

**SIEMENS**



Hans Berger

# Automatisieren mit SIMATIC S7-1200

Projektieren, Programmieren und Testen  
mit STEP 7 Professional

4. Auflage

Berger Automatisieren mit SIMATIC S7-1200



# Automatisieren mit SIMATIC S7-1200

Programmieren, Projektieren und  
Testen mit STEP 7

von Hans Berger

4. wesentlich überarbeitete und erweiterte  
Auflage, 2017

Publicis Publishing

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

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

[www.publicis-books.de](http://www.publicis-books.de)

Lektorat: Dr. Gerhard Seifudem, [gerhard.seifudem@publicispixelpark.de](mailto:gerhard.seifudem@publicispixelpark.de)

**Print ISBN 978-3-89578-469-9**

**ePDF ISBN 978-3-89578-960-1**

4., wesentlich überarbeitete und erweiterte Auflage, 2017

Herausgeber Siemens Aktiengesellschaft, Berlin und München

Verlag: Publicis Publishing, Erlangen

© 2017 by Publicis Pixelpark Erlangen – eine Zweigniederlassung der Publicis Pixelpark GmbH

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

Printed in Germany

# Vorwort

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

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

Das vorliegende Buch beschreibt das Automatisierungssystem SIMATIC S7-1200. Eine S7-1200-Steuerung ist kompakt aufgebaut und modular erweiterbar. Für viele Kleinanwendungen genügt bereits der Einsatz der CPU-Baugruppe mit der Onboard-Peripherie. Die in der CPU-Baugruppe integrierten technologischen Funktionen machen das Gerät äußerst vielseitig einsetzbar. Mit Kontaktplan (KOP), Funktionsplan (FUP) und Structured Control Language (SCL) stehen drei etablierte Programmiersprachen für die Lösung der Automatisierungsaufgabe zur Verfügung.

Passend zu den S7-1200-Steuerungen sind mit den SIMATIC HMI Basic Panels neue Bedien- und Beobachtungsgeräte konzipiert worden mit einer für Kleinanwendungen optimierten Leistung und Funktionalität. Ein Touchscreen in verschiedenen Monitorgrößen und eine aufeinander abgestimmte Kommunikation über Industrial Ethernet sind ideale Voraussetzungen für das Zusammenspiel mit S7-1200.

Die Engineeringsoftware STEP 7 im TIA-Portal erschließt alle Möglichkeiten der S7-1200-Controller. STEP 7 ist das gemeinsame Werkzeug für die Konfiguration des Hardware-Aufbaus und die Programmierung des Anwenderprogramms genauso wie für den Programmtest und die Diagnose. Mit der mit STEP 7 gelieferten Projektierungssoftware SIMATIC WinCC Basic werden die Basic Panels projektiert. Die moderne und intuitive Benutzerführung gestattet ein effizientes, aufgabenorientiertes Engineering von Steuerungs- und Visualisierungsgeräten.

Das vorliegende Buch beschreibt das Projektieren, Programmieren und Testen von S7-1200-Controllern mit Firmware Version 4.2 und von HMI Basic Panels mit der Engineeringsoftware STEP 7 V14 Basic SP1 und WinCC Basic V14 SP1 einschließlich der Simulationssoftware PLCSIM V14 SP1.

Erlangen, im September 2017

Hans Berger

# Der Inhalt des Buchs auf einen Blick

---

## 1 Start

Das Automatisierungssystem SIMATIC S7-1200 in der Übersicht.

Die Engineeringsoftware SIMATIC STEP 7 in der Übersicht.

Adressierung, Datentypen und Programmbearbeitung in der Übersicht.

---

## 2 Einführung in STEP 7

STEP 7 installieren und starten. Die Benutzeroberfläche von STEP 7.

Die Grundlage der Automatisierungslösung: Ein Projekt erstellen und bearbeiten

---

## 3 Automatisierungssystem SIMATIC S7-1200

Die Baugruppen von SIMATIC S7-1200 in der Übersicht: Aufbau eines Automatisierungssystems, CPU-Baugruppen, Signal- und Kommunikationsbaugruppen.

---

## 4 Gerätekonfiguration

Eine Station konfigurieren, Baugruppen parametrieren und Stationen vernetzen. Hardware-Objekte adressieren.

---

## 5 Programmbearbeitung

Wie sich die CPU-Baugruppe in den Betriebszuständen ANLAUF, RUN und STOP verhält.

Wie das Anwenderprogramm durch Bausteine strukturiert wird, welche Eigenschaften die Bausteine haben und wie sie aufgerufen werden, mit Programmierempfehlungen.

Wie das Anwenderprogramm bearbeitet wird: Anlaufverhalten, Hauptprogramm, Alarmbearbeitung, Fehlerbehandlung und Diagnose.

---

## 6 Der Programmeditor

Operanden, Adressierung und Datentypen, mit der PLC-Variablen-tabelle arbeiten.

PLC-Datentypen, Codebausteine und Datenbausteine erstellen und bearbeiten, Bausteine übersetzen und Programminformationen auswerten.

---

## 7 Die Programmiersprache Kontaktplan KOP

Das Charakteristische der KOP-Programmierung; Reihen- und Parallelschaltung von Kontakten; der Einsatz von Spulen, Standard-Boxen, Q-Boxen und EN/ENO-Boxen.

---

## 8 Die Programmiersprache Funktionsplan FUP

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

---

## 9 Die Programmiersprache Structured Control Language SCL

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

---

---

## **Die Beschreibung der Steuerungsfunktionen**

**10 Basisfunktionen:** Funktionen für Binärsignale: binäre Verknüpfungen, Speicherfunktionen, Flankenauswertungen, Zeit- und Zählfunktionen.

**11 Digitalfunktionen:** Funktionen für Digitalvariablen: Übertragungs-, Vergleichs-, Arithmetik-, Mathematik-, Konvertierungs-, Schiebe-, Logik- und Zeichenkettenfunktionen.

**12 Programmsteuerung:** Sprungfunktionen, Bausteine aufrufen und beenden, Bausteinparameter adressieren, beschalten und übergeben; Datenbausteinfunktionen

---

## **13 Online-Betrieb, Diagnose und Programmtest**

Ein Programmiergerät an die PLC-Station anschließen, den Online-Betrieb einschalten. Die Anwenderbausteine laden, ändern, löschen und vergleichen. Mit der Hardware-Diagnose arbeiten. Das Anwenderprogramm testen. Messwertaufzeichnung mit der Trace-Funktion.

---

## **14 Dezentrale Peripherie**

Wie ein PROFINET IO System, ein PROFIBUS DP Mastersystem und ein Aktor/Sensor-Interface System projektiert werden und welche Eigenschaften sie haben. DPV1-Alarme.

---

## **15 Kommunikation**

Welche Eigenschaften und welche Kommunikationsfunktionen die Open User Communication, die S7-Kommunikation und die Punkt-zu-Punkt-Kommunikation haben und wie sie projektiert werden. Kommunikationsfunktionen für USS-Antriebe, Modbus RTU und TCP.

---

## **16 Visualisierung**

Einführung: Die Basic Panels in der Übersicht.

Start: Ein HMI-Projekt einrichten, der Bediengeräte-Assistent.

Verbindung zur PLC: HMI-Variablen und Bereichszeiger anlegen.

Bilder erstellen: Die Projektierung der Prozessbilder – Vorlagen, Ebenen und Bildwechsel.

Arbeiten mit Bildelementen: Bedien- und Anzeigeelemente anordnen und editieren.

Ein Meldesystem projektieren, Rezepturen anlegen und Datensätze übertragen, die Benutzerverwaltung verwenden, Variablen- und Meldearchive einrichten, Aufgaben planen.

Das HMI-Programm fertig stellen: Das HMI-Programm mit PLC-Station oder mit Variablen-tabelle simulieren. Das HMI-Programm zur HMI-Station übertragen.

---

## **17 Anhang**

Integrierte und technologische Funktionen: High Speed Counter, Impulsgenerator, Motion Control, PID Control. Mit TeleService und TeleControl Daten übertragen.

Wie der Webserver der CPU-Baugruppe projektiert wird und welche Möglichkeiten er bietet.

Mit der Datenprotokollierung und mit Rezepturen in der PLC-Station Prozesswerte übertragen und speichern.

Mit der Simulationssoftware S7-PLCSIM ein Anwenderprogramm offline testen.

---

# Inhaltsverzeichnis

<b>1 Einführung, Übersichten</b> .....	21
1.1 Übersicht Automatisierungssystem S7-1200 .....	21
1.2 Übersicht STEP 7 .....	24
1.3 Datenhaltung im SIMATIC-Automatisierungssystem .....	27
1.4 Übersicht Adressierung .....	31
1.5 Übersicht Datentypen .....	32
1.6 Bearbeitung des Anwenderprogramms .....	35
1.7 Bedienen und Beobachten mit Prozessbildern .....	38
<b>2 Einführung in STEP 7</b> .....	40
2.1 STEP 7 installieren und starten .....	40
2.1.1 STEP 7 installieren .....	40
2.1.2 Automation License Manager .....	40
2.1.3 STEP 7 Basic starten .....	41
2.2 Die Benutzeroberfläche von STEP 7 .....	41
2.2.1 Portalansicht .....	41
2.2.2 Die Fenster der Projektansicht .....	42
2.2.3 Informationssystem .....	45
2.2.4 Bedienoberfläche anpassen .....	46
2.2.5 Mehrsprachige Projekte .....	46
2.3 Ein SIMATIC-Projekt bearbeiten .....	47
2.3.1 Strukturierte Darstellung der Projektdaten .....	48
2.3.2 Projektdaten und Editoren für eine PLC-Station .....	49
2.3.3 Mit Projekten arbeiten .....	52
2.3.4 Mit Referenzprojekten arbeiten .....	57
2.3.5 Mit Bibliotheken arbeiten .....	57
<b>3 Automatisierungssystem SIMATIC S7-1200</b> .....	62
3.1 Komponenten einer S7-1200-Station .....	62
3.2 CPU-Baugruppen S7-1200 .....	63
3.2.1 CPU-Varianten .....	63
3.2.2 Onboard-Peripherie .....	65
3.2.3 PROFINET-Anschluss .....	66
3.2.4 Status-Leuchtdioden .....	66
3.2.5 SIMATIC Memory Card .....	67
3.2.6 Erweiterungen der CPU-Baugruppe .....	67

---

3.3	Signalbaugruppen	69
3.3.1	Digitalein-/ausgabebaugruppen	69
3.3.2	Analogein-/ausgabebaugruppen	70
3.3.3	Eigenschaften der Peripherieanschlüsse	70
3.4	Technologiebaugruppen	72
3.4.1	IO-Link-Master	72
3.4.2	Wägemodule SIWAREX	73
3.4.3	Power Signal Booster	73
3.5	Kommunikationsbaugruppen	73
3.5.1	Punkt-zu-Punkt-Kommunikation	74
3.5.2	PROFIBUS DP	74
3.5.3	Aktor/Sensor-Interface	74
3.5.4	Industrial Remote Communication	75
3.5.5	Kommunikationsbaugruppe RF120C	76
3.6	Weitere Baugruppen	77
3.6.1	Anschlussvervielfacher	77
3.6.2	Externe Spannungsversorgung	77
3.6.3	TS-Adapter IE Basic	77
3.6.4	Simulatormodule SIM 1274	78
3.7	SIPLUS S7-1200	79
<b>4</b>	<b>Gerätekonfiguration</b>	<b>80</b>
4.1	Einführung	80
4.2	Eine Station konfigurieren	82
4.2.1	Eine PLC-Station hinzufügen	82
4.2.2	Eine Baugruppe hinzufügen	82
4.2.3	Eine HMI-Station hinzufügen	83
4.3	Baugruppen parametrieren	84
4.3.1	Die CPU-Eigenschaften parametrieren	84
4.3.2	Baugruppen adressieren	87
4.3.3	Digitaleingaben parametrieren	89
4.3.4	Digitalausgaben parametrieren	90
4.3.5	Analogeingaben parametrieren	91
4.3.6	Analogausgaben parametrieren	91
4.3.7	Kommunikationsbaugruppen parametrieren	92
4.4	Hardware-Objekte adressieren	92
4.5	Konfigurationssteuerung projektieren	94
4.6	Eine Vernetzung projektieren	96
4.6.1	Einführung	96
4.6.2	Eine Station vernetzen	97
4.6.3	Teilnehmeradressen in einem Subnetz	99
4.6.4	Verbindungen	100
4.6.5	Ein PROFINET-Subnetz projektieren	103
4.6.6	Ein PROFIBUS-Subnetz projektieren	106
4.6.7	Ein AS-i-Subnetz projektieren	107

<b>5 Anwenderprogramm bearbeiten</b> .....	109
5.1 Betriebszustände .....	109
5.1.1 Betriebszustand STOP .....	110
5.1.2 Betriebszustand ANLAUF .....	111
5.1.3 Betriebszustand RUN .....	113
5.1.4 Remanenzverhalten von Operanden .....	113
5.2 Anwenderprogramm erstellen .....	114
5.2.1 Programmbearbeitung .....	114
5.2.2 Strukturierung des Anwenderprogramms .....	116
5.2.3 Bausteinararten .....	119
5.2.4 Bausteineigenschaften .....	121
5.2.5 Bausteinschnittstelle .....	121
5.2.6 Einen Codebaustein aufrufen .....	126
5.2.7 Programmierempfehlungen .....	128
5.3 Anlaufprogramm .....	136
5.3.1 Organisationsbausteine für das Anlaufprogramm .....	136
5.3.2 Baugruppenadresse ermitteln .....	137
5.4 Hauptprogramm .....	141
5.4.1 Organisationsbausteine für das Hauptprogramm .....	141
5.4.2 Prozessabbild-Aktualisierung .....	142
5.4.3 Zykluszeit .....	143
5.4.4 Reaktionszeit .....	146
5.4.5 Programmbearbeitung stoppen .....	147
5.4.6 Uhrzeit .....	148
5.4.7 Betriebsstundenzähler .....	152
5.5 Alarmbearbeitung .....	154
5.5.1 Einführung zur Alarmbearbeitung .....	154
5.5.2 Uhrzeitalarme .....	159
5.5.3 Verzögerungsalarme .....	163
5.5.4 Weckalarme .....	167
5.5.5 Prozessalarme .....	170
5.5.6 Alarmer zur Laufzeit zuordnen .....	173
5.5.7 Alarmer und Asynchronfehler verzögern und freigeben .....	174
5.5.8 Startinformation lesen .....	175
5.6 Fehlerbehandlung, Diagnose .....	176
5.6.1 Fehlerursachen und Fehlerreaktionen .....	176
5.6.2 Lokale Fehlerbehandlung .....	178
5.6.3 Zeitfehler OB 80 .....	182
5.6.4 Diagnosealarm OB 82 .....	183
5.6.5 Ziehen/Stecken-Alarm OB 83 .....	184
5.6.6 Baugruppenträgerfehler OB 86 .....	185
5.6.7 Diagnosefunktionen im Anwenderprogramm .....	186
5.6.8 Anwender-Diagnosemeldung erzeugen .....	194

---

<b>6 Programmeditor</b> .....	198
6.1 Einführung .....	198
6.2 Globale Operanden, Konstanten und Adressierung .....	199
6.2.1 Eingänge, Peripherie-Eingänge .....	199
6.2.2 Ausgänge, Peripherie-Ausgänge .....	199
6.2.3 Operandenbereich Merker .....	200
6.2.4 Absolute Adressierung .....	202
6.2.5 Symbolische Adressierung .....	203
6.2.6 Adressierung von Konstanten .....	205
6.2.7 Indirekte Adressierung (Übersicht) .....	206
6.3 Datentypen .....	207
6.3.1 Datentypen für Bitfolgen .....	207
6.3.2 Datentypen für BCD-Werte .....	207
6.3.3 Datentypen für Ganzzahlen .....	208
6.3.4 Datentypen für Gleitpunktzahlen .....	209
6.3.5 Datentypen für Zeitwerte .....	210
6.3.6 Datentypen für Zeichen .....	211
6.3.7 Datentyp ARRAY .....	214
6.3.8 Datentyp STRUCT .....	217
6.4 PLC-Variablen-tabelle .....	219
6.4.1 PLC-Variablen-tabelle anlegen und bearbeiten .....	219
6.4.2 PLC-Variablen definieren .....	219
6.4.3 PLC-Variablen-tabellen vergleichen .....	222
6.4.4 PLC-Variablen-tabelle exportieren und importieren .....	222
6.4.5 Konstanten-tabellen .....	223
6.5 PLC-Datentypen .....	224
6.5.1 Einen PLC-Datentyp programmieren .....	224
6.5.2 Einen PLC-Datentyp anwenden .....	225
6.5.3 PLC-Datentypen vergleichen .....	226
6.5.4 Einen PLC-Datentyp umnummerieren .....	227
6.6 Einen Codebaustein programmieren .....	228
6.6.1 Einen neuen Codebaustein anlegen .....	228
6.6.2 Arbeitsbereich des Programmeditors für Codebausteine .....	229
6.6.3 Bausteineigenschaften für Codebausteine festlegen .....	230
6.6.4 Einen Codebaustein schützen .....	234
6.6.5 Bausteinschnittstelle programmieren .....	237
6.6.6 Operandenbereich temporäre Lokaldaten .....	240
6.6.7 Allgemeines Vorgehen beim Programmieren der Steuerungsfunktion .....	241
6.6.8 Instanzdaten anlegen .....	244
6.7 Einen Datenbaustein programmieren .....	248
6.7.1 Einen neuen Datenbaustein anlegen .....	248
6.7.2 Arbeitsbereich des Programmeditors für Datenbausteine .....	249
6.7.3 Bausteineigenschaften für Datenbausteine festlegen .....	250
6.7.4 Datenvariablen deklarieren .....	252
6.7.5 Datenvariablen in Global-Datenbausteinen eingeben .....	253

6.8 Bausteine übersetzen .....	255
6.8.1 Übersetzung starten .....	255
6.8.2 SCL-Bausteine übersetzen .....	256
6.8.3 Fehler nach der Übersetzung beheben .....	256
6.9 Programminformationen .....	258
6.9.1 Querverweisliste .....	258
6.9.2 Belegungsplan .....	261
6.9.3 Aufrufstruktur .....	262
6.9.4 Abhängigkeitsstruktur .....	263
6.9.5 Konsistenzprüfung .....	264
6.9.6 Speicherauslastung der CPU .....	265
<b>7 Kontaktplan KOP .....</b>	<b>266</b>
7.1 Einführung .....	266
7.1.1 Mit Kontaktplan programmieren .....	266
7.1.2 Programmelemente des Kontaktplans .....	272
7.2 Binäre Verknüpfungen mit KOP programmieren .....	272
7.2.1 Schließerkontakt und Öffnerkontakt .....	272
7.2.2 Reihen- und Parallelschaltung von Kontakten .....	274
7.2.3 T-Abzweig, offener Parallelzweig .....	277
7.2.4 Verknüpfungsergebnis negieren im Kontaktplan .....	278
7.2.5 Flankenbewertung einer Binärvariablen im Kontaktplan .....	278
7.2.6 Gültigkeitsprüfung einer Gleitpunktvariablen im Kontaktplan .....	279
7.2.7 Vergleich von Variablen im Kontaktplan .....	280
7.2.8 VARIANT-Zeiger testen und vergleichen im Kontaktplan .....	281
7.3 Speicherfunktionen mit KOP programmieren .....	282
7.3.1 Einfache und negierende Spule .....	282
7.3.2 Setzen- und Rücksetzen-Spule .....	283
7.3.3 Speicherndes Verhalten durch Selbsthaltung .....	284
7.3.4 Flankenbewertung mit Impulsausgabe im Kontaktplan .....	285
7.3.5 Mehrfaches Setzen und Rücksetzen (Bitfeld füllen) im Kontaktplan ..	286
7.3.6 Spulen mit Zeitverhalten .....	286
7.4 Q-Boxen mit KOP programmieren .....	287
7.4.1 Speicher-Boxen im Kontaktplan .....	288
7.4.2 Flankenbewertung des Stromflusses .....	288
7.4.3 Beispiel Binäruntersetzer im Kontaktplan .....	289
7.4.4 Zeitfunktionen im Kontaktplan .....	291
7.4.5 Zählfunktionen im Kontaktplan .....	292
7.5 EN/ENO-Boxen mit KOP programmieren .....	293
7.5.1 Flankenbewertung mit einer EN/ENO-Box .....	293
7.5.2 Übertragungsfunktionen im Kontaktplan .....	295
7.5.3 Arithmetische Funktionen für Zahlenwerte im Kontaktplan .....	296
7.5.4 Arithmetische Funktionen für Zeitwerte im Kontaktplan .....	298
7.5.5 Mathematische Funktionen im Kontaktplan .....	298

---

7.5.6	Konvertierungsfunktionen im Kontaktplan	299
7.5.7	Schiebefunktionen im Kontaktplan	300
7.5.8	Logikfunktionen im Kontaktplan	300
7.5.9	Funktionen für Zeichenketten im Kontaktplan	301
7.5.10	Die CALCULATE-Box im Kontaktplan	302
7.6	Programmsteuerung mit KOP	302
7.6.1	Sprungfunktionen im Kontaktplan	303
7.6.2	Sprungliste im Kontaktplan	304
7.6.3	Sprungverteiler im Kontaktplan	304
7.6.4	Bausteinende-Funktion im Kontaktplan	305
7.6.5	Bausteinaufruf-Funktionen im Kontaktplan	306
<b>8</b>	<b>Funktionsplan FUP</b>	<b>308</b>
8.1	Einführung	308
8.1.1	Mit Funktionsplan programmieren	308
8.1.2	Programmelemente des Funktionsplans	314
8.2	Binäre Verknüpfungen mit FUP programmieren	314
8.2.1	Abfrage auf Signalzustand „1“ und auf Signalzustand „0“	314
8.2.2	Eine binäre Verknüpfung im Funktionsplan programmieren	316
8.2.3	T-Abzweig im Funktionsplan	319
8.2.4	Verknüpfungsergebnis negieren im Funktionsplan	319
8.2.5	Flankenauswertung einer Binärvariablen im Funktionsplan	321
8.2.6	Gültigkeitsprüfung einer Gleitpunktvariablen im Funktionsplan	321
8.2.7	Vergleich von Variablen im Funktionsplan	322
8.2.8	VARIANT-Zeiger testen und vergleichen im Funktionsplan	323
8.3	Standard-Boxen mit FUP programmieren	324
8.3.1	Zuweisung und negierende Zuweisung	324
8.3.2	Setzen- und Rücksetzen-Box	326
8.3.3	Flankenauswertung mit Impulsausgabe im Funktionsplan	326
8.3.4	Mehrfaches Setzen und Rücksetzen (Bitfeld füllen) im Funktionsplan	327
8.3.5	Standardboxen mit Zeitverhalten	328
8.4	Q-Boxen mit FUP programmieren	329
8.4.1	Speicher-Boxen im Funktionsplan	329
8.4.2	Flankenauswertung des Verknüpfungsergebnisses	330
8.4.3	Beispiel Binäruntersetzer im Funktionsplan	330
8.4.4	Zeitfunktionen im Funktionsplan	331
8.4.5	Zählfunktionen im Funktionsplan	333
8.5	EN/ENO-Boxen mit FUP programmieren	334
8.5.1	Flankenauswertung mit einer EN/ENO-Box	334
8.5.2	Übertragungsfunktionen im Funktionsplan	336
8.5.3	Arithmetische Funktionen für Zahlenwerte im Funktionsplan	338
8.5.4	Arithmetische Funktionen für Zeitwerte im Funktionsplan	339
8.5.5	Mathematische Funktionen im Funktionsplan	340
8.5.6	Konvertierungsfunktionen im Funktionsplan	340
8.5.7	Schiebefunktionen im Funktionsplan	341

8.5.8 Logikfunktionen im Funktionsplan .....	342
8.5.9 Funktionen für Zeichenketten im Funktionsplan .....	343
8.5.10 Die CALCULATE-Box im Funktionsplan .....	344
8.6 Programmsteuerung mit FUP .....	344
8.6.1 Sprungfunktionen im Funktionsplan .....	344
8.6.2 Sprungliste im Funktionsplan .....	346
8.6.3 Sprungverteiler im Funktionsplan .....	347
8.6.4 Bausteinende-Funktion im Funktionsplan .....	347
8.6.5 Bausteinaufruf-Funktionen im Funktionsplan .....	347
<b>9 Structured Control Language SCL .....</b>	<b>350</b>
9.1 Einführung .....	350
9.1.1 Mit SCL programmieren .....	350
9.1.2 SCL-Anweisungen und Operatoren .....	355
9.1.3 Datentyp-Konvertierung bei SCL .....	357
9.2 Übertragungsfunktionen .....	359
9.2.1 Wertzuweisung einer Binärvariablen .....	359
9.2.2 Flankenbewertung bei SCL .....	360
9.2.3 Wertzuweisung einer Digitalvariablen .....	361
9.2.4 Übertragen von VARIANT-Variablen bei SCL .....	362
9.2.5 Datenbereiche übertragen und füllen .....	363
9.2.6 Bytes tauschen .....	363
9.2.7 Datenübertragung mit PEEK und POKE .....	364
9.3 Logische Ausdrücke und Logikfunktionen .....	366
9.3.1 Binäre Verknüpfungen mit SCL programmieren .....	366
9.3.2 Wortverknüpfungen bei SCL .....	369
9.3.3 Codier-, Auswahl- und Begrenzerfunktionen bei SCL .....	370
9.4 Arithmetische Ausdrücke .....	371
9.4.1 Arithmetische Funktionen bei SCL .....	371
9.4.2 Mathematische Funktionen bei SCL .....	372
9.5 Vergleichsausdrücke .....	373
9.5.1 Vergleichsfunktionen bei SCL .....	373
9.5.2 VARIANT-Zeiger testen .....	373
9.6 Weitere Funktionen für SCL .....	375
9.6.1 Zeitfunktionen bei SCL .....	375
9.6.2 Zählfunktionen bei SCL .....	375
9.6.3 Schiebefunktionen bei SCL .....	377
9.6.4 Zeichenkettenfunktionen bei SCL .....	378
9.7 Programmsteuerung mit SCL .....	378
9.7.1 Kontrollanweisungen .....	378
9.7.2 Bausteinende-Funktion bei SCL .....	389
9.7.3 Aufruf einer Funktion (FC) bei SCL .....	389
9.7.4 Aufruf eines Funktionsbausteins (FB) bei SCL .....	390
9.7.5 Versorgung von Parametern .....	391
9.8 Arbeiten mit Quelldateien .....	392

---

<b>10 Basisfunktionen</b> .....	396
10.1 Binäre Verknüpfungen .....	396
10.1.1 Einführung .....	396
10.1.2 Arbeiten mit Binärsignalen .....	397
10.1.3 UND-Funktion, Reihenschaltung .....	399
10.1.4 ODER-Funktion, Parallelschaltung .....	399
10.1.5 Exklusiv-ODER-Funktion, Antivalenzfunktion .....	402
10.1.6 Verknüpfungsergebnis negieren, NOT-Kontakt .....	402
10.2 Speicherfunktionen .....	403
10.2.1 Einführung .....	403
10.2.2 Einfache und negierende Spule, Zuweisung .....	404
10.2.3 Einzelnes Setzen und Rücksetzen .....	404
10.2.4 Mehrfaches Setzen und Rücksetzen .....	406
10.2.5 Vorrangiges Setzen und Rücksetzen, Speicher-Boxen .....	407
10.3 Flankenbewertungen .....	407
10.4 Zeitfunktionen .....	413
10.4.1 Wissenswertes zu Zeitfunktionen .....	413
10.4.2 Impulsbildung TP .....	414
10.4.3 Einschaltverzögerung TON .....	415
10.4.4 Ausschaltverzögerung TOF .....	416
10.4.5 Akkumulierende Einschaltverzögerung TONR .....	418
10.4.6 Eine Zeitfunktion mit einer Zeitdauer laden .....	419
10.4.7 Eine Zeitfunktion zurücksetzen .....	419
10.5 Zählfunktionen .....	420
10.5.1 Wissenswertes zu Zählfunktionen .....	420
10.5.2 Vorwärtszähler CTU .....	422
10.5.3 Rückwärtszähler CTD .....	423
10.5.4 Vorwärts-Rückwärtszähler CTUD .....	424
<b>11 Digitalfunktionen</b> .....	426
11.1 Übertragungsfunktionen .....	426
11.1.1 Variable kopieren, MOVE- und S_MOVE-Box bei KOP und FUP .....	427
11.1.2 Wertzuweisungen bei SCL .....	429
11.1.3 VARIANT-Variable lesen und schreiben .....	431
11.1.4 Datenbereich kopieren mit MOVE_BLK_VARIANT .....	432
11.1.5 Datenbereich kopieren mit MOVE_BLK und UMOVE_BLK .....	433
11.1.6 Datenbereich füllen mit FILL_BLK und UFILL_BLK .....	435
11.1.7 Variable von und zu einen BYTE-Feld übertragen .....	437
11.1.8 Variable von und zu einem Bitfeld übertragen .....	440
11.1.9 Byte-Reihenfolge ändern .....	442
11.1.10 Bereichsgrenzen ermitteln .....	442
11.2 Vergleichsfunktionen .....	442
11.2.1 Vergleich von Variablenwerten .....	443
11.2.2 Bereichsvergleich (KOP, FUP) .....	445

11.2.3	Gleitpunkt-Variable testen (KOP, FUP)	445
11.2.4	VARIANT-Zeiger testen (KOP, FUP)	446
11.2.5	VARIANT-Zeiger testen (SCL)	448
11.3	Arithmetische Funktionen für Zahlenwerte	451
11.3.1	Grundrechnungsarten	451
11.3.2	Division mit Rest als Ergebnis MOD	453
11.3.3	Dekrementieren DEC, Inkrementieren INC	453
11.4	Arithmetische Funktionen für Zeitwerte	454
11.4.1	Addition T_ADD	454
11.4.2	Subtraktion T_SUB	455
11.4.3	Differenz T_DIFF	456
11.4.4	Zusammenfassen T_COMBINE	456
11.5	Mathematische Funktionen	456
11.5.1	Winkelfunktionen SIN, COS, TAN	457
11.5.2	Arcusfunktionen ASIN, ACOS, ATAN	458
11.5.3	Quadrat bilden SQR	458
11.5.4	Quadratwurzel ziehen SQRT	458
11.5.5	Potenzieren zur Basis e EXP	459
11.5.6	Natürlichen Logarithmus berechnen LN	459
11.5.7	Nachkommastellen extrahieren FRAC	460
11.5.8	Potenzieren zu einer beliebigen Basis EXPT	460
11.5.9	Absolutwertbildung ABS	460
11.5.10	Negation NEG	460
11.6	Konvertierungsfunktionen (Datentypwandlung)	461
11.6.1	Implizite Datentypkonvertierung	462
11.6.2	Variablen überlagern (Datentypsichten)	464
11.6.3	Konvertierungsfunktionen CONVERT, S_CONV und T_CONV	466
11.6.4	Konvertierungsfunktionen für Gleitpunktzahlen	472
11.6.5	Konvertierungsfunktionen STRG_TO_CHARS und CHARS_TO_STRG	473
11.6.6	Konvertierungsfunktionen STRG_VAL und VAL_STRG	475
11.6.7	Konvertierungsfunktionen ATH und HTA	478
11.6.8	Konvertierungsfunktionen SCALE_X und NORM_X	480
11.7	Schiebefunktionen	480
11.8	Logikfunktionen	483
11.8.1	Wortverknüpfungen	483
11.8.2	Invertieren INVERT	485
11.8.3	Codierfunktionen DECO und ENCO	485
11.8.4	Auswahlfunktionen SEL, MUX und DEMUX	487
11.8.5	Auswahlfunktionen MIN und MAX	488
11.8.6	Begrenzer LIMIT	490
11.9	Zeichenketten bearbeiten	490
11.10	Rechnen mit der CALCULATE-Box (KOP, FUP)	496
11.11	Symbolnamen lesen	498

---

<b>12 Programmsteuerung</b> .....	503
12.1 Sprungfunktionen .....	503
12.1.1 Sprung abhängig vom Verknüpfungsergebnis (KOP, FUP) .....	504
12.1.2 Sprungliste JMP_LIST (KOP, FUP) .....	505
12.1.3 Sprungverteiler SWITCH (KOP, FUP) .....	506
12.1.4 Absoluter Sprung GOTO (SCL) .....	507
12.2 Bausteinende-Funktion .....	508
12.3 Aufruf von Codebausteinen .....	509
12.3.1 Wissenswertes zum Bausteinaufruf .....	509
12.3.2 Aufruf einer Funktion (FC) .....	510
12.3.3 Aufruf eines Funktionsbausteins (FB) .....	510
12.3.4 Asynchron arbeitende Systembausteine .....	511
12.3.5 EN/ENO-Mechanismus .....	513
12.4 Arbeiten mit Bausteinen .....	515
12.4.1 Bausteine mit optimiertem und Standardzugriff .....	515
12.4.2 Datentypen der lokalen Variablen .....	517
12.4.3 Bausteinparameter adressieren .....	519
12.4.4 Bausteinparameter versorgen .....	521
12.4.5 Parametertyp VARIANT .....	523
12.4.6 Übergabe von Bausteinparametern .....	525
12.5 Datenbausteinfunktionen .....	528
12.5.1 Hardware-Datentyp DB_ANY .....	529
12.5.2 Konvertierung von DB_ANY .....	530
12.5.3 Operandenbereich Daten .....	531
12.5.4 Adressierung von Datenbausteinen .....	533
12.5.5 Datenbausteinattribute lesen .....	534
12.5.6 CPU-Datenbausteine .....	535
12.5.7 Ladespeicher lesen und schreiben .....	538
<b>13 Online-Betrieb, Diagnose und Test</b> .....	542
13.1 Programmiergerät an die PLC-Station anschließen .....	543
13.2 Projektdaten übertragen .....	545
13.2.1 Die Projektdaten erstmalig laden .....	546
13.2.2 Die Projektdaten nachladen .....	548
13.2.3 Das Anwenderprogramm schützen .....	549
13.2.4 Mit Online-Projektdaten arbeiten .....	552
13.2.5 Mit der Memory Card arbeiten .....	554
13.3 Mit Bausteinen im Online-Betrieb arbeiten .....	557
13.3.1 Einführung .....	557
13.3.2 Einen Online-Baustein bearbeiten .....	558
13.3.3 Einen Baustein laden und zurückladen .....	559
13.3.4 Ohne Reinitialisierung laden .....	560
13.3.5 Mit Momentaufnahmen arbeiten .....	562

13.3.6 Mit Einstellwerten arbeiten	563
13.3.7 Bausteine vergleichen	565
13.4 Hardware-Diagnose	568
13.4.1 Status-Anzeigen an den Baugruppen	569
13.4.2 Diagnosefenster	570
13.4.3 Diagnosepuffer	571
13.4.4 Online-Tools	572
13.4.5 Weitere Diagnose-Informationen über das Programmiergerät	573
13.5 Anwenderprogramm testen	574
13.5.1 Aufrufumgebung definieren	575
13.5.2 Testen mit Programmstatus	575
13.5.3 PLC-Variablen beobachten	580
13.5.4 Datenvariablen beobachten und steuern	580
13.5.5 Testen mit Beobachtungstabellen	581
13.5.6 Testen mit der Force-Tabelle	588
13.6 Messwertaufzeichnung mit der Trace-Funktion	591
13.6.1 Einführung	591
13.6.2 Erstellen der Trace-Konfiguration	591
13.6.3 Messwerte aufzeichnen	592
13.6.4 Messungen auswerten	593
13.6.5 Messungen speichern	595
<b>14 Dezentrale Peripherie</b>	<b>596</b>
14.1 Einführung, Übersicht	596
14.2 PROFINET IO	597
14.2.1 Komponenten von PROFINET IO	597
14.2.2 Adressen bei PROFINET IO	599
14.2.3 PROFINET IO projektieren	601
14.2.4 Echtzeit-Kommunikation bei PROFINET IO	606
14.3 PROFIBUS DP	607
14.3.1 Komponenten von PROFIBUS DP	607
14.3.2 Adressen bei PROFIBUS DP	610
14.3.3 PROFIBUS DP projektieren	612
14.4 Systembausteine für PROFINET IO und PROFIBUS DP	615
14.4.1 Peripheriedaten lesen und schreiben	615
14.4.2 Datensätze übertragen	618
14.4.3 Dezentrale Station aktivieren/deaktivieren	621
14.5 DPV1-Alarme	622
14.6 Aktor/Sensor-Interface	624
14.6.1 Komponenten von Aktor/Sensor-Interface	624
14.6.2 AS-i-Master CM 1243-2 projektieren	624
14.6.3 Aktor/Sensor-Interface projektieren	624
14.6.4 Schnittstelle zum Anwenderprogramm	626

---

<b>15 Kommunikation</b>	628
15.1 Übersicht	628
15.2 Open User Communication	631
15.2.1 Grundlagen	631
15.2.2 Datenstruktur der Open User Communication	631
15.2.3 Verbindung aufbauen und Daten senden mit TSEND_C	632
15.2.4 Verbindung aufbauen und Daten empfangen mit TRCV_C	634
15.2.5 Open User Communication projektieren	636
15.2.6 Weitere Funktionen für die Open User Communication	637
15.3 S7-Kommunikation	639
15.3.1 Grundlagen	639
15.3.2 Datenaustausch mit GET und PUT	640
15.3.3 S7-Kommunikation projektieren	642
15.4 Punkt-zu-Punkt-Kommunikation	643
15.4.1 Einführung in die Punkt-zu-Punkt-Kommunikation	643
15.4.2 Konfigurieren der Kommunikationsbaugruppe CM 1241	643
15.4.3 Funktionen für die Punkt-zu-Punkt-Kommunikation	646
15.5 Weitere Kommunikationsfunktionen	649
15.5.1 USS-Protokoll für Antriebe	649
15.5.2 Modbus RTU	651
15.5.3 Modbus TCP	653
<b>16 Visualisierung</b>	656
16.1 Einführung in die Visualisierung	656
16.1.1 Übersicht HMI Panels in STEP 7 (TIA Portal)	657
16.1.2 Ein Projekt mit einer HMI-Station anlegen	659
16.1.3 Querverweise für HMI-Objekte	661
16.2 HMI-Variablen und Bereichszeiger anlegen	663
16.2.1 Einführung HMI-Variablen	663
16.2.2 Eine HMI-Variable anlegen	664
16.2.3 Einen Bereichszeiger anlegen	665
16.3 Prozessbilder projektieren	667
16.3.1 Einführung in die Projektierung von Prozessbildern	667
16.3.2 Arbeitsfenster für Prozessbilder	668
16.3.3 Arbeiten mit Bildebenen	668
16.3.4 Arbeiten mit Vorlagen	669
16.3.5 Arbeiten mit Funktionstasten	670
16.3.6 Ein neues Bild anlegen	672
16.3.7 Einen Bildwechsel projektieren	672
16.3.8 Arbeiten mit Objekten in Prozessbildern	673
16.3.9 Bildobjekte zur Laufzeit verändern	674
16.3.10 Basisobjekte für die Bildprojektierung	674

16.4	Bedien- und Beobachtungsfunktionen	675
16.4.1	Eingabe und Anzeige von Prozesswerten	675
16.4.2	Arbeiten mit Meldungen	678
16.4.3	Arbeiten mit Rezepturen	687
16.4.4	Arbeiten mit der Benutzerverwaltung	692
16.4.5	Arbeiten mit Archiven	695
16.4.6	Aufgaben planen	697
16.5	HMI-Projektierung fertig stellen	697
16.5.1	HMI-Projektierung übersetzen (Konsistenzprüfung)	697
16.5.2	Simulation der HMI-Projektierung	698
16.5.3	Projektierung in die HMI-Station laden	699
16.5.4	Wartung der HMI-Station	701
<b>17</b>	<b>Anhang</b>	<b>704</b>
17.1	Integrierte und technologische Funktionen	704
17.1.1	High Speed Counter (HSC)	704
17.1.2	Impulsgenerator	710
17.1.3	Technologieobjekte für Motion Control	713
17.1.4	Technologieobjekte für PID Control	718
17.2	Fernverbindung mit TeleService	722
17.3	TeleControl mit CP 1242-7	724
17.4	Webserver	726
17.4.1	Webserver aktivieren	726
17.4.2	Zugriff auf den Webserver	726
17.4.3	Standard-Webseiten	727
17.5	Daten protokollieren und mit Rezepturen arbeiten	729
17.5.1	Einführung in die Datenprotokollierung	729
17.5.2	Funktionen für die Datenprotokollierung	730
17.5.3	Einführung zur Rezepturenübertragung	732
17.5.4	Funktionen für die Rezepturenübertragung	734
17.6	Simulation mit S7-PLCSIM	735
17.6.1	S7-PLCSIM allgemein	735
17.6.2	Die Bedienoberfläche von S7-PLCSIM	736
17.6.3	In S7-PLCSIM mit STEP-7-Testfunktionen testen	737
17.6.4	Mit einem Simulationsprojekt arbeiten	739
17.6.5	Mit dem Adressbereich testen	740
17.6.6	Mit der SIM-Tabelle testen	741
17.6.7	Mit der Sequenztabelle testen	742
	<b>Stichwortverzeichnis</b>	<b>746</b>

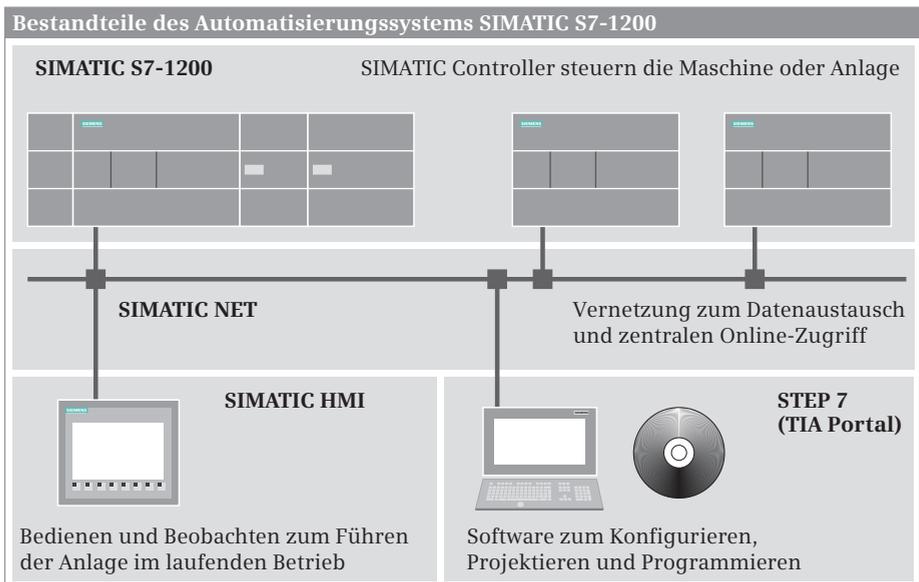
# 1 Einführung, Übersichten

## 1.1 Übersicht Automatisierungssystem S7-1200

SIMATIC S7-1200 ist das modulare Automatisierungssystem für den unteren und mittleren Leistungsbereich. Verschiedene Varianten der Controller passen die Leistungsfähigkeit an den jeweiligen Einsatzfall an. Je nach Bedarf kann das Automatisierungsgerät mit Signalbaugruppen sowie mit Kommunikationsbaugruppen modular erweitert werden. Das Automatisierungssystem SIMATIC S7-1200 ist nahtlos eingebettet in die SIMATIC-Systemarchitektur (Bild 1.1).

Die **Controller SIMATIC S7-1200** bilden als speicherprogrammierbare Steuerungen (SPS oder PLC: Programmable Logic Controller) die Basis des Automatisierungssystems. Mehrere Controller mit abgestufter Leistungsfähigkeit überdecken den unteren und mittleren Bereich der industriellen Steuerungen.

**SIMATIC HMI** bedeutet Bedienen und Beobachten (HMI: Human Machine Interface). Die Basic Panels sind so ausgelegt, dass sie optimal mit SIMATIC S7-1200 zusammenarbeiten. Die Geräte sind in Displaygrößen von 3,6 Zoll bis 15 Zoll verfügbar und werden über Touchscreen bedient. Bis auf das 15-Zoll-Gerät haben die Geräte zusätzlich Funktionstasten.



**Bild 1.1** Bestandteile des Automatisierungssystems SIMATIC S7-1200

**SIMATIC NET** verbindet alle SIMATIC-Stationen und sorgt für komplikationslosen Datenaustausch. SIMATIC S7-1200 mit PROFINET-Schnittstelle verwendet das Industrial Ethernet Netzwerk, um mit anderen PLC-Stationen, HMI-Stationen und Programmiergeräten Daten auszutauschen. Kommunikationsbaugruppen erweitern die Kommunikationsfähigkeit auf andere Netze, wie PROFIBUS DP, Aktor/Sensor-Interface oder Punkt-zu-Punkt-Kopplung auf RS232- oder RS485-Basis.

Mit der Engineering-Software **STEP 7** werden die SIMATIC-Komponenten konfiguriert, projektiert und parametrisiert, mit STEP 7 wird das Anwenderprogramm erstellt und getestet. Das *TIA Portal* ist die zentrale Bedienoberfläche zum Verwalten der Werkzeuge und Automatisierungsdaten. STEP 7 im TIA Portal gibt es in den Varianten STEP 7 Professional und STEP 7 Basic. Mit beiden Varianten kann eine S7-1200-Station projektiert und programmiert werden. In diesem Buch ist die Anwendung von STEP 7 Basic beschrieben.

### Komponenten und Anschlüsse

Das Bild 1.2 zeigt einen Überblick über die Anschlussmöglichkeiten an einen Basic Controller S7-1200.

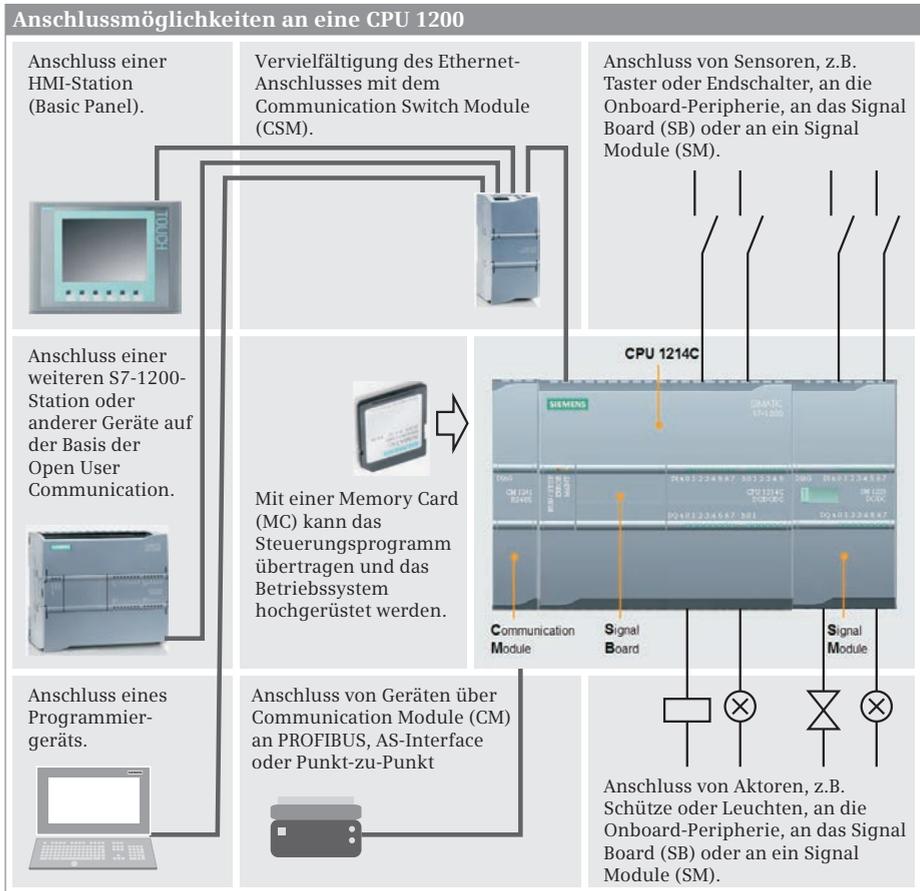
Die Zentralbaugruppe (**Central Processor Unit, CPU**) enthält das Betriebssystem und das Anwenderprogramm. Das Anwenderprogramm befindet sich spannungsausfallsicher im Ladespeicher. Die ablaufrelevanten Teile des Anwenderprogramms werden in einem Arbeitsspeicher mit schnellem Zugriff bearbeitet. Variablenwerte, die auch bei Spannungsausfall oder beim Aus- und Einschalten ihren Wert behalten sollen, werden im Remanenzspeicher abgelegt.

Mithilfe einer steckbaren Speicherkarte (**Memory Card, MC**) kann das Anwenderprogramm auf die CPU-Baugruppe übertragen werden – alternativ zum Übertragen über eine Online-Verbindung mit dem Programmiergerät. Die Speicherkarte kann auch als externer Ladespeicher oder zum Firmware-Update eingesetzt werden.

Die Verbindung zur Anlage oder zum Prozess übernehmen **Onboard-Ein- und -Ausgaben**, die je nach Version der Zentralbaugruppe in unterschiedlicher Anzahl vorhanden sind. Die Onboard-Ein- und -Ausgaben sind auch speziell für den Betrieb der integrierten Hochgeschwindigkeitszähler (HSC: High Speed Counter) ausgelegt. Das Betriebssystem enthält außerdem Impulsgeneratoren mit weitenmodulierter Impulsausgabe sowie die Technologieobjekte *Achse* zur Ansteuerung von Schrittmotoren und Servomotoren mit Pulsschnittstelle und *PID Compact*, einen PID-Regler mit optimierter Selbsteinstellung.

Mit einem Signal Board (SB) können die Onboard-Ein- und -Ausgaben erweitert werden. Das Communication Board (CB) schafft einen Punkt-zu-Punkt-Anschluss für die CPU-Baugruppe und das Battery Board (BB) verlängert die Gangreserve der integrierten Hardware-Uhr auf etwa ein Jahr.

Sind zusätzliche Ein- und Ausgänge erforderlich, können – je nach Version der CPU-Baugruppe – Ein-/Ausgabebaugruppen (**Signal Modules, SM**) an die CPU-Baugruppe angesteckt werden. Sie sind für Digital- und Analogsignale verfügbar.



**Bild 1.2** Anschlussmöglichkeiten an eine PLC-Station mit CPU 1200

Die **PROFINET-Schnittstelle** verbindet die CPU-Baugruppe mit dem Subnetz Industrial Ethernet. An ihr wird das Programmiergerät angeschlossen, wenn beispielsweise das Anwenderprogramm online zur CPU-Baugruppe übertragen und an der Maschine getestet werden soll. Über diese Schnittstelle geschieht auch der Datenaustausch mit HMI-Stationen oder anderen Automatisierungsgeräten.

Wird die CPU-Baugruppe mit nur einem Gerät über Ethernet verbunden, kann es direkt mit einem Standard- oder „Cross-over“-Kabel geschehen. Bei der Vernetzung von mehr als zwei Geräten, die nur eine PROFINET-Schnittstelle haben, müssen die Verbindungskabel über einen Vervielfacher, z. B. das **Communication Switch Module (CSM)** geführt werden. CPU 1215 und CPU 1217 haben zwei mit einem Switch verbundene Anschlüsse, so dass sie mit dem nächsten Automatisierungsgerät ohne zwischengeschalteten Anschlussvervielfacher vernetzt werden können.

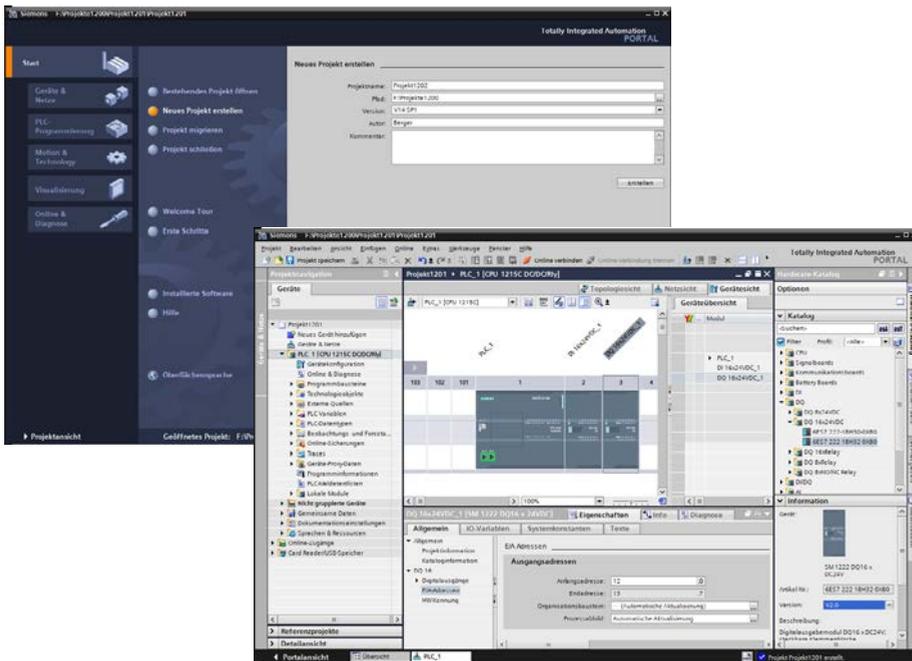
Kommunikationsbaugruppen (**Communication Modules, CM**) gestatten den Betrieb an weiteren Bussystemen, wie beispielsweise PROFIBUS DP. Hier kann eine

S7-1200-Station in einem DP-Mastersystem sowohl der DP-Master als auch ein DP-Slave sein. An AS-Interface kann eine S7-1200-Station der AS-Interface-Master sein und bis zu 62 AS-Interface-Feldgeräte steuern. Die Kommunikationsbaugruppe für die Punkt-zu-Punkt-Kopplung gibt es mit RS232- oder RS485-Schnittstelle, an die beispielsweise ein Barcode- oder RFID-Leser angeschlossen werden kann.

## 1.2 Übersicht STEP 7

STEP 7 ist das zentrale Automatisierungswerkzeug für SIMATIC. STEP 7 benötigt zum Betrieb eine Nutzungsberechtigung (Lizenz) und läuft auf den jeweils aktuellen Betriebssystemen Microsoft Windows. Mit STEP 7 Basic oder STEP 7 Professional können die S7-1200-Controller und – mit dem mitgelieferten WinCC Basic – die Basic Panels projektiert werden. Die Projektierung geschieht mit zwei Ansichten: die Portalansicht und die Projektansicht (Bild 1.3):

Die **Portalansicht** bietet eine aufgabenorientierte Sicht auf die installierten Werkzeuge. Auf der linken Seite sind die Portale für die Werkzeuge aufgelistet und in der Mitte stehen die Aktionen zum ausgewählten Portal. Das Fenster auf der rechten Seite zeigt die Objekte, die mit der ausgewählten Aktion bearbeitet werden können. Falls es erforderlich ist, wird zur Bearbeitung des ausgewählten Objekts zur Projektansicht gewechselt.



**Bild 1.3** Bedienoberflächen im TIA Portal. Portalansicht und Projektansicht

Die **Projektansicht** bietet eine objektorientierte Ansicht mit mehreren Fenstern, deren Inhalt je nach ausgeführter Tätigkeit wechselt. In der *Gerätekonfiguration* steht der Arbeitsbereich mit dem zu konfigurierenden Gerät im Mittelpunkt. In der Gerätesicht wird der Baugruppenträger mit den bereits platzierten Baugruppen gezeigt. Ein weiteres Fenster – das Inspektorfenster – enthält die Eigenschaften der selektierten Baugruppe, das Aufgaben-Fenster (Task Card) gibt Unterstützung durch den Hardware-Katalog mit den zur Verfügung stehenden Baugruppen. Die Netzsicht gestattet die Vernetzung zwischen PLC- und HMI-Stationen.

Bei der *PLC-Programmierung* bearbeiten Sie im Arbeitsbereich den ausgewählten Baustein. Wiederum sehen Sie im Inspektorfenster die Eigenschaften des markierten Objekts und können diese einstellen. Das Aufgabenfenster enthält diesmal den Programmelemente-Katalog mit den zur Verfügung stehenden Anweisungen und Funktionen. Ähnliches gilt für die Bearbeitung der PLC-Variablen, beim Online-Programmtest durch Beobachtungstabellen oder beim Projektieren eines HMI-Geräts.

Und stets haben Sie die *Projektnavigation* im Blickfeld. Sie enthält alle Objekte des STEP-7-Projekts. So können Sie jederzeit ein Objekt, beispielsweise einen Programmbaustein oder eine Beobachtungstabelle, auswählen und dieses Objekt mit den entsprechenden Editoren, die beim Öffnen des Objekts automatisch starten, bearbeiten.

### **Allgemeine Vorgehensweise für die Erstellung eines Steuerungsprogramms**

Alle für die Steuerung und Bedienung der Maschine oder Anlage notwendigen Automatisierungsdaten sind in einem *Projekt* zusammengefasst. Ein Projekt ist eine Ordnerstruktur im Dateisystem des Programmiergeräts.

Mit STEP 7 legen Sie ein neues Projekt an und fügen ein Automatisierungsgerät (eine Station) hinzu. Dabei bestimmen Sie den Typ des Controllers, den Sie für Ihr Vorhaben benötigen. Je nach Bedarf erweitern Sie die Station mit zusätzlichen Baugruppen, die Sie dem Hardware-Katalog entnehmen. Mit dem Konfigurationseditor stellen Sie alle Hardware-Parameter ein, beispielsweise das Verhalten der Zentralbaugruppe beim Einschalten der Versorgungsspannung oder die Adressen der Baugruppen. Im weiteren Verlauf der Projektbearbeitung können Sie jederzeit zusätzliche Baugruppen hinzustecken, vorhandene Baugruppen aus der Konfiguration entfernen oder die Einstellungen der Baugruppen ändern.

Bei der Hardware-Konfiguration erhält jeder Ein- und Ausgang eine Adresse, die sich aus der Adresse der Signalbaugruppe ableitet. Dieser Absolutadresse ordnen Sie in der PLC-Variablen-tabelle einen Namen (eine Symboladresse) zu. Der Programmeditor arbeitet primär mit Symboladressen, die er – falls keine (Anwender-) Symboladressen vergeben werden – mit laufender Nummer generiert. Diese automatisch generierten Symboladressen sagen nichts über die Bedeutung des Eingangs oder Ausgangs aus. Es wird empfohlen, nur aussagekräftige Namen zu verwenden, denn dadurch sinkt die Fehleranfälligkeit und das Anwenderprogramm ist leichter lesbar.

Das Steuerungsprogramm (das Anwenderprogramm) besteht aus einzelnen Unterprogrammen, die man *Bausteine* nennt. Alle Anwenderbausteine sind im Ordner *Programmbausteine* gespeichert. Beim Anlegen einer Station wird in diesen Ordner ein Organisationsbaustein mit der Nummer 1 (OB 1) und der Symboladresse *Main* (Hauptprogramm) eingefügt. Organisationsbausteine sind die Schnittstelle zwischen dem Betriebssystem der CPU und dem Anwenderprogramm. Für die Strukturierung des Anwenderprogramms gibt es zusätzlich Funktionsbausteine und Funktionen und für die Anwenderdaten die Datenbausteine. Ein Doppelklick auf einen Baustein öffnet ihn und Sie können das Bausteinprogramm eingeben. Die Programmiersprache für einen Baustein können Sie frei wählen, je nachdem, welche Vorlieben Sie haben oder welche Funktion der Baustein erfüllen soll. Im Steuerungsprogramm können also alle Programmiersprachen vertreten sein.

Die benötigten Programmelemente entnehmen Sie dem Programmelemente-Katalog, dessen Inhalt sich automatisch der verwendeten Programmiersprache anpasst. Dazu „ziehen“ Sie mit der Maus das Programmelement aus dem Katalog in den geöffneten Baustein („drag and drop“). Anschließend beschriften Sie das Programmelement mit den gewünschten Variablen. Das sind die (globalen) Variablen aus der PLC-Variablen-tabelle (Ein- und Ausgänge) und die (lokalen) Variablen, die Sie im Baustein selbst definieren, zum Beispiel Zwischenspeicher für temporäre Ergebnisse oder Bausteinparameter für die Übergabe von Werten an das Bausteinprogramm.

Sie können jederzeit die Projektierung oder Programmierung unterbrechen, die eingegebenen Daten speichern und die Arbeiten zu einem späteren Zeitpunkt fortsetzen. Es wird immer das komplette Projekt gespeichert. Dabei spielt es keine Rolle, ob das eingegebene Programm und die Daten vollständig oder in sich konsistent sind.

Ein Projekt können Sie „offline“ – das heißt ohne Steuerungs-Hardware – erstellen. Zum Testen und Betreiben an der Maschine müssen Sie die Projektdaten in die Steuerung „laden“. Beim Laden werden die Projektdaten „übersetzt“. Dabei wird der Code erzeugt, den der Steuerungsprozessor bearbeiten kann. Jetzt muss das Anwenderprogramm vollständig und in sich konsistent sein, sonst wird der Ladevorgang nicht ausgeführt. Sie können eine Übersetzung auch während der Programmerstellung anstoßen und so beispielsweise prüfen, ob der bis dahin eingegebene Programmcode konsistent ist. Mit einer im Programmiergerät simulierten (Software-)Station können Sie auch einem „Vorabtest“ durchführen.

STEP 7 stellt Ihnen zum Testen und Fehlersuchen umfangreiche Werkzeuge zur Verfügung. So können Sie beispielsweise den Verlauf der Signalzustände im bearbeiteten Programm beobachten, Variablenwerte vorgeben und vom getesteten Programm Zwischensicherungen erstellen. Nach der Inbetriebnahme können Sie das Anwenderprogramm in komprimierter Form archivieren.

Ein Projekt muss mindestens eine Station enthalten. Das kann eine PLC-, eine HMI- oder eine PC-Station sein. In einem Projekt können sich auch mehrere Stationen unterschiedlicher Art befinden. Mit der Netzprojektierung erstellen Sie dann die Kommunikationsverbindungen für den Datenaustausch zwischen den Stationen.

## 1.3 Datenhaltung im SIMATIC-Automatisierungssystem

Die Automatisierungsdaten liegen im Automatisierungssystem an verschiedenen Speicherplätzen. Da ist zuerst einmal das Programmiergerät. Auf dessen Festplatte werden in einem STEP-7-Projekt sämtliche Automatisierungsdaten gespeichert. Beim Konfigurieren, Projektieren und Programmieren bearbeiten Sie die Projektdaten mit STEP 7 im Hauptspeicher des Programmiergeräts (Bild 1.4).

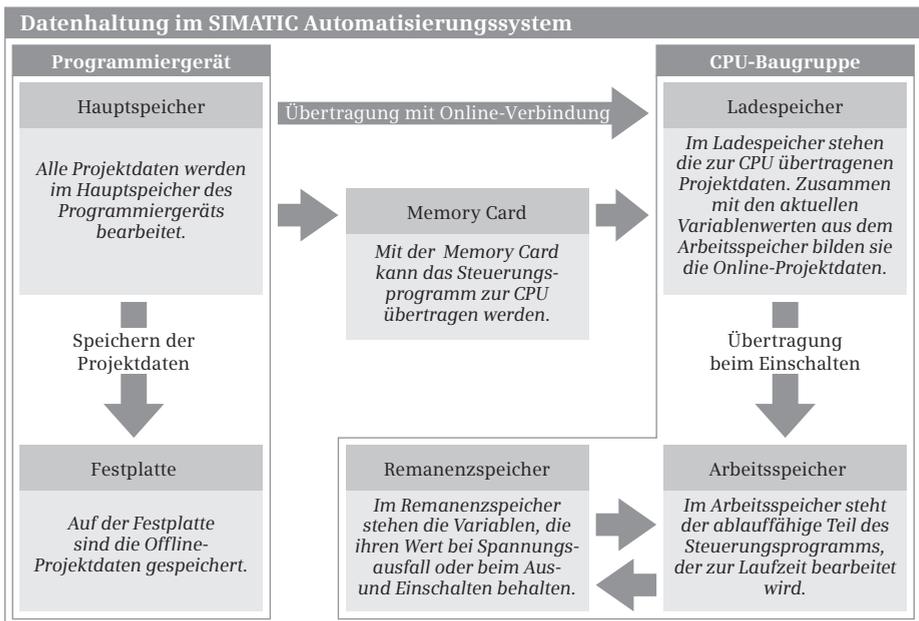
Die Automatisierungsdaten im Programmiergerät sind die *Offline-Projektdaten*. Wenn STEP 7 die Automatisierungsdaten entsprechend aufbereitet („übersetzt“) hat, können sie in ein angeschlossenes Automatisierungsgerät geladen werden. Die in den Anwenderspeicher der CPU-Baugruppe geladenen Daten sind die *Online-Projektdaten*.

Die Übertragung der Projektdaten zwischen Programmiergerät und CPU-Baugruppe kann mithilfe der SIMATIC Memory Card erfolgen. Der Normalfall ist eine Online-Verbindung zum Übertragen, Testen und Diagnostizieren.

### Anwenderspeicher

Der Anwenderspeicher in der CPU-Baugruppe besteht aus dem Ladespeicher, dem Arbeitsspeicher und dem Remanenzspeicher.

Im **Ladespeicher** steht das gesamte Anwenderprogramm einschließlich Konfigurationsdaten (Systemdaten). Reicht der Umfang des internen, auf der CPU-Bau-



**Bild 1.4** Datenhaltung im SIMATIC-Automatisierungssystem

gruppe liegenden Ladespeichers nicht aus, kann er auf die Memory Card „verlagert“ werden (externer Ladespeicher). 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 Anwenderprogramm bearbeitet. Datenbausteine, die beispielsweise Rezepte enthalten, können als „nicht ablaufrelevant“ gekennzeichnet werden und werden dann nicht in den Arbeitsspeicher übertragen. Mit Systembausteinen kann vom Anwenderprogramm aus auf diese Datenbausteine zugegriffen werden. Wie das Anwenderprogramm aus Bausteinen strukturiert ist, entnehmen Sie dem Kapitel 1.6 „Bearbeitung des Anwenderprogramms“ auf Seite 35.

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“ Anwenderprogramm wird im Arbeitsspeicher bearbeitet.

Beim Zurückladen des gesamten Anwenderprogramms in das Programmiergerät werden die Bausteine aus dem Ladespeicher geholt, ergänzt um die aktuellen Werte der Daten aus dem Arbeitsspeicher.

Der **Remanenzspeicher** enthält die als remanent definierten Merker und Datenvariablen. Die Werte im Remanenzspeicher bleiben nach einem Spannungsausfall oder beim Aus- und Einschalten der Versorgungsspannung erhalten. Die Werte werden gelöscht bei einem Umräumen oder beim Zurücksetzen der CPU-Baugruppe auf die Werkseinstellungen.

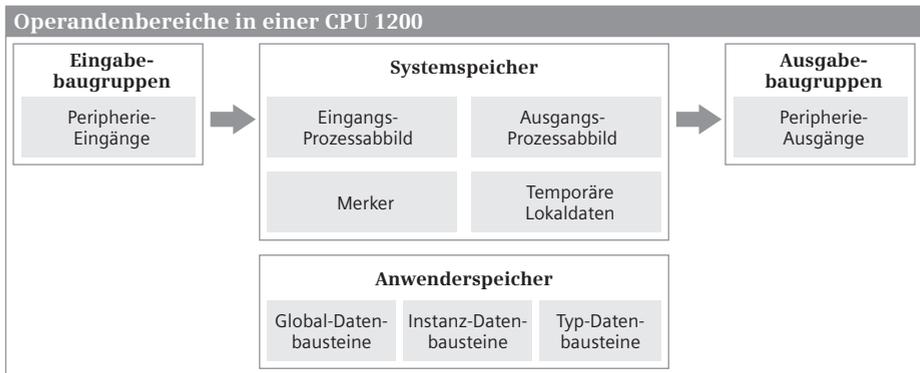
Der **Systemspeicher** enthält die Prozessabbilder für die Ein- und Ausgänge. Das sind Kopien der Ein- und Ausgangssignale von den Baugruppen. Ebenfalls im Systemspeicher liegen die Operandenbereiche Merker und die temporären Lokaldaten. Die temporären Lokaldaten sind Zwischenspeicher für die Programmbearbeitung in den Bausteinen des Anwenderprogramms.

Die Memory Card kann als *Übertragungskarte* das Anwenderprogramm in den CPU-Speicher übertragen oder als *Programmkarte* der externe Ladespeicher sein. Bei der Anwendung als Programmkarte muss die Memory Card zur Laufzeit in der CPU-Baugruppe gesteckt bleiben.

### Operandenbereiche

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. CPU 1200 stellt für diese variablen Werte die im Bild 1.5 gezeigten Speicherbereiche zur Verfügung.

- ▷ *Peripherie-Eingänge* sind die Speicherbereiche auf den Eingabebaugruppen. Sie sind die direkte Schnittstelle zur gesteuerten Maschine, um z. B. die Einstellung von Bedienelementen oder Sensoren abzufragen.



**Bild 1.5** Operandenbereiche in einer PLC-Station

- ▷ *Eingänge* sind ein Abbild der Peripherie-Eingänge im Systemspeicher der CPU. Mit ihnen arbeitet im Normalfall das Anwenderprogramm, wenn Signalzustände der Maschine abgefragt und verknüpft werden sollen. Die Gesamtheit der Eingänge ist das Eingangs-Prozessabbild.
- ▷ *Peripherie-Ausgänge* sind die Speicherbereiche auf den Ausgabebaugruppen. Sie sind die direkte Schnittstelle zur gesteuerten Maschine, um z. B. Anzeigen, Ventile oder Schütze zu steuern.
- ▷ *Ausgänge* sind ein Abbild der Peripherie-Ausgänge im Systemspeicher der CPU. Mit ihnen arbeitet im Normalfall das Anwenderprogramm, wenn die Ergebnisse der Steuerungsfunktionen ausgegeben werden sollen. Die Gesamtheit der Ausgänge ist das Ausgangs-Prozessabbild.
- ▷ *Merker* sind ein Speicherbereich im Systemspeicher der CPU, der als globaler Zwischenspeicher bevorzugt für Binärsignale verwendet wird.
- ▷ *Temporäre Lokaldaten* sind Speicherbereiche, die die CPU einem Codebaustein während der Bearbeitung zuweist. Hier kann das Programm im Baustein Signalzustände und Zahlenwerte temporär zwischenspeichern, die mit dem Bearbeitungsende des Bausteins ihre Gültigkeit verlieren.
- ▷ *Daten* sind Variablen im Anwenderspeicher, die in Datenbausteinen mit verschiedenen Strukturen zusammengefasst sind. Ein Datenbaustein, der einem Codebaustein zugeordnet ist (Instanz-Datenbaustein), enthält den Operandenbereich *statische Lokaldaten*.

Weitergehende Beschreibungen finden Sie in den Kapiteln 6.2 „Globale Operanden, Konstanten und Adressierung“ auf Seite 199, 6.6.6 „Operandenbereich temporäre Lokaldaten“ auf Seite 240 und 12.5.3 „Operandenbereich Daten“ auf Seite 531.

### Globale und lokale Variablen

Variablen enthalten die im Anwenderprogramm bearbeiteten Werte. Man kann sie grob in zwei Kategorien einteilen: in globale und lokale Variablen.