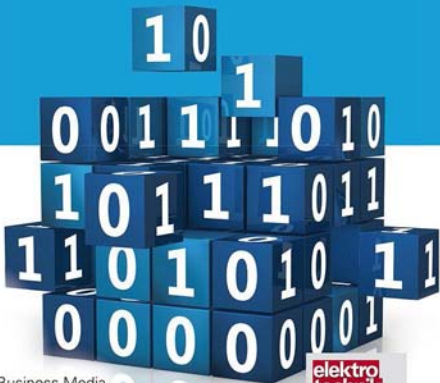


Jürgen Kaftan

SPS-Grundkurs mit SIMATIC S7



Vogel Business Media

elektro
technik

Der Onlineservice InfoClick bietet unter www.vogel-buchverlag.de nach Codeeingabe eventuell zusätzliche Informationen und Aktualisierungen. Fordern Sie für Ihr E-Book den Code unter buch@vogel-buchverlag.de an!

Jürgen Kaftan

SPS-Grundkurs mit SIMATIC S7

**Aufbau und Funktion
speicherprogrammierbarer Steuerungen,
Programmieren mit SIMATIC S7**

6. Auflage

Vogel Business Media

JÜRGEN KAFTAN

1967–1971 Ausbildung zum Elektromechaniker
1971–1973 Facharbeiter im Elektromechaniker-Handwerk
1973–1975 Studium zum staatl. gepr. Elektrotechniker
1975–1977 Tätigkeit als Techniker
1977–1978 Meisterschule, Elektromechanikermeister
1979–1992 Berufsbildungswerk Nürnberg für Hör- und
Sprachgeschädigte, Ausbildungsmeister
1985–1992 Kursleiter für SPS-Steuerungen in der Beruflichen
Fortbildung (BFH) Hör- und Sprachgeschädigter
1985–1992 Kursleiter für SPS an der Handwerkskammer
1992–1995 IKH Elektrogerätebau – Systemschulungen (Geschäftsführer)
seit 1995 Leiter der Bildungseinrichtung IHK Systemschulungen
für Hard- und Software in Weißenburg und Roth
(Mittelfranken)

Jürgen Kaftan ist Autor folgender Vogel-Fachbücher:

SPS-Grundkurs 1
SPS-Grundkurs 2
SPS-Grundkurs mit SIMATIC S7
SPS-Aufbaukurs mit SIMATIC S7
SPS-Beispiele mit SIMATIC S7
SPS-Beispiele mit SIMATIC S7-1200
LOGO!-Kurs

Weitere Informationen: www.vogel-buchverlag.de



<http://twitter.com/vogelbuchverlag>



www.facebook.com/vogel.buchverlag



www.vogel-buchverlag.de/rss/buch.rss

ISBN 978-3-8343-3368-1

6. Auflage. 2015

Alle Rechte, auch der Übersetzung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlages reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Hiervon sind die in §§ 53, 54 UrhG ausdrücklich genannten Ausnahmefälle nicht berührt.

Printed in Germany

Copyright 1998 by Vogel Business Media GmbH & Co. KG,
Würzburg

Umschlaggrafik: Fotolia Titelgrafik: © vege-Fotolia.com

Vorwort

Speicherprogrammierbare Steuerungen, kurz SPS, sind für Automatisierungsaufgaben im unteren, mittleren und oberen Leistungsbereich nicht mehr wegzudenken. Durch immer größere und umfangreichere Automatisierungsprojekte bei Hard- und Software stellte sich für die Industrie die Aufgabe, schnellere und leistungsfähigere Automatisierungssysteme zu entwickeln sowie das Handling für den Anwender zu vereinfachen. Alle Übungen in diesem Buch wurden mit der SPS-Steuerung SIMATIC S7-300 der Firma Siemens erarbeitet und getestet. Der Befehlssatz dieser Steuerung reicht von der Binärverarbeitung bis zur 32-Bit-Gleitpunktarithmetik. Die Programmierung der Steuerung erfolgt unter dem Betriebssystem Windows XP und setzt die entsprechenden Grundkenntnisse voraus. Alle Vorgänge zur Programmierung der SIMATIC S7-300 werden exakt aufgezeigt und sind vom Anwender leicht nachvollziehbar. Das Thema ist vom «Einfachen zum Schwierigen» aufgebaut und für den Einsatz in Berufsschulen, Technikerschulen usw., aber auch für das Selbststudium sehr gut geeignet.

Ich bedanke mich bei der Firma Wükro-Lehrsysteme in Würzburg sowie bei allen, die das Zustandekommen dieses Buches gefördert haben. Resonanz aus dem Leser- und Benutzerkreis ist mir stets willkommen.

Weißenburg/Heuberg (Mittelfranken)

Jürgen Kaftan

Inhaltsverzeichnis

Vorwort.....	5
1 Einleitung	13
1.1 Zahlensystem	13
1.2 Begriffe aus der Informatik	14
1.2.1 Bit	14
1.2.2 Byte	14
1.2.3 Wort	14
1.2.4 Bitadresse	15
1.2.5 Byteadresse	15
1.2.6 Wortadresse	15
2 Aufbau einer SPS.....	17
2.1 Zusammensetzung einer SPS	18
2.2 Zusammensetzung eines Automatisierungsgerätes.....	18
2.3 Hardware-Voraussetzungen.....	18
2.3.1 Hardware-Aufbau	20
2.4 Software-Voraussetzungen.....	20
2.4.1 Programmiersprache STEP 7	20
2.4.2 Objekte	20
2.4.3 Projekte	21
2.4.4 Konfigurieren einer S7-300	22
2.4.5 Parametrisieren	22
3 Funktionsweise einer SPS.....	23
3.1 Baugruppen.....	23
3.1.1 Stromversorgungsbaugruppe	23
3.1.2 Programmspeicher: RAM, ROM, EPROM, REPROM, EEPROM, EAPROM.....	24
3.1.3 Zentralbaugruppe (CPU)	25

3.1.4	Bussystem	26
3.1.5	Ein- und Ausgabebaugruppen	27
4	Programmbearbeitung und Programmierung	29
4.1	Lineare Programmierung	29
4.2	Strukturierte Programmierung: Organisationsbaustein (OB), Funktionsbaustein (FB), Funktion (FC), Datenbaustein (DM), Systembausteine: Systemfunktionsbaustein, Systemfunktion, Systembaustein	30
4.3	Steuerungsanweisung	31
4.3.1	Operationsteil	32
4.3.2	Beispiele für digitale Operationen	32
4.3.3	Beispiele für binäre Operationen	32
4.3.4	Beispiele für organisatorische Operationen	33
4.3.5	Operandenteil	33
4.4	Adressierung	34
4.4.1	Symbolische Adressierung	34
4.4.2	Absolute Adressierung	34
4.4.3	Unmittelbare Adressierung	34
4.4.3.1	Direkte Adressierung	34
4.4.3.2	Speicherindirekte Adressierung	34
4.5	Programmdarstellung	35
4.5.1	Kontaktplan (KOP)	36
4.5.2	Funktionsplan (FUP)	36
4.5.3	Anweisungsliste (AWL)	36
4.6	Merker	37
4.6.1	Remanente Merker	37
4.6.2	Nichtremanente Merker	38
5	Verknüpfungen	39
5.1	Grundverknüpfungen	39
5.1.1	CPU urlöschen	40
5.1.2	Projekte anlegen	42
5.1.3	Einfügung der SIMATIC-300-Station	43
5.1.4	Konfigurieren und Parametrisieren	43
5.1.5	Anordnung der Stromversorgungsbaugruppe	45
5.1.6	Anordnung der CPU 314	46
5.1.7	Anordnung der Eingabebaugruppe	46
5.1.8	Anordnung der Ausgabebaugruppe	47
5.1.9	Parametrisieren der CPU 314	47
5.1.10	Speichern der Gesamtkonfiguration	49
5.1.11	Übertragung der Konfiguration in die CPU	49
5.2	Anwendungsprogramm UND-Verknüpfung	50
5.2.1	Eingabe von FC (FC1)	50
5.2.2	S7-Bausteinfunktion	52
5.2.3	Eingabe von OB 1	55
5.2.4	Laden	58
5.2.5	Testen	59
5.2.6	Festlegung der Triggerbedingungen	60
5.2.7	FUP-Programmstatus ausschalten	63
5.2.8	Testen mit AWL	63
5.2.9	Testen mit KOP	65
5.2.10	Erweiterung von 2 auf 3 Eingänge (3. Eingang E0.2)	67
5.2.10.1	Erweiterung mit AWL	67
5.2.10.2	Erweiterung mit KOP	69

5.2.10.3	Erweiterung mit FUP	71
5.2.11	Reduzierung von 3 auf 2 Eingänge (E0.2 löschen)	73
5.2.11.1	Reduzierung mit AWL	73
5.2.11.2	Reduzierung mit KOP	74
5.2.11.3	Reduzierung mit FUP	74
5.3	Anwendungsprogramm ODER-Verknüpfung	75
5.3.1	Eingabe des Programmes mit dem PC (FUP)	75
5.3.2	Projekt anlegen	76
5.3.3	SIMATIC-Station in ein anderes Projekt kopieren	77
5.3.4	OB 1 ändern	80
5.3.5	Laden	82
5.3.6	Testen	82
6	Programmeingabe	85
6.1	UND vor ODER	85
6.2	ODER vor UND	88
6.3	Abfrage auf Signalzustand 0	91
6.4	Exklusiv-ODER-Verknüpfung	94
6.5	Abfrage von Ausgängen	96
6.6	Einfügung von Netzwerken	98
6.7	Selbsthaltung mit dem PC	101
6.8	Praktische Steuerungsbeispiele mit dem PC	105
6.8.1	Temperaturdifferenz	105
6.8.2	Getränkeautomat	106
6.8.3	Sprechanlage	108
6.8.4	Generator	110
6.8.5	Kesselsteuerung	112
6.8.6	Schmelzöfen	113
6.9	Speicherschaltung	116
6.9.1	R-S-Speicherschaltung	116
6.9.2	Eingabe des Programmes	118
6.9.3	Pumpensteuerung	122
7	Erzeugen von Wischimpulsen (Flankenoperationen)	127
7.1	Wischimpuls mit steigender Flanke	127
7.2	Wischimpuls mit fallender Flanke	128
7.3	Programmeingabe	128
7.4	Quittierschaltung	132
8	Zeitfunktionen.	135
8.1	Zeitwertvorgabe	135
8.2	Zeit freigeben	136
8.3	Zeitwert	136
8.4	Zeit rücksetzen	137
8.5	Auswahl der Zeiten (5 verschiedene Zeiten)	137
8.5.1	Zeitstufe als SI-Impuls	137
8.5.2	Zeitstufe als verlängerter SV-Impuls	139
8.5.3	Zeitstufe als SE-Einschaltverzögerung	140
8.5.4	Zeitstufe als speichernde SS-Einschaltverzögerung	142
8.5.5	Zeitstufe als SA-Ausschaltverzögerung	143
8.6	PC-Programmeingabe von Zeitfunktionen	144
8.6.1	Garagenbeleuchtung	147

8.6.2	Abfüllanlage	149
8.6.3	Kompressoranlage	150
9	Taktgeber	153
9.1	PC-Programmeingabe mit Taktgeber	154
9.1.1	Kanalumschalter	156
9.1.2	Personensuchanlage	157
9.1.3	Luftversorgung	159
10	Zähler	163
10.1	Lade- und Transferfunktionen	163
10.2	Zählerfunktionen	164
10.2.1	Freigeben	164
10.2.2	Vorwärtszählen	164
10.2.3	Rückwärtszählen	164
10.2.4	Setzen	165
10.2.5	Zählwertvorgabe	165
10.2.6	Rücksetzen	165
10.2.7	Zählwertabfrage	165
10.2.8	Signalzustandsabfrage	165
10.3	PC-Programmeingabe Reinigungsbad	167
11	Vergleicher	169
11.1	Vergleichsfunktionen	169
11.1.1	Gleich	169
11.1.2	Ungleich	170
11.1.3	Größer als gleich	170
11.1.4	Größer	170
11.1.5	Kleiner als gleich	170
11.1.6	Kleiner	171
11.2	PC-Programmeingabe Landefeuere	171
11.3	PC-Programmeingabe Ablauffunktion	174
12	Praktische Beispiele mit Simulatoren	177
12.1	7-Segment-Anzeige	177
12.2	Stern-Dreieck-Anlauf	179
12.3	Ampelsteuerung	181
12.4	Bandsteuerung	183
12.5	Reaktionsgefäß	186
12.6	Behälterfüllanlage	188
12.7	Tablettenfüllautomat	190
12.8	Schleusentürsteuerung	193
12.9	Pumpensteuerung	195
13	Ablaufsteuerungen	199
13.1	Einführung	199
13.2	Bestandteile	200
13.3	Darstellungsart	201
13.4	Lineare Ablaufkette	201
13.5	Blechbiegevorrichtung	204

14 Sicherheitsbestimmungen	207
14.1 Regeln	207
14.2 NOT-AUS-Steuerungsfreigabe.....	208
14.3 Beispiel einer Steuerungsfreigabe.....	209
 Anhang	 211
 Lösungen zu den Beispielen.....	 211
Abschnitt 6.8.1	212
Abschnitt 6.8.2	215
Abschnitt 6.8.3	218
Abschnitt 6.8.4	221
Abschnitt 6.8.5	223
Abschnitt 6.8.6	226
Abschnitt 6.9.3	232
Abschnitt 7.4	237
Abschnitt 8.6.1	240
Abschnitt 8.6.2	243
Abschnitt 8.6.3	249
Abschnitt 9.1.1	255
Abschnitt 9.1.2	257
Abschnitt 9.1.3	260
Abschnitt 10.3	263
Abschnitt 11.2	269
Abschnitt 11.3	277
Abschnitt 12.1	282
Abschnitt 12.2	297
Abschnitt 12.3	303
Abschnitt 12.4	312
Abschnitt 12.5	322
Abschnitt 12.6	328
Abschnitt 12.7	337
Abschnitt 12.8	356
Abschnitt 12.9	370
Abschnitt 13.5	386

1 Einleitung

Die speicherprogrammierbare Steuerung (SPS) hat die Aufgabe, Einzeloperationen einer Maschine oder Anlage nach einem vorgegebenen Funktionsablauf in Abhängigkeit von Gebersignalen zu steuern oder zu regeln.

1.1 Zahlensystem

Für die Bearbeitung der Adressen von Speicherzellen, Eingängen, Ausgängen, Zeiten, Merkern usw. durch eine speicherprogrammierbare Steuerung wird nicht das Dezimalsystem, sondern das **duale Zahlensystem** verwendet.

Das duale Zahlensystem kennt nur die Ziffern 0 und 1, die in der Datenverarbeitungstechnik leicht darstellbar und auswertbar sind. **Es ist ein binäres Zahlensystem.**

Die Stellenwerte einer Dualzahl sind, wie unten dargestellt, den Potenzen von 2 zugeordnet.

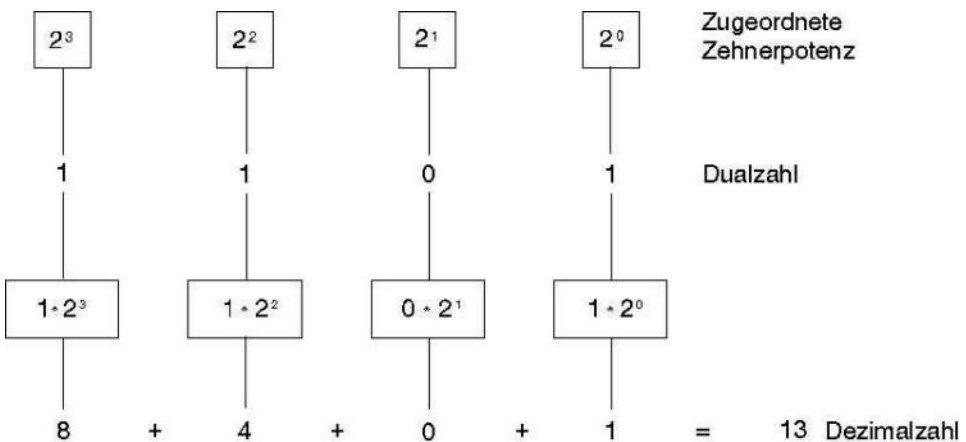


Bild 1.1 Jeder Ziffer innerhalb einer Dualzahl ist eine Zweierpotenz zugeordnet

1.2 Begriffe aus der Informatik

Im Zusammenhang mit speicherprogrammierbaren Steuerungen werden häufig aus der Daten- bzw. Informationsverarbeitung stammende Begriffe wie Bit, Byte, Wort und Doppelwort verwendet.

1.2.1 Bit

Das Bit (das Wort ist eine Abkürzung für Binärziffer, engl.: binary digit) ist die kleinste binäre (zweiwertige) Informationseinheit.

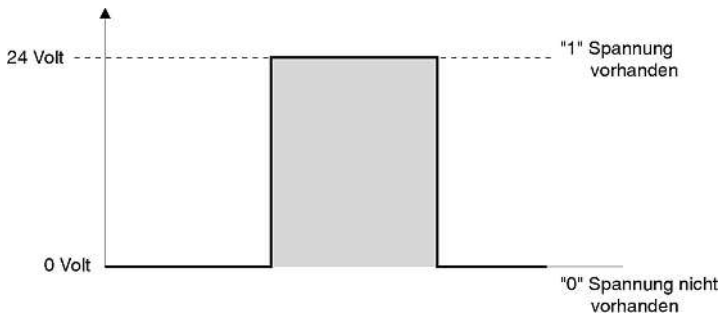


Bild 1.2
Ein Bit kann den Signalzustand «1» oder «0» annehmen

1.2.2 Byte

Für eine Einheit von 8 Binärzeichen steht der Begriff Byte.

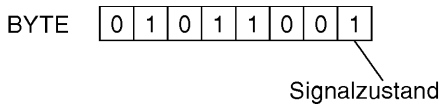


Bild 1.3
Ein Byte hat die Größe von 8 Bit

1.2.3 Wort

Ein Wort setzt sich aus 2 Byte oder 16 Binärzeichen zusammen. Mit «Wörtern» können z.B. dargestellt werden:

- Dualzahlen,
- Buchstaben,
- Steuerungsanweisungen.

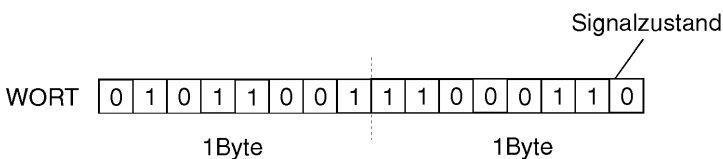
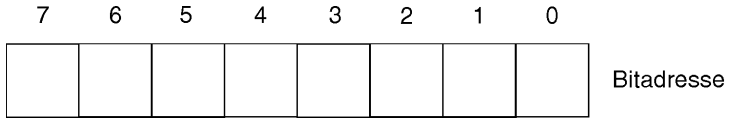


Bild 1.4
Ein Wort hat die Größe von 2 Byte oder 16 Bit

1.2.4 Bitadresse

Damit Bits erkannt und angesprochen werden können, wird jedem einzelnen Bit innerhalb eines Bytes eine Ziffer, eine sog. Bitadresse, zugeordnet.

Bild 1.5
In jedem Byte bekommt das Bit rechts außen die Bitadresse 0 und das links außen stehende Bit die Bitadresse 7



1.2.5 Byteadresse

Die einzelnen Bytes erhalten ebenfalls Nummern, die Byteadressen. Zusätzlich wird noch der Operand gekennzeichnet. So steht dann z.B. EB 2 für Eingangsbyte 2 oder AB 4 für Ausgangsbyte 4. Einzelne Bits sind eindeutig durch die Kombination von Bit- und Byteadresse adressiert.

Die Bitadresse wird durch einen Punkt von der Byteadresse getrennt. Rechts vom Punkt steht die Bitadresse, links davon die Byteadresse.

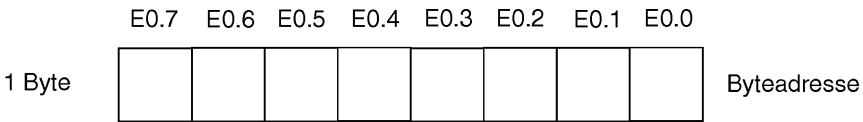
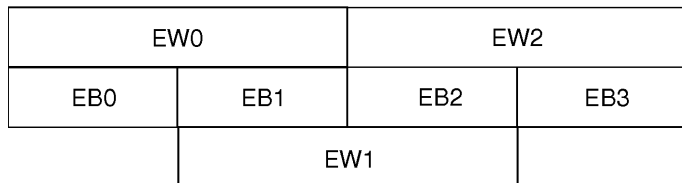


Bild 1.6 Beispiel: Byteadresse E0.0 oder A4.5

1.2.6 Wortadresse

Die Numerierung von Wörtern ergibt die Wortadresse. Bei der Verwendung von Wörtern, z.B. EW (Eingangswort), AW (Ausgangswort), MW (Merkerwort), DW (Datenwort) usw. ist die Wortadresse immer die kleinere Byteadresse der 2 dazugehörigen Bytes.

Bild 1.7
Wortadresse



2 Aufbau einer SPS

Speicherprogrammierbare Steuerungen werden in Serie gefertigt. Sie haben zunächst noch keine Aufgabe. Alle für die Steuerungstechnik benötigten Verknüpfungsglieder, Speicherfunktionen, Zeiten, Zähler usw. sind vom Hersteller integriert und werden über die Programmierung zu einer funktionierenden Steuerung verbunden. Es gibt eine Vielzahl von Steuergeräten, die sich durch folgende Funktionseinheiten unterscheiden:

- Ein- und Ausgänge,
- Speicherplätze,
- Zähler,
- Zeiten,
- Merkerfunktionen,
- Sonderfunktionen,
- Arbeitsgeschwindigkeit,
- Art der Programmbearbeitung.

Größere Steuergeräte werden aus einzelnen Baugruppen in Modulbauweise individuell zusammengesetzt. Mit diesem modularen System lassen sich, von der Grundausstattung ausgehend, SPS-Systeme zusammenstellen, die für den Anwendungsfall angepaßt werden können. Für kleinere Steuerungsaufgaben werden kompakt aufgebaute Steuergeräte angeboten. Sie sind in sich geschlossene Einheiten, mit einer fest vorgegebenen Anzahl von Ein- und Ausgängen.

2.1 Zusammensetzung einer SPS

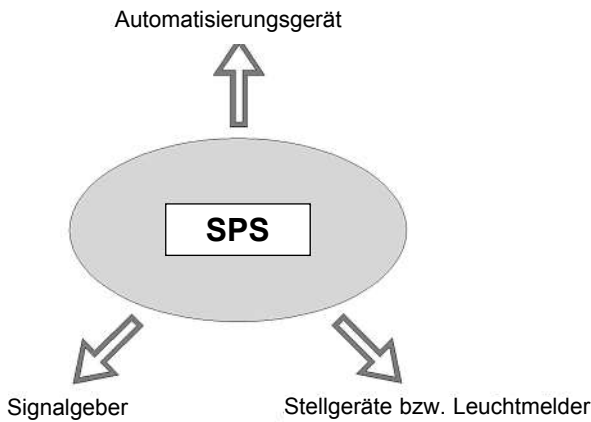


Bild 2.1
Grundsätzliche
Zusammensetzung
einer speicherpro-
grammierbaren
Steuerung

2.2 Zusammensetzung eines Automatisierungsgerätes

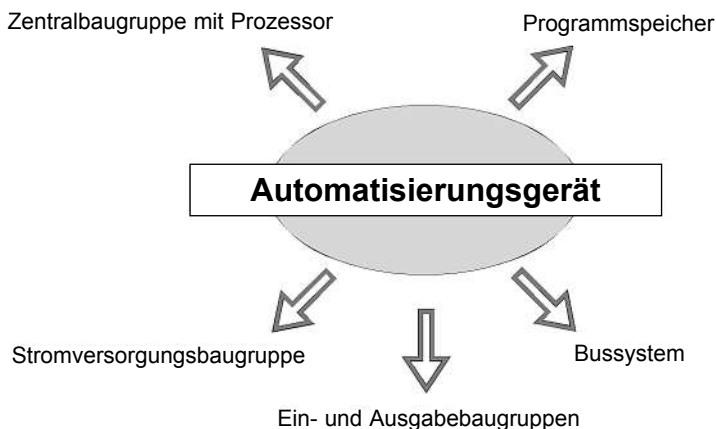


Bild 2.2
Wesentliche
Zusammensetzung
von einem Auto-
matisierungsgerät

- Signalgeber liefern digitale und analoge Signale zur Weiterverarbeitung an die SPS,
- Stellgeräte bzw. Leuchtmelder empfangen digitale und analoge Signale von der SPS.

2.3 Hardware-Voraussetzungen

Für das Arbeiten mit den nachfolgenden Beispielen werden folgende Hardware-Komponenten benötigt.

Profilschiene

Auf der Profilschiene werden die einzelnen Baugruppen montiert.

Stromversorgungsbaugruppe (PS)

Die Stromversorgungsbaugruppe wandelt die Netzspannung von 120/230 V AC in eine Gleichspannung 24 V DC für die Versorgung der S7-300 Baugruppen um.

Zentralbaugruppe (CPU)

Die Zentralbaugruppe dient zur Ausführung der Anwenderprogramme. Sie kommuniziert über die MPI-Schnittstelle mit anderen Baugruppen.

Ein- und Ausgabebaugruppe

Über die Ein- oder Ausgabebaugruppe werden Signale über Geber in die S7-300 ein- oder an die Ausgabebaugruppe ausgegeben. Der Signalzustand wird an den Baugruppen über eine LED-Anzeige angezeigt.

MPI-Kabel

Das MPI-Kabel verbindet den Computer mit der Zentralbaugruppe (CPU). Ohne dieses Kabel ist keine Kommunikation mit dem PC möglich.

Personalcomputer (PC)

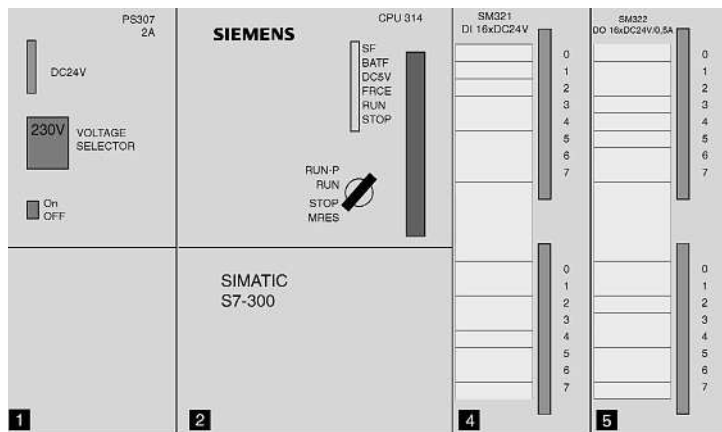
Der PC dient dem Konfigurieren, Parametrisieren und Programmieren der S7-300.

Programmiergerät (PG)

Das PG dient ebenfalls dem Konfigurieren, Parametrisieren und Programmieren der S7-300.

Für dieses Buch wird als Konfiguration eine S7-300 (CPU 314) mit einer digitalen Eingabebaugruppe (16 Eingänge) und einer digitalen Ausgabebaugruppe (16 Ausgänge) verwendet. Dabei enthält die Eingabebaugruppe die Adressen E 0.0...E 1.7 (EW0) und die Ausgabebaugruppe die Adressen A 4.0...A 5.7 (AW4).

Bild 2.3
Aufbau des Übungsgeräts



2.3.1 Hardware-Aufbau

Beim Aufbau einer S7-300 sind Steckplatzregeln zu beachten. Die Baugruppen werden von links nach rechts gesteckt:

- Die *Stromversorgungsbaugruppe (PS)* muß immer als 1. Baugruppe auf der Profilschiene aufgesteckt werden,
- die *Zentralbaugruppe (CPU)* muß als 2. Baugruppe aufgesteckt werden,
- die *Ein- und Ausgabebaugruppen* werden im Anschluß an die CPU aufgesteckt,
- neben der Zentralbaugruppe (CPU) dürfen maximal 8 Signalbaugruppen stecken,
- die Baugruppen dürfen waagrecht oder senkrecht gestellt werden.

2.4 Software-Voraussetzungen

Für das Programmieren der S7-300 wird das Betriebssystem **Windows 95** verwendet sowie das **Softwarepaket STEP 7, Version 3.xx**. In dieser Version sind die drei Darstellungsarten Anweisungsliste (AWL), Kontaktplan (KOP) und Funktionsplan (FUP) integriert. Die Vorgängerversionen z.B. 2.xx enthalten die Darstellung Funktionsplan (FUP) nicht.

Für das Programmieren der Übungen im Buch wird vorausgesetzt, daß das Softwarepaket STEP 7 der Fa. Siemens bereits installiert ist. Auch für den Umgang mit der Programmiersprache STEP 7 sind Kenntnisse im Betriebssystem Windows 95 (Dateiverwaltung) nötig. Das Installieren ist unter der Bedieneroberfläche Windows 95 durchzuführen.

2.4.1 Programmiersprache STEP 7

Infolge der Ablösung der SIMATIC S5 durch SIMATIC S7 im Herbst 1995 wurde eine neue Programmiersprache (STEP 7), basierend auf der Norm IEC 1131, entwickelt. Diese Programmiersprache bietet die gesamte Funktionalität für Parametrisieren, Konfigurieren und Programmieren des Automatisierungsgerätes S7-300. STEP 7 läuft unter Windows 95 und arbeitet objektorientiert. Die Symbole für die Objekte sind auf der grafischen Bedieneroberfläche abgebildet. In STEP 7 ist eine Onlinehilfe integriert, die wertvolle Hinweise und Tips gibt.

2.4.2 Objekte

Es werden unter der grafischen Bedieneroberfläche Objekte mit Symbolen dargestellt. Ein Symbol ist einem bestimmten Objekt zugeordnet. STEP-7-Objekte bieten den Zugang zu folgenden Bearbeitungsfunktionen:

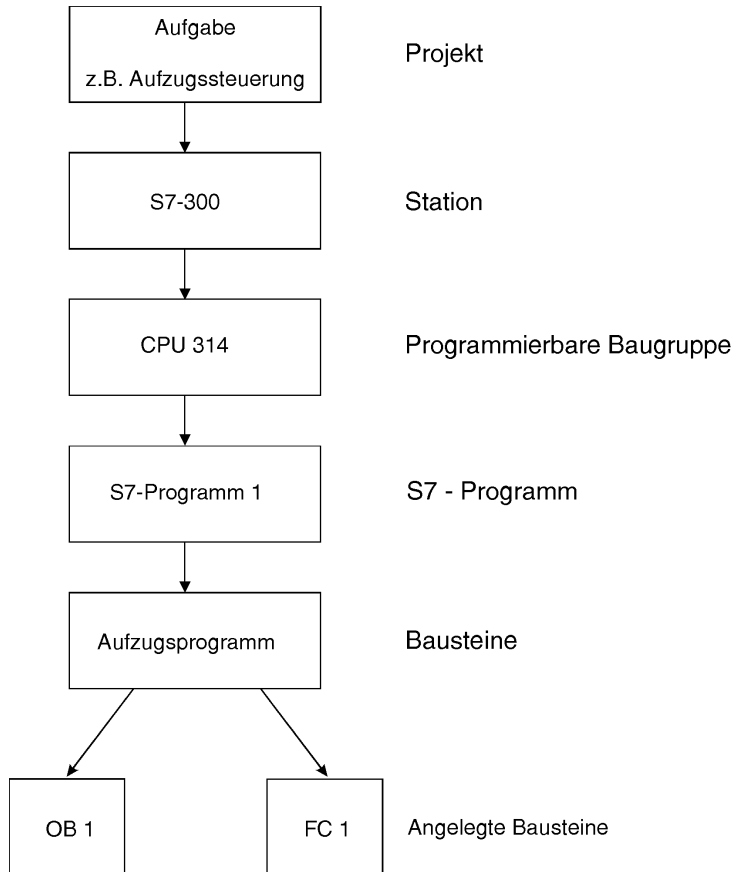
- Objekte anlegen und öffnen,
- Objekte bearbeiten und speichern,
- Objekte umbenennen,
- Objekte löschen,
- Objekte kopieren und einfügen.

Ein Objekt kann z.B. ein Projekt, eine SIMATIC-Station, ein S7-Programm usw. als Symbol enthalten.

2.4.3 Projekte

STEP 7 bietet die Möglichkeit, eine Anlage in Projekte zu gliedern. Ein Projekt enthält die gesamten Daten für eine Automatisierungslösung. Man kann als wichtigstes Objekt das Projekt bezeichnen.

Bild 2.4
Projektstruktur mit
SIMATIC S7



Die Daten für eine Automatisierungslösung werden in einem Projekt verwaltet. In unserem Beispiel ist das Projekt eine Aufzugsteuerung. In diesem Projekt liegt die gesamte Automatisierungslösung. Das Projekt enthält z.B. die Station S7-300 sowie das Automatisierungsgerät (AG) mit der CPU 314. In dieses AG wird das S7-Programm eingegeben. Das Programm gliedert sich dann in die entsprechenden Bausteine wie z.B. in OB 1 und FC 1.

2.4.4 Konfigurieren einer S7-300

Beim Konfigurieren werden die S7-Baugruppen in einer Konfigurationstabelle angeordnet. Die Baugruppen können aus einem elektronischen Katalog gewählt und in der Konfigurationstabelle dem Steckplatz entsprechend eingetragen werden. Der Steckplatz in der Konfigurationstabelle entspricht dem Steckplatz auf dem Baugruppenträger. Jeder Baugruppe wird dann automatisch eine Adresse zugeordnet.

2.4.5 Parametrisieren

Die Eigenschaften der einzelnen Baugruppen können verschiedenartig eingestellt werden. Ein Parameter ist z.B. die Zykluszeitüberwachung bei der CPU S7-300. Der Parameter ist veränderbar.

3 Funktionsweise einer SPS

3.1 Baugruppen der SPS

Die verschiedenen Baugruppen einer SPS erläutert Bild 3.1.

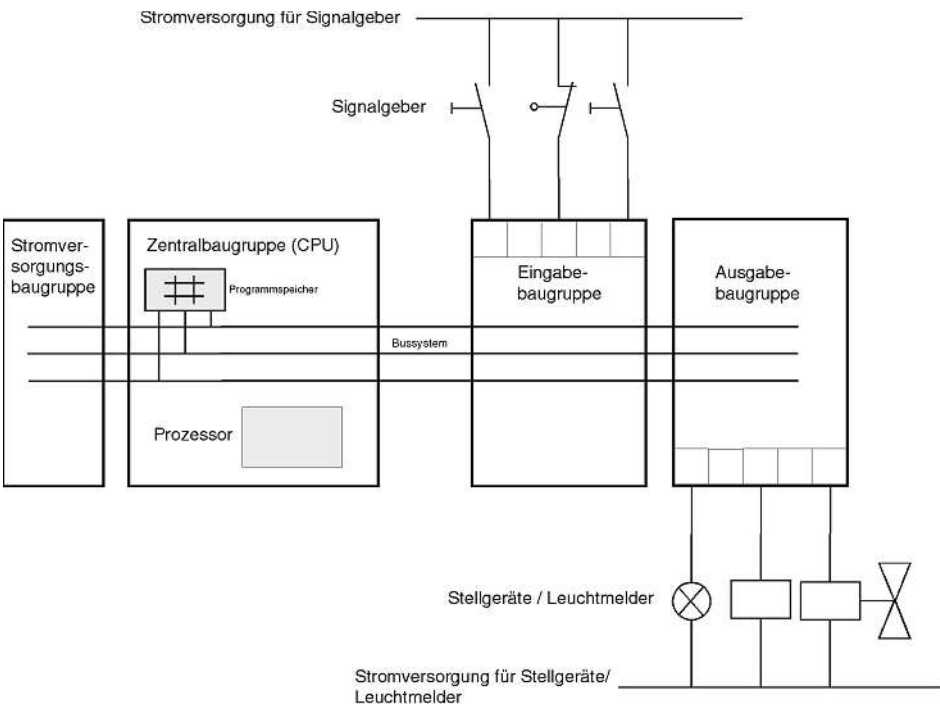


Bild 3.1 Zusammenwirken und Anordnung der Baugruppen einer SPS

3.1.1 Stromversorgungsbaugruppe

Die Stromversorgungsbaugruppe erzeugt aus der Netzspannung die Spannung für die elektronischen Baugruppen des Automatisierungsgerätes. Die Höhe dieser Spannung beträgt 24 V Gleichspannung. Spannungen für Signalgeber, Stellgeräte und Leuchtmelder, die über 24 V liegen (24...220 V), liefern zusätzlich dafür vorgesehene Netzgeräte bzw. Steuertransformatoren.

3.1.2 Programmspeicher

Speicherelemente sind Bauelemente, in denen Informationen in Form von binären Signalen hinterlegt (gespeichert) werden können. Als Programmspeicher werden überwiegend Halbleiterspeicher eingesetzt. Ein Speicher besteht aus:

- 512,
- 1024,
- 2048 usw. Speicherzellen.

Es ist üblich, die Kapazität des Programmspeichers (d.h. die Anzahl der Speicherzellen) in **Vielfachen von 1 K** (1 K steht hier für 1024) anzugeben. In jeder Speicherzelle kann mit Hilfe eines Programmiergerätes eine Steueranweisung geschrieben (programmiert) werden. Jedes Binärelement einer Speicherzelle kann dabei den Signalzustand «1» oder «0» annehmen.

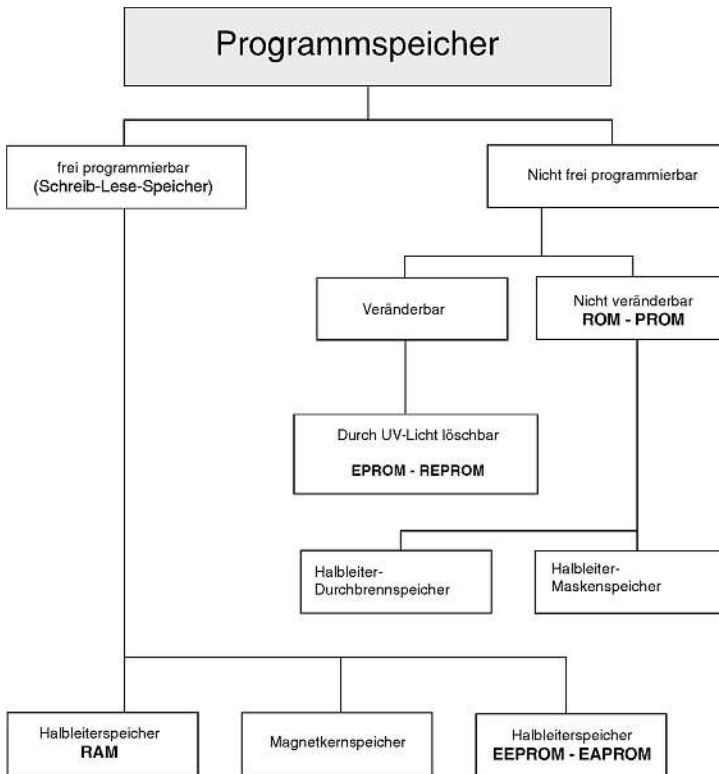


Bild 3.2
Schema eines Programmspeichers

RAM (random-access memory)

Mit RAM bezeichnet man einen in Halbleitertechnik gebauten *Schreib-Lese-Speicher*. Die einzelnen Speicherplätze sind mit Adressen gekennzeichnet, mit deren Hilfe frei auf die Speicherzellen zugegriffen werden kann.

In die Speicherzellen können beliebig oft Informationen geschrieben werden. Die Information wird ausgelesen, ohne daß der Informationsinhalt verlorenght.

RAM-Speicher sind jedoch flüchtige Speicher, d.h., daß deren Informationsinhalt bei Ausfall der Speisespannung verlorenght. RAM-Speicher werden elektrisch gelöscht.

ROM (read-only memory)

Festwertspeicher ROM enthalten eine nicht löschbare und nicht veränderbare Information. Die Bezeichnung ROM steht für *Nur-Lese-Speicher*. Die Information wird vom Hersteller eingegeben und kann nur von ihm verändert werden.

EPROM (erasable read-only memory)

EPROM steht für *löschrbarer, programmierbarer* Festwertspeicher. Der Inhalt des EPROM ist in seiner Gesamtheit durch UV-Licht löschrbar und danach wieder programmierbar. Er ist deswegen gut geeignet, um ohne Datenverlust transportiert zu werden.

REEPROM (reprogrammable read-only memory)

REEPROM steht für *neuprogrammierbarer* Festwertspeicher. Er ist ebenfalls nur durch UV-Licht löschrbar.

EEPROM (electrically erasable read-only memory)

EEPROM steht für *elektrisch löschrbarer, programmierbarer* Festwertspeicher. Im EEPROM läßt sich jede Speicherzelle elektrisch löschen und beschreiben.

EAPROM (electrically alterable read-only memory)

EAPROM steht für *elektrisch umprogrammierbarer* Festwertspeicher.

3.1.3 Zentralbaugruppe (CPU)

Die von den Signalen kommende Spannung wird auf die Anschlußleiste der Eingabebaugruppe geschaltet. In der CPU (Zentralbaugruppe, s. Bild 3.3) bearbeitet der Prozessor das im Speicher stehende Programm und fragt dabei ab, ob die einzelnen Eingänge des Gerätes Spannung führen oder nicht. Abhängig von diesem Zustand an den Eingängen und von dem im Speicher stehenden Programm weist der Prozessor die Ausgabebaugruppe an, auf die entsprechenden Anschlüsse der Anschlußleiste Spannung zu schalten. Wiederum abhängig vom Spannungszustand an den Anschlüssen der Ausgabebaugruppen werden die angeschlossenen Stellgeräte bzw. Leuchtmelder ein- oder ausgeschaltet.

Der Adressenzähler fragt den Programmspeicher Anweisung für Anweisung nacheinander (seriell) ab und bewirkt die programmabhängige Informationsübertragung aus dem Programmspeicher zum Anweisungsregister. Alle Speicher in einem Prozessor nennt man üblicherweise Register.

Das Steuerwerk erhält seine Anweisungen vom Anweisungsregister. Während das Steuerwerk die aktuelle Anweisung bearbeitet, schiebt der Adressenzähler die nächste Anweisung in das Anweisungsregister.

Auf die Statusübertragung der Eingänge in das Prozeßabbild der Eingänge (PAE) folgen Verknüpfung, der Einsatz der Zeitglieder, Zähler, Akkus und die Übertragung der Verknüpfungsergebnisse (VKE) in das Prozeßabbild der Ausgänge (PAA). Wird nach Abarbeitung des Anwenderprogrammes Bausteinende (BE) erkannt, so erfolgt die Übertragung des jeweiligen Status aus dem PAA an die Ausgänge.

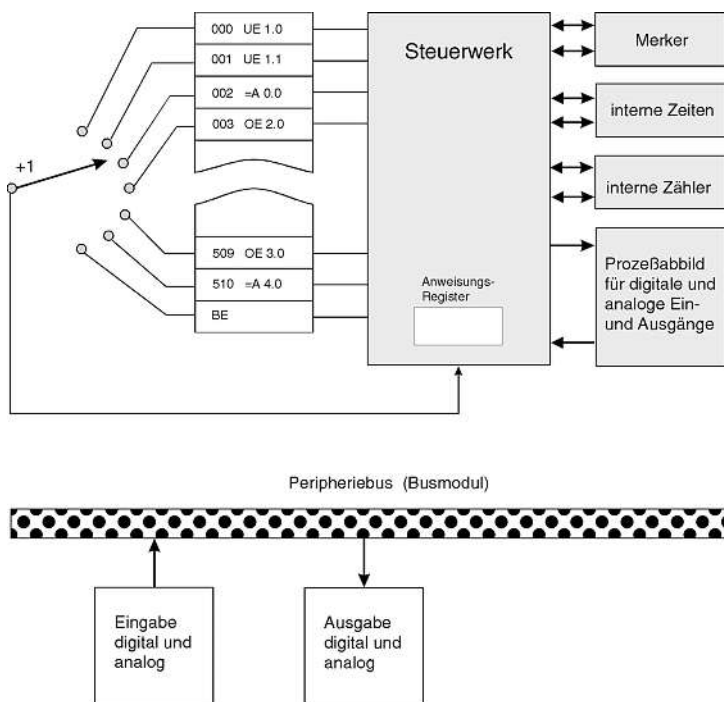


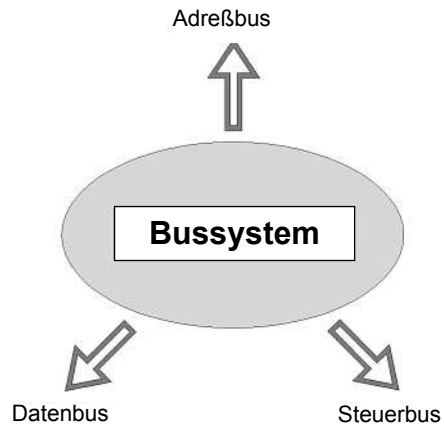
Bild 3.3
Zentralbaugruppe
(CPU)

Der Peripheriebus wickelt den Datenaustausch zwischen Zentralbaugruppe und Peripherie ab. Zur Peripherie gehören die digitalen Eingabe- und Ausgabebaugruppen, analoge Eingabe- und Ausgabebaugruppen sowie Zeit-, Zähl- und Grenzwertbaugruppen.

3.1.4 Bussystem

Das Bussystem (s. Bild 3.4) ist eine Sammelleitung zur Übertragung von Signalen. So erfolgt der Signalaustausch im Automatisierungsgerät (AG) zwischen dem Prozessor und den Ein- und Ausgabebaugruppen über ein sog. Prozeßbussystem. Der Prozeßbus besteht aus 3 parallelen Signalleitungen:

Bild 3.4
Das Bussystem



- mit dem Adreßbus werden Adressen auf den einzelnen Baugruppen angesprochen,
- mit dem Datenbus werden Daten z.B. von Eingabe- nach Ausgabebaugruppen übertragen,
- auf dem Steuerbus werden die Signale zur Steuerung und Überwachung des Funktionsablaufs innerhalb des Automatisierungsgerätes übermittelt.

3.1.5 Ein- und Ausgabebaugruppen

Über die Ein- und Ausgabebaugruppen erfolgt der Datenaustausch mit dem zu steuernden oder zu regelnden Prozeß.

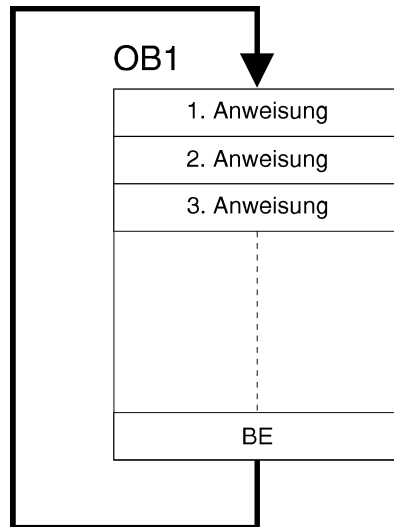
4 Programmbearbeitung und Programmierung

Bei der Programmbearbeitung gibt es die Möglichkeit der linearen und der strukturierten Bearbeitung.

4.1 Lineare Programmbearbeitung

Die lineare Programmbearbeitung (s. Bild 4.1) wird meist für einfache, nicht zu umfangreiche Steuerungen verwendet und kann in einem einzigen Organisationsbaustein (OB) realisiert werden. Dabei werden die Anweisungen vom Steuergerät in der Reihenfolge bearbeitet, in der sie im Programmspeicher hinterlegt sind. Wenn das Programmende (BE) erreicht ist, beginnt die Programmbearbeitung wieder von vorne. Die Zeit, die ein Gerät für eine einmalige Bearbeitung aller Anweisungen benötigt, wird Zykluszeit genannt. Man spricht deshalb auch von einer zyklischen Bearbeitung.

Bild 4.1
Lineare Programm-
bearbeitung



4.2 Strukturierte Programmierung

Das Programm unterteilt man bei umfangreichen Steuerungsaufgaben in kleine, überschaubare und nach Funktion geordnete Programmbausteine. Das hat den Vorteil, Programmteile einzeln austesten zu können und sie bei Funktionieren zu einer Gesamtfunktion zusammenzuführen. STEP 7 bietet dafür folgende Anwenderbausteine an:

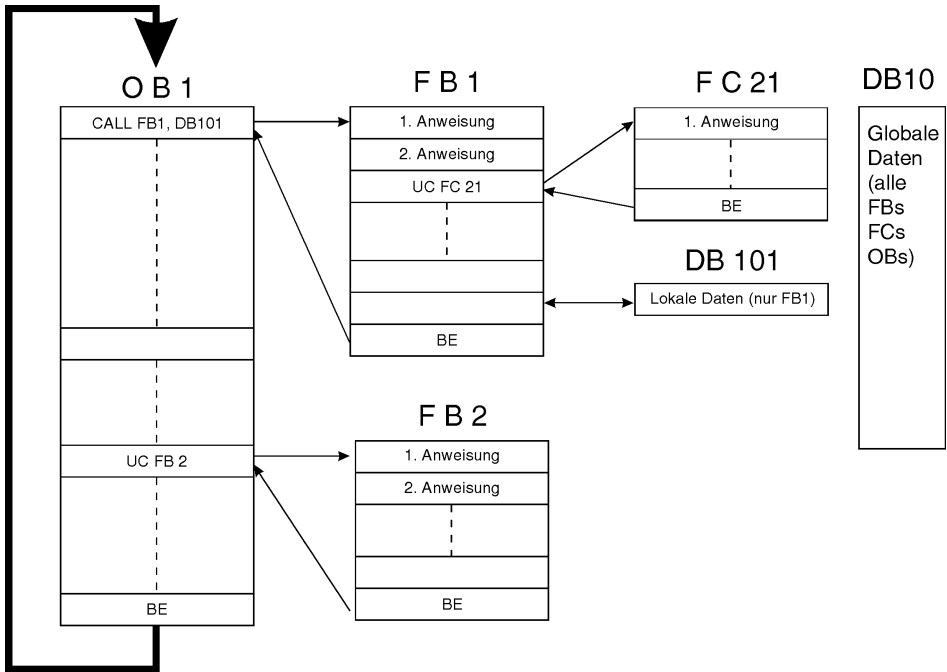


Bild 4.2 Strukturierte Programmierung mit Anwenderbausteinen mit: OB Organisationsbaustein, FB Funktionsbaustein, FC Funktion, DB Datenbaustein

Organisationsbaustein (OB)

Ein OB wird vom Betriebssystem zyklisch aufgerufen und bildet die Schnittstelle zwischen Anwenderprogramm und Betriebssystem. Im OB wird dem Steuerwerk des Automatisierungsgeräts (AG) über Bausteinaufrufbefehle mitgeteilt, welche Programmbausteine es bearbeiten soll.

Funktionsbaustein (FB)

Der FB verfügt über einen zugeordneten Speicherbereich. Wenn ein FB aufgerufen wird, kann ihm ein Datenbaustein (DB) zugeordnet werden. Auf die Daten in diesem Instanz-DB kann über Aufrufe aus dem FB zugegriffen werden. Ein FB kann verschiedenen DBs zugeordnet werden. Weitere FBs und FCs können über Bausteinaufrufbefehle in einem Funktionsbaustein aufgerufen werden.