Steuerungsfunktion programmieren	106
5.4 Variablen und Konstanten editieren	110
5.5 Grundsätzliche Angaben zu den Beispielen	113
5.6 Beispielprogramm „2Grundlagen“	117
6 Globaldaten	128
6.1 Mit Datenbausteinen arbeiten	129
6.2 Datenvariablen in Global-Datenbausteinen	134
6.3 Datenbausteine im Ladespeicher	139
6.4 Datentyp ARRAY	144
6.5 ARRAY-Datenbausteine (S7-1500)	149
6.6 Beispielprogramm „3Globaldaten“	153
7 Funktionen	160
7.1 Eine Funktion (FC) programmieren	160
7.2 Bausteinschnittstelle	165
7.3 Übergabe von Bausteinparametern	168
7.4 EN/ENO-Mechanismus	172
7.5 Datentypkonvertierung	173
7.6 Funktionen im Programmelemente-Katalog	176
7.7 Sprungfunktionen	178
7.8 Beispielprogramm „4Funktionen“	182
8 Funktionsbausteine	192
8.1 Einen Funktionsbaustein programmieren	193
8.2 Instanzdaten anlegen	199
8.3 Beispielprogramm „5Bausteine“	204
8.4 Software Units (S7-1500)	212
8.5 Beispielprogramm „6SoftwareUnits“	216
9 PLC-Datentypen und Parametertypen	220
9.1 PLC-Datentypen	220
9.2 Parametertyp DB_ANY	227
9.3 Parametertyp ARRAY[*]	230
9.4 Parametertyp VARIANT	232
9.5 Parametertyp REFERENCE (S7-1500)	240

10 Binärsignale	245
10.1 Wissenswertes zu Binärsignalen	245
10.2 Binäre Verknüpfungen	248
10.3 Speicherfunktionen	250
10.4 Flankenauswertungen	253
10.5 Beispiele zu den Binärfunktionen	258
11 Bitfolgen	261
11.1 Datentypen für Bitfolgen und BCD-Zahlen	261
11.2 Bitfolge-Datentypen konvertieren	262
11.3 Byte-Reihenfolge	263
11.4 Slice-Adressierung	264
11.5 Wortverknüpfungen	265
11.6 Schiebefunktionen	266
11.7 Codierfunktionen	268
11.8 Beispiel zu den Bitfolge-Datentypen	269
12 Ganzzahlen und Gleitpunktzahlen	271
12.1 Datentypen für Ganzzahlen	271
12.2 Datentypen für Gleitpunktzahlen	272
12.3 Übertragungsfunktionen	274
12.4 Arithmetische Funktionen	277
12.5 Mathematische Funktionen	280
12.6 Rechnen mit der CALCULATE-Box	283
12.7 Vergleichsfunktionen	285
12.8 Zahlen-Datentypen konvertieren	288
12.9 Auswahl- und Begrenzerfunktionen	292
12.10 Beispiel zu Zahlenwerten	295
13 Zeit- und Zählerfunktionen	301
13.1 Instanzdaten für Zeit- und Zählerfunktionen	301
13.2 Zeitfunktionen	302
13.3 Datentypen für Zeitdauern	309
13.4 Datentypen für Zeitpunkte	310
13.5 Zeit-Datentypen konvertieren	312
13.6 Zeitdauern und Zeitpunkte bearbeiten	315
13.7 Uhrzeitfunktionen der CPU-Baugruppe	317
13.8 Beispiele zu den Zeitfunktionen	324
13.9 Zählerfunktionen	326
13.10 Beispiel zu den Zählerfunktionen	331

14 Zeichen und Zeichenketten	333
14.1 Datentypen für einzelne Zeichen	333
14.2 Datentypen für Zeichenketten	334
14.3 Mit Zeichen und Zeichenketten arbeiten	334
14.4 Zeichen-Datentypen konvertieren	336
14.5 Zeichenketten bearbeiten	343
14.6 Symbolnamen lesen	349
14.7 Beispiel zu den Zeichenformaten	351
15 Online-Betrieb	353
15.1 Programmiergerät anschließen	353
15.2 Projektdaten übertragen	355
15.3 Mit Online-Projektdaten arbeiten	357
15.4 Mit Bausteinen im Online-Betrieb arbeiten	359
15.5 Mit Datenvariablen im Online-Betrieb arbeiten	361
15.6 Diagnose im Online-Betrieb	366
16 Programmtest	371
16.1 Testen im Programmstatus	371
16.2 PLC-Variablen beobachten	375
16.3 Datenvariablen beobachten und steuern	375
16.4 Testen mit Beobachtungstabellen	377
16.5 Testen mit der Force-Tabelle	381
16.6 Peripherie-Ausgänge freischalten (S7-1200)	384
16.7 Testen mit S7-PLCSIM	385
Stichwortverzeichnis	395

1 Controller SIMATIC S7-1200/1500

1.1 Übersicht Automatisierungssystem

Das Industrie-Automatisierungssystem SIMATIC besteht aus Komponenten, die durch das Konzept „Totally Integrated Automation“ (TIA) aufeinander abgestimmt sind. Totally Integrated Automation bedeutet Automatisierung mit durchgängiger Projektierung, Programmierung, Datenhaltung und Datenübertragung (Bild 1.1).

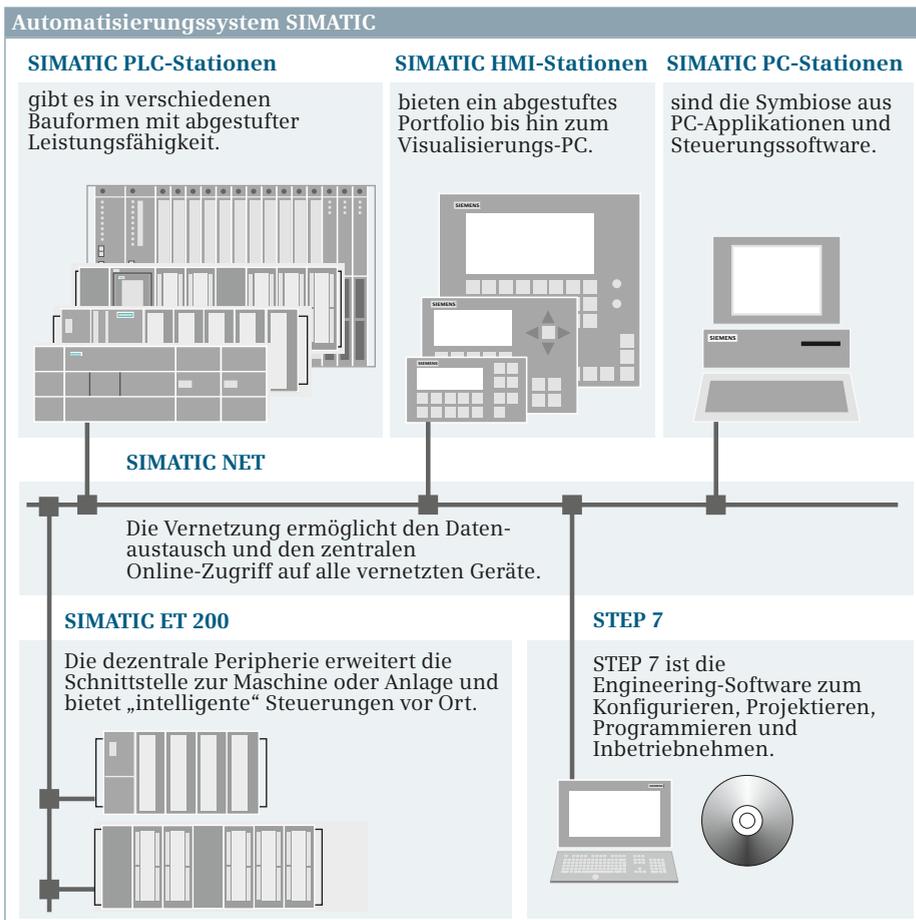


Bild 1.1 Die Komponenten des Automatisierungssystems SIMATIC

Die Controller **SIMATIC S7** bilden als speicherprogrammierbare Steuerungen (SPS) die Basis des Automatisierungssystems. Die Basic Controller SIMATIC S7-1200 sind die richtige Wahl für einfache und autark zu lösende Aufgaben im unteren bis mittleren Leistungsbereich. Die Advanced Controller SIMATIC S7-1500 sichern höchste Leistungsfähigkeit und Benutzerfreundlichkeit für mittlere und High-End-Anwendungen der Maschinen- und Anlagenautomatisierung.

SIMATIC ET 200 erweitert die Schnittstelle zwischen zentralem Controller und Maschine oder Anlage um Peripheriebaugruppen direkt vor Ort. Sind autark arbeitende, „intelligente“ Steuerungen auf der Feldebene notwendig, kommen die Distributed Controller ET 200 CPU zum Einsatz. SIMATIC ET 200 bietet ein multifunktionales, modulares und fein skalierbares System für die dezentrale Automatisierung: für Lösungen im Schaltschrank oder ohne Schaltschrank direkt an der Maschine sowie für den Einsatz im explosionsgefährdeten Bereich.

SIMATIC PC-based Automation ist die ideale Basis, um die Kombination aus dem Betriebssystem Microsoft Windows sowie aus Windows-Applikationen und SIMATIC-Steuerungssoftware effizient und wirtschaftlich zu realisieren. Die Software Controller mit dem Leistungsumfang der Controller-Familie S7-1500 eignen sich besonders für eine flexible Steuerung von Sondermaschinen mit hohen Performance- und Funktionsanforderungen.

SIMATIC HMI bedeutet Bedienen und Beobachten mit SIMATIC Panels in vielen verschiedenen Leistungsklassen. Leistungsfähige HMI-Software zeigt mit Betriebs- und Störungsmeldungen den Anlagenzustand, verwaltet Rezepturen und Messwertarchive und unterstützt den Anlagenbetreiber bei Fehlersuche, Wartung und Instandhaltung.

SIMATIC NET verbindet alle SIMATIC-Stationen und sorgt für komplikationslose Industriekommunikation. Verschiedene Bussysteme mit abgestufter Leistungsfähigkeit gestatten die Kopplung auch zu Fremdgeräten, seien es Feldgeräte in der Anlage oder Prozessrechner in der Leitebene. Der Datenverkehr funktioniert auch über die Grenzen verschiedener Subnetze hinweg, beispielsweise die Übertragung von Automatisierungsdaten wie Messwerte und Meldungen oder die Inbetriebsetzung und Störungssuche ausgehend von einer zentralen Stelle im Netzwerk.

STEP 7 (TIA Portal) ist die Engineering-Software, mit der alle SIMATIC-Komponenten konfiguriert, projiziert, parametrisiert und programmiert werden. Es ist das zentrale Werkzeug zum Verwalten der Automatisierungsdaten und der dazugehörigen Editoren in Form eines hierarchisch gegliederten Projekts. Unter anderem fügen Sie mit STEP 7 dem Projekt eine (weitere) PLC-Station mit einem Controller hinzu, bestücken die Station mit Baugruppen und stellen die Baugruppeneigenschaften ein. Sie erstellen eine Variablenliste mit den Schnittstellensignalen zur Anlage oder Maschine. Und nicht zuletzt programmieren, ändern, testen und archivieren Sie das Steuerungsprogramm (das Anwenderprogramm).

Das Anwenderprogramm kann in den Programmiersprachen Kontaktplan (KOP), Funktionsplan (FUP) und Structured Control Language (SCL) erstellt werden. Bei S7-1500 stehen zusätzlich Anweisungsliste (AWL) und Ablaufsteuerung (GRAPH) zur Verfügung.

1.2 Basic Controller SIMATIC S7-1200C

Aufbauvarianten

Der zentrale Baugruppenträger einer S7-1200-Station besteht aus der Profilschiene mit zwölf Steckplätzen und der CPU-Baugruppe auf Steckplatz 1. Links von der CPU-Baugruppe können bis zu drei Kommunikationsgruppen gesteckt werden (Steckplätze 101, 102 und 103). Rechts von der CPU-Baugruppe werden die Signalbaugruppen angeordnet, je nach CPU-Typ keine, zwei oder acht (Steckplätze 2 bis 9). Jede Baugruppe belegt unabhängig von ihrer Breite einen Steckplatz. Mit einem 2 m langen Erweiterungskabel kann der Aufbau zweizeilig ausgeführt werden. Die Anzahl der einsetzbaren Baugruppen ändert sich hierbei nicht (Bild 1.2).



Bild 1.2 S7-1200-Station mit CPU S7-1214C

CPU-Varianten

Ein S7-1200-Controller ist in der Standardausführung ein Kompakt-Controller (CPU 1200C für „compact“). Bei ihm sind auch Ein- und Ausgabekanäle in das Gehäuse integriert. Die CPU-Baugruppe allein kann also für Kleinanwendungen eine komplette PLC-Station sein. Die FailSafe-Controller CPU 1200F gestatten den Aufbau eines fehlersicheren Automatisierungssystems. Im Buch wird die Standard-Variante der CPU-Baugruppe beschrieben.

Die CPU-Baugruppen werden mit unterschiedlicher Leistungsfähigkeit angeboten. Sie unterscheiden sich beispielsweise in den Anschlussvarianten (DC/DC/DC, DC/DC/Relais, AC/DC/Relais), in der Größe des Anwenderspeichers, in der Bearbeitungsgeschwindigkeit, in der Anzahl der integrierten Ein-/Ausgänge und der Erweiterungsmöglichkeit mit Signalbaugruppen. Der Befehlsumfang der Programmiersprachen ist bei allen Varianten gleich.

Modulsteckplatz auf der CPU-Baugruppe

An der Frontseite der CPU-Baugruppe befindet sich ein Steckplatz, der Möglichkeiten für Erweiterungen bietet, ohne die Abmessungen der CPU-Baugruppe zu verändern. Zur Verfügung stehen Signal Boards (SB, Bild 1.3) mit Digital- und Analogkanälen, ein Communication Board (CB) für seriellen Datenaustausch und ein Battery Board (BB), um die Laufzeit der internen Uhr von typisch 20 Tagen auf bis zu einem Jahr zu verlängern.



Bild 1.3 Signal Board SB 1223

Simulatormodul

Auf die Onboard-Eingaben kann ein Simulatormodul mit Schaltern oder mit einem Potentiometer gesteckt werden für eine einfache Testmöglichkeit des Anwenderprogramms.

Schnittstellen

Alle CPU-Baugruppen haben mindestens einen Anschluss für Industrial Ethernet. Über diesen Anschluss können beispielsweise ein Programmiergerät und andere PLC- oder HMI-Stationen angeschlossen werden. In einem PROFINET-IO-System kann eine CPU 1200C der IO-Controller oder ein „intelligentes“ IO-Device sein. Für den Betrieb an PROFIBUS-DP stehen der CM 1243-5 (DP-Master) und der CM 1242-5 (DP-Slave) zur Verfügung. CM 1243-2 AS-i ermöglicht den Anschluss von Feldgeräten über Aktor/Sensor-interface.

Betriebszustände

Eine CPU 1200 kennt die Betriebszustände STOP, ANLAUF und RUN. Im STOP wird das Anwenderprogramm nicht bearbeitet, die CPU-Baugruppe ist jedoch kommunikationsfähig. Es können beispielsweise das Anwenderprogramm oder Teile davon übertragen werden oder die Online-Funktionen ausgeführt werden. Nach dem Einschalten durchläuft die CPU-Baugruppe den Betriebszustand ANLAUF, in dem die Baugruppen konfiguriert werden und ein Anlaufprogramm bearbeitet wird. Im Betriebszustand RUN schließlich bearbeitet das CPU-Betriebssystem das Hauptprogramm zyklisch immer wiederkehrend.

Mit Leuchtdioden an der Frontseite der Zentralbaugruppe wird der Betriebszustand angezeigt. Im Betriebszustand STOP leuchtet die RUN/STOP-LED gelb, im Betriebszustand RUN leuchtet sie grün und im Betriebszustand ANLAUF blinkt sie. Die rote ERROR-LED blinkt bei einem Fehler oder leuchtet dauerhaft bei defekter Hardware. Eine Wartungsanforderung wird mit der gelben MAINT-LED angezeigt. Bei defekter Firmware blinken alle LEDs.

Eine CPU 1200 hat keinen Betriebsartenschalter. Sie läuft an, wenn man sie an Spannung legt. Im Online-Betrieb können die Betriebszustände mit einem Programmiergerät über die Online-Tools eingestellt werden.

1.3 Advanced Controller SIMATIC S7-1500

Aufbauvarianten

Der zentrale Baugruppenträger einer S7-1500-Station besteht aus einer Profilschiene mit 32 Steckplätzen und der CPU-Baugruppe auf Steckplatz 2. Links von der CPU-Baugruppe kann bei Bedarf eine Systemstromversorgungsbaugruppe (PS) angeordnet werden. Auf den 30 Steckplätzen rechts der CPU-Baugruppe stecken die Peripheriebaugruppen (SM, TM, CM). Jede Baugruppe belegt unabhängig von ihrer Breite einen Steckplatz. Je nach Stromverbrauch werden die Baugruppen in bis zu drei Powersegmenten zusammengefasst, die jeweils eine eigene Systemstromversorgungsbaugruppe haben.

CPU-Varianten

Ein Standard-Controller in der typischen S7-1500-Bauform mit der Schutzart IP 20 hat ein aufklappbares, abnehmbares und verschließbares Bedienpanel. Die Standard-Controller gibt es auch in der FailSafe-Ausführung für den Einsatz in einem fehlersicheren Automatisierungssystem. Beide Ausführungen gibt es in verschiedenen Varianten, die sich beispielsweise in den Busanschlüssen, in der Größe des Anwenderspeichers und in der Bearbeitungsgeschwindigkeit unterscheiden. Im Buch wird die Standard-Variante der CPU-Baugruppe beschrieben.

Ein Kompakt-Controller S7-1500C in der typischen S7-1500-Bauform hat integrierte Ein- und Ausgabekanäle, sodass er für Kleinstanwendungen eine komplette PLC-Station sein kann. Ein Technologie-Controller in der Bauform eines Standard-Controllers bietet einen erweiterten Umfang an Motion-Control-Funktionen, wie z. B. Getriebe- und Kurvengleichlauf.



Bild 1.4 S7-1500-Station mit CPU 1516-3 PN/DP

Die redundanten Controller S7-1500R und S7-1500H werden paarweise eingesetzt, wenn die Verfügbarkeit höher als bei Standard-Controllern sein soll. Zwei identische CPU-Baugruppen bearbeiten parallel das Anwenderprogramm. Wenn eine der Baugruppen ausfällt, kann die andere das Programm weiterbearbeiten. Die CPU 1500R tauschen die Redundanzdaten über PROFINET aus, die CPU 1500H besonders schnell über eigene Lichtleiter-Verbindungen. Eine redundante Station hat keine zentralen Peripheriebaugruppen. Alle Peripheriebaugruppen sind über eine PROFINET-IO-Ringstruktur an die CPU-Baugruppen angeschlossen und müssen Systemredundanz S2 und Medienredundanz unterstützen, wie z. B. die PROFINET-IO-Devices ET 200SP IM 155 ab FW-Stand V4.2.

Weiterhin gibt es Standard- und FailSafe-Controller in der ET 200SP- und ET 200pro-Bauform sowie Software Controller für Industrie-PC und Open Controller in der Bauform ET 200SP als Software-Applikation, die unter dem Betriebssystem Microsoft Windows laufen. Der Befehlsumfang der Programmiersprachen ist in allen Varianten gleich.

Bedienpanel und Schnittstellen

Die Frontseite der CPU-Baugruppe besteht aus einem aufklappbaren, abnehmbaren und verplombbaren Bedienpanel mit einem Farbdisplay und Bedientasten. Das Farbdisplay zeigt – strukturiert in mehreren Menüs – unter anderem den Status und die Eigenschaften der CPU-Baugruppe und Informationen zu den vorhandenen Baugruppen.

Unter dem Bedienpanel liegen der Betriebsarten-schalter und die Busanschlüsse, deren Art und Anzahl von der CPU-Version abhängen. Jede CPU 1500 hat mindestens eine PROFINET-Schnittstelle mit zwei durch einen Switch verbundenen Ports. Über diesen Anschluss können beispielsweise ein Programmiergerät und andere PLC- oder HMI-Stationen angeschlossen werden. In einem PROFINET-IO-System kann eine CPU 1500 der IO-Controller oder ein „intelligentes“ IO-Device sein. Einige Controller haben einen Anschluss für den Betrieb an PROFIBUS-DP als DP-Master. Mit zusätzlichen Funktionen ausgestattet übernehmen Kommunikationsbaugruppen den Betrieb als PROFINET-IO-Controller, als PROFIBUS-DP-Master oder als PROFIBUS-DP-Slave (Bild 1.5).



Bild 1.5 Anschlüsse einer CPU 1516-3 PN/DP

Betriebszustände

Eine CPU 1500 kennt die Betriebszustände STOP, HALT, ANLAUF und RUN. Im Betriebszustand STOP wird das Anwenderprogramm nicht bearbeitet, die CPU ist jedoch kommunikationsfähig, d. h. es kann beispielsweise das Anwenderprogramm geladen oder der Diagnosepuffer ausgelesen werden.

Der Betriebszustand kann an der CPU-Baugruppe über das Display, mit dem Betriebsartenschalter oder mit einem Programmiergerät im Online-Betrieb geändert werden. Wenn man den Betriebszustand RUN einschaltet, wird einmalig der Betriebszustand ANLAUF durchlaufen, in dem die Baugruppen parametrieren werden und ein Anwender-Anlaufprogramm bearbeitet wird. Im Betriebszustand RUN findet die zyklische Bearbeitung des Anwender-Hauptprogramms statt.

Mit Leuchtdioden an der Frontseite der Zentralbaugruppe (und auf dem Display) wird der Betriebszustand angezeigt. Im Betriebszustand STOP leuchtet die RUN/STOP-LED gelb, im Betriebszustand RUN leuchtet sie grün und im Betriebszustand ANLAUF blinkt sie. Die rote ERROR-LED blinkt bei einem Fehler oder leuchtet dauerhaft bei defekter Hardware. Eine Wartungsanforderung wird mit der gelben MAINT-LED angezeigt. Bei defekter Firmware blinken alle LEDs.

Eine CPU 1500 kennt noch den Betriebszustand HALT für das Testen mit Einzelschritt bei SCL und AWL. Für die Programmierung mit Kontaktplan ist dieser Betriebszustand nicht relevant.

1.4 Anwenderspeicher und Memory Card

Ladespeicher und Arbeitsspeicher

Das Anwenderprogramm liegt auf der CPU-Baugruppe in zwei Bereichen: im Ladespeicher und im Arbeitsspeicher. Der Ladespeicher enthält das gesamte Anwenderprogramm einschließlich Konfigurationsdaten. Der Arbeitsspeicher ist ein in der CPU-Baugruppe integrierter schneller RAM-Speicher, der den ablaufrelevanten Programmcode und die Anwenderdaten enthält.

Eine CPU 1200 hat einen auf der CPU-Baugruppe integrierten Ladespeicher. Ist der interne Ladespeicher zu klein, kann auf der Memory Card ein externer Ladespeicher eingerichtet werden. In diesem Fall muss während des Betriebs die Memory Card dauerhaft gesteckt bleiben.

Eine CPU 1500 hat keinen integrierten Ladespeicher. Der Ladespeicher liegt immer auf der Memory Card. Das bedeutet, zum Betrieb benötigt eine CPU 1500 immer eine gesteckte Memory Card.

Das Programmiergerät überträgt das komplette Anwenderprogramm einschließlich der Konfigurationsdaten in den Ladespeicher. Das Betriebssystem interpretiert beim Einschalten die Konfigurationsdaten und parametrieren die Baugruppen. Der ablaufrelevante Programmcode und die Anwenderdaten werden in den Arbeitsspeicher kopiert.

Das Anwenderprogramm ist im Ladespeicher spannungsausfallsicher abgelegt. Ein zusätzlicher Remanenzspeicher kann einen Teil der Anwenderdaten aufnehmen und sorgt dafür, dass die Werte nach einem Spannungsausfall oder beim Aus- und Einschalten der Versorgungsspannung erhalten bleiben. Der Remanenzspeicher wird beim Umräumen oder beim Zurücksetzen auf die Werkseinstellungen gelöscht.

SIMATIC Memory Card

Die SIMATIC Memory Card ist eine von Siemens vorformatierte SD-Speicherkarte (Bild 1.6). Sie gibt es in den Größen von 4 MB bis 32 GB. Die Memory Card kann als Programmkarte (Ladespeicher), als Übertragungskarte (bei S7-1200) oder als Update-Karte für die CPU-Firmware verwendet werden. Vor einem erneuten Einsatz sollte die Memory Card mit STEP 7 formatiert werden.

Achtung! Ein Formatieren mit Windows-Betriebsmitteln macht die Memory Card für einen Einsatz in einer SIMATIC-CPU unbrauchbar!



Bild 1.6
SIMATIC Memory Card

Eine Memory Card formatieren Sie in der CPU-Baugruppe. Schalten Sie die CPU-Baugruppe ohne Memory Card ein und stellen Sie eine Online-Verbindung her. Bei einer CPU 1200 schalten Sie mit den Online-Tools den Betriebszustand STOP ein. Stecken Sie die Memory Card in die CPU-Baugruppe und doppelklicken Sie in der Projektnavigation auf *Online & Diagnose*. Im Diagnosefenster wählen Sie *Funktionen > Memory Card formatieren* und dann die Schaltfläche *Formatieren*. Bei einer CPU 1500 können Sie die Memory Card auch über das Display formatieren.

Zur Einstellung des Kartentyps stecken Sie die Memory Card in den Kartenleser des Programmiergeräts. In der Projektnavigation öffnen Sie den Ordner *Cardreader/USB-Speicher* und die unterlagerten Ordner bis zur SD-Karte (bis zum Laufwerksbuchstaben). Markieren Sie die SD-Karte und klicken Sie im Kontextmenü auf den Befehl *Eigenschaften*. Im aufgeblendeten Dialogfenster wählen Sie im Feld *Kartentyp* aus der Klappliste einen der Einträge *Programm*, *Übertragung* oder *Firmware aktualisieren*.

Zum Übertragen der Projektdaten vom Programmiergerät auf die Memory Card markieren Sie in der Projektnavigation die PLC-Station und ziehen diese mit der Maus auf die SD-Karte. Bestätigen Sie die angezeigten Dialoge.

Die Memory Card darf nur bei ausgeschalteter Versorgungsspannung oder im Betriebszustand STOP gezogen und gesteckt werden. Achten Sie darauf, dass der Schalter für den Schreibschutz an der Memory Card ausgeschaltet ist.

S7-1200: Wenn Sie die CPU mit einer Übertragungskarte mit Projektdaten starten, werden die Projektdaten in den internen Ladespeicher übertragen. Die Memory Card kann dann wieder gezogen werden. Wenn die Übertragungskarte leer ist, wird der interne Ladespeicher gelöscht. Wenn Sie die CPU mit einer Programmkarte mit Projektdaten starten, nimmt die CPU die Memory Card als externen Ladespeicher. Wenn beim Starten die Programmkarte leer ist, wird der Inhalt des internen Ladespeichers auf die Memory Card kopiert. In beiden Fällen muss die Memory Card als externer Ladespeicher im Betrieb gesteckt bleiben.

1.5 Signalbaugruppen

Digitalbaugruppen

Digitalbaugruppen sind Signalformer für binäre Prozess-Signale. Über Digital-Eingabebaugruppen kann das Anwenderprogramm Betriebszustände des Prozesses und Bedienvorgänge abfragen und über Digital-Ausgabebaugruppen steuernd in den Prozess eingreifen. Ein Digitalkanal belegt ein Bit. Auf den Baugruppen sind die Digitalkanäle zu Bytes (= 8 Bits) zusammengefasst. Es gibt Baugruppen mit 8, 16 oder 32 Kanälen entsprechend einem, zwei oder vier Bytes (Bild 1.7).

Analogbaugruppen

Analogbaugruppen sind Signalformer für analoge Prozess-Signale. Analog-Eingabebaugruppen wandeln analoge Messwerte in digitale Werte um, die vom Anwenderprogramm verarbeitet werden können. Analog-Ausgabebaugruppen erzeugen analoge Sollwerte aus den vom Anwenderprogramm vorgegebenen Digitalwerten. Ein Analogkanal belegt 16 Bits. Es gibt Analogbaugruppen mit 2, 4 oder 8 Kanälen, entsprechend 4, 8 oder 16 Bytes.

Ein physikalischer Messwert, z. B. ± 10 V Spannung, wird in einem Digitalwert im Zahlformat INT (Integer, Ganzzahl) abgebildet: Die Bits 0 bis 14 enthalten den Wert und das Bit 15 enthält das Vorzeichen des Analogwerts. Bei einem „bipolaren“ Messwert kann das Vorzeichen wechseln, wie z. B. bei einem Messwert im Messbereich von ± 10 V. Ein „unipolarer“ Messwert, wie z. B. ein Messwert im Bereich von 4...20 mA, hat ein gleichbleibendes (positives) Vorzeichen.

Der Nennbereich eines Messwerts ($\pm 100\%$ bzw. 0...100%) wird bei bipolaren Messwerten im Zahlenbereich von $-27\,648$ bis $+27\,648$ und bei unipolaren Messwerten im Zahlenbereich von 0 bis $+27\,648$ abgebildet.

Beispiel: Wenn bei einem Messbereich von ± 10 V ein Messwert von $+10$ V anliegt, liefert die Analogbaugruppe den Wert $+27\,648$. Ein Messwert, der größer als 100% ist, liegt im „Übersteuerungsbereich“; ein Messwert, der kleiner als 0% bzw. -100% ist, liegt im „Untersteuerungsbereich“. Der höchste darstellbare Zahlenwert $+32\,767$ bedeutet „Überlauf“ und der kleinste darstellbare Wert $-32\,768$ bedeutet „Unterlauf“.



Bild 1.7 Signalbaugruppen in den Bauformen S7-1200 und S7-1500

Wertstatus (S7-1500)

Entsprechend ausgelegte Baugruppen liefern mit dem Wertstatus (Quality Information, QI) eine Aussage über die Gültigkeit des Peripheriesignals. Bei aktiviertem Wertstatus steht die Information im Eingangs-Prozessabbild zur Verfügung. Pro Digital- oder Analogkanal zeigt ein Bit im Wertstatus mit Signalzustand „0“ an, dass der zugeordnete Prozesswert ungültig ist. Bei Eingabebaugruppen liegt der Wertstatus im Anschluss an die E/A-Adressen im Prozessabbild, bei Ausgabebaugruppen werden für den Wertstatus zusätzlich Eingangsbytes belegt. Der Wertstatus wird zusammen mit den Peripherie-Ein-/Ausgängen übertragen. Die Aktivierung des Wertstatus geschieht mit der Hardware-Konfiguration.

1.6 Eingänge, Ausgänge, Merker

Signalweg

Mit der Verdrahtung der Maschine oder Anlage legen Sie fest, welche Signale wo an die PLC-Station angeschlossen werden (Bild 1.8). Ein Eingangssignal, z. B. das Signal vom Taster +HP01-S10 mit der Bedeutung „Motor einschalten“, wird auf eine Eingabebaugruppe geführt, wo es an einer bestimmten Klemme angeschlossen wird. Auf welchem Steckplatz die Baugruppe steckt, projizieren Sie mit der Hard-

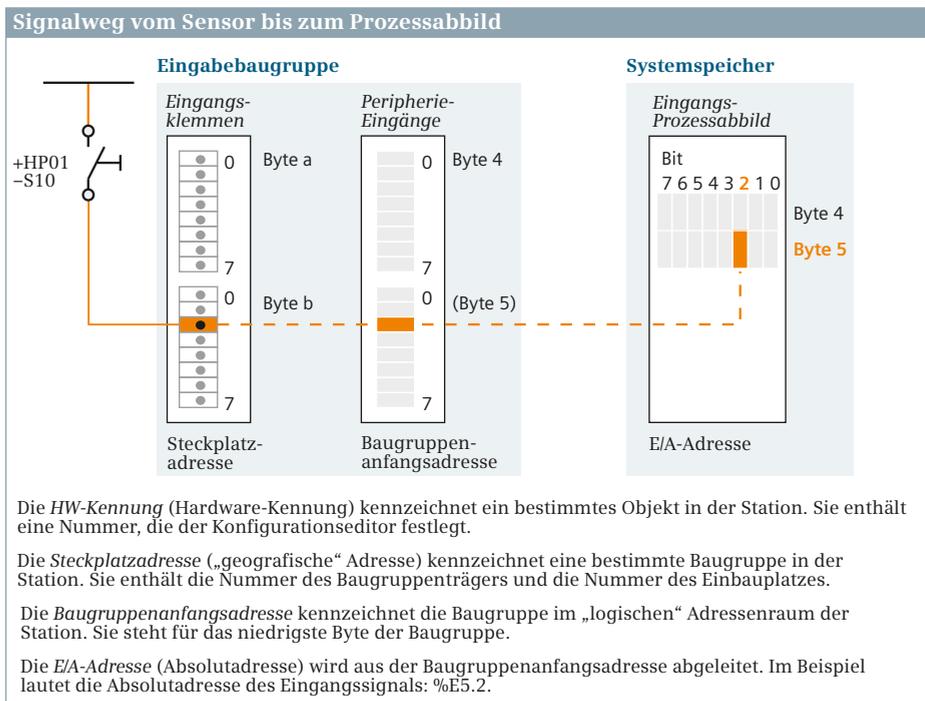


Bild 1.8 Weg eines Signals vom Sensor bis zum Prozessabbild

ware-Konfiguration. Der Konfigurationseditor vergibt für die Eingabebaugruppe eine eindeutige ID, die HW-Kennung. Mit der Hardware-Konfiguration stellen Sie die Baugruppenanfangsadresse ein, aus der die Adressen der Baugruppenkanäle abgeleitet werden. Die CPU kopiert standardmäßig jedes Mal vor Beginn der zyklischen Programmbearbeitung das Signal von der Eingabebaugruppe in das Eingangs-Prozessabbild. Die Adresse auf der Baugruppe ist gleichzeitig die Adresse im Prozessabbild.

Für die Ausgangssignale gilt sinngemäß das Gleiche. Sie projektieren bei der Hardware-Konfiguration den Steckplatz der Ausgabebaugruppe und legen die Baugruppenanfangsadresse fest. Das ist dann auch die Adresse im Ausgangs-Prozessabbild.

Symbolische Operandenadresse

Das Signal vom Taster +HP01-S10 hat in Bild 1.8 die E/A-Adresse %E5.2. Das ist die Absolutadresse. Im Anwenderprogramm wird dieses Signal mit der Symboladresse verarbeitet, die beispielsweise „Motor starten“ lautet. Die Symboladresse legen Sie fest und ordnen sie in der PLC-Variablentabelle der absoluten Adresse zu. Wenn Sie keine Symboladresse festlegen, erzeugt der Programmierer automatisch eine Symboladresse nach dem Muster „Tag_n“ mit n als laufender Nummer.

Operandenbereiche in einer SIMATIC-PLC

Beim Steuern einer Maschine oder eines Prozesses werden Signalzustände und Zahlenwerte verarbeitet. Eingänge werden abgefragt, deren Signalzustände gemäß der Steuerungsaufgabe miteinander verknüpft und mit den Ergebnissen die Ausgänge gesteuert. Ähnlich ist es mit Zahlenwerten, die ausgewählt, berechnet, verglichen und gespeichert werden. Die PLC-Station stellt für diese variablen Werte die in Bild 1.9 gezeigten Speicherbereiche zur Verfügung.

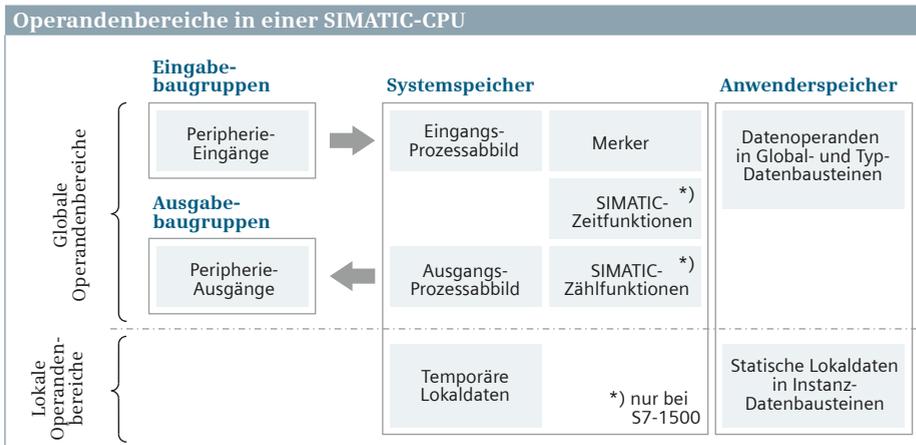


Bild 1.9 Operandenbereiche in einer SIMATIC-CPU

Operandenbereich Peripherie-Eingänge

Die Peripherie-Eingänge sind der Speicherbereich auf den Eingabebaugruppen. Sie sind die direkte Schnittstelle zur Maschine oder zum Prozess, um beispielsweise Sensoren oder die Einstellungen von Bedienelementen abzufragen, die über die Klemmen der Eingabebaugruppen geliefert werden. Peripherie-Eingänge dürfen nur gelesen werden, der Programmierer warnt bei einem schreibenden Zugriff. Die Peripherie-Eingänge werden vom CPU-Betriebssystem vor jedem Bearbeitungszyklus des Anwenderprogramms automatisch in das Eingangs-Prozessabbild kopiert.

Operandenbereich Eingänge

Das Eingangs-Prozessabbild enthält den Operandenbereich Eingänge. Die Eingänge werden verwendet, um im Anwenderprogramm Binärsignale abzufragen und deren Signalzustände zu verknüpfen. Die Eingänge können gelesen und geschrieben werden. Im Normalfall arbeitet das Anwenderprogramm mit den Eingängen und nicht mit den Peripherie-Eingängen. Das Eingangs-Prozessabbild ist groß genug, um sämtliche Peripherie-Eingänge aufzunehmen.

Operandenbereich Peripherie-Ausgänge

Die Peripherie-Ausgänge sind der Speicherbereich auf den Ausgabebaugruppen. Sie enthalten die Signalzustände und Zahlenwerte, mit denen die Maschine oder der Prozess über die Verdrahtung gesteuert wird. Peripherie-Ausgänge dürfen nur geschrieben werden, der Programmierer warnt vor einem lesenden Zugriff. Das Schreiben zu den Peripherie-Ausgängen führt automatisch das Ausgangs-Prozessabbild nach, sodass es keine Differenz der Signalzustände von Ausgängen und Peripherie-Ausgängen mit gleichen Adressen gibt. Das CPU-Betriebssystem überträgt vor jedem Bearbeitungszyklus des Anwenderprogramms automatisch die Signalzustände des Ausgangs-Prozessabbilds zu den Peripherie-Ausgängen.

Operandenbereich Ausgänge

Das Ausgangs-Prozessabbild enthält den Operandenbereich Ausgänge. Die Ausgänge werden verwendet, um im Anwenderprogramm die Maschine oder den Prozess gemäß der Steuerungsaufgabe zu beeinflussen. Die Ausgänge können gelesen und geschrieben werden. Im Normalfall arbeitet das Anwenderprogramm mit den Ausgängen und nicht mit den Peripherie-Ausgängen. Das Ausgangs-Prozessabbild ist groß genug für alle Peripherie-Ausgänge.

Operandenbereich Merker, System- und Taktmerker

Ein dritter, ähnlicher Operandenbereich wie die Eingänge oder Ausgänge sind die Merker. Sie liegen wie das Prozessabbild im Systemspeicher der CPU, haben aber keine Verbindung zu den Baugruppen. Man kann sie als globale Variablen verwenden, um beispielsweise Betriebsartenbits oder Fehlernummern zu speichern.

Die CPU stellt zwei Merkerbytes mit besonderer Funktion zur Verfügung. Die Systemmerker zeigen den ersten Durchlauf des Anwenderprogramms im Betriebszu-

stand RUN (kann zur Voreinstellung verwendet werden) und die Änderung des Diagnosestatus an (kann zur Fehleranzeige verwendet werden) sowie zwei Merkerbits, die immer „1“ (TRUE) oder „0“ (FALSE) sind (können als Ersatz für BOOL-Konstanten verwendet werden). Im Taktmerkerbyte wechseln die Merkerbits mit verschiedenen Perioden den Signalzustand (kann zur Steuerung eines Blinklichts verwendet werden). Die Aktivierung dieser speziellen Merkerbytes geschieht bei der Hardware-Konfiguration in den CPU-Eigenschaften. Nach der Aktivierung werden diese Merker in die Standard-Variablen-tabelle übernommen. Die vorgegebenen Namen können Sie ändern. Diese System- und Taktmerker dürfen vom Anwenderprogramm nur abgefragt werden.

Remanenzeinstellungen für Merker

Einen Teil des Merkerbereichs können Sie „remanent“ einstellen, d. h. sein Inhalt bleibt auch nach Abschalten der Versorgungsspannung und nach Einschalten beim Übergang von STOP nach RUN erhalten. Die Remanenz können Sie in der Variablen-tabelle einstellen. Klicken Sie auf das Symbol für Remanenz in der Funktionsleiste des Arbeitsfensters und geben Sie die Anzahl der remanenten Bytes ein. Der Remanenzbereich beginnt immer mit dem Merkerbyte 0 und endet an der eingestellten Obergrenze. Liegt eine Merkerdeklaration innerhalb dieses Bereichs, erhält sie in der Spalte *Remanenz* einen Haken. Eine Variable, die mehr als ein Byte belegt, darf die Grenze zwischen remanentem und nicht remanentem Bereich nicht überschreiten.

Weitere Operandenbereiche

Der Operandenbereich Daten wird zum Speichern der Anwenderdaten verwendet. Die Datenvariablen sind in Datenbausteinen zusammengefasst, die im Anwenderspeicher liegen. Sie können remanent eingestellt werden, sodass deren Werte bei Spannungswiederkehr weiterhin zur Verfügung stehen.

- ▷ Datenvariablen, die man global (im gesamten Anwenderprogramm) adressieren kann, liegen in Global- und Typ-Datenbausteinen (Kapitel 6.2 „Datenvariablen in Global-Datenbausteinen“ auf Seite 134).
- ▷ Datenvariablen, die lokal (in einem Funktionsbaustein) Gültigkeit besitzen, liegen in Instanz-Datenbausteinen (Kapitel 8.2 „Instanzdaten anlegen“ auf Seite 199). Das sind dann die statischen Lokaldaten.

Die temporären Lokaldaten werden für Zwischenergebnisse genutzt, die außerhalb des Bausteinprogramms nicht benötigt werden. Sie liegen im Systemspeicher und stehen in jedem Codebaustein zur Verfügung (siehe Abschnitt „Temporäre Lokaldaten“ auf Seite 167).

S7-1500: Die SIMATIC-Zeitfunktionen und SIMATIC-Zählfunktionen sind für die Migration von S7-300/400-Programmen noch vorhanden. Für Neuanwendungen werden stattdessen die IEC-Zeitfunktionen und IEC-Zählfunktionen empfohlen.

1.7 Übersicht Adressierung

Die absolute Adressierung wird für Ein- und Ausgänge verwendet, um die Klemmen an den Baugruppen zu spezifizieren, mit denen sie verbunden sind. In der PLC-Variablen-tabelle können Sie den Ein- und Ausgängen symbolische Adressen zuordnen, deren Verwendung im Anwenderprogramm empfohlen wird. Aus historischen Gründen können Sie die Merker wie die Ein- und Ausgänge behandeln, obwohl sie mit keiner Baugruppe verbunden sind.

Alle anderen Variablen (Globaldaten, statische und temporäre Lokaldaten) werden symbolisch adressiert. Das aktivierte Bausteinattribut *Optimierter Bausteinzugriff* sorgt für eine optimale Behandlung beim Speichern und Zugriff auf die Variablen. Aus Gründen der Migration von S7-300/400-Anwenderprogrammen können bei nicht aktiviertem Attribut *Optimierter Bausteinzugriff* diese Variablen auch absolut adressiert werden. Die Verwendung dieser Möglichkeit bei Neuanwendungen wird nicht empfohlen.

Absolutadresse

Die absolute Adressierung greift auf Operanden mithilfe der Speicheradresse zu. In Bild 1.8 auf Seite 21 ist es die Adresse %E5.2. Das Prozentzeichen ist die Kennzeichnung für eine Absolutadresse, danach folgt der Kennbuchstabe für den Operandenbereich. Die Angabe vor dem Punkt ist die Adresse des Bytes und die Angabe nach dem Punkt die Adresse des Bits in diesem Byte. Die Bezeichnung %E5.2 adressiert also den Eingang am Bit 2 im Byte 5.

Die Adressen der Peripherie-Ein- und -Ausgänge (die Ein- und Ausgabekanäle auf den Baugruppen) werden bei der Projektierung des Stationsaufbaus bei der Hardware-Konfiguration festgelegt. Die zugeordneten Ein- und Ausgänge im Prozessabbild haben die gleichen Adressen. Zur Kennzeichnung einer Peripherie-Adresse wird an die Eingangs- bzw. Ausgangsadresse die Peripheriezugriffskennung „:P“ angehängt (Tabelle 1.1).

Tabelle 1.1 Operandenkennzeichen und absolute Adressierung für E/A/M

Operandenbereich	Operandenkennzeichen	Bit (1 Bit)	Byte (8 Bits)	Wort (16 Bits)	Doppelwort (32 Bits)
Eingang	E	%Ey.x	%EBy	%EWy	%EDy
Peripherie-Eingang	:P an den Eingangsoperanden „angehängt“	%Ey.x:P	%EBy:P	%EWy:P	%EDy:P
Ausgang	A	%Ay.x	%ABy	%AWy	%ADy
Peripherie-Ausgang	:P an den Ausgangsoperanden „angehängt“	%Ay.x:P	%ABy:P	%AWy:P	%ADy:P
Merker	M	%My.x	%MBy	%MWy	%MDy

y = Byteadresse; x = Bitadresse

S7-1500: Ein (64 Bits breites) Langwort kann nicht absolut adressiert werden. Der Programmierer zeigt bei einem symbolisch adressierten Langwort die Absolute Adresse in Form eines Zeigers an, Beispiel: P#M16.0, wenn das Merker-Langwort am Byte 16 beginnt (und mit Byte 23 endet).

Bit- und Bytebelegung

Die Bits in einem Byte werden von rechts nach links gezählt, beginnend mit null. Für jedes Byte beginnt die Zählung von neuem. Jeder Operandenbereich ist byteweise organisiert. Die Zählung der Bytes beginnt am Bereichsanfang bei null. Bei einem bytebreiten Operanden wird als Byteadresse die Nummer des Bytes angegeben, bei einem wortbreiten die Nummer des niedrigeren Bytes und bei einem doppelwortbreiten die niedrigste Byteadresse in dem Doppelwort. In Bild 1.10 wird dieses Prinzip am Beispiel der Merkerbytes MB 24 bis MB 27 erläutert. Die Absolute Adressierung einer 64 Bits breiten Variablen (eines Langworts) ist nicht möglich. Derartige Variablen werden nur symbolisch adressiert.

Die absolute Adressierung von Datenvariablen ist im Abschnitt „Globaldaten absolut adressieren“ auf Seite 138 beschrieben.

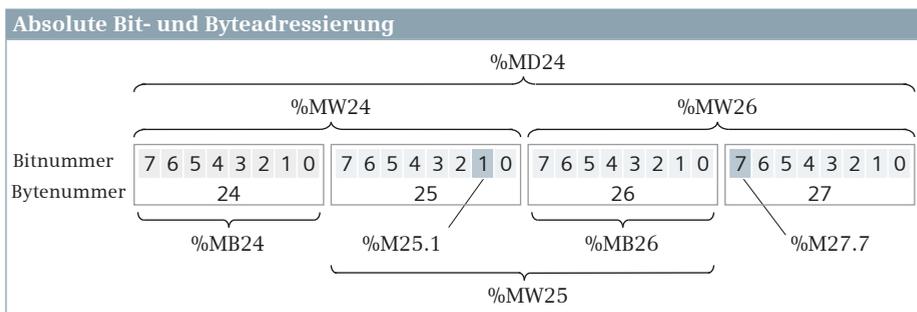


Bild 1.10 Bit- und Byte-Belegung

Symboladresse

Bei der symbolischen Adressierung wird ein Operand mit einer alphanumerischen Bezeichnung (Name, Symbol) und einem Datentyp versehen. Man spricht dann von einer *Variablen*. Beispielsweise könnte der Operand %E5.2 den Namen „Motor starten“ und den Datentyp BOOL erhalten. Die Zuordnung der Symboladresse zur Absolute Adresse eines Operanden nehmen Sie in einer PLC-Variablen-tabelle vor. Variablen-namen können Buchstaben, Ziffern und Sonderzeichen – außer doppelten Anführungszeichen – enthalten. Groß- und Kleinbuchstaben werden bei der Namensprüfung nicht unterschieden. Im Anwenderprogramm können beim Programmieren sowohl die Absolute Adresse als auch die Symboladresse verwendet werden. Die Anzeige ist wählbar: nur absolut (%E5.2), nur symbolisch („Motor starten“) oder beides.

Die Adressen der Peripherie-Ein- und -Ausgänge (die Ein- und Ausgabekanäle auf den Baugruppen) sind den Ein- und Ausgängen in den Prozessabbildern direkt 1:1 zugeordnet und übernehmen auch deren Symbolnamen. Zur Unterscheidung wird an den Symbolnamen die Peripheriezugriffskennung „:P“ angehängt. Beispiel: Das Symbol „Motor starten“ bezeichnet einen Eingang im Prozessabbild, das Symbol „Motor starten“:P bezeichnet den zugeordneten Peripherie-Eingang auf der Baugruppe.

Der Programmierer zeigt die Symboladresse von globalen Variablen in Anführungszeichen an. Er vergibt zusätzlich zur Absolutadresse immer eine Symboladresse. Wenn Sie beispielsweise im Programm einen Eingang E 4.0 erstmalig verwenden, legt der Programmierer eine Symboladresse „Tag_n“ mit n als laufender Nummer an. Diese Symboladresse können Sie ändern. Es wird empfohlen, nur die symbolische Adressierung mit aussagekräftigen Namen zu verwenden.

Weitere Adressierungsmöglichkeiten (Übersicht)

Variablen mit PLC-Datentypen können hierarchisch aufgebaut sein, d. h. aus Komponenten bestehen, die unterschiedliche Datentypen haben können. Es kann die komplette Variable oder eine einzelne Komponente adressiert werden (Kapitel 9.1 „PLC-Datentypen“ auf Seite 220).

Variablen mit dem Datentyp ARRAY bestehen aus Komponenten, die alle den gleichen Datentyp haben. Es kann die komplette Variable oder eine einzelne Komponente adressiert werden. Eine Komponente wird mit einem Index ausgewählt, der eine Konstante oder eine Variable sein kann (Kapitel 6.4 „Datentyp ARRAY“ auf Seite 144).

Bei Variablen mit dem Datentyp STRING oder WSTRING können einzelne Zeichen mit einem Index adressiert werden (Kapitel 14 „Zeichen und Zeichenketten“ auf Seite 333).

Mit der Slice-Adressierung ist es möglich, einzelne Teile einer symbolisch adressierten Variablen, z. B. ein Bit, anzusprechen (Kapitel 11.4 „Slice-Adressierung“ auf Seite 264).

1.8 Übersicht Datentypen

Die Übersicht in der Tabelle 1.2 ist so gegliedert, wie die Datentypen mit den entsprechenden Funktionen in diesem Buch beschrieben sind. Die Datentypen BCD16 und BCD32 sind keine „echten“ Datentypen; sie werden nur in Verbindung mit bestimmten Konvertierungsfunktionen verwendet. Der Datentyp S5TIME wird nur in Verbindung mit einer SIMATIC-Zeitfunktion verwendet. Der Datentyp STRUCT sollte nicht mehr angewendet werden. Ein PLC-Datentyp – quasi eine Variablenstruktur mit einem Namen – ist ein vollwertiger Ersatz und wesentlich besser geeignet.

Parametertypen sind Datentypen für Bausteinparameter. Der Datentyp INSTANCE dient nur der Beschreibung. Dahinter verbergen sich die über einen Bausteinpara-

meter übergebenen Instanzdaten für Funktionsbausteine sowie für Zeit- und Zählfunktionen. Der Datentyp REFERENCE steht allgemein für eine Referenz auf eine Variable. Zu den in der Tabelle aufgeführten Datentypen kommen noch die fest vorgegebenen Systemdatentypen und Hardware-Datentypen hinzu.

Tabelle 1.2 Übersicht Datentypen

Kategorie		Datentypen S7-1200	Datentypen S7-1500	Seite
Elementare Datentypen	Binär	BOOL	BOOL	245
	Bitfolge	BYTE, WORD, DWORD, BCD16, BCD32 *)	BYTE, WORD, DWORD, LWORD, BCD16, BCD32 *)	261
	Ganzzahlen	USINT, UINT, UDINT, SINT, INT, DINT	USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT	271
	Gleitpunktzahlen	REAL, LREAL	REAL, LREAL	272
	Zeichen	CHAR, WCHAR	CHAR, WCHAR	333
	Zeitdauern	TIME	TIME, LTIME, S5TIME	309
	Zeitpunkte	DATE, TOD	DATE, TOD, LTOD, LDT	310
Strukturierte Datentypen	Zeitpunkte	DTL	DT, DTL	310
	Zeichenketten	STRING, WSTRING	STRING, WSTRING	334
	Feld	ARRAY	ARRAY	144
	Struktur	STRUCT, PLC-Datentyp **)	STRUCT, PLC-Datentyp **)	220
Parametertypen	Funktionswert	VOID	VOID	167
	Datenbaustein	DB_ANY	DB_ANY	227
	Variables Feld	ARRAY[*]	ARRAY[*]	230
	Zeiger	VARIANT	VARIANT	232
	Instanz	INSTANCE	INSTANCE	199
	Referenz	–	REFERENCE	240

*) BCD16 und BCD32 sind keine „echten“ Datentypen

**) Die Anwendung von STRUCT wird nicht empfohlen, Ersatz: PLC-Datentypen

Wenn Variablen miteinander verknüpft werden, müssen diese Variablen einen kompatiblen Datentyp aufweisen. Die implizite Datentypwandlung geschieht quasi automatisch beim Beschalten der Bausteinparameter und hängt vom Bausteinattribut *IEC-Prüfung* ab. Für die explizite Datentypwandlung gibt es entsprechende Konvertierungsfunktionen. Das Kapitel 7.5 „Datentypkonvertierung“ auf Seite 173 zeigt die Datentypkonvertierungen in der Übersicht.

2 Anwenderprogramm

2.1 Programmbearbeitung

Schnittstelle zum Betriebssystem

Das Gesamtprogramm einer CPU-Baugruppe besteht aus dem Betriebssystem und dem Anwenderprogramm (Steuerungsprogramm). Das Betriebssystem ist die Gesamtheit aller Anweisungen und Vereinbarungen geräteinterner Betriebsfunktionen (z. B. Sicherstellung von Daten bei Ausfall der Versorgungsspannung, Aktivieren von Ereignisklassen usw.). Das Betriebssystem ist fester Bestandteil der CPU und kann von Ihnen nicht verändert werden. Das Betriebssystem können Sie jedoch z. B. bei einem Programm-Update neu laden. Das Anwenderprogramm ist die Gesamtheit aller von Ihnen programmierten Anweisungen und Vereinbarungen für die Signalverarbeitung, durch die eine zu steuernde Anlage (der Prozess) gemäß der Steuerungsaufgabe beeinflusst wird.

Die Schnittstellen zwischen Betriebssystem und Anwenderprogramm sind die Organisationsbausteine. Die Organisationsbausteine sind Teil des Anwenderprogramms und werden vom Betriebssystem bei bestimmten Ereignissen aufgerufen und bearbeitet. Ereignisse können beispielsweise der einmalige Start des Anlaufs, der zyklisch wiederkehrende Start der Hauptprogrammbearbeitung, ein Prozessalarm oder ein Programmierfehler sein. Ist dem Ereignis ein Organisationsbaustein zugeordnet, ruft das Betriebssystem beim Auftreten des Ereignisses diesen Organisationsbaustein auf, in dem weitere Codebausteine aufgerufen werden können.

Ist einem Ereignis kein Organisationsbaustein zugeordnet oder ist der zugeordnete Organisationsbaustein nicht vorhanden, führt das Betriebssystem beim Auftreten des Ereignisses die voreingestellte Systemreaktion aus: Das Betriebssystem ignoriert das Ereignis oder wechselt in den Betriebszustand STOP. Die Organisationsbausteine sind in Ereignisklassen eingeteilt, die beim Auftreten mehrerer Ereignisse die Reihenfolge der Programmbearbeitung (die gegenseitige Unterbrechbarkeit) festschreiben.

Übersicht Organisationsbausteine

Tabelle 2.1 zeigt die vorhandenen Organisationsbausteine mit ihrer voreingestellten Bearbeitungspriorität. Wenn Sie einen Organisationsbaustein neu anlegen, müssen Sie eine Ereignisklasse angeben und legen damit die Zuordnung zum Ereignis fest. Unter Umständen müssen Sie in den Bausteineigenschaften das Ereignis definieren, z. B. die Uhrzeit für einen Uhrzeitalarm oder das Intervall für einen Weckalarm.