

Hans Berger

Automatisieren mit STEP 7 in KOP und FUP

Speicherprogrammierbare Steuerungen
SIMATIC S7-300/400

SIEMENS

KOP

FUP

DB10.DBW12 -IN IN1
C#123
DB10.DBW20 -IN IN2

E10.0

DB10.DBW20 -IN1

DB10.DBW14 -IN2

E5.0

E1.3

E1.4

E1.6

E1.7

E1.5

A3.0

M3.1

≥ 1

&

6. Auflage

Berger Automatisieren mit STEP 7 in KOP und FUP

Automatisieren mit STEP 7 in KOP und FUP

Speicherprogrammierbare Steuerungen
SIMATIC S7-300/400

von Hans Berger

6., überarbeitete und erweiterte Auflage, 2012

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.

Dem Buch liegt eine Demo DVD der Siemens AG bei.

„**SIMATIC STEP 7 Professional, Edition 2010 SR1, Trial License**“ umfasst: SIMATIC STEP 7 V5.5 SP1, S7-GRAPH V5.3 SP7, S7-SCL V5.3 SP6, S7-PLCSIM V5.4 SP5 und ist 14 Tage zu Testzwecken nutzbar.

Die Software ist nur unter Microsoft Windows XP 32 Bit Professional Edition SP3 oder Microsoft Windows 7 32/64 Bit Professional Edition SP1 oder Microsoft Windows 7 32/64 Bit Ultimate Edition SP1 ablauffähig.

Weitere Informationen erhalten Sie im Internet unter

<http://www.siemens.de/sce/promotoren>

<http://www.siemens.de/sce/module>

<http://www.siemens.de/sce/tp>

Die Programmierbeispiele sind dafür vorgesehen, schwerpunktmäßig die AWL- und SCL-Funktionen zu beschreiben und Anwendern von SIMATIC S7 Anhaltspunkte zu geben, wie bestimmte Aufgabenstellungen mit dieser Steuerung programmtechnisch gelöst werden können.

Bei den im Buch gezeigten Programmierbeispielen handelt es sich um Lösungsideen ohne Anspruch auf Vollständigkeit oder auf Ablauffähigkeit bei zukünftigen Ausgabeständen von STEP 7 oder S7-300/400. Für die Einhaltung entsprechender Sicherheitsvorschriften ist bei der Anwendung zusätzliche Sorge zu tragen.

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.

Für Anregungen zum Inhalt weiterer Auflagen sind Autor und Verlag immer dankbar.

Publicis Publishing

Postfach 3240

91050 Erlangen

E-Mail: publishing-distribution@publicis.de

ISBN 978-3-89578-411-8

6. Auflage, 2012

Herausgeber Siemens Aktiengesellschaft, Berlin und München

Verlag: Publicis Publishing, Erlangen

© 2012 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. Durch diese Totally Integrated Automation (TIA) wird eine durchgängige Projektierung, Programmierung, Datenhaltung und Kommunikation im gesamten Automatisierungssystem erreicht. Aufeinander abgestimmte Kommunikationsmechanismen ermöglichen die harmonische Zusammenarbeit von Speicherprogrammierbaren Steuerungen, Visualisierungssystemen und dezentraler Peripherie.

STEP 7 übernimmt als Basiswerkzeug für SIMATIC die Klammerfunktion bei Totally Integrated Automation. Mit STEP 7 erfolgt die Projektierung und Programmierung der Automatisierungssysteme SIMATIC S7, SIMATIC C7 und SIMATIC WinAC. Als Betriebssystem wurde Microsoft Windows gewählt und somit die Welt der Standard-PCs mit der auch in der Bürowelt weit verbreiteten und damit bekannten Bedienoberfläche erschlossen.

Für die Bausteinprogrammierung stellt STEP 7 nach DIN EN 6.1131-3 genormte Programmiersprachen zur Verfügung: AWL (Anweisungsliste, eine Assembler-ähnliche Sprache), KOP (Kontaktplan, eine Stromlaufplan-ähnliche Darstellung), FUP (Funktionsplan) und das Optionspaket S7-SCL (Structured Control Language, eine Pascal-ähnliche Hochsprache). Zusätzliche Optionspakete ergänzen diese Sprachen: S7-GRAPH (Ablaufkettensteuerung), S7-HiGraph (Programmierung als Zustandsgraph) und CFC (Funktionsplan-ähnliche Verschaltung von Bausteinen). Die verschiedenen Darstellungsarten geben jedem Anwender die Möglichkeit, sich die für ihn geeignete Be-

schreibung der Steuerungsfunktion auszuwählen. Die so gewonnene weitgehende Übereinstimmung mit der Darstellung der zu lösenden Steuerungsaufgabe vereinfacht wesentlich die Handhabung von STEP 7.

Dieses Buch beschreibt die Programmiersprachen KOP und FUP für S7-300/400. Als wertvolle Ergänzung zur Sprachbeschreibung bietet es nach einer Einführung in das Automatisierungssystem S7-300/400 wertvolle, praxisnahe Hinweise für die grundsätzliche Handhabung von STEP 7 bei der Projektierung der SIMATIC-Steuerungen, ihrer Vernetzung und Programmierung. Die Beschreibung der „Basisfunktionen“ einer binären Steuerung, wie z.B. logische Verknüpfungen oder Speicherfunktionen, erleichtern besonders für Anfänger oder Umsteiger von Schützensteuerung den Einstieg in STEP 7. Die Digitalfunktionen erläutern die Verknüpfung digitaler Werte, z.B. Grundrechnungsarten, Vergleiche oder Datentypwandlung.

Das Buch zeigt, wie Sie mit KOP bzw. FUP die Programmbearbeitung (den Programmfluss) steuern und ein Programm strukturiert aufbauen können. Zusätzlich zum zyklisch bearbeiteten Hauptprogramm können Sie ereignisgesteuerte Programmteile einbinden sowie das Verhalten der Steuerung im Anlauf und im Fehlerfall beeinflussen. Eine Gesamtübersicht der Systemfunktionen und des Funktionsvorrats für KOP und FUP runden das Buch ab. Der Inhalt des vorliegenden Buchs beschreibt die Programmiersoftware STEP 7 in der Version 5.5.

Erlangen, im Mai 2012

Hans Berger

Der Inhalt des Buchs auf einen Blick

Überblick über das
Automatisierungssystem
S7-300/400

SPS-Funktionen
vergleichbar mit einer
Schützensteuerung

Umgang mit Zahlen
und digitalen Operanden

Einführung	Basisfunktionen	Digitalfunktionen
1 Automatisierungssystem S7-300/400 Aufbau des Automatisierungssystems SIMATIC; Speicherbereiche Dezentrale Peripherie (PROFIBUS DP); Kommunikation (Subnetze); Baugruppenadressen; Operandenbereiche	4 Binäre Verknüpfungen UND-, ODER- und Exklusiv-ODER-Funktion; Zusammengesetzte Verknüpfungen	9 Vergleichsfunktionen Vergleiche nach INT, DINT und REAL
2 Programmiersoftware STEP 7 Projekt bearbeiten; Station konfigurieren; Netz projektieren; Symboleditor; KOP/FUP-Programmator; Online-Funktionen; KOP- und FUP-Programme testen	5 Speicherfunktionen Zuweisung, Setzen und Rücksetzen; Flankenauswertung; Beispiel Förderbandsteuerung	10 Arithmetische Funktionen Grundrechnungsarten nach INT, DINT und REAL;
3 SIMATIC S7- Programm Programmbearbeitung; Bausteinarten; Codebausteine und Datenbausteine editieren; Variablen adressieren, Konstantendarstellung, Beschreibung Datentypen	6 Übertragungsfunktionen Laden und Transferieren; Systemfunktionen zur Datenübertragung	11 Mathematische Funktionen Winkelfunktionen; Arcusfunktionen; Quadrat, Quadratwurzel Potenzieren, Logarithmieren
	7 Zeitfunktionen SIMATIC-Zeiten starten mit fünf verschiedenen Verhaltensweisen, rücksetzen und abfragen; IEC-Zeitfunktionen	12 Umwandlungsfunktionen Datentypwandlung; Komplementbildung
	8 Zählfunktionen SIMATIC-Zähler vorwärtszählen, rückwärtszählen, setzen, rücksetzen und abfragen; IEC-Zählfunktionen	13 Schiebefunktionen Schiebefunktionen, Rotierfunktionen
		14 Wortverknüpfungen Digitale UND-, ODER- und Exklusive-ODER-Verknüpfung

Steuerung des
Programmablaufs,
Bausteinfunktionen

Bearbeitung des
Anwenderprogramms

Ergänzungen zu KOP/FUP;
Bausteinbibliotheken,
Funktionsübersichten

Programmfluss-Steuerung

15 Statusbits

Binäranzeigen,
Digitalanzeigen;
Setzen und Auswerten der
Statusbits;
EN/ENO-Mechanismus

16 Sprungfunktionen

Absoluter Sprung;
Sprung bei VKE = „1“
Sprung bei VKE = „0“

17 Master Control Relais

MCR-Abhängigkeit,
MCR-Bereich,
MCR-Zone

18 Bausteinfunktionen

Bausteinaufrufe,
Bausteinende;
temporäre und statische
Lokaldaten;
Datenoperanden adressieren
Datenbaustein aufschlagen

19 Bausteinparameter

Formalparameter,
Aktualparameter;
Deklarieren, Versorgen und
„Weiterreichen“

Programmbearbeitung

20 Hauptprogramm

Programmstruktur;
Zyklussteuerung
(Zykluszeit, Reaktionszeit,
Startinformation,
Hintergrundbearbeitung);
Programmfunktionen;
Kommunikation mit
PROFIBUS und PROFINET;
GD-Kommunikation;
S7- und S7-Basis-
Kommunikation

21 Alarmbearbeitung

Uhrzeitalarme;
Verzögerungsalarme;
Weckalarme;
Prozessalarme;
DPV1-Alarme
Mehrprozessoralarm;
Alarmereignisse hantieren

22 Anlaufverhalten

Kaltstart, Warmstart,
Wiederanlauf;
STOP, HALT, Urlöschen;
Baugruppen parametrieren

23 Fehlerbehandlung

Synchronfehler;
Asynchronfehler;
Systemdiagnose

Anhang

24 Ergänzungen zur grafi- schen Programmierung

Bausteinschutz
KNOW_HOW_PROTECT;
Beschreibung der Zeiger,
Indirekte Adressierung;
Kurzbeschreibung
„Beispiel Telegramm“

25 Bausteinbibliotheken

Organisationsbausteine;
Systembausteine;
IEC-Funktionen;
S5/S7-Konverterfunktionen;
TI/S7-Konverterfunktionen;
Regelungsfunktionen;
DP-Funktionen

26 Funktionsübersicht KOP

Basisfunktionen;
Digitalfunktionen;
Programmfluss-Steuerung

27 Funktionsübersicht FUP

Basisfunktionen;
Digitalfunktionen;
Programmfluss-Steuerung

Die Programmierbeispiele auf einen Blick

Das vorliegende Buch bietet viele Abbildungen zur Darstellung und Anwendung der Programmiersprachen KOP und FUP. Alle im Buch gezeigten Programmteile sowie zusätzliche Beispiele finden Sie in den zwei Bibliotheken KOP_Buch und FUP_Buch, die Sie aus dem Download-Bereich von der Website des Verlags heruntergeladen können:

www.publicis.de/books.

Die Bibliotheken KOP_Buch und FUP_Buch enthalten acht Programme, die im Wesentlichen Anschauungsbeispiele zur grafischen Darstellung sind. Zwei umfangreichere Beispiele zeigen die Programmierung von Funktionen, Funktionsbausteinen und Lokalinstanzen (Beispiel Fördertechnik) und den Umgang mit Daten (Beispiel Telegramm).

Bibliothek KOP_Buch

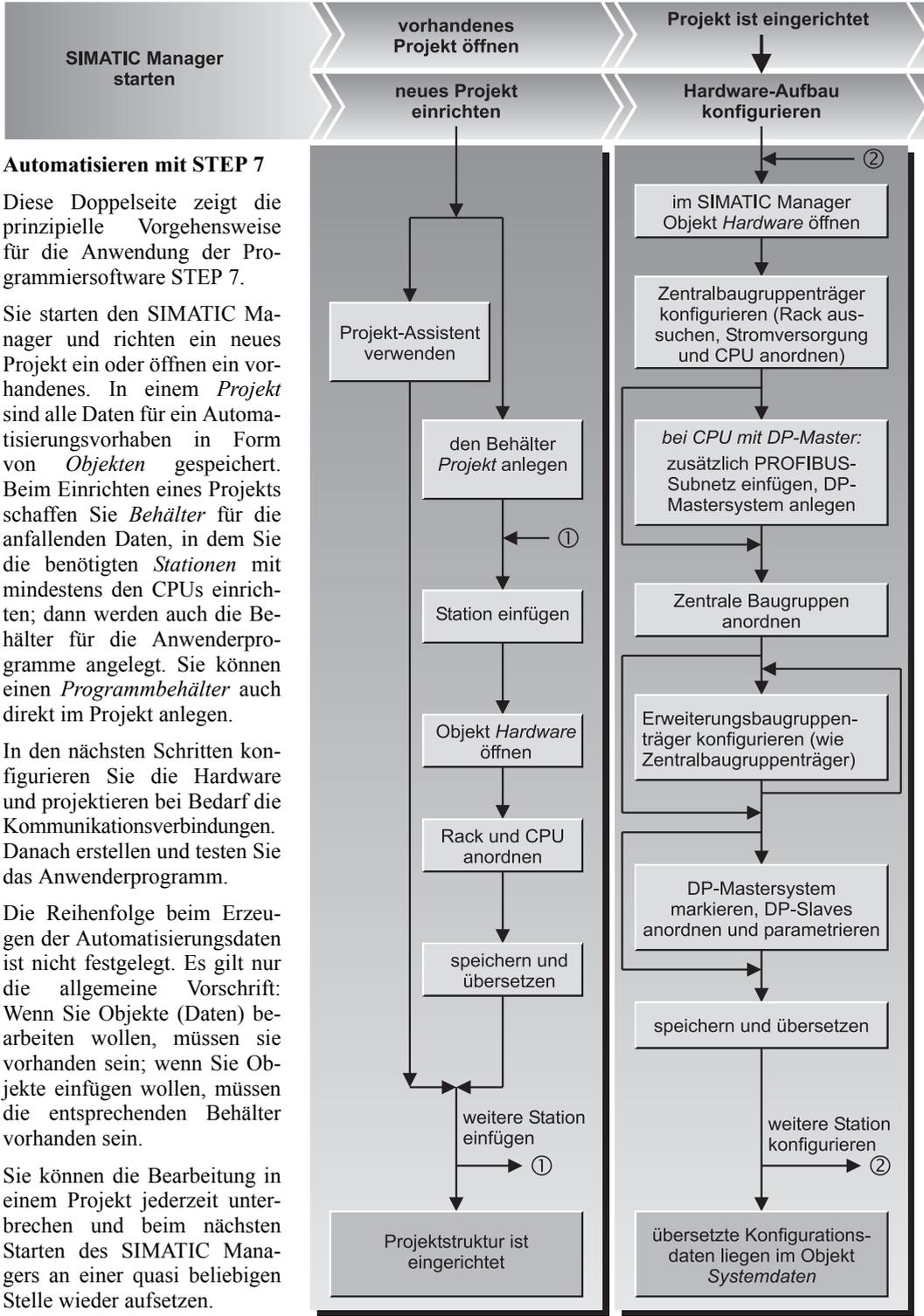
Datentypen Beispiele zur Definition und Anwendung	Programmbearbeitung Beispiele SFC-Aufrufe
FB 101 Elementare Datentypen FB 102 Zusammengesetzte Datentypen FB 103 Parametertypen	FB 120 Kapitel 20: Hauptprogramm FB 121 Kapitel 21: Alarmbearbeitung FB 122 Kapitel 22: Anlaufverhalten FB 123 Kapitel 23: Fehlerbehandlung
Basisfunktionen Beispiele zur KOP-Darstellung	Beispiel Fördertechnik Beispiele zu Basisfunktionen und Lokalinstanzen
FB 104 Kapitel 4: Reihen- und Parallelschaltung FB 105 Kapitel 5: Speicherfunktionen FB 106 Kapitel 6: Übertragungsfunktionen FB 107 Kapitel 7: Zeitfunktionen FB 108 Kapitel 8: Zählfunktionen	FC 11 Förderbandsteuerung FC 12 Fördergutzähler FB 20 Zuförderung FB 21 Förderband FB 22 Stückgutzähler
Digitalfunktionen Beispiele zur KOP-Darstellung	Beispiel Telegramm Beispiele zum Umgang mit Daten
FB 109 Kapitel 9: Vergleichsfunktionen FB 110 Kapitel 10: Arithmetische Funktionen FB 111 Kapitel 11: Mathematische Funktionen FB 112 Kapitel 12: Umwandlungsfunktionen FB 113 Kapitel 13: Schiebefunktionen FB 114 Kapitel 14: Wortverknüpfungen	UDT 51 Datenstruktur Header UDT 52 Datenstruktur Telegramm FB 51 Telegramm aufbereiten FB 52 Telegramm speichern FC 51 Uhrzeit abfragen FC 52 Datenbereich indirekt kopieren
Programmfluss-Steuerung Beispiele zur KOP-Darstellung	Beispiele allgemein
FB 115 Kapitel 15: Statusbits FB 116 Kapitel 16: Sprungfunktionen FB 117 Kapitel 17: Master Control Relay FB 118 Kapitel 18: Bausteinfunktionen FB 119 Kapitel 19: Bausteinparameter	FC 41 Bereichsüberwachung FC 42 Grenzwertmeldung FC 43 Zinseszins-Rechnung FC 44 Doppelwortweise Flankenauswertung

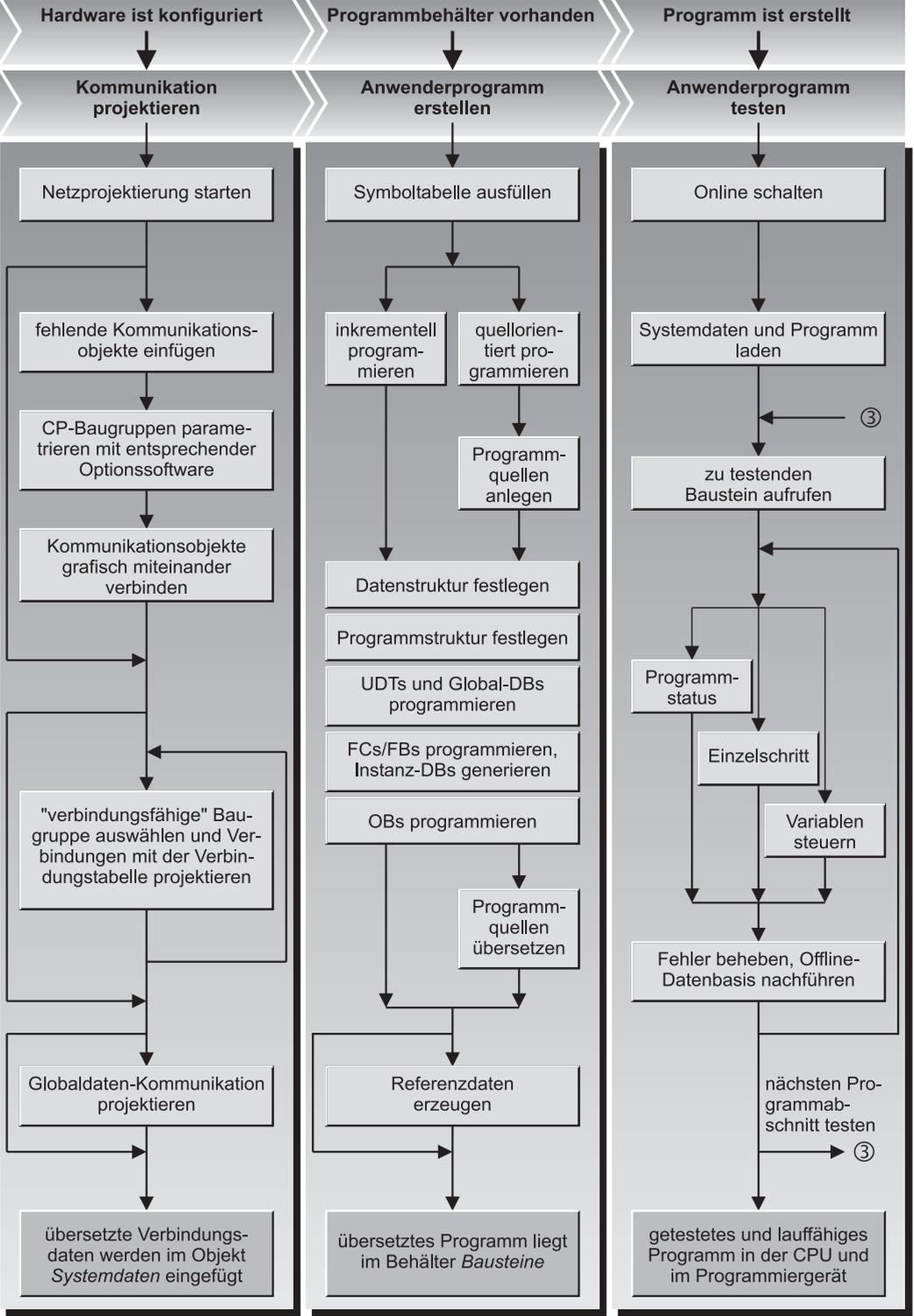
Die Bibliotheken liegen im archivierten Zustand vor. Bevor Sie mit ihnen arbeiten können, müssen Sie die Bibliotheken dearchivieren. Wählen Sie im SIMATIC Manager den Menüpunkt DATEI → DEARCHIVIEREN und folgen Sie den Anweisungen (siehe auch die Datei LIESMICH.TXT zum Herunterladen).

Zum Ausprobieren richten Sie ein Projekt ein, das der Ihnen vorliegenden Hardware-Konfiguration entspricht und kopieren das Programm einschließlich der Symboltabelle von der Bibliothek in das Projekt. Nun können Sie die Beispielprogramme aufrufen, für eigene Anwendungen abwandeln und online testen.

Bibliothek FUP_Buch

Datentypen Beispiele zur Definition und Anwendung	Programmbearbeitung Beispiele SFC-Aufrufe
FB 101 Elementare Datentypen FB 102 Zusammengesetzte Datentypen FB 103 Parametertypen	FB 120 Kapitel 20: Hauptprogramm FB 121 Kapitel 21: Alarmbearbeitung FB 122 Kapitel 22: Anlaufverhalten FB 123 Kapitel 23: Fehlerbehandlung
Basisfunktionen Beispiele zur FUP-Darstellung	Beispiel Fördertechnik Beispiele zu Basisfunktionen und Lokalinstanzen
FB 104 Kapitel 4: Binäre Verknüpfungen FB 105 Kapitel 5: Speicherfunktionen FB 106 Kapitel 6: Übertragungsfunktionen FB 107 Kapitel 7: Zeitfunktionen FB 108 Kapitel 8: Zählfunktionen	FC 11 Förderbandsteuerung FC 12 Fördergutzähler FB 20 Zuförderung FB 21 Förderband FB 22 Stückgutzähler
Digitalfunktionen Beispiele zur FUP-Darstellung	Beispiel Telegramm Beispiele zum Umgang mit Daten
FB 109 Kapitel 9: Vergleichsfunktionen FB 110 Kapitel 10: Arithmetische Funktionen FB 111 Kapitel 11: Mathematische Funktionen FB 112 Kapitel 12: Umwandlungsfunktionen FB 113 Kapitel 13: Schiebefunktionen FB 114 Kapitel 14: Wortverknüpfungen	UDT 51 Datenstruktur Header UDT 52 Datenstruktur Telegramm FB 51 Telegramm aufbereiten FB 52 Telegramm speichern FC 51 Uhrzeit abfragen FC 52 Datenbereich indirekt adressieren
Programmfluss-Steuerung Beispiele zur FUP-Darstellung	Beispiele allgemein
FB 115 Kapitel 15: Statusbits FB 116 Kapitel 16: Sprungfunktionen FB 117 Kapitel 17: Master Control Relay FB 118 Kapitel 18: Bausteinfunktionen FB 119 Kapitel 19: Bausteinparameter	FC 41 Bereichsüberwachung FC 42 Grenzwertmeldung FC 43 Zinseszins-Rechnung FC 44 Doppelwortweise Flankenauswertung





Inhaltsverzeichnis

Einführung	19	2 Programmiersoftware STEP 7	51
1 Automatisierungssystem SIMATIC S7-300/400	20	2.1 STEP 7 Basis	51
1.1 Aufbau des Automatisierungssysteme	20	2.1.1 Installation	51
1.1.1 Komponenten	20	2.1.2 Automation License Manager	51
1.1.2 S7-300-Station	20	2.1.3 SIMATIC Manager	52
1.1.3 S7-400-Station	23	2.1.4 Projekte und Bibliotheken	55
1.1.4 Hochverfügbare SIMATIC	23	2.1.5 Multiprojekte	56
1.1.5 Sicherheitsgerichtete SIMATIC	24	2.1.6 Online-Hilfe	56
1.1.6 Speicherbereiche der Zentralbaugruppe	25	2.2 Projekt bearbeiten	57
1.2 Dezentrale Peripherie	29	2.2.1 Projekt anlegen	57
1.2.1 PROFIBUS DP	29	2.2.2 Verwalten, reorganisieren und archivieren	58
1.2.2 PROFINET IO	31	2.2.3 Projektversionen	59
1.2.3 Aktor-/Sensor-Interface	32	2.2.4 Multiprojekt anlegen und bearbeiten	59
1.2.4 Netzübergänge	33	2.3 Station konfigurieren	61
1.3 Kommunikation	36	2.3.1 Baugruppen anordnen	63
1.3.1 Einführung	36	2.3.2 Baugruppen adressieren	63
1.3.2 Subnetze	38	2.3.3 Baugruppen parametrieren	64
1.3.3 Kommunikationsdienste	42	2.3.4 Baugruppen mit MPI vernetzen	64
1.3.4 Verbindungen	43	2.3.5 Baugruppen beobachten und steuern	65
1.4 Baugruppenadressen	44	2.4 Netz projektieren	65
1.4.1 Signalweg	44	2.4.1 Netzansicht konfigurieren	66
1.4.2 Steckplatzadresse	44	2.4.2 Dezentrale Peripherie mit der Netzprojektierung konfigurieren	67
1.4.3 Logische Adresse	44	2.4.3 Verbindungen projektieren	68
1.4.4 Baugruppenanfangsadresse	46	2.4.4 Netzübergänge	72
1.4.5 Diagnoseadresse	46	2.4.5 Verbindungsdaten laden	72
1.4.6 Adressen für Busteilnehmer	46	2.4.6 Projekte im Multiprojekt abgleichen	73
1.5 Operandenbereiche	47		
1.5.1 Nutzdatenbereich	47		
1.5.2 Prozessabbild	48		
1.5.3 Konsistente Nutzdaten	49		
1.5.4 Merker	50		

11	Mathematische Funktionen . . .	215	16	Sprungfunktionen	241
11.1	Bearbeitung einer mathematischen Funktion	215	16.1	Bearbeitung einer Sprungfunktion	241
11.2	Winkelfunktionen	217	16.2	Sprung absolut	242
11.3	Arcusfunktionen	217	16.3	Sprung bei VKE = „1“	243
11.4	Sonstige mathematische Funktionen	217	16.4	Sprung bei VKE = „0“	243
12	Umwandlungsfunktionen . . .	220	17	Master Control Relay	244
12.1	Bearbeitung einer Umwandlungsfunktion.	220	17.1	MCR-Abhängigkeit	244
12.2	Umwandeln von INT- und DINT-Zahlen.	222	17.2	MCR-Bereich	245
12.3	Umwandlung von BCD-Zahlen	223	17.3	MCR-Zone	246
12.4	Umwandlung von REAL-Zahlen	223	17.4	Peripheriebits setzen und rücksetzen.	248
12.5	Sonstige Umwandlungsfunktionen	225	18	Bausteinfunktionen	249
13	Schiebefunktionen	226	18.1	Bausteinfunktionen für Codebausteine.	249
13.1	Bearbeitung einer Schiebefunktion	226	18.1.1	Allgemeines zu Baustein aufrufen	250
13.2	Schieben	228	18.1.2	Aufruf-Box	251
13.3	Rotieren	229	18.1.3	CALL-Spule/Box	253
14	Wortverknüpfungen	230	18.1.4	Bausteinendefunktion.	254
14.1	Bearbeitung einer Wortverknüpfung	230	18.1.5	Temporäre Lokaldaten	254
14.2	Beschreibung der Wortverknüpfungen	232	18.1.6	Statische Lokaldaten	256
	Programmfluss-Steuerung	233	18.2	Bausteinfunktionen für Datenbausteine	259
15	Statusbits	234	18.2.1	Zwei Datenbausteinregister	259
15.1	Beschreibung der Statusbits	234	18.2.2	Zugriff auf Datenoperanden	260
15.2	Setzen der Statusbits	235	18.2.3	Datenbaustein aufschlagen	261
15.3	Auswertung der Statusbits	237	18.2.4	Besonderheiten bei der Datenadressierung.	262
15.4	Anwendung des Binärergebnisses	238	18.3	Systemfunktionen für Datenbausteine	263
15.4.1	Speichern des Binärergebnisses BIE.	238	18.3.1	Erzeugen eines Datenbausteins im Arbeitsspeicher	264
15.4.2	Hauptstrompfad, EN-/ENO-Mechanismus.	238	18.3.2	Erzeugen eines Datenbausteins im Ladespeicher.	264
15.4.3	ENO bei selbstgeschriebenen Bausteinen	239	18.3.3	Löschen eines Datenbausteins	266
			18.3.4	Testen eines Datenbausteins	266

19 Bausteinparameter	267	20.4 Kommunikation über dezentrale Peripherie	304
19.1 Bausteinparameter allgemein	267	20.4.1 PROFIBUS DP adressieren	304
19.1.1 Festlegung der Bausteinparameter	267	20.4.2 PROFIBUS DP projektieren	309
19.1.2 Bearbeitung der Bausteinparameter	268	20.4.3 Sonderfunktionen für PROFIBUS DP	318
19.1.3 Deklaration der Bausteinparameter	268	20.4.4 PROFINET IO adressieren	322
19.1.4 Deklaration des Funktionswerts	269	20.4.5 PROFINET IO projektieren	327
19.1.5 Versorgung von Bausteinparametern	269	20.4.6 Sonderfunktionen für PROFINET IO	333
19.2 Formalparameter	270	20.4.7 Systembausteine für die dezentrale Peripherie	342
19.3 Aktualparameter	272	20.5 Globaldatenkommunikation	351
19.4 „Weiterreichen“ von Bausteinparametern	275	20.5.1 Grundlagen	351
19.5 Beispiele	276	20.5.2 GD-Kommunikation projektieren	353
19.5.1 Beispiel Förderband	276	20.5.3 Systemfunktionen für GD-Kommunikation	355
19.5.2 Beispiel Stückgut-zähler	277	20.6 S7-Basiskommunikation	355
19.5.3 Beispiel Zuförderung	277	20.6.1 Stationsinterne S7-Basiskommunikation	355
Programmbearbeitung	285	20.6.2 Systemfunktionen für stationsinterne S7-Basiskommunikation	356
20 Hauptprogramm	286	20.6.3 Stationsexterne S7-Basiskommunikation	358
20.1 Programmgliederung	286	20.6.4 Systemfunktionen für stationsexterne S7-Basiskommunikation	359
20.1.1 Programmstruktur	286	20.7 S7-Kommunikation	361
20.1.2 Programmorganisation	287	20.7.1 Grundlagen	361
20.2 Zyklussteuerung	288	20.7.2 Zweiseitiger Datenaustausch	363
20.2.1 Prozessabbild-Aktualisierung	288	20.7.3 Einseitiger Datenaustausch	365
20.2.2 Zyklusüberwachungszeit	290	20.7.4 Druckdaten übertragen	365
20.2.3 Mindestzyklusdauer, Hintergrundbearbeitung	291	20.7.5 Steuerfunktionen	366
20.2.4 Reaktionszeit	292	20.7.6 Überwachungsfunktionen	368
20.2.5 Startinformation	293	20.8 IE-Kommunikation	371
20.3 Programmfunktionen	295	20.8.1 Grundlagen	371
20.3.1 Uhrzeit	295	20.8.2 Verbindungen auf- und abbauen	372
20.3.2 Systemzeit lesen	297	20.8.3 Datenübertragung mit TCP native oder ISO-on-TCP	374
20.3.3 Betriebsstundenzähler	297	20.8.4 Datenübertragung mit UDP	376
20.3.4 CPU-Speicher komprimieren	299	20.9 PtP-Kommunikation bei S7-300C	378
20.3.5 Warten und Stoppen	299	20.9.1 Grundlagen	378
20.3.6 Mehrprozessorbetrieb	299	20.9.2 ASCII-Treiber und Prozedur 3964(R)	379
20.3.7 OB-Programm-laufzeit ermitteln	300	20.9.3 Rechnerkopplung RK512	380
20.3.8 Programmschutz ändern	303		

20.10	Configuration in RUN	383	21.9	Alarmereignisse hantieren	405
20.10.1	Konfigurationsänderungen vorbereiten	385	21.9.1	Alarmer sperren und freigeben	405
20.10.2	Konfiguration ändern	385	21.9.2	Alarmer verzögern und freigeben	406
20.10.3	Konfiguration laden	386	21.9.3	Alarmzusatzinformation lesen	406
20.10.4	CiR-Synchronisationszeit	386	22	Anlaufverhalten	409
20.10.5	Auswirkungen auf die Programmbearbeitung	387	22.1	Allgemeines	409
20.10.6	CiR-Vorgang steuern	387	22.1.1	Betriebszustände	409
21	Alarmbearbeitung	388	22.1.2	Betriebszustand HALT	410
21.1	Allgemeines	388	22.1.3	Sperren der Ausgabebaugruppen	410
21.2	Uhrzeitalarme	389	22.1.4	Anlauf-Organisationsbausteine	410
21.2.1	Bearbeitung der Uhrzeitalarme	390	22.2	Einschalten	411
21.2.2	Uhrzeitalarme mit STEP 7 projektieren	391	22.2.1	Betriebszustand STOP	411
21.2.3	Systemfunktionen für Uhrzeitalarme	391	22.2.2	Urlöschen	412
21.3	Verzögerungsalarmer	393	22.2.3	Auslieferungszustand wieder- herstellen	412
21.3.1	Bearbeitung der Verzögerungs- alarmer	393	22.2.4	Remanenzverhalten	412
21.3.2	Verzögerungsalarmer mit STEP 7 projektieren	394	22.2.5	Anlaufparametrierung	413
21.3.3	Systemfunktionen für Verzöge- rungsalarmer	394	22.3	Anlaufarten	413
21.4	Weckalarmer	395	22.3.1	Betriebszustand ANLAUF	413
21.4.1	Bearbeitung der Weckalarmer	396	22.3.2	Kaltstart	414
21.4.2	Weckalarmer mit STEP 7 projektieren	397	22.3.3	Warmstart (Neustart)	414
21.5	Prozessalarmer	398	22.3.4	Wiederanlauf	416
21.5.1	Auslösung eines Prozessalarms	398	22.4	Baugruppenadresse ermitteln	417
21.5.2	Bearbeitung der Prozessalarmer	399	22.5	Baugruppen parametrieren	420
21.5.3	Prozessalarmer mit STEP 7 projektieren	399	22.5.1	Allgemeines zum Parametrieren von Baugruppen	420
21.6	DPV1-Alarmer	400	22.5.2	Systembausteine zur Baugruppen- parametrierung	421
21.7	Mehrprozessoralarm	400	22.5.3	Bausteine zur Datensatz- übertragung	424
21.8	Taktsynchronalarmer	403	23	Fehlerbehandlung	426
21.8.1	Bearbeitung der Taktsynchron- alarmer	403	23.1	Synchronfehler	426
21.8.2	Prozessabbild taktsynchron aktualisieren	404	23.2	Synchronfehlerereignisse hantieren	428
21.8.3	Taktsynchronalarmer mit STEP 7 projektieren	404	23.2.1	Fehlermasken	429
			23.2.2	Synchronfehlerereignisse maskieren	429
			23.2.3	Synchronfehlerereignisse demaskieren	429
			23.2.4	Ereignisstatusregister lesen	430
			23.2.5	Ersatzwert eintragen	430
			23.3	Asynchronfehler	431

23.4	Systemdiagnose	433	25.2	System Function Blocks	450
23.4.1	Diagnoseereignisse und Diagnosepuffer	433	25.3	IEC Function Blocks	453
23.4.2	Anwendereintrag in den Diagnosepuffer schreiben	434	25.4	S5-S7 Converting Blocks	454
23.4.3	Auswertung des Diagnosealarms	435	25.5	TI-S7 Converting Blocks	456
23.4.4	Systemzustandsliste lesen	435	25.6	PID Control Blocks	456
23.5	Webserver	437	25.7	Communication Blocks	456
23.5.1	Webserver aktivieren	437	25.8	Miscellaneous Blocks	457
23.5.2	Web-Informationen auslesen	438	25.9	SIMATIC_NET_CP	457
23.5.3	Web-Informationen	438	25.10	Redundant IO MGP V31	458
Anhang		440	25.11	Redundant IO CGP V40	459
24	Ergänzungen zur grafischen Programmierung	441	25.12	Redundant IO CGP V51	459
24.1	Bausteinschutz	441	26	Funktionsvorrat KOP	460
24.2	Indirekte Adressierung	442	26.1	Basisfunktionen	460
24.2.1	Zeiger allgemein	442	26.2	Digitalfunktionen	461
24.2.2	Bereichszeiger	442	26.3	Programmfluss-Steuerung	463
24.2.3	DB-Zeiger	444	27	Funktionsvorrat FUP	464
24.2.4	ANY-Zeiger	444	27.1	Basisfunktionen	464
24.2.5	„Variabler“ ANY-Zeiger	445	27.2	Digitalfunktionen	465
24.3	Kurzbeschreibung „Beispiel Telegramm“	445	27.3	Programmfluss-Steuerung	467
25	Baustein-Bibliotheken	449	Stichwortverzeichnis	468	
25.1	Organization Blocks	449	Abkürzungsverzeichnis	476	

Einführung

Dieser Teil des Buches stellt Ihnen das Automatisierungssystem SIMATIC S7-300/400 im Überblick vor.

Das **Automatisierungssystem S7-300/400** ist modular aus Baugruppen aufgebaut. Die Baugruppen können zentral (in der Nähe der CPU) oder dezentral vor Ort in der Anlage angeordnet sein, ohne dass Sie hierfür besondere Einstellungen und Parametrierungen benötigen. Die dezentrale Peripherie ist bei SIMATIC S7 integraler Bestandteil des Systems. Die Zentralbaugruppe mit ihren verschiedenen Speicherbereichen bildet die hardwaremäßige Grundlage für die Bearbeitung der Anwenderprogramme. Ein Ladespeicher enthält das komplette Anwenderprogramm; die ablaufrelevanten Programmteile befinden sich in einem Arbeitsspeicher, der mit kurzen Zugriffszeiten Voraussetzung für eine schnelle Programmbearbeitung ist.

STEP 7 ist die Programmiersoftware für S7-300/400, das Werkzeug zum Automatisieren ist der SIMATIC Manager. Er ist eine Applikation für die Windows-Betriebssysteme von Microsoft und enthält alle Funktionen zum Einrichten eines Projekts. Bei Bedarf startet der SIMATIC Manager weitere Werkzeuge, um z.B. Stationen zu konfigurieren, Baugruppen zu parametrieren, Programme zu erstellen und zu testen.

Mit den Programmiersprachen von STEP 7 formulieren Sie Ihre Automatisierungslösung. Das **SIMATIC S7-Programm** ist strukturiert aufgebaut, d.h. es besteht aus Bausteinen mit abgegrenzter Funktion, unterteilt nach Netzwerken und Anweisungen. Verschiedene Prioritätsklassen gestatten eine abgestufte Unterbrechbarkeit des gerade laufenden Anwenderprogramms. STEP 7 arbeitet mit Variablen unterschiedlicher Datentypen, angefangen von Binärvariablen (Datentyp BOOL) über Digitalvariablen (z.B. mit den Datentypen INT oder REAL für Rechenaufgaben) bis zu zusammengesetzten Datentypen wie Feldern oder Strukturen (Zu-

sammenfassung von Variablen verschiedener Datentypen zu einer einzigen Variablen).

Das erste Kapitel enthält einen Überblick über die Hardware des Automatisierungssystems S7-300/400, das zweite Kapitel den gleichen Überblick über die Programmiersoftware STEP 7. Grundlage der Beschreibung ist der Funktionsumfang für STEP 7 Version 5.5.

Das Kapitel 3 „SIMATIC S7-Programm“ führt Sie in die wesentlichen Elemente eines S7-Programms ein und zeigt die Programmierung einzelner Bausteine in den Programmiersprachen KOP und FUP. In den weiteren Kapiteln des Buchs werden dann ausführlich die Funktionen und Anweisungen von KOP und FUP beschrieben. Alle Beschreibungen sind durch kurze Darstellungsbeispiele erläutert.

- 1 Automatisierungssystem SIMATIC S7-300/400**
Aufbau des Automatisierungssystems; dezentrale Peripherie; Kommunikation; Baugruppenadressen; Operandenbereiche
- 2 Programmiersoftware STEP 7**
SIMATIC Manager; Projekt bearbeiten; Station konfigurieren; Netz projektieren; Programm erstellen (Symboltabelle, Programmeditor); Online schalten; Programm testen
- 3 SIMATIC S7-Programm**
Programmbearbeitung mit Prioritätsklassen; Programmaufbau aus Bausteinen; Bausteine mit KOP und FUP programmieren; Variablen adressieren; Konstanten; Datentypen (Übersicht)

1 Automatisierungssystem SIMATIC S7-300/400

1.1 Aufbau des Automatisierungssystems

1.1.1 Komponenten

Das Automatisierungssystem SIMATIC S7-300/400 ist eine modular aufgebaute speicherprogrammierbare Steuerung, die aus folgenden Komponenten besteht:

- ▷ Baugruppenträger (Racks); nehmen die Baugruppen auf und verbinden sie untereinander
- ▷ Stromversorgung (PS); liefert die internen Versorgungsspannungen
- ▷ Zentralbaugruppe (CPU); speichert und bearbeitet das Anwenderprogramm
- ▷ Anschaltungsbaugruppen (IM); verbinden die Baugruppenträger untereinander
- ▷ Signalbaugruppen (SM); passen die Signale der Anlage an den internen Signalpegel an oder steuern Stellgeräte über digitale und analoge Signale
- ▷ Funktionsbaugruppen (FM); bearbeiten komplexe oder zeitkritische Prozesse unabhängig von der Zentralbaugruppe
- ▷ Kommunikationsbaugruppen (CP) stellen den Anschluss zu Subnetzen her
- ▷ Subnetze verbinden die Automatisierungssysteme untereinander oder mit anderen Geräten

Ein Automatisierungssystem (eine Station) kann aus mehreren Baugruppenträgern bestehen, die untereinander über Buskabel verbunden sind. Im Zentralbaugruppenträger stecken die Stromversorgung, die CPU und Peripheriebaugruppen (SM, FM und CP). Reicht der Platz im Zentralbaugruppenträger für die Peripheriebaugruppen nicht aus oder möchte man räumlich entfernt

zum Zentralbaugruppenträger Peripheriebaugruppen anordnen, stehen Erweiterungsbaugruppenträger zur Verfügung, die über Anschaltungsbaugruppen die Verbindung zum Zentralbaugruppenträger herstellen (Bild 1.1). Zusätzlich besteht die Möglichkeit, dezentrale Peripherie an eine Station anzuschließen (siehe Kapitel 1.2.1 „PROFIBUS DP“).

Die Baugruppenträger verbinden die Baugruppen mit zwei Bussen: dem Peripheriebus (P-Bus) und dem Kommunikationsbus (K-Bus). Der P-Bus ist für den schnellen Signalaus-tausch von Ein- und Ausgabesignalen ausgelegt, der K-Bus für den Austausch größerer Datenmengen. Der K-Bus verbindet die CPU und die Programmiergeräteschnittstelle (MPI) mit Funktionsbaugruppen und Kommunikationsbaugruppen.

1.1.2 S7-300-Station

Zentraler Aufbau

Bei S7-300 können bis zu 8 Peripheriebaugruppen im Zentralbaugruppenträger gesteckt werden. Reicht dieser einzeilige Aufbau nicht aus, können Sie ab CPU 313

- ▷ entweder einen zweizeiligen Aufbau wählen (mit IM 365 bis zu 1 m)
- ▷ oder einen bis zu vierzeiligen Aufbau wählen (mit IM 360 und IM 361 bis zu 10 m zwischen den Baugruppenträgern).

Sie können maximal 8 Baugruppen in einem Baugruppenträger betreiben. Die Anzahl der Baugruppen wird durch den maximal zulässigen Strom pro Baugruppenträger von 1,2 A begrenzt.

Die Baugruppen werden durch einen seriellen Rückwandbus miteinander verbunden, der die Funktionen des P-Busses und des K-Busses vereint.

Modularer Aufbau einer S7-300-Station



Modularer Aufbau einer S7-400-Station

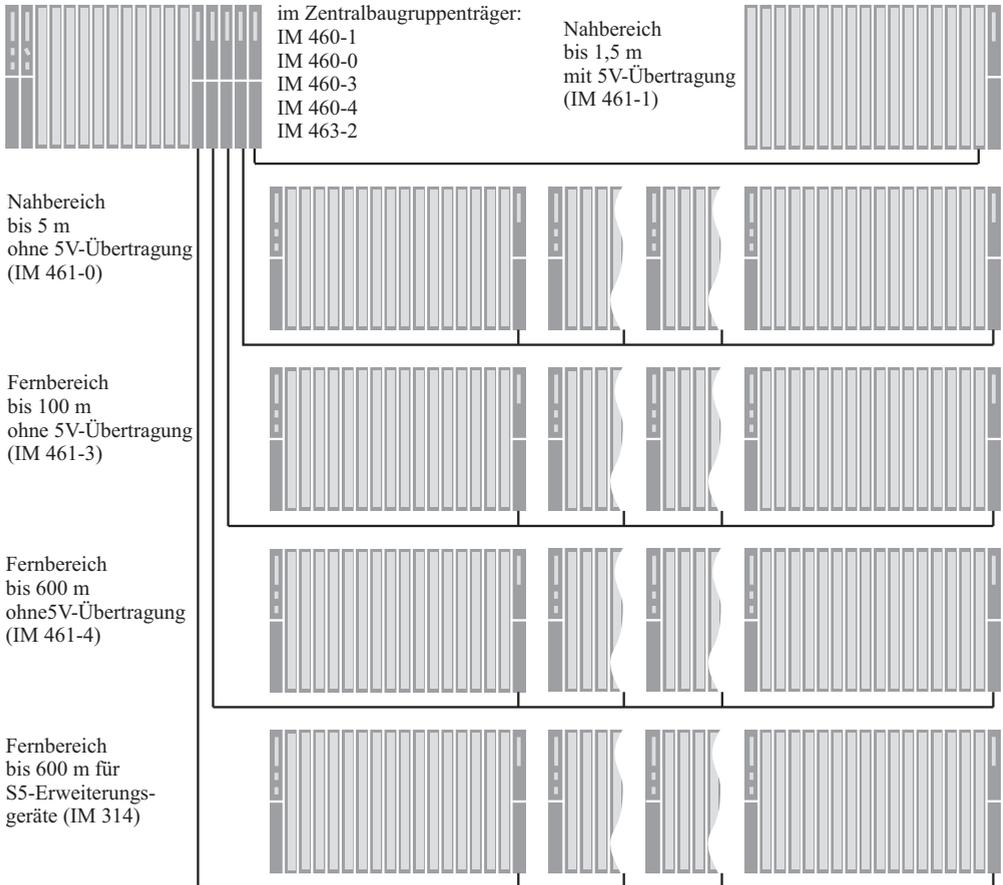


Bild 1.1 Hardware-Aufbau S7-300/400

Lokalbusegment

Eine Besonderheit bezüglich der Baugruppenanordnung stellt der Einsatz der Applikationsbaugruppe FM 356 dar. Eine FM 356 ist in der Lage, den Rückwandbus eines Baugruppenträgers „aufzutrennen“ und die Ansteuerung der restlichen Baugruppen im abgetrennten „Lokalbus-Segment“ selbst zu übernehmen. Für die Baugruppenanzahl und die Stromaufnahme gelten auch in diesem Fall die oben genannten Begrenzungen.

Standard-CPUs

Die Standard-CPUs gibt es in verschiedenen Ausführungen bezüglich Speicherumfang und Verarbeitungsgeschwindigkeit. Sie reichen von der „kleinsten“ CPU 312 für kleinere Anwendungen mit moderaten Anforderungen an die Verarbeitungsgeschwindigkeit bis zur CPU 319-3 PN/DP mit großem Programmspeicher und hoher Programmbearbeitungsleistung für branchenübergreifende Automatisierungsaufgaben. Mit den entsprechenden Schnittstellen ausgestattet können einige CPUs als Zentrale für die dezentrale Peripherie über PROFIBUS und PROFINET eingesetzt werden.

Zum Betrieb der Standard-CPUs ist – wie bei allen innovierten S7-300-CPU – eine Micro Memory Card (MMC) erforderlich. Dieses Speichermedium eröffnet neue Anwendungsmöglichkeiten gegenüber der bisher eingesetzten Memory Card (siehe Kapitel 1.1.6 „Speicherbereiche der Zentralbaugruppe“).

Die mittlerweile abgekündigte CPU 318 kann durch die CPUs 317 und 319 ersetzt werden.

Kompakt-CPUs

Die CPUs 3xxC gestatten einen kompakten Aufbau von Kleinststeuerungen. Je nach Ausführung enthalten sie bereits

- ▷ integrierte Peripherie
Digital- und Analog-Ein/Ausgänge
- ▷ integrierte technologische Funktionen
Zählen, Messen, Regeln, Positionieren
- ▷ integrierte Kommunikationsschnittstellen
PROFIBUS-DP-Master oder -Slave, Punkt-zu-Punkt-Kopplung (PtP)

Die technologischen Funktionen sind Systembausteine, die die On-Board-Peripherie der Zentralbaugruppe verwenden.

Technologie-CPUs

Die CPUs 3xxT vereinen Steuerungsfunktionen mit einfachen Motion-Control-Funktionen. Der Steuerungsanteil ist wie bei einer Standard-CPU ausgelegt. Er wird mit STEP 7 projektiert, parametrisiert und programmiert. Die Technologieobjekte und der Motion-Control-Teil benötigen das Optionspaket S7-Technology, das nach der Installation im SIMATIC Manager integriert ist.

Die Technologie-CPUs besitzen eine PROFIBUS-DP-Schnittstelle, die den Betrieb als DP-Master oder DP-Slave gestattet. Die CPUs werden für branchenübergreifende Automatisierungsaufgaben im Serienmaschinen-, Sondermaschinen- und Anlagenbau eingesetzt.

Fehlersichere CPUs

Die CPUs 3xxF werden in Fertigungsanlagen mit erhöhten Sicherheitsanforderungen eingesetzt. Entsprechende PROFIBUS- und PROFINET-Schnittstellen gestatten das Betreiben von sicherheitsgerichteter dezentraler Peripherie über das Busprofil PROFIsafe (siehe „Safety Integrated für die Fertigungsindustrie“ unter 1.1.5 „Sicherheitsgerichtete SIMATIC“). Parallel zum sicherheitsgerichteten Betrieb können Standardbaugruppen für normale Anwendungen eingesetzt werden.

SIPLUS

Die Produktfamilie SIPLUS bietet Baugruppen, die in rauer Umgebung eingesetzt werden können. Die Basis der SIPLUS-Komponenten sind Standardgeräte, die speziell für den jeweiligen Einsatz umgerüstet werden, beispielsweise für einen erweiterten Temperaturbereich, erhöhte Rüttel- und Stoßfestigkeit oder verschiedene, vom Standard abweichende Spannungsbereiche. Beachten Sie deshalb bitte die technischen Daten der jeweiligen SIPLUS-Baugruppe. Zur Projektierung mit STEP 7 verwenden Sie den Äquivalenztyp (die zugrunde liegende Standardbaugruppe), der beispielsweise auf dem Typenschild der Baugruppe angegeben ist.

1.1.3 S7-400-Station

Zentraler Aufbau

Die Zentralbaugruppenträger bei S7-400 gibt es in den Ausführungen UR1 (18 Steckplätze), UR2 (9 Steckplätze) und CR3 (4 Steckplätze). UR1 und UR2 können auch als Erweiterungsbaugruppenträger eingesetzt werden. In den Baugruppenträgern belegen auch die Stromversorgung und die CPU Steckplätze, evtl. sogar zwei oder mehr pro Baugruppe. Bei Bedarf wird mit Erweiterungsbaugruppenträgern die verfügbare Anzahl an Steckplätzen erhöht: UR1 und ER1 haben je 18 Steckplätze, UR2 und ER2 je 9.

Mit den Anschaltungen IM 460-1 und IM 461-1 kann pro Schnittstelle ein Erweiterungsbaugruppenträger bis 1,5 m entfernt vom Zentralbaugruppenträger angeordnet werden, hierbei wird die 5V-Versorgungsspannung mit übertragen. Ebenfalls im Nahbereich bis 5 m können bis zu 4 Erweiterungsbaugruppenträger über IM 460-0 und 461-0 betrieben werden. Im Fernbereich schließlich ist es mit IM 460-3 und IM 461-3 bzw. IM 460-4 und 461-4 möglich, bis zu 4 Erweiterungsbaugruppenträger bis zu 100 m bzw. 600 m entfernt zu betreiben.

Maximal sind bis zu 21 Erweiterungsbaugruppenträger an einen Zentralbaugruppenträger anschließbar. Zur Unterscheidung stellen Sie die Nummer des Baugruppenträgers am Kodierschalter der Empfangs-IM ein.

Der Rückwandbus besteht aus einem parallelen P-Bus und einem seriellen K-Bus. Die Erweiterungsbaugruppenträger ER1 und ER2 sind für „einfache“ Signalbaugruppen ausgelegt, die keine Prozessalarmlen auslösen, nicht mit 24 V über den P-Bus versorgt werden müssen, keine Pufferspannung benötigen und keinen K-Bus-Anschluss haben. Der K-Bus ist in den Baugruppenträgern UR1, UR2 und CR2 dann vorhanden, wenn sie entweder als Zentralbaugruppenträger oder als Erweiterungsbaugruppenträger mit den Nummern 1 bis 6 verwendet werden.

Segmentierter Baugruppenträger

Eine Besonderheit stellt der segmentierte Baugruppenträger CR2 dar. Sie können hier zwei CPUs in einem Baugruppenträger mit gemein-

samer Stromversorgung funktionsmäßig getrennt betreiben. Beide CPUs können über den K-Bus miteinander Daten austauschen, haben jedoch vollständig getrennte P-Busse für ihre eigenen Signalbaugruppen.

Mehrprozessorbetrieb

Bei S7-400 können sich in einem Zentralbaugruppenträger UR bis zu 4 entsprechend ausgelegte CPUs am Mehrprozessorbetrieb beteiligen. Jede Baugruppe in dieser Station wird einer einzigen CPU zugeordnet, sowohl mit ihrer Adresse als auch mit ihren Alarmlen. Näheres siehe Kapitel 20.3.6 „Mehrprozessorbetrieb“ und 21.7 „Mehrprozessoralarm“.

Anschluss von SIMATIC S5-Baugruppen

Mit der Anschaltungsbaugruppe IM 463-2 können Sie S5-Erweiterungsgeräte (EG 183U, EG 185U, EG 186U sowie ER 701-2 und ER 701-3) an eine S7-400 anschließen und diese Erweiterungsgeräte wiederum zentral erweitern. Im S5-Erweiterungsgerät übernimmt eine IM 314 die Kopplung. Sie können alle in den angegebenen Erweiterungsgeräten zugelassenen Digital- und Analogbaugruppen betreiben. Eine S7-400 kann maximal 4 IM 463-2 aufnehmen; an jeder der beiden Schnittstellen einer IM 463-2 ist der Anschluss von maximal vier S5-Erweiterungsgeräten möglich.

1.1.4 Hochverfügbare SIMATIC

Für Anwendungen mit hohen Anforderungen an die Ausfallsicherheit von Maschinen und Prozessen gibt es bei SIMATIC S7 hochverfügbare Automatisierungssysteme mit redundant ausgelegtem Ausbau in zwei Ausführungen: Software-Redundanz und S7-400H/FH.

Software-Redundanz

Mit SIMATIC S7-300/400-Standard-Komponenten können Sie ein auf Softwarebasis redundantes System aufbauen, in dem eine Masterstation den Prozess steuert und, bei deren Ausfall, eine Reservestation die Steuerung übernimmt.

Hochverfügbarkeit durch Software-Redundanz eignet sich für langsame Prozesse, denn die

Umschaltung auf die Reservestation kann je nach Ausbau der Automatisierungsgeräte mehrere Sekunden benötigen. Während dieser Zeit sind die Prozesssignale „eingefroren“. Danach arbeitet die Reservestation mit den zuletzt in der Masterstation gültigen Daten weiter.

Die Redundanz der Ein-/Ausgabebaugruppen wird mit dezentraler Peripherie realisiert (ET 200M mit Anschaltung IM 153-2 für redundant aufgebautem PROFIBUS DP). Die Software Redundanz kann mit STEP 7 ab Version 5.2 projektiert werden.

Hochverfügbare SIMATIC S7-400H

SIMATIC S7-400H ist ein hochverfügbares Automatisierungssystem mit redundant ausgelegtem Aufbau aus zwei Zentralbaugruppenträgern mit je einer H-CPU und einem Synchronisationsmodul zum Datenabgleich über Lichtwellenleiter. Beide Geräte arbeiten „hot standby“; im Fehlerfall übernimmt das intakte Gerät durch automatische stoßfreie Umschaltung die alleinige Bearbeitung. Der Baugruppenträger UR2-H mit zweimal neun Steckplätzen bietet die Möglichkeit, ein hochverfügbares System auch in einem einzigen Baugruppenträger aufzubauen.

Die Peripherie kann entweder normal verfügbar (einkanaliger einseitiger Aufbau) oder erhöht verfügbar (einkanaliger geschalteter Aufbau mit ET 200M) aufgebaut sein. Die Kommunikation erfolgt mit einfachem oder mit redundantem Bus.

Das Anwenderprogramm ist das gleiche wie für ein nicht redundantes Gerät; die Redundanzfunktion wird ausschließlich und vom Anwender verborgen durch die eingesetzte Hardware erbracht. Das für die Projektierung erforderliche Softwarepaket ist in STEP 7 ab V5.3 enthalten. Die mitgelieferten Standardbibliotheken *Redundant IO* enthalten Bausteine zur Unterstützung der redundanten Peripherie.

1.1.5 Sicherheitsgerichtete SIMATIC

Fehlersichere Automatisierungssysteme steuern Prozesse, in denen der sichere Zustand durch unmittelbare Abschaltung erreicht werden kann. Sie werden in Anlagen mit erhöhten Sicherheitsanforderungen eingesetzt.

Die Sicherheitsfunktionen liegen schwerpunktmäßig im sicherheitsgerichteten Anwenderprogramm einer entsprechend ausgelegten CPU und in den fehlersicheren Ein- und Ausgabe-Baugruppen. Eine F-CPU erfüllt die Sicherheitsanforderungen bis AK 6 nach DIN V 19250/DIN V VDE 0801, bis SIL 3 nach IEC 61508 und bis Kategorie 4 nach EN 954-1. Die Sicherheitsfunktionen können in der gleichen CPU parallel zu einem nicht sicherheitsgerichtet ausgelegten Anwenderprogramm betrieben werden.

Die sicherheitsgerichtete Kommunikation über PROFIBUS DP – bei S7 Distributed Safety auch über PROFINET IO – verwendet das Busprofil PROFIsafe. Es erlaubt die Übertragung von sicherheitsgerichteten und nicht sicherheitsgerichteten Daten auf einer einzigen Busleitung.

Safety Integrated für die Fertigungsindustrie

S7 Distributed Safety ist ein fehlersicheres Automatisierungssystem mit dem Einsatzschwerpunkt im Bereich Maschinen- und Personalschutz für Maschinensteuerungen und in der Prozessindustrie.

Als F-CPU stehen Controller aus dem Gerätespektrum von SIMATIC S7-300, S7-400 und ET 200S zur Verfügung. An S7-400 werden die sicherheitsgerichteten E/A-Baugruppen über PROFIBUS DP oder PROFINET IO mit dem sicherheitsgerichteten Busprofil PROFIsafe angeschlossen. Bei S7-300 und ET 200S ist der Betrieb von sicherheitsgerichteten E/A-Baugruppen auch im Zentralbaugruppenträger möglich.

Die Hardware-Projektierung und die Programmierung des nicht sicherheitsgerichteten Anwenderprogramms erfolgt mit den Standard-Applikationen von STEP 7.

Zur Programmierung der sicherheitsgerichteten Programmteile ist das Optionspaket *SIMATIC S7 Distributed Safety* erforderlich. Mit diesem Optionspaket erstellen Sie mit den Programmiersprachen F-KOP oder F-FUP die Bausteine, die das sicherheitsgerichtete Programm enthalten. Die Anbindung an die Peripherie erfolgt wie beim Standardprogramm über das Prozessabbild. S7 Distributed Safety enthält auch eine Bibliothek mit TÜV-zertifizierten Sicherheitsbausteinen. Zusätzlich gibt es eine Bibliothek mit F-Bausteinen für Pressen- und Brenner-Steuerungen.

Das sicherheitsgerichtete Anwenderprogramm kann parallel zum Standard-Anwenderprogramm laufen. Wird im sicherheitsgerichteten Teil des Programms ein Fehler entdeckt, geht die CPU in den Betriebszustand STOP.

Safety Integrated für die Prozessindustrie

S7 F/FH Systems ist ein fehlersicheres Automatisierungssystem auf der Basis von S7-400 mit dem Einsatzschwerpunkt in der Prozessindustrie. Die sicherheitsgerichteten E/A-Baugruppen werden über PROFIBUS DP mit dem sicherheitsgerichteten Busprofil PROFIsafe angeschlossen.

Eine S7-400-F-CPU erhält die sicherheitsgerichteten Steuerungsfunktionen durch den Einsatz einer *S7 F Systems Runtime-Lizenz*. Parallel zum sicherheitsgerichteten Anlagenteil kann ein nicht sicherheitsgerichtetes Anwenderprogramm laufen.

Zusätzlich zur Fehlersicherheit bietet S7-400FH eine erhöhte Verfügbarkeit. Führt ein erkannter Fehler zum STOP der Master-CPU, wird auf die im Hot-Stand-By laufende Reserve-CPU rückwirkungsfrei umgeschaltet. Für den Betrieb als S7-400FH ist zusätzlich das Optionspaket *S7 H Systems* erforderlich.

Die Hardware-Projektierung und die Programmierung des nicht sicherheitsgerichteten Anwenderprogramms erfolgt mit den Standard-Applikationen von STEP 7.

Für die Programmierung der sicherheitsgerichteten Programmteile ist das Optionspaket *S7 F Systems* erforderlich, zusätzlich das Optionspaket *CFC* ab V5.0 SP3 und das Optionspaket *S7-SCL* ab V5.0.

Die Programmierung des sicherheitsgerichteten Programms geschieht mit CFC (Continuous Function Chart, funktionsplanähnliche Verschaltung von Bausteinen). Hierbei werden sicherheitsgerichtet programmierte Funktionsbausteine aus der mitgelieferten F-Bibliothek aufgerufen und verschaltet. Sie enthalten neben Funktionen zur Programmierung von Sicherheitsfunktionen auch Funktionen zur Fehlererkennung und Fehlerreaktion. So wird sichergestellt, dass bei Ausfällen und Fehlern das F-System in einem sicheren Zustand gehalten wird oder in einen sicheren Zustand überführt

wird. Wird im Sicherheitsprogramm ein Fehler entdeckt, schaltet sich der sicherheitsgerichtete Teil der Anlage ab, während der restliche Teil noch weiterlaufen kann.

Fehlersichere Peripherie

Für den Sicherheitsbetrieb sind fehlersichere Signalbaugruppen (F-Baugruppen bzw. F-Module) erforderlich. Die Fehlersicherheit wird durch die integrierten Sicherheitsfunktionen und eine entsprechende Verdrahtung der Sensoren und Aktoren erreicht.

Die F-Baugruppen sind auch im Standardbetrieb mit erhöhten Anforderungen an die Diagnose einsetzbar. Sowohl im Standard- als auch im Sicherheitsbetrieb können bei S7 F/FH Systems die F-Baugruppen zur Erhöhung der Verfügbarkeit redundant betrieben werden.

Die fehlersichere Peripherie gibt es in verschiedenen Ausführungen:

- ▷ Die fehlersicheren Signalbaugruppen in der S7-300-Bauform werden im dezentralen Peripheriegerät ET 200M oder – bei S7-Distributed Safety – auch zentral eingesetzt.
- ▷ Fehlersichere E/A-Module gibt es für die dezentralen Peripheriegeräte mit den Bauformen ET 200S, ET 200pro und ET 200eco.
- ▷ Für die dezentralen Peripheriegeräte ET 200S und ET 200pro sind auch fehlersichere Interface-Module als F-CPU erhältlich.
- ▷ Es können fehlersichere DP-Norm-Slaves und – bei S7-Distributed Safety auch IO-Norm-Devices –, die das Busprofil PROFIsafe beherrschen, eingesetzt werden.

Fehlersichere CPUs und Signalbaugruppen gibt es auch in SIPLUS-Ausführung.

1.1.6 Speicherbereiche der Zentralbaugruppe

Das Bild 1.2 zeigt die für Ihr Programm wichtigen Speicherbereiche im Programmiergerät, in der Zentralbaugruppe und in den Signalbaugruppen.

Das Programmiergerät enthält die *Offline-Daten*. Sie bestehen aus dem Anwenderprogramm (Programmcode und Anwenderdaten), aus den

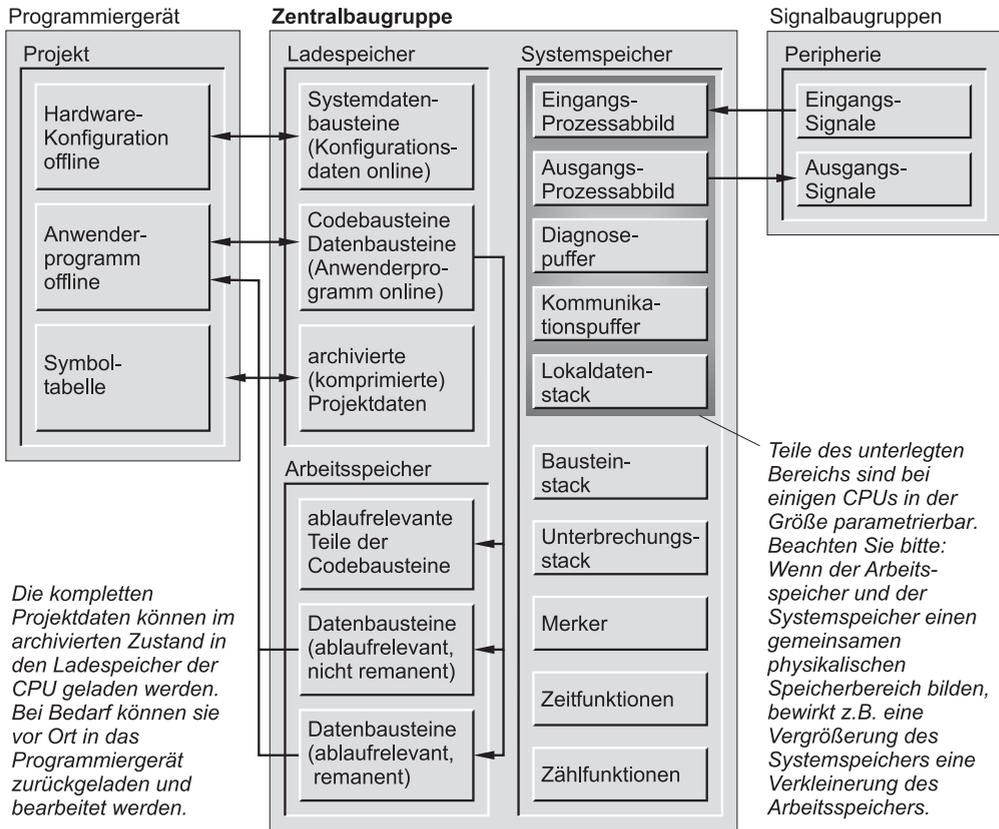


Bild 1.2 Speicherbereiche auf der Zentralbaugruppe

Systemdaten (z.B. Hardware-Projektierung, Netz- und Verbindungsprojektierung) und weiteren projektspezifischen Daten wie z.B. die Symboltabelle und die Kommentare.

Die *Online-Daten* bestehen aus dem Anwenderprogramm und den Systemdaten auf der Zentralbaugruppe, die in zwei Speicherbereichen, dem Ladespeicher und dem Arbeitsspeicher, untergebracht sind. Zusätzlich ist hier noch der Systempeicher vorhanden.

Die Peripheriebaugruppen schließlich enthalten Speicher für die Signalzustände der Ein- und Ausgänge.

Die Zentralbaugruppen haben einen Schacht für ein steckbares *Speichermodul*. Auf diesem Speichermodul befinden sich der Ladespeicher oder Teile davon (siehe „Physikalische Ausführ-

ung des CPU-Speichers“, weiter unten). Das Speichermodul ist als Memory Card (S7-400-CPU) oder als Micro Memory Card (S7-300-CPU) und davon abgeleitete ET200-CPU) ausgeführt. Über das Speichermodul kann auch ein Firmware-Update des CPU-Betriebssystems erfolgen.

Memory Card

Das Speichermodul für die S7-400-CPU ist die Memory Card (MC). Es gibt zwei Arten: RAM Memory Cards und Flash EPROM Memory Cards.

Wenn Sie ausschließlich den Ladespeicher erweitern wollen, verwenden Sie eine RAM Memory Card. Mit einer RAM Memory Card können Sie das komplette Anwenderprogramm

online ändern. Das ist beispielsweise für größere Programme beim Test und bei der Inbetriebnahme erforderlich. RAM Memory Cards verlieren ihren Inhalt, wenn sie gezogen werden.

Wenn Sie nach Test und Inbetriebnahme das Anwenderprogramm einschließlich der Konfigurationsdaten und Baugruppenparameter auch ohne Pufferbatterie spannungsausfallsicher halten wollen, verwenden Sie eine Flash EPROM Memory Card. Hierbei laden Sie das gesamte Programm offline auf die Flash EPROM Memory Card, wenn sie im Programmiergerät steckt. Bei entsprechend ausgelegten CPUs können Sie das Programm auch online laden, wenn die Memory Card in der CPU steckt.

Micro Memory Card

Das Speichermodul bei den neueren S7-300-CPU's ist eine Micro Memory Card (MMC). Die Daten auf der MMC sind nullspannungsfest abgelegt, können jedoch wie bei einem RAM-Speicher gelesen, geschrieben und gelöscht werden. Dieses Verhalten erlaubt eine Datenpufferung ohne Batterie.

Auf der MMC befindet sich der komplette Ladespeicher, so dass zum Betrieb immer eine MMC erforderlich ist. Die MMC kann als portables Speichermedium für Anwenderprogramme oder Firmware-Updates genutzt werden. Mit speziellen Systemfunktionen lesen oder schreiben Sie vom Anwenderprogramm aus Datenbausteine auf der MMC und können so z.B. Rezepturen von der MMC lesen oder ein Messwertarchiv auf der MMC anlegen und mit Daten versorgen.

Ladespeicher

Im Ladespeicher steht das gesamte Anwenderprogramm einschließlich Konfigurationsdaten (Systemdaten). Vom Programmiergerät aus wird das Anwenderprogramm immer zuerst in den Ladespeicher übertragen und von hier aus in den Arbeitsspeicher. Das Programm im Ladespeicher wird nicht als Steuerungsprogramm bearbeitet.

Bei einer CPU 300 und einer CPU ET 200 liegt der Ladespeicher komplett auf der Micro Memory Card. Somit bleibt der Inhalt des Lade-

speichers auch im spannungslosen Zustand der CPU erhalten.

Besteht bei einer CPU 400 der Ladespeicher aus integriertem RAM oder einer RAM Memory Card, ist eine Pufferbatterie erforderlich, um das Anwenderprogramm remanent zu halten. Bei integriertem EEPROM oder bei zusteckbarer Flash EPROM Memory Card als Ladespeicher kann die CPU batterieelos betrieben werden.

Ab STEP 7 V5.1 können Sie bei entsprechend ausgelegten CPUs die gesamten Projektdaten als komprimierte Archivdatei im Ladespeicher ablegen (siehe Kapitel 2.2.2 „Verwalten, reorganisieren und archivieren“).

Arbeitsspeicher

Der Arbeitsspeicher ist als schneller, in vollem Umfang in der CPU integrierter RAM-Speicher ausgelegt. Das Betriebssystem der CPU kopiert den „ablaufrelevanten“ Programmcode und die Anwenderdaten in den Arbeitsspeicher. „Ablaufrelevant“ ist eine Eigenschaft der vorhandenen Objekte, nicht gleichbedeutend mit der Tatsache, dass ein bestimmter Codebaustein auch aufgerufen und bearbeitet wird. Das „eigentliche“ Steuerungsprogramm wird im Arbeitsspeicher bearbeitet.

Produktspezifisch kann der Arbeitsspeicher entweder als ein zusammenhängender Bereich oder unterteilt nach Programm- und Datenspeicher und letzterer auch unterteilt nach remanent und nicht remanent ausgebildet sein.

Beim Zurückladen des Anwenderprogramms in das Programmiergerät werden die Bausteine aus dem Ladespeicher geholt, ergänzt um die aktuellen Werte der Datenoperanden aus dem Arbeitsspeicher (weitere Informationen hierzu in den Kapiteln 2.6.4 „Anwenderprogramm in die CPU laden“ und 2.6.5 „Bausteinhandlung“).

Systemspeicher

Der Systemspeicher enthält die Operanden (Variablen), die Sie von Ihrem Programm aus ansprechen. Die Operanden sind zu Bereichen (Operandenbereiche) zusammengefasst, die eine für jede CPU spezifische Menge an Operan-

den enthalten. Operanden sind z.B. Eingänge, mit denen Sie die Signalzustände von Tastern und Endschaltern abfragen, und Ausgänge, mit denen Sie Schütze und Lampen steuern.

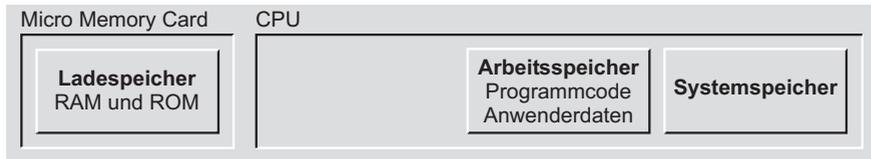
Folgende Operandenbereiche liegen im Systempeicher der CPU:

- ▷ Eingänge (E)
Sie sind ein Abbild („Prozessabbild“) der Digitaleingabebaugruppen.
- ▷ Ausgänge (A)
Sie sind ein Abbild („Prozessabbild“) der Digitalausgabebaugruppen.
- ▷ Merker (M)
Sie sind Informationsspeicher, die im Anwenderprogramm von jeder Stelle aus direkt ansprechbar sind.

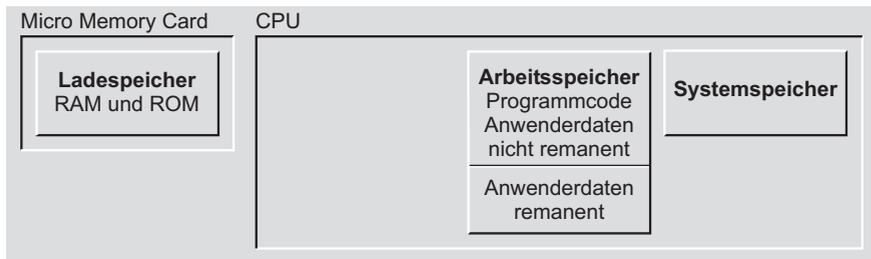
- ▷ Zeitfunktionen (T)
Sie stellen Zeitglieder dar, mit denen Warte- und Überwachungszeiten realisiert werden.
- ▷ Zählerfunktionen (Z)
Sie sind Softwarezähler, die vorwärts und rückwärts zählen können.
- ▷ temporäre Lokaldaten (L)
Sie dienen als dynamische Zwischenspeicher während der Bausteinbearbeitung. Die temporären Lokaldaten stehen im L-Stack, den die CPU während der Programmbearbeitung dynamisch belegt.

In Klammern steht das Kurzzeichen, mit dem Sie die einzelnen Operanden beim Programmieren ansprechen können. Sie können jeder Variablen auch ein Symbol zuordnen und dann mit der symbolischen Bezeichnung anstelle der Operandenbezeichnung arbeiten.

S7-300- und ET-CPU's ohne einstellbare Daten-Remanenz



S7-300- und ET-CPU's mit einstellbarer Daten-Remanenz



S7-400-CPU

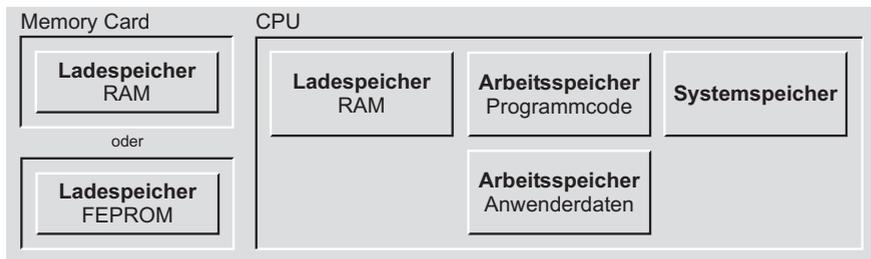


Bild 1.3 Physikalische Ausführung des CPU-Speichers

Zusätzlich enthält der Systemspeicher Puffer für Kommunikationsaufträge und Systemmeldungen (Diagnosepuffer). Die Größe dieser Datenpuffer sowie die Größe des Eingangs-Prozessabbaus, des Ausgangs-Prozessabbaus und des L-Stacks sind bei einigen CPUs parametrierbar.

Physikalische Ausführung des CPU-Speichers

Die physikalische Ausführung des Ladespeichers unterscheidet sich bei den verschiedenen CPU-Typen (Bild 1.3).

Eine CPU 300 oder eine CPU ET 200 besitzt keinen integrierten Ladespeicher. Zum Betrieb muss immer eine Micro Memory Card stecken, auf der der Ladespeicher liegt. Der Ladespeicher kann wie ein RAM-Speicher beschrieben und gelesen werden. Physikalisch bedingt ist die Anzahl der Schreibzugriffe beschränkt (kein zyklisches Schreiben durch das Anwenderprogramm). Mit dem Menübefehl RAM NACH ROM KOPIEREN können die aktuellen Werte der Datenoperanden aus dem Arbeitsspeicher in den Ladespeicher übertragen werden.

Bei einer CPU 300 ab Firmwareversion V2.0.12 besteht der Arbeitsspeicher für die Anwenderdaten aus einem remanenten und einen nicht remanenten Teil. Im nicht remanenten Teil liegt auch das Steuerungsprogramm.

Der integrierte RAM-Ladespeicher bei einer CPU 400 ist für kleine Programme bzw. für das Ändern von einzelnen Bausteinen ausgelegt, wenn der Ladespeicher eine Flash EPROM Memory Card ist. Ist das komplette Steuerungsprogramm größer als der integrierte Ladespeicher, benötigen Sie zum Testen eine RAM Memory Card. Das getestete Programm wird dann mit dem Programmiergerät auf eine Flash EPROM Memory Card übertragen, die Sie zum Betrieb in die CPU stecken.

Der Arbeitsspeicher einer CPU 400 ist zweigeteilt: Ein Bereich nimmt den Programmcode auf, der andere Teil die Anwenderdaten. System- und Arbeitsspeicher bilden bei einer CPU 400 eine (physikalische) Einheit. Wird z.B. das Prozessabbild in der Größe verändert, hat dies Auswirkungen auf die Größe des Arbeitsspeichers.

1.2 Dezentrale Peripherie

Unter dezentraler Peripherie versteht man Baugruppen, die über PROFIBUS DP oder PROFINET IO angeschlossen sind. PROFIBUS DP verwendet das Subnetz PROFIBUS für die Datenübertragung, PROFINET IO das Subnetz Industrial Ethernet (weitere Informationen siehe Kapitel 1.3.2 „Subnetze“).

1.2.1 PROFIBUS DP

PROFIBUS DP bietet eine standardisierte Schnittstelle für die Übertragung von überwiegend binären Prozessdaten zwischen einer „Anschaltungsbaugruppe“ im (zentralen) Automatisierungsgerät und den Feldgeräten. Diese „Anschaltungsbaugruppe“ nennt man DP-Master und die Feldgeräte DP-Slaves.

Der DP-Master und alle von ihm gesteuerten DP-Slaves bilden ein DP-Mastersystem. Maximal können 32 Stationen in einem Segment und bis zu 127 im gesamten Netz vorhanden sein. Ein DP-Master kann eine ihm spezifische Anzahl DP-Slaves steuern. Sie können auch Programmiergeräte an das PROFIBUS-DP-Netz anschließen, ebenso wie z.B. Geräte zum Bedienen und Beobachten, ET 200-Geräte oder SIMATIC-S5-DP-Slaves.

DP-Mastersystem

PROFIBUS DP wird üblicherweise als „Mono-Mastersystem“ betrieben, d.h. ein DP-Master steuert mehrere DP-Slaves. Der DP-Master ist, abgesehen von einem temporär vorhandenen Programmiergerät (Diagnose- und Servicegerät), der einzige Master am Bus. Der DP-Master und die ihm zugeordneten DP-Slaves bilden ein DP-Mastersystem (Bild 1.4).

Sie können auf einem PROFIBUS-Subnetz auch mehrere DP-Mastersysteme installieren (Multi-Mastersystem). Dann jedoch vergrößert sich die Reaktionszeit im Einzelfall, denn wenn ein DP-Master „seine“ DP-Slaves versorgt hat, bekommt die Zugriffsrechte der nächste DP-Master, der nun wiederum „seine“ DP-Slaves versorgt, usw.

Sie können die Reaktionszeit verringern, wenn nur wenige DP-Slaves in einem DP-Mastersystem vorhanden sind. Da es möglich ist, in einer