

Speicherprogrammierbare Steuerungen für die Fabrik- und Prozessautomation

Strukturierte und objektorientierte SPS-Programmierung, Motion Control, Sicherheit, vertikale Integration

Bearbeitet von
Matthias Seitz

4., neu bearbeitete Auflage 2015. Buch. 351 S. Gebunden

ISBN 978 3 446 44273 3

Format (B x L): 16,4 x 24,1 cm

Gewicht: 662 g

Weitere Fachgebiete > Technik > Technische Instrumentierung > Mess- und Automatisierungstechnik

schnell und portofrei erhältlich bei


DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.



Leseprobe

Matthias Seitz

Speicherprogrammierbare Steuerungen für die Fabrik- und
Prozessautomation

Strukturierte und objektorientierte SPS-Programmierung, Motion Control,
Sicherheit, vertikale Integration

ISBN (Buch): 978-3-446-44273-3

ISBN (E-Book): 978-3-446-44418-8

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-44273-3>

sowie im Buchhandel.

Vorwort

Der Begriff „Speicherprogrammierbare Steuerung (SPS)“ mag dem einen oder anderen ein wenig kompliziert oder sogar antiquiert vorkommen. Er beschreibt aber den *Klassiker* unter den Automatisierungsgeräten, der millionenfach in Produktionsbetrieben eingesetzt wird und erheblich zu einer hocheffizienten, erfolgreichen Industrieproduktion beiträgt.

Das vorliegende Buch will dem Leser einen Leitfaden an die Hand geben, wie er typische Aufgaben der Fabrik- und Prozessautomatisierung mit speicherprogrammierbaren Steuerungen lösen kann. Dabei wird sowohl der Systemaufbau als auch die Programmierung von speicherprogrammierbaren Steuerungen behandelt.

Zunächst wird in Kapitel 2 der prinzipielle *Hardware-Aufbau* von SPSen erläutert. Im weiteren Verlauf des Buchs wird die Zusammenschaltung mehrerer SPSen mit PC-basierten Systemen betrachtet. Damit wird einerseits die Bedienung und Beobachtung des automatisierten Prozesses möglich. Andererseits können dadurch ganze Produktionsbetriebe automatisiert werden (siehe Kapitel 9). Für die Automatisierung von Fertigungsstraßen, in denen schnelle synchrone Bewegungen von Robotern, Förderbändern und Bearbeitungsmaschinen ausgeführt werden müssen, wird in Kapitel 6 das Zusammenspiel von SPSen mit Robot-Control-, Machine-Vision- und Motion-Control-Systemen behandelt. Somit erhält der Leser einen umfassenden Überblick über die gängigen Systemstrukturen industrieller Automatisierungssysteme.

Im Mittelpunkt des Buchs steht jedoch die *Programmierung* von Automatisierungssystemen. Das Buch versucht, den Stoff anwendungsorientiert zu vermitteln. Dabei wird nur am Rande auf die Programmiersysteme einzelner Hersteller und deren Programmiersyntax eingegangen, sondern im Vordergrund steht die Umsetzung einer verbalen Aufgabenstellung in SPS-Software. Hierfür wird eine Systematik vorgestellt, die

- eine objektorientierte Softwarestrukturierung vorschlägt (siehe Kapitel 3),
- verschiedene Entwurfsverfahren zum Logikentwurf beschreibt (siehe Kapitel 4 – 7),
- und die Programmierung *strukturiert* (siehe Kapitel 4+5) oder *objektorientiert* vornimmt (siehe Kapitel 7).

Um die Software zuverlässig und nachvollziehbar entwickeln zu können, wurden *Entwurfsverfahren* aus der Informatik für SPS-Programme angepasst:

- Modellierung der Software durch Unified Modeling Language (UML),
- Automatenentwurf für Verknüpfungssteuerungen (siehe Kapitel 4),
- Petri-Netze zur Koordination paralleler Ablaufsteuerungen (siehe Kapitel 5),
- Structured Analysis and Design Technique (SADT) zur Programmierung flexibler Rezeptsteuerungen (siehe Kapitel 7).

Um den hohen Sicherheitsanforderungen in Produktionsbetrieben gerecht zu werden, behandelt Kapitel 8 Methoden zur Risikoeinschätzung, Fehlervermeidung und Fehlerbeherrschung.

Da zukünftige Produktionskonzepte unter dem Schlagwort „*Industrie 4.0*“ auf noch höhere Flexibilität abzielen, wurde in diesem Buch die Entwicklung modularer und allgemein verwendbarer Bausteine für Automatisierungssoftware in den Mittelpunkt gestellt. Im Ausblick in Kapitel 10 werden weitere Anforderungen diskutiert, durch die sich die „*SPS4.0*“ nahtlos in moderne Industriestrukturen einfügen kann.

Zahlreiche Beispiele, Übungen und Wiederholungsfragen unterstützen den Leser beim Erlernen der erläuterten Methoden und Werkzeuge. Alle Beispiel- und Übungsprogramme sind systemneutral konzipiert, d. h. sie können prinzipiell in jedem Programmiersystem (CoDeSys, STEP7 o. a.) so wie im Text beschrieben umgesetzt werden. Da die Firma 3S das *Programmiersystem CoDeSys* zum kostenlosen Download zur Verfügung stellt, wurden die Beispiele und Übungsaufgaben damit erstellt. Sie stehen auf der Internetseite

<http://www.es.hs-mannheim.de/sps40>

zum Download und zur Simulation mit CoDeSys zur Verfügung ebenso wie Bausteinbibliotheken für den praktischen Einsatz. Die *Internetseite* soll dem Leser als SPS-Lern- und Übungsseite dienen, um mithilfe von Frage-Antwortspielen, Videos zur Bedienungsanleitung, Beispielprogrammen und Übungsaufgaben Erfahrung im System- und Programm-entwurf für SPSen zu gewinnen.

In diesem Zusammenhang bedanke ich mich bei den Firmen 3S-Smart Software Solutions GmbH, ABB, NetXautomation, Siemens, TheImagingSource und Wonderware für die Bereitstellung von Software und Bildmaterial.

Frau M. A. Franziska Jacob und Frau Dipl.-Ing. F. Kaufmann vom Hanser Verlag danke ich herzlich für die Übernahme des Lektorats bzw. die Herstellung des Buchs.

Besonderen Dank für viele fruchtbare Diskussionen und die Durchsicht von Teilen des Manuskripts verdienen mein Vater, Herr Dipl.-Ing. M. Seitz, meine Kollegen von der Hochschule Mannheim und der Hochschule Darmstadt und nicht zuletzt Herr Dipl.-Ing. H. Peter, Laborbetriebsleiter des Instituts für Automatisierungssysteme, mit dem mich eine enge Zusammenarbeit auf dem Gebiet der SPS-Technik verbindet.

Schließlich gilt mein Dank meinen Studierenden für ihre Mitarbeit an den Programmierübungen und den vielen Lesern, die durch ihre Rückmeldungen zur Verbesserung der Darstellung und Korrektur von Fehlern beigetragen haben.

Mannheim, im Juni 2015

Matthias Seitz

Inhalt

	Verzeichnis der Abkürzungen und Symbole	12
1	Einleitung	17
1.1	Definition einer Steuerung	17
1.2	Aufgaben der Steuerungstechnik	18
1.2.1	Einsatzfelder für Steuerungen	18
1.2.2	Tätigkeitsfelder des Steuerungstechnikers	20
1.3	Steuereinrichtungen	21
1.3.1	Entwicklung der Steuerungstechnik	22
1.3.2	Stand der Technik	24
2	Aufbau und Strukturen industrieller Steuerungen	27
2.1	Aufbau einer SPS	27
2.1.1	Central Processing Unit	28
2.1.2	Ein- und Ausgangskarten	28
2.1.3	Programmiergerät (PG)	28
2.1.4	Human Machine Interface (HMI)	29
2.2	SPS-Arten	30
2.2.1	Hardware-SPS	30
2.2.2	Slot-SPS	30
2.2.3	Soft-SPS	31
2.2.4	Vor- und Nachteile PC-basierter SPSen	31
2.3	Informationsverarbeitung in der SPS	32
2.4	Konventionelle Ankopplung der Feldgeräte	33
2.4.1	Binäre Eingänge der SPS	33
2.4.2	Binäre Ausgänge der SPS	34
2.4.3	Analoge Eingänge der SPS	35
2.4.4	Analoge Ausgänge der SPS	37
2.5	Busan Kopplung der Feldgeräte	38
2.5.1	Feldbussysteme und -strukturen	38
2.5.2	Datenübertragung zwischen SPS und Feldgeräten	41
2.6	Bedienen und Beobachten	43
2.6.1	Aufbau von Bedien- und Beobachtungssystemen	44
2.6.2	Darstellung der Prozessgrafik	46

3	Strukturierte SPS-Programmierung nach IEC 61131	54
3.1	Das Softwaremodell	55
3.1.1	Steuerungskonfiguration und Ressourcen	55
3.1.2	Tasks	57
3.1.3	Programmorganisationseinheiten	60
3.1.4	Variablen	65
3.2	Das Kommunikationsmodell	67
3.2.1	Datenaustausch innerhalb eines Programms	67
3.2.2	Datenaustausch zwischen Programmen	68
3.3	Das Programmiermodell	69
3.3.1	Programmiersprachen	69
3.3.2	Anwender-Datentypen	72
3.3.3	Anwender-Funktionsbausteine	73
3.4	Strukturierte Programmierung in der Automatisierungstechnik	75
3.4.1	Analyse der User Requirements	75
3.4.2	Objektorientierte Softwarestrukturierung	77
3.4.3	Entwurf der Funktionsbausteine	79
3.4.4	Entwurf der Ansteuerprogramme	80
3.4.5	Implementierung in der SPS	82
3.4.6	Simulation des Anlagenverhaltens	83
3.4.7	Testdurchführung und -protokollierung	86
4	Verknüpfungssteuerungen	93
4.1	Entwurf von Schaltnetzen	94
4.1.1	Wahrheitstabelle	94
4.1.2	Karnaugh-Veitch-Diagramme	96
4.2	Entwurf von Schaltwerken	98
4.2.1	Speicherschaltungen mit Flip-Flops	99
4.2.2	Zählschaltungen mit Countern	101
4.2.3	Zeitschaltungen mit Timern	104
4.2.4	Automatenentwurf	105
4.3	Einzelsteuerfunktionen	114
4.3.1	Motorbausteine	115
4.3.2	Ventilbausteine	117
4.3.3	Schutzmaßnahmen	119
4.3.4	Betriebsartenkonzepte	122
4.4	Sensordatenverarbeitung	126
4.4.1	Einlesen binärer Sensordaten	126
4.4.2	Einlesen analoger Sensorsignale	127
4.5	Regelungen	129
4.5.1	Reglerbetriebsarten	130
4.5.2	Schaltende Regler	131
4.5.3	Kontinuierliche Regler	133
4.5.4	Selbsteinstellende Regler	141

5	Ablaufsteuerungen	152
5.1	Aufbau von Schrittketten	153
5.1.1	Aktionen und Transitionen	153
5.1.2	Strukturen von Schrittketten	155
5.2	Analyse und Entwurf von Ablaufketten	157
5.2.1	Erreichbarkeitsgraf	157
5.2.2	Schrittkettenentwurf aus Zustandsgraf	158
5.3	Verknüpfung zwischen SFCs und CFCs	159
5.4	Schutzfunktionen und Betriebsarten	163
5.4.1	HALT und ABBRUCH einer Schrittkette	163
5.4.2	Betriebsarten	165
5.5	Schrittkettenentwurf durch anlagenneutrale Grundfunktionen	167
5.6	Koordination paralleler Prozesse durch Petri-Netze	171
5.6.1	Modellierung paralleler Prozessabläufe durch Petri-Netze	172
5.6.2	Algebraischer Entwurf zur Koordination paralleler Prozesse ..	175
5.6.3	Programmmentwurf aus Petri-Netzen	177
6	Bewegungssteuerungen	184
6.1	Motion-Control-Systeme	185
6.1.1	Aufbau von Motion-Control-Systemen	185
6.1.2	Komponenten von Motion-Control-Systemen	187
6.1.3	Standard-Funktionsbausteine in Motion-Control-Systemen	189
6.2	Steuerung einer Bewegungsachse	190
6.2.1	Interpolation	192
6.2.2	Lageregelung	195
6.3	Steuerung von Werkzeugmaschinen	197
6.3.1	Bahnplanung durch CNC-Programmierung	197
6.3.2	Bewegungsvorgaben durch Kurvenscheiben	200
6.4	Robotersteuerungen	203
6.4.1	Kinematische Transformationen	204
6.4.2	Programmierung von Bewegungsabläufen	205
6.4.3	Bildverarbeitung zur Steuerung von Robotern	208
6.5	Bewegungsplanung für eine Fertigungszelle	213
6.5.1	Fertigungsplanung mit Vorranggraf	214
6.5.2	Schrittkettenentwurf durch Petri-Netze	216
7	Objektorientierte SPS-Programmierung	224
7.1	Einsatz von Methoden und Eigenschaften	224
7.2	Klassen und Objekte	227
7.3	Vererbung	230
7.4	Objektorientierte Ansteuerung der Feldgeräte	232
7.4.1	Ablaufsteuerungen mit Methoden und Eigenschaften	232
7.4.2	Anlagenneutrale Grundfunktionen	234
7.4.3	Ansteuerung von Interfaces	235
7.4.4	Verallgemeinerung durch abstrakte Klassen	237

7.5	Rezeptsteuerung mit polymorphen Grundfunktionen	238
7.5.1	Entwurf von Rezeptsteuerungen	239
7.5.2	Prozessanalyse	240
7.5.3	Rezeptsynthese	244
7.5.4	Flexible Automatisierungssoftware	249

8 Sicherheitskonzepte für industrielle Steuerungen 254

8.1	Gefahrenanalyse und Gegenmaßnahmen	254
8.1.1	Ereignisbaumanalyse	255
8.1.2	Fehlerbaumanalyse	255
8.1.3	Risikoanalyse	256
8.1.4	Gegenmaßnahmen	257
8.1.5	Safety Integrity Levels (SIL)	258
8.1.6	Berechnung der Ausfallwahrscheinlichkeiten	259
8.2	Sicherheitsgerichtete Steuerungen	261
8.2.1	Redundanz und Diversität	261
8.2.2	Aufbau sicherheitsgerichteter SPSen (SSPSen)	263
8.2.3	Sicherheitsgerichtete Feldbussysteme	265
8.3	Engineering zuverlässiger Steuerungen	267
8.3.1	Gute Automatisierungspraxis	267
8.3.2	Planung und Projektierung	268
8.3.3	Realisierung der Hard- und Software	272
8.3.4	Inbetriebnahme und Verifizierung	277
8.3.5	Wartung und Instandhaltung	280

9 Vertikale Integration betrieblicher Abläufe 284

9.1	Horizontale Integration der Feld- und Steuerungsebene	285
9.1.1	Vernetzung mit Feldbus	285
9.1.2	Werkzeuge zur Netzwerkintegration	286
9.1.3	Vernetzung mit Ethernet	287
9.1.4	Vernetzung mit Industrial Ethernet	291
9.2	Vertikale Integration mit den höheren Ebenen	293
9.2.1	Datenaustausch durch OPC	293
9.2.2	Internet in der Automatisierungstechnik	296
9.3	Prozessleitsysteme	301
9.4	Integrierte Betriebsführung	303
9.4.1	Betriebsdateninformationssysteme	304
9.4.2	Produktionsplanung und -steuerung	308
9.4.3	Ausführung von Steuerrezepten	310
9.4.4	Lagerverwaltungssysteme	315
9.4.5	Supply Chain Management	316

10	Fazit	321
	10.1 Systematischer Programmentwurf	321
	10.2 Modularer Systementwurf	323
	10.3 Ausblick – SPS 4.0	324
	Anhang	327
A	Wichtige Funktionsbausteine und Funktionen nach IEC 61131	327
B	Bibliotheken wichtiger Anwenderfunktionsbausteine	330
	B1 Bausteine der automation.library	330
	B2 Bausteine der automationOOP.library	332
	B3 Bausteine der SimAT.library	333
	B4 Weitere Bibliotheken	334
C	Umsetzung der strukturierten Programmierung mit STEP 7	335
D	Fachbegriffe Deutsch/Englisch	336
	Literatur und Links	341
	Index	345
	Hinweise zur Internetseite www.es.hs-mannheim.de/sps40	356

2

Aufbau und Strukturen industrieller Steuerungen

In einem Automatisierungsprojekt ist die Hardware des Steuerungssystems so auszuwählen, dass sie auf die Anforderungen der Aufgabe zugeschnitten ist. Hierfür ist es notwendig, den *Aufbau* der wichtigsten Komponenten und die Strukturen des Steuerungssystems zu kennen.

Deshalb werden im Folgenden Aufbau und Strukturen von Automatisierungssystemen und ihre grundsätzliche *Informationsverarbeitung* beschrieben. Im Anschluss daran werden Verdrahtungskonzepte vorgestellt, um Sensoren und Aktoren entweder über Gleichstromkreise oder mit Feldbustechnik an die Steuerung anzuschließen.

Schließlich wird das Zusammenspiel zwischen Steuerungen und *Prozessvisualisierungssystemen* erläutert, durch die der Anlagenfahrer den Prozess bedienen und beobachten kann.

■ 2.1 Aufbau einer SPS

Eine klassische SPS (engl. PLC) besteht aus den in Bild 2.1 dargestellten Hardwaremodulen. Die Stromversorgungskarte PS (Power Supply) wandelt die Netzspannung in eine 24-V-Gleichspannung, mit der die Elektronik der SPS versorgt wird.

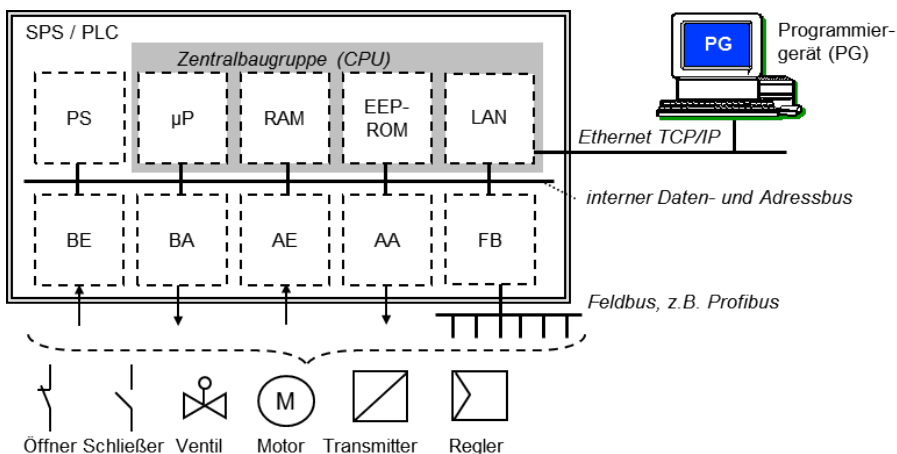


Bild 2.1 Hardwareaufbau einer SPS mit Stromversorgung (PS), µP, RAM, EEPROM, Ethernet-Schnittstelle (LAN) auf einer Zentralbaugruppe und binären Ein- (BE) und Ausgangskarten (BA) sowie analogen Ein- (AE) und Ausgangskarten (AA) und Feldbusschnittstellen (FB)

2.1.1 Central Processing Unit

Das Kernstück ist die Zentralbaugruppe oder CPU (Central Processing Unit) mit einem Mikroprozessor (μP) zum Ausführen der Steuerungsprogramme. Die aktuell im μP abgearbeiteten Programme stehen online im Arbeitsspeicher (Random Access Memory, RAM) zur Verfügung. Außerdem werden im RAM die von den Programmen benötigten Variablenwerte gespeichert. Der Speicherinhalt des RAMs geht aber bei Spannungsausfall verloren.

Anstatt einer Festplatte besitzt die SPS ein EEPROM (Electrically Erasable Programmable Read Only Memory), in dem alle Anwender- und Betriebssystemprogramme wie in einem Archiv offline gespeichert werden können. Der EEPROM ist häufig als steckbare Memory-Card, wie man sie von Digitalkameras kennt, realisiert. Bei Ausführung eines Programms wird es vom EEPROM in den RAM kopiert, wo die CPU schnellen Zugriff auf das Programm hat. Der Speicherinhalt des EEPROMs bleibt bei Spannungsausfall erhalten [100].

Die Auswahl der CPU erfolgt gemäß der Anforderungen der jeweiligen Anwendung. Auswahlkriterien sind z. B. der verfügbare Arbeitsspeicher, die Verarbeitungsgeschwindigkeit von Binärbefehlen (AND, OR o. ä.), der Umfang an E/A-Adressen und die Anzahl möglicher Ethernet-TCP/IP-Verbindungen zwischen SPSen und PCs [85].

2.1.2 Ein- und Ausgangskarten

Eine weitere Besonderheit einer SPS sind die Ein-/Ausgangskarten zum Einlesen von Sensorinformationen und zum Ausgeben von Befehlen an die Aktoren. Dabei wird ein Sensor oder Aktor mit zwei Leitungen zum Aufbau eines *Gleichstromkreises* an eine binäre bzw. analoge Ein-/Ausgangskarte angeschlossen. Für die Ankopplung busfähiger Sensoren und Aktoren verfügt die SPS über Feldbusschnittstellenkarten (siehe Abschnitte 2.4 und 2.5).

Der *interne Daten- und Adressbus* verbindet die Module der SPS und ermöglicht den Datenaustausch zwischen ihnen. Das Programmiergerät (PG) wird in einem Local Area Network (LAN) mit einem Ethernetkabel an die SPS angekoppelt. Ethernet ermöglicht auch die Ankopplung an andere SPSen oder an ein Visualisierungssystem zum Bedienen und Beobachten des Prozesses (siehe Abschnitt 2.6).

2.1.3 Programmiergerät (PG)

Das Programmiergerät ist ein PC oder Notebook. Es dient hauptsächlich zur Erstellung der Anwenderprogramme, also zur Programmierung des zu automatisierenden Prozesses. Hierfür befindet sich auf dem Programmiergerät eine spezielle Software, die sogenannte Programmierumgebung, die das Programmieren in festgelegten Programmiersprachen ermöglicht. Die damit vom Anwender erstellten Programme werden in SPS-spezifischen Code übersetzt und an die SPS übertragen.

Dieses *Laden* der Programme in die SPS verursacht im Allgemeinen, dass die laufenden Programme gestoppt und die Variablen neu initialisiert werden. Deshalb ist entweder an der SPS ein Schlüsselschalter angebracht oder das Programmiersystem ist mit einem Passwort gesichert, damit der Zugriff auf die SPS nur für einen autorisierten Bediener möglich ist. Um zu verhindern, dass z. B. während einer Reparatur unerwünscht von der SPS ein

Programm abgefahren wird, verfügen die meisten SPSen über einen RUN/STOP-Wahlschalter [85]. In der Betriebsart STOP werden alle laufenden Programme angehalten und die SPS-Ausgänge stromlos geschaltet. Einige SPSen erlauben wahlweise auch einen Online-Change, bei dem nur Änderungen übersetzt und ohne Anhalten der Steuerung oder Verlust der Variablenwerte geladen werden [1].

In jedem Fall sollten die Programme zunächst auf dem Programmiergerät *simuliert* und ggf. korrigiert werden, bevor sie in die SPS geladen werden und eine eventuell empfindliche Anlage steuern.

2.1.4 Human Machine Interface (HMI)

Außer dem Programmiersystem laufen auf dem PC auch Programme, die es Menschen erlauben die weitgehend automatisierte Anlage zu bedienen und zu beobachten. Diese Programme dienen als Schnittstelle zwischen der Steuerung und dem Bediener (Human Machine Interface, HMI). Sie zeigen den Zustand der Anlage an, beispielsweise Behälterfüllstände, aktive Pumpen, geöffnete Rohrleitungswege etc. Außerdem ermöglichen sie es dem Bediener, einzelne Geräte, wie z. B. Pumpen, Ventile oder Regler, von Hand zu aktivieren und somit manuell in den Prozess einzugreifen. Aufbau und Funktionsweise solcher HMIs werden in Abschnitt 2.6 beschrieben.

Im Zuge immer komplexerer Anwendungen entstehen somit Systemstrukturen wie in Bild 2.2. Diese bestehen beispielsweise aus mehreren SPSen, an die zahlreiche Feldgeräte angeschlossen sind. Außerdem kommunizieren sie mit dem Programmiergerät und den HMIs über ein industrielles Ethernet.

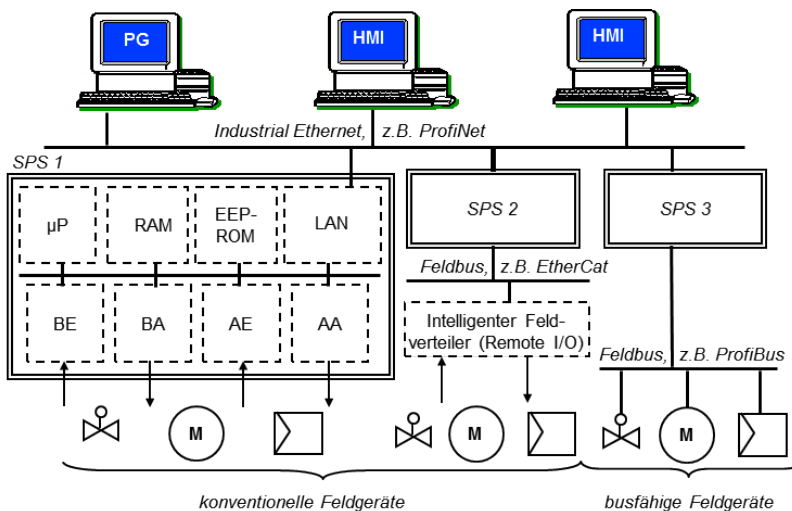


Bild 2.2 Struktur eines modernen Steuerungssystems mit Feldbustechnik oder Remote-I/O, mehreren SPSen sowie Anzeige- und Bedienkomponenten (Human Machine Interfaces, HMIs)

Die Feldgeräte können dabei auf unterschiedliche Arten an die SPS angekoppelt werden:

- *konventionell* durch Kupferdrahtleitungen (siehe Bild 2.2 links),
- über *Feldbus*, was jedoch busfähige Feldgeräte erfordert (siehe Bild 2.2 rechts),
- über eine *dezentrale Peripherie* (Remote-I/O), die einerseits über Feldbus mit der SPS verbunden ist und andererseits nicht busfähige Feldgeräte über dezentrale E/A-Karten anbindet (siehe Bild 2.2 Mitte).

■ 2.2 SPS-Arten

Grundsätzlich unterscheidet man drei verschiedene Aufbauarten bei SPSen, nämlich als:

- Hardware-SPS,
- Slot-SPS und
- Soft-SPS.

2.2.1 Hardware-SPS

Der im vorigen Abschnitt beschriebene Aufbau einer SPS bezieht sich auf die klassische Aufbauform einer Hardware-SPS. Ihre Komponenten sind als Einsteckkarten in einem *Schaltschrank* oder *Gehäuse* angeordnet. Über einen Rückwandbus sind die Einsteckkarten miteinander verbunden.

Im Allgemeinen gibt es eine gemeinsame Zentralbaugruppe (CPU) mit μ P, RAM und EEPROM. Außerdem gibt es Einsteckkarten zur Feldbusankopplung oder mit klassischen Eingangs- und Ausgangs-Kanälen zur Verdrahtung der Sensoren und Aktoren. Eine Hardware-SPS bedarf eines externen PCs als Programmiergerät (vgl. Bild 2.1).

2.2.2 Slot-SPS

Eine Slot-SPS ist eine *Einsteckkarte* für den PC, die alle Module einer SPS enthält. Anstatt einer CPU besitzt sie einen Co-Prozessor, auf dem ein eigenes multitaskingfähiges Betriebssystem mit einem multi-ported RAM (für PC und SPS zugänglicher, gemeinsamer Speicher) läuft. Außerdem befindet sich auf der Slot-SPS eine Feldbuskopplung zur Anbindung der Sensoren und Aktoren.

Im Grunde nutzt also die Slot-SPS lediglich die Stromversorgung des PCs. Durch den multi-ported RAM können die CPU des PCs und der Co-Prozessor der Slot-SPS aber gleichzeitig auf den RAM zugreifen, was den Datenaustausch zwischen beiden vereinfacht.

2.2.3 Soft-SPS

Eine Soft-SPS ist dagegen reine *Software*, die komplett auf der CPU eines PCs läuft. Der PC kann ein handelsüblicher Windows-PC, ein Notebook oder ein Industrie-PC sein. Vielen Industrie-PCs sieht man den Unterschied zur Hardware-SPS nicht an. Sie sind jedoch leistungsfähiger und bieten die typischen Schnittstellen über USB- oder VGA-Ports, Webserver etc.

Auf der Internetseite zu diesem Buch¹ ist die Soft-SPS CoDeSys der Firma 3S-Smart Software Solutions verfügbar, mit der der Leser SPS-Programme erstellen und in Simulation testen kann. Zur Ankopplung realer Sensoren und Aktoren wäre eine Einsteckkarte zur Feldbuskopplung notwendig, die mit einem Prozessor zur Buskommunikation ausgestattet ist.

2.2.4 Vor- und Nachteile PC-basierter SPSen

Die Vorteile der SPS im PC ergeben sich hauptsächlich dadurch, dass die rasante Entwicklung der PC-Leistung für SPSen genutzt werden kann:

- PC-basierte Steuerungen erreichen höhere *Verarbeitungsgeschwindigkeiten* als Hardware-SPSen [14].
- Ein PC kann zur Steuerung, Programmierung und Visualisierung verwendet werden. Somit ergeben sich *preisgünstigere*, einfachere und durchgängige Systemstrukturen, mit denen der Anwender gewohnt ist umzugehen.
- Es entstehen *offenere* Systeme, weil der Datenaustausch auf einer einheitlichen Plattform unter Windows standardisiert wird. Somit wird die Ankopplung von Bedien- und Beobachtungssystemen sowie von übergeordneten Planungssystemen an die SPS vereinfacht.

Neben diesen Vorteilen muss der PC aber auch die industriellen *Anforderungen* erfüllen, wie z.B. Robustheit des Betriebssystems, Echtzeitfähigkeit, Kommunikationsstandards für E/A-Anbindung, Funktionssicherheit, EMV (elektromagnetische Verträglichkeit), Ex-Schutz (Explosionsschutz in Chemiebetrieben), Temperaturüberwachung und USV (unterbrechungsfreie Spannungsversorgung).

Insbesondere bei einer Soft-SPS besteht die Gefahr, dass die SPS-Programme von anderen im PC ablaufenden Programmen gestört werden, was im industriellen Einsatz nicht toleriert werden kann. Durch Einsatz von Industrie-PCs mit Echtzeit-Kernel-Betriebssystemen, die die Abarbeitung der Programme in Echtzeit gewährleisten, werden diese Anforderungen zunehmend erfüllt [93]. Dennoch ist die konventionelle SPS-Hardware für kleinere Steuerungsaufgaben, z.B. zur Gebäudeautomatisierung, häufig kostengünstiger, wenn kein Visualisierungssystem zusätzlich zur SPS gebraucht wird.

¹ www.es.hs-mannheim.de/sp40 (Weitere Hinweise zur Installation von CoDeSys und zum Inhalt der SPS-Lern- und Übungsseite finden Sie auf der letzten Seite des Buchs)

■ 2.3 Informationsverarbeitung in der SPS

Die Informationsverarbeitung in einer SPS verläuft zyklisch. Die Verarbeitungsschritte lassen sich vereinfacht wie in Bild 2.3 dargestellt mit dem EVA-Prinzip beschreiben:

- *Einlesen* der Sensordaten,
- *Verarbeiten* der Informationen im SPS-Programm und
- *Ausgeben* der Stellsignale an die Aktoren.

Die Messsignale von den Sensoren werden zunächst in den Eingangskarten elektronisch angepasst. Die CPU fragt nacheinander alle Eingangskanäle ab und legt die Eingangsdaten im Arbeitsspeicher (RAM) ab. Dieser Speicherbereich wird auch *Eingangsabbild* genannt, weil die hier abgelegten Eingangsdaten nicht die aktuellen, sondern die zum Abtastzeitpunkt anliegenden Daten sind.

Die Programme werden dann von der CPU jeweils Schritt für Schritt abgearbeitet. Dabei werden die im Arbeitsspeicher abgelegten Operationen, wie LD (LOAD), AND und ST (STORE) in Bild 2.3, einzeln adressiert, interpretiert und mit den angegebenen Operatoren ausgeführt. Die Operatoren können direkte Adressen des Ein- oder *Ausgangsabbildes* sein, sollten aber im Allgemeinen als Variablen (z.B. A, B, C in Bild 2.3) deklariert werden. Prinzipiell werden in einem Programm nicht nur Daten des Ein- und Ausgangsabbildes, sondern auch Parameter und Zwischenwerte, wie Zählerstände, Sollwerte, Zustände etc., verarbeitet und im RAM gespeichert. Wenn ein Programm Stellwerte für die Aktoren der Anlage berechnet, werden diese im Ausgangsabbild des Datenspeichers abgelegt.

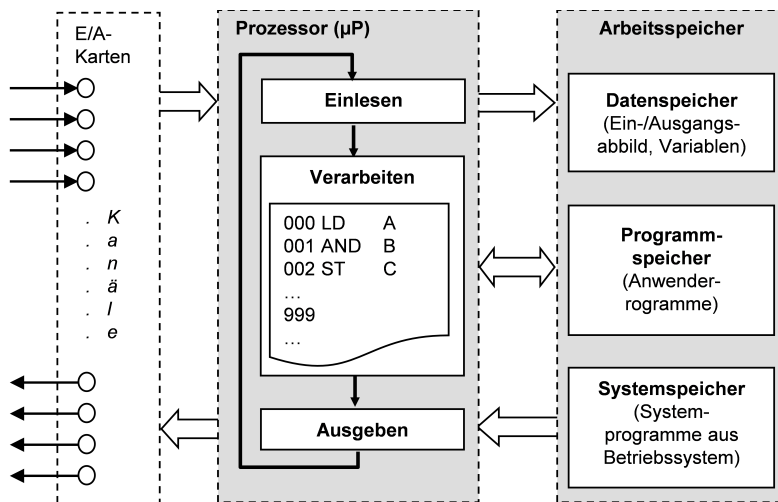


Bild 2.3 Signalverarbeitung und Arbeitsweise einer SPS

Erst nach Abarbeitung aller Programme werden die im Ausgangsabbild abgelegten Stellwerte nacheinander an die Ausgangskanäle übertragen. Dabei erfolgt wiederum eine elektronische Anpassung. Der *Arbeitsspeicher* lässt sich also wie in Bild 2.3 dargestellt in drei Teile gliedern:

- den Datenspeicher mit Ein- und Ausgangsabbild sowie den verwendeten Variablen,
- den Programmspeicher mit den aktuell abzuarbeitenden Anwenderprogrammen und
- den Systemspeicher mit den benötigten SPS-internen Systemprogrammen.

Wie Sensordaten eingelesen und Stellsignale ausgegeben werden, beschreibt der folgende Abschnitt.

■ 2.4 Konventionelle Ankopplung der Feldgeräte

Sensoren und Aktoren werden auch als Feldgeräte bezeichnet und sind häufig durch zwei Kupferleitungen an die SPS oder ein anderes Automatisierungssystem angeschlossen. Dadurch werden *Stromkreise* aufgebaut, in denen die Signale von den Sensoren zur SPS bzw. von der SPS zu den Aktoren als Gleichströme oder Gleichspannungen übertragen werden.

Für Automatisierungssysteme wie die SPS gibt es spezielle Ein-/Ausgangskarten zum Einlesen binärer oder analoger Sensordaten sowie zum Ausgeben binärer oder analoger Stellsignale.

2.4.1 Binäre Eingänge der SPS

Eine binäre Eingangskarte (BE) kann zwischen einem hohen und einem niedrigen Pegel der anliegenden Eingangsspannung unterscheiden. Wie in Bild 2.4 skizziert, wird die Eingangskarte von einer 24-V-Gleichspannung (DC) versorgt. Ist der Schalter geschlossen, liegt also – innerhalb gewisser Toleranzen – ein Spannungspegel von 24 V am Eingangskanal an. Bei geöffnetem Schalter liegt dagegen ein Spannungspegel von 0 V an.

Die Spannung wird durch ein RC-Filter von überlagertem Rauschen entzerrt. Das Filter dient auch als Verzögerung des Eingangssignals, um Störungen durch Schalterprellen zu unterdrücken. Anschließend wird das Eingangssignal durch einen Optokoppler von der restlichen Verarbeitungselektronik *galvanisch* getrennt. Somit können keine Ausgleichsströme zwischen dem Sensor und dem Steuerkreis fließen.

Schließlich wird das Eingangssignal durch einen *Schwellwertschalter* (Triggerstufe) einem High- (5 V) bzw. Low-Zustand (0 V) zugeordnet. Bei einem Spannungspegel von 5 V interpretiert die Elektronik die binäre Information als TRUE, bei einem Spannungspegel von 0 V dagegen als FALSE [8, 97]. Einige SPSen arbeiten heutzutage zur Energieeinsparung nur noch mit einem Spannungspegel von 3,3 V für den High-Zustand.

Bei einem Zustandswechsel des binären Sensors erzeugt dessen Elektronik wie in Bild 2.4 skizziert ein Signal, das die binäre Eingangskarte einliest.

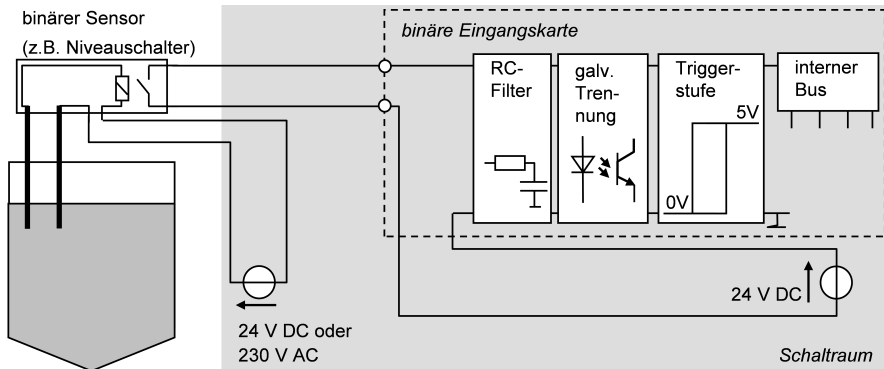


Bild 2.4: Ankopplung eines Sensors an eine binäre Eingangskarte

Beispiel 2.1: Binäre Eingangskarte

Wird der Niveauschalter in Bild 2.4 von Flüssigkeit bedeckt, zieht das Relais in seiner Elektronik an, und der Schalter im Messkreis der binären Eingangskarte wird geschlossen. Der Messkreis wird von der SPS mit 24 V Gleichspannung versorgt, die an den Eingangsklemmen der Karte anliegt. Durch Filter, galvanische Trennung und Triggerstufe wird das Signal in ein 5-V-Spannungssignal umgewandelt und in der SPS als Boole'scher Datenwert TRUE interpretiert.

2.4.2 Binäre Ausgänge der SPS

Binäre Ausgangskarten erzeugen aus den Boole'schen Ausgangsvariablen mit TRUE oder FALSE-Werten ein Spannungssignal von 5 V bzw. 0 V.

Wie in Bild 2.5 skizziert, verfügen binäre Ausgangskarten ebenso wie die Eingangskarten über Optokoppler zur galvanischen Trennung. Darüber hinaus sorgen Signalverstärker zusammen mit einem Transistor für die Anpassung an den 24-V-Pegel, denn bei einem TRUE-Signal schließt der Transistor den Stromkreis und bei einem FALSE-Signal öffnet er ihn.

Der Nachteil dieser reinen *Transistorausgänge* ist, dass sie einen Strom von bis zu 0,5 A ausgeben können, der nur zur Ansteuerung von Lampen, kleineren Schützen und Magnetventilen ausreicht. Dagegen können sog. *Relaisausgangskarten* einen Ausgangsstrom bis zu 2 A ausgeben und damit auch größere Antriebe ansteuern. Hierfür besitzen diese *Relaisausgangskarten* wie in Bild 2.5 dargestellt, zusätzlich zu dem Transistor noch ein Relais, dessen Schalter im Ausgangsstromkreis auch mit höheren Schaltspannungen, wie z. B. mit 230 V und Wechselstrom (AC), betrieben werden kann [8, 23, 97].

Beispiel 2.2: Binäre Ausgangskarte

Um den Dreiphasenmotor in Bild 2.5 zu aktivieren, steuert das Stellsignal der SPS in der binären Ausgangskarte einen Transistor an. Dadurch wird das Relais A1 mit Strom versorgt und der Relaiskontakt im externen Stromkreis geschlossen. Dies hat zur Folge, dass auch die Relais-Spule A2 mit Strom versorgt und der Hauptschütz umgelegt wird. Der Motor ist jetzt mit den drei Phasen der Drehstromversorgung verbunden und läuft an. Damit beim Ausschalten der Entladestrom des Relais den Transistor nicht zerstört, wird eine Diode in Sperrichtung parallel zu ihm geschaltet.

Index

Symbole

1002-Systeme 264
2003-Systeme 265
2004-Systeme 265
4...20-mA-Signal 35

A

Ablaufsprache 71 f., 153
Ablaufsteuerungen 152
Ablauftypicals 169
Abwasserreinigung 160
ACTION 160, 244, 247
Actions 152
Aktionen 153
Alarmmeldung 121
Alternativverzweigung 155, 158
Ampelanlage 147
Ampelsteuerung 181
Analog-/Digital-Umwandlung 36
Analogeingangskanal 128
Änderung 280
Anlagenschema 46, 268
Anlagensimulation 274, 333
Anlagenteile 272
Anti-Reset-Windup-Maßnahme 137
Anweisungsliste 69, 72
Anwender-Datentypen 72
Anwender-Funktionen 61
Anwender-Funktionsbausteine 73
Anzeige- und Bedienkomponente (ABK) 301
Arbeitsstromprinzip 122
ASi-Bus 40
Auftragsüberwachung 310
Auftragsveranlassung 310
Aufzugsteuerung 148, 180
Ausgangsabbild 32
Ausgangskanäle 57
Ausgangskarte
– analog 37
– binär 34
Ausgangsschaltnetz 99, 107

Auslagern 171
Ausschaltverzögerung 104
Autotuning 144

B

Bahninterpolation 207
Bahnplanung 187, 197
Bahnsteuerung 199
Bahnvorgabe 200
Basic Function 241
Bedienereingaben 308
Bedienphilosophie 269
Beschaffungsrechnung 317
Bestimmungszeichen 154
Betriebsarten 122, 165, 226
Betriebsartenhierarchie 125
Betriebsartenumschaltung 123
Betriebsdaten 304
Betriebsdatenauswertung 305, 318
Betriebsdatenerfassung 318
Betriebsdateninformationssystem (BDIS) 304
Bewegungssteuerungen 184
Bibliothek 272, 330
Bildaufbereitung 209
Bildaufnahme 209
Bildsegmentierung 209
Bildverarbeitung 208
Bildverarbeitungsstufen 222
Bimetallschalter 120
Black-Box-Tests 277
Box-Slope-Algorithmus 304
Brennersteuerung 260
Bridge 286
Busprotokoll
– sicherheitsgerichtet 265

C

CAM-Editor 200
CAN-Bus 40
Cause-and-Effect-Matrices 258
Cause-and-Effect-Matrix 81, 92, 119, 278

Change Management 281
 Change-Order 280
 Charge 310
 Chargenprotokoll 312
 Client-Server-Verbindung 45
 Cloud 324
 CNC-Editor 198
 Collision Avoidance 287
 Computerized Numerical Control (CNC) 185
 Condition Monitoring 307
 Continuous Function Charts (CFC) 93
 CPU 28
 CSMA/CD-Verfahren 287
 CTUD 102
 Cyber Physical Systems (CBS) 324

D

Datenbaustein 74, 79 f., 159, 234, 336
 Datenspeicher 33
 Datentyp 66
 Deadlock 157
 Device Type Manager (DTM) 55
 Digital-/Analog-Umwandlung 37
 direkte kinematische Transformation (DKT) 204
 direkte perspektivische Transformation (DPT) 210
 disjunktive Normalform (DNF) 95
 Disposition 249, 312, 319
 Diversität 262
 Drehmaschine 202
 Drehzahlregler 196
 Drehzahlveränderbare Motoren 116
 Dreipunktregler 131, 251
 Dreitankanlage 181, 251
 Durchflussregler 161

E

E/A-Kanäle 57
 E/A-Karten 33
 E/A-Liste 270
 EEPROM 28
 EIB 40
 Eigenschaft (Property) 225
 Ein-/Ausgangskanäle 81
 Eingänge 64
 Eingangsabbild 32
 Eingangskanäle 57
 Eingangskarte
 – analog 35
 – binär 33

Eingangskennlinie 127
 Eingangsschaltnetz 99, 111
 Einlagern 171
 Einschaltverzögerung 104
 Einzelsteuerfunktion 64
 Electronic Device Description, EDD 55
 Endlagenüberwachung 149
 Energieverbrauch 306
 Enterprise Ressource Planning Systems (ERP) 308
 Ereignisbaumanalyse 255
 Erreichbarkeitsgraf 157, 177
 Ethernet 28, 287
 EVA-Prinzip 32, 57
 EXTENDS 230

F

Fabrikautomatisierung 321
 Factory Acceptance Test (FAT) 278
 Fail-Safe-Prinzip 261
 Fehler
 – gefährliche 264
 – sichere 264
 Fehlerarten 119
 Fehlerbaumanalyse 255
 Fehlerbeherrschung 258
 Fehlervermeidung 257
 Fehlerwahrscheinlichkeit 259
 Feinplanung 313
 Feldbussysteme 40
 Feldgerät 77
 Feldgeräte-Liste 270
 Feldverteiler 38
 Fernbedienung 296
 Fernwartung 296, 299
 Fertigungsplanung 214
 Fertigungstechnik 19
 Fertigungszelle 213, 223
 Firewall 300
 fliegende Säge 188, 221
 Förderband 100
 Frequenzumrichter 116
 Füllstandmessung 128
 Füllstandregelung 137, 283
 Füllstandregler 160
 Function Block, FB 63
 Function Code, FC 60
 Funktionen 60
 Funktionsbaustein 63, 272, 327
 Funktionsbausteinsprache 69
 Funktionsplan 71, 72
 Funktionsprüfung 278

G

galvanische Trennung 34
GAMP-Leitfaden 267
Gateway 286
Gedächtnis 98 f.
Gefahrenabwendung 256
Gefahrenidentifikation 254
Gelenkkordinaten 204
Gepäckanlage 106
Gerätebaum 186
Gerätespezifikation 268
Gerätestammdatei (GSD) 55
Geschwindigkeitsprofil 192
Gleichlaufsysteme 188
Gray-Code 108 f.
Grobplanung 313
Grundfunktion 168, 234, 238, 241, 244, 274
Grundoperationen 238, 244
Grundstellung 163

H

H_3BIT 109
Halteglied 108
Handshake 290
Hardwarekonfiguration 83
Hardware-SPS 30
Hardwarestrukturplan 270
HAZOP-Methode 254
HMI 191, 199
Hochreallager 181
Hochregallager 74, 168, 171, 215, 272, 320
HTTP-Protokoll 298
Hub 286
Human Machine Interface (HMI) 29, 43, 166
Hysterese 131

I

IEC 61131 54
IMPLEMENTS 235
Inbetriebnahme 86
Individual Drive Function, IDF 64
Industrial Ethernet 186, 291
Industrie 4.0 24, 323
Init-Schritt 161
Inkrementalgeber 275
Inkrementalweggeber 195
In-Prozess-Kontrolle 308
Installationsprüfung 278
Instandhaltung 280

Instanzausteine 78, 247
Instanziierung 74
Integrationstest 87
Interpolation 187, 192
Interpolationstask 194
Interpolator 208
inverse Kinematik 204
inverse perspektivische Transformation (IPT) 211
isochron 186
isochroner Realtime-Kanal 292
Items 294

K

Kapazitätsplanung 310
Karnaugh-Veitch-Diagramm (KV-Diagramm) 96
Klasse 74, 77
Klassendiagramm 77, 159, 169, 275
Kommunikationsmodell 67
kompakte Lösung 177
Komplexität 96
Königswelle 200
konjunktive Normalform (KNF) 95
Kontaktplan 72
Koordinatentransformation 188
Koordinationsprogramm 179
Kurvenscheibe 190, 200 f., 221

L

Labordateninformationssysteme (LDIS) 308
Lageregelkreis 196
Lageregelung 188, 195, 220
Lagerverwaltung 320
Lagerverwaltungssystem 315
Lastenheft 21, 268
Laufmeldung 121
Laufzeitfehler 121
Lebenszyklus 280
Leiten 302
Linearinterpolation 194
Logikentwurf 113
Loop-Check 278
LS 47

M

main 336
Manipulated Variable, MV 116
manueller Stellwert 136
Manufacturing Execution System (MES) 310
Mapping-Variablen 86

Markierung 173
 Masterachse 188, 202
 Master-Slave-Protokoll 285
 Master-Slave-Verfahren 42
 Materialbedarfsplanung 309
 Materialverbrauch 306
 Maxterm 94
 MC_CamIn 190, 201
 MC_CamOut 201
 MC_MoveAbsolute 189, 191, 206
 MC_Power 189
 Mean Time to Failure (MTTF) 259
 Mehrproduktanlagen 313
 Mehrstranganlagen 313
 Mengenplanung 309
 Merkmalsextraktion 209
 Messbereichsgrenze 127
 Methoden 224
 Mikrocontroller 185
 Mikroprozessor 28
 Minterm 94
 MODES 124
 modulare Lösung 178
 Modultest 86, 278
 Moore-Automat 99, 106
 Motion-Control-System 185, 293
 Motortypen 115
 Multi-Master-Betrieb 286
 Multiplexer 109
 Multitasking 59

N

Nachstellzeit TN 138
 Näherungsschalter 102
 Netzmatrix 173
 Netzwerkvariablen 287
 Niveauschalter 18
 Nocken 201
 NS 47, 77

O

Objekte 77
 objektorientiert 332
 objektorientierte Programmierung 229
 Objektorientiertheit 77
 OPC 293, 302, 318
 OPC-Client 295
 OPC-Server 294
 OPC-Standard 46
 Optokoppler 33f.

Organisationsbausteine 336
 OSCAT 335

P

PAC 25
 Package Units 273
 Parallelverzweigung 155
 PCE
 – Kategorie 46
 – Verarbeitungsfunktion 46
 Performance Monitoring 307
 Petri-Netz 152, 172, 176, 213
 Pflichtenheft 21, 270
 PID-Regler 134
 Plant-Asset-Management-System (PAM) 307
 Plant-Lifecycle-Management-Systeme (PLM) 280
 Planung 267
 PLCopen 189, 335
 Polling-Verfahren 43
 Polumschaltbarer Motor 115
 polymorph 237
 Portalfräsmaschine 198
 Positioner 37
 Positioniersysteme 187
 Positionsermittlung 210
 Positionssensorik 195
 Power Supply 27
 Priorisierung 292
 Probability of Failure on Demand (PFD) 258, 260
 Probability of Failure per Hour (PFH) 258f.
 Process Control Engineering, PCE 46
 Process Value, PV 127
 Produktionsplanung 310
 Produktionsprogrammplanung 309
 Produktionsprüfung 279
 Produktionssteuerung 310
 Produktsicherheit 267
 Profibus 40, 57
 ProfiNet 40
 ProfiSafe 265
 Programmable Automation Controller (PAC) 209
 Programme 80
 Programmentwurf 321
 Programmiergerät 28
 Programmorganisationseinheiten, POE 60
 Programmspeicher 33
 Program Organization Units, POUs 60
 Proportionalbeiwerts KP 137
 Proxy-Server 300
 Prozessablauf 160, 274
 Prozessanalyse 239
 Prozessautomatisierung 321

Prozessgrafik 48
Prozessleitsystem (PLS) 301
prozessnahe Komponente (PNK) 301
Prozessspezifikation 268
Prozessvisualisierung 44, 125, 298
Prozesszerlegung 240
Pulsweitensteuerung 148
Punkt-zu-Punkt-Bewegung 206

Q

Qualitätskontrolle 308
Qualitätsmanagementsystem (QMS) 308

R

RAM 28
Randbedingungen 175
Rangierverteiler 38
RC-Filter 33
Realtime-Kanal 292
Redundanz 261
– aktiv 261
– passiv 261
Regeldifferenz 129
Regelgröße 129
Regelung 17
Regelventil 37, 139
Reglerbaustein 160, 233
Reglerbetriebsarten 130
Reglereinstellung 138
Reglerparameter 139
Relaisausgänge 34
Remote-I/O-System 39
Repeater 285 f.
Ressourcen 55, 186
Review 277
Rezept 238, 244
Rezeptparameter 240, 311
Rezeptsynthese 244
R+I-Schema 46
Risiko 256
Risikoanalyse 257
Risikograf 256
Robot Control, RC 185
Roboter 171, 215, 221 f.
Router 286
RS-Flip-Flop 64, 99
Rückmeldung 121
Rückwärtstransformation 204, 206
Ruhestromprinzip 79, 85, 120
Rührkesselreaktor 91

S

SADT-Methode 240
Safety Integrity Level (SIL) 258
SCARA-Roboter 203
Schadensausmaß 256
Schaltfunktion 95, 127
Schaltnetze 94
Schaltwerk 98, 158
Schlüsselschalter 28
Schnittstelle (Interface) 235
Schrittkette 152, 158, 208, 233, 241
– ABBRUCH 163
– HALT 163
– Stop 165
– unerreichbar 156
– unsicher 156
Schrittmerker 153
Schrittmotor 105
Schrittvektor 173
Schutzschalter 120
Schwingungsanalyse 141
Selbsteinstellende Regler 141
Sensor
– aktiv 35
– passiv 35
Sequential Function Chart (SFC) 71, 152
Sercos 40
SFCCurrentStep 163
Sicherheitsanforderungen 269
Sicherheitsfunktionen 258
sicherheitsgerichtete Steuerung (SSPS) 254, 261
Simulation 85
Simulationsbaustein 83, 162, 275
Simulieren 333
Singletasking 59
Site Acceptance Tests (SAT) 278
Slaveachse 188
Slot-SPS 30
Smart-Camera 209
Socket Library 290
Soft-SPS 30
Softwaremodell 55
Softwarestrukturplan 270
Softwaretest 86
Softwarstrukturierung
– objektorientiert 77
Speicherkomparator 263
Sprungantwort 139
SPS 17, 21
– Aufbau 27
– PC-basiert 31
SPS 4.0 324

SR-Flip-Flop 100
Standard-Funktionen 61
Standard-Funktionsbaustein 63
Statusvariablen 159
Stellwert 17
STEP 7 92, 336
Steuerkreis 17
Steuerrezept 311
Steuerung 17
– speicherprogrammiert 23
– verbindungsprogrammiert 22
Steuerungshardware 272
Steuerungskonfiguration 55, 186
Steuerungstechnik 17
Steuervariablen 159
Stromregelkreis 196
Strukturen
– STRUCT 73
Strukturierter Text 70, 72
Supply Chain Management 317
Swinging-Door-Algorithmus 304
Switch 286, 291
Symbolkonfiguration 295
Systementwurf 323
Systemspeicher 33

T

Taktgenerator 105
Tanklager 126
Task 57
Taskkonfiguration 60
Taskzuordnung 82
TCP/IP 288
Teamviewer 299
Teilanalagen 272
TeKa-Anlage 243, 248
Temperaturregelung 143, 150
Terminfeinplanung 310
Timer 104, 155
TIPP 165
TOF 104, 111
Token-Passing-Verfahren 286
TON 104
TP 104
Transistorausgänge 34
Transition 153
Transitionsvektor 173
Transportbandsteuerung 234
Transportfahrzeug 216
Trennanlage 159
Trennverstärker 38

Triggerstufe 33
TYP_2PT 132
TYP_AIN 127
TYP_AOUT 116, 137
TYP_BIN 78f., 126
Typical 271
TYP_IDF1 64, 78, 224
TYP_IDF2 100, 103, 230
TYP_PID 134
TYP_POL 115
TYP_PULSE 102
TYP_SMOT 105
TYP_TUNE 142

U

Überwachen 19
UML 75
UML component diagram 270
UML deployment diagram 270
UML-Zeitdiagramm 106
UML-Zustandsdiagramm 109
Umrichter 185
Use-Case Diagramm 76, 167
User Data Types, UDT 72
User Requirements 75

V

Validierung 278
Variablen 65
– global 68
Variablendeklaration 66
VAR_INPUT 67
VAR_OUTPUT 67
Ventile 117
verbindungsprogrammierte Steuerung (VPS) 259
Vererbung 224, 231
Verfahrenstechnik 19
Verfügbarkeitsanforderungen 269
Verifizierung 277
Verkehrsampele 153
Verknüpfungssteuerung 93
Verriegelung 119
Verriegelungseingänge 100
Verschlüsselung 299
vertikale Integration 285, 303
Vertikal-Knickarm-Roboter 203
Vierleiterschaltung 36
Vierwegeventil 149
Virtualisierung 326
Virtual Private Network (VPN) 299

V-Modell 20, 267
Vorhaltzeit TV 138
Vor-Ort-Bedientableau 44
Vorranggraf 214
Vorschubantriebe 197
Vorwärtstransformation 204

W

Wahrheitstabelle 94
Warenaufzug 102
Warenverwaltung 73
Warteschritte 218
Wartung 280
Wasserfahrt 278
Watchdogschaltung 263
Web-Browser 298
Webvisualisierung 45
Weiterschaltbedingung 152 f.
Weltkoordinaten 204
Werkzeugmagazin 90
Werkzeugmaschine 146, 197
White-Box-Tests 276
Wirkungslinien 81
Wirkungsweg 17

X

XML 55

Y

YS 47, 77

Z

Zähler 101
Zeitdiagramm 111
zeitredundant 263
Zentrifuge 182
Ziegler-Nichols 141
Zusatzlogik 81, 278
Zusatzzustände 176
Zustandscodierung 109
Zustandsermittlung 107
Zustandsgleichungen 99 f., 102, 104, 173
Zustandsgraf 110, 158
Zustandstabelle 108
Zustandsübergangstabelle 111
Zuverlässigkeit 259
Zweileiterschaltung 36
Zweipunktregler 131, 161
Zwei- und Vierleitertechnik 38
Zykluszeit 43, 58, 90, 104
Zylinderkolben 118